Autonomic Operation of Massively Multiplayer Online Games in Clouds

Vlad Nae, Radu Prodan
Institute of Computer Science
University of Innsbruck
Technikerstr. 21a, A-6020 Innsbruck, Austria
vlad, radu @dps.uibk.ac.at

Alexandru Iosup
Dept. of Software and Computer Technology
Delft University of Technology
Mekelweg 4, 2628 CD, Delft, Netherlands
A.Iosup@tudelft.nl

ABSTRACT

To support the variable load of Massively Multiplayer Online Games (MMOGs) with millions of registered users and thousands of active concurrent players, game operators overprovision a large static infrastructure capable of sustaining the peak load with guaranteed Quality of Service (QoS). This leads to inefficient resource utilisation, high service prices, and limited market participation accessible only to the large companies. To address this problem, we propose a new autonomic ecosystem for hosting and operating MMOGs based on cloud computing principles involving four smaller and better focused business actors whose interaction is regulated through Service Level Agreements (SLAs): resource provider, game operator, game provider, and client. In our model, game providers acquire operation SLAs from game operators to satisfy client requests and manage multiple distributed MMOG sessions. Game operators lease on-demand cloud resources based on the dynamic MMOG load and guarantee the required QoS to all clients. We evaluate through simulations based on real MMOG traces and commercial cloud SLAs different methods of ranking MMOG operation offers. We show that considering compensations for SLA faults in the offer selection can lead to over 11% gains in game providers' income, and that adequate ranking of offers can reduce operational costs by up to 60%.

Categories and Subject Descriptors

K.6.2 [**Management of Computing and Information Systems**]: Installation Management—*Pricing and resource allocation*; I.6.8 [**Simulation and Modelling**]: Types of Simulation—*Gaming*

General Terms

Management, Economics

Keywords

MMOG, SLA, QoS, autonomic cloud computing

CAC'13, August 5–9, 2013, Miami, FL, USA.

1. INTRODUCTION

Massively Multiplayer Online Games (MMOGs) are a new type of large-scale distributed application characterised by seamless virtual worlds with millions of world-wide players interacting in real-time. Although for the past decade the number of MMOG players has grown exponentially to the current tens of millions, this very growth may now hamper the progress of this important branch of the entertainment business. Today, most MMOG companies have to be both game providers by developing the creative part of games, and game operators by over-provisioning a multi-server infrastructure, using for this operation up to 40% of the total game revenues in an annual market of over 24 billion dollars. For example, Blizzard spends approximately $50 million every year just for the upkeep of the World of Warcraft infrastructure. An important reason for this approach is the lack of business models and supporting autonomic middleware to enable outsourcing the operation of MMOGs.

Cloud computing promises to solve the infrastructure problems of the MMOG ecosystem through on-demand resource leasing under contractual terms specified in form of *Service Level Agreements (SLAs)*. By leveraging this new infrastructure model, companies such as MMOG operators may avoid the large costs of buying and maintaining depreciable hardware, and can join the MMOG operation market with nearly zero initial investment. Although SLAs are a classic and well-studied [17, 19] mechanism for specifying and managing strict user requirements in distributed systems, an autonomous middleware for MMOGs still does not exist, as many of the current approaches are complex to implement and difficult to map to specific application domains [4]. We have tackled in [11, 12] many of the technical challenges of on-demand provisioning and allocation of resources to MMOGs under Quality of Service (QoS) constraints. However, commercial clouds still cannot readily be used for MMOG operation, as their SLAs mostly focus on hardware characteristics and lack support for negotiating MMOG-friendly SLAs. Moreover, although focused on infrastructure hardware, even large commercial cloud providers, such as Amazon and Microsoft, have experienced black-outs and variable performance over the past years. Thus, game operators cannot outsource their infrastructure services without a clear specification of responsibility and of penalties associated with risks of unavailability, lower performance, and other QoS violations.

We introduce a new *autonomic MMOG ecosystem* that extends our previous work [12] with a comprehensive SLA specification and negotiation mechanism between two ac-

tors: (1) *game providers* that negotiate operation SLAs with game operators to satisfy all clients' requests, and (2) *game operators* that efficiently lease and provision resources to MMOGs from cloud providers based on their dynamic load so that the required QoS parameters are maintained to all clients. Our proposed ecosystem includes a comprehensive specification of the SLA terms underneath the negotiation protocol, and of compensations for temporary QoS violations introduced by the stringent QoS requirements and the dynamic nature of MMOGs. On top of this, we propose a method for ranking operational offers based on price, compensation and resource fitness, and study its impact on game provider's profit in an environment with several providers competing for SLAs from multiple game operators.

The main contributions of our work are: (1) we propose a comprehensive SLA-based middleware for the MMOG application domain (in Section 3); (2) we design an SLA-based negotiation protocol between the game providers, who offer MMOGs to the clients, and the game operators, who rent appropriate resources to fulfill QoS requirements (in Section 4); (3) we investigate various SLA-based policies and the operation of a sample yet complete MMOG ecosystem. Our experimental approach is simulation-based, but uses real six-month-long MMOG traces and the real SLAs of over ten commercial clouds (in Section 5).

2. MODEL

In this section, we introduce the computational model, autonomic ecosystem, and QoS metrics for MMOGs.

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 and are mapped to one avatar in the game world to whom they send their play actions and receive appropriate responses. Based on the actions sent, the avatar dynamically interacts with other avatars within a *game session*, influencing each others' state. The state update responses must be delivered within a given time frequency to ensure a smooth and responsive experience. The load of the game server is proportional 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.

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.

2.2 Autonomic MMOG Ecosystem

We propose a new autonomic business ecosystem for distributed MMOG operation and provisioning consisting of three actors: game providers, game operators, and resource providers (see Figure 1). The interaction between these actors is negotiated and regulated through bipartite SLAs, representing wrappers around QoS parameters which they agree to deliver (e.g. state-update rate for a certain price).

Game providers offer a selection of MMOGs by contracting new games from development companies (this offline interaction is not covered here). Based on the clients' requests, game providers assign clients to game zones, which are delegated to game operators for QoS-based execution. The quality of game play (see Section 2.3) is monitored by the MMOG client program and, in case of *SLA faults* (e.g. state update rate below the minimum threshold), the client is compensated.

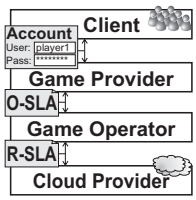

Figure 1: Autonomic MMOG ecosystem.

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 [11], the game operators construct SLA templates, negotiate SLAs with the game providers, and allocate resources accordingly (i.e. start new zones, allow client connections). This interaction is detailed in Sections 3.2 and 4. To fulfill their agreements with game providers, game operators acquire the correct amount of resources from cloud providers. At predefined *measurement timesteps* during the game play, the game provider analyses the QoS information from the MMOG servers (see Section 2.3) and, whenever SLA faults are detected, they are compensated by the operators.

Resource Providers are data centres such as *Infrastructure-as-a-Service (IaaS)* clouds that lease computing and storage resources to game operators for running game servers with guaranteed QoS. We studied in [11, 12] the opportunity of employing IaaS clouds 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 cost penalties.

2.3 QoS Metrics

One main challenge in our ecosystem is mapping the QoS requirements of MMOGs to SLA contracts which can only be enforced through best-effort mechanisms using today's cloud or Internet-based resource allocation mechanisms. We define two important QoS metrics for the quality of game play.

Instantaneous non-interruption ratio represents 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: $\frac{36}{40} = 90\%$.

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, of 24 hours, only 23.98 hours of game play during which the MMOG session was accessible and the state update rate was above the minimal frequency, the total non-interruption ratio is: $\frac{23.98}{24} = 99.9\%$.

3. SLA-BASED RELATIONSHIPS

We present now the business relationships between the actors in our MMOG operational 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 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 for particular MMOGs 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 in case of contract violations.

3.2 Game Provider and Game Operator

The interaction between the game provider and the game operator is fully automated. Based on the total number of accounts and its service policy, the game provider estimates the maximum number of clients for each game zone. We define the provider's *service policy* as a quintuple:

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

with the terms: target instantaneous non-interruption ratio s_{ini}, the total non-interruption ratio s_{tni}, the interval of acceptable SLA validity periods s_{time} to avoid excessively long operation agreements, target hourly price per client $P^{(T)}$, and target compensation per client per minute $C^{(T)}$.

To implement their service policies, the game providers negotiate the most appropriate terms for hosting each zone by establishing *Operation SLAs (O-SLA)* with different operators (see Section 4). In turn, operators publish comprehensive *O-SLA templates*, which provide detailed operation information, and compete for selection by the game providers:

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

consisting of eight terms with scalar or range values: (1) *MMOG name* and *version* $G^{(O)}_{type}$; (2) *client count* $t^{(O)}_{cli}$ (range) that the game operator is ready to service; (3) *instantaneous non-interruption ratio* $t^{(O)}_{ini}$ (range); (4) *validity period* $t^{(O)}_{time}$ (range) representing the SLA lifetime offered by the game operator, with typical granularity from daily to semestrial; (5) *total non-interruption ratio* $t^{(O)}_{tni}$ (range) representing the percentage of QoS fulfillment for the entire O-SLA validity time, to be evaluated only after the O-SLA validity period; (6) *geographical area* $\sigma^{(O)}$ in which the game operator will service the clients; (7) *base price* $P^{(O)}$ for accepting an SLA supporting the lowest values of terms 2–5; (8) the *compensation* $C^{(O)}$ for violating an SLA temporarily. For simplicity, 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 here.

The compensation term $C^{(O)}$ is an important contribution of this work. We define compensation as the aggregate penalty $C^{(O)} = \mathbb{P}\left(C_{cli}, C_{ini}, C_{tni}\right)$, where $\mathbb{P}$ represents a polynomial aggregation function that the operator has to pay in case of O-SLA faults. $C^{(O)}$ consists of three QoS-related components: client number compensation C_{cli}, instantaneous non-interruption ratio compensation C_{ini}, and total non-interruption ratio compensation C_{tni}. O-SLA policies can be made more attractive through $C^{(O)}$, for example, expensive O-SLAs can make the C_{ini} compensation term more significant. In this work, we only consider the additive function $\mathbb{P}\left(C_{cli}, C_{ini}, C_{tni}\right) = C_{cli} + C_{ini} + C_{tni}$ with each term is expressed through a *compensation function*:

$$C_x : \left[0; b^{(\max)}_x\right] \to \mathbb{R}^+,\ C_x\left(b_x\right) = \frac{c^{(u)}_x \cdot b_x}{u_x} \cdot f_x\left(\frac{b_x}{b^{(\max)}_x}\right), \tag{3}$$

where $x \in \{cli, ini, tni\}$), $\mathbb{R}^+$ is the set of positive real numbers, $c^{(u)}_x$ the compensation for an O-SLA fault of one *term unit* u_x, b_x represents the fault severity for the term x, $b^{(\max)}_x$ is its maximum possible fault severity, and f_x is a *shape function* with the signature:

$$f_x : [0; 1] \to \mathbb{R}^+, \tag{4}$$

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. For example, we can define a class of logarithmic parameterised shape functions:

$$f_x\left(b_x\right) = \frac{\log\left(a \cdot b_x + 1\right)}{\log\left(a + 1\right)}, \tag{5}$$

where $a \in \mathbb{R}^+$ is a coefficient shaping the distribution of compensations for different fault severities attempting to make the O-SLAs more appealing to the game providers.

3.3 Game Operator and Resource Provider

The business interaction between the game operator and the resource provider is also fully automated. As mentioned in Section 2.2, the game operator selects from different resource providers appropriate cloud resources to run the MMOG zones. The result of this interaction is a *Resource SLA* (R-SLA) with six terms: (1) *issuer* or the resource (cloud) 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 time for which the resources are available to the game operator from the time the R-SLA is accepted (usually hourly-grained and seldom weekly or monthly-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, the game operator employs a simple request-offer matching algorithm instead of a complex negotiation. However, the terms offered by current cloud providers, such as the Amazon "ECU" defined as "the equivalent CPU capacity of a $1 - 1.2$ Gigahertz 2007 Opteron or Xeon processor", or the FlexiScale "vCPU" units, are not precise. Thus, the resource descriptions in the R-SLA resource bulk term cannot be precise too. In turn, we cannot define finer-grained compensation terms other than for resource downtime, for example for lower processor performance or network bandwidth. Our approach to making offers more precise is to use application-specific benchmarks, such as the *RS unit* benchmark employed in Section 5.1, to quantify the performance offered by new cloud providers before establishing R-SLAs.

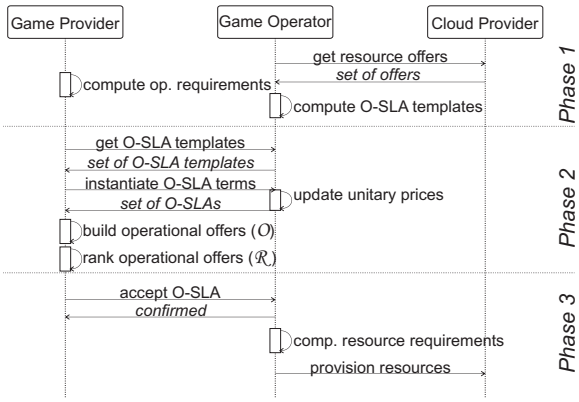

Figure 2: O-SLA negotiation protocol.

4. O-SLA NEGOTIATION

A key component of our middleware model is the negotiation between game operators and game providers, which we design as a decision process in which two parties interact with each other for mutual gain. The goal of game providers and operators is to maximise income and keep expenditures low. The game provider's income includes 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 paid to the clients when providing low QoS. Conversely, the game operator's income results from the O-SLAs provisioned to the game provider. Operator expenditures include the renting of resources from the cloud providers and the O-SLA compensations to the game providers. The accounting, billing and auditing aspects of SLA management fall outside the scope of this work, but solutions already exist.

The three negotiation phases depicted in Figure 2 cover the game providers specifying their operational requirements (phase 1), the game providers instantiating and ranking O-SLA offers (phase 2), and the binding agreement (phase 3). A simpler approach like a one-phase request-offer matching algorithm, although desirable, cannot be employed because it would not be fair towards the game operators. The resource providers' pricing policies can change during the time between the operator publishing an offer and the game provider accepting it, which would enable a game provider to profit from delaying the answer to an offer. Introducing a validity deadline for offers to prevent this unfair behaviour could have negative effects on both game provider and game operator actors, as one might not have enough time for the ranking process, while the other would have to assume the risk of the cloud resources price changes (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 final adjustments in price during the second O-SLA instantiation phase.

4.1 Phase 1: Operational Requirements

In the first phase of the negotiation, the game provider computes the *operational requirements*:

$$R = \left(G^{(R)}_{type}, t^{(R)}_{cli}, t^{(R)}_{ini}, t^{(R)}_{tni}, \sigma^{(R)}, P^{(T)}, C^{(T)}\right) \quad (6)$$

for each geographical area $\sigma^{(R)}$ based on the current state of its provisioned O-SLAs and the estimated number of clients for the next provisioning time frame, where: (1) $G^{(R)}_{type}$ is the MMOG type; (2) $t^{(R)}_{cli}$ is the estimated number of active accounts; (3) $t^{(R)}_{ini}$ is the required instantaneous total non interruption ratio, initially set to the game provider's service policy value, s_{ini} (defined in Section 3.2); (4) $t^{(R)}_{tni}$ is the total non-interruption ratio, initially set to s_{tni}, the provider's service policy value; (5) $t^{(R)}_{time}$ is the estimated time period for these requirements expressed in hours; (6) $P^{(T)}$ is the target hourly price per client defined by the game provider's service policy; and (7) $C^{(T)}$ is the target compensation per client per minute defined by the game provider's service policy.

4.2 Phase 2: Ranking Offers

In the second phase, the game provider gathers the O-SLA templates from all the game operators and instantiates them with the "best" values permitted by the O-SLA 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}:

$$P_x = \frac{t^{(R)}_x - t^{(O_{\min})}_x}{u_x} \cdot p^{(u)}_x \cdot f_x\left(t^{(O)}_x\right), \forall x \in \{cli, ini, tni\}, \quad (7)$$

where $t^{(R)}_x$ is the operational requirement for the term $x \in \{cli, ini, tni\}$ (see Equation 6), $t^{(\min)}_x$ is the minimum value of the term x allowed by the operator through an O-SLA, $p^{(u)}_x$ represents the price per term unit u_x, and $f_x\left(t^{(O)}_x\right)$ is a shape function defined as in Equation 4. The final price the game provider is charged when accepting the O-SLA is:

$$P^{(O)} = P_{base} + \left(P_{cli} + P_{ini} \cdot t^{(R)}_{cli} + P_{tni} \cdot t^{(R)}_{cli}\right) \cdot T_{coeff}, \quad (8)$$

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:

$$T_{coeff} = \left\lceil \frac{t^{(R)}_{time}}{t^{(O_{\min})}_{time}} \right\rceil \cdot f_{time}\left(t^{(R)}_{time}\right), \quad (9)$$

where $\lceil \cdot \rceil$ is the ceiling function, $t^{(O_{\min})}_{time}$ represents the lowest O-SLA validity period allowed by the operator, and f_{time} is a shape function defined as in Equation 4.

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

$$O = \bigcup_{i=1}^{M} O\text{-}SLA_i, \quad (10)$$

Consider 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: pricing rank $\mathcal{P}_{O\text{-}SLA}$ (directly proportional), compensation rank $\mathcal{C}_{O\text{-}SLA}$ (inversely proportional) and resource fitness rank $\mathcal{F}_{O\text{-}SLA}$ (inversely proportional):

$$\mathcal{R} = \lambda_p \cdot \mathcal{P}_{O\text{-}SLA} - \lambda_c \cdot \mathcal{C}_{O\text{-}SLA} - \lambda_f \cdot \mathcal{F}_{O\text{-}SLA}, \quad (11)$$

where $\lambda_p, \lambda_c, \lambda_f \in [0;1]$ and $\lambda_p + \lambda_c + \lambda_f = 1$. We define in the following the computation of the pricing, compensation and resource fitness ranks by the game provider.

The *pricing rank* $\mathcal{P}_{O\text{-}SLA}$ of an operational offer is a quantification how expensive a resource is, determined as the ratio between the aggregated hourly price $\frac{P_i^{(O)}}{t_{time_i}^{(O)}}$ of all M O-SLAs of an operational offer and the target price $P^{(T)} \cdot t_{cli_i}^{(O)}$ for servicing all clients in all M O-SLAs (see Equation 2):

$$\mathcal{P}_{O\text{-}SLA} = \frac{\sum_{i=1}^{M} \frac{P_i^{(O)}}{t_{time_i}^{(O)}}}{P^{(T)} \cdot \sum_{i=1}^{M} t_{cli_i}^{(O)}}. \tag{12}$$

The *compensation rank* quantifies the penalties the operator pays for O-SLA faults computed based on a *compensation gain* representing the area of the compensation function C_x within its definition interval $\left[0; b_x^{(\max)}\right]$ (see Equation 3):

$$\mathcal{A}_x = \int_0^{b_x^{(\max)}} C_x\left(b_x\right) \cdot db_x = \frac{c_x^{(u)}}{u_x} \cdot \int_0^{b_x^{(\max)}} b_x \cdot f_x\left(\frac{b_x}{b_x^{(\max)}}\right) \cdot db_x\,. \tag{13}$$

By substituting $y = \frac{b_x}{b_x^{(\max)}}$ in Equation 13, we obtain:

$$\mathcal{A}_x = \frac{c_x^{(u)} \cdot \left(b_x^{(\max)}\right)^2}{u_x} \cdot \int_0^1 y \cdot f_x(y) \cdot dy\,. \tag{14}$$

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

$$\delta_x : \left[0; b_x^{(\max)}\right] \rightarrow [0; \Delta_{\max}]\,, \tag{15}$$

where $\Delta_{\max}$ represents 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 SLA fault. By superimposing δ_x to the compensation gain, we compute an adjusted metric called *characteristic compensation gain* which defines the compensation function for a specific MMOG:

$$\mathcal{A}_x^{(ch)} = \frac{c_x^{(u)} \cdot \left(b_x^{(\max)}\right)^2}{u_x} \cdot \int_0^1 y \cdot \delta_x\left(b_x^{(\max)} \cdot y\right) \cdot f_x(y) \cdot dy\,. \tag{16}$$

We compute the characteristic compensation gain through a finite sum approximation:

$$\mathcal{A}_{C_x}^{(ch)} \approx \frac{c_x^{(u)} \cdot \left(b_x^{(\max)}\right)^2}{u_x} \cdot \sum_{i=1}^{N} \frac{i}{N} \cdot \delta_x\left(\frac{b_x^{(\max)} \cdot i}{N}\right) \cdot f_x\left(\frac{i}{N}\right), \tag{17}$$

where N is the *integration granularity* representing the number of interval partitions. Using $\mathcal{A}_{C_x}^{(ch)}$, 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 $x \in \{cli, ini, tni\}$:

$$\mathcal{C}_{O\text{-}SLA} = \sum_{i=1}^{M} \sum_{x \in \{cli, ini, tni\}} \psi_x \cdot \frac{\mathcal{A}_{x_i}^{(ch)}}{\mathcal{A}_x^{(REF)}}, \tag{18}$$

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

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

$$\mathcal{F}_{O\text{-}SLA} = \sum_{x \in \{cli, ini, tni, time\}} \phi_x \cdot \frac{S_x\left(t_{x_i}^{(O)}\right)}{t_x^{(R)}}, \text{ where} \tag{19}$$

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

$\phi_{cli}, \phi_{ini}, \phi_{tni}, \phi_{time} \in [0;1]$ indicate again the provider's preference for each O-SLA term ($\phi_{cli} + \phi_{ini} + \phi_{tni} + \phi_{time} = 1$) and S_x is an aggregation function (i.e. sum for client number, 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.

As a final step, the operational offers are sorted in ascending order by their rank. It is worth mentioning 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 Equation 16). This ensures that game providers do not constantly reach the same apparently-optimal operator, but are able to discover those whose offers most accurately match their needs.

4.3 Phase 3: Binding Agreement

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 RESULTS

We present in this section an evaluation of our MMOG middleware stack focused 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

Table 1: Summary of commercial cloud R-SLAs.

Cloud provider	*VM types*	*Locations*	*Price [$/.. RSU/h]*	*GB/h]*	*Valid. [h]*	*VM inst. [seconds]*
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]

real data corresponding to MMOG workloads (number of players online) and real commercial IaaS cloud SLAs. We cover an evaluation space with two dimensions: the compensation rank function $\mathcal{C}_{O\text{-}SLA}$ (Sections 5.2 and 5.3), and the fitness rank function $\mathcal{F}_{O\text{-}SLA}$ (Section 5.4). The pricing rank is employed in the O-SLA negotiation, but is not a focal point of our evaluation as we covered it in [12]. Our aim is to demonstrate that considering compensations is paramount when ranking SLAs for reducing the operational cost and maintaining QoS, and to determine guidelines for balancing the operational terms of $\mathcal{F}_{O\text{-}SLA}$ (see Equation 19) for maximising the provider's profit.

5.1 Experimental Setup

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 have collected execution traces for a period of six months from 150 servers on four continents by sampling the number of players every two minutes. The number of players ranges from 0 to 2000, the maximum capacity of a RuneScape server [11]. We simulate game providers that provision O-SLAs according to the number of active client accounts. For client–game provider interaction, we use the real player subscription model of RuneScape ($5.95 per month, August 2012).

We employ 115 R-SLAs based on the resources provided by 16 real commercial cloud providers, described in Table 1. The hourly prices are presented based on the processing power and memory availability. The prices include the upstream and downstream network traffic which may have an important impact on the final R-SLA prices, as in the case of CloudCentral. For each cloud provider we use its geographical location, memory size, and price. We express the VM processing power using an MMOG-centric metric called *RS unit*, representing the equivalent 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 [8]. The VM instantiation overhead (column "VM inst.") is the variable duration of instantiating a new VM instance. We consider a 100% resource uptime because most cloud providers promise very high resource availability.

We characterise the client – game provider contracts through the service policies displayed in Table 2. We further sample the design space of operational ranking functions through

Table 2: Service policies of game providers, where sets of policies are defined (min; max; step) **triplets and stochastic values by** $[min; max]$ **interval ranges.**

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

Table 4: RuneScape-related O-SLA templates.

Name	$t^{(O)}_{cli}$ (x10^3)	$t^{(O)}_{ini}$	$t^{(O)}_{time}$	$t^{(O)}_{tni}$	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

the 45 functions summarised in Table 3, by varying in RK1–RK6 the class of the compensation ranking function (results in Section 5.2), in RK7–RK17 the computational complexity of the compensation ranking function (Section 5.3), and in RK18–RK37 the fitness ranking function (Section 5.4).

Finally, we employ an extensive set of O-SLAs designed to cover all aspects of the negotiation described in Section 4, based on the three O-SLA templates presented in Table 4 and generated by varying one or more of their term values. We keep the pricing functions constant, since we covered them in [12]. We present the compensation functions C_x introduced in Equation 3 as a set of two parameters: the shape function type f_x and its shape coefficient a. We use two classes of parameterised shape functions: logarithmic defined as in Equation 5 and exponential defined as:

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

We adjust the shape coefficient for different O-SLAs and evaluate the resulting compensation functions using the compensation gain metric defined in Equation 14. We use a uniform distribution of the serviced geographical areas.

We consider metrics for both cost and performance (SLA). We analyse the financial aspect of our MMOG operation using two metrics: (1) *gross profit* representing the difference between the business actor's revenue and the cost of providing its services, excluding taxation and other overheads; (2) *total compensation* (a fraction of the gross profit) representing the total cost a business actor pays as a compensation for any SLA fault for the entire simulation period. We analyse the QoS through two O-SLA metrics defined in Section 2.3: the instantaneous non-interruption ratio t_{ini} and the total non interruption ratio t_{tni}. For a better understanding of t_{tni}, we also analyse the *average non-serviced clients* representing the average number of clients who were denied service within a measurement time step because of improper O-SLA provisioning by the game provider, or because of improper resource allocation by the game operator.

5.2 Compensation Ranking Selection

The goal of this first experiment is to study how game providers can select operational offers (sets of O-SLA instances, see Section 4) from game operators based on compensation terms. We study six different compensation rank-

Table 3: Operational ranking configurations, where sets of functions are defined as (min; max; step) value ranges.

Ranking acronym *(function)*	*Compensation rank* ($\mathcal{C}_{O\text{-}SLA}$)		*Fitness rank* ($\mathcal{F}_{O\text{-}SLA}$)			
	Type	*Integration granularity* (N)	ϕ_{cli}	ϕ_{ini}	ϕ_{tni}	ϕ_{time}
`RK1(max)`	max	-	0.2	0.5	0.2	0.1
`RK2(avg-3)`	avg	3	0.2	0.5	0.2	0.1
`RK3(avg-9)`	avg	9	0.2	0.5	0.2	0.1
`RK4(gain)`	$\mathcal{A}_{C_x}$	-	0.2	0.5	0.2	0.1
`RK5(cgain-9)`	$\mathcal{A}^{(ch)}_{C_x}$	9	0.2	0.5	0.2	0.1
`RK6(cgain-30)`	$\mathcal{A}^{(ch)}_{C_x}$	30	0.2	0.5	0.2	0.1
`RK[7;17] (cgain-[1;30])`	$\mathcal{A}^{(ch)}_{C_x}$	1,(**3**; **30**; **3**)	0.2	0.5	0.2	0.1
`RK[18;22](cli-[10;90])`	$\mathcal{A}^{(ch)}_{C_x}$	30	(**0.1**; **0.9**; **0.2**)	$\frac{1-\phi_{cli}}{3}$		
`RK[23;27](ini-[10;90])`	$\mathcal{A}^{(ch)}_{C_x}$	30	$\frac{1-\phi_{ini}}{3}$	(**0.1**; **0.9**; **0.2**)	$\frac{1-\phi_{ini}}{3}$	
`RK[28;32](tni-[10;90])`	$\mathcal{A}^{(ch)}_{C_x}$	30	$\frac{1-\phi_{tni}}{3}$		(**0.1**; **0.9**; **0.2**)	$\frac{1-\phi_{tni}}{3}$
`RK[33;37](time-[10;90])`	$\mathcal{A}^{(ch)}_{C_x}$	30	$\frac{1-\phi_{time}}{3}$			(**0.1**; **0.9**; **0.2**)

ing methods. First, `max` (`RK1`) (see Table 3 and Figure 3) ranks offers by the compensation value $C_x(k)$ corresponding to the most frequent O-SLA fault (i.e. most frequent fault k for which $\delta_x(k) = \Delta_{\max}$, defined in Equation 15):

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

where $\delta_x(k) = \Delta_{\max}$, M is the total number of O-SLAs in an operational offer, $C^{(T)}$ is the target minutely compensation per client (see Section 6), and $\psi_x \in [0;1]$ with $\sum_{x\in\{cli,ini,tni\}} \psi_x = 1$. Second and third, `avg-3` (`RK2`) and `avg-9` (`RK3`) rank based on the average compensation values:

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

for $N = 3$ and $N = 9$ with uniformly distributed SLA faults (see Figure 3). Fourth, `gain` (`RK4`) is based on Equation 18 with the compensation gain defined as in Equation 14 (not 16). Fifth and sixth, `cgain-9` (`RK5`) and `cgain-30` (`RK6`) are variants of the compensation rank proposed in Section 4 with $N = 9$ and $N = 30$, where N is the integration granularity (i.e. the number of partitions in the Riemann sum approximation) of Equation 16. We defined a separate game provider for each of the six compensation ranking methods and the same `PP1` service policy (see Table 2). We further use 65 game operators, each offering a different O-SLA based on the `OSLA-1` template (see Table 4) and differentiated by their compensation function, its shape and other parameters, as defined in Section 5.1. We run simulations and evaluate the total compensation, representing the fraction of compensation obtained by game providers from their gross profit.

The top graph of Figure 4 depicts the total compensation of all game providers relative to the total compensation when using the basic `max` method. We observe that while the `max`, `avg` and `gain` perform roughly the same (variation less than 3%), `cgain` leads to 11–16% increases in income from compensations. As each MMOG exhibits an individual load pattern, which, in turn, results in a particular fault distribution (as exemplified in Figure 3), by employing the characteristic gain ranking method game providers automatically tune their offer selection process to favour the O-SLAs bringing the highest compensations from these unique fault distributions. We further analyse the impact employing these methods has on the QoS offered by game providers and depict the results in the bottom graph of Figure 4. The results indicate only slight QoS variations for all methods: the instantaneous non-interruption ratio t_{ini} is above the target value $s_{ini} = 0.9$ of the game providers' service policy `PP1` between the first and third quartiles, and the median value is near 1 (optimal). The average number of non-serviced clients is around 8 (from a maximum of 2000).

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) that it is possible to significantly increase the providers' income (up to 16%) through these offer selection methods without negative effects on the QoS offered to the clients.

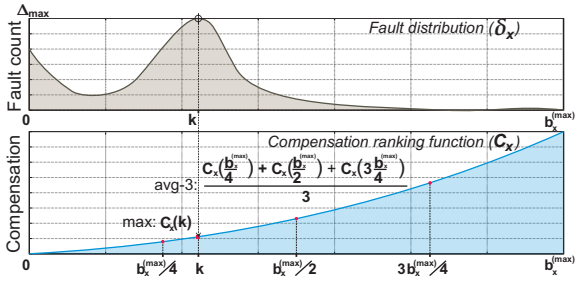

Figure 3: `max` versus `avg-3` compensation ranking.

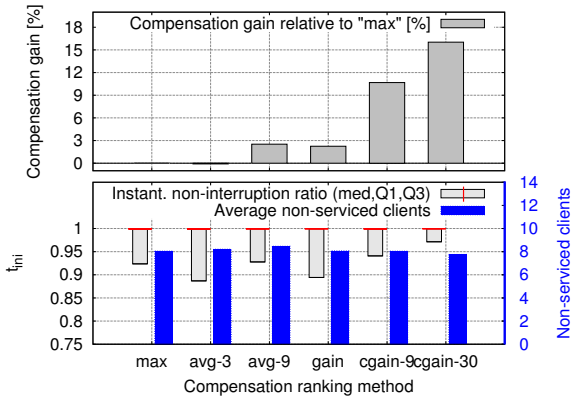

Figure 4: Compensation ranking comparison.

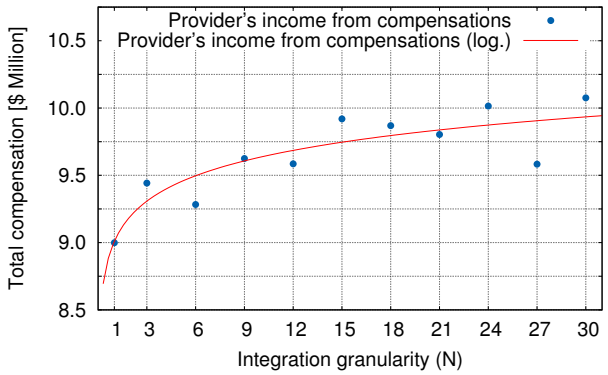

Figure 5: Increase in the game provider's income resulting from better offer compensation clause ranking, with increasing integration granularity(N).

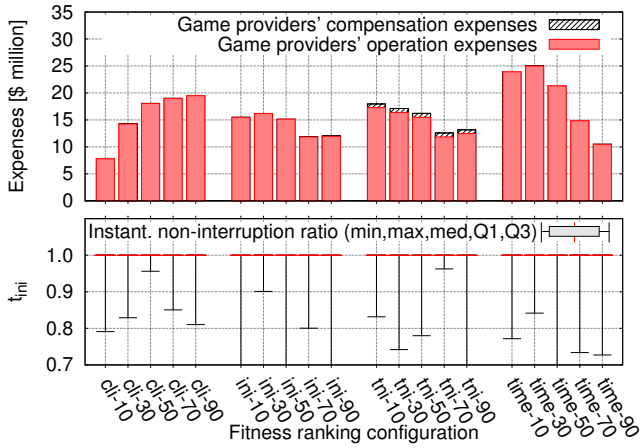

Figure 6: Game provider expenses for different fitness rankings.

5.3 Tuning the Compensation Ranking

As depicted in Figure 4, the characteristic compensation granularity ranking `cgain` involving the integration granularity N as a tuning parameter (see Equation 17), leads to an increase of the game provider's income from total compensation with increasing values of N. The goal of this experiment is to quantify the impact of this integration granularity N 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 used in the previous experiment, except for the game provider's offer ranking configurations. From Table 3, we use in this experiment the `RK7` configuration ($N = 1$) and ten other configurations (i.e., `RK8` to `RK17`), whose integration granularities range from 3 to 30 with a step of 3. We run a separate simulation for each ranking configuration and compute the fraction of the game provider's profit representing compensations for O-SLA faults.

We observe in Figure 5 that the compensation increases logarithmically with the integration granularity. The identified trend is not strictly monotonous because, as the integration granularity is increased, game providers select other game operators which use different cloud resources, which, in turn, influence the number and intensity of faults, leading to variations in the total compensation. Over the six-month period we simulated, 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 further be tuned to obtain a logarithmic increase in total compensation income through the increase of the integration granularity parameter.

5.4 Weighting the Fitness Ranking

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 Equation 19). In our experimental setup, the game operators offer different O-SLAs, generated starting from the `OSLA-2` template (see Table 4) and varying in turn the ranges of the four negotiable terms $t_x^{(O)}, x \in \{cli, ini, tni, time\}$. We use five game providers and vary over 20 simulations the ranking configuration for all involved game providers from `RK18` to `RK37` (described in Table 3). Each of the 20 configurations is designed to gradually increase the weight ϕ_x of one of the four negotiable O-SLA terms. In each simulation, the game providers employ different service policies. For the `cli-[10;90]` and `ini-[10;90]` ranking configurations, each game provider uses one of the service policies `PP2-PP6` defined in Table 2 (note that for $t_{cli}^{(O)}$ there is no corresponding service policy term). For the `tni-[10;90]` ranking configurations, each game provider uses one of the service policies `PP7-PP11`. Finally for the `time-[10;90]` ranking configurations, each game provider uses one of the service policies `PP12-PP16`. This experimental setup effectively explores the much larger space of `RK-PP` policy pairs without exhaustively considering each possible pair. The fitness ranking 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 6 shows that the fitness ranking configuration has a significant impact on the expenses of the game providers. The maximum difference we observed in this experiment is a $14.5 million reduction of expenses for the case of the t_{time} validity period (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 $t_{ini}^{(O)}$ and $t_{tni}^{(O)}$ QoS terms leads to a decrease in the game providers' expenses. Conversely, increasing of the weight of the client number term $t_{cli}^{(O)}$ 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 Figure 6 (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 ($s_{tni} \geq 0.996$) to compensate the clients.

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

6. RELATED WORK

We survey three large bodies of related work: SLA-based operation of MMOGs, SLA stacks for large-scale systems, and cloud operation of services with millions of customers.

6.1 SLA-based operation of MMOGs

Wong [18] proposed a resource provisioning algorithm focused on network QoS guarantees, whereas we considered more resource types, including virtualised cloud-based [12].) Briceno et al. [3] studied resource allocation for MMOGs but, unlike our work, considered a simplified workload model (not traces from a real MMOG). Complementary to our study, Lee and Chen [9] investigate MMOG server consolidation techniques focusing on the energy consumption. Middleton et al. [10] proposes a four-actor business model for online games, but does not consider its mapping to QoS models and cloud hosting. Complex business models are also proposed by Alves et. al [1] with focus on higher-level business interactions and goals for MMOG operation only. In contrast, we study a novel operation model and its effects on the profits of both game and resource operators. Complementing our work, Nojima [13] studies the relationship between pricing models and MMOG player motivation, while Oh and Ryu [14] analyse different pricing models for gaming service.

6.2 SLA stacks for large-scale systems

There has been a lot of work researching SLA stacks since at least the early 1980s [15], with recent focus on grids [17] and clouds [19]. Despite the large amount of work, existing SLA stacks cannot be directly employed for MMOG operation because the mapping of business SLA terms to resource-centric real-time operational terms required by MMOGs was missing. Unlike traditional SLAs considered in grids (e.g. SNAP [5], NextGrid [6], SLA@SOI [4]) and other distributed systems (e.g. Galaxy [16], Oceano [2]), our proposed SLAs include a comprehensive specification of compensations for temporary QoS violations introduced by the stringent QoS requirements and the dynamic nature of MMOGs. Nevertheless, our negotiation protocol is inspired from previous work on SLA-based scheduling [17] by considering a three-stage negotiation mechanism: SLA generation, ranking operational offers (also considering penalties for SLA violation), and renegotiation. In addition, our work proposes a new ranking mechanism adapted to MMOGs and non-uniform SLA violations, and focuses on various compensation and other SLA policies.

6.3 Cloud operation of services with millions of customers

The entertainment industry has already started to migrate from the in-house to a cloud-based infrastructure. Zynga for example uses Amazon EC2 resources for operating online gaming services for over 250 million users in 2011. However, the games supported by Zynga require much less computational and network resources than MMOGs. On-demand gaming is provided by companies such as Geelix [7], OnLive, Gaikai, and OTOY by offloading their computation to the cloud and streaming back the video output to remote clients. This model is not (yet) considered by MMOG operators, in part because of the high network requirements imposed on the players. Since late 2011, Amazon Web Services has been used for video streaming by Netflix, and for offloading web browsing on mobile devices with Android OS. Our work adapts this model to the specifics of MMOGs and proposes an in-depth study of a variety of scenarios with application to other branches of the entertainment industry.

7. DISCUSSION

Although designed for MMOGs, the autonomic ecosystem and underlying SLA mechanisms proposed in this paper can generalised and applied to other scientific and industrial application domains in response to their ever larger demand in computing and storage resources. Similar to MMOG companies, research institutes have to take a triple role to conduct scientific computing research: *providers* by developing new domain-specific research applications (or purchasing required software licences), *operators* by running the applications on own resources so that user-centric QoS requirements are fulfilled, and *data centres* by purchasing expensive high-performance hardware including its hosting, maintenance, and periodic renewal. The new autonomic SLA-driven ecosystem proposed in this paper can represent useful input for research institutes or industrial companies to focus their activities based on their needs, interest, and size. Domain scientists as application providers can outsource their codes to computer scientists specialised in middleware tools capable of running them with guaranteed QoS. Similarly, computer scientists can delegate the application hosting to on-demand to specialised cloud providers owning modern hardware infrastructure facilities. The interaction of the three actors is regulated through SLA negotiation protocols such as the one proposed in this paper that establish the price, terms of operation, and compensation for service violations. The MMOG ecosystem proposed in this paper can be adapted to new scenarios by specifying new provider-specific service policies, operational terms, and compensation functions for O-SLA faults, and mapping them onto the generic O-SLA negotiation protocol, including a mechanism for building and ranking operational offers.

Finally, although designed for MMOGs, our autonomic ecosystem and its underlying SLA negotiation and compensation mechanism is valid for game genres such as *First Person Shooter (FPS)* action games. Due to their highly dynamic nature and more stringent QoS requirements, FPS games scale to a small number of users and may have more severe compensation consequences than MMOGs. We have studied in [11] the QoS-based resource and scalability requirements of a real-world FPS game, and plan to study its impact on the operational and compensation terms in future work. Moreover, we believe that our proposed ecosystem presents great potential of being employed by the next generation massively multiplayer online FPS games (MMOFPS) that are being released right now, for example Planteside 2 with more titles being currently in development.

8. CONCLUSION

The current MMOG ecosystem with tens of millions of players across hundreds of games forces game providers to also become game and infrastructure operators, leading to inefficient resource utilisation, high service prices, and limits its market participation to only the largest companies. We proposed a new ecosystem based on cloud computing principles for hosting and operating MMOGs, and focused on the formulation and negotiation of SLAs encompassing price, operational terms, and compensation policies. For ranking MMOG operational offers, our model considers and 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 in a variety of scenarios using operational traces collected from a real MMOG and SLAs from over ten commercial cloud providers. We demonstrated that a ranking method which considers the yield from compensations in the given environment is necessary. We approached this through a new metric called characteristic compensation gain that leads to $11-16\%$ higher financial gain without QoS deterioration, and is logarithmic improved by its approximation precision. Furthermore, tuning the O-SLA terms in the fitness rank to reflect the MMOG load can lead to a $20-60\%$ reduction in the operational expenses.

In the future we intend to generalise this work and apply it to other domains such as FPS games or other computationally-intensive scientific and industrial applications.

9. ACKNOWLEDGMENTS

Austrian Science Fund project TRP 72-N23 funded this research.

10. REFERENCES

[1] T. Alves and L. Roque. Using value nets to map emerging business models in massively multiplayer online games. In *9th Pacific Asia Conference on Information Systems*. AIS Electronic Library, 2005.

[2] K. Appleby, S. A. Fakhouri, L. Fong, G. S. Goldszmidt, M. H. Kalantar, S. M. Krishnakumar, D. P. Pazel, J. A. Pershing, and B. Rochwerger. Océano – SLA based management of a computing utility. In *International Symposium on Integrated Network Management Proceedings*, pages 855–868. IEEE, 2001.

[3] L. D. Briceño, H. J. Siegel, A. A. Maciejewski, Y. Hong, B. Lock, M. N. Teli, F. Wedyan, C. Panaccione, C. Klumph, K. Willman, and C. Zhang. Robust resource allocation in a massive multiplayer online gaming environment. In *4th International Conference on Foundations of Digital Games*, pages 232–239. ACM, 2009.

[4] P. Chronz and P. Wieder. Integrating ws-agreement with a framework for service-oriented infrastructures. In *11th International Conference on Grid Computing*, pages 225–232. IEEE Computer Society, 2010.

[5] K. Czajkowski, I. T. Foster, C. Kesselman, V. Sander, and S. Tuecke. SNAP: A protocol for negotiating service level agreements and coordinating resource management in distributed systems. In *Job Scheduling Strategies for Parallel Processing*, volume 2537 of *LNCS*, pages 153–183. Springer, 2002.

[6] P. Hasselmeyer, H. Mersch, B. Koller, H.-N. Quyen, L. Schubert, and P. Wieder. Implementing an SLA negotiation framework. In *eChallenges*, pages 154–161. IOS Press, October 2007.

[7] O.-I. Holthe, O. Mogstad, and L. A. Rønningen. Geelix livegames: Remote playing of video games. In *6th IEEE Consumer Communications and Networking Conference*, pages 1–2. IEEE, 2009.

[8] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(6):931–945, June 2011.

[9] Y.-T. Lee and K.-T. Chen. Is server consolidation beneficial to MMORPG? A case study of World of Warcraft. In *International Conference on Cloud Computing*, pages 435–442. IEEE Computer Society, 2010.

[10] S. E. Middleton, M. Surridge, B. I. Nasser, and X. Yang. Bipartite electronic SLA as a business framework to support cross-organization load management of real-time online applications. In *Euro-Par 2009 âĂŞ- Parallel Processing Workshops*, number 6043 in LNCS, pages 245–254. Springer, 2010.

[11] V. Nae, A. Iosup, and R. Prodan. Dynamic resource provisioning in massively multiplayer online games. *IEEE Transactions on Parallel and Distributed Systems*, 99(82):380–395, 2010.

[12] V. Nae, R. Prodan, A. Iosup, and T. Fahringer. A new business model for massively multiplayer online games. In *2nd International Conference on Performance Engineering*, pages 271–282. ACM, 2011.

[13] M. Nojima. Pricing models and motivations for MMO play. In *Conference of the Digital Games Research Association*, pages 672–681, Tokyo, September 2007.

[14] G. Oh and T. Ryu. Game design on item-selling based payment model in Korean online games. In *Conference of the Digital Games Research Association*, pages 650–657, Tokyo, September 2007.

[15] R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12):1104, December 1980.

[16] W. Vogels and D. Dumitriu. An overview of the galaxy management framework for scalable enterprise cluster computing. In *International Conference on Cluster Computing*, pages 109–118. IEEE Computer Society, 2000.

[17] P. Wieder, J. Seidel, O. Wäldrich, W. Ziegler, and R. Yahyapour. Using SLA for resource management and scheduling – a survey. In *Grid Middleware and Services*, pages 335–347. Springer, 2008.

[18] K. Wong. Resource allocation for massively multiplayer online games using fuzzy linear assignment technique. In *5th Consumer Communications and Networking Conference*, pages 1035–1039. IEEE, 2008.

[19] L. Wu and R. Buyya. Service level agreement (SLA) in utility computing systems. In *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*, Advances in Web Technologies and Engineering, chapter 1, pages 1–25. IGI Global, July 2011.