NTNU
Norwegian University of
Science and Technology

Evaluating performance impact of performing computations on storage nodes

Batch on Eos Extra Resources (BEER)

Håvard Tollefsen

Master of Science in Informatics
Submission date: July 2018
Supervisor: Magnus Själander, IDI

Norwegian University of Science and Technology
Department of Computer Science

Summary

The Batch on EOS Extra Resources (BEER) project is a response to observations of available computing resources on EOS storage system. BEER introduces sharing computing resources between storage and compute nodes. The project have gone through several iterations to insure that the service provided by the EOS storage system would not be compromised. The success of the project have provided the Batch system with an estimate of over 2'000 extra cores, and with further scaling have a huge potential to give a lot more.

Preface

This Master's thesis is written by Håvard Tollefsen for the Norwegian University of Science and Technology (NTNU). The thesis is written as a report of the project the author worked on during his Technical Student programme at The European Organization for Nuclear Research (CERN).

I want to thank my supervisor Ben Jones for giving me the opportunity to come work on the project, the Infrastructure Services and the Storage teams for their work on the project, and my thesis supervisor Magnus Själander for accepting to supervise my thesis without much control on the project.

Table of Contents

List of Figures

Abbreviations

ALICE	=	A Large Ion Collider Experiment, one of the detector experiments at the LHC
Atlas	=	A Toroidal LHC ApparatuS, one of the detector experiments at the LHC
BEER	=	Batch on EOS Extra Resources
CERN	=	European Organization for Nuclear Research
CPU	=	Central Processing Unit
LHC	=	Large Hydron Collider
I/O	=	input/output, the action of reading or writing data to a disk

Chapter 1

Background and current state

For every iteration of the LHC experiments, detectors are upgraded to extract even more data from events. With no plans to build any new data centers we are required to maximize the use of any computer we can. The projections of data generated in Run 3 shows an extreme increase in generated data, creating a huge compute load. Therefore we need to be able to run batch on any hardware we can get our hands on. When buying hardware for our data centers, CERN has typically bought the same hardware for batch and disk servers.

1.1 WLCG

The LHC produces an enormous amount of data. With over 99.9% of the data filtered it is expected to gather around 50 Petabytes in 2018. This is beyond the capabilities of CERN to process. Therefore the WLCG project was started.

The WLCG project is a global collaboration of more than 170 computing centres in 42 countries, linking up national and international grid infrastructures.

The mission of the WLCG project is to provide global computing resources to store, distribute and analyse the ~50 Petabytes of data expected in 2018, generated by the LHC at CERN on the Franco-Swiss border.

CERN (2018)

Eck et al. (2005)

Bird et al. (2014)

1.2 Batch

CERN Batch system is a system to process CPU intensive workload. Key goals include maximizing utilization, throughput and efficiency, and to provide a simple platform for physics. Physicists are able to run a command on some input, and receive the output.

The batch system is distributed on the WLCG The sites are divided up into tiers dependent on location and type of computing it performs.

- Tier 0 (CERN): data recording, reconstruction and distribution
- Tier 1: permanent storage, re-processing, analysis
- Tier 2: Simulation, end-user analysis

All together these computer centers add up to approximately 350’000 cores and 500 PB of storage, and are running more than 2 million jobs per day.

Jones (2018)

1.3 EOS

EOS is a multi-protocol disk-only storage system in use at CERN. The system have been deployed at CERN since 2011. In 2015 the system had grown to 140 PB storage provided by 44.000 hard disks.

Peters et al. (2015)

1.4 HammerCloud

HammerCloud is a stress testing tool. The tool provides possibilities to submit a number of analysis jobs, or a steady flow of them, to operational sites. The running jobs can be seen and analyzed through a web-interface, which also provides historical data.

van der Ster et al. (2011)

1.5 Puppet

Puppet is a configuration management framework. The framework uses a Domain Specific Language to describe the desired state of a server. Puppet looks at the servers current state and only does the necessary configuration that is needed to bring the server to the desired state, making it idempotent.

Loope (2011)

1.6 LHC@Home

LHC@Home is a system for user around the world to donate their available CPU-time to CERN to boost CERNs own computational power. The system is built on the Berkeley Open Infrastructure for Network Computing (BOINC), enabling CERN to access large amounts of computational power otherwise not available to them.

Lombraa Gonzlez et al. (2012)

Chapter 2

Hypothesis

The CERN large storage system is made up by nodes with similar, or exactly the same, hardware as the CERN batch system. At the time of writing there is 1339 nodes in the large storage system.

These nodes work mainly with I/O operations. As a result of this they are often I/O bound[1]. Most nodes are not CPU saturated even when they are I/O saturated.

As we can see from this snapshot in figure (2.1) of the average CPU utilization on the ALICE EOS cluster is less than 20. Some of it is IOWait, which can be used by other processes. This hints at significant potential to utilize the CPU on these nodes for other purposes.

This lets us ask two important questions to find areas of focus to work towards.

Can we make use of some of these nodes? The nodes have been set up with software and build processes to facilitate storage. To utilize their CPU for different processes requires additional software. This inclusion of extra use cases should also not do any damage to the existing service of the nodes.

What value does this correspond to? The potential gain of computer resources should negate the increase in software complexity and strain on the primary service for the project to be considered successful.

[1] I/O bound is a condition in which operations are limited by time used on reading or writing to disk, rather than the processing time

Figure 2.1: Average CPU and Network utilization on the ALICE EOS cluster. Idle CPU shown in green.

Chapter 3

Project

The project was iterated over a number of PoCs (Proof of Concept) with increasingly larger scope. Each iteration was conceived after the results of the previous iteration. This chapter is built on and elaborates on the proceeding at CHEP 2018.

Smith (2018)

3.1 Proof of concept I

The initial proof of concept tests was done by Andrey Kirianov. To test the concept two different environments was simulated. The systems needed nodes to simulate the EOS disk servers, clients to generate I/O load, and computational load. The computational load was generated by running LHC@Home on the disk servers.

3.1.1 Setup

The setup of the test environments are shown in figure 3.1 and 3.2.

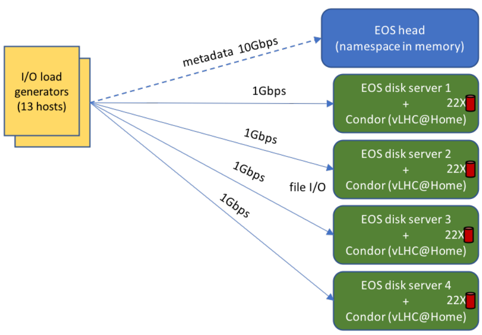

Figure 3.1: Environment 1

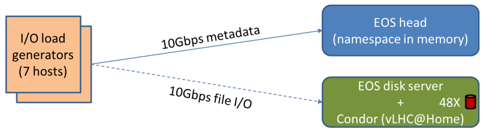

Figure 3.2: Environment 2

3.1.2 Results

When only running EOS the environment averaged 161 MB/s read and 93 MB/s write. Shown in figure 3.3, the eminent CPU operations while running EOS is IOWait.

Shown in figure 3.4, the CPU usage is similar to figure 3.3. The IOWait operation have been utilized. The environment averaged 156 MB/s read and 93 MB/s write.

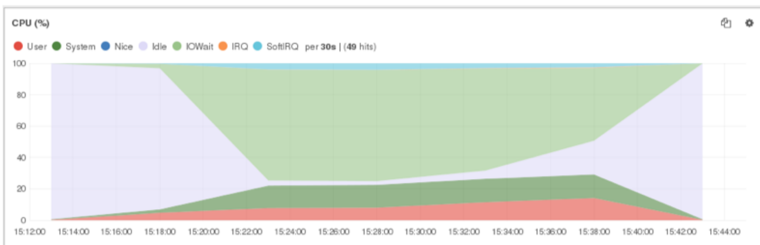

Figure 3.3: Environment 1 running with EOS

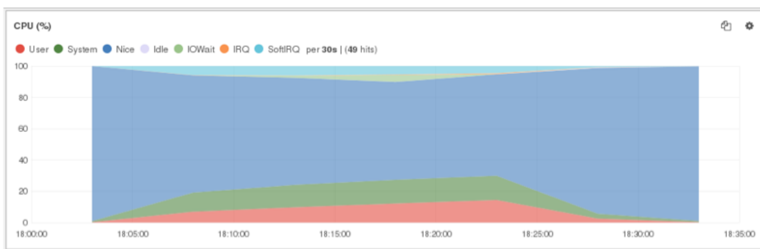

Figure 3.4: Environment 1 running with EOS and LHCHome

Environment 2 shown almost identical I/O result when running without compute payload and when running with, as shown in figure 3.5 and figure 3.6 respectively.

3.1.3 Conclusion

The results indicate that the computational service does not seem to significantly impact the storage service.

From this point on we need to consider which requirements emerges to turn this into production.

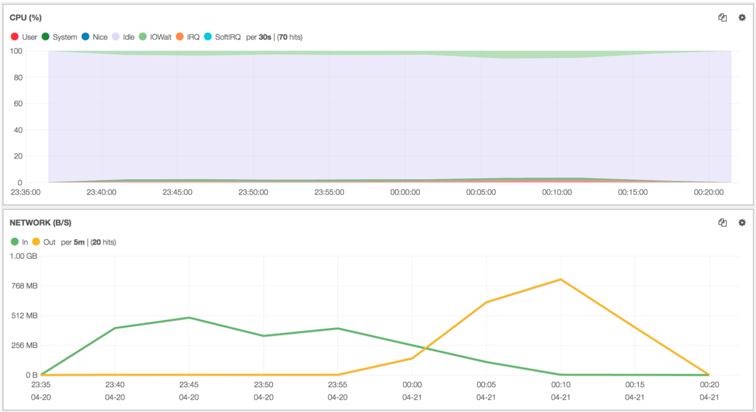

Figure 3.5: Environment 2 running with no compute payload

Figure 3.6: Environment 2 running with compute payload

The service needs to be deployable through our configuration management system. We are introducing two, possible competing, services on the same node. There will also be two teams responsible for maintaining the system.

The nodes main service is the large storage system. When using computing resources to perform batch operations, it is crucial that we don't compromise the EOS service. There should be a way of halting the computational tasks on demand.

We want to partition the resources. With this approach we can get a few benefits, including a guarantee that storage performance is not crippled. This makes it possible to provide accountable resources, which was not possible with LHC@Home. Using Cgroups[1] to limit the resource usage, we are effectively partitioning the resources needed by Condor. Condor will run the jobs in containers using docker.

[1] A linux kernel feature that isolates resource usage of a collection of processes, making it possible to introduce limits and accounting on the resources.

3.2 Proof of concept II

Named the BEER Pilot, an approach was made to explore the conclusions made in the first iteration. This new iteration were performed with participation from both the storage and batch team at CERN.

3.2.1 Setup

A puppet configuration is made based on the EOS hostgroup and using a modified EOS module and cerncondor module. This configuration is set up on three disk servers. The three disk serves had the following hardware.

- 48 x 6TB HSS
- 2 x E5-2630 v3
- 2 x 800GB SSD
- 10 Gbit network

A number of limits are set on these nodes. Using the Cgroup that condor is running in, memory is limited to 98GB and 4 physical CPU cores are entirely excluded. Condor is configured to offer 24 job slots and 96GB ram. The number of processes have been limited to 8000.

A question about security emerged. The EOS data disks should not be accessible to the Condor system. Data on the node is protected by UNIX ownership, and separation between users is done by Condor. In addition, jobs are run in containers which means that EOS data disks are not visible within the containers.

To generate I/O load a few clients clients would connect to the nodes. These clients used an internal tool called xrdstress to read and write files to the disk servers. The computational load was generated by submitting specific jobs called ATLAS Pile jobs to the batch system and specify in the submission that the jobs were only to be run on BEER nodes.

3.2.2 Results

The ATLAS Pile jobs consists of different stages, and are generalized to perform different tasks. Firstly it will download instructions and inputs from a remote disk server. Then performs the computations described in the instructions, and lastly uploads the output of the computations back to the remote disk server. This makes Pile jobs possible to submit in large bulks and still perform different instructions. These also means that the jobs utilizes different resources in the different stages, namely I/O in the first and last stage and CPU wall time in the computational stage.

When first submitting jobs to the BEER nodes we found that these stages created a spike in resource metrics for the current stage. The reason for this is that when all the job slots on the nodes started performing their newly received job they would all be in the first stage and start downloading their instructions and input. Likewise, since most jobs had similar duration in each stage, afterwards they would all be using their CPU to do the computation task they had received and later all would be uploading the output. This meant that our metrics would show a large number of I/O used in both the first and last stage, and the same for CPU wall time in the computational stage.

The metric spikes would not simulate the average I/O and CPU wall time for the system during production. We introduced a delay in the submission to saturate the nodes with jobs at different stages.

Seen in figure 3.7, there is three phases in the results. There is I/O simulation from the start until 08:00, and BEER Pilot jobs running after 20:00.

- Phase 1: I/O on, job off
- Phase 2: I/O on, job on
- Phase 3: I/O off, job on

There is no difference in I/O performance in phase 1 and 2.

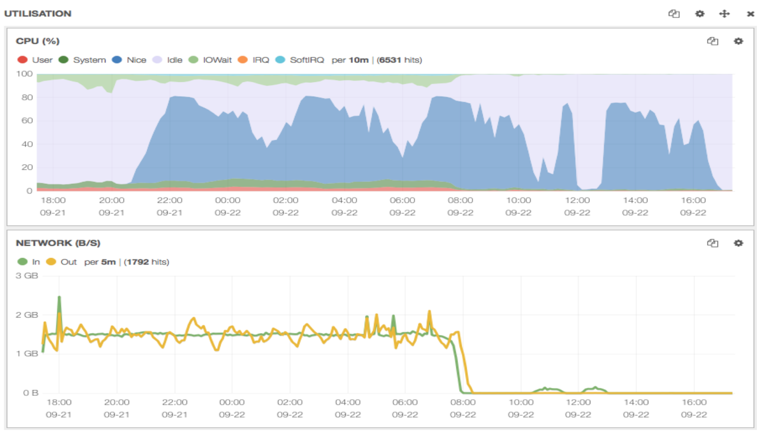

Figure 3.7: Results of BEER Pilot

3.2.3 Conclusion

The BEER Pilot confirmed our theory that both the batch and the storage service can coexist in the same nodes without interfering on each other. With that confirmation we could procede with the project and move towards setting the system up in our production nodes.

3.3 Pre-production

A step before actual nodes in production was to set the system up on nodes in an pre-production environment. This environment exist to root out any errors or faults that could occur when the system exist in production. The storage team provided four nodes from the EOS pre-production cluster. HammerCloud was used to submit jobs to Condor, which would get scheduled and started like a standard grid job.

3.4 Production

With a successful pre-production the project was confirmed safe to introduce to production nodes. 70 disk servers from ATLAS EOS Production was set up with the BEER system. Initial computational loads are submitted by ATLAS, but the nodes will at a later stage become part of CERN resources and the batch system in general.

Figure 3.8: BEER nodes in production

As seen in figure 3.8 the nodes have a similar I/O pattern both with jobs running, and without.

Chapter 4

Impact and further work

4.1 Next steps

Monitoring has been used throughout the whole project to examine the impact of the different services. With more detailed monitoring we can increase our chances to discover strains and benefits with the system.

There needs to be done a study of different experiments and processing steps to understand how the system behaves with mixed workloads.

The EOS large storage system have a much larger number of nodes still not CPU saturated, therefore it is desired to scale the system up to benefit from these resources.

4.2 Impact

The following are conservative assumption about the impact of the BEER project, both for the near future and with long term in mind.

In the following calculations we are assuming that we are able to utilize 40% of a machines CPU. Our measurements show that this is far less than what can be done when the node is fully saturated. We are also generalizing all machines to have 32 core CPUs.

In the near future we can expect 350 machines running BEER.

$$350 \times 32cores \times 0.4 = 4480cores \tag{4.1}$$

The number of machines running BEER should increase as the system is scaled up. Using an estimate of 1200 machines, almost all machines in ALICE EOS Cluster, we find this number.

$$1200 \times 32cores \times 0.4 = 15360cores \tag{4.2}$$

These numbers of cores will continue to scale parallel to the scaling of the EOS storage system. This makes BEER make an impact on the number of computational cores in the Batch system that will continue to grow in the years to come.

Bibliography

Bird, I., Buncic, P., Carminati, F., Cattaneo, M., Clarke, P., Fisk, I., Girone, M., Harvey, J., Kersevan, B., Mato, P., Mount, R., Panzer-Steindel, B., Apr 2014. Update of the Computing Models of the WLCG and the LHC Experiments. Tech. Rep. CERN-LHCC-2014-014. LCG-TDR-002.
URL `http://cds.cern.ch/record/1695401`

CERN, Apr. 2018. About — worldwide lhc computing grid.
URL `https://wlcg-public.web.cern.ch/about`

Eck, C., Knobloch, J., Robertson, L., Bird, I., Bos, K., Brook, N., Dllmann, D., Fisk, I., Foster, D., Gibbard, B., Grandi, C., Grey, F., Harvey, J., Heiss, A., Hemmer, F., Jarp, S., Jones, R., Kelsey, D., Lamanna, M., Marten, H., Mato-Vila, P., Ould-Saada, F., Panzer-Steindel, B., Perini, L., Schutz, Y., Schwickerath, U., Shiers, J., Wenaus, T., 2005. LHC computing Grid: Technical Design Report. Version 1.06 (20 Jun 2005). Technical Design Report LCG. CERN, Geneva.
URL `http://cds.cern.ch/record/840543`

Jones, B. D., 2018. How does batch work, internal document for Infrastructure Service team.

Lombraa Gonzlez, D., et al., 2012. LHC@home: a Volunteer Computing System for Massive Numerical Simulations of Beam Dynamics and High Energy Physics Events. Conf. Proc. C1205201, 505–507.

Loope, J., 2011. Managing Infrastructure with Puppet: Configuration Management at Scale. O'Reilly Media.
URL `https://books.google.ch/books?id=hYb2U-ZZByMC`

Peters, A., Sindrilaru, E., Adde, G., 2015. Eos as the present and future solution for data storage at cern. In: Journal of Physics: Conference Series. Vol. 664. IOP Publishing, p. 042042.

Smith, D., 2018. Sharing server nodes for storage and compute. Presented at CHEP 2018.

van der Ster, D. C., Elmsheuser, J., beda Garca, M., Paladin, M., 2011. Hammercloud: A stress testing system for distributed analysis. Journal of Physics: Conference Series 331 (7), 072036.
URL http://stacks.iop.org/1742-6596/331/i=7/a=072036