BOOSTED METAHEURISTIC ALGORITHMS FOR QOE-AWARE SERVER SELECTION IN MULTIPLAYER CLOUD GAMING

M S . Saranya[1], A. Vijayakumar[2]

[1]*Dept of Computer Science, School of Arts and Science, Vinayaka Mission's Research Foundation, Chennai, India.*
Email : mssaranya452@gmail.com

[2] *Asst. Professor Grade-II, Dept of Computer Science, School of Arts and Science, Vinayaka Mission's Research Foundation, Chennai, India.*
Email: vijayakumar.mca@avit.ac.in

ABSTRACT

Cloud Gaming (CG) provides a high performance and cost-effective solution where players with low-end devices can play high-end games without the need for advanced hardware. A cloud-based video game system offloads all the computational tasks to the cloud. Considering the dynamic nature of game workloads and resource capacity, resource management is still a significant challenge. Since CG is a real-time gaming service, graphics processing units (GPUs) are necessary to accelerate game scene rendering. GPUs are one of the most expensive resources in a CG platform. Therefore, service providers have a strong incentive to utilize GPUs efficiently to maximize their economic profit. In addition, players' quality of game experience (QoE) is a crucial parameter that can directly affect a service provider's profit and must be taken into account in any resource scheduling optimization. To satisfy both parties, in this paper we propose two efficient methods for GPU based server selection in CG. The proposed methods are an improved version of two well-known metaheuristic algorithms called Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), which we refer to as Boosted-PSO and Boosted-GA, respectively. The proposed methods consider service providers' profits and players' experience simultaneously. Our objective is to maximize GPU utilization, which will not only lead to the service provider's economic benefit, but also increase the player's QoE. Our simulation results show that compared to the existing methods to solve such an NP-Hard optimization problem, our Boosted-PSO method, followed by Boosted-GA, achieves the highest efficiency in terms of GPU utilization, capacity wastage, and player's QoE.

Keywords: Cloud gaming, server selection, GPU utilization, genetic algorithm, particle swarm optimization, quality of experience, cloud resource allocation.

1. INTRODUCTION

According to Newzoo's report, there were 2.5 billion active gamers worldwide in 2019, with an expected global games market of $152.1 Billions. Remarkable advancements in cloud computing provide inexpensive and flexible opportunities for game service providers to deploy their games in the cloud, known as Cloud Gaming (CG). In the CG model, Game as a Service (GaaS), or gaming on demand computationally intensive tasks such as the game engine, graphics rendering, encoding the game scenes is performed on remote servers in the cloud, and the game video is streamed to the player's end device. The cloud servers have higher processing power and memory compared to the players' device, and the only requirement on the client-side is a broadband internet connection and ability to play video, not the need for high-end hardware. These features are attractive for game providers and have encouraged even big stakeholders such as Google, Amazon, Verizon, Apple, and Electronic Arts to develop their own video game streaming services.

2. RELATED WORKS

“Model-based evaluation: from dependability to security,”The development of techniques for quantitative, model-based evaluation of computer system dependability has a long and rich history.A wide array of model-based evaluation techniques is now available, ranging from combinatorial methods, which are useful for quick, rough-cut analyses, to state-based methods, such as Markov reward models, and detailed, discrete-event simulation. The use of quantitative techniques for security evaluation is much less common, and has typically taken the form of formal analysis of small parts of an overall design, or experimental red team-based approaches. Alone, neither of these approaches is fully satisfactory, and we argue that there is much to be gained through the development of a sound model-based methodology for quantifying the security one can expect from a particular design.In this work, we survey existing model-based techniques for evaluating system dependability, and summarize how they are now being extended to evaluate system security.

“Markov renewal processes: definitions and preliminary properties,”This paper contains the definition of and some preliminary results on Markov Renewal processes and Semi-Markov processes. The close relationship between these two types of processes is described. The concept of regularity is introduced and characterized. A classification of the states of a Markov Renewal process is described and studied.

“The Probabilistic Model Checking Landscape,”Randomization is a key element in sequential and distributed computing. Reasoning about randomized algorithms is highly non-trivial. In the 1980s, this initiated first proof methods, logics, and model-checking algorithms.The field of probabilistic verification has developed considerably since then. This paper surveys the algorithmic verification of probabilistic models, in particular probabilistic model checking. We provide an informal account of the main models, the underlying algorithms, applications from reliability and dependability analysis-and beyond and describe recent developments towards automated parameter synthesis.

“Model checking for survivability!”Business and social life have become increasingly dependent on large-scale communication and information systems.A partial or complete breakdown as a consequence of natural disasters or purposeful attacks might have severe impacts. Survivability refers to the ability of a system to recover from such disaster circumstances. Evaluating survivability should therefore be an important part of communication system design. In this paper we take a model checking approach toward assessing survivability. We use the logic CSL to phrase survivability in a precise manner.The system operation is modelled through a labelled CTMC. Model checking algorithms can then decide automatically whether the system is survivable. We illustrate our method by evaluating the survivability of the Google file system using stochastic Petri nets.

“Model- Checking Algorithms for Continuous-Time Markov Chains,”Continuous-time Markov chains (CTMCs) have been widely used to determine system performance and dependability characteristics. Their analysis most often concerns the computation of steady-state and transient-state probabilities. This paper introduces a branching temporal logic for expressing real-time probabilistic properties on CTMCs and presents approximate model checking algorithms for this logic. The logic, an extension of the continuous stochastic logic CSL of Aziz et al. (1995, 2000), contains a time-bounded until operator to express probabilistic timing properties over paths as well as an operator to express steady-state probabilities. In CG, players operating on heterogeneous devices; e.g., PC, laptop, tablet, game consoles, desktops, set-top boxes, and smart phones, send their commands to the cloud game server which runs the game engine and makes the appropriate decision to produce the corresponding video frames rendered using a GPU and encoded by a video codec.Afterwards, the compressed frames are streamed to players through the network, and finally decoded and played on user's devices. The tasks that need to be performed on the server-side require a well-organized resource management system to satisfy both the service provider's and end-user's requirementsThis management system would be more challenging for Multiplayer CG (MCG) cases.In our previous work, we proposed a method that optimally assigns cloud servers to the game sessions requested by game players. The method assesses the requested game and its required video quality in terms of frame rate and the load of the frames while considering the capacity of the eligible data centers to allocate an appropriate cloud server to the player for the requested game session. We modeled the problem of server selection as an optimization problem focusing on maximizing both the GPU utilization and players' QoE. Our objective was the maximization of GPU utilization, which leads to the service provider's economic benefit, while increasing the player's QoE. To determine the priority of the two objectives, an end-to-end lag model weights them adaptively.

3. ARCHITECTURE

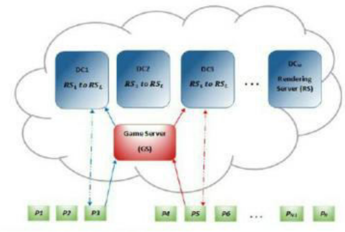

Fig 1: System Design

4. PROCESS AND METHODOLOGY

A. Cloud computing:

According to Newzoo's report, there were 2.5 billion active gamers worldwide in 2019, with an expected global games market of $152.1 Billions. Remarkable advancements in cloud computing provide inexpensive and flexible opportunities for game service providers to deploy their games in the cloud, known as Cloud Gaming (CG). In the CG model, a.k.a Game as a Service (GaaS), or gaming on demand computationally intensive tasks such as the game engine, graphics rendering, encoding the game scenes is performed on remote servers in the cloud, and the game video is streamed to the player's end device. The cloud servers have higher processing power and memory compared to the players' device, and the only requirement on the client-side is a broadband internet connection and ability to play video, not the need for high-end hardware.

B. Multiplayer:

In CG, players operating on heterogeneous devices; e.g., PC, laptop, tablet, game consoles, desktops, set-top boxes, and smart phones, send their commands to the cloud game server which runs the game engine and makes the appropriate decision to produce the corresponding video frames rendered using a GPU and encoded by a video codec. Afterwards, the compressed frames are streamed to players through the network, and finally decoded and played on user's devices. The tasks that need to be performed on the server-side require a well-organized resource management system to satisfy both the service provider's and end-user's requirements. This management system would be more challenging for Multiplayer CG (MCG) **cases.**

C. Game server:

CG system, players can be assigned to the servers. As shown in the depicted framework, the game server (GS) is the point that manages the connections between players and the cloud gaming system. Processing the state of the game, evaluating the resource requirement of the game, and allocating the resources to players are the main functions of the GS to arrange a game session between a group of players. For each received command, the GS evaluates the state of the game and the required resources of the game scenes and then assigns the game player to a server in an eligible datacenter. It is assumed that the eligible datacenter has the minimum required delay, bandwidth, and processing power for the assigned game sessions.

D. Gpu utilization model

GPUs play the leading role in speeding up the rendering of the game's video and reducing the processing delay of the CG system and consequently have a considerable effect on end user's experience. Besides, they are the most expensive hardware rented by providers. Furthermore, not all of the servers in datacenters are equipped with a GPU. For these reasons, management of this resource is highly significant. Therefore, we consider its utilization as one dimension of the proposed server selection method. To this end, we formulate the GPU utilization model, and the formulation for the resource which is utilized by one player.

5. RESULTS

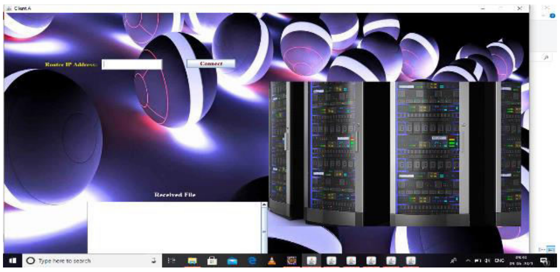

Fig 2: User 1 which is named as client A whose file are transferred based on IP address

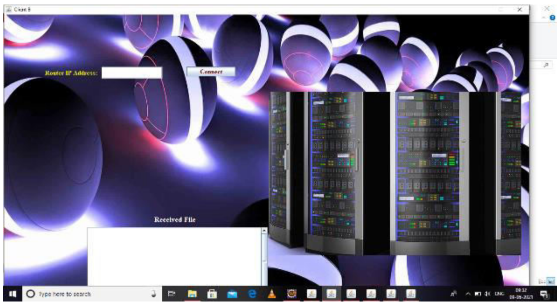

Fig 3: User 2 which is named as client B whose file are transferred based on IP address

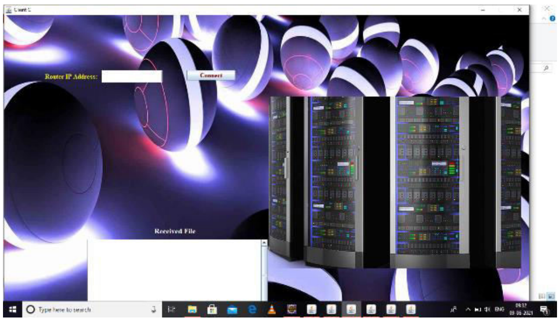

Fig 4.User 3 which is named as client C whose file are transferred based on IP address

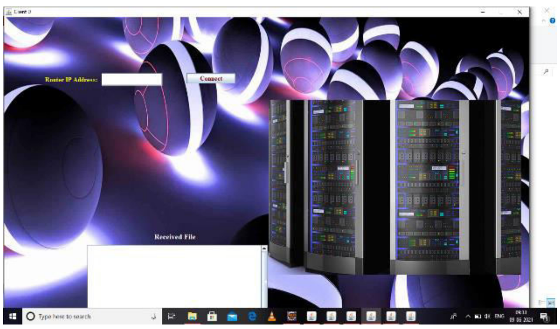

Fig 5.User 4 which is named as client D whose file are transferred based on IP address

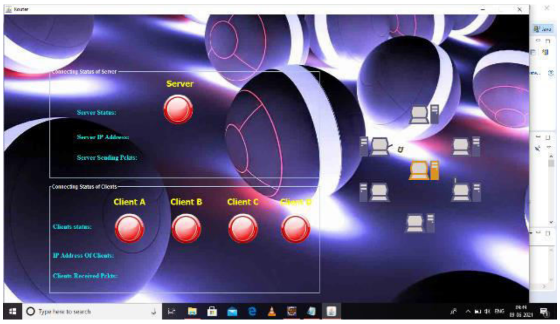

Fig 6: Which controls the default path and send files by converting packages respectively

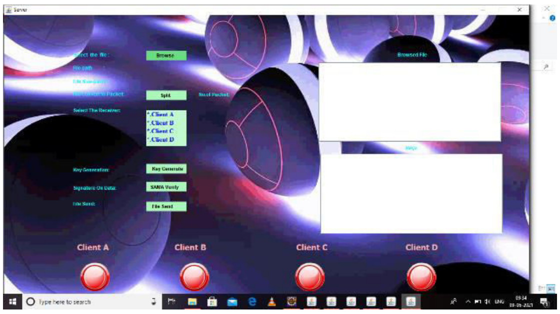

Fig 7. Select a file to browser

Select a Reciever and click split to find how many packages

Then generate key and send files

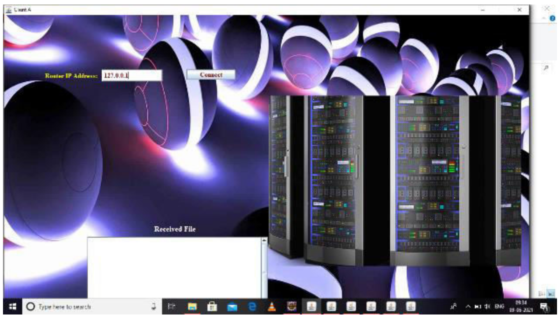

Fig 8. Assigning IP Address for all clients(A,B,C,D)

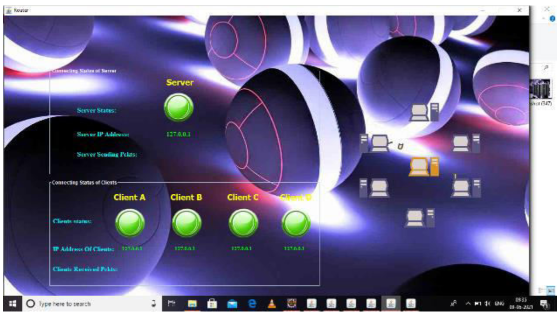

Fig 9. The place which files is selected to send,find packages,send files

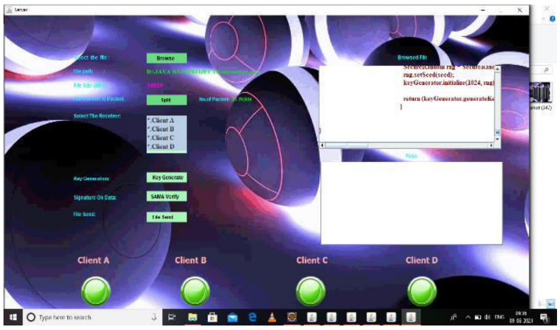

Fig 10. The place which files is selected to send,find packages,send files

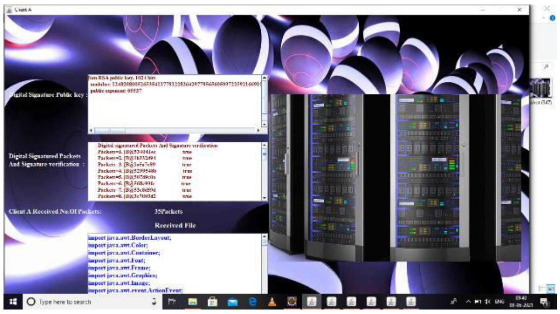

Fig 11: Packets are running in client (A,B,C,D)

Fig 12: All clients had received packages

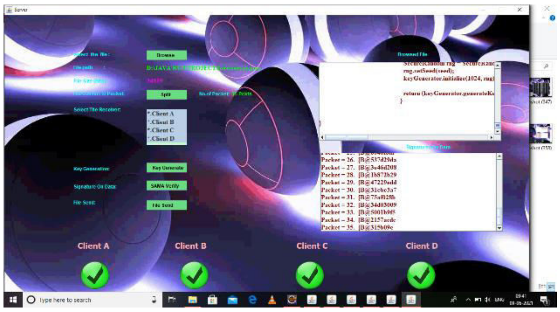

Fig 13: Client Received,All are verified successfully

6. CONCLUSION

In CG, finding an appropriate resource allocation method to satisfy the service provider's profit and the end user's quality of experience simultaneously, is still a challenge. In this work, we address this challenge by presenting a server selection method that considers the provider's and the client's concerns together. The presented objective maximizes GPU utilization while increasing the player's experience. To solve the problem, we proposed two metaheuristic algorithms, Boosted-PSO and Boosted-GA. The proposed methods considerably compensate for the lack of efficiency in our previous work, especially for the low number of players. We evaluated the boosted algorithms in two aspects, utilization and player's experience. Also, we compared the proposed algorithms to the GA and PSO methods presented in and four popular bin-packing algorithms.

7. Future Enhancement:

The simulation results showed that our Boosted-PSO method achieves the highest efficiency among the other methods, including Boosted-GA, FFA, BFA, NFA, and WFA in terms of GPU utilization, capacity wastage, and player's QoE. Also, it has remarkable stability for the different number of players. In our future work, we will consider other quality metrics and also the network parameters to evaluate the problem from the other aspects and to attain a more comprehensive solution.

References:

[1] "The Global Games Market Will Generate $152.1 Billion in 2019 | Newzoo." [Online]. Available: https://newzoo.com/insights/articles/the-global-games-market-will-generate-152-1-billion-in-2019-as-the-u-s-overtakes-china-as-the-biggest-market/. [Accessed: 20-Mar-2020].

[2] M. . .Semsarzadeh, M. Hemmati, A. Javadtalab, A. Yassine, and S. Shirmohammadi, "A video encoding speed-up architecture for cloud gaming," in 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2014, pp. 1–6.

[3] "Sony says cloud gaming could be future threat to PlayStation business - Business Insider." [Online]. Available: https://www.businessinsider.com/sony-says-cloud-gaming-future-threat-to-playstation-business-2019-2. [Accessed: 20-Mar-2020].

[4] "Global Cloud Gaming Market Will Reach USD 6,944 Million By 2026: Zion Market Research." [Online]. Available: https://www.globenewswire.com/news-release/2018/12/31/1679151/0/en/Global-Cloud-Gaming-Market-Will-Reach-USD-6-944-Million-By-2026-Zion-Market-Research.html. [Accessed: 20-Mar-2020].

[5] "Sony Computer Entertainment to Acquire Gaikai Inc., A Leading Interactive Cloud Gaming Company SCE To Build A Cloud Service Bringing Gaikai's Cloud Based-Streaming Technologies into Its Network Business." [Online]. Available: https://www.sie.com/en/corporate/release/2012/120702.html. [Accessed: 20-Mar-2020].

[6] "OnLive." [Online]. Available: http://onlive.com/. [Accessed: 20-Mar-2020].

[7] "PlayStation Now game-streaming service coming summer 2014 (update) - Polygon." [Online]. Available:

https://www.polygon.com/2014/1/7/5284504/sony-playstation-now-gaikai-based-streaming-service-ps-ps2-ps3. [Accessed: 20-Mar-2020].

[8] "Company Profile | Broadmedia GC Corporation." [Online]. Available: https://www.broadmediagc.co.jp/en/corporation/. [Accessed: 20-Mar-2020].

[9] "Stadia Founder's Edition - One place for all the ways we play - Google Store." [Online].

Available:
https://store.google.com/ca/product/stadia_learn.
[Accessed: 20-Mar-2020].

[10] “PlayStation Now – Online Streaming Services on PS4 or PC - PlayStation.” [Online]. Available: https://www.playstation.com/en-ca/explore/playstation-now/. [Accessed: 20-Mar-2020].