Table of Contents

Introduction

The Challenge
  Read Workload Scaling .................................................................1
  Write Workload Scaling ...............................................................2

The Solution
  HGST FlashMAX PCIe SSD ..........................................................3
  Addressing Read Workloads Using FlashMAX PCIe SSDs – Minimize
    Slave Servers .................................................................4
  Addressing Write-loads Using FlashMAX PCIe SSDs – Scale
    Without Sharding ..........................................................5

Data Layout and Configuration Parameters, MySQL Version
  Locating I/O Intensive Files on the FlashMAX PCIe SSD ..................6
  MySQL Server Options ...............................................................7
  New Architecture Using HGST FlashMAX PCIe SSDs ......................7

HGST FlashMAX® PCIe SSD: Solving the MySQL Scaling Challenge
  Scaling MySQL Cost-effectively with HGST FlashMAX PCIe SSDs .......8
Introduction

Data continues to grow at an exponential rate across organizations of all sizes. This growth is fueled by the popularity of portal, search, media, and e-commerce sites such as Amazon, Yahoo and eBay, as well as the exponential growth of social networking sites such as Facebook and Twitter. In the enterprise, data is growing as companies introduce new services to customers and various stakeholders. Not only is the amount of data growing, but as organizations strive to grow more responsive, the way data is accessed is also changing. Data analysis, for example, has transitioned from a traditional, batch mode, reporting-style to an ad-hoc, on-demand real-time access. Both of these trends – growing volume of data and the growing need for random access to this data by applications such as MySQL – are exposing shortcomings in traditional disk drive-based storage infrastructures that are no longer able to keep up with the growing performance demands. Consequently, IT staff are being challenged with scaling MySQL database and infrastructure to handle this exponential increase in data and traffic in a cost-effective manner. Traditional approaches take time and resources while increasing complexity and driving up the costs in hardware, data center footprint and power.

Companies have traditionally tried to address these problems either using operational techniques such as sharding or throwing more hardware, such as DRAM or servers, into their IT infrastructure, with limited success. This paper focuses on how to solve these scaling and data management issues cost-effectively and easily by using Flash-based PCIe solid-state drives (SSDs) from HGST. HGST’s FlashMAX® PCIe SSDs deliver high predictable performance as well as enterprise-class reliability. It utilizes the industry-standard PCI Express interface and employs an innovative hardware and software architecture for high performance and sustained random IOPS. Delivering up to 15X faster MySQL performance versus traditional hard drive solutions means that up to 15 MySQL servers can be replaced with a single server comprised of FlashMAX SSDs, with the following benefits:

- Lower power consumption – no additional power required with no impact on OpEx
- Decreased space in racks (and payments for data center) – reduced CapEx
- Reduced architectural complexity – reduced support and administrative costs

The Challenge

**Scaling MySQL cost-efficiently to handle exponential growth in data and traffic**

Web 2.0 companies and enterprises alike have adopted MySQL as the database platform of choice to manage their data. While MySQL and the associated infrastructure is easy to manage and deploy when the dataset sizes are small, scaling the database and infrastructure to handle the growing datasets create demands on the database that pose significant challenges. Traditionally, IT departments have adopted a multi-pronged strategy to address this problem. Typically, they separate out the read and write traffic on the data by creating a master-slave configuration. Masters absorb the write traffic, while the majority of the read traffic is directed to the slaves.

**Read Workload Scaling**

Scaling of the read workload is addressed, initially, by using MySQL Replication, and by cloning slaves. However, as the demand on the database grows, this approach can result in a large number of slave boxes, leading to a management and operational nightmare – not only requiring maintenance of many machines but
also managing several copies of data. In addition, slaves require application modification to accommodate for the asynchronous MySQL Replication. More importantly, customers very often find that this approach does not scale very well as the associated write workload grows. The number of slaves that can be added to address the read workload is, in fact, limited by the number of writes/updates that have to be replicated across all the slaves and the single threaded nature of MySQL Replication. The net result is that scaling of read workload is not independent of the write workload.

**Write Workload Scaling**

Scaling for writes is much more complex. The following approaches are used to address write scaling:

**Scaling Up: Increase DRAM on Master Servers**

This looks like an easy fix but works only up to a certain point. As the write workload and dataset size grows, this solution becomes prohibitively expensive due to the non-linear price-to-density relationship for DRAM. Apart from the physical limitation of adding memory to a server, there is an additional problem.

As MySQL expects data to be persistent, there is a need to periodically flush the data from memory to the backing store. As a result, the steady state performance of such a solution is limited by the performance of backing store HDDs and hence performance can be poor. Another limitation of this approach is the significantly high “warm-up time”, which is the length of time between server startup and the time it can accept high loads.

**Using SAN/External Storage**

While SANs can deliver decent I/O throughput, the I/O latency still remains a problem. Latency is critical for MySQL performance especially for routine management operations such as replication, long queries and batch jobs. Many of these operations are run serially, hence their response time and performance is directly impacted by the large latencies of a SAN device. Also, SAN-based solutions are expensive and complex to manage.

**Sharding**

Scaling by changing application design to include sharding and partitions is another option. But this is never an easy solution and requires complex data management along with changes to applications. The architectural design should be done in advance and often it leads to limitations on applications (e.g. it becomes difficult/impractical to execute some types of queries because data is placed in different locations – complex queries such as joins, etc.). The data gets married to one form of architecture and a subsequent change to the architecture, to meet new business needs, could become very difficult.

According to Vadim Tkachenko, CTO of Percona, Inc., sharding should be last resort for scaling and is fairly complex for the following reasons:

- The application developer has to write more code to be able to handle sharding logic
- Operational issues become more difficult (backing up, adding indexes, changing schema)

Thus, scaling for read and write workloads using traditional methods leads to a more complex infrastructure deployment and cost as shown in the following diagram:
Flash-based storage, such as the SSDs, has created a paradigm shift in the way in which data is stored, managed and accessed. The read and write performance issues are instantly alleviated, delivering significant scalability to the database architecture. Most applications should see instant improvement in I/O performances.

Of the various types of SSDs, the highest performance (highest bandwidth and lowest latencies) is delivered by PCIe-based SSDs.

**HGST FlashMAX PCIe SSD**

The HGST FlashMAX PCIe SSD is the best Flash storage solution available on market, in terms of addressing performance (I/O throughput and latency) and data reliability requirements.

**Performance**

Benchmarking and customer deployments have shown that, depending on workload and active dataset, users can get up to 15X performance gain, compared to HDDs, by simply moving data from traditional storage to the HGST FlashMAX SSD. By doing so, companies can scale their MySQL infrastructure without having to add expensive DRAM, SAN or having to shard the database, while using their existing MySQL and storage engines. HGST FlashMAX SSDs can deliver over 330,000 4K read IOPS and 150,000 fully sustained mixed (70:30) 4k IOPS.
Latency
FlashMAX PCIe SSD connects directly with the CPU using PCIe; giving the best latencies (10s of us) better by orders of magnitude compared with a SAN or even SATA- or SAS-based SSDs. These low latencies, as seen earlier, are extremely critical to ensure fast response times for many MySQL operations such as replication, long queries, batch jobs etc.

Capacity
FlashMAX PCIe SSD is available in usable capacities ranging from 550GB – 4.8TB and has a low profile form factor, making it compatible with all kinds of servers.

Modular, Scalable, Reliable
The FlashMAX SSD design is modular, consisting of a base card with field replaceable Flash modules. This modularity combined with an on-board, Flash-aware RAID5, ensures high data availability at all times, even in the event of a Flash module failure. This is in addition to the ECC implemented at the Flash-level.

Compact
Low profile HGST FlashMAX PCIe SSDs can be installed into any server chassis and deliver blazing performance without the need for additional power.

Addressing Read Workloads Using FlashMAX PCIe SSDs – Minimize Slave Servers
Using FlashMAX PCIe SSDs for read workload scaling in slave servers as the primary store for MySQL data results in up to 15X improvement in transactions per second (TPS) compared to HDD arrays. Using FlashMAX SSDs in this configuration results in the following benefits:

- Consolidation of slave servers
- Allow dataset size to exceed the size of DRAM without compromising TPS

Figure 2 below shows the benefits of having FlashMAX SSDs on the slave. When the dataset size exceeds the size of DRAM and data spills out of memory, the FlashMAX drive outperforms HDD RAID by up to 15X.
The FlashMAX PCIe SSD is not only good at handling reads but is very efficient in handling write traffic as well. FlashMAX offers the most balanced read-write performance in the industry. Also, the FlashMAX SSD’s write performance advantage over competing products increases with capacity utilization. This capability enables FlashMAX SSDs to improve MySQL Replication performance. As a result, companies can now scale the slaves to address the read workload independent of the amount of write and not be bottlenecked by the single threaded nature of MySQL Replication.

Using HGST FlashMAX PCIe SSDs in the slave servers delivers the following benefits:

- Scale the slave infrastructure to handle the read workload, independent of the amount of writes to the master
- Improved response time for critical MySQL operations such as replication, long queries etc.
- Scale read traffic by 15X with the existing infrastructure – or –
- Consolidate up to 15 slaves into 1, resulting in lower IT infrastructure costs

**Addressing Write-loads Using FlashMAX PCIe SSDs – Scale Without Sharding**

The write workload scaling problem can be addressed by using FlashMAX PCIe SSDs in the MySQL master to store all of the MySQL data.

As the chart in Figure 3 below shows, a single HGST FlashMAX SSD enables the MySQL server to scale nearly 10X, thereby eliminating the need to shard the database to handle heavy write workloads.

Furthermore, the chart in Fig. 4 shows that, even when the entire dataset can fit into DRAM, using a FlashMAX SSD delivers a 5X improvement in performance and the ability of the MySQL master server to handle write workloads.

**TPCC-MySQL Benchmark FlashMAX vs RAID**

![Figure 3. Write scaling, with HGST FlashMAX PCIe SSD vs. RAID 10 HDD array (SAS drives)](image)

![Figure 4. Write scaling – Dataset fits into DRAM](image)
Data Layout and Configuration Parameters, MySQL Version

If the entire dataset fits on a single card (up to 4.8TB usable capacity), the above illustrated performance improvement can be achieved by moving all the data into the FlashMAX PCIe SSD. No additional changes are required.

If the dataset size is too large, there are three ways to layout the data.

- **Stripe**: Stripe the data across multiple FlashMAX cards on a single server.
- **Tier**: Locate the most I/O-intensive files on the FlashMAX SSD (refer to Section 3.1).
- **Cache**: Implement a Flash-memory-friendly caching solution (refer to the HGST Virident ClusterCache Solution Brief).

**Locating I/O Intensive Files on the FlashMAX PCIe SSD**

You may also consider separating the files in the following manner:

Put the transactional and binary logs on a RAID 10 SAS HDD and the entire remaining index and data files on the FlashMAX SSD. Locate transactional logs and binary logs on separate SAS HDD storage since binary logs and slow-logs can consume valuable space on the PCIe SSD. A RAID 10 HDD array with BBU would be a good choice. The same goes for all additional logs you have (error.log, slow-query.log).

Performance can be further improved by putting system tablespace (ibdata1) on separate HDD storage (as I/O patterns for this tablespace are different from I/O patterns of data and index files). As can be seen from the chart below, this additional improvement can be as much as 1.45X. This improvement is over and above all the performance improvement obtained by using FlashMAX SSDs for write intensive workloads where the size of the hot data is significantly larger than the DRAM size.

![Figure 5. Additional performance with ibdata1 on RAID and rest on FlashMAX](image-url)
MySQL Server Options
An alternate approach to improving the overall performance would be to use either MySQL ver 5.5 or the Percona Server. The Percona Server uses advanced and improved techniques resulting in higher performance than the standard MySQL Server.

New Architecture Using HGST FlashMAX PCIe SSDs
The diagram below represents the new MySQL architecture with HGST FlashMAX PCIe SSDs. The new architecture does not require any sharding of the database and uses far fewer slave servers to support a much larger database with an order of magnitude higher traffic (or TPS).

Figure 6. New Architecture Using HGST FlashMAX PCIe SSDs
HGST FlashMAX PCIe SSD: Solving the MySQL Scaling Challenge

As seen above, HGST FlashMAX PCIe SSDs not only improve the performance of MySQL databases and infrastructure over HDD arrays, they also deliver the highest sustained steady performance to the application compared to all other SSDs in the market.

Scaling MySQL Cost-effectively with HGST FlashMAX PCIe SSDs

- Up to 15X performance improvement over standard configurations
- Minimize sharding – application team can focus on introducing newer services etc. to their end users
- Fewer master and slave servers – reduces management workload on the operations team
- Freedom to scale slaves independently of the amount of write – avoid MySQL Replication bottlenecks
- Decreased warm-up time from hours/days to minutes
- Reduced time for schema changes e.g. like ALTER TABLE add column/index
- Reduced time for maintenance and operational tasks: replication, back up, recovery, slave set up etc.
- Significant reduction in CapEx and OpEx

Contact Information

Percona, LLC
2300 Benson Road S, Suite #B5
Renton, WA 98055 USA
Phone: +1-888-401-3401 Ext. 510
http://www.percona.com

HGST, a Western Digital company
3403 Yerba Buena Rd.
San Jose, CA 95135
Phone: +1-800-801-4618
http://www.hgst.com