Autonomic Service Hosting for Large-Scale Distributed MOVE-Services

Bruno Van Den Bossche

Supervisor(s): Filip De Turck, Bart Dhoedt

I. Introduction

Recent years have shown an increasing popularity of Massively Online Virtual Environments (MOVEs). A prime example of such an application is the Massively Multiplayer Online Game (MMOG) World of Warcraft which currently has over 10 million paying users. At peak usage, over 500,000 players are online at the same time, although spread across multiple independent parallel instances of the virtual world. Furthermore, do current trends indicate that MOVEs are no longer restricted to games but are finding their use in online communities and corporate environments, for example virtual meeting rooms and the IBM Metaverse project.

Due to this increasing demand for MOVEs, we argue that there is a need for a generic service hosting platform to support the increasing demand for distributed online environments. Current MMOG hosting infrastructures are focused on specific game implementations, lack the flexibility to host multiple virtual worlds and are often not capable of dynamic resource usage optimization. Current Service Hosting platforms focus on distributing parallel or independent service components to optimize the load distribution. We propose the creation of a platform geared toward service hosting providers which is capable of hosting multiple independent MOVEs and dynamic resource usage optimization.

A successful hosting solution for MOVEs will require support in both the MOVE-application itself as the MOVE-hosting platform. To enable fine grained load balancing the virtual world can be divided into small parts called microcells [1]. By dynamically reassigning these microcells to different servers it is possible to shift the load generated by a certain cell from one server to another and as such perform load balancing. As such, virtual worlds with a very uneven player distribution can still be supported with a minimal set of resources. Furthermore does this scalable approach allow the creation of continuous virtual wolds where the number of players does not need to be limited due to server capacity limits.

II. Problem Description

Traditional load balancing techniques typically redirect client requests to separate servers running the same service, possibly with fail-over and session replication to increase the robustness of the platform. These techniques have already been applied to middleware platforms to make this application independent [2]. This approach is not suitable for use with MOVEs as it is geared toward parallel deployments of the same application with independent non-interaction clients. The authors of [3] do propose a method for dynamically relocating services at runtime. However, the services are assumed to be atomic units which can be replicated without any additional synchronization or session management costs, nor do they offer support for service to service communication.

B. Van Den Bossche is with the Department of Information Technology, Ghent University (UGent), Gent, Belgium. E-mail: bruno.vandenbossche@intec.ugent.be .

The requirements of a MOVE-service hosting platform include both load balancing to optimize the Quality of Experience of the end users and minimizing the required resources and performing these actions at runtime without human intervention.

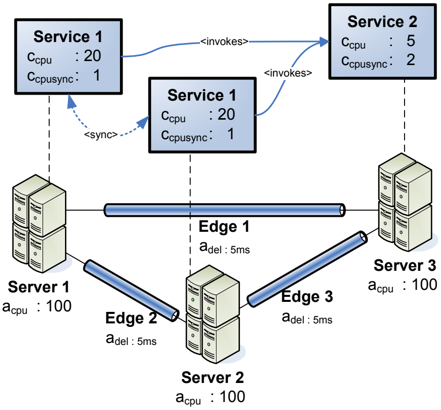

Figure 1. An example of cost and load parameters used in the formal problem formulation and possible interactions between service components.

A middleware platform capable of autonomic management requires the knowledge of both the application and the available resources as shown in Figure 1. Each application component requires a given number of processing cycles, consumes a given amount of memory, generates given amount of network traffic etc. However, these resources are only available in limited amounts. We assume that the costs of executing a given service or application component, as well as the interactions between different components can be sufficiently accurately measured or estimated. This includes the number of invocations and the associated delays.

It is possible to generate an optimal solution for this problem using an Integer Linear Programming (ILP) approach. However this is not scalable and only suitable to solve relatively small problems. Therefor an optimizing heuristic has been developed to generate possible solutions. This heuristic analyzes the dependencies between services, the amount of bandwidth required for the communication between services and any synchronization costs that might be required to maintain a consistent virtual world. These dependencies are assigned a weight which allow us to optimize the deployment for minimal delay, minimal bandwidth consumption, minimal required processing power or a combination thereof. Choosing the actual weights allows us to tune the heuristic to the specific needs of a given MOVE.

III. Conclusions

In this paper we introduced the concept of a generic Service Hosting platform targeted toward Massively Online Virtual Environments or MOVEs. Such a platform makes it possible to let third party providers efficiently host multiple MOVEs with the flexibility of dynamic resource allocation. This means it is no longer necessary to provide a dedicated infrastructure for every single MOVE capable of handling any peak load.

A heuristic was developed to minimize the resource usage and optimize the overall deployment on a minimal resource set. The heuristic is validated using an Integer Linear Programming solution and can be tuned to generate a deployment targeted with specific requirements such as a minimal delay experienced by the end-user.

References

[1] Bart De Vleeschauwer, Bruno Van Den Bossche, Tom Verdickt, Filip De Turck, Bart Dhoedt, and Piet Demeester, “Dynamic microcell assignment for massively multiplayer online gaming,” in *Proceedings of Netgames 2005: 4th Workshop on Network and Systems Support for Games*, New York, USA, 10–11 October 2005.

[2] Jaiganesh Balasubramanian, Douglas C. Schmidt, Lawrence Dowdy, and Ossama Othman, “Evaluating the performance of middleware load balancing strategies,” in *EDOC ’04: Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International*, Washington, DC, USA, 2004, pp. 135–146, IEEE Computer Society.

[3] Constantin Adam and Rolf Stadler, “Service middleware for self-managing large-scale systems,” *IEEE Transactions on Network and Service Management*, vol. 4, no. 3, pp. 50–64, December 2007.