Multimedia Systems (2014) 20:521–544
DOI 10.1007/s00530-013-0352-y

SPECIAL ISSUE PAPER

SLA-based operations of massively multiplayer online games in clouds

Vlad Nae · Radu Prodan · Alexandru Iosup

Published online: 5 January 2014

Abstract Successful massively multiplayer online games (MMOGs) have today millions of registered users and hundreds of thousands of active concurrent players. To be able to guarantee quality of service (QoS) to a highly variable number of concurrent users, game operators statically over-provision a large infrastructure capable of sustaining the game peak load, even though a large portion of the resources is unused most of the time. To address this problem, we introduce in this work a new MMOG ecosystem for hosting and provisioning of MMOGs which effectively splits the traditional monolithic MMOG companies into three main service providers: game providers, game operators, and resource providers. Their interaction is regulated through comprehensive service level agreements (SLA) that establish the price, terms of operation, and compensation for service violations. In our model, game operators efficiently provision resources for MMOGs from multiple cloud providers, based on dynamic load forecasts, and ensure proper game operation that maintains the required QoS to all clients under varying resource availability. Game providers manage multiple distributed MMOGs for which they lease services under strict operational SLAs from game operators to satisfy all client requests. These three self-standing, smaller, more agile service providers enable access to the MMOG market for the small and medium enterprises, and to the current commercial cloud providers. We evaluate, through simulations based on real-life MMOG traces and commercial cloud SLAs, the impact of resource availability on the QoS offered to the MMOG clients. We find that our model can mitigate the negative effects of resource failures within four minutes and that MMOG server consolidation can accentuate the negative effects of failures in a resource-scarce environment. We further investigate different methods of ranking MMOG operational offers with either single or multiple (competing) MMOG providers. Our results show that compensations for SLA faults in the offer selection process can lead up to 11–16 % gain in the game providers' income. We also demonstrate that adequate ranking of offers can lead to MMOG operational cost reductions from 20 up to 60 %.

Keywords MMOG · Cloud computing · Fault tolerance · QoS · SLA

V. Nae · R. Prodan (✉)
Institute of Computer Science, University of Innsbruck, Innsbruck, Austria
e-mail: radu@dps.uibk.ac.at

V. Nae
e-mail: vlad@dps.uibk.ac.at

A. Iosup
Faculty of Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands
e-mail: A.Iosup@tudelft.nl

1 Introduction

Massively multiplayer online games (MMOGs) are a new type of large-scale distributed application characterised by seamless virtual worlds in which millions of world-wide players interact in real-time. Although, in the past decade, the number of MMOG players has grown exponentially to the current tens of millions, growth may now hamper the progress of this important branch of the entertainment business. Figure 1a shows a rapid increase in the peak number of concurrently connected players to the first five most popular MMOGs in Asia in the past 8 years, according to a survey conducted by [1]. The biggest Asian

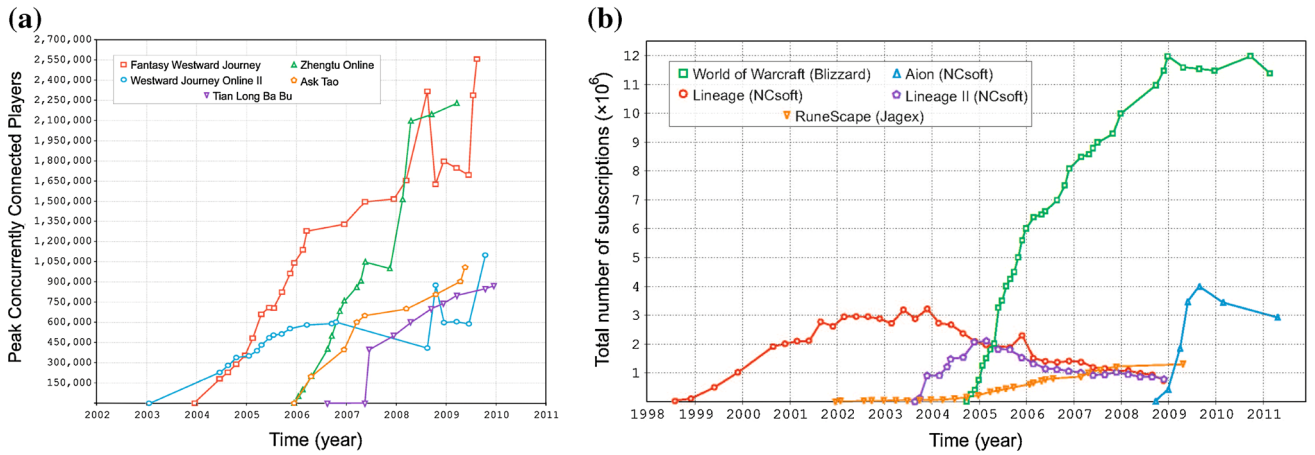

Fig. 1 Top-five MMOGs in number of subscribed and concurrent users. **a** Top-five Asian MMOGs in the peak number of concurrent users. **b** Top-five MMOGs in the total number of subscriptions [1]

MMOG reached a record of 2.5 million concurrent players in 2009, while the second biggest MMOG registered a jump of one million in the number of concurrently connected players within only 4 months (December 2008–March 2009).

Today, most MMOG companies have to be both *game providers* by investing in the creative part of games, and *game operators* by purchasing and managing an over-provisioned multi-server infrastructure (to comply with the *quality of service* (QoS) requirements of players generating highly variable computational and latency-sensitive network demands [2]), using for its operation up to 40 % of the total game revenue in an annual market of over 24 billion dollars. For example, leading MMOGs companies such as Blizzard (developer of World of Warcraft) and Jagex Ltd. (RuneScape) own and operate tens of thousands of cores in hundreds of physical locations across all continents. The MMORPG World of Warcraft has operational costs of over $50 million per year. An important reason for this approach is the lack of middleware and business models to enable outsourcing the operation of MMOGs, which should include comprehensive methods for specifying and negotiating operational terms, responsibilities, and risk-related penalties. Another downside of this approach is the high initial investment in purchasing and running the data centres required to join the MMOG market.

Today, *cloud computing* promises to solve the infrastructure problems of the MMOG ecosystem through on-demand resource leasing under well-defined *service level agreements* (SLAs). By leveraging the new on-demand resource provisioning model, companies such as MMOG operators may avoid the large costs of buying and maintaining hardware, and the rapid deprecation of hardware investments. We have tackled in [2, 3] many of the technical challenges of on-demand provisioning and allocation of virtualised cloud resources to MMOGs under QoS constraints.

To enable cloud use, operational MMOG models and middleware must account for complex SLA and operational conditions, which raises numerous challenges and motivates this work. Mapping SLAs to a specific application domain remains challenging [4], although SLAs are a classic and well-studied [5, 6] mechanism for specifying and managing strict user requirements in distributed systems. As a consequence, working middleware and adequate modeling tools for MMOGs do not yet exist. Selecting among multiple cloud providers with different and complex SLAs remains a distributed systems' challenge [7, 8], in particular raising the challenge of mapping various business-level SLAs to the real-time QoS requirements of MMOGs. Adapting to varying resource availability due to variable performance [9] and zonal (correlated resource) blackouts that occur even for large commercial cloud providers, such as Amazon and Microsoft, is also crucial.

To address these QoS-related challenges, we introduce in this work a new *MMOG ecosystem* for hosting and operating MMOGs which effectively splits the traditional monolithic MMOG companies into three lighter and more focussed service providers: game providers, game operators, and resource providers. The proposed ecosystem regulates their interaction through comprehensive SLA negotiation protocols that establish the price, terms of operation, and compensation for service violations as *penalties*. In our model, game operators efficiently provision cloud resources for MMOGs based on their dynamic load and ensure proper game operation that maintains the required QoS to all clients. Game providers lease operation SLAs from the game operators to satisfy all client requests and manage multiple distributed MMOG sessions.

Our new ecosystem enables faster access to the MMOG market for the small and medium enterprises, and to the current commercial cloud providers. It equips game operators with a clear specification of responsibility and of penalties associated with risks of unavailability, lower performance, and other QoS violations. It comprehensively defines the roles of each service provider in the ecosystem, such that they can be fulfilled by self-standing, smaller, more agile service providers. This comes in contrast to today's practice that considers game providers to also be operators, and even worse, resource providers. We show in this work that it is possible for our comprehensive yet seemingly complex model to be useful in realistic scenarios, and in particular in relationship with a realistic multi-cloud, multi-priced, dynamically available resources.

Our SLA-based middleware requires few changes to current commercial MMOG deployments, as we do not introduce game client changes and only require the game operators to update their operational approach with minimal intrusion in their deployed gaming platforms. Most importantly, we do not require resource operators, such as IaaS clouds, to change their underlying infrastructure to support our negotiation protocols. Unlike traditional SLAs considered in grids (e.g. SNAP [7], OGSA, WSRF [10]) and other distributed systems (e.g. Galaxy [11], Oceano [12]), our proposed SLAs include a comprehensive specification of compensations for temporary QoS violations. Thus, our middleware addresses transparently the new challenges in SLA specification and negotiation introduced by the stringent QoS requirements and the dynamic nature of MMOGs. Recently, SLAs adapted to IaaS clouds have been used by the online gaming company Zynga, but without a publicly available middleware stack and only for games that, as essentially non-communicating web applications, have much lower and less diverse QoS requirements than MMOGs. Several research studies related to games [3, 13, 14], including our pioneering work in the field of cloud-based MMOGs [15] and of SLAs that include penalties [3], still lack much in comprehensiveness and realism.

The remainder of this article is structured as follows. Section 2 presents the MMOG, business ecosystem, and QoS models underneath our approach. The following three sections contain the main contributions of our work:

1. We propose in Sect. 3, a new ecosystem that separates the problem of provisioning and operation of MMOGs between two business actors: *game providers* offering MMOGs to the clients and *game operators* leasing appropriate resources from cloud providers to fulfil QoS requirements and, as a novel contribution, considering various elements that affect resource availability as QoS parameters.
2. We model in Sect. 4, the MMOG QoS parameters as SLA terms and design a negotiation protocol between the game providers and the game operators which includes fitness-based ranking of SLA offers and penalties (compensations for QoS violations);
3. Based on simulations using real MMOG traces and multi-cloud, multi-class, dynamically available resources from ten commercial cloud providers described in Sect. 5, we analyse in Sect. 6:
   - the impact of resource availability on the MMOG operation in isolated and competition-based environments and formulate best practices guidelines towards game operators for improving client's QoS;
   - the impact of SLA compensation and fitness ranking methods on the game provers' income in both isolated and competition-based environments, and formulate again best practices guidelines oriented towards maximising providers' income and maintaining clients' QoS.

The paper ends with a related work summary in Sect. 7 and concluding remarks in Sect. 8.

2 Model

In this section, we introduce the computational, business ecosystem, and QoS models underneath our approach.

2.1 Computational model

Online games can be seen as a collection of networked *game servers* that are concurrently accessed by a number of *players* (or *clients*). Clients connect directly to one game server, send their play actions (e.g. movements, collection of items, shooting), and receive appropriate responses. Each player is mapped to one *avatar* located at precise coordinates in the game world. Based on the actions sent, the avatar dynamically interacts with other avatars within a *game session*, influencing each others' state. The state updates must be delivered within a given time frequency to ensure a smooth and responsive experience.

Most MMOGs share the following computational model. The game server runs a loop in which the state of all entities is first computed and then broadcast to the clients. All entities within a specific avatar's area of interest are considered to be interacting with it and have an impact on its state. The load of the game server is proportional to the population size in the avatars' areas of interest and to the number of interactions between entities. An overloaded game server delivers state updates to its clients at a lower frequency than the players expect, which makes the overall

environment fragmented and unplayable; players may quit if similar games with better service exist in the market. Depending on the game, typical update frequencies for a fluent play must be of 1–10 Hz for most MMOGs, and even higher for other game genres.

To concurrently support millions of active players and many more other server-driven entities (non-playing characters and other game objects) with guaranteed QoS, MMOG operators provision a large static infrastructure with hundreds to thousands of computers hosting a single distributed game session. The most common game session distribution technique is "zoning", which is based on spatial partitioning of the game world into geographical zones to be handled independently by separate machines. Other techniques, such as "instancing" and "replication", divide the entities contained in a zone across several machines. Software libraries like the real time framework (RTF) [16] implement such low overhead techniques for distributed game operation under QoS constraints for both client and server. In this work, we consider the RTF library as a prototypical low-level software layer which allows for all these load distribution mechanisms.

2.2 Business ecosystem

To enable small and medium enterprises enter the MMOG market, we propose a new middleware for distributed MMOG operation and provisioning that separates the responsibilities across three different actors: game providers, game operators, and resource providers (see Fig. 2). The interaction between these actors is negotiated and regulated through bipartite SLAs, representing wrappers around QoS parameters which they agree to deliver (e.g. sustained state-update rate for a specified price). These SLA-based relationships will be discussed in detail in Sect. 3.

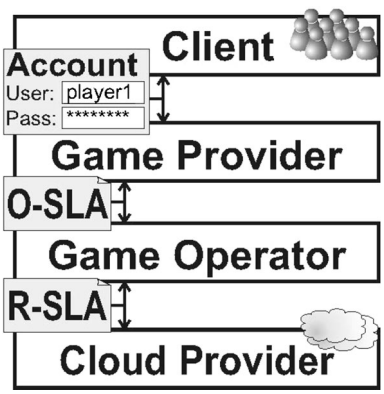

Fig. 2 Layered MMOG ecosystem

Game providers offer clients a selection of MMOGs by contracting new games from game developers (the interaction provider-developer is not covered in this work). Based on received requests, game providers assign clients to game zones, which are then delegated to game operators for QoS-based execution. The quality of gameplay is monitored by the MMOG client program and, in case of *SLA faults* (e.g. state update rate below the minimum threshold) corroborated by server-side monitoring, the client is compensated.

Game operators receive requests from the game providers for operating zones of different MMOG sessions with guaranteed QoS. Based on resource utilisation estimations covered in [2] employing a dead reckoning-based load prediction method [17], the game operators construct SLA template offers, negotiate SLAs with the game providers, and allocate resources accordingly (e.g. start new game zones, allow client connections). This interaction is detailed in Sects. 3.2 and 4. To fulfil their agreements with game providers, game operators acquire the correct amount of resources from cloud providers. At predefined *measurement timesteps* during the game play, QoS information from the MMOG servers (e.g. game loop tick rate, utilised memory and network bandwidth, client-server connection latency) is analysed. Whenever SLA faults are detected, the operators compensate the providers.

Resource providers are data centres such as the IaaS cloud providers available on the market, from which game operators lease computing and storage resources to run game servers with guaranteed QoS. We studied in [2, 3], the opportunity of employing IaaS-based cloud infrastructures for MMOG hosting with respect to the performance penalties incurred by the virtualisation overheads. In this paper, we add an essential new dimension to our previous work by considering resource availabilities and cost penalties.

2.3 QoS parameters

In the proposed MMOG operational model, we identify two principal types of failures that produce disturbances in the game play for the clients with a negative impact on the offered QoS: resource and management failures.

2.3.1 Resource failures

The game operator runs the MMOG sessions on distributed and heterogeneous cloud resources which are subject to a multitude of unexpected events that can lead to failures. If a machine crashes, hangs or becomes unreachable through the network, the MMOG server running on it is compromised disrupting the normal operation of the session. Upon the detection of this type of failure, the game operator

provisions a new resource with the same (or better) characteristics as the failing one, starts a new MMOG server which is included in the session and to which it instructs the clients to reconnect. Although the MMOG session is salvaged, the clients connected to the failing MMOG server will experience a *total interruption* in game play for a certain amount of time.

2.3.2 Management failures

The necessary amount of cloud resources for proper MMOG session operation is estimated by the game operator and provisioned from the resource provider. The main challenge in this architecture is the mapping of the stringent QoS requirements of MMOGs to SLA contracts that must be honoured at all times, which can only be enforced through best-effort mechanisms using today's resource allocation mechanisms in cloud and Internet-based infrastructures. In case of erroneous estimations or sudden surges in the number of clients, the provisioned resources are not sufficient to handle the generated load, which leads to the degradation of the QoS (i.e. fragmented, unrealistic game play) for the clients connected to the overloaded servers. In this situation, the game operator can either redistribute the clients to other MMOG servers within the same session aiming a better load distribution and a more efficient use of the provisioned resources, or it can provision more resources to include in the affected session. We call this type of disturbance, where the clients are not disconnected from the MMOG session but their game play experience is degraded, as a *partial interruption*. In previous work, we covered several aspects related to the MMOG QoS topic, such as the impact of the performance of resources [2] and of the cloud virtualisation overheads [18]. In this work, we expand our investigation by studying the effect of the two identified types of disruptions on the MMOG session QoS.

We define two important QoS metrics that define MMOG management failures. First, the *instantaneous non-interruption ratio* is the ratio between the measured state update frequency within one measurement timestep and the required minimal frequency. For example, if the minimal update frequency given by the game developer is 40 Hz and the measured update frequency is above 36 Hz in this measurement step, the instantaneous non-interruption ratio is: $\frac{36}{40} = 90\,\%$. Second, the *total non-interruption ratio* is the percentage of time the MMOG session has been accessible and the state update frequency equal or greater than the required frequency, over a given time interval (e.g. an SLA's validity time). For example, if the game operator provided, over the last 24 h, only 23.98 h of accessible MMOG sessions during which the state update rate was above the minimal frequency, the total non-interruption ratio is: $\frac{23.98}{24} = 99.9\,\%$. It is important to mention that the total and instantaneous non-interruption ratios are high-level metrics that take into account other lower-level QoS metrics such as insufficient CPU, memory or network resources. Concretely, if a set of clients are assigned to an overloaded resource, they will experience a partial interruption characterised by its total and instantaneous non-interruption ratios.

3 SLA-based relationships

We present in this section, the business relationships between the actors participating in our cloud-based middleware model.

3.1 Client and game provider

The interaction between the client and the game provider requires human intervention only from the client. The relationship is regulated by the client account, created through a Web portal by the client upon agreeing on a *contract* with the game provider. The contract includes generic mutual obligations valid for all MMOGs, while further refinements and extensions can be added in the form of *annexes*.

Typical client obligations include subscription costs, client community interaction rules, and costs for accessing MMOG sessions. Typical game provider obligations include guaranteed services, such as community support, player support (player status and achievements, inventory, and detailed play statistics), mediation of client connections to MMOGs, access and availability to game world areas, and *compensation* (penalties) in case of contract violations. Client accounts can have unlimited duration, while MMOG contract annexes have well-defined validity periods, typically of 1 month.

Initially, the client selects a game provider based on offered MMOGs and contract terms. The client then creates an account by accepting (once) the contract offered by the selected game provider. After accepting the contract, the client selects the MMOG to play and, after agreeing upon the presented contract annex (once, during the first access), is allowed to connect to an MMOG session. The client may be refused access to an MMOG session, either because the game provider underestimated the demand for the requested MMOG, or because the game operator has not allocated enough resources for the MMOG session. In these situations, the game provider compensates the client according to the terms of the contract.

3.2 Game provider and game operator

We focus in this paper on formalising a *fully automated* interaction between the game provider and the game operator. To facilitate the read of the underlying formalism, we summarise in Table 1 the most important symbols used in this paper. Based on the total number of accounts and its service policy, the game provider computes the maximum number of clients for each game zone. We define the provider's *service policy* as a quintuple:

$$\left(s_{\text{ini}}, s_{\text{tni}}, s_{\text{time}}, P^{(\text{T})}, C^{(\text{T})}\right) \tag{1}$$

composed of five terms: (1) target instantaneous non-interruption ratio s_{ini} that should be minimised, (2) total non-interruption ratio s_{tni} that should also be minimised, (3) interval of acceptable SLA validity periods s_{time} to avoid excessively long operation agreements, (4) target hourly price per client $P^{(\text{T})}$, and (5) target compensation per client per minute $C^{(\text{T})}$.

Using the estimated client requests (see Fig. 1b), the game provider negotiates the most appropriate terms for hosting each zone by establishing O-SLAs with different operators (see Sect. 4). Based on its policies and available cloud resources, each operator publishes an *O-SLA template*:

$$\text{O-SLA} = \left(G^{(\text{O})}_{\text{type}}, t^{(\text{O})}_{\text{cli}}, t^{(\text{O})}_{\text{ini}}, t^{(\text{O})}_{\text{tni}}, t^{(\text{O})}_{\text{time}}, \sigma^{(\text{O})}, P^{(\text{O})}, C^{(\text{O})}\right), \tag{2}$$

consisting of a set of eight terms with either scalar or range values:

Table 1 Notation summary

Notation	Semantic
O-SLA	Operational SLA
t_{cli}	Number of clients to service
$t^{(\text{O})}_{\text{ini}}/t^{(\text{R})}_{\text{ini}}$	Instantaneous non-interruption ratio
$t^{(\text{O})}_{\text{tni}}/t^{(\text{R})}_{\text{tni}}$	Total non-interruption ratio
$t^{(\text{O})}_{time}/t^{(\text{R})}_{time}$	SLA validity period
$P^{(\text{O})}/P^{(\text{T})}$	Hourly SLA price
$C^{(\text{O})}_x/C^{(\text{T})}_x$	Compensation for SLA term $x \in \{\text{cli}, \text{ini}, \text{tni}\}$
$b_x/b^{(\max)}_x$	O-SLA/maximum O-SLA fault severity for term $x \in \{\text{cli}, \text{ini}, \text{tni}\}$
f_x	Shape function for SLA term $x \in \{\text{cli}, \text{ini}, \text{tni}\}$
$\mathcal{P}_{\text{O-SLA}}$	Pricing rank
$\mathcal{C}_{\text{O-SLA}}$	Compensation rank
$\mathcal{F}_{\text{O-SLA}}$	Fitness rank
$A_x/\mathcal{A}^{(\text{ch})}_x$	Compensation gain/characteristic compensation gain

O offered, *R* requested, *T* targeted

- *MMOG name* and *version* $G^{(\text{O})}_{\text{type}}$;
- *client number* $t^{(\text{O})}_{\text{cli}}$ (range), the number of clients that the game operator is ready to service;
- *instantaneous non-interruption ratio* $t^{(\text{O})}_{\text{ini}}$ (range) representing the minimum percentage from the state update frequency, the operator guarantees to maintain for all clients during the O-SLA validity period;
- *total non-interruption ratio* $t^{(\text{O})}_{\text{tni}}$ (range) representing the percentage of QoS fulfilment from the entire O-SLA validity time and evaluated only at the end of the O-SLA validity period;
- *validity period* $t^{(\text{O})}_{\text{time}}$ (range) representing the SLA lifetime offered by the game operator (with variable granularity from daily to semestrial);
- *geographical area* $\sigma^{(\text{O})}$ in which the game operator will service the clients;
- *base price* $P^{(\text{O})}$ for accepting an SLA utilising the lowest values of all terms in the given ranges;
- *compensation* $C^{(\text{O})}$ for O-SLA faults, defined as the aggregate penalty:

$$C^{(\text{O})} = \mathcal{P}(C_{\text{cli}}, C_{\text{ini}}, C_{\text{tni}}), \tag{3}$$

where $\mathcal{P}$ is a polynomial function, the operator has to pay in case of O-SLA faults, consisting of three QoS terms: (1) compensation for unserved clients C_{cli}, (2) compensation for instantaneous non-interruption ratio C_{ini}, and (3) compensation for total non-interruption ratio C_{tni}. Although the aggregation function may carry a lot of weight in the game operators' policies (e.g. make an expensive O-SLA template more attractive, if it favours important compensation terms like C_{ini}), we consider for the sake of clarity an additive function:

$$\mathcal{P}(C_{\text{cli}}, C_{\text{ini}}, C_{\text{tni}}) = C_{\text{cli}} + C_{\text{ini}} + C_{\text{tni}}. \tag{4}$$

We define these three compensation terms using a *compensation function*:

$$C_x : \left[0; b^{(\max)}_x\right] \to \mathbb{R}^+,\ C_x(b_x) = \frac{c^{(\text{u})}_x \cdot b_x}{u_x} \cdot f_x\left(\frac{b_x}{b^{(\max)}_x}\right),$$
$$\forall x \in \{\text{cli}, \text{ini}, \text{tni}\}, \tag{5}$$

where $\mathbb{R}^+$ denotes the positive real numbers, $c^{(\text{u})}_x$ is the compensation for an O-SLA fault of one *term unit* u_x (i.e. one client or 0.1 % of the instantaneous/total non-interruption ratio), b_x is the O-SLA fault severity for term x (i.e. measured client number or instantaneous/total non-interruption ratio), $b^{(\max)}_x$ is its maximum severity, and f_x is a *shape function* with the signature:

$$f_x : [0; 1] \to \mathbb{R}^+, \tag{6}$$

employed for changing the importance of different fault classes. A game operator could make its offer more appealing by employing a shape function offering higher

compensations for most frequent, low-severity faults rather than for the infrequent, higher severity ones (see Sect. 5.5). We present and analyse in Sect. 5.6, Eq. 21, two important shape functions: logarithmic and exponential. Non-negotiable O-SLA terms such as the issuer (i.e. game operator) and the measurement timestep describing the time interval between consecutive QoS evaluations are not represented for simplicity reasons.

3.3 Game operator and resource provider

The interaction between the game operator and the resource provider is also fully automated. As mentioned in Sect. 2.2, the game operator selects, from different providers, appropriate cloud resources on which to host and run the MMOG zones. The result of this interaction is a *Resource SLA* (R-SLA) with the following terms: (1) *issuer* or resource provider; (2) *geographical location* of the issuer's data centre; (3) *resource bulk* representing the set of rented resources comprising processor speed, memory size, internal and external network bandwidth; (4) *validity period* representing the availability duration of the resources to the game operator after accepting an R-SLA (usually hourly grained); (5) *compensation terms* in case of resource faults; (6) *price* representing the requested non-negotiable price.

The R-SLA terms provided by commercial clouds in today's market have fixed, non-negotiable values. Therefore, no negotiation but a simple request-offer matching algorithm is employed by the game operator. Another complication is the fact that the terms offered by cloud providers, such as Amazon ECU (equivalent CPU capacity of a 1–1.2 GHz 2007 Opteron or Xeon processor) and FlexiScale (vCPU units), are not precise and difficult to map to finer-grained agreements and compensations other than for resource downtime. Our approach to making offers more precise is to use application-specific benchmarks to quantify the performance offered by cloud providers before establishing R-SLAs, such as the *RS unit* benchmark employed in Sect. 5.1. We studied the interaction between the game operator and the resource in [2, 19] and no longer address it here.

4 Game operators–game providers interaction

We define the O-SLA negotiation as a decision process in which two parties interact with each other for mutual gain 20. (i.e. maximise income and keep expenditures low). The game provider's income comprises the MMOG subscription sales and the compensations paid by the game operator in case of O-SLA faults, while its expenditures consist of the O-SLA acquisitions and the compensations to the clients for low QoS. The game operator's income results from the O-SLAs provisioned to the game provider, and its

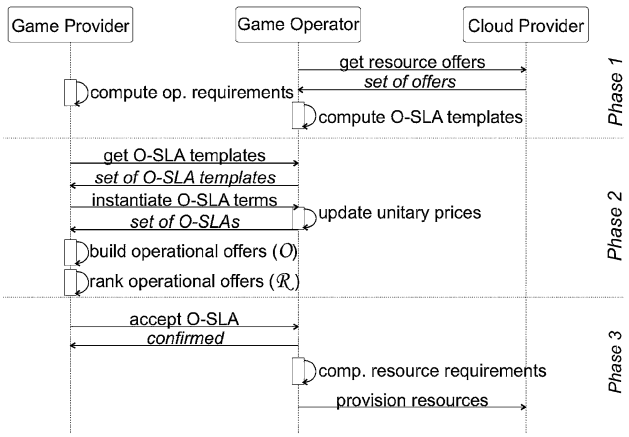

Fig. 3 O-SLA negotiation protocol

expenditures comprise the acquisition of resources from the cloud providers and the O-SLA compensations to the game providers. The accounting, billing and auditing aspects of SLAs fall outside our scope (but solutions exist).

The three negotiation phases depicted in Fig. 3 cover the game operators generating *O-SLA templates* based on cloud resource pricing and availability (phase one), the game providers instantiating and ranking O-SLA offers (phase two) and finally, the binding agreement (phase three). A simpler one-phase request-offer matching algorithm cannot be employed because it would not be fair towards the game operators. The resource providers' pricing policies can change between the time the operator published an offer until the game provider accepted it, which would enable a game provider profit from delaying the answer. Introducing a validity deadline for offers to prevent this unfair behaviour could have negative effects on both game provider and game operator, as one might not have enough time for the ranking process, while the other would have to assume the risk of changes in resource prices (within the offer validity time). Thus, the proposed negotiation involves dynamic offers (determined by the available cloud resources) and a possibility for game operators to propose small changes in price during the final agreement phase.

4.1 First phase

In the first phase, the game operator checks the resources offered by different providers and publishes an *O-SLA template*, as defined in Eq. 2. The game provider computes the *operational requirements*:

$$R = \left(G^{(\mathrm{R})}_{\mathrm{type}}, t^{(\mathrm{R})}_{\mathrm{cli}}, t^{(\mathrm{R})}_{\mathrm{ini}}, t^{(\mathrm{R})}_{\mathrm{tni}}, t^{(\mathrm{R})}_{\mathrm{time}}, \sigma^{(\mathrm{R})}, P^{(\mathrm{T})}, C^{(\mathrm{T})}\right) \quad (7)$$

based on the current state of its provisioned O-SLAs and the estimated number of clients for the next provisioning time frame, where:

- $G^{(\mathrm{R})}_{\mathrm{type}}$ is the required MMOG type;
- $t^{(\mathrm{R})}_{cli}$ is the estimated number of active accounts;
- $t^{(\mathrm{R})}_{\mathrm{ini}}$ is the required instantaneous total non-interruption ratio, initially equal to the minimum instantaneous non-interruption ratio s_{ini} of the game provider's service policy (defined in Eq. 1);
- $t^{(\mathrm{R})}_{\mathrm{tni}}$ is the total non-interruption ratio, initially equal to the minimum total non-interruption ratio s_{tni} from the provider's service policy;
- $t^{(\mathrm{R})}_{\mathrm{time}}$ is the estimated time period for these requirements expressed in hours;
- $\sigma^{(\mathrm{R})}$ is the geographical area;
- $P^{(\mathrm{T})}$ is the target hourly price per client defined by the game provider's service policy;
- $C^{(\mathrm{T})}$ is the target compensation per client per minute defined by the game provider's service policy.

4.2 Second phase

In the second phase, the game provider gathers O-SLA templates from all game operators and instantiates them with the "best" values allowed by the template for the operational requirements. When instantiating an O-SLA template, it also calculates the price increase for the client number P_{cli}, the instantaneous non-interruption ratio P_{ini}, and the total non-interruption ratio P_{tni}:

$$P_x = \frac{t^{(\mathrm{R})}_x - t^{(\mathrm{O}_{\min})}_x}{u_x} \cdot p^{(u)}_x \cdot f_x\left(t^{(\mathrm{O})}_x\right), \quad \forall x \in \{\mathrm{cli}, \mathrm{ini}, \mathrm{tni}\}, \tag{8}$$

where $t^{(\mathrm{R})}_x$ is the operational requirement for the term $x \in \{\mathrm{cli}, \mathrm{ini}, \mathrm{tni}\}$ (see Eq. 7), $t^{(\min)}_x$ is the minimum value of the term x allowed by the operator through an O-SLA, $p^{(\mathrm{u})}_x$ represents the price per term unit u_x, and $f_x(t^{(\mathrm{O})}_x)$ is a shape function defined as in Eq. 6. The final price the game provider is charged when accepting the O-SLA is:

$$P^{(\mathrm{O})} = P_{\mathrm{base}} + \left(P_{\mathrm{cli}} + P_{\mathrm{ini}} \cdot t^{(\mathrm{R})}_{\mathrm{cli}} + P_{\mathrm{tni}} \cdot t^{(R)}_{\mathrm{cli}}\right) \cdot T_{\mathrm{coeff}}, \tag{9}$$

where P_{base} is the base price and T_{coeff} is the *validity period coefficient* that adjusts the price in case of changes in validity time requested by the provider:

$$T_{\mathrm{coeff}} = \left\lceil \frac{t^{(\mathrm{R})}_{\mathrm{time}}}{t^{(O_{\min})}_{\mathrm{time}}} \right\rceil \cdot f_{\mathrm{time}}\left(t^{(\mathrm{R})}_{\mathrm{time}}\right), \tag{10}$$

where $\lceil \cdot \rceil$ is the ceiling function, $t^{(\mathrm{O}_{\min})}_{\mathrm{time}}$ is the lowest O-SLA validity period allowed by the operator, and f_{time} is a shape function defined as in Eq. 6.

Next, the O-SLA instances are grouped by the game provider into a set of M feasible *operational offers*:

$$O = \bigcup_{i=1}^{M} \text{O-SLA}_i, \tag{11}$$

Consider for example, the operational requirements of 50 thousand clients and three O-SLAs (*O-SLA*[1;3]) with the maximum of 25, 20 and 30 thousand clients. The resulting operational offers are {*O-SLA1*, *O-SLA3*} and {*O-SLA2*, *O-SLA3*} ($M = 2$ in both cases). The combination {*O-SLA1*, *O-SLA2*} is not feasible because it does not meet the minimum operational requirements of 50 thousand players ($25 + 20 < 50$).

The game provider assigns to each operational offer an *operational rank* computed based on the weighted sum of three individual ranks: the pricing rank $\mathcal{P}_{\mathrm{O-SLA}}$ (directly proportional), the compensation rank $\mathcal{C}_{\mathrm{O-SLA}}$ (inversely proportional) and the resource fitness rank $\mathcal{F}_{\mathrm{O-SLA}}$ (inversely proportional):

$$\mathcal{R} = \lambda_p \cdot \mathcal{P}_{\mathrm{O-SLA}} - \lambda_c \cdot \mathcal{C}_{\mathrm{O-SLA}} - \lambda_f \cdot \mathcal{F}_{\mathrm{O-SLA}}, \tag{12}$$

where $\lambda_p, \lambda_c, \lambda_f \in [0; 1]$ and $\lambda_p + \lambda_c + \lambda_f = 1$. The main goal of this paper is to determine best practices for a game provider for computing these compensation and fitness weights in an environment with multiple competing providers and game operators (we studied the pricing weight in [3]). We define in the following, the computation of the pricing, compensation and resource fitness ranks by the game provider.

The *pricing rank* $\mathcal{P}_{\mathrm{O-SLA}}$ of an operational offer is a quantification of the resources' price, determined as the ratio between the aggregated hourly price $\frac{P^{(\mathrm{O})}_i}{t^{(\mathrm{O})}_{\mathrm{time}_i}}$ of all M O-SLAs of an operational offer and the target price $P^{(\mathrm{T})} \cdot t^{(\mathrm{O})}_{cli\_i}$ for servicing all clients in all M O-SLAs (see O-SLA definition in Eq. 2):

$$\mathcal{P}_{\mathrm{O-SLA}} = \frac{\sum_{i=1}^{M} \frac{P^{(\mathrm{O})}_i}{t^{(\mathrm{O})}_{\mathrm{time}_i}}}{P^{(\mathrm{T})} \cdot \sum_{i=1}^{M} t^{(\mathrm{O})}_{cli_i}}. \tag{13}$$

The *compensation rank* quantifies the penalties the operator pays for O-SLA faults, based on a *compensation gain* metric representing the area of the compensation function C_x within its definition interval $[0; b^{(\max)}_x]$ (see Eq. 5):

$$\mathcal{A}_x = \int_0^{b^{(\max)}_x} C_x(b_x) \cdot db_x = \frac{c^{(u)}_x}{u_x} \cdot \int_0^{b^{(\max)}_x} b_x \cdot f_x\left(\frac{b_x}{b^{(\max)}_x}\right) \cdot db_x. \tag{14}$$

By substituting $y = \frac{b_x}{b^{(\max)}_x}$ in Eq. 14, we obtain:

$$\mathcal{A}_x = \frac{c_x^{(u)} \cdot \left(b_x^{(\max)}\right)^2}{u_x} \cdot \int_0^1 y \cdot f_x(y) \cdot dy. \quad (15)$$

While the compensation gain completely characterises the compensation function for uniformly distributed SLA faults, it does not accurately reflect its behaviour in a realistic system with a non-uniform fault distribution. To compensate for this drawback, we introduce an *SLA fault distribution function*:

$$\delta_x : \left[0; b_x^{(\max)}\right] \rightarrow [0; \Delta_{\max}], \quad (16)$$

where $\Delta_{\max}$ is the maximum value of the SLA fault distribution function. We dynamically compute the SLA fault distribution for each MMOG zone by continuously monitoring the game play and recording each fault. By superimposing δ_x to the compensation gain, we compute an adjusted metric called *characteristic compensation gain* as the compensation function for an MMOG:

$$\mathcal{A}_x^{(\mathrm{ch})} = \frac{c_x^{(\mathrm{u})} \cdot \left(b_x^{(\max)}\right)^2}{u_x} \cdot \int_0^1 y \cdot \delta_x\left(b_x^{(\max)} \cdot y\right) \cdot f_x(y) \cdot dy. \quad (17)$$

We approximate the characteristic compensation gain through a finite sum:

$$\mathcal{A}_x^{(\mathrm{ch})} \approx \frac{c_x^{(\mathrm{u})} \cdot \left(b_x^{(\max)}\right)^2}{u_x} \cdot \sum_{i=1}^{N} \frac{i}{N} \cdot \delta_x\left(\frac{b_x^{(\max)} \cdot i}{N}\right) \cdot f_x\left(\frac{i}{N}\right), \quad (18)$$

where N is the *integration granularity* representing the number of interval partitions. In Sect. 6.4, we evaluate the impact of the integration granularity on the game provider's yield from compensations and its effects on the QoS.

Using the term $\mathcal{A}_x^{(\mathrm{ch})}$, we can finally compute the compensation rank of an operational offer as the sum as the weighted sum of the normalised characteristic compensation gains for all O-SLA terms $x \in \{\mathrm{cli}, \mathrm{ini}, \mathrm{tni}\}$:

$$\mathcal{C}_{\mathrm{O-SLA}} = \sum_{i=1}^{M} \sum_{x \in \{\mathrm{cli,ini,tni}\}} \psi_x \cdot \frac{\mathcal{A}_{x_i}^{(\mathrm{ch})}}{\mathcal{A}_x^{(\mathrm{REF})}}, \quad (19)$$

where $\mathcal{A}_x^{(\mathrm{REF})}$ represents a reference compensation gain considered ideal by the game provider (e.g. minimum compensation function from all operators), and ψ_{cli}, ψ_{ini}, $\psi_{\mathrm{tni}} \in [0; 1]$ indicate the provider's preference for each specific O-SLA term, where $\psi_{\mathrm{cli}} + \psi_{\mathrm{ini}} + \psi_{\mathrm{tni}} = 1$. Section 6.3 and Fig. 9 give examples and evaluate several compensation ranking alternatives.

The *fitness rank* reflects how well the operational offer matches the requirements, computed as a weighted sum of the ratio between the offered $t_x^{(\mathrm{O})}$ and the requested $t_x^{(\mathrm{R})}$ O-SLA terms (i.e. t_{cli}, t_{ini}, t_{tni}, and t_{time}—see Eq. 2):

$$\mathcal{F}_{\mathrm{O-SLA}} = \sum_{x \in \{\mathrm{cli,ini,tni,time}\}} \phi_x \cdot \frac{S_x\left(t_{x_i}^{(\mathrm{O})}\right)}{t_x^{(\mathrm{R})}},$$

$$\text{where } S_x\left(t_{x_i}^{(\mathrm{O})}\right) = \begin{cases} \sum_{i=1}^{M} t_{x_i}^{(\mathrm{O})}, & x = \mathrm{cli}; \\ \frac{\sum_{i=1}^{M} t_{x_i}^{(\mathrm{O})}}{M}, & x \in \{\mathrm{ini}, \mathrm{tni}\}; \\ \min_{i \in [1;M]} \left\{ t_{x_i}^{(\mathrm{O})} \right\}, & x = \mathrm{time}, \end{cases} \quad (20)$$

ϕ_{cli}, ϕ_{ini}, ϕ_{tni}, $\phi_{\mathrm{time}} \in [0;1]$ indicate again the provider's preference for each O-SLA term ($\phi_{\mathrm{cli}} + \phi_{\mathrm{ini}} + \phi_{\mathrm{tni}} + \phi_{\mathrm{time}} = 1$) and S_x is an aggregation function (i.e. sum for number of clients, average for instantaneous and total non-interruption ratios, and minimum for validity period). The offer is unfit if the fitness rank is lower than one, is a perfect match if equal to one, or contains too many resources if higher than one.

Finally, the operational offers are sorted in ascending order by their rank. It is worth noting that, although the price ranking is relatively static between successive negotiations (provided that the operators do not adjust their offers dynamically), the fitness and compensation rankings constantly vary based on the current operational demands and the operators' SLA fault history (see characteristic compensation gain function in Eq. 17). This ensures that game providers do not reach the same apparently optimal operator, but are able to discover those whose offers most accurately match their needs.

4.3 Third phase

In the third phase, the game provider attempts to accept an operational offer starting with the best ranked one, and continues through the list in case other competing providers already provisioned it. At this stage, the operators are allowed to propose small updates in the O-SLA terms to compensate for changes in the cloud providers' R-SLAs. In turn, the game providers will either recompute the rank for the O-SLA in question, or will simply skip to the next best offer according to their internal policy. After the negotiation, the provider tries to enforce the accepted O-SLA for the entire interaction with the clients and the game operator. To achieve this, the game provider collects and aggregates data from two sources: the game operator's QoS data collected from MMOG servers and the client that regularly reports (in the background) on the quality of game play. The game provider enforces the O-SLAs by compensating the clients according to their contractual

terms (not covered here) and by penalising the game operators in case of QoS violations.

5 Experimental setup

We present in this section, the experimental setup used for evaluating our MMOG middleware stack focussed on the O-SLA-based negotiation process between the game providers and the game operators. We conduct our evaluation in simulation, but use as input real data corresponding to MMOG workloads (number of players online) and real commercial IaaS cloud SLAs.

5.1 RuneScape

We use traces from RuneScape, a real MMOG ranked second after World of Warcraft by the number of active paying customers in the US and European markets. We collected execution traces from the official RuneScape web page for a period of 6 months from over 130 servers spread across five geographical regions by sampling the number of players every two minutes (see Table 2). Our traces contain the number of players over time for each server group used by the RuneScape operators. These numbers are used by the game providers to provision O-SLAs according to the number of active client accounts. Since the total number of MMOG subscriptions is only occasionally published by the game provider, we approximate it from the total number of active accounts using the concurrent active account ratio (around 15 % in case of RuneScape), computed at each moment of time when the total number of MMOG subscriptions is publicly known (see Fig. 4). We need this metric in our experiments for computing the game provider's income from client subscriptions. For the client–game provider interaction, we use the real monthly subscription model of RuneScape (e.g. $5.95 as of August 2012).

5.2 MMOG simulator

To validate our theory, we developed an MMOG simulator based on the model proposed in Sect. 3. While the simulator considers many relevant computing and resource-level parameters of an MMOG (e.g. load computation for CPU, memory, multiple network connections), we focus its presentation on the business relationships targeted in this paper. Figure 5 presents an overview of the simulation process centred around the three main actors: game provider, game operator and resource provider. The simulator uses MMOG traces (such as the RuneScape ones) as input to simulate the clients.

Table 2 Number of server groups in different regions

Region	Server groups
Europe	40
North America East Coast	37
North America Continental	12
North America West Coast	39
Australia	6

We configured the game providers through two inputs: the service policy (PP) parameters from Eq. 1 and the operational ranking policy (RK) parameters from Eq. 12. Each game provider is assigned a subset of the RuneScape traces. Based on the number of client access requests and their geographical origin, the game providers estimate the demand for each geographical area. Based on the estimated demand, the providers construct their operational requirements (R) which, along with the operational ranking policies, represent their basis for negotiating with the game operators.

We configured the game operators through O-SLA templates as the basis for their O-SLA offers. The negotiation with the game providers results in some O-SLAs being accepted and, as a consequence, in game providers assigning clients to be serviced by the operators. Based on historical data and the currently assigned clients, each operator predicts the load for predetermined time intervals (shorter for dynamic MMOGs and longer otherwise) using the method presented in [17]. The prediction consists of the distribution of the clients in the MMOG world and a load model for translating it into resource requirements. In case of RuneScape, we determined that a prediction interval of two minutes is adequate. At each prediction interval, the operators evaluate their provisioned resources and make adjustments if necessary by allocating/releasing resources and redistributing the load using the RTF [16] mechanisms (e.g. game zone replication and clients migration).

The role of the resource providers in the simulation is to periodically evaluate their allocated resources and to eventually offer new R-SLAs based on the remaining free resources.

Each prediction/negotiation step of the simulation is preceded by an evaluation of the state of the allocated resources, the QoS provided to the clients, and an analysis of the SLA faults. The output of this step consists of O-SLA and R-SLA (de-)allocation traces, along with accounting traces of all financial transactions between the three actors, including all compensations. This output is logged and represents the simulation output (not shown in Fig. 5).

5.3 Cloud providers' R-SLAs

We employ 115 R-SLAs based on the resources provided by 16 commercial cloud providers, described in Table 3. We present the hourly prices relative to the processing

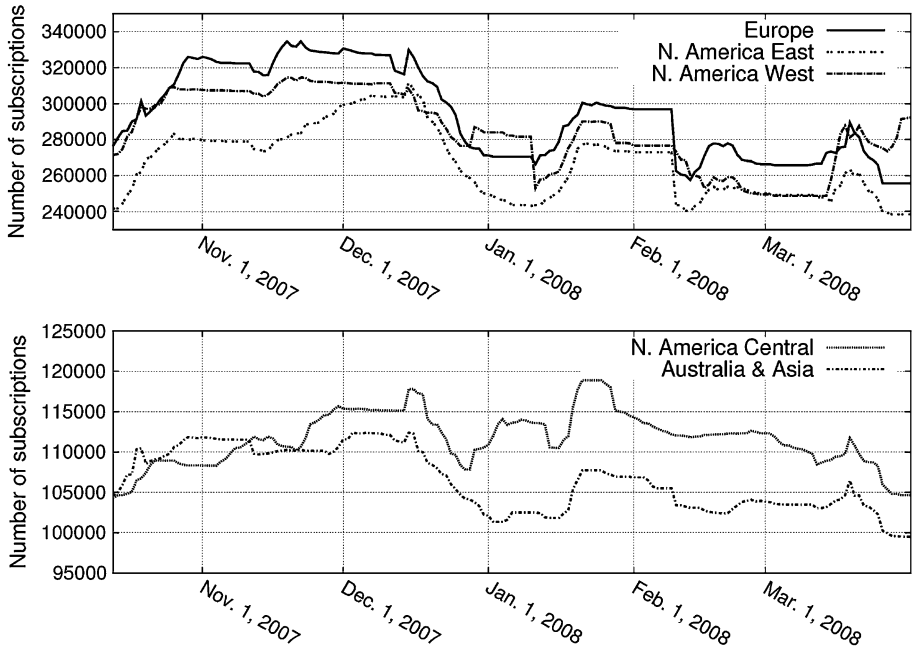

Fig. 4 Total number of RuneScape client subscriptions aggregated for all servers in each region

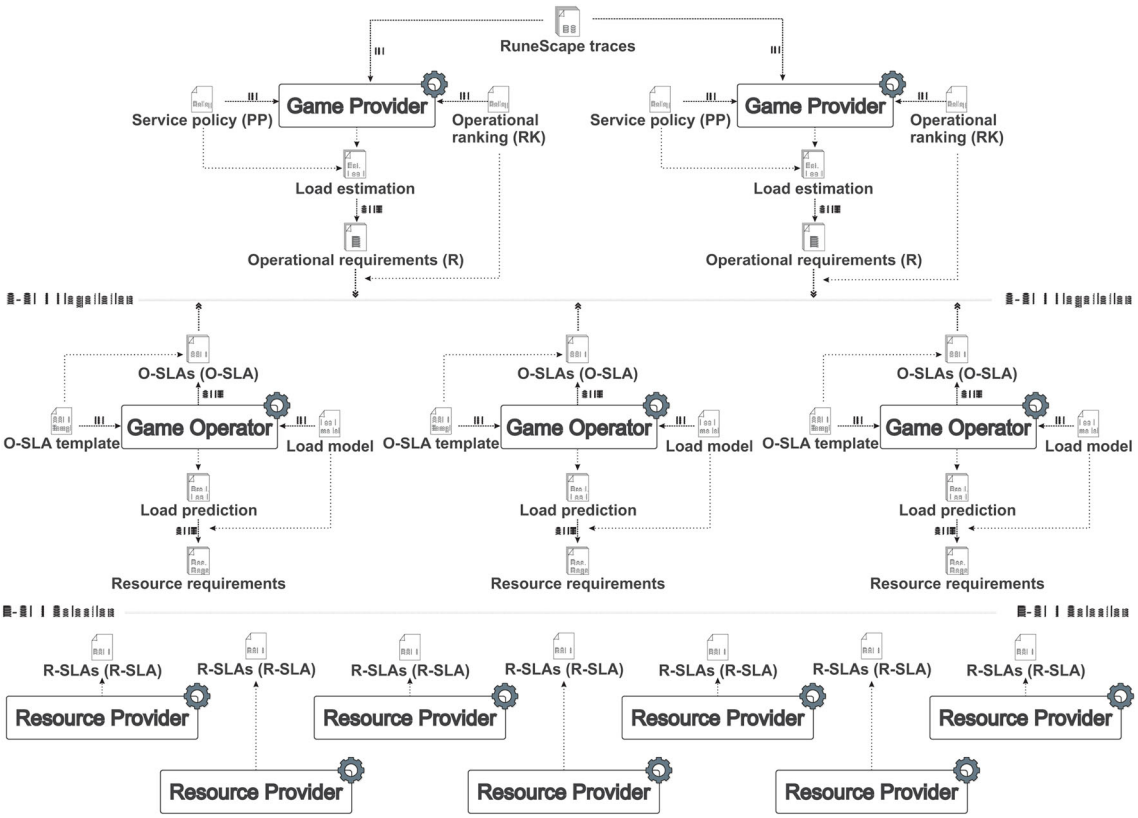

Fig. 5 MMOG simulation architecture

power and the memory capacity, including the upstream and downstream network traffic which may have an important impact on the final R-SLA prices, as in the case of the CloudCentral provider. For each cloud provider, we use its geographical location, memory size, and price. We express the VM processing power using an MMOG-oriented metric called *RS unit* (RSU), representing the equivalent maximum computational requirements of one

RuneScape server servicing 2000 clients. We compute this metric, including the virtualisation overheads, based on benchmarking and analysis data from our previous work [20]. We quantified in our simulation, the entire traffic generated by the RuneScape servers on external networks and included its costs in the hourly RSU presented in Table 3 (fourth column). The VM instantiation overhead is the variable duration of instantiating a new VM.

Table 3 Summary of commercial cloud R-SLAs

Cloud provider	VM types	Locations	Price [$/..		Validity (h)	VM instantiation (seconds)
			RSU/h]	GB/h]		
Amazon	6	4	1.21	0.81	1	[65; 105]
CloudCentral	5	1	11.07	35.25	1	[50; 120]
ElasticHosts	4	1	1.22	2.73	1	[45; 120]
FlexiScale	4	1	0.72	1.46	1	[40; 50]
GoGrid	4	1	2.07	7.15	1	[60; 120]
Linode	5	1	0.67	2.37	24	[45; 120]
NewServers	5	1	0.38	0.71	1	[30; 120]
OpSource	6	1	0.09	0.15	1	[300; 540]
RackSpace	4	2	1.54	5.56	1	[100; 300]
ReliaCloud	3	1	0.96	1.04	1	[45; 60]
SoftLayer	4	3	0.70	1.75	1	[180; 300]
SpeedyRails	3	1	1.76	8.43	24	[80; 120]
Storm	6	2	0.99	1.54	1	[600; 900]
Terremark	5	1	1.40	6.14	1	[40; 60]
Voxel	4	3	0.83	0.94	1	[300; 600]
Zerigo	2	1	1.96	3.16	1	[60; 120]

Table 4 Resource availability parameters and statistical characterisation

Metric	Distribution	Statistical properties of failure trace						Avail-ability (%)
		Min	Max	Average	Q1	Q2	Q3	
DUR[a]	LN[b] (2.12, 0.31)	1	37	8	6	8	10	–
SIZE[c]	W[d] (4, 5)	0	6	3	3	3	4	–
IAT[e]	W (13600, 7)	2073	19668	12722	11381	12906	14254	99.5
IAT	W (7000, 7)	888	10123	6548	5860	6645	7336	99.6
IAT	W (4750, 7)	535	6988	4443	3976	4509	4977	99.7
IAT	W (3550, 7)	476	5146	3320	2971	3370	3720	99.8
IAT	W (2830, 7)	341	4119	2647	2369	2686	2965	99.9

[a] Failure duration (in minutes)

[b] Log-Normal distribution

[c] Failure size (in number of machines)

[d] Weibull distribution

[e] Inter-arrival time (in seconds)

Table 5 Game providers' service policies, defined in bold by (min; max; step) value ranges; stochastic parameters are defined by [min; max] intervals

Policy	s_{ini}	s_{tni}	s_{time} (hours)	$P^{(T)}$ ($)	$C^{(T)}$ ($)
PP1	0.9	0.99	[12; 168]	0.01	0.05
PP2-PP6	**(0.86; 0.98; 0.03)**	0.992	[168; 336]	0.002	0.05
PP7-PP11	0.92	**(0.986; 0.998; 0.003)**	[168; 336]	0.002	0.05
PP12-PP16	0.92	0.992	**[(24; 312; 72); (336; 624; 72)]**	0.002	0.05

5.4 Resource availability

We associated each of the modelled Cloud providers' data centres with generated failure traces with tunable availability. We employ the resource availability model proposed in [21] and generate failure traces with average availabilities ranging from 99.5 to 99.9 %. The traces are characterised through their failures' duration, size and inter-arrival time (IAT), each modelled through a statistical distribution. The distributions' parameters are presented in Table 4, along with the statistical properties of the resulting traces. For all traces, we employ the same distribution for the failure duration and size, but we vary the resource availability by adjusting the failure IAT. We generated both independent and correlated failures, the ratio between the two being 3:2.

5.5 Game provider's service policies

We characterise the client-game provider contracts through the service policies displayed in Table 5 (see terms in Eq. 1). We further sample the design space of operational ranking functions through the 45 functions summarised in Table 6, by varying the class of the compensation ranking function in RK1–RK6 (see Sect. 6.3), the computational complexity of the compensation ranking function in RK7–RK17 (see Sect. 6.4), the fitness ranking function (Sect. 6.5) in RK18–RK37, and the complete operational ranking function (Sect. 6.6) in RK38–RK45.

5.6 O-SLA templates

We employ an extensive set of O-SLAs designed to cover all aspects of the negotiation described in Sect. 4, based on the three O-SLA templates presented in Table 7 and generated by varying one or more of their term values. We keep the pricing functions constant, since we covered them in previous work [3]. As compensation functions (see Eq. 5 in Sect. 3.2), we use two classes of parameterised shape functions (logarithmic and exponential):

Table 6 Operational ranking configuration parameters [sets of functions defined in bold by the (min; max; step) value ranges]

Ranking acronym (function)	$\mathcal{C}_{\text{O-SLA}}$		Fitness rank ($\mathcal{F}_{\text{O-SLA}}$)				Oper. rank ($\mathcal{R}$)	
	Type	N	ϕ_{cli}	ϕ_{ini}	ϕ_{tni}	ϕ_{time}	λ_f	λ_b
RK1(*max*)	max	–	0.2	0.5	0.2	0.1	0.1	0.8
RK2(*avg-3*)	avg	3	0.2	0.5	0.2	0.1	0.1	0.8
RK3(*avg-9*)	avg	9	0.2	0.5	0.2	0.1	0.1	0.8
RK4(*gain*)	$\mathcal{A}_x$	–	0.2	0.5	0.2	0.1	0.1	0.8
RK5(*cgain-9*)	$\mathcal{A}_x^{(ch)}$	9	0.2	0.5	0.2	0.1	0.1	0.8
RK6(*cgain-30*)	$\mathcal{A}_x^{(ch)}$	30	0.2	0.5	0.2	0.1	0.1	0.8
RK[7; 17] (*cgain-[1;30]*)	$\mathcal{A}_x^{(ch)}$	**1, (3; 30; 3)**	0.2	0.5	0.2	0.1	0.1	0.8
RK[18; 22](*cli-[10;90]*)	$\mathcal{A}_x^{(ch)}$)	30	**(0.1; 0.9; 0.2)**	$\frac{1-\phi_{\text{cli}}}{3}$			0.6	0.2
RK[23; 27](*ini-[10;90]*)	$\mathcal{A}_x^{(ch)}$	30	$\frac{1-\phi_{\text{ini}}}{3}$	**(0.1; 0.9; 0.2)**	$\frac{1-\phi_{\text{ini}}}{3}$		0.6	0.2
RK[28; 32](*tni-[10;90]*)	$\mathcal{A}_x^{(ch)}$	30	$\frac{1-\phi_{\text{tni}}}{3}$		**(0.1; 0.9; 0.2)**	$\frac{1-\phi_{\text{tni}}}{3}$	0.6	0.2
RK[33; 37](*time-[10;90]*)	$\mathcal{A}_x^{(ch)}$	30	$\frac{1-\phi_{\text{time}}}{3}$			**(0.1; 0.9; 0.2)**	0.6	0.2
RK[38; 45](*or-[1;8]*)	$\mathcal{A}_x^{(ch)}$	30	0.1	0.3	0.3	0.3	**(0.1; 0.8; 0.1)**	**(0.8; 0.1; −0.1)**

Table 7 RuneScape-related O-SLA templates

Name	$t_{\text{cli}}^{(\text{O})}$(x$10^3$)	$t_{\text{ini}}^{(\text{O})}$	$t_{\text{time}}^{(\text{O})}$	$t_{\text{tni}}^{(\text{O})}$	C_{cli}		C_{ini}		C_{tni}	
					f_{cli}	a	f_{ini}	a	f_{tni}	a
OSLA-1	[2; 20]	[0.85; 0.95]	[24; 168]	[0.99; 0.999]	exp	1.5	exp	1.3	exp	1.3
OSLA-2	[3; 10]	[0.90; 0.98]	[144; 336]	[0.99; 0.999]	log	15	exp	1.3	exp	1.3
OSLA-3	=OSLA-1	=OSLA-1	=OSLA-1	=OSLA-1	log	10	log	10	log	10

$$f_x^{(\log)}(b_x) = \frac{\log(a \cdot b_x + 1)}{\log(a+1)}; \quad f_x^{(\exp)}(b_x) = \frac{e^{a \cdot b_x} - 1}{e^a - 1}, \tag{21}$$

where $a \in \mathbb{R}^+$ is the shape coefficient. We adjust this coefficient for different O-SLAs and evaluate the resulting compensation functions using the compensation gain defined in Eq. 15. Finally, we use a uniform distribution of the serviced geographical areas.

5.7 Evaluation metrics

We evaluate our new model using several metrics for QoS, cost, and SLA. First, we define three important QoS metrics that define MMOG management failures, alongside the instantaneous non-interruption ratio t_{ini} and the total non-interruption ratio t_{tni} defined in Sect. 2.3:

1. *number* of interruptions in a time interval (e.g. the simulation period);
2. *duration* of the interruptions measured from the start of the event (resource/management failure) to the moment when all affected clients recover (i.e. when they are reconnected to the MMOG session in case of total interruptions, or when the QoS is above the promised level in case of partial interruptions);
3. *severity* of the interruptions, which represents the percentage of affected players of the MMOG session.

For a better understanding of t_{tni}, we also analyse the *average non-serviced clients* as the average number of clients who were denied service within a measurement timestep because of improper O-SLA provisioning by the game provider, or because of improper resource allocation by the game operator.

Second, we analyse the financial MMOG operation using three metrics:

1. *gross profit* representing the difference between the actor's revenue and the cost of providing its services, excluding taxation and other overheads;
2. *total compensation* (fraction of gross profit) representing the cost paid by an actor as compensation for any SLA fault for the entire simulation period;
3. *compensation events* representing a breakdown of the total compensation for all faults. The sum of all compensation events is the total compensation.

6 Experimental results

We presents in this section, the results of our evaluation. We cover an evaluation space with four dimensions explored in the following subsections: (1) MMOG QoS in Sects. 6.1 and 6.2, (2) compensation rank function in Sects. 6.3 and 6.4, (3) fitness rank function in Sect. 6.5, and (4) operational ranking function with multiple game providers competing for resources in Sect. 6.6. Table 8 shows an overview of the experiments.

6.1 Impact of resource availability on MMOGs' QoS

In the first experiment, we investigate the impact of resource failures on the quality of game play under different resource availability conditions. We employ the cloud resources and failure traces introduced in Sect. 5.4. We run one experiment for each average resource availability and evaluate the experienced QoS through the total interruptions metric, defined in Sect. 5.7.

Figure 6 depicts the number, the average duration and the severity of the total interruptions registered in the MMOG sessions over the 6 months simulation, as a function of resource availability. We observe stable, constant values for the severity and duration of total interruptions (the two bottom graphs) across all resource availability values, which indicates a proper recovery from resource failures. The median duration of a total interruption is two minutes and just below four minutes (i.e. approximately two resource allocation cycles) for more than 75 % of the events. The median percentage of affected players is below 2 %, as shown by the bottom interruption severity plot. The

Table 8 The evaluation space (the focus of each section is in bold)

Section	QoS		Ranking function			Game providers
	Availability	Contention	Pricing	Compensation	Fitness	
6.1	**Yes**	No	No	No	No	Single
6.2	**Yes**	**Yes**	No	No	No	Single
6.3	No	No	Yes	**Yes**	No	Single
6.4	No	No	Yes	**Yes, details**	No	Single
6.5	No	No	Yes	No	**Yes**	Single
6.6	No	No	**All yes**			**Multiple**

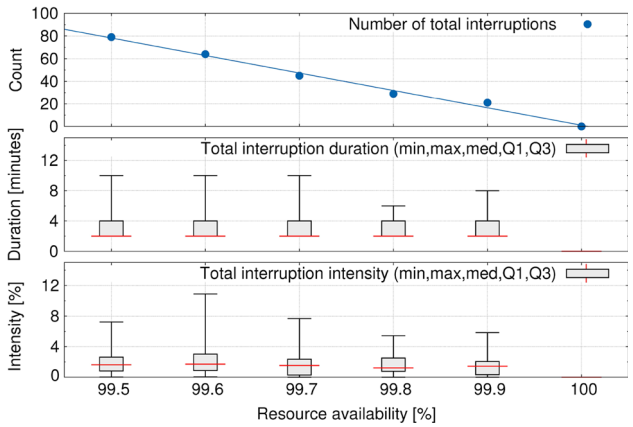

Fig. 6 Total interruptions under different resource availability conditions (*all graphs* show resource availability on horizontal axis with the *bottom values*)

trend in the number of total interruptions (top graph) is inversely proportional to the average resource availability. This could be alleviated by a fault prediction method [2] that migrates MMOG servers away from failing resources.

Running MMOGs on real, limited availability cloud resources can potentially have a strongly negative impact on the QoS for the clients due to prolonged recovery times and the need for human intervention for restoring the game session. However, we conclude based on this first experiment that this performance degradation can be mitigated by employing our proposed recovery techniques, which effectively limits the duration of the resulting interruptions to a constant value (lower than four minutes for 75 % of the events in our concrete scenario) independent of the duration of the underlying resource failure.

6.2 MMOG QoS in competition-based environments

The goal of the second experiment is to analyse the impact of resource failures on MMOGs in resource scarcity scenarios. We add this new dimension to our study by generating an increasing resource contention through gradually reducing the amount of resources in our setup. Thus, we run a set of simulations employing the same 6 month long RuneScape traces, but varying the amount of resources so that the peak load requires between 5 and 95 % of the available resources. Concretely, for example, the setup in which the RuneScape peak load requires 60 % of the total amount of resources has a resource contention value of 60 %. As in the previous experiment, we shape the availability of the resources from 99.5 to 100 % by employing the traces presented in Table 4. We cover these two dimensions with six values each, for a total of 36 simulations.

Figure 7 shows the number of interruptions experienced by the clients during the 6-month simulation in the different resource availability and contention scenarios. We observe a slanting in the number of total interruptions (seen in Fig. 7a), which is consistent with the decrease in availability for all resource contention values, confirming the previous experiment's conclusions. The central positive finding of this investigation is the fact that the number of total interruptions remains constant with increasing resource contention, even in the limit case of 95 % resource contention. The only observed particularity is the lower number of total interruptions for the other limit case with an extreme resource abundance (5 % resource contention) over all availability values. The number of partial interruptions (Fig. 7b) appears not to be impacted either by the competition for resources, or by the resource availability.

For a complete assessment of the effectiveness of our MMOG operational model, the number of interruptions in isolation is not sufficient, as it does not concretely present the extent to which these failures impact the game session. We therefore present, in Fig. 8, a statistical analysis of all interruptions for two metrics: their duration and severity. Regarding the duration of the total interruptions (Fig. 8a, top), we notice a step-wise increase proportional to the resource contention, but a very stable behaviour with changing resource availability. Overall, the median time needed for recovery from a total resource failure is of two minutes (i.e. one resource allocation step) for a

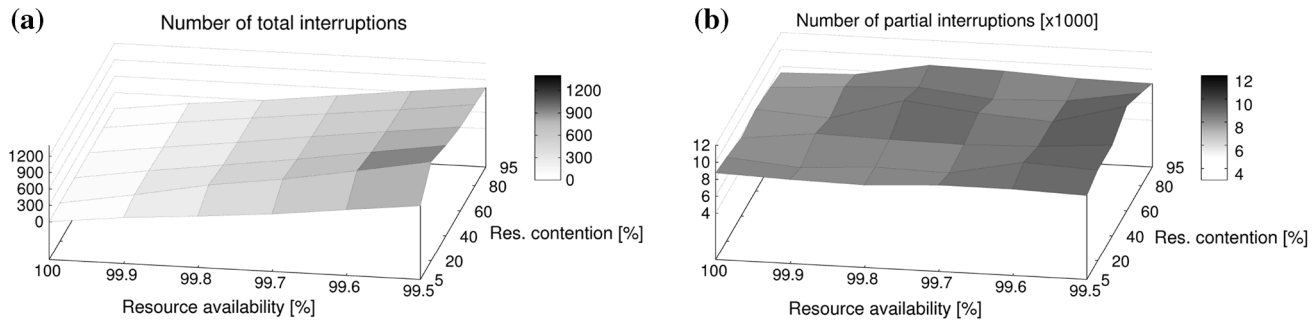

Fig. 7 Interruption events in varying resource availability and contention. (a) Total interruptions. (b) Partial interruptions

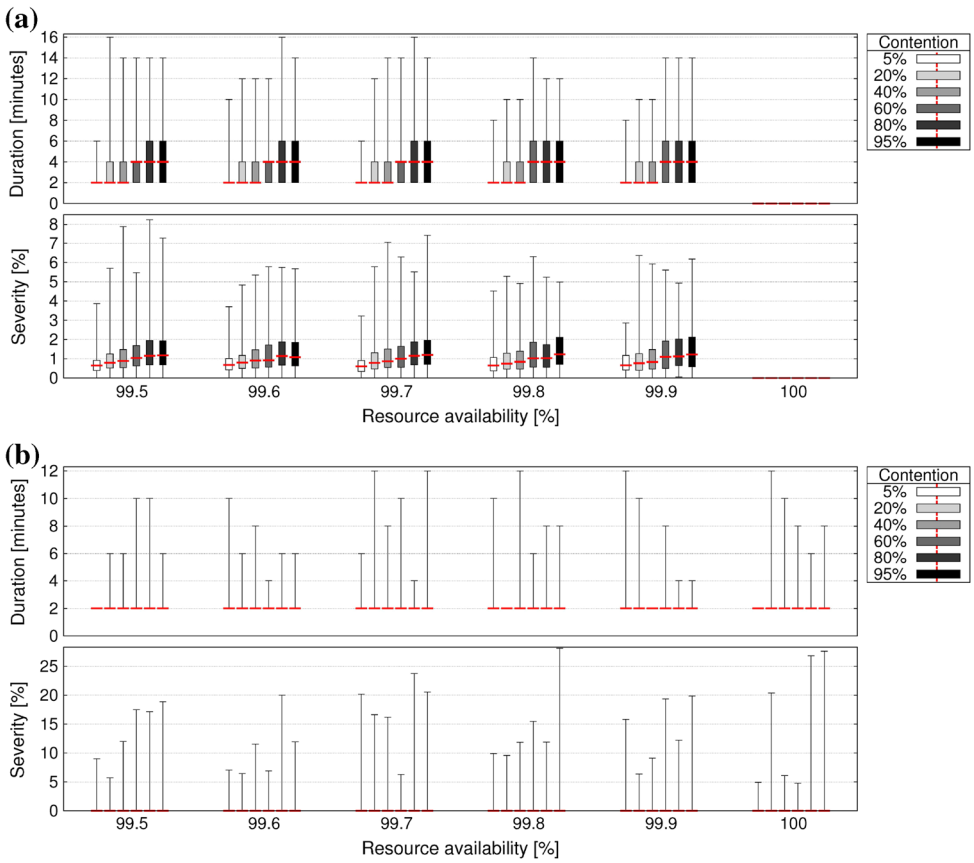

Fig. 8 Event duration and severity QoS analysis in competition-based conditions with increasing resource availability and resource contention. (a) Total interruptions. (b) Partial interruptions

Table 9 Impact of resource availability and contention on MMOG QoS

Metric	Impact on QoS					
	Total interruptions			Partial interruptions		
	Number	Duration	Severity	Number	Duration	Severity
Resource availability	Strong	None	None	None	None	None
Resource contention	Light	Strong	Strong	None	None	Light

resource contention of up to 40 %, and four minutes (i.e. two allocation steps) for higher resource contention. The step-wise variation is due to the cyclic nature of the recovery evaluation. More precisely, an MMOG session might recover, for example, in 90 s, but the evaluation of the resource allocation state is only done every 120 s. Thus, the reported duration of recovery will be 120 s. In a real implementation, this issue would be easily circumvented by employing an event-driven monitoring system. Regarding the severity of the total interruptions (Fig. 8a, bottom), we notice a gradual increase proportional to the resource contention, from a median of approximately 0.7 to 1.4 % correlated with an increase of resource contention from 5 to 95 %. This variation is similar across

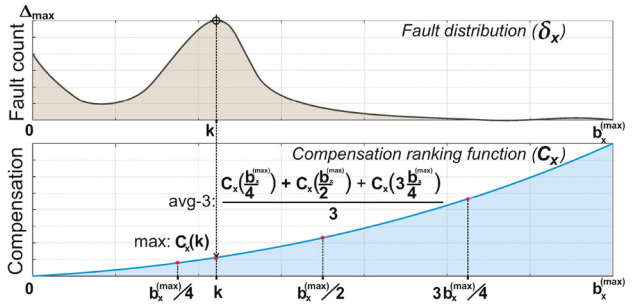

Fig. 9 *max* versus *avg-3* compensation ranking methods: *max* ranks by the compensation value $C_x(K)$ of the most frequent O-SLA fault K; *avg-3* ranks by the average compensation values of three uniformly distributed SLA faults

different resource availability levels and, regardless of the individual configuration, at least 75 % for the events affect <2 % of the clients. In contrast to the total interruptions, the behaviour of the partial interruptions' duration and severity, shown in Fig. 8b, is clearly not dependent on either of the studied metrics (resource availability and contention). The recovery time is for the vast majority of events one allocation cycle long (i.e. two minutes), with only some outliers (<5 % of events) reaching 12 minutes. The severity of the partial interruptions is also very steady at 0.01 % of the number of clients for 95 % of the events, regardless of resource availability and contention. However, a general growing trend of the outliers coherent with the increase of resource contention can be observed.

We conclude that clouds can be used for MMOG hosting with high QoS, even in cases of resource contention, as summarised in Table 9. Concretely:

1. Resource availability strongly impacts the number of total interruptions, but does not influence their duration and severity;
2. Contention for resources has a low negative impact on the number of total interruptions, but strongly affects their duration and severity;
3. Resource availability has little or no visible impact on partial interruptions, while an increased resource contention might lead to a slight increase in the severity of the partial interruptions.

6.3 Selecting the compensation ranking method

The goal of this next experiment is to study how game providers can select operational offers (sets of O-SLA instances, see Sect. 4) from game operators based on compensation terms. We study six compensation ranking methods. First, *max* (RK1) (see Table 6; Fig. 9) ranks offers by the compensation value $C_x(K)$ (see Eq. 5) corresponding to the most frequent O-SLA fault K for which $\delta_x(K) = \Delta_{\max}$ (see Eq. 16):

$$\mathcal{C}_{\text{O-SLA}}^{(\max)} = \sum_{i=1}^{M} \sum_{x \in \{\text{cli,ini,tni}\}} \psi_x \cdot \frac{C_x(K)}{C^{(\text{T})}}, \tag{22}$$

where $\delta_x(K) = \Delta_{\max}, M$ is the total number of O-SLAs in an operational offer, $C^{(\text{T})}$ is the target minutely compensation per client (see Eq. 7), and $\psi_{\text{cli}}, \psi_{\text{ini}}, \psi_{\text{tni}} \in [0;1]$ indicate the provider's preference for each O-SLA term ($\phi_{\text{cli}} + \phi_{\text{ini}} + \psi_{\text{tni}} = 1$, see Eq. 19). Second and third, *avg-3* (RK2) and *avg-9* (RK3) rank based on the average of the compensation values:

$$\mathcal{C}_{\text{O-SLA}}^{(\text{avg}-N)} = \sum_{i=1}^{M} \sum_{x \in \{\text{cli,ini,tni}\}} \psi_x \cdot \frac{\sum_{k=1}^{N} C_x \cdot \left(\frac{k}{N+1} \cdot b_x^{(\max)}\right)}{N \cdot C^{(\text{T})}}, \tag{23}$$

for $N = 3$ and $N = 9$, with uniformly distributed SLA faults (see Fig. 9). Fourth, *gain* (RK4) is based on Eq. 19, in which the compensation gain is defined as in Eq. 15 (and not 17). Fifth and sixth, *cgain-9* (RK5) and *cgain-30* (RK6) are variants of the compensation rank proposed in Eq. 19 with $N = 9$ and $N = 30$, where N is the integration granularity (i.e. number of partitions in the Riemann sum approximation) of Eq. 17. We define a separate game provider for each of the six compensation ranking methods and the same PP1 service policy (see Table 5). We further use 65 game operators, each offering a different O-SLA based on the OSLA-1 template (Table 7) differentiated by their compensation function, its shape and other parameters, as defined in Sect. 5.5. We evaluate the total compensation, representing the fraction of compensation obtained by game providers from their gross profit.

The top graph of Fig. 10 depicts the total compensation of all game providers relative to the total compensation of game providers using the basic *max* method. We observe that, while the *max, avg* and *gain* perform roughly the same (variation <3 %), the *cgain* class leads to 11–16 % increases in income from compensations. As each MMOG exhibits an individual load pattern which results in a particular fault distribution (as exemplified in Fig. 9), game providers automatically tune their offer selection using the characteristic gain ranking method to favour the O-SLAs bringing the highest compensations. We further analyse the impact of employing these methods on the QoS offered by game providers. The results depicted in the bottom graph of Fig. 10 indicate only slight QoS variations for all methods: the instantaneous non-interruption ratio t_{ini} is above the target value $s_{\text{ini}} = 0.9$ of the game providers' service policy PP1 between the first and third quartiles and the median value is nearly one (optimal). Furthermore, the average number of non-serviced clients is around eight (out of a 2,000 maximum).

We conclude that for an optimal selection of MMOG operational offers: (1) it is necessary to employ a method that accurately captures the characteristics of the offered compensations; (2) it is essential to account for the dynamic behaviour of the O-SLA faults; and (3) it is possible to significantly increase the providers' income (up to 16 %) through these offer selection methods without negative effects on the QoS.

6.4 Tuning the compensation ranking method

The goal of this experiment is to quantify the impact of the integration granularity *N of the characteristic compensation granularity ranking method cgain* on the efficiency of the operational offer selection, by evaluating the game provider's income from total compensations for different values of *N*. The experimental setup is similar to the one in the previous experiment, except for the game provider's offer ranking configurations. We use in this experiment the RK7 configuration ($N = 1$) and ten other configurations (i.e. RK8–RK17), whose integration granularity ranges from 3 to 30 with a step of 3 (see Table 6). We run a separate simulation for each ranking configuration and compute the fraction of game provider's profit representing compensations for O-SLA faults.

We observe in Fig. 11, that the compensation has a logarithmic increase with the integration granularity. This trend is not strictly monotonous because, as the integration granularity is increased, game providers select other game operators using different cloud resources, which influence the number and severity of faults leading to variations in the total compensation. Over the 6-month simulation, the RK17 provider with the highest integration granularity ($N = 30$) registered an income of approximately \$10 million from O-SLA fault compensations, which is 12 % higher than by employing RK7 ($N = 1$).

The findings of this experiment show that the best performing O-SLA ranking method, the proposed characteristic compensation gain, can be further tuned to obtain a logarithmic increase in total compensation income by increasing the integration granularity.

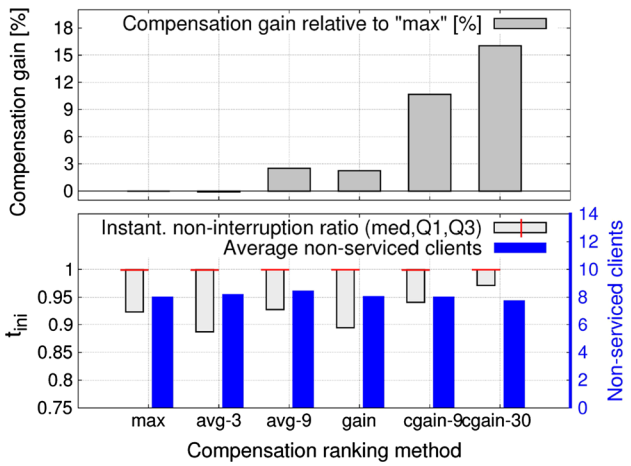

Fig. 10 Comparison of compensation ranking methods

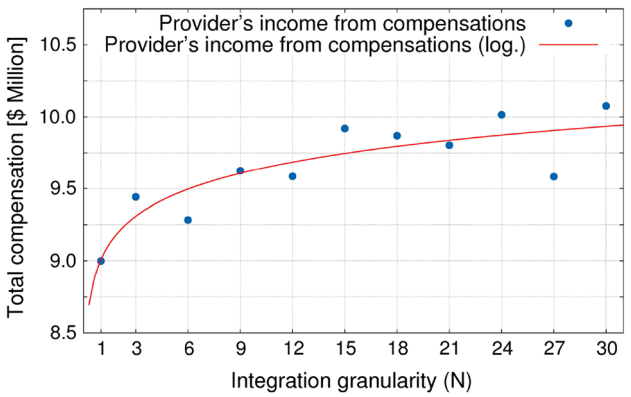

Fig. 11 Increase in provider's income from better offer compensation ranking with increasing integration granularity

6.5 Weighting the fitness ranking components

In this experiment, we analyse techniques for maximising the game providers' profit by proper operational offer selection based solely on the fitness ranking. Concretely, we determine how the game operators can weight each of the four negotiable O-SLA terms in the fitness ranking process (ϕ_x weights in Eq. 20). The setup for this experiment differs from previous setups by effectively exploring the much larger space of RK-PP policy pairs without exhaustively considering each possible pair, as follows. The game operators offer different O-SLAs generated starting from the OSLA-2 template (see Table 7) and vary the ranges of the four negotiable terms $t_x^{(O)}, x \in \{\text{cli}, \text{ini}, tni, time\}$. We use five game providers and vary the ranking configuration for all involved game providers from RK18 to RK37 (described in Table 6) over 20 simulations. We designed each of the 20 configurations to gradually increase the weight ϕ_x of one of the four negotiable O-SLA terms. For each simulation, the game providers employ different service policies, as follows. For *cli-[10;90]* and *ini-[10;90]* ranking configurations, each game provider uses one of the service policies PP2-PP6 defined in Table 5 (for the $t_{\text{cli}}^{(O)}$ term, there is no corresponding service policy term). For *tni-[10;90]* ranking configurations, each game provider uses one of the service policies PP7-PP11. Finally, for *time-[10;90]* ranking configurations, each game provider uses one of the service policies PP12-PP16. The fitness ranking process considers only those offers which meet at least the minimum operational requirements. Thus, expenses are the key part to analyse in the game providers' budget. An improper offer selection leads to over-provisioning and consequently, to higher expenses.

Figure 12 shows that the fitness ranking configuration has a significant impact on the expenses of the game providers. We observe a reduction in expenses of \$14.5

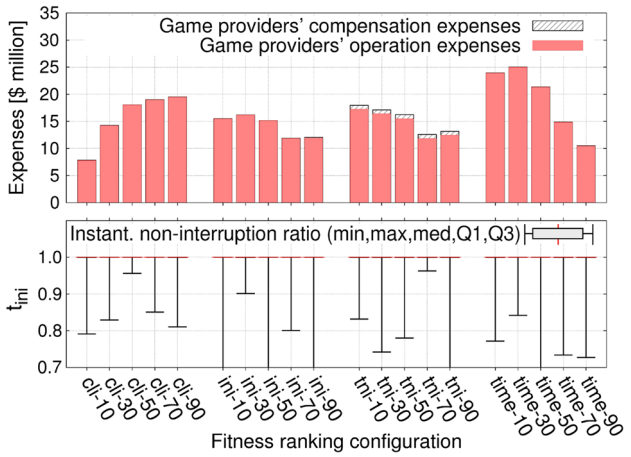

Fig. 12 Game providers expenses for different fitness ranking configurations: expenditure fraction of the game providers gross profit (*top*); instantaneous non-interruption ratio (*bottom*)

million for the case of the t_{time} validity period term (ranking configurations *time-[10;30]*) or about 60 % from the maximum expenses. In contrast, the game providers' income is approximately $37.8 million for all simulations (not shown in the graph). We further observed that an increase in the weight of the other QoS terms, namely $t_{\text{ini}}^{(\text{O})}$ and $t_{\text{tni}}^{(\text{O})}$, leads to a decrease in the game providers' expenses. Conversely, increasing of the weight of the client number $t_{\text{cli}}^{(\text{O})}$ leads to an increase of the game providers' expenses. During all simulations, the QoS provided to the clients was constantly high due to the wide range of O-SLA templates, which effectively accommodate most of the game providers' needs. The increased client compensation expenses that appear exclusively for the *tni-[10;90]* ranking configurations is also notable. Even though the O-SLA faults are not severe [see Fig. 12 (bottom)], they often account for over 0.4 % of the total O-SLA duration, which forces the game providers with the PP10-PP11 service policies promising a high total non-interruption ($s_{\text{tni}} \geq 0.996$) to compensate the clients.

We conclude that for maximising their profit, the game providers should attempt to find an optimal, Pareto-efficient balance between the client number $t_{\text{cli}}^{(\text{O})}$ and the other three negotiable terms. In our experiments, the optimal weights are $\phi_{\text{cli}} = 0.1$, $\phi_{\text{ini}} = 0.3$, $\phi_{\text{tni}} = 0.3$ and $\phi_{\text{time}} = 0.3$; however, the actual weights depend on the available O-SLA templates.

6.6 Configuring operational offer ranking in a competitive environment

The goal of this experiment is to evaluate the impact of the operational offer ranking on the game providers' gross profit in a *competitive* environment, that is, when multiple game providers compete for operational offers (O-SLAs). We aim at determining best practices guidelines for maximising the profit of game providers in a realistic setting involving competition. The setup for this experiment differs from previous setups as follows. A set of 15 game providers compete for operational offers, each employing one of the service policies PP2-PP16 (defined in Table 5). Each group of five providers employing the policies PP2-PP6, PP7-PP11, and PP12-PP16 runs a part of the RuneScape MMOG traces. We run eight simulations, each changing the ranking configuration employed by all the game providers from RK38 to RK45 (also or-[1;8] in Table 6). The compensation ranking and fitness ranking parameters are fixed to the optimal values determined in the previous experiments. By employing the or-[1;8] ranking configurations, we start with a high emphasis on the importance of the compensation ranking, and implicitly a low importance on the fitness ranking, and continue by gradually changing the emphasis until reaching the opposite scenario (from $\lambda_c = 0.8$ and $\lambda_f = 0.1$ to $\lambda_c = 0.1$ and $\lambda_f = 0.8$—see Eq. 12 and Table 6). The setup also considers a number of 74 game operators employing O-SLAs based on the OSLA-3 template (shown in Table 7), which cover the whole value range of each parameter (operational and compensation terms) and effectively offering to game providers an operator market with maximum offer diversity.

Figure 13a shows the variation of three gross profit fractions: the MMOG operation expenses, client compensation expenses, and the income from O-SLA fault compensations. The top chart presents the trend of the aggregated profit fractions for all game providers, while the bottom chart shows their proportional composition. The income from client subscriptions of all game providers is constant throughout all runs at around $113.67 million (not shown in the graphs). The increasing trend of the game providers' gross profits from or-1 to or-4 is due to a decrease in the expenses caused by client compensations and resulting from a better selection of operational offers, a consequence of the increased weight of fitness ranking. The descending trend from or-4 to or-6 is due to increased expenses with operational offers, which is a consequence of further increasing the weight of fitness ranking. These expenses are slowly being compensated by further decreases in client compensations, which eventually lead to another increasing trend for or-6 to or-8. Overall, the best gross profit value is reached when employing or-4. We observe that increasing the weight of fitness ranking leads to an increase of the operational offer expenses and to a decrease in the client compensations. Conversely, increasing the weight of the compensation rank leads to higher operational expenses, but also to an increase of the income from O-SLA fault compensations.

To analyse the impact of the operational rank weights on the game providers employing different service policies, we group them into two classes: (1) the *Low* s_{ini} class, offering

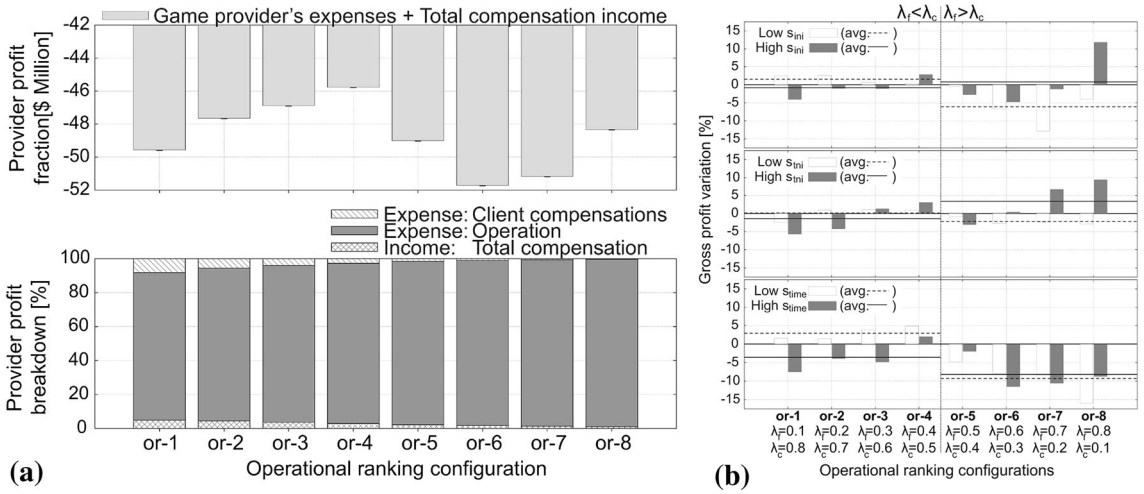

Fig. 13 Gross profit variation with different operational ranking configurations (the constant $113.67 million profit fraction from client accounts is not shown). **a** Variation of expenses and income from compensations (*top*); breakdown of profit fractions (*bottom*). **b** Gross profit variation in percentage to the average of or-4 and or-5

the clients low QoS, in terms of the targeted instantaneous non-interruption ratio (employing PP1 and PP2 with low s_{ini}), and (2) the *High* s_{ini} class, targeting a high QoS by employing PP4 and PP5 with high values for the same term. The *Low* s_{tni} and *High* s_{tni} classes and, respectively, the *Low* s_{time} and *High* s_{time} classes are constructed similarly for the other QoS term and the targeted SLA validity. Figure 13b shows the gross profit variation of the game provider classes reported to an average of the or-4 and or-5 runs. We observe that the *Low* s_{ini} class favours the lower fitness rank and higher compensation rank weights, while *High* s_{tni} is only marginally affected by changes in the operational ranking weights. The *Low* s_{tni} slightly favours lower fitness rank weights, while the *High* s_{tni} is positively influenced by the increased fitness rank and lower compensation rank weights. The strongest impact of the operational ranking weights is observed for the s_{time} term: *Low* s_{time} favours lower fitness rank weights and has a strong negative reaction to higher compensation ranking weights, while *High* s_{time} performs best when the two weights are balanced and significantly worse otherwise.

We conclude that the game providers' profit can be maximised in a competition-based environment by balancing the fitness and compensation ranking weights. However, the game providers' service policies strongly impact the influence of the operational rank weights on the profit, as summarised in Table 10.

Table 10 Best practice rank weight configurations for maximising game providers' profit

Service policy term	Low		High	
s_{ini}	$\lambda_f\downarrow$	$\lambda_c\uparrow$	$\lambda_f=$	$\lambda_c=$
s_{tni}	$\lambda_f=$	$\lambda_c=$	$\lambda_f\uparrow$	$\lambda_c\downarrow$
s_{time}	$\lambda_f\downarrow$	$\lambda_c\uparrow$	$\lambda_f=$	$\lambda_c=$

'↑', represents a high value; '↓', a low value; and '=', balanced

7 Related work

We survey in this section five large bodies of related work: cloud-based resource provisioning, SLA-based operational models for MMOGs, traditional SLA stacks for large-scale systems, reliability, and cloud-based operation of services with millions of customers.

7.1 Cloud-based resource provisioning for MMOGs

The problem of operating MMOGs in distributed computing environments with recent focus on cloud infrastructures is receiving increasing interest in the parallel and distributed systems community. Existing efforts focus on increasing the scalability of FPS games [16], latency fairness to players [22], dynamic provisioning and load balancing on distributed grid resources [2, 15], or the impact of virtualisation overheads on MMOG operation.

Closest to our work, [13] proposes a greedy dynamic provisioning algorithm for resizing the resource pool of a MMOG service to react to workload variability. In contrast, our model is more realistic in that it considers multiple clouds as resource providers, resources with different costs and capabilities, and particularly important, resources that can be unavailable. Conceptually, we consider compensation terms and their operational impact, without which we believe the use of clouds in commercial gaming

is made difficult or even impossible. Also conceptually, we consider a hybrid pro-active (prediction-based) and reactive (monitoring-based) approach to trigger system reconfiguration, whereas Marzolla et al. consider only a monitoring-based approach.

Also, close to our work is the study on QoS-aware revenue-cost optimisation algorithm for latency-sensitive services in clouds [14]. The authors consider a model similar, but more restricted (e.g. less realistic modeling of cloud resources, no period of validity for offers, no comprehensive modeling of QoS aspects besides latency and response time) than the one proposed by us in an early work on this topic [3]. They do not consider any of the specific improvements we made in this work to our previous model, including penalties for breaches of services or resource availability. The experiments in [14] consider only a single cloud and no cost variation. In contrast, our results indicate that significant cost differences can be observed in a multi-cloud environment due to various resource capabilities and pricing.

To conclude: in contrast to previous work on cloud-based MMOG operation, ours is the first to study MMOG-related SLA terms as a basis for the negotiation and QoS-based provisioning of resources to MMOG servers, including compensations (penalties) for QoS violations and fitness ranking of SLA offers, and across an ecosystem that includes multiple service providers at each level. Several specific advances, both conceptual (the complex O-SLA relationship between the game provider and the game operator, the hybrid reactive-proactive reconfiguration, etc.) and technical (the experimental study in a multi-cloud setting, including unavailability of resources), significantly differentiate our work from previous studies. Last, but not least, we are the first to evaluate comprehensively both the technical operation of MMORPGs (with various metrics related to game-level QoS) and the financial operation of these games (with profit- and compensation-related metrics). Our comprehensive approach shows evidence of the complexity of interactions in a realistic, competitive, multi-service provider environment.

7.2 SLA-based operational models for MMOGs

Much recent work focusses on (soft) QoS guarantees for MMOG operation [23–25]. Wong [23] proposes a resource provisioning algorithm with QoS guarantees, but considers only networking aspects, whereas we consider here more resource types and a more realistic cloud model (including cloud and virtualisation overheads [3]). Briceno et al. [24] study resource allocation for MMOGs but, unlike our work, consider only computational requirements and use a simplified workload model (not traces from a real MMOG). Lee and Chen [25] investigate MMOG server consolidation techniques focussing on complementary energy consumption issues. We also extend our previous work [3] with the formulation, negotiation, and use of SLAs for MMOGs.

Several market-based operational models for MMOGs have been recently investigated. Our proposed business model for MMOGs is closest in concept to the four-actor business model of Middleton et al. [26], but studies in addition the connection between the business and hosting models, and methods for controlling the provided QoS. Complex business models are proposed by Alves et al. [27] and by Andersson et al. [28, 29], but focus only on higher level business interactions and goals for MMOG operation. In contrast, we study a novel operational model and its effects on the profits of both game and resource operators. Complementary to our work, Nojima [30] studies the relationship between pricing models and MMOG player motivation, and Oh and Ryu [31] analyse different pricing models for the gaming service.

7.3 Traditional SLA stacks for large-scale systems

Much work has focussed on SLA stacks since at least the early 1980s [32]. Two recent surveys focus on SLA stacks for grids [5] and clouds [6]. Complete SLA stacks have been implemented, for example, by Galaxy [11], Oceano [12], Globus [7], NextGrid [33], and SLA@SOI [4]. Close to our work is also the study of Wu et al. [8]. In contrast to these approaches, we focus on a (popular) domain-specific application, for which we extend traditional approaches with MMOG-specific considerations, propose a comprehensive SLA formalism, and investigate specific operational policies.

The *formulation* of SLAs has received considerable attention, notably from standardisation bodies such as W3C, OASIS, and OGF. The resulting formalisms OGSA [7], WSLA [34], and WS-Agreement [10] and its extensions are general but, due to size and complexity, difficult to implement and map to each other [4]. We believe that our SLA formalism offers a better tradeoff between coverage and implementation that eases the MMOG operation by omitting elements that are not essential in this context. Our work also differs from previous *negotiation* approaches [7, 35–38] through a focus on MMOGs: detailed SLA key performance indicators, a mechanism for ranking offers based on several elements including penalties, and several classes of MMOG service and compensation policies. Similar to the previous work on SLA-based *scheduling* [8, 39–43], our work considers a three-stage mechanism: the MMOG operator first auto-generates SLAs, then finds and ranks game operator offers (also considering penalties for SLA violation), and finally renegotiates with the game operator. In addition, our work proposes a new ranking mechanism adapted to MMOGs

and non-uniform SLA violations, investigates multiple game and multiple cloud operators, and focusses on various compensation and other SLA policies.

7.4 Reliability

There have been a number of research activities in assessing the performance of virtualised resources in cloud computing environments, some also considering the availability of cloud resources [44]. In contrast to these studies, ours targets realistic computational cloud resources with limited availability for a new application class (MMOG). In the area of reliability, there are studies which investigate the characteristics of resource and workload failures, but do not assess their effects on the underlying systems' performance [45, 46]. Others consider uncorrelated failures in distributed systems [47] and evaluate the resulting performance of the affected systems [21], but restricted to high-performance computing jobs. In contrast, we employ the failure model introduced in [21], apply it to cloud resources, and evaluate the consequences of utilising such resources on the QoS of MMOGs.

7.5 Cloud-based operation of services with millions of customers

The entertainment industry has already started to migrate from the in-house to cloud-based infrastructure. Zynga, which operated in 2011 online gaming services for over 250 million users, uses Amazon EC2 resources for operating games up to several months after their launch. Nevertheless, the games supported by Zynga require much less computational and network resources than MMOGs. On-demand gaming, which offloads gaming computation to the cloud and streams back to remote clients the video output of the game, is provided by companies such as Geelix [48], OnLive, Gaikai, and OTOY. We do not consider this game operational model because major MMOG operators have yet to switch to this model, in part due to the high network requirements imposed on the players. Since late 2011, Amazon Web Services has been used by Netflix for video streaming and for offloading web browsing for mobile devices with Android operating system. In contrast to these approaches, our work adapts this model to the specifics of MMOGs and proposes an in-depth study of a variety of scenarios applicable to other branches of the entertainment industry.

8 Conclusion

The current MMOG ecosystem, which includes tens of millions of players across hundreds of games, forces game providers to also become game and infrastructure operators. This leads, in general, to inefficient resource utilisation, high service prices, and limits market participation to only the largest game providers. In this work, we proposed a new ecosystem and middleware model for hosting and operating MMOGs based on cloud-computing principles, focussing on the formulation and negotiation of SLAs that encompass price, operational terms and novel compensation policies. In our model, game operators efficiently provision cloud resources for MMOGs based on their dynamic load and ensure proper game operation that maintains the required QoS to all clients. Game providers lease operation SLAs from the game operators to satisfy all client requests and manage multiple distributed MMOG sessions. These three self-standing, smaller, more agile service providers enable access to the MMOG market for the small and medium enterprises, and to the current commercial cloud providers. For ranking MMOG operational offers, our model balances among three criteria: pricing, fitness for operation, and compensation. For each criterion, we provided comprehensive ranking mechanisms. We evaluated the operation of the proposed MMOG ecosystem, which can include multiple service providers of various kinds, in a variety of scenarios through realistic simulations using traces collected from real MMOGs and real SLAs from over ten commercial clouds. Our main findings in this paper are:

1. Regarding the QoS impact of unreliable resources on MMOG operation:
   (a) Our MMOG ecosystem successfully mitigates the performance degradation of running MMOGs on cloud resources with limited availability to game play disruptions of <4 min, independently of the duration of the underlying resource failure;
   (b) The majority of resource failures affect <2 % of the clients participating in autonomously operated MMOG sessions;
   (c) A low resource availability increases the number of game play disruptions, while a high resource contention results in longer disruptions affecting more clients.
2. Regarding the business impact of resource and allocation failures:
   (a) For the compensation criterion, a ranking method which considers the yield from compensations in the given environment is necessary. Our proposed approach to this problem called characteristic compensation gain leads to 11–16 % higher financial gain without QoS deterioration;
   (b) For the fitness of operation criterion, tuning the four operational terms to reflect the MMOG

generated load can lead to a 20–60 % reduction in the operational expenses compared to the non-tuned approach;

(c) For the overall ranking of operational offers, we provide guidelines for balancing the three criteria and find that their impact depends on the service terms used in the MMOG provider–client (player) relation.

Acknowledgments Austrian Science Fund (FWF) project TRP 72-N23 funded this research.

References

1. MMOData.net: Mmodata.net keeping track of the mmog scene (v3.4). Online. Available: http://mmodata.net/ (2011). http://mmodata.net/
2. Nae, V., Iosup, A., Prodan, R.: Dynamic resource provisioning in massively multiplayer online games. IEEE Trans. Parallel Distrib. Syst. **22**(3), 380–395 (2011). doi:10.1109/TPDS.2010.82
3. Nae, V., Prodan, R., Iosup, A., Fahringer, T.: A new business model for massively multiplayer online games. In: Proceeding of the Second Joint WOSP/SIPEW International Conference on Performance Engineering, pp. 271–282. ACM, New York, (2011). doi:10.1145/1958746.1958785
4. Chronz, P., Wieder, P.: Integrating WS-Agreement with a framework for service-oriented infrastructures. In: 11th IEEE/ACM International Conference on Grid Computing, pp. 225–232, IEEE (2010)
5. Talia, D., Yahyapour, R., Ziegler, W., Wieder, P., Seidel, J., Waldrich, O., Ziegler, W., Yahyapour, R.: Using SLA for resource management and scheduling—a survey. In: Talia, D., Yahyapour, R., Ziegler W. (eds.) Grid Middleware and Services: Challenges and Solutions, pp. 335–347. Springer, Berlin (2008)
6. Wu, L., Buyya, R.: Service level agreement (sla) in utility computing systems. CoRR **abs/1010.2881** (2010)
7. Czajkowski, K., Foster, I.T., Kesselman, C., Sander, V., Tuecke, S.: SNAP: a protocol for negotiating service level agreements and coordinating resource management in distributed systems. In: Revised papers from the 8th International Workshop on Job Scheduling Strategies for Parallel Processing, pp. 153–183. Springer-Verlag, London, UK (2002)
8. Wu, L., Garg, S.K., Buyya, R.: SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments. In: 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 195–204, IEEE Computer Society Washington, DC, USA (2011)
9. Iosup, A., Yigitbasi, N., Epema, D.H.J.: On the performance variability of production cloud services. In: 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 104–113, IEEE Computer Society Washington, DC, USA (2011)
10. Andrieux A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web services agreement specification (WS-Agreement). Protocol, v.1.0, OGF, GRAAP Work Group (2007). Available online: http://www.ogf.org/documents/GFD.107.pdf
11. Vogels, W.D.D.: An overview of the galaxy management framework for scalable enterprise cluster computing. In: International Conference on Cluster Computing, pp. 109–118, IEEE (2000)
12. Appleby, K., et al.: Océano—SLA based management of a computing utility. In: IEEE/IFIP International Symposium on Integrated Network Management, pp. 855–868, IEEE (2001)
13. Marzolla, M., Ferretti, S., D'Angelo, G.: Dynamic resource provisioning for cloud-based gaming infrastructures. ACM Comput. Entertain. **10**(3), 4:1–4:20 (2012)
14. Duong, T.N.B., Li, X., Goh, R.S.M., Tang, X., Cai, W.: Qos-aware revenue-cost optimization for latency-sensitive services in iaas clouds. In: 16th International Symposium on Distributed Simulation and Real Time Applications, pp. 11–18. IEEE Computer Society (2012)
15. Nae, V., Iosup, A., Podlipnig, S., Prodan, R., Epema, D.H.J., Fahringer, T.: Efficient management of data center resources for massively multiplayer online games. In: ACM/IEEE Conference on Supercomputing, Article No. 10, IEEE Press Piscataway, NJ, USA, p. 10 (2008)
16. Glinka, F., Ploss, A., Müller-Iden, J., Gorlatch, S.: RTF: a real-time framework for developing scalable multiplayer online games. In: NetGames '07 6th ACM SIGCOMM workshop on Network and system support for games, pp. 81–86. ACM (2007). doi:10.1145/1326257.1326272
17. Prodan, R., Nae, V.: Prediction-based real-time resource provisioning for massively multiplayer online games. Future Gener. Comput. Syst. (FGCS) **25**(7), 785–793, Elsevier (2009). doi:10.1016/j.future.2009.03.003. http://www.sciencedirect.com/science/article/B6V06-4V0MJ6H-1/2/45b9cd3f8de8d176aebbeb9594231a1d
18. Nae, V., Iosup, A., Prodan, R., Fahringer, T.: The impact of virtualization on the performance of massively multiplayer online games. In: NetGames'09 proceedings of the 8th Annual Workshop on Network and systems support for games, Article No. 9, IEEE Press Piscataway, NJ, USA (2009). doi:10.1109/NETGAMES.2009.5446227. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5446227
19. Iosup, A., Nae, V., Prodan, R.: The impact of virtualization on the performance and operational costs of massively multiplayer online games. Int. J. Adv. Media Commun. (IJAMC) **4**, 364–386 (2011). doi:10.1504/IJAMC.2010.036836
20. Iosup, A., Ostermann, S., Yigitbasi, N., Prodan, R., Fahringer, T., Epema, D.H.J.: Performance analysis of cloud computing services for many-tasks scientific computing. IEEE Trans. Parallel Distrib. Syst. **22**(6), 931–945 (2011)
21. Iosup, A., Jan, M., Sonmez, O., Epema, D.H.J.: On the dynamic resource availability in grids. In: 8th IEEE/ACM International Conference on Grid Computing, pp. 26–33. IEEE Computer Society (2007). doi:10.1109/GRID.2007.4354112
22. Kohana, M., Okamoto, S., Ikegami, A.: Optimal data allocation for keeping fairness of online game. In: 26th International Conference on Advanced Information Networking and Applications Workshops, pp. 1209–1214. IEEE Computer Society (2012)
23. Wong, K.: Resource allocation for massively multiplayer online games using fuzzy linear assignment technique. In: 5th IEEE Consumer Communications and Networking Conference, pp. 1035–1039. IEEE (2008)
24. Briceño, L.D., et al.: Robust resource allocation in a massive multiplayer online gaming environment. In: 4th International Conference on Foundations of Digital Games, pp. 232–239. ACM (2009). doi:10.1145/1536513.1536556
25. Lee, Y.T., Chen, K.T.: Is server consolidation beneficial to MMORPG? A case study of World of Warcraft. Cloud Computing, IEEE International Conference on Cloud Computing, pp. 435–442, IEEE Computer Society Washington, DC, USA (2010). doi:10.1109/CLOUD.2010.57
26. Middleton, S., Surridge, M., Nasser, B., Yang, X.: Bipartite electronic SLA as a business framework to support cross-organization load management of real-time online applications. In: Lin, H.-X., Alexander, M., Forsell, M., Knüpfer, A., Prodan, R., Sousa, L. and Streit, A. (eds.) Euro-Par 2009 – Parallel

Processing Workshops, Lecture Notes in Computer Science, vol. 6043, 2010, pp 245–254, Springer, Berlin, Heidelberg (2009)
27. Alves, T., Roque, L.: Using value nets to map emerging business models in massively multiplayer online games. In: Ninth Pacific Asia Conference on Information Systems, pp. 1356–1367, PACIS proceedings. Available online: http://www.pacis-net.org/ (2005)
28. Andersson, B., Bergholtz, M., Edirisuriya, M., Ilayperuma, T., Jayaweera, P., Johannesson, P., Zdravkovic, J.: On the alignment of goal models and business models. In: A Celebration of the REA Enterprise Ontology, Stockholm University, Faculty of Social Sciences, Department of Computer and Systems Sciences. Available online: http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-12144 (2007)
29. Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Jayaweera, P., Johannesson, P., Zdravkovic, J.: Enterprise sustainability through the alignment of goal models and business models. In: 3rd International Workshop on Business/IT Alignment and Interoperability, vol. 336, pp. 73–87, CAiSE 2008, Montpelier, June (2008)
30. Nojima, M.: Pricing models and motivations for MMO play. In: B. Akira (ed.) Situated Play: Proceedings of the 2007 Digital Games Research Association Conference, pp. 672–681. Tokyo (2007)
31. Oh, G., Ryu, T.: Game design on item-selling based payment model in Korean online games. In: Akira, B. (ed.) Situated Play: Digital Games Research Association Conference, pp. 650–657. Tokyo (2007)
32. Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans. Comput. **29**(12), 1104–1113 (1980)
33. Hasselmeyer, P., Mersch, H., Koller, B., Quyen, H.N., Schubert, L., Wieder, P.: Implementing an SLA negotiation framework. In: Cunningham, P., Cunningham, M., (eds.) Expanding the Knowledge Economy: Issues, Applications, Case Studies, ISBN: 978-1586038014, pp. 154–161 (2007)
34. Keller, A., Ludwig, H.: The WSLA framework: Specifying and monitoring service level agreements for web services. J. Netw. Syst. Manage. **11**(1), 57–81, (2003)
35. Siddiqui, M., Villazón, A., Fahringer, T.: Grid allocation and reservation—grid capacity planning with negotiation-based advance reservation for optimized QoS. In: ACM/IEEE Conference on Supercomputing, p. 103, ACM, New York, NY, USA (2006)
36. Yan, J., Kowalczyk, R., Lin, J., Chhetri, M.B., Goh, S., Zhang, J.Y.: Autonomous service level agreement negotiation for service composition provision. Future Gener. Comp. Syst. **23**(6), 748–759 (2007)
37. Kertész, A., Kecskemeti, G., Brandic, I.: Autonomic SLA-aware service virtualization for distributed systems. In: 19th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, pp. 503–510, IEEE Computer Society (2011)
38. Netto, M.A.S.: Canpro: a conflict-aware protocol for negotiation of cloud resources and services. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R., (eds.) Service-Oriented Computing, Lecture Notes in Computer Science, vol. 7084, pp. 541–548, Springer, Berlin, Heidelberg (2011)
39. Irwin, D.E., Grit, L.E., Chase, J.S.: Balancing risk and reward in a market-based task service. In: 13th IEEE International Symposium on High Performance Distributed Computing, pp. 160–169, IEEE Computer Society Washington, DC, USA (2004)
40. In, J.-U.K., Avery, P., Cavanaugh, R., Ranka, S.: Policy based scheduling for simple quality of service in grid computing. In: 18th International Parallel and Distributed Processing Symposium, IEEE Computer Society (2004)
41. Ranjan, R., Harwood, A., Buyya, R.: SLA-based coordinated superscheduling scheme for computational grids. In: IEEE International Conference on Cluster Computing, p. 8, (2006)
42. Dumitrescu, C., Raicu, I., Foster, I.T.: The design, usage, and performance of GRUBER: a grid usage service level agreement based brokering infrastructure. J. Grid Comput. **5**, 99–126 (2007)
43. Netto, M.A.S., Bubendorfer, K., Buyya, R.: SLA-based advance reservations with flexible and adaptive time qos parameters. In: 5th International Conference on Service-Oriented Computing, pp. 119–131, Springer-Verlag, Berlin, Heidelberg (2007)
44. Nagarajan, A.B., Mueller, F., Engelmann, C., Scott, S.L.: Proactive fault tolerance for hpc with xen virtualization. In: Proceedings of the 21st Annual International Conference on Supercomputing, ICS '07, pp. 23–32. ACM (2007). doi:10.1145/1274971.1274978
45. Nurmi, D., Brevik, J., Wolski, R.: Modeling machine availability in enterprise and wide-area distributed computing environments. In: Euro-Par 2005 Parallel Processing, LNCS, vol. 3648, pp. 612–612. Springer (2005). doi:10.1007/11549468_50
46. Schroeder, B., Gibson, G.: A large-scale study of failures in high-performance computing systems. IEEE Trans. Dependable Secur. Comput. **7**(4), 337 –351 (2010). doi:10.1109/TDSC.2009.4
47. Bhagwan, R., Savage, S., Voelker, G.: Understanding availability. In: Peer-to-Peer Systems II, LNCS, vol. 2735, pp. 256–267. Springer (2003). doi:10.1007/978-3-540-45172-3_24
48. Holthe, O.I., Mogstad, O., Ronningen, L.A.: Geelix LiveGames: Remote playing of video games. In: 6th IEEE Consumer Communications and Networking Conference (CCNC) (2009). pp. 758–759, IEEE Press Piscataway, NJ, USA, ISBN:978-1-4244-2308-8