Monitoring Enterprise Applications and the Future of Self-Healing Applications

Shuchih Ernest Chang, National Chung Hsing University, Taiwan
Boris Minkin, National Chung Hsing University, Taiwan

ABSTRACT

With the drastic growth of the Internet and the advance of hardware and software technologies, the enterprise information systems supporting business operations and functions have become more and more complex. The need of monitoring the behavior of such systems is becoming apparent, since it allows detecting problems early and resolving them before they become fatal and affect business seriously. In addition to covering the concept and related technologies of various monitoring approaches and their corresponding advantages and disadvantages, this article illustrates how self-healing application monitoring can facilitate the performance and availability management of Java based enterprise applications. The creating of enterprise strength monitoring solutions, together with the criteria of monitoring technology adoption and vendor selection, is also presented in this article.

Keywords: application monitoring; information system management; multi-tier system; performance and usage measurement; quality of services (QoS); system availability

INTRODUCTION

Application monitoring is essential for observing and improving the performance and availability of enterprise applications which are usually large-scale, distributed, multi-tiered and complicated. In general, the monitoring process should be carried out with appropriate techniques and mechanism to examine and control the applications for serving the needs of businesses. In terms of business needs, people’s life and living styles have been, in recent years, deeply influenced by the Internet and the World Wide Web, which enable electronic commerce (EC) for companies and their business partners to conduct business and perform electronic transactions. In addition to the purchase of products and services over the Internet, EC also encompasses all electronically conducted business activities, operations, and transaction processing within and cross companies. Through EC, companies can alleviate constraints (upon time, space and cost) to
enhance the way they connect to and interact with their EC counterparts by serving customers and collaborating with business partners electronically and intelligently. However, to catch this revolutionary opportunity offered by EC, enterprises are facing complex challenges. “Cost-effectively protecting the availability of mission-critical applications has as much to do with business planning than with technical capability,” and “A solid decision-making process must be followed to remove the fear factor from the equation”, stated by Meta Group (Garry, 2004), may serve as examples of such highly important challenges in ensuring a responsive management of EC applications.

Availability is defined as a function of mean time between failures (MTBF) and mean time to repair (MTTR) (Zimmerman, Yuhanna, Heffner, Schreck, Rankine, & Garbani, 2004). It is calculated much like a probability of failure, using the following formula:

\[
\text{Availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}
\]

In an IT system, the application availability is the result of the aggregation of all the availability factors of all architectural components supporting the application. This aggregation differs whether the components are serial or parallel, and it is a commonly agreed goal that single points of failure (in terms of considering serial components) tend to be avoided, and critical components are doubled or clustered (in terms of building parallel components). The Information Technology Infrastructure Library (ITIL), which is the most widely accepted approach to IT service management in the world, provides a set of service management best practices which include availability management (Behr, Kim, & Spafford, 2005). According to ITIL, availability management ensures that users can use IT services when they need them, at the level agreed to in the Service Level Agreement (SLA), by managing factors such as the reliability, complexity, serviceability, and maintainability of software, hardware, contracts and procedures.

The definition of what highly available means with respect to a particular application is crucial in determining what problem is really being solved. If the application is performing poorly, it might be considered unavailable by its users. If mobile users cannot connect because the phone organization’s cable is accidentally cut, the application is unavailable. If a table in the database is currently off-line, the entire application could be unavailable. From the users’ perspective, the answer may be different depending on when they connect, how they connect, and what they access. Therefore, every user has a different definition of “highly available”. In general, the user requirements for availability are defined in the SLAs, the agreements between business and IT departments on the major characteristics that application shall provide. Good SLAs should address availability and stability issues, as well as security, performance and others. The industry measures availability in nines (e.g., 99%, 99.99%, 99.999%), representing the percentage of time the application is deemed available throughout an entire year. Vendors use terms such as basic availability (95%—e.g., tape backup), enhanced availability (e.g., 99.9%—RAID disks), highly available (99.999%—application data protection), and fault tolerant (>99.9999%—life and death applications).

According to a survey conducted by Contingency Planning Research (Kembel, 2000), mission critical brokerage applications have much higher costs for downtime and require higher availability and performance requirements than retail applications. A later Contingency Planning Research survey conducted in 2001 found that 18% of the participating companies reported an hourly loss of downtime between $251,000 and $1,000,000, and another 8% of the companies reported an hourly downtime damage of over one million US dollars (Eagle Rock Alliance, 2001). According to Patterson (2002), a simple way to estimate cost of downtime can be summarized as follows:

\[
\text{Estimated Average Cost of one hour of downtime} = \]

Copyright © 2008, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
11 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/article/monitoring-enterprise-applications-future-self/2140

Related Content

Study on an Information System Integration Scheme of Enterprises Based on RFID Technology with SOA
www.irma-international.org/article/study-on-an-information-system-integration-scheme-of-enterprises-based-on-rfid-technology-with-soa/190620/

Project Management in Portuguese Metalworking Industry
www.irma-international.org/chapter/project-management-in-portuguese-metalworking-industry/133090/

Software Architectures and Requirements for a Web-Based Survey System
www.irma-international.org/chapter/software-architectures-requirements-web-based/26844/

Machine Learning Techniques for Wrapper Maintenance
www.irma-international.org/chapter/machine-learning-techniques-wrapper-maintenance/30865/

Initial Adoption vs. Institutionalization of E-Procurement in Construction Firms: The Role of Government in Developing Countries
De Chun Huang, Quang Dung Tran, Thi Quynh Trang Nguyen and Sajjad Nazir (2014). *International Journal of Enterprise Information Systems* (pp. 1-21).
www.irma-international.org/article(initial-adoption-vs-institutionalization-of-e-procurement-in-construction-firms/119166/)