Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 81 (2008) 1591–1608
The Journal of Systems and Software
www.elsevier.com/locate/jss

Virtualization-based autonomic resource management for multi-tier Web applications in shared data center

Xiaoying Wang [a], Zhihui Du [a,*], Yinong Chen [b], Sanli Li [a]

[a] Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[b] Computer Science and Engineering Department, Arizona State University, Tempe, AZ 85287, USA

Received 9 July 2007; received in revised form 20 November 2007; accepted 21 November 2007
Available online 9 January 2008

Abstract

As large data centers emerge, which host multiple Web applications, it is critical to isolate different application environments for security reasons and to provision shared resources effectively and efficiently to meet different service quality targets at minimum operational cost. To address this problem, we developed a novel architecture of resource management framework for multi-tier applications based on virtualization mechanisms. Key techniques presented in this paper include (1) establishment of the analytic performance model which employs probabilistic analysis and overload management to deal with non-equilibrium states; (2) a general formulation of the resource management problem which can be solved by incorporating both deterministic and stochastic optimizing algorithms; (3) deployment of virtual servers to partition resource at a much finer level; and (4) investigation of the impact of the failure rate to examine the effect of application isolation. Simulation experiments comparing three resource allocation schemes demonstrate the advantage of our dynamic approach in providing differentiated service qualities, preserving QoS levels in failure scenarios and also improving the overall performance while reducing the resource usage cost.

Keywords: Autonomic computing; Resource allocation; Multi-tier Web application; Performance modeling

1. Introduction

Recent years, large data centers have received significant attentions, as they usually host many third party Web applications providing comprehensive services. In such systems, there is a critical need to provide quality-of-service (QoS) performance guarantees for each class of differentiated services. The QoS of the hosted applications plays a crucial role in attracting and retaining customers, directly impacting on providers' profits. Hence, the service providers guarantee a certain level of QoS for each application. In return, the clients agree to pay the service provider based on the specified level of QoS. Such issues of QoS requirements are often negotiated based on service level agreements (SLAs), in which the bound of anticipated performance level and the cost model involving both revenue and penalty will be clearly defined. Thus, SLA acts as a bridge between service providers and their clients, because the user satisfaction levels of the service quality experienced by them have a direct relationship with the profits of service providers. As a result, maximizing the SLA-based profits implicitly indicates that the QoS provided to a client would be optimized simultaneously. However, one main issue of preserving the QoS is the high variability of the workload, which makes it difficult to estimate the resource requirement in advance. Planning for the worst case scenario is either infeasible or inefficient. In order to efficiently utilize resources while satisfying the SLA under fluctuating workload and unpredictable failures, adaptive self-managing techniques are required to dynamically assign resources among applications of different clients on the base of

* Corresponding author. Tel.: +86 10 62782530; fax: +86 10 62771138.
E-mail addresses: wangxy@tirc.cs.tsinghua.edu.cn (X. Wang), duzh@tsinghua.edu.cn (Z. Du), yinong@asu.edu (Y. Chen), lsl-dcs@mail.tsinghua.edu.cn (S. Li).

0164-1212/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2007.11.719

short-term demand estimates. Since typical Internet applications employ a multi-tier architecture, with each tier providing certain functionality, this paper focuses on the design of autonomic resource management framework in multi-tier environments.

Nevertheless, there are many open challenges for a large data center to achieve adaptive self-management, such as application isolation, system administration, and high availability. To isolate different applications provided for independent organizations, server nodes should be dedicated for a certain application environment. Consequently, nodes may have to be shifted from one environment to another when adaptive reactions are taken according to the varying load, which needs exhausting provisioning and reconfiguring work (Diep et al., 2005). If it takes too long to complete provisioning before the node can work normally, the adaptation actions would not be able to catch the quick variation in workload, which leads to low efficiency. Fortunately, since virtualization techniques have been proposed as a solution for maintaining security and reliability in data centers (Banga et al., 1999), the fulfillment of resource multiplexing was greatly enhanced. Since hardware dependency can be broken, two virtual machines are better isolated than two services deployed in the same operating system, reducing the probability of a single error casting as multiple errors. Moreover, virtualization mechanism helps to partition resources into small slices, thus driving the resource sharing to a much finer level.

In this paper, we present the system in which virtualization mechanisms are employed for autonomic resource management of a large data center hosting different multi-tier services. First, the architecture of the service platform is proposed, in which heterogeneous physical nodes are divided into groups and shared by separate application environments. A virtualization-based self-management framework is presented to facilitate the autonomic features of the architecture. Then, we formulate a non-linear constrained optimization problem, in which fine-grained resource partitioning and multi-tier co-allocation are considered. We augment traditional model-based techniques with probabilistic analysis and overload management, and then validate the model we have established. An optimizing algorithm incorporating both deterministic and stochastic methods is adopted to solve the problem. Finally, performance evaluation results based on simulation experiments demonstrate the feasibility and efficiency of our QoS management approach that maximizes providers' profits associated with reduction of total operational costs, and also the impact of different failure rates on the system performance.

The main contributions of this paper include (1) a novel architecture of autonomic resource management framework with the employment of virtual servers to enhance various self-managing features, (2) establishment of an analytic probabilistic model which is also able to deal with non-equilibrium states, (3) a general formulation of the resource management problem which is later verified and further solved by incorporating both deterministic and stochastic optimizing algorithms, (4) fine-grained resource allocation which not only provides differentiated service qualities but also improves the overall performance and reducing the resource usage cost, and (5) investigation of the impact of failure rates on performance, which shows the advantages of the valuable features provided by virtualization mechanisms.

The remainder of this paper is structured as follows. In Section 2, we present the design of the hosting platform infrastructure and the autonomic self-management framework. In Section 3, the problem of autonomic resource provisioning is formulated. In Section 4, we establish the analytic performance model used to solve the target problem. Then, several resource allocation approaches are stated in Section 5. Section 6 demonstrates the results of performance evaluation experiments. In Section 7, we review related work in relevant areas. Concluding remarks and discussion about future work are given in Section 8.

2. System design

This section first presents the architecture of the hosting platform required in our work. Then, we design an autonomic self-management framework on such an infrastructure and discuss the autonomic features facilitated by virtualization techniques.

2.1. Architecture overview

The architecture of a shared data center is shown in Fig. 1, which is comprised of heterogeneous physical nodes,

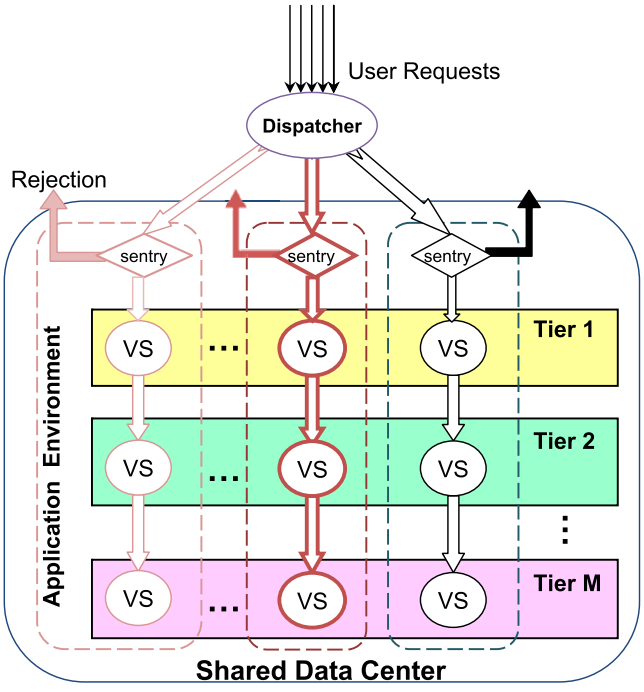

Fig. 1. Data center architecture.