Massively Multiplayer Online Games on Unreliable Resources

Vlad Nae, Lukas Köpfle, Radu Prodan
University of Innsbruck,
Technikerstr. 21a, A-6020 Innsbruck, Austria

Alexandru Iosup
Delft University of Technology,
Mekelweg 4, 2628 CD, Delft, The Netherlands

***Abstract*—We have previously introduced a cloud-based model for Massively Multiplayer Online Games (MMOGs) operation, which allows small and medium enterprises to join the MMOG market through near-zero initial infrastructure investment.**

In this work we propose an architectural extension to this MMOG operational model designed to increase and guarantee the MMOG session availability through fault tolerance mechanisms.

I. Introduction

MMOGs are a new type of large-scale distributed applications characterised by a real-time virtual world entertaining millions of players spread across the globe. To comply with the variable computational and latency-aware resource demands of MMOGs, the MMOG operators over-provision an own multi-server infrastructure with sufficient capabilities for guaranteeing the *Quality of Service (QoS)* requirements and a smooth game play at all times. This statically provisioned infrastructure has two major drawbacks: it has high operational costs and is vulnerable to capacity shortages in case of unexpected increases in demand.

In contrast to static provisioning, the new cloud computing technology based on resource virtualisation has the potential to provide an on-demand infrastructure for MMOGs, where resources are provisioned and paid for only when they are actually needed. Conversely, this technology can introduce virtualisation overheads which may cancel out the benefits. In previous work we studied the consequences of using virtualised resources regarding the technology-incurred overheads and their effects on the QoS offered to the clients [1] and the economical consequences [2], while considering ideal resources in terms of availability.

In this work we propose an architectural extension to the cloud-based MMOG operational model, introduced in [1], designed to increase and guarantee the MMOG session availability through fault tolerance mechanisms.

II. Architecture

We propose a three tiered architecture composed of game session and resource management services supporting and steering the execution of MMOGs. Our MMOG platform, depicted in Figure 1, consists of three actors, each fulfilling distinct roles, namely:

1) the *client* which connects to the game operator's MMOG sessions;
2) the *game operator* that provisions resources from the resource provider and ensures the proper execution of the MMOG sessions;
3) the *resource provider* which offers the physical or virtualised machines on which the game servers will run.

Resource providers are scattered around the world and pool together resources that may serve multiple game operators simultaneously. Similarly, there are geographically distributed game operators offering MMOG titles to clients and ensuring proper game operation by allocating resources from providers.

A. Clients

Clients can join MMOG sessions offered by game operators on the basis of MMOG subscriptions. An MMOG subscription represents a contract between a client and a game operator based on which the client is allowed, under certain terms and with certain QoS guaranties, to join a MMOG session managed by the game operator.

B. Game operators

The game operators interact with clients and offer them a selection of MMOGs, usually by contracting new games from game development companies (interaction not modelled in this work). The operators execute distributed MMOG sessions with guaranteed QoS comprising interconnected MMOG servers. The game operator runs four main services.

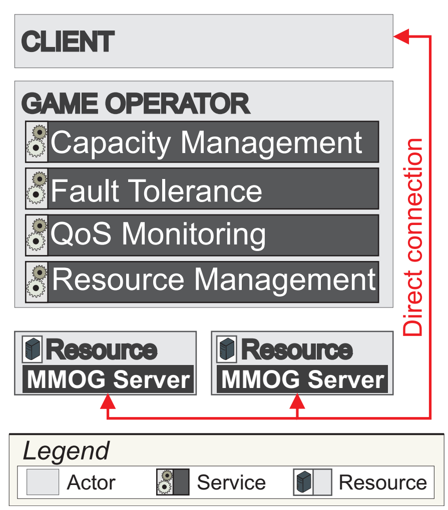

Fig. 1. MMOG ecosystem architecture.

1) The capacity management service: Using a game state prediction mechanism (i.e. distribution of clients' avatars within the game world), the capacity management service estimates the resource requirements for short-medium time intervals (order of minutes). Based on these load estimates, it instructs the resource management service to provision the correct amount of resources in the next allocation cycles. It also decides if MMOG server start/stop/migrate -actions are necessary to accommodate and balance the generated load. For example, by timely foreseeing critical hot-spots in the game world (i.e. excessively populated areas of the game world generating a large number of interactions), one can dynamically provision additional MMOG servers on newly leased resources and take proactive load balancing actions that redistribute the game load before the existing game servers become overloaded. We investigated viable approaches to capacity requirement estimation in [3].

2) The fault tolerance service: is designed to ensure a high level of tolerance to resource faults which can lead to low QoS or even MMOG session unavailability. Although the utilised resources are from commercial Cloud providers which promise relatively high levels of availability, they are nevertheless subject to a multitude of unexpected events which can lead to failures. The fault tolerance service is responsible for maintaining the MMOG sessions' integrity in case of faults appearing in resources which host MMOG servers, such as independent or correlated machine failures, and local or Cloud site network connectivity loss. These events result in disruptions in game play which the fault tolerance service can minimise, or even completely hide to the clients by timely taking the appropriate counter measures: redistributing the clients connected to a failing server to others within the same MMOG session (transparent measure), or starting other MMOG servers on new resources, in tandem with the resource management service (minimally disrupting measure).

3) The QoS monitoring service: collects and analyses information about the state of the MMOG sessions and the QoS delivered by the running game servers. The QoS metrics, like client update frequencies, utilised memory and network bandwidth, and average client-server connection latency are gathered from the MMOG servers. The QoS monitor analyses the collected information and aggregates it into monitoring reports which are then utilised by the other services.

4) The resource management service: interacts with the resource providers, negotiating for the resources best fitting the capacity requirements provided by the capacity management service. The negotiation process takes into account multiple resource parameters: computational power, amount of memory, network capacity, geographical location in relation to the requests, overheads introduced by the virtualisation technology and price. It also provides the necessary low-level mechanisms for game session management, such as MMOG server start/stop/migrate -actions, as well as high-level paralellisation mechanisms, such as zoning, replication, and instancing which distribute the client-generated load among computing resources. In previous work we thoroughly analysed some of the challenges this operator-provider interaction raises: the effects of dynamic resource allocation for MMOG hosting [4] and the operator-provider resource negotiation process [2]. In consequence, these aspects of MMOG operation will not be further detailed in this work.

The game operator is also responsible for running a persistence service which ensures the continuity of the MMOG session throughout the lifetime of the game, but given that all MMOG operators already utilise advanced fault tolerance mechanisms for this particular service, we leave this aspect outside the scope of our investigation.

C. Resource providers

We consider Cloud resource providers which employ the paradigm of Infrastructure provided as a Service (IaaS) and offer resources for fine grained time intervals (i.e. hours) though a virtualisation platform that allows automated software deployment and maintenance. The resource providers lease virtual machines with fuzzy definitions of their characteristics, but with much more precise guaranties in terms of resource availability. For example, for the processor performance, Amazon [5] employs "EC2" units defined as "the equivalent CPU capacity of a 1.0-1.2 Gigahertz 2007 Opteron or Xeon processor" and FlexiScale [6] uses the "vCPU unit" which represents a computational unit of unknown power associated to 0.5 Gigabytes of memory. Conversely, the availability of an Amazon resource is defined as 99.95% over a period of one year and Flexiscale promises 100% availability over one month, along with an additional provision that, should a resource fail, the recovery time will be limited to 15 minutes.

III. Ongoing Work

We are currently in the process of evaluating the viability of the proposed MMOG operational model on real-world, failing cloud resources by analysing a set of OoS metrics in different resource availability scenarios. In particular we aim to evaluate the impact of employing limited availability resources on the game-play experience of the clients. We are also investigating different resource allocation policies for alleviating the negative effects of resource failures have on the MMOG sessions.

Acknowledgments: This work is partially founded by the FWF Austrian Science Fund. Project number TRP 72-N23.

References

[1] V. Nae, A. Iosup, R. Prodan, and T. Fahringer, "The impact of virtualization on the performance of Massively Multiplayer Online Games," in *NetGames'09*, 2009.

[2] V. Nae, R. Prodan, A. Iosup, and T. Fahringer, "A new business model for Massively Multiplayer Online Games," in *Proceeding of 2nd WOSP/SIPEW international conference on Performance engineering*. ACM, 2011, pp. 271–282.

[3] R. Prodan and V. Nae, "Prediction-based real-time resource provisioning for massively multiplayer online games," *Future Generation Computer Systems (FGCS)*, vol. 25, no. 7, pp. 785–793, 2009.

[4] V. Nae, A. Iosup, S. Podlipnig, R. Prodan, D. Epema, and T. Fahringer, "Efficient management of data center resources for Massively Multiplayer Online Games," in *Proceedings of the ACM/IEEE conference on Supercomputing*. IEEE, 2008, pp. 1–12.

[5] Amazon Inc., "Amazon Elastic Compute Cloud (Amazon EC2)," http://aws.amazon.com/ec2/, May 2012.

[6] FlexiScale, "Utility computing on demand," http://flexiscale.com/, February 2012.