

Master of Science in Telecommunication Systems
June 2018



Performance evaluation of Gaming Anywhere server in a virtual environment

Bandhavi Vishnumolakala

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Telecommunication Systems. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:

Author(s):

Bandhavi Vishnumolakala

E-mail: bavi16@student.bth.se

University advisor:

Dr. Yong Yao

Department of Communication Systems

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ABSTRACT

In the recent years, cloud services are dominating the internet world. Streaming video content is drastically increased. This trend created a way for the cloud gaming industry. Gaming Anywhere is one such an open source cloud gaming system designed specifically for gaming users to experience high-quality gaming while eliminating the burden of upgrading hardware or software whenever the gaming system becomes outdated. Gaming Anywhere client can stream the desired game from a remote server with high-quality resolution and fps. Gaming Anywhere launches almost all type of games as per the request of the client.

In this paper, the performance of a Gaming Anywhere server is evaluated in a virtual windows environment. Performance is categorically divided into two types. One is server metrics and the other is power consumption. Server metrics deals with CPU utilization, GPU utilization and multi-player capability. Power consumption deals with the CPU power usage of a virtual machine. The main aim of this paper is to conduct measurement studies on Gaming Anywhere server in two virtual implementation kits, VMware and VirtualBox, using monitoring tools. The analysis of the outcome is evaluated against the Gaming Anywhere physical server.

Keywords: Gaming Anywhere server, Multiplayer capability, Virtual gaming server.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Yong Yao for the constant support and encouragement in the thesis by dedicating his maximum time. I had the opportunity to learn many things from him, including his patience filled attitude.

I sincerely thank my examiner Dr. Kurt Tutschku for his support and guidance without which I couldn't have submitted my thesis.

I would like to thank the Department of communication systems for providing me an opportunity to explore a lot of things and to do research in my interesting field of study.

CONTENTS

Abstract	3
Acknowledgement	4
Contents	5
List of Figures	6
1 Introduction	7
1.1 Motivation.....	8
1.2 Aims and Objectives.....	8
1.3 Research Questions.....	8
1.4 Research Methodology.....	8
1.5 Thesis Outline.....	9
2 Background	10
2.1 Overview of Cloud gaming.....	10
2.2 Gaming Anywhere Architecture.....	10
2.3 Gaming Anywhere server.....	12
3 Related Work	14
4 Methodology	16
4.1 Overview.....	16
4.2 Performance metrics.....	16
4.3 Experimental test-bed.....	17
4.4 Experimental Procedure.....	18
5 Results and Analysis	20
5.1 CPU Utilization.....	20
5.2 Multi-player Capability.....	21
5.3 GPU Utilization.....	23
5.4 CPU power consumption.....	24
6 Conclusions and Future Work	25
References	26

LIST OF FIGURES AND TABLES

2.2.1 Architecture block diagram.....	11
2.2.2 Game server and client.....	12
4.3.1 GA server in event-driven mode.....	17
5.1.1 CPU utilization tabular form.....	21
5.1.2 Graphical representation of CPU usage.....	21
5.1.3 fps for GA server.....	21
5.2.1 one-encoder-for-all in physical server	22
5.2.2 one-encoder-for-all in VMware server	22
5.2.3 one-encoder-for-all in VirtualBox server.....	22
5.2.4 encoder-for-each-client for all servers	23
5.3.1 GPU Utilization tabular form.....	23
5.3.2 Graph for GPU Utilization.....	24
5.4.1 CPU power usage.....	24

According to '2013 State of Online gaming report' by SpilGames, 44 percent of the world's online population play computer games which is approximately 700 million users [1, Para. 2]. These figures show that the demand for gaming is huge and is eventually increasing day-by-day. Despite this tremendous interest in games, gamers often face difficulty with upgrading the hardware or software to ensure better video quality. High-end equipment like PC, gaming consoles are required to achieve better user experience. Also, due to the poor quality of graphics and resolution, users are not satisfied to play games on their mobile phones. Gaming industry can benefit a lot from these mobile gamers if there is a scope to deliver the same kind of user experience on all devices.

To overcome this problem, a new concept of online gaming has emerged in the recent years, called cloud gaming. Cloud gaming, also known as gaming-as-a-service is a unique way of incorporating the techniques of cloud computing into the online gaming industry. The cloud gaming system will continue to expand and increase further [2]. Through cloud gaming, a user can play his desired game on any mobile device, laptop, PC or gaming console without compromising on the gaming experience. The desired game can be selected from the company's remote server. Any player from around the world can play his desired game anytime and anywhere.

There are many popular cloud gaming systems available in the market like OnLive, Gaikai, Vortex and Gaming Anywhere. All the mentioned cloud gaming systems are closed [3][4][5] except Gaming Anywhere.

Gaming Anywhere is the first open source cloud gaming system [6, Sec. 7]. This system has two important components, server and a client. The server streams the desired game to end-user and the client is the device in which the user plays the game. The hardware and software installation or upgrading is done on the server side, the user is independent of this process. Also, the overall performance of gaming anywhere system is calculated, and it is found to be better than both OnLive and Gaikai [7]. Gaming Anywhere system supports many operating systems like Windows, Linux and Android.

Due to the openness of Gaming Anywhere system, it can be used by developers to carry out experiments on cloud gaming. In this paper, the idea is to study the Gaming Anywhere server in a virtual environment using virtual implementation kit and evaluate its performance. This measurement study includes CPU power consumption of Gaming Anywhere server along with the server-side metrics which affect the performance of server.

1.1 Motivation

With the emerging new technology, sophisticated techniques like cloud gaming are vastly increasing. Cloud gaming has become one of the fastest growing industrial sectors. With Microsoft launching its Vortex cloud gaming system, the potential of cloud gaming market can increase further. The cost of upgrading is reduced for the users whereas the profits for cloud gaming developers are rapidly growing. The scope of this cloud gaming market is huge. The next step in cloud gaming technology is to virtualize the cloud gaming servers to further decrease the investment revenue of developers thereby increasing the overall profits.

1.2 Aims and Objectives

The main aim of this paper is to study the behaviour of Gaming Anywhere server in a virtual environment by conducting measurement studies. This paper further investigates the impact of the type of virtual implementation kit on the performance of server. The objectives of this experiment include:

- Installing Gaming Anywhere server in a virtual environment
- The virtual implementation kits considered are VMware and VirtualBox
- The measurement studies are conducted on Gaming Anywhere server using monitoring tools
- The performance of server is evaluated in different configurations of the Gaming Anywhere system and the outcome is analysed

1.3 Research Question

1. How to measure the performance of Gaming Anywhere server in a virtual environment?
2. What is the impact of virtual implementation kit on the performance of Gaming Anywhere server?
3. What is the impact of moving Gaming Anywhere server from physical machine to virtual machine?

1.4 Research Methodology

The research methodology followed in this paper can be divided into the below five phases.

Literature study:

- Literature study focuses on the technology of cloud gaming and different cloud gaming systems. The emphasis is on the open source cloud gaming system, Gaming Anywhere. The study includes the architecture of Gaming Anywhere, internal modules of the server and the performance analysis on different cloud gaming systems.
- This study also focuses on understanding the challenges faced by previous authors in evaluating the performance of cloud gaming systems.

Experimental set-up:

- The hardware equipment used to conduct this experiment includes windows desktop with 2.6 GHz Intel Core i5 processor which acts as a server and an android mobile phone which acts as a client.
- The latest versions of VMware Workstation and VirtualBox are required.
- Any of the three server configuration game files are selected from the available games list of the server.
- The category of games selected are action-adventure, first-person shooter and real-time strategy games.

Formulating metrics:

To measure the performance of Gaming Anywhere server, it is necessary to formulate the relevant metrics depending on the internal mechanism and different modules of Gaming Anywhere server. After the literature study on the architecture of Gaming Anywhere, the formulated metrics are:

- CPU and GPU utilization
- Multi-player capability
- CPU power consumption

Performance evaluation:

The performance of Gaming Anywhere server in VMware and VirtualBox is calculated along with the performance of server in physical machine. This performance is calculated using windows monitoring tools.

Results:

The outcome of this experiment is analysed depending on the game genre and the environment in which Gaming Anywhere server is launched. The conclusions are drawn based on the comparisons of results from the above-mentioned scenarios.

1.5 Thesis Outline

Chapter 1: Introduction describes the concepts of cloud gaming and Gaming Anywhere in detail.

Chapter 2: Background section is the brief study of Gaming Anywhere server to understand its internal modules and different configuration settings.

Chapter 3: Related work describes the common methods that are used to evaluate the performance of cloud gaming systems.

Chapter 4: Results and Analysis gives a detailed overview of the performance of GA server in VMware, VirtualBox and physical machine.

Chapter 5: This section concludes the paper by summarizing our work and suggesting further work to be carried in virtualizing the Gaming Anywhere server.

2.1 Overview of Cloud gaming

Before cloud gaming, computer games are accessed by the end-users using two different methods. One is to directly install the desired game onto the PC and the other is to use gaming consoles. Installing a game on PC takes much time and the PC should support latest gaming configurations. This creates problems for the users like upgrading hardware or software which sometimes include changing the entire console if a single component is outdated.

Cloud gaming overcomes all the problems of gaming consoles but most of the cloud gaming systems are not suitable for gaming developers as they are all closed. Gaming Anywhere is the first open source cloud gaming platform which enables the developers to extend their research.

2.2 Gaming Anywhere Architecture

To work on the Gaming Anywhere server, it is necessary to study the basic features and the internal mechanism of Gaming Anywhere system. It consists of three important components:

- Game server
- Portal server
- Game client

Before understanding the way in which this cloud gaming system works, it is important to understand the nature of each component. Portal server is like a web-based server. This deals with user authentication of the client device and acts as a communication gateway between the server and the client until a connection is established between the both. The server manages all the server configuration files of available games. They are stored on the server and can be accessed by the client upon request. The client can be any device like PC, mobile phone or laptop which is used to play games at the end-user level. The steps involved in using the Gaming Anywhere system is as follows:

- The first step to play the game, using this system, is to log into the portal server.
- Now, the user is on the portal server and can select his desired game from the list.
- Gaming Anywhere is compatible with almost all the games and it is also compatible with Windows, Linux and Android operating systems.
- As the game selection is done by the user, the same game will be launched on the available GA server by the portal server.

- After the game is launched on the GA server, the user will have access to the game.
- The game server will share a URL with the user and any observer can watch the game on the game server with the help of this URL. This URL can be accessed from any device throughout the globe.

This procedure is shown in the block diagram given below.

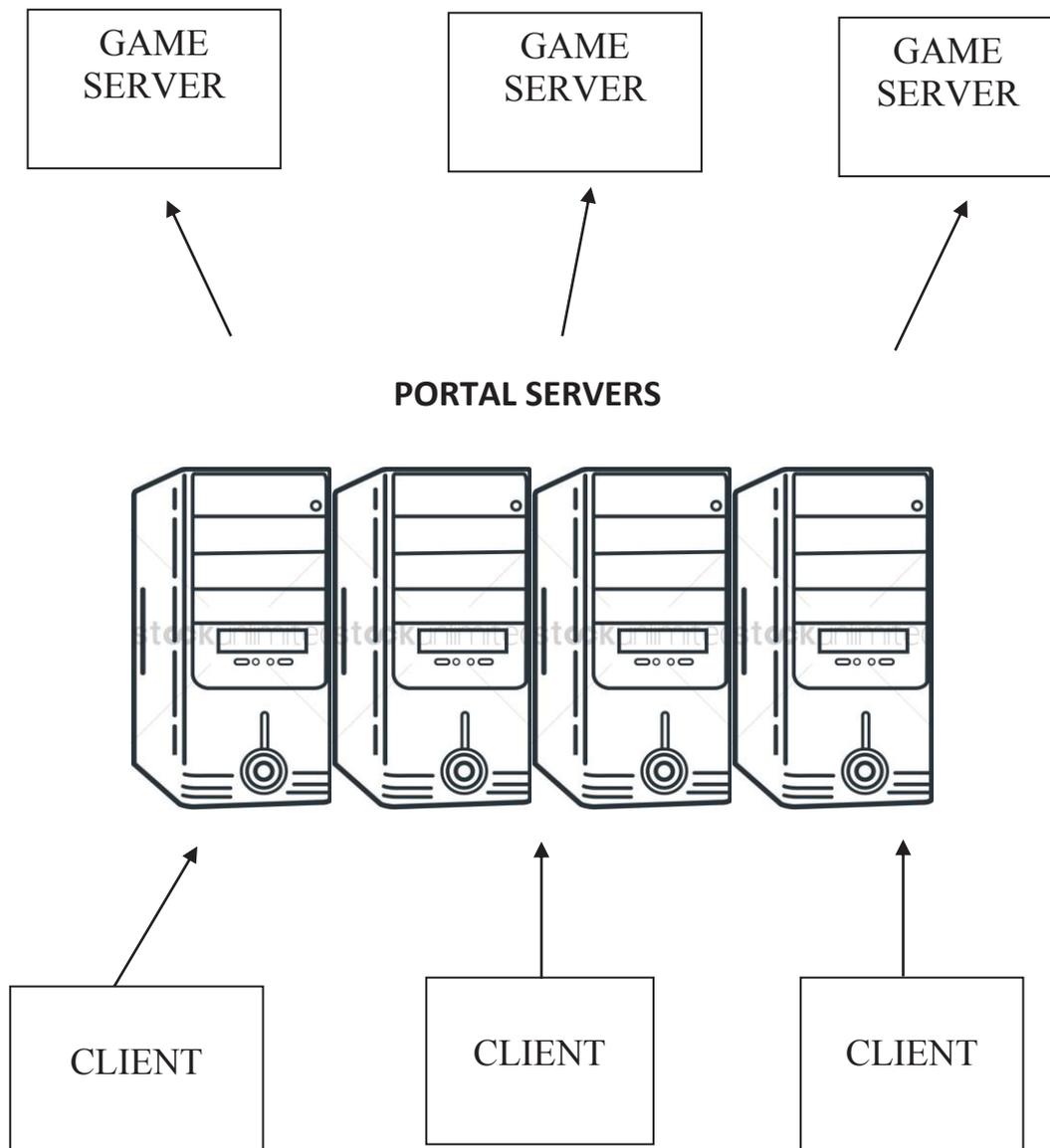


Figure 2.2.1: Block diagram of the basic components of Gaming Anywhere and the interaction between server and client.

The GA server and GA client interact with each other over the internet with the help of two types of network flows, the data flow and the control flow. The frames generated by any frame source is encoded by the server and is sent to the Game client via the data flow and the frames that are generated by the client like mouse

cursor movements or keyboard inputs are received and decoded by the game server via the control flow. The below figure [8] shows the internal architecture and different modules of Gaming Anywhere.

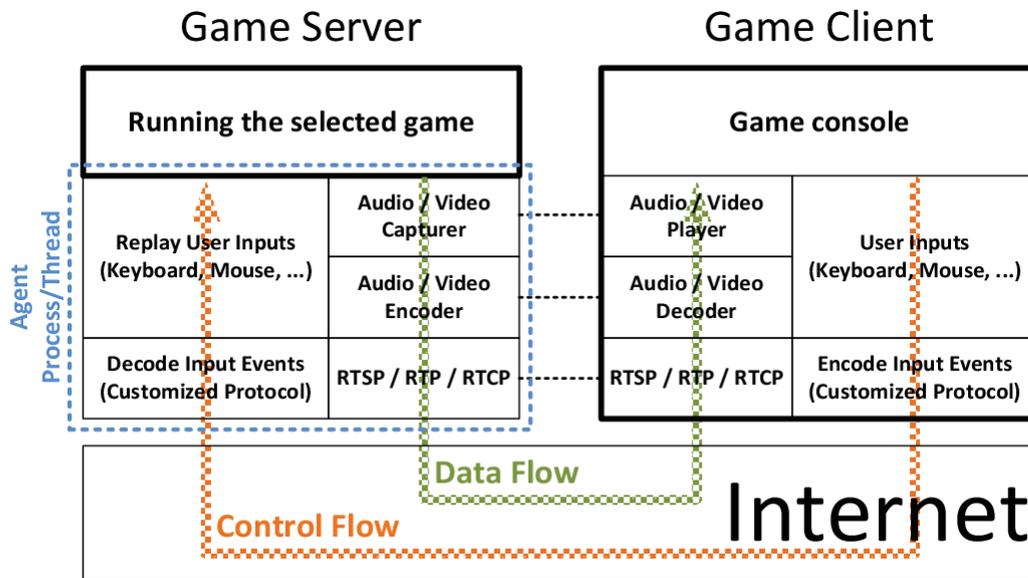


Figure 2.2.2: Game server and Game client modules

This architecture makes clear that the performance of GA system is highly dependent on the type of the game.

2.3 Gaming Anywhere server

The work of GA server is to stream audio and video frames to the client side. This is achieved by an agent process or a thread that is injected into the selected game and it captures the audio/video frames, encodes them and delivers them to the client side using the data flow [9, Sec. 5.1].

GA server has five modules and they are launched as soon as the agent process is launched. The five modules are:

- RTSP server
- Video source
- Audio source
- Frame encoding
- Input handling.

RTSP server: RTSP server is the first module to launch among the five other modules because it waits for the client to establish a connection and listens for it. When a connection is established, it sends encoded frames to the client side.

After the encoded video and audio frames are streamed via the data flow, the user inputs need to be decoded at the GA server side. To perform this action, a protocol

which mimics the behaviour of user inputs is needed but such protocol is not available. RTSP/RTP protocol is designed and implemented to overcome this problem.

Frame Encoding: The generated video and audio frames are to be encoded before delivering to the client side. The data flow carries only encoded frames. There are two types of methods to encode frames, one-encoder-for-all and encoder-for-each-client. All the video frames are encoded by a single encoder and all audio frames by another single encoder but the difference in two encoding methods has to do with the number of clients or observers.

If all the frames are encoded once and then delivered to the client, the GA system is operating in one-encoder-for-all mode. If each client or observer has a separate encoder, then the configuration mode is in encoder-for-each-client. Performance of a system is highly dependent on this module. In the first type, CPU usage is the same regardless of the increasing number of clients, but efficiency will be higher in this case whereas in the later, the desired resolution and encoder combinations can be changed which impacts the CPU utilization.

Video source: Video source is an important module for the GA server. It captures the game screen, so it determines the video quality thereby affecting the user experience. However, the game screen capture is platform-dependent. There are two different ways of capturing the game screen. One is desktop capture module and the other is API intercept module. Desktop capture module is the method of capturing the game scenes at a specific interval of time and the captured time interval depends on the desired frame rate.

API intercept module works based on the events. If the recently updated screen is rendered, then the module will capture the screen for streaming. Again, the screen capture depends on the desired frame rate per second.

Based on these two methods of screen capturing, the GA server can be launched in two different modes. The command for GA server in desktop capture mode is `ga-server-periodic server-configuration-file.conf` and the command to launch server in API intercept mode is `ga-server-event-driven server-configuration-file.conf`

Audio source: Audio device and audio buffer are two components of this audio source. The output of the audio device is periodically captured by the audio source module. This capturing method imposes a problem known as frame discontinuity problem. When no audio signal is being received, the audio source should render a frame with all zero's or it should indicate that no frames are available, but this is hard to achieve mainly because audio buffers expect frames periodically. Gaming anywhere has resolved this issue by using their Windows audio session API (WASAPI) enabled audio source.

Cloud gaming is a relatively new technique and the research work in this area started booming only around 2013. According to the literature survey, there are only a few research papers that deal with this topic of cloud gaming especially Gaming Anywhere. The research on virtualization of gaming anywhere will be a new milestone for cloud gaming.

Authors in [10] have considered three different games while comparing the performance of Gaming anywhere system with OnLive and Gaikai. The three different games they used can be categorically divided into three types. They are first-person shooter, action-adventure and real-time strategy. According to the author's estimation, the type of games indeed shows a clear impact on the performance.

Authors in [11] evaluated the different types of game genres and their impact on the user experience. They also divided the games into four different types including racing, puzzle, strategy and action games. This shows that the different game genres will impact the performance of cloud gaming system thereby impacting the user experience at the client-side.

Authors in [12] calculated processing delay of the physical server by implementing the same method of Measuring latency. Authors of the later defined processing delay as the time difference between the frame sent from the client and the frame received by the client with network delay. The network delay is independent of the cloud gaming system and it is not controlled by any cloud gaming systems so, in the case of evaluating the performance of a server, network delay becomes invalid.

Authors in [13] calculated processing delay of the server using the response delay and the network delay. They defined response delay as the sum of processing delay, playout delay and network delay. Processing delay is generated by the server and the playout delay is due to the client. The amount of time difference between the frame sent by the client and the frame received by the client along with the network delay is defined as processing delay. As the frame sent by the client is processed by the server and as this is an internal process, it is difficult to derive the time. To handle this issue, the authors have used Gaussian probability approach where the approximate time range is estimated and is calculated with the help of 95% confidence intervals.

Authors in [14, Sec. 2] carried out measurement studies on the performance of cloud gaming server in virtual machines but they have not considered multi-player capability of the virtual server which in combination with different configurations of frame encoding module can highly affect the performance of a virtual game server. The multi-player capability and the server configuration settings can be discussed in

chapter 4. To the best of our knowledge, this paper is the first one to evaluate measurement studies by including not only the multi-player capability scenario but also considering the power performance of the server. This paper analyses the behaviour of GA server in different configuration modes of the server internal modules in addition to the power usage at the server-side.

This chapter discusses the procedure for formulating and evaluating the performance metrics of GA virtual server. It describes the required hardware set-up, step-by-step procedure of carrying out the experiment and finally, different tools used to calculate performance metrics.

4.1 Overview

This experiment is carried out using the latest source binaries of Gaming Anywhere in a virtual windows environment. The host machine is windows desktops with 2.6 GHz Intel Core i5 processors with integrated GPU. Gaming Anywhere server can be run using the server configuration files from installed source binaries.

As the performance of Gaming Anywhere is dependent on the type of the game, three different games are selected from the list of available configuration files. In this paper, three games are selected. They are:

- Assaultcube
- OpenTTD
- Armagetron Advanced

Assaultcube is a first-person shooter game, OpenTTD is a strategy game and Armagetron Advanced is an adventure game.

Although the focus is on the server alone, it is necessary to download both the server and client configuration files for the three games. This is because multi-player capability can only be decided if there is at least one client. The number of observers can be varied to find the multi-player capability.

4.2 Performance metrics

The main aim of this experiment is to calculate the performance of server alone, so the network parameters and client-side metrics are neglected because the focus is on the server side. The metrics considered are:

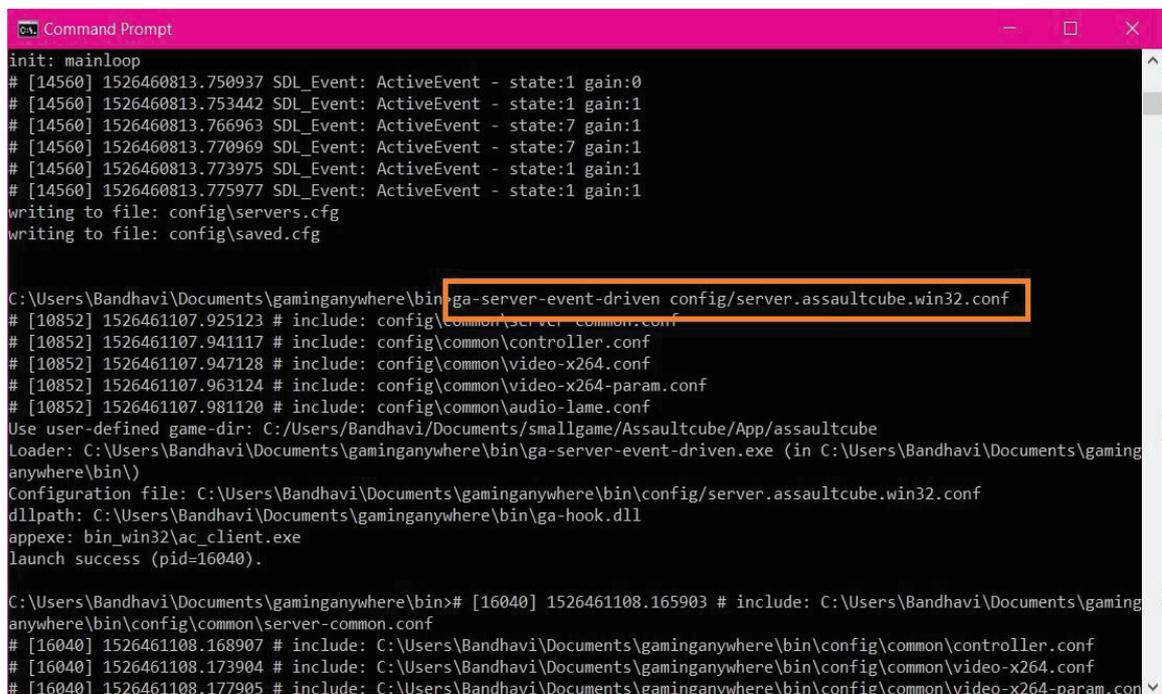
- i. CPU utilization: The average CPU usage of GA server
- ii. Multi-player capability: The number of clients or observers who can simultaneously access the virtual server in two different configurations of frame encoding, one-encoder-for-all and one-encoder-for-each-client
- iii. GPU utilization: The average GPU usage of GA server
- iv. CPU power usage: The power consumed by the CPU in launching the game on the GA server

When the client is connected to the virtual GA server, observations are taken during a 5-minute session initiated after the server is launched. This is true whenever a new observer is added to the network through server URL.

In addition to the above-mentioned metrics, frame rate per second (fps) is also observed and any changes in the fps are noted down.

4.3 Experimental test-bed

This section gives a detailed description of the tools and components used in this experiment. The experiment requires a single GA server in a virtual machine. As the experiment is conducted in both VMware and VirtualBox, two host machines with VMware and VirtualBox are required and the virtual implementation kits are dumped into host machines respectively. GA server is launched in VMware and VirtualBox to calculate the performance metrics.



```
init: mainloop
# [14560] 1526460813.750937 SDL_Event: ActiveEvent - state:1 gain:0
# [14560] 1526460813.753442 SDL_Event: ActiveEvent - state:1 gain:1
# [14560] 1526460813.766963 SDL_Event: ActiveEvent - state:7 gain:1
# [14560] 1526460813.770969 SDL_Event: ActiveEvent - state:7 gain:1
# [14560] 1526460813.773975 SDL_Event: ActiveEvent - state:1 gain:1
# [14560] 1526460813.775977 SDL_Event: ActiveEvent - state:1 gain:1
writing to file: config\servers.cfg
writing to file: config\saved.cfg

C:\Users\Bandhavi\Documents\gaminganywhere\bin\ga-server-event-driven config/server.assaultcube.win32.conf
# [10852] 1526461107.925123 # include: config\common\server-common.conf
# [10852] 1526461107.941117 # include: config\common\controller.conf
# [10852] 1526461107.947128 # include: config\common\video-x264.conf
# [10852] 1526461107.963124 # include: config\common\video-x264-param.conf
# [10852] 1526461107.981120 # include: config\common\audio-lame.conf
Use user-defined game-dir: C:/Users/Bandhavi/Documents/smallgame/Assaultcube/App/assaultcube
Loader: C:\Users\Bandhavi\Documents\gaminganywhere\bin\ga-server-event-driven.exe (in C:\Users\Bandhavi\Documents\gaminganywhere\bin\)
Configuration file: C:\Users\Bandhavi\Documents\gaminganywhere\bin\config/server.assaultcube.win32.conf
dllpath: C:\Users\Bandhavi\Documents\gaminganywhere\bin\ga-hook.dll
appexe: bin_win32\ac_client.exe
launch success (pid=16040).

C:\Users\Bandhavi\Documents\gaminganywhere\bin># [16040] 1526461108.165903 # include: C:\Users\Bandhavi\Documents\gaminganywhere\bin\config\common\server-common.conf
# [16040] 1526461108.168907 # include: C:\Users\Bandhavi\Documents\gaminganywhere\bin\config\common\controller.conf
# [16040] 1526461108.173904 # include: C:\Users\Bandhavi\Documents\gaminganywhere\bin\config\common\video-x264.conf
# [16040] 1526461108.177905 # include: C:\Users\Bandhavi\Documents\gaminganywhere\bin\config\common\video-x264-param.conf
```

Figure 4.3.1: GA server running in event-driven mode

Performance metrics can be divided into two types. They are server metrics and power performance. AIDA64 Extreme, a performance monitoring tool is used to calculate CPU utilization whereas GPU-z is used to calculate the GPU utilization. Joulemeter, a power monitoring tool is used to calculate CPU power usage of the server. Server metrics like CPU utilization and GPU utilization are calculated using AIDA64 extreme and GPU-z respectively but not multi-player capability because multi-player capability is highly dependent on the internal mechanism of the Gaming Anywhere server and cannot be calculated using external tools. Any change

in the two different modes of frame encoding module when connected with a client can affect the multi-player capability thereby affecting the CPU utilization.

An android mobile device is connected to the server which acts as a client and any PC, laptop or mobile device is connected to the server through the server URL. The devices which access the server URL are called observers.

4.4 Experimental Procedure

The experiment is carried on the single server with all available resources in a virtual environment. The virtual implementations considered are VMware and VirtualBox. The source binaries of Gaming Anywhere are installed both in VMware and VirtualBox. To launch the GA server, the server configuration files for the games are to be changed according to the settings of the game.

The two possible options for running Gaming Anywhere server are periodic mode and event-driven mode. These options are dependent on the video source module of GA server. This module is platform-dependent. This paper concentrates on the virtual machine more than the underlying operating system, so either of the two options works fine if the operating system and the option selected are consistent throughout the experiment. In this experiment, the GA virtual server is operating in the event-driven mode. The desired fps of video source module is set to 50 fps.

For each of the three different games, the configuration settings are same. Now, the GA server is launched in VMware and VirtualBox with the desired game server configuration file in the event-driven mode. CPU utilization is calculated in both the virtual machines and the performance is compared against the physical server. This means that, in addition to calculating CPU utilization in both the virtual machines, it is also calculated in the windows physical GA server. The values of CPU utilization for the 5-minute session when the server is running are averaged.

To use a configuration mode in frame encoding module, the server should be connected to at least a client, so it is necessary to set-up a GA client in Android mobile phone. The game is launched in a mobile phone and a 5-minute gameplay session is initiated. Note that the virtual machine is being used only for the server but not for the client and observers. One-encoder-for-all mode is enabled, and the encoder used in this experiment is x264. CPU utilization and fps for a single client are observed. After the observations are recorded, the session is terminated, and at this point a single observer is added to the network along with a mobile client through the URL provided by the game server. The CPU utilization and fps for this newly added observer along with the client is recorded. This process repeats for a minimum of 3 observers along with the mobile client and the changes are observed.

In the encoder-for-each-client mode, CPU utilization and fps are recorded in a similar way but in this mode, new observers are added to the network and this process repeats until the server cannot handle any more requests from new devices. The total number of observers in addition to a single client in the network

until the server reaches its maximum capacity are recorded. This shows the multi-player capability of virtual GA server. Again, all the observations are calculated in both VMware GA server and VirtualBox GA server. This frame encoding settings are implemented without considering the different game genres because, in a real-time environment, game clients will select their desired games, so the different type of games were neglected. In addition to the above-mentioned performance metrics, fps is calculated and the game which has lowest fps is selected.

The mobile client playing the game and the number of observers watching the game from the server's URL are disconnected. The GA server is launched in both VMware and VirtualBox, GPU values are observed for the 5-minute session when server is running in the event-driven mode are recorded. All the values are averaged, and GPU utilization is calculated. The tool used for calculating GPU is GPU-z. GPU utilization is calculated in the same way for all the three games in both VMware and VirtualBox.

The CPU power usage of the GA virtual server can be calculated by using Joulemeter, a power monitoring tool. The values obtained from this observation are averaged for 5-minute session after the required game is launched by the server.

This chapter describes the outcome of the monitoring tools which are used to calculate server metrics and power metrics. The average values obtained from these tools are calculated for three different games as mentioned earlier. The values of each metric depend on the type of the game. The results were taken when the server is running for an approximate period of 5-minute sessions for each metric. These results depict the overall performance of GA virtual server against the physical machine.

5.1 CPU utilization

In this section, the CPU utilization values obtained from AIDA64 extreme monitoring tool for GA server in VMware and VirtualBox are recorded. The below percentages of CPU utilization were obtained for the average values in every 5-minute session.

The CPU utilization for the physical machine for all the three games is lower than the virtual machine. When the server is launched using Assaultcube configuration file in the GA physical server, the CPU usage is at its minimum compared to other games and maximum for OpenTTD.

Games	CPU usage (%) in Physical Machine	CPU usage (%) in VMware	CPU usage (%) in VirtualBox
Assault Cube	53	66	89
OpenTTD	71	98	99
Armagetron Advanced	58	67	98

Figure 5.1.1: CPU utilization of GA server

The graphical representation of figure 5.1.1 shows the impact of game genre on the performance of GA server in both the physical machine and the virtual machine. By observations drawn from the graph, it is evident that:

- Different virtual implementations result in different CPU utilization
- The performance of GA server in virtual machine also depends on the type of the game
- Armagetron Advanced is a basic game, VirtualBox has the highest CPU utilization compared to VMware

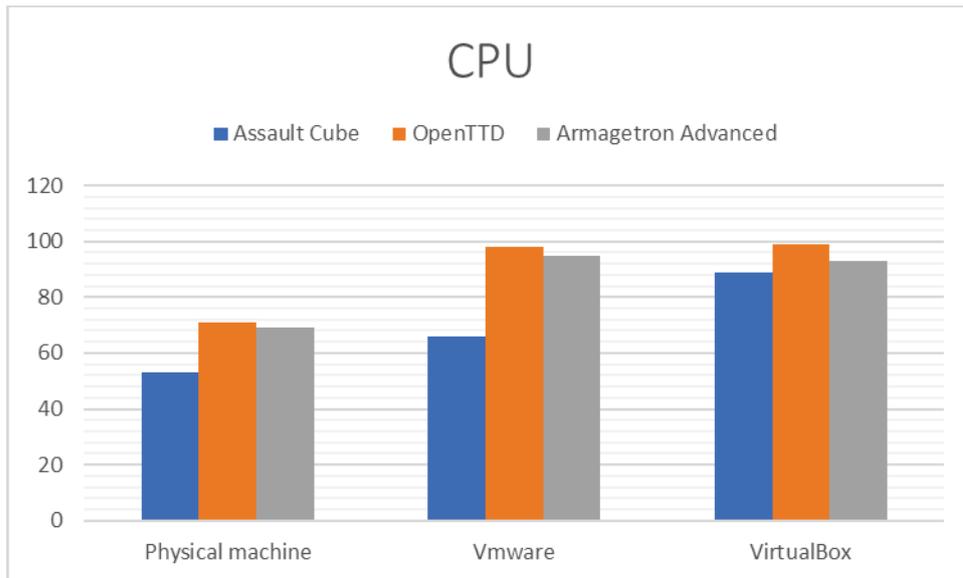


Figure 5.1.2: Graphical representation of CPU utilization

As mentioned earlier, in addition to this CPU utilization, fps is also observed for the three games. In case of OpenTTD, a simulation-based strategy game, the fps is lower compared to the other two games in both the physical and virtual machine. For the physical machine, the fps is high than the virtual machine. The fps for physical machine for all the three games is more than 40 fps at any given point of the 5-minute session whereas in case of VMware and VirtualBox the fps is largely varied and is always less than the physical machine.

Games	Fps in Physical Machine	Fps in VMware	Fps in VirtualBox
Assault Cube	53	39	24
OpenTTD	41	24	24
Armagetron Advanced	46	33	32

Figure 5.1.3: fps values for GA server

From the above table, the observations drawn can be as follows:

- The more the CPU usage is for a game, the lesser the frame rate per second is achieved
- This is true in all the cases of Gaming Anywhere server

5.2 Multi-player capability

First, the server is launched in one-encoder-for-all mode and then the mode is switched to encoder-for-each-client. When the server is running in both the modes, the OpenTTD game is considered. In one-encoder-for-all mode, the CPU utilization

is stable for all the observers/client in physical machine for every single game even as the number of observers are increased.

Number of Observers	CPU utilization	Frame rate per second
1	71	41
2	71	40
3	74	38
4	75	35

Figure 5.2.1: CPU utilization and fps for one-encoder-for-all mode in Physical GA server

The CPU utilization in a specific virtual machine is almost the same value obtained when a single server is launched on that virtual machine without any clients. The reason behind calculating the fps along with CPU utilization in this encoder-for-all mode is to evaluate the sever tolerance when a high load is generated on the system.

The below figure shows the CPU utilization and fps for one-encoder-for-all mode in VMware and VirtualBox respectively.

Number of Observers	CPU utilization	Frame rate per second
1	98	24
2	98	24
3	98	24
4	98	20

Figure 5.2.2: CPU utilization and fps for one-encoder-for-all mode in VMware GA server

Number of Observers	CPU utilization	Frame rate per second
1	98	24
2	99	22
3	99	20
4	100	20

Figure 5.2.3: CPU utilization and fps for one-encoder-for-all mode in Virtual Box GA server

In the encoder-for-each-client mode, there is an increase in the CPU utilization in both virtual server and physical server. Because of this increase, the fps keeps on decreasing for every new observer added.

When 2 observers are added to the server along with the single mobile client, the CPU utilization in VirtualBox reached its maximum and it couldn't handle any new

observer, but the physical machine and the VMware can handle a few observers, so more observers are added to these servers until they reached the maximum CPU utilization level. When a 3rd observer is added to the VirtualBox GA server, the recorded fps is 20. The physical GA server tolerated 8 observers along with a mobile client which makes it 9 and the VMware tolerated 3 observers along with the mobile client. The resolution for all the observers and clients in both physical GA server and the virtual GA server is 1280X720.

Number of Observers	Gaming Anywhere server
9	Physical machine
4	VMware
3	VirtualBox

Figure 5.2.4: Number of observers in three GA servers

5.3 GPU utilization

The tool used to calculate GPU utilization is GPU-z. For the three different games, the GPU utilization is calculated when the server is running for a 5-minute session. The average value obtained in each GA server, both physical and virtual is recorded. The below table shows the GPU utilization in GA servers for all the three different games.

Games	GPU usage (%) in Physical Machine	GPU usage (%) in VMware	GPU usage (%) in VirtualBox
Assault Cube	76	82	88
OpenTTD	69	94	91
Armagetron Advanced	54	92	95

Figure 5.3.1: GPU utilization in percentages for all GA servers

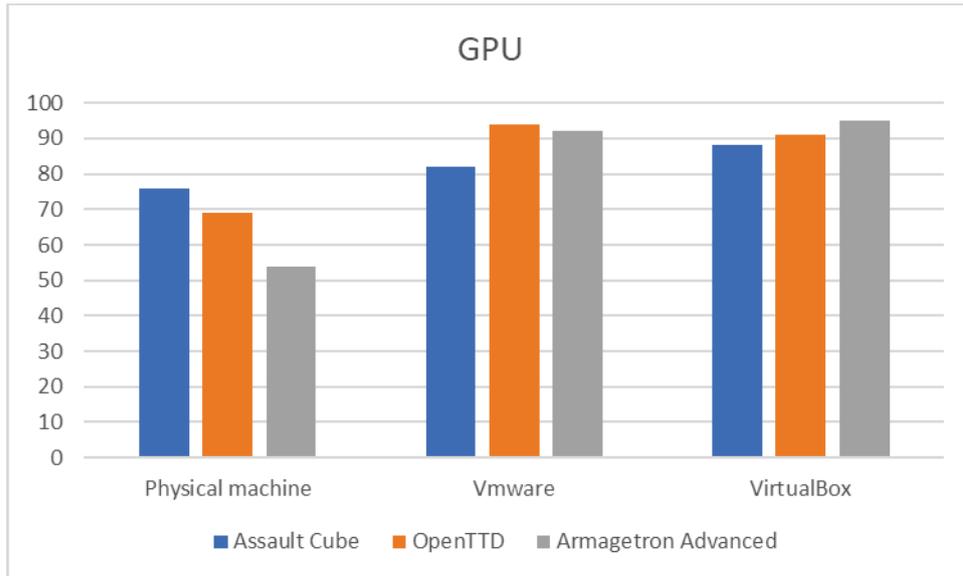


Figure 5.3.2: Graphical representation of GPU utilization

The graph depicts the performance of GPU for three different games in GA physical and virtual servers.

5.4 CPU power consumption

The CPU power usage is calculated using the Joulemeter monitoring tool. The average value over a 5-minute session after the GA server is launched in both the physical machine and virtual machine is recorded.

Games	CPU power usage in physical machine	CPU power usage in VMware	CPU power in VirtualBox
Assault Cube	11.8 W	12.2 W	15.8 W
OpenTTD	12.4 W	11.6 W	11.4 W
Armagetron Advanced	10.9 W	11.2 W	10.3 W

Figure 5.4.1: CPU power consumption for all three GA servers

The observations from the above power performance calculations are as follows:

- OpenTTD, a simulation adventure game requires more power than the rest of the games
- Power performance of server is also dependent on the type of the game
- Physical machine requires comparatively less power than the virtual machine
- Power performance also varies on the type of virtual implementation kit used

By the outcome of this experiment and detailed analysis, the observations for the server metrics in GA server is that the performance depends on the virtual implementation kit used. From this experiment, it is evident that GA server in VMware can handle 4 observers but VirtualBox can handle only 3 observers. Also, the CPU/GPU utilization is lower for the VMware than for the VirtualBox. Although the power performance of both the virtual machines is varied, we can clearly see that adventure game consumed more power than the other two games. This shows that power performance is also dependent on the type of virtual implementation kit.

From the above analysis, the conclusion is that the performance of server metrics and power performance depends on the type of virtual implementation kit.

Therefore, we suggest that by developing efficient scheduling algorithms for a specific virtual machine will improve its performance because the power consumption in virtual machines can be regulated with the help of energy-aware scheduling algorithms to be used in the virtualized cloud gaming system. This virtualization of cloud gaming systems based on the energy-aware scheduling algorithms will decrease the investment revenue for the gaming developers.

REFERENCES

- [1] "Study: 1.2 billion people are playing games worldwide; 700M of them are online," *GeekWire*, 25-Nov-2013. [Online]. Available: <https://www.geekwire.com/2013/gaming-report-12-billion-people-playing-games-worldwide/>. [Accessed: 17-May-2018].
- [2] W. Cai *et al.*, "A Survey on Cloud Gaming: Future of Computer Games," *IEEE Access*, vol. 4, pp. 7605–7620, 2016.
- [3] R. Shea, J. Liu, E. C. H. Ngai, and Y. Cui, "Cloud gaming: architecture and performance," *IEEE Netw.*, vol. 27, no. 4, pp. 16–21, Jul. 2013
- [4] S. Ungureanu, A. Panu, and L. Alboaie, "A Cloud Gaming system design insights," in *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, 2015, vol. 02, pp. 144–151.
- [5] "Vortex - Cloud Gaming for PC and Android." [Online]. Available: vortex.gg/. [Accessed: 17-May-2018].
- [6] "GamingAnywhere - An Open Source Cloud Gaming System." [Online]. Available: <http://gaminganywhere.org/index.html>. [Accessed: 17-May-2018].
- [7] "GamingAnywhere - Performance." [Online]. Available: <http://gaminganywhere.org/perf.html>. [Accessed: 17-May-2018].
- [8] "GamingAnywhere: An Open Cloud Gaming System." [Online]. Available: http://gaminganywhere.org/gaming_anywhere.html. [Accessed: 17-May-2018].
- [9] "GamingAnywhere: An Open-Source Cloud Gaming Testbed." [Online]. Available: http://www.iis.sinica.edu.tw/~swc/pub/gaming_anywhere_open_source.html. [Accessed: 17-May-2018].
- [10] "GamingAnywhere - Performance." [Online]. Available: <http://gaminganywhere.org/perf.html>. [Accessed: 17-May-2018].
- [11] P. Quax, A. Beznosyk, W. Vanmontfort, R. Marx, and W. Lamotte, "An evaluation of the impact of game genre on user experience in cloud gaming," in *2013 IEEE International Games Innovation Conference (IGIC)*, 2013, pp. 216–221.
- [12] "GamingAnywhere: An Open-Source Cloud Gaming Testbed." [Online]. Available: http://www.iis.sinica.edu.tw/~swc/pub/gaming_anywhere_open_source.html. [Accessed: 17-May-2018].
- [13] K.-T. Chen, Y.-C. Chang, P.-H. Tseng, C.-Y. Huang, and C.-L. Lei, "Measuring the Latency of Cloud Gaming Systems," in *Proceedings of the 19th ACM International Conference on Multimedia*, New York, NY, USA, 2011, pp. 1269–1272.
- [14] H. J. Hong, D. Y. Chen, C. Y. Huang, K. T. Chen, and C. H. Hsu, "Placing Virtual Machines to Optimize Cloud Gaming Experience," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 42–53, Jan. 2015.