

Amund Hauglie-Hanssen
Sveinung Bergum

Software Defined Storage med IBM Spectrum Scale

Med fokus på effektivisering og
ressursbesparelse

Bacheloroppgave i Informatikk, Drift av Datasystemer

Veileder: Jostein Lund

Mai 2020

Amund Hauglie-Hanssen
Sveinung Bergum

Software Defined Storage med IBM Spectrum Scale

Med fokus på effektivisering og ressursbesparelse

Bacheloroppgave i Informatikk, Drift av Datasystemer
Veileder: Jostein Lund
Mai 2020

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Innhold

TITTEL	SIDE
ORDLISTE	8
FORSTUDIERAPPORT	10
HOVEDRAPPORT	39
SLUTTRAPPORT	129

Innledning

Dette dokumentet er sluttinnleveringen for bachelorprosjektet i Informatikk, Drift av datasystemer, våren 2020. Prosjektet omfatter lagringssystemet IBM Spectrum Scale, og er en oppgave gitt av Kongsberg Seatex i forbindelse med deres problemstilling rundt håndtering av Big Data i bedriften. De ønsker å finne en løsning på hvordan de skal håndtere de store datamengdene som samles opp av deres autonome båter.

Rapportene som ligger i dette dokumentet er forstudierapporten, hovedrapporten og sluttrapporten, som dekker hver sin fase i prosjektet. Som vedlegg til denne innleveringen kommer prosjekthåndboken som inneholder planer, logger, timelister og møterefater. I tillegg følger MS Project-filen, presentasjonen av prosjektet og timelistene for prosjektet med.

Sammendrag

Dagens IT-bedrifter utfordres av stadig større mengder data, ellers kjent som «Big Data». Dette kommer av at mer og mer av teknologien vår digitaliseres og samler informasjon om omgivelsene rundt seg. For at bedriftene skal kunne dra nytte av dataen og sette den i sammenheng må den lagres et sted i mellomtiden. Problemet er at lagring hverken er gratis eller logisk problemfritt. Big Data krever smarte og sikre lagringsløsninger som kan hjelpe til med å strukturere og sortere dataen riktig. Dette er essensielt for å holde kostnader, ytelseskrav og kompleksitet under kontroll. I tillegg har ofte større bedrifter mange systemer og enheter med ulikt operativsystem som skal kommunisere med hverandre, noe som krever en felles plattform som kan fungere som et samlepunkt.

I forstudierapporten ble Kongsberg Seatex sine problemstillinger rundt akkurat dette temaet belyst. Vi så her på hva bedriften kan gjøre for å håndtere de store datamengdene som kommer inn fra sin autonome flåte. Her kom vi frem til at lagringssystemet IBM Spectrum Scale kunne være et relevant alternativ for bedriften. I hovedrapporten har vi gått dypere inn på programmets funksjonalitet, og undersøkt om det har det som skal til for å dekke kravene til bedriften.

Filsystemer som er designet for strategisk styring av data, kalles for «Software Defined Storage», og IBM Spectrum Scale er et eksempel på dette. Her tar man i bruk et konsept som kalles «Information Lifecycle Management», som ser på hvordan data bør styres gjennom livssyklusen basert på dataens verdi. Ut ifra resultatet av slik styring, bestemmer man hvordan lagringssystemet skal analysere og behandle dataene som importeres. IBM Spectrum Scale innebærer funksjonalitet for å automatisk migrere filer til et lagringsmedium som passer for typen data, noe som både sparer kostnader og øker responstiden til filsystemet. Slik kan man bruke billige og trege medier for arkivering, mens man bruker raske og dyrere medier for hurtiglesing. Spectrum Scale er også svært skalerbart, har høy ytelse, og sikrer dataen mot tap og korrupsjon.

I bachelorprosjektet har vi utarbeidet en omfattende dokumentasjon om Spectrum Scale, som har til hensikt å gi leseren en innføring i hvordan systemet fungerer, hvilke oppgaver det kan løse, og hvordan det installeres og tas i bruk. Vi har også satt opp et testsystem i Azure i liten skala, for å prøve ut og å dokumentere programmet i praksis. Hensikten med å ta i bruk en løsning som Spectrum Scale over et tradisjonelt lineært filsystem, er at man sparer betydelige mengder ressurser på lagring av Big Data. Samtidig er systemet designet for å takle fremtidig skalering, og belastningene slike datamengder skaper.

Bachelorprosjektet skal gi Kongsberg Seatex verdi i form av innsikt i filsystemets funksjoner, og Spectrum Scales relevans i forhold til problemstillingen. Vi har igjennom arbeidet med dette prosjektet konkludert med at IBM Spectrum Scale er et sterkt alternativ for bedriftens utfordringer, og som dekker kravene som ble satt i begynnelsen av prosjektet. Likevel må bedriften selv avgjøre om nytten av et slikt system veier opp for kostnadene, og hvordan dette kan forsvares økonomisk sett. Spectrum Scale krever innkjøp av en del maskinvare og programvare, og da vi ikke har oversikt over prisen på dette utstyret, kan vi ikke si mer om virkelig nytte bedriften vil ha av dette.

Ordliste

Denne seksjonen beskriver ord og begreper som kan være ukjent for leseren, og kan være hendig som oppslagsverk ved tvil om meningen bak noe som blir beskrevet i rapporten.

<i>Begreper</i>	<i>forklaring</i>
ACL	Access Control list. Liste med brukertillatelser for filsystemer
AFM	Active File Management. En Spectrum Scale funksjon. Samarbeid mellom clustere på ulike lokasjoner.
AWS	Amazon Web Services. Skytjenesten til Amazon
Azure	Skytjenesten til Microsoft
Big data	Begrep som beskriver datasett som er svært store eller komplekse å behandle.
Buffering	Midlertidig lagring av data i en I/O prosess.
CES	Cluster Export Services. Funksjon i Spectrum Scale for dataoverføring
Cluster	En samling av datamaskiner som samarbeider i et nettverk.
CN	Common Name. En del av et DN som brukes i katalogtjenester som LDAP
DC	Domain Component. En del av et DN som brukes i katalogtjenester som LDAP
DN	Distinguished Name. Unikt navn som brukes for identifisering av brukere i katalogtjenester som LDAP. En kommaseparert setning som består av CN DC og OU.
GPFS	General Parallel File System. Filsystemet som brukes av IBM Spectrum Scale
GUI	Graphical User Interface. Begrep for programmer som har en visuell plattform for brukere til å benytte programmets funksjoner
HA	High availability. Står for høy oppetid. Datasystemet kommer raskt på banen etter feil.
Hadoop	Open-source rammeverk fra Apache for distribuert lagring
HDD	Hard Disk Drive, Disk for lagring av data, hvor dataen er lagret på magnetiske skiver som leses av en mekanisk arm.
I/O	Input/output. Beskrivelse av noe som sender og/eller mottar data.
IBM	International Business Machines. Teknologiselskap som leverer blant annet Spectrum Scale.
IBM Spectrum Scale	Distribuert lagringssystem fra IBM. En del av IBM Spectrum Storage pakken.
IDI	Institutt for datateknologi og informatikk
ILM	Information Lifecycle Management. Hvordan data håndteres gjennom livsløpet.
IT	Informasjons Teknologi
Kerberos	Protokoll for autentisering og sikker dataoverføring

<i>LDAP</i>	Lightweight Directory Access Protocol. Protokoll for tilgangsstyring av maskiner og brukere
<i>LTFs</i>	Linear Tape File System. Filsystem som brukes for tape
<i>LTO</i>	Linear Tape-Open. Standard for magnetbånd / tape
<i>LUN</i>	Logical Unit Number. Representerer en lagringsenhet
<i>NAS</i>	Network-Attached Storage. Nettverksmontert lagringsenhet
<i>NFS</i>	Network File System. Protokoll for deling av lagring over nettverket
<i>Node</i>	En datamaskin som representerer en funksjon eller en oppgave i et cluster
<i>NSD</i>	Network Shared Disk. Disk som deles av en node i clusteret
<i>NSD-server</i>	Node som deler en NSD med de andre nodene
<i>Object</i>	Brukt i sammenheng med CES. Protokoll for dataoverføring mot enkelte skytjenester
<i>OU</i>	Organizational Unit. En del av et DN som brukes i katalogtjenester som LDAP
<i>PAM</i>	Pluggable Authentication Module. Brukes på klientmaskiner for å konfigurere pålogging mot LDAP
<i>Policy</i>	En definert regel
<i>Quorum</i>	Minimum antall noder som må kommunisere for drift av Spectrum Scale
<i>RAID</i>	Redundant Array of Inexpensive Disks. Teknologi for redundant lagring av data over flere disk
<i>SAN</i>	Storage Area Network. Nettverk av ulike lagringsenheter, eksempelvis et nettverk av flere NAS
<i>SDS</i>	Software Defined Storage. Lagring som styres av programvare
<i>Single point of failure</i>	Et enkelt ledd som resten av et datasystem avhenger av
<i>SMB</i>	Server Message Block. Protokoll for dataoverføring
<i>SSD</i>	Solid State Disk. Disk for lagring av data uten bevegelige deler, hvor dataen lagres elektronisk med flashminne
<i>SSH</i>	Secure Shell. Protokoll for sikker dataoverføring
<i>SSL</i>	Secure Sockets Layer. Protokoll for kryptert dataoverføring
<i>Storage pool</i>	Definert område for lagring med samme egenskaper
<i>Tape</i>	Medium for lagring av data. Tape har magnetiske spor som optiske sensorer bruker for å lese av. Krever ikke strøm for lagring.
<i>VPN</i>	Virtual Private Network. Protokoll for kryptert dataoverføring. VPN lager broer over flere ulike nettverk som gjør at fysisk adskilte maskiner tror de er i samme nettverk



Software Defined Storage med IBM Spectrum Scale

Med fokus på effektivisering og ressursbesparelse

Forstudierapport

For bachelorprosjektet v2020 i Drift av Datasystemer

Amund Hauglie-Hanssen, Sveinung Bergum
15.05.2020

Innhold

Figurliste.....	2
Tabelliste.....	2
Revisjonshistorie.....	3
1. Introduksjon.....	4
2. Bakgrunn for prosjektet.....	4
2.1 Dagens systemer og rutiner.....	4
2.2 Hvorfor IBM Spectrum Scale?.....	5
3. Prosjekt mål.....	7
3.1 Krav til systemet:.....	8
3.2 Resultatmål.....	8
3.3 Effektmål.....	8
3.4 Prosessmål.....	8
4. Prosjektets omfang.....	9
4.1 Hva bachelorprosjektet <i>skal</i> gjøre:.....	9
4.2 Hva bachelorprosjektet <i>ikke</i> skal gjøre:.....	9
5. Interessentanalyse.....	11
6. Rammebetingelser.....	12
7. Kritiske suksessfaktorer.....	13
7.1 Bachelorprosjekt.....	13
7.2 Reell gjennomføring.....	13
8. Risikoanalyse.....	14
9. Kost/nytte-analyse.....	18
9.1 Kvantifiserbar nytte.....	18
9.2 Ikke-kvantifiserbar nytte.....	18
9.3 Bortfall av direkte kostnader.....	19
9.3.1 Bortfall av direkte kostnader i prosjektoppgaven.....	19
9.3.2 Bortfall av kostnader i reell gjennomføring.....	19
9.4 Estimerte kostnader.....	19
9.5 Sammenstilling kost/nytte.....	20
10. Prosjektorganisering og ansvarsområder.....	23
11. Retningslinjer og standarder.....	24
11.1 Dokumentasjon.....	24

11.2 Ukentlig planlegging or rapportering.....	24
11.3 Møter med veileder	25
11.4 Møter med oppdragsgiver	25
11.5 Prosjektplanlegging i MS Project	25
11.6 Timeføring.....	25
11.7 Informasjon om bruk av kilder.....	25
12. Aktivitets- og ressursplan.....	26
12.1 Plan for hele prosjektet.....	26
13. Konklusjon og anbefalinger om videre arbeid	27
Kilder	28

Figurliste

Figur 1 - Dagens løsning	5
Figur 2 - Systemtrening med data.....	7
Figur 3 - Abstrakt illustrasjon av arkiveringssystemet	10
Figur 4 - Risikokart	17
Figur 5 - Kostnadsbesparelse tapeløsning	21
Figur 6 - Kostnadsfordeling tapeløsning	22
Figur 7 - Prosjektorganisasjon.....	23
Figur 8 - Utkast av Projectfil.....	26

Tabelliste

Tabell 1 - Revisjonshistorie	3
Tabell 2 - Sannsynlighet og konsekvens.....	14
Tabell 3 - Sammenstilling kost/nytte for et reelt prosjekt.....	20
Tabell 4 - Oversikt kost/nytte bachelorprosjekt	21

Revisjonshistorie

Tabell 1 - Revisjonshistorie

Dato	Utgave	Kommentar
05.02.2020	1.0	Første fullstendige utgave av forstudierapporten. Klargjøring for vurdering av oppdragsgiver og veileder.
28.02.2020	2.0	Revisjon i forhold til oppdatert prosjektfokus og rettelser i forhold til designrapport
02.04.2020	3.0	Siste utgave
24.04.2020	3.1	Finpuss
15.05.2020	3.2	Klar for levering

1. Introduksjon

Dette er et dokument som skal opplyse leseren om omfanget til bachelorprosjektet om «Software Defined Storage» med IBM Spectrum Scale gitt av Kongsberg Seatex. Gjennom forstudierapporten utarbeider vi i prosjektgruppe 115 prosjektets entydige problemstilling, mål knyttet til prosjektet og de rammer som skal gjelde. Dokumentet beskriver prosjektets utgangspunkt på et overordnet nivå. Hensikten er å gjennomføre en innledende undersøkelse rundt prosjektets levedyktighet for å avgjøre om det er verdt å gjennomføre. Dette vil innebære ulike analyser, slik som kost/nytte-analyse, risikoanalyse og interessentanalyse. I tillegg setter vi standarden for dokumentasjon og prosessgjennomføring.

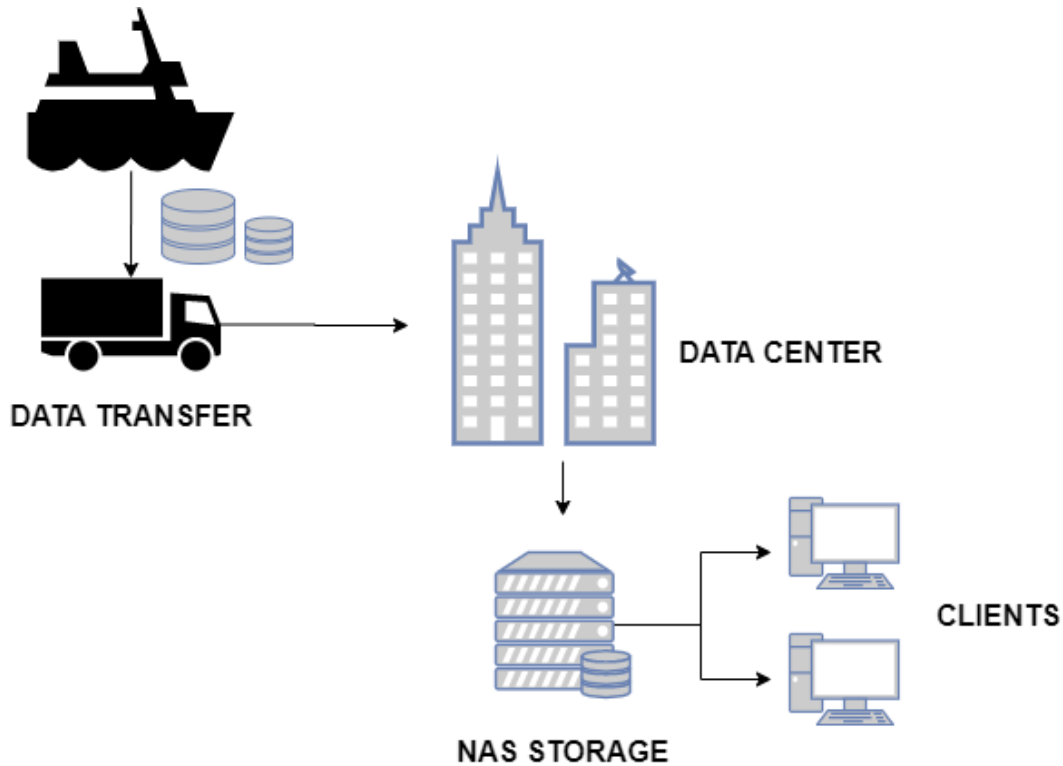
2. Bakgrunn for prosjektet

Kongsberg Seatex er et datterselskap av Kongsberg Maritime som utvikler, produserer og selger maritimt utstyr på verdensbasis. Bedriften benytter blant annet autonome skip til forskning og utvikling. Disse selvkjørende skipene samler store mengder data når de er i drift. Dataene samles opp, arkiveres og gjøres tilgjengelig for videre analyse. Utfordringen med dette er mengden data som fartøyene produserer; det er her snakk om mellom en til to terrabyte med data per skip på en dag. Disse datamengdene er for store til at det er hensiktsmessig å bruke en ren skyløsning for å håndtere dem, da dette blir dyrt og ressurskrevende. Kongsberg Seatex ønsker seg heller derfor et sikkert system for å sentralisere og lagre denne dataen. Deler av dataen skal ligge tilgjengelig for hurtiglesing basert på innhold, slik at utviklere raskere kan hente ut dataen som er interessant, mens resten arkiveres på et tregere medium.

I denne oppgaven har vi derfor fått i oppdrag å se på løsninger som kan effektivisere og forenkle lagringen av filer, med hensikten å gjøre systemet mer fleksibelt og oversiktlig, samtidig som at analysen av filene blir raskere og enklere for utviklere. Oppdragsgiver har foreslått IBM Spectrum Scale som utgangspunkt for denne nye løsningen. Programmet er designet for å løse problemstillinger knyttet til lagring av store mengder data på flere ulike medium, slik som flaskehals i ytelsen, kostnadseffektiv skalering av filsystemet, og generell oversiktighet med sammenheng mellom enheter. Filhierarkiet som brukeren ser vises uavhengig av hvordan dataen er lagret, så brukerne trenger ikke å forholde seg til fysisk filplassering. Vi tror at Spectrum Scale er et godt alternativ for å løse Kongsbergs Seatex sin problemstilling. Derfor vil vi i denne rapporten vurdere dette systemet i forhold til bedriftens krav og behov.

2.1 Dagens systemer og rutiner

I dagens system (se figur 1) samles data fra de autonome fartøyene på disk i et Network Attached Storage (NAS), som senere fraktes over til datasenteret. Et NAS er kort forklart en enkel diskløsning som kobles på nettet, og kan nås av andre maskiner som har tilgang til dette nettverket. På datasenteret lastes filene over til et annet NAS ved behov, noe som tar lang tid. Her arkiveres filene hierarkisk basert på dato. Dagens måte å lagre data på er lite skalerbar, uoversiktlig og lite optimalisert. Dette er fordi NAS lagrer all data på samme sted og på samme type medium, noe som raskt fører til at tilgangen til filsystemet til en flaskehals. Kostnadene ved en slik løsning vil stige lineært med kravet til kapasitet. I tillegg gir NASet oss et «single point of failure», noe som hverken er ønskelig i forhold til beskyttelse mot tap eller for høy opptid og tilgjengelighet av data.



Figur 1 - Dagens løsning

2.2 Hvorfor IBM Spectrum Scale?

Problemstillingen i dette prosjektet er igjen hovedsakelig mengden data som produseres og skal lagres. Hvert fartøy produserer mellom 1 og 2 terrabyte med data i døgnet. Lagres all data på disker, enten "on premise" eller i skyen kommer systemet til å bli for dyrt til det at lønner seg å bruke, og kompleksiteten øker over tid. Et av kravene til systemet er dermed å bruke magnetiske bånd for å lagre deler av dataene. Magnetiske bånd, eller tape, er mye billigere enn disker og kan lagre store mengder data per enhet. Skal man lagre flere petabytes med data, vil tape være et valg som gir mer mening økonomisk sett. Slike kassetter er robuste og kan lagres uten behov for strøm over lang tid. Ulempen er at de er mye tregere enn disker, og derfor er prioritering av data viktig. For å ha et fungerende system må det lages en hybridløsning hvor magnetiske bånd blir brukt for langtidslagring av data, mens data som blir eller skal bli behandlet er tilgjengelig på disker.

Her kommer IBM Spectrum Scale inn i bildet. Informasjonen her er et utdrag fra dokumentasjonen om Spectrum Scale sine nettsider (IBM 2020a), og brukermanualen om Spectrum Scale (Quintero et al, 2019). Spectrum Scale er et intelligent, fleksibelt, parallelt filsystem med mange funksjoner for å dekke ulike behov og konfigurasjoner. Programmet installeres på en eller flere noder i et cluster, altså maskiner som samarbeider, som sammen sørger for høy oppetid og tilgjengelighet. Disse nodene kommuniserer med hverandre over et høyhastighetsnettverk. Applikasjonene som kommuniserer mot filsystemet sender forespørsler til nodene som er koblet mot lagringsmediet om å få tilgang til data.

Lagringsmediet er typisk et Storage Area Network (SAN) eller en annen form for redundant løsning. Programmet er en del av programvare-suiten IBM Spectrum Storage, som inneholder flere alternativer for Software Defined Storage (SDS).

Dokumentasjonen for Spectrum Scale er omfattende, og systemet kan konfigureres på mange forskjellige vis. Noen av hovedfunksjonene som kan nevnes er:

- Skalerbart enterprise-programvare som kan brukes på standard maskinvare
- Høyhastighets parallelt filsystem som kan etableres på kryss av geografiske lokasjoner
- Enorm kapasitet for lagring av data
- Bredt spekter av konfigurasjoner for å møte bedriftsbehov
- Konfigurasjonen kan gå fra å være veldig enkel til veldig kompleks avhengig av behov
- Høy grad av kompatibilitet med operativsystemer og protokoller samler ulike enheter på samme plattform
- Global tilgang til filsystemet, selv med dårlig internett-tilkobling
- Live-migrering av filsystemet mellom ulike clustre legger til rette for høy ytelse, redundans og backup
- Innebygde funksjoner for høy oppetid, failover og kryptering

Filsystemet er designet slik at det sprer data systematisk over flere disker i RAID. Programvaren legger til rette for flere funksjoner, slik som HA, høy I/O-båndbredde, god skalerbarhet og sikker lagring. Ved å gruppere disker med ulik pris og hastighet vil Spectrum Scale kunne flytte dataene dit de trengs etter behov, noe som sparer kostnader ved unødvendig bruk av dyr lagring. Effekten av dette vil også være at man ikke trenger å grue seg for å bruke lagring med høy ytelse, da man kun trenger å gjøre dette der det er behov for det. Ved å definere egne regler for dataen man importerer i filsystemet vil programmet kunne behandle dataen automatisk for å optimalisere for både ytelse og kostnadsbesparelse.

Når vi ser på IBM Spectrum Scale kontra dagens NAS er det mange grunner til å velge Spectrum Scale:

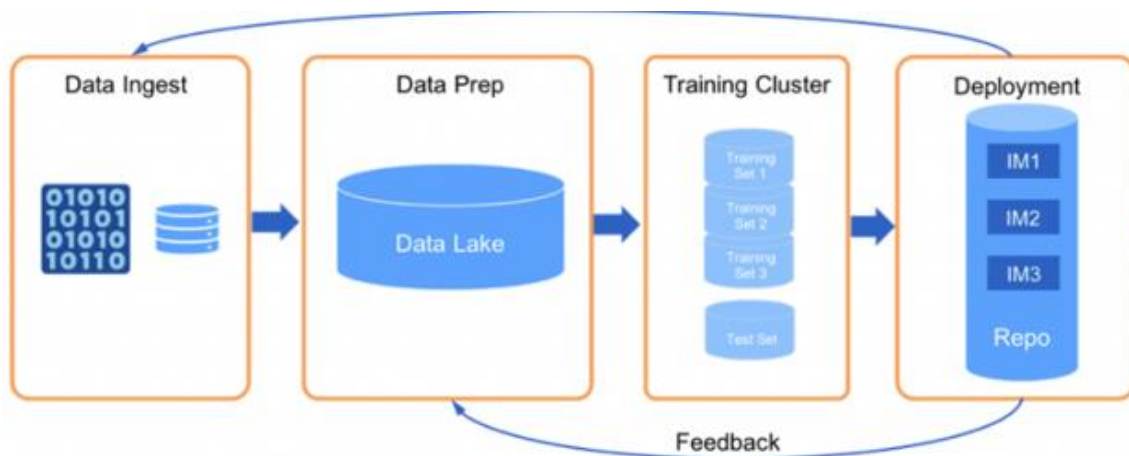
- Spectrum Scale er et parallelt filsystem designet med høy ytelse og skalerbarhet for Big Data og AI-workload i tankene. Et NAS er derimot et lineært filsystem der ytelse raskt begrenses ved mange forespørsler og høye krav til lese- og skrivehastigheter.
- Filsystemet har virtuelt sett ubegrenset med lagringskapasitet.
- Lagring av data er policy-basert, og skjer på den type lagringsmedium som er hensiktsmessig, og sparer derfor kostnader. Spectrum Scale kan defineres til å kun ta vare på de data man trenger, og kaste resten. Et NAS er derimot en enkel lagringsenhet som kun er designet for å lagre data.
- Løsningen er multifunksjonell, mens NAS kun har en enkelt funksjon.
- Overlegen kapasitet for båndbredde, oppetid og redundans. Et NAS vil derimot være et «single point of failure» ved uhell og feil.
- Spectrum Scale kan spres over ulike geografiske lokasjoner der spesifiserte datasett sendes over og synkroniseres automatisk, noe som maskerer forsinkelser i nettverket.

3. Prosjektmål

I dette prosjektet skal vi utforske nærmere om IBM Spectrum Scale løser problemstillingene til Kongsberg Seatex. Vi vil se på hvordan systemet støtter oppunder prosessen for behandling av data.

Proessen med å behandle og analysere store mengder data går over 4 trinn:

1. Innsamling og lagring av data
2. Klargjøring og tilpasning av data fra systemene
3. Analyse av data
4. Trene opp systemene ut fra analysen av data



Figur 2 - Systemtrening med data

Å preparere dataene (trinn 2 på figuren) er det mest ressurskrevende trinnet i prosessen, og det er dette som er hovedoppgaven til Spectrum Scale. Systemet som lagrer data, blir også styrende for de andre trinnene. Effektiv og godt planlagt preparasjon av data vil spare en virksomhet for tid og penger, og kan lesse de andre trinnene for unødvendig belastning. I relasjon til dette er det viktig med god bruk av Information Lifecycle Management (ILM), som vi skal komme tilbake til senere. Dette er viktig for å holde oversikt over dataene som bedriften har til enhver tid, noe som blant annet kan forhindre tap av kritisk informasjon.

Igjennom bachelorprosjektet skal vi undersøke og dokumentere komponentene og funksjonene i IBM Spectrum Scale, og sette systemet i sammenheng med bedriftens behov. Som en del av denne oppgaven har vi dermed besluttet å sette opp og teste et lagringssystem basert på denne programvaren, med de funksjoner som er viktigst for bedriftens scenario. Prosjektet har til hensikt å presentere IBM Spectrum Scale på en måte som kan være til hjelp når oppdragsgiver skal ta en endelig beslutning om hva slags system de skal benytte. Oppdragsgiveren må videre vurdere om produktets nytte veier opp for kostnad og kompleksitet.

Krav til systemet og effektmål definerer vi for en reell installasjon av systemet for Kongsberg Seatex. Vår oppgave med prosjektet er å dokumentere at programmet kan dekke disse målene. Resultatmål og prosessmål er mål vi har satt i forhold til utviklingen av vårt eget bachelorprosjekt.

3.1 Krav til systemet:

- Skal være skalerbart for store mengder data
- Søking i arkiver skal være raskt
- Data skal sorteres, plasseres og kopieres der det er hensiktsmessig basert på policyer for ressursbesparing
- Data skal være lett tilgjengelig for utviklere og brukere
- Deler av dataen skal kunne hurtigleses

3.2 Resultatmål

Resultatmålene beskriver konkret hva som skal gjøres i bachelorprosjektet.

- Dokumentere komponentene i IBM Spectrum Scale, samt løsningens bruksområder og funksjonalitet
- Dokumentere hvordan systemet kan brukes i sammenheng med oppdragsgiverens behov
- Installere et IBM Spectrum Scale cluster og teste funksjonene
- Konfigurere policyer i IBM Spectrum Scale til å sortere og arkivere data
- Vise at filsystemet er tilgjengelig for alle noder i nettverket

3.3 Effektmål

Effektmålene til prosjektet beskriver hvilken virkning systemet skal få for Kongsberg Seatex

- Øke verdien i produsert data
- Kutte ned tid og ressurser brukt på databehandling
- Forenkle arbeidsoppgaver knyttet til databehandling, og gjøre jobben mer oversiktlig
- Spare kostnader på lagringsmedier
- Verdien av produsert data må være større enn kostnadene til systemet

3.4 Prosessmål

Prosessmålene tar for seg resultatene som skal fremkomme av selve prosessen

- Øke kunnskap om problemstillinger knyttet til lagring av store datamengder
- Undersøke hvordan IBM Spectrum Scale kan brukes for å løse en Big Data-problemstilling
- Lære om hvordan et større prosjekt gjennomføres
- Danne erfaring om ulike faser i prosjektet som forstudie, design, gjennomføring og avslutning
- Lære om hvordan å jobbe effektivt i par
- Få erfaring om møtekultur og kommunikasjon med andre ansatte

4. Prosjektets omfang

Systemet skal forenkle og effektivisere analyse og behandling av data, og legge til rette for sparing av ressurser. Verdien som ligger i dataen som blir produsert kommer ikke til nytte hvis den ikke er lett tilgjengelig. For brukerne som skal analysere data ønskes det et system som gjør all data tilgjengelig som individuelle filer under ett hierarki, uten at de trenger å vite noe om hvordan filene er lagret og behandlet.

Å håndtere flere terrabyte med data om gangen på fornuftig vis, særlig når man skal finne tilbake til de rette filene i etterkant, krever god planlegging og et skalerbart, raskt og oversiktlig datasystem. Et eksempel på en slik løsning er IBM Spectrum Scale. Dette systemet er designet for å håndtere store mengder data med ulik prioritet på en sikker og stabil måte. Systemet klassifiserer og flytter dataen automatisk etter viktighet basert på policyer, og tar av seg mange av utfordringene som kommer med å håndtere slike datamengder. Uavhengig av lagringsmedium vil filsystemet fremstå som ett filsystem, noe som gjør at sluttbrukeren ikke trenger å ta hensyn til hva som skjer på lavere nivåer. Dette muliggjør høy grad av kostnadseffektivitet fordi filene vil distribueres til et lagringsmedium som samsvarer med prioritet og krav om tilgjengelighet.

4.1 Hva bachelorprosjektet *skal* gjøre:

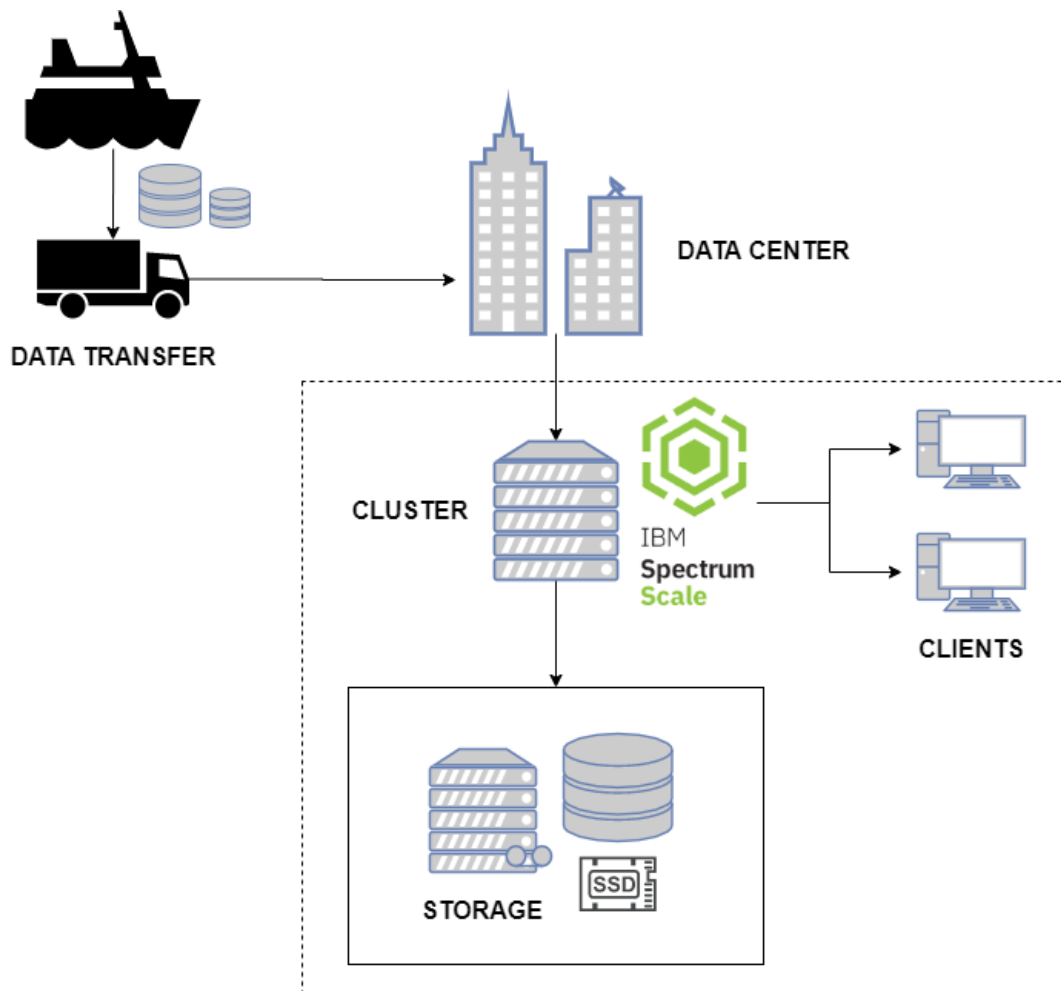
- Ta utgangspunkt i hvilken nytte arbeidet kan ha for bedriften
- Undersøke og vurdere IBM Spectrum Scale som tjeneste i forhold til problemstillingen
- Ta for seg installasjon, testing og vurdering av IBM Spectrum Scale som lagringsløsning tiltenkt Kongsberg Seatex sine autonome båter
- Sette sammen en fungerende prototype som presenteres med pilotprosjektet
- Dokumentere designet og installasjonen av testsystemet
- Demonstrere programvarens funksjonalitet
- Produsere innsikt og veiledning som Kongsberg Seatex eventuelt kan bruke i fremtidig implementasjon av en slik løsning

4.2 Hva bachelorprosjektet *ikke* skal gjøre:

- Installere systemet på, eller jobbe direkte mot Kongsberg Seatex sitt utstyr
- Håndtere faktiske data som tilhører bedriften
- Ta høyde for infrastruktur, slik som aktivitet på båten, i mellomledd og under transport av disker og liknende
- Ta høyde for kostnader knyttet til områder utover det som angår bachelorprosjektet
- Ta i bruk funksjoner som ikke ansees for å være nødvendige for oppgaven

Oppsummert baserer prosjektet seg på å designe og utvikle en prototype for å demonstrere hvilke muligheter og fordeler Kongsberg Seatex kan oppnå ved å oppgradere sitt lagringssystem. Løsningen forholder seg til og tar utgangspunkt i en abstrakt modell av komponentene, og fungerer som en selvstendig prototype. Prosjektet har med andre ord ingen tilknytning til Kongsbergs faktiske systemer, og vi vil heller ikke vurdere disse direkte da vi ikke har tilgang til slik informasjon. Prosjektet vil videre kunne brukes av Kongsberg Seatex for å vurdere nytten av å gjennomføre et reelt prosjekt.

Under har vi laget en abstrakt illustrasjon over hvordan vi ser for oss det nye systemets oppbygning. Kort forklart fraktes data på tape fra båten til et datasenter, og plasseres inn i filsystemet. Deretter håndteres de nye dataene av Spectrum Scale som analyserer dataene og fordeler dem til rett lagringsmedium. Overføringen bør skje så raskt som mulig slik at tape-kassetten kan transporteres tilbake for gjenbruk.



Figur 3 - Abstrakt illustrasjon av arkiveringssystemet

5. Interessentanalyse

Interessentene er partene som berøres av bachelorprosjektet. Analysen beskriver den forventede effekten prosjektet vil ha på de ulike interessentene.

(Eksterne) Representanter fra Kongsberg Seatex:

- Prosjekteier Henrik Foss
 - Beskrivelse
 - Eier prosjektet. Han kommer med krav til systemet og bistår med hjelp og kunnskap om løsningen, og stiller problemstillingen til disposisjon for å utforske mulighetene med systemet.
 - Konsekvens
 - Får større innsikt i IBM Spectrum Scale som løsning i forhold til Kongsberg Seatex sine systemer
 - Sparer forsknings- og utviklingskostnader
- Brukere
 - Beskrivelse
 - Sluttbrukere av systemet som skal dra nytte av dataene som lagres. Med andre ord er dette utviklere som primært skal lese og skrive til filsystemet.
 - Konsekvens
 - Kan bruke rapporten som veiledning
 - Får en oversikt over systemets funksjonalitet
- IT-administrasjon
 - Beskrivelse
 - Ansvarlige for drift av et slikt system. Skal ha kunnskap om rutiner og drift av systemet.
 - Konsekvens
 - Får en grunnleggende innføring i både det tekniske og det abstrakte ved Spectrum Scale.
 - Driftsjobben blir enklere og mer oversiktlig ved bruk av dokumentasjonen

(Interne) Representanter fra NTNU:

- Prosjektgruppe (Amund Hauglie-Hanssen og Sveinung Bergum)
 - Beskrivelse
 - Ansvarlige for gjennomføring og kvalitet i prosjektet. Prosjektet er både en avsluttende oppgave for å vise tilegnet kompetanse fra universitet, og for å fordype seg innen et relevant og spennende fagområde. Utvikler prosjektets system og dokumenterer prosessen.
 - Konsekvens
 - Tilegner seg kunnskap om parallelle filsystemer, og mer spesifikt om lagringssystemet IBM Spectrum Scale
 - Får erfaring innen prosjektarbeid
- Veileder Jostein Lund
 - Beskrivelse
 - Bistår med erfaring og kunnskap fra tidligere studentprosjekter. Setter rammer for prosjektet og er med å vurdere den endelige kvaliteten på prosjektet.

6. Rammebetingelser

Rammebetingelsene er konkrete krav som gjelder for bachelorprosjektet, og som prosjektgruppen må holde seg innenfor.

Starttidspunkt:

Offisiell oppstartsdato er 13.01.2020.

Prosjektets frist:

Vi har satt prosjektets frist til å være 16.05.2020. Da skal alt av rapporter og materiale være klart for innlevering, og pilotprosjektet klart for presentasjon. Innleveringsdato satt av instituttet på Inspira er 20.05.2020 klokken 14.00. Videre vil presentasjonen av prosjektet avholdes i løpet av uken etter.

Arbeidstimer:

Standarden for antall timer brukt til bacheloroppgaven er 500 timer per person eller cirka 14 arbeidsuker, med et slingringsmonn på 5%. Dette skal dokumenteres i timelister og i prosjekthåndboken.

Kostnader:

Prosjektet vil basere seg på å benytte ressurser og utstyr som allerede er tilgjengelig eller kan tilbys av NTNU og gratis-/studentlisenser til programvare. Der dette ikke er mulig blir vi nødt til å finne andre alternativer. Vi vil ikke se på alle kostnader for en faktisk implementasjon av systemet, men gjøre kost/nytte analysen basert på konsekvensene av bachelorprosjektet, og for områdene der vi får tak i konkrete tall. Hensikten med kost/nytte-analysen er å illustrere hvordan systemet kan spare bedriften for kostnader.

Myndighetskrav:

Oppgaven er tilrettelagt av oppdragsgiver slik at det ikke skal bli konflikt i forhold til opphavsrett. I tillegg vil vi ikke få tilgang til informasjon som vil være av betydning i forhold til taushetsplikt.

Dokumentasjon og fremføring:

Oppgaven skal dokumenteres i form av rapporter som beskriver prosjektets fremgang. Når det nærmer seg innleveringsdato arrangeres en fremføring og demonstrasjon av pilotprosjektet. Mer om dette i kapitlet om retningslinjer og standarder.

Bruk av lisenser og utstyr:

Under gjennomføringen av testing, design og oppsett av bacheloroppgaven vil vi benytte oss av gratislisenser for testmiljø og lisenser for skytjenester knyttet til NTNU. Vi vil også benytte oss av servere og datautstyr som instituttet tilbyr for installasjon.

7. Kritiske suksessfaktorer

7.1 Bachelorprosjekt

Disse kritiske suksessfaktorene tar utgangspunkt i kriteriene vi setter for at bachelorprosjektet skal anses som vellykket.

De kritiske suksesskriteriene er:

- Veldokumentert vurdering av IBM Spectrum Scale for Kongsberg Seatex. Prosjektgruppen klarer å formidle bruksområdene og nytteverdien til en slik løsning
- Bedriftens ansatte finner rapporten nyttig i vurderingen av et slikt system
- Vi klarer å installere testsystemet som skal være velfungerende og skalerbart, og demonstrere funksjonaliteten til systemet
- Prosjektgruppen holder seg innenfor gitte rammebetingelser

7.2 Reell gjennomføring

Videre seg vi på de overordnede kritiske suksessfaktorene som gjelder for en virkelig innføring av IBM Spectrum Scale.

De kritiske suksesskriteriene er:

- Ivareta krav til informasjonsforvaltning
- Systemets effektivitet må være målbart forbedret
- Kostnader per byte som lagres skal være betydelig lavere
- Administrasjon, drift og bruk av data skal være oversiktlig og enkelt
- Prosjektet må holde seg innenfor gitte rammebetingelser

8. Risikoanalyse

Risikoanalysen er til for å øke sannsynligheten for å lykkes med prosjektet ved å forutse og drøfte utfordringer og problemer som kan oppstå underveis i prosjektet. For de ulike utfordringene lager vi en handlingsplan for hva vi skal gjøre for å håndtere risikoen. En risiko beskriver vi som et produkt av konsekvens og sannsynlighet for at en uønsket hendelse inntreffer. Konsekvens og sannsynlighet har en skala hver, fra en til fem, og produktet til risikoen kommer da på en skala fra en til 25. Sannsynlighet er beskrevet på en skala fra lav til høy. Lav sannsynlighet betyr at hendelsen sannsynligvis ikke inntreffer i løpet av prosjektet, mens høy sannsynlighet betyr at det sannsynligvis skjer. Konsekvens beskriver i hvor stor grad hendelsen påvirker kvalitet, mål, ambisjoner og ressursbruk på gjennomføringen.

Når det kommer til tiltakene for en risiko, kan man håndtere dette på ulike måter. Det første man kan gjøre er å etablere tiltak for å redusere sannsynligheten for at noe skjer. Videre kan man forsøke å redusere omfanget til konsekvensen om det skulle skje. Et annet alternativ er å overføre risikoen til andre parter ved å endre kontraktsvilkårene, eller eliminere risikoen fullstendig ved å endre på rammebetingelsene til prosjektet.

Sannsynlighet	Forekommer i prosjektet
1	Lav sannsynlighet
2	Noe sannsynlighet
3	Middels sannsynlighet
4	Moderat sannsynlighet
5	Høy sannsynlighet

Konsekvens	Påvirkning av gjennomføringen
1	Lav påvirkning av gjennomføringen
2	Noe påvirkning
3	Middels påvirkning
4	Høy påvirkning
5	Vil forhindre gjennomføring

Tabell 2 - Sannsynlighet og konsekvens

1. Sykdom eller skader blant nøkkelpersoner

Sannsynlighet: 4

Vi kan regne med at minst en av personene som er involvert i viktige deler av prosjektet blir syk i løpet av prosjektets gang, og at dette vil få en viss påvirkning på prosjektet.

Konsekvens: 2

Mest sannsynlig er det ikke snakk om alvorlig sykdom dersom noen av deltakerne blir syke. Det kan likevel ha en viss effekt på effektiviteten og eventuelt føre til dager uten jobbing fra deler av gruppen. Vi setter derfor konsekvensen til 2.

Risikokarakter: 8

Tiltak:

Det er ikke mye man kan gjøre for å unngå å bli syk. Tiltak blir derfor å etablere rutiner for å videreføre prosjektet i best mulig grad om noen blir midlertidig fraværende. Det er mulig å søke om utsettelse av prosjektet. Eventuelt må en av deltakerne kompensere for den andre med tanke på arbeidsmengde.

2. Manglende ressurser og lisenser

Sannsynlighet: 3

Sannsynligheten for at vi støter på utfordringer i forbindelse med ressurser vi enten ikke har eller får tilgang til i løpet av prosjektet er relativt stor. Dette er med tanke på at vi ikke er fullstendig klare over omfanget til prosjektet og hvilke ressurser systemet krever.

Konsekvens: 3

Det er usikkert hva slags konsekvens det får for prosjektet om det mangler ressurser. Dette bestemmes naturligvis av hvor essensiell ressursen er for å gjennomføre prosjektet. Vi setter konsekvensen til 3 fordi vi ser for oss at et slikt problem kan være relativt omfattende.

Risikokarakter: 9

Tiltak:

Tiltaket for å redusere denne risikoen blir å gjøre godt forarbeid og å starte tidlig med å kartlegge alle ressursene som kreves for hver arbeidsoppgave. Dette er oppgaver vi tar fatt i når vi kommer til designfasen der vi ser på systemets design og spesifikasjoner.

3. Feil definering av mål eller omfang

Sannsynlighet: 4

Vi ser på det som en mulighet at vi oppdager at definisjonen av prosjektet enten er problematisk eller at vi over-/underestimerer arbeidsmengden i prosjektet. Vi kan støte på utfordringer som følge av valgt arbeidsområde. Da vi aldri tidligere har arbeidet med IBM Spectrum Scale og er usikre på hva som kreves for å ta det i bruk, er det ganske stor sannsynlighet for at vi kommer til å enten overestimere eller underestimere arbeidsmengden som kreves for å få det opp og gå

Konsekvens: 4

Om vi underestimerer arbeidsmengden, kan dette få relativt store konsekvenser. Dette vil kunne føre til lange arbeidsdager, forskjøvet tidsplan og i verstefall at vi ikke kommer i mål med det vi hadde planlagt. Dette kan føre til at rammebetingelser og prosjektmål må justeres eller sløyfes.

Risikokarakter: 16

Tiltak:

Her vil en kombinasjon av godt forarbeid og effektiv jobbing kunne redusere risikoen. Å sette oss godt inn i dokumentasjonen til systemet for å forstå prinsippene og kravene til systemet, vil bedre evnen vår til å vurdere omfanget av oppgaven. Dette vil hjelpe oss med å fastslå realistiske mål i begynnelsen av prosjektet. Å komme raskt i gang med designfasen slik at vi ikke slurver med tiden vil også bedre sannsynligheten for at vi oppdager uforutsette utfordringer og kan revurdere omfanget tidlig nok.

4. Dårlig kommunikasjon mellom interessenter

Sannsynlighet: 2

Det kan hende at det oppstår misforståelser i kommunikasjonen mellom deltakere i prosjektet, både internt i prosjektgruppen, men også eksternt mellom prosjektgruppe og veiledere.

Konsekvens: 4

Misforståelser av prosjektdefinisjon og status vil kunne styre prosjektet i en negativ retning, der arbeid eventuelt må gjøres på nytt, eller mål og betingelser må endres. Dette avhenger av graden av misforståelse. Dersom misforståelsen er stor rundt essensielle faktorer av prosjektet, slik som oppgavedefinisjonen, vil konsekvensen være stor for gjennomføringen.

Risikokarakter: 8**Tiltak:**

Jevnlige møter og felles gjennomgang av agenda, sakslister og referater vil hjelpe til med å gjøre at interessentene har en felles forståelse for oppgavens status. Man bør heller gjenta beskjeder en gang for mye enn å satse på at alle er innforstått med det som kommuniseres.

5. Dårlig samarbeid i prosjektgruppa

Sannsynlighet: 2

Igjennom prosjektet kommer vi sannsynligvis til å møte på problemstillinger vi er uenige om. Her vil diskusjon og fremgangsmåte for å komme videre fra problemet ha konsekvenser for hvordan samarbeidet videre i prosjektet kommer til å bli. Sannsynligheten er beregnet ut ifra hvor vidt vi tror vi støter på uenigheter som kan skape konflikter imellom oss. Det kan også hende at deltakere i prosjektet ikke bidrar, og at arbeidsfordelingen blir skjev, noe som fører til misnøye.

Konsekvens: 4

Samarbeidet i prosjektgruppa er avgjørende for kvalitet og nytte i prosjektet. Dårlig samarbeid fører til lavere motivasjon og tar fokuset vekk fra målet til prosjektet.

Risikokarakter: 8**Tiltak:**

Ved konflikter som vedvarer kontaktes veileder for å lage en avtale på retningslinjer for å minske konflikten. Kontinuerlig bør prosjektgruppen ha en positiv holdning til konflikter og holde en god dialog gjennom prosjektet.

6. Tap av prosjektmateriale

Sannsynlighet: 1

At vi taper prosjektmateriale av betydning er lav, da alt lagres på OneDrive som har beskyttelse mot sletting og skade. Om noe skulle skje ville det vært et resultat av enten sabotasje eller feil fra Microsoft sin side, noe som i begge tilfeller er lite sannsynlig.

Konsekvens: 5

Tap av prosjektmateriale vil kunne ha fatale konsekvenser for prosjektet, avhengig av hva som går tapt. Å miste viktige deler av arbeidet slik som rapporter og liknende, vil kunne sette oss langt tilbake i prosessen og i verstefall føre til at vi ikke får gjennomført oppgaven på tilfredsstillende vis.

Risikokarakter: 5

Tiltak:

I tillegg til at OneDrive tar backup av filene våre, er det viktig at vi jevnlig tar sikkerhetskopi av viktige filer på en egen disk. Igjen er det ikke sannsynlig at vi kommer til å trenge dette, men det er best å være på den sikre siden.



Figur 4 - Risikokart

9. Kost/nytte-analyse

Dette prosjektet baserer seg på at prosjektgruppen gjør undersøkelser av IBM Spectrum Scale og setter opp en prototype for testing. Vi utformer kost-/nytteanalysen med utgangspunkt i nytten vi gjør for bedriften i gjennomføringen av bachelorprosjektet. Fokuset legger vi på ikke-kvantifiserbar nytte og bortfall av kostnader.

Prisen på et reelt system blir påvirket av mange ulike faktorer, slik som konfigurasjon av maskinvare og tjenester som inngår i avtalen. Dersom bedriften går for en slik løsning, vil det mest sannsynlig bli inngått en skreddersydd pakkeavtale mellom IBM og Kongsberg med egne fastsatte priser som vi ikke har oversikt over. Vi vet derfor lite om det virkelige økonomiske omfanget til et slikt prosjekt. Videre vil denne analysen derfor fungere mer som et eksempel for å demonstrere det økonomiske aspektet ved et slikt prosjekt. Kjernen i oppgaven er å lage et kostnadsbesparende system ved bruk av tape med SDS. Analysen tar utgangspunkt i å illustrere hvilke økonomiske fordeler dette vil ha.

Som studenter har vi tilgang til gratis lisenser for ulike it-systemer tiltenkt opplæring som tilbys både av NTNU og av tjenesteleverandørene. Bedriftene tjener på å gi vekk gratis lisenser til studenter fordi vi muligens tar det i bruk når vi kommer ut i arbeidslivet. I tillegg har NTNU ressurser tilgjengelig som kan være nyttige for prosjektet som fysiske maskiner. Dette tas også med i analysen som en del av bortfallende kostnader.

Eksempel:

Vi forestiller oss at bedriften kommer til å starte med et lagringsbehov på 1 Petabyte, med en årlig økning på 10% av kapasiteten, lagringstid av data på 3 år og at 5% av dataen hentes ut årlig for lesing.

9.1 Kvantifiserbar nytte

Prosjektet har ingen kvantifiserbar nytteverdi som vi har oversikt over. Fokuset ligger som sagt på å skape ikke-kvantifiserbar nytte som Kongsberg Seatex kan ha nytte av ved en faktisk implementasjon, samt bortfall av kostnader knyttet til arbeidet vi gjør i dette prosjektet.

9.2 Ikke-kvantifiserbar nytte

Bedre forutsetninger for å oppnå effektmålene:

Prosjektmålene i 3.2 definerer prosjektets effektmål, hvor nytteverdien av dette prosjektet er å gjøre målene lettere å oppnå.

Innsikt i IBM Spectrum Scale:

For prosjekteieren har det en verdi å ha «hands-on» kunnskap om systemet. Dette gir grunnlag for bedre beslutninger og færre feiltrinn i vurderingen av en eventuell investering. I tillegg er dette nyttig informasjon å kunne referere til i en drifts-sammenheng.

Senke risikoen for mislykket prosjekt:

Hvis systemet skal inn i bedriften er sannsynligheten for at det vellykkes høyere. Gjennom dokumentasjon og testing av systemet blir kravspesifikasjoner og behov klarere.

9.3 Bortfall av direkte kostnader

Bortfall av direkte kostnader er en av hovedtankene bak dette systemet. Ifølge case-studiene fra IBM kan Spectrum Scale spare mange ressurser på flere områder avhengig av bedriftstypen (IBM, 2020b). Kostnadene som unngås er i stor grad knyttet til frigjøring av unødvendig plassbruk, raskere responstid, høyere oppetid, enklere administrasjon og mindre bruk av dyre lagringsmedier. Systemet sparer også strøm ved bruk av tape som lagringsmedium og ved frigjøring av diskbruk.

9.3.1 Bortfall av direkte kostnader i prosjektoppgaven

Arbeidstimer for research og utvikling:

Timer som prosjektgruppen bruker på test av lagringssystemet er timer som bedriften ellers ville brukt på sine ansatte. Dette frigjør arbeidskraft for bedriften og sparer penger i lønnskostnader. Den totale arbeidsmengden estimert for bachelorprosjektet er 1000 arbeidstimer. Med en antatt timelønn på 350 kroner utgjør dette 350 000 kroner i sparte lønnskostnader.

Lisenskostnader:

Som studenter har vi tilgang til lisenser på programvare for opplæring. Dette gjør det enklere å prøve ulike systemer. Lisensene vi har tilgjengelig er 2000 kr i Azure og 2000 kr i AWS.

Testsystem:

Av NTNU får vi støtte til bruk av Azure for oppsett av testsystemet. Kostnadene for dette er basert på kontinuerlig bruk av 8 noder med tilkoble disk over en måned. Siden testsystemet ikke skal brukes i produksjon løper det bare på kostnader når vi faktisk bruker systemet. En måned tilsvarer cirka 730 timer. Kostnadene vi estimerer for dette er på 2269 kr.

9.3.2 Bortfall av kostnader i reell gjennomføring

Bytte til tape

Kostnaden på en terrabyte på disk er fra 330 kr (Dustin, 2020b), mens en terrabyte med tape kommer i fra 100 kr (Dustin, 2020a). Kostnaden for tape er i dette eksemplet er 70 % lavere enn kostnaden for harddisk per TB. Denne besparelsen er en av de største nyttene som kommer frem ved bruk av et SDS som IBM Spectrum Scale, og er en av hovedgrunnene til å implementere et slikt system. I eksempelet vårt bruker vi 90% av lagringen til tape, og de resterende 10% til harddisk. Dette gir $0.7 \cdot 0.9 =$ minst 63% kostnadsbesparelse kun på bruk av billigere lagringsmedium. Man vil også spare penger fordi man kan bruke «standard» hardware i en Spectrum Scale løsning, og fortsatt oppnå enterprise-ytelse.

For 900TB med lagring gir dette oss en kostnadsbesparelse på kr 207 000 fordi:

$(900\text{TB med HDD koster } 297\ 000) - (900\text{TB med tape koster } 90\ 000) = \underline{207\ 000 \text{ kr.}}$ (Se 9.4)

9.4 Estimerte kostnader

Lisenskostnader:

Lisenskostnadene vil være inkludert i en pakkeavtale som inngås mellom IBM og Kongsberg Seatex, der et fullstendig system leveres til en fast pris, med ubegrenset brukskapasitet. Prisen for en slik avtale har vi ikke tilgang til. Derfor tar vi ikke lisenskostnader med i regnestykket. Skal man derimot installere

systemet på egen maskinvare kreves en selvstendig lisens som enten går på kapasitet eller antall kjerner som skal brukes i clusteret. Dette avhenger av hvilket formål og lisens løsningen har.

Lagringskostnader:

Kostnadene for prosjektet vil primært dreie seg rundt lagringsmediene som skal brukes for å arkivere de store datamengdene, og det er her kostnadene vil øke mest over tid sammen med potensialet for sparing. Å få tak i eksakte tall på hva dette vil koste er krevende, særlig når vi ikke vet nøyaktig hvor mye bedriftens behov for lagring er over tid, og hva slags fordeling mellom høy-hastighets og lav-hastighets lagring som trengs. Vi har derfor tatt utgangspunkt i en pris per TB, og med 90% av lagringen på tape, mens resten er på disk. Estimert databehov er 1PB (1000TB). Dermed blir behovet for lagring for tape på 900TB og 100TB for disk.

For å dekke behovet på 900TB med tape kreves 150 LTO Ultrium 7 kassetter (Dustin 2020a). Hver av disse har en kapasitet på 6TB. Med en pris på 600 kroner per kassett gir dette en pris på 90 000 kroner for tape.

For 100TB disklagring ser vi på 8 stykk 14TB disk (Dustin 2020b). Hver av disse koster 4600 kroner, blir total pris for disk 36 800 kroner.

9.5 Sammenstilling kost/nytte

Kost/nytte for reell gjennomføring:

	År 0	År 1	År 2	År 3
Kvantifiserbar nytte	0	0	0	0
Bortfall kostnader	207 000	20 700	20 700	20 700
Sum nytte	207 000	20 700	20 700	20 700
Utviklingskostnader	?	0	0	0
Utstyrskostnader	126 800	12 680	12 680	12 680
Drifts- og forv. Kostnader	?	?	?	?
Sum kostnader	126 800	12 680	12 680	12 680
Beregnet nytte (nyttekostnader)	80 200	8 020	8 020	8 020

Tabell 3 - Sammenstilling kost/nytte for et reelt prosjekt

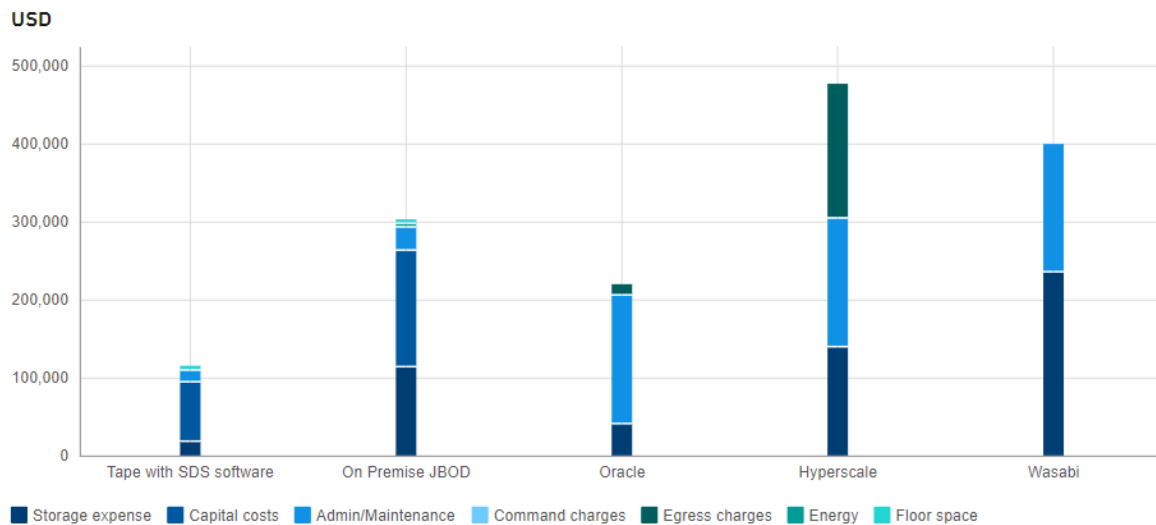
Kost/nytte av bachelorprosjekt beregnet ut ifra gjennomføring av prosjektet:

	År 0
Bortfall kostnader prosjekt	350 000
Bortfall kostnader lisens	4000
Bortfall kostnader testsystem	2269
Sum nytte	356 269
Sum kostnader	0
Beregnet nytte (nyttekostnader)	356 269

Tabell 4 - Oversikt kost/nytte bachelorprosjekt

På illustrasjonen nedenfor ser du et estimat fra IBM sin TCO-kalkulator for kostnadsbesparelse ved bruk av tape og SDS (Software Defined Storage), med andre ord en slik ordning som dette prosjektet baserer seg på. Tallene som er brukt i beregningen er de samme som eksempelet vi har tatt utgangspunkt i for kost/nytte-analysen (IBM, 2020c).

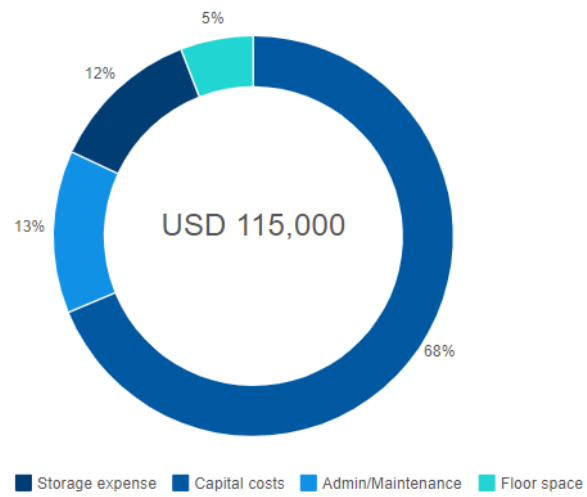
Total solution cost



Figur 5 - Kostnadsbesparelse tapeløsning

Vi har også illustrert en mer detaljert inndeling av hvordan kostnadene vil fordele seg med tape og SDS software.

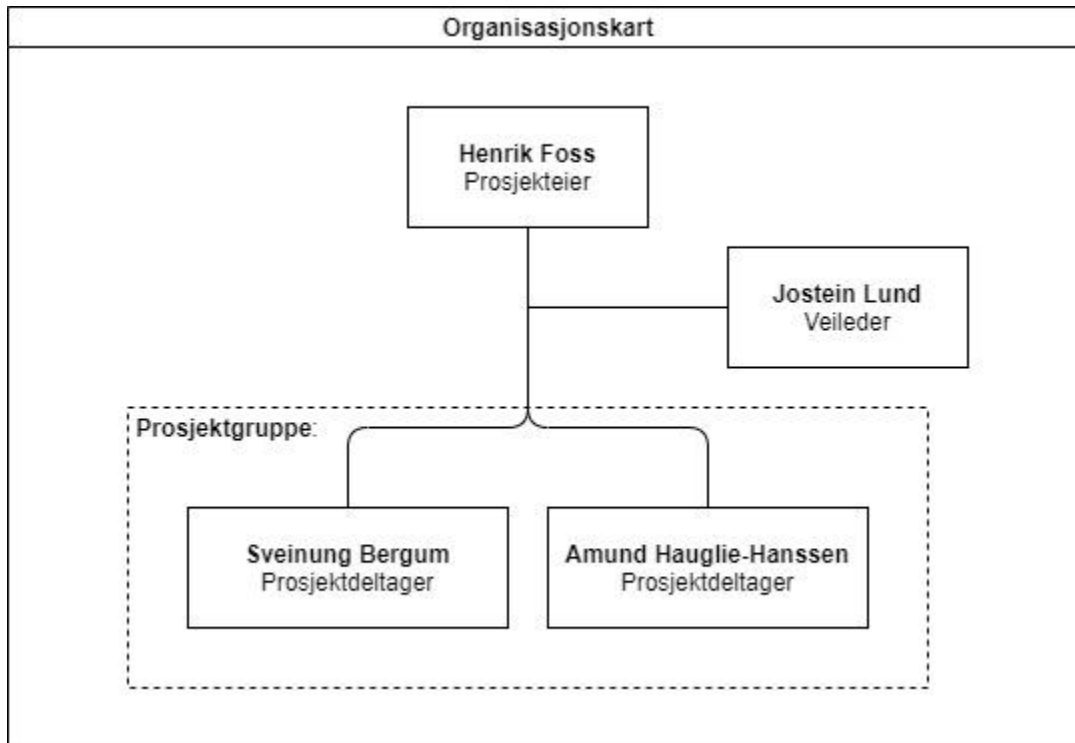
Detailed results



Figur 6 - Kostnadsfordeling tapeløsning

10. Prosjektorganisering og ansvarsområder

Her ser vi på struktur av prosjektorganisasjonen, og hvilke ansvarsområder deltakerne har.



Figur 7 - Prosjektorganisasjon

Ansaret til prosjekteier:

Prosjekteiers ansvarsområde er å bidra til bachelorprosjektet som en støttespiller og rådgiver til prosjektgruppen. Prosjekteier stiller med informasjon som er relevant for oppgaven og til hjelp for prosjektgruppen. Dersom prosjektgruppen sliter med å gjennomføre deler av oppgaven skal prosjekteier bistå med nødvendig hjelp.

Ansaret til prosjektgruppen:

Hovedansvaret i oppgaven ligger på studentene i prosjektgruppen. Det er prosjektgruppen som skal gjøre mesteparten av arbeidet knyttet til forskning og utvikling, samt produksjon av materiell for bacheloroppgaven. Studentene har ansvaret for å få fremgang i prosjektet, avtale møter og utarbeide dokumentasjon.

Ansaret til veileder

Veileder fungerer som rådgivende støtte for prosjektgruppen underveis i prosessen, og ved behov dersom studentene har utfordring med oppgaven. Veileder vil også delta aktivt i vurderingen av oppgaven og forløpet, og dermed også studentenes individuelle innsats.

11. Retningslinjer og standarder

11.1 Dokumentasjon

I bachelorprosjektet vil det utarbeides 3 rapporter:

Forstudierapport:

I forstudierapporten skal hensikten bak prosjektet beskrives. Rapporten utleder mål for prosjektet og ansvarsforhold mellom de ulike partene. Sentralt kommer en kost/nytte-analyse som skal illustrere noen av mulighetene for kostnadsbesparelser, og gir innblikk i nytten av bachelorprosjektet. Vi ser her på prosjektets omfang på overordnet nivå, og gjør grovere anslag på arbeidsmengde og ressurskrav.

Hovedrapport:

Denne rapporten vil ta for seg bacheloroppgaven i sin helhet. Dokumentet repeterer først prosjektets utgangspunkt og problemstilling. Videre dokumenteres oppbygningen av Spectrum Scale, både på teknisk og konseptuelt nivå. Målet her er å finne en balanse som gjør at leseren henger med, og får forståelsen som er nødvendig uten å gå for mye i detalj.

Hovedrapporten kan ansees som todelt, der den først delen til og med kapittel 3, tar for seg teorien og konseptene bak løsningen, samt prosjektets utgangspunkt. Videre fra kapittel 4 designes og installeres et testsystem med IBM Spectrum Scale, der hensikten er å vise funksjonaliteten i praksis, slik at leseren får et innblikk i hvordan operasjonen av et slikt system ser ut.

Poenget med hovedrapporten er å demonstrere hva IBM Spectrum Scale kan gjøre, og mer spesifikt hvordan systemet kan brukes i sammenheng med oppdragsgivers utfordringer. Gjennom tegninger, teori og skjermbilder av installasjon og drift, vil hovedrapporten forhåpentligvis gi Kongsberg Seatex et godt innblikk i hva systemet handler om, og hvilke muligheter som finnes.

Avslutningsrapport:

Dette er en oppsummering av prosjektet. Her blir prosessen gjengitt i sin helhet, og de ulike fasene som prosjektet har bestått av drøftes. Vi ser på hvilke utfordringer vi har hatt i prosjektet, og hvilke erfaringer vi har gjort oss underveis. Rapporten skal vise hvordan samarbeidet i prosjektgruppen har gått, og dokumentere resultatet av prosjektet opp imot forventningene.

11.2 Ukentlig planlegging or rapportering

Det er fastsatt at det skal avholdes møter ukentlig internt i prosjektgruppen. Dette for å rapportere prosjektets status og den enkelte medlems bidrag til prosjektet ved slutten av hver arbeidsuke. Dette er for å sørge for jevn og oversiktlig fremgang i prosjektet, og minimere risiko for feilberegning av arbeidsmengden. Å rapportere den enkeltes bidrag er også en viktig faktor i forhold til vurdering av bacheloroppgaven. Resultatet av dette møtet skrives som en oppsummering for hver uke i prosjekthåndboken som ligger vedlagt sluttinnleveringen. Her vil også antall timer for hver uke knyttet opp mot aktivitetene i prosjektet føres.

For å holde oversikt over oppgaver, frister og ansvar bruker vi Teams med Planner-applikasjonen. Her kan vi prioritere ulike oppgaver etter viktighet, og holde oversikt over oppgaver som er i gang, hvilke som står på vent, og hvilke som er fullført.

11.3 Møter med veileder

I fellesveiledningen for bachelorprosjektene ved instituttet for datateknologi og informatikk (IDI), har vi fått beskjed om at det skal holdes jevnlig oppfølgingsmøter med veileder underveis i prosjektet. Dette skal skje hver andre til tredje uke. Her vil prosjektets fremgang og utfordringer stå på agendaen, og veileder vil kunne stå som en støttespiller dersom prosjektgruppa trenger hjelp. Disse møtene rapporteres og vil ligge i prosjekthåndboken ved sluttinnleveringen.

11.4 Møter med oppdragsgiver

Møter med oppdragsgiver vil vi avtale etter behov, men vi sikter oss inn på møter cirka hver andre uke over Skype for å følge opp prosessen. Dersom det oppstår utfordringer og vi trenger å ha et spontant møte vil vi planlegge dette etter faktisk behov. Det er fastsatt at det skal være minst to møter med både oppdragsgiver og veileder, ett for oppstart og ett for den avsluttende presentasjonen.

11.5 Prosjektplanlegging i MS Project

Innledningsvis har vi laget en grov skisse av gangen i prosjektet i MS Project (Se kapittel 12: «Aktivitets- og ressursplan»). Gantt-diagrammet viser hvilke aktiviteter som skal skje til hvilke tider, i hvilken rekkefølge, og antatt tidsforbruk for hver aktivitet. I forbindelse med det interne ukentlige oppsummeringsmøtet vil vi også gå igjennom planen for neste uke, og oppdatere Gantt-diagrammet om det skjer endringer i planene.

11.6 Timeføring

I tillegg til den ukentlige timeføringen som ligger ved prosjekthåndboken vil det også føres totalt antall timer i et eget excel-dokument.

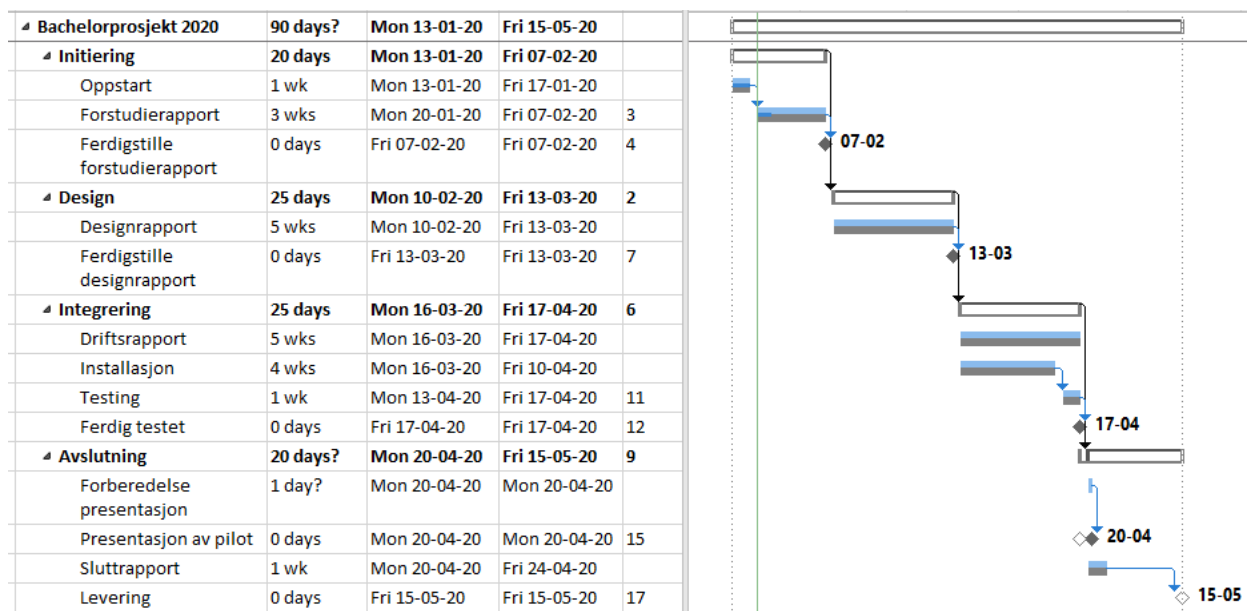
11.7 Informasjon om bruk av kilder

Alle kilder som blir brukt for å støtte oppunder arbeidet føres i en kildeliste for hvert dokument som leveres med sluttinnleveringen. Dette gjør vi for å holde det ferdige arbeidet mest mulig ryddig og oversiktlig, og slik at det blir lettere å knytte kildene opp imot stoffet. Kildene føres i Harvard-stil basert på NTNUs retningslinjer for kildeføring.

12. Aktivitets- og ressursplan

12.1 Plan for hele prosjektet

Som vedlegg til sluttinnleveringen, ligger Project-filen som skisserer prosjektets forløp fra start til slutt. Her presenteres hovedaktivitetene med milepæl i grove trekk. Vi har laget et utdrag som viser Gantt-diagrammet (Figur 8). Her ser vi at prosjektet deles inn i fasene initiering, design, integrering og avslutning. Under hver av disse fasene kommer det tilhørende oppgaver. Oppgavene i hver fase består primært av en rapport som skal utarbeide arbeidet som hører til fasen, eksempelvis designrapporten for designfasen. På dette tidspunktet velger vi å ikke gå dypere inn på oppgavene i hver fase. Dette er noe vi kommer til å gjøre ved oppstart av neste fase i oppdateringen av project-filen. Derfor ser du her kun et førsteutkast av Gantt-diagrammet.



Figur 8 - Utkast av Projectfil

13. Konklusjon og anbefalinger om videre arbeid

Igjennom utviklingen av denne forstudierapporten vil vi konkludere med at dette er et spennende prosjekt som kan gjennomføres. Med dette mener vi at bacheloroppgaven anses som et verdifullt forskningsprosjekt for bedriften Kongsberg Seatex. Igjen er hensikten med prosjektet å undersøke og vurdere filsystemet IBM Spectrum Scale i henhold til bedriftens problemstilling, slik at de i ettertid kan bruke dette materialet i sin vurdering av en virkelig implementasjon.

Usikkerhetene for den videre gangen i prosjektet ligger hovedsakelig i omfanget av oppgaven. Det er viktig at oppgaven snevres inn nok til at vi klarer å produsere meningsfulle resultater. Dette er en usikkerhet som er vanskelig å ta stilling til nå fordi vi ikke kan forutse hvor mye jobb prosjektet krever. Vi anser disse utfordringene som overkommelige, og gir med dette utgangspunktet grønt lys til gjennomføringen av prosjektet.

Videre kommer vi til å gå grundigere til verks på dokumentasjonen til IBM Spectrum Scale, og lese oss opp på ulike kilder som vi kan bruke i oppgaven. IBM har utgitt flere bøker om sine produkter i tillegg til dokumentasjonen på nett, så vi har tilgang på den informasjonen vi trenger. Ved hjelp av kunnskapen vi tilegner oss, skal vi skrive om programmet i en rapport satt opp imot bedriftens problemstilling. Deretter skal vi designe, installere, teste og dokumentere en testløsning med IBM Spectrum Scale i et cluster. Dette er planen videre for å oppnå prosjektmålene slik vi har definert dem i denne rapporten.

Kilder

- Dustin (2020a) *HPE HPE Ultrium Non-Custom Labeled Data Cartridge, LTO Ultrium 15TB 20st*. Tilgjengelig fra: <https://www.dustin.no/product/5010896839/hpe-ultrium-non-custom-labeled-data-cartridge> (Hentet 24.02.2020)
- Dustin (2020b) *Seagate IronWolf Pro, 14TB 3.5" Serial ATA-600*. Tilgjengelig fra: <https://www.dustin.no/product/5011094950/ironwolf-pro> (Hentet 24.02.2020)
- IBM (2020a) *IBM Spectrum Scale Documentation*. Tilgjengelig fra: https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/ibmspectrumscale504_welcome.html (Hentet: 16.04.2020).
- IBM (2020b) *IBM Storage Case Studies*. Tilgjengelig fra: <https://ibm.lookbookhq.com/l/storage-case-study-hub> (Hentet: 24.februar 2020)
- IBM (2020c) *Tape TCO Calculator*. Tilgjengelig fra: <https://www.ibm.com/it-infrastructure/resources/tools/storage-tco-calculator/> (Hentet 24.02.2020)
- Quintero, D., et al. (2015). *IBM Spectrum Scale (formerly GPFS)*. Tilgjengelig fra: <http://www.redbooks.ibm.com/redbooks/pdfs/sg248254.pdf> (Hentet: 16.04.2020)



Software Defined Storage med IBM Spectrum Scale

Med fokus på effektivisering og ressursbesparelse

Hovedrapport

For bachelorprosjektet v2020 i Drift av Datasystemer

Amund Hauglie-Hanssen, Sveinung Bergum
15.05.2020

Innhold

Figurliste.....	3
Tabelliste.....	5
Sammendrag.....	6
1 Innledning	8
1.1 Dokumentets hensikt.....	8
1.2 Bachelorprosjektets deler.....	8
1.3 Definisjoner og forkortelser.....	8
2. Problemstilling	9
2.1 Kort om kunden og behov	9
2.2 Resultat av forstudiet.....	9
2.3 Avgrensning	10
2.4 Rapportens oppbygning.....	10
2.4.1 Oversikt	10
2.4.2 Roller	11
2.4.3 Innhold i rapporten.....	12
3. Hva er IBM Spectrum Scale?.....	14
3.1 Bakgrunn for SDS (Software Defined Storage).....	14
3.1.1 Information Lifecycle Management.....	14
3.2 Software Defined Storage i praksis	15
3.2.1 IBM Spectrum Scale som SDS	15
3.2.2 Hensikten med IBM Spectrum Scale.....	16
3.3 Nøkkelpunkter og funksjoner	16
3.3.1 Smart filtrering og migrering av data.....	16
3.3.2 Active File Management	18
3.3.3 Cluster Export Services.....	19
3.3.4 Stabilitet og høy oppetid.....	19
3.3.5 Skalerbarhet og høy ytelse.....	20
3.3.6 Enkel administrasjon.....	20
3.4 Komponenter	20
3.4.1 Programvaren	20
3.4.2 GPFS-clusteret.....	22

3.4.3 Nodenes roller.....	22
3.5 Spectrum Scale Lagring	24
3.5.1 Lagringsmedier.....	24
3.5.2 Storage pools	26
3.5.3 Failure groups	27
3.5.4 Backup.....	27
3.5.5 IBM Spectrum Archive	28
3.6 Lisensiering.....	29
3.7 Begrensninger	31
4. Installasjon av IBM Spectrum Scale	32
4.1 Clustertopologi og definisjon av noder.....	34
4.1.1 Roller	34
4.1.2 Quorum.....	35
4.2 Forhåndsinstallasjon	35
4.2.1 Krav for bruk av installasjonsverktøyet.....	35
4.2.2 Klargjøring av installasjonsverktøyet	37
4.2.3 Kommunikasjon mellom noder.....	39
4.2.4 Manuell installasjon	41
4.2.5 Klargjøring av CES.....	41
4.2.6 OpenLDAP med phpLDAPadmin	42
4.3 Installasjon av filsystemet.....	52
4.3.1 Oppsett av noder i clusteret	52
4.3.2 Legge til disker og storage pools.....	53
4.3.3 Utrulling av filsystemet	54
5. Administrasjon og drift av IBM Spectrum Scale.....	56
5.1 Initiell konfigurasjon	56
5.1.1 Oppsett og bruk av kommandolinjen	56
5.1.2 Oppsett av management GUI	57
5.1.3 Oversikt/Hjemmeside	58
5.1.4 Sette opp brukere	59
5.1.5 Koble til ekstern brukerdatabase.....	60
5.1.6 Egendefinert dashbord	63
5.1.7 Opprette filsett	65

5.1.8 Oppsett av policyer	67
5.2 Daglig drift.....	71
5.2.1 Systemhelse	71
5.2.2 Inntak av data	78
5.2.3 Feilhåndtering.....	79
5.2.4 Snapshot	81
5.3 Sluttbrukere	83
5.3.1 Applikasjonsnode.....	83
5.3.2 Delt filsystem over NFS	84
6. Konklusjon og anbefalinger om videre arbeid	87
6.1 Konklusjon.....	87
6.2 Vedlegg	88
Kilder	89
Bibliografi	89
Referanseliste	89

Figurliste

Figur 1 - Enkel systemoversikt	7
Figur 2 - Kapitteltemaer	11
Figur 3 - Importering av filer	17
Figur 4 - Active File Management	18
Figur 5 - Cluster Export Services	19
Figur 6 - Nodeoppbygning.....	21
Figur 7 - Cluster	22
Figur 8 - NSD Nettverk	23
Figur 9 - Clusterkonfigurasjon 1	25
Figur 10 - Clusterkonfigurasjon 2	25
Figur 11 - Clusterkonfigurasjon 3	26
Figur 12 - Storage pool nivåer.....	27
Figur 13 - Valgte for lisensiering	30
Figur 14 - Systemoversikt.....	32
Figur 15 - Systemdesign for testing.....	33
Figur 16 - Noderoller.....	34
Figur 17 - Installasjon på noder.....	36
Figur 18 - LDAP Gui 1.....	47
Figur 19 - LDAP Gui 2.....	47

Figur 20 - LDAP User Configuration	48
Figur 21 - LDAP Directory	48
Figur 22 - Noderoller 2	52
Figur 23 - Storage pool nivåer 2	53
Figur 24 - GUI Innloggingsportal	57
Figur 25 - GUI Hjemmeside	58
Figur 26 - GUI Meny	58
Figur 27 - GUI Brukertilgang.....	59
Figur 28 - GUI Ny bruker	59
Figur 29 - GUI File Authentication.....	60
Figur 30 - GUI Config FA	60
Figur 31 - GUI Config FA 2	61
Figur 32 - GUI Config FA 3	61
Figur 33 - GUI Config FA 4	62
Figur 34 - GUI Config FA 5	62
Figur 35 - GUI Dashboard.....	63
Figur 36 - GUI Edit Dashboard.....	63
Figur 37 - GUI Configure Widget	64
Figur 38 - GUI Widget.....	64
Figur 39 - GUI Dashboard Profiles.....	64
Figur 40 - GUI Create Dashboard	65
Figur 41 - GUI Opprett Filsett.....	65
Figur 42 - GUI Opprett Filsett 2.....	66
Figur 43 - GUI Oversikt Filsett	66
Figur 44 - GUI Policy 1	67
Figur 45 - GUI Add Rule.....	68
Figur 46 - GUI Import Placement	68
Figur 47 - GUI Migration Rule	69
Figur 48 - GUI Migration Rule Config.....	69
Figur 49 - GUI Syntaks for regel	70
Figur 50 - GUI Regler Oversikt.....	70
Figur 51 - GUI Skriver til tape	71
Figur 52 - GUI Dashboard 2	71
Figur 53 - GUI Meny 2	72
Figur 54 - GUI tips	72
Figur 55 - GUI Tips detaljert	72
Figur 56 - GUI Notifikasjon.....	73
Figur 57 - GUI hendelse.....	73
Figur 58 - GUI markere hendelse	73
Figur 59 - GUI Epost	74
Figur 60 - GUI statistikk.....	74
Figur 61 - GUI detaljert statistikk	75
Figur 62 - GUI Nodeinformasjon	75
Figur 63 - GUI Clusterinformasjon	76

Figur 64 - GUI Filsystem informasjon	76
Figur 65 - GUI Storagepools informasjon	77
Figur 66 - GUI NSD informasjon	77
Figur 67 - Data Ingest	78
Figur 68 - GUI Systemhendelser	79
Figur 69 - GUI Hendelsesinformasjon	80
Figur 70 - GUI Fix Procedure	80
Figur 71 - GUI Create Snapshot	81
Figur 72 - GUI Snapshot Path	82
Figur 73 - GUI Snapshot Rule	82
Figur 74 - GUI Create Fileset 2	83
Figur 75 - NFS med autentisering mot LDAP	84
Figur 76 - GUI Create ACL Template	85
Figur 77 - GUI CES Export	86

Tabelliste

Tabell 1 - Revisjonshistorie	5
------------------------------------	---

Tabell 1 - Revisjonshistorie

Dato	Utgave	Kommentar
11.03.2020	1.0	Ferdigstilt førsteutkast av design-del
27.03.2020	2.0	Ferdigstilt førsteutkast av drifts-del
07.04.2020	3.0	Siste utgave, førsteutkast
24.04.2020	3.1	Finpuss og gjennomgang
15.05.2020	3.2	Klar til levering

Sammendrag

Dagens IT-bedrifter utfordres av stadig større mengder data, ellers kjent som «Big Data». Dette kommer av at mer og mer av teknologien vår digitaliseres og samler informasjon om omgivelsene rundt seg. For at bedriftene skal kunne dra nytte av dataen og sette den i sammenheng må den lagres et sted i mellomtiden. Problemet er at lagring hverken er gratis eller logisk problemfritt. Big Data krever smarte og sikre lagringsløsninger som kan hjelpe til med å strukturere og sortere dataen riktig. Dette er essensielt for å holde kostnader, ytelseskrav og kompleksitet under kontroll. I tillegg har ofte større bedrifter mange systemer og enheter med ulikt operativsystem som skal kommunisere med hverandre, noe som krever en felles plattform som kan fungere som et samlepunkt.

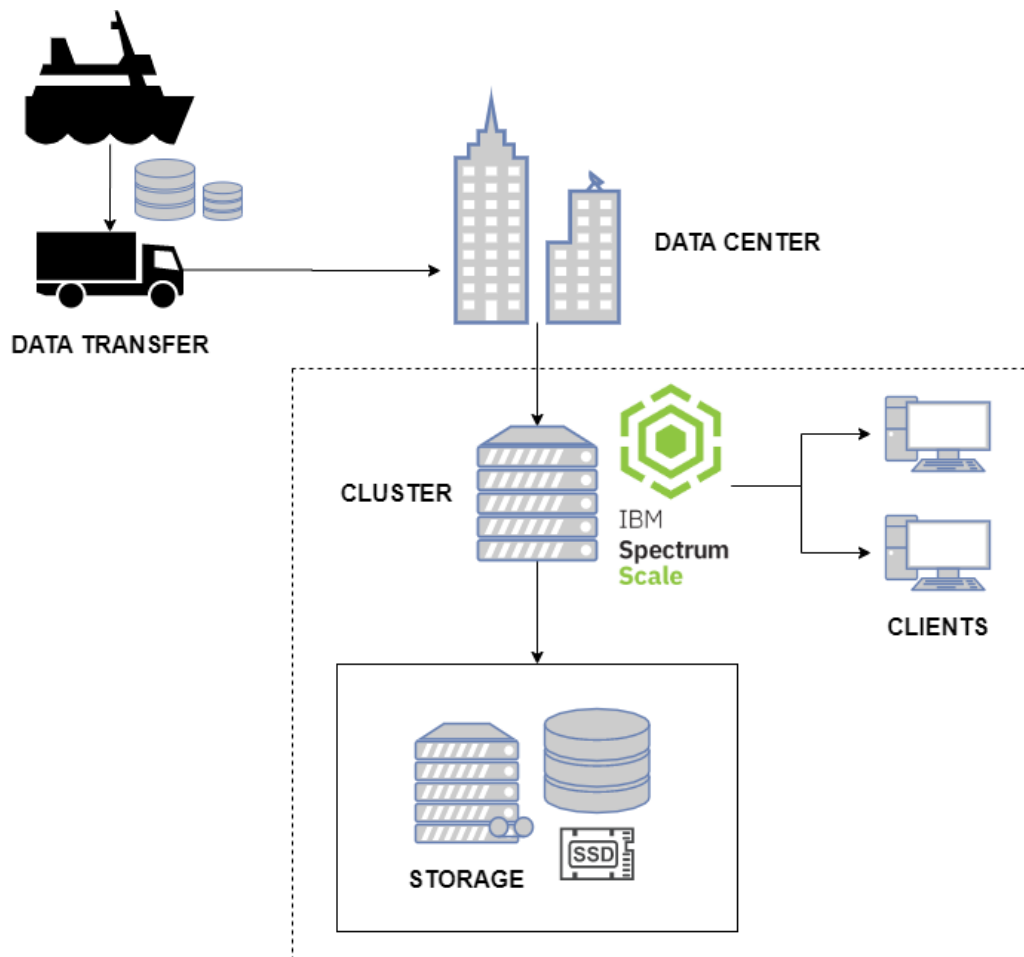
I forstudierapporten ble Kongsberg Seatex sine problemstillinger rundt akkurat dette temaet belyst. Vi så her på hva bedriften kan gjøre for å håndtere de store datamengdene som kommer inn fra sin autonome flåte. Her kom vi frem til at lagringssystemet IBM Spectrum Scale kunne være et relevant alternativ for bedriften. I denne rapporten har vi gått dypere inn på programmets funksjonalitet, og undersøkt om det har det som skal til for å dekke kravene til bedriften.

Filsystemer som er designet for strategisk styring av data, kalles for «Software Defined Storage», og IBM Spectrum Scale er et eksempel på dette. Her tar man i bruk et konsept som kalles «Information Lifecycle Management», som ser på hvordan data bør styres gjennom livssyklusen basert på dataens verdi. Ut ifra resultatet av slik styring, bestemmer man hvordan lagringssystemet skal analysere og behandle dataene som importeres. IBM Spectrum Scale innebærer funksjonalitet for å automatisk migrere filer til et lagringsmedium som passer for typen data, noe som både sparer kostnader og øker responstiden til filsystemet. Slik kan man bruke billige og trege medier for arkivering, mens man bruker raske og dyrere medier for hurtiglesing. Spectrum Scale er også svært skalerbart, har høy ytelse, og sikrer dataen mot tap og korrupsjon.

I bachelorprosjektet har vi utarbeidet en omfattende dokumentasjon om Spectrum Scale, som har til hensikt å gi leseren en innføring i hvordan systemet fungerer, hvilke oppgaver det kan løse, og hvordan det installeres og tas i bruk. Vi har også satt opp et testsystem i Azure i liten skala, for å prøve ut og å dokumentere programmet i praksis. Hensikten med å ta i bruk en løsning som Spectrum Scale over et tradisjonelt lineært filsystem, er at man sparer betydelige mengder ressurser på lagring av Big Data. Samtidig er systemet designet for å takle fremtidig skalering, og belastningene slike datamengder skaper.

Bachelorprosjektet skal gi Kongsberg Seatex verdi i form av innsikt i filsystemets funksjoner, og Spectrum Scales relevans i forhold til problemstillingen. Vi har igjennom arbeidet med dette prosjektet konkludert med at IBM Spectrum Scale er et sterkt alternativ for bedriftens utfordringer, og som dekker kravene som ble satt i begynnelsen av prosjektet. Likevel må bedriften selv avgjøre om nytten av et slikt system veier opp for kostnadene, og hvordan dette kan forsvares økonomisk sett. Spectrum Scale krever innkjøp av en del maskinvare og programvare. Da vi ikke har oversikt over prisen på dette utstyret, kan vi ikke si mer om virkelig nytte bedriften vil ha av dette.

Figur 1 illustrerer utgangspunktet til bachelorprosjektet. Vi har med andre ord sett på hvordan nye data importeres til filsystemet fra båtene, hvordan den prosesseres og arkiveres, og hvordan den videreformidles til sluttbrukeren. Enkelt forklart transporteres data på tape inn til systemet, der det føres over på disk. Videre vil IBM Spectrum Scale håndtere dataen, og flytte den dit den skal være basert på regler satt i konfigurasjonen. Dermed kan klientene på høyre side lese og skrive data til filsystemet. Programmet tilbyr et brukervennlig grensesnitt for å overvåke filsystemets helse og ytelse, som kan nåes fra en klientmaskin i nettverket.



Figur 1 - Enkel systemoversikt

1 Innledning

Dette dokumentet er hovedrapporten for bachelorprosjektet om «Software Defined Storage» ved studiet «Informatikk, drift av datasystemer», gitt av Kongsberg Seatex AS. Prosjektgruppen (115) tar i prosjektet for seg lagringsløsningen IBM Spectrum Scale. Sett opp imot prosjektmålene som ble utarbeidet under forstudiefasen, skal prosjektet dokumentere oppbygningen og mulighetene til dette systemet, og sette dette i sammenheng med bedriftens utfordringer.

1.1 Dokumentets hensikt

Tidligere i bachelorprosjektet har prosjektgruppen opparbeidet seg grunnleggende kunnskap om IBM sitt lagringssystem med navn Spectrum Scale gjennom et forstudium. Med utgangspunkt i resultatene fra forstudierapporten og prosjektmålene som kom frem av denne, viderefører denne rapporten undersøkelsene av systemet. Dokumentet skal gi leseren en grunnleggende forståelse for IBM Spectrum Scale sin funksjonalitet, og demonstrere hvordan oppdragsgiver kan dra nytte av systemet. Rapporten skal fungere som vurderingsgrunnlag for bedriften under eventuell innføring av et nytt lagringssystem.

1.2 Bachelorprosjektets deler

Bachelorprosjektet består av et forstudium, en designfase og en driftsfase. Designfasen skal ta for seg teori bak Spectrum Scale og design av testløsningen, mens driftsfasen skal se på installasjon, konfigurasjon og bruk av testløsningen. Ut ifra prosjektmålene fra forstudierapporten, så vi at det ble enklere og mer oversiktlig å gjennomføre disse to fasene i ett. Det gjør dokumentasjonen bedre sammenflettet, og gjør det lettere for leseren å følge med. Derfor har vi slått sammen design og drift til en hovedrapport. Rapporten er fortsatt klart delt mellom teori og praksis som gjør det lett å se sammenhengen.

Innleveringen av prosjektet vil dermed bestå av forstudierapporten, denne hovedrapporten, og en kort avslutningsrapport. Som vedlegg kommer prosjekthåndboken med statusrapporter, møterefater og Gantt-diagram. I tillegg kommer en ordliste som forklarer tekniske begreper i rapportene. Prosjektet avsluttes med en felles presentasjon av prosjektet for veileder, sensor og bedrift.

1.3 Definisjoner og forkortelser

I forbindelse med dette prosjektet kommer det frem flere nye begreper og forkortelser som leseren kanskje ikke er kjent med. Derfor er lagt ved et oppslagsverk som beskriver og forklarer begreper og forkortelser. Likevel er dokumentet utformet for lesere med gode IT-kunnskaper, og vil derfor ikke forklare begreper som ikke er spesifikke for dette prosjektet.

2. Problemstilling

2.1 Kort om kunden og behov

Kongsberg Seatex benytter autonome skip til forskning og utvikling. Disse selvkjørende skipene samler store mengder data i drift som skal arkiveres og gjøres tilgjengelig for videre analyse. Bedriften har en utfordring med mengden data som fartøyene produserer. Hvert skip produserer mellom en til to terrabyte med data hver dag. Denne datamengden gjør det både komplisert og kostbart å lagre dataen over lengre tid, enten i skytjenester eller i ren diskløsning. Dagens løsning er at all dataen lagres på et NAS, noe som ikke er spesielt raskt, sikkert eller skalerbart på sikt. Et NAS er en enkel lagringsløsning som kobles på nettverket.

For å håndtere de store datamengdene ønsker oppdragsgiveren å se på muligheten for å benytte tape for billigere lagring av data over tid. Utfordringen med tape er at det har mye lavere lese- og skrivehastigheter enn disker. For å gjøre systemet responsivt og effektivt nok for analyse trengs derfor et hybridssystem som både benytter disker for data som behandles og tape for langtidslagring av data. Men ved å kombinere ulike filsystemer oppstår et problem med å betjene dem og holde oversikt over hvor dataene befinner seg. Oppdragsgiveren ønsker derfor at alt samles under ett sikkert filsystem der dataen blir flyttet mellom de ulike lagringsmediene automatisk, uten å være til bry for sluttbrukeren.

2.2 Resultat av forstudiet

Av oppdragsgiveren fikk vi i oppgave å se på IBM Spectrum Scale som alternativ for å dekke behovene som ble beskrevet i 2.1. Bedriften har vært i dialog med IBM om en slik løsning tidligere, og kan tjene på at det utføres et bachelorprosjekt om dette temaet. I den forrige fasen i bachelorprosjektet har prosjektgruppen utarbeidet en forstudierapport som tar for seg oppdragsgiverens problemstilling, systemet på overordnet nivå, og temaer rundt selve gjennomføringen av prosjektet. Her har vi sett på bedriftens behov, og hvordan bachelorprosjektet kan utformes i forhold til behovene.

Gjennom arbeidet med forstudiet kom vi frem til prosjektets krav og mål, som vi skal følge opp i denne rapporten. Med utgangspunkt i disse målene (se kapittel 3 i forstudierapporten), ser vi på noen generelle områder som vi anser som relevant. Både det teoretiske og det praktiske i rapporten vil dreie seg om disse temaene, som beskrevet i kapittel 2.4. Noe av det vi skal se på er følgende:

- 1. Oppbygningen av Spectrum Scale på konseptuelt og teknisk nivå:**

Å gjennomgå oppbygningen av Spectrum Scale er viktig for å få generell forståelse for hva systemet kan gjøre. Dette står sentralt i å dekke målene om dokumentasjon av funksjonaliteten i Spectrum Scale, og i vurderingen av systemet i forhold til Kongsberg Seatex sitt Big Data problem.

- 2. Design, konfigurasjon og installasjon av et testmiljø for bruk av IBM Spectrum Scale:**

Design, konfigurasjon og installasjon av testmiljøet viser grunnleggende oppsett av IBM Spectrum Scale. Her tar vi for oss resultatmålet fra forstudierapporten som dreier seg om å teste og installere systemet, og demonstrerer mulighetene for skalering. For å teste ut programvaren i praksis er det essensielt å ha et fungerende testmiljø.

3. Hvordan data importeres, håndteres og sorteres i Spectrum Scale:

En viktig problemstilling handler om hvordan data skal overføres fra båten og inn til filsystemet. Videre må dataen behandles av systemet, og sorteres imellom ulike lagringsmedium basert på type. Å forstå verdien av dataen, og sette dette i sammenheng med dataens livssyklus er et nøkkelkriterium for god bruk av Spectrum Scale. Dette området er sentralt for effektmålene i forstudierapporten. Effektmålene omfatter virkningen Spectrum Scale har for Kongsberg, der hovedkriteriet er enklere behandling av data.

4. Tilgjengeliggjøring av data for sluttbrukere og prosessering:

Ulike brukere krever ulik kapasitet og tilgang til systemet, og tilgjengelighet for brukerne er derfor viktig. Tilgjengelighet omfatter opetid og tilgang til systemet for brukere. God tilgjengelighet til systemet er en forutsetning for Kongsberg Seatex.

5. Daglige driftsoppgaver som overvåkning av systemets helse og respons på hendelser:

Videre må den daglige driften av systemet være enklere og gi bedre støtte enn å manuelt behandle data. Dette vil dekke effektmålet om å forenkle arbeidsoppgaver.

2.3 Avgrensning

I en reell installasjon vil systemets krav fastsettes i spesielle avtaler, og bli levert som en ferdig tilpasset pakke for bedriften satt sammen av IBM sine konsulenter. Bachelorprosjektet handler derfor mer om å vise hvordan Spectrum Scale er bygget opp og hvilke praktiske problemer det kan løse for bedriften, enn å se på de spesifikke detaljene for et virkelig system. Avgrensningen for dette prosjektet vil dermed være knyttet til installasjonen av testsystemet, og generell forståelse og bruk av Spectrum Scale.

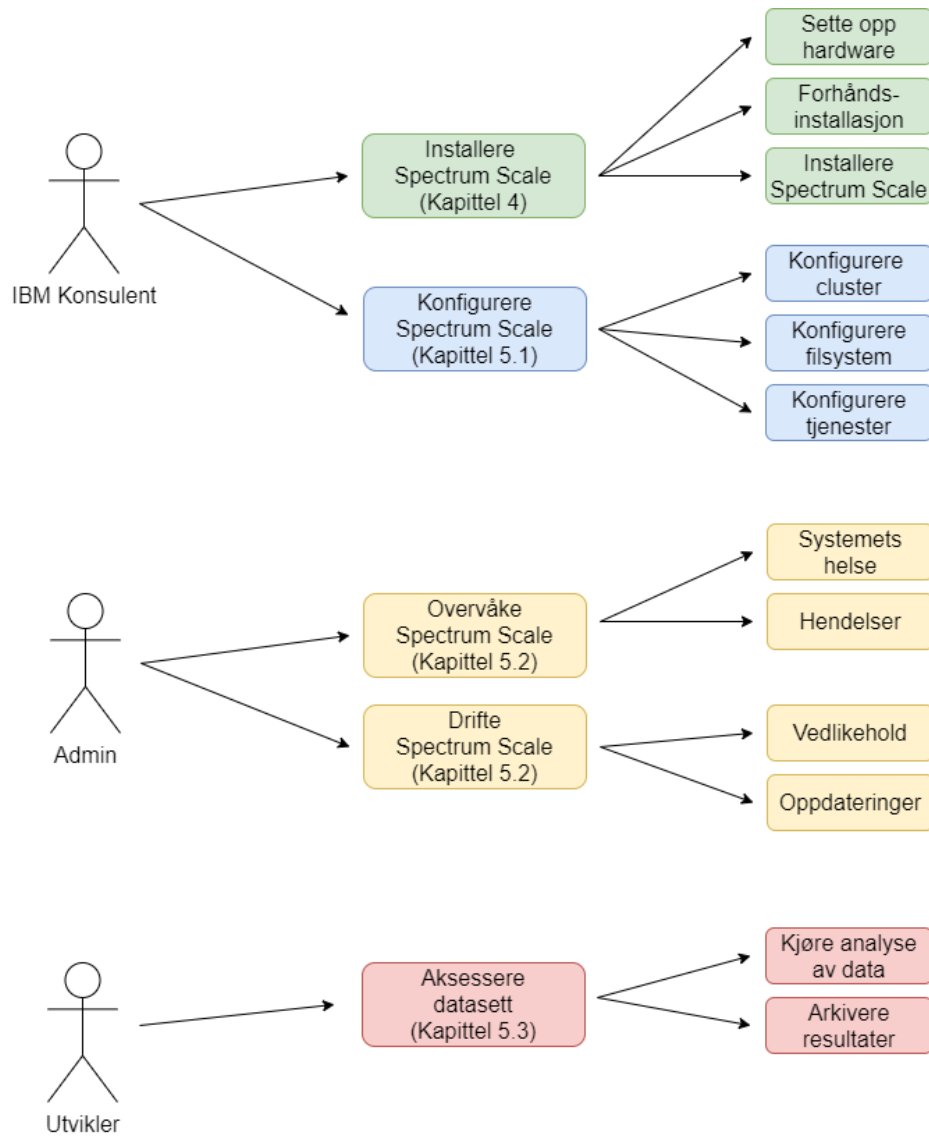
I rapporten dokumenteres produktets funksjoner slik vi mener det er hensiktsmessig for å oppnå målene i bachelorprosjektet. Det finnes flere funksjoner ved programvaren som ikke vil være relevante eller mulige å undersøke for denne oppgaven med tanke på ressursbegrensninger. I tillegg ser prosjektgruppen kun på de parter av systemet som vi har opplysninger om, og som direkte kan knyttes til bachelorprosjektet.

2.4 Rapportens oppbygning

Hensikten med denne rapporten er å dekke de områdene som er beskrevet i kapittel 2.2, som igjen tar utgangspunkt i prosjektets mål. Rapporten er kronologisk inndelt etter hvordan prosessen går fra dokumentasjon til installasjon, og gjør det derfor enkelt for leseren å forstå sammenhengen mellom teori og praksis. Her beskrives innholdet i de ulike kapitlene, og hvilke temaer de tilhører.

2.4.1 Oversikt

Figur 2 viser hvilke oppgaver som inngår i installasjonen, driften og bruk av IBM Spectrum Scale. Her illustreres hvilke kapitler som inneholder informasjon om dette. Videre ser vi på rollene som er oppført på figuren og hvilke ansvarsområder disse har. I kapittel 2.4.3 beskriver vi tydeligere sammenhengen mellom rapportens kapitler og de ulike områdene beskrevet i 2.2.



Figur 2 - Kapittemaer

2.4.2 Roller

Her diskuteres rollene som vises på figur 2.

IBM Konsulent

I et virkelig prosjekt vil installasjon og konfigurasjon mest sannsynlig bli gjort av en sertifisert konsulent fra IBM ut ifra kravene satt i en avtale med bedriften. Konsulentens sin oppgave er å sette opp maskinvare, installere programvare på maskinvare, konfigurere Spectrum Scale og teste at systemet fungerer slik det er tiltenkt. Videre vil konsulentens bidra med support i oppstartsfasen for å sørge for at de som skal drifte systemet tilegner seg den kompetansen de trenger for å drive systemet. Konsulentens kan også kontaktes underveis dersom det trengs ytterligere veiledning.

Admin

Systemadministratoren har ansvaret for daglig drift og vedlikehold av lagringssystemet. Administratoren skal sørge for at systemets helse er optimal, og følge med på eventuelle hendelser i systemet. Selv om systemet vil fungere automatisk mesteparten av tiden, er det viktig å ha en rolle som er forberedt på problemer som kan oppstå underveis.

Utvikler

Denne rollen representerer alle sluttbrukere som skal aksessere systemet, samt lese og skrive data til disk. Sluttbrukeren ser ikke hva som foregår bak kulissene, og forholder seg kun til et vanlig filsystem med den mappestrukturen han ønsker.

2.4.3 Innhold i rapporten

Dette delkapitlet ser nærmere på områdene fra kapittel 2.2, og hvor disse temaene befinner seg i denne rapporten.

Programvarens oppbygning

Oppbyggingen av Spectrum Scale beskrives i kapittel 3. Her ser vi på hvilke prinsipper som gjelder for systemet, og hvilke komponenter som utgjør et såkalt Spectrum Scale Cluster. Dette er en beskrivelse av de ulike funksjonene og komponentene i Spectrum Scale. Videre i rapporten er kapittel 3 viktig for teoretisk forståelse og referanse under installasjon og bruk av programvaren.

Installasjon og konfigurasjon

I kapittel 4 designes testsystemet, før vi går videre med forhåndsinstallasjonen som kreves for bruk av programvaren. Deretter tar vi for oss installasjonen av nodene og den grunnleggende konfigurasjonen som er nødvendig for å få filsystemet til å fungere. Kapitlet dekker dermed temaet om design, installasjon og konfigurasjon. Dette er noe som IBM leverer som en pakke ved kjøp av systemet.

Konfigurasjon og bruk av Spectrum Scale

Kapittel 5 bygger videre på konfigurasjonen av systemet. Denne konfigurasjonen gjøres på et fungerende system og skal tilpasse systemet til kundens behov. Her benytter vi kommandoene og det grafiske brukergrensesnittet til Spectrum Scale. Kapittel 5.1.1 til 5.1.6 utfyller området om installasjon og konfigurasjon, og er leverandørens ansvar.

Kapittel 5.1.7, 5.1.8 og 5.2.2 beskriver bruksområdet om import, håndtering og behandling av data. Dette er en prosess som utføres av konsulentene til IBM, men kan være svært nyttig for IT-administrasjon selv å vite hvordan fungerer for bedre bruk og tilpasning til nye eller endrende behov.

Kapittel 5.2.1, 5.2.3 og 5.2.4 hører til bruksområdet om overvåking av systemets helse, og respons på hendelser. Dette er oppgaver som hører til den daglige driften av systemet og vil håndteres av Kongsbergs IT administrasjon.

Kapittel 5.3 dekker temaet om tilgjengeliggjøring av data for sluttbrukerne av systemet. Dette kapitlet er rettet mot den daglige bruken til systemet. Utviklerne trenger ikke å ha noe kjennskap til Spectrum Scale, utover at de kun har ett filsystem å forholde seg til, og at plassering og behandling av filene foregår gjennom automatiske prosesser.

Konklusjon

Kapittel 6 oppsummerer rapporten i sin helhet. Kapitlet gjentar prosjektets mål, hensikt og plan, og tar for seg resultatene av bachelorprosjektets hovedrapport. Innleveringen vil også inneholde en sluttrapport som går mer i dybden på selve prosessen, og hvordan bachelorprosjektet har gått. Temaer her vil være utfordringer underveis, hva vi har lært og hva vi har oppnådd med prosjektet. Vi ser på graden av måloppnåelse og resultater i forhold til forventninger.

3. Hva er IBM Spectrum Scale?

For å danne en felles forståelse for hvordan systemet er satt sammen, beskrives i dette kapitlet komponentene og funksjonene til IBM Spectrum Scale. Vi ser også på bakgrunnen for slike systemer. Det som står i dette kapitlet er et kondensert utdrag av dokumentasjonen for systemet hentet fra IBM sine nettsider (IBM, 2020) og boka om Spectrum Scale (Quintero, 2015). Kapitlet legger til rette for teknisk forståelse av funksjonaliteten for Spectrum Scale hos leseren.

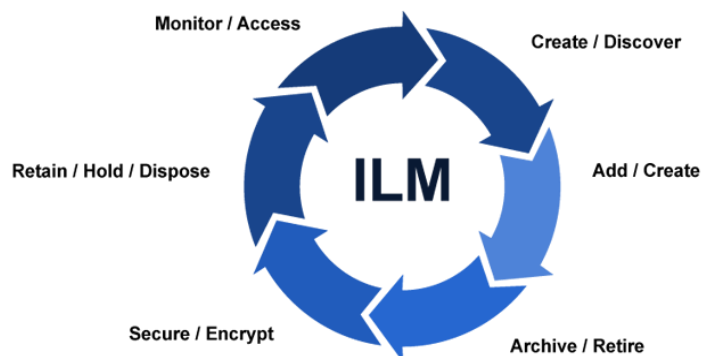
For å hindre forvirring poengteres det aller først at IBM Spectrum Scale for sluttbrukeren fremstår som et «vanlig» filsystem. Faktisk vil hverken applikasjonene på operativsystemet eller brukerne av filsystemet se forskjell på dette og et tradisjonelt filsystem. Men programvaren gjør mye mer under overflaten enn å bare plassere filer i en tre-struktur.

3.1 Bakgrunn for SDS (Software Defined Storage)

Dagens bedrifter, og særlig IT-bedrifter, utfordres av stadig større mengder data. Dette kommer av at mer og mer av teknologien vår digitaliseres og samler store mengder informasjon om omgivelsene rundt seg. For at bedriftene skal kunne dra nytte av dataen og sette den i sammenheng må den lagres et sted i mellomtiden. Problemet er at lagring hverken er gratis eller logisk problemfritt. Store mengder data, eller «Big Data», krever optimaliserte lagringsløsninger som kan hjelpe til med å strukturere og sortere dataen riktig. Dette er essensielt for å holde kostnader, ytelseskrav og kompleksitet under kontroll. I tillegg har ofte større bedrifter mange systemer og enheter med ulikt operativsystem som skal kommunisere med hverandre, noe som krever en felles plattform som kan fungere som et samlepunkt. Systemer som er designet for strategisk styring av data, kalles for «Software Defined Storage», og IBM Spectrum Scale er et eksempel på dette.

3.1.1 Information Lifecycle Management

ILM (Information Lifecycle Management), omfatter den sentrale tankegangen bak en løsning som IBM Spectrum Scale (figur 3). Dette konseptet handler om å styre hvordan data og metadata lagres hele veien gjennom livssyklusen fra den skapes til den ikke trengs lenger, og består av strategier, prosesser og teknologi for å håndtere denne problemstillingen (Rouse, 2005).

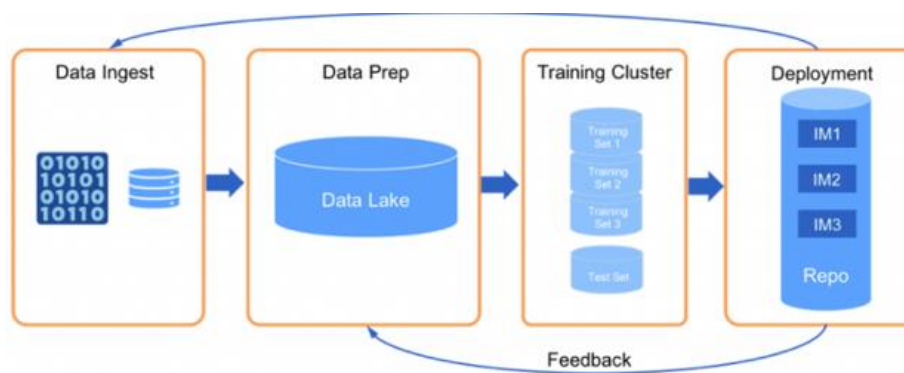


Figur 3 - Information Lifecycle Management

Gjennom proaktiv bruk av systemer for å håndtere informasjon, kan bedrifter redusere kostnader og risiko samtidig som de skaper verdier (Deloitte, 2015). Med stigende mengder data, blir også viktigheten ved god bruk av ILM større. Har vi ikke kontroll på hvilke data som er lagret hvor, og hva slags verdi dataene våre har for bedriften, blir driften av lagringssystemene mindre effektiv, koster mer, og kan i verstefall lede til tap av bedriftskritisk informasjon.

3.2 Software Defined Storage i praksis

Når vi behandler Big Data, er noe av det som tar aller mest tid å behandle og preparere dataene for videre analyse og maskinlæring (se Data Prep på figur 4). Derfor er det viktig, særlig i forbindelse med store datamengder, å ha et filsystem som er raskt, skalerbart, og kan utføre mange operasjoner samtidig.

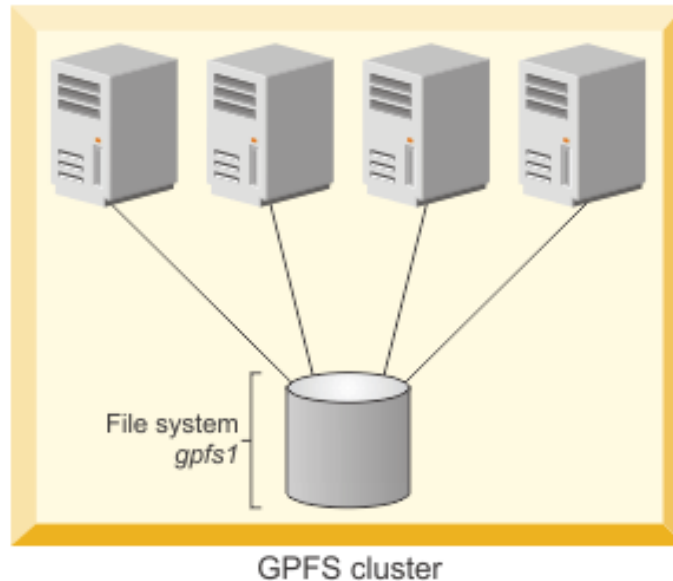


Figur 4 - Systemtrening med data

3.2.1 IBM Spectrum Scale som SDS

Ved hjelp av en eller flere noder arrangert i et *cluster* (se figur 5), installeres Spectrum Scale over flere lagringsenheter i redundant konfigurasjon. Filsystemet kalles for GPFS (General Parallel File System), og Spectrum Scale er bare en nyere betegnelse på dette. Med andre ord samarbeider flere datamaskiner med ulike roller over et designert nettverk for å utføre filsystemets oppgaver og dele data med hverandre. Hver enkelt node får GPFS-programvaren installert lokalt slik at det kan kommunisere med operativsystemet, og kan fungere som en selvstendig tjener for filsystemet. Alle nodene i et cluster har enten direkte eller indirekte tilgang til lagringsmediene som filene befinner seg på.

Dette filsystemet er unikt fordi det distribuerer oppgavene over flere maskiner, og filene strategisk over mange ulike lagringsenheter i et nettverk, noe som sørger for høy ytelse, oppetid, stabilitet og grad av redundans. Programmet er designet for å unngå flaskehalsene vi ser i større sammenhenger i tradisjonelle filsystemer. Dermed blir filsystemet svært skalerbart, og kan håndtere virtuelt sett ubegrensede mengder data. Videre kan man koble på så mange noder man ønsker i clusteret, gjerne noder som skal arbeide med dataene i sine applikasjoner. Dette blir da en «Spectrum Scale Klient» i clusteret. Som vi skal komme tilbake til kan også andre maskiner som ikke har filsystemets komponenter installert få tilgang til filsystemet gjennom egne protokoller.



Figur 5 - GPFS-cluster

3.2.2 Hensikten med IBM Spectrum Scale

Hensikten med Spectrum Scale er å effektivisere, forenkle, og ikke minst spare kostnader ved lagring av store mengder data. Spectrum Scale kan kombinere flash, disk, cloud og tape i ett felles filsystem med høyere ytelse og lavere kostnader enn tradisjonelle systemer. Dette er mulig på grunn av programvarens funksjonalitet og dets skalerbare design. Systemet minimerer kompleksiteten ved å samle alt under ett filsystem, slik at brukeren og applikasjonene slipper å tenke på hva som foregår i bakgrunnen.

3.3 Nøkkelpunkter og funksjoner

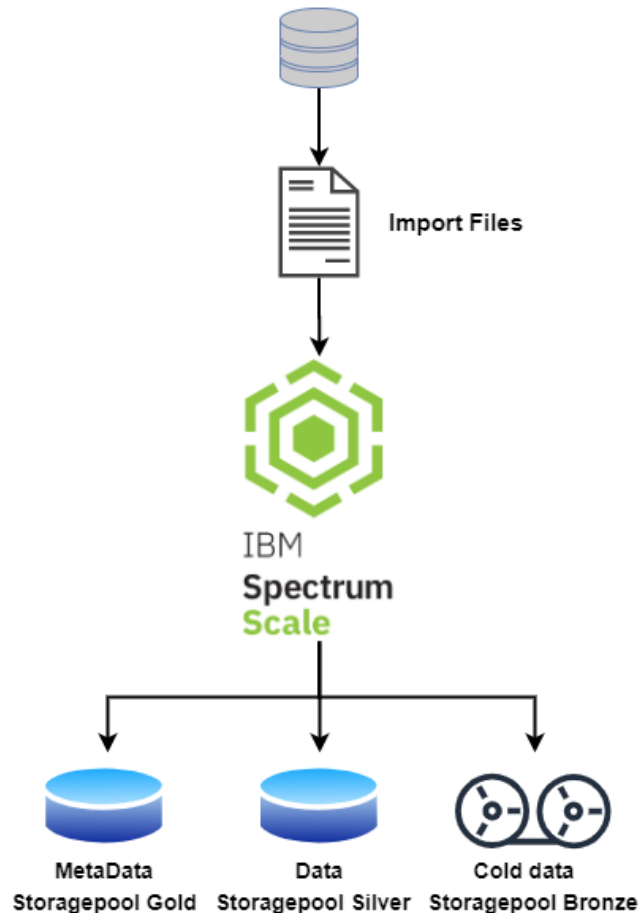
Her ser vi på noen av de viktigste styrkene IBM Spectrum Scale har i forhold til tradisjonelle, lineære filsystemer. Spectrum Scale utgjør mer enn bare et filsystem; Det er intelligent programvare som ved hjelp av noder med ulike roller kan utføre flere funksjoner.

3.3.1 Smart filtrering og migrering av data

Når systemet tar inn nye og bedre datasett mister den gamle dataen verdi. Systemet til Kongsberg Seatex skal kunne håndtere store mengder data fra tidligere og kontinuerlig hente inn ny data. Etter hvert vil systemet fylle seg opp, og dermed må enten kapasiteten utvides eller data slettes. Å utvide kapasiteten med flere disk kan være kostbart når dataen sjeldent skal benyttes. Men å slette data har også en risiko, for selv om dataen har liten verdi nå, kan dataen øke i verdi når nyere verktøy og teknologi blir tilgjengelig, eller hvis man plutselig finner ut at man trenger den likevel. Det er derfor viktig å se på hvordan man mest mulig effektivt kan sortere dataen uten masse ekstra jobb for IT, og kun slette data man er sikker på at man ikke trenger.

IBM Spectrum Scale muliggjør en kombinasjon av flash, disklagring, tape og sky for lagring av data. Dersom man kombinerer disse lagringsmediene på en god måte, vil man kunne spare betydelige mengder ressurser uten store merkbare forskjeller i ytelse. Dette er mulig fordi vi gjennom policyer kan

bestemme hvor dataene skal lagres basert på type, viktighet og tilgjengelighet. Eksempelvis vil vi typisk definere at filer som skal brukes ofte plasseres på raskere disk, og filer som kun skal arkiveres plasseres på tape. Dermed kan vi spare kostnader ved å bruke billigere lagring for arkivering uten merkbar endring i ytelse, da filene vi vanligvis aksesserer er raskt tilgjengelige på dyrere medier. For at dette skal fungere er det viktig at dataen passer med det lagringsmediet den befinner seg på, og disse reglene må derfor settes med omhu i forhold til bedriftens data og hvordan de ønsker at den skal sorteres. Å ha dette godt planlagt på forhånd vil gjøre denne jobben mye lettere, og redusere risiko for feiltrinn. I kapittel 5 ser vi konkret på hvordan vi konfigurerer regler for å håndtere data i systemet i Spectrum Scale.



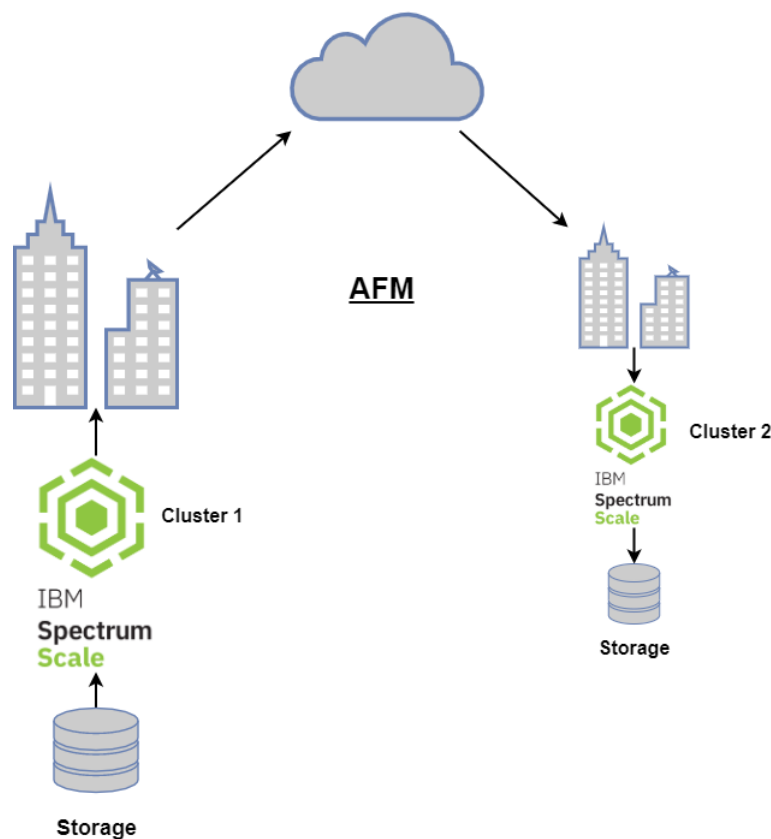
Figur 3 - Importering av filer

GUI-tjenesten til Spectrum Scale har en del innebygde kriterier for håndtering av data. Det går også an å spesifisere og importere sine egne avanserte regler for hvordan ulike datatyper skal gjenkjennes og hva som skal skje med dem. Dette defineres i egne skript. Vi skiller her mellom «file placement policies» og «file management policies». «File placement policies» bestemmer hvordan nye filer som opprettes skal lagres, mens «file management policies» styrer dataen gjennom livssyklusen. Når ny data plasseres inn i

filsystemet, vil disse automatisk plasseres der vi ønsker å ha dem basert på disse to typene av regler vi har satt i Spectrum Scale.

3.3.2 Active File Management

Spectrum Scale er designet for å kunne brukes på kryss av geografiske lokasjoner og ulike plattformer med høy ytelse og responstid, selv med dårlig internettforbindelse. Dette gjøres mulig med en funksjon som kalles AFM (Active File Management). Denne funksjonen bruker intelligent caching for å gjøre filsystemet tilgjengelig over internett, og minimere forsinkelser. Dette betyr at deler av filsystemet mellomlagres på de ulike lokasjonene for å klargjøre for lesing og skriving. En kan dermed sette opp filsystemer på kryss av lokasjoner med hele eller deler av de delte filsettene lokalt tilgjengelig. Filsystemet vil da oppleves som om det befant seg lokalt i nettverket. Dette er gunstig både for samarbeid mellom avdelinger, og for sikring av data og høy oppetid på filsystemet. Under 3.5.4 står det også litt om hvordan slik replikasjon kan brukes som en form for backup. Dersom filsystemet går ned på en lokasjon, kan man «feile over» til instansen som befinner seg på en annen lokasjon, og dermed opprettholde drift som normalt.



Figur 4 - Active File Management

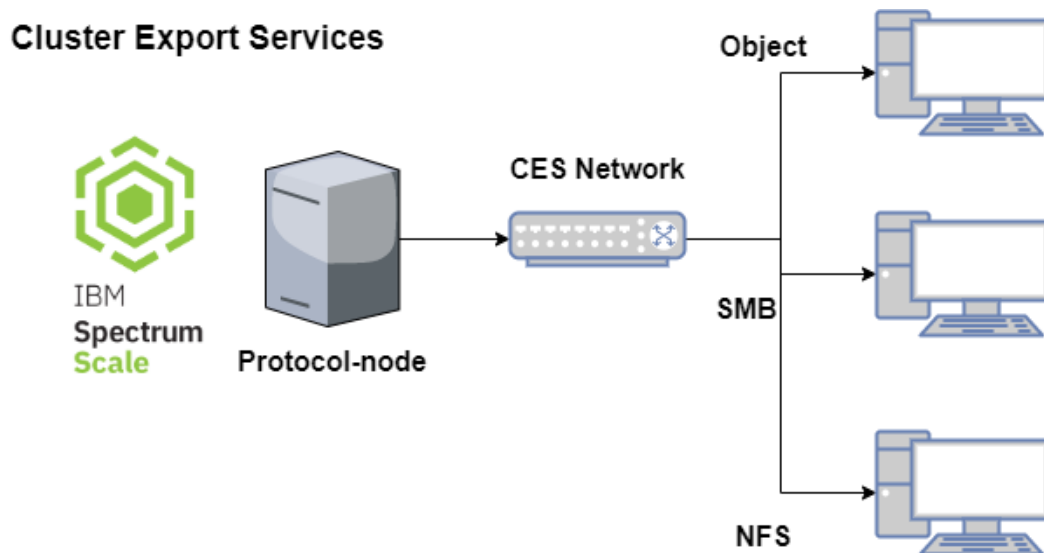
En av de største fordelene dette gir er å kunne ha lagring og dataprosessering fysisk nærmere hverandre på flere lokasjoner. Lagring og dataprosessering ønsker du å ha nært for å unngå en flaskehals og

beslaglegning av båndbredde i nettverket ved store datamengder. For Kongsberg kan dette for eksempel være et aktuelt samarbeidsprosjekt mellom flere avdelinger, der en større Spectrum Scale løsning kan distribueres mellom de ulike lokasjonene.

3.3.3 Cluster Export Services

Ved bruk av CES (Cluster Export Services), kan man dele filsystemet over NFS-, SMB- og Object-protokollene, noe som er gunstig for å samlokalisere data på tvers av ulike plattformer. Man setter da opp egne noder i clusteret som protokollnoder, som dermed brukes for å viderefremde disse protokollene. I tillegg er det mulighet for å koble systemet opp imot et eksisterende Hadoop-filsystem. Dermed er man ikke avhengig av at alle klienter som skal nå filene i GPFS-filsystemet også har GPFS installert lokalt. En slik kryssplattform løsning tar hånd om et vanlig problem i større bedrifter som oppstår når ulike avdelinger bruker forskjellig utstyr.

Eksempelvis blir det lettere å koble maskiner med ulike operativsystemer og programvare på det samme filsystemet. Alt som kreves av klientmaskinene er at de også har NFS installert, og at de settes opp som klienter mot clusteret over et nettverk. Det delte filsystemet vil normalt hete «ces_shared».



Figur 5 - Cluster Export Services

3.3.4 Stabilitet og høy oppetid

IBM Spectrum Scale sitt design sikrer høy grad av redundans og unngår «single point of failure». Med andre ord finnes det ingen enkeltledd i systemet som resten av komponentene avhenger av. Dette oppnås ved bruk av flere enheter med mulighet til å utføre de samme oppgavene i alle ledd av systemet. Eksempelvis vil et SAN koblet til et GPFS-cluster gi mulighet for flere fysiske veier til det samme målet om en enhet i nettverket skulle feile eller trenge vedlikehold. Dette er også årsaken til at man benytter flere noder i et cluster som samarbeider om de samme oppgavene. Dersom en node feiler, vil de andre nodene fortsette driften av clusteret.

Selve filsystemet er designet med dette konseptet i tankene, og sprer blant annet dataen systematisk over flere diskene i RAID, noe som sikrer mot tap og nedetid av data. Om en disk skulle feile, vil man fortsatt kunne aksessere dataene på grunn av dette, og filsystemet kan reparere seg selv basert på «striping» mellom diskene. Dette prinsippet kan man justere etter behov for oppetid og sikring av data. Om man forvalter sensitiv og kritisk informasjon, vil kravene for integritet og oppetid i systemet øke. Dette kan dermed justeres både i designet av clusteret og i parametere i programvaren (Taylor, 2019).

3.3.5 Skalerbarhet og høy ytelse

Filsystemet er designet med skalerbarhet og høy ytelse i tankene. Filsystemet er kostnadseffektivt når det kommer til skalering, da man ved hjelp av policyer og bruk av ILM sørger for å ikke sløse med lagringskapasitet og bruk av dyrere medier. Det logiske designet av filsystemet tillater et svært høyt antall filer og størrelse, noe som gir virtuelt sett ubegrenset med lagringskapasitet.

At filsystemet er parallelt betyr at det kan håndtere mange ulike forespørsler på en gang. Dataene som er lagret på diskene er spredt utover mellom dem slik at flere klienter kan få tilgang til data samtidig. Dette er en av styrkene til et parallelt filsystem sett opp imot et tradisjonelt lineært filsystem, da et tradisjonelt filsystem gjerne bare har en enkelt funksjon, og kun kan håndtere ett forespørsel om gangen. Filsystemet bruker flere metoder for å øke ytelsen, for eksempel analyse av databruken, og adaptasjon deretter for å maksimere responstiden. Ved å lagre data som cache både i minne og på raskere lagringsmedier som SSD, øker responstiden ved aksessering av filsystemet.

3.3.6 Enkel administrasjon

Gjennom et grafisk brukergrensesnitt får man tilgang til hele filsystemet på en oversiktlig måte. Her kan man monitorere systemets komponenter, generere statistikk og utføre diverse administrative oppgaver. Dette kan også gjøres gjennom kommandoskallet til nodene i clusteret. Disse funksjonene er eksempelvis å legge til eller slette diskene, opprette filsystemer og filsett, og å konfigurere noder i clusteret. Mange av disse funksjonene vil vi demonstrere i kapittel 5.

3.4 Komponenter

Dette underkapitlet går lenger ned i dybden på hvordan filsystemet fungerer. Her finner du en oversikt over programmets komponenter som skal gi leseren forståelse for hva som foregår «under panseret» i en Spectrum Scale løsning. Naturligvis er det mye mer en kunne sagt om dette, men vi prøver å holde det kort og konsist.

3.4.1 Programvaren

IBM Spectrum Scale består av tre hovedkomponenter: En multitråd daemon-prosess, administrative kommandoer, og en kjerneutvidelse. På figur 9 ser du hvordan disse komponentene befinner seg på en Spectrum Scale node i forhold til lagmodellen for operativsystemet.

Daemon-prosess

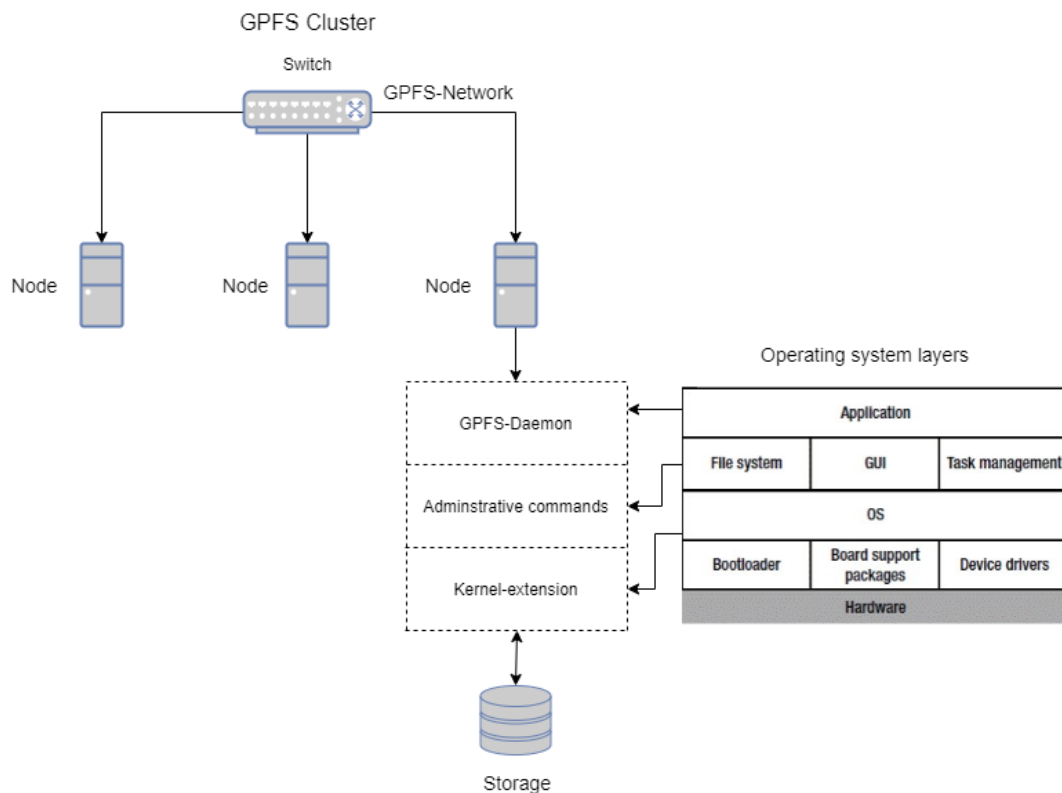
Dette er en prosess som kjører på hver node. Den har flere oppgaver, men i hovedsak sørger den for kommunikasjon og oppdatering av konfigurasjon med de andre nodene i clusteret. For den lokale noden styrer daemon-prosessen I/O aktivitet og buffering, som igjen styres av fil-systemmanageren i clusteret. Mer om noderes roller finner du i kapittel 3.4.3.

Administrative kommandoer

De administrative kommandoene inkluderes i noderes kommandoskall og brukes for å administrere og overvåke GPFS-clusteret. Hver node har myndighet til å utføre oppgaver, og vil bruke et dedikert GPFS-nettverk for kommunikasjon med de andre nodene. Senere vil vi demonstrere hvordan noen av disse kommandoene kan brukes for å administrere filsystemet.

Kjerneutvidelse

For hvert operativsystem bygges det en unik kjerneutvidelse som muliggjør sømløs kommunikasjon mellom applikasjoner og GPFS-filsystemet. Når en applikasjon på datamaskinen sender systemkall til operativsystemet, for eksempel for å lese en fil, blir den henvist til kjerneutvidelsen som tar hånd om forespørselen. Kjerneutvidelsen leter først i minnet etter filen, og deretter på disk. Slik fremstår filsystemet som et vanlig filsystem, og applikasjonene kan operere som normalt. Man kan enkelt legge til nye klienter for Spectrum Scale med GPFS-funksjonalitet for å kjøre dataen gjennom de applikasjonene man bruker. Filsystemet fremstår som lokalt for klientmaskinene, og har lav responstid fordi dataen sendes over et høyhastighets nettverk mellom klienten og den maskinen som har direkte tilgang til disken (NSD Server).

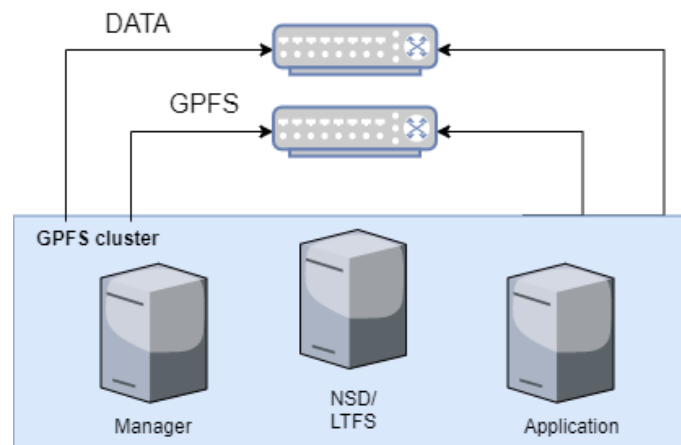


Figur 6 - Nodeoppbygning

3.4.2 GPFS-clusteret

GPFS-clusteret er bærebjelken i et Spectrum Scale filsystem. Det er sammensetningen av maskiner som samarbeider og kommuniserer for å drive filsystemet. I et GPFS-cluster har vi én eller flere noder som samarbeider med å utføre GPFS-kommandoer (se figur 10). En node kan betegnes som en selvstendig maskin i nettverket, og disse kommuniserer over et dedikert GPFS-nettverk. Nodene deler også data med hverandre over et data-nettverk til den noden som spør om tilgang. Disse nettverkene skilles fra hverandre for å hindre flaskehals.

Hver av disse nodene får GPFS-programvaren installert når de velges under installasjonen. På figur 5 ser du en abstraksjon av hvordan programvaren fungerer på nodene i henhold til lagmodellen for operativsystemet. En såkalt GPFS-daemon er igjen prosessen som kjører på operativsystemet, og kommuniserer med instansene som befinner seg på de andre nodene. Det installeres som nevnt også kjernemoduler og egne administrative kommandoer slik at programvaren kan kommunisere med operativsystemet gjennom ulike kall. Dermed vil applikasjoner som kjører på maskinen kunne aksessere filsystemet som om det var et «vanlig» filsystem.



Figur 7 - Cluster

3.4.3 Nodenes roller

En node i GPFS-clusteret kan ha flere roller som bestemmes ved installasjon og ved opprettelse av nye noder. En rolle har ulike oppgaver og funksjoner i clusteret, og det er viktig å velge tildelingen av roller med omhu for å oppnå optimal ytelse og stabilitet.

Cluster-manager

Det finnes én cluster-manager for hvert cluster. Denne har i oppgave å monitorere diskbruk blant noder, oppdage og rette feil på nodene, distribuere konfigurasjoner, velge filesystem-manager og overvåke clusterets helse. Cluster-manageren velges automatisk fra samlingen quorum-noder.

Filesystem-Manager

I hvert cluster har man én «filesystem-manager» som har hovedansvaret for å koordinere de ulike nodenes tilgang til filsystemet. Dette skjer ved hjelp av et kø-system der hver enkelt node som ønsker tilgang til en fil først må henvise seg til file-system manageren. Dermed gis det som kalles en «token»

eller en billett ut til noden som ønsker tilgang, og den kan dermed aksessere filsystemet frem til noen andre ønsker tilgang. Om flere ønsker tilgang til samme fil samtidig settes det opp en kø som filesystem-manageren håndterer. Denne rollen gjør også andre oppgaver som å konfigurere og reparere filsystemet.

File-system manageren velges av cluster-manageren, og rollen vil overføres til en annen node dersom filesystem-manageren skulle feile. Slik kan operasjonen av filsystemet fortsette uhindret. En bør unngå å kjøre intensive applikasjoner på den samme noden som har denne rollen.

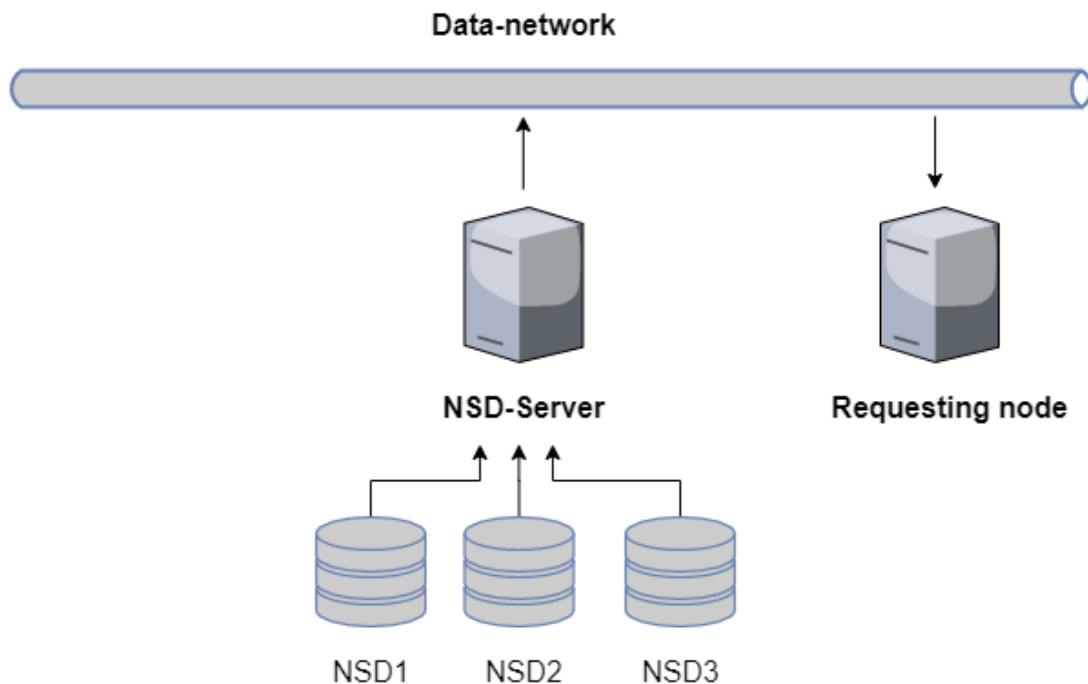
Admin node

En admin-node sørger for å hjelpe til med konfigurasjon og installasjon av komponenter i clusteret. Dette kan for eksempel være når vi legger til en ny NSD-server eller en disk.

NSD-server

NSD står for "Network Shared Disk", og er en disk som deles over datanettverket av en NSD-server. Hensikten med en NSD-server er å gi tilgang til disk som andre noder selv ikke har direkte tilgang til. Noden spør da NSD-serveren om tilgang til disken. Det er anbefalt å definere NSD-servere som backup dersom en node skulle miste sin fysiske tilgang til en disk. Dermed kan noden gå via NSD-serveren i stedet.

For at en node skal bli NSD-server må den konfigureres med denne rollen, og ha definert en eller flere disk som den har direkte tilkobling til som NSD. Dette gjøres enten under installasjonen av GPFS eller senere ved hjelp av egne kommandoer. Alle NSD-servere må ha tilgang til de samme NSD-ene i clusteret.



Figur 8 - NSD Nettverk

GUI-Server

En kan definere opptil flere GUI-servere i clusteret. På disse nodene installeres en webserver som man kan koble til for å overvåke filsystemet gjennom et grafisk grensesnitt. Dette gjør man gjennom nettleseren ved å bruke serverens adresse i adressefeltet. Det anbefales å etablere 2-3 slike for å sikre høy oppetid på GUI-tjenesten. En node som er GUI-server må også være admin-node.

Protocol node

I et GPFS-cluster kan man definere flere protokoll-noder. En protokoll-node er en maskin som viderefremidler filsystemet ved bruk av CES, og deler dem over protokollene NFS, SMB og Object. I praksis betyr dette at maskiner som ikke har GPFS installert kan få tilgang til GPFS-filsystemet ved å benytte seg av en av disse protokollene, og henvende seg som klienter til protokollnoden. Filsettene som er satt til å deles over protokollene vil da bli tilgjengelig for disse maskinene.

Quorum node

Quorum er en betegnelse på samlingen «mandater» i clusteret. Majoriteten av mandatene som kreves for Quorum må være til stede for at filsystemet skal være operativt. Hensikten med å implementere quorum er å beskytte integriteten til filsystemet, og hindre korrupsjon. Om nok av disse nodene blir utilgjengelige avmonteres filsystemet inntil videre. Eksempelvis må minst 3 av 5 quorum noder være funksjonelle for at driften av GPFS skal fortsette. Alternativet til Quorum er å bruke en eller flere «tiebreaker»-disker. Disse diskene må til enhver tid være koblet til minst en av nodene for at systemet skal operere. Det anbefales ikke å bruke flere enn 8 quorum-noder, og de bør helst velges i oddetall.

3.5 Spectrum Scale Lagring

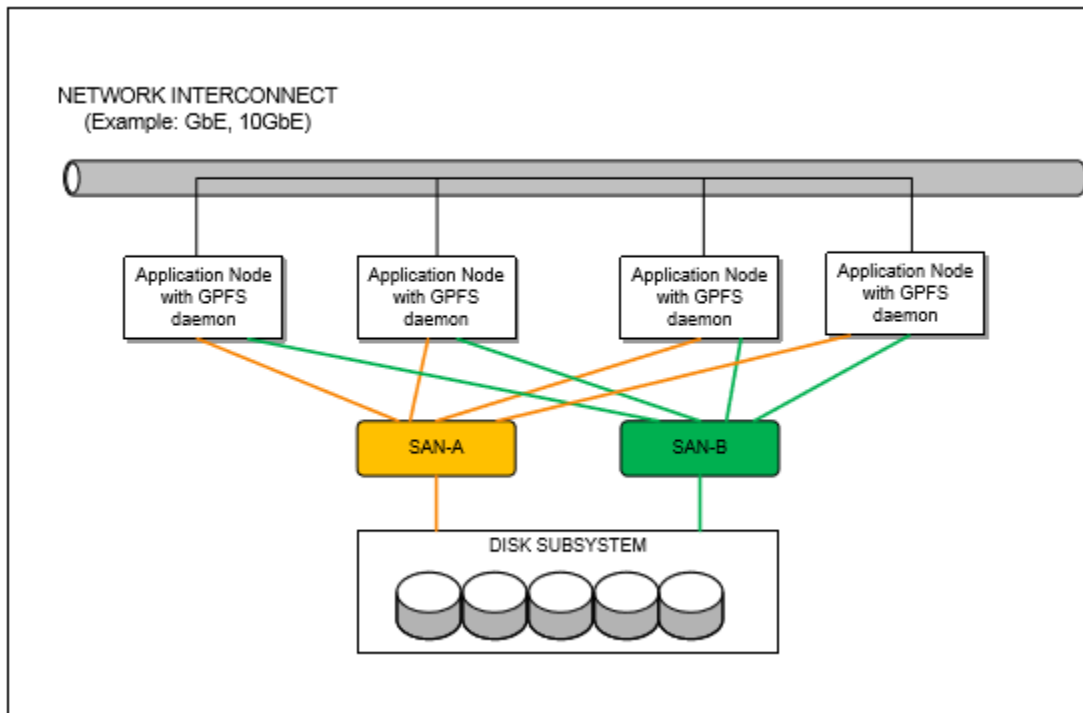
3.5.1 Lagringsmedier

I et Spectrum Scale cluster bruker man gjerne en blanding av disk, flash og tape til å lagre data. Diskene og flashdiskene er typisk koblet rett til nodene i clusteret, mens tape håndteres av et eget system i et eksternt storage pool (se 3.5.2 og 3.5.5). Typisk lagres data som brukes ofte på diskene, og data som skal arkiveres på tape, da tape har lengre responstid og tregere skrive- og lese hastigheter.

Topologi

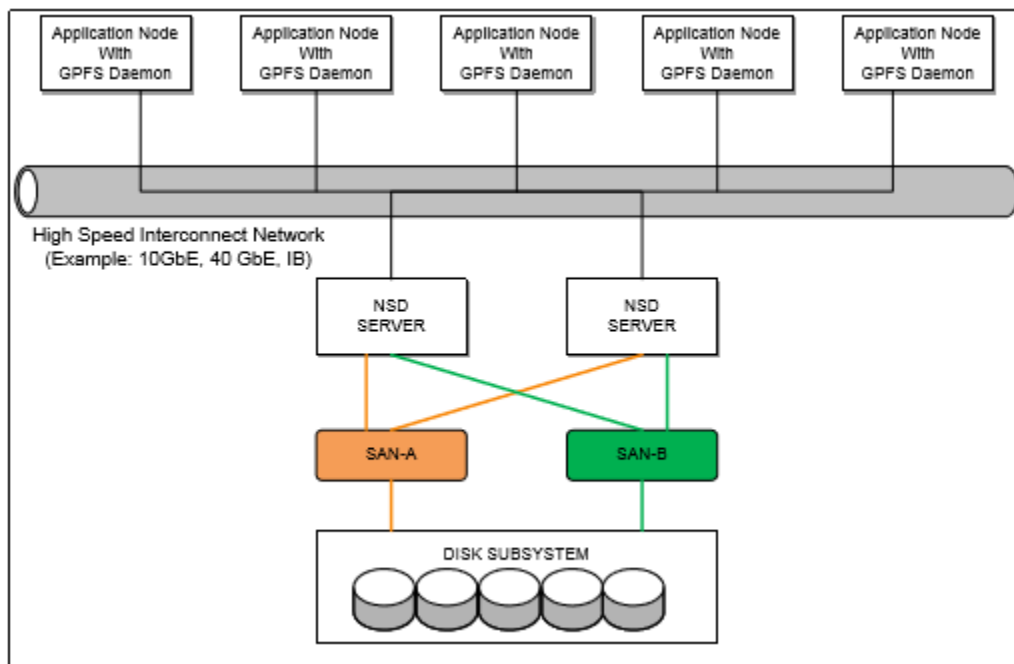
Avhengig av clustertopologien vil diskene enten være koblet direkte på alle nodene, indirekte til nodene gjennom en NSD-server, eller en blanding av disse to. En disk som deles gjennom en NSD-server kalles for en NSD, altså en «Network Shared Disk». (Se 3.4.3 for NSD-server rollen).

Her ser vi at alle nodene har direkte tilkobling til diskene gjennom to SAN:



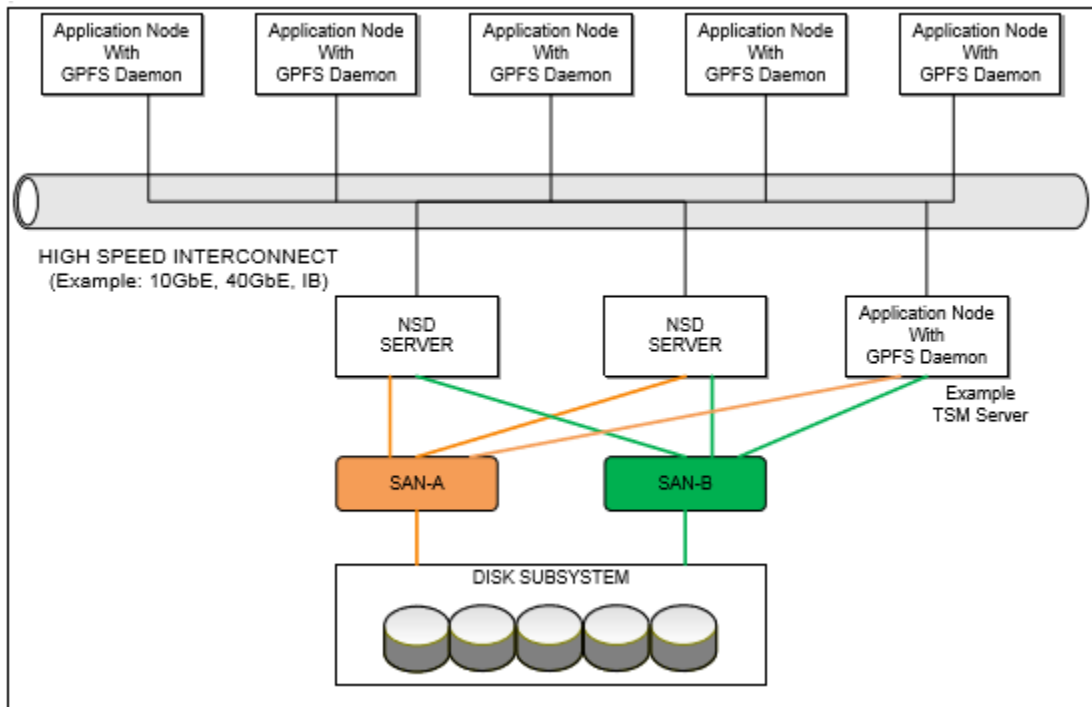
Figur 9 - Clusterkonfigurasjon 1

Under ser du en konfigurasjon der alle diskene er koblet til NSD-serverne gjennom to SAN. Videre deler NSD-serverne dataene med nodene i nettverket. Dette krever at nettverket har høy nok båndbredde for å dekke krav til ytelse.



Figur 10 - Clusterkonfigurasjon 2

Den tredje konfigurasjonen er som sagt en blanding av disse to typene:



Figur 11 - Clusterkonfigurasjon 3

3.5.2 Storage pools

Spectrum Scale benytter noe som kalles storage pools for å lagre data. Et storage pool fysisk sett er en samling diskere eller andre lagringsmedier, typisk i et RAID-oppsett. Man grupperer disse lagringsmediene basert på egenskaper som ytelse, plassering og pålitelighet. Et storage pool kan være enterprise-grade diskere, mens et annet kan være billige harddiskere i store kvantum. Slik lager man ulike nivåer av lagring som tilpasses den dataen som skal lagres der. De fysiske diskene legges til i et storage pool i Spectrum Scale, og brukes for å gruppere data. Vi kan ta i bruk storage pools til en rekke oppgaver, slik som replikering av data og speiling. I tillegg vil samsvar mellom kvaliteten på dataen og lagringsmediet føre til økt ytelse og sparte ressurser.

Det finnes to typer storage pools, interne og eksterne:

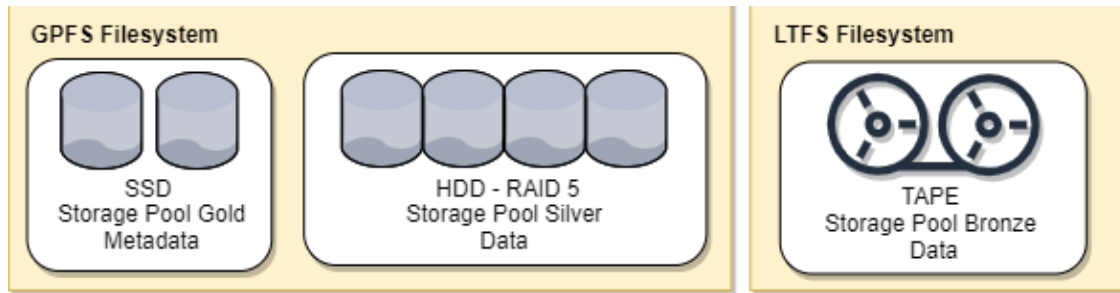
Interne storage pools

Interne storage pools er de områdene som styres av GPFS-filsystemet. I disse lagringsmiljøene lagres data som skal aksesseres av filsystemet jevnlig. Vi skiller her mellom det obligatoriske system storage poolet, og valgfrie user storage pools. Lagringsmiljøet for systemet lagrer både data og metadata, mens user storage pool kun inneholder brukerdata.

Eksterne storage pools

Eksterne storage pools styres av en ekstern applikasjon. Med andre ord styrer ikke GPFS noe annet enn å flytte dataen til det eksterne lagringsmiljøet. Deretter tar den eksterne applikasjonen, eksempelvis

IBM Tivoli Storage Manager over for datahåndteringen. Et slikt lagringsmiljø brukes som oftest til arkivering og «offline»-lagring.



Figur 12 - Storage pool nivåer

3.5.3 Failure groups

En «failure group» er betegnelsen på lagringsenheter som deler de samme kritiske punktene, altså punkter som ved feil kan gjøre lagringen utilgjengelig. Dette kan for eksempel være diskene på samme strømforsyning, i samme rom, eller på samme lokasjon. Hvordan man definerer disse grensene avhenger av hvor stor spredning man ønsker mellom disse gruppene med tanke på sikkerhet. Hensikten med å definere slike grupper er å gjøre programmet oppmerksomt på hvor den skal spre kopier av data for replikering (se 3.5.4). Spectrum Scale sørger for å ha en kopi av både metadata og data i hver failure group slik at operasjonen av filsystemet kan fortsette selv om en hel gruppe går ned. Hvilken failure group en disk tilhører konfigurerer man når man kobler en ny disk til clusteret.

3.5.4 Backup

Alle bedrifter som lagrer kritisk informasjon, med andre ord de fleste bedrifter nå til dags, må ha en eller annen form for backupløsning. Uten backup er man ekstremt sårbar for uhell, katastrofer og hærverk, og kan fort lede til store tap for virksomheten. Avhengig av krav til sikring av data og personvern må bedriften vurdere hvilke tiltak som passer for dem. Vi vet ikke hvordan Kongsberg Seatex håndterer denne problemstillingen, men vi vet at de trenger et system for å sikre dataene sine. Her ser vi på noen av alternativene for backup, og hvordan dette kan brukes med IBM Spectrum Scale.

Replikering

Som du leste om i forrige delkapittel bruker Spectrum Scale noe som kalles failure groups for å avgjøre spredningen av data mellom ulike lagringsgrupper. En failure group sier som sagt noe om hvilke lagringsenheter som ligger nært hverandre når det kommer til feil, slik som strømforsyning, lokasjon og liknende. Ved opprettelsen av et filsystem kan vi ta i bruk en funksjon som kalles replikering, som betyr at filsystemet sprer kopier av filene imellom diskene. På denne måten øker man sikringen mot tap av data og nedetid på filsystemet dersom uhell skulle inntreffe. Spectrum Scale vil automatisk håndtere en feilsituasjon ved å ta i bruk de replikerte filene om det skulle oppstå feil, og varsle om dette.

Maksimum replikasjonsfaktor er 3, noe som betyr at filsystemet oppbevarer 3 kopier av de definerte filene som skal replikeres. Det som er viktig å ta hensyn til ved replikering er at dette tar opp ekstra plass på diskene, og sinker hastigheten på grunn av økt skriveid. Det er derfor kun ønskelig å bruke dette dersom det har et bestemt formål.

Bruk av remote cluster

Under 3.3.2 kan du lese om «Active File Management». Spectrum Scale muliggjør samarbeid mellom flere filsystemer på ulike geografiske lokasjoner. Dette kan brukes til flere ting, blant annet å dele ulike filsett imellom avdelinger i bedriften for å forbedre ytelse. Men en slik løsning kan også brukes som «off-site» backup ved at filene befinner seg på flere fysiske lokasjoner samtidig. Dermed sikrer vi oss mot trusler som brann, flom, tyveri og annen skade. Dette sørger for å sikre mot tap av data, og at filsystemet kan fortsette driften selv om en gruppe med disketter skulle gå ned.

IBM Spectrum Protect

Gjennom programvaren IBM Spectrum Protect tilbyr IBM en egen løsning for backup og arkivering av filer, da Spectrum Scale ikke har denne funksjonen. Dette er også en del av programvareporteføljen Spectrum Storage. Kort sagt er Spectrum Protect en plattform for sikker og skalerbar backup, og kan bruke flere typer lagringsmedier for dette. Fordelen med å bruke dette vil være økt grad av kompatibilitet og integritet i filsystemet, og support fra IBM.

Tredjeparts programvare

Det finnes mange løsninger for å håndtere backup som ikke er knyttet til Spectrum Scale, både open-source og betalte programmer. Dette kan være et rimelig alternativ dersom man ikke krever fordelene ved å bruke samme leverandør til flere tjenester. I praksis vil man da velge de delene av filsystemet som er interessante for backup, og kopiere disse til ønsket medium via backup-programmet.

Snapshot

Spectrum Scale har innebygd funksjonalitet for å ta snapshots av filsystemet. Et snapshot er et tilstandsbilde som kan brukes for å gjenopprette filsystemet slik det var før snapshotet ble tatt. Denne funksjonen er god for enkel backup og sikkerhetskopiering dersom noe skulle gå galt ved testing eller oppdatering. Likevel kan ikke snapshot ansees som en fullverdig backup eller kopi fordi det avhenger av at disken som ble kopiert fungerer og er tilgjengelig når vi skal gjenopprette tilstanden.

Backup med NFS

Som beskrevet i 3.3.3 om «Cluster Export Service», legger Spectrum Scale opp til å eksportere filsystemet gjennom ulike protokoller. Dersom man har satt opp en klient-serverløsning for backup av filsystemet, kan man bruke NFS eller en av de andre protokollene til å eksportere de filene man ønsker å ta backup av, ved bruk av CES.

3.5.5 IBM Spectrum Archive

IBM Spectrum Archive er en del av IBMs lagringsportefølje, IBM Spectrum Storage (Miyoshi, 2020). Tidligere var Spectrum Archive kjent som Linear Tape File System (LTFS), noe som fortsatt brukes i teknisk sammenheng. Programvaren jobber mot et tape basert filsystem og tilbyr tilgang for lesing og skriving mot blokker i det lineære tape-filsystemet. Målet med å benytte et filsystem på tape er å spare lagringskostnader, da tape er billigere, mindre ressurskrevende og har lengre levetid enn disketter.

IBM Spectrum Archive Enterprise Edition er integrert mot IBM Spectrum Scale og vil tilby clusteret mulighet til å migrere filer mellom disk og tape. Tapeløsningen vil da brukes som et eksternt storage

pool i filsystemet. Da vi ikke har tilgang på denne programvaren, vil vi ikke se nærmere på dette i prosjektet. I testkonfigurasjonen vil vi i stedet bruke variasjoner av diskere for å representere de ulike storage poolene, og hvordan data kan flyttes mellom disse.

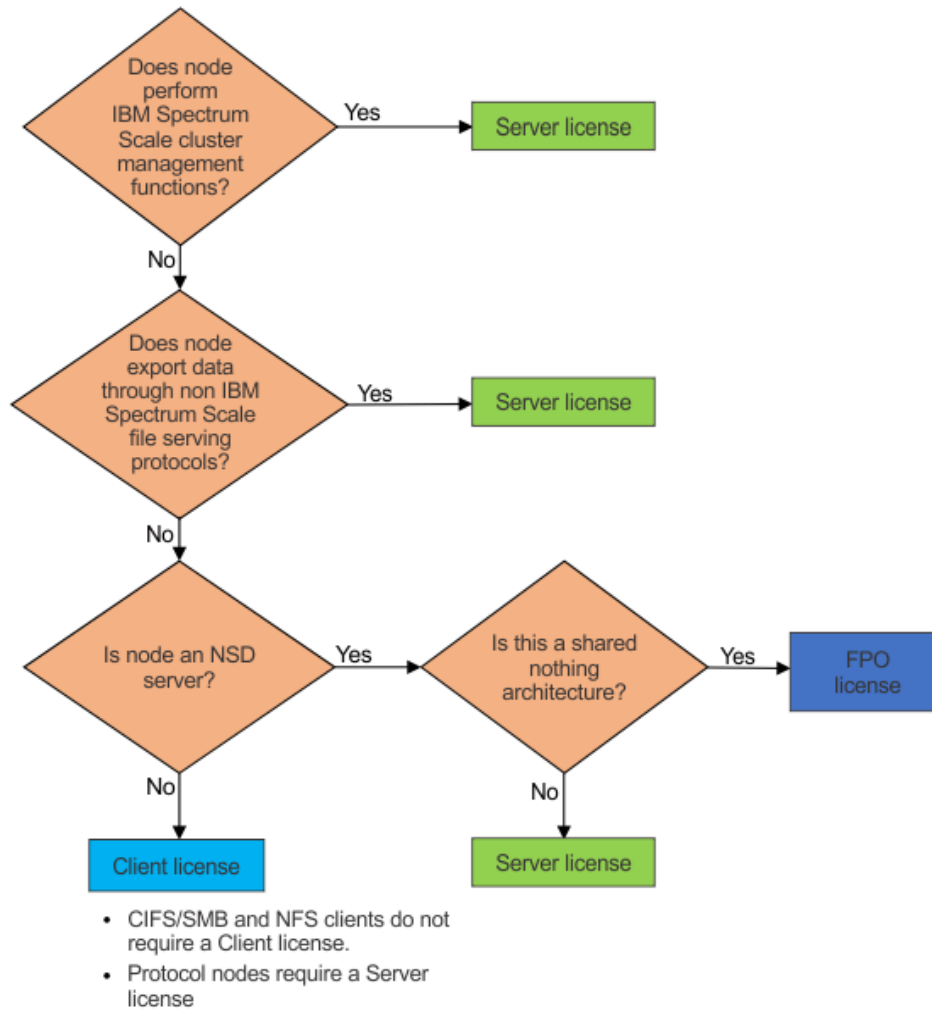
3.6 Lisensiering

Hvilken lisens som kreves for å drive et cluster bestemmes av hvilken utgave av programmet en skal bruke og hva slags funksjonalitet som brukes på den enkelte noden. Det kreves en lisens for hver node i clusteret som skal brukes med IBM Spectrum Scale. Dersom Spectrum Scale leveres i en fullstendig løsning fra IBM, vil alt av kostnader, inkludert lisenser for ubegrenset bruk, være inkludert i prisen. Dette betyr at man kan legge til så mange klienter man vil, og legge til så mye diskplass man ønsker. I vårt tilfelle kreves ingen spesiell lisens, da programvaren brukes i et testmiljø.

Ved installasjon på egen maskinvare som skal brukes i produksjon må en skaffe lisenser på egenhånd. Lisenskostnadene kan avhengig av lisenstypen bestemmes enten på basis av prosessorbruk eller diskbruk. Figur 16 viser et kart som skal hjelpe brukeren med å forstå lisenskravene for Spectrum Scale. Som du ser, avhenger valget av lisens av hvilke oppgaver en node skal benyttes for i clusteret.

Lisensen kan enten gjelde for lagringskapasitet eller for prosessorkraft (kjerner). Lisens basert på lagringskapasitet gjelder for IBM Spectrum Scale Data Access Edition, Data Management Edition og Erasure Code Edition. Dersom man virtualiserer noden vil lisensen gå på antall CPU-kjerner som er tilgjengelig for Spectrum Scale gjennom host-maskinen.

Applikasjonsnoder som kun skal aksessere dataene tildeles en klient-lisens. Dersom denne skal bruke andre GPFS-funksjoner enn kun å aksessere data internt i clusteret, eksempelvis ved å fungere som NSD-server eller eksportør av data vil denne kreve en server-lisens i stedet.



Figur 13 - Valgtre for lisensiering

3.7 Begrensninger

Det skal poengteres at IBM Spectrum Scale ikke er en alt-mulig løsning. Selv om prosjektet handler mest om å se på mulighetene og funksjonene til programmet, er det også viktig å balansere bildet litt ved å se på noen av utfordringene, begrensningene og ulempene ved å ta i bruk et slik system.

Kostnader

Prislappen på et fullverdig Spectrum Scale system, og særlig hvis det brukes i kombinasjon med IBMs andre produkter, vil være relativt høy. Å kjøpe inn servere, hardware, disk og programvare er ingen billig investering, og bedriften må være sikre på at dette er hensiktsmessig. Gjennom grundige kost-nytte analyser, må bedriften avgjøre om de økonomiske utgiftene kan rettferdiggjøres av nytten til prosjektet. Ut ifra vår oppfatning passer et slikt system best for store bedrifter med høy omsetning som behandler enorme datamengder. Dersom dette ikke er tilfellet, kommer man langt med enklere og mer tradisjonelle løsninger, slik som NAS, filtjenere og skytjenester.

Skala

Knyttet opp imot kostnader for å innføre et system har vi også omfanget og kompleksiteten av selve prosjektet. Å innføre et nytt lagringssystem vil for de fleste bedrifter kreve en omfattende formalisert prosess, noe som krever mye ressurser. Å gjennomføre et slikt prosjekt vil naturligvis også føre med seg en del risiko, og kan hemme bedriften i en periode. For et lite firma med enkle formål, vil det derfor kanskje lønne seg å benytte et mindre avansert system. Med mindre man vil være fullstendig avhengige av IBM for drift, vil man også trenge en god del kompetanse om slike filsystemer for å kunne drive dem og gjennomføre servicerutiner.

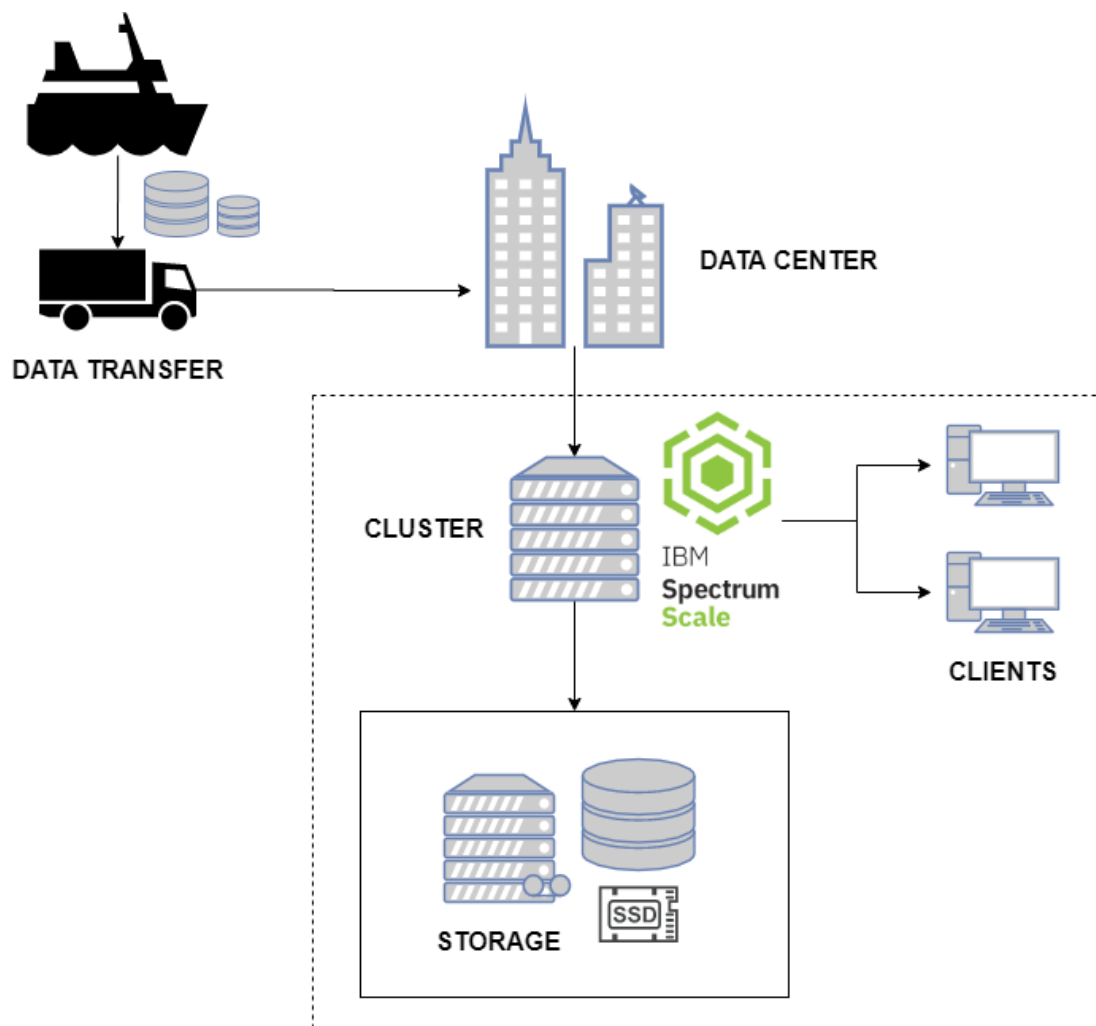
Funksjonalitet

Spectrum Scale er i hovedsak et filsystem. Dette betyr at hensikten og funksjonen til programvaren er å lagre data på disk. Med andre ord vil dette ha lite å si for sluttbrukeren, utenom en potensiell økning i ytelse og tilgjengelighet. Systemet ser for eksempel ikke på andre problemstillinger som organisering og presentasjon av filer for sluttbrukeren. Spectrum Scale har heller ingen funksjoner for ren backup av data uten å introdusere tredjepartsløsninger eller andre IBM-produkter, noe som vil føre til økte kostnader og kompleksitet. En må her se på behovene til bedriften, og hvilke funksjoner man egentlig trenger.

4. Installasjon av IBM Spectrum Scale

Som en del av bachelorprosjektet har vi kommet frem til at det skal utvikles og installeres et testsystem for IBM Spectrum Scale. Ved å konfigurere et eget oppsett for test, kan vi enklere formidle hva Spectrum Scale kan gjøre for Kongsberg Seatex. Derfor vil vi i dette kapitlet beskrive vi design, installasjon og konfigurasjon av testsystemet, før vi går videre med å se på bruksområdene for bedriften i kapittel 5. Hensikten er å få praktisk kunnskap om løsningen, samt bruke dette oppsettet til å demonstrere noen av mulighetene med IBM Spectrum Scale, og å koble dette opp mot prosjektets utgangspunkt.

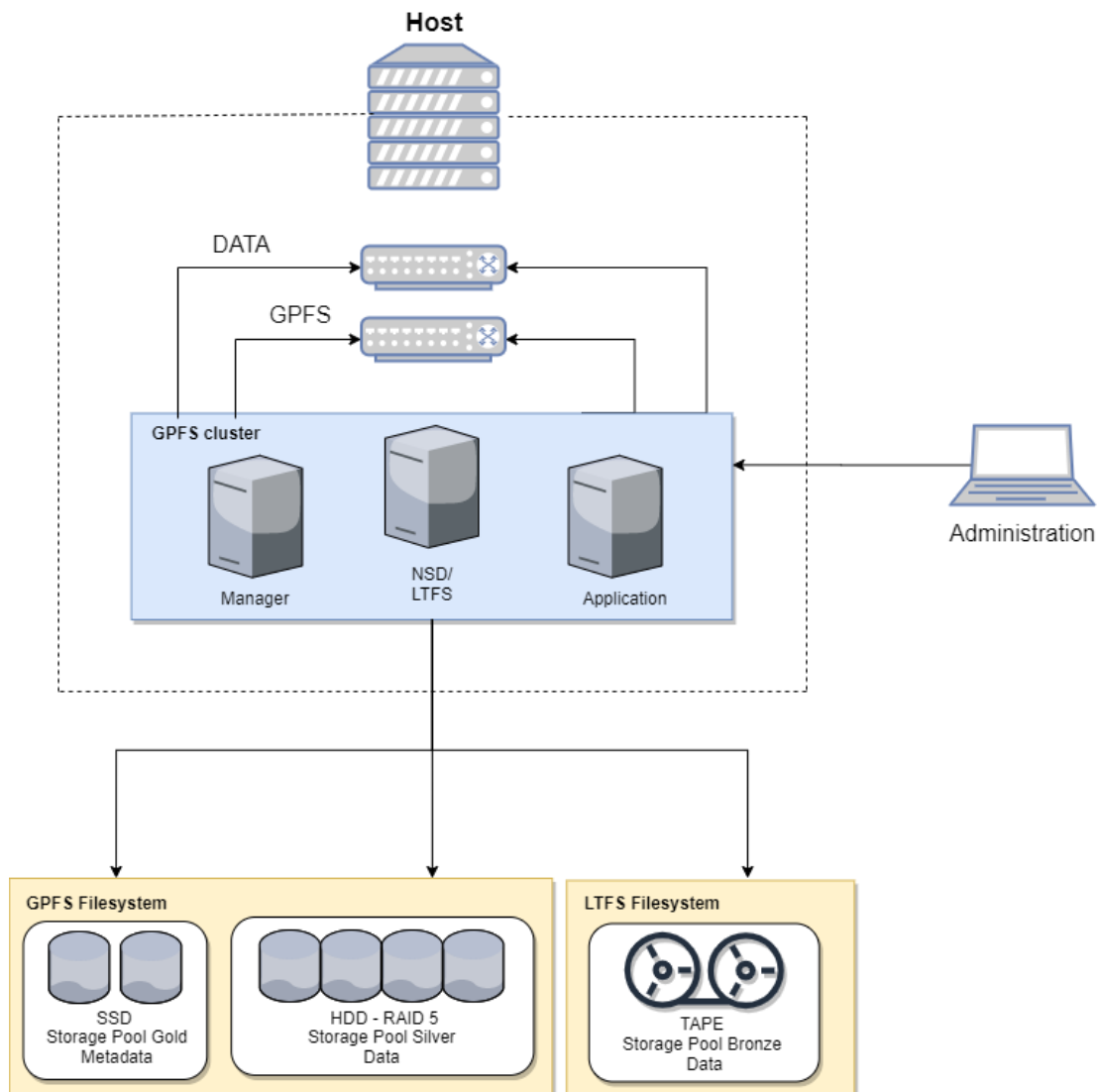
Som en repetisjon ser vi på figuren under at dataen transporteres fra båten til bedriftens lokaler, og deretter importereres til filsystemet. Filsystemet består av et IBM Spectrum Scale cluster, lagringsenhetene, og klienter som skal bruke filsystemet. Her demonstreres hvordan systemet brukes til å hente inn ny data, forvalte dataen i livssyklusen, og drifte systemet underveis.



Figur 14 - Systemoversikt

Figur 18 viser systemdesignet vi har bestemt for testløsningen til prosjektet, og tar utgangspunkt i figur 17. På tegningen ser du en hostmaskin og tre virtuelle maskiner. Disse virtuelle maskinene er koblet sammen i to lokale nett, ett for data og ett for GPFS-kommunikasjon, og utgjør clusteret i konfigurasjonen. Inn på det lokale nettet har vi også en VPN-tilkobling fra utsiden for administrasjon, som vi kommer til å bruke for å kontrollere clusteret og ta i bruk GUI-tjenesten. Clusteret er også koblet til lagringsløsningen gjennom NSD-serveren (noden i midten).

Videre mens vi går igjennom installasjon og konfigurasjon, vil vi forklare bakgrunnen for valgene vi gjør. Clusteret vi skal installere består av de tre virtuelle maskinene med ulike roller, altså nodene i clusteret. Disse maskinene er de tre nodene du ser i den blå firkanten, og kjører alle Ubuntu 18.04. NSD/LTFS-serveren vil være koblet til tre ulike storage pools: gull, sølv og bronse, der gull er SSD, sølv er HDD og bronse er tape. Maskinene er koblet sammen i et lokalt nett, der kun en av maskinene har begrenset internettilgang for fjernadministrasjon. Vi vil her se på hva som kreves for å gjennomføre en vellykket installasjon.



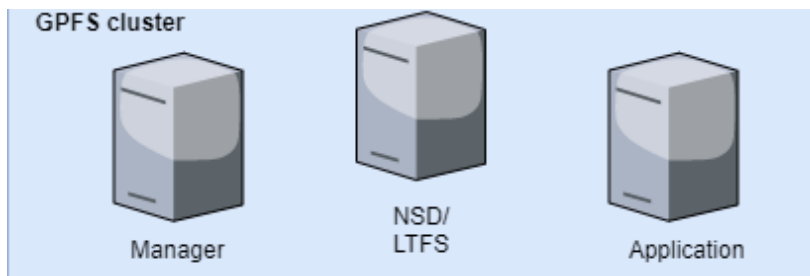
Figur 15 - Systemdesign for testing

4.1 Clustertopologi og definisjon av noder

Her ser vi på nodene som utgjør GPFS-clusteret vi skal installere, og hvilke roller og egenskaper de har.

4.1.1 Roller

Slik figur 18 og 19 illustrerer velger vi å benytte tre noder med ulike roller i denne clusterkonfigurasjonen: en managernode (også installernode), en NSD/LTFS-node og en applikasjonsnode. I kapittel 3.4.2 finner du nærmere beskrivelse av rollene. Grunnen til at vi gjør det slik er for å kunne demonstrere samarbeidet mellom flere noder, og tildele dem dedikerte roller. Vi kunne satt opp clusteret med flere eller færre noder, men vi mener at tre noder er et bra antall for å vise funksjonaliteten til Spectrum Scale. Her ser vi nærmere på hva hver enkelt node gjør, og hvilke roller den har.



Figur 16 - Noderoller

Managernode

Managernoden har hovedansvaret for å håndtere driften av clusteret. Den skal håndtere forespørsler fra de andre nodene, gi redundans og fungerer også som quorum node. Denne noden brukes også som installer-node ved bruk av installasjonsverktøyet til IBM Spectrum Scale for å etablere filsystemet og installere programvaren på nodene. I tillegg har den en instans av GUI-serveren installert.

Roller:

- Admin
- Cluster-manager
- Filesystem-manager
- GUI
- Installer
- Quorum

NSD/LTFS-node

Denne noden brukes primært for å dele systemets NSD-er med de andre nodene i clusteret. I tillegg er den konfigurert som quorum-node og host for GUI-serveren, samt protokollnode. Siden den er definert som protokollnode betyr dette at andre protokoll-klienter, eksempelvis NFS-klienter, kan henvende seg til denne noden for å få tilgang til ces_shared filsystemet. Dette er altså delen av filsystemet som skal deles utenfor clusteret.

Roller:

- NSD-server
- Admin
- GUI
- Protocol
- Quorum

Application-node

Applikasjonsnoden er tiltenkt prosessering av data på diskene, og er hovedsakelig den noden som kommer til å benytte seg av filene på lagringsenhetene. Dette vil være en maskin med høy prosessorkapasitet og I/O-båndbredde for å kunne analysere dataene. Noden er også med på å opprettholde quorum i clusteret.

Roller:

- Quorum

4.1.2 Quorum

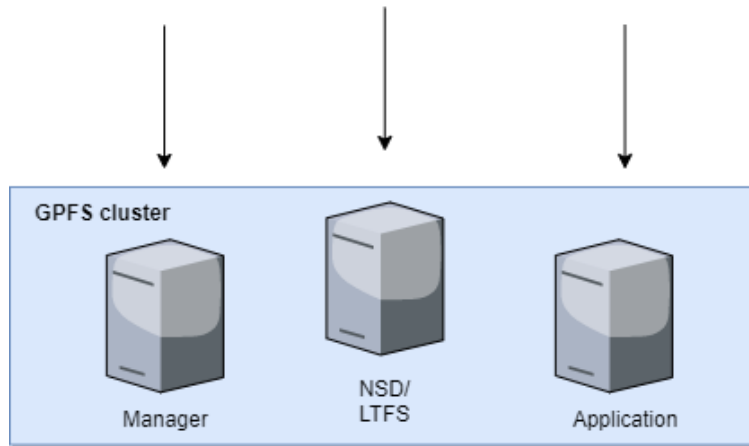
Alle nodene i clusterkonfigurasjonen er satt til å være quorum-noder. Dette er også en av grunnene til at vi har valgt tre noder i clusteret. Med tre quorum-noder betyr det at en av maskinene kan tas ned uten at det påvirker driften av filsystemet. Dersom to av nodene går ned, vil driften stoppe til nodene kommer opp igjen. Dette er gunstig både med tanke på redundans og oppetid, men også dersom det kreves vedlikehold på en av nodene.

4.2 Forhåndsinstallasjon

4.2.1 Krav for bruk av installasjonsverktøyet

Før man begynner å installere programvaren på nodene er det en rekke krav som må dekkes. Vi har valgt å installere Spectrum Scale på Ubuntu 18.04, da det er et fleksibelt operativsystem som vi er godt kjent med. Programmet kan også installeres på andre distribusjoner av Linux, og på Windows-maskiner. På Windows er en del av funksjonaliteten begrenset. Vi bruker installasjonsverktøyet som følger med, da dette gjør det svært enkelt å sette opp og konfigurere filsystemet i et cluster.

Kravene til installasjonen finner man i dokumentasjonen for IBM Spectrum Scale. Her installerer vi de nødvendige pakkene for installasjonen. Noter at dette gjøres på alle tre nodene:



Figur 17 - Installasjon på noder

Kjører først en apt update for å sørge for at alt er oppdatert:

```
bachelor@ManagerNode:~$ sudo apt update
```

```
bachelor@ManagerNode:~$ sudo apt upgrade
```

Sjekker at Python2.7 er installert, da dette er en nødvendighet:

```
bachelor@ManagerNode:/usr/bin$ ls /usr/bin | grep python2.7  
python2.7
```

Sjekker at RPM ikke er installert:

```
bachelor@ManagerNode:/usr/bin$ rpm  
Command 'rpm' not found, but can be installed with:  
sudo apt install rpm
```

Sjekker at SELinux ikke er installert:

```
bachelor@ManagerNode:/usr/bin$ selinux  
selinux: command not found
```

Nødvendig for bruk av noen funksjoner:

```
bachelor@ManagerNode:/usr/bin$ sudo apt install sqlite3
```

```
bachelor@ManagerNode:/usr/bin$ sudo apt install numactl
```

Sjekker at UFW ikke kjører:

```
bachelor@ManagerNode:/usr/bin$ sudo ufw status  
Status: inactive
```

Pakker for å bygge kernel-extension:

```
bachelor@ManagerNode:/usr/bin$ sudo apt install cpp
```

```
bachelor@ManagerNode:/usr/bin$ sudo apt install gcc
```

```
bachelor@ManagerNode:/usr/bin$ sudo apt install g++
```

Pakke for å bruke performance-monitoring:

```
bachelor@ManagerNode:/usr/bin$ sudo apt install libboost-regex-dev
```

Installerer NTP-pakker på alle noder

```
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# apt install ntp
```

```
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# apt install ntpdate
```

4.2.2 Klargjøring av installasjonsverktøyet

Her laster vi ned installasjonsverktøyet fra IBM sine nettsider, og pakker det ut slik at det kan tas i bruk.

Laster ned pakken til installer-noden:

```
C:\WINDOWS\system32>powershell C:\Users\Amund\Desktop\ss\ss.zip bachelor@52.174.186.13:/home/bachelor  
bachelor@52.174.186.13's password:  
ss.zip | 992012 kB | 43131.0 kB/s | ETA: 00:00:24 | 48%
```

```
bachelor@ManagerNode:~$ ls  
ss.zip
```

Pakker ut installasjonsverktøyet:

```
bachelor@ManagerNode:~$ unzip ss.zip -d ss
Archive:  ss.zip
  creating: ss/Spectrum Scale 5.0.4.1 Developer Edition/
  inflating: ss/Spectrum Scale 5.0.4.1 Developer Edition/README
  inflating: ss/Spectrum Scale 5.0.4.1 Developer Edition/SpectrumScale_public_key.pgp
  extracting: ss/Spectrum Scale 5.0.4.1 Developer Edition/Spectrum_Scale_Developer-5.0.4.1-x86_64-Linux-install.md5

bachelor@ManagerNode:~/ss/Spectrum Scale 5.0.4.1 Developer Edition$ ls
README
SpectrumScale_public_key.pgp
Spectrum_Scale_Developer-5.0.4.1-x86_64-Linux-install
Spectrum_Scale_Developer-5.0.4.1-x86_64-Linux-install.md5

root@ManagerNode:/home/bachelor/ss/Spectrum Scale 5.0.4.1 Developer Edition# ./Spectrum_Scale_Developer-5.0.4.1-x86_64-Linux-install
```

Aksepterer lisensavtalen:

```
LICENSE INFORMATION

The Programs listed below are licensed under the following
License Information terms and conditions in addition to the
Program license terms previously agreed to by Client and
IBM. If Client does not have previously agreed to license
terms in effect for the Program, the International License
Agreement for Non-Warranted Programs (Z125-5589-05) applies.

Program Name (Program Number):
IBM Spectrum Scale Developer Edition V5.0.4.1 (Community)

The following standard terms apply to Licensee's use of the
Program.

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.
1
```

Ser at filene har blitt pakket ut:

```
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1# ls
ganesha_debs  gpfs_rpms  installer  object_debs  smb_rpms  zimon_rpms
ganesha_rpms  hdfs_debs  license    object_rpms  tools
gpfs_debs     hdfs_rpms  manifest   smb_debs     zimon_debs
```

4.2.3 Kommunikasjon mellom noder

Før man kan bruke installasjonsverktøyet er det et par ting som må være på plass. Det første er programvare-kravene som beskrives i 6.1.1, det andre er at maskinene i nettverket må ha fri tilgang til hverandre gjennom passord-løs SSH med root-rettigheter. Slik setter vi opp dette:

Passordløs SSH-tilgang

Det er en risikofaktor å la brukeren root ha tilgang til systemet over ssh, da et vellykket brute-force angrep utenifra lar angriperne få full tilgang til systemet. For vårt system er det nødvendig å la root få tilgang over ssh, og dermed må vi endre standardinnstillingene. Innstillinger knyttet til ssh finner vi i mappen «`/etc/ssh/sshd_config`».

Med kommandoen «`# nano /etc/ssh/sshd_config`» går vi inn i filen og ser etter en linje med «`#PermitRootLogin`». Denne setter vi til «yes» og fjerner hash tegne som kommenterer ut linja.

```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Med ssh kan du nå logge inn på maskinen med root, foreløpig må du fylle ut passord hver gang. I neste avsnitt ser vi på hvordan du setter opp ssh-nøkler for innlogging uten passord.

SSH nøkkelpar

For at de ulike nodene i clusteret skal kunne kommunisere med hverandre må vi sette opp passordløs ssh mellom de ulike nodene. Ssh nøkkelpar gjør at maskinene har tilgang til hverandre uten å fylle ut passord hver gang noen logger inn. Med kommandoen «`# ssh-keygen -t rsa`» oppretter du et nøkkelpar. Nøkkelparet legger seg i mappen «`/.ssh`» i hjemmemappen til brukeren. Her er det dermed viktig å bruke root som bruker og ikke sudo, fordi filene med sudo legges i hjemmemappen til brukeren.

I mappen «`/root/.ssh/`» er det nå opprettet 3 filer: «`authorized_keys`», «`id_rsa`» og «`id_rsa.pub`». Den private nøkkelen ligger i «`id_rsa`», mens den offentlige nøkkelen ligger i «`id_rsa.pub`». Vi ønsker å bruke nøkkelparet på alle maskinene i clusteret, derfor kopierer vi nøkkelen over til «`authorized_keys`».

Deretter endrer vi rettighetene med «`# chmod 600`» for filen. Det gir eieren av filen Read og Write rettigheter. Her er eieren root, som betyr at bare root har tilgang til den passordløse tilkoblingen over ssh.

```

root@ManagerNode:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:leRsN4gT0jqmX2kWU2FBfNfavtmL++XQRdcp50jrGM4 root@ManagerNode
The key's randomart image is:
+---[RSA 2048]-----+
|
|  ...o*o . . |
|  ..Boo= . + |
|  .o.Boo+ ++ |
|  + o+.. .o.o |
|  o .S+ . . . |
|  . = . . .o |
|  . + . . . * |
|  . o + . =o |
|    E . oooo |
+----[SHA256]-----+
root@ManagerNode:~# cd ~/.ssh/
root@ManagerNode:~/.ssh# ls
authorized_keys  id_rsa  id_rsa.pub
root@ManagerNode:~/.ssh# cp id_rsa.pub authorized_keys
root@ManagerNode:~/.ssh# chmod 600 authorized_keys
root@ManagerNode:~/.ssh#

```

Når maskin A prøver å kommunisere med maskin B, ser A etter om den offentlige nøkkelen som ligger i «**authorized_keys**» stemmer med den private nøkkelen til B i B's «**id_rsa**». Derfor ønsker vi å oppdatere alle maskiner i clusteret med samme nøkler.

Dermed oppretter vi mappen «**.ssh**» på de andre maskinene med «**# mkdir /root/.ssh**» og setter rettighetene til «**# chmod 700 /root/.ssh**». I mappen «**/root/.ssh**» bruker vi kommandoen touch til å opprette filene «**id_rsa**» og «**authorized_keys**». Vi går inn i filene og kopierer de samme filene vi hadde i manageroden over til de andre maskinene. Rettighetene setter vi til 600 på begge filene.

```

root@ApplicationNode:~/.ssh# nano authorized_keys
root@ApplicationNode:~/.ssh# chmod 600 authorized_keys
root@ApplicationNode:~/.ssh# touch id_rsa
root@ApplicationNode:~/.ssh# nano id_rsa
root@ApplicationNode:~/.ssh# chmod 600 id_rsa
root@ApplicationNode:~/.ssh# █

```

Med «**# ssh «maskinnavn eller ip adresse**» kan vi nå få tilgang mellom de ulike maskinene i nettverket uten bruk av passord.

4.2.4 Manuell installasjon

Det er mulig å installere Spectrum Scale-pakkene manuelt. Dette kan være ønskelig i spesielle tilfeller der konfigureringen begrenses av installasjonsverktøyet, slik som på spesielle distribusjoner eller operativsystemer. I vårt tilfelle er ikke dette et problem, og installasjonsverktøyet er enkelt og raskt å ta i bruk. Derfor ser vi ikke videre på dette i prosjektet.

4.2.5 Klargjøring av CES

IBM Spectrum Scale støtter ulike protokoller for eksport av filsystemet. Dette kan du lese mer om under 3.3.3. Med CES kan vi sette opp kommunikasjon med klienter over protokoller som NFS og SMB. Dette gjør at maskiner som ikke har installert Spectrum Scale kan få tilgang til filene på GPFS-filsystemet.

For å konfigurere CES med de ulike protokollene må et par elementer være på plass. I clusteret må det være minst en node som er satt opp til å håndtere protokoller. Clusteret og noden må også ha en skygge-ip som clusteret kan administrere. En skygge-ip er en måte å tilordne flere ip-adresser til samme nettverkskort på. Dermed kan nettverkskortet lytte på flere ip-adresser samtidig. Spectrum Scale vil sette opp skygge-ipen som «export ip address». Adressen blir delt mellom de ulike protokollnodene, som gjør at klientmaskiner kan forholde seg til samme ip-adresse om en protokollnode i clusteret går ned.

Her viser vi hvordan du oppretter en skygge-ip på storagenoden og setter den som en «export ip» for clusteret:

I filen «/etc/network/interfaces» skriver vi inn konfigureringen som vist under. «Eth0» er nettverkskortet til maskinen. Ved å sette et kolon med et tall bak nettverkskortet, kan vi gi en maskin en skygge-ip.

```
root@managernode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale node add storagenode -p
```

```
auto eth0:1
iface eth0:1 inet static
name Ethernet alias Lan card
address 10.0.0.12
netmask 255.255.255.0
broadcast 10.0.0.255
network 10.0.0.0
```

Vi aktiverer den nye nettverkskonfigureringen med kommandoen «# ifup» etterfulgt det gjeldene nettverkskortet. Du kan sjekke om innstillingene er gjeldene med kommandoen «# ifconfig», som viser nåværende ip-konfigurasjon:


```
# ifup eth0:1
```

På installasjonsnoden setter vi opp skygge-ipen til å peke mot storagenoden. Dette er nødvendig for å finne ipen under installasjon.

```
127.0.0.1 localhost
10.0.0.4 managernode
10.0.0.5 storagenode.internal.cloudapp.net
10.0.0.6 applicationnode.internal.cloudapp.net
10.0.0.12 storagenode.internal.cloudapp.net
```

Videre må clusteret gjøre ip-adressen tilgjengelig som export-ip. Dette må gjøres etter installasjonen er utført og før clusteret deployes. Ip-adressen gjøres tilgjengelig for CES med «# mmces address add --ces-ip 10.0.0.12». For å sette ip-adressene som skal eksporteres brukes kommandoen «# config protocols -e». Taggen «-e» står for export-ip.

```
root@managernode:/usr/lpp/mmfs/5.0.4.1/installer# mmces address add --ces-ip 10.0.0.12
```

```
root@managernode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale config protocols -e 10.0.0.12
```

CES er nå klar for å deployes.

4.2.6 OpenLDAP med phpLDAPAdmin

Spectrum Scale har støtte for flere ulike katalogtjenester for tilgangsstyring til systemet og filbehandling. Dette gjelder for bruk av CES, der vi deler filsystemet over protokoller. Eksempelvis er dette Windows Active Directory, Samba, eller som vi skal sette opp her: OpenLDAP. LDAP står for «Lightweight Directory Access Protocol» og er en protokoll for tilgangsstyring. I katalogen ligger en liste med alle brukere og maskiner som er tilgjengelig for ulike tjenester i domene. OpenLDAP har åpen kildekode og er gratis å benytte. Programmet kan lastes ned på flere ulike Linux distribusjoner.

For Spectrum Scale er det nødvendig med en katalogtjeneste for å sette rettigheter på over ulike filer ved bruk av CES. Clusteret må dermed ha tilgang til ulike brukere. Klientmaskinene når filene med en tjeneste som NFS eller SMB, og for å få tilgang må klientmaskinene logge inn med en bruker for katalogtjenesten.

Vi vil her vise 4 ulike trinn av installasjonen: Installasjon av OpenLDAP, installasjon av phpLDAPAdmin for grafisk brukerkontroll av LDAP serveren, legge til bruker i phpLDAPAdmin og hvordan å gi en klientmaskin tilgang til brukere i clusteret.

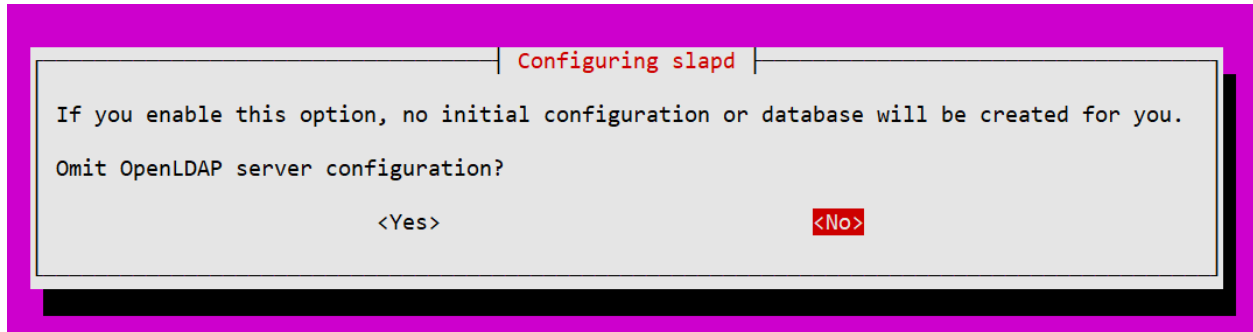
Installasjon av OpenLDAP

Installasjonen foregår via å installere 2 pakker: «slapd» og «ldap-utils». Dette gjør vi på en ubuntu-maskin i samme nettverk som clusteret. Når vi installerer de første pakkene, blir vi spurt om å fylle ut et administrativt passord for OpenLDAP.

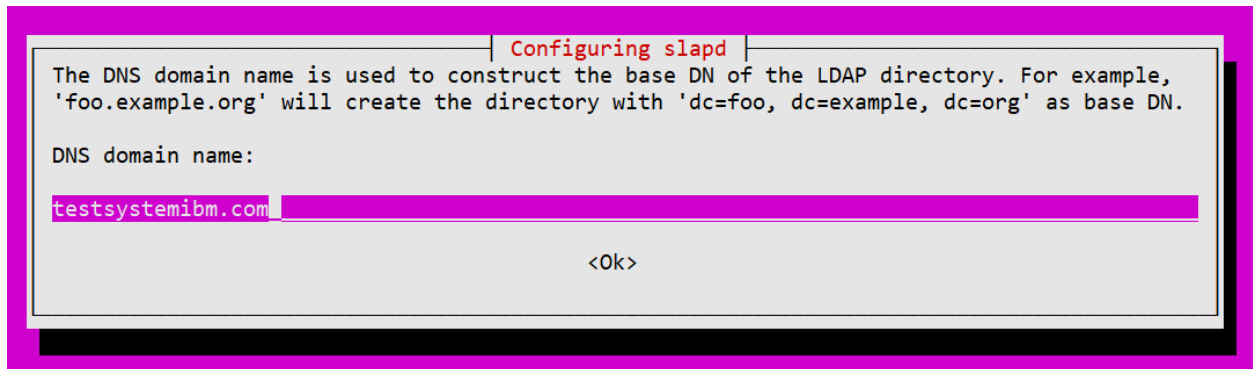
```
root@LDAP:~# apt-get install slapd ldap-utils
```

Etter installasjonen konfigurerer vi tjenesten med «dpkg-reconfigure slapd». Her blir vi tatt gjennom en konfigurasjon i terminalen.

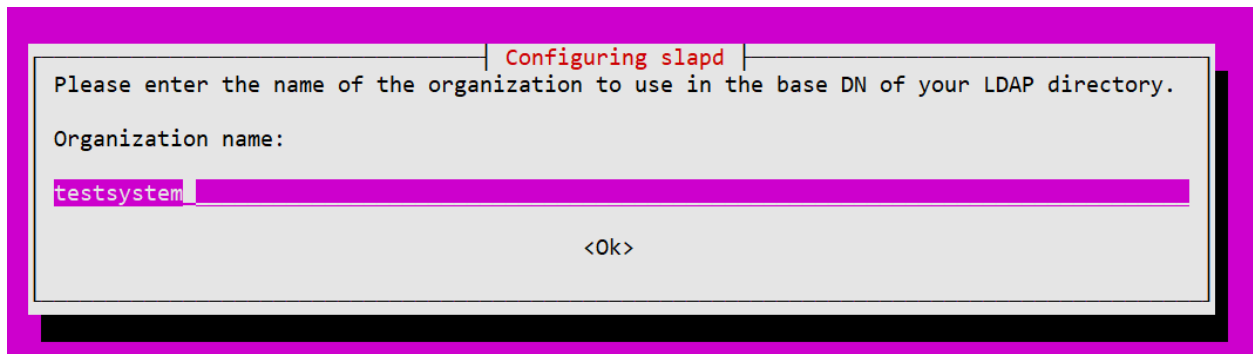
```
root@LDAP:~# dpkg-reconfigure slapd
```



Dn står for «distinguished name» eller unikt navn og brukes til å nå tjenesten fra klientmaskiner. Her benytter vi kun et testdomene som vi ikke eier, hvor det er mulig å i stedet benytte ip-adresse til å benytte tjenesten. Fremdeles er det viktig å forstå at dn til tjenesten blir «dc=testsystemibm,dc=com», dette skal benyttes i senere konfigurasjoner.

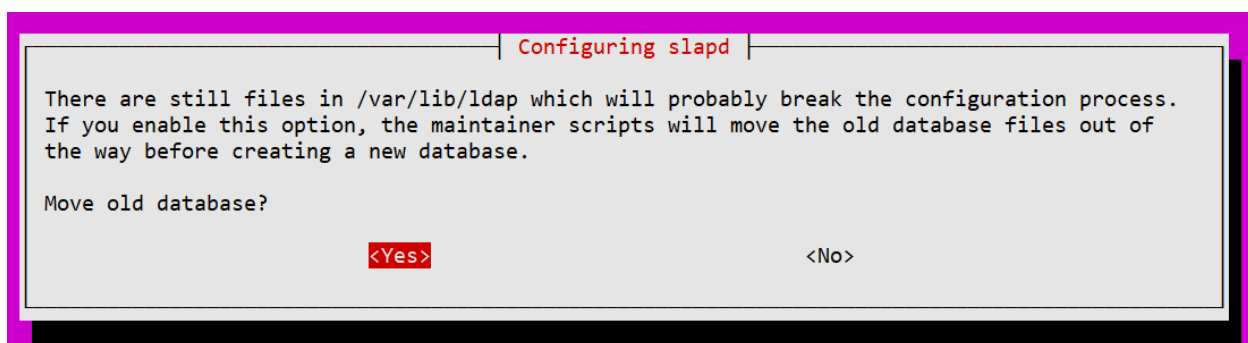
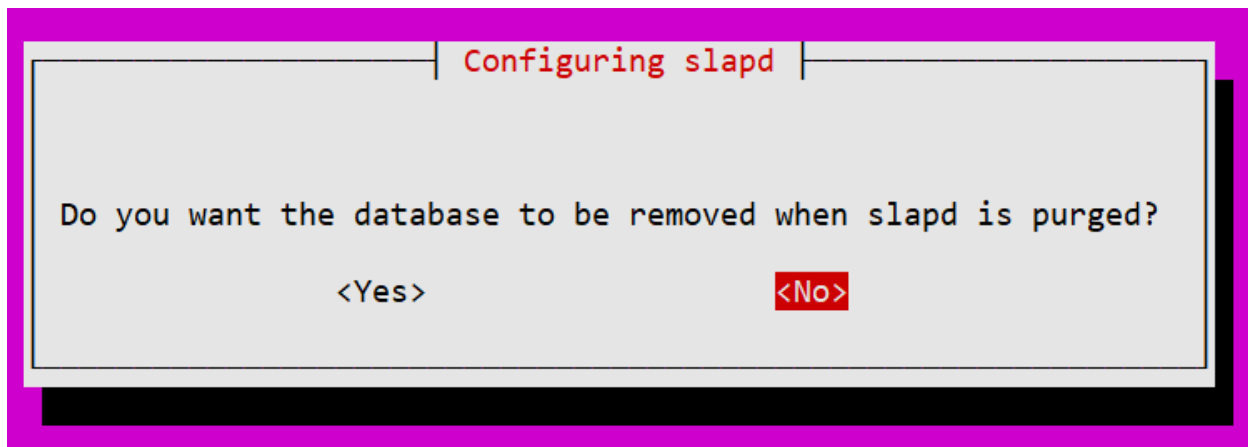
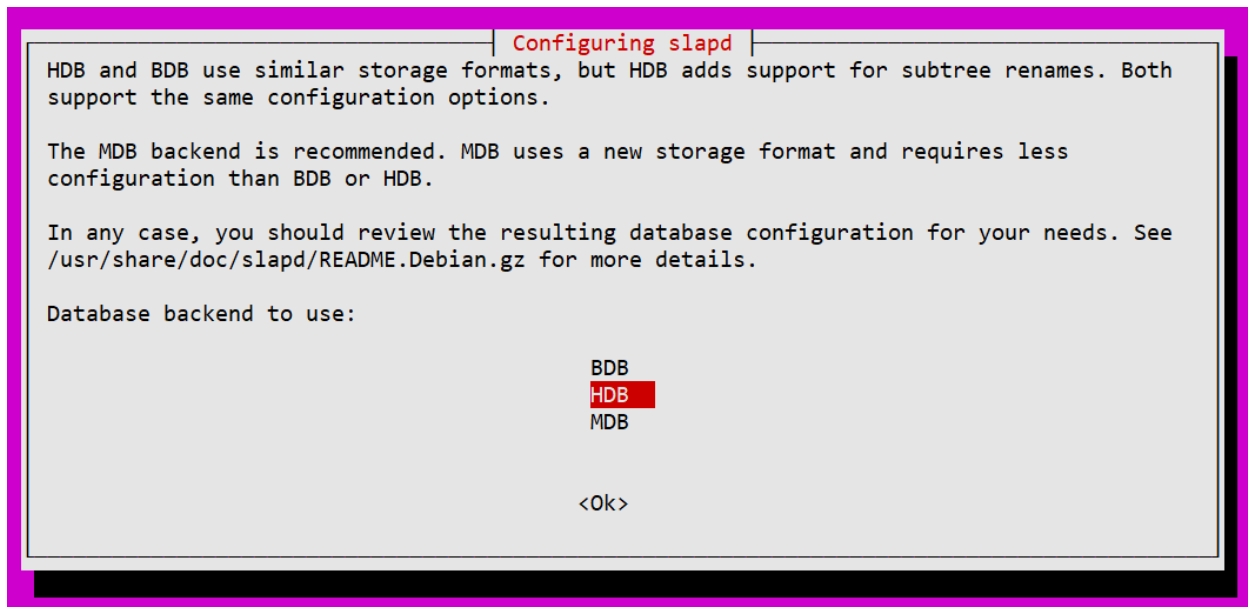


Organisasjonsnavn er ikke essensielt for vår konfigurasjon, men vi setter den til «testsystem».



Etter vi har satt organisasjonsnavn blir vi bedt om å bestemme navn og passord for administrasjonsbrukeren for LDAP tjenesten. Her er setter vi brukernavnet til admin, hvor brukerens dn blir «cn=admin,dc=testsystemibm,dc=com».

For de neste konfigurasjonene gjør vi følgende:



Etter denne konfigurasjonen fungerer nå OpenLDAP tjenesten. Videre vil vi installere og benytte phpLDAPadmin for å konfigurere brukere til katalogtjenesten.

Installasjon av phpLDAPAdmin

PhpLDAPAdmin er et program som ligger med åpen lisens på github. Programmet kjøres som en pakke på en webtjener. I installasjonen vil vi opprette en apache webserver, laste ned kildekoden og konfigurere webserveren og phpLDAPAdmin mot LDAP maskinen.

Vi begynner med å installere alle nødvendige pakker. Når vi installerer apache2 har vi en webserver som er tilgjengelig på den offentlige ip-adressen til serveren.

```
root@LDAP:~# sudo apt-get install php php-ldap php-xml apache2 git
```

Kildekoden lastes ned fra github med kommandoen «# git clone».

```
git clone https://github.com/breisig/phpLDAPAdmin.git
```

Neste vi ønsker å gjøre er å konfigurere phpLDAPAdmin mot OpenLdap. Filen vi skal konfigurere finner vi i «/root/phpldapadmin/config». Vi kopierer og endrer navnet på eksempelfilen til «config.php». Deretter åpner vi filen og redigerer innholdet.

```
cp config.php.example config.php
```

```
nano /root/phpldapadmin/config/config.php
```

I filen ønsker vi å sette konfigurasjonen mot LDAP serveren. På linjene med «`$servers->setValue`» endrer vi verdiene for host, port, base og login. Host er ip-adressen til LDAP maskinen. Dette kunne ha vært det unike navnet vi brukte, men siden det er et domene vi ikke styrer bruker vi ip-adressen på det interne nettverket i stedet. Port er standardport 389. Base er domenenavnet «`dc=testsystem,dc=com`». Login er administrasjonsbrukeren til OpenLDAP med domenenavn «`cn=admin,dc=testsystem,dc=com`». Bare for å presisere er linjen under med `cn=Manager` kommentert ut med «#» og er ikke en del av installasjonen.

```
/* Examples:
   'ldap.example.com',
   'ldaps://ldap.example.com/',
   'ldapi://%2fusr%2flocal%2fvar%2frun%2fldapi'
   (Unix socket at /usr/local/var/run/ldap) */
$servers->setValue('server','host','10.0.0.9');

/* The port your LDAP server listens on (no quotes). 389 is standard. */
$servers->setValue('server','port',389);

/* Array of base DNSs of your LDAP server. Leave this blank to have phpLDAPAdmin
   auto-detect it for you. */
$servers->setValue('server','base',array('dc=testsystemibm,dc=com'));
```

```
/* The DN of the user for phpLDAPadmin to bind with. For anonymous binds or
'cookie','session' or 'sasl' auth_types, LEAVE THE LOGIN_DN AND LOGIN_PASS
BLANK. If you specify a login_attr in conjunction with a cookie or session
auth_type, then you can also specify the bind_id/bind_pass here for searching
the directory for users (ie, if your LDAP server does not allow anonymous
binds. */
$servers->setValue('login','bind_id','cn=admin,dc=testsystemibm,dc=com');
# $servers->setValue('login','bind_id','cn=Manager,dc=example,dc=com');
```

For å gjøre konfigurasjonen av dette testsystemet lettere fjerner vi også varsler.

```
/* Hide the warnings for invalid objectClasses/attributes in templates. */
$config->custom->appearance['hide_template_warning'] = true;
```

Systemet er nå konfigurert og vi kopierer filene i «phpLDAPadmin» over til «/var/www/html/». Denne mappen inneholder filene som apache webtjeneren deler.

```
cp /root/phpldapadmin/ /var/www/html/
```

Vi restarter webtjeneren for å gjøre filene tilgjengelig. Med «# systemctl start slapd» starter vi tjenesten og sørger for den alltid starter ved oppstart med «# systemctl enable slapd».

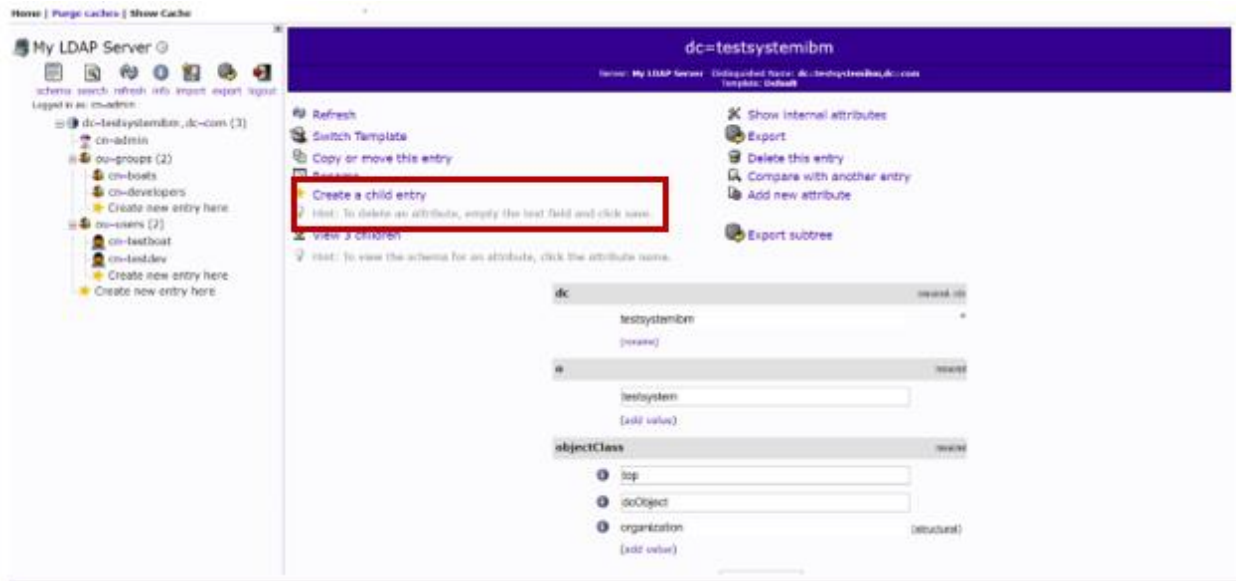
```
systemctl restart apache2
systemctl start slapd
systemctl enable slapd
```

Brukeradministrasjon med phpLDAPadmin

Etter installasjonen av phpLDAPadmin har du nå muligheten til å åpne programmet. Du går i en nettleser og fyller ut ip-adressen til serveren. Du vil bli spurt etter en administratorbruker og passord, hvor det er admin brukeren fra OpenLDAP konfigurasjonen som skal brukes. Under ser du brukergrensesnittet.

I oppsettet ønsker vi å ha 2 ulike brukere: testdev og testboat. Hver bruker trenger å ha en gruppe så vi lager en gruppe for utviklere og en gruppe for båter. For å ha system over kontoene, lager vi en organizational unit (ou) for brukere og en ou for grupper.

Du starter installasjonen ved å trykke på domene, for å deretter velge «create a child entry». Første som må opprettes er ou-ene. Deretter opprettes gruppene i ou-en for grupper. Brukerne testboat og testdev legges i ou-en for brukere. Når brukerne lages må vi fylle inn hvilken gruppe brukerne tilhører, henholdsvis boats og developers. Når brukerne er opprettet må vi gå inn i gruppene igjen og trykke på «add new attribute». Her velger vi brukeren vi ønsker skal ligge i gruppen.



Figur 18 - LDAP Gui 1



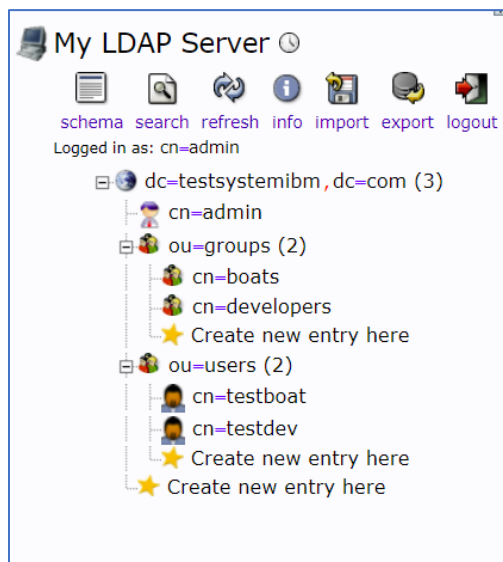
Figur 19 - LDAP Gui 2

Under til venstre ser vi hvordan en bruker er satt opp. Her er det viktig å merke seg cn-navnet og at brukeren har en hjemmemappe. Når brukeren er riktig, trykker du commit for å legge brukeren til i databasen.

Attribute	New Value	Skip
cn= testdeveloper,ou=users,dc=testsystemibm,dc=com		
Last name	testdeveloper	<input type="checkbox"/>
Common Name	testdeveloper	<input type="checkbox"/>
User ID	testdeveloper	<input type="checkbox"/>
Home directory	/home/users/testdeveloper	<input type="checkbox"/>
UID Number	1000	<input type="checkbox"/>
GID Number	500	<input type="checkbox"/>
Password	*****	<input type="checkbox"/>
objectClass	inetOrgPerson posixAccount	<input type="checkbox"/>

Commit Cancel

Figur 20 - LDAP User Configuration



Figur 21 - LDAP Directory

Under til høyre ser vi oppsettet av brukere og grupper i ulike ou-er. Viktig å nevne at koblingen mellom brukere grupper som vi beskrev ovenfor, ikke blir vist her.

Klientkonfigurasjon av LDAP

Hensikten med å konfigurere en klientmaskin mot LDAP er at brukere i LDAP kan logge inn på maskinen og dermed få tilgang til filer i Spectrum Scale.

På en klientmaskin med Ubuntu logger vi inn med root og utfører følgende kommando:

```
apt install libnss-ldap libpam-ldap ldap-utils nscd
```

Når pakkene blir installert får vi opp konfigurasjonen mot LDAP. Vi begynner med å sette ip-adressen til LDAP-serveren.

Configuring ldap-auth-config

Please enter the URI of the LDAP server to use. This is a string in the form of ldap://<hostname or IP>:<port>/. ldaps:// or ldapi:// can also be used. The port number is optional.

Note: It is usually a good idea to use an IP address because it reduces risks of failure in the event name service problems.

LDAP server Uniform Resource Identifier:

ldap://10.0.0.9

<Ok>

Dn:

Configuring ldap-auth-config

Please enter the distinguished name of the LDAP search base. Many sites use the components of their domain names for this purpose. For example, the domain "example.net" would use "dc=example,dc=net" as the distinguished name of the search base.

Distinguished name of the search base:

dc=testsystemibm,dc=com

<Ok>

Hvilken versjon av LDAP vi bruker.

Configuring ldap-auth-config

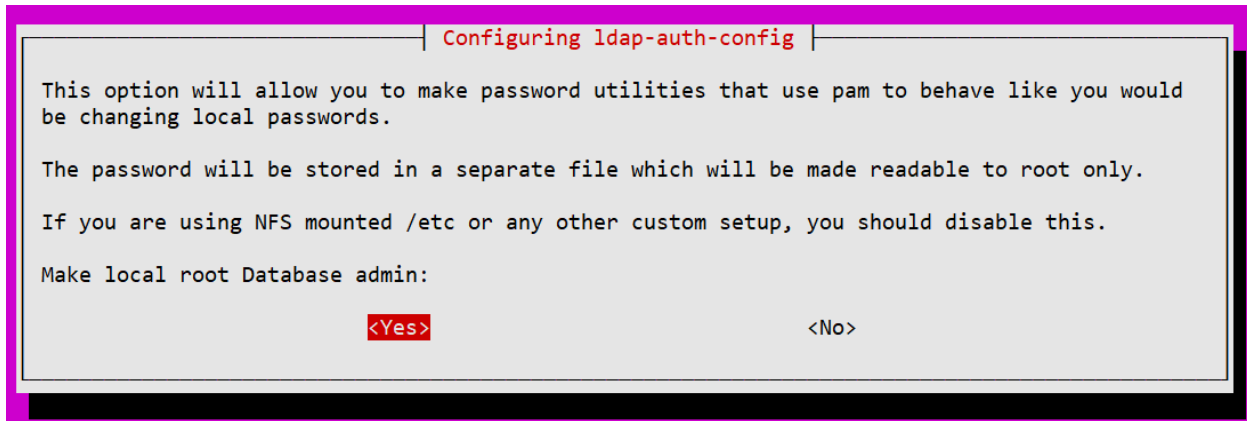
Please enter which version of the LDAP protocol should be used by ldapns. It is usually a good idea to set this to the highest available version.

LDAP version to use:

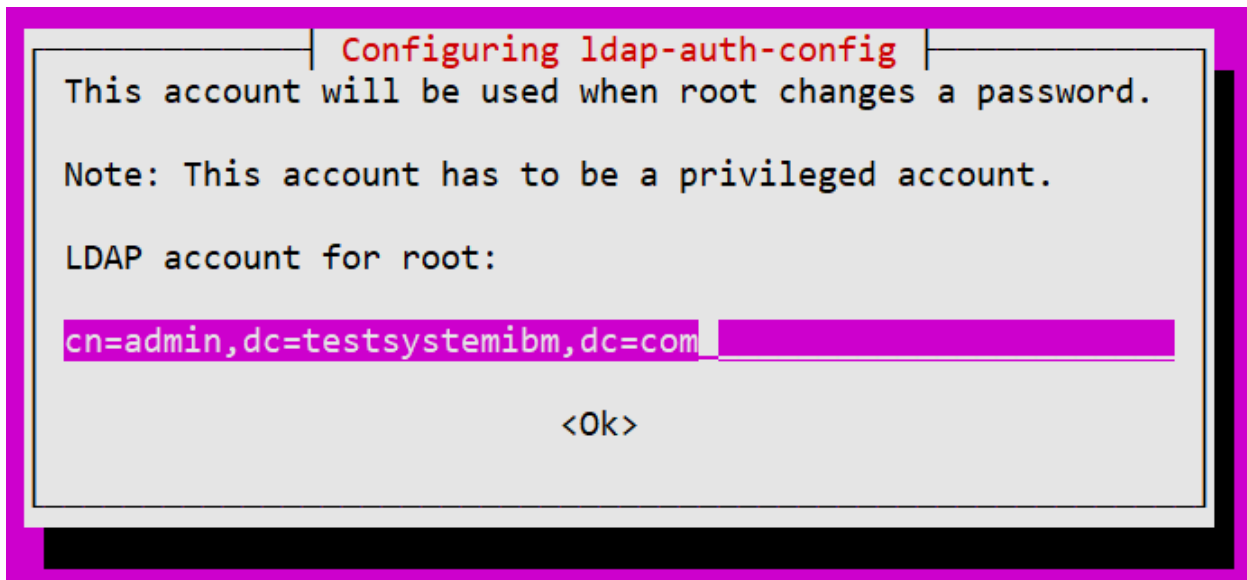
3
2

<Ok>

For vårt system vil vi kun benytte NFS som en ekstra disk i pcen, og velger yes her.



Neste punkt er å fylle ut admin brukeren for domene, her spør de først om navnet for så passord.

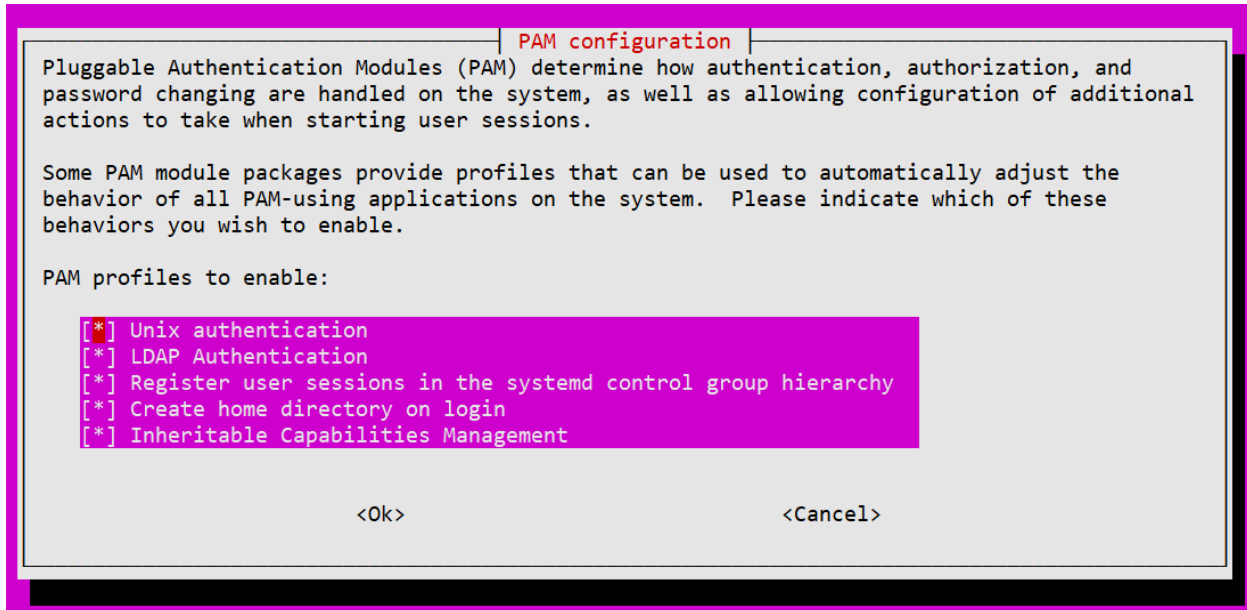


Neste som må gjøres er å konfigurere LDAP mot NSS. NSS står for Network Security Services som er et bibliotek for sikkerhetskonfigurering mellom ulike enheter i et nettverk. Det gjøres ved å konfigurere auth-client.

Videre konfigurerer du pam. Pam er et bibliotek i linux som styrer tilgang til maskinen fra andre brukerdata-baser. Her setter vi at pam skal godkjenne Unix og LDAP.

```
auth-client-config -t nss -p lac_ldap
pam-auth-update
```

Pam-auth-update sender en prompt. Her velger vi alle punktene.



For at brukere skal ha tilgang til hjemmemappe må vi også legge til en linje i «`/etc/pam.d/common-session`».

For å teste at systemet nå fungerer kan du logge inn med en domenebruker med ssh til klientmaskinen:

```
# and here are more per-package modules (the "Additional" block)
session required pam_unix.so
session optional pam_ldap.so
session optional pam_systemd.so
session optional pam_mkhomedir.so
session required pam_mkhomedir.so skel=/etc/skel umask=007
# end of pam-auth-update config
```

```
root@failover:~# ssh testdev@10.0.0.11
testdev@10.0.0.11's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.0.0-1032-azure x86_64)
```

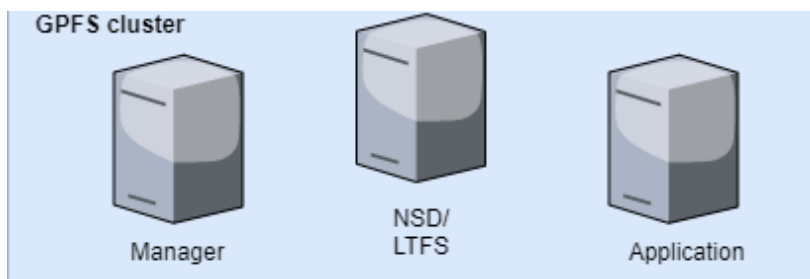
Vi kan også se at en bruker har hjemmemappe, her er LDAP brukeren riktig konfigurert med developers som gruppe.

```
$ ls -l /home/users
total 4
drwxrwxrwx 4 testdev developers 4096 Apr  3 12:28 testdev
```

4.3 Installasjon av filsystemet

Med IBM Spectrum Scale pakken følger det med et svært hendig installasjonsverktøy som raskt og enkelt installerer alt av nødvendig programvare på nodene i clusteret og etablerer filsystemet kun med et par kommandoer. Verktøyet sjekker også for alt av nødvendigheter før installasjon og melder klart ifra om feil og mangler. Slik unngår man korrupt installasjon og frustrasjon ved feil. I oppsett av testsystemet kommer vi til å benytte oss av dette skriptet for installasjon.

Maskinen som kjører installasjonsverktøyet kalles for installer-noden. Herfra kan man gjøre oppdateringer på clusteret og filsystemet, eksempelvis ved å legge til nye NSDer og endre navn på cluster. Installer-noden er i dette tilfellet den samme som manager-noden, da vi ikke lager en egen VM for denne. Se figur 18 for referanse.



Figur 22 - Noderoller 2

4.3.1 Oppsett av noder i clusteret

Når vi først setter i gang med oppsettet av Spectrum Scale, begynner vi med å sette opp installer-noden. Dette blir igjen den samme som managernoden. Etter vi pakket ut installasjonsverktøyet har filene vi skal bruke blitt plassert under `</usr/lpp/mmfs/5.0.4.1/installer>`. Her ligger den kjørbare filen «**spectrumscale**» som kjører installasjonen. Fremover vil du se at vi bruker denne filen i en rekke kommandoer.

```
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale setup -s 10.0.0.4
[ INFO ] Installing prerequisites for install node
[ INFO ] Chef successfully installed and configured
[ INFO ] Your control node has been configured to use the IP 10.0.0.4 to communicate with other nodes.
[ INFO ] Port 8889 will be used for chef communication.
[ INFO ] Port 10080 will be used for package distribution.
[ INFO ] Install Toolkit setup type is set to Spectrum Scale (default). If an ESS is in the cluster, run this command to set ESS mode: ./spectrumscale setup -s server_ip -st ess
[ INFO ] SUCCESS
[ INFO ] Tip : Designate protocol, nsd and admin nodes in your environment to use during install: ./spectrumscale -v node add <node> -p -a -n
```

Legger så til installernoden som en del av clusteret med kommandoen «**# spectrumscale node add**». Den får rollene admin, GUI-server, manager og quorum. Flaggene du ser etter ip-adressen definerer hvilke roller noden får, henholdsvis «**-a**» for admin, «**-g**» for GUI, «**-p**» for protocol, «**-m**» for manager og «**-q**» for quorum.

```

root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale node add 10.0.0.4 -a -g -m -q
[ INFO ] Adding node managernode.internal.cloudapp.net as a GPFS node.
[ INFO ] Adding node managernode.internal.cloudapp.net as a quorum node.
[ INFO ] Adding node managernode.internal.cloudapp.net as a manager node.
[ INFO ] Setting managernode.internal.cloudapp.net as an admin node.
[ INFO ] Configuration updated.
[ INFO ] Tip : Designate protocol or nsd nodes in your environment to use during install:./spectrumscale node
add <node> -p -n
[ INFO ] Setting managernode.internal.cloudapp.net as a GUI server.

```

Legger til applikasjonsnoden og gir den rolle som quorum-node.

```

root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale node add 10.0.0.6 -q
[ INFO ] Adding node applicationnode.internal.cloudapp.net as a GPFS node.
[ INFO ] Adding node applicationnode.internal.cloudapp.net as a quorum node.

```

Legger til storage noden og gir den rollene admin, NSD-server, quorum, protocol og GUI-server.

```

./spectrumscale node add 10.0.0.5 -a -q -g -p -n
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale node add 10.0.0.5 -a -n -q -g
[ INFO ] Adding node storagenode.internal.cloudapp.net as a quorum node.
[ INFO ] Node storagenode.internal.cloudapp.net is already an NSD server.
[ INFO ] Tip :If all node designations are complete, add NSDs to your cluster definition and de
[ INFO ] Setting storagenode.internal.cloudapp.net as an admin node.
[ INFO ] Configuration updated.

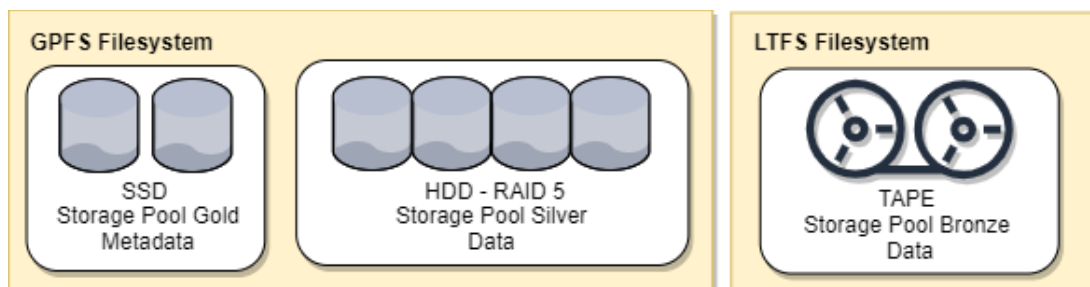
```

Ved å bruke kommandoen «**# ./spectrumscale node list**» får vi skrevet ut clusterkonfigurasjonen slik den ser ut til nå. Dette er en god måte å dobbeltsjekke at alt har blitt gjort riktig, og at adressene stemmer. Her ser vi oversikten av systemet vårt så langt. Vi ser at dette stemmer overens med det som er beskrevet ovenfor av roller for nodene:

GPFS Node	Admin Node	Quorum Node	Manager Node	NSD Server	Protocol Node	GUI Server	Perf Mon Collector	OS	Arch
applicationnode.internal.cloudapp.net		X						ubuntu18	x86_64
managernode.internal.cloudapp.net	X	X	X			X	X	ubuntu18	x86_64
storagenode.internal.cloudapp.net	X	X		X	X	X	X	ubuntu18	x86_64

4.3.2 Legge til disker og storage pools

Videre skal vi definere disker og storage pools for clusteret. Slik vi har definert oppsettet tidligere, har vi tre storage pools: en av typen «gold», en av typen «silver», og en «bronze». Dette er uttrykk vi bruker som betegnelse på kvaliteten og ytelsen til diskene. I vår konfigurasjon er alle diskene oppført som NSD-er. Med andre ord er dette disker som deles over data-nettverket, og som alle noder får tilgang til gjennom NSD-serveren.



Figur 23 - Storage pool nivåer 2

Først begynner vi med å legge til en SSD i «system» storage poolen. Denne disken hører til gruppen Gold, som du ser til venstre på figur 11. Denne disken defineres kun til å inneholde metadata. Dette gjør vi for å øke responstiden til filsystemet uten å bruke for mye plass på dyre lagringsmedier. Skriver inn IP-adressen til NSD-serveren og stien til disken. Definerer også at disken skal høre til filsystemet «gpfs_device».

```
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale nsd add /dev/sdd -p 10.0.0.5 -fs gpfs_device -u metadataOnly -po system -fg 1
[ INFO ] Connecting to storagenode.internal.cloudapp.net to check devices and expand wildcards.
[ INFO ] Looking up details of /dev/sdd.
[ INFO ] The installer will create the new file system gpfs_device if it does not exist.
[ INFO ] Adding NSD None on storagenode.internal.cloudapp.net using device /dev/sdd.
[ INFO ] Configuration updated
```

Som representant for Silver-gruppen, legger vi til en tregere HDD disk, og sier at den kun skal inneholde data. Denne hører også til storage poolen «system», og filsystemet gpfs_device:

```
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale nsd add /dev/sdc -p 10.0.0.5 -fs gpfs_device -u dataOnly -po system -fg 1
[ INFO ] Connecting to storagenode.internal.cloudapp.net to check devices and expand wildcards.
[ INFO ] Looking up details of /dev/sdc.
[ INFO ] Adding NSD None on storagenode.internal.cloudapp.net using device /dev/sdc.
[ INFO ] Configuration updated
```

For å demonstrere migrering av data mellom ulike storage pools i forskjellig klasse, oppretter vi storage poolen «tape», og legger til den tredje disken her. Dette vil være lagring av typen Bronze.

```
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale nsd add /dev/sde -p 10.0.0.5 -fs gpfs_device -u dataOnly -po system -fg 1
[ INFO ] Connecting to storagenode.internal.cloudapp.net to check devices and expand wildcards.
[ INFO ] Looking up details of /dev/sde.
[ INFO ] Adding NSD None on storagenode.internal.cloudapp.net using device /dev/sde.
[ INFO ] Configuration updated
```

Vi legger også til en egen disk for ces_shared filsystemet som skal brukes med CES.

```
root@managernode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale nsd add /dev/sdf -p 10.0.0.5 -fs ces_shared -u dataAndMetadata -po system -fg 1
[ INFO ] Connecting to storagenode.internal.cloudapp.net to check devices and expand wildcards.
[ INFO ] Looking up details of /dev/sdf.
[ INFO ] The installer will create the new file system ces_shared if it does not exist.
[ INFO ] Adding NSD None on storagenode.internal.cloudapp.net using device /dev/sdf.
[ INFO ] Configuration updated
[ INFO ] Tip : If all node designations and any required protocol configurations are complete, proceed to check the installation con
```

4.3.3 Utrulling av filsystemet

Det er nesten klart for å installere GPFS og rulle ut filsystemet. Først konfigureres NTP-serverne som skal brukes av clusteret for å holde klokken oppdatert mellom nodene:

```
./spectrumscale config ntp -s 10.0.0.4,10.0.0.5,10.0.0.6
```

```
root@ManagerNode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale config ntp -l
[ INFO ] Current settings are as follows:
[ INFO ] Upstream NTP Servers(comma separated IP's with NO space between multiple IPs) is 10.0.0.4,10.0.0.5,10.0.0.6.
```

Konfigurerer så filsystemet med navn gpfs_01:

```
./spectrumscale config gpfs -c gpfs_01
```

Skrur av call-home funksjonen, da vi ikke skal bruke den i demonstrasjonen:

```
./spectrumscale callhome disable
```

Kjører precheck før installasjon for å verifisere at alt er i orden:

```
./spectrumscale install --precheck
```

Får beskjed om at alt er i orden for installasjon.

```
[ INFO ] Pre-check successful for install.
```

Installerer Spectrum Scale på nodene, og ser at alt går bra:

```
./spectrumscale install
```

```
[ INFO ] Installation successful. 3 GPFS nodes active in cluster gpfs_01.managernode. Completed in 2 minutes 59 seconds.
```

Så monteres GPFS-filsystemet under «/gpfs/gpfs_device»:

```
root@managernode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale filesystem modify gpfs_device -m /gpfs/gpfs_device
```

Ruller så til slutt ut filsystemet:

```
root@managernode:/usr/lpp/mmfs/5.0.4.1/installer# ./spectrumscale deploy
```

Ser at dette var vellykket. Spectrum Scale er nå ferdig installert og klart til bruk.

```
[ INFO ] Successfully installed protocol packages on 0 protocol nodes. Components installed: Chef (deployment tool), Filesystem, Performance Monitoring, GUI, FILE AUDIT LOGGING. it took 5 minutes and 5 seconds.
```

5. Administrasjon og drift av IBM Spectrum Scale

I dette kapitlet demonstreres en rekke av funksjonene i IBM Spectrum Scale ut ifra bedriftens behov og målene i prosjektet. Dokumentasjonen vil primært dreie seg rundt å se på hvordan man kan utføre diverse oppgaver og bruksområdene beskrevet i 2.2, som vi ser for oss er relevante når bedriften skal ta i bruk og drifte systemet. I hovedsak ser vi på hvordan dette gjøres med det grafiske brukergrensesnittet, da dette er brukervennlig og enkelt. Vi legger også til alternativet for noen av funksjonene ved bruk av kommandolinjen.

5.1 Initiell konfigurasjon

Nå som filsystemet er oppe og går, skal vi gjøre litt konfigurasjon i overgangen til daglig drift for å tilpasse grensesnittet slik vi ønsker det, automatisere oppgaver, og gjøre systemet klart for bruk.

5.1.1 Oppsett og bruk av kommandolinjen

Det meste vi skal gå igjennom her vil dreie seg om bruk av GUI-tjenesten til Spectrum Scale. Da denne tjenesten er den enkleste måten å håndtere vanlige arbeidsoppgaver i clusteret med, er det også mulig å bruke kommandolinjen for dette. I flere tilfeller blir vi nødt til å bruke kommandolinjen for å få gjort oppgaver, da GUI-et er en sekundær tjeneste, som ikke inneholder all funksjonaliteten.

For å bruke kommandolinjen må kommandoene inkluderes i kommandoskallet til nodene i clusteret. Dette gjøres ved å legge til stien til disse som en del av PATH-variabelen i kommandoskallet. For å automatisere dette gjør vi følgende på alle noder i clusteret:

```
«# cd /root»
```

```
«# nano .bashrc»
```

Legger til følgende nederst i filen:

```
«export PATH="$PATH:/usr/lpp/mmfs/bin"»
```

Vi kan dermed benytte Spectrum Scale kommandoer. Eksempelvis «**mmhealth**», som viser status på noder og cluster:

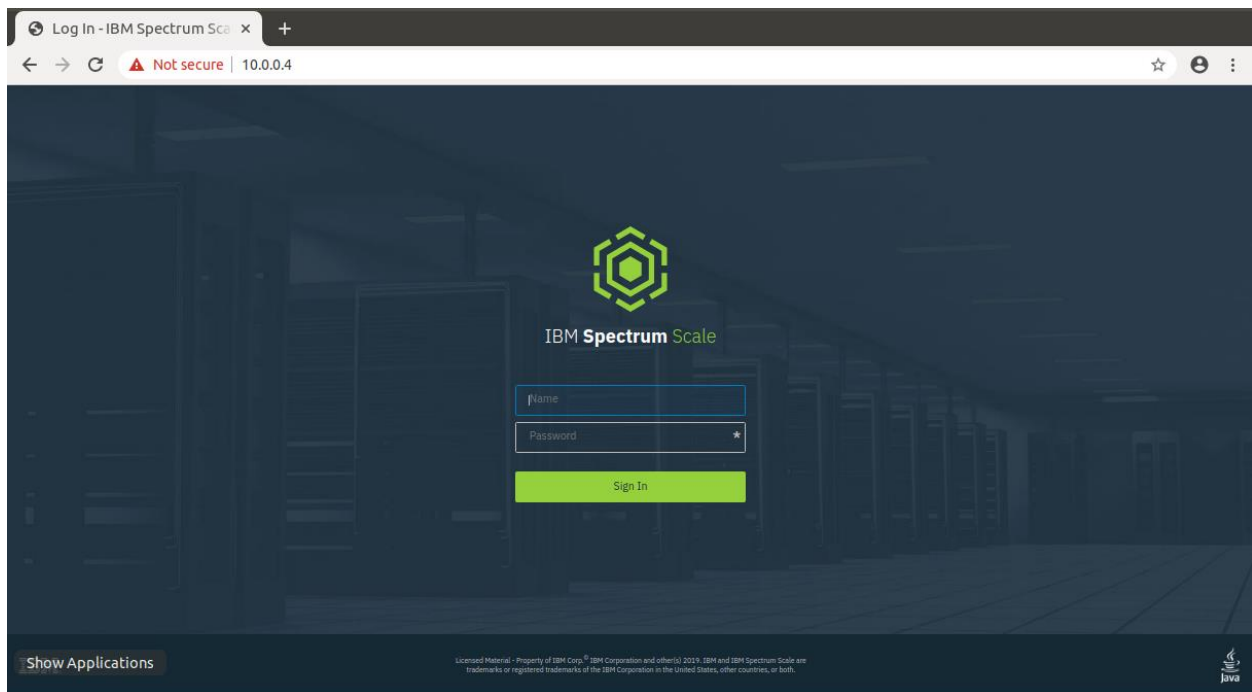
```
root@managernode:~# mmhealth node show
Node name:      managernode
Node status:    TIPS
Status Change: 32 min. ago

Component      Status      Status Change      Reasons
-----
GPFS           TIPS       32 min. ago        gpfs_pagepool_small
NETWORK        HEALTHY    32 min. ago        -
FILESYSTEM     DEGRADED   26 min. ago        pool-data_high_error(gpfs_device)
CESIP          HEALTHY    31 min. ago        -
GUI            HEALTHY    32 min. ago        -
PERFMON        HEALTHY    32 min. ago        -
THRESHOLD      HEALTHY    32 min. ago        -
root@managernode:~#
```

5.1.2 Oppsett av management GUI

Tilkobling og første gangs oppsett

Etter oppsett av GPFS-clusteret slik vi har dokumentert i denne rapporten, er det svært enkelt å ta i bruk GUI for Spectrum Scale. Dette er en oversiktlig tjeneste som gjør administrasjon av filsystemet lettere, og tilbyr en visuell fremstilling av hva som foregår i systemet. Ved å koble seg på adressen til en av GUI-serverne, får man tilgang til innloggingsportalen til GUI-tjenesten:

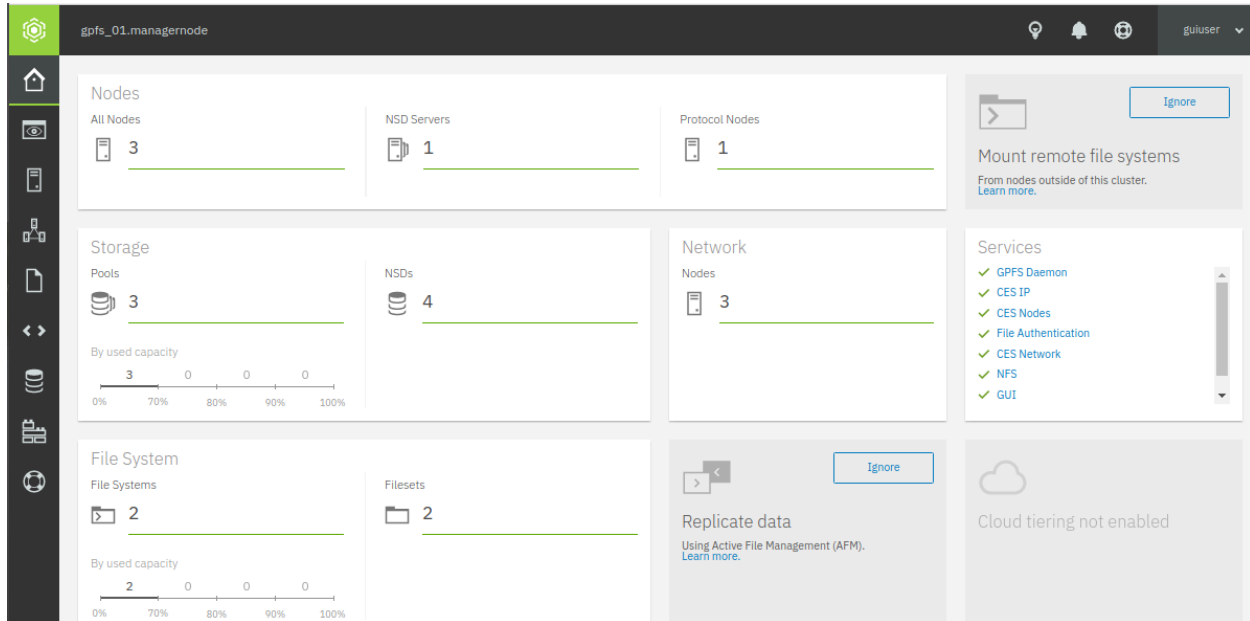


Figur 24 - GUI Innloggingsportal

Ved førstegangs innlogging får man opp noen instruksjoner om hvordan man legger til bruker i kommandoskallet. Dermed kan man logge seg inn i portalen.

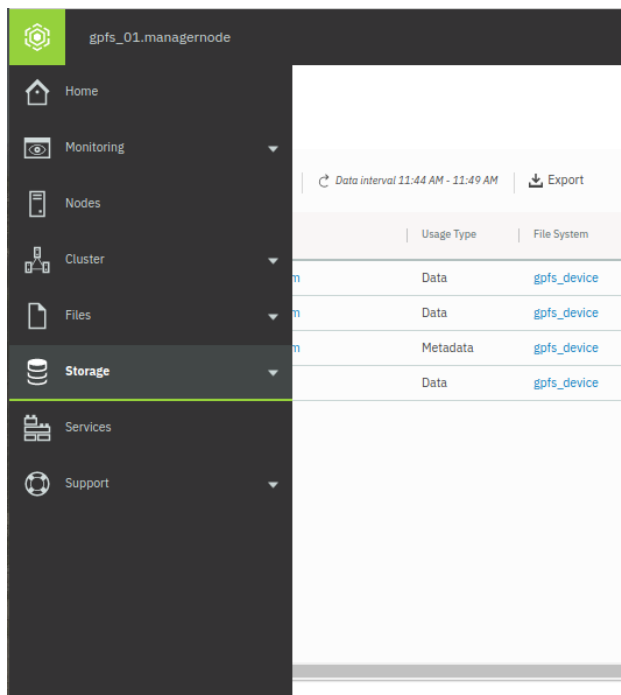
5.1.3 Oversikt/Hjemmeside

Etter at man har opprettet en bruker og logget seg inn på GUI-tjenesten, får man en generell oversikt over tilstanden til clusteret. Her ser vi antall noder, servere, lagring og filsystemer som er i drift. Delte filsystemer vil også dukke opp her.



Figur 25 - GUI Hjemmeside

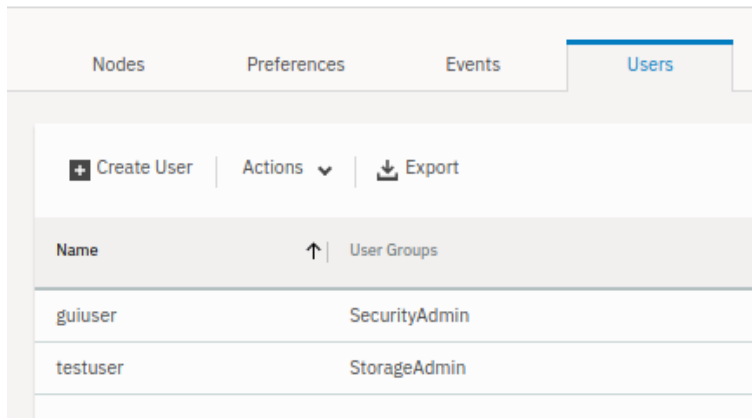
På venstre side ser vi en meny som kommer frem. Her finner vi de ulike kategoriene i grensesnittet.



Figur 26 - GUI Meny

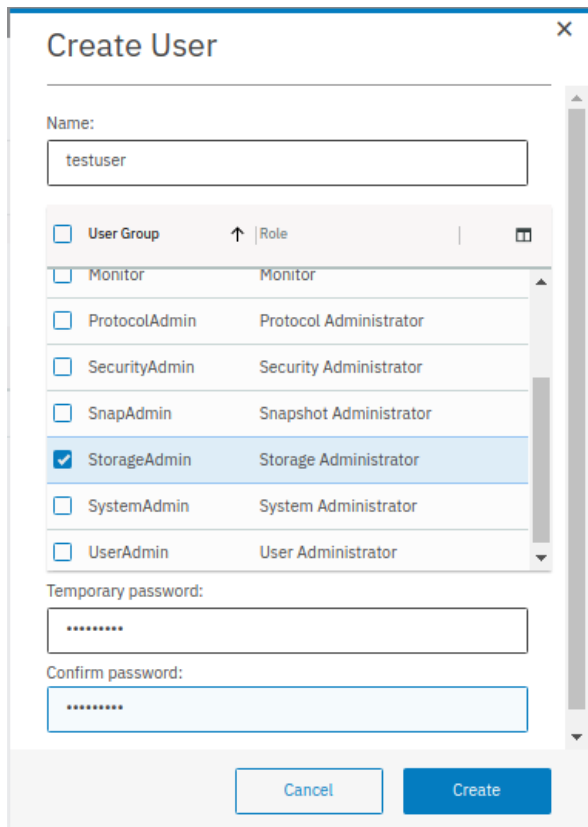
5.1.4 Sette opp brukere

Det kan hende at vi ønsker å konfigurere flere brukere med bestemte rettigheter i Spectrum Scale. For å gjøre dette går vi inn på **Services – GUI – Users** og klikker på «create user».



Figur 27 - GUI Brukertilgang

Her velger vi navnet på brukeren, hvilke rettigheter den skal ha, og et midlertidig passord. Vi velger at denne brukeren kun skal være storage administrator.



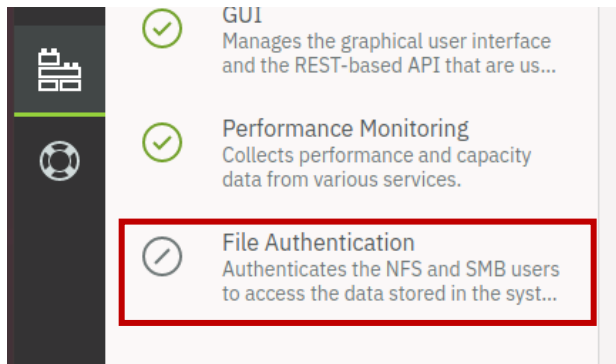
Figur 28 - GUI Ny bruker

5.1.5 Koble til ekstern brukerdatabase

File Authentication muliggjør å gi eksterne bruke rettigheter over filer og mapper når brukerne kobler på filsystemet via NFS eller SMB. For at CES skal fungere må File Authentication konfigureres.

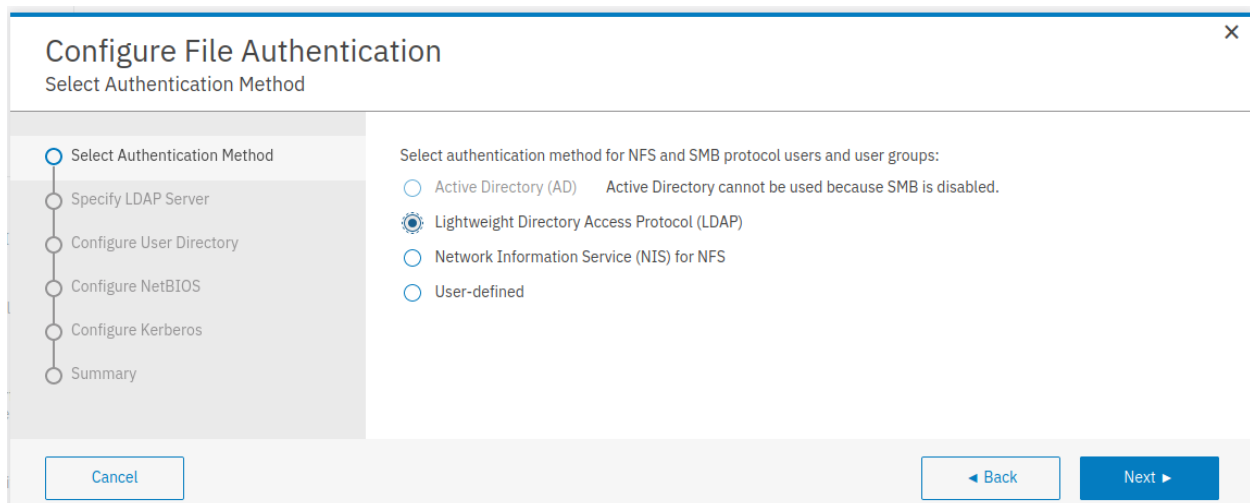
Konfigurasjonen gjøres på brukergrensesnittet til Spectrum Scale. Før du installerer må du ha en aktiv katalogtjeneste. Vi bruker i vårt oppsett LDAP, oppsettet av LDAP vises i kapittel 4.2.6.

Konfigurasjonen starter ved å gå services og File Authentication. Der velger vi å konfigurere «file authenticaton».



Figur 29 - GUI File Authentication

Vi velger å koble systemet mot LDAP serveren. Vi ser her at Spectrum Scale også har mulighet for å koble seg mot Windows Active Directory. Vi ser her på en ren Linux installasjon og trenger dermed bare LDAP.



Figur 30 - GUI Config FA

Her fyller vi ut tilsvarende informasjon som vi gjorde i oppsettet av phpLDAPadmin. Skriver inn IP-adressen til LDAP serveren og admin brukeren med passord.

The screenshot shows the 'Configure File Authentication' dialog box with the 'Specify LDAP Server' step selected. The left sidebar contains a vertical list of steps: 'Select Authentication Method' (checked), 'Specify LDAP Server' (selected), 'Configure User Directory', 'Configure NetBIOS', 'Configure Kerberos', and 'Summary'. The main area contains the following fields and options:

- LDAP server:** Text input field containing '10.0.0.9'.
- Port:** Text input field containing '389' with a dropdown arrow and '+'/'-' buttons.
- Use SSL connection
- Bind user DN:** Text input field containing 'cn=admin'.
- Password:** Password input field with a '*' icon and a Show password checkbox.

At the bottom, there are three buttons: 'Cancel', 'Back', and 'Next'.

Figur 31 - GUI Config FA 2

Vi fyller så ut det unike navnet til LDAP serveren

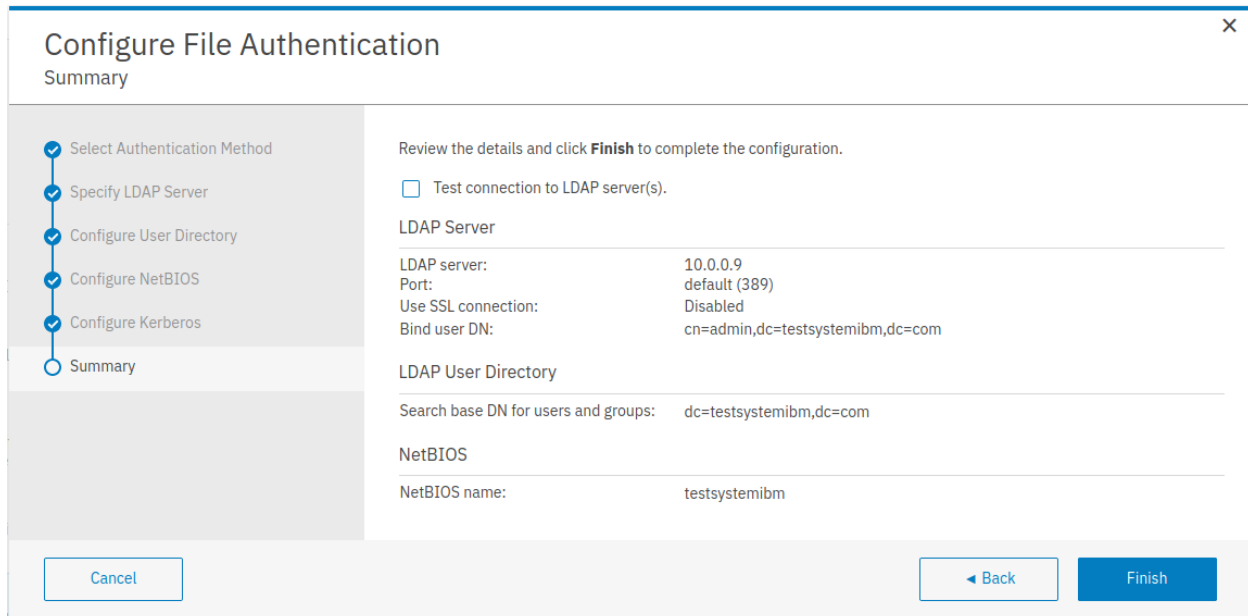
The screenshot shows the 'Configure File Authentication' dialog box with the 'Configure User Directory' step selected. The left sidebar contains a vertical list of steps: 'Select Authentication Method' (checked), 'Specify LDAP Server' (checked), 'Configure User Directory' (selected), 'Configure NetBIOS', 'Configure Kerberos', and 'Summary'. The main area contains the following fields:

- Search base DN for users and groups:** Text input field containing 'dc=testsystemibm,dc=com'.
- User suffix:** Text input field.
- Group suffix:** Text input field.
- Netgroup suffix:** Text input field.
- User object class:** Text input field.
- Group object class:** Text input field.
- User name attribute:** Text input field.
- User ID attribute:** Text input field.
- User email attribute:** Text input field.

At the bottom, there are three buttons: 'Cancel', 'Back', and 'Next'.

Figur 32 - GUI Config FA 3

Her ser vi den endelige konfigurasjonen av LDAP i Spectrum Scale. NetBIOS er navnet på tilkoblingen clusteret har mot LDAP serveren. Vi har her ikke konfigurert Kerberos eller SSL som gir sikker og kryptert kommunikasjon mellom LDAP, klientmaskinen og clusteret.



Configure File Authentication Summary

Select Authentication Method
Specify LDAP Server
Configure User Directory
Configure NetBIOS
Configure Kerberos
Summary

Review the details and click **Finish** to complete the configuration.

Test connection to LDAP server(s).

LDAP Server

LDAP server: 10.0.0.9
Port: default (389)
Use SSL connection: Disabled
Bind user DN: cn=admin,dc=testsystemibm,dc=com

LDAP User Directory

Search base DN for users and groups: dc=testsystemibm,dc=com

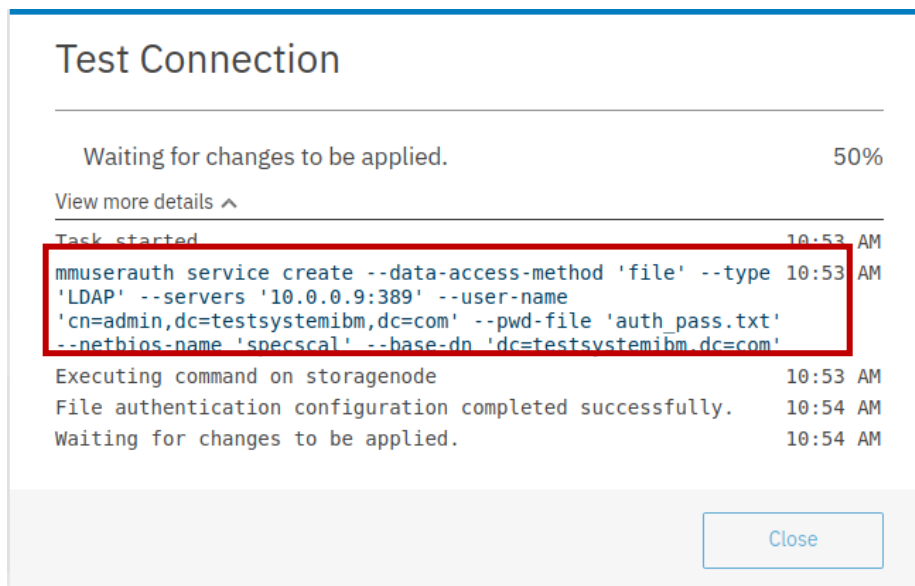
NetBIOS

NetBIOS name: testsystemibm

Cancel Back Finish

Figur 33 - GUI Config FA 4

Under ser vi oppgaven er fullført og venter på bekreftelse fra storagenoden. Her ser vi også kommandoen brukergrensesnittet sendte mot clusteret for konfigurasjon.



Test Connection

Waiting for changes to be applied. 50%

View more details ^

```
Task started 10:53 AM
mmuserauth service create --data-access-method 'file' --type 'LDAP' --servers '10.0.0.9:389' --user-name 'cn=admin,dc=testsystemibm,dc=com' --pwd-file 'auth_pass.txt' --netbios-name 'specscal' --base-dn 'dc=testsystemibm,dc=com' 10:53 AM
Executing command on storagenode 10:53 AM
File authentication configuration completed successfully. 10:54 AM
Waiting for changes to be applied. 10:54 AM
```

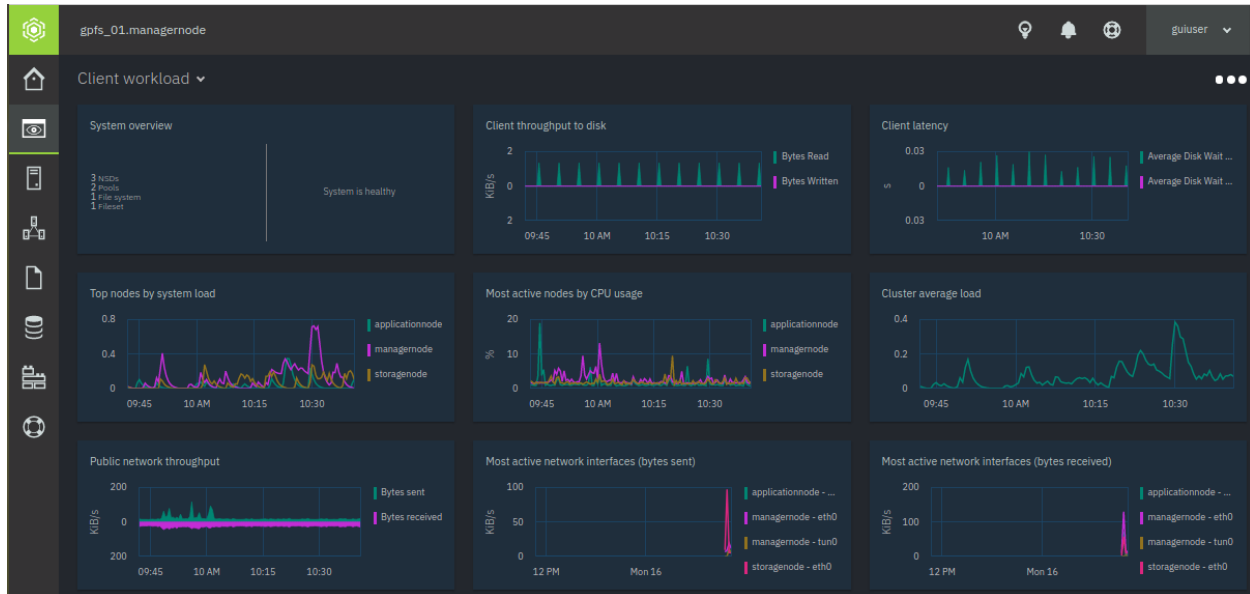
Close

Figur 34 - GUI Config FA 5

5.1.6 Egendefinert dashboard

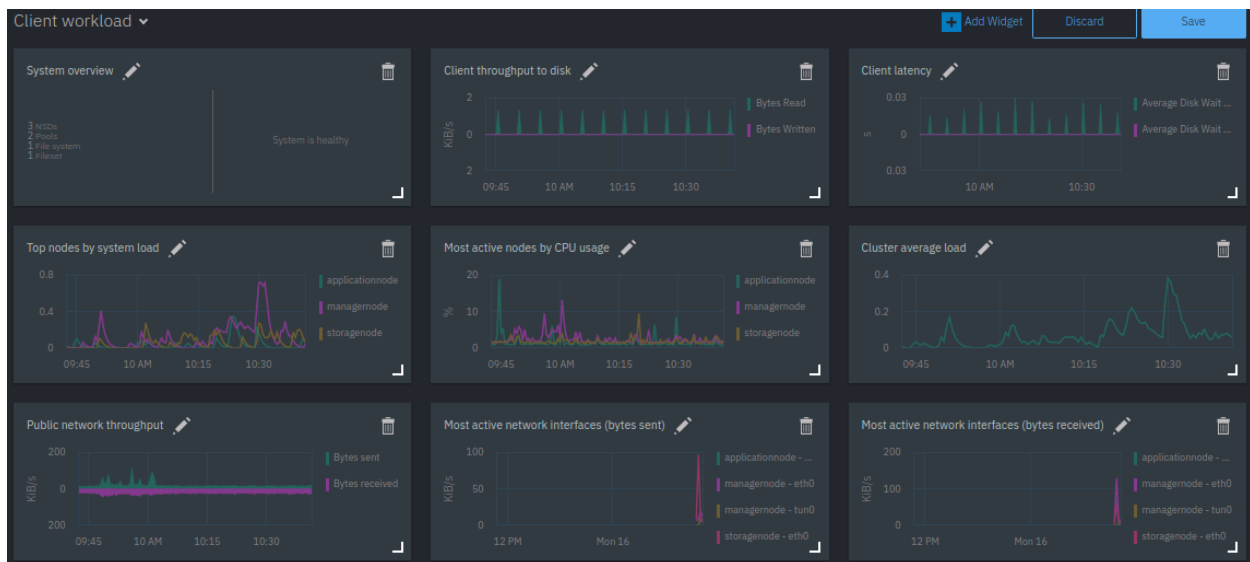
I Spectrum Scale GUI kan man egendefinere hvilke data og statistikk man ønsker å få presentert på dashboardet sitt. Slik kan man tilpasse oversikten utifra hva man trenger av informasjon og hvordan systemet er satt opp.

Ved å gå til **Monitoring – Dashboard** får vi se dashboardet.



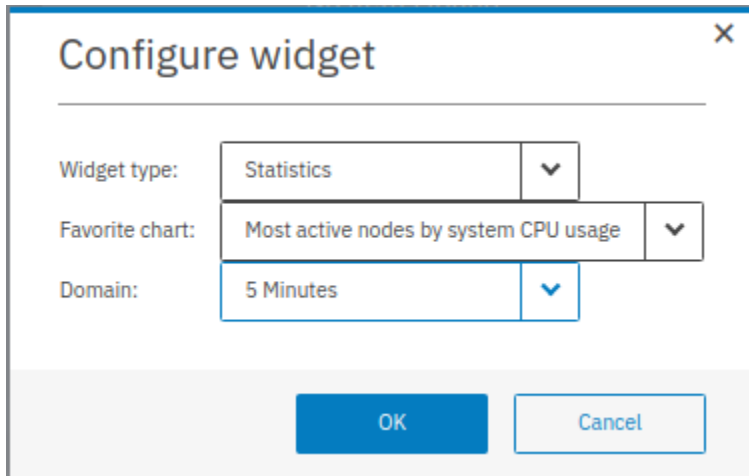
Figur 35 - GUI Dashboard

Øverst i høyre hjørne ser vi tre prikker. Dersom man trykker her kan man redigere hvilke widgets som vises på dashboardet:

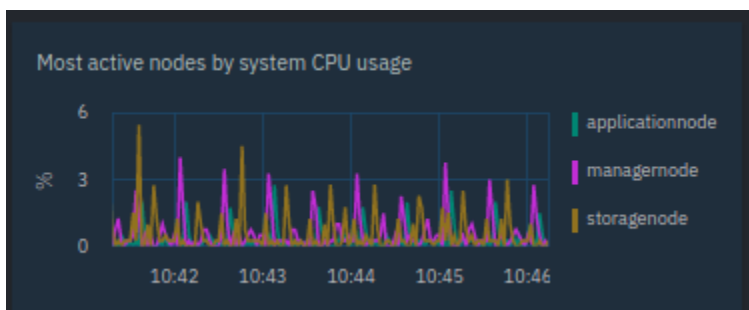


Figur 36 - GUI Edit Dashboard

Klikker vi på pennen på en av dem, får vi opp mulighet for å konfigurere det enkelte vinduet slik vi vil ha det. Vi kan velge mellom forskjellige typer widgets, for eksempel statistikk eller hendelser, hvilket fokusområde vi skal ha og tidsperspektiv.



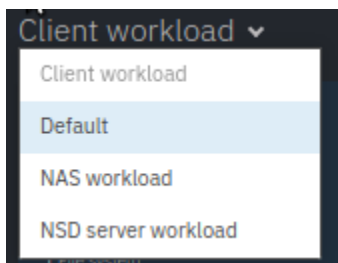
Figur 37 - GUI Configure Widget



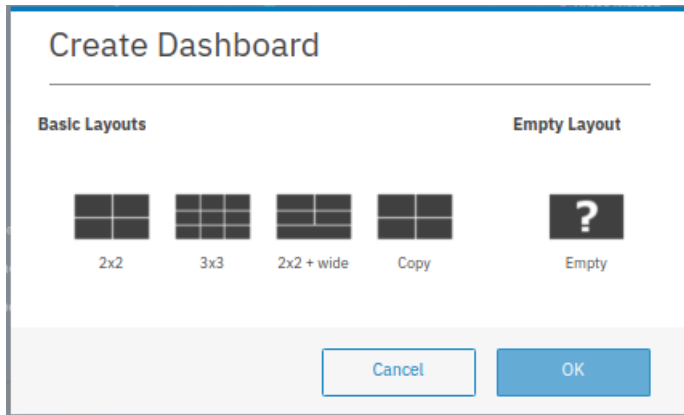
Figur 38 - GUI Widget

Vi kan også flytte dem dit vi vil ha dem, og slette eksisterende widgets.

Man kan velge forskjellige profiler, og lage sine egne:



Figur 39 - GUI Dashboard Profiles

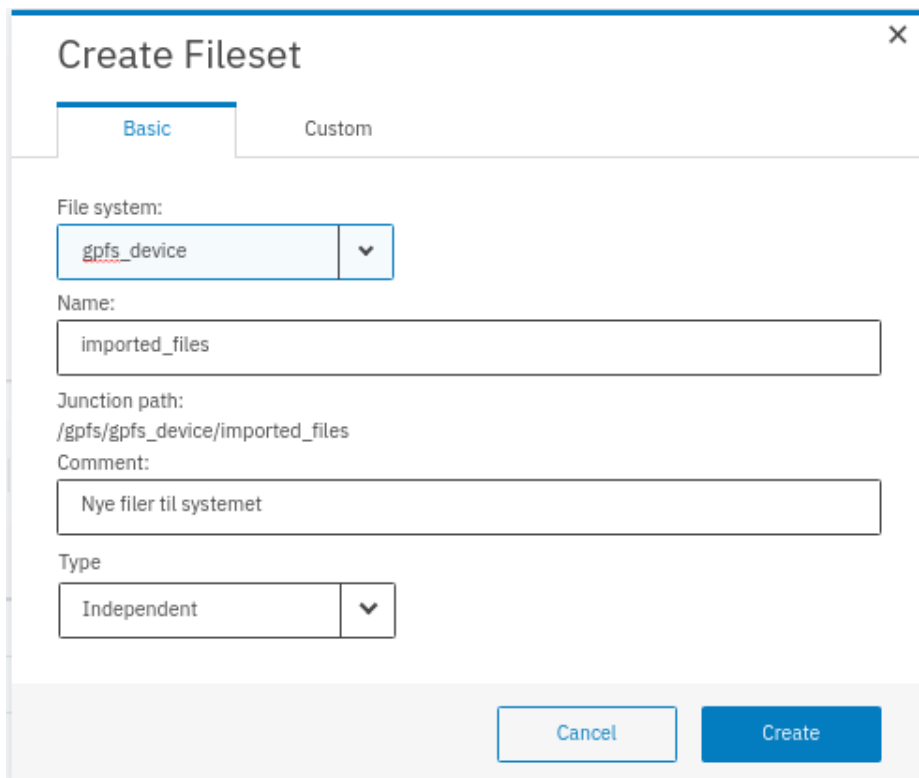


Figur 40 - GUI Create Dashboard

5.1.7 Opprette filsett

Et filsett er et subsett med filer innenfor filsystemet med et spesifikt formål. Her setter vi opp et filsett med navn «**imported_files**» i filsystemet «**gpfs_device**» med formålet å ta imot nye filer. Her kopieres alt av usorterte filer fra båten, før vi senere ved hjelp av regler flytter disse dit de skal være.

Vi oppretter et nytt filsett ved å manøvrere til **Files – Filesets – Create fileset**. Som du ser, blir det nye filsettet liggende under «**/gpfs/gpfs_device/imported_files**». Dersom vi ønsker, kan vi også endre stien til filsettet og eieren ved å klikke på «custom».



Figur 41 - GUI Opprett Filsett

Videre opprettes et filsett for filer som har blitt kopiert inn til systemet og prosessert av reglene våre. Her er tanken at man etter hvert kan flytte filer som nylig har blitt kopiert til filsystemet over i en mer permanent lokasjon. Ved å skille filene i ulike filsett kan vi redusere belastningen på systemet fordi vi ikke kjører unødvendig prosessering på filer.

Create Fileset [X]

Basic Custom

File system:
gpfs_device

Name:
processed_files

Junction path:
/gpfs/gpfs_device/processed_files

Comment:
Filer etter import

Type
Independent

Edit Access Control

Owner: root Owning group: root [Edit]

[Cancel] [Create]

Figur 42 - GUI Opprett Filsett 2

Her ser vi de nye filsettene i oversikten:

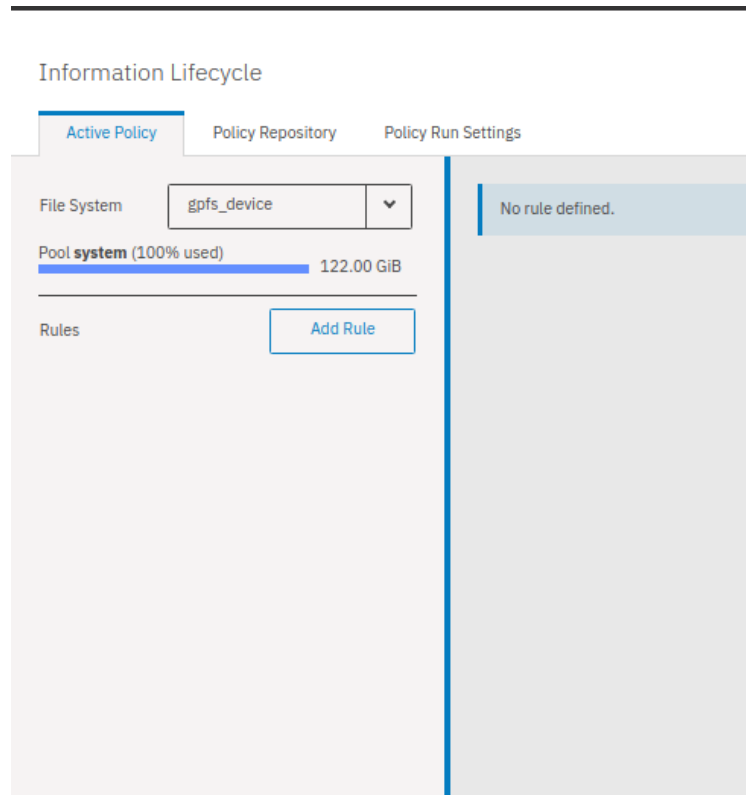
Name	↑ Remote Files...	File System	Parent	Used Capacity	Used Inodes	Inode Space Used
imported_files		gpfs_device		0 bytes	0	2
processed_files		gpfs_device		0 bytes	0	0
root		ces_shared		0 bytes	0	4127
root		gpfs_device		0 bytes	0	4123

Figur 43 - GUI Oversikt Filsett

5.1.8 Oppsett av policyer

Nå som vi har opprettet filsett innenfor filsystemet er det tid for å lage noen regler for hvordan data skal håndteres når de kommer inn i systemet. Det første vi trenger er en policy som bestemmer hva som skal skje med nye filer som opprettes innenfor filsystemet. Dette kalles en «file placement policy». Siden alle filer først skal kopieres inn i storage pool «silver» for videre analyse, lager vi en regel som sier at nye filer skal plasseres på disse diskene. I eksempelet tilsvarer dette «system» storage poolen. Du kan lese mer om policyer under 3.3.1.

For å sette opp en policy navigerer man til **Files – Information Lifecycle**.

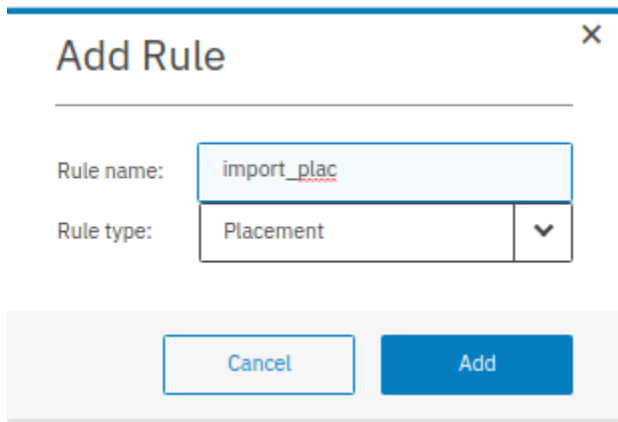


Figur 44 - GUI Policy 1

File placement policy

Først oppretter vi som sagt en policy for å håndtere alle nye filer som kommer inn i systemet. Her bestemmer vi at de nye filene først plasseres i storage pool «system». Dette tilsvarer storage pool «silver», altså vanlige harddisker.

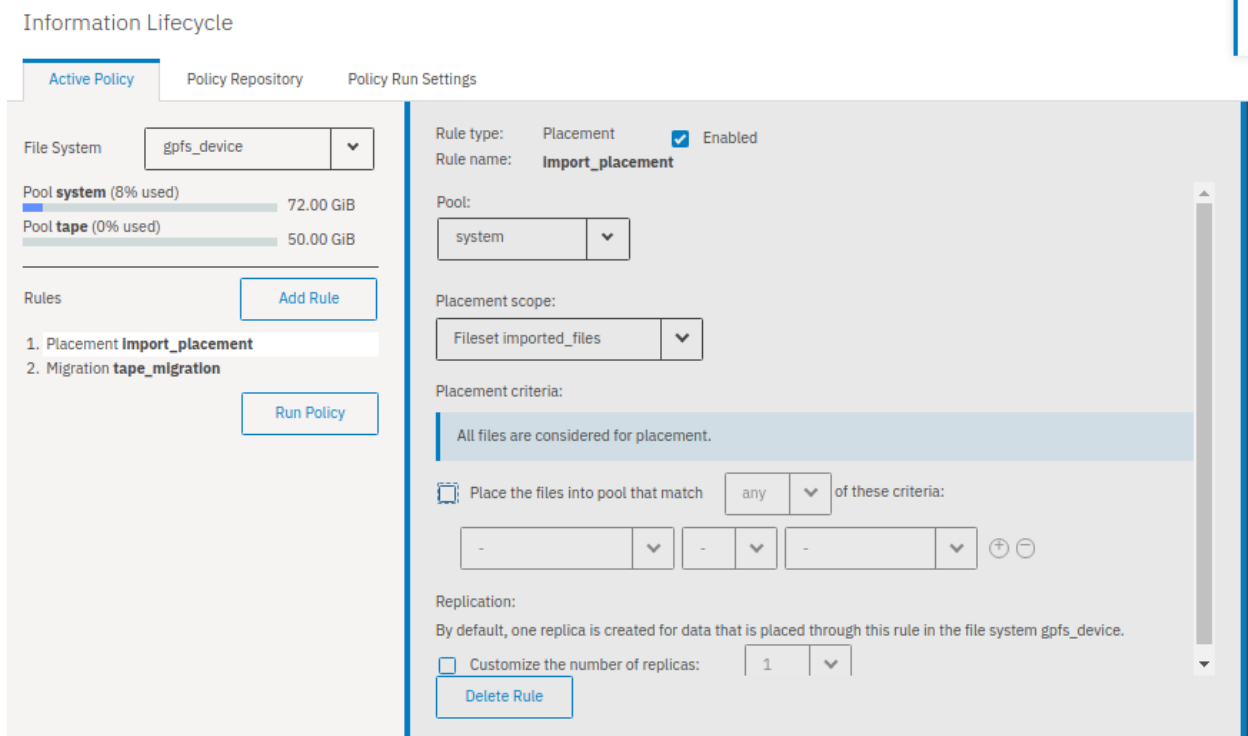
Klikker på «add rule», og får opp menyen for å legge til regler. Her ser vi at vi kan velge kilde og mål, grenseverdier for å trigge regelen, hvilke filsett som skal migreres og hvordan filene identifiseres.



The screenshot shows a dialog box titled "Add Rule" with a close button (X) in the top right corner. It contains two input fields: "Rule name:" with the text "import_plac" and "Rule type:" with a dropdown menu showing "Placement". At the bottom, there are two buttons: "Cancel" and "Add".

Figur 45 - GUI Add Rule

Her velges det at filene skal plasseres i «system» storage pool, og at filene som skal vurderes ligger under filsettet «imported files». Dette er altså filsettet som nye filer skal kopieres inn i.

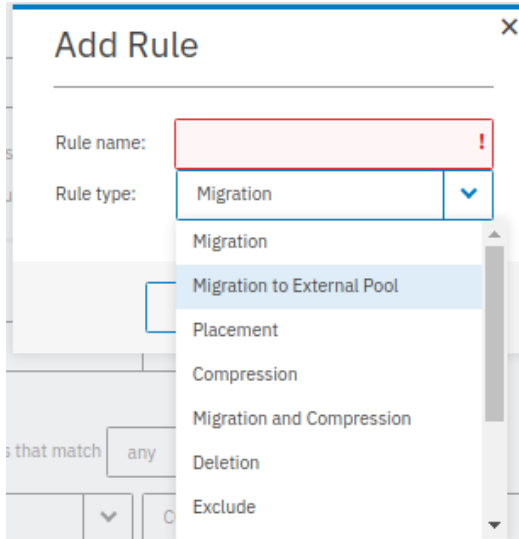


The screenshot shows the "Information Lifecycle" configuration page. The "Active Policy" tab is selected. The "File System" is "gpfs_device". The "Pool system" is 8% used (72.00 GiB) and "Pool tape" is 0% used (50.00 GiB). The "Rules" list shows "1. Placement import_placement" and "2. Migration tape_migration". The "Add Rule" button is visible. The "Run Policy" button is also visible. The "Rule type" is "Placement" and "Enabled". The "Rule name" is "import_placement". The "Pool" is "system". The "Placement scope" is "Fileset imported_files". The "Placement criteria" section shows "All files are considered for placement." and "Place the files into pool that match any of these criteria:". The "Replication" section shows "By default, one replica is created for data that is placed through this rule in the file system gpfs_device." and "Customize the number of replicas: 1". The "Delete Rule" button is visible.

Figur 46 - GUI Import Placement

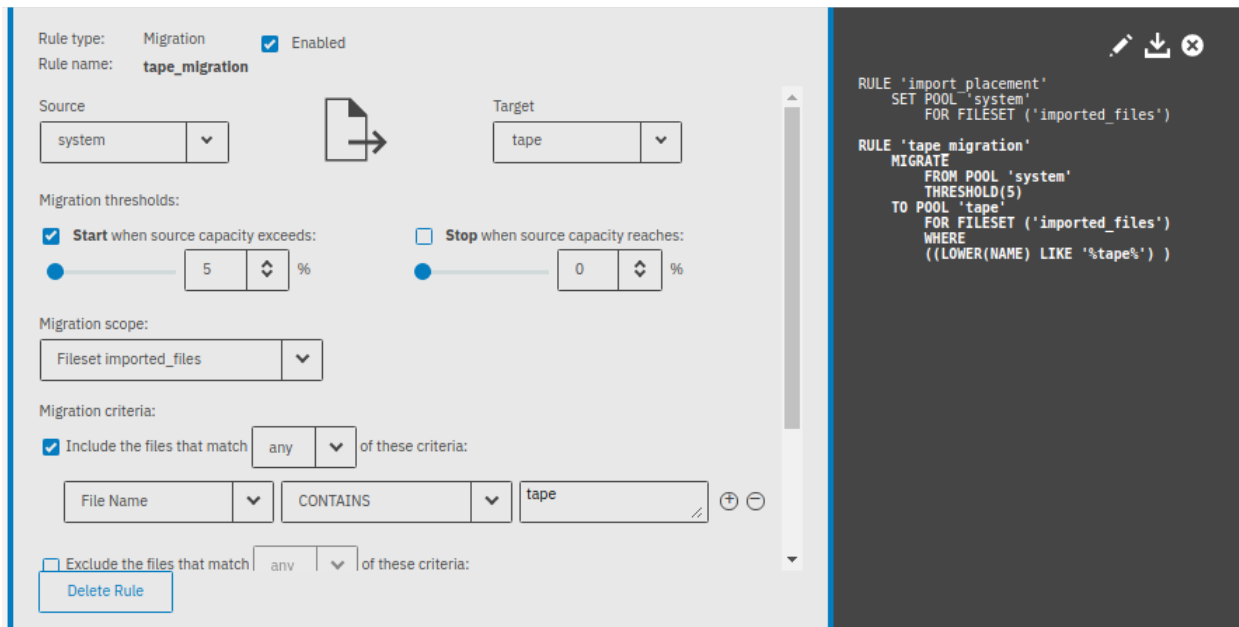
File management policy

I dette eksempelet ser vi på hvordan en kan sette opp en policy for å migrere spesielle filer til egne områder. Dersom filene skal migreres til en ekstern pool, slik som ved bruk av LTFS og Spectrum Archive, velger vi «migration to external pool». Dette vil da typisk være tilfellet om vi skal eksportere filer til tape.



Figur 47 - GUI Migration Rule

Vi setter opp en enkel policy for å migrere alle filer som inneholder ordet «tape» i filnavnet til storage pool «tape». For regelen velger vi at data skal flyttes fra kilden «system» til storagepool «tape». Vi velger en grenseverdi på 5%. Filsettet som skal vurderes er hele processed_files. Dette vil da være filer som har gjennomgått den initielle prosesseringen, og som skal vurderes for arkivering.



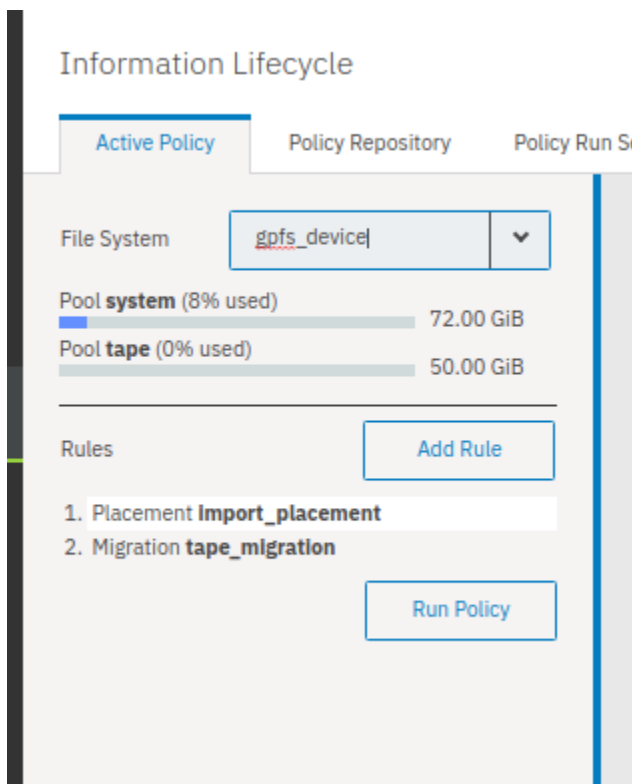
Figur 48 - GUI Migration Rule Config

Til høyre ser vi også hvordan syntaksen for regelen blir. I dette vinduet kan man redigere koden manuelt om det er endringer man ønsker å gjøre med regelen. Slik vil også syntaksen se ut dersom vi skal skrive våre egendefinerte regler og importere dem til systemet. Reglene skrives i et spørrespråk som ligner på SQL. Mer om dette finner man i dokumentasjonen for Spectrum Scale. For å importere egne regler går man under **Files – Information Lifecycle – Policy Repository – Create Policy**, og velger «select file».

```
RULE 'tape_migration'  
MIGRATE  
  FROM POOL 'system'  
  THRESHOLD(5)  
  TO POOL 'tape'  
  FOR FILESET ('imported_files')  
  WHERE  
    ((LOWER(NAME) LIKE '%tape%') )
```

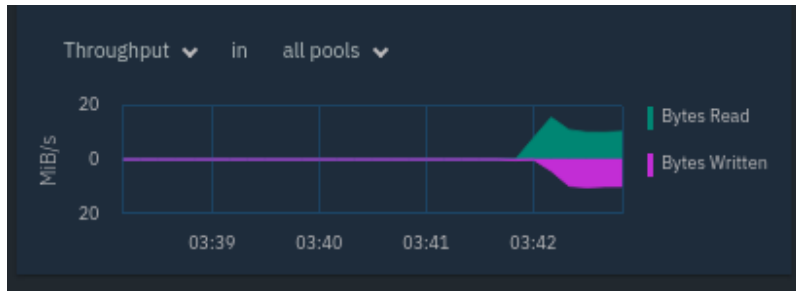
Figur 49 - GUI Syntaks for regel

Når regelen er opprettet kan vi velge å kjøre den ut ved å trykke på «run policy».



Figur 50 - GUI Regler Oversikt

Umiddelbart ser vi at filsystemet begynner å skrive over fra storagepool «system» til «tape».



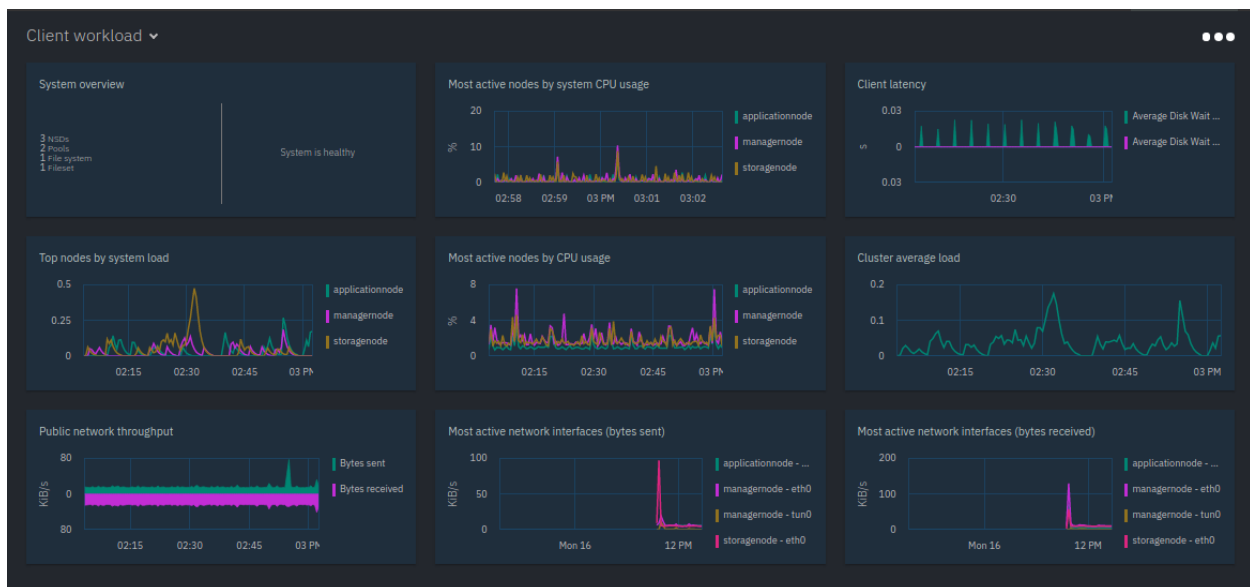
Figur 51 - GUI Skriver til tape

5.2 Daglig drift

Dette delkapittelet omhandler den daglige driften av IBM Spectrum Scale. Her ser vi på bruksområder som vi ser for oss vil være typiske i det daglige for IT hos bedriften. Dette er oppgavene som systemet hovedsakelig skal brukes til, slik som å hente inn ny data, håndtere dataen gjennom livsløpet, og overvåke systemet underveis.

5.2.1 Systemhelse

Noe av det viktigste vi ønsker å bruke GUI-et til er å holde oversikt over hvilke ressurser som blir brukt og hvordan helsen til systemet er for å forutse og unngå problemer. Slik du så i kapittel 5.1.3 får vi en generell oversikt over systemets tilstand allerede når vi kommer inn på hjemmesiden til brukergrensesnittet. Men det finnes flere måter å overvåke helsen til Spectrum Scale på. Et godt utgangspunkt er gjennom dashbordet vi konfigurerte i 5.1.6.

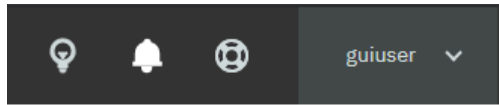


Figur 52 - GUI Dashboard 2

Her kan vi samle all den informasjonen vi ønsker å se om systemet under daglig drift, alt fra hendelser til filsett eller statistikk over noders ressursbruk.

Tips

Enten ved å gå til **Monitoring – Tips**, eller ved å trykke på lyspæren får man opp forslag til endringer for å øke ytelsen og stabiliteten.



Figur 53 - GUI Meny 2

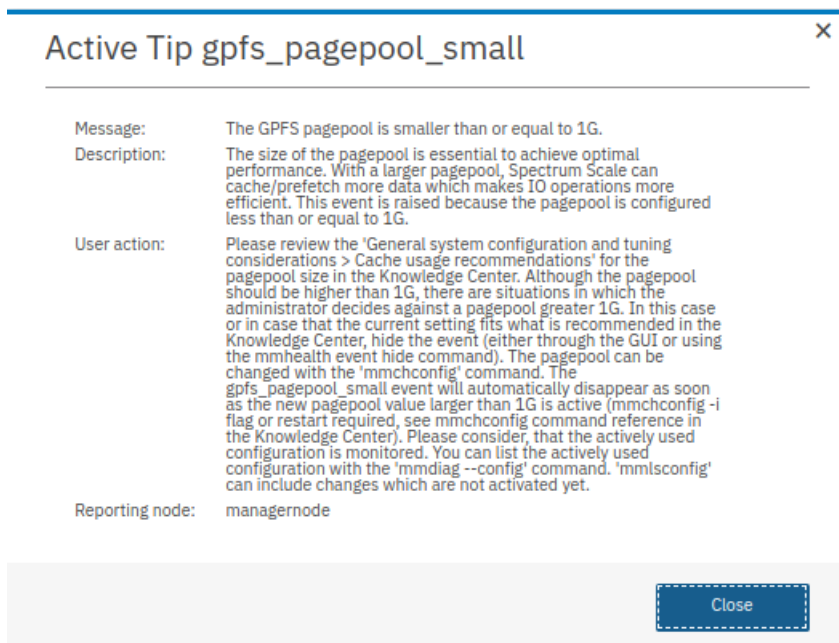
Her ser vi for eksempel at pagepool-variabelen er lavere enn anbefalt, noe som påvirker ytelsen. Meldingen sier at vi burde sette denne størrelsen til over 1GB.

Tips

Current	Tip Name	Event Time	Active Until	Message
🟡	gpfs_pagepool_small	04.03.20 09:01:59		The GPFS pagepool is smaller than or equal to 1G.
🟡	gpfs_pagepool_small	04.03.20 08:31:01		The GPFS pagepool is smaller than or equal to 1G.
🟡	gpfs_pagepool_small	04.03.20 08:30:36		The GPFS pagepool is smaller than or equal to 1G.

Figur 54 - GUI tips

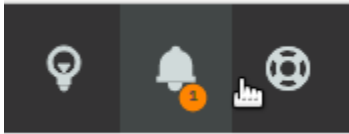
Ved å høyreklikke på meldingen og velge «properties» får vi en mer detaljert beskrivelse av meldingen. Her ser vi hva som er problemet, hvorfor, hva vi kan gjøre, og hvilken node som har rapportert.



Figur 55 - GUI Tips detaljert

Notifikasjoner og hendelser

Videre under **Monitoring – Events** får vi opp alle hendelser som har oppstått i systemet. Dette kan man også se ved å trykke på bjellen i øvre høyre hjørne:



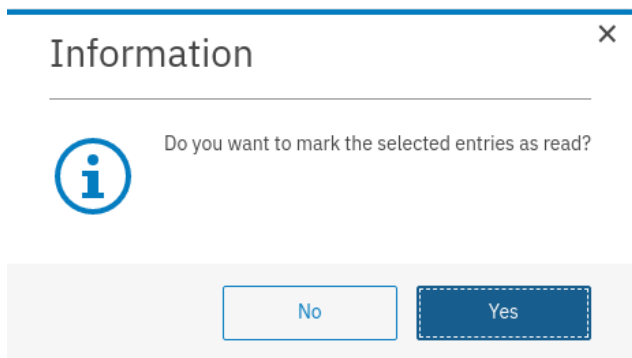
Figur 56 - GUI Notifikasjon

Her ser vi at det har skjedd et mislykket forsøk på innlogging til GUI-tjenesten

Severity	Event Time	Reporting Node	Event Name
Warning	3/4/20 10:47:36 AM	managernode	gui_login_attempt_failed

Figur 57 - GUI hendelse

Markerer meldingen som lest



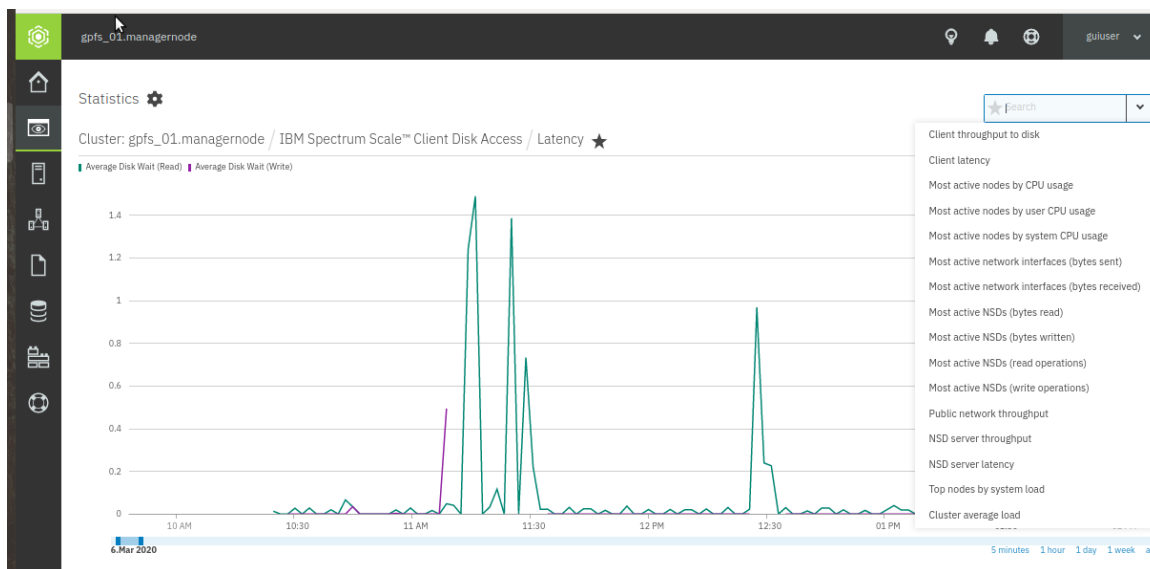
Figur 58 - GUI markere hendelse

Dersom man ønsker å få tilsendt informasjon om hendelser kan man under **Monitoring – Event Notifications** konfigurere sending til epost:

Figur 59 - GUI Epost

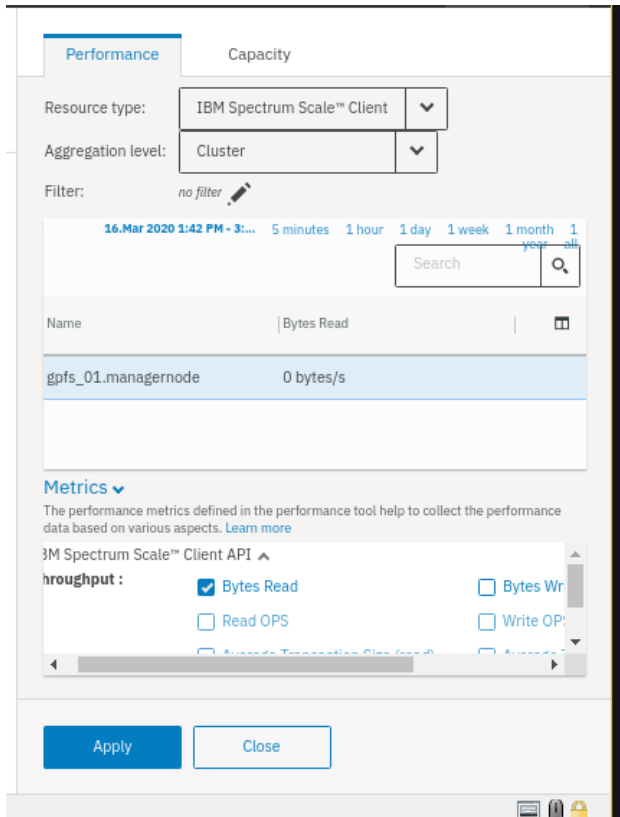
Statistikk

Dersom vi ønsker mer detaljert statistikk for spesifikke deler av systemet, kan vi gå under **Monitoring – Statistics**. Som du ser til høyre kan vi også søke etter den informasjonen vi ønsker. Her kommer det opp forslag til de vanligste dataene vi kan hente inn.



Figur 60 - GUI statistikk

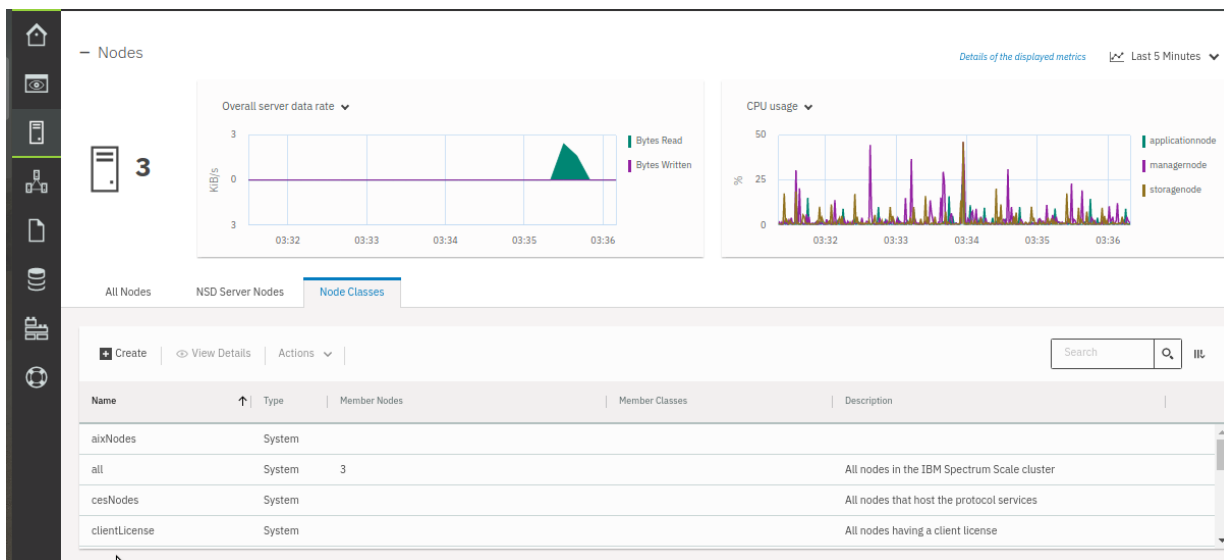
Ved å klikke på prikkene i høyre hjørne får man mulighet til ytterligere definering av statistikken:



Figur 61 - GUI detaljert statistikk

Noder

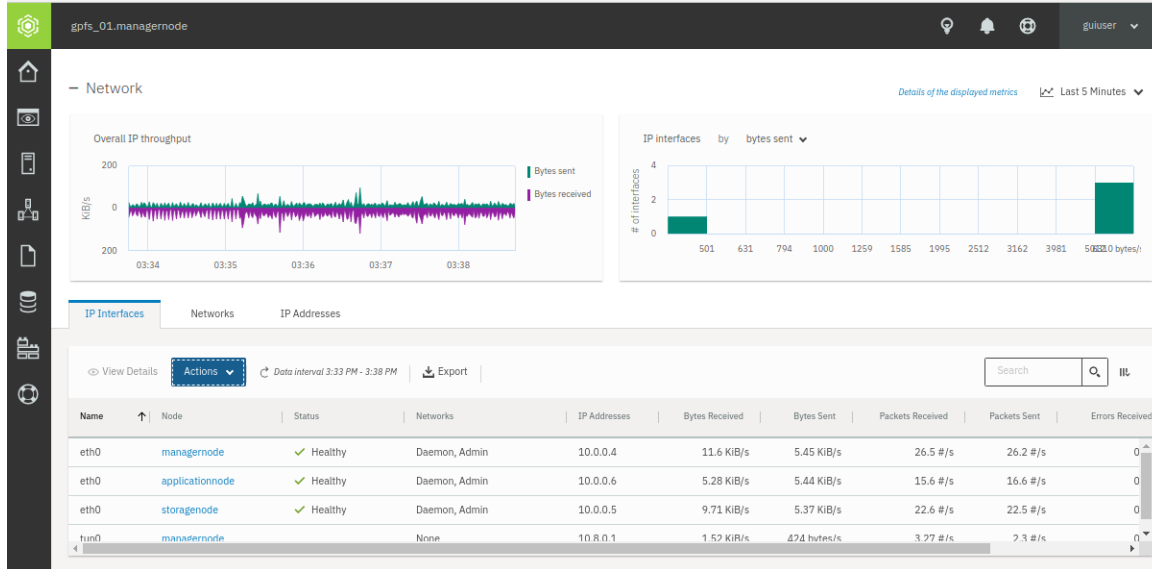
Vi kan se helsen til de ulike nodene i clusteret under **Nodes**. Her kan vi også sortere nodene basert på type og lage egne klasser som vi legger nodene våre inn i.



Figur 62 - GUI Nodeinformasjon

Cluster

For å se hvordan ting står til med selve clusteret kan vi enten gå inn på **Cluster – Network** eller **Cluster – Remote connections**. «Remote connections» er bare aktuelt dersom vi har koblet eksterne clustre til filsystemet vårt. «Network» viser nettverksenheter, nettverk og ip-adresser, og status på disse:

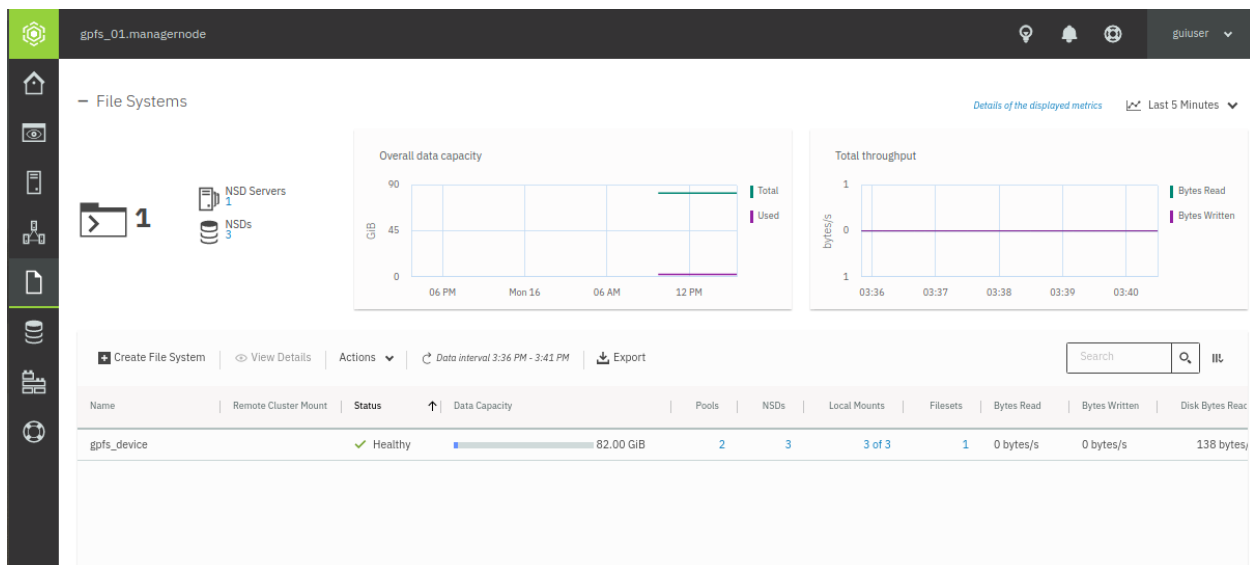


Figur 63 - GUI Clusterinformasjon

Filsystemet

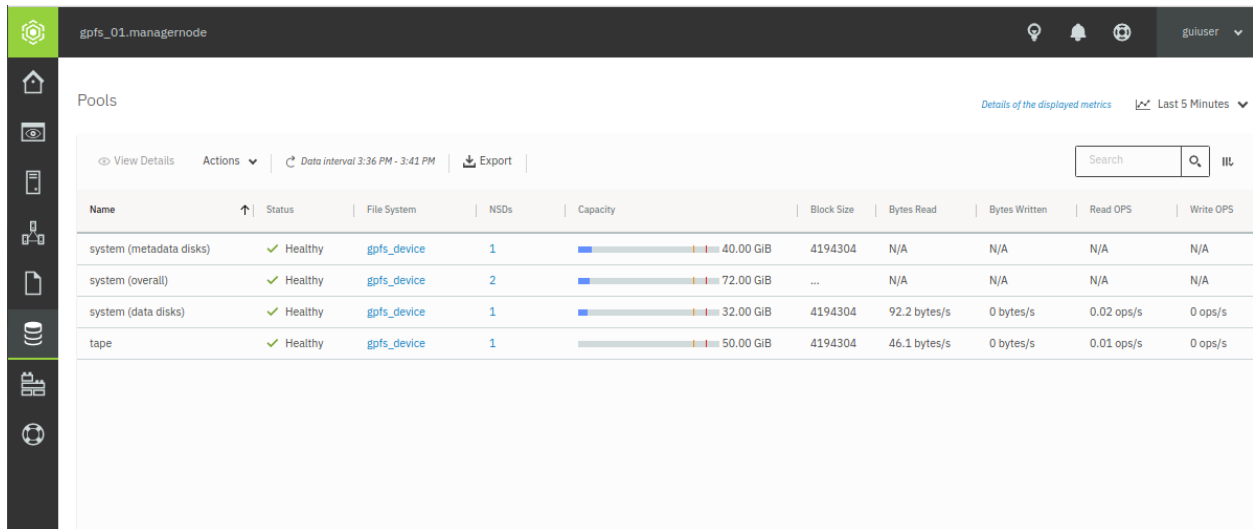
For å vise status og kapasitet på filsystemet, kan vi enten gå under **Files – File systems**, eller **Storage – Pools**:

Her ser vi kapasitet, skrivehastighet og status på filsystemet.



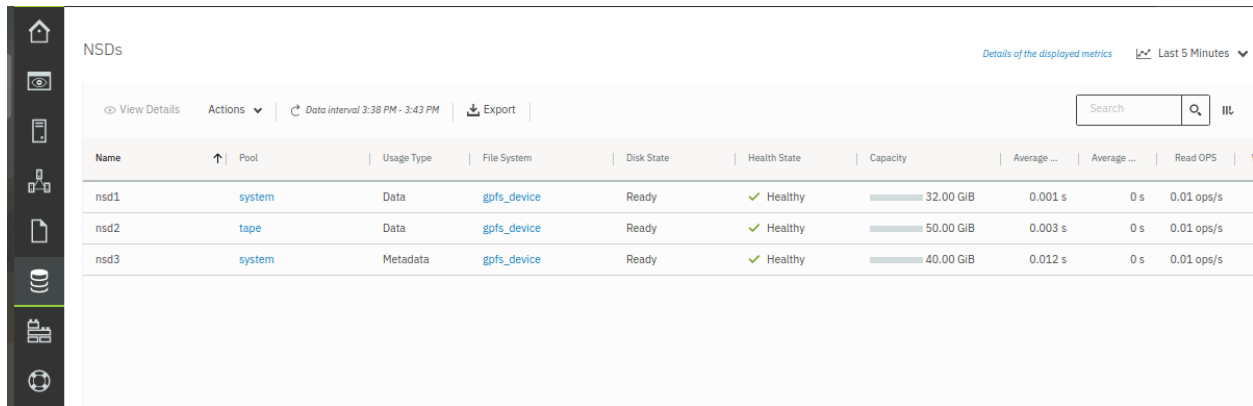
Figur 64 - GUI Filsystem informasjon

Under ser du kapasitetsforbruk på de ulike storage-poolene vi har definert for filsystemet.



Figur 65 - GUI Storagepools informasjon

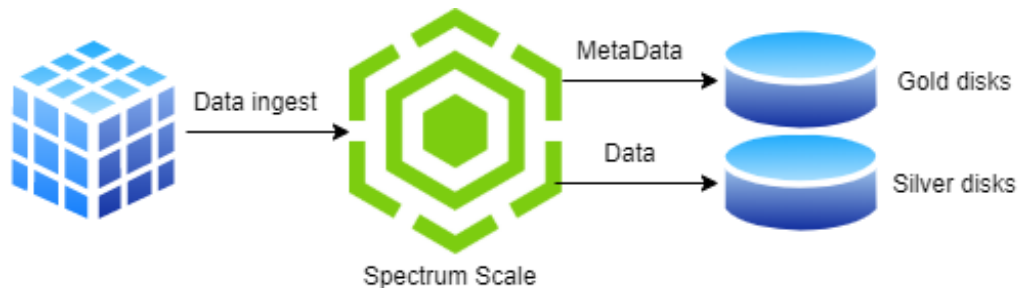
Vi kan også se på de respektive NSDene som hører til filsystemet:



Figur 66 - GUI NSD informasjon

5.2.2 Inntak av data

Den viktigste funksjonen til systemet er at det klarer å importere data effektivt og oversiktlig, og plassere den på riktig medium basert på datatype. Et av kravene til Kongsberg Seatex er å kunne forholde seg til ett filhierarki uten å vite hvor dataen er fysisk plassert, og derfor må dette skje sømløst. Når det kommer inn ny data fra båtene, skal alt av filer på disse diskene lempes inn i filsystemet, og policyene som vi har satt opp skal automatisk håndtere dette. På figur 70 er ny data betegnet som «Data ingest», og representerer det som kommer inn av filer ved importering til filsystemet. I praksis vil dette være en disk- eller tapestasjon som kobles til clusteret, og filene kopieres over til filsettet som vi opprettet i 5.1.7, «`/gpfs/gpfs_device/imported_files`».



Figur 67 - Data Ingest

Metadata er data som beskriver filen. Det er standard fildata som eier, tidspunkt og størrelse. Spectrum Scale støtter også metadata som brukerstyrt metadata, Access Control List (ACL) og log filer. Til metadataen ønsker vi så lav aksestid som mulig, så vi legger den på raske SSD-disker i Storagepool Gold. Dette har vi allerede definert i oppsettet av diskene i clusteret (se 4.3.2), så metadata vil automatisk havne her. Tagger kan legges til i metadataen og brukes til å filtrere data. Dette kan effektivisere søk og analyse av data.

Selve dataen legges i førsteomgang inn i Storagepool Silver. Her er det derfor viktig at vi har nok plass til buffer når nye filer kommer inn i systemet. Dette lagringsområdet skal ha relativt høy ytelse og håndtere store mengder data. Herfra analyseres dataen og flyttes den dit den skal være. Hvordan vi tilpasser innholdet kan effektivisere og automatisere filbehandling i systemet. Det er derfor viktig at vi har så god oversikt som mulig over hvilke data vi kommer til å samle inn, slik at vi kan konfigurere gode regler for å håndtere disse.

Ved hjelp av policyene vi satt opp i 5.1.8 kan vi prosessere den nye dataen. Disse oppgavene vil dreie seg om å plassere den på rett lagringsmedium gjennom livssyklusen. Dette vil avhenge av typen data bedriften har, og hvilke regler de setter for deres data.

5.2.3 Feilhåndtering

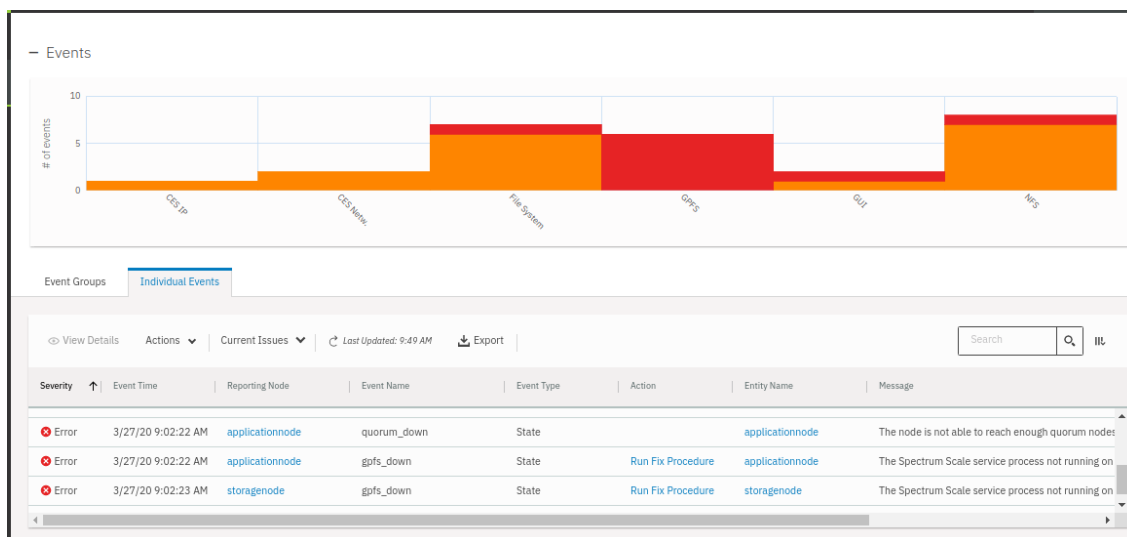
Det kan hende at det oppstår feil på filsystemet under den daglige driften. Dersom systemet er satt opp med tilstrekkelig grunnlag for failover og med passende grad av redundans, vil forhåpentligvis ikke filsystemet bli skadet eller driften stanse, og feilhåndteringen vil dreie seg om å få enheten som har feilet opp igjen så fort som mulig. Det er viktig at man planlegger gode rutiner for å håndtere kjente feil, slik at vi minimerer risikoen for at flere ting går galt samtidig, noe som kan lede til korrupsjon og datatap. Her ser vi på hvordan vi kan planlegge og reagere på vanlige feil. Bedriften bør utarbeide en omfattende risikoanalyse knyttet til driften av et virkelig system, der man også tar mer alvorlige hendelser i betraktning.

Komponenten som sannsynligvis kommer til å ryke oftest er diskene. En kan regne med at det bare er et spørsmål om tid før en disk slutter å fungere i slike sammenhenger. Da er det viktig at man har gjort klart prosedyrer for å skifte disker raskest mulig, og få filsystemet tilbake til optimal tilstand. Da filsystemet bruker redundante disk-array, vil det ikke ødelegge filsystemet dersom en disk går ned. Å fikse problemet vil dreie seg om å kontrollere disken, og eventuelt bytte den ut med en ny. Dermed må man legge til en ny disk i den storage poolen som den gamle hørte til, og la filsystemet reparere seg selv.

Det kan også oppstå andre feil slik som problemer i nettverket på grunn av enten fysiske eller logiske årsaker. I tillegg kan vi få nodefeil i clusteret. Dersom en node går ned kan dette skyldes flere ting. Maskinen kan ha mistet nettverkstilkobling, det kan ha skjedd en programfeil, eller maskinen kan ha mistet strøm. Uansett hva som skjer, må vi være forberedt på å håndtere feilen. Igjen bør en risikoanalyse fortelle bedriften hvilke feil en kan vente seg, hvilke konsekvenser dette vil ha, og tiltak for å unngå kompromitterte systemer.

Håndtere feil i GUI

Slik du så i 5.2.1, får du opp alle hendelser i høyre hjørne under notifikasjons-bjellen. I tillegg kan man navigere til **Monitoring – Events** for å finne den samme menyen. Her får vi en oversikt over alle advarsler og feil på filsystemet, og informasjon om hvert enkelt problem:



Figur 68 - GUI Systemhendelser

Her ser vi for eksempel at GPFS er nede på storgenoden. Vi kan dobbeltklikke på innslaget for å få mer informasjon. For å forsøke å fikse problemet automatisk, kan vi trykke på «Run fix procedure».

The screenshot shows a window titled 'gpfs_down' with a dark theme. At the top, there is a breadcrumb trail: Error > 3/27/20 9:02:23 AM > storagenode > gpfs_down > State > Run Fix Procedure > storagenode. Below the breadcrumb, there are two tabs: 'Overview' (selected) and 'Event Thread'. To the right of the tabs are two buttons: 'Mark as read' and 'Run Fix Procedure'. The main content area is titled 'Component:' and contains a list of key-value pairs:

- Event name: gpfs_down
- Component: GPFS
- Entity type: Node
- Entity name: storagenode
- Occurrences: 1
- Event time: 3/27/20 9:02:23 AM
- Active until:
- Message: The Spectrum Scale service process not running on this node. Normal operation cannot be done
- Description: The Spectrum Scale service is not running. This can be an expected state when the Spectrum Scale service is shutdown
- Cause: The Spectrum Scale service is not running
- User action: Check the state of the Spectrum Scale filesystem daemon, and check for the root cause in the /var/adm/ras/mmfs.log.latest log
- Reporting node: storagenode
- Event type: Active health state of an entity which is monitored by the system.

Figur 69 - GUI Hendelsesinformasjon

Veilederen viser hvilke alternativer vi kan forsøke for å fikse problemet.

The screenshot shows a dialog window titled 'Fix Procedure: Start IBM Spectrum Scale™ Daemon' with a close button (X) in the top right corner. Below the title is the word 'Introduction'. On the left side, there is a vertical navigation pane with two items: 'Introduction' (selected with a blue circle) and 'Summary' (with a grey circle). The main content area contains the following text:

The Spectrum Scale service process not running on this node. Normal operation cannot be done

- Start the IBM Spectrum Scale™ daemon on the node storagenode.
- Start the IBM Spectrum Scale™ daemon on all 3 nodes.

Press Next to perform the selected action.

At the bottom of the dialog, there are three buttons: 'Cancel', 'Back', and 'Next' (which is highlighted with a dashed border).

Figur 70 - GUI Fix Procedure

Andre problemer

I noen tilfeller vil ikke problemet kunne løses automatisk, og særlig dersom problemet ligger utenfor rekkevidden til programmet. Da må vi bruke informasjonen vi får fra hendelsene i kombinasjon med dokumentasjonen til Spectrum Scale for å kartlegge hva problemet er, slik at vi fortrest mulig kan fikse det.

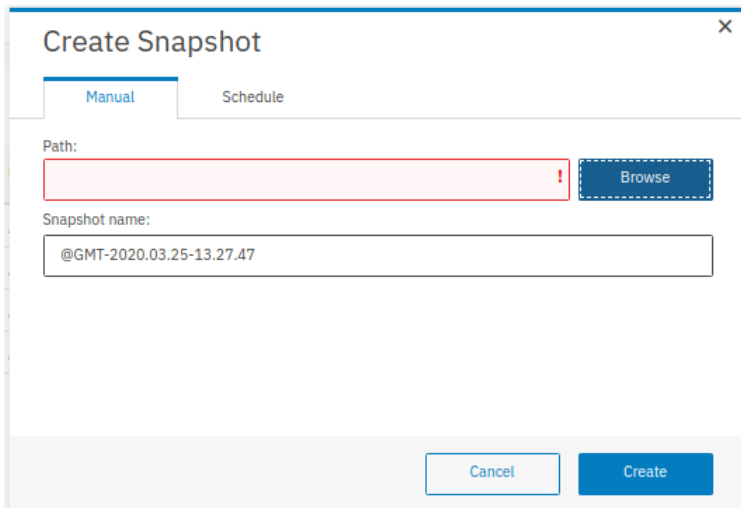
5.2.4 Snapshot

Som nevnt i kapittel 3.5.4 om backup finnes det mange alternativer for dette avhengig av behov og preferanser. Her ser vi på enkel backup av hele eller deler av filsystemet ved bruk av snapshot.

Snapshot

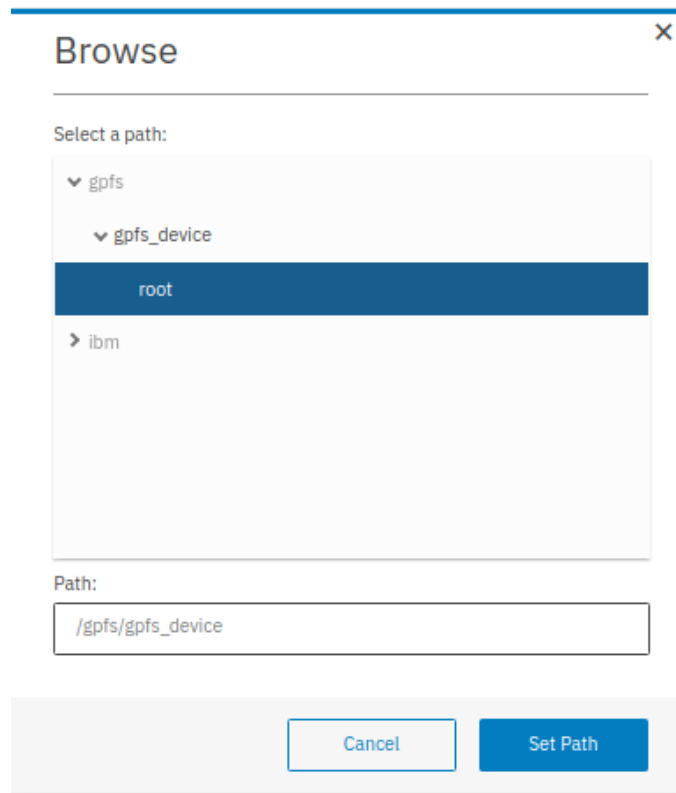
Å ta et snapshot av filsystemet kan enkelt gjøres fra GUI-tjenesten til Spectrum Scale, eventuelt med «mmcrsnapshot»-kommandoen. Et slikt snapshot vil være nyttig som en enkel backupløsning, særlig før man har tenkt å gjøre endringer på filsystemet.

For å ta et snapshot går man til **Files – Snapshots – Create Snapshot**. Her kan man velge stien til snapshottet man ønsker å ta, og eventuelt sette opp regler for automatisk snapshot.

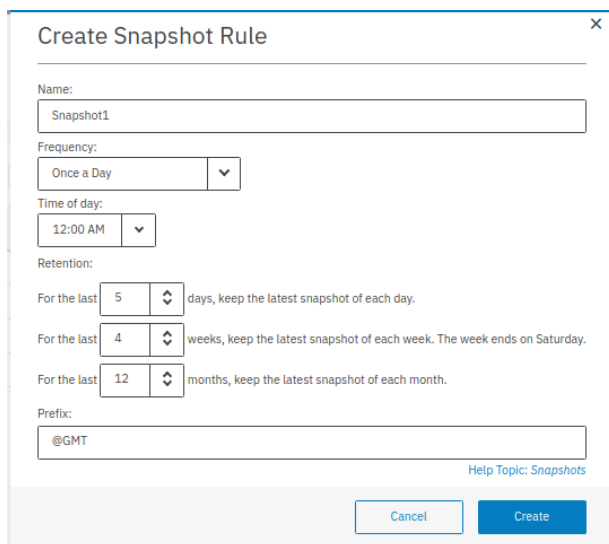


Figur 71 - GUI Create Snapshot

Dersom vi velger «browse» kan vi velge hvilke deler av filsystemet som skal tas snapshot av:



Figur 72 - GUI Snapshot Path



Figur 73 - GUI Snapshot Rule

Her kan vi se alle snapshots som har blitt lagret på filsystemet.

```
root@managernode:/gpfs/gpfs_device/.snapshots# ls
@GMT-2020.03.06-10.07.48 @GMT-2020.03.25-13.27.47 @GMT-2020.03.25-13.33.23
```

Gjenoppretting av snapshots:

Dersom vi ønsker å bruke snapshottet vi har tatt til å gå tilbake til en tidligere tilstand av filsystemet, kan vi bruke kommandoen «`mmstorefs filesystem snapshot`». Dermed gjenoprettes filsystemet til tilstanden den var før snapshottet ble tatt, men sletter ikke eventuelle nye innslag.

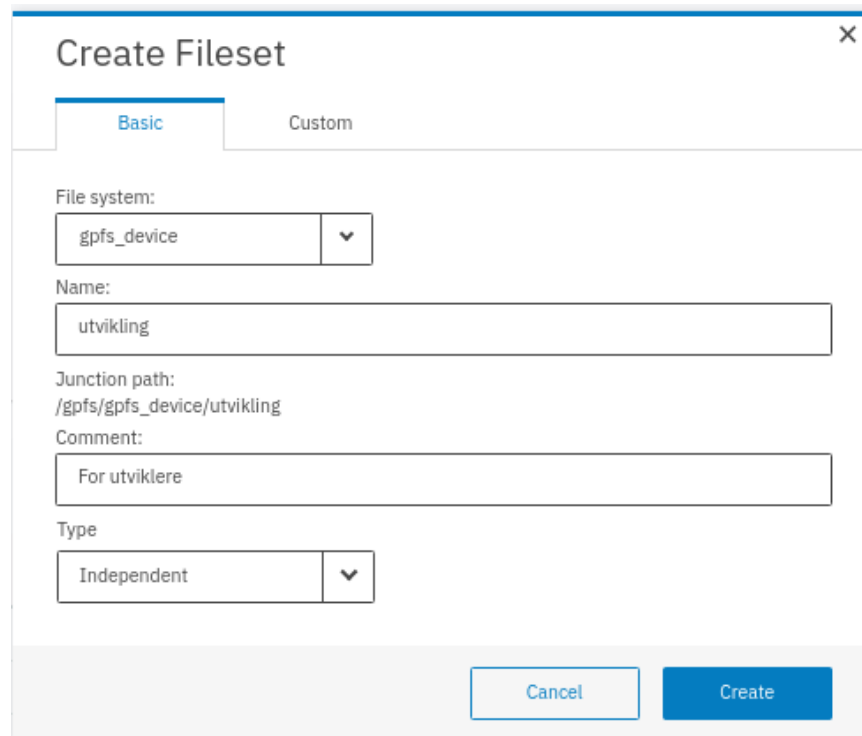
5.3 Sluttbrukere

Et av nøkkelpunktene til dette systemet er at det skal være enkelt å forholde seg til, og at det skal fremstå som ett filsystem for brukeren. Dette gjelder også for sluttbrukerne, altså utviklerne i dette tilfellet, som skal lese data, prosessere den, og skrive tilbake til filsystemet. Et forslag til hvordan filadministrasjon for utviklere kan fungere er å benytte egne filsett som inneholder kun de filene som disse brukerne skal ha tilgang til. Slik blir det mest mulig oversiktlig for sluttbrukerne, og minst mulig sannsynlig at det blir kluss i filsystemet.

5.3.1 Applikasjonsnode

I clusteret har vi satt opp en dedikert applikasjonsnode med hensikten å brukes til prosessering av data i clusteret. Grunnen til at vi har en egen node for dette er å oppnå raskest mulig responstid fra lagringssystemet, da noden er direkte koblet til disken i clusteret. Dette vil også være en maskin som har god prosessorkapasitet, og er tiltenkt tyngre prosessering av store mengder data. Noden har ingen andre roller i clusteret enn quorum-rollen, og er kun tiltenkt prosessering av data. Dette kunne også vært en selvstendig maskin satt opp som klient i clusteret.

Som eksempel setter vi opp et filsett som vist i 5.1.7 for utviklere som skal benytte seg av applikasjonsnoden:



The image shows a 'Create Fileset' dialog box with the following fields and values:

- File system: gpfs_device
- Name: utvikling
- Junction path: /gpfs/gpfs_device/utvikling
- Comment: For utviklere
- Type: Independent

Buttons: Cancel, Create

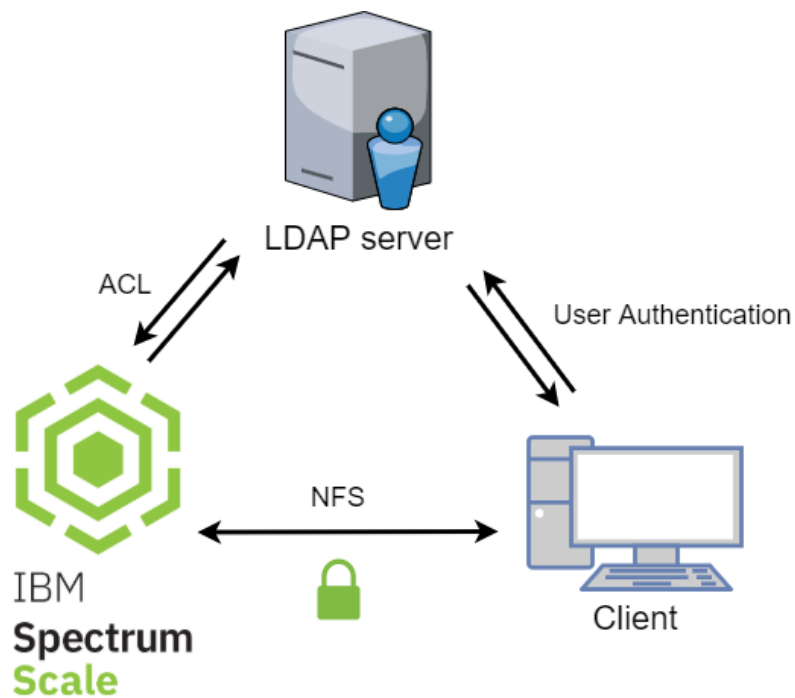
Figur 74 - GUI Create Fileset 2

Dermed kan man gi brukere som skal ha tilgang til filene under dette filsettet rettigheter til mappen «/gpfs/gpfs_device/utvikling», slik at de fritt kan jobbe med filene som ligger under her.

5.3.2 Delt filsystem over NFS

Et annet alternativ for sluttbrukere er å koble seg på som klient til NFS-filsharet eller en annen delt protokoll. Her vil klientene få tilgang til «ces_shared»-filsystemet, der filer som er tiltenkt disse brukerne vil plasseres. Dette er praktisk fordi det ikke krever at klientmaskinene kjører GPFS. Så lenge de har NFS installert kan de koble seg på og jobbe uten å være en del av selve clusteret. Dette vil være mest praktisk for oppgaver som ikke krever intens I/O. Her viser vi oppsettet av NFS i Spectrum Scale mot en klientmaskin.

Før vi begynner oppsettet av NFS trenger vi en del forberedelser. I Spectrum Scale trenger vi en protokollnode som har klargjort CES og NFS, som vi har beskrevet i 4.2.5 klargjøring av CES. Vi må sette opp brukertilgang over filer for eksterne brukere, som vi har beskrevet i 5.1.4. Vi trenger tilgang til en domenebruker på en klient-pc, som beskrevet i 4.2.6 OpenLDAP med phpLDAPadmin. Det siste er å ha et filsett som er tilgjengelig for deling som beskrevet i 5.1.7.



Figur 75 - NFS med autentisering mot LDAP

Konfigurasjonen går over 3 korte trinn. Det første er å gi brukeren tilgang til filen ved å sette opp en ACL til filene. ACL kan settes opp fra Spectrum Scale GUI og verifiserer brukerne i LDAP katalogen. Tilgang til filene må settes opp med en NFS-regel i clusteret før en til slutt klargjør tilgang til filsystemet fra en klient-pc. Klienten må benytte en domenebruker fra LDAP med tiltalelse med ACL i filsystemet.

ACL en setter du opp i brukergrensesnittet under «File system ACL». Du starter med å opprette en template som gjelder for brukeren og gruppen vi ønsker å gi rettigheter til. Deretter velger vi hvilken fil vi ønsker å tilegne templatens til.

Create ACL Template ×

Name:

Owner: Owning group:

Type	Name	Access	Read	Write	Execute	Access Mask	Inheritance
User	testdev	Allow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	rwxmdaA ...	fd-- ⊕ ⊖

 [Help Topic:Access Control List](#)

Figur 76 - GUI Create ACL Template

Det neste er å opprette en export av filsystemet. Her velger vi hvilken fillokasjon vi ønsker å dele, i dette tilfellet «**/ibm/ces_shared/utvikling_shared**». Deretter setter vi en «pseudo path» til «**/ces**», som er filplasseringen klientmaskinen ser via NFS.

Edit Export /ibm/ces_shared/utvikling_shared ×

Path:

Pseudo path:

Add NFS client. Reorder the clients as needed.

Name or IP Address	Access Type	Root Squashing	NFS Protocol
10.0.0.11	Read/Write	Root Squash	v3 and v4

Neste del av konfigurasjon gjøres fra klientmaskinen. Her starter du med å laste ned «nfs-common», som inneholder kommandoer for NFS.

```
apt-get install nfs-common
```

Du oppretter så en mappe som du vil koble NFS mot

```
sudo mkdir /home/users/testdev/gpfs
```

Deretter monterer du filsystemet i Spectrum Scale mot mappen du opprettet. Her bruker du en adresse til en av protokollknodene (10.0.0.5), etterfulgt av pseudo pathen «/ces» som du opprettet i Spectrum Scale. Det siste argumentet i kommandoen «mount» er mappen du ønsker å finne filsystemet i.

```
mount 10.0.0.5:/ces /home/users/testdev/gpfs
```

Med kommandoen «df» kan du se om filsystemet blir riktig montert. Her ser vi størrelsen og tilgjengelig plass på de ulike filsystemene:

```
df -h
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            1994712         0   1994712   0% /dev
tmpfs           401564         688   400876   1% /run
/dev/sda1       30309264 1415244 28877636   5% /
tmpfs           2007820         0   2007820   0% /dev/shm
tmpfs           5120           0     5120   0% /run/lock
tmpfs           2007820         0   2007820   0% /sys/fs/cgroup
/dev/sda15      106858         3668   103190   4% /boot/efi
/dev/sdb1       8190328         36856 7717712   1% /mnt
tmpfs           401564         0     401564   0% /run/user/1000
10.0.0.5:/ces  20971520 3997696 16973824  20% /home/users/testdev/gpfs
$ cat /etc/fstab
```

Figur 77 - GUI CES Export

6. Konklusjon og anbefalinger om videre arbeid

6.1 Konklusjon

Igjennom denne rapporten har vi sett på de grunnleggende funksjonene og prinsippene bak lagringsløsningen IBM Spectrum Scale. Vi har forsøkt å presentere dette på en helhetlig måte, slik at leseren sitter igjen med kunnskap som kan brukes i videre diskusjon. I den teoretiske biten av prosjektet har vi sett på hvilke komponenter som utformer et Spectrum Scale cluster, hvilke roller clusteret må bestå av, og hvordan komponentene samarbeider for å utføre ulike funksjoner. Oppsettet av et Spectrum Scale cluster i Azure har gitt oss mulighet til å teste og dokumentere funksjonene i praksis.

Ut ifra det teoretiske arbeidet ser vi at Spectrum Scale tilbyr et raskt og sikkert filsystem med god skalerbarhet, ytelse og høy oppetid. Løsningen er fleksibel og kan tilpasses bedriftens behov, da systemet kan konfigureres på mange måter, og har en rekke funksjoner som kan være nyttige i ulike scenarier. Filsystemet kan etableres både på kryss av operativsystemer, plattformer og geografiske distanser samtidig som det opprettholder god ytelse og integritet i dataene. Spectrum Scale kan enkelt skaleres til å håndtere gigantiske datamengder, og kan bruke en kombinasjon av ulike lagringsmedier for dette. Styrken ved å kunne benytte forskjellige typer lagringsmedium, er at systemet både kan arkivere og behandle data, uten at kostnadene blir for store.

Videre har vi installert et testsystem i liten skala, der vi har tatt i bruk grunnleggende funksjonalitet og demonstrert noen av mulighetene som IBM Spectrum Scale tilbyr. Med dette viser vi at systemet er lett å administrere, konfigurere og tilpasse etter behov. Vi har vist at data enkelt kan aksesseres fra brukere og konfigureres med andre systemer for sikkerhet og tilgangsstyring. Fra et drifts-perspektiv kan vi anbefale Spectrum Scale, da programmet er brukervennlig og oversiktlig, og er godt dokumentert.

Arbeidet videre mot et nytt lagringssystem, må for Kongsberg bli å sammenstille kostnader mot nytteverdien av et slikt system. Vårt inntrykk fra møtet med IBM, og etter arbeid med prosjektet, er at Spectrum Scale er ment for større og mer ressurskrevende datamengder enn hva Kongsberg Seatex har behov for akkurat nå. Det kan argumenteres for å anskaffe et slikt system likevel for å være tidlig forberedt på fremtidens krav til kapasitet. Eventuelt kan man vurdere å innføre Spectrum Scale som en løsning på kryss av avdelinger i bedriften, slik at man får bruk for noen av stordriftsfordelene ved et slikt skalerbart filsystem. Alternativet som har vært diskutert er å gå for en rimeligere og mindre kompleks løsning i førsteomgang, slik som en enkel tape-løsning. Å enten starte smått eller samarbeide med andre om en stor løsning vil kunne være lettere å forankre i bedriften. Dette vil muligens også gi mer mening økonomisk sett enn å gå løs på en fullverdig Spectrum Scale løsning uten videre.

Uansett er det viktig å være klar på hva bedriften ønsker av en ny lagringsløsning, og måle dette opp imot hva Spectrum Scale kan og ikke kan gjøre. Med utgangspunkt i bedriftens problemstillinger og behov er Spectrum Scale åpenbart godt egnet som et alternativ, og som et tema for videre diskusjon. Spørsmålet blir igjen om nytten veier opp for kostnadene knyttet til innkjøp av nytt lagringssystem, og omstillingen i bedriften dette fører med seg. Med stadig økende datamengder er det uansett viktig å ha en god plan for hvordan man skal håndtere dette på sikt, enten det blir gjennom en enkel tapeløsning for arkivering eller et fullt lagringssystem som Spectrum Scale. Vi håper at bachelorprosjektet har vært til hjelp for Kongsberg Seatex i vurderingen av IBM Spectrum Scale som løsning, og at rapportene har gitt innsikt i hvilke muligheter dette systemet kan tilby dem som bedrift

6.2 Vedlegg

Som vedlegg til denne rapporten finner du en sluttrapport som tar for seg evaluering av prosessen. Her vurderer vi hvordan teamet har fungert, hvordan oppgaven har vært å jobbe med, hvilke utfordringer vi har hatt og hvilke erfaringen vi har gjort oss i løpet av disse månedene. Videre finner du prosjekthåndboken som inneholder planer for prosjektet, samt logger og referater fra møtene som har tatt sted under prosjektet. Vedlagt ligger en også egen ordliste som forklarer begreper brukt i rapportene.

Kilder

Bibliografi

Tate, J. et al. (2017). *Introduction to Storage Area Networks. 9. Edition*. Tilgjengelig fra: <http://www.redbooks.ibm.com/abstracts/sg245470.html> (Hentet 14.02.2020)

NAS vs. SAN: Differences and Use Cases (2018) Tilgjengelig fra: <https://www.enterprisestorageforum.com/storage-networking/nas-vs.-san-differences-and-use-cases.html> (Hentet: 10. februar 2020).

Robb, D. (2018) *Network Attached Storage in the Enterprise*. Tilgjengelig fra: <https://www.enterprisestorageforum.com/storage-networking/network-attached-storage-in-the-enterprise.html> (Hentet: 10. februar 2020).

Indiana University (2019) *About NFS*. Tilgjengelig fra: <https://kb.iu.edu/d/adux> (Hentet: 11. februar 2020).

Taylor, C. *What Is iSCSI and How Does It Work?* Tilgjengelig fra: <https://www.enterprisestorageforum.com/storage-hardware/iscsi.html> (Hentet: 11. februar 2020).

Beal, V. (2020) *DAS – Direct Attached Storage*. Tilgjengelig fra: https://www.webopedia.com/TERM/D/direct_attached_storage.html (Hentet: 12. februar 2020).

LTO (2020) *LTFs* Tilgjengelig fra: <https://www.lto.org/technology/ltps/> (Hentet: 13. februar 2020).

Referanseliste

Deloitte (2015) *Information Lifecycle Management (ILM)* Tilgjengelig fra: <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/finance/us-advisory-information-lifecycle-management.pdf> (Hentet 26.02.2020).

IBM (2020) *IBM Spectrum Scale 5.0.1 Documentation*. Tilgjengelig fra: https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/b11ins_protocolsprerequisites.htm (Hentet 02.03.2020)

Quintero, D., et al. (2015). *IBM Spectrum Scale (formerly GPFS)*. Tilgjengelig fra: <http://www.redbooks.ibm.com/redbooks/pdfs/sg248254.pdf> (Hentet: 16.04.2020)

Rouse, M. (2005) *Information Lifecycle Management (ILM)* <https://searchstorage.techtarget.com/definition/information-life-cycle-management> (Hentet 26.02.2020).

Taylor, C. (2019) *RAID Controllers*. Tilgjengelig fra: <https://www.enterprisestorageforum.com/storage-hardware/raid-controllers.html> (Hentet: 10. februar 2020).

Miyoshi, H., et al. (2020) *IBM Spectrum Archive Enterprise Edition* Tilgjengelig fra: <https://www.redbooks.ibm.com/redbooks/pdfs/sg248333.pdf> (Hentet 13.05.2020)



Software Defined Storage med IBM Spectrum Scale

Med fokus på effektivisering og ressursbesparelse

Sluttrapport

For bachelorprosjektet v2020 i Drift av Datasystemer

Amund Hauglie-Hanssen, Sveinung Bergum
15.05.2020

Innhold

Figurliste.....	1
Tabelliste.....	2
Revisjonshistorie.....	2
1. Introduksjon.....	3
2. Resultater.....	4
2.1 Forstudierapport.....	4
2.2 Hovedrapport.....	4
2.3 Testsystemet.....	4
2.4 Måloppnåelse.....	6
2.4.1 Tidsplan.....	6
2.4.2 Prosjektets mål.....	6
2.4.3 Timebruk.....	7
3. Prosessen.....	9
3.1 Samling av materiale.....	9
3.2 Dokumentasjon.....	9
3.3 Installasjon av testsystem.....	9
3.4 Samarbeidet i gruppen.....	9
3.4.1 Arbeidsfordeling.....	9
3.5 utfordringer underveis.....	10
3.5.1 Utforme en konkret hensikt.....	10
3.5.2 Avgrensning av prosjektet.....	10
3.5.3 Balanse mellom det tekniske og konseptuelle.....	11
3.5.4 Coronaviruset.....	11
4. Oppsummering og videre arbeid.....	12

Figurliste

Figur 1 - Clusteroppsett.....	5
Figur 2 - Tidsplan.....	6

Tabelliste

Tabell 1 - Revisjonshistorie	2
Tabell 2 - Timebruk	7

Revisjonshistorie

Tabell 1 - Revisjonshistorie

Dato	Utgave	Kommentar
13.05.2020	1.0	Førsteutkast
15.05.2020	2.0	Klar til levering

1. Introduksjon

I løpet av det siste halvåret har prosjektgruppe 115, bestående av Amund Hauglie-Hanssen og Sveinung Bergum, i samarbeid med Kongsberg Seatex utarbeidet et bachelorprosjekt. Prosjektet omfatter temaet «Software Defined Storage», og spesifikt lagringssystemet IBM Spectrum Scale. Oppgaven gikk ut på å sette problemstillingen til bedriften om lagring av store mengder data i sammenheng med et lagringssystem som Spectrum Scale. Et slikt system setter fokus på «Information Lifecycle Management», altså hvordan data skal håndteres gjennom livssyklusen. Dette er med hensikt å få kontroll over hvilke data bedriften har til enhver tid, hvordan den er lagret, og hva det koster. I tillegg setter Big Data naturligvis større krav til ytelse og parallelle arbeidsoppgaver, noe som må dekkes av et slikt system.

I løpet av prosessen har vi lært om ulike problemstillinger knyttet til håndtering av Big Data. Vi ser det klart at dagens IT-bedrifter ganske raskt blir nødt til å ta stilling til den gigantiske datamengden som samles inn av alle mulige sensorer og kontaktpunkter i IT-løsningene, og hvordan disse tallene bare kommer til å stige fremover. For å håndtere dette trenger man smarte og effektive systemer for å skille ut viktig data, og etablere rutiner for å håndtere dataen gjennom hele livssyklusen på billigst mulig vis. Dataen som samles opp er en stor og viktig ressurs for mange bedrifter, og kan i mange tilfeller være kritisk for videre drift. Derfor blir datasikkerhet og redundante lagringssystemer viktige temaer.

I denne rapporten skal vi kort oppsummere hva vi har gått igjennom i løpet av bachelorprosjektet, og hvordan prosessen har vært. Vi skal se på hvilke områder vi har vært borti, hva vi har lært, utfordringer vi har støtt på, og hva sluttproduktet ble.

2. Resultater

Dokumentasjonen av selve bachelorprosjektet, forstudierapporten og hovedrapporten, tar for seg IBM Spectrum Scale både på et konseptuelt og teknisk nivå. Resultatet av bachelorprosjektet er denne dokumentasjonen som fungerer som en innføring i oppbygningen av Spectrum Scale og programmets funksjoner, samt en demonstrasjon av noen av mulighetene ved denne programvaren. Vi har satt det som vi mener er nyttig for bedriftens problemstillinger i fokus, og dette er noe som bedriften forhåpentligvis får bruk for videre.

2.1 Forstudierapport

Hensikten med forstudierapporten var først og fremst å gjøre en vurdering av levedyktigheten til prosjektet, og å undersøke produktet på overflaten. Videre gjorde vi en rekke analyser, slik som interessentanalyse og kost/nytte analyse, før vi kom med en anbefaling om videre arbeid.

Resultatet av forstudiet var at vi anså Spectrum Scale som et spennende tema å undersøke videre, og at dette kunne være hensiktsmessig for bedriftens del. Rapporten ble vi stort sett fornøyd med, og føler at vi fikk frem poengene på en konkret og oversiktlig måte. Resultatet av analysene, eksempelvis kost/nytte-analysen, ble noe begrenset. Dette var både fordi vi i stor grad manglet data, men også fordi slike analyser er ressurskrevende i utgangspunktet, og kan gjøres svært komplekse om ønskelig. Likevel mener vi at vi fikk illustrert poengene våre gjennom analysene, og at de derfor var vellykket.

2.2 Hovedrapport

Resultatet av hovedrapporten er en omfattende dokumentasjon, som tar for seg både tekniske og abstrakte forklaringer av Spectrum Scale som produkt, og hvordan det kan tas i bruk for å dekke behovene til Kongsberg Seatex. Vi har jobbet hardt for å gjøre rapporten mest mulig leselig, og konseptene forståelige gjennom kontinuerlig diskusjon i gruppen, og omstrukturering av innholdet. Til tross for dette er innholdet ganske teknisk, og krever at leseren har grunnleggende forståelse for mange av konseptene for å henge med. Slik måtte det bli for å unngå unødvendig mye forklaring.

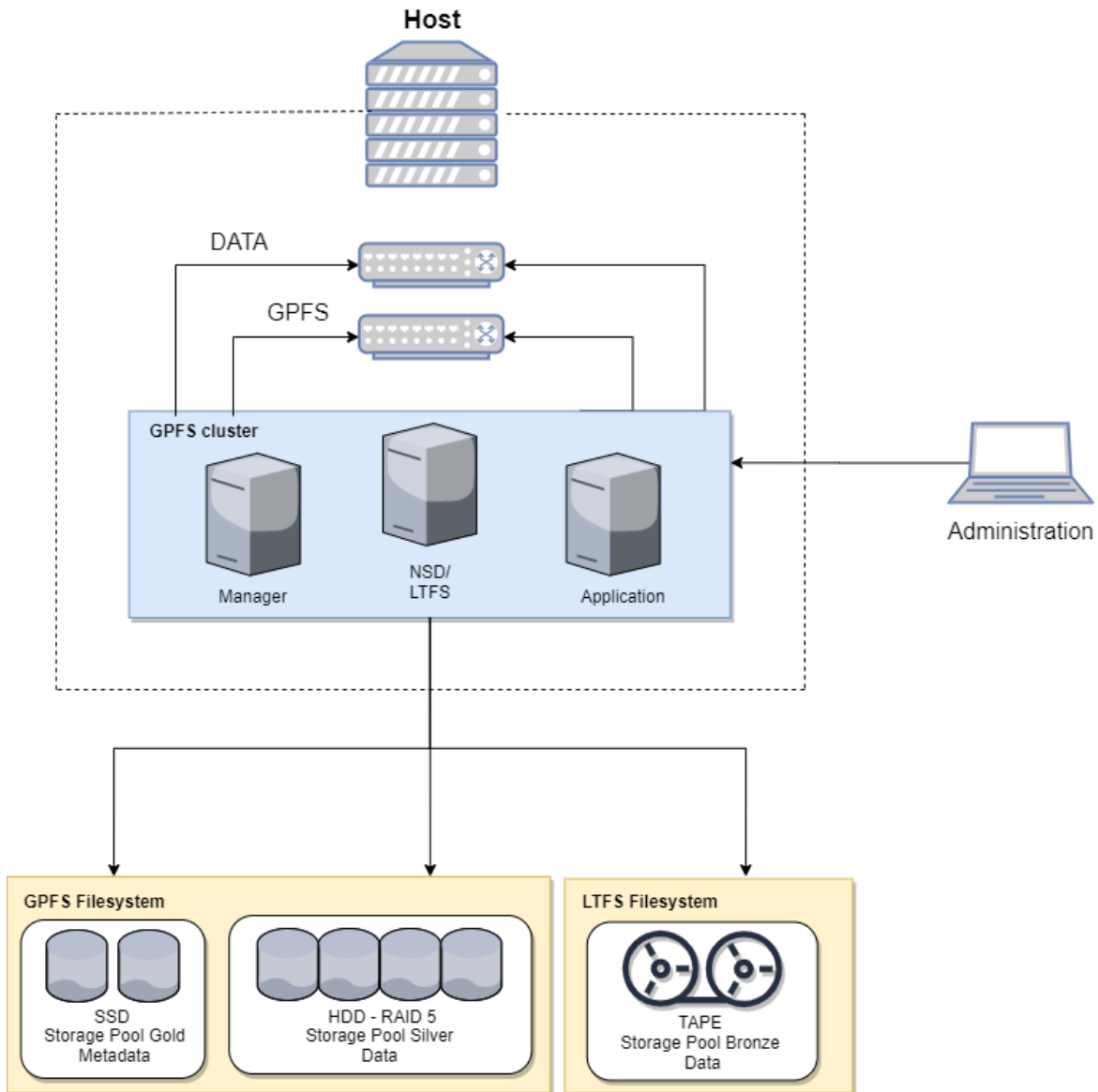
Noe av det vi har jobbet mest med i prosessen er å vinkle prosjektet på en måte som er til hjelp for Kongsberg Seatex i vurderingen av et slikt system. Med utgangspunkt i dette satte vi sammen en rekke «use cases» basert på antagelser av de viktigste bruksområdene. Eksempler på dette er installasjon av komponenter i clusteret, konfigurering av disk og filsystem, og overvåking av systemet under daglig drift. Dette har vi dokumentert i hovedrapporten, både gjennom praktiske eksempler og teori.

2.3 Testsystemet

Den sentrale oppgaven i dette prosjektet har vært å fordype seg i, installere og teste IBM Spectrum Scale på våre egne maskiner for å få best mulig forståelse for systemet. I begynnelsen var det stor usikkerhet rundt kompleksiteten i gjennomføringen, og vi visste ikke hvor mye arbeid det kom til å kreve å få installert programvaren. Vi visste ikke engang om vi fikk tilgang til lisens, eller om vi hadde kompatibel maskinvare. Heldigvis viste det seg at dette ikke skulle by på større utfordringer.

På IBM sine nettsider fant vi dokumentasjonen vi trengte, i tillegg til en gratis versjon av IBM Spectrum Scale som vi kunne bruke på våre virtuelle maskiner i Azure. Slik du kan lese om i hovedrapporten, og som du ser på figur 1, bestod vårt testcluster av tre noder koblet til tre ulike storage pools. Spectrum

Scale består av en programvarepakke som man kan laste ned fra IBM sine nettsider. Vi opprettet tre VM-er i Azure, og installerte programmet på disse maskinene ved hjelp av installasjonsverktøyet som følger med. Videre, med hjelp fra installasjonsvideoen til IBM, var det relativt enkelt å få clusteret til å fungere. Deretter kunne vi utforske mulighetene og grensesnittet til programmet, og dokumentere dette i rapporten.

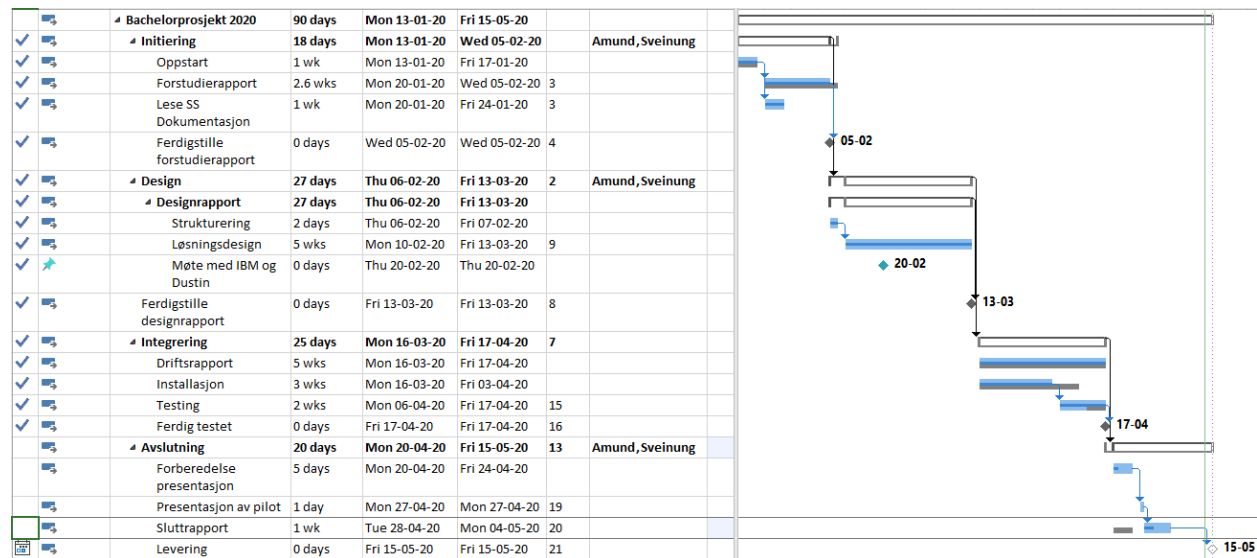


Figur 1 - Clusteroppsett

2.4 Måloppnåelse

2.4.1 Tidsplan

Figuren viser den grove planleggingen av prosjektet gjort i MS Project tidlig i prosjektet. Det har blitt gjort noen endringer og oppdateringer av planen underveis, men stort sett har vi fulgt planen hele veien igjennom. Datoene har blitt overholdt, og vi har brukt planen som retningslinje hele veien igjennom de ulike fasene i prosjektet.



Figur 2 - Tidsplan

2.4.2 Prosjektets mål

Da vi utarbeidet forstudierapporten, konkretiserte vi en rekke mål for bachelorprosjektet. Disse målene skulle brukes som retningslinjer for resten av prosjektet, og dokumentasjonen dreide seg rundt disse. Målene ble delt opp i prosessmål, effektmål og resultatmål. Prosessmålene omhandlet resultater som skulle komme frem av selve prosessen, effektmålene var målene for hvilke effekter systemet skulle få for bedriften, og resultatmålene konkret beskrev hva som skulle gjøres i prosjektet.

Dersom vi måler grad av måloppnåelse basert på hvordan prosjektet gikk i forhold til disse målene, kan vi trygt si at prosjektet ble vellykket. Vi har lært om Spectrum Scale, dokumentert dette i en omfattende rapport, og satt opp et fungerende testsystem med de mest sentrale funksjonene installert. Vi har demonstrert at data kan plasseres inn i filsystemet, og automatisk sorteres til ulike lagringsmedier basert på regler vi har satt opp. Filene er også tilgjengelig fra alle nodene i clusteret, og kan eksporteres via NFS til eksterne noder. I tillegg har vi sett på bedriftens problemstillinger, og argumentert for hvordan systemet kan passe inn med deres behov. Slik har vi dekket målene om å sette seg inn i, dokumentere, installere og konfigurere Spectrum Scale, og med det mener vi at vi har fått til det vi ønsket da vi satt i gang.

2.4.3 Timebruk

Tabell 2 viser det faktiske timebruket på de ulike aktivitetene. Dette gir oss grunnlag til å vurdere den faktiske timebruken mot tidsplanen (kapittel 2.4.1) og lar oss reflektere rundt det helhetlige omfanget til prosjektet. Noe har forandret seg underveis, slik som sammenslåingen av design og driftsdokumentet.

Tabell 2 - Timebruk

Aktiviteter	Timebruk
Strukturering og dokumentasjon	161
Forstudierapport	106
Informasjonshenting	127
Installasjon / Testing	132
Designrapport	97
Møter og planlegging	37
Driftsrapport	34
Hovedrapport	101
Presentasjon	60
Totalsum	855

«Strukturering og dokumentasjon» omfatter generell planlegging, diskusjon og rapportskrivning rundt prosjektet. Eksempelvis er dette oppsett av tidsplaner og prosjekthåndboken, og omstrukturering av rapporter. Vi har satt av timer på posten under alle faser av prosjektet, noe som har gjort posten til en av de mest tidskrevende. Posten har ikke innvirket direkte på produktet i bacheloroppgaven, men har vært viktig for det helhetlige omfanget.

Forstudierapporten brukte vi kortere tid på enn antatt. Estimert hadde vi rundt 200 timer på rapporten, men har kun brukt 106. Mye av grunnen til dette er at vi også brukte fasen på å lese oss opp å gjøre en testinstallasjon av systemet. Her er det dermed gått en del tid til postene «informasjonsinnhenting» og «installasjon / testing». Selve fasen ble vi ferdig med før milepælen som lot oss komme tidligere i gang med designrapporten.

«Informasjonsinnhenting» omfatter all tidsbruk vi har brukt for å sette oss inn i ulike programdokumentasjon og kilder. Posten har vært sentral siden vi ved starten av prosjektet ikke hadde noen kunnskap om systemet vi satte opp. Vi ser derfor at det har gått med en god del tid til dette.

«Installasjon / Testing» omfatter alle timer brukt på installasjon, konfigurering og testing av IBM Spectrum Scale. Denne posten er slått sammen av flere poster fra tidsplanen. På grunn av endringene underveis i prosjektet fant vi det mest naturlig å ha denne som en post. Siden vi fokuserte på å dokumentere de ulike funksjonene og konseptet til Spectrum Scale, falt testing, installasjon og konfigurering sammen. Noe som er viktig å merke seg er at de første 61 timene av installasjon og testing kun ble brukt til å lære oss systemet. Tidsbruken på det endelige systemet er dermed bare på 71

timer. Dette viser at Spectrum Scale er et enkelt program å konfigurere og installere og har gode tilpasningsmuligheter til ulike miljøer.

Designrapport og driftsrapporten slo vi sammen av praktiske årsaker til én hovedrapport. Dermed må vi se på det totale timeforbruket på rapportene i ett. Det totale timeforbruket her er 241 timer.

Endringene av rapporten var nødvendig for å gi en helhetlig forståelse av IBM Spectrum Scale i samme rapport. Det høye tidsforbruket er dermed forsvarlig fordi det representerer innsatsen for både design- og driftsfasen. Samtidig måtte vi bruke mye tid på å gjøre teksten forståelig for leseren.

«Møter og planlegging» er en post som vi skilte ut av «strukturering og dokumentasjon». Vi innså etter hvert at innholdet i disse to postene ble forskjellig. «Møter og planlegging» omfatter tid brukt på innkalling og gjennomføring av møter, referatskriving og planlegging for videre arbeid.

«Presentasjon» er posten som dekker arbeidet i uke 21 med forberedelser til presentasjon. Vi anslår timeforbruket på denne posten i forkant, da vi leverer oppgaven før presentasjonen er helt ferdig.

Det totale timeforbruket ble dermed på 855 timer. Målet for prosjektet lå på 1000 timer. Vi ser ikke på dette som et problem, da vi har jobbet jevnt og kontinuerlig med prosjektet fra start, og har hatt god fremdrift hele veien. Noe tid gikk bort i forbindelse med Covid-19. Heldigvis lå vi godt an med oppgaven, og kunne ta oss råd til å prioritere eksamen i uke 18 og 19.

3. Prosessen

3.1 Samling av materiale

For å i det hele tatt kunne begynne å produsere meningsfylt arbeid, måtte prosjektgruppen fordype seg i temaene gjennom den dokumentasjonen som var tilgjengelig. Heldigvis har IBM dokumentert systemene sine godt både på nettet og i egne bøker. Ut ifra dette lagde vi et kondensert utdrag av dokumentasjonen i en egen fil for å få oversikt over hvordan temaene hang sammen. Vi har dermed i stor grad brukt både internett og publiserte bøker som grunnlag for arbeidet i bachelorprosjektet. Da vi hadde brukt de første ukene i prosjektet på å lese oss opp på stoffet, var vi mye bedre forberedt på å takle de praktiske oppgavene med å sette opp et testsystem.

3.2 Dokumentasjon

Helt fra begynnelsen i prosjektet har gruppen jobbet iherdig med å sette sammen dokumentasjon som både dekker prosjektmålene, og formidler den viktigste kunnskapen om systemet til leseren. Resultatet av dette ble som sagt to rapporter, en forstudierapport og en hovedrapport. Hovedrapporten ble dokumentet som inneholdt mesteparten av den tilegnede kunnskapen, da forstudiet handlet mest om å få et overblikk.

3.3 Installasjon av testsystem

Med god kunnskap om systemets funksjoner og sammenheng ble det ingen stor utfordring å sette opp et fungerende testsystem. Ved hjelp av en instruksjonsvideo og dokumentasjonen på nettet fant vi ut hvordan vi skulle laste ned og installere programmet. Spectrum Scale er kompatibelt med Ubuntu, og vi fant alt vi trengte av forhåndsinstallasjon i dokumentasjonen. Dermed gikk dette relativt raskt unna.

3.4 Samarbeidet i gruppen

Samarbeidet i prosjektgruppen har gjennomgående vært bra. Begge deltakerne har møtt opp til avtalt tid, og gjort sitt beste for å bidra positivt til gjennomføringen av prosjektet. Helt fra begynnelsen har vi vært enige om å møte opp på skolen, og jobbe som om det var en normal arbeidssituasjon. Selv om timeantallet generelt ble noe lavere enn antatt, har vi stort sett jobbet fulle arbeidsdager, og med god flyt gjennom hele prosjektet. Arbeidsrutinen og interessen for emnet gjorde at vi kunne jobbe med høy effektivitet. Vi er begge fornøyde med arbeidsinnsatsen og kvaliteten på sluttproduktet.

3.4.1 Arbeidsfordeling

Selv om det ikke har eksistert noen klar rollefordeling i gruppen, og medlemmene har tatt for seg de arbeidsoppgavene som krevdes av dem underveis, har fordelingen av oppgavene mellom deltakerne vært noe forskjellig.

Amund Hauglie-Hanssen

Amund har jobbet mest med Spectrum Scale-produktets oppbygning og tekniske detaljer. Han har satt seg grundig inn i dokumentasjonen til IBM om hvordan de ulike komponentene fungerer og utgjør programmets funksjoner. Videre har han overført denne kunnskapen både i installasjonen av programvaren på en virtuell testløsning og den tekniske dokumentasjonen av programmet i rapportene. Han har dokumentert mange av de praktiske oppgavene som kan gjennomføres med Spectrum Scale gjennom det grafiske brukergrensesnittet, og satt dette i sammenheng med bedriftens behov.

Sveinung Bergum

Sveinung har i hovedsak tatt for seg et mer helhetlig bilde av et lagringssystem med IBM Spectrum Scale, og har hatt større fokus på støttesystemene rundt en installasjon enn de tekniske detaljene. Han har sett på temaer som dreier seg rundt tapeløsninger med IBM Spectrum Archive og tilleggsfunksjoner som CES (Cluster Export Services) med NFS og brukerdatabase med LDAP. I rapportene har han bidratt med å installere og dokumentere disse funksjonene og sette dem i sammenheng med Spectrum Scale.

Felles

Sammen har Amund og Sveinung tatt for seg Kongsbergs problemstilling, og diskutert hvilken løsning som kan være den beste for bedriften. Gruppens medlemmer har sammen stått for å sette sammen dokumentasjon med et helhetlig fokus, og har brukt mye tid på å skape sammenheng i teksten. Dette har ført til mange timer med omstrukturering og vurdering av den røde tråden i dokumentasjonen.

3.5 Utfordringer underveis

Prosjektgruppen har naturligvis støtt på noen utfordringer underveis i arbeidet. Vi har erfart at det er krevende å jobbe i store prosjekter, og å skrive store rapporter. Dette krever god planlegging og struktur for å produsere et tilfredsstillende resultat, og det er viktig at man kommer fort i gang med prosessen slik at man best mulig kan kartlegge prosjektets utfordringer. Her ser vi litt på hvilke problemer vi har støtt på i prosjektet, og hvordan vi gikk frem for å løse disse.

3.5.1 Utforme en konkret hensikt

Oppdragsgiveren, Kongsberg Seatex, hadde et konkret problem med sine datamengder. Kravene til selve systemet var klare. Det som derimot ikke var klart, var hva som skulle til for å dekke disse behovene. I begynnelsen var hensikten med prosjektet å sette opp et pilotprosjekt for bedriften som skulle fungere som en slags testløsning eller en «salgspitch» for et virkelig Spectrum Scale-system. Prototypen skulle speile bedriftens virkelige implementasjon i størst mulig grad, og i prinsippet kunne overføres til deres datasystemer.

Senere i prosjektet innså vi etter hvert at dette ble for omfattende for to studenter uten forhåndskunnskaper om systemet. Et system som skal håndtere bedriftskritisk informasjon må igjennom en formell prosess utenfor vårt kunnskaps- eller ressursområde. I tillegg fant vi ut i møte med IBM at de sannsynligvis leverer en egen skreddersydd løsning for bedriften, og at vårt testsystem dermed i stor grad ble overflødig. Vi valgte derfor å kurse prosjektet i en annen retning, der hensikten i større grad dreide seg rundt å eksperimentere med Spectrum Scale, og demonstrere noe av nytteverdien systemet kunne ha for Kongsberg Seatex i fremtiden. I prosjektet valgte vi derfor å fokusere på å illustrere konseptene og grunnleggende funksjonalitet i løsningen på et mer abstrakt nivå, for best mulig å gjøre en nytteverdi for bedriften. Denne forandringen i prosjektmålene gjorde at vi måtte revurdere mange av punktene vi hadde planlagt, og skrive om en del i rapportene.

3.5.2 Avgrensning av prosjektet

Det er vanskelig å si noe om det man ikke vet noe om. Da vi satte i gang med prosjektet, hadde vi kun en svært abstrakt forståelse av hva systemet gikk ut på. Vi hadde ingen anelse om omfanget til de ulike områdene av den tekniske løsningen, og slik det ble nevnt tidligere visste vi ikke engang om vi kom til å greie å laste ned og installere Spectrum Scale i det hele tatt. Den eneste måten å finne ut dette på, var å sette seg ned og lese om temaet. Dermed ble det også vanskelig å gjøre en realistisk avgrensning av

prosjektet, noe som førte til at vi i flere omganger måtte gå tilbake til prosjektmålene og endre på dem i forhold til kunnskapen vi tilegnet oss underveis.

Det var mange retninger vi kunne tatt prosjektet i, og det gjaldt dermed å finne middeelveien mellom de ulike temaene rundt problemstillingen. For eksempel måtte vi etter hvert beslutte at fokuset måtte holde seg til kun selve datasystemet, og ikke for eksempel transport av data fra båten. Det ble også viktig at vi ikke brukte for mye tid på å utdype hvert tema og funksjon, da man kunne ha skrevet flere bacheloroppgaver om noen av disse temaene. Heller fokuserte vi på å vise litt av hvert, slik at leseren får en oversikt over hva som er mulig å gjøre.

3.5.3 Balanse mellom det tekniske og konseptuelle

Et av de temaene vi har brukt mest tid på å diskutere er utformingen av prosjektet, og da også dokumentasjonen som helhet i forhold til grad av teknisk fokus. I begynnelsen av prosjektet lå fokuset veldig på det tekniske bak Spectrum Scale som produkt. Vi så på hvordan de ulike komponentene i løsningen var bygget opp og hvilke oppgaver de hadde. Selv om dette var veldig nyttig i å danne forståelse for programmet, noe som kom til hjelp i å videreformidle kunnskapen, ble fokuset fort for snevert og arbeidsomfanget ble stort og uoversiktlig.

Problemet med å bli for orientert rundt detaljene er at arbeidsmengden blir uendelig stor. Vi hadde rett og slett ikke tid til å sette oss inn i og dokumentere alle funksjonene i Spectrum Scale. Vi tror heller ikke at dette hadde vært spesielt nyttig for bedriften. Dette henger også tett sammen med selve oppgavedefinisjonen, og hva som er hensikten med prosjektet.

Slik endte vi med å måtte justere fokuset fra detaljnivå til abstrakt nivå. Dette ble gjort ved å inkludere flere illustrasjoner av konseptene, forklare dem så enkelt som mulig, og fjerne unødvendige tekniske detaljer. Det var her viktig å hele tiden holde prosjektmålene og selve hensikten med rapporten i bakhodet for å ikke skli ut i en sidegate som ikke var relevant for prosjektet.

Å skrive rapporten på denne måten var også essensielt for å gjøre den mest mulig forståelig for leseren. Vi har erfart at det er utfordrende å sette seg inn i hvordan det er å lese teknisk dokumentasjon når man ikke har den samme kunnskapen som den som har skrevet den. Igjen har vi derfor brukt mye tid på strukturering, og å forsøke å holde en rød tråd gjennom rapportene på best mulig vis, uten å gjenta oss selv for mange ganger.

3.5.4 Coronaviruset

Effektiviteten i prosjektet ble mot slutten noe rammet av Covid-19-situasjonen. Da vi ikke fikk sitte på skolen lenger, merket vi fort at det ble vanskeligere å stå opp til riktig tid og jobbe effektivt. For å håndtere dette justerte vi ned målet fra 8 timer til 6 timer om dagen, og inkluderte heller tirsdager som arbeidsdag i tillegg. Dette har hjulpet litt på effektiviteten, da vi kunne jobbe mer intensivt på denne måten. Likevel har dette problemet sakkert ned på prosessen, og tærte på motivasjonen i gruppen.

4. Oppsummering og videre arbeid

Vi er stort sett veldig fornøyde med gjennomføringen av bachelorprosjektet. Vi mener at vi har oppnådd de målene vi satte i begynnelsen i høy grad, at vi har lært masse, og produsert et tilfredsstillende resultat. Oppsummert er resultatet av dette bachelorprosjektet en grunnleggende innføring i IBM Spectrum Scale som produkt. Dokumentasjonen skal hjelpe Kongsberg Seatex i vurderingen videre av hvordan de skal håndtere sine data. For bedriften blir veien videre å sette seg inn i dokumentasjonen vi har utarbeidet for å avgjøre om en reell innføring av Spectrum Scale eller et liknende produkt kan være noe for dem. Vi håper at prosjektet har utgjort en forskjell og at det kommer til hjelp i tiden fremover.

