Resource Management for Cognitive Cloud Gaming

Wei Cai[1], Min Chen[2], Conghui Zhou[3], Victor C.M. Leung[1], Henry C.B. Chan[4]
[1]Department of Electrical and Computer Engineering, The University of British Columbia, Canada
[2] School of Computer Science and Technology, Huazhong University of Science and Technology, China
[3] We Software Limited, Hong Kong
[4] Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[1] {weicai, vleung}@ece.ubc.ca, [2] minchen2012@hust.edu.cn [3] neio.zhou@gmail.com, [4] cshchan@comp.polyu.edu.hk

***Abstract*—In contrary to conventional gaming-on-demand solution, cognitive cloud gaming platform facilitates gaming component migration from the cloud server to the players' terminal, a novel flexible solution to provide Gaming as a Service. In this work, we model the component-based game and investigate the capacity of intelligent resource management for different optimization targets, including cloud resource minimization and throughput-oriented optimization. Experimental results show that, with the cognitive resource management, cloud system is adaptive to various service requirements, such as increasing the quantity of supported devices and reducing the network throughput of user terminals, while satisfying players' quality of experience.**

***Index Terms*—cognitive; resource allocation; cloud; game**

I. Introduction

Recently, there has been increasing interest from researchers, system designers, and application developers on a new type of cloud service mode generally known as Everything as a Service (EaaS) [1]. Along with this trend and the developments of game industry, the transformation of traditional gaming software into Gaming as a Service (GaaS) becomes one of the most active research topics [2]. The GaaS model exhibits several advantages: i) *Scalability*: it overcomes the constraints of terminal gaming hardware, including processing capacity, data storage and battery in mobile devices; ii) *Cost-effective*: it reduces the production cost with a unified development approach; iii) *Ubiquitous and Multiple-platform Support*: it provides cross-platform and seamless gaming experience; iv) *Effective Anti-Piracy Solution*: it transforms the game developing companies to game service providers, providing a potential solution to the troublesome piracy problems; vi) *Click-and-Play*: it supports a play-as-you-go mode, in which the players start their gaming sessions without downloading and setting up the game copies. Therefore, not only the academia, but also the industry raise great expectations on the development of GaaS solutions.

Therefore, game companies such as OnLive[1] and Gaikai[2], G-Cluster[3], start to provide commercialized GaaS to the public. For this purpose, they utilize gaming-on-demand (i.e. Remote Rendering GaaS, RR-GaaS) model, which is the most popular and mature cloud gaming service model. In this model, video games are executed in cloud servers, and the gaming video frames are transmitted to televisions or desktop PCs over the Internet. In reverse, the interactive information of game players are sent to the cloud server over the same network [3]. In this context, the cloud is intrinsically an interactive video generator and streaming server, while the mobile devices function as the event controllers and video receivers that can run sophisticated games despite the capacity of their restricted hardware. This results in longer battery life for the device and longer gaming times for the user at the expense of higher energy consumption in to of communications and network. It is obvious that the workload of gaming video rendering in the cloud is extremely heavy, and the video frame transmissions via Internet also incur huge amount of throughput.

With the improvement of hardware performance, most gaming terminals, including mobile devices, are capable to perform complicated graphical rendering for game scenes. Under this circumstance, rendering module is implemented in the gaming terminal for Local Rendering GaaS (LR-GaaS) model to eliminate the high network burden of real-time video transmission. However, for typical LR-GaaS model, e.g. browser games [4], the workload in the cloud is still very heavy, when the quantity of users increased to a certain level. Moreover, it is not a flexible solution: in fact, some procedure can be manipulated locally, to reduce the responsive latency and also reduce the cloud workload.

As a conventional assumption, both of these two types of GaaS models require continuous Internet access. However, players are suffering from temporary disconnection to the cloud, especially when they are connecting with mobile networks. In this case, the gaming session will be suspended or even destroyed, which disrupts players's quality of experience (QoE). In some particular scenarios, e.g. the players discover some valuable items in the battlefield, or encounter some special Non-Player Characters (NPCs), the disconnection will make them very disappointed. In [5] a cognitive cloud gaming platform is presented to facilitate "click-and-play" and flexible resource allocation. The proposed component-based game design also allows the players to continue their gaming sessions in particular gaming scenes without network connection. Accordingly, the optimization of the cloud-terminal workload in order to provide an efficient and QoE-oriented gaming services, is a very interesting topic.

In this paper, we model the resource management problem

[1] http://www.onlive.com
[2] http://www.gaikai.com
[3] http://www.g-cluster.com

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

for the cognitive cloud gaming platform and perform preliminary experiments to illustrate the benefits of flexible resource allocation. The remaining sections of the paper are organized as follows. We review related work in Section II and give an overview of the implementation of cognitive cloud gaming platform in Section III. Then, in Section IV, we describe the Modeling and optimization targets. Afterwards, experiments and preliminary results are presented in Section V. Section VI concludes the paper.

II. Related Work

A. Dynamic Partitioning

Resource allocation optimization problem is intrinsically a group of dynamic partitioning problem for multiple devices. Researches on the dynamic partitioning between cloud and users' mobile terminal have been conducted for general-purpose applications. The work [6] first introduces a K-step algorithm to compute partitioning on-the-fly, when a phone connects to the server and specifies its resources and requirements. Furthermore, the work [7] formulates the dynamic partitioning problem and discusses the supporting platform to facilitate it. Afterwards, a dynamic partitioning system named CloneCloud is designed in the work [8]. As a flexible application partitioner and execution runtime, CloneCloud enables unmodified mobile applications running in an application-level virtual machine to seamlessly offload part of their execution from mobile devices onto device clones operating in a computational cloud. However, a basic requirement of these work is that, identical application copies shall be resided on both cloud and terminal in priori, which conflicts with our design requirement of "click-and-play" for the cloud-based games.

B. Cognitive Cloud Gaming Platform

The work [5] first designs and implements a cognitive gaming platform that supports both "click-and-play" and cognitive resource allocation. The designated cognitive platform, consists of Information Collector, Performance Evaluator, Onloading Manager, Synchronization Controller, and Partitioning Coordinator, is able to dispatch selected gaming components from cloud to players' terminal and later facilitate dynamic partitioning to adapt its service QoS to the real-time systemic environment. As a development-friendly environment, the platform also provides a set of application programming interfaces (APIs), so that the game developers need not to understand the lower layer resource management details but only focus on the design of game programs.

C. QoE for Cloud Gaming

Maintaining an acceptable QoE is an imperative design concern for cloud gaming systems. To provide GaaS, the relationships between cloud gaming QoE and QoS are different for distinct implementation architectures. For remote rendering GaaS, various subjective user studies have been conducted to demonstrate the relationship between cloud gaming QoE and QoS, including game genres, video encoding factors, CPU load, memory usage, and link bandwidth utilization [9], response latency and the game's real-time strictness [10], network characteristics (bit rates, packet sizes, and inter-packet times and an empirical network traffic analysis of On-Live and Gaikai [11]. However, the QoE to QoS mapping shall be redefined given the rendering component is resided on the local terminal. Due to multiple remote invoke between components, the impact of network QoS parameters will intensively impact the QoE for players.

III. System Overview

A. Cognitive Resource Management

In this section, we provide an overview for the proposed cognitive resource management. As a cognitive system, it is cognitive of resources and characteristics of the cloud, the access network, and the end-user devices, to enable dynamic utilization of these resources. To the best of our knowledge, a QoE-oriented cognitive system is not yet investigated for cloud gaming systems.

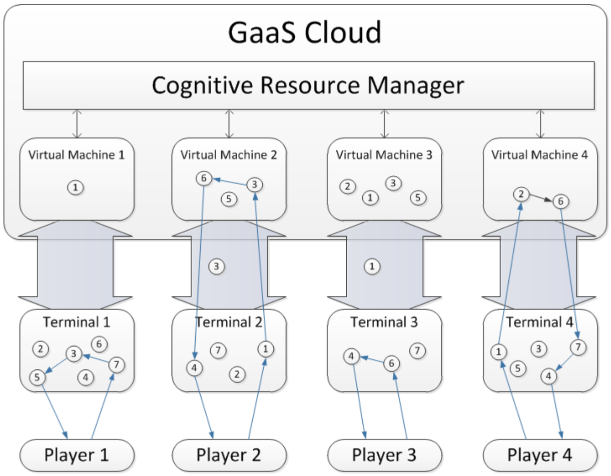

Fig. 1. Architecture Framework for Cognitive Mobile Cloud Gaming

In a typical scenario, the GaaS cloud hosts a number of terminals with diversity of user-end devices and frequent changes in network QoS and cloud responses. The *Cognitive Resource Manager* monitors and assesses all terminals' working status and dynamically determines partitioning solution for each of them according to the system status, e.g. the available network bandwidth and remaining resources in the cloud, in order to provide cognitive capabilities across the cloud gaming system. As shown in Fig. 1, the four terminal hosts different number of components locally, since their device status is distinct from each other. In the mean time, the *Cognitive Resource Manager* schedules the Virtual Machine 2 and 3 to dispatch components to their corresponding terminals, since the available bandwidth allows the background downloads.

B. Implementation of Cognitive Cloud Gaming Platform

Fig. 2 illustrates the implementation of the cognitive platform to facilitate the dynamic partitioning. *Information Collectors* on both the cloud and mobile sides monitor resource usage at cloud, access network and mobile terminal. These

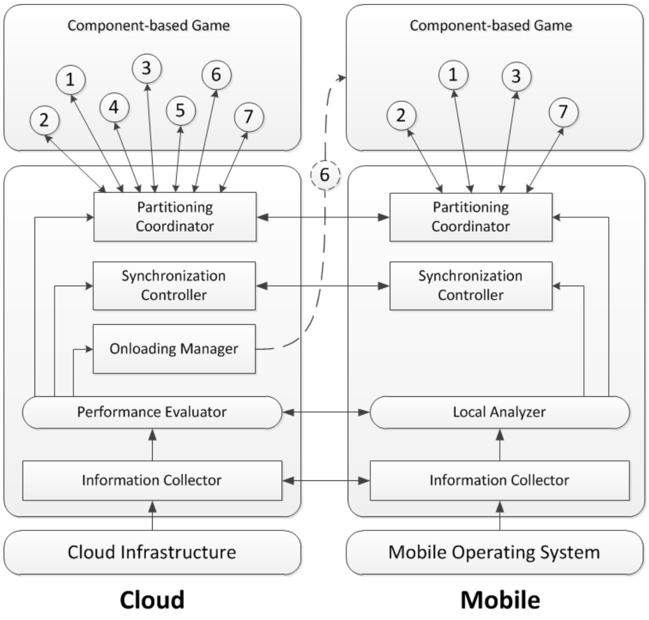

Fig. 2. Cognitive Platform for Mobile Cloud Gaming

surveillance data are reported to the *Performance Evaluator* and *Local Analyzer*. Note that the *Performance Evaluator* is able to guide the procedure in *Local Analyzer*, while the *Local Analyzer* reports its results to *Performance Evaluator* periodically for further evaluations. Games designed for the cognitive platform consist of a number of inter-dependent game components. These components are able to migrate from the cloud to the mobile terminal via a network, under the instruction of the *Onloading Manager*. As the message gateway between components, the *Partitioning Coordinator* intelligently selects destination components, locally or remotely, to achieve dynamic resource allocations. The *Synchronization Controller* is designed to guarantee the synchronization of data in identical components distributed in the cloud and mobile terminals. Note that the *Onloading Manager*, *Partitioning Coordinator* and *Synchronization Controller* are manipulated by the *Performance Evaluator* and *Local Analyzer* for the purpose of maintaining an acceptable QoE for players.

IV. System Modeling

In this section, we model the resource management problem from the perspectives of game components, hard QoS control, and optimization targets, respectively.

A. Game Components

In cognitive cloud gaming platform, games are consisted of inter-dependent components that work collaboratively to provide gaming services for players. As shown in Fig. 3, we denote the dependency of game components as a directed graph $G = \{C, E\}$, where every vertex in C is a component c_i and every edge $e_{i,j}$ in E is a dependency between c_i and c_j . Each component c_i is characterized by following parameters in Table I:

TABLE I
Notation 1

Symbol	Definition
r_i	the resource consumption of c_i
s_i	the size of the compiled code of c_i
$in_{j,i}$	the amount of data that c_i takes in input from c_j
$out_{j,i}$	the amount of data that c_i sends in output to c_j
$fin_{j,i}$	the frequency that c_i takes data input from c_j
$fout_{j,i}$	the frequency that c_i sends data output to c_j

Once the mobile terminal fetches the game components from the cloud, *Partitioning Coordinator* should work with *Performance Evaluator* and *Local Analyzer* to solve the dynamic partitioning problem, in order to provide a QoE-oriented resource optimization. In this framework, all input and output data from the components are sent to the *Partitioning Coordinator*, which provides a routing service for invoke messages by intelligently selecting the destination components when an application cycle is determined.

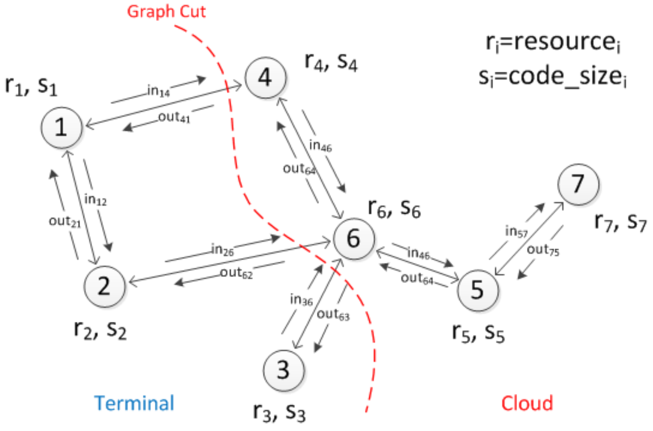

Fig. 3. Components Partitioning for Proposed Cognitive Platform

As depicted in Fig. 3, the partitioning problem intrinsically seeks to find a cut in the consumption graph such that some components of the game execute on the client side and the remaining ones on the cloud side. The optimal cut maximizes or minimizes an objective function O, which expresses the general goal of a partition, e.g., minimizing the end-to-end interaction time between the mobile terminal and the cloud, minimizing the resource consumption in the cloud, or minimizing the average throughput for the game terminals. Therefore, the resource management problem is to seek an optimal set of all connecting terminals' partitioning solutions that meets the system's optimization target.

B. Hard QoS Control

As a gaming service provision system, to satisfy all players' QoE is a fundamental requirement. Therefore, it is a mandatory to guarantee the minimal QoS parameters for all connecting terminals, so called "Hard QoS". We formulate the hard QoS constraints with additional notations in Table II.

TABLE II
NOTATION 2

Symbol	Definition
$\{n \vert n \in N\}$	the set of N terminals
σ	the set of components allocated in cloud
τ	the set of components allocated in terminal
$r(n)_i$	source consumption of c_i for terminal n
$d(n)_{i,j}$	throughput between c_i and c_j for terminal n
$l(n)_p$	process latency for terminal n
$l(n)_c$	communication latency for terminal n
L_M	maximal latency that player can tolerate
$T(n)_{RTT}$	Round Trip Time for terminal n
$F(n)_T$	time efficiency factor for terminal n
$F(n)_C$	time efficiency factor for cloud n
$R(n)_T$	the available resources in terminal n
$B(n)_T$	the available bandwidth in terminal n
R_C	the available resources in the cloud
B_C	the available bandwidth in the cloud

1) Terminal Resource Constraint: The terminal device needs sufficient resources to host a set of downloaded gaming components. To simplify our model, we investigate available memory as the only indicator of resources. The constraint on terminal resource is formulated as:

$$\sum_{c_i \in \tau} r(n)_i \leq R(n)_T, \forall n \in N \tag{1}$$

2) Terminal Throughput Constraint: The remote invokes between components introduce bandwidth consumption to the gaming procedure. To guarantee the QoE for players, the system need to control all terminals' gaming throughput. We formulate the expected throughput for terminal n as:

$$d(n)_{i,j} = fin(n)_{j,i} \cdot in(n)_{j,i} + fout(n)_{i,j} \cdot out_{i,j} \tag{2}$$

and the constraint on terminal throughput is formulated as:

$$\sum_{\substack{c_i \in \tau \\ c_j \in \sigma}} d(n)_{i,j} \leq B(n)_T, \forall n \in N \tag{3}$$

3) Interaction Latency Constraint: Interaction latency represents the time interval between players' input and the system response. In GaaS system, the latency is consisted of two parts: *processing latency* and *communication latency*. *Processing latency* $l(n)_p$ is formulated as the sum of processing time for all components, which is proportional to the components' resource consumption with efficient factor $F(n)_T$ for terminal and efficient factor $F(n)_C$ for cloud:

$$l(n)_p = \sum_{c_i \in \tau} r(n)_i F(n)_T + \sum_{r_j \in \sigma} r(n)_j F(n)_C \tag{4}$$

The *communication latency* $l(n)_c$ is formulated as the sum of communication delays for all remote invokes between components:

$$l(n)_c = \sum_{\substack{c_i \in \tau \\ c_j \in \sigma}} f(d(n)_{i,j}, T(n)_{RTT}) \tag{5}$$

where f is the function to calculate the latency from specific communication package and RTT. Accordingly, we conclude the constraint on interaction latency as

$$l(n)_c + l(n)_p \leq L_M, \forall n \in N \tag{6}$$

4) Cloud Resources Constraint: As a service provider, the cloud consumes its resources to host a set of components for terminals. Therefore, to guarantee the resource provision capacity is a critical issue in QoS assurance. The same as previous assumption, we adopt memory as the only indicator of the required resources and formulate the constraint on cloud resources as:

$$\sum_{n \in N} \sum_{c_i \in \sigma} r(n)_i \leq R_C \tag{7}$$

5) Cloud Throughput Constraint: During the gaming session, the cloud handles network connections from the terminals. Therefore, we also need to formulate the constraint on the overall cloud throughput as follows:

$$\sum_{n \in N} \sum_{\substack{c_i \in \tau \\ c_j \in \sigma}} d(n)_{i,j} \leq B_C \tag{8}$$

C. Optimization Targets

As a cognitive system, the GaaS provider is capable to adapt its service to the various status of cloud, terminal and access network. In this work, we design a set of optimization targets to demonstrate the flexibility and efficiency of the proposed system.

1) Cloud Resource Minimization: Cloud is considered as an infinite resource provider in ideal context. However, the capacity of cloud is still restricted by existing virtualization techniques. Moreover, there is no free lunch. GaaS companies's usage of cloud services are charged in a pay-per-use way. Therefore, to minimize the cloud resource utilization is of great importance from the economic perspective, especially when the number of terminals reaches a peak in a certain period of time. For this purpose, an optimization target for the cognitive gaming platform is "Cloud Resource Minimization", in which the resource manager monitors the cloud resource pool in real-time and dynamically allocates components to achieve the optimization. For instance, when the workload of gaming cloud reaches a certain threshold, the cloud intelligently allocates more components to selected terminals to reduce its own resource consumption. Note that, the selecting procedure shall be under the supervision of the hard QoS control, thus, to guarantee QoE satisfactory for all players.

2) Throughput-Oriented Optimization: As a cloud system, network quality is always the most critical bottleneck, especially for gaming, a latency-sensitive application. On the other hand, players who accesses gaming services via paid mobile network also concern their bill amount. Hence, the system network throughput is an important factor that impacts both players' QoE and users' interests on GaaS model. Correspondingly, another optimization target for the proposed cognitive resource manager is a "throughput-oriented optimization" mode, in which the cognitive platform monitors the resource status for each mobile device and dynamically determines an optimized partitioning solution to minimize the terminals' average throughput, thus, to provide best QoE for the players.

V. Simulations

A. Simulation Setup

To validate the performance of the cognitive resource management of the proposed gaming platform, we set up the following simulations. For cloud side, we initiate a resource pool with 100 gigabyte (GB) memory and 1 gigabit per second (Gbps) gateway bandwidth. To the calculate of the *Communication Latency* $l(n)_c$, we assume the function f described in Section IV-B follows the formula below:

$$f(d(n)_{i,j}, T(n)_{RTT}) = d(n)_{i,j} T(n)_{RTT} F_{comm} \qquad (9)$$

where F_{comm} is the scale factor representing the relationship between communication data size and latency. The default simulation parameters for terminal devices are listed in Table III, in which all random variables follows uniform distribution.

TABLE III
Parameters for Terminal Devices

Parameter	Value
Round Trip Time (T_{RTT})	10ms∼25ms
Maximal Bandwidth	80kbps∼120kbps
Available Memory	80MB∼160MB
$F(n)_C$	0.01
$F(n)_T$	0.02
F_{comm}	0.1

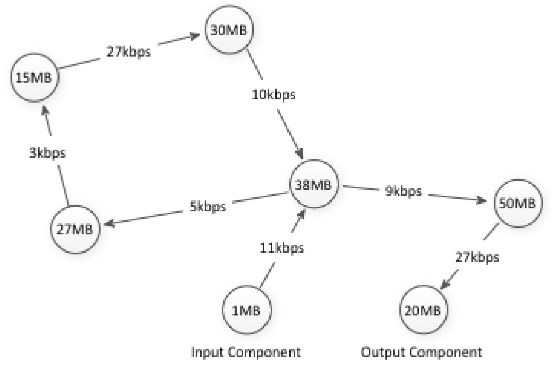

Fig. 4. Components Partitioning for Proposed Cognitive Platform

We design a set of component invokes to simulate a gaming interaction procedure, as shown in Fig. 4. The number in the circle represents the resource consumption for components, while the number on the arrow represents the average throughput for the remote invokes. To simplify the QoE measurement, we adopt 120 ms as the maximal tolerable latency value as the indicator of QoE. With these settings, we compare our cognitive resource management solutions, named *Cognitive Mode*, with *Pure-Cloud Mode*, the conventional cloud gaming architecture, which implements the *Input Component* and *Output Component* in the terminals while offloading the other components to the cloud. All simulations are repeated for 1000 times with distinct random seeds to yield an average value.

B. Cloud Resource Minimization

In this section, we compare the performances of *Pure-Cloud Mode* and *Cognitive Mode* for the cloud resource minimization. In our implementation, the cognitive resource manager adopts an exhaustion approach to seek the optimal component allocation solution that minimizes the cloud resource consumption while guarantees all players' QoE requirement.

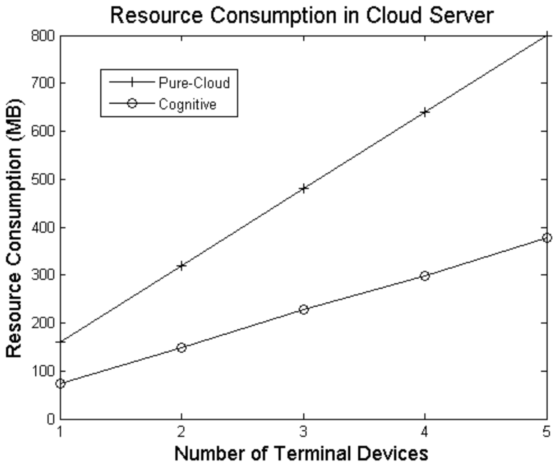

Fig. 5. Resource Comparison for Cloud Resource Optimization

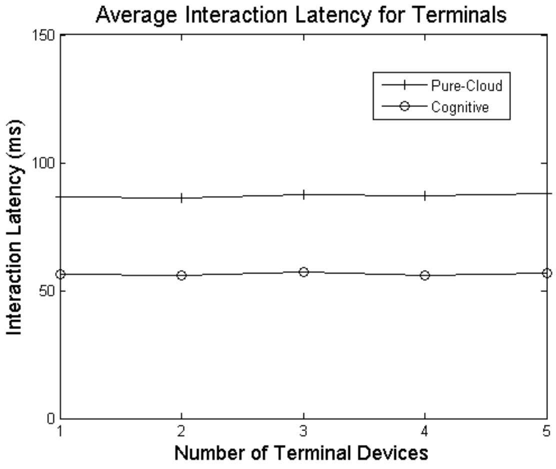

Fig. 6. Latency Comparison for Cloud Resource Optimization

Experimental results on the cloud resource consumption are illustrated in Fig. 5. The resource consumption of *Pure-Cloud Mode* increases proportionally to the quantity of terminal devices, since its resource allocation is not adaptive to the system status. In contrast, the *Cognitive Mode* reduces the resource usage by up to 50%. In other words, with certain amount of the cloud resources, our cognitive resource management enables the gaming server to support terminals of double quantity. To ensure the QoE for game players, we also investigate the average interaction latency for terminals, as depicted in Fig. 6. The comparison indicts that, in our experimental settings, the proposed *Cognitive Mode* also outperforms the conventional *Pure-Cloud Mode* in terms of interaction latency.

C. Throughput-Oriented Optimization

In this section, we compare the performances of *Pure-Cloud Mode* and *Cognitive Mode* in terms of throughput-oriented optimization. The same as the cloud resource minimization, exhaustive approach is applied to determine the best resource allocation solution that minimizes the throughput for the terminals while satisfying the QoE requirements for all players.

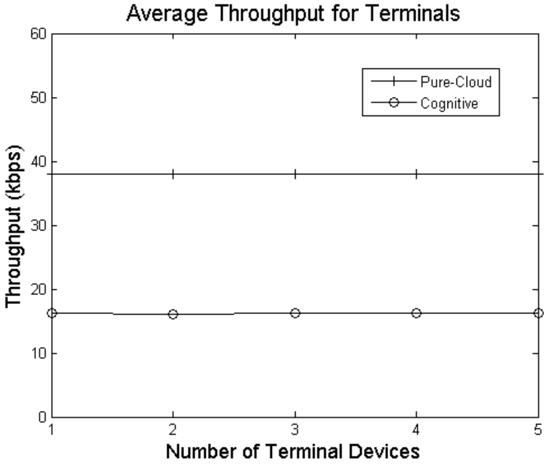

Fig. 7. Throughput Comparison for Throughput Oriented Optimization

Fig. 7 shows the comparison on the average throughput for the terminal devices. In fact, the throughput for *Pure-Cloud Mode* is determined by the transmission rate of *Input Module* and *Output Module*. In contrast, the *Cognitive Mode* intelligently select the optimal partitioning solution with minimal throughput, which provides a 55% deduction on network throughput in our experimental settings.

D. Computation Complexity Analysis

Given the cloud-based game is consisted of C components and provides the GaaS for N terminals, the quantity of existing component allocation solutions is 2^{NC}, which implies an extremely high computational complexity of exhaustive approach, especially when more and more terminals access the cloud for gaming services. Hence, we only provide preliminary results with limited number of terminal devices in this work, due to the constraints of simulation time. Our future work is to investigate a more sophisticated and efficient approach, e.g. an iterative genetic algorithm, to support the real-time resource management for the cognitive cloud gaming platform.

E. Discussion on Game Design

In our experiments, we realize that the game design, including the resource consumption of each component and the data communication between components, will substantially impact the system performance. Hence, one of our future work is to investigate the features of diverse genres of games, to further explore the optimization potential of the cloud-based gaming system.

VI. Conclusion

The cognitive cloud gaming platform introduces a flexible component allocation solution for the promising GaaS provision. In this paper, we provide a formulation for the resource management problem and demonstrate the proposed system's flexibility of diverse optimization targets. Preliminary simulation results suggest that, the cognitive platform provides great efficiency potential in terms of resource minimization and throughput optimization, while guaranteeing the QoE requirements for game players.

Acknowledgement

This work is supported by a University of British Columbia Four Year Doctoral Fellowship and by funding from the Natural Sciences and Engineering Research Council.

References

[1] P. Banerjee, R. Friedrich, C. Bash, P. Goldsack, B. Huberman, J. Manley, C. Patel, P. Ranganathan, and A. Veitch, "Everything as a service: Powering the new information economy," *Computer*, vol. 44, no. 3, pp. 36–43, 2011.

[2] P. Ross, "Cloud computing's killer app: Gaming," *IEEE Spectrum*, vol. 46, no. 3, pp. 14–14, 2009.

[3] C. Huang, C. Hsu, Y. Chang, and K. Chen, "Gaminganywhere: an open cloud gaming system," in *Proceedings of the 4th ACM Multimedia Systems Conference*, ser. MMSys '13, New York, NY, USA, 2013, pp. 36–47.

[4] J. Vanhatupa, "Browser games: The new frontier of social gaming," in *Recent Trends in Wireless and Mobile Networks*, ser. Communications in Computer and Information Science, vol. 84. Springer Berlin Heidelberg, 2010, pp. 349–355.

[5] W. Cai, C. Zhou, V. Leung, and M. Chen, "A cognitive platform for mobile cloud gaming," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2013.

[6] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, "Calling the cloud: enabling mobile phones as interfaces to cloud applications," in *Proceedings of the ACM/IFIP/USENIX 10th international conference on Middleware*, ser. Middleware'09, Berlin, Heidelberg, 2009, pp. 83–102.

[7] B. Chun and P. Maniatis, "Dynamically partitioning applications between weak devices and clouds," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, ser. MCS '10, New York, NY, USA, 2010, pp. 7:1–7:5.

[8] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*, ser. EuroSys '11, New York, NY, USA, 2011, pp. 301–314.

[9] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hossfeld, "An evaluation of qoe in cloud gaming based on subjective tests," in *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, June 30-July 2 2011, pp. 330 –335.

[10] Y. Lee, K. Chen, H. Su, and C. Lei, "Are all games equally cloud-gaming-friendly? an electromyographic approach," in *Proceedings of IEEE/ACM NetGames 2012*, Oct 2012.

[11] M. Manzano, J. Hernandez, M. Uruena, and E. Calle, "An empirical study of cloud gaming," in *Network and Systems Support for Games (NetGames), 2012 11th Annual Workshop on*, 2012, pp. 1–2.