Web Performance Enhancement with Solid State Disk

By Dr. Mitchell Wyle, Co-Founder, Euphorian Inc.

Dr. Wyle is not associated with Texas Memory Systems, Inc. He has given us permission to post his white paper on solid state disks.

Abstract

Certain Web-enabled Internet and Intranet systems can derive great benefits from the use of solid state disk (SSD). The application of SSD provides beneficial side-effects including reliability, flexibility and the scalability needed for extremely high web transaction volume. This document describes some of the benefits of SSD to enhance web application performance.

Contents

Introduction

Web technology is the foundation of a wonderful communications medium. The web's growth on the public Internet and also in private Intranets is a testimony to it's power. Because of the web's popularity many web-based applications run into performance bottlenecks that drastically decrease the throughput and therefore the usability of the **content** (transactions or other information) delivered over the web. The Wild Wild Web can become the World Wide Wait!

Some web applications require very intensive disk access. All web applications must -- at least initially-- load their data from disk and log all of their transactions to disk. Regardless of how aggressively the web subsystems try to cache content in RAM, some number of cache misses and cache stalls always occur. Tuning a complex multi-tiered cache can be a difficult, on-going configuration and maintenance headache.

These and other disk-related web system delay problems can be overcome simply by adding solid state disk (SSD) hardware below the operating system (OS) and web software.
Different types of web applications can suffer from disk performance bottlenecks. Applications based on commercial database software are usually the biggest causes of excessive disk input/output (I/O). CPU-intensive systems, e.g. systems that exclusively use common gateway interface (CGI) also tend to write to disk a lot and may suffer from slow disk speeds. Interestingly enough, even content-only web sites, i.e. sites that serve only hypertext mark-up language (HTML) files, can gain a significant performance boost by using SSD in addition to aggressive, multi-tiered caching.

Performance Bottlenecks

It is important to note that most web throughput bottlenecks are **not** caused by disk I/O. The most prevalent and most commonly overlooked bottleneck in high-volume web sites is the operating system. There is a factor of ten (10X) difference between two popular operating systems running on the same hardware! No amount of raw CPU power or high-performance hardware can overcome such severe problems.
The first step in planning for a high-performance web site is to choose the best operating system for the need. If an existing application is suffering because of sudden growth in web transaction volume, the first action to take for performance enhancements is by tuning the OS. The most popular commercial and free-ware web daemon vendors note this fact in their installation documentation. However, a perfectly tuned OS does not by itself guarantee high performance.
Another important and sometimes hidden bottleneck for a public Internet web application's performance is the Internet itself. When designing high-volume public web applications it is critical to consider methods that overcome the unpredictable, chaotic latencies (delays) of the public Internet. The best technique for attacking this problem is *client-side caching*. Several products are emerging that pull web content non-interactively, that is, with no end-user direct manipulation. These products pull web content to an end-user's personal disk cache during the wee hours of the night. Such proxy "agents" allow for fast interactive response and also for off-line document browsing. Corporate web-based white pages and magazines become accessible on disconnected notebook or palm-top computers. However all interactive web applications that rely heavily on web forms, can**not** take advantage of this technique.
Two other commonly blamed, but rarely guilty factors that limit web system performance include:

- **Network bandwidth**: Many public Internet connections go over T1 lines that can pump out 0.1544 million Bytes per second. Assume that the average "hit" on a web site retrieves a very large document like this ten thousand-byte page. The number of "hits per day" a T1 line can support is then:

```
1,544,000 bit   1 byte     1 "hit"     60 second   60 minute   24hour
------------- * ------ * ----------- * --------- * --------- * ------
   second       10 bit   10,000 byte    1 minute     1 hr       1 day
```

```
= 1,334,080, over 1.3 million hits per day
```

This analysis assumes the bandwidth required for client requests is insignificant. Large corporate networks have aggregate bandwidth in the trillions of bits per second. A single Ethernet segment within such a network (ten million bits per second) can support roughly ten times the network traffic, or **well over over 8.6 million hits per day**.

- **CPU performance:** Most modern systems can flood the entire bandwidth of one Ethernet segment. In fact, they can usually stuff two or three such ether wires full of bytes. Poor web application design does cause some performance degradation. However, all web daemon vendors offer methods for tuning an application's CPU resource utilization to overcome these difficulties. A **good web daemon on strong hardware should be able to serve 10 to 70 million hits per day**.

Disk Bottlenecks

After tuning other system parameters to eliminate common bottlenecks in a web application, eventually the seek times of disk become the key problem causing the longest delays. At that point the usual reaction is to add more cache either as a "write-back" on the disk system or as virtual memory for the operating system. These approaches are non-optimal.

Solid State Disk Benefits

Solid state disk is a better solution. In cache systems there will always be cache misses. At that point the factor of 100 difference between DRAM (electronic) data access and mechanical movement for data access favors SSD.
The total cost of analysis and cache tuning in human labor alone usually exceeds the small cost difference between processor DRAM and SSD.
SSD is a safer storage medium than rotating disks because there are no moving parts to fail. SSD has **full data protection** including built-in error detection, correction, and battery backup.
SSD is most often applied in "pure" production database systems because these systems have very intense I/O requirements. With the explosion of the public Internet, some of these applications are adding web front-ends which will dramatically increase their moderate-volume transaction loads up into transaction volumes requiring the application

of SSD. Systems on the public Internet with web transaction volumes in the dozens of millions of hits per day can definitely make good use of SSD.
SSD is so much faster than normal disk that it can be shared by up to four different machines. These machines can be configured to share the web transaction load but instead of synchronizing their slow disk subsystems, the SSD can be used as a common data access system among them. Furthermore, if one or more of the load sharing machines should fail, the data on the SSD are still accessible to the other systems. Thus the SSD system can facilitate load balancing and fail-over capabilities when reliability is an important objective.

Other Benefits

High Reliability

SSD, with its built-in error detection and correction, full data protection (battery backup), and no moving parts has a higher mean time between failures (MTBF) than any disk subsystem.

Scalability

Access to data on SSD is more than two orders of magnitude faster than accessing data on rotating disks. It's use in web applications will, therefore, pro-actively prevent secondary storage access from becoming the performance bottleneck.

Flexibility

SSD "pretends" it is a disk to the machine accessing it. This transparent architecture enables the use of SSD only in the troublesome "hot spots" of very high disk activity. As a web application scales up to meet growing demand, more and more disk activity will start to bog down performance. SSD can swiftly be applied as needed. Dropping in SSD is just as simple as installing another disk.

Conclusion

SSD is a flexible, simple-to-install, low-maintenance performance and reliability enhancement option for use in high-volume Internet and Intranet applications. SSD provides for full data protection and enables simple architectures for CPU load balancing and fail-over. We have analyzed and explored some of the true --as opposed to perceived-- bottlenecks in Web application performance. When write-transaction volume becomes high enough, SSD can greatly enhance performance.