primaryio

Technology Brief

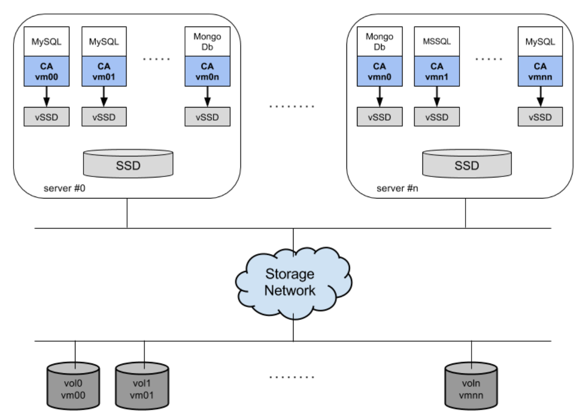
8/01/2015

Introduction

Flash technology, SSDs in particular, have found their way into the modern datacenter as a replacement for slower, traditional hard disks (HDD). SSDs are now found deployed within direct attached storage (DAS), network attached storage (NAS) or a storage area network (SAN). Cloud service providers such as AWS deploy flash as well in order to comply with SLAs for high application performance and low latency. Clearly, almost all public and private infrastructures today offer some form of flash based storage technology.

However, the blazingly fast IO performance of flash often comes at a significant price tag. Although it's true that flash densities and costs continue to improve, they are still much higher than installing traditional hard disk solutions. One approach to leveraging flash technology is to install a new hybrid or all-flash storage array, but this can be very disruptive to a business as applications are taken off-line during the upgrade. For all these reasons, leveraging SSDs as server-level cache is now considered a more cost-effective and far less disruptive deployment alternative.

An as illustrated in Figure 1, server-side SSD caching harnesses flash in the form of a direct server attached SSD device, which brings many advantages due to its closer proximity to the application. However, almost all the SSD caching software available today is limited to generic LUN based performance acceleration. These caching solutions have no intelligence about application's I/O behavior patterns or the business needs driving the application I/O requests. The main limitation of generic LUN acceleration solutions is that they require very large cache capacities in order to be effective. However, this makes them sub-optimal and inefficient for enterprise use, beyond the very basic and simplest of deployments. Applications, and the admins who manage them, are in the best position to provide insights on what they need to accelerate performance, since they understand data beyond just blocks and LUNs. For example databases like Oracle and SQL Server prioritize RAM usage for storing of index blocks over say log or other miscellaneous data.



Block diagram of servers and storage in a datacenter. Servers are virtualized via VmWare or Xen. SSDs are direct attached to the physical server. The SSD is split into smaller volumes (vSSD) and assigned to individual Virtual Machines. Persistent VM storage (vol0 …) is delivered via the network using iSCSI.

Figure 1: Direct attached flash storage architecture in a typical datacenter

Application Performance Acceleration

PrimaryIO's Application Performance Acceleration (APA) technology is the next generation of server-side SSD caching software that is built to precisely and optimally accelerate applications deployed in enterprise datacenters. This unique APA caching technology manages flash much more efficiently and delivers significantly improved levels of application I/O performance over traditional solutions. The software integrates closely with enterprise applications whether they are installed on bare-metal servers or within a virtual machine. Yet, no changes are required to the application or the underlying storage infrastructure. The PrimaryIO APA technology also provides improved SSD cache utilization and higher virtual machine densities both of which are important goals for an enterprise datacenter.

The PrimaryIO APA software leverages intelligent algorithms to optimally intercept and cache a copy of a database application's working data set based on its relative value and ability to accelerate workload performance. Only the most important application components such as frequently accessed tables or indexes that speed up queries are optimized, while less critical elements such as log records, replicas, audit entries, or ad hoc user activity are de-prioritized. Application administrators in particular benefit from easier management since they do not have to become system experts. Rather, they can optimize performance based on familiar application components without the risk of adversely impacting user access or storage operations.

PrimaryIO APA Technology – A Deeper Dive

The PrimaryIO APA™ software is very effective at filtering out I/O streams that are not relevant to core application performance. To better understand the power of the PrimaryIO APA technology, let's examine an application deployment scenario involving the MySQL database and an Online Transaction Processing (OLTP) workload. As illustrated in Figure 2, MySQL databases dbA and dbB are replicated across the network to a slave MySQL server. The key point to note here is the additional I/O that is incurred by the master server due to replication and backup storage functions. Although these are important functions to the enterprise, they also have a direct and negative impact on the transaction performance of the MySQL databases.

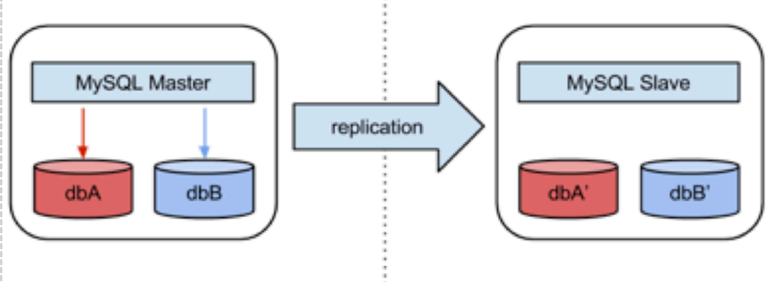


Figure 2.

Traditional LUN caching solutions do not have the intelligence to see beyond I/O activity and blocks of data so they cannot differentiate between a primary data request that is most important such as databases transactions stored in tables from a secondary request that is less important and often generated as a consequence of primary data or other unrelated activities within the system such as log records or in this case a replication process. (see Figure 3). Generic LUN caching solutions rely solely on temporal reference to make caching decisions, which results in inefficiencies and lower application performance. The PrimaryIO APA technology overcomes this limitation through its ability to distinguish between different application I/O streams.

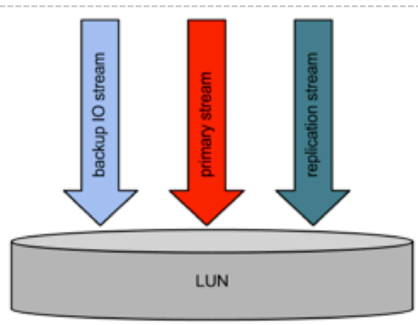


Figure 3.

PrimaryIO APA Deployment Process

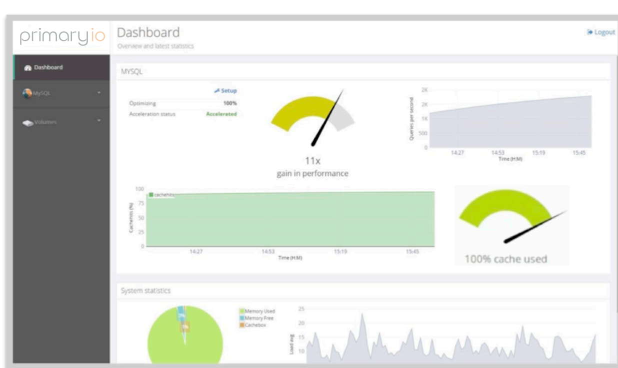
The installation process for the PrimaryIO APA software is very simple and requires only very basic administration skills that most application and system owners already possess. To better understand the deployment process, we've summarized the key steps into three phases.

Step One: Pre-Installation
This step involves running the integrated installer/analyzer on the server that will be optimized. During this phase ongoing I/O activity is monitored and the PrimaryIO APA software determines whether the application workload can be optimized and how much flash capacity is required for optimal performance. This process does not require installation of any core caching components and can be run safely without any concerns of affecting running applications or system operations.

Step Two: Installation / Configuration
The installer automatically determines which operating systems and database applications are running on the server and has the intelligence to install the necessary components required for PrimaryIO APA. Once installed, the configuration wizard proceeds to guide the admin through a set of simple steps to configure the application for optimization.

Step Three: Optimization
Once configured, the application components are further analyzed and recommendations are presented to the administrator to select appropriate components for optimization. These components are typically database objects like indexes and tables that are presented in a familiar format to the application owner. The admin can then choose to apply the PrimaryIO APA recommendations or override it with custom optimizations.

The entire optimization process is easy and intuitive and can be completed with just a few clicks of a mouse. The PrimaryIO APA software has the intelligence to automatically discover and identify all storage devices attached to the server. As illustrated below, the admin is presented with an intuitive web-based GUI dashboard that provides many important insights and updates about system and application performance such as TPS and latency improvements. For advanced users, detailed drill down of storage and system health is also available.



Performance Summary

In an effort to quantify the benefits of PrimaryIO APA technology, we performed a series of tests on the MySQL database in PrimaryIO's Performance Lab across a variety of application workloads. We compared the performance of pure hard disk, generic optimization using Flashcache open source caching software and PrimaryIO's APA technology.

The results in Table 1 below clearly show that PrimaryIO APA technology is up to 2 times better than competing offerings, and up to 19x better than hard disk based non-caching setups.

Table 1.	Pure HDD	Generic Optimization	PrimaryIO APA	% Improvement
MySQL[1]	11 TPS	84 TPS	147 TPS	1,236%
SQL Server	41 TPS	70 TPS	222 TPS	441%
Web Server[2]		2,181,245 page fetches	3,571,240 page fetches	63%
mongoDB[3]	251 OPS	4,638 OPS	5,111 OPS	1,936%, @ 50% cost savings per mo.
VDI[4]	49 minutes	18 minutes	12 minutes	308%

[1] OLTP MySQL database running in a KVM virtual machine.
[2] Bare-metal web server configured with WordPress web portal behind an apache2 server.
[3] OLAP use case on Amazon cloud comparing PrimaryIO APA powered AWS cluster vs Mongo 4K AMI in run phase of YCSB
[4] VDI use case boot time comparison for 100 thin clients.

Conclusion

The PrimaryIO selective acceleration policy avoids the noise generated due to log updates, replication while efficiently utilizing server-side cache to accelerate only the most important components. This approach improves performance for the accelerated database table but also reduces the load on traditional hard disk that positively impacts the performance of other database tables as well.

Enterprises benefit beyond just performance acceleration for database applications. The PrimaryIO APA technology dynamically adjusts memory utilization as per the system memory available rather than the boundaries of the SSD disk size, making the best use of system memory without compromising application performance. CPU consumption is maintained within an acceptable limit to ensure no adverse impact on production servers. The PrimaryIO solution also works to avoid unnecessary I/Os on the SSD drives thus increasing their useful life.