The Impact of Virtualization on the Performance and Operational Costs of Massively Multiplayer Online Games

Alexandru Iosup*

Parallel and Distributed Systems Group,
Delft University of Technology, the Netherlands
E-mail: A.Iosup@tudelft.nl
*Corresponding author

Vlad Nae*

Institute for Computer Science,
University of Innsbruck, Austria
E-mail: Vlad@dps.uibk.ac.at
*Corresponding author

Radu Prodan

Institute for Computer Science,
University of Innsbruck, Austria
E-mail: Radu@dps.uibk.ac.at

Abstract:
Today's highly successful Massively Multiplayer Online Games (MMOGs) have millions of registered users and hundreds of thousands of active concurrent players. Motivated by performance and portability, MMOG operators currently pre-provision and then maintain throughout the lifetime of the game tens of thousands of compute resources in data centers located across the world. In contrast, virtualization is a new technology that promises minimal overhead and a uniform, easy to port to, computing platform. To investigate the potential of this new technology, in this paper we propose a new hybrid resource provisioning model that uses a smaller and less expensive set of self-owned data centers, complemented by virtualized cloud computing resources during peak hours. Through trace-based simulation and empirical experimentation, we assess the impact of provisioning virtualized cloud resources, analyze the components of virtualization overhead, and compare provisioning of virtualized resources with direct provisioning of data center resources. Using a simple cost model, we also investigate the costs of hosting MMOGs on the resources leased independently from three commercial cloud providers, including the current market leader Amazon. We find that the virtualization characteristics, policy, and provider can have an important impact on the performance and cost of MMOG operation. We also find that resource under-allocation, which occurs more often for static resource provisioning and non-virtualized platforms, has a significant impact on a key performance metric, latency.

Keywords: virtualization impact; Massively Multiplayer Online Games; MMOG; game operation; resource provisioning; on-demand; data centers; clouds; cloud computing; Amazon EC2; performance; cost; virtualization overhead; simulation; empirical results.

Reference to this paper should be made as follows: Iosup, A., Nae, V., and Prodan, R., (2010) 'The Impact of Virtualization on the Performance and Operational Costs of Massively Multiplayer Online Games', *Int. J. Adv. Media and Comput.*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Alexandru Iosup received his Ph.D. in Computer Science in 2009 from the Delft University of Technology (TU Delft), the Netherlands. He is currently an Assistant Professor with the Parallel and Distributed Systems Group at TU Delft. He is the co-founder of the Grid Workloads, the Peer-to-Peer Trace, and the Failure Trace Archives, which provide open access to workload and resource operation traces from large-scale distributed computing environments. He is the author of over 50 scientific publications and received several awards and distinctions, including best paper awards at IEEE CCGrid 2010, Euro-Par 2009, and IEEE P2P 2006.

Vlad Nae received his Diploma Engineer degree in Computer Science from the "Politehnica" University of Bucharest, Romania, in 2006. He is currently a Research Assistant and PhD candidate at the Institute of Computer Science, University of Innsbruck. His research interests are in the areas of resource management in distributed systems, ML-based load prediction, and resource provisioning for massively multi-user environments. His latest work focuses on virtualized cloud resources for servicing massively multiplayer online games. He is the author of 15 publications including three book chapters and three journal articles.

Radu Prodan is an Assistant Professor at the Institute of Computer Science, University of Innsbruck (UIBK). He received his Ph.D. in 2004 from the Vienna University of Technology and his Habilitation in 2009 from UIBK. His research interests are on programming methods, compiler technology, performance analysis, and scheduling for parallel and distributed systems. Prodan participated in several national and European projects, including the IST-034601 edutain@grid project. He is the author of over 70 journal and conference publications, one book, and received one IEEE best paper award.

1 Introduction

Massively Multiplayer Online Games (MMOGs) have emerged in the past decade as a new type of large-scale distributed application characterized by a real-time virtual world entertaining millions of players spread across the globe [4]. To comply with the variable computational and latency-aware resource demands of the players distributed worldwide [10, 14, 23], the MMOG operators maintain a multi-server distributed infrastructure with sufficient computational and network capabilities necessary to guarantee the quality of service requirements and a smooth game session play at all times. This statically provisioned infrastructure has several major drawbacks: is subject to over-provisioning, increases the operational costs of MMOGs, and is vulnerable to sudden increases in demand. In contrast to static provisioning, the new cloud computing technology based on resource virtualization promises to provide an on-demand infrastructure for MMOGs, where resources are provisioned and paid for only when they are actually needed. While virtualization alleviates the problem of porting and deploying MMOGs on on-demand resources by providing homogeneous resources for on-demand MMOG hosting, the virtualization overheads may cancel out the benefits. To understand the trade-offs of virtualization, in this work we investigate the impact of virtualization on the performance and cost of hosting of MMOGs.

The current industry approach to ensure that resources are available when needed is based on resource ownership and over-provisioning of resources. For example, the operating infrastructures of the leading MMOGs such as World of Warcraft [6] and RuneScape [29] comprise each thousands of computers in hundreds of physical locations, and resource ownership can take up to 40% of the game revenue [16]. However, the resource demand of MMOGs is highly dynamic [34], which raises two problems: on the one hand, for most of the operational time a large portion of the pre-provisioned resources are unnecessary and used during peak hours only; on the other hand, when unforeseen increases in demand occur, there are not enough resources to serve them. For example, in RuneScape the servers dedicated to Pest Control, a popular minigame, are most of the time under-utilized, but during the evening they may become overloaded and users would have to wait before entering the minigame world.

Cloud computing has emerged recently as a new resource provisioning model, under which resources are leased only when and for how long they are needed, for fixed cost and ensured by Service Level Agreements. Thus, cloud computing may eliminate the need for permanent over-provisioning of occasionally needed resources. Thus, clouds may free companies such as MMOG operators from the large provisioning costs of buying and maintaining hardware, and from the rapid deprecation of hardware investments. Commercial Infrastructure as a Service (IaaS) clouds, such as Amazon Web Services, FlexiScale, and NewServers, employ hardware *virtualization*, which present the user with the illusion of a homogeneous platform, regardless of the physical underlying resources. While this illusion allows MMOG operators to alleviate the problems raised by developing and deploying games on remote resources, achieving this illusion incurs performance overheads that can be significant for particular workload characteristics [3, 32, 11, 42] and for large infrastructures [37, 38, 26].

We have proposed in previous work [34, 33] a new MMOG ecosystem consisting of game operators and multiple data centers, and studied several resource lease policies. We have shown through simulations and under the assumption of zero performance overheads that dynamic resource provisioning can reduce the MMOG operational costs considerably, with a reasonable performance loss. More recently, we started to investigate the impact of virtualization overheads on the performance of MMOGs [35] by extending our MMOG ecosystem to accommodate both non-virtualized and virtualized resources, and evaluated through trace-based simulation scenarios involving one or more MMOG providers. In this paper, we extend this preliminary investigation in three main directions: more in-depth analysis of the virtualization impact on the MMOG performance, new results that quantify the cost of hosting MMOGs on virtualized resources provided by commercial clouds, and new experiments involving a real prototype game.

The remainder of this paper is structured as follows. We present in Section 2 the MMOG ecosystem and a background on resource virtualization and compute clouds. We continue in Section 3 with a state-of-the-art in resource provisioning, including a comparison to our previous work. We model MMOGs operating on virtualized and non-virtualized resources, either separately or in conjunction, in Section 4. In Section 5, we investigate through trace-based simulation the impact of virtualization on the performance and operational costs of MMOGs. We continue, in Section 6, with an empirical evaluation of the impact of virtualization on several performance metrics associated with the quality of service experienced by the game player. Last, we conclude our work in Section 7.

2 Background

In this section we summarize our previous work on the MMOG ecosystem and the virtualization concept.

2.1 The MMOG Ecosystem

We have previously modeled [34, 33] an MMOG ecosystem with two components: the MMOG operators and the MMOG platform providers (*hosters*). Each MMOG is managed by one *game operator* responsible for the real-time experience of the connected players, and for negotiating resource leases with data center and cloud providers in order to achieve this goal. The hosters act as a global network of conventional *data centers* complemented by a set of virtualized *cloud computing providers* that host in cooperation multiple MMOG sessions, simultaneously.

The relative success of an MMOG is characterized by the number of registered players. We have shown in previous work [33] that the number of MMOG players within the MMORPG category is growing under an exponential trend, with about 25 million players in 2008 and, potentially, over 60 million players by 2011. The large number of MMOG players, for the whole ecosystem and for each game in part, is an important motivation for our work.

Hosters The hosters operate two major services. A *load prediction* service, presented in detail in [36], is in charge of projecting the future distribution of entities in the game world that is demonstrated to have the highest impact on the server load. We devised accurate analytical models for translating the entity distribution prediction and possible interactions into estimating the game server load. Based on the projected load, a *resource allocation* service [34] provisions additional local servers to the game session (through the zoning, replication, or instancing parallelization techniques) that accommodate the player load while guaranteeing the real-time quality of service constraints. For example, by timely foreseeing critical hot-spots (i.e. excessively populated area of interest generating a large number of interactions), one can dynamically provision additional servers on some new resources and take timely load balancing actions that transparently redistribute the game load before the servers become overloaded.

Our hosting model considers the *size* and *duration* of the minimal resource allocation which may be not only for a resource as a whole (e.g. a server in Web data centers [2] or a multi-processor node in a Grid system [27]), but also for a fraction of that resource (e.g. a virtual machine running on a physical node [30], or a channel of an optical network). Similarly, the minimal duration for which a resource may be allocated may be between a few seconds (servicing one user request by a Web service) to several months (a typical value for Web server hosting). We define the *resource bulk* as the minimum number of resources that can be allocated for one request, expressed as the multiple of a minimal resource size. Similarly, we define the *time bulk* as the minimum duration for which a resource allocation can be performed expressed as multiple of a minimal time period.

Operators The game operators handle simultaneously MMOG sessions of different types. They issue resource requests based on the predicted load of the games they operate (either statically or dynamically [36, 33] computed), and the hosters respond with offers based on their local time-space renting policy. Depending on the hoster's service model, either best-effort or advance reservation-based, resource requests are queued or immediately scheduled, respectively. Using one or several important metrics (e.g., virtualization overheads, geographical proximity, data locality, resource proportionality), the game operator applies a *resource selection policy* using one or several of the following four operations: *classifying* groups several resources into classes based on metric value ranges; *sorting* orders the resources based on metric values; *filtering* eliminates resources based on metric values; *prioritizing* gives higher allocation priority to resources with important metric values.

The operators submit resource requests to hosters by specifying the type and amount of resources, and the duration for which the resources are desired. The allocated resources are reserved for the entire duration of the game operator's request, that is, preemption or migration are not supported. As with our previous work [34, 33], we consider four resource types that are relevant for MMOG hosting: *CPU*, *memory*, input from (*ExtNet[in]*) and output to (*ExtNet[out]*) the external network of the data center. Once the available resources are selected, they are *allocated* to the game operators, which run MMOG servers on them. From the game operator's point of view, we say that the resources have been *provisioned*. (We use the terms resource allocation and resource provisioning interchangeably.)

2.2 Virtualization and Clouds

Virtualization is a key technology for hiding from the users the low-level physical characteristics of a computing platform by showing another abstract, higher-level emulated platform instead represented by a so called *Virtual Machine*(VM). The most common virtualization environments today are VMWare [43] and Xen [13], while there exist other solutions used by smaller communities such as Oceano [1], VMPlants [31], Kadeploy [9], and XGE [17]. Using virtualization, a computing cloud may address with the same shared set of physical resources a large user base with different needs, cheaply.

Similar to scientific computing software and other complex software, MMOGs have many library dependencies that make the deployment process complex and requiring manual intervention. In most cases, special requirements have to be fulfilled as discussed in [7] (for scientific applications), which makes the automatic deployment and provisioning of resources impractical without additional platform support such as virtualization. Although the virtualization layer adds overheads (that we will model in Section 4.2) and thus reduces performance, correctly configured VM images can be used to automatically deploy and further operate MMOGs on many hardware platforms. Thus, the use of virtualization promises to ensure deployment and provisioning scalability; we characterize the trade-offs raised by MMOG virtualization in the remainder of this work.

3 Related Work

Our investigation into the impact of virtualization on the performance and operational costs of MMOGs is related with a large body of previous work that spans three directions from resource provisioning research: data centers, grid and cloud computing, and resource virtualization.

Data Centers The case when resources from one data center are shared between multiple applications with statistical performance guarantees has received much attention [41, 30, 2]. In these approaches, the variables characterizing requests (such as "service time") are independent of the system state. In contrast, an important component of the resource demands in the MMOG ecosystem is the interaction between players [34]; thus, these demands are system state-dependent.

Grid and Cloud Computing The problem of dynamically allocating geographically distributed resources to applications has been a popular topic in Grid computing [27] and, more recently, in cloud computing research [39, 42]. Work in these areas investigates mechanisms for resource allocation across single- and multi-cluster grids and clouds, but only for scientific workloads and for unitary resource allocations. Closest to our work, the industrial game hosting platform Butterfly.net Grid (now renamed the Emergent Platform) [20] uses Grid technology to provide on-demand access to cluster resources. Their hosting policy only considers multi-unitary resource bulks and long time bulks; as such, this platform fits well into our MMOG ecosystem as a typical large hoster.

Virtualization Overhead The overhead incurred by virtualization has been the focus of numerous performance studies using general-purpose benchmarks. This overhead can be below 5% for computation [3] and below 15% for networking [3, 32]. Similarly, the performance loss due to virtualization for parallel I/O and web server I/O has been shown to be below 30% [46] and 10% [11], respectively. Much interest for the use of virtualization has been shown by the HPC community, spurred by two seminal studies [45, 25] that find virtualization overhead to be negligible for compute-intensive HPC kernels and applications such as the NAS NPB benchmarks; other studies have investigated virtualization performance for specific HPC application domains [21, 45]. Recently, there has been a spur of research activity in assessing the performance of virtualized resources, in cloud computing environments [15, 39, 42, 37, 40, 38, 26]. In contrast

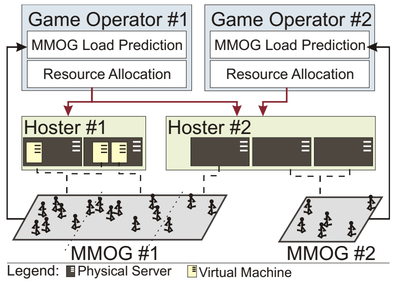


Figure 1 The MMOG ecosystem.

to this body of research, in this work we focus on a different, domain-specific, benchmark-like application with a large number of paying users (see Section 2.1).

Comparison with Our Previous Work This work extends and complements our own previous investigation into the impact of virtualization on the performance of MMOGs [35] in three main ways. First, we extend the analysis of the impact of virtualization on the performance of MMOGs with additional performance metrics. Second, we present new results about the costs of hosting MMOGs based on the real costs of three commercial clouds. Third, using experiments with a real MMOG prototype we show evidence that resource under-allocation, which occurs more often for static resource provisioning and non-virtualized platforms, has a significant impact on key MMOG performance metrics.

4 The Extended MMOG Model

In this section we extend the MMOG ecosystem introduced in [34] from multiple MMOG sessions operated by a single game operator on non-virtualized resources to multiple MMOG sessions operated by several competing game operators on non-virtualized and virtualized resources.

4.1 MMOG Hosting and MMOG Operation

Hosters Our enhanced hosting model consists of *both* conventional data centers that operate pre-installed game servers [34] *and* additional general purpose providers of virtualized cloud resources (see Figure 1). Thus, the extended hosting platform consists of both data centers and cloud providers scattered around the world, where each hoster pools together resources that may serve several games simultaneously. We add to the model a virtualization policy, which describes the characteristics of the virtualization process employed by a hoster; these characteristics are detailed in Section 4.2.

Operators In our previous work [34] the MMOG operators employed four operations to select resources: classifying, sorting, filtering, and prioritizing (see Section 2.1). The actual implementation and use four operations form a versatile policy-based resource selection mechanism. In this work we consider three such policies: The *"efficient"* policy targets low over-allocation by selecting the offers closest to the current request (sorting) and shortest allocation time (classifying). The *"good service"* policy targets low under-allocation by first selecting the offers proportional to the request (sorting), then the offers geographically closest to the request (classifying), and finally the offers with the lowest virtualization overhead (prioritizing). The *"balanced"* policy tries to achieve low under-allocation with reasonable over-allocation, through a compromise between the "efficient" and the "good service" policies. This policy first selects the proportional offers to the request (sorting) and then classifies on virtualization overheads.

4.2 Virtualization Overhead

Instantiation VM instantiation is the process by which a VM is started on a selected resource. Our model for VM instantiation that considers four performance aspects expressed by the corresponding *virtualization overheads*: VM image preparation (characterized by the time t_c), VM transfer (t_x), VM start (t_s), and VM removal (t_r). Performance-wise, the total time needed to instantiate a VM can be expressed as $T = t_c + t_x + t_s + t_r$. Conventional data centers do not exhibit this overhead, but are restricted to pre-deployed software and thus lack the flexibility of dynamic provisioning and on-the-fly deployment of MMOG servers.

We now present the four performance aspects formally; Table 1 presents realistic values for the modeled parameters, and summarizes them into a theoretical average VM (the "Avg. VM" row):

$$t_c = \begin{cases} 0, & \text{VM image pre-created and/or cached,} \\ t_{suspend}, & \text{VM to be suspended,} \\ t_{image}, & \text{"on-the-fly" creation of VM image and/or VM configuration.} \end{cases}$$

Note: $t_{suspend} = f(size(VM_{memory}))$, where VM_{memory} is the size of the instantiated VM's memory (RAM and disk).

$$t_x = \begin{cases} 0, & \text{VM already present,} \\ t_x^z + t_x^{cz} + t_x^u, & \text{VM zipped, the zip is copied, then unzipped,} \\ t_x^c, & \text{VM is copied "as-is".} \end{cases}$$

Note: $t_x^z = f(size(VM_{image}))$, $t_x^c = f(size(VM_{image}), |VM_{instances}|)$, $t_x^u = f(size(VM_{image}))$, where VM_{image} is the size of the un-instantiated VM's data (disk).

$$t_s = \begin{cases} 0, & \text{VM to be used by the same user under the same configuration,} \\ t_{resume}, & \text{VM to be resumed (was suspended),} \\ t_{boot} + t_{cfg}, & \text{otherwise.} \end{cases}$$

Note: $t_{resume} = f(size(VM_{memory}))$. the size of the instantiated VM's memory (RAM and disk).

$$t_r = \begin{cases} 0, & \text{VM to be used by the same user under the same configuration,} \\ t_{stop}, & \text{VM just stops, no VM removal necessary,} \\ t_{cleanup}, & \text{VM removal necessary.} \end{cases}$$

Note: $t_{stop} = f(size(VM_{memory}), |VM_{instances}|)$ and $t_{cleanup} = f(size(VM_{memory}), |VM_{instances}|)$.

<table>
<tr><th>Real VM middleware</th><th>Image Creation (t_c)</th><th>Transfer (t_x)</th><th>Start (t_s)</th><th>Removal (t_r)</th></tr>
<tr><td>Oceano [1]</td><td>0 (pre-created)</td><td>0 (present)</td><td>$t_{boot} + t_{cfg} = 130s$</td><td>n/a</td></tr>
<tr><td>VMPlants [31]</td><td>$t_{image} = 27s + 100s \times S$</td><td colspan="2">$t_x + t_s = 90s + 80s \times S$</td><td>n/a</td></tr>
<tr><td>Kadeploy [9]</td><td>0 (pre-created)</td><td colspan="2">$t_x + t_s = 200s + 0.33s \times N$</td><td>$t_{c/up} = 0.2s \times N$</td></tr>
<tr><td>Shirako [28]</td><td colspan="3">$t_{image} + t_x^z = 100s,\ t_x^{cz} + t_x^u + t_s = 20s + 2s \times N$</td><td>n/a</td></tr>
<tr><td>VW [18]</td><td>0 (pre-created)</td><td colspan="2">$t_x + t_s = 110s + 54s \times N$</td><td>n/a</td></tr>
<tr><td>XGE [17]</td><td>$t_{suspend} = 15s + 22.5s \times S$</td><td>n/a</td><td>$t_{resume} = 20s + 18.3s \times S$</td><td>n/a</td></tr>
<tr><td>VD caches [12]</td><td colspan="2">$t_c + t_x = 24.8s + 0.09s \times N$</td><td>n/a</td><td>n/a</td></tr>
<tr><td>Avg. VM</td><td>$t_c = 0s$</td><td>$t_x = Xfer(S) + 0.09s \times N$</td><td>$t_s = 80s$</td><td>$t_r = 0s$</td></tr>
</table>

Table 1 Realistic parameter values for the VM instantiation model, where S is the size of the data to transfer in GB, N is the number of VMs to instantiate, and $Xfer(S)$ is the transfer time for data of size S from the data source to the resource that instantiates the VM (e.g., for a 1Gbps transfer bandwidth in ideal conditions $Xfer^{ideal}_{1Gbps}(S) = 10s \times S$).

In the remainder of this work, we set the VM preparation time to zero after assuming conservatively that MMOG images can only be pre-created, the VM start time to 80 seconds as determined in our evaluation of real commercial clouds [38, 26], and the VM removal time to zero as images are not saved to avoid data corruption across sessions.

Execution Performance studies using general-purpose benchmarks have shown that the overhead incurred by virtualization can be below 5% for computation and 15% for networking [3], and similarly low for other MMOG-relevant resources; we call *virtualization penalty* the aggregate performance overheads. We expect that future MMOG-specialized hosters that will employ cloud computing/virtualization technology will achieve a similarly low virtualization penalty.

5 Simulation Results

Based on the extended model introduced in the previous section, we have evaluated through trace-based simulation the use of virtualized and non-virtualized resources, either separately or in conjunction, in scenarios involving one or more MMOG providers.

Summary and implication of findings We find that the use of virtualized resources without adequate policies can incur severe degradation of the gameplay experience (Section 5.2). We also find that the average resource under-allocation increases linearly with the VM size and the VM start time, and may depend on the VM transfer bandwidth and the virtualization penalty (Section 5.3). Last, we find that the resource selection mechanism determines the achieved performance when the game operators compete for virtualized resources (Section 5.4), and that the MMOG hosters have strong incentives to support MMOGs through adequate virtualization policies in a multi-hoster market (Section 5.5).

5.1 Experimental Setup

Input Workload We perform experiments using the same traces as in our previous work [34, 35]. These traces have been collected from RuneScape [29], a real MMOG ranked second after World of Warcraft by number of active paying customers in the US and European markets [34]. RuneScape is a generic MMOG: RuneScape combines an overall MMORPG theme with elements of other game genres, so that in specific parts of the game world (minigames) the player

Location		Data Centers	Machines (total)
Continent	Country		
Europe	Finland	2	8
	Sweden	2	8
	U.K.	2	20
	Netherlands	2	15
North America	U.S. (West)	2	35
	Canada (West)	1	15
	U.S. (Central)	1	15
	U.S. (East)	2	32
	Canada (East)	1	10
Australia	Australia	2	8
Total	World-Wide	17	166

Table 2 Physical characteristics of the data centers.

interaction follows different rules and patterns than in MMORPGs. Various levels of player interactivity coexist in RuneScape, so the game load cannot be trivially computed (i.e., using linear models [44]). The traces correspond to two weeks of RuneScape operation; each server group used by the game operators is sampled every two minutes (called *simulation steps*) for the number of active players. The detailed resource demand is determined from these traces by employing the game load models introduced and analyzed in our previous work [35].

Environment We perform experiments in a simulated RuneScape-like environment. The data centers are located on three continents and seven countries; their characteristics are summarized in Table 2. For virtualized resources, we use the VM overhead model introduced in Section 4.2.

Performance Metrics We evaluate the quality of the game experience using a *resource under-allocation* metric that characterizes the percentage of resources that have not been allocated from the amount necessary for the seamless execution of the MMOG. We define resource under-allocation $U(t)$ within one simulation step T as: $U(t) = \frac{(A-L)\cdot t}{T\cdot L_{\max}} \cdot 100\%$, where T is the duration of the under-allocation event, A is the amount of allocated resources, L is the amount of needed resources (measured from the traces), and $L_{\max}$ is the maximum load determined by game design. Resource under-allocation has an important impact on the gameplay experience: not having enough resources quickly increases the game response time and degrades the game experience. For example, in a FPS game, during an under-allocation event, the minimum required client update frequency will not be met (e.g. updates at less than 25Hz). As a result, players may quit, which may further trigger mass-departure events [34]. We also define the resource over-allocation $O(t)$ as: $O(t) = \frac{(L-A)\cdot t}{T\cdot L_{\max}} \cdot 100\%$, which characterizes allocation of a resource in surplus, relative to the necessary for the seamless MMOG execution.

5.2 Verifying the Virtualization Premise

An important premise of our work is that the virtualization policy has an important impact on the resource under-allocation, and thus on the quality of the game experience. We now show that this is indeed the case. Toward this end, in this experiment we assess the resource under-allocation for five virtualization policies that can be used by hosters in practice. These policies combine the realistic hoster policies specified by real hosters within the edutain@grid project, and the real technical limitations we have observed in our experiments with

Policy	CPU [units]	Mem. [units]	Time [min.]	t_s [s]	t_x [s]	Virt. penalty
Ideal	0.4	0.25	360	0	0	0
Policy 1	0.4	0.25	360	30	5	1%
Policy 2	0.4	0.25	360	60	10	3%
Policy 3	0.4	0.25	360	90	20	5%
Policy 4	0.4	0.25	360	120	40	10%

Table 3 Virtualization policies for hosters.

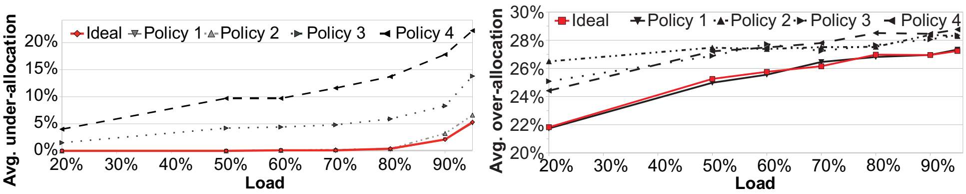


Figure 2 The impact of the virtualization policy, under increasing load, (left) on the resource under-allocation; (right) on the resource over-allocation (non-zero start for the vertical axis).

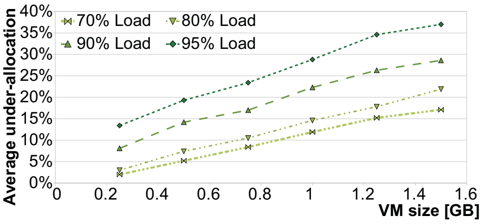
clouds and data centers [38, 26]. The five policies are summarized in Table 3. The "Ideal" policy has no virtualization overheads, while the policies "Policy 1" through "Policy 4" have increasingly larger virtualization overheads and thus become increasingly unfavorable for MMOG hosting. A cloud operator may decide to employ a policy that is more favorable to MMOG hosting, but this can be detrimental to other cloud users, will incur higher up-front and system tuning costs, and may require R&D investments to adapt the existing cloud to MMOG hosting. We use the five virtualization policies, in turn, with the setup presented in Section 5.1.

The virtualization policy has an important impact on the resource under-allocation (see Figure 2 (left)). The maximal vertical distance between the five curves, which correspond each to a policy, indicates that different virtualization policies have an important impact on the average resource under-allocation. In particular, Policies 3 and 4 incur large resource under-allocations, which would affect severely the quality of the game experience. The small difference between Policies 1 and 2, and the Ideal virtualization policy, is due to the use of a load prediction service (see Section 2.1) that effectively cancels the overheads of virtualization as resources are allocated before they are actually required. conclude that the virtualization policy has an important impact on resource under-allocation.

The virtualization policy does not have an important impact on the resource over-allocation (see Figure 2 (right)). The resource over-allocation is less dynamic than the resource under-allocation for the same load variation, showing only a slight increase at higher loads. The policy selection is more important when the ecosystem is lightly loaded. Because the exhibited variations of the resource over-allocation with both load and virtualization policies is relatively low, we conclude that the main factors determining resource over-allocation are the efficiency of the allocation algorithm and the proportionality between the resource requests and resource offers, which concurs with our findings in previous work [33].

Experiment Focus	Value Ranges for Characteristic t_s [seconds]	S size [GB]	Xfer BW [Mbit]	Virt. penalty
VM start	20 – 170	0.5	100	5%
VM size	80	0.25 – 1.5	100	5%
VM x'fer. bw.	80	0.5	100 – 1000	5%
Virt. penalty	80	0.5	100	2 – 12%

Table 4 The value ranges for the characteristics investigated in Section 5.3. Each row presents the value or value range used in the experiment that investigates one focus characteristic.



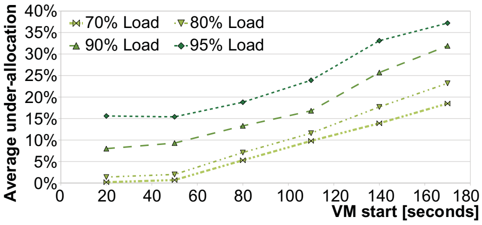


Figure 3 Variation of under-allocation, for different loads, (left) with the image size; (right) with the image boot time.

5.3 *Analysis of Virtualization Policies*

To explain the effects of the virtualization policy on the game experience we explore the impact of each of the components of the virtualization policy on the resource under-allocation, when a single MMOG is hosted.

We used the experimental setup presented in Section 5.1, and varied in turn each of the four characteristics affecting the virtualization overheads introduced in Section 4.2: the VM start time (t_s), the VM size (S), the bandwidth available for transferring the VM image, and the virtualization penalty. The ranges of values explored in the four separate experiments are summarized in Table 4. In each experiment, the single values are selected based on realistic data published by major providers of virtualized resources and other benchmark reports. We set the VM size set to 0.5 GB (which is average for base images with pre-deployed software), the transfer bandwidth to 100 Mbps, and the performance penalty of the game servers running inside virtualized resources to 5% [3]. We also study in each experiment the impact of the system load, for which we look at four values that are common in practice: 70%, 80%, 90%, and 95%.

The average resource under-allocation increases linearly with the VM size and the VM start time (see Figure 3). Since the curves in Figure 3 do not show artifacts, we conclude that the impact of these parameters on the quality of gameplay is predictable, with the resource under-allocation roughly growing by 10% per GB of VM image size and for each minute of VM start time.

The average resource under-allocation may depend on the VM transfer bandwidth and the virtualization penalty (see Figure 4). The figure shows that the VM transfer bandwidth and virtualization penalty have little or no impact on the resource under-allocation at loads lower than 90%, since their effect is negligible compared to the other virtualization parameters and can be easily hidden by the proactive (prediction-based) resource allocation. However, the under-allocation exhibits irregular behavior at loads of 90% and higher.

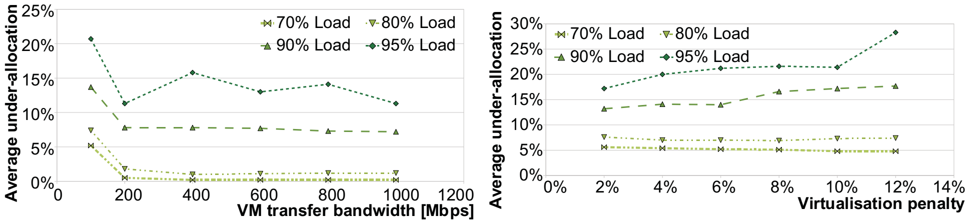


Figure 4 Variation of under-allocation, for different loads, (left) with the image transfer bandwidth; (right) with the with virtualization penalty.

Policy	VM start [s]	Image size [GB]	Xfer BW [Mbps]	Virt. penalty	Time [min.]
Ideal	0	0	0	0%	60
Virtualization 1	60	0.25	400	2%	120
Virtualization 2	80	0.50	300	5%	180
Virtualization 3	120	0.75	200	10%	240
Virtualization 4	180	1.00	100	12%	300

Table 5 Virtualization policies for the multi-MMOG hosting scenario.

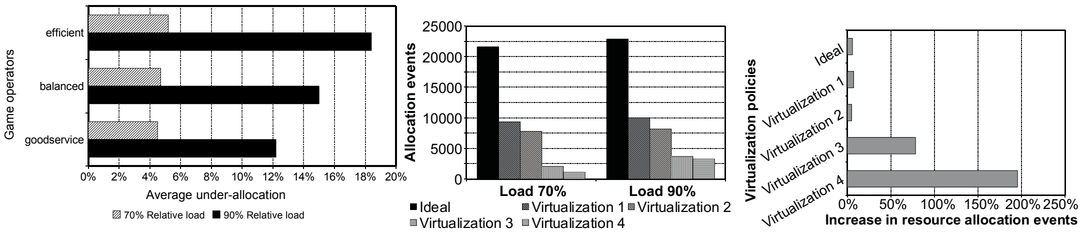


Figure 5 Multi-MMOG hosting scenario: (left) Performance of three resource selection policies; (middle) Number of allocation events at 70% and 90% load; (right) Percentage of additional allocation events at 90% vs 70% load.

5.4 Resource Competition for Multi-MMOGs

In this experiment we investigate the impact of the policy for resource allocation in a multi-hoster environment. We used the experimental setup presented in Section 5.1 and considered several MMOG operators who use different policies to compete for resources. The three resource allocation policies for MMOG operators considered here are assigned in equal proportion (33%) to the operators. Additionally, we considered the five different policies summarized in Table 5, which we assigned to hosters in equal proportion (20%). Policy “Ideal” represents the allocation policy for non-virtualized resources (ideal from a virtualization standpoint), while the policies “Virtualization 1” through “Virtualization 4” are defined with increasingly unfavorable virtualization parameter values (overheads). The competition for resources is ensured by setting the average load high; we experiment with two values, 70% and 90%.

The selection mechanism is crucial for achieving good performance when the resources are in short supply and game operators compete for offers (see Figure 5 (left), which depicts the average under-allocation for the three resource allocation policies, for 70% and 90% load.) The similar performance of the policies at 70% load is due to the lack of competition for resources. In contrast, for the 90% load the “good service” game operator provides the best under-allocation,

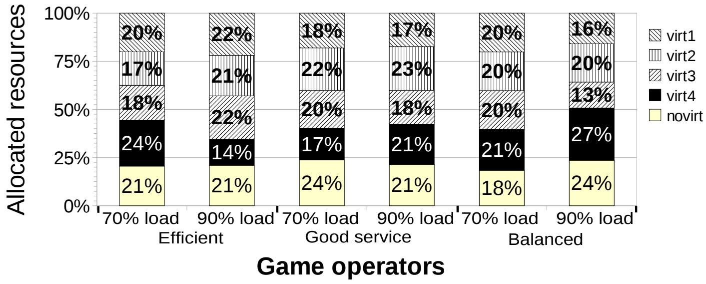


Figure 6 Distribution of virtualized resources among the game operators.

followed by the "balanced" and the "efficient" operators. To better understand the behavior of the resource allocation policies, Figure 6 shows the distribution of allocated resources for each policy. The "good service" exhibits a strong preference toward the resources with low virtualization overheads and is using the least amount of resources provisioned from Virtualization 4 (unfavorable) hosting. The "efficient" game operator favors Virtualization 1 resources, which have the shortest time bulk but high virtualization overheads.

How many allocations/reallocations occur under different virtualization policies? A large number of allocation events is indicative of an environment that requires more (human) monitoring. Figure 5 (middle) compares the behavior of the MMOG ecosystem investigated in this section by number of allocation events, under different load conditions; the relative changes between the 70% and the 90% loads are depicted in Figure 5 (right). The figures show that at high load the ideal and low virtualization overhead resources are almost fully allocated, which forces the MMOG operators to utilize virtualized resources with high virtualization overhead regardless of their selection policies.

5.5 Cloud Hosting Costs

To estimate the costs for hosting MMOGs on cloud resources, we modeled the virtualized resources starting from the popular Amazon Elastic Cloud Computing (EC2). The resource requirements of the RuneScape game servers are approximately equivalent to the standard small instance (especially in terms of memory requirements), which is priced between \$0.085 and \$0.095 per hour. Throughout this section, the static cost represents the cost for obtaining all the resources from the cloud and keeping them allocated from the beginning until the end of the experiment, which is akin to static over-provisioning of resources. Conversely, the dynamic cost represents the cost for obtaining the resources temporarily under our dynamic resource allocation strategy. We do not consider here the network consumption.

For our first cost-related experiment, we varied the average session load from 20% to 95%, representing the amount of clients participating in the distributed game session relative to the total maximum serviceable clients with the given resources. Figure 7 compares the hosting costs for the conventional static provisioning and for our dynamic allocation method for the entire time span. For the dynamic allocation, the cost is directly proportional to the session load, the small cost step between the 50% and 60% session loads being generated by

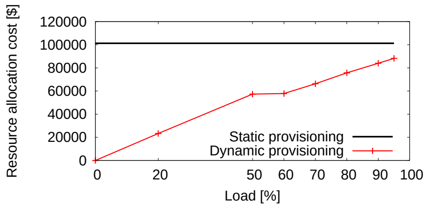


Figure 7 Estimated costs of hosting a RuneScape session with a maximum of 280, 000 clients on Amazon EC2 standard small instances.

Cloud provider	*Instance type*	*Description*	*Price/hour*
Amazon EC2 (`aws.amazon.com/ec2/`)	Standard small	1.1 GHz Xeon, 2 GB	$0.085
FlexiScale (`flexiscale.com/`)	2 GB	Mid-range Xeon, 2 GB	$0.159
NewServers (`www.newservers.com/`)	Medium	2 × 3.2 GHz Xeon, 2GB	$0.170

Table 6 Cloud resource providers (hosters).

Load	Estimated yearly MMOG hosting costs					
	Amazon EC2		FlexiScale		NewServers	
	Dynamic	Static	**Dynamic**	Static	**Dynamic**	Static
0%	**$0**	$101,266	**$0**	$189,426	**$0**	$202,531
20%	**$23,326**	$101,266	**$40,920**	$189,426	**$38,468**	$202,531
50%	**$57,345**	$101,266	**$100,404**	$189,426	**$97,495**	$202,531
60%	**$57,830**	$101,266	**$101,829**	$189,426	**$98,179**	$202,531
70%	**$66,299**	$101,266	**$116,458**	$189,426	**$114,775**	$202,531
80%	**$75,709**	$101,266	**$133,111**	$189,426	**$129,119**	$202,531
90%	**$84,007**	$101,266	**$147,055**	$189,426	**$142,578**	$202,531
95%	**$88,199**	$101,266	**$155,039**	$189,426	**$149,793**	$202,531

Table 7 Estimated yearly Cloud hosting costs for a RuneScape session with maximum 280, 000 clients.

the minimum allocation time interval of one hour for Amazon resources. This interval can sometimes represent a too coarse-grain measure for highly dynamic game session loads generating higher over-provisioning values. In this case, **hosters should tune their clouds to provide better MMOG support**.

In the second cost-related experiment, we compared different Cloud providers by selecting instances closest to the RuneScape hardware requirements from Amazon EC2, FlexiScale, and NewServers (see Table 6). Using the same RuneScape trace data, we ran sequential simulations on each of these resource types and extrapolated the yearly costs for hosting the MMOG session. We find that **dynamic provisioning achieves significant cost reduction vs static over-provisioning, even when the load is high** (see Table 7). The results detailed in Table 7 show that our method achieves a cost decrease from two up to five times in the most favorable case (excluding the sessions without load) compared to the RuneScape static over-provisioning. Even when the actual resource consumption becomes close (95%) to the statically provisioned resources, dynamic provisioning leads to cost reductions of at least 10%; the actual gain depends on the hourly resource cost. From the three Cloud providers, Amazon EC2 offers the best RuneScape hosting price in all cases because of the cheap standard small instances match best the RuneScape processor and memory requirements. On the other hand, NewServers offers the most expensive static allocation price because of the small amount of memory in one instance, which prohibits the servers from using the entire expensive processor power available. Our dynamic allocation solution is capable of efficiently handling these disproportioned NewServers instances and achieves a lower provisioning cost than static resource provisioning from FlexiScale.

6 Empirical Results

In this section we investigate the impact of resource provisioning on game play experience. Specifically, we study how resource under-allocation and over-allocation affect performance metrics that determine the game play experience, such as the game response latency [10, 14, 23].

Summary and implication of findings We find that players experience a substantial decrease in the quality of gameplay during periods of resource under-allocation, even when the under-allocation is low. Thus, it is crucial to avoid or to limit the duration of under-allocation periods. Conversely, the over-allocation situations have the potential of improving the game play quality, but the gain is marginal compared to the large amount of wasted resources (significant improvements are observed only at 80% resource over-allocation or higher).

6.1 Experimental Setup

Environment To determine the exact relationship between resource under- and over-allocation and game experience in our proposed environment, as opposed to traditional gameplay testing [19], we utilize a real game prototype built with RTF, a game parallelization library developed by University of Münster [22] within edutain@grid. Even though our prototype does not include recent latency-improving and bandwidth-reducing approaches, see [24, 5, 8], it can support up to 250 concurrent clients interacting in close encounters, on a single machine, and can scale up to thousands of concurrent clients in a single game session hosted on a distributed, heterogeneous system, which is comparable to state-of-the-art MMOGs such as EVE-Online (`http://www.eveonline.com/`). The experimental setup for the our game prototype consists of a game world with uniform terrain (i.e. no obstacles), transparently divided in two game zones. Each game zone also supports the replication technique for load distribution. To generate load on the game servers we used "bots", computer controlled clients, driven by Artificial Intelligence (AI) profiles which determine their behaviour within the game: *aggressive*, *team player*, *scout*, *camper*. The experiments are run utilizing two dual-core machines equiped with 2.66 Gigahertz Intel Core 2 Duo processors and 2 Gigabytes of memory for hosting the game servers and eight other machines located in different networks for running the bots.

Input Workload The setup consists of one machine running a game prototype server program to which clients can connect. In order to achieve the different load levels required by our experiments we utilize *bots*, artificial-intelligence driven clients, which are running on several other machines. We allocate the resources of a single machine for running the game server and we generate increasing load by sequentially connecting *bots*, starting from low and reaching high loads (high over-allocation to high under-allocation, respectively).

Performance Metrics We study the evolution of the *network latency*, *network incoming* and *outgoing bandwidth*. The network latency measurement is computed as an average of all existing client-server connections' instantaneous latencies, while the network bandwidth represents the total bandwidth utilized by all existing connections. The results are presented as distributions of these metrics'

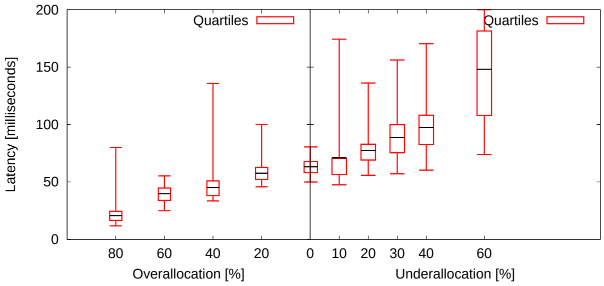


Figure 8 Statistical properties of the network connection latency under different over- and under-allocation conditions.

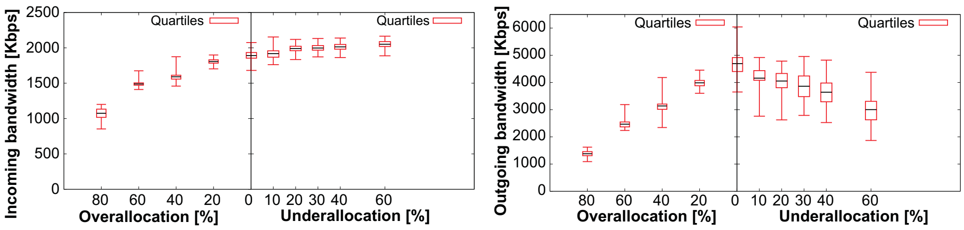


Figure 9 Statistical properties of the network bandwidth under different over- and under-allocation conditions: (left) incoming; (right) outgoing.

values for different degrees of resource over-/under-allocation. Each distribution is based on 1,000 measurements.

6.2 Experiments on a Game Prototype

Resource under-allocation leads quickly to unplayable games (see Figure 8). This batch of experiments evaluates the behavior of our real game prototype in conditions of both over- and under-allocation of resources. Figure 8 shows a stable and slowly rising network latency with the decrease resource over-allocation. The under-allocation side of the graph shows not only that the network latency quickly increases to levels which make the game unplayable (values higher than $80ms$ in the case of our game pilot) but also an increasing jitter which can introduce unfairness toward some clients. The large variance of the network latency is characteristic to traditional game servers, when overloaded.

Outgoing bandwidth gains due to under-allocation may be canceled by incoming bandwidth increase and high latency (see Figure 9 (left)). During our experiments, the utilized network bandwidth exhibits a significant decrease in the outgoing bandwidth, although the number of connected clients is increasing (see Figure 9 (right)). This happens because, as the processor gets saturated, more time is spent computing the state updates and less dispatching updates to clients. This further results in the “freezing” between client updates of the client-controlled entities in the game world, which is truly detrimental to the quality of game play. Moreover, the decrease in the consumed outgoing bandwidth adds a new dimension to the network latency results (Figure 8) because it indicates that some of the client updates are dropped (in effect, postponed to the next cycle) in order to maintain the necessary client update frequency. Since network is a provisioned resource, even though under-allocation reduces outgoing bandwidth consumption, it may lead at the same time to higher incoming bandwidth consumption and game latency. Thus, the network latency and bandwidth need to be considered in conjunction when evaluating the gameplay experience.

7 Conclusion

The MMOG operators who can afford it currently operate dedicated multi-server infrastructures spread across the world to support the highly variable resource demand and to cope with the players' sensitivity to latency. The recent introduction of virtualization in data centers promises much lower costs through uniform-behaving resources that can be provisioned on-demand for MMOG hosting from specialized commercial clouds. However, the expected cost reduction also comes with a loss of performance due to virtualization overheads. Moreover, the costs incurred by various commercial clouds differ.

In this work we have investigated the impact of virtualization on the performance and operational costs of MMOGs. Toward this end, we have modeled an MMOG ecosystem that comprises both non-virtualized and virtualized resources. In our ecosystem, multiple competing game operators manage several MMOG sessions, and virtualized cloud computing resources can be provisioned for MMOG hosting together with conventional data center machines. We have analyzed through trace-based simulation the effect of using virtualized resources on the performance of MMOGs. We have shown that both the virtualization policies employed by MMOG hosters and the resource allocation policies used by MMOG operators are important for achieving good performance. We have found that MMOG hosting on virtualized resources is practical even if the virtualization overheads are taken into account, although non-virtualized resources are still preferable if performance is the only selection criterion. We have also evaluated empirically the impact of (the lack of) virtualization on the quality of gameplay. We have learned that using virtualized resources can negatively affect the quality of gameplay when the system is heavily loaded, but also that resource under-allocation, which is common in self-owned and non-virtualized infrastructures, increases latency by a significant margin.

For the future, we intend to explore more in-depth the dynamic provisioning policy, toward finding policies that optimize the performance-cost trade-off for highly dynamic game populations. We will also investigate how hosters can improve their support for MMOGs, without affecting their current users' workloads. We plan to also study empirically the validity of our virtualization model at large-scale, where management overheads and resource sharing may be non-trivial to capture.

Acknowledgements This research is partially funded by the Austrian Research Promotion Agency (FFG) project 824925, by the Austrian Science Fund (FWF) project TRP 72-N23, and by a TU Delft ICT Talent Grant.

References

[1] K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt, M. Kalantar, S. Krishnakumar, D. Pazel, J. Pershing, and B. Rochwerger. Oceano-SLA based management of a

computing utility. In *INM*, pages 855–68, 2001.

[2] M. Aron, P. Druschel, and W. Zwaenepoel. Cluster reserves: a mechanism for resource management in cluster-based network servers. In *SIGMETRICS*, 2000.

[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. L. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *SOSP*, 2003.

[4] R. Bartle. *Designing Virtual Worlds*. New Riders, New York, USA, 2003.

[5] A. R. Bharambe, J. R. Douceur, J. R. Lorch, T. Moscibroda, J. Pang, S. Seshan, and X. Zhuang. Donnybrook: enabling large-scale, high-speed, peer-to-peer games. In *SIGCOMM*, pages 389–400, 2008.

[6] I. Blizzard Entertainment. World of warcraft. http://www.worldofwarcraft.com/.

[7] R. Bradshaw, N. Desai, T. Freeman, and K. Keahey. A scalable approach to deploying and managing appliances. In *TeraGrid*, Madison, WI, 2007.

[8] E. Buyukkaya and M. Abdallah. Efficient triangulation for p2p networked virtual environments. *Multimedia Tools Appl.*, 45(1-3):291–312, 2009.

[9] F. Cappello, E. Caron, M. J. Daydé, F. Desprez, Y. Jégou, P. V.-B. Primet, E. Jeannot, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, B. Quétier, and O. Richard. Grid'5000: a large scale and highly reconfigurable grid experimental testbed. In *Grid*, pages 99–106, 2005.

[10] K.-T. Chen, P. Huang, and C.-L. Lei. How sensitive are online gamers to network quality? *Commun. ACM*, 49(11):34–38, 2006.

[11] L. Cherkasova and R. Gardner. Measuring cpu overhead for i/o processing in the xen virtual machine monitor. In *USENIX ATC*, pages 387–390, 2005.

[12] T. Chiba, T. Endo, and S. Matsuoka. High-performance mpi broadcast algorithm for grid environments utilizing multi-lane nics. In *CCGrid*, pages 487–494, 2007.

[13] D. Chisnall. *The Definitive Guide to the Xen Hypervisor*. Prentice Hall, 2007.

[14] M. Claypool and K. T. Claypool. Latency and player actions in online games. *Commun. ACM*, 49(11):40–45, 2006.

[15] E. Deelman, G. Singh, M. Livny, J. B. Berriman, and J. Good. The cost of doing science on the cloud: the Montage example. In *SC*, page 50, 2008.

[16] DFC Intelligence Team. Online games report, 2004. [Online] Available: `http://www.dfcint.com`.

[17] N. Fallenbeck, H.-J. Picht, M. Smith, and B. Freisleben. Xen and the art of cluster scheduling. In *VTDC*, pages 4–11, 2006.

[18] I. T. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, and X. Zhang. Virtual clusters for grid communities. In *CCGrid*, pages 513–520, 2006.

[19] T. Fullerton, C. Swain, and S. Hoffman. *Game Design Workshop: Designing, Prototyping, and Playtesting Games*. CMP Books, 2005.

[20] Gamebryo. Butterfly Grid/Emergent Platform. [Online] Available: `http://www.emergent.net/`, Aug 2008.

[21] L. Gilbert, J. Tseng, R. Newman, S. Iqbal, R. Pepper, O. Celebioglu, J. Hsieh, and M. Cobban. Performance implications of virtualization and hyper-threading on high energy physics applications in a grid environment. In *IPDPS*, 2005.

[22] F. Glinka, A. Ploß, J. Müller-Iden, and S. Gorlatch. Rtf: a real-time framework for developing scalable multiplayer online games. In *NetGames*, pages 81–86, 2007.

[23] T. Henderson. Latency and user behaviour on a multiplayer game server. In *Networked Group Communication*, volume 2233 of *LNCS*, pages 1–13, 2001.

[24] S.-Y. Hu, J.-F. Chen, and T.-H. Chen. Von: a scalable peer-to-peer network for virtual environments. *IEEE Network*, 20(4):22–31, 2006.

[25] W. Huang, J. Liu, B. Abali, and D. K. Panda. A case for high performance computing with virtual machines. In *ICS*, pages 125–134, 2006.

[26] 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 Trans. on Parallel and Distrib. Sys.*, 2010. (in print).

[27] A. Iosup, T. Tannenbaum, M. Farrellee, D. H. J. Epema, and M. Livny. Inter-operating grids through Delegated MatchMaking. *Sci. Prog.*, 16(2-3):233–253, 2008.

[28] D. E. Irwin, J. S. Chase, L. E. Grit, A. R. Yumerefendi, D. Becker, and K. Yocum. Sharing networked resources with brokered leases. In *USENIX ATC*, 2006.

[29] Jagex, Ltd. Runescape. http://www.runescape.com.

[30] A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder, M. Sviridenko, and A. N. Tantawi. Dynamic placement for clustered web applications. In *WWW*, pages 595–604, 2006.

[31] I. Krsul, A. Ganguly, J. Zhang, J. A. B. Fortes, and R. J. O. Figueiredo. Vmplants: Providing and managing virtual machine execution environments for grid computing. In *SC*, page 7, 2004.

[32] A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel. Diagnosing performance overheads in the xen virtual machine environment. In *VEE*, pages 13–23, 2005.

[33] V. Nae, A. Iosup, and R. Prodan. Dynamic resource provisioning in massively multiplayer online games. *IEEE Trans. on Parallel and Distrib. Sys.*, 2009. (in print, DOI: `10.1109/TPDS.2010.82`).

[34] V. Nae, A. Iosup, R. Prodan, D. Epema, and T. Fahringer. Efficient management of data center resources for massively multiplayer online games. In *SC*, 2008.

[35] V. Nae, A. Iosup, R. Prodan, and T. Fahringer. The impact of virtualization on the performance of massively multiplayer online games. In *NetGames*, pages 1–6, 2009.

[36] V. Nae, R. Prodan, and T. Fahringer. Neural network-based load prediction for highly dynamic distributed online games. In *Euro-Par*, 2008.

[37] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The Eucalyptus open-source cloud-computing system. 2008. UCSD Tech.Rep. 2008-10. [Online] Available: `http://eucalyptus.cs.ucsb.edu/`.

[38] S. Ostermann, A. Iosup, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. An early performance analysis of cloud computing services for scientific computing. In *CloudComp 2009*, volume 34 of *LNICST*, pages 1–10.

[39] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel. Amazon S3 for science grids: a viable solution? In *DADC*, pages 55–64, 2008.

[40] B. Quétier, V. Néri, and F. Cappello. Scalability comparison of four host virtualization tools. *Grid Comput.*, 5(1):83–98, 2007.

[41] J. Rolia, X. Zhu, M. F. Arlitt, and A. Andrzejak. Statistical service assurances for applications in utility grid environments. *Perform. Eval.*, 58(2-3):319–339, 2004.

[42] E. Walker. Benchmarking Amazon EC2 for HP Scientific Computing. *Login*, 33(5):18–23, 2008.

[43] B. Walters. VMware virtual platform. *Linux Journal*, page 6, 1999.

[44] M. Ye and L. Cheng. System-performance modeling for massively multiplayer online role-playing games. *IBM Systems Journal*, 45(1):45–58, 2006.

[45] L. Youseff, R. Wolski, B. C. Gorda, and C. Krintz. Paravirtualization for HPC systems. In *ISPA Workshops*, volume 4331 of *LNCS*, pages 474–486, 2006.

[46] W. Yu and J. S. Vetter. Xen-based HPC: A parallel I/O perspective. In *CCGrid*, pages 154–161, 2008.