Shifting Game Processing and Simulation of Massively Multi Player Online Games onto Compound Systems Connected over a Network

Ramachaitanya Kavuluru
University of Southern California
Department of Computer Science
kavuluru@usc.edu

Abstract

This paper will initially explore the current trends and problems of sustaining Massively Multi Player Online Games with the modern day technology and assess the best optimization model for running these games from a distributed systems and cloud computing stand point.

After achieving this most optimized model, the paper will explore the hardware requirements to buttress this architecture.

Once a thorough analysis about the hardware requirements is made, an inspection about the hardware capabilities on the existing standalone devices like Iphones, Androids and HTC devices will be studied specifically.

The paper would then emphasize on bridging the gap, between these incapabilities by enabling these relatively low computation intensive devices to acquire some computation power from the other devices connected on the network.

1. Introduction

Today there are several highly successful Massively Multiplayer Online Games (MMOGs) and high speed network communication is burgeoning. Shifting game processing, simulation and emulation in these killer applications between machines which are on a single network will provide the ability for devices with reduced processing power to play the game that requires higher processors and memory requirements. This paper will initially build a simple problem scenario and explore the difficulties of such a system with respect to standard architectures, assumption of shared designs and geographic network latencies [1].

There are studies about the impact of Virtualization on the Performance of MMOGS, where virtualization is a technology that alleviates the problem of porting a MMOG software services to different platforms. A notable contribution to this belief would be the influence of virtualization [2].

Along the stand point of distributed systems, Shinya and Murata et al [3] propose a distributed event delivery method for MMORPG. In this method, the whole game space is divided into multiple sub spaces with same size and some player nodes are selected as responsible nodes to deliver game events occurring in their responsible sub spaces. Furthermore, cluster servers by Fengyun et al [4] talk about servers which allow great flexibility for routine maintenance or upgrading of servers involved in the system. This system presents an approach to load balancing. Finally, a system [5] which deals with offloading the AI computation from game servers to game clients is used to further optimize the computation power in a decentralized fashion.

In networking community, a notable system [6] which exhibits the benefits of using proxy architecture for multiplexing games streams with TCP for an MMOG is used to optimize the server load, allow congestion control and reduce lag of the game. Furthermore, the relation between game playing time, network QOS and user satisfaction is well known [7]. The problem the author intends to address is the type of networks and the upper limit of network latency which will provide users to experience best QOS. The author will consider the problem of the hardware incapacity to simulate/emulate the current system and if need be the possibility of future hardware which could make the system possible.

This paper will examine the issues raised by protocols involved in preserving the state in computers communicating over networks [8] and will look at the possibility of integrating the ideas and concepts made use in that system. Some system like cameo [9]; have the ability to support the social networks built around MMOGs. The author intends to provide an overview of this system and its ability to blend with social networks through continuous analytics and cloud computing.

Ultimately the problem deals with exploring the possibility of playing computation intensive MMOGS on standalone instruments in the current day which are incapable of supporting the required processing power.

A thorough analysis of the current day hardware and the hardware required by the system will be made. If both are found compatible, the problem would be solved; else the hardware scalability will be explored for which the system will function.

The constraints on the solution would be the ability of the system to enable the stand alone instrument to process and recognize the content forwarded to it by the compound network. Adding to the constraints would be the ability of the system to effectively deal with latency.

2. Prolog: The Present Day Systems

Massively Multiplayer Online Games are a genre of Computer Games in which a very large number of players interact with each other within a game world.

Games like World of Warcraft, Starcraft, Diablo, Ragnarok, and World War II online; constitute this category. (The paper is not concerned with browser based MMOGs which require minimal computation on the machine to run and simulate; like Farmville or Travian.)

2.1 Nuts and Bolts:

a. Security and Management Trade Offs [3]

Some of the MMOGs have a client server system where the global game state and all the states of players in the game are managed in a centralized way. In these systems maintaining security is very convenient but managing and coping with the growing demand becomes a problem.

b. Non Standard Architectures [1]

There is no consensus among game designers to build games for a single platform. Thus each game and every succeeding version of the game has and requires different dedicated software and hardware architectures.

c. Content Production, Workload Distribution and Server over Provisioning Phenomenon [1]

"Sharded Architecture" is a term that refers to; maintaining consistent game content over multiple views and worlds, which has substituted graphic game content like scenes, creatures and scenarios for huge number of players. This resolves the scalability issues but increases the server over provisioning phenomenon since as long as one player remains in a realm, the machines serving that realm cannot retire.

d. Multiple Games Hosted on Multiple Data Centers [1]

With the growth in the number of Massively Multiplayer Online Games, there is a constant need for a data center to support multiple games; as an example the Garena Game Server [15] supports games like Warcraft III, Counter Strike, and Age of Empires etc. This decision further alleviates the need for optimizing and consolidating to an architecture that is not dependent on the respective game hardware and software.

e. Deployment of Data Centers in Geographically Distributed Locations [1]

To protect gamers from excessive network latencies, there must be data centers for the relevant game distributed in multiple geographic locations so that the service is rendered from the most physically close location. This leads to excessive energy and hardware consumption

3. The Vanguard

During the last five years there have been developments which account for the issues that are listed in the previous section.

a. Spatial Locality Property [1]

Server consolidation based on spatial locality property, where a game world is partitioned into a certain continuous region called “zones”, serving each zone with a separate Virtual Machine. This strategy reallocates zones among a set of server clusters regularly and evaluate its impact in terms of the number of servers required and energy.

Each zone is a perfect unit for workload dispatching based on virtualization.

b. The Peer-2-Peer System (p2p) [3]

The game state is composed only of player nodes in a p2p fashion without specific servers or high speed networks. The game state of the MMOG is managed in a distributed way by multiple player nodes with P2P overlay techniques. This will reduce the cost in managing a centralized server.

In this system, three concerns are mainly handled:

b.1. The computation and communication overhead of each node is regulated below the specified threshold independently of the number of players.

b.2. The node failure does not influence the progress of the game.

b.3. The update interval of game state is small enough.

“A game space is dynamically divided depending on the positions of player characters using a technique of Voronoi graph partition algorithm so that player nodes whose characters in the same partitioned region directly exchange game events and keep the same game state” [3].

The multiple sub states in the system and for each sub state a node is assigned to manage it and any other node that requires that sub state can easily retrieve it through distributed hash tables such as Pastry [11]; the above procedure can function as back up too.

The overhead of communication messages is handled by multicast technique Scribe [10] to deliver game events to all player nodes with the same sub state and regulates the communication messages which each node must forward, can be regulated within a threshold.

The game space is divided into multiple sub spaces with the same size. For each sub space there is a responsible node to deliver events which occur in the sub space.

Dynamic load balancing is done whenever there is a large overhead in communication and computation. This is performed by constructing a load balancing tree from the responsible node. Another node is assigned as the backup node to this responsible node which delivers the events to other nodes in the same space. There is a threshold for each node which satisfies the smooth function of communication and computation overhead. The tree expands if there are more participants in the same sub space. As a consequence the end to end delay increases in the tree. In this case to bring the latency to accepted levels the backup node can participate in event delivery too.

Intermediate nodes are substituted by back up nodes, so the delivery can happen from these intermediate nodes instead always from the root. Subscriber nodes leave the tree once the space is empty. Root can only leave once the game is over, this way some computation can be saved up.

4. A Spell of Virtualization

Virtualization opens a whole new window of opportunities at the data centers. It also reduces the hardware set up cost and functions on demand.

The system in context, [2] speaks about an ecosystem where Virtualized and Non Virtualized resources co-exist. This allows the design of the current system to opportunistically take advantage of the competency of a given node on the network and appropriately decide whether or not to instantiate Virtual Machines on the node.

Furthermore [2] also shows that dynamic resource provisioning has the potential to considerably reduce the MMOG operation with a reasonable performance loss.

According to the studies [2] the overhead for instantiation and execution of a virtual machine are below 5% for computation and 15% for networking. This aggregate figure is called the “Virtualization Penalty” and due to the improvement in the virtualization techniques it is expected that a lower penalty is achieved in the future.

However in this paper the virtualization penalty can be compensated by the computation overhead saved by the distributed event delivery mechanism.

5. The Intelligent Optimization and Proxy Architecture

Until this far, the server is optimized and a distributed event delivering mechanism is scheduled to work. Though the system is peer 2 peer and distributed, the Artificial Intelligence for the game is provided by the centralized server and is accordingly distributed to different zones and subsequently to different nodes. It will be ideal to use the resource pool of the participating nodes to build and compute the AI in the system and allow a light weight process on the main server to keep track of the game AI. However, [5] has shown that offloading the AI computation to the clients adds a significant latency and this issue becomes non trivial.

However [5] talks about AI partitioning where the system splits the game AI into tight-loop controlled server side and a tuning client side that supplies parameters to the server side. This process reduces the computation overhead but opens the system for security pit falls as the client side parameters could be tweaked and back doors to the game could be opened.

This is one security to overhead trade off which the design in this paper will make as the aim partly is achieve optimized software architecture. Furthering this notion, there is a continuous dependence of the client machines to generate AI in the game, which requires the expected client machines to be operational. Thus in case a node goes down, it would inadvertently affect the AI in the game as the aggregate computation is lost. To avoid such a scenario, in the distributed event delivery mechanism, it is effective to replicate the client AI architecture on the intermediate nodes.

The system [6] discusses the general communication protocol between the server and the client, through a time critical unicast event stream. This system builds a case by using a non-backing of TCP stream and sets it up against a UDP stream; the system [6] also goes on to that the TCP stream does not have to be slower than the UDP stream and argues that repeated time outs is the only way of having a bad game experience. In regard to this problem [6] comes up with proxy architecture to a centralized server and set of clients which multiplexed saves around 40% of the resources and reduce lag. The observed system could opportunistically be subjected to the design mechanism in [8] to increase the efficiency. This system [6] is also tested on three Pentium machines and has been recorded to function smoothly below the 400 millisecond margin [3], where the users observe ambiguity in the game.

Abstracting the idea from a centralized cluster of servers, to certain responsible nodes in the load balancing tree, a proxy architecture adhering to the above norms could be established between the root, intermediate and subscriber nodes. This would further reduce the management overhead on server nodes in the peer 2 peer system.

6. Virtualization, Virtualization and more Virtualization with a touch of Load Balancing

With the above system in place the standalone devices will join the network as a simple node. In this case each standalone device will have a virtual machine simulated on the nearest node.

The system also facilitates a lobby server which is widely used and built in many gaming clients like the Battle.Net [14] for Blizzard's Warcraft and Starcraft.

This lobby server simply allocates and assigns machines as nodes in the distributed system based on features like proximity, compatibility and complement computation overhead required by the participating nodes.

When a lobby server receives a request from a device, the system with some facts will be able to classify whether or not the incoming request is from a standalone device or a fully capable machine. Once this demarcation is done and if the request is identified as from a standalone device the lobby server could run some minimalistic computation and decide to instantiate a virtual machine on a compliment machine which could support the stand alone device. If such a device could not be found the standalone device will not be able to participate in the system. This is one of the constraints for the architected system to function.

On the contrary if such a device is found, a new virtual machine is instantiated and it is dedicated to the device and now the standalone device would behave like any other node on the network. Here the network latency is now compensated as the standalone device instead of communicating with the server will interact with the nearest node.

Considering the server consolidation into zones where each zone is a virtual machine; another virtual machine could be instantiated within the existing virtual machine. Though this is definitely an optimization, the convoluted virtual machine overhead can compromise performance on some machines. Thus the system with must carry out strict constraint checking before such a decision is made. Though it is useful to do it on a machine which has abundance of resources it is not fruitful to carry on this intricate virtualization on machines with scarcity of resources and thus increase the computation overhead onto some participating server cluster.

The system referred here [4] talks about maintaining a server cluster to resolve scalability issues. The current system can use the implementation of server clusters and provide it on demand. This is a counter measure for handling convoluted virtualization, when some machines are incapable of doing the required amount of virtualization, it could be transferred to the cluster. However this does not resolve the need for a dedicated virtual machine for the standalone device on the nearest node and still acts a constraint.

7. Multimedia Pipelining

Making progress from the previous juncture, since each standalone device is participating on the network through a virtual machine on the nearest computationally compliment instrument, the game data corresponding to the standalone device resides on this machine.

The next task would be to generate the multimedia (audio and video) content for the standalone device on home machine. The following section explores how this is achieved.

The complete development and delivery of the multimedia content is called as Multimedia Pipelining. It starts from the creation of the content; from capturing devices, stimulated images, graphics or image manipulation and ends at rendering the final output to the end user over the network or in form of stored media. The creation starts at the server, local computers or hand held devices. It relates to the creation of graphical content at the very initial stage. By initial we mean we give a format and a set of rendering options. Once the creation is done we compress, encode and deliver it to the end user using the internet structure or stored media or radio or any communicative measures depending upon the audiences. The end part of this pipeline is the rendering which states the interactivity, communication and actions that can be performed on the final output.

In the above said case we implement the usage of MPEG-4 for the multimedia content creation and rendering. MPEG-4 is not very new but has been enormously developed in the last few years in terms of interactivity and features.

MPEG-4 is a very structural and well defined format as it requires description and defining the functionality for every object it carries. All objects or graphic content in multimedia are termed as “audio visual objects” [18].

The MPEG-works by carrying streams for each visual or audio content; it is supposed to render. These streams are called as “elementary streams” [18]. The streams are further made up of units called “access units” [18]. The units are the fundamental units that carry the data that can be a frame or an audio sample.

There are two more streams known as BIFS [18] Binary Format for Scenes and “Object Description stream” [18] that carry the description for the objects; The BIFS carries the scene description for a specific instance.

The systems start at a server where the server gets the bits from different machines and renders a panoramic view of the background or the image that will be used. The server also generates the stream for the character that will be interactive with the user. The stream carrying the character can be a “2D/3D stream” [18] which are well defined in MPEG-4 format. The streams are rendered to the user on a computer or a hand held device. Due to the presence of the hand held device the image and the stream needs to be compressed according to the device's rendering capabilities. This is already defined in MPEG-4 as it has the capability to make many streams for each “audio visual objects” [18] and render the stream according to the bandwidth available.

The user can interact with the video on a player that holds the capabilities to understand the user inputs. The device parses the input and then feeds the inputs to the server, and then the server gets notified. As the character is of constant width and height (as with the case of any MMOG) and movement events are already defined for character in the server the server computes the total motion of the character and computes the frames that are to be rendered using motion compensation technique. And with the help of the sync parameters the server produces the capability of rendering media with synchronization. Thus the video is rendered to users in a synchronized manner and the user has further capabilities to interact with the system. Quality of the stream can be decided based on the device's capabilities and the network capabilities. New interactive methods can be set up in MPEG-4 as it carries the way to describe the motion and the movements and the events that can occur at the rendering end. The interactivity can be defined by the vendor for the audience using the inbuilt structure.

The character stream needs to be considered more importantly as it deals with motion of the character. The character frames are to be created with the use of the alpha channels which determine the visibility of the object pixels in an area. If set to one the pixel is visible else if zero then it is not visible. The number of bits can be defined on the network capabilities. If we want to reduce the bits we can define one bit but this fails at the borders of the object as the audience can perceive the superimposition. On the other hand, if 8 bits are used we can set the alpha channel so that the borders at the character object mix with the background easily and no artifacts are produced.

The crux of the plot here employs the use of MPEG 4 format for game data to multimedia content which can be captured and interpreted on the standalone device and the user input is again transmitted to the home instrument where the dedicated virtual machine respect to the standalone device resides.

8. Over the modem and Out, Wireless Data Transmission

The previous section addressed the issue about assimilating the game content into multimedia content and vice versa. Now the problem about data transmission between different nodes is investigated.

In this section several devices will be explored which can actually cope with the current system design. In particular devices that are capable of supporting the IEEE 802.11 n standard [12].

802.11 n was intended to address the streaming-media needs of the early-adopter consumer those who want to play multiplayer games on an Xbox 360 and talk on a Vonage VoIP line at the same time-the truth is that there is only so much bandwidth available. [13]

Under the best conditions, many 802.11 n routers deliver throughput between 100 Mbps and 120 Mbps near the router. However, speeds degrade quickly to about 45 Mbps in the 300- to 600-foot range, or if other wireless networks are present. It's also critically important to use the matching client adapter with the router. [13]

Practically, with the advent of 4 G technologies, the 802.11 n with LTE are capable of supporting 2-5 Mbps downlink and 6-12 Mbps Uplink. The average user response in games is measured in "Actions Per Minute" or (APM) which is around 60 per minute; though this figure is just an approximation even double or triple the rate which is 180 actions by the gamer per minute or 3 actions (mouse clicks and keyboard strokes) per second which is extremely fast will not affect the designed system as the it is scaled to function at update rate of 400 milliseconds. The studies from the system at [7] provide a scale of response variations which fit with the design of the system. To condition the argument, the Uplink speed of 6-12 Mbps also adds to the advantage.

These are the speeds that are required to seamlessly emulate the game content on the mobile phone to create a real time experience.

Furthermore, looking at a case where home modem is used while playing an online game, the practical data transfer speed experienced between the NIC card on the laptop and the wireless modem will be around 5 Mbps; even though the router claims a speed on 11Mbps or higher only half of it will be utilized due to noise, interference and other external reasons. Concluding this case, a reasonable data transfer speed of 5 Mbps wireless will be sufficient to stimulate real time online gaming experience.

Primarily, this decision is made over facts presented here [3], where a required response time for a game to function smoothly is around 400 milliseconds.

8.1. Apple Iphone [19]

The Iphone 3 running on the 802.11 b/g standard will support a data transfer rate to a maximum upto 1.19 Mbps; this falls short of creating a real time gaming experience.

There have been techniques like adding "chillers" to Iphones to increase the data transfer rate but, it has proven to damage the phone on the long run. Thus Iphone 3 will not qualify for the required specification.

On the other hand Iphone 4 in the market utilizes the 802.11 n standard and is also compliant with legacy standards, but Iphone only utilizes part of the functions of the 4 G technology, thus though Iphone 4 uses the 802.11 n compliant NIC card, it will meet the desired goals.

8.2. HTC Thunderbolt [20]

The thunderbolt on the other hand runs the 802.11 n standards with the LTE compatibility. This has the ability to, react to data at rates of 5 Mbps. This device will be ideal for as a standalone device for the system design architected in this paper.

8.3 Motorola Droid Bionic [21]

The Droid Bionic is Motorola equivalent of the HTC Thundebolt. It is yet to have a release but with the growing industry it is expected to be compliant with whatever Thunderbolt is compliant with. This would also qualify as another contender for a standalone device.

9. New Frontiers

In the current age, cloud based gaming is rapidly growing companies like Gaikai[16] and OnLive[17] use the open gaming cloud where in high end graphic games are played on the browser using mundane Flash and Java technologies. Though most of the games are being tested on these platforms taking them to mobile devices like Verizon HTC Thunderbolt and Verizon Motorola Bionic which supports 802.11 n standards compliant with 4G technology will mark the cusp of mobile gaming for the next decade. The demos for Gaikai include Mass Effect 2, Spore, The Sims 3 and Dead Space 2 which are currently being run and tested.

On the other side of the spectrum is OnLive which delivers instant play, high end video games to TVs. In this case there is some effort in making the set up; OnLive provides a game system bundle that consists of a MicroConsole TV adapter that synchs the broadband modem connection and Television. The idea of Virtualization can be extended in these MicroConsoles and the data can be transmitted to the cellular phone, though the scope of which is beyond this paper.

10. Epilog: Towards Unchartered Waters and Contribution of this Paper

With the advent of Fiber Optics Services and growing wireless markets the NIC cards are becoming faster and faster day by day. The technology is driven towards providing applications that are mobile and easier to port. Unveiling cloud computing technologies, the growth in the hardware industry is becoming comprehensible by the software industry. The gap between the dream killer application and obscure hardware is becoming narrower and narrower.

The system devised in here could as well be extended to Apple i-pads, though that could open up, data transmission over HDMI cables, in a way reduce the virtualization overhead and remove the need of a dedicated nearest node. The scope of that discussion is beyond this paper and could be the work of future.

This paper consolidates the effort of several views on MMOGS and provides an optimized architecture that can function. The paper also explores newer avenues of utilizing the 802.11 n standards to the fullest potential. Furthermore, keying the idea of creating a real time gaming experience by simulating the game as multimedia content was also crucial. In the modern day turning a blind eye towards social networks implies losing a big market; this paper also leaves doors open for the technology implemented in cloud and social networking systems [9].

11. References

[1] Yeng-Ting Lee, Kuan-Ta Chen, "Is Server Consolidation Beneficial to MMORPG? A Case Study of World of Warcraft," cloud, pp.435-442, 2010 IEEE 3rd International Conference on Cloud Computing, 2010

[2] Vlad Nae, Alexandru Iosup, Radu Prodan, Thomas Fahringer, The Impact of Virtualization on the Performance of Massively Multiplayer Online Games, 8th Annual Workshop on Network and Systems Support for Games (NetGames), 2009, p. 1 – 6

[3] Shinya Yamamoto , Yoshihiro Murata , Keiichi Yasumoto , Minoru Ito, A distributed event delivery method with load balancing for MMORPG, Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games, October 10-11, 2005, Hawthorne, NY

[4] Fengyun Lu , Simon Parkin , Graham Morgan, Load balancing for massively multiplayer online games, Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games, October 30-31, 2006, Singapore

[5] J. R. Douceur, J. R. Lorch, F. Uyeda, and R. C. Wood. Enhancing game-server AI with distributed client computation. In Proceedings of NOSSDAV ’07, 2007.

[6] Carsten Griwodz and Pål Halvorsen. The fun of using TCP for an MMORPG. Proceedings of NOSSDAV’ 06, Newport, RI, USA, pages 1–7, may 2006.

[7] Kuan-Ta Chen , Polly Huang , Chin-Laung Lei, How sensitive are online gamers to network quality?, Communications of the ACM, v.49 n.11, November 2006.

[8] Iyengar, Arun K. (Yorktown Heights, NY) 1999 Preserving state information in a continuing conversation between a client and server networked via a stateless protocol United States International Business Machines Corporation (Armonk, NY).

[9] A. Iosup, A. Lascateu, and N. Tapus, CAMEO: Enabling Social Networks for Massively Multiplayer Online Games through Continuous Analytics and Cloud Computing, InACM/IEEE Symposium on Network and Systems Support for Games (NetGames 2010), Taipei, Taiwan, November 16-17, 2010 (accepted, acceptance ratio 30%=9/27)

[10] M. Castro, P. Druschel, A-M. Kermarrec and A. Rowstron: “SCRIBE: A large-scale and decentralized application-level multicast infrastructure”, IEEE Journal on Selected Areas in Communications (JSAC) (Special issue on Network Support for Multicast Communications), 2002

[11] A. Rowstron and P. Druschel: “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems”, Proc. of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware2001), LNCS2218, pp. 329–350, 2001.

[12] Wi-Fi CERTIFIED n: Longer-Range, Faster-Throughput, Multimedia-Grade Wi-Fi® Networks .Wi-Fi Alliance . September 2009.

[13] Repository: http://archive.laptopmag.com/features/common-802-11n-myths-debunked.ht Date Consulted: April 29, 2011, Fact vs. Fiction

[14] Repository: http://us.battle.net/en/faq/ Date Consulted: April 29, 2011, Battle.Net FAQ

[15] Repository: http://forum.garena.com/forumdisplay.php?8-Garena-Support-amp-FAQs Date Consulted: April 29, 2011, Garena FAQ

[16] Repository: http://www.gaikai.com/faq#general Date Consulted: April 29, 2011, Gaikai FAQ

[17] Repository: http://www.onlive.com/support Date Consulted: April 29, 2011, OnLive Support

[18] C. Herpel and A. Eleftheriadis, “MPEG-4 Systems: Elementary Stream Management”, Signal Processing Image Communication (Netherlands), 15, pp. 299-320, 2000.

[19] Repository: http://www.apple.com/iphone/iphone-3gs/specs.html Date Consulted: April 29, 2011, Apple Iphone Specification

[20] Repository: http://www.htc.com/us/products/thunderbolt-verizon#tech-specs Date Consulted: April 29, 2011, HTC Thunderbolt Specification

[21] Repository: http://www.phonearena.com/phones/Motorola-DROID-BIONIC_id5122 Date Consulted: April 29, 2011, Motorola Droid Bionic Specification