An Oracle White Paper
April 2010

MySQL Cluster for Web and E-Commerce Applications: Growing Revenues and Enhancing Customer Loyalty

Introduction

As organizations bring more of their operations to the Web, from e-commerce storefronts to content delivery services and user collaboration, the pressure to keep systems running continuously has never been higher. Services must be delivered with the highest and most predictable levels of availability and performance, sustained even under the heaviest peak loads, to ensure a quality customer experience and to cement loyalty.

The most successful businesses built on the Web live on the principle of “failing fast, scaling fast.” This principle dictates that the operational environment will be a key differentiator in the success of service adoption and monetization over the Web. Services are required to fail over and scale transparently without any disruption to the consumer. The ability to handle explosive growth in user volume, transactions, and storage capacity is critical to delivering online services over a Web-based platform.

For many online applications such as e-commerce, information and access to it forms not only the infrastructure of the application but also the entire revenue model of the business. With the emergence of highly participative “Web 2.0” technologies, new generations of online applications are dependent on having fast and constantly available access to data. Such data enables the online experience of users to reflect, in real time, their own preferences, history, and requirements as well as ensuring a seamless and reliable experience as services are consumed and transactions are committed over the Web.

MySQL offers a range of technologies to deliver high availability (HA) and scalability, and one that has become increasingly relevant to specific use cases in Web and e-commerce applications is the MySQL Cluster database.

MySQL Cluster Carrier Grade Edition has proven itself as the HA database of choice for mission-critical telecommunications applications where 99.999 percent availability and real-time performance are key requirements. MySQL Cluster is now seeing rapid adoption by providers of Web applications and services in areas such as e-commerce, social networking, content delivery, and real-time data analytics. These services demand databases characterized by their ability to support

- 99.999 percent availability to minimize both planned and unplanned downtime
- Extreme levels of real-time read/write performance with predictable ultralow-latency response times
- Dynamic scalability to handle the most-unpredictable volumes of users and transactions

The purpose of this white paper is to explore the increasing demands being placed on databases to support Web and e-commerce applications. It presents an overview of the MySQL Cluster database, demonstrating capabilities designed to meet the needs described above. It also discusses sample workloads and real-world use cases, enabling you to understand the types of Web applications that benefit from the use of MySQL Cluster.

The Increasing Demands of Web and E-Commerce Applications

For nearly two decades, the promise of the internet has been to enable small startup companies to compete with large global businesses over the Web, driving competition and choice for consumers. However, it has been only the reduction of infrastructure costs—enabled by the rise of open source software technologies running on commodity hardware systems—that has made this promise a reality.

Online businesses can differentiate themselves at multiple levels, such as the services and products offered, pricing, fulfillment, and brand reputation. All of these, however, are meaningless if the infrastructure used to power the Web application, such as the e-commerce engine, is offline or response times are unacceptably slow.

"By building its infrastructure on MySQL Cluster, go2 has achieved a more stable environment, improved its user experience, and can now efficiently scale its platform with the growth of the mobile Web."
Dan Smith, Cofounder and CEO, go2 Media

As a result, any provider of an online application needs to focus on the availability and the performance of the service it is delivering, which, in turn, is a function of the infrastructure used to deliver the service. An additional dimension to consider is both the capital and operational costs of the service, because this has a direct impact on ROI.

There have been many studies over the years to assess the costs of downtime, and it has become clear that such costs are highly dependent on factors such as the type of application, the industry, and the number of users or transactions affected. Looking specifically at e-commerce applications used in online retail businesses begins to reveal the true financial impacts of downtime, with recent studies[1] calculating a cost to the business of $30,000 per minute, or $1.8 million per hour, in lost revenue. Direct revenue loss arising from the inability to process orders is only one aspect of the total revenue impact caused by systems downtime. To gain a complete picture, it is also necessary to consider the following factors:

- Damage to the brand image
- Impact on customer relationships, satisfaction, and loyalty
- Loss in employee productivity
- Potential regulatory issues if customer or financial data is corrupted or lost

To deliver a highly available Web-based service, an organization must consider multiple factors during technology selection:

[1]blog.dssdatacenter.com/2009/07/17/a-pragmatic-view-of-downtime-cost

- Ability to minimize both planned and unplanned downtime
- Instant detection of failure events
- Fast system recovery
- High throughput and scalability, with consistently short response times

Only when all these factors are considered is it possible to select technologies that can achieve the levels of uptime and performance demanded by Web and e-commerce applications.

Figure 1 plots availability measures expressed as percentages against real-world downtime experienced by users:

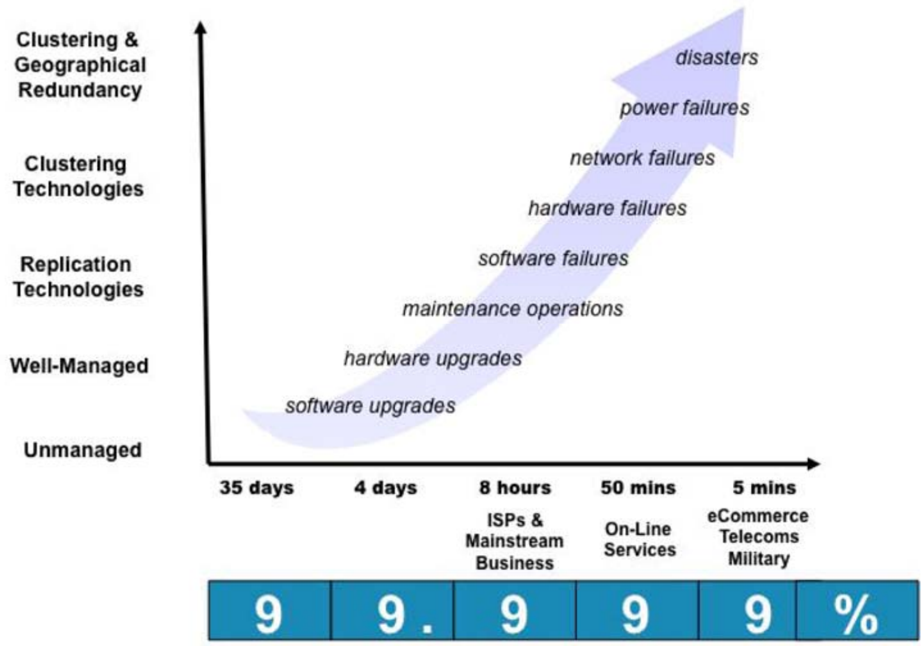

Figure 1. This chart maps availability against downtime.

Even higher levels of availability are possible with proprietary systems, such as servers and custom middleware, that provide full fault tolerance, but the cost and complexity of implementing such solutions often prove to outweigh any possible business benefits.

In summary, an infrastructure that delivers extreme levels of availability and scalability with instantaneous response and low cost provides a compelling business differentiation for an online business.

Using Databases to Achieve High Availability and Scalability

Whether you're racing to introduce a new service or trying to manage an avalanche of data in real time, your database has to be scalable, fast, and highly available to handle ever-changing market conditions and demanding customer requirements.

A database designed for Web-based applications such as e-commerce must support a real-time live data experience. It must be able to couple the advantages of relational databases with the responsiveness demanded by online Web applications.

Data and transactional states are usually the hardest parts of an online application to make highly available. Implementing a database that is itself highly available makes it easier for the application to become highly available as well. This approach permits delegating the complexity of data management and transactional states to the database layer. The clear advantage of this design is that the database will always be more competent, efficient, and reliable mechanism in handling these duties than other system components.

The essential capabilities of the database to be used to deliver HA, performance, and scalability are discussed below.

Minimizing Planned and Unplanned Downtime

A system architected for HA has been designed specifically to minimize downtime by providing resilience to system components that fail or are taken offline for maintenance. It is important to recognize that downtime can be divided into planned and unplanned downtime, as Table 1 shows:

TABLE 1. DIFFERENTIATING PLANNED FROM UNPLANNED DOWNTIME

PLANNED DOWNTIME	UNPLANNED DOWNTIME
Hardware upgrades	Hardware component failure
Software upgrades	Software bugs and crashes
Bug fixes	Operator error and malicious code
Backup routines	Datacenter failure (physical disaster)

To create an HA environment for Web-based applications, it is important to select only those databases that are designed to minimize both planned and unplanned downtime.

Combining technologies such as clustering, redundancy, online operations, and geographic replication can enable applications to meet (and often exceed) 99.999 percent availability, totaling just five minutes of downtime per year, by minimizing both planned and unplanned downtime.

Fast Failover and Recovery

The goal of any technology used to maximize system uptime must be to instantly detect failures and then fail over to clustered systems, typically in less than one second, and then continue delivering services and processing transactions without any impact on the user. Multiminute failover times cannot be tolerated by applications such as e-commerce storefronts and payment processing engines.

Database recovery is another element that is critical in delivering HA. It is all very well to ensure that the online application continues to provide services in the event of a failure or a maintenance operation, but if the remaining active system is unable to sustain millisecond response times and manage high throughput, the original objectives of designing for HA are compromised.

For this reason, it is important to select databases that enable either replacement systems or recovered systems to quickly rejoin and resynchronize with the active database cluster so that normal system resource levels are re-established to meet the performance requirements of the service. Database recovery also needs to be automatic, without requiring manual intervention by administrators or complex programming logic embedded in the applications.

"We have experienced two instances of hardware failure over the past couple of years. MySQL Cluster was able to instantly fail over, avoiding any service interruption to our users. When we added replacement hardware, MySQL automatically resynchronized it with the rest of the cluster, with no intervention from my team. It is an awesome technology."
Dave Dalrymple, VP Engineering, Shopatron Inc.

Performance

As with availability, there are multiple dimensions to achieving high performance of Web and e-commerce applications.

From the user perspective, the service they are using must be able to provide immediate responsiveness as it handles requests and perform transactions, such as placing orders, over the Web. To achieve this, the database must be able to do the following:

- Ensure consistently low levels of latency as users interact with the service in real time.
- Support peak levels of throughput, typically measured as the volume of simultaneous users and transactions per second.
- Support constant updates (writes). Online applications, especially those related to e-commerce and session management, have very high update (write) demands on their underlying database, so the database must be able to handle such transaction-intensive environments and not be limited to basic SELECTs that may be typical of nontransactional Web applications.
- Deliver predictability of performance, regardless of the load on the system, so that low latency and high throughput from the database can be maintained.

As with HA, delivering low latency, high throughput, and predictable response times can help ensure customer satisfaction, resulting in higher loyalty and revenue.

Scalability

It is impossible to predict how successful a new online service can become. A service that starts out appealing to several hundred users can quickly spiral to attract millions of online devotees through the increasingly viral nature of communications enabled by Web 2.0 technologies. Similarly, e-commerce applications may start by offering just a few products for sale online, but rapid expansions of product catalogs and recommendations, pricing promotions, and user adoption can require the environment to scale rapidly to support increasing sales volumes.

Scaling Out

In traditional enterprise IT environments, scaling was achieved by deployment of applications and databases to larger symmetric multiprocessing (SMP) servers, but it has long been understood that such a design approach is completely inappropriate for scaling Web-based applications. The cost of such solutions, coupled with their inability to ever scale sufficiently to handle exploding Web volumes, has driven the rise of “scale-out” architectures, where the processing and database load is deployed and scaled over low-cost commodity servers.

To achieve efficient levels of scalability, it is therefore necessary to select databases that can be distributed across multiple systems to handle the rapid increases in capacity and performance demanded by online systems. Even for scaling across many systems, it is also critical that the underlying database can continue to support the very high levels of write scalability demanded by the application.

“We’ve always known that our capacity to handle rapid Web traffic growth would be critical to the success of Zillow. We selected MySQL Cluster as a low-cost, high-throughput database with easy scalability—and it has delivered. MySQL Cluster has enabled us to manage our growth smoothly and cost-effectively and intelligently plan for future expansion.”
Brian Milnes, Director of IT Operations, Zillow.com

Scaling Up

Recent advances in microprocessor design have resulted in CPUs that boast multiple cores and threads. To take advantage of these new processing resources, the database must be multithreaded, enabling it to be scaled up even within low-cost commodity systems.

Scaling Online

Coupling scalability with HA is especially critical for online applications such as e-commerce engines. If a service has to be restarted to recognize and use new servers deployed into an infrastructure, the service will experience downtime. Therefore, databases must be selected that enable the infrastructure to be scaled online with no interruption to either the application or the users. Such online scaling ensures that the service can accommodate the most-unpredictable workloads typical in online applications without disrupting application availability.

Introducing the MySQL Cluster Carrier Grade Edition Database

MySQL Cluster is a real-time transactional database combining the flexibility of a highly available relational database with the low TCO of open source.

Featuring a shared-nothing distributed architecture with no single point of failure (see Figure 2), MySQL Cluster is designed to deliver the 99.999 percent availability demanded by Web and e-commerce applications.

MySQL Cluster's real-time design delivers predictable millisecond response times with the ability to service tens of thousands of transactions per second. Support for in-memory and disk-based data, automatic data partitioning with load balancing, and the ability to add nodes to a running cluster with zero downtime enable linear database scalability for handling the most-unpredictable Web-based workloads.

MySQL Cluster Carrier Grade Edition is already proven in the toughest telecommunications environments, delivering higher database throughput and shorter response times at a tenth of the cost of proprietary, clustered, shared-disk databases[2], with the added benefit of running on commodity hardware and operating systems. Customers include Alcatel-Lucent, BT Plusnet, Cisco, Deutsche Telekom, Ericsson, Telenor, and UTStarcom.

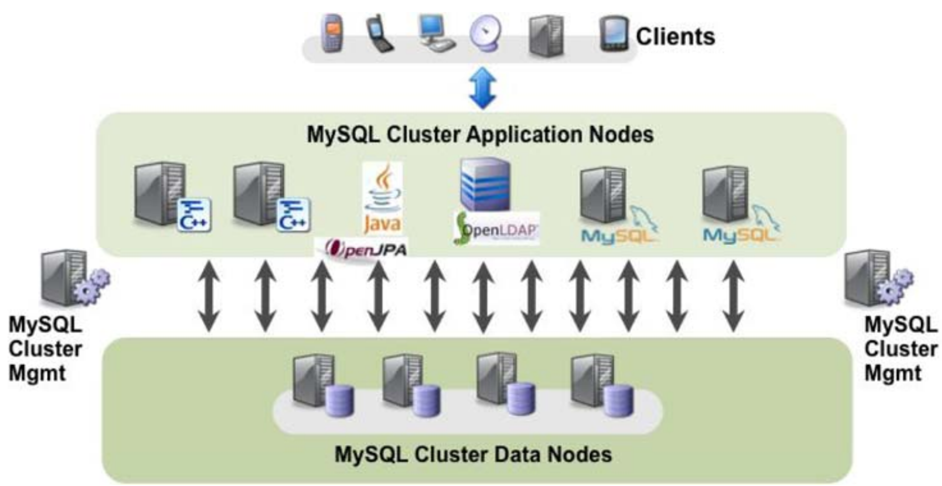

Figure 2. The MySQL Cluster architecture eliminates any single point of failure.

To learn more about the MySQL Cluster architecture, see the "MySQL Cluster 7: Architecture and New Features Whitepaper," at mysql.com/why-mysql/white-papers/mysql_wp_cluster7_architecture.php.

[2]mysql.com/why-mysql/case-studies/mysql-cs-alcatel.php

Identifying Target Applications for MySQL Cluster

It is important to recognize that because MySQL Cluster is a distributed shared-nothing database, certain ranges of application characteristics represent ideal candidates for it. Other applications are better served by other storage engines available with MySQL Server, due to specific requirements of the applications. Table 2 identifies the key characteristics to consider when deciding whether MySQL Cluster represents a good fit for a particular application.

TABLE 2. IDENTIFYING THE OPTIMAL APPLICATION FIT FOR MYSQL CLUSTER

GOOD FIT FOR MYSQL CLUSTER	COMPLEMENT WITH OTHER MYSQL STORAGE ENGINES
Fast write throughput with low-latency response	Database size of more than 2 TB
99.999 percent availability with subsecond failover and automatic recovery	Rows demanding more than 8 KB of memory and 128 fields
Unpredictable scalability demands	Requirement to store objects larger than 2 MB
Mainly primary key access, or subcomponents of multifield primary key	Full table scans with complex joins
Simple table joins	Foreign key support

By coupling MySQL Cluster with additional MySQL technologies, it is possible to use MySQL Cluster within applications that may not, at first glance, represent a good fit. These are discussed in the "MySQL Cluster Case Studies in Web and E-Commerce Applications" section later in this white paper.

Use Cases for MySQL Cluster in Web and E-Commerce Applications

The following section presents potential use cases for MySQL Cluster, which may frequently be part of a broader range of MySQL storage engines typically deployed within Web and e-commerce applications. These examples are by no means exhaustive but are designed to present potential areas in which MySQL Cluster brings unique value to Web-based applications.

MySQL Cluster for E-Commerce Applications

E-commerce applications have the most-stringent requirements for HA and real-time responsiveness, so specific modules of these applications can benefit most directly from the capabilities of MySQL Cluster.

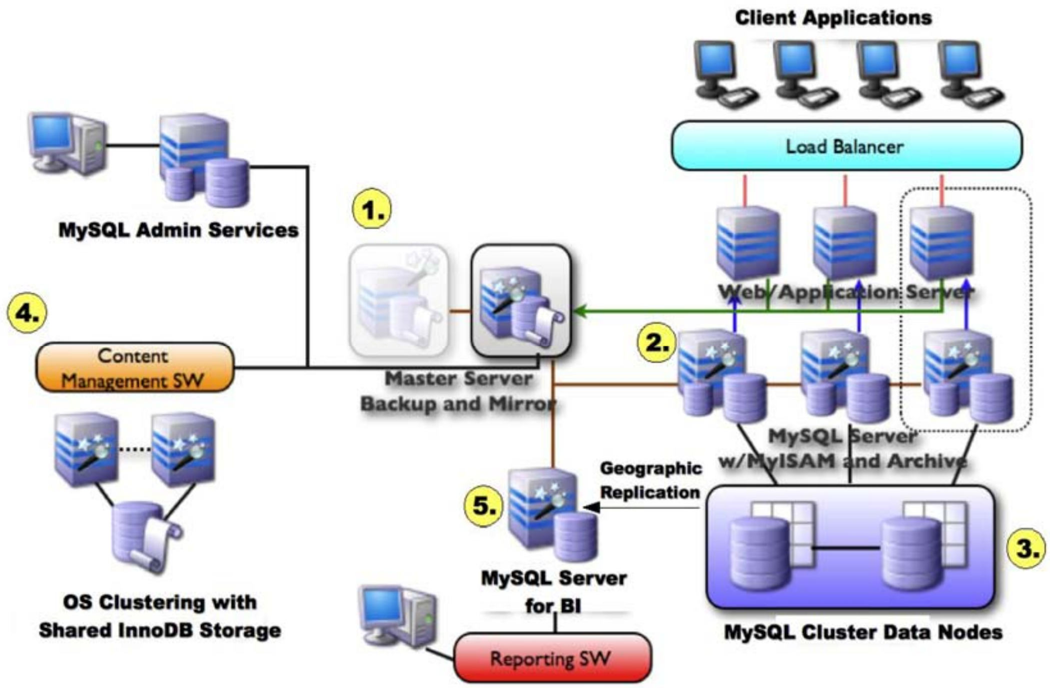

Figure 3. This figure illustrates the anatomy of an e-commerce application.

In Figure 3, the database architecture has been standardized on MySQL, with a variety of MySQL storage engines providing specific functionality for each module of the e-commerce application:

1. The master MySQL Server coordinates the various data management functions of the application.
2. MySQL Server with the MyISAM and Archive storage engines supports and captures user behavior while browsing and searching products. These lookup activities are load-balanced across the MyISAM slaves to ensure fast read scalability and HA.
3. MySQL Cluster manages user data for registration, access, authentication, and accounting. The MySQL Cluster data nodes are also responsible for managing session state, including the shopping basket as well as product codes and product availability. User orders and payments are captured in MySQL Cluster and then replicated to InnoDB tables for invoicing and shipping.
4. MySQL Server powers the content management, and InnoDB replicates product content to the master servers.
5. As users browse the product catalog, click-stream data is captured and stored in the Archive storage engine. The data is then loaded into MyISAM tables that the business intelligence server uses for analysis. Product recommendations can then be loaded back into the MyISAM tables that the browse and search modules of the application use. The application then displays the results to the user for cross-sell and up-sell purposes.

Based on the example above, the following modules represent the best candidates to be managed by MySQL Cluster:

TABLE 3. FOR THE PREVIOUS EXAMPLE, THE BEST CANDIDATES FOR MYSQL CLUSTER MANAGEMENT

APPLICATION TYPE	DATA ACCESS METHOD	DATA CHARACTERISTICS	DATA REQUIREMENTS
User data	Primary key—user ID	Read- and write-intensive (activity accounting, purchase history, order status)	HA, real-time access
Product codes and availability	Primary key—product code	Read- and write-intensive	HA, real-time access, scalability for simultaneous users and updates
Session management	Primary key (session ID, user ID)	Write-intensive	Subsecond failover, write scalability, real-time data access
Orders and payments	Primary key (user ID)	Read- and write-intensive	HA, subsecond failover, write scalability, real-time data access

In addition to the use cases above, MySQL Cluster can be used within a broader set of data management requirements for applications such as payment processing. This is the most complex element of any e-commerce application, requiring the coordinated flow of transactions across a complex network of processes and systems.

Typically, a single transaction touches systems holding user data and product catalogs and involves external providers such as credit card companies, issuing banks, merchant banks, and payment authorization engines. The transaction—and its state—must be persisted across each of these systems and is therefore ideally suited to MySQL Cluster, which can coordinate the flow of data across these systems without corruption or loss due to system failures.

"MySQL Cluster enables us to incrementally scale our database infrastructure to meet our growth needs, which we could not have done with our legacy system. Furthermore, this scale-out architecture gives us the flexibility to allocate development and administrative resources to where they are needed most."
Project manager, European e-commerce company

MySQL Cluster for Social Networking Applications

Social networking is arguably the poster child of the Web 2.0 revolution. For example, Facebook has witnessed stunning growth since its initial launch, in 2004. Only five years later, it served 400 million users, 200 million of which accessed the service every day, collectively uploading a billion images per month[3]. Facebook is a large MySQL customer, relying on MySQL to enable its infrastructure to keep pace with these huge demands.

Whereas Facebook is a dedicated social networking site, many other organizations have integrated social networking into their core online applications as a means to foster community building and collaboration among members.

The modules in Table 4 represent the best candidates to be managed by MySQL Cluster:

TABLE 4. FOR THE PREVIOUS EXAMPLE, THE BEST CANDIDATES FOR MYSQL CLUSTER MANAGEMENT

APPLICATION TYPE	DATA ACCESS METHOD	DATA CHARACTERISTICS	DATA REQUIREMENTS
User profile	Primary key—user name	Read-intensive	HA, real-time access
Status updates	Primary key—user name	Write-intensive	HA, real-time access

MySQL Cluster for Online Content Delivery

"In comparison to other solutions, MySQL Cluster is superior to other databases in terms of maintenance, safety, stability, scalability, and reliability."
Jo Rogner, Technical Manager, KURIER.at

Content management systems (CMSs) are critical for Web properties to manage the workflow needed to collaboratively create, edit, review, index, search, publish, and archive various kinds of digital media. However, MySQL Server is the recommended database for core CMS functions, ensuring that user access and monetization (subscriptions) are modules that can be effectively managed by MySQL Cluster.

Table 5 characterizes the components of a CMS:

TABLE 5. THE COMPONENTS OF A CMS

APPLICATION TYPE	DATA ACCESS METHOD	DATA CHARACTERISTICS	DATA REQUIREMENTS
User data and user authorization	Primary key—user ID	Read-intensive	HA, real-time access
Content catalog	Primary key—subject,	Read-intensive	HA, real-time access, scalability for

[3]facebook.com/press/info.php?statistics

	author		simultaneous users
Session management	Primary key (session ID, user ID)	Write-intensive	Subsecond failover, write scalability, real-time data access
Online transaction processing (OLTP)	Primary key (user ID)	Read- and write-intensive	HA, subsecond failover, write scalability, real-time data access

MySQL Cluster for Service Access, Authorization, and Monetization

Online application providers are dependent on reaching new customers and building new revenue streams by deploying large-scale services over fixed and wireless networks.

All these services are highly dependent on technologies such as IP management; Lightweight Directory Access Protocol (LDAP); and centralized authentication, authorization, and accounting (AAA) protocols, enabling users to connect and consume network services and network administrators to capture usage information. Before clients can start to access and consume services on a network, they must be authenticated to the network and then authorized to use the services to which they are entitled. Their consumption of network resources then needs to be captured via the accounting processes. Collectively, AAA is a cornerstone of today's network security; management; and in many cases, monetization.

"MySQL Cluster delivers the high availability that enables us to guarantee continuous broadband internet access and VoIP services to our subscribers. This has had an immediate impact on significantly improving customer satisfaction and has reduced the cost of operating our network."
Lars-Ake Norling, B2, Telenor

To ensure that AAA services function, user data accessed by the authentication and authorization processes and user activity recorded in the accounting details need to be persisted to back-end AAA data stores.

With the online user population continuing to explode as bandwidth increases, communication costs decrease, and services available over fixed and wireless networks proliferate, the back-end data stores of AAA and LDAP services are now more mission-critical than ever.

As these networks grow, limitations can occur that add administrative overhead, inhibit flexible scaling, and affect the timely synchronization of data across the AAA and LDAP environments.

To address these challenges, MySQL has collaborated with the leading AAA[4] and LDAP server[5] vendors to integrate MySQL Cluster into their solutions to provide a robust back-end data store designed to meet the performance and availability requirements of online applications.

Components of an AAA and LDAP infrastructure that can be managed by MySQL Cluster are characterized in Table 6:

TABLE 6. THE COMPONENTS OF AN AAA AND LDAP INFRASTRUCTURE THAT MYSQL CLUSTER CAN MANAGE

APPLICATION TYPE	DATA ACCESS METHOD	DATA CHARACTERISTICS	DATA REQUIREMENTS
User authentication	Primary key—internal LDAP user identifier	Read-intensive	HA, real-time access, high scalability for simultaneous user access
User authorization	Primary key—internal LDAP user identifier	Read-intensive	HA, real-time access
User accounting	Primary key—internal LDAP user identifier	Write-intensive	Subsecond failover, write scalability, real-time data access
LDAP access and updates	Primary key—internal LDAP user identifier	Read-intensive (write volumes growing)	HA, scalability, real-time data access

Advantages of MySQL Cluster for Web and E-Commerce Applications

Table 7 details the Web- and e-commerce-related capabilities of MySQL Cluster:

TABLE 7. MYSQL CLUSTER MEETS THE TOUGHEST CHALLENGES OF WEB-BASED APPLICATIONS

REQUIREMENT OF ONLINE APPLICATION	MYSQL CLUSTER CAPABILITY
High reliability and HA	• Support for atomic, consistent, isolated, durable (ACID) transactions • Distributed shared-nothing architecture • Synchronous data replication • Automated subsecond failover

[4]For more information, see “Delivering Scalable & Highly Available Authentication & Accounting Services”: mysql.com/why-mysql/white-papers/mysql_wp_ha_auth_account.php.

[5]For more information, see “Guide to Scaling OpenLDAP with MySQL Cluster”: mysql.com/why-mysql/white-papers/mysql_wp_openldap-scaling-guide.php.

	• Automatic recovery and data synchronization • Local and global checkpoints to disk for durability • Geographic replication between clusters • Online upgrades • Online add nodes • Online backup • HA consulting • 24/7 technical support • Professional training and certification
Performance and scalability	• Hybrid in-memory and disk-based storage • Parallel server architecture for high read/write performance • Automatic data partitioning enabling load balancing • Distributed database for scaling out • Multithreaded data nodes for scaling up • Online add nodes for on-demand scaling
Interoperability	• World's most popular open source database • Support for multiple hardware platforms and operating systems • SQL and native APIs: Java/JPA, C++, LDAP, HTTP for data access • Support for full range of MySQL Connectors
Low TCO	• Open source freedom, standards, and economics • Low-cost service offerings • Ability to run on commodity hardware, including rackmounts and blades • Elimination of costly shared storage • Simplified monitoring and management

Achieving High Availability with MySQL Cluster

The architecture of MySQL Cluster is designed to deliver 99.999 percent availability, which includes regularly scheduled maintenance operations as well as system failures (planned and unplanned downtime).

This level of availability is achieved via a distributed shared-nothing architecture and by synchronous replication of data, which automatically propagates transaction information to all appropriate data nodes, before completion of the transaction. If one or more database nodes fail during a transaction, the application will simply retry the transaction and the remaining data nodes will successfully satisfy the request.

The advantage of synchronous replication is that it eliminates the time shared disk architectures consume in having to re-create and replay log files for the application to fail over.

MySQL Cluster detects any failures instantly, and control is automatically failed over to other active nodes in the cluster without interruption of service to the clients (see Figure 4). In the event of a failure, the MySQL Cluster database nodes can automatically restart, recover, and dynamically reconfigure themselves—all completely transparently to the application.

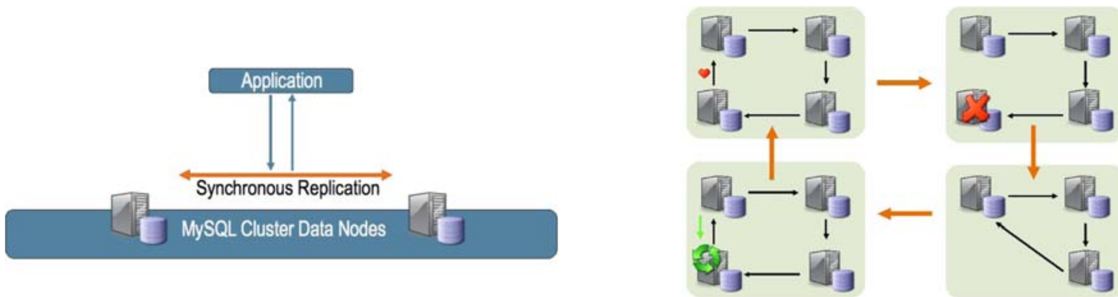

Figure 4. With synchronous replication and subsecond failover, MySQL Cluster minimizes downtime.

If all nodes fail due to a power failure or a multisystem hardware failure, MySQL Cluster will ensure that an entire system can be safely recovered to a consistent state. It does so by maintaining a series of checkpoints on the local disks of each data node, giving durability to the database.

To further support continuous operation, MySQL Cluster Carrier Grade Edition can add nodes online to a running cluster by automatically repartitioning data as new node groups are added, ensuring that the cluster will maintain continuous operation and application connectivity. MySQL Cluster also enables online updates to a live database schema, in addition to upgrades to and maintenance of the underlying hardware and software infrastructure.

The ability to withstand site failures by replicating clusters across multiple remote locations is an important capability for many global online services. Geographic replication (see Figure 5) is a feature of MySQL Cluster, commonly implemented in order to

- Achieve higher availability within the datacenter or across a geographic wide area network (WAN)
- Provide lower-latency data access in different geographies
- Replicate data to another near-real-time database for complex data analysis, without having an impact on the performance of the main, real-time production database

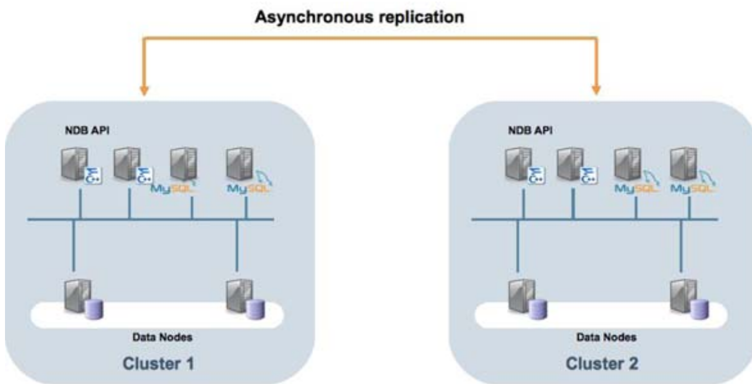

Figure 5. Geographic replication provides multisite protection for critical online applications.

Through the capabilities described above, MySQL Cluster is able to eliminate both planned maintenance and unplanned downtime in order to deliver the 99.999 percent availability demanded by Web-based applications.

Achieving High Performance and Linear Scalability with MySQL Cluster

In addition to 99.999 percent availability, high performance is a critical requirement for accommodating the massive volumes of database requests and transactions typically associated with Web-based services, especially e-commerce applications. Database performance requirements are typically tens of thousands of operations per second, with a consistent response time latency of just a few milliseconds.

As a distributed database, MySQL Cluster employs a parallel server architecture with multiple active master nodes. This ensures that transactions (both reads and writes) can be load-balanced and automatically scaled across multiple SQL servers simultaneously, with each SQL node able to access and update data across any node in the cluster.

MySQL Cluster also offers a flexible architecture with the ability to store indexes as well as data in memory or on disk. As a result of this in-memory characteristic, MySQL Cluster can limit disk-based I/O bottlenecks by asynchronously writing transaction logs to disk, therefore maintaining real-time performance.

MySQL Cluster 7.0 was recently benchmarked with the DBT2 test suite (see Figure 6) and achieved 251,000 transactions per minute (TPM) with just four data nodes[6]. Each transaction involved approximately 30 database operations, and MySQL Cluster was able to sustain approximately 125,000

[6]blogs.sun.com/hasham/entry/mysql_cluster_7_performance_benchmark

operations per second, with an average response time of only 3 milliseconds. This performance increase represented a 4x improvement in scalability over previous versions of MySQL Cluster.

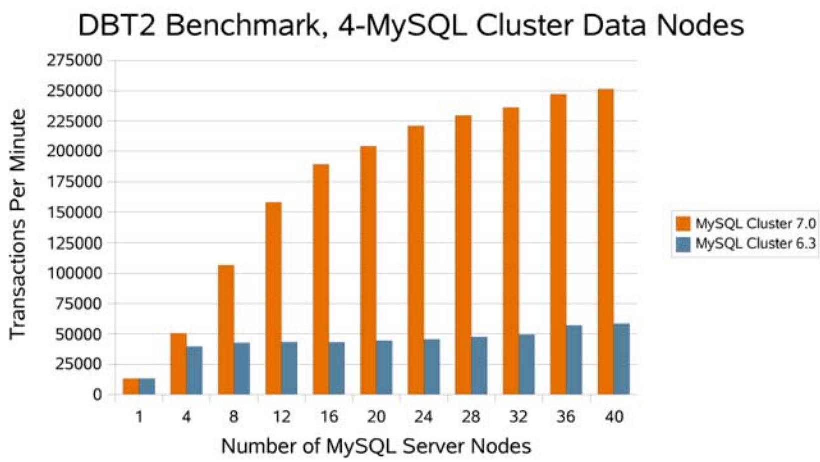

Figure 6. MySQL Cluster achieves more than 250,000 TPM, or 125,000 operations per second, with an average latency of only 1.5 milliseconds.

Note: *More than one MySQL Server node was installed on each physical server, with several of the server instances used as load generators to the MySQL Cluster database. An alternative deployment could use multiple connections from each MySQL Server node, which would have resulted in the use of fewer MySQL Server nodes.*

In addition to scaling performance, MySQL Cluster can be deployed into a range of scaling scenarios. The distributed architecture enables it to be easily scaled across multiple low-cost servers. Using the latest multithreaded data nodes, MySQL Cluster can be used in server architectures equipped with multiple cores and threads. MySQL Cluster also supports the addition of nodes to a running cluster without interruption from the standpoint of the applications or users, enabling it to handle the most-unpredictable and volatile workloads demands of Web-based applications.

With support for as many as 255 nodes, organizations have the flexibility to start small and make incremental investments to increase capacity as service adoption and e-commerce revenue streams grow. This eliminates the need for a large initial hardware and software investment in what tends to be over-powered and costly configurations.

Through the combination of a distributed, parallel server architecture coupled with in-memory index and data support and a choice of scaling options, MySQL Cluster is highly suited to demanding, transaction-intensive e-commerce and Web-based applications.

"MySQL Cluster has enabled us to meet our demands for scalability, performance, and continuous uptime, at a much lower cost than proprietary technologies'. We would be dead in the water without it."
Richard McCluskey, Senior Engineer, go2 Media

Flexible Data Access

Application developers can easily integrate new and legacy applications, using their preferred database-independent method. MySQL Cluster Carrier Grade Edition provides multiple data access methods that work together. These include SQL and native APIs such as C++, Java, LDAP, and Web services.

Designed for Java developers, the new MySQL Cluster Connector for Java implements a high-performance, easy-to-use native Java interface and OpenJPA plug-in that directly maps Java objects to relational tables stored in the MySQL Cluster database.

With the elimination of data transformations into SQL, users get lower data access latency and higher throughput. In addition, Java developers have a more natural programming method for directly managing their data, with a complete, feature-rich solution for object-relational mapping. As a result, the development of Java applications is simplified, with faster development cycles resulting in shortened time to market for new services.

Simplified Management and Monitoring

According to industry research, staffing and downtime comprise almost 50 percent of database TCO[7]. At the same time, IT budget and staffing levels are declining or flat, whereas demand for IT services continues to increase. Simplified management and monitoring tools for MySQL Cluster Carrier Grade Edition help address these challenges by

- Maximizing administrator productivity with tools for managing database clusters more efficiently
- Minimizing database downtime caused by human error
- Doing proactive monitoring of real-time usage statistics to ensure seamless database operations

Management and monitoring tools for MySQL Cluster Carrier Grade Edition serve to reduce database TCO and create a more agile, highly available IT environment by slashing management complexity, increasing administrator efficiency, and reducing the risk of downtime.

MySQL Cluster Manager

MySQL Cluster Manager simplifies the creation and management of MySQL Cluster Carrier Grade Edition by automating common management tasks. As a result, DBAs and systems administrator are more productive, so they can focus on strategic IT initiatives that support the business and respond more quickly to changing user requirements. At the same time, risks of database downtime previously attributable to manual configuration errors are significantly reduced.

[7]"Maximizing the Business Value of Enterprise Database Applications" (IDC)

For example, management operations requiring rolling restarts of a MySQL Cluster database that previously demanded 46 manual commands[8] and consumed 2.5 hours of DBA time[9] can now be performed with a single command and are fully automated, serving to reduce

- Cluster management complexity and overhead
- Custom scripting of management commands
- The risk of downtime through the automation of configuration and change management processes

You can learn more about MySQL Cluster Manager at mysql.com/cluster/mcm.

NDBINFO

NDBINFO presents real-time status and usage statistics from the MySQL Cluster data nodes as SQL tables and views, giving developers and administrators a simple and consistent means of proactively monitoring and optimizing database performance and availability.

NDBINFO makes it simple to

- Monitor resource consumption against configured limits, including the consumption of data and index memory, along with buffers and log spaces. This approach ensures that administrators can reconfigure resources or optimize their tables before performance or availability is affected.
- Expose statistics on database operations such as transactions, reads, writes, table and range scans, and aborts. These statistics can be especially useful to developers in optimizing application access to the database.
- Check the status and uptime of each data node and its connections with all other nodes in the cluster.

Rather than manually searching through logs to find root cause errors or optimize configurations, NDBINFO enables administrators to quickly and easily monitor MySQL Cluster operations in real time and proactively tune configuration parameters and application access to ensure seamless database operations.

[8]Based on a MySQL Cluster configuration comprising four MySQL Cluster data nodes, two MySQL Server SQL nodes, and two MySQL Cluster management nodes implemented across individual servers (eight nodes total). Total operation included the following commands: one preliminary check of cluster state, eight Secure Shell commands per server, eight per-process stop commands, four Secure Copies of configuration files (two mgmd and two mysqld), eight per-process start commands, eight checks for started and rejoined processes, eight process completion verifications, and one verify completion of the whole cluster. The total command count does not include manual editing of each configuration file.

[9]Based on a DBA's restarting four MySQL Cluster data nodes, each with 6 GB of data, and performing 10,000 operations per second (http://www.clusterdb.com/mysql-cluster/mysql-cluster-data-node-restart-times/)

Complementary MySQL Technologies for Web and E-Commerce Applications

By coupling MySQL Cluster with additional MySQL technologies, it is possible to bring the benefits of MySQL Cluster to applications that may not, at first glance, represent a good fit.

Using the MySQL Pluggable Storage Engine Architecture to Meet Diverse Application Needs

Many online applications are highly modular, with each module providing specific functionality. For example, an e-commerce application may include a range of modules to handle specific tasks, such as

- Management of user data
- Surfing and searching of the product catalog
- Handling session state, especially for shopping baskets
- Reporting of purchase recommendations (that is, a user who purchased x also purchased y)
- Payment processing and checkout

MySQL users have long benefited from the MySQL pluggable storage engine architecture, which enables a developer or a DBA to select a specialized storage engine for a particular application need. A MySQL storage engine is a low-level engine that manages the storage and retrieval of data and can be accessed through a MySQL server instance and, in some situations, directly by an application. MySQL Cluster is implemented as a MySQL storage engine.

The flexibility of MySQL extends to being able to select a specific storage engine for each table accessed by an application. Therefore, in the e-commerce example above, certain modules of the application can use MySQL Cluster to achieve the highest levels of availability and performance when handling product and user data, whereas the recommendations reporting involving complex queries is handled by another MySQL storage engine such as MyISAM.

The use of the storage engine is transparent to both the application and the user.

Using MySQL Cluster's Flexible Replication for Online Applications

One of the major advantages of using MySQL Cluster for online applications is the ability to handle demanding write-intensive workloads. Geographic replication capabilities can then be used to replicate the data, typically within a second of commit, to another MySQL server storage engine such as InnoDB or MyISAM (possibly running on a different host or site). This data can then be used in multiple ways. For example, it can be presented back to users as they are shopping online, to support cross-sell or up-sell opportunities. The data can be used to automatically trigger additional business processes, such as stock replenishment, or can be presented to business analysts to enable them to make better and more judicious decisions based on real-time data feeds.

In this example, geographic replication is implemented to replicate data from MySQL Cluster to a remote cluster to provide multisite resilience (see Figure 7).

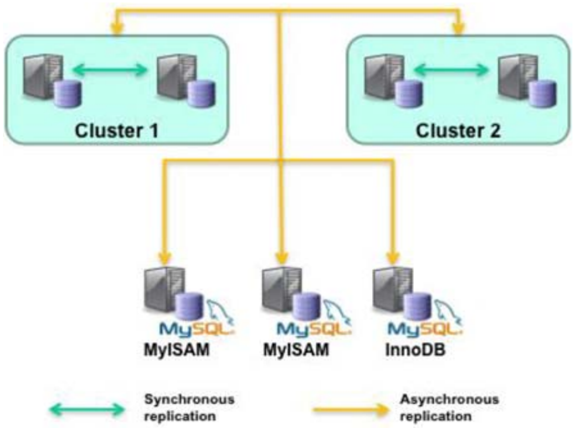

Figure 7. Flexible replication enables near-real-time data analytics.

The data is also replicated by MySQL Server from the MySQL Cluster storage engine to an InnoDB or MyISAM storage engine, against which complex queries involving full table scans and multilevel joins can be run.[10]

With this type of functionality, MySQL Cluster can still be used to handle write-intensive workloads demanding the highest levels of scalability and real-time performance, even though the application itself demands the ability to run complex queries involving operations that may not be suitable for MySQL Cluster.

MySQL Cluster Case Studies in Web and E-Commerce Applications

The following section presents case studies describing MySQL Cluster deployments in Web and e-commerce applications.

E-Commerce Case Study 1: Shopatron

Shopatron is one of the world's leading consumer goods, e-commerce, and order management service providers. With Shopatron's e-commerce services, orders placed on manufacturers' Websites are

[10]Learn how to deploy this type of replication scenario: johanandersson.blogspot.com/2009/05/ha-mysql-write-scaling-using-cluster-to.html.

offered to a managed network of global partners for fulfillment, typically for assignment to local retailers.

Since its founding, in 2000, Shopatron has grown to provide the e-commerce and order management operations for more than 700 global and local manufacturers, with more than 10,000 retail partners operating in eight currencies and three languages across North America and Europe.

As Shopatron's business began to grow, so did the demands placed on its Web and e-commerce infrastructure. Each new manufacturer using Shopatron's services added 100 to 1,000 new retailers. All new orders were loaded into the e-commerce system and persisted to a back-end database, typically in daily batch runs via XML-based transactions. All new orders were then accessed over a secure Web connection once a day by the manufacturers' retail partners, which would then bid to fulfill the order, depending on their local stock availability.

The daily cycles of new order entry and then access by retail partners placed enormous peak demands, which were quadrupled between the Thanksgiving and Christmas holiday periods, on Shopatron's e-commerce systems.

Shopatron had built its e-commerce infrastructure on a largely open source stack of software components, including Debian Linux, the Apache Web server, and PHP. The back-end database had been deployed on an SMP server running a single proprietary database instance.

As the demands to scale the infrastructure grew, the database became a bottleneck. Shopatron was aware that it needed to employ a scale-out approach to its database layer, distributing the database over smaller commodity nodes to manage costs and keep pace with the volume of orders and retailers as their business grew.

Shopatron needed an e-commerce database that could handle very high volumes of concurrent reads by the retail partners of its manufacturer customers. To achieve continuous availability and meet stringent service-level agreements (SLAs), it needed to eliminate any single point of failure from its database infrastructure. Shopatron also needed to make sure it selected a database that offered operating system flexibility, especially one that would run and be supported on Debian Linux.

In 2006 Shopatron began investigating clustered database solutions that would provide the performance, scale, and availability demanded by its growing e-commerce business. Having used the MySQL server for several internal projects, it was aware of the performance and reliability offered by MySQL technologies. It was also aware that MySQL offered a real-time HA implementation of the MySQL Cluster database, so it made the decision to download the product and begin evaluations.

Shopatron's developers and DBAs attended the annual MySQL user conference, where they were able to meet directly with MySQL Cluster engineers and consultants, from whom they captured best practices in configuration, schema, and query optimization. Following the user conference, Shopatron was able to apply these best practices to its evaluation, optimizing all its queries for significant performance improvements over its legacy database. Primary-key-based lookups and queries that took three seconds to run with its existing database were executed by MySQL Cluster in only 2 milliseconds. Through additional testing, the company discovered that the distributed and parallel architecture of

MySQL Cluster was able to handle a much higher load of read and write operations than other clustered databases under evaluation.

As a result of its evaluation, Shopatron selected MySQL Cluster to power its e-commerce services and deployed the technology into production in 2006. MySQL Cluster provides the back-end database for the entire e-commerce fulfillment engine, from storing all order data to managing retailer stocking lists, tracking order fulfillment status, and providing user authentication services.

The MySQL Cluster database has supported double year-to-year growth in data volumes, with throughput rates of several thousand queries per second.

Shopatron purchased licenses as well as support for MySQL Cluster at a fraction of the cost of alternative products. Thanks to the reliability and stability of MySQL Cluster, Shopatron was also able to eliminate the costs it incurred from a third-party vendor that had been monitoring and supporting its existing legacy database. Shopatron annually renews its support contract for MySQL Cluster, securing additional expertise when needed.

Shopatron relies on MySQL Cluster to support the high performance, scalability, and availability demands of its e-commerce application. It is now exploring advanced replication technologies for loading data from MySQL Cluster tables into database engines optimized for complex query analysis and business reporting.

Learn more about Shopatron's experiences with MySQL Cluster by reading the case study and listening to the on-demand Webinar posted at mysql.com/customers/view/?id=1080.

E-Commerce Case Study 2: Leading European Online Mail Order Company

This organization launched its e-commerce shopping site in 1995 and has become one of the largest online retailing companies in Europe, receiving millions of visitors each day and stocking several hundred thousand items for online purchase.

Website performance, scalability, and reliability are critical to the company's success. At the core of the e-commerce site is the Web session management application. Session management is required by all online shopping applications for securely managing user profile information and e-commerce shopping cart applications.

The online retailer was using SMP servers to store user profiles and manage session data. To add additional capacity to meet the upcoming Christmas shopping demand, the company had to add capacity to it Website. At the same time, it needed to find a way to make better use of its financial resources by lowering capital expenditures and operating costs and improving flexibility. The retailer was able to reach this goal by replacing its proprietary scale-up environment with MySQL Cluster running on a grid of commodity Intel-based blade server hardware.

The MySQL Cluster implementation delivers the performance, availability, and scalability necessary to enable customers to enjoy a positive online shopping experience. Plus, using MySQL Cluster, the company is able to implement retail merchandising to up-sell and cross-sell additional products and services based on customer shopping behavior.

MySQL Cluster is the highly available repository for dynamic session data. By tracking user activity, the retailer is able to provide fast and easy navigation and ensure that users can smoothly complete an e-commerce transaction. The retailer understands that lengthy wait times have a negative impact on the user experience and a direct effect on revenues. Although the session data is stored on the cluster for only short amounts of time, the company relies on the performance of MySQL Cluster for instantaneous read and write data operations.

One of the retailer's requirements was the flexibility to cost-effectively grow its infrastructure. The MySQL Cluster parallel server architecture enables scalability in a linear fashion, making it possible for the company to make incremental investments to increase capacity as its needs grow.

The retailer needs to ensure that customers can always browse, search, and place orders, especially under peak loads. MySQL Cluster uses synchronous replication to replicate session data across multiple database nodes. If a database node fails, the application will continue uninterrupted so customers can complete their transactions successfully.

By combining a scale-out methodology with open source software and commodity components, the retailer has been able to lower costs and accelerate ROI while improving the customer experience.

Social Networking Case Study: Zillow.com

Zillow.com is one of the most visited real estate communities where homeowners, buyers, sellers, and real estate agents can find and share vital information about homes for free. Launched in early 2006 to provide "Zestimate" values and data on millions of U.S. homes, Zillow has since opened up the site to community social networking input, data, and dialogue. Zillow's goal is to help people become more informed about real estate—what homes are worth, what's for sale, and what local experts have to say about real estate and individual properties. In an average month, Zillow.com handles more than four million unique visitors.

As soon as Zillow launched, it immediately generated a massive amount of Web traffic. As the company expanded its data and services and started to promote its distinctive brand, interest in Zillow continued to climb. Zillow's founders knew from the company's inception that the site's ability to quickly process and manage massive amounts of data in real time would be a key factor in its success. The company identified a need for a low-latency, reliable Web database that would enable it to continue to increase the capacity of its infrastructure indefinitely without sacrificing performance.

Zillow uses MySQL Cluster as the high-transaction, low-latency, Web-centric database for conducting critical data processing for the Zillow.com community in real time. The open source database has helped Zillow scale on a single interface to meet many of the database requirements of a high-growth Website. Although Zillow uses a variety of databases to address different data processing needs, MySQL Cluster on inexpensive hardware has proven to be a high-availability, low-cost solution for its specialized Website application.

Scaling out in a horizontal architecture is made possible through MySQL's simple-yet-powerful replication technology, enabling Zillow to grow efficiently and cost-effectively.

As a result of its success with MySQL Cluster, Zillow is continuing to evaluate the database for new applications that could benefit from an affordable scale-out architecture.

Learn more about Zillow's experiences in this case study: mysql.com/customers/view/?id=908.

Online Content Delivery Case Study: kurier.at

More than one million Austrians read *Kurier* on the internet, making kurier.at one of the most successful news Websites in the German-speaking world. The internet presence of Austria's largest quality newspaper provides editorial news and comprehensive information around the clock. As the Web supplement to the daily newspaper, Kurier.at provides additional 24-hour news updates and interactive online options.

When the kurier.at Website was being built, a reliable and high-performance database solution was required for the management of user data and sessions. Particularly against the background of the tight fiscal pressure of online media, some products were excluded from the outset for cost reasons. Approximately three years ago, kurier.at decided in favor of a solution based on the MySQL Cluster database. One of the most important goals for the developers was to implement a highly flexible system that involved no additional costs as the service evolved and scaled.

For more than 10 years, kurier.at had been using MySQL as a database server. Centralized Web services such as session management, user data, and the in-house CMS are hosted on MySQL servers. Using MySQL Cluster for the central login and user data area of kurier.at proved superior to using solutions from other providers, in terms of maintenance, stability, scalability, and reliability. Currently MySQL Cluster is used to manage more than 150,000 users; track more than one million page views every day; and handle important transactions, such as newspaper subscriptions, that are processed online.

Learn more about *Kurier*'s experiences in this case study: mysql.com/customers/view/?id=998.

Service Access and Authorization Case Study: Bredbandsbolaget (B2)/Telenor

B2, part of the Telenor group, is one of Sweden's largest broadband internet providers, enjoying the most satisfied user base in the country, according to independent surveys. Following major investments in its network infrastructure, B2 is able to provide multiple services across its converged network, including broadband access, Voice over IP (VoIP), and broadcast television.

To quickly add new services and provision new users, B2 needed an AAA infrastructure that provided continuous availability, high performance, and fast failover in the event of hardware or software errors. The previous database solution was not able to keep pace with the increased demands, causing customer satisfaction issues and major network administration overhead, so B2 turned to MySQL Cluster, which has provided it with the following benefits:

- No single point of failure, as a result of the distributed architecture
- Subsecond failover to guarantee continuous availability

- Ability of failed nodes to automatically restart and reconfigure themselves without operator intervention
- Price/performance ratio several times better than competing offerings'
- Ability to handle 30,000 transactions per second on commodity hardware
- Improved customer satisfaction and reduced network operation costs

Learn more about B2's experiences in this case study: mysql.com/customers/view/?id=499.

Conclusion

This white paper has explored how MySQL Cluster is increasingly being used by Web-based application providers and e-commerce companies to cost-effectively deliver services that meet user requirements and business SLAs and provide competitive differentiation.

Originally developed and proven within the telecommunications industry, MySQL Cluster, with its HA, real-time performance, and linear scalability characteristics, is an ideal component of a complete database platform for Web and e-commerce providers that must achieve 99.999 percent availability.

With successful deployments in applications including e-commerce, social networking, content delivery, and service access/authorization, MySQL Cluster makes it easy for any online applications provider to get started today by downloading the product from the MySQL Website[11] and using the resources listed in Appendix A to start an evaluation.

Appendix A: Additional Resources

MySQL Cluster on the Web:

mysql.com/cluster

MySQL Cluster Customer Case Studies:

mysql.com/customers/cluster/

Getting Started with MySQL Cluster:

mysql.com/products/database/cluster/get-started.html

[11]mysql.com/downloads/cluster/

MySQL Cluster 7: Architecture and New Features Whitepaper:

mysql.com/why-mysql/white-papers/mysql_wp_cluster7_architecture.php

MySQL Cluster Evaluation Guide:

mysql.com/why-mysql/white-papers/mysql_cluster_eval_guide.php

Oracle is committed to developing practices and products that help protect the environment

MySQL Cluster for Web and E-Commerce Applications
April 2010

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com