MASSIVIZING ONLINE GAMES USING CLOUD COMPUTING: A VISION

Alexandru Iosup[1], Siqi Shen[1], Yong Guo[1], Stefan Hugtenburg[1], Jesse Donkervliet[1], Radu Prodan[2]

[1]Delft University of Technology, Delft, the Netherlands. Contact: A.Iosup@tudelft.nl
[2]University of Innsbruck, Austria. Contact: Radu@dps.uibk.ac.at

ABSTRACT

Online gaming, a large market with hundreds of millions of active players, is still struggling to scale without risky investments in infrastructure. In this work, we propose a cloud-based platform to *massivize* online gaming—the challenges and opportunities of scaling on-demand, while paying only for what is used. We discuss the major aspects of cloud-based gaming, virtual-world management, game-data processing, and game-content generation.

1. INTRODUCTION

Online gaming systems are already providing services to an increasing player base, but also to enterprise training, disaster-scenario analysis, and education. Although the current approach services overall hundreds of millions, the predominant industry approach is self-hosting, that is, to buy and operate large-scale infrastructure. This practice cannot scale when the number of players per game surges, is too costly when the number of players decreases, and is too restrictive for small companies due to up-front costs. In recent years, when many self-hosted games were shut down or not even deployed, a new approach has emerged: *cloud-based gaming*, that is, hosting games on an elastic set of resources, provisioned flexibly, when needed, only for as long as needed, and only paying for what is consumed, from Infrastructure-as-a-Service (IaaS) clouds. In this work, we discuss the challenges and opportunities of hosting games in the cloud.

According to a recent survey in `games`, Online Games (OGs) entertain over 1.2 billion online gamers in a global market of tens of billions of Euros per year. About two-thirds of the Americans play games [1], with similar numbers reported in most developed countries. Besides entertainment, the techniques developed first for online gaming are increasingly used in enterprise training and evaluation, for example using complex simulations that require cooperation across multiple continents and advanced visualizations; in disaster scenarios such as fire-fighting in crowded neighborhoods and evacuation of large-scale disaster areas; and education, for example in popular Massive Open Online Courses.

The designers of OG systems are increasingly searching for ways to scale their games, thus being able to provide service to an ever-increasing user-base. However, modern OGs raise difficult scalability challenges. The technology is often limiting the most interactive designs, such as the fast-paced first person shooter (FPS) games, to a few tens of players per game instance. Managing the data corresponding to tens of millions of players is vital for the operation of successful games, yet raises complex challenges even for slow-paced analytics. Generating player-customized content is done only on a small-scale, even by the most popular and big budgeted games. Collectively, and when the user-based is massive, these challenges are part of the family of challenges related to *massivizing online games*.

We argue in this vision paper that the traditional OG self-hosting technique is failing but for a few companies, and that cloud computing is already and will increasingly be part of the solution to massivizing online games. Although the industry numbers tens of thousands of successful OGs, operated by hundreds of companies, the market is in constant turmoil. Because game popularity is fashion-driven, few of the games can operate longer than several years. This volatility increases the risk of investment and also means that newcomers have a technological advantage, as the ICT infrastructure advances and becomes cheaper. But only for so long. Thus, the traditional approach of self-hosting has become increasingly riskier and ephemeral; a number of traditional companies have shut down entire games when they became unprofitable, and others have postponed deployment.

In contrast to self-hosting, cloud computing offers the mechanisms to lease resources and ICT services elastically, growing and shrinking with the user base, and paying (nearly) only for the resources actually consumed. The basic use of cloud computing for online gaming has already been explored. For example, Zynga pioneered hosting its social games in its hybrid private-public cloud infrastructure; OnLive has hosted an entire portfolio of games developed and published elsewhere, streaming to its users only the video output of the game; various researchers have looked at hosting models [2, 3]; etc. Beside the traditional challenges inherited from non-cloud-based online games, for example scalability- and latency-related, cloud-based gaming introduces new challenges and opportunities such as elastic resource scheduling. In this work, we discuss some of these challenges and identify various opportunities.

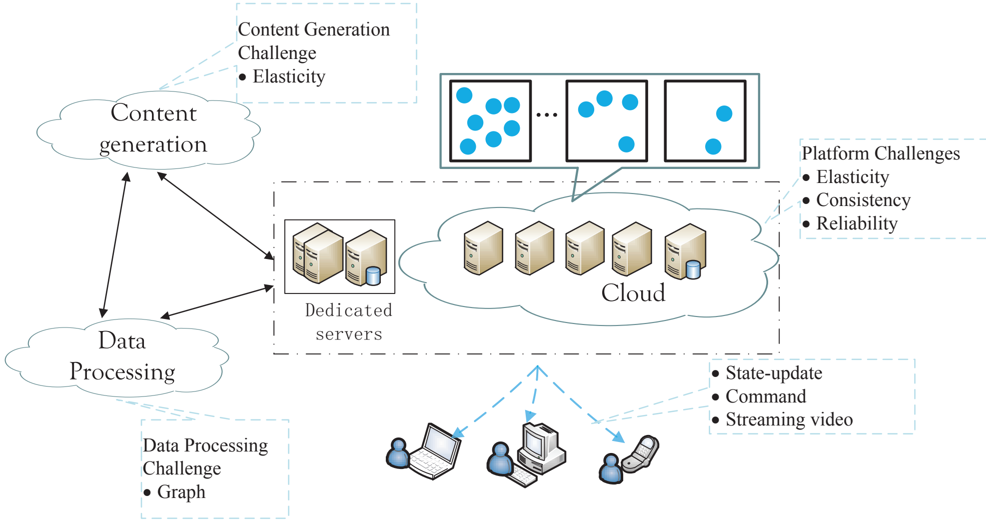

Fig. 1. A generic cloud-based platform for massivizing online games. The three pillars are virtual-world management, game-data processing, and game-content generation.

2. A CLOUD-BASED PLATFORM TO MASSIVIZE ONLINE GAMING

In this section we discuss the requirements and the design of a generic platform to massivize OGs.

2.1. Requirements for Cloud-Based Gaming Platforms

We identify three main requirements:

1. **Virtual-World Management**: The cloud-based game platform should be scalable to serve millions of players online, match elastically the number of players, be always available, and be consistent and low latency.

2. **Game-Data Processing**: The massive number of players in the virtual world leads to massive amounts of data: user interactions, uploaded screenshots and videos, social networking, etc. Analyzing these data can help the system designers to understand player behavior and to gain insight into system operation, thus allowing them to build better games for the players. Cloud-based data processing could enable an elastic, and thus efficient, platform for time-based and graph analytics.

3. **Game-Content Generation**: Game content, from bits such as textures to abstract puzzles and even entire game designs, is at the core of the entertainment value of games. Until the early 2000s, manual labor has ensured that the quality and quantity of game content matched the demands of the playing community, but is not scalable due to exponential growth in number of users and production costs. A cloud-based game platform providing elastic, cost-effective, procedural generation of player-customized content, could lead to a vast improvement over the capabilities of today's generation of games.

2.2. Cloud-Based Platform Architecture

To address the three main requirements, we propose the generic architecture of a cloud-based platform depicted in Figure 1. The architecture is based on three pillars: *virtual-world management*, *game-data processing*, and *game-content generation*. Responding to Requirement 1, The *virtual-world management* pillar addresses game hosting and the management of players in the virtual world. The *game-data processing* pillar addresses Requirement 2: it analyzes time-based and graph data corresponding to players and their games. Focusing on Requirement 3, the *game-content generation* pillar generates player-customized content at massive scale. The virtual-world management pillar provides in-game data to the game-data processing pillar and uses content produced by the game-content generation pillar.

We describe the challenges and opportunities of the virtual-world management pillar in Section 3, and focus on the other two pillars in the remainder of this section. We focus on systems challenges; others, such as finding new ways to use data and to generate content, fall outside the scope of this work.

Challenge 1: Collecting and processing time-based data in the cloud. Using elastic cloud resources can facilitate game operators and analysts to mine game data under different processing scenarios. In our previous work, we have designed the CAMEO [4] platform for Massively Multiplayer Online Game (MMOG) data analysis. By dynamically leasing and releasing computational and storage resources from cloud environments, CAMEO is able to support regular periodical analysis (e.g. daily workload distribution) and special burst processes (e.g. popular in-game events with massively participants). However, processing time-based data in workflows, that is, sets of inter-dependent tasks, remains challenging [5].

Challenge 2: Processing graph data in the cloud. Social data (friendships, team-mates, communities, etc.) is commonly collected in games. Social data can be easily presented as graphs, whose vertices represent game entities (e.g., players) and edges represent relationships (e.g., "is part of the same clan with"). In our previous work [6], we have evaluated the performance of several popular graph-processing systems when processing gaming graph data, and found that the performance of different data-processing applications can be very different, even for the same dataset. With performance characterization still elusive, statical deployments can be wasteful or insufficient, and provisioning from clouds the right amount and type of resources is challenging.

Challenge 3: Procedural Content Generation for Games (PCG-G) in the cloud. PCG-G automates or aids in the generation of game content, from game bits to entire designs and even derived game content [7]. PCG-G is already difficult, since the algorithm has to generate content, satisfy constraints imposed by the artist, and return interesting instances for gamers, all within a resource budget. With the help of cloud computing, the algorithms can at least remove the risky investment in content generation, deferring it until the content is actually needed and enabling player-customization. In our previous work, we have shown that on-demand resources can generate large amounts of content [8], but much remains to be done in enabling efficient generation.

3. 10 CHALLENGES AND OPPORTUNITIES IN VIRTUAL-WORLD MANAGEMENT

To support millions of players concurrently online, the game platform should be elastic, reliable, and consistent.

Challenge 4: Using basic offloading models. Combining basic offloading models into hybrid, more efficient offloading models. In general, there are two types of offloading for cloud-based games: server-side offloading and client-side offloading. For server-side offloading, server operation is moved (offloaded) from self-hosted servers to the cloud. For client-side offloading, the clients can also offload operation to the cloud. Offloading can refer to computational, communication, and other types of workload. Cloud gaming approaches, such as [9, 10], offload the client-side *computation* and *rendering* to servers, so that the client only receives the streaming video from the servers. For *communication offloading*, the clients can let servers in the cloud download the needed information [11]. Using these basic offloading models to be efficient while maintaining game responsiveness is a challenge [12].

With hybrid offloading, only *parts* of the computation and communication loads are offloaded to the cloud. A game operator can use self-owned servers to serve its main market, while also using the cloud to service customers in other markets. The main challenge for hybrid offloading remains the concerted decision of how to partition the load between clients, self-owned servers, and cloud; of when to offload; and of which cloud resources to use for each offloaded part.

Challenge 5: Using basic cloud deployment models. Combining basic cloud deployment models. In the traditional model, games run on self-owned dedicated server/servers cluster. The main inefficiency and risks associated with this model can be alleviated through the deployment of a *private cloud* on this self-owned infrastructure, which can also serve external customers, thus gaining economies of scale and revenue even if the gaming unit has a temporary slowdown. The *public cloud* model only uses resources leased from public IaaS clouds, which is easy to operate but can lead to vendor lock-in, that is, to the problem of being tied, through code and data, to a single cloud provider. Using these models has already explored [2, 3, 13]. However, the provisioning and allocation of resources can still raise challenges for new game types and services.

The *hybrid cloud* model uses both self-owned infrastructure and public IaaS clouds, promising to save costs by running necessary tasks in self-owned clusters, and only leasing cloud resources for unexpected load and for access to emerging markets. The *sharded cloud* model is an extension of any other cloud model, where the deployments are essentially non-communicating clusters (shards); this leads to excellent scalability at the expense of segregated in-game worlds. Using hybrid clouds efficiently and extending the size of each shard remain very challenging issues.

Challenge 6: Efficient provisioning and allocation of resources. How to provision the right amount of the right resources to the right place? Through *horizontal scaling* [3, 2], new compute machines are leased from the cloud and lead to increased service capability; through *vertical scaling*, the same machine is loaded more, thus utilizing its full capacity. Hybrid resource scheduling, which can make use of both horizontal scaling and vertical scaling, remains a challenge.

Challenge 7: Cost-aware provisioning and allocation of resources. Typical for an IaaS cloud provider, Amazon EC2 offers three billing models: on-demand, spot, and reserved. How to use efficiently each billing models? Our previous work focuses on using on-demand and reserved instances, to reduce the cost of cloud hosting [14]. We also propose an ecosystem [13] for hosting OGs, which effectively splits the traditional monolithic MMOG companies into three main service providers: game providers, game operators, and resource providers. This ecosystem can help the small, medium enterprizes enter the MMOG market with low startup cost. Using a combination of billing models may further reduce operational costs, but remains challenging.

Challenge 8: Socially aware provisioning and allocation of resources. Players like to play with their friends [15]. How to use the explicit or implicit social structure of games to provision and allocate resources to improve the user experience? For social network game such as FarmVille, it is more efficient to place a group of friends who interact frequently in a same server. There are some possible directions to improve the socially-aware scheduling, skill-aware which takes into the players' skill into account, location-aware which try to place the players with close-proximity, or combination of those two.

Challenge 9: Deciding on the consistency-scalability trade-off. Achieving consistency in distributed system is difficult, and the distributed servers of OGs are particularly challenging [16] [17]. State needs to be replicated for many users to see, and game state changes not only due to user-initiated actions but also with the advance of in-game time [18]. Although early work exists on the consistency-scalability trade-off, such as using dead-reckoning [19] and distributed transactions [20], and varying consistency by scenario [21], we believe this challenge remains open.

Challenge 10: 100% availability for OGs, using cloud resources and agreements. Although reliability, availability, and similar non-functional system properties have received much attention in distributed computing, they have been less studied in the context of online gaming. Due to good management and to promising service level agreements (SLAs), clouds may eventually tolerate machine failures without losing data or annoying users. We have already looked [22] at mechanisms to help cloud operators to provide good service, although other considerations (e.g., low penalties) may actually entice cloud operators to cut service occasionally. However, much remains to be done to address this challenge: designing new availability mechanisms for OGs, building high-availability servers for OGs, etc.

4. TOWARD SOLVING THE CHALLENGES

OGs can greatly benefit from the leasing and releasing of cloud resource. In this work, we have presented a generic cloud-based gaming architecture, and ten of its challenges. The future belongs to our passionate gaming community.

5. REFERENCES

[1] The Entertainment Software Association, "Essential facts about the computer and video game industry: Sales, demographics, and usage data," Jul 2013.

[2] V. Nae, A. Iosup, and R. Prodan, "Dynamic resource provisioning in massively multiplayer online games," *IEEE TPDS*, no. 3, pp. 380–395, 2011.

[3] Y.-T. Lee and K.-T. Chen, "Is server consolidation beneficial to mmorpg? a case study of world of warcraft," in *CLOUD*, 2010.

[4] A. Iosup, A. Lascateu, and N. Tapus, "Cameo: Enabling social networks for massively multiplayer online games through continuous analytics and cloud computing," in *NETGAMES*, 2010.

[5] T. Hegeman, B. Ghit, M. Capota, J. Hidders, D. H. J. Epema, and A. Iosup, "The btworld use case for big data analytics: Description, mapreduce logical workflow, and empirical evaluation," in *BigData Conference*, 2013, pp. 622–630.

[6] Y. Guo, M. Biczak, A. L. Varbanescu, A. Iosup, C. Martella, and T. L. Willke, "How well do graph-processing platforms perform? an empirical performance evaluation and analysis," in *IPDPS*, 2014.

[7] M. Hendrikx, S. Meijer, J. V. D. Velden, and A. Iosup, "Procedural content generation for games: A survey," *TOMCCAP*, vol. 9, no. 1, p. 1, 2013.

[8] A. Iosup, "Poggi: generating puzzle instances for online games on grid infrastructures," *Concurrency and Computation: Practice and Experience*, vol. 23, no. 2, pp. 158–171, 2011.

[9] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere: an open cloud gaming system," in *MMSys*, 2013.

[10] M. Hemmati, A. Javadtalab, A. A. Nazari Shirehjini, S. Shirmohammadi, and T. Arici, "Game as video: Bit rate reduction through adaptive object encoding," ser. NOSSDAV 2013.

[11] S. Hu, C. Wu, E. Buyukkaya, C.-h. Chien, T. Lin, M. Abdallah, J. Jiang, and K. Chen, "A spatial publish subscribe overlay for massively multiuser virtual environments," in *ICEIE*, 2010.

[12] A.-C. Olteanu, N. Tapus, and A. Iosup, "Extending the capabilities of mobile devices for online social applications through cloud offloading," in *CCGRID*, 2013, pp. 160–163.

[13] V. Nae, R. Prodan, A. Iosup, and T. Fahringer, "A new business model for massively multiplayer online games," in *ICPE*, 2011.

[14] S. Shen, K. Deng, A. Iosup, and D. H. J. Epema, "Scheduling jobs in the cloud using on-demand and reserved instances," in *Euro-Par 2013*.

[15] A. Iosup, R. van de Bovenkamp, S. Shen, A. L. Jia, and F. A. Kuipers, "An analysis of implicit social networks in multiplayer online games," *IEEE Internet Computing*, 2014.

[16] B. Knutsson, H. Lu, W. Xu, and B. Hopkins, "Peer-to-peer support for massively multiplayer games," in *INFOCOM*, 2004.

[17] D. Ahmed and S. Shirmohammadi, "Zoning issues and area of interest management in massively multiplayer online games," in *Handbook of Multimedia for Digital Entertainment and Arts*, 2009.

[18] M. Mauve, J. Vogel, V. Hilt, and W. Effelsberg, "Local-Lag and Timewarp: Providing Consistency for Replicated Continuous Applications," *IEEE Multimedia*, vol. 6, no. 1, 2004.

[19] Y. Bernier, "Latency compensating methods in client/server in-game protocol design and optimization," in *Game Developers Conference*, vol. 98033, no. 425, 2001.

[20] M. T. Najaran, S.-Y. Hu, and N. C. Hutchinson, "Spex: Scalable spatial publish/subscribe for distributed virtual worlds without borders," ser. MMSys 2014.

[21] L. Krammer, G. Schiele, D. Koch, and C. Becker, "Quality of experience-aware event synchronization for distributed virtual worlds," in *ICPADS*, 2012, pp. 604–611.

[22] V. Nae, L. Köpfle, R. Prodan, and A. Iosup, "Autonomous massively multiplayer online game operation on unreliable resources," in *C3S2E*, 2013, pp. 95–103.