Extending Online Communities through Virtual Parallel Systems

Chu Jong
Illinois State University, USA

Pruthikrai Mahatanankoon
Illinois State University, USA

INTRODUCTION

Internet technology has extended human social interactions across geographical boundaries. Using information and communication technology (ICT) as ways to establish social relationships (Palmer, 1998), virtual communities (VCs) provide the means for people to communicate and to store and share information for various social meetings regardless of time and distance. VCs offer a variety of social interactions ranging from casual and leisure activities to strictly business-related transactions. Many online retail stores use message boards as ways for customers to share information and ideas with other customers. Online forums and chat rooms provide discussion and support facilities for any interest group imaginable from cooking and gardening to stocks and politics.

The value of VCs derives from various functions being delivered. Armstrong and Hagel (1996) suggest that the success of virtual communities rely on the ability to meet "multiple social and commercial needs." To facilitate these needs, online communities generally charge their members minimal fees, either for usage, content, or advertising. The goals are to generate revenue and in some cases cover the operating expenses. However, with new evolutional VCs such as virtual museums, virtual libraries, virtual organizations, and telemedicine, the operating costs for providing such services can be tremendous.

VCs require significant investment in their infrastructures, which can be reduced by implementing an enhanced parallel *p*-split mechanism on their existing servers. The goal of this article is to introduce new practical ideas to manage VCs by focusing on the underlying hardware and software infrastructures that support virtual operations. The structure of this article includes (1) an introduction to the functions and services that a VC system should have and its potential growth, and (2) the concept and basic mechanisms of the *p*-split with the benefit of using *p*-split to generate virtual servers. This article then summarizes the current progress and provides an outline for the future VC system research. The appendix provides some technical aspects in terms of the software module and system overheads of the *p*-split.

SERVICES OF VCs

ICTs, such as e-mails, chat rooms, and so forth, are the main enablers for VCs, and needless to say, these cybercommunities need state-of-the-art information technology to support their operations. To achieve this end, a fast and reliable information and communication infrastructure is probably the most important facilitating factor for any VC. In general, this infrastructure delivers, manages, and hosts various user-related services on several centralized servers, which are accessible through the Internet. Community members may subscribe to these services with minimal fees; however, in many cases, VCs hosting operators cover their operating cost through advertising or product promotion. For example, online book retailers create message broads for reader groups to post their recommended book list. Such necessities are needed for any VCs to reach critical mass and become valuable to cybercitizens. Web site operators continuously need to upgrade their hardware and software, which are becoming a cost burden to many VCs.

Outsourcing provides cost savings by not having to hire additional in-house technical expertise, lowering software acquisition and maintenance cost, and not having to invest in technological infrastructure. However, additional costs come from rental or lease agreement paid to outsourcers. On the other hand, the increasing demand on new and creative applications to support cybercitizens requires higher computation power and increased reliability, storage, security, dynamic adaptability, flexibility, and interoperability. Essentially, the ultimate goals of VCs are to create high-quality services through various service differentiation strategies, such as the ability to offer specialized services that focus on a very well-defined target population. This requires a significant investment to augment the existing technological resources. This article suggests a *p*-split mechanism that will help VCs increase their service performance through resource distribution and virtual process customization, thus reducing their investment cost.

To better understand the idea conceptually, Figure 1 shows typical servers used to support various VCs based

Copyright © 2006, Idea Group Inc., distributing in print or electronic forms without written permission of IGI is prohibited.

Figure 1. Virtual communities servers' arrangement

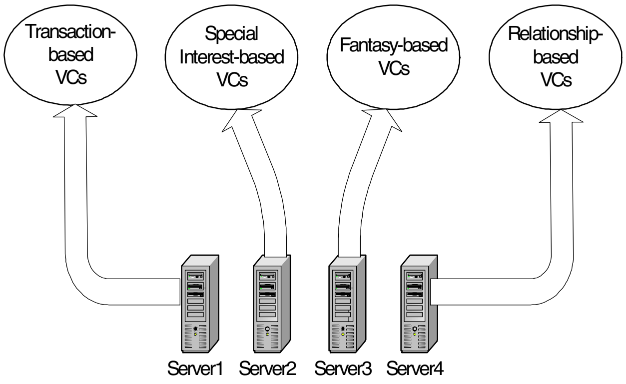
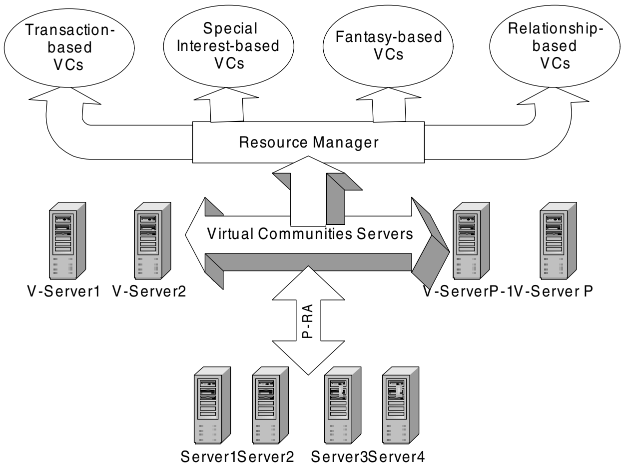


Figure 2. Virtual communities servers created through the p-split mechanism



on the four types of VCs of Armstrong and Hagel (1996). If this were a single site, a Web site operator would occasionally need to upgrade the existing infrastructures when the communities expand, and this would increase additional operating costs. However, with the *p*-split mechanism shown in Figure 2, the Web site operator only needs to optimize the existing server into *p* virtual servers at the software level. The benefits of this single-to-virtual server scalability can be realized through the use of the *p*-split mechanism. In the following section, we describe the technical concepts behind this mechanism and suggest a practical example that can be applied in real situations.

THE *P*-SPLIT MECHANISM

The basic principle of the *p*-split is to divide every resource of a physical node into *p* equivalent slices, grouping one slice from each resource to form a portion of resources and use each portion as a virtual node. In other words, *p*-split is a virtualization mechanism that splits a single server into a number of virtual servers. Theoretically the value of *p* can be any positive integer to split as many virtual servers as one needs; the reality is that the more virtual servers, the higher the system overhead and the heavier the network traffic. The increased overhead and traffic mainly come from increased virtual server process switching, resource management, and inter-/intraprocess communication. However, with a proper choice of *p*, one can easily generate a large number of virtual servers and use them as groups of VCs segments.

Provided that the resources of a server can be evenly divided into *p* equivalent portions and each portion can be used to replicate the function of a physical server, we can generate *p* smaller subservers. Assuming that the dividing mechanism generates negligible overhead then each subserver becomes a *p*-scaled–down server of the original one. Thus, we can easily create a larger number of scaled-down servers and each can be used for a different VC's purpose. This provides flexibility for better utilization of the server resources.

Figure 3 shows a conceptual diagram of a server before and after it has been divided. Let R_n denote the total resources (CPU cycles, main storage, network capacity, peripheral devices, and system control software) of server N. B_n denotes the backbone network capacity (bandwidth and latency) that connects N to other external servers. For example, by choosing a divider $n = 10$, we can create a one-order magnitude scaled-down parallel system of 10 N servers connected by a network with $10 \times B_n$ capacity.

Dividing a server requires all system resources to be divisible. Since resources of a server are integral entities of that system, physically dividing them is impossible. Our approach is to logically separate each resource into *p* functional equivalent slices using mechanisms such as space and time multiplexing. For example, the space multiplexing is used to separate both main storage (physical memory) and secondary storage (disk), and the time multiplexing is used to share the CPU cycles. The network bandwidth can be shared by time multiplexing, and the network capacity can be divided by both time and space multiplexing.

Figure 4 shows that a server can be homogeneously scaled down into *p* virtual servers, all of which have the same computation capacity as the physical server (all units are composed by one slice from each resource). Depending on the need of various virtual communities,

3 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/extending-online-communities-through-virtual/18073

Related Content

The Effect of Experience-Based Tangible User Interface on Cognitive Load in Design Education
Zahid Islam (2020). *International Journal of Virtual and Augmented Reality (pp. 1-13).*
www.irma-international.org/article/the-effect-of-experience-based-tangible-user-interface-on-cognitive-load-in-design-education/283062

An Interactive Space as a Creature: Mechanisms of Agency Attribution and Autotelic Experience
Ulysses Bernardet, Jaume Subirats Aleixandriand Paul F.M.J. Verschure (2017). *International Journal of Virtual and Augmented Reality (pp. 1-15).*
www.irma-international.org/article/an-interactive-space-as-a-creature/169931

Gendered Experiences of Mobile Gaming and Augmented Reality: Engagement with Pokémon Go among University Students
William Goette, Julie A. Delelloand Rochell R. McWhorter (2019). *International Journal of Virtual and Augmented Reality (pp. 54-67).*
www.irma-international.org/article/gendered-experiences-of-mobile-gaming-and-augmented-reality/239898

Agent-Based Network Infrastructure for E-Communities
M. Mari, A. Poggiand M. Tomaiuolo (2011). *Virtual Communities: Concepts, Methodologies, Tools and Applications (pp. 428-438).*
www.irma-international.org/chapter/agent-based-network-infrastructure-communities/48683

Metaverse: Transforming the User Experience in the Gaming and Entertainment Industry
Pooja Swami (2024). *Research, Innovation, and Industry Impacts of the Metaverse (pp. 115-128).*
www.irma-international.org/chapter/metaverse/349151