Session Management with MySQL

How to increase performance, scalability and security within a Session Management architecture using MySQL

A MySQL® Technology White Paper

May 2006

Table of Contents

1 Executive Summary

As the web continues to provide one of the most cost-effective ways for businesses to market and sell their goods & services, we will continue to see the investment in web-based applications rise. An inherent characteristic about the web and one of the most common technical issues businesses face when designing and maintaining their web infrastructure, is addressing the stateless interaction between the user's browser and the web server from which the pages are requested. This stateless interaction may be excellent for viewing pages at random or clicking through hyper links, however, many web applications require knowledge of a user's interactions and current state on the website, as in online banking or ecommerce. The inability to properly store and access security credentials or reliably add and remove items that may have been placed in a virtual shopping cart, for example, can lead to an end result that is disastrous or unfeasible. In this paper we explore how MySQL and MySQL Cluster can be used as part of a cost-effective, high-performance, open-source solution for database–centric session management.

2 Session Management Solutions

Because the web has commoditized many goods & services, companies are always striving for users to have better then expected experiences when visiting their sites. Higher customer satisfaction and brand loyalty are all key elements to ensuring a company can continue to be successful on the web. Ensuring that situations do not occur, like a potential customer switching to a competitors site in frustration because all the items in their shopping cart were lost while browsing, affects the bottom-line. An effective session management strategy should be a component of every company's online strategy. When designing a session management architecture, it makes sense to look at ways in which these sessions can be accessed on an open, reliable and high performance manner.

The technical reason why there is a need for session management in the first place, has to do with the HTTP protocol itself. This protocol is used in the interaction between browsers and web servers, and is stateless by nature. At a high level what is meant by "stateless" is that the HTTP protocol does not persist any information about what pages the user has requested from the Web server or what interactions/transactions they have engaged in. Many traditional database applications use sessions to help control user interaction, while other applications use sessions to reduce processing and resource consumption on the server-side.

Browser vs. Server-side Solutions

Applications often need to use the result of one request when processing another. For example, a request that adds an item, say a book to a shopping cart while browsing in a virtual bookstore, needs to be remembered when the request is made to create the order at checkout time. In other words, the state of the application needs to be stored between these HTTP requests. There are two common ways to achieve this: variables defining the state of the session can be stored in the browser via cookies and included with each request or cookies can be used in conjunction with storing these variables on the server (web, application or database tier).

Cookies are a good solution if the application does not require large or complex data sets to be kept between the browser and server requests. However, there are some limitations involving cookies to be aware of. For example, browsers can have hard limits on the number of cookies they can store from any given domain and even place limits on their size. Privacy and security concerns by individuals and businesses about cookies should also factor into your session management strategy, especially if they are disabled in the browser by the end user.

Another common approach is to store and manage sessions on the server (web, application or database tier). With server-side session management, it is important to remember that session variables and data need to be stored for each session. So, as volume and traffic increases, so do the sessions that will need to be supported on the server. Coupled with the size and complexity of the data, scalability and performance issues may quickly arise.

Another consideration is for how long the session data should be stored for security purposes. Because the nature of HTTP is stateless, there is no way to know when a user has finished with a session and is no longer still on the site. Some interactions may require the user to login, giving the opportunity to use a logout, to trigger the appropriate deletion of stale session data. However, this is not always practical or feasible based on the specifications of your application. There is a risk in not removing these sessions, not only because of the resources they consume on the server, but these stale sessions may also allow for what is known as "session-hijacking". How long the timeout should be before deleting these stale session states depends on the needs of the application and your business rules.

3 Session Management at the Database Tier

One popular strategy for session management is to leverage a MySQL database to store session variables. A session variable keeps a value on all pages for the appropriate session handling functionality you may be leveraging, for example, within the PHP language. Using MySQL to store this information over say, a filesystem or other storage mechanism has many advantages. They include, but are not limited to:

- Platform Interoperability
- Performance & Throughput
- Efficient Locking and Transactional Support
- Scalability
- Load Balancing
- Security
- High Availability and MySQL Cluster
- Managing Session Data & Timeouts
- A Unique and Customized End-User Experience

3.1 Platform Interoperability

Whatever the platform you have chosen to host your application on, MySQL is very likely to support it. A short listing of supported platforms include:

- Red Hat Linux
- SuSE Linux
- Solaris
- Windows
- HP-UX

The other benefit of leveraging MySQL is its support for various languages. So, whether you are using PHP, .Net, Java or Perl, MySQL offers support and connectors for interfacing with the database.

The very popular and well-documented session management approach involving the open-source LAMP stack (Linux, Apache, MySQL and PHP) is by default, a MySQL "sure-fit". This allows a session management strategy which takes full advantage of LAMP's open-source characteristics and lower TCO within all the components comprising the architecture. A basic session management architecture is illustrated below in Figure 1.

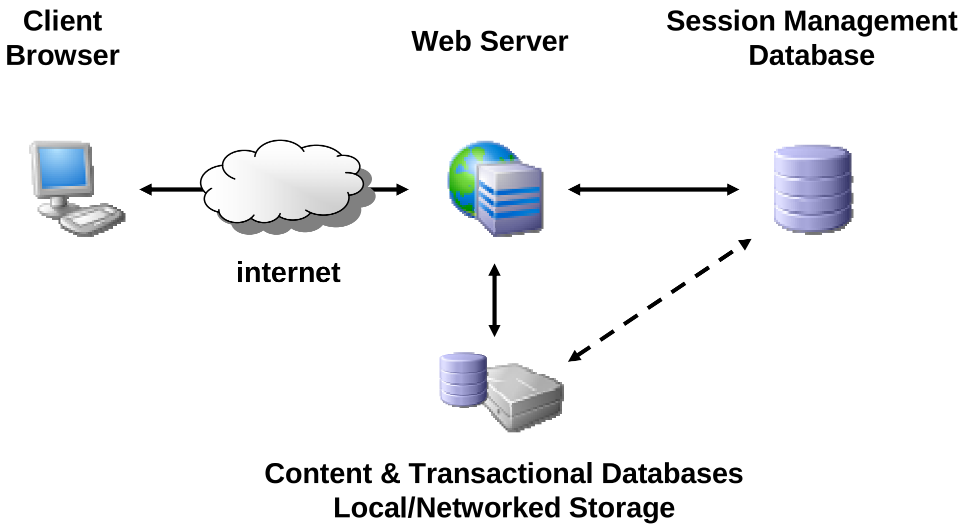

Client Browser
Web Server
Session Management Database
internet
Content & Transactional Databases Local/Networked Storage

Figure 1

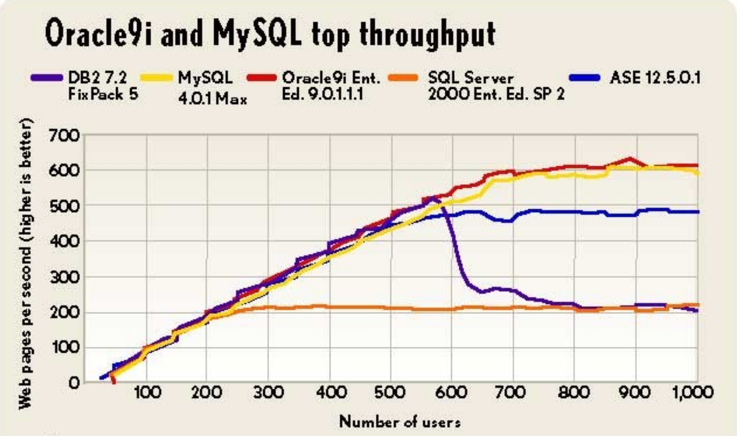
3.2 Performance & Throughput

Because of the stateless nature of the HTTP protocol, an application may need to repeatedly request a resource taxing operation from the server.

For example, a user may be at a banking or financial website that requires complex computations and the reconciliation of various accounts or ledgers to be presented on several resultant pages. This in turn may create and execute many complex SQL statements on various databases or call mathematical operations and functions on the application server. An application that uses a session variable can reuse the results that have already been requested by the user, so that this performance hit does not have to occur more then is necessary.

Using MySQL over a traditional file system, or even proprietary/closed-source databases to store these session variables, can give a better overall performance boost to your session management infrastructure.

In Figure 2, is a graphic depicting the performance characteristics of MySQL against proprietary/closed source databases published by eWeek in “Server Databases Clash”, February 25, 2002.



Throughput is in returned Web pages per second from the application server. Number of users is number of concurrent Web clients driving the load. Response time is the time to complete the six bookstore user action sequences, weighted by frequency of each sequence in the mix. All tests were conducted on an HP NetServer LT 6000r with four 700MHz Xeon CPUs, 2GB of RAM, a Gigabit Ethernet Intel Corp. Pro/1000 F Server Adapter and 24 9.1GB Ultra3 SCSI hard drives used for database storage.

Figure 2

For more information about MySQL performance characteristics, please see: http://www.mysql.com/why-mysql/benchmarks/

Regardless if transactional integrity is required for this potentially highly transient data, MySQL offers a storage engine to meet the needs of the session management design. MySQL ships with several storage engines with differing characteristics. MyISAM, InnoDB, and NDB are popular choices for session management. Some of their more notable traits are listed below:

- **MyISAM:** High Performance, Efficient Storage
- **InnoDB:** Transactional Support, Row-Level Locking
- **NDB:** Clustered - High Availability, Memory and Disk Storage Support, Transactional Support

Another reason to use MySQL over a filesystem is the amount of I/O that may be generated by requests. Individual reads and writes to a file system can result in a bottleneck that may impact the entire website. Out of the box implementations of PHP's session variable handling, for example, relies on a file system. This may be fine for supporting a small amount of users whose session requests are limited and the number of concurrent sessions is low. But the potential for creating a bottleneck rises as more sessions need to be stored and modified, and more requests are issued as a result. Fortunately, this is merely a default, and PHP can easily accommodate MySQL as a session management store.

3.3 Efficient Locking & Transactional Support

When using a file-based approach to session management, one of the bottlenecks that may surface is the inability for the filesystem to provide an efficient locking mechanism. MySQL offers several storage engines which are natively more efficient at handling locks, offering support for various transactional levels if the application requires it. Inserts, updates, deletes and concurrent access will be far more scalable and robust using MySQL within your session management system, then a file system.

3.4 Scalability & Load Balancing

An application that uses sessions is inherently limited, in so much that, each HTTP request will have to be processed in the context of the session variables for which that request belongs. Information about the state is recorded, because the result of one request needs to be available to the subsequent requests. Typically, applications that implement sessions store their session variables on the web server. For example, a session is created when a user browses an initial page on your website, all the subsequent requests, like further browsing, searching, adding/deleting items in a cart, or logging in or out, must be processed on the web server that holds the session variables. This architecture prevents many applications from using HTTP to distribute the requests across multiple web servers and therefore cannot easily implement a scale-out methodology in order to handle ever increasing numbers of requests. A diagram of an application that stores session variables on the web server is depicted in Figure 4.

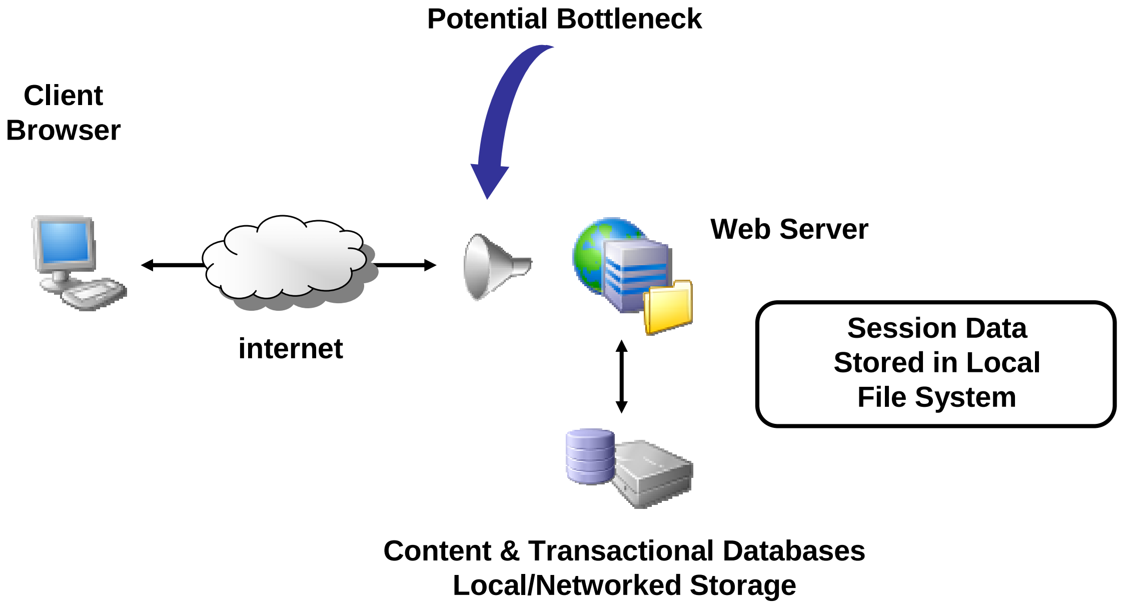

Potential Bottleneck
Client Browser
Web Server
internet
Session Data Stored in Local File System
Content & Transactional Databases Local/Networked Storage

Figure 4

Two basic strategies exist to help balance the load between the processing components of the system: static and dynamic load balancing.

Static load balancing methods determine the performance of the processing or “worker” components at the beginning. Depending on their performance the load is distributed and assigned. Static load balancing has only little overhead in communication and run-time complexity thus it is quite fast. The disadvantage is the inability to adjust work distribution during run-time.

Dynamic load balancers determine the distribution of load at run-time. The balancer distributes the work load depending on the most recent information collected. Many algorithms exist, but they all have to measure and monitor the performance of each processing component in some way to change the distribution of the work load. One of the most common dynamic load balancing strategies is the monitoring of network traffic to individual processing components. Based on the amount of traffic to each single “worker”, the work load distribution is adjusted at run-time.

A strategy one can employ that will allow an application to leverage multiple web servers is to store these session variables in a MySQL database. MySQL enables the ability to scale-out with this technique as illustrated below in Figure 5:

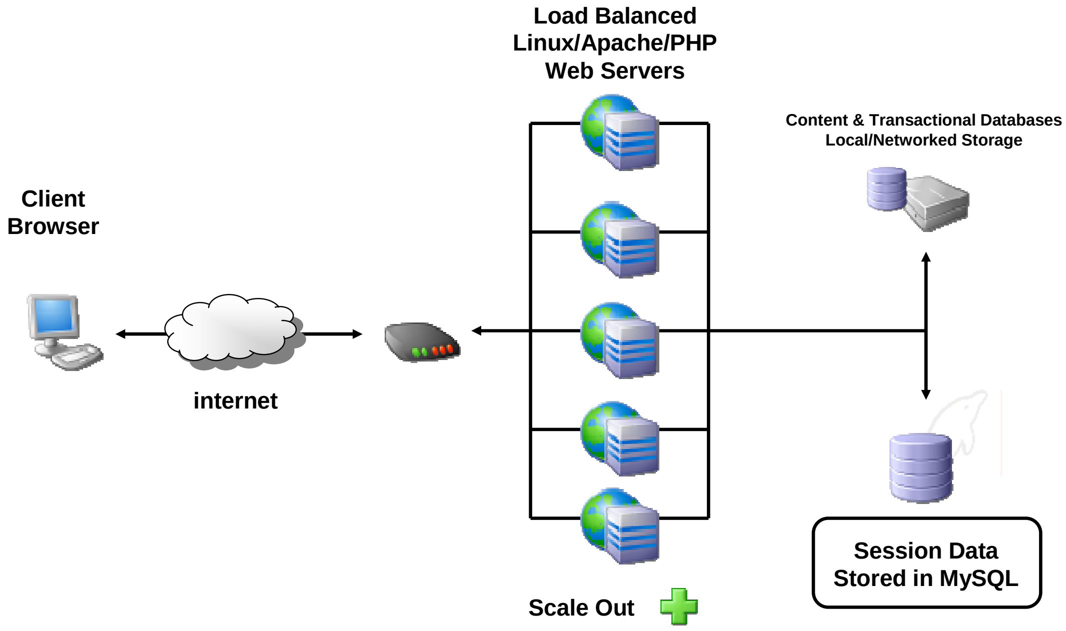


Figure 5

MySQL provides the level of scalability demanded by some of the world's most heavily trafficked web sites, not just for session management, but also for other application components, like order databases, transactional and business intelligence/data warehousing. The ability to adopt a scale-out architecture using LAMP components means that commodity hardware in conjunction with open-source software can allow your application to scale and still keep costs under control. Another benefit is that with MySQL's optional, easy to use, replication, another degree of scalability can be achieved for fail-over, performance or maintenance operations.

3.5 Security

The introduction of sessions into a web architecture can in and of themselves provide a way for a hacker to break into the system. Ineffective controls on sessions are often claimed to be the most vulnerable area of a web-based application. There are various types of session-based attacks and many of these can be characterized as impersonation attacks, also known as "session hijacking". In this type of an attack, the attacker attempts to access another user's session by posing as that user. For example, hijacking a user's session could permit a user to view saved settings which may include personal or sensitive information, or make posts in a forum under the assumed identity. At the very least, these types of attacks require that the malicious user obtain a valid session identifier, because this is the minimum amount of information that must be used for identification.

There are at least three ways that a valid session identifier can be obtained by an attacker:

- Prediction
- Capture
- Fixation

A "prediction" attack involves guessing a valid session identifier to gain access. Note, that the PHP language has native session handling that makes this type of attack inherently difficult.

A “capture” attack is little easier and therefore much more common. This can be brought about when cookies are used to store session identifiers. Therefore, a web browsers weaknesses can be exploited to obtain this information.

A session “fixation” attack tricks the victim into using a session identifier chosen by the attacker. If successful, it represents the simplest method with which a valid session identifier can be obtained.

Using a MySQL database as part of your session management implementation can provide an added layer of security by moving these session identifiers to the database layer. Although using a database as a session store does not ensure that your application cannot be hacked, it becomes much more difficult and complex to do so.

MySQL comes with an evolved set of security features built right in. These include:

- An ACL-based security model
- Support for yaSSL
- Data encryption
- Fine-grained access system from hosts and IPs down to the column-level
- A privilege system for managing the ability to connect and perform data operations
- Connection and statement logging

3.6 High Availability and MySQL Cluster

Sometimes the session management architecture demands that the database being employed also approach five 9’s of availability, without compromising performance or budgets in the process. Typically expensive closed-source clustering solutions or ever larger SMP systems are used to achieve availability and scalability requirements. With MySQL Cluster, one can effectively enable scale out, achieve high availability, retain high-performance characteristics and still only cost a fraction of what comparable closed source systems would cost to implement.

MySQL Cluster is a high-end version of the MySQL database specifically designed for fault-tolerant, mission-critical applications that need scalability and strict high-availability. It supports the ability to store data in memory or on disk, and utilizes a shared-nothing approach to clustering, enabling scale-out without requiring expensive shared storage. In Figure 6 is an example of a session management architecture that leverages MySQL Cluster as a session store.

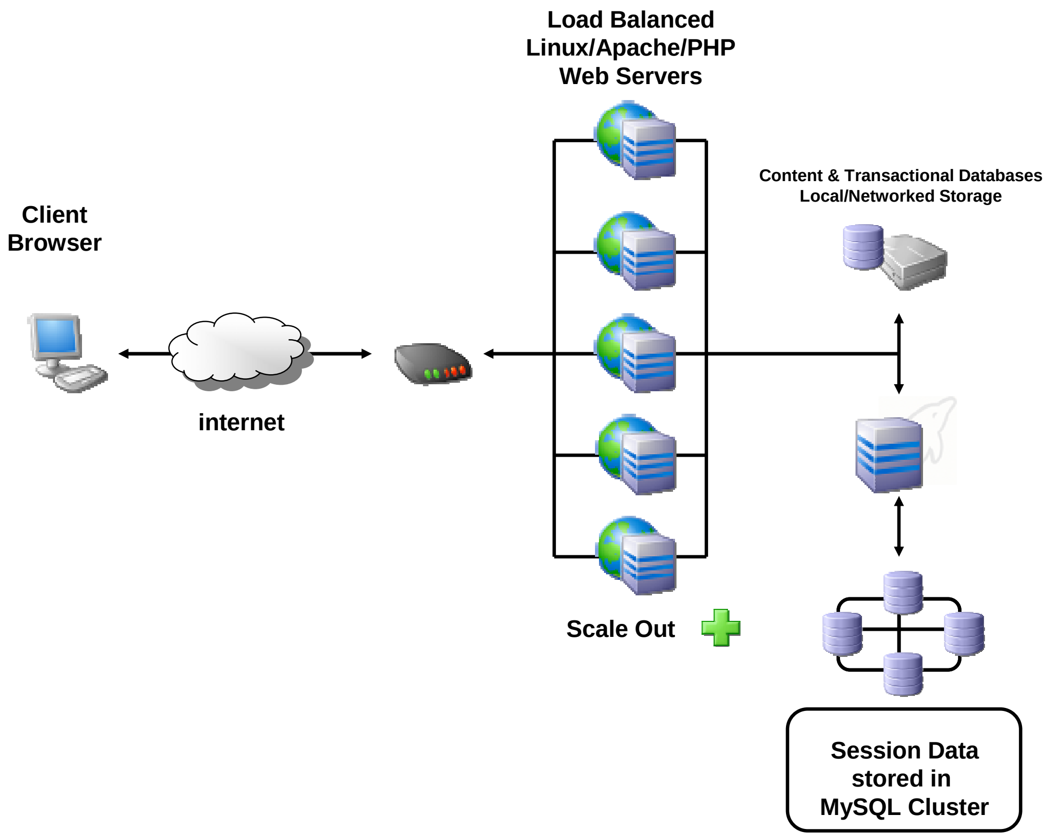


Figure 6

3.7 Managing Session Data and Timeouts

Another consideration when working with sessions is that their use can lead to synchronization problems. Because of the stateless nature of HTTP, there is no way of knowing when a user has really finished with the application. In contrast, applications that do not make use of the HTTP protocol can catch the fact that a connection has been dropped and clean up the state that was held on behalf of that user even if the user forgets to use a logout routine, which can often be the case. However, when a user is conducting a session over the Web, the server has no way of knowing if the user has ended his session accidentally or on purpose. Therefore a decision has to be made as too how long to keep the data about the session.

With the introduction of stored procedures in MySQL 5.0, we now provide an effective way of dealing with the routine analyzing and deletion of session data and managing timeouts. The key benefit is that it provides a more efficient way of interacting with the database since you don't have to prepare and execute several distinct queries to perform the same operation. (It should be noted that Perl, PHP, Java or .NET can be leveraged in a session management architecture with or without the use of stored procedures.)

Within the MySQL database we could create tables to hold the following information about our sessions:

- Variables – the name of the variable and the it's value. For example, an idle sessions timeout variable in seconds
- Users – store a users email, name, password and password, last login and access times
- Sessions – store cookie, user id, session start and access times

- Logins – this may include a history of past logins and login failures

We could then use stored procedures to minimize round trips and impact to the database in the following capacities:

- Login – check login credentials
- Cookies – return a user id based on a cookie
- Password – change or create logins and passwords
- Session Maintenance – delete old sessions and historically archive them

An illustration showing the calling of stored procedures within a MySQL database is depicted in Figure 7.

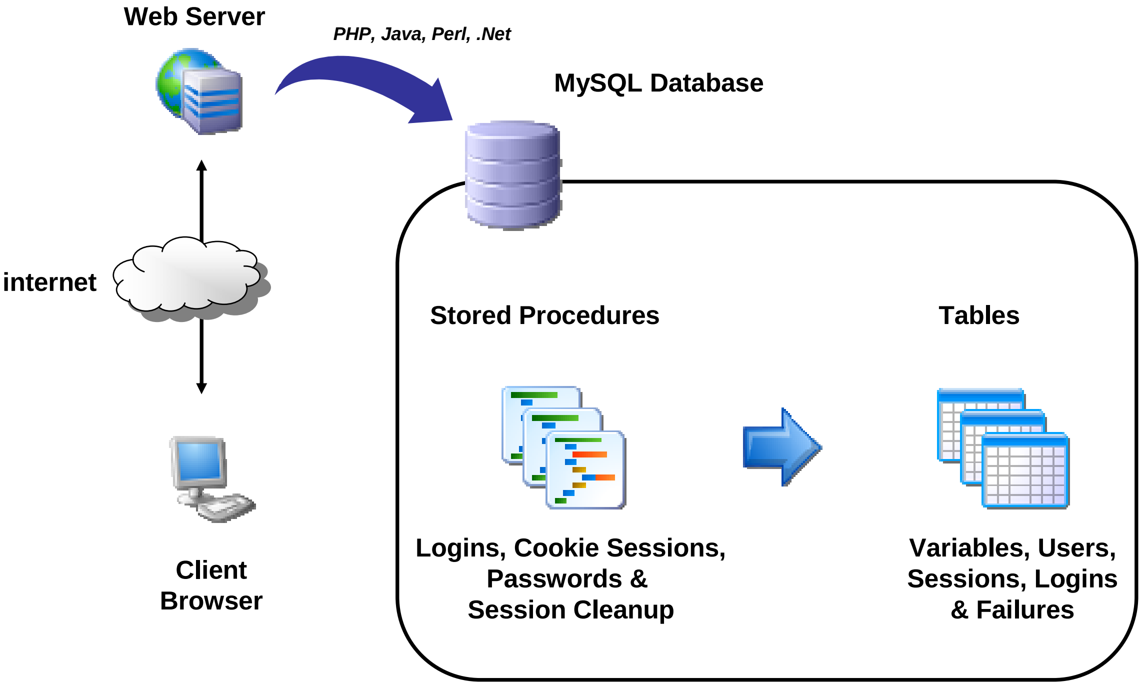

Web Server
PHP, Java, Perl, .Net
MySQL Database
internet
Stored Procedures
Tables
Logins, Cookie Sessions, Passwords & Session Cleanup
Variables, Users, Sessions, Logins & Failures
Client Browser

Figure 7

3.8 A Customized & Unique End User Experience

Sessions can be leveraged to create a customized and unique end user experience for those visiting the web site. This can materialize in the creation of pages that appeal directly to the user's interests. Whether it be to highlight products or advertisements they may be interested in, page presentation formats, or alerting them to specials or promotions. Information about sessions not only leads to better overall customer satisfaction, but increases the likelihood they will develop loyalty and return to purchase additional goods and services. No matter how complex or volatile the session data you intend to store, MySQL provides an excellent storage medium in any case. As we will see in a case study with Evite, the level of personalization that is necessary to satisfy customers could not be achieved without MySQL.

4 Evite Relies on MySQL for Session Management

Launched in 1998, Evite is a free online event planning service from InterActiveCorp, one of the world's leading interactive media companies. The Evite Web site serves 6 million users and delivers up to 9 million invitations each month. Evite's great success has resulted in phenomenal traffic growth of 80% year over year.

Evite had been using Oracle for four years to try to meet all of their data management requirements. However, they found that the Oracle software was very expensive and difficult for less-experienced staff to use. The company also needed a database that delivered the performance and scalability to meet their exponential growth in traffic. After only one year, MySQL has become a key part of the IT infrastructure of Evite, taking the place of expensive, proprietary solutions. MySQL sits behind nearly every page of the Evite website and drives business-critical data warehousing, tracking and session management applications.

Evite users can create personalized online invitations. Users can view local events that match specific interests, maintain a personalized address book, and share photos making it a highly customized and rich experience. Evite enables this by using MySQL to track the state of each users Evite session. Any of Evite's web servers can participate in any user session without data or state loss, regardless of problems with any particular web server or group of servers.

After briefly experimenting with Oracle, Evite chose MySQL for its superior performance MySQL flawlessly answers over 10 million queries during peak hours, servers more then 5 million pages a day, and has allowed Evite to cost-effectively scale traffic growth of 80% per year.

5 Neckermann.de and MySQL Cluster

KarstadtQuelle AG's e-commerce subsidiary neckermann.de replaced its legacy database system with MySQL Cluster. When neckermann.de began planning a new platform for the session data for its online store, the company's focus was on lowering operating costs and increasing flexibility. The German e-retailer found what it was looking for in the combination of MySQL Cluster running on Linux and commodity Intel-based blade server hardware.

When it came time to implement the new solution, neckermann.de employed MySQL AB's professional service organization to provide assistance in integrating MySQL Cluster into their application, including helping tune and optimize performance.

"With the help of MySQL consultants, we were able to set up a high-performing database system in a short period of time that meets our demanding requirements," stated Michael Sorg, head of new media for neckermann.de GmbH. "Administration of the session data requires the database to quickly and reliably process frequent, repeated read and write access requests. We have now successfully completed the MySQL Cluster deployment for neckermann.de and will proceed to roll this solution out for our other subsidiaries one by one."

In order to respond more effectively and rapidly to the needs of its growing number of customers, neckermann.de sought to replace its current scale-up strategy using multiprocessor systems with a flexible scale-out solution based on clustered commodity blade servers. To achieve this, the company successfully evaluated MySQL Cluster.

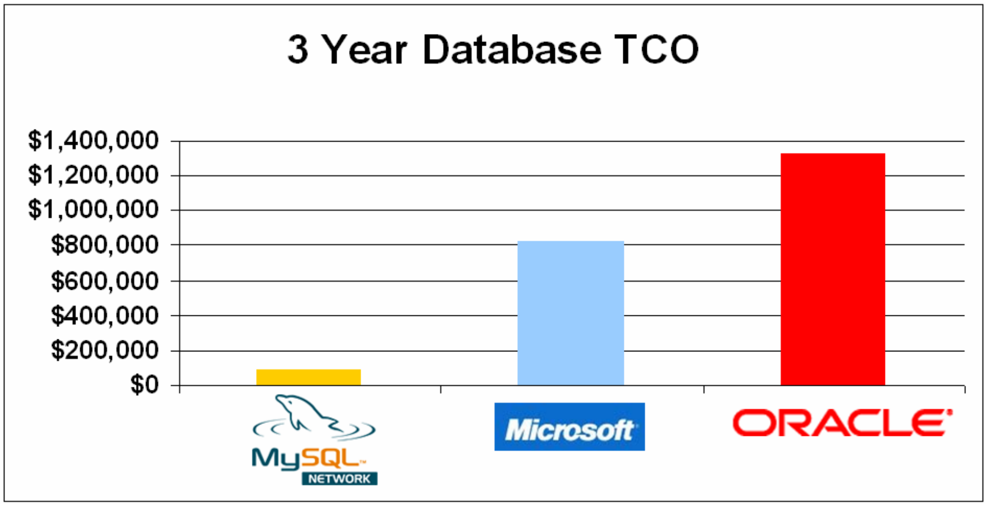
Neckermann.de's original 24-way Unix servers have been replaced with dual-processor blade servers based on Intel technology. MySQL Cluster is the main management system for session data administration.

6 Why MySQL?

MySQL provides an ideal feature set for companies looking to incorporate a database in their session management architecture.

Feature	Benefit
Pluggable Storage Engine Architecture	We won't lock you into any particular way data should be stored and the characteristics of that container. You decide what storage engine will meet the demands of the your application and end-users based on your session management strategy.
Performance	As we have see in Figure 2, MySQL is a high performance database, even when stacked against proprietary RDBMS vendors. Whether you need in-memory or disk based data, lower I/O or just plain speed at a lower cost, MySQL has the capacity to deliver.
Transactions	Whatever the transactional characteristics the database which stores your session data needs to have, we have the appropriate locking and isolation levels to satisfy the needs of your application.
Replication	Using MySQL master-slave replication, organizations are able to cost-effectively provide the scalability and the high availability necessary to accommodate the exponentially growing capacity demands. MySQL Replication is easy to set up and allows for complex chained replication topologies to achieve huge scalability.
Query Cache	Sessions may often need to repeatedly run the same queries on the same data set, returning the same results each time. The MySQL Query Cache caches result sets to improve performance by eliminating the overhead of running through the data over and over.
Optimized Connection Handling	MySQL connection handling is web optimized, providing very high performance, and is very scalable. MySQL does not require complex connection pooling to deliver scalability.
Security	An evolved set of security features including data encryption, SSL support and fine grained access control from host to column.
Clustering – High Availability	Increase the availability of your session data to users and web servers using MySQL Cluster. High availability and performance at a fraction of the price of traditional clustering solutions.

7 TCO comparison: MySQL vs. MS SQL Server vs. Oracle



Sample Hardware Configuration:

Number of Server Machines:	10 (commodity x86 machines)
Number of CPUs (Per Machine):	2 (Dual CPU)
Number of Cores (Per CPU):	2 (Dual Core)
Total CPUs in deployment:	20
Total Cores in deployment:	40

Microsoft SQL Server Enterprise:

Software License:	$24,995 Per CPU
Annual Support and Maintenance:	22% of Software License

Oracle Enterprise:

Software License:	$40,000 Per Core
Annual Support and Maintenance:	22% of Software License

MySQL Network Gold:

Software License:	NONE *(MySQL eliminates the Database License Tax*)
Annual Network Subscription:	$2,995 Per Server

3 Year TCO Comparison

	MySQL NETWORK	Microsoft	ORACLE
Software License	$0	$499,900	$800,000
3 Year Support & Maintenance	$89,850	$329,934	$528,000
Total Cost	$89,850	$829,834	$1,328,000
TCO Savings ($) using MySQL		$739,984	$1,238,150
TCO Savings (%) using MySQL		89%	93%

8 Conclusion

As we have seen, there are many reasons to consider moving your session handling and data, off the client and web server to the database tier. MySQL is an open source database solution which delivers high-performance, scalability and low administrative overhead, in whatever platform, language or technology stack you choose to deploy or are currently invested in. MySQL helps your company deliver a customized, rich and better then expected experience for end-users. It also enables your business to adopt a scale-out methodology, using commodity components and open-source software. All this means lower costs to you and a higher return on investment for your company’s web-based technology investments.

9 About MySQL

MySQL AB develops, markets, and supports a family of high performance, affordable database servers and tools. The company's flagship product is MySQL, the world's most popular open source database, with more than six million active installations. Many of the world's largest organizations, including Google, Sabre Holdings, The Associated Press, Suzuki and NASA, are realizing significant cost savings by using MySQL to power web sites, business-critical enterprise applications and packaged software. For more information about MySQL, please go to http://www.mysql.com/

MySQL Network

MySQL® Network™ provides a comprehensive set of enterprise-grade software, support and services directly from the developers of MySQL to ensure the highest levels of reliability, security and uptime. As a proactive service that helps you eliminate problems before they occur, MySQL Network gives you everything you need in a single, unified offering to successfully develop and deploy business critical applications using MySQL. To learn more, visit http://www.mysql.com/network/

MySQL Consulting

MySQL AB offers a full range of consulting services. Whether you are starting a new project, needing to optimize an existing MySQL application, or migrating from a proprietary database to MySQL, we have an affordable solution for you. Using industry best practices and proven methodologies, your MySQL certified consultant will help you deliver on-time and on-budget. To see the complete list of consulting offerings, visit http://www.mysql.com/consulting/

MySQL High Availability Solution
MySQL senior consultants use proven methodologies and expertise in database Clustering, Replication, Fail-Over, Fault-Tolerance and other HA techniques to work closely with your team to deliver a High Availability MySQL solution. This solution covers the process of designing, testing, deploying and operating a modern database infrastructure that delivers on your precise requirements for performance and high-availability.

MySQL Remote DBA Solution
A dedicated MySQL Remote DBA will work closely with your DBA team to design, install, configure, tune, secure, monitor, manage and administer your MySQL databases. MySQL senior consultants use proven methodologies and expertise in database Administration, Security, Performance Tuning, Replication, Backup, and Fault-Tolerance to ensure success.

MySQL Training

MySQL offers a comprehensive set of MySQL training courses. To see the complete list of courses and schedules, visit http://www.mysql.com/training/

MySQL for Developers
This instructor-led, hands-on class will teach Developers techniques for building database applications on MySQL. This course covers essential SQL for data design, querying, and programming. In addition, the class will prepare students for the MySQL Developer certification.

MySQL for DBAs
This instructor led, hands-on class will teach DBAs to install, configure, tune, secure, monitor, and manage their MySQL database servers. In addition, the class will prepare students for the MySQL DBA certification.

10 Resources

White Papers

A Guide to Lower Database TCO, MySQL AB
http://www.mysql.com/tco

A Computerworld article, "MySQL Breaks Into the Data Center", revealed how MySQL has become the world's most popular open source database and why corporations intent on lowering their cost of operations are using it to further commoditize their IT infrastructure. In this white paper we'll show you how. You'll also learn how organizations such as Cox Communications, NASA, Sabre Holdings and Yahoo! have improved database reliability, performance and TCO using MySQL.

Guide to Cost-effective Database Scale-Out using MySQL
http://www.mysql.com/scaleout.html

As more and more organizations rely on the Internet, they must be able to cost-effectively increase capacity of their infrastructure without sacrificing performance in order to grow their business. MySQL enables organizations to eliminate the "database tax" associated with proprietary closed source software and reduce the Total Cost of Ownership (TCO) enabling them to more readily accommodate growing demands for database capacity.

Case Studies

Evite Relies on MySQL to Deliver Millions of Invitations
http://www.mysql.com/why-mysql/case-studies/mysql-evite-casestudy.pdf

The Evite Web site serves 6 million users and delivers up to 9 million invitations each month. Evite had been using Oracle for four years to try to meet all of their data management requirements. However, they found that the Oracle software was very expensive and difficult for less-experienced staff to use. After only one year, MySQL has become a key part of the IT infrastructure of Evite, taking the place of expensive, proprietary solutions.

MySQL Contributes up to 80% Reduction in Total Cost of Ownership for Sabre Holdings
http://www.mysql.com/why-mysql/case-studies/mysql-sabre-casestudy.pdf

Sabre Holdings uses MySQL to power its air shopping and pricing application for Internet-based travel booking. Sabre estimated a 40% TCO savings when budgeting for the $100 million project. In many cases, the ongoing TCO savings are expected to be up to 80% - twice what was estimated. This translates into tens of millions of dollars saved upfront, and millions saved annually thereafter.

Citysearch Saves Over $1 Million Using MySQL
http://www.mysql.com/why-mysql/case-studies/mysql-citysearch-casestudy.pdf

Citysearch is a leading online local search service, providing the most up-to-date information on businesses, from restaurants and retail, to travel and professional services. The Citysearch web site receives over 10 million unique visitors each month. As a result of an exponentially growing user base, the front end Oracle system could no longer keep up with the traffic coming from the Web site. By implementing replicated MySQL servers to handle the Web traffic and a terabyte of data, Citysearch was able to deliver an extremely high performance solution and save more than $1 million in database licenses and hardware costs

Press Releases

Neckermann.de Migrates to MySQL Cluster
http://www.mysql.com/news-and-events/press-release/release_2006_09.html

Scaled-Out Database & Commodity Hardware Reduce Operating Costs for Leading German e-Retailer