NSF Cloud Workshop: *Cloud Gaming – Provisioning for large real time online environments*

Magda El Zarki and Nalini Venkatasubramanian, UC Irvine

Introdution

Although cloud gaming is a promising direction for the game industry, the current stream-based offerings are not the solution that the gaming world is ready to adopt for all gaming scenarios. Achieving good user experience without excessive hardware investment is a tough problem. This is because gamers are hard to please, as they concurrently demand for high responsiveness, game accuracy and a level playing field where network characteristics do not play to the advantage of some players. But gamers do not want to pay too much. Therefore, service providers have to not only design the systems to meet the gamers' needs but also take latency, error resiliency, scalability, synchronization, resource allocation and cost into consideration. This renders the design and implementation of cloud gaming systems very challenging.

The traditional view of cloud platforms primarily focuses on how cloud providers supply resources and services on-demand from large pools of resources installed in data centers. The goal is to realize economies of scale and increased utilization by sharing resources/services as available through technologies such as virtualization and multi-tenancy. Examples include Amazon EC2, Google Compute Engine, Windows Azure Cloud Services, Rackspace, etc. In the multiuser game context, this translates to techniques to offload compute and data intensive tasks from end-clients (who have limited game context and resources) to cloud servers where such context can be gathered, assimilated, and processed. While public clouds provide resources at scale; there are a limited number of public cloud data centers within close proximity of end-users resulting in large communication latencies within the network infrastructure.

The idea of remote execution of resource-intensive tasks to alleviate resource constraints is not new; it has been recently explored, especially in the context of mobile cloud applications. Such approaches constantly monitor resource consumption and availabilities (e.g., CPU, network, etc.) to further optimize the resource usage. Recent efforts, e.g., Cloudlets [SAT2009] and MAPCLOUD [RAH2013], have demonstrated the role of local resources within close proximity of the user in ensuring improved application latencies. What is missing in the above efforts is an explicit consideration of the game QoE that are required for efficiently distributing the game workflow components among multiple clouds. In addition, user dynamicity, if not addressed properly, can result in suboptimal resource mapping choices and ultimately in diminished game QoE.

We argue that a new view of the cloud infrastructure is necessary to address the next generation of rich applications such as immersive networked games - a view that supports convergence of the service, compute, communication, and storage infrastructure. In such a view, the networks and the associated servers in the network are key components of the cloud infrastructure. The key idea is to overcome the resource limitations of mobile devices and networks by leveraging resources available in distributed cloud environments. For example the Network-as-a-Service (NaaS) frameworks [COS2012] integrate current cloud computing offerings with direct access to the network infrastructure. The idea is to enable tenants/users to easily deploy custom protocols for routing and multicast; modify the content of packets on-path and efficiently implement advanced network services, such as in-network data aggregation, and smart caching.

Research Thrusts

Multiple research challenges need to be addressed to realize the benefits for *group-based interactive and immersive applications* in a cloud setting. Techniques must be developed to efficiently allocate the shared cloud resources across multiple cloud gaming users with common states so as to achieve high QoE for users and high system utilization for the cloud. In this section we discuss some of these key problems and shed some light on potential approaches to solve them.

Modern games, especially MMOGs, are extremely complex. Hence it is not directly obvious how one can map the different game functions onto cloud resources while guaranteeing the QoE of the online game. Distributing the workflow components among servers in multiple clouds without suffering from high and unpredictable latencies is challenging due to the limitations of current cloud platforms

We will begin with a simple workflow based model of online games; the figure below illustrates the generic workflow of online computer games. In each game scene, a client begins by checking if it exists in the zone. Next, it checks if some prediction has to be done about the other player positions to reduce network traffic. This flow is part of the client side optimization techniques that are very popular in game development to counter network idiosyncrasies. The player then proceeds with an objective that is *allowed* given the current location and zone view. The client then executes the chosen objective either locally or remotely based on the type of action required to achieve the objective and also on the update frequency policy with the server. On the server side of a game scene, the server receives requests from multiple clients to perform actions. These requests are added to a list of pending requests maintained per server execution window. At the end of each execution window, requests in the list are ordered based on the server optimization algorithm implemented. The server executes these requests, updates global zone information and sends the changes to the clients. The model is fairly simple, but is a start in trying to understand the tasks and assign them to cloud components.

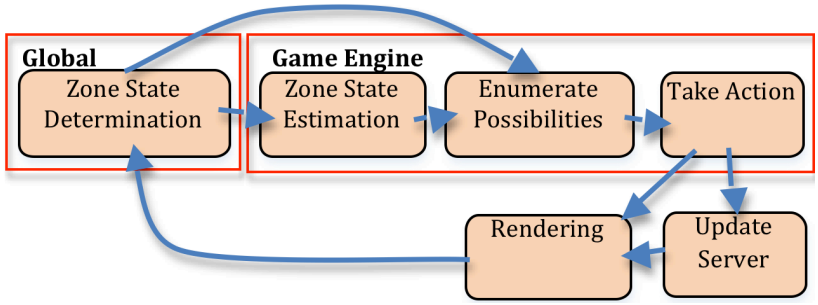
Real-time interactive applications, such as cloud games, are vulnerable to erroneous game states, which are due to network/hardware delay and unreliability. Exposing more lower-level system information to cloud games allows game developers to address quality/cost tradeoffs and prioritize contents in games. Questions that naturally arise include: (a) What is the degree of system/infrastructure awareness required to adequately execute latency-sensitive online games in an outsourced setting? (b) To what extent must online games be aware of the underlying latencies in the network/devices? To answer these two questions, we must determine the factors that will influence game behavior and outcome. Parameters that a game designer may vary to study game behavior exist at multiple levels; some of these are listed below:

- *Client* - #players, link bandwidth, and execution latency.
- *Server* - CPU consumption for game logics, server architecture, and access bandwidth.
- *Network* - topology, access networks, and protocols.
- *Game* - game actions/objectives and scenes.

An understanding of these parameters and their correlations can help determine multiple performance metrics can be used to capture the efficacy of cloud games. Capacity and scalability metrics on the server and network side can help the game designer with proper sizing of required resources for game deployment. Game playability metrics determine whether the game will be attractive to users. A game designer can now associate the metrics with suitable system parameter values for an online game; cost and deployment constraints dictate how these parameters at different levels can be varied to achieve the desired playability metric values. A game scene, for example, can be modified with fewer bots and less dynamic scenes if the responsiveness is not at

the expected level. Given many alternatives to achieving certain playability targets, the challenge faced by the game designer is that of choosing the one that best suits the purpose of the game and the desired player experience bearing in mind cost, scalability, and resource constraints. The interplay between the allocated resources (e.g., CPU and bandwidth utilization) and its impact on QoE metrics is not well understood.

Referring to the flow model below and playability parameters, the average response time can be considered to be the measure of the *responsiveness* of the game. *Fairness* can be modeled as the difference between zone states across all clients for a given zone at a given instance of time. This can be derived from the periodic logs of the zone info. The difference between the client zone state and the server global zone state gives us the measure of *precision* in the game. Depending on the type of game being played, the values for each of those three performance parameters will vary. Some require a higher precision than others, some require higher responsiveness either in actions or in communications. Knowledge of relevant metrics at different levels is required to design optimized state prediction and resource allocation strategies. [CHE2011]



While local resources at the game device or a machine in its local network can be effectively utilized to reduce operational latencies, creation of a consistent global view in the virtual environment is a task that is more effectively performed at a sever where input information from multiple users are collated. We argue that public clouds are likely to be very effective creating reasonably accurate global snapshots of the virtual environment, especially when there are a large number of users.

We want to study the techniques for scaling to a very large number of end users. In particular, we want to develop scalable techniques for task partitioning within the public cloud, that will determine the exact specifics of how many public cloud instances are needed and how gaming tasks will be mapped. This work will build on our prior work and existing approaches for big data processing and warehousing in datacenters. We argue that the computation of an optimized allocation of the underlying resources across users is well suited to problems that typically execute in datacenters, using existing distribution platforms such as MapReduce. Our key intuition is to (a) exploit locality in the virtual space and group users that require information about each other in order to make progress in the activity and (b) exploit social connections to communicate with others involved in the activity. While the former set can yield a straightforward mapping of users to groups based on current positions, creating and maintaining a view of a social group is more complex since it can involve obtaining information about other users in the vicinity. This implies that the subset of users in the view to be created is varied.

[SAT2009] M. Satyanarayanan, P. Bahl, R. Cceres, N. Davies, “The Case for VM-Based Cloudlets in Mobile Computing,” PerCom, 2009.

[COS2012] P. Costa, M. Migliavacca, P. Pietzuch, and A. Wolf, “NaaS: Network-as-a-Service in the Cloud,” Hot-ICE, 2012.

[RAH2013] M. Rahimi, N. Venkatasubramanian, A. Vasilakos, “MuSIC: On Mobility-Aware Optimal Service Allocation in Mobile Cloud Computing," CLOUD, 2013.

[CHE2011] P. Chen and M. El Zarki, “Perceptual View Inconsistency: An Objective Evaluation Framework for Online Game Quality of Experience (QOE),” NetGames, 2011.