

Mulighetsvindu for en cloudbasert simulator

Christoffer Askeland

Master i kommunikasjonsteknologi
Innlevert: januar 2016
Hovedveileder: Jan Arild Audestad, ITEM

Norges teknisk-naturvitenskapelige universitet
Institutt for telematikk

Window of opportunity for a cloud based simulator

Christoffer Askeland

January 2016

PROJECT / MASTER THESIS

Department of Telematics

Norwegian University of Science and Technology

Supervisor 1: Terje Heierstad

Supervisor 2: Professor Jan A. Audestad

Preface

This is a Master's thesis in Tele-Economics at NTNU as a part of the study program Communication Technology. It was carried out during the autumn semester of 2015. This project has been carried out for Kongsberg Maritime, and is a result of the research I did as an intern for Kongsberg Maritime during the summer of 2015. The idea was brought up as a study of opportunities for a cloud-based simulator turning that time.

The assumed background of any reader of this report is a technical background (I.e a Bachelor degree or higher in computer engineering)

Trondheim, 2016-01-17

Christoffer Askeland

Acknowledgment

I would like to thank the following persons for their great help during the time writing this thesis:

Jan A. Audestad as my supervisor at NTNU, and Terje Heierstad as my supervisor at Kongsberg Maritime.

I would also like to thank Kongsberg Maritime for giving me the opportunity to write a Master's thesis for them.

C.A.

Summary and Conclusions

Kongsberg Maritime wanted to research the possibilities of running one of their lower-end simulators in a cloud-based environment. The simulator from Kongsberg Maritime is one of the world leading within the Offshore segment, so we knew the technology had to be further developed to support the heavy calculations and visualisation of the simulator. I have researched different solutions to this project, and tried to give reasoning for and against for every solution. Virtualization of different kinds and levels has is the solution that has proven to be the best. It has scalability, flexibility, the potential to reach to new customers and markets who where inaccessible due to various reasons (remote, turbulent etc.) and finally reduce the cost of each simulator, making Kongsberg Maritime more competitive in a already tough market. With a cloud simulator Kongsberg Maritime is able to meet customer demands of easier accessibility, no requirement to purchase extra hardware.

When it comes to the actual technical solution, the best alternative is to build a local server (at which site Kongsberg Maritime has offices) and use the GPU pass-through technology. Even though the NVIDIA GRID solution is more developed and is more fancy, it can not provide the same performance as a 1:1 GPU - Virtual Machine. This is still scalable, and meets all the customer's demands. Scaling with the GRID might be easier, but it will be more expensive for Kongsberg Maritime in the long run. By moving from a traditional CAPEX to OPEX means lower margins, since Kongsberg Maritime no longer can bundle hardware to the simulator like they do with the traditional simulator. By the time Kongsberg Maritime will start building a cloud-based simulator, new and improved hardware might have been release with regards to NVIDIA GRID, making this the better choice for the future.

Pricing the cloud-based simulator is not easy with just one pricing model. The market is so diverse, that making a model that fits everyone is difficult. Therefore, I proposed a mix of three different pricing models to easier satisfy the customers.

Sammendrag

Kongsberg Maritime ville undersøke mulighetene til å kjøre en av sine low-end simulatorer opp i en skybasert server. Kongsberg Maritime er verdensledende når det gjelder simulatorer innen offshoresektoren, så vi visste at teknologien måtte videreutvikles for å støtte tunge kalkulasjoner og visualisering av simulatoren. Jeg har undersøkt ulike løsninger i dette prosjektet og forsøkt å begrunne fordeler og ulemper ved hver løsning. Virtualiseringer på ulike måter og nivåer som har vist seg å være den beste løsningen. Den har skalerbarhet, fleksibilitet og potensial til å nå ut til nye kunder og markeder som var utilgjengelige av ulike årsaker og til sist redusere kostnadene av hver enkelt simulator, noe som gjør Kongsberg Maritime mer konkurransedyktig i et allerde tøft marked. Med en Skybasert simulator blir Kongsberg Maritime i stand til å møte kundekrav om lettere tilgang uten krav om å kjøpe ekstra hardware.

Når det gjelder den aktuelle tekniske løsningen, er det beste alternativet å bygge en lokal server på et sted hvor Kongsberg Maritime har kontorer og bruke GPU gjennomgang teknologi. Selv om NVIDIA GRID løsningen er bedre utviklet og mer fancy, så kan den ikke gi den samme ytelsen som 1:1 GPU Virtual Machine. Den er fremdeles skalbar og imøtekommer alle kravene fra kundene. Skalering med GRID er kanskje enklere, men det vil være mye dyrere for Kongsberg Maritime i det lange løp. Ved å bytte fra en tradisjonell CAPEX til OPEX vil det bli en mindre margin siden Kongsberg Maritime ikke lenger kan binde hardware til simulatoren som en del av en pakkeløsning. Når Kongsberg Maritime vil begynne å lage en skybasert simulator, vil antagelig ny og forbedret hardware ha kommet på markedet med hensyn til NVIDIA GRID, noe som vil gjøre dette til et bedre valg i fremtiden.

Det er ikke lett å prisse en skybasert simulator ved hjelp av en prismodell. Markedet er så mangfoldig så det er vanskelig å lage en modell som passer for alle. Derfor har jeg foreslått en blanding av tre ulike modeller for å tilfredsstille kundene.

List of Figures

2.1	Traditional Architecture vs Virtual Architecture	6
2.2	GPU manufacturers market share(1)	9
2.3	An overview of the three different types of GPU Virtualization	9
2.4	A simple overview of the vSGA architecture	10
2.5	GPU Pass-through	11
2.6	Detailed view of NVIDIA vGPU	12
2.7	Overview GRID K2's profiles(2)	12
2.8	Difference between Native and hosted hypervisor	13
2.9	WebRTC's architecture(3)	16
4.1	A visualization of Horizon 6 with NVIDIA GRID(4)	30
4.2	A visualization of View Composer(5)	31

Contents

Preface	i
Acknowledgment	ii
Summary and Conclusions	iii
1 Introduction	2
1.1 Background	2
1.2 Objectives	3
1.3 Limitations	3
1.4 Approach	3
1.5 Structure of the Report	4
2 Technology	5
2.1 Server	5
2.1.1 Virtualization	5
2.1.2 Graphical Processing Unit	8
2.1.3 Hypervisor	13
2.2 Alternative Solutions	14
2.2.1 Simulator directly in the browser	15
2.2.2 Thin clients	16
2.2.3 Run the Visualization locally	17
3 Business Model and Marked Analysis	18
3.1 Marked Analysis	18
3.1.1 Software-as-a-Service	19

<i>CONTENTS</i>	1
3.2 Pricing Model	19
3.2.1 Volume based pricing	19
3.2.2 One-time fee	19
3.2.3 Flat rate	20
3.3 Business Model	21
3.3.1 Timing	21
3.3.2 Customer Segments	21
3.3.3 Value Proposition/Key Activities	22
3.3.4 Channels	22
3.3.5 Customer Relationship	23
3.3.6 Revenue Streams	23
3.3.7 Key Resources	24
3.3.8 Cost Structure	25
4 Discussion	27
4.1 Similar Products	27
4.2 Build or Rent Server	28
4.3 Product configuration (with NVIDIA GRID)	29
4.4 Product configuration (without NVIDIA GRID)	30
4.5 How will a Cloud simulator affect existing products?	31
5 Summary	33
5.1 Summary and Conclusions	33

Chapter 1

Introduction

1.1 Background

Kongsberg Maritime wants to create a cloud-based simulator for their low-end simulators, which in this case is the machine room simulator. At present time, the simulator runs locally on every computer, which makes for a huge cost in regards of both hardware and service/maintenance. By moving the the simulator "into the cloud", they hope to reduce costs and create a product that can potentially scale well.

For the last 40 years, Kongsberg Maritime has been the world leading company in providing Offshore simulators. As the competition has become harder, Kongsberg Maritime looks to new ways of grabbing market shares. The next natural step is to move the simulator into the cloud. This Master's thesis is a research for what they call "a window of opportunity for a cloud based simulator", and will be the premise of an eventually new product line.

In this section, you should present the problem that you are going to investigate or analyze; why this problem is of interest; what has, so far, been done to solve the problem, and which parts of the problem that remain.

The literature used in this thesis is mostly based on articles from developers within virtualization, GPUs/ graphics cards and business models. Since the only platform Kongsberg supports today is VMware, they became the natural source for information about virtualization. They support NVIDIA GRID, the basis of the study.

1.2 Objectives

Window of opportunity for a cloud-based simulator

- What technology is required to to run a cloud based simulator?
- Does this technology exist today? Which problems needs to be solved?
- What changes must be made to the software to make this possible, but still deliver what the end-user want?

A Market analysis and a business model for a cloud based simulator

- What does the end-user want?
- Go from CAPEX to OPEX, pros and cons
- Which markets does KM intend to reach? Will new markets be created?
- what shall the price of the product be?
- How will this affect already existing products from KM?
- Define a product configuration that can be run in a cloud.

1.3 Limitations

The only limitation to this project where the possibility to test the GRID K2 cards for performance. With the price of 30000 NOK, they are not something to purchase to test and throw away if it not sufficient.

1.4 Approach

For "window of opportunity" the approach is to look at similar products and see how they are developed. I started at VMware and NVIDIA since they were the providers that KM already used. As for the marked analysis, I had many conversations with employees at Kongsberg Maritime during my internship this summer as well as the times I visited this summer.

1.5 Structure of the Report

In this report I will go through the existing technology and how the market looks today in the first chapter. The second chapter will be about the market analysis and the creation of a business plan. The next chapter will be where I discuss solutions and try to answer every objective if it has not already been answered in the report.

Chapter 2

Technology

2.1 Server

The server's purpose is to share software or hardware resources with its clients. In this case, the server's role is to distribute, handle and serve every connected client. The machine room simulator will be installed as well as other software required to handle virtualization. What software and hardware the server actually contains will be discussed throughout this chapter, and whether to build a private server or rent a server from i.e. Amazon.

2.1.1 Virtualization

So what is virtualization? Virtualization is the act of creating a virtual version of an instances. An operating system, storage device, a resource in a computer network or a computer hardware platform are some of the instances that can be created in a virtual environment. The physical characteristics of the computing platform are hidden, and instead presented as another abstract computing platform. There are several different types of virtualization:

- **Server Virtualization:** This is the idea of divorcing yourself from the limitations of hardware, and be able to run multiple virtual machines, or clients, on a single physical server.
- **Desktop Virtualization:** This is taking it one step further. Instead of talking about just individual workload, we're talking about arrays of desktops and distribute them to users.

- **Application Virtualization:** This is server back to the server workload, but now the server is distributing applications as a service out to the end-user. The user should not notice any difference on running the application on a server compared to running it locally.

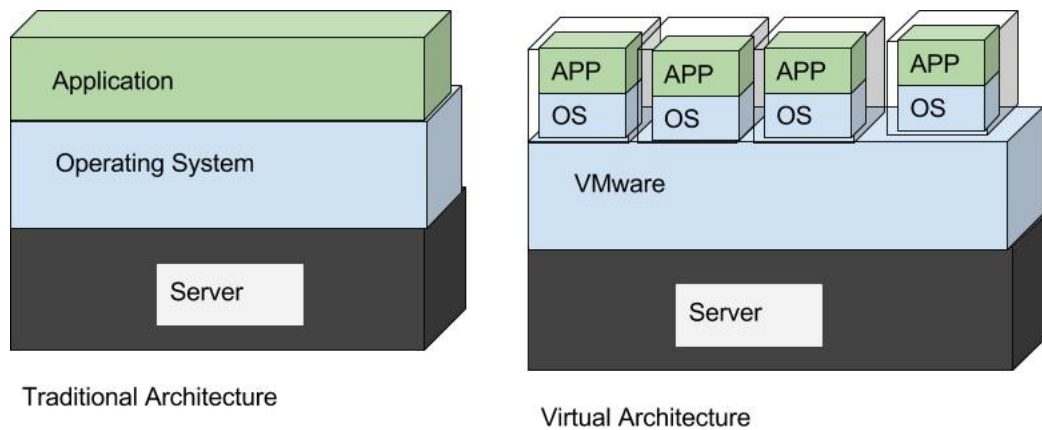


Figure 2.1: Traditional Architecture vs Virtual Architecture

In figure 2.1, the difference between the two types of architecture are visualized. On the left we have the traditional server with a server rack, a host OS and an application running on top of that. On the right however, a VMware Hypervisor is implemented (more on hypervisors later in this chapter) with four virtual machines running on top of it. With virtualization, we get an increased density from a physical storage perspective. In the the physical world, we get a 1:1 for each server, but in the virtual world we get a n:1 where many users run simultaneously on one server. This again means an improved resource optimization but without sacrificing performance.

Other advantages of virtualizing a server includes:

- **Partitioning:** The server is now able to run multiple operating systems on one physical machine. It is also able to share physical resources between virtual machines.
- **Portability:** A Virtual Machine (VM) is very portable due to the fact the entire machine is saved as a file, which makes moving, copying or exporting a machine just as easy as with any other file.
- **Security:** The hardware is isolated from the the operating system, limiting the access a user can to the physical server. This also means that one VM cannot see another from the

client perspective. Also, recovery is as easily done as restoring a file.

- **Agnostic:** The user can export the VM and use it on a similar, or different, physical server.

These advantages apply to the server side. But what about the clients or desktops? What are the challenges they face with the traditional setup compared to a virtual setup? We still face the same problem with a 1:1 computer to worker ratio with traditional computers. Whenever a company hires a new employee, the company must buy a new PC. You also face the problem of not being able to share the resources because every PC "belongs" to each separate employee. It all becomes a resource management issue where computers are ordered in bulks, and they are limited to a few, often just one, types of computers because homogeneity saves money. The employer ends up buying for the most demanding group of users, which is basically an overkill for the rest. Then you have the high end users, where each computer often is custom made. These are expensive, and the user is stuck since they can't do their work without this computer. Often it requires more processing power or the GPU unit in it. And then you have the people who have to maintain and manage all of these computers. The company would need staff to manage it all. Security issues, Anti-virus for every computer, updates, patches etc. all becomes more challenging and expensive for every new computer brought into the company. With a centralized server, you will only face hardware issues in this one instance. In an environment where every user has its own computer, potentially every computer can experience hardware failure and it will take longer and be more expensive to fix every problem. And what if you have smaller offices spread across a larger geographic area?

All of these issues have made more businesses move towards a virtualized work environment. The cost savings alone should be enough to push a company into setting up a virtualized system. Much money can be saved on hardware, service, software and staff. It also brings more mobility and flexibility to worker. But it is not only the business themselves that benefits from these changes. The end user often wants the latest and greatest when it comes to computers. They are looking for frequent refreshes of operating systems and applications. Upgrading the graphics or power as the demand increases is much more cost efficient in a virtualized setting for the company than to upgrade each computer separately.

The last type of virtualization that will be covered here is the application virtualization. Application virtualization means distribution of multiple copies of an app from a single physical server. The app can be streamed to any device that supports it. It will be centrally managed, which makes pushing out updates really easy. The ease of support is also important to remember.

A few examples of solutions are:

- Citrix Metaframe, now called XenApp. This is an solution where they basically install an OS and run an instance of it and deliver it with a remote graphic type solution. This has been around for 20 years.
- VMware ThinApp
- Microsoft App-v

All of these solutions are tools to centralize the app, deploy it in the data center one time and then allow a large amount of users to use it without having to deploy it themselves.

2.1.2 Graphical Processing Unit

The Graphical Processing Unit (GPU) is the processor of every dedicated video card on the market or in some cases integrated on the motherboard. It is a specialized electronic circuit built and designed to alter and rapidly manipulate memory to accelerate image creation in a frame buffer. The GPU has become very good at manipulating graphics and processing of images. The highly parallel structure makes them even more efficient than a CPU at processing big blocks of visual data. The CPU is much faster at processing a linear task, but the GPU excels at parallel processing. This has made it very valuable to research and in processing linear algebra and statistics.

The NVIDIA were the first ones to come up with the term "GPU" when they launched their GeForce 256, with the advertisement of "the world's first GPU"(6). With the introduction of CUDA in 2007, the GPU became a more generalized computing device. As figure 2.2 shows, NVIDIA is by far the largest provider of desktop GPU, with a market share as high as 82%.

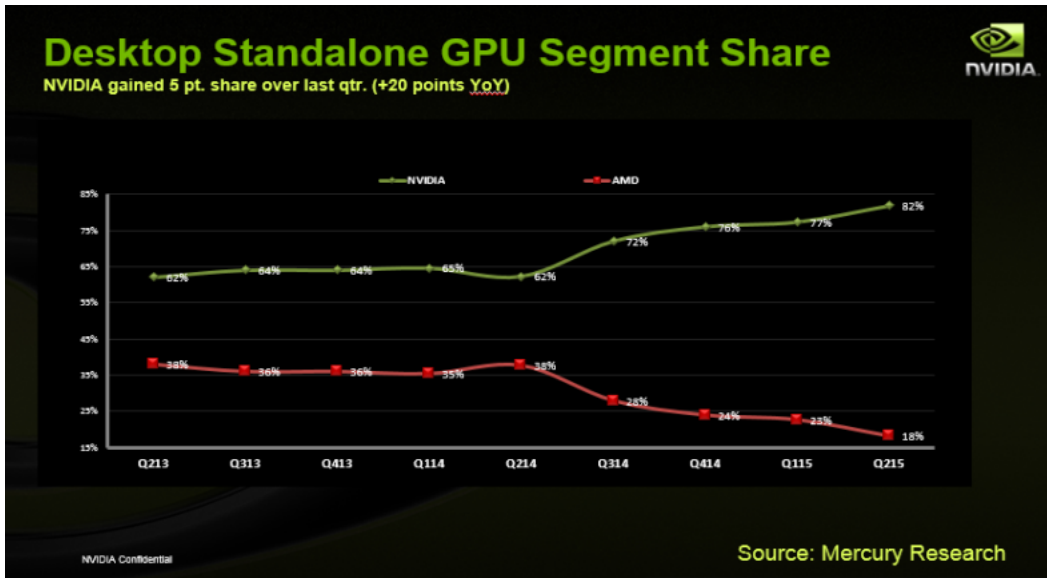


Figure 2.2: GPU manufacturers market share(1)

NVIDIA is also the provider of graphic cards for KM and their simulators. KM uses one of the standard GeForce GTX cards (which specific model is business confidential). The card is used to render images and to do all calculations within the simulator. The simulator calculates every internal factor as speed, course etc. as well as every external factor (i.e waves, wind) affecting the ship. NVIDIA where also the first to take virtualization seriously, and launched their GRID technology in 2013. Figure 2.3 shows an overview of the three different types of graphics virtualization.



Figure 2.3: An overview of the three different types of GPU Virtualization

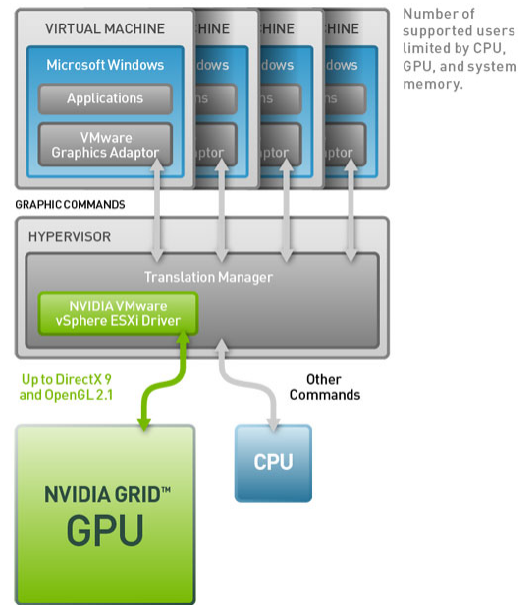


Figure 2.4: A simple overview of the vSGA architecture

2.1.2.1 Shared GPU

Shared GPU (or vSGA) is the simplest and most primitive use of GPUs in virtualization. With vSGA, every user shares the GPU equally, and the performance is not very good compared to other GPU virtualizations. It is however significantly better than to not have vSGA. With the vSGA driver installed in the hypervisor, a VM can use the capabilities and benefits of image rendering (or other heavy graphical processes). Figure 2.4 shows how the vSGA is implemented in a virtualized environment.¹

2.1.2.2 GPU Pass-through

This technology is also called VDGA (Virtual Dedicated Graphics). To describe it simple, the GPU "drills a hole" in the hypervisor so that every VM gets a dedicated GPU. The PCI device (here a GPU) is routed directly to the VM from the host.

Figure 2.5 shows a visualization of how a VM gets a dedicated GPU from the host. The GPU will show up in the device manager, so the VM believes it has its own GPU, and GPU is only available for that one VM. This gives the user 100% compatibility that it would get from a phys-

¹From the site: <http://www.nvidia.com/object/vmware-horizon.html>

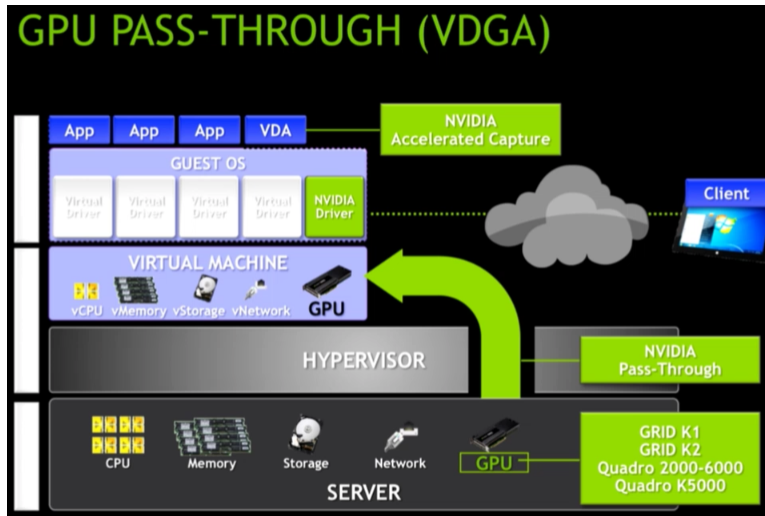


Figure 2.5: GPU Pass-through

ical unit. The problem with this solution is density. There is only room for a limited amount of cards in a server. The amount of VMs a server can support is drastically reduced, and the virtualization loses a some of its purpose.

So how is the density problem solved? NVIDIA has come up with a solution which is called vGPU.

2.1.2.3 vGPU

vGPU is NVIDIA's name for emulated GPU. vGPU is similar to VDGA in many ways, but as seen in figure 2.6, there is no "hole" in the hypervisor. Instead, a piece of software called GRID vGPU Manager is placed in the hypervisor which gives the host the ability to create instances of the GPU.

This solution is sort of a compromise between the shared GPU and VDGA, where the host supports more VMs with dedicated GPUs than the number of cards placed in the host. As of now, it is only NVIDIA GRID K1 and K2 cards who supports this technology. Intel has their own open source project called INTEL GVT-G, but this is still only considered work in progress(7). This is just a software solution, not a complete product with a GPU card and supplementing software. When it comes to hypervisors, both VMware and Citrix supports vGPU (see 2.1.3).

But how good is the performance of a NVIDIA GRID K1/K2 compared to a regular GPU? It

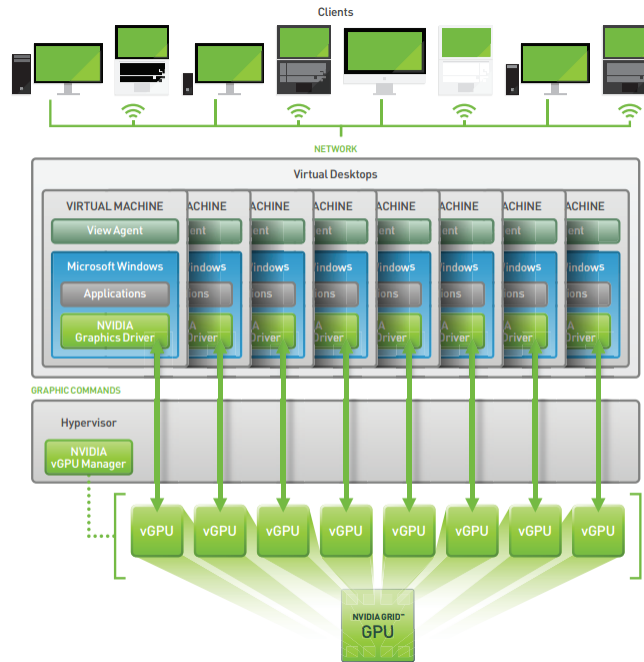


Figure 2.6: Detailed view of NVIDIA vGPU

is important to remember that the partitioning is dynamic, meaning that performance depends on the numbers of VM connected to the host. The access is governed by the manager. The best place to start the analysis of the GRID technology is look at the specs of their best card; NVIDIA GRID K2.

Use Case	Virtual GPU			Per-VM Resources		
	vGPU Profile	# per GPU*	# per Board**	CUDA Cores (time-slice shared)	Frame Buffer (Dedicated)	Max Displays (2560x1600)
Designer	GRID K280Q	1	2	1536	4GB	4
Designer	GRID K260Q	2	4	1536	2GB	4
Designer/Power User	GRID K240Q	4	8	1536	1GB	2
Designer/Power User	GRID K220Q	8	16	1536	512MB	2
Entry Designer	GRID K180Q	1	4	192	4GB	4
Power User	GRID K160Q	2	8	192	2GB	4
Power User	GRID K140Q	4	16	192	1GB	2
Power User	GRID K120Q	8	32	192	512MB	2

Figure 2.7: Overview GRID K2's profiles(2)

In figure 2.7 a overview over the different profiles of the K2 card is shown. It is known from KM's side that the simulator uses GeForce cards with at least 1536 CUDA cores (which excludes K1) and a dedicated frame buffer of 2GB. In this case, if the profile supports 2 or 4 displays are irrelevant since the machine room simulator will only run on two displays anyway. This means

that each GRID K2 card can support up to 4 VMs with acceptable performance. This means it now becomes a question of cost / benefit whether to buy one GRID K2 card or four GTX GeForce cards with the same performance. This will be discussed further in chapter 4

2.1.3 Hypervisor

The hypervisor is the hardware, firmware or software between the physical server and the VMs. It is also called a Virtual Machine Monitor, and is the instance that creates and runs VMs. The guest operating system (or clients) is presented with a virtual operating platform, and the hypervisor manages the execution of the client. A variety of operating and multiple instances may share virtualized hardware resources. There are two different types of hypervisors: native(bare-metal) and hosted. In figure 2.8 the difference between the two are illustrated.

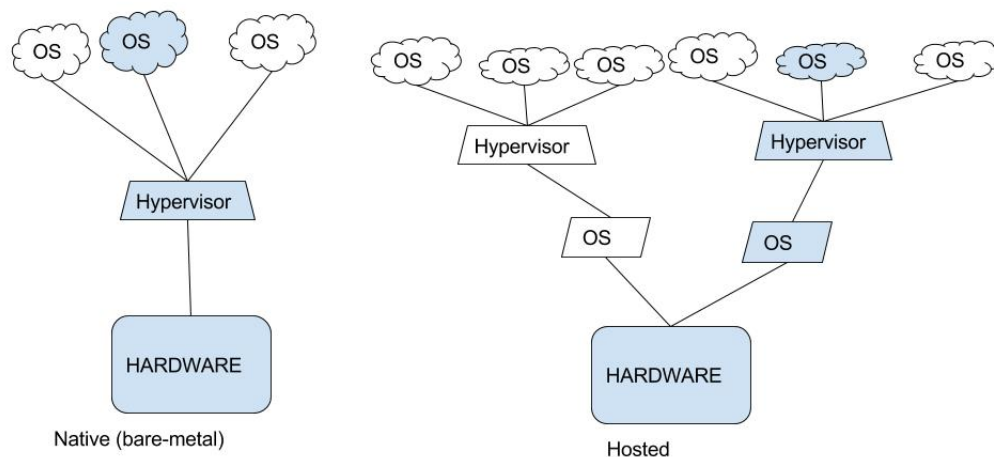


Figure 2.8: Difference between Native and hosted hypervisor

- **Native:** A bare-metal hypervisor runs directly on top of the servers hardware. This is to control and manage any guest operating system. Any process the client (guest OS) runs is on the host.
- **Hosted:** A hosted hypervisor is software that runs on a operating systems the same way other programs do. The guest OS is abstracted from the hosts OS by the hosted hypervisor.

So which one to choose for a cloud based simulator? Since a simulator is being deployed, performance and speed are what we're looking for. By using a hosted hypervisor, another layer

is added between the VM and the host, a lot of problems can quickly show themselves. Performance can go down, and latency up. With this extra layer, the VMs are no longer isolated, creating potential security issues. For regular desktop use the hosted hypervisor is often enough, but for a server like this the native hypervisor is the best choice. Within the segment of native hypervisors, there are three main providers:

- VMware vSphere
- Citrix XenServer
- Windows Hyper-v

All three of these products are good alternatives. VMware has been in the business for the longest, but are usually more expensive(8). Citrix XenServer is an open source alternative, which is used of the largest IaaS provider Amazon Web Services (AWS)(9). Windows Hyper-V works very well with other Microsoft products, but it is not yet decided whether Linux or Windows shall be used as the operating system on the server. KM has traditionally been using VMware, and is what they support today. So to avoid resources being used on adapting the system to a new hypervisor, choosing VMware seems like the best choice here. When vGPU where released by NVIDIA, it was only Citrix XenServer who supported the NVIDIA GRID cards(10). However, at the end of 2015 VMware and NVIDIA finally announced that VMware Horizon (thus vSphere too) now supports vGPU as well(4).

2.2 Alternative Solutions

NVIDIA's GRID combined with VMware's Horizon 6 solution has been tested by a number of different customers with just as many success stories with it. However, there are other ways to solve the challenges of creating a cloud based simulator. By running the simulator in many independent instances of the simulator on the server, and then distributing the output to users through different channels opens up for many opportunities (and challenges).

2.2.1 Simulator directly in the browser

With the technology available in the browser with the launch of HTML5, running the simulator straight into a browser as a video stream from the server should not be a problem. HTML5 is supported in every browser, so this should be possible regardless of which browser the user uses. By compressing the rendered video on the server using NVIDIA video codec, more precisely the NVENC(11), the bandwidth required to stream the video is reduced a lot without adding much latency. NVENC is an open-source project hosted by NVIDIA as a software which enables the GPU to perform hardware accelerated video H.264 encoding. As mentioned in section 2.1.2, the GPU excels at parallel image processing. A similar option for GPU accelerated video encoding is Main Concepts CUDA H.264/AVC(12), but this is basically the same as NVENC. With the raw video now converted to H.264, or mpeg4 if you like, a protocol is needed for streaming the video from the server to the client:

- **WebRTC:** Web Real-time Communication is an open source protocol designed for real time communication in the browser(3). It uses VP8 video encoding, which has full support with HTML5. There is little to no difference between VP8 and H.264 in quality(13), so this protocol is good for streaming video from the server to the browser. An application must be made in both ends, as shown in figure 2.9
- **OBS:** Open Broadcast Software is an open source software made for video recording and live streaming(14). The system supports GPU accelerated video encoding, which is good to keep the latency as low as possible.

There are a few drawbacks with a simulator directly in the browser. The problem comes when the browser shall handle the input from the user. Registering which actions the user does is difficult with a video stream, and the browsers have little to no support for this to this date. By adding some sort of software layer above the machine room simulator since it will be a static placement of every button, input could be recorded and sent back to the server as user input. This would mean that every instance of the simulator must have a software overlay, and would be very time consuming to update. The user experience will be worse since handling more than one display with a video stream in the browser becomes messy. This solution does not scale very

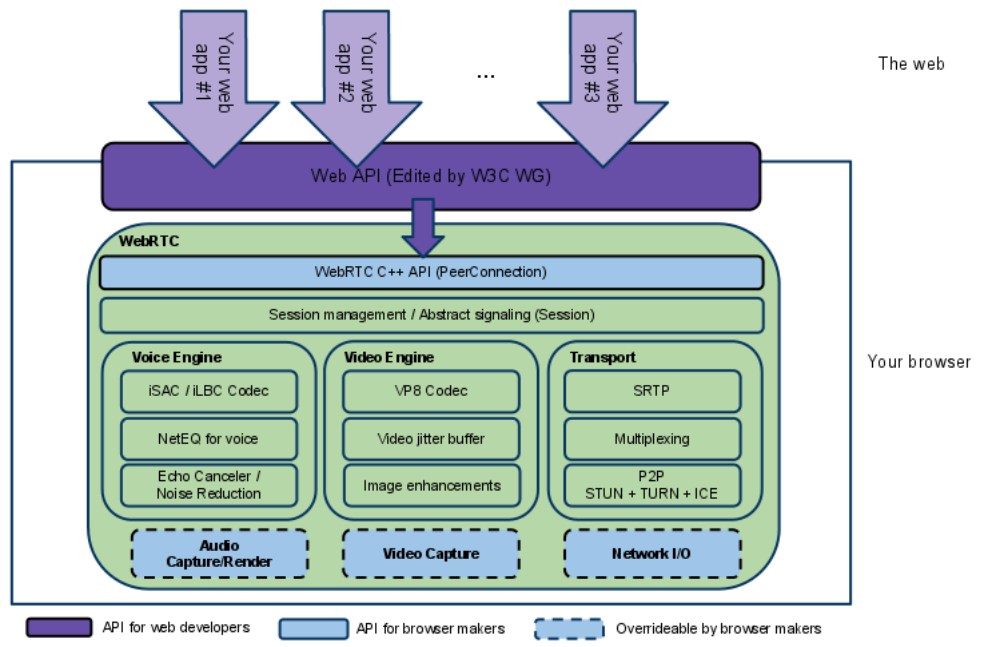


Figure 2.9: WebRTC's architecture(3)

well either, as it probably needs one new GPU per customer to handle all the calculations and image rendering, and a new GPU dedicated to video encoding for every X new customer. This solution quickly becomes really expensive.

2.2.2 Thin clients

By centralizing the software into a dedicated server and then have thin clients access it through VDI (Virtual desktop infrastructure) is another solution to solve some of the issues KM faces today with their simulator. This is sort of a compromise between the product KM is delivering today and the remote cloud simulator they wish to create. With the application centralized, it sets almost no requirements for the thin client. All the thin client need is ethernet connection to the server as well as hardware compatible to handle a VDI application, which is basically a motherboard with any integrated GPU. Even though this solution will reduce the expenses in hardware, it still does not solve the problems KM addressed. It will not be able to access more remote or turbulent areas as KM will still have to send a crew to the location to install (or perform service on) the product. Not only that, this product line will probably cannibalize on the other low-end simulator sold by KM, doing more harm than good for the company as a whole.

2.2.3 Run the Visualization locally

This solution is similar to the one in section [2.2.2](#), but there are some differences that might reduce some of the cost and not cannibalize as hard on other products. By centralizing the simulator on a server, making it do all the calculations and then send output to one of the connected clients. The client then visualizes the given information locally. This solution splits the work load in half, which again sets a lower requirement to hardware for the client.

Chapter 3

Business Model and Market Analysis

In this chapter a Business model will be developed based on the market analysis given by KM.

3.1 Market Analysis

As one of the world leaders in Offshore simulators, there is a constant pressure from the customers and competitors to stay a head of the rest to gain higher market shares. With Transas launching a local cloud service for SUNY, KM must start developing a similar product to not fall behind. Whenever KM is out negotiate new contracts with customers, they are asked to deliver a technology they do not yet possess; easier access to the simulator without buying KM's hardware¹. This especially applies to lower end simulators like the machine room simulator that does not require heavy visualization on the client side. This is the reason why KM has started to look at a cloud based simulator to remove all need of dedicated hardware and easier access to the simulator. What the customer still want is pre-defined training exercises, scenarios basic course material. They also want to be able to be the instructor, and both rights as instructor in the trainer as well the other material is included in the package when they buy access to the cloud.

¹It's hard to document this, because it comes from conversations with the supervisor at KM

3.1.1 Software-as-a-Service

By centralizing the simulator on a cloud server, KM moves from a traditional product, and into the world of offering a product as a service. This is called Software-as-a-Service, or SaaS for short.

From a traditional product to SaaS Going from selling a product to offering it as a service, changes the dynamic of the revenue stream drastically. With moving from a oriented CAPEX revenue stream to OPEX, the price of the product goes down and so does the cost. It will lower the margin, but KM will be able to reach a wider market and with a more flexible approach.

3.2 Pricing Model

This project will use three different pricing models to utilize the market potential better. Because there is such a large difference in the customer segment, a flexible pricing model is good for both parts.

3.2.1 Volume based pricing

In a volume based pricing model, the customer pays for the amount of volume consumed. This model is most common in the telecom business, but with the accession of SaaS, it is becoming more popular in other markets as well. Volume-based pricing will be used on the most congested servers. This model is not suited for the schools or businesses where they have access to internet or computers at all time. Instead this model will be used for on-board trainers on ships which have a more dynamic schedule. The customer does not want to pay a flat rate if they do not know how often or much they get to use the simulator. Both AWS and Google Cloud uses this model.

3.2.2 One-time fee

Having a one-time fee pricing model is how the traditional simulator is priced, so it is only natural that the cloud simulator follows the same model. This model will apply to more than one

type of customers:

- Buying the technology: KM will not only be selling the simulator as a SaaS. The customer is able to by the technology behind it and with a given number of floating licenses. Each licence will be an additional charge, however quantity discounts might be given. In this case, installation and service will also cost extra.
- Buying full access to a server: If a customer wants full access to the server instead of just buying time at given hours. With full access they can use the simulator whenever they want. However, there will be different prices depending on how much resources and how many licences they want.
- Customers who wants to pay a fee once instead of less over time. They will have the same access as the flat rate customers, and will not be charge based on the amount of data used.

3.2.3 Flat rate

Since the cloud simulator now is considered a SaaS, the customers should consider that they are renting access to the server more than actually buy a product. With a flat rate pricing model, the users will pay a given yearly fee regardless of the amount of traffic used. There are some important factors to why flat rate is a popular pricing model:

- Insurance effect: With a higher, stable fee the user avoid the discomfort of a unexpected huge bill, despite the fact that the user most likely would have paid less each year with a volume based model.
- Mental accounting: Users are often satisfied with paying for more than they use just to simply the evaluation of each months cost.
- Users tend to expect higher usage than they actually end up with, which means they are paying more than what a volume based solution would have cost them.

3.3 Business Model

"A business model describes the rationale of how an organization creates, delivers and captures value" -Osterwalder (15)

To create a business model for a Cloud-based simulator, this report uses Osterwalders business plan canvas as template. The business plan is divided into nine different segment to structure the model. However, "key activities" and "Value proposition" are done together since they are supplements to each other.

3.3.1 Timing

Having the "First-mover advantage" will be critical for KM in this market. The competition is tough, with companies as Transas always pushing KM to deliver better products. If KM is successful in launching a cloud-based simulator before everyone else, they will be able to capitalize on a large part of the market. Transas has already deployed what they call a "cloud simulator" at the New York State University(16), but this is only a local cloud and can only be accessed at the university. KM plans to launch a cloud based simulator which can be accessed from any part of the world. It will most likely be deployed on sites where KM already has established offices (i.e. China, Australia, Singapore, Norway).

3.3.2 Customer Segments

The customers of the cloud-based simulator will probably be the same as the ones already using their products today. This means schools and shipping companies that wants to train and educate the users. KM is already established in Asia, Europe and North-America, but with a cloud-simulator, they will now be able to reach more remote areas since they do not have to travel to different sites to set up a simulator. This will expand their markets to pretty much every corner of the earth, given that a server is close enough to access with acceptable ping. By reaching out to new areas, some customers might be satisfied with the cloud simulator that they wishes to buy a larger scale simulator like the K-Sim Offshore simulator. Low volume customers will be much easier to satisfy since they are now buying access (or license) to a server, which will save them the costs of service, installation and hardware. Turbulent areas can now too get

access to the simulator, as KM does not have to send any of their employees into these areas.

Many of the larger shipping and offshore customers wants to have on-board trainers whenever a ship is in port and has the option of connecting to Internet. With a cloud based solution, they will need much less hardware installed on-board and no service on the ship if something goes wrong.

3.3.3 Value Proposition/Key Activities

In Osterwalders Canvas, under the point "Value propositions", he asks two very important questions: "Which one of our customer's problems are we helping to solve?" and "Which customer needs are we satisfying?" (15). The problems solved by delivering a cloud based simulator are that customers are no longer forced to buy hardware along with the software/application they want. They can now use the simulator on any computer or laptop they find suitable as long as it can handle a VM. In reality, the customer just want a simpler access to the simulator, that is what they will get with this solution.

Some customers also want a cheaper solution than what they are getting today. With installation, service, upgrades, hardware etc. the cost starts racking up. By using a remote server as basis for the simulator instead of a in-house solution, the customer can skip every expenditure except for the cost of access to the simulator.

The needs of the customer that will be satisfied are more mobility, flexibility and scalability. It is much easier to purchase an additional license/access code than to order a new setup of computers that have to be shipped from Norway. The customer can access the simulator from basically any location as long as it has internet connection.

Being the first-mover here is, as mentioned in 3.3.1, very important. This might be the most important activity KM does. Churning is not very common in this business, so when you first get a customer, it is likely to stay.

3.3.4 Channels

Since KM is already one of the worlds largest provider of Offshore simulators, reaching out to customers should be simple. KM can choose to contact already existing customers about this

product, or enter fights over contracts with this product as their selling point. Partners or already existing customers of KM will be a good way to spread the word too.

KM is represented at user conferences for simulators all over the world, which is a good opportunity to reach out to new customers. Some money should be invested in advertisement as well, as a way to reach out to customers unable to attend user conferences or other events with KM present.

3.3.5 Customer Relationship

To not lose every obtained customer, a good customer relationship must be established. Even though not every customer will be able to negotiate the price of either access or a license, some of larger ones will have this option. A quantity discount could be accepted with a purchase of a large amount of licenses. If a customer purchases access for 100 students to a cloud simulator, the customer is able to customize the amount of access each student should have. Maybe 10% will have access 100% of the time, while others are limited to a give time each day/week.

It is important that the servers are always up and running. Down-time may affect the relationship to the customer, especially if it happens often and takes long time to fix. That is why the servers must be monitored 24/7 at all locations so that a problem can be fixed as soon as it appears. On sites where KM has only sold the technology, and not access a cloud server, a service team should be able to respond in short notice. This will of course cost extra, and should be optional for the customer.

3.3.6 Revenue Streams

For this product there will be two main streams of income; buying access to a server or buying licenses to host internal grids/clouds. Each stream will have separate pricing models. For buying access to the server, a mix of three different pricing models will be used (For more info, see 3.2): one-time fee, flat rate and volume based. With the use of different pricing models and dividing the users into "classes", the market should self-regulate(17). The higher the class, the higher priority the user should get. This should make the most expensive classes less congested. This is done by having the customer willing to pay the most will have access anytime of the day,

and will be prioritized over other customers if the server is congested. The one not willing to pay for this will have to pay for a cheaper version(18). Every user can choose to either for a flat rate or a volume based usage of the server.

The amount of users the cloud simulator will get is hard to predict. The offshore and shipping industry is conservative, and are often sceptical to new technology. The schools however, will most likely be using this product instead of the original simulator due to the reduced cost. With a basis in that a server serving 12 customers would cost between \$8000 and \$10000 (see 3.3.8.1), a regular license have to cost more than \$ 700 to \$850 just to be even (and this is without any licenses for Windows or VMware). The price for access to the server should therefore lay between \$2000 and \$6000, which is little lower than what the traditional costs. These calculations will be used as a basis to find a price for the volume based customers and for the one-time fee.

The future market potential is expected to grow steadily as shipping is not gonna decline in any foreseeable future, thus the industry still need simulators to educate new crews. With access to more remote locations than earlier, the market is expected to grow with the introduction of the cloud simulator.

The other income stream will be from selling the technology and floating licenses. A big customer might buy the technology to set up their own grid/cloud server, and then buy floating licenses for every user. Here, quantity discount will be considered, and it will most likely be a one-time transaction. Any upgrades to the system must of course be purchased unless an other deal has been made.

3.3.7 Key Resources

The key resources will be the technology behind the cloud based simulator. All the revenue is based on either selling, licensing out or allowing access to the cloud simulator. Which resource the customer choose to purchase depends on the volume of intended use.

3.3.8 Cost Structure

Reduction of costs is one of, if not the main, motivations behind making a cloud based server. KM is losing out to competitors because they are too expensive, and need to find alternatives on how to be cheaper. Three areas will be the subject to cost reduction:

- **Centralized software:** By moving the entire simulator into the cloud, the cost of hardware is massively reduced. Even though KM makes a lot of money by selling hardware as a part of the simulator, it is too expensive for the low-end segment of the market. By letting the user use it's own computer lowers the entry barrier, and makes the product easier to sell.
- **No installation cost:** With a remote server, all the customer need is an access code and a VM to access the simulator, regardless of where in the world it is. This means the installation cost is reduced to 0. It is one thing to travel to Rotterdam to fix a problem, doing the same in Nigeria is a completely different price range.
- **Reduced cost of service:** By having a centralized server in one of KM's locations (or if KM decides to rent a server), the staff no longer needs to travel to whatever site the problem occur. With a service staff located where the server is, no travel expenses will apply and thus cutting that cost as well.

The cost structure is divided into two segments: CAPEX and OPEX.

3.3.8.1 CAPEX

The capital required to acquire or upgrade an asset is called capital expenditures (or CAPEX). In this case the capital expenditures will be building the physical server, and acquire license for Windows and whichever virtualization tool chosen (lets say VMware to be consistent).

Lets take a look at what it would cost to build and set up two different kind of servers: a server with GRID cards and a server with a stack of GTX GeForce cards. The server should be able to support 12 concurrent VMs. The license for Windows and VMware Horizon will be the same for each server so we can neglect that, and with Linux it would be free.

- NVIDIA GRID server: Server average cost $\$2500 + 3 * \text{GRID K2} (\$2500 \text{ each}) = \$10000$

- Server with a stack of GeForce GTX 660 cards: Dells PowerEdge R630 supports up to 13 PCI-E graphics cards(19), and costs \$2500 + 12* GTX 660 (\$480 each) = \$8260.

This means it is almost \$2000 cheaper for every 12 customers with GPU pass-through model compared to the GRID model.

3.3.8.2 OPEX

OPEX is the cost of a project's operations. For the simulator, the operations would be renting a venue where a server is placed, power and internet. In other words, small expenditures. If KM decides to rent a server from I.e AWS or Google cloud, the rent (and data rates) of the server would be the operational expenditures.

Chapter 4

Discussion

In this chapter I will discuss challenges with technical propositions made in previous chapters.

4.1 Similar Products

With the new vGPU technology on the rise, new markets business ideas pops up. As of now, there are two products on the commercial market that can be compared to the cloud simulator KM is developing;

- PlayStation Now
- NVIDIA Shield

Both are using virtualization as a method to deliver games to any sort of device with a display attached to it. NVIDIA SHIELD delivers what they like to call Gaming-as-a-Service, or GaaS. With a monthly subscription, any user can stream and play any game SHIELD has in it's database. NVIDIA SHIELD where first released as a streaming platform for TV shows and movies, but has now taken the natural step into video/computer games. This feature is called GeForce Now. It uses the GRID technology to process and deliver games in 1080p and at 60 FPS(20). NVIDIA usually releases new cards in every series every year, but they have yet to announce any new GRID cards. I believe the reason for this is because they want to protect NVIDIA SHIELD from competition. If they had released any better GRID cards than the K2, then any company could have created a similar service, and probably taken the market from NVIDIA. Just imagine if Steam or

XBOX could have gotten their hands on this technology this simple? NVIDIA is just protecting their interest, which is fair.

PlayStation NOW is a similar concept, but they are a little more protective on what kind of infrastructure they are using. All that is know, is that it is a streaming service for PS3 games to a PS4 console for a monthly subscription fee. Sony bought GAIKAI in 2012, who were a company working with streaming of high-end video games.

These products both differentiate from the classic cloud simulator. They both have their own hardware in the hands of the client. For NVIDIA, it is the hand-controller which handles all input. With this solution, they have fixed the biggest problem found in streaming the KM simulator directly in the browser (see 2.2.1); namely handling input. PlayStation can take this even further and do local rendering which will ease up the work load for the server a lot. While NVIDIA only has the game-controller, PlayStation have both a controller handling input and they have the entire console to process and handle information.

As long as NVIDIA are making money on GeForce now, I can't see them releasing a significant better GRID card because they do not want more competition in that market. KM then needs to either improve their simulator in a way that it does not require as hard processing or they must develop their own software which can emulate GPUs (I will touch more on that later).

4.2 Build or Rent Server

Whether KM should rent or build a server locally depends on a few factors from the IaaS-provider, which in this case is Amazon Web Services (AWS). AWS is the only large IaaS-provider that supports NVIDIA technology (21), which is a advantage. With the service EC2 (Elastic Compute Cloud), Amazon provides in cloud platform which can be might be interesting for KM to use. There is just one problem; EC2 NVIDIA support is only available in North America, making it unpractical for use in Europe and Asia to the least. This means that KM must build a local server, at least for the European and Asian market.

4.3 Product configuration (with NVIDIA GRID)

With the technology described earlier in this chapter, it is now time to try defining the product configuration. The first place to start is with design considerations. It is important to know what the goals of the product are, and how to get to define and create a product that satisfies this. Starting with design considerations, there are a few tasks that must be done:

- Determine Business Goals
- Target specific 3D Applications
- Size Hardware
- Select Architecture
- Acquire Hardware

Task one and two are already done. The business goals were determined in chapter 3.3, and the specific 3D application is the machine room simulator. Next step is sizing hardware. In this case, we choose to not rent a server and instead build a server that KM runs itself. NVIDIA has a list over certified GRID servers, which is a good place to start(22). With this list, it is certain that the server supports both vSphere and NVIDIA GRID. Even if the server is built in-house by purchasing every hardware part separately, this is a good reference to what hardware it should contain. The CPU should have the highest clock speed possible, then think about the number of cores. As for memory the more, the better. The VM will at least have 8 GB of memory which is dedicated and not dynamically shared. In regards of storage, IOPS (Input/output per second) is what matters and the higher, the better. At last there is the most important piece of hardware, the GRID. How many cards required depends on the amount of VMs this server will handle, but most servers can only accommodate maximum four GRID cards due to their size and need of PCI-E ports. Even with just one card, sufficient cooling is important and extra should be acquired.

Now that the hardware is in place, the next step is selecting architecture. This part is based on VMware's Horizon 6(23) The logical place to start, would be at hypervisor, which in this case is the ESXi from VMware. This will run inside vSphere where all the VMs will run. Within the

vSphere infrastructure, vCenter Server and ESXi will run. The vCenter Server is the centralized platform for managing a pool of ESXi resources. This is the part that lets the host do dynamic resource scheduling. The ESXi is the bare-metal hypervisor. With the hypervisor in place, the GRID cards are next on the list. For the VMs to get access to the GRID cards, NVIDIA VIB is installed to let every driver talk directly to the GPU.

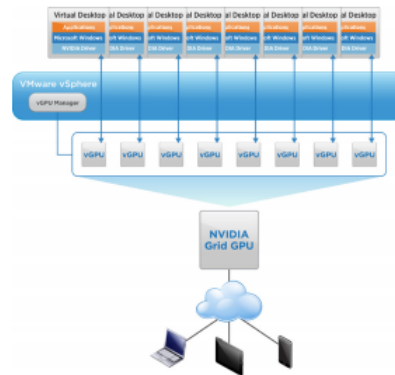


Figure 4.1: A visualization of Horizon 6 with NVIDIA GRID(4)

With the cards in place, it is time to start managing the VMs and this is done in the View Manager. The View Manager is tied to the vCenter and it allows you to create pools of end-users (VMs)(24). The View Composer is a part of the View infrastructure that allows you to create linked clones to save resources. In the View Composer a single image can be created and cloned a X amount of times. This saves storage and makes it more secure as the host knows that if the first instance is secure, then all the clones are secure.

Figure 4.2 shows a overview of what the View Composer looks like. To create different pools of users, the tool VM Pools is used. A pool is a set of VM that you entitle specific users to, based on the role and workload the user has in the given organisation or company. This lets the host not having to build a whole new set of users, instead they a clone can just be assigned to a pool.

4.4 Product configuration (without NVIDIA GRID)

By choosing to go for a server without NVIDIA GRID might not be a bad option. It is cheaper (see 3.3.8.1), and provides a better performance for every VM. The ideal solution would be to create

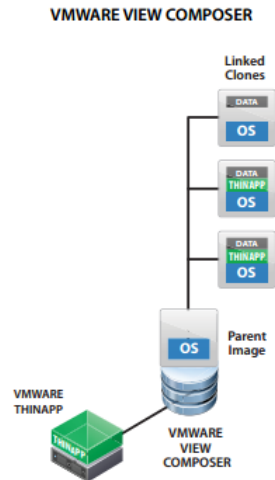


Figure 4.2: A visualization of View Composer(5)

a software that can work as an emulator in the hypervisor like the NVIDIA vGPU Manager. By emulating GPUs, each instance of the simulator will think they have dedicated a GPU for the VM, while it in fact is only emulated from a stack of Graphic cards. With this solution, scaling should be no problem since the server can just add either a better or more graphic cards to the stack. This is basically what NVIDIA GRID does today, but it does not look like they will release any improved cards soon (as mentioned earlier).

This configuration will be identical to the one in 4.3, just without the GRID technology. By using the View Composer to create instance, and VM pool to distribute each VM's place in the hierarchy based on the license they have paid for. Now, we do not have to create a separate database to handle right for each user using cloud simulator.

4.5 How will a Cloud simulator affect existing products?

By not underpricing the cloud simulator, but instead advertise it as a alternative to the traditional simulator, the simulator should not cannibalize on the other low-end simulator. Instead it should supplement it, and making the simulator more accessible for every potential user. The machine room simulator operating in the cloud is no where near as complex and good as the original, and should be seen as a substitute for it, but rather a supplement. However, in the long run , a full scale simulator in the cloud might be a possibility, but that is many years into the

future.

Chapter 5

Summary and Recommendations for Further Work

5.1 Summary and Conclusions

Kongsberg Maritime wanted to research the possibilities of running one of their lower-end simulators in a cloud-based environment. The simulator from Kongsberg Maritime is one of the world leading within the Offshore segment, so we knew the technology had to be further developed to support the heavy calculations and visualisation of the simulator. I have researched different solutions to this project, and tried to give reasoning for and against for every solution. Virtualization of different kinds and levels has is the solution that has proven to be the best. It has scalability, flexibility, the potential to reach to new customers and markets who where inaccessible due to various reasons (remote, turbulent etc.) and finally reduce the cost of each simulator, making Kongsberg Maritime more competitive in a already though market. With a cloud simulator Kongsberg Maritime is able to meet customer demands of easier accessibility, no requirement to purchase extra hardware.

When it comes to the actual technical solution, the best alternative is to build a local server (at which site Kongsberg Maritime has offices) and use the GPU pass-through technology. Even though the NVIDIA GRID solution is more developed and is more fancy, it can not provide the same performance as a 1:1 GPU - Virtual Machine. This is still scalable, and meets all the cus-

customer's demands. Scaling with the GRID might be easier, but it will be more expensive for Kongsberg Maritime in the long run. By moving from a traditional CAPEX to OPEX means lower margins, since Kongsberg Maritime no longer can bundle hardware to the simulator like they do with the traditional simulator. By the time Kongsberg Maritime will start building a cloud-based simulator, new and improved hardware might have been released with regards to NVIDIA GRID, making this the better choice for the future.

Pricing the cloud-based simulator is not easy with just one pricing model. The market is so diverse, that making a model that fits everyone is difficult. Therefore, I proposed a mix of three different pricing models to easier satisfy the customers. I believe the new simulator will supplement and not cannibalize other products based on the analysis done in this report.

References

- [1] J. Evangelho, “Gpu market share.” <http://www.forbes.com/sites/jasonevangelho/2015/08/19/nvidia-increases-desktop-gpu-market-share-again-despite-multiple-amd-radeon-7870-7890-7900-7930-7950-7970-7990-8000-8020-8030-8040-8050-8060-8070-8080-8090-8100-8110-8120-8130-8140-8150-8160-8170-8180-8190-8200-8210-8220-8230-8240-8250-8260-8270-8280-8290-8300-8310-8320-8330-8340-8350-8360-8370-8380-8390-8400-8410-8420-8430-8440-8450-8460-8470-8480-8490-8500-8510-8520-8530-8540-8550-8560-8570-8580-8590-8600-8610-8620-8630-8640-8650-8660-8670-8680-8690-8700-8710-8720-8730-8740-8750-8760-8770-8780-8790-8800-8810-8820-8830-8840-8850-8860-8870-8880-8890-8900-8910-8920-8930-8940-8950-8960-8970-8980-8990-9000-9010-9020-9030-9040-9050-9060-9070-9080-9090-9100-9110-9120-9130-9140-9150-9160-9170-9180-9190-9200-9210-9220-9230-9240-9250-9260-9270-9280-9290-9300-9310-9320-9330-9340-9350-9360-9370-9380-9390-9400-9410-9420-9430-9440-9450-9460-9470-9480-9490-9500-9510-9520-9530-9540-9550-9560-9570-9580-9590-9600-9610-9620-9630-9640-9650-9660-9670-9680-9690-9700-9710-9720-9730-9740-9750-9760-9770-9780-9790-9800-9810-9820-9830-9840-9850-9860-9870-9880-9890-9900-9910-9920-9930-9940-9950-9960-9970-9980-9990-10000>
#2715e4857a0b7887f851250e.
- [2] NVIDIA, “Nvidia grid vgpu: Delivering scalable graphics-rich virtual desktops.” <http://images.nvidia.com/content/pdf/grid/whitepaper/NVIDIA-GRID-WHITEPAPER-vGPU-Delivering-Scalable-Graphics-Rich-Virtual-Desktops.pdf>.
- [3] webRTC, “Webtrd.” <http://webrtc.org/faq/#video>.
- [4] V. NVIDIA, “Vmware horizon with nvidia grid vgpu.” <http://images.nvidia.com/content/grid/vmware/horizon-solution-brief.pdf>, 2015.
- [5] VMware, “Vmware view™ composer.” <https://www.vmware.com/files/pdf/VMware-View-4-Composer-DS-EN.pdf>.
- [6] NVIDIA, “The world’s first gpu.” <http://www.nvidia.com/page/geforce256.html>.
- [7] H. WANG, “Intel gvt-g (xengt) public release.” <https://01.org/igvt-g/blogs/wangbo85/2015/intel-gvt-g-xengt-public-release-q32015>, 2015.
- [8] S. J. Vaughan-Nichols, “Why you want a bare metal hypervisor and how to choose.” <http://www.itworld.com/article/2992463/virtualization/why-you-want-a-bare-metal-hypervisor-and-how-to-choose.html>.
- [9] X. project, “Amazon web services.” <http://www.xenproject.org/project-members/141-amazon-web-services.html>.

- [10] NVIDIA, “Virtual gpu technology.” <http://www.nvidia.com/object/virtual-gpus.html>.
- [11] NVIDIA, “Nvidia video codec sdk.” <https://developer.nvidia.com/nvidia-video-codec-sdk>.
- [12] M. Concept, “Cuda h.264/avc.” <http://www.mainconcept.com/products/sdks/gpu-acceleration-sdk/cuda-h264avc.html>.
- [13] J. Ozer, “Vp8 vs. h.264.” <http://www.streamingmedia.com/conferences/west2010/presentations/SMWest-2010-H264-VP8.pdf>.
- [14] OBS, “Open broadcast software.” <https://obsproject.com/>.
- [15] A. Osterwalder and Y. Pigneur, “Business model generation,” 2010.
- [16] Transas, “Clouds and suny in the transas techsim forecast.” <http://transas.com/clouds-and-suny-in-the-transas-techsim-forecast>, 2014.
- [17] Cisco, “Usage based pricing strategies.” <http://www.cisco.com/web/about/ac79/docs/clmw/Usage-Based-Pricing-Strategies.pdf>.
- [18] B. Tuffin and P. Maillé, *Telecommunication Network Economics From Theory yo applications*. Cambridge University Press, 2014.
- [19] Dell, “Poweredge r630 rack server.” <http://www.dell.com/us/business/p/poweredge-r630/pd>.
- [20] NVIDIA, “Geforce now.” <https://shield.nvidia.com/game-streaming-with-geforce-now>.
- [21] NVIDIA, “Gpu cloud computing, howpublished=["http://www.nvidia.com/object/gpu-cloud-computing-services.html"](http://www.nvidia.com/object/gpu-cloud-computing-services.html),”
- [22] NVIDIA, “Nvidia grid certified servers.” <http://www.nvidia.com/object/grid-certified-servers.html>.

- [23] VMware, “Vmware horizon 6 reference architecture.” <http://www.vmware.com/files/pdf/techpaper/VMware-Reference-Architecture-Horizon-6-View-Mirage-Workspace.pdf>.
- [24] VMware, “Introduction to vmware view manager.” https://www.vmware.com/pdf/viewmanager_intro.pdf.