

# Admission Control for E-Commerce Web Sites

## (Extended Abstract)

Sameh Elnikety<sup>†</sup>, Erich Nahum<sup>‡</sup>, John Tracey<sup>‡</sup>, Willy Zwaenepoel<sup>†</sup>  
<sup>†</sup> School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland  
<sup>‡</sup> IBM Thomas J. Watson Research Center, Hawthorne, New York, USA

### ABSTRACT

Autonomic systems require self-protection. In the context of a dynamic E-commerce Web site, this includes preventing the site from crashing as a result of unpredictable load. We present a flexible, transparent method for admission control for multiply-tiered e-Commerce Web sites, achieving both stable behavior during overload and improved peak performance. Our method is embodied in an implementation, called Gatekeeper, which we evaluate using standard software components on the Linux operating system. Driving the system with the industry-standard TPC-W workload generator, we show consistent performance during overload. Peak throughput increases up to 10 percent and response time improves up to 15 percent.

### 1. OVERVIEW

The goal of self-managing or *autonomic* systems is to require as little human attention as possible. Ideally, these systems should be self-configuring, self-monitoring, self-protecting, self-repairing, and self-optimizing. They should be resilient, flexible, adaptive, transparent, and require the minimum number of “knobs” to tune in order to operate.

E-commerce is a prime candidate for an autonomic system. Online commerce is a growing phenomenon as consumers gain experience and comfort with shopping on the Internet [2]. Online merchants desire to maintain a continuous, consistent presence on the Web in order to keep customers satisfied and maximize both revenues and returns on their infrastructure. However, E-commerce sites can actually suffer from too much success, where they become overloaded. The volume of requests for content at a site may exceed the capacity for serving them, which renders the site unusable. This is frequently referred to as the “slashdot effect.” Ideally, an E-commerce site should be able to monitor itself and protect itself against such “catastrophic success.”

This paper presents a step toward the goal of an autonomic multiply-tiered E-Commerce Web sites. We present a method for providing admission control that is self-monitoring, self-protecting, and mostly self-configuring. Our approach externally measures execution costs

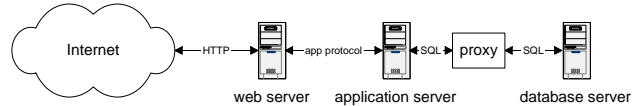


Figure 1: Gatekeeper Placement within a 3-Tiered Web Site

online, differentiating between different types of requests, enabling overload protection and improvements in response time. Our approach is completely resource-independent and does not discriminate between various bottlenecks (e.g., data contention or physical resource contention). Our admission control scheme *accounts for variations in service costs*. By measuring service times online, our system is more robust to overload than approaches which assume that measurements taken under light load are applicable to heavy load situations. Our approach also accounts for variation in the execution times of different types of requests; most other approaches only account for a single metric such as overall response time or queue length, or assume a simple linear model of service costs. In contrast, we track the amount of work generated by each request directly. The only configuration required is a simple binary-search process, performed at setup time, that determines the available capacity of the system, and we believe this can be automated in the future as well.

Other proposals require extensive modifications to the operating system or a complete re-write of the server, contrary to the transparency and flexibility desired in autonomic systems. Our implementation requiring no changes to the source code, server software, application programs, or to the database. The benefits of such an approach are clear: the use of unmodified commodity software components reduces development effort tremendously. As a result, we are able to demonstrate our approach using *standard software components and workload generators*.

Our method is embodied in a proxy, called *Gatekeeper*. A key feature is that it is *transparent to the database and application server*. For our evaluation, we use a standard testbed environment of a multiple-tiered e-Commerce Web site, with Linux, Apache, Tomcat, MySQL. In this environment, the database is the bottleneck, and thus Gatekeeper is placed between the application server and the database server so as to transparently intercept requests from the application server to the database. Driving the system with the industry-standard TPC-W benchmark, Gatekeeper achieves both stable behavior during overload and improved throughput and response time.

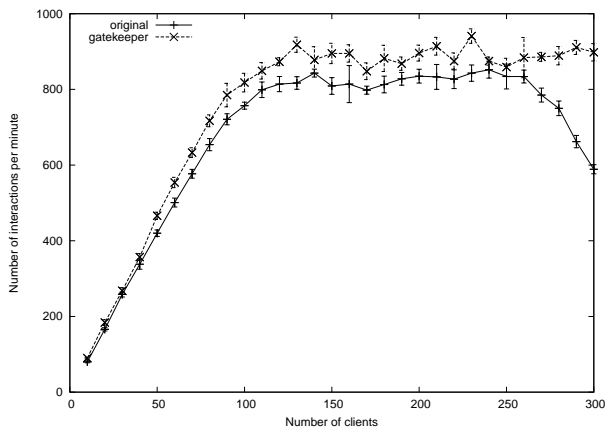


Figure 2: Throughput (MySQL, Locking in App. Server)

## 2. SAMPLE RESULTS

In this section, we present a brief sample of our results. For full details and results, please see our tech report [1].

Figure 2 shows the throughput of the system, when locking is done in the application server. The X-axis is the number of emulated clients, and the Y-axis is throughput in interactions per minute. Two curves are presented: the original system without admission control, marked “original”, and the system using the Gatekeeper proxy, marked “gatekeeper”. The original system reaches a maximum throughput of about 852 interactions per minute, but then starts degrading above 250 clients, falling to 589 interactions per minute at 300 clients. The system using the Gatekeeper proxy maintains a consistent throughput even at the higher loads, and exhibits a higher peak throughput than the original system, of 941 interactions per minute at 230 clients.

Note that in this case, when admission control is used, peak throughput is increased roughly 10 percent. To understand why, we ran a number of profiling experiments using the performance counters for the Athlon CPU. Space constraints preclude showing the data, but two main factors appear to be the cause of the difference: a reduction in the number of database processes on the system and better memory cache behavior.

One question that might occur is whether our results are dependent on artifacts of a particular implementation, in this case MySQL. To test this notion, we ran our experiments using DB2 instead of MySQL. Again, space constraints do not permit showing the data, but we observed similar behavior to that seen in Figure 2.

Figure 3 shows the throughput of the system when locking is performed by the database. Note that the scales of this graph are different than in Figure 2; the peak throughput here is much lower than in the previous graph. This shows how locking in the database is more expensive than locking in the application server. In this graph, the two curves show similar performance up to about 70 clients, after which the original system degrades quickly, but the experiments using Gatekeeper again demonstrate consistent performance even during overload. The original system reaches the peak throughput of 515 interactions per minute at 70 clients. At this point, the CPU utilization is 70 percent and all other system resources are far less utilized. As the load increases beyond this point, contention for the

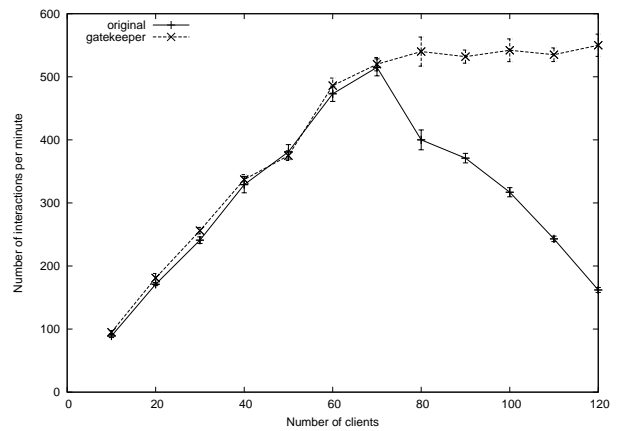


Figure 3: Throughput (MySQL, Locking in Database)

data locks causes thrashing, as two phase locking is used and many database queries become blocked.

## 3. SUMMARY AND CONCLUSIONS

Autonomic E-commerce Web sites require self-monitoring and self-protection. This work presents a method for providing admission control for multiply-tiered e-Commerce Web sites that is resilient to overload and dynamically adjusts for variations in load. By externally measuring service costs online and distinguishing between different types of requests, our approach can achieve both stable behavior during overload and improved response times. Other proposals require extensive modifications to the operating system or a complete re-write of the server. Our approach requires no changes to the source code, server software, application programs, or to the database. This allows ease of deployment, database independence, and use of standard software components. Our work is thus the first to evaluate admission control for multiply-tiered e-Commerce Web sites.

For full self-protection, each stage of a multiple-tiered Web site will require admission control. A natural next step for future work is to evaluate our approach using other dynamic workloads. In cases where the bottleneck resource is the application server, rather than the database, it likely makes sense to place the Gatekeeper proxy between the front-end Web server and the application server.

Reducing the amount of configuration is also a logical next step. While our servlet execution times are currently estimated automatically using a simple moving average, the process for estimating the overall capacity of the database server can be automated and performed adaptively and on-line. This would greatly simplify deployment and manageability issues for the proxy, and move us closer to the goal of a fully autonomic E-commerce Web site.

## 4. REFERENCES

- [1] Sameh Elnikety, Erich Nahum, John Tracey, and Willy Zwaenepoel. A method for transparent admission control and request scheduling in dynamic e-commerce web sites. Technical report, IBM Research Report, Hawthorne, NY, Mar 2003.
- [2] News.Com. E-commerce strong in third quarter. <http://news.com.com/2100-1017-971123.html>, November 2002.