“Virtual MySQL Assistance” for the DBA

Enterprise Monitor, Advisors, and Query Analyzer

A MySQL® White Paper

November, 2008

Table of Contents

Introduction

For over 12 years, the MySQL database server has been the heart of data-driven applications that serve a growing and intensely demanding customer base. The “M” in the LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python), MySQL has been battle-tested by heavy transaction processing applications, terabyte-sized data warehouses, as well as high-traffic Web sites, and found to be the proven leader in open source database technology. No other open source database comes close to the popularity of the MySQL database, with over 10 million production installations existing worldwide and more than 60,000 downloads occurring daily on the MySQL web site.

Having proven itself in the bleeding-edge world of technology start-up’s, Web 2.0 and other such rapidly-advancing companies, MySQL is now gaining wide acceptance in enterprises that have traditionally only used proprietary database software to handle their information management needs. More modern enterprises, deemed Enterprise 2.0, are discovering the incredible advantage of using open source for data-driven, online applications throughout their IT infrastructure. With this fast-increasing adoption rate, MySQL has evolved to become an enterprise-class database complete with must-have features and a supporting system of must-have services and production support that successful organizations demand.

While the adoption rate of the LAMP architecture continues to grow in momentum, it poses some unique challenges for traditional DBAs or application owners and the Enterprise organizations they work for. A Forrester survey of CIOs reveals that 74% have plans to adopt open source software (OSS) as part of their on-going data management strategy. Interestingly, these same CIOs expressed concerns regarding their current staffing levels and the availability of skilled DBAs to manage the technology[1]. Forrester has also reported that enterprise data is growing at a clip of 48% a year and that the ratio of DBA to managed server that stands at 1:21 is expected to be at 1:30 by the year 2010.[2] Understandably, the lower cost of scaling-out with MySQL has to be counter-balanced with a support mechanism designed to scale at the same rate. In short, as the LAMP architecture expands deeper into the corporate data center, DBA productivity and core competency with MySQL, specifically as it relates to scale-out with MySQL Replication, will need to scale at a much faster rate to allow for a proactive versus a reactive approach to data management.

To help DBAs meet this challenge, MySQL Enterprise provides the MySQL Enterprise Monitor. Whether they are working for large companies who are scaling out on MySQL, or for small businesses with only a handful of MySQL servers, the MySQL Enterprise Monitor is designed to scale DBA resources to include MySQL expertise by providing a unified, informed view into the health, security, performance and availability of the entire MySQL server environment. This paper explores the MySQL Enterprise Monitor in detail and explains how it can be leveraged as a “Virtual MySQL DBA” assistant to help over-extended DBAs proactively manage more MySQL servers with less time and effort.

Who is the MySQL DBA?

It is a fair bet to say that most MySQL DBAs wear many hats when it comes to supporting the database applications under their care. Further, while some larger organizations may have the luxury to split DBA responsibilities to align with database platform specific expertise, supporting MySQL is most commonly added to the already heaping plate of a DBA who is tasked with supporting one to many other proprietary database platforms. To this end, MySQL is most often not the sole responsibility of those tasked with managing it. While there are seemingly countless DBA titles for practically every data management discipline, MySQL DBAs typically fall into one of two designations:

- Operational DBA and
- Application DBA

[1] Forrester Research, February 2004.
[2] Forrester Research, April 2005

The Operational DBA

The operational DBA is most commonly responsible for support of MySQL database applications after they are released to the production environment. MySQL DBAs in this category typically work in mid to large scale organizations and have some level of specialization in regard to the proprietary database platforms they support. An example would be the Oracle or SQL Server DBA or System Administrator who initially “inherits” MySQL as a part-time support responsibility.

The operational DBA focuses on the global management aspects of the MySQL environment. The key responsibilities include hardware and software inventory, monitoring, trouble-shooting problems, performance turning, security administration, back up and recovery, implementing patches and updates and consulting with application teams (and other DBAs) on project related database implementations. For the operational DBA, performance and availability are of the utmost concern with a large percentage of their time being spent on activities directly related to ensuring production SLAs are met. A day in the life of the operational DBA reveals that most daily activities are spent trouble-shooting problems or tuning server configurations and queries in response to user demand for improved availability and faster response times.

Operational DBAs are fully engaged with the technical aspects of supporting the databases under their care, and ideally are involved in the strategic decision-making process for things such as capacity planning, infrastructure, scale-out, and server consolidations.

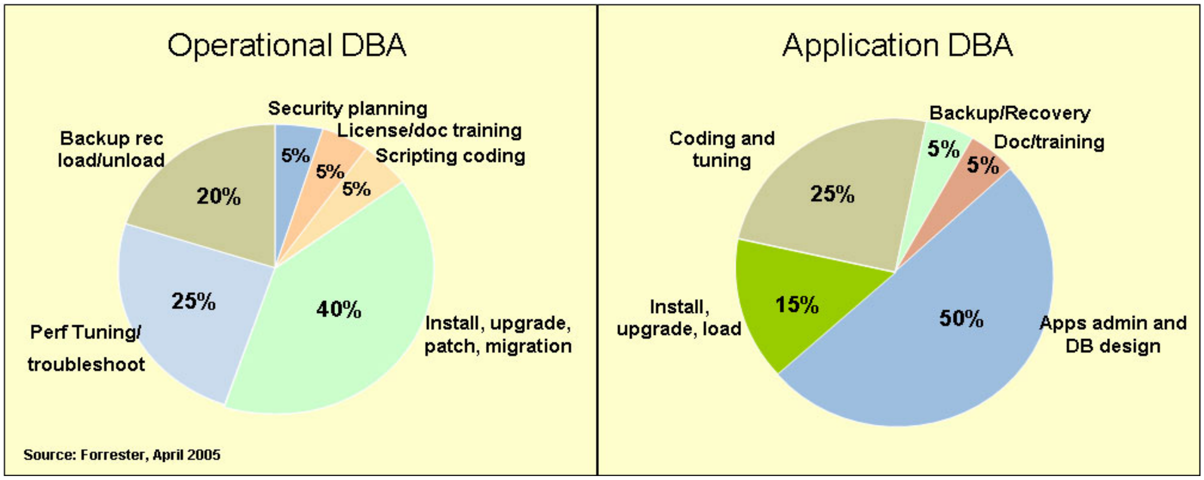
The Application DBA

Also known as a project DBA, or developer, the application DBA is responsible for designing, developing, tuning and maintaining applications that run on MySQL. Application DBAs who work for medium to large scale organizations typically have only project development related DBA responsibilities with the “heavy lifting” of supporting MySQL in production falling on the operational DBA. In smaller companies with no full-time operational DBAs it is not uncommon for the application DBA to be jack of all trades and to have both development and production level support responsibilities.

The application DBA focuses on database design, application development and the ongoing support and administration of databases for a specific set of production systems. Focusing on business critical project work, the application DBA is likely to be an expert at writing and debugging complex SQL and application code and understands the best ways to incorporate user requirements and database requests into application programs. In this role the application DBA performs most of the administration tasks that are usually reserved for the operational DBA, before, during and sometimes after the development cycle.

How Do DBAs Spend Their Time?

A common thread between the operational and application DBA is that they both spend a disproportionate amount of their time simply “keeping the lights on” when it comes to building and maintaining their database environments. As seen in the Figure 1, while operational DBAs spend 40% of their time proactively maintaining their production environments, they spend 30% of their time on scripting, performance tuning, and troubleshooting. Grouping these activities together as one very related statistic is not much of stretch when you consider that most scripting is done to monitor database security, performance and availability and most tuning that DBAs do on their production environments is done reactively in response to script diagnostics, user response time issues or production SLA violations. Further, while scripting encompasses a reported 5% of the operational DBA’s time, it is important to note there is most likely time spent every working day maintaining script code, compiling results or ensuring that key scripts are actually running. Also consider that much of the time spent on reactive performance tuning is burned identifying the “what” and “how” aspects of the specific tuning opportunity.



Source: Forrester, April 2005

Figure 1

Typically application DBA work is somewhat proactive in nature with time spent on activities that ensure the optimization of an application and the underlying database before they are introduced into the production environment. Referring again to figure 1, on average 75% of an application DBA's time, or 6 hours of a typical 8 hour work day, is spent on application administration, database design, script and application coding and database tuning related tasks. Given that most application development teams are working on multiple projects in parallel, most driven by business supporting initiatives, application DBAs are often under a great deal of pressure to shorten time to market delivery windows often at the expense of the time they spend at home or the time they devote to these key areas of application and database development.

Outside of these activities, application DBAs also maintain their own development environments, document and maintain hardware inventories, train new staff, and perform operational DBA administration tasks for their new applications until the time they are released for production implementation. As stated above, many times the application DBA shoulders the additional burden of serving as the operational DBA for the MySQL database applications that his/her team developed. In these cases, the application DBA is taxed to be proactive in designing and developing new systems, while most often maintaining a reactive approach to supporting existing production systems.

Challenges They Face

The popularity of open source software (OSS) continues to grow with Gartner reporting that by 2008 more than 70% of all organizations will be using an open source database solution to power their business applications. This trend in OSS adoption as it relates to MySQL poses a unique set of challenges for DBAs and the companies for which they work. Chief among these challenges are:

Scale-Out means more MySQL servers and data assets to manage

A recent DBMS study conducted by Gartner reports that of the companies they researched 49% are currently using MySQL and heading into 2007, 17% have plans to deploy MySQL as an alternative to its more pricey alternatives.

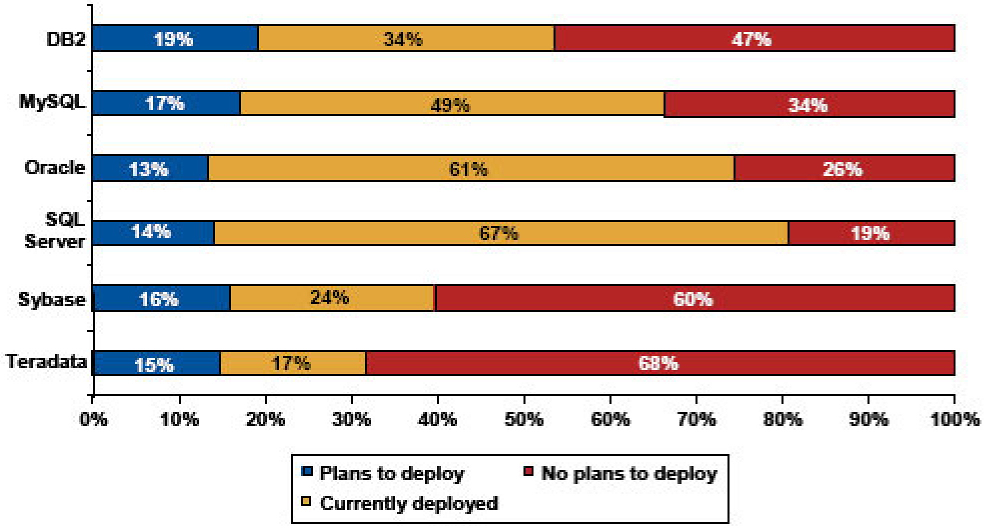


Figure 2

So what does this mean to the DBAs currently supporting MySQL? Because MySQL runs and performs well on commodity hardware, these organizations are enabled to scale-out via MySQL Replication on inexpensive Linux-based hardware as they add new MySQL powered applications. This is a double-edged sword to the MySQL DBA; while MySQL is known for its ease-of-use, steady performance and rock-solid reliability, the sheer number of MySQL servers (both Master and Slave), each with its own administrative and support needs, will most likely grow in rabbit-like fashion. Given that many MySQL DBAs are operating at capacity with their current server and application loads, managing even more MySQL servers seems out of the realm of possibility.

Supporting these findings is a recent Forrester study that reports that today a single enterprise DBA manages 1 Terabyte of data across 21 production servers. As seen in this figure, these numbers are expected to increase at an increasing rate over the next 3 years.

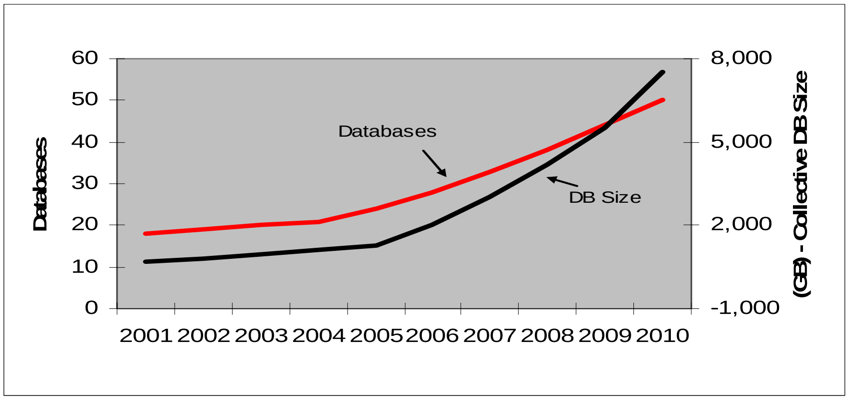


Figure 3

The demand for MySQL DBAs far exceeds the supply

Gartner reports that 58% of North American companies plan to increase their database related spending by 5% or more in 2007.[3] Assuming some of these dollars will be used for staffing those seeking MySQL specific skill sets may find stiff competition for the best and brightest. For reference, a search of www.monster.com for "MySQL DBA" returns the default max of 1,000 job postings, a number equal to the number returned for a similar search done for "Oracle DBA".

Service Level Agreements are becoming more demanding

[3] Source: Gartner DBMS Report, 2006

No list of challenges would be complete without a reminder of the production SLAs that serve as guidelines for the performance and availability of business-critical applications. While some applications can tolerate some amount of user interruption, those with SLAs are most often defined by how much revenue a company loses when those systems are down or performing badly. For web and e-commerce based applications, a single minute of downtime can cost a business between $300,000 and $1M.[4]

Companies are requiring their DBAs to be more business-minded

While database related spending is forecast to increase, there will always be increased pressure on the MySQL DBA to ensure that all production environments are tuned for optimal performance. While it's tempting to simply scale-out on commodity hardware, especially when production-ready servers can be purchased for $5,000 or less, companies will look to the MySQL DBA to smartly consolidate smaller, less critical applications where possible.

To meet these challenges companies scaling-out on MySQL need to address the pain points their DBAs face as their MySQL responsibilities grow. In short, DBAs need the following to be successful in supporting an aggressive move to MySQL:

- Visibility into the status and inventory of all of their MySQL servers, regardless of physical location
- Direction on where and how to spend their limited time
- Ability to scale their existing skill sets to include MySQL expertise
- Assistance finding problems and tuning opportunities they cannot find themselves

The balance of this paper addresses how the above needs can be met when DBAs are empowered by the proactive monitoring and manageability features found in the MySQL Enterprise Monitor.

Virtual DBA Assistant: MySQL Enterprise Monitor

Provided as part of the MySQL Enterprise subscription, the Enterprise Monitor is a "Virtual DBA" assistant that helps MySQL DBAs manage more MySQL servers in a scale-out environment, tune their current MySQL servers and find and fix problems with their MySQL database applications *before* they can become serious problems or costly outages. Running completely within the corporate firewall, the Enterprise Monitor proactively monitors enterprise database environments and provides expert advice on how even those new to MySQL can tighten security, optimize performance and reduce downtime of their MySQL powered systems.

Built on Web 2.0 Technology

Built on a Web 2.0 framework, and running fully contained within the corporate firewall, the MySQL Enterprise Monitor is powered by a distributed web-application. It is comprised on three main components: a lightweight agent that is deployed to each of your monitored MySQL servers to collect MySQL and OS specific data, a centralized server that stores and evaluates the collected data and a web-enabled client application that serves as the portal for the MySQL Enterprise Monitor user interface. The user interface is designed to provide a rich, user experience found in the most common client based applications, all from a very thin, browser based solution.

The architecture for the MySQL Enterprise Monitor and each of its components is shown here:

[4] Source: Ecommerce Times, 2006

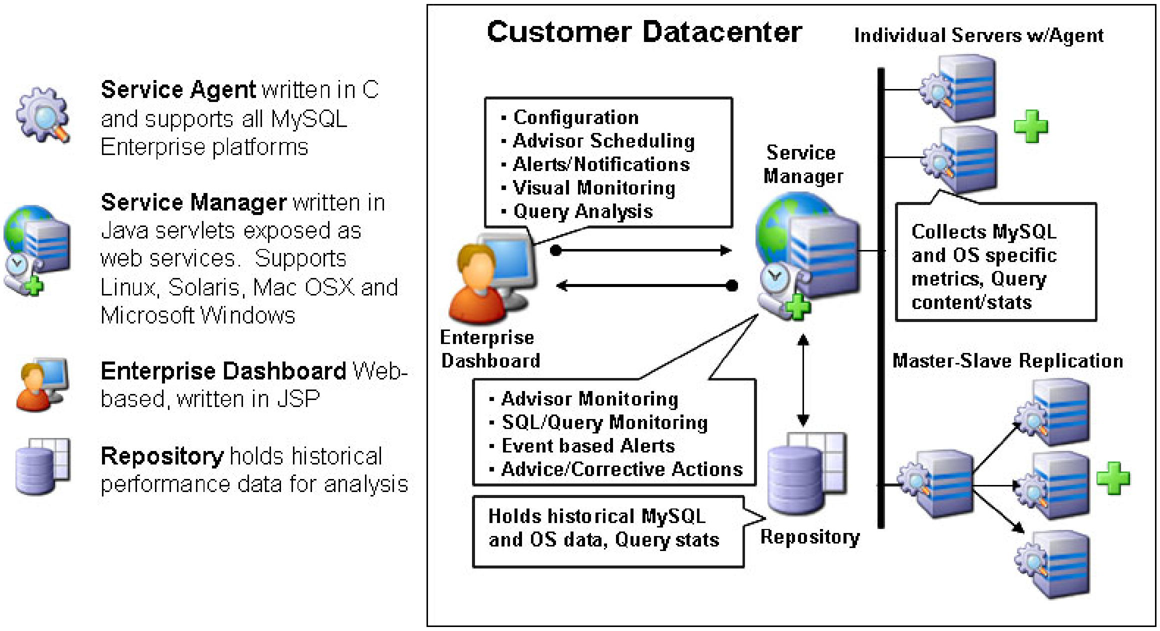

Service Agent written in C and supports all MySQL Enterprise platforms
Service Manager written in Java servlets exposed as web services. Supports Linux, Solaris, Mac OSX and Microsoft Windows
Enterprise Dashboard Web-based, written in JSP
Repository holds historical performance data for analysis
Customer Datacenter
Individual Servers w/Agent
• Configuration
• Advisor Scheduling
• Alerts/Notifications
• Visual Monitoring
• Query Analysis
Service Manager
Collects MySQL and OS specific metrics, Query content/stats
Enterprise Dashboard
Master-Slave Replication
• Advisor Monitoring
• SQL/Query Monitoring
• Event based Alerts
• Advice/Corrective Actions
Holds historical MySQL and OS data, Query stats
Repository

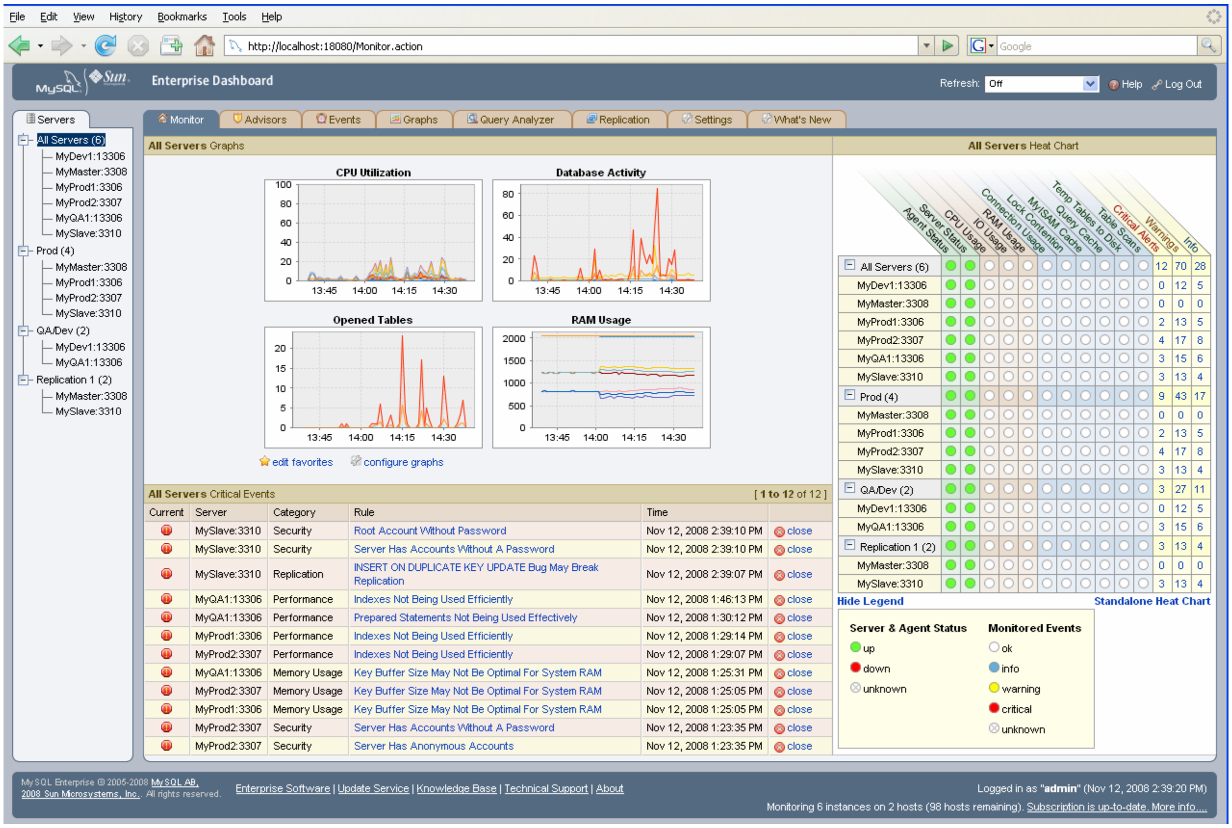
Figure 4: MySQL Enterprise Monitor Architecture

How the Enterprise Monitor Helps the DBA

The Enterprise Monitor is designed to help companies scale their existing MySQL DBA resources by providing a single, consolidated view in the health, security, performance and availability of all of their MySQL servers. It proactively monitors all MySQL servers using a set of MySQL provided expert advisors, to identify and alert DBAs of problems, security vulnerabilities and tuning opportunities so they can be acted upon well in advance of a problem or outage occurring.

Enterprise-Ready Dashboard for Monitoring All MySQL Servers

As stated earlier, one of the biggest challenges the MySQL DBA faces is managing an ever-growing number of MySQL servers and databases. Regardless of the size of the MySQL environment, each server requires specific attention when it comes to basic administration, security, performance monitoring and availability. For dynamic, replicated environments DBAs must document and maintain Master/Slave relationships to ensure they can monitor for replication related latency issues. To give the MySQL DBA a proactive advantage in all of these areas, the MySQL Enterprise Monitor provides the Enterprise Dashboard. Using the Enterprise Dashboard, DBAs can monitor MySQL and OS specific metrics for single or groups of servers, and can stay on top of all their replication topologies. The Enterprise Dashboard is designed so DBAs can easily understand the complete security, availability, and performance picture of all their MySQL servers in one place, all from a thin, browser-based console.



All Servers Critical Events [1 to 12 of 12]

Current	Server	Category	Rule	Time	
●	MySlave:3310	Security	Root Account Without Password	Nov 12, 2008 2:39:10 PM	close
●	MySlave:3310	Security	Server Has Accounts Without A Password	Nov 12, 2008 2:39:10 PM	close
●	MySlave:3310	Replication	INSERT ON DUPLICATE KEY UPDATE Bug May Break Replication	Nov 12, 2008 2:39:07 PM	close
●	MyQA1:13306	Performance	Indexes Not Being Used Efficiently	Nov 12, 2008 1:46:13 PM	close
●	MyQA1:13306	Performance	Prepared Statements Not Being Used Effectively	Nov 12, 2008 1:30:12 PM	close
●	MyProd1:3306	Performance	Indexes Not Being Used Efficiently	Nov 12, 2008 1:29:14 PM	close
●	MyProd2:3307	Performance	Indexes Not Being Used Efficiently	Nov 12, 2008 1:29:07 PM	close
●	MyQA1:13306	Memory Usage	Key Buffer Size May Not Be Optimal For System RAM	Nov 12, 2008 1:25:31 PM	close
●	MyProd2:3307	Memory Usage	Key Buffer Size May Not Be Optimal For System RAM	Nov 12, 2008 1:25:05 PM	close
●	MyProd1:3306	Memory Usage	Key Buffer Size May Not Be Optimal For System RAM	Nov 12, 2008 1:25:05 PM	close
●	MyProd2:3307	Security	Server Has Accounts Without A Password	Nov 12, 2008 1:23:35 PM	close
●	MyProd2:3307	Security	Server Has Anonymous Accounts	Nov 12, 2008 1:23:35 PM	close

All Servers Heat Chart

	Agent Status	Server Status	CPU Usage	IO Usage	RAM Usage	Connection Usage	Lock Contention	MyISAM Cache	Query Cache	Temp Tables to Disk	Table Scans	Critical Alerts	Warnings	Info
All Servers (6)	●	●	○	○	○	○	○	○	○	○	○	12	70	28
MyDev1:13306	●	●	○	○	○	○	○	○	○	○	○	0	12	5
MyMaster:3308	●	●	○	○	○	○	○	○	○	○	○	0	0	0
MyProd1:3306	●	●	○	○	○	○	○	○	○	○	○	2	13	5
MyProd2:3307	●	●	○	○	○	○	○	○	○	○	○	4	17	8
MyQA1:13306	●	●	○	○	○	○	○	○	○	○	○	3	15	6
MySlave:3310	●	●	○	○	○	○	○	○	○	○	○	3	13	4
Prod (4)	●	●	○	○	○	○	○	○	○	○	○	9	43	17
MyMaster:3308	●	●	○	○	○	○	○	○	○	○	○	0	0	0
MyProd1:3306	●	●	○	○	○	○	○	○	○	○	○	2	13	5
MyProd2:3307	●	●	○	○	○	○	○	○	○	○	○	4	17	8
MySlave:3310	●	●	○	○	○	○	○	○	○	○	○	3	13	4
QA/Dev (2)	●	●	○	○	○	○	○	○	○	○	○	3	27	11
MyDev1:13306	●	●	○	○	○	○	○	○	○	○	○	0	12	5
MyQA1:13306	●	●	○	○	○	○	○	○	○	○	○	3	15	6
Replication 1 (2)	●	●	○	○	○	○	○	○	○	○	○	3	13	4
MyMaster:3308	●	●	○	○	○	○	○	○	○	○	○	0	0	0
MySlave:3310	●	●	○	○	○	○	○	○	○	○	○	3	13	4

Hide Legend Standalone Heat Chart

Server & Agent Status	Monitored Events
● up	○ ok
● down	● info
⊗ unknown	● warning
	● critical
	⊗ unknown

MySQL Enterprise © 2005-2008 MySQL AB, 2008 Sun Microsystems, Inc.. All rights reserved.

Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as "**admin**" (Nov 12, 2008 2:39:20 PM)

Monitoring 6 instances on 2 hosts (98 hosts remaining). Subscription is up-to-date. More info....

Figure 5: MySQL Enterprise Monitor Dashboard

Auto Detection, Grouping and Monitoring of Replication/Scale-Out Topologies

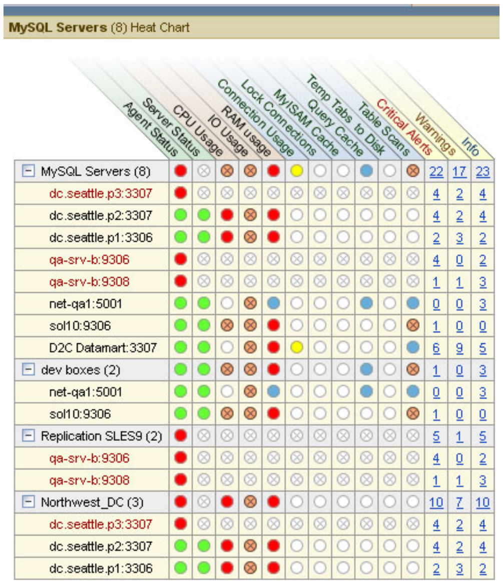
While MySQL Replication provides a robust solution for businesses leveraging MySQL as part of a scale-out versus a scale-up strategy, managing MySQL in such an environment has been a continual challenge, particularly when it comes to understanding or keeping track of master/slave hierarchical relationships. This is a common DBA pain point and can be the result of several factors:

- MySQL Replication topologies are often not documented or existing documentation is not maintained to reflect hierarchy changes and additions.
- For scale out, slaves can easily be added to an existing replication topology to keep up with demand and agreed upon service level agreements. In this scenario, status checking of new slaves is only as good as the last time the documented topology inventory was updated.
- Slaves are commonly promoted to master to correct a master/slave out of synch condition.
- DBAs new to MySQL or to an existing MySQL Replication topology may not understand the hierarchies they are inheriting and often must spend upfront time learning the relationships that are in place.

The Enterprise Dashboard makes it easier to scale out using MySQL Replication by providing industry-leading auto detection, grouping, documenting and monitoring of all MySQL Replication master/slave hierarchical relationships. Changes and additions to existing replication topologies are also auto detected and maintained providing DBAs instant visibility into newly implemented updates. This helps reduce the learning curve for DBAs new to MySQL Replication or to specific scale-out environments.

At-a-Glance Availability and Performance Diagnosis

The Enterprise Dashboard includes a color-coded Heat Chart that provides an at-a-glance view into the availability and performance of all of the MySQL servers across the enterprise. From the Heat Chart the DBA can instantly tell:



MySQL Servers (8) Heat Chart

	Agent Status	Server Status	CPU Usage	IO Usage	RAM usage	Connection Usage	Lock Connections	MyISAM Cache	Query Cache	Temp Tabs to Disk	Table Scans	Critical Alerts	Warnings	Info
⊟ MySQL Servers (8)	🔴	ⓧ	⊗	⊗	🔴	🟡	○	○	🔵	○	⊗	22	17	23
dc.seattle.p3:3307	🔴	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	4	2	4
dc.seattle.p2:3307	🟢	🟢	🔴	⊗	🔴	○	○	○	○	○	○	4	2	4
dc.seattle.p1:3306	🟢	🟢	🔴	⊗	🔴	○	○	○	○	○	○	2	3	2
qa-srv-b:9306	🔴	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	4	0	2
qa-srv-b:9308	🔴	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	1	1	3
net-qa1:5001	🟢	🟢	○	⊗	🔵	○	○	○	🔵	○	🔵	0	0	3
sol10:9306	🟢	🟢	⊗	⊗	🔴	○	○	○	○	○	⊗	1	0	0
D2C Datamart:3307	🟢	🟢	○	⊗	🔴	🟡	○	○	○	○	🔵	6	9	5
⊟ dev boxes (2)	🟢	🟢	⊗	⊗	🔴	○	○	○	🔵	○	⊗	1	0	3
net-qa1:5001	🟢	🟢	○	⊗	🔵	○	○	○	🔵	○	🔵	0	0	3
sol10:9306	🟢	🟢	⊗	⊗	🔴	○	○	○	○	○	⊗	1	0	0
⊟ Replication SLES9 (2)	🔴	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	5	1	5
qa-srv-b:9306	🔴	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	4	0	2
qa-srv-b:9308	🔴	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	1	1	3
⊟ Northwest_DC (3)	🔴	ⓧ	🔴	⊗	🔴	○	○	○	○	○	○	10	7	10
dc.seattle.p3:3307	🔴	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	4	2	4
dc.seattle.p2:3307	🟢	🟢	🔴	⊗	🔴	○	○	○	○	○	○	4	2	4
dc.seattle.p1:3306	🟢	🟢	🔴	⊗	🔴	○	○	○	○	○	○	2	3	2

- The Up/down status of all MySQL servers
- Key OS metrics that may be affecting MySQL
- What MySQL servers need attention and
- Where and how they need to spend their limited time

The Heat Chart provides smart groupings that allow a DBA to collapse or expand a particular MySQL group of servers and view only the information they need to see. Each MySQL server's availability is indicated so a DBA knows what servers are on and offline. In addition, key performance statistics are collected and measured against pre-defined MySQL best practice thresholds, with the end result being a series of colorized displays that indicate what performance metrics are out of line with acceptable limits. Finally, a summarized count of performance and best practice exceptions are displayed, with a DBA being able to easily drill down to see what exceptions have been encountered for any of their monitored servers.

Traditional MySQL Database Monitoring

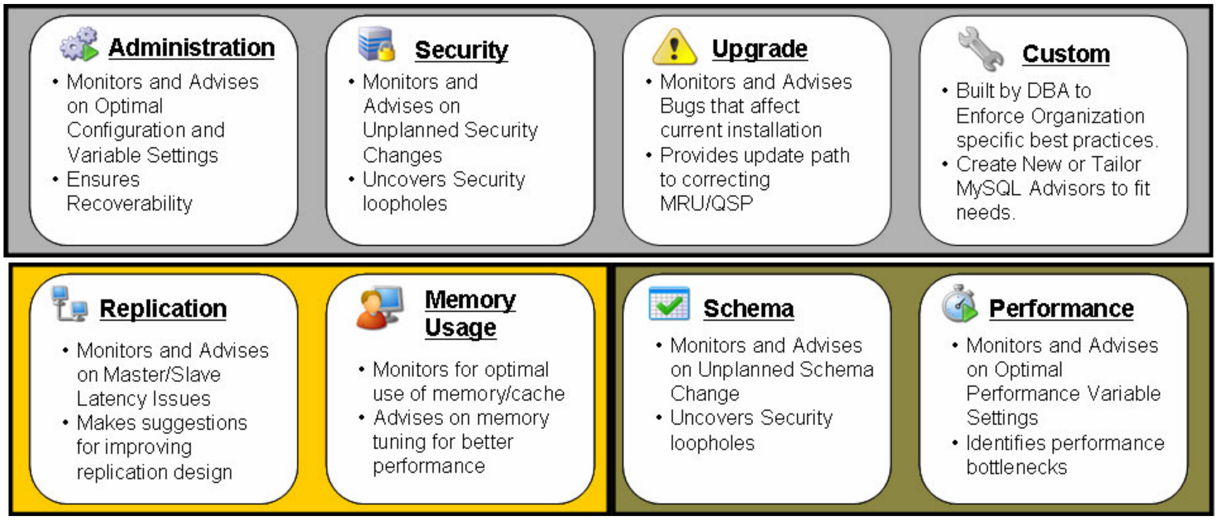
DBAs employ a variety of solutions to monitor their MySQL servers. Most commonly, monitoring is done using homegrown scripts, third party commercial products or some combination of the two. While most DBAs will agree they like using scripts because they are customizable to fit their exact needs, they will also be quick to point out that their use limits the efficiency, quality, and depth of the monitoring that can be done. All monitoring done via scripts is done in real time with no collection or retention of monitored data, so users cannot easily draw correlations between current problems and past events. Typically, script based monitoring is hampered by limited scheduling capabilities and few options for built-in notifications, so often DBAs are left to manually check results or to rely on network operation center (NOC) operators to notify them of problems.

It is far more strategic for a business to rely on a third-party tool provider to make sure their database monitoring application remains current with the latest in scheduling, notifications and versioning of the MySQL server they are monitoring. But even this is not an optimal solution as even the best of these tools are only designed to report on monitored metrics and provide little value when it comes to proactively identifying problems or recommending tuning opportunities.

The MySQL Advisors – Find and Fix Problems Before They Occur

The MySQL Enterprise Monitor differs from traditional third-party database monitors in that it supplies, right out of the box, a set of MySQL Advisors that are designed to automatically examine a MySQL server's configuration, security, and performance levels, identify problems and tuning opportunities and provide the MySQL DBA with specific corrective actions.

The MySQL Enterprise Monitor ships with the following set of MySQL Advisors:



Administration

- Monitors and Advises on Optimal Configuration and Variable Settings
- Ensures Recoverability

Security

- Monitors and Advises on Unplanned Security Changes
- Uncovers Security loopholes

Upgrade

- Monitors and Advises Bugs that affect current installation
- Provides update path to correcting MRU/QSP

Custom

- Built by DBA to Enforce Organization specific best practices.
- Create New or Tailor MySQL Advisors to fit needs.

Replication

- Monitors and Advises on Master/Slave Latency Issues
- Makes suggestions for improving replication design

Memory Usage

- Monitors for optimal use of memory/cache
- Advises on memory tuning for better performance

Schema

- Monitors and Advises on Unplanned Schema Change
- Uncovers Security loopholes

Performance

- Monitors and Advises on Optimal Performance Variable Settings
- Identifies performance bottlenecks

Figure 6: MySQL Enterprise Advisors

Each of the MySQL Advisors is designed to cover specific DBA “areas of concern” and is comprised of a set of MySQL Advisor Rules that help DBAs proactively find problems and tuning opportunities they may not have the time or expertise to find themselves.

What are the MySQL Advisor Rules?

The MySQL Advisor Rules are a set of MySQL supplied best practices that allow DBAs to implement new MySQL servers with confidence and to proactively manage the dynamic nature of all of their MySQL servers over time. The MySQL Advisor Rules do this by monitoring all MySQL servers for adherence to MySQL recommended configuration and server settings and notifying the DBA with specific instructions on how to proactively address found exceptions to align with MySQL best practices. To this end, the MySQL Enterprise Monitor employs 100+ MySQL Advisor Rules that monitor over 600 MySQL and OS specific variables and metrics that track and report on the overall health, security, availability and performance of each MySQL server.

For new servers there are specific Advisor Rules that monitor configuration variables and settings for optimal recoverability, performance, and security. From a security standpoint, the Security Advisor Rules monitor for the presence of security loopholes that leave a new MySQL server vulnerable to malicious attack. For existing MySQL servers, there are specific Memory Usage, Performance, Security, Schema and Replication Advisor Rules that track and alert on dynamic metrics, so DBAs can proactively address things like security violations, poor cache hit ratios, poorly utilized indexes or replication synch issues before they impact application performance and ultimately the end user.

For DBAs with advanced monitoring needs, the MySQL Advisors can be extended to include custom Advisor Rules that monitor existing metrics or that the DBA opts to collect from the MySQL server. DBAs can easily copy and edit the MySQL provided Advisor Rules to better meet their organization’s specific needs. They can also create Rules and evaluated expressions from scratch using the MySQL provided template and the MySQL and OS variables and counters collected by the MySQL Enterprise Monitor.

MySQL Advisor Rules – Productivity through Automation

The MySQL Enterprise Monitor makes MySQL DBAs more productive by allowing them to automate each the MySQL Advisor Rules for unattended, around the clock operations. This helps minimize human errors, improves overall productivity and lowers the total cost of associated with managing MySQL.

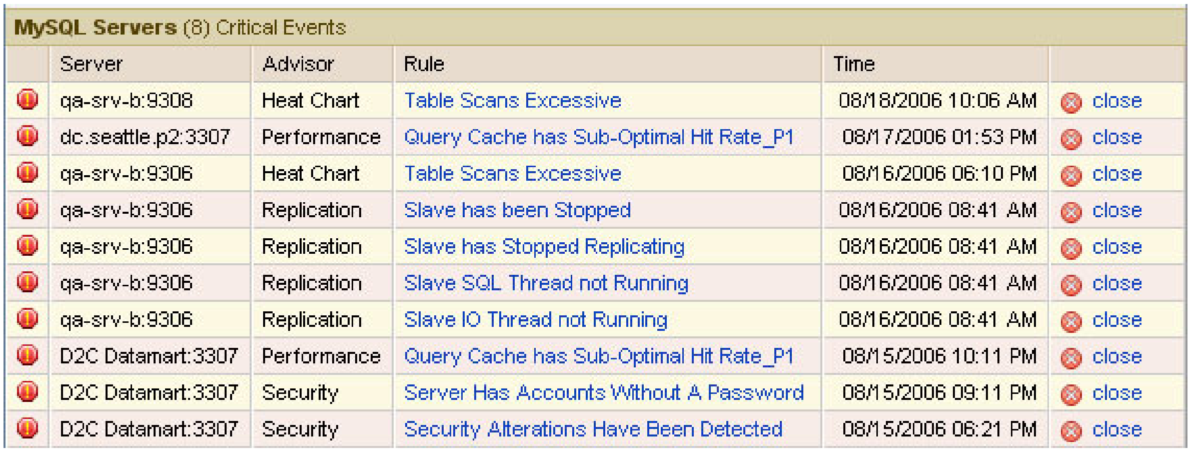
Forrester estimates that enterprises can typically save 15% or more of their total cost of database ownership by way of these reduced administrative efforts.[5]

Enabled via “set it and forget it” scheduling options the MySQL Advisor Rules provide immediate results upon implementation by identifying existing problems with your MySQL servers and continue to work to ensure that even optimized systems remain at peak performance.

Customized Thresholds and Alerts

Each of the MySQL Advisor rules allows the MySQL DBA to customize the thresholds that are acceptable for specific MySQL servers. As an example, a DBA using the MySQL supplied Performance Advisor Rule “Query Cache has Sub-Optimal Hit Ratio” may use higher threshold values for their MySQL servers running customer-facing OLTP applications, while lower thresholds may be acceptable for less-critical intranet-based applications. Additionally, thresholds can be designated to trigger Informational, Severe or Critical severity level alerts allowing DBAs to assess whether a reported issue requires immediate attention.

For all of the MySQL Advisors, Advisor Rule violations trigger notification events that are sent to the Enterprise Dashboard and to customized groups via SMTP and/or SNMP notification options. To help DBAs quickly determine where to spend their time the severity of each event is based on the alert level of the violated threshold. Using the Enterprise Dashboard, DBAs can filter by event severity and by server to easily identify the MySQL servers with issues that demand immediate attention.



MySQL Servers (8) Critical Events

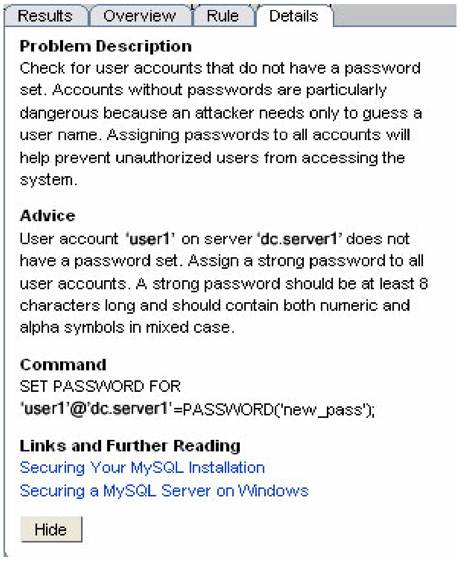
	Server	Advisor	Rule	Time	
!	qa-srv-b:9308	Heat Chart	Table Scans Excessive	08/18/2006 10:06 AM	⊗ close
!	dc.seattle.p2:3307	Performance	Query Cache has Sub-Optimal Hit Rate_P1	08/17/2006 01:53 PM	⊗ close
!	qa-srv-b:9306	Heat Chart	Table Scans Excessive	08/16/2006 06:10 PM	⊗ close
!	qa-srv-b:9306	Replication	Slave has been Stopped	08/16/2006 08:41 AM	⊗ close
!	qa-srv-b:9306	Replication	Slave has Stopped Replicating	08/16/2006 08:41 AM	⊗ close
!	qa-srv-b:9306	Replication	Slave SQL Thread not Running	08/16/2006 08:41 AM	⊗ close
!	qa-srv-b:9306	Replication	Slave IO Thread not Running	08/16/2006 08:41 AM	⊗ close
!	D2C Datamart:3307	Performance	Query Cache has Sub-Optimal Hit Rate_P1	08/15/2006 10:11 PM	⊗ close
!	D2C Datamart:3307	Security	Server Has Accounts Without A Password	08/15/2006 09:11 PM	⊗ close
!	D2C Datamart:3307	Security	Security Alterations Have Been Detected	08/15/2006 06:21 PM	⊗ close

Figure 7: Events Tab

Expert Advice Straight from the Pros at MySQL

In addition to notifying the DBA of MySQL best practice security, design, and performance best practice violations, the MySQL Enterprise Monitor also provides expert advice on the specific problem that has been reported. Each Advisor Rule supplies tailored recommendations, straight from the engineers at MySQL, that communicate in step-by-step fashion how to go about rectifying a particular problem.

Advice from the MySQL Enterprise Monitor is available via drill downs in the Enterprise Dashboard or through customized notifications that can be sent to individual or



Problem Description

Check for user accounts that do not have a password set. Accounts without passwords are particularly dangerous because an attacker needs only to guess a user name. Assigning passwords to all accounts will help prevent unauthorized users from accessing the system.

Advice

User account **‘user1’** on server **‘dc.server1’** does not have a password set. Assign a strong password to all user accounts. A strong password should be at least 8 characters long and should contain both numeric and alpha symbols in mixed case.

Command

```
SET PASSWORD FOR
'user1'@'dc.server1'=PASSWORD('new_pass');
```

Links and Further Reading

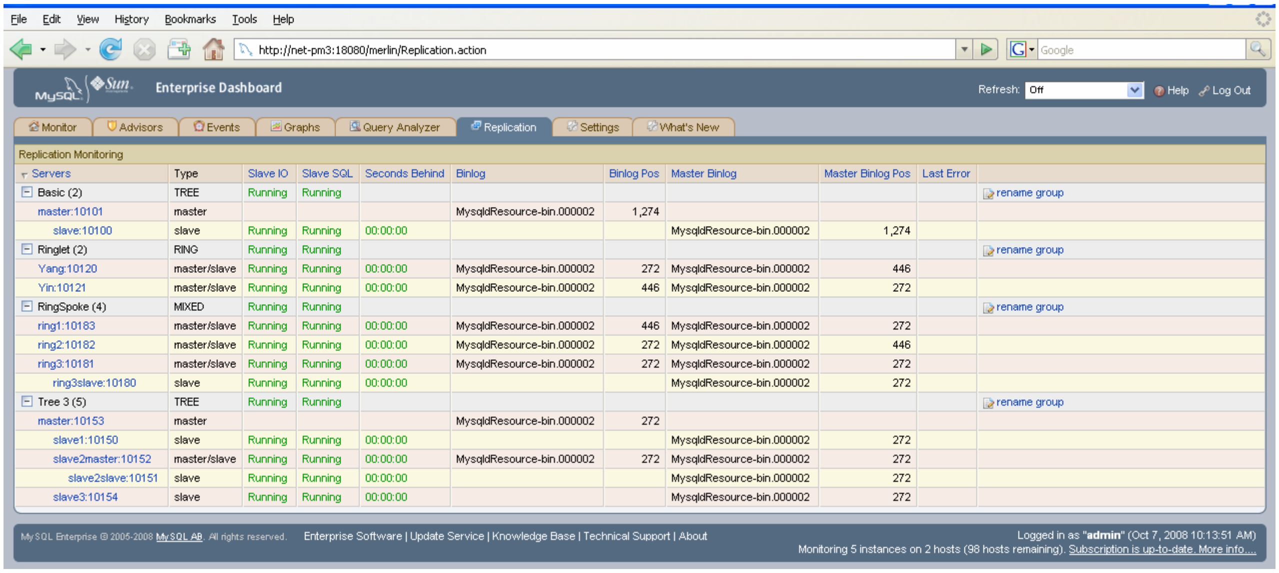
Securing Your MySQL Installation
Securing a MySQL Server on Windows

[5] Forrester, September 2006

groups of recipients. Each recommendation clearly communicates what the identified issue is, the recommended way to correct the problem, and, where applicable, the appropriate MySQL command that can be used to make things right. Finally, links to the MySQL Enterprise online knowledge base and other external resources are supplied so a DBA can easily locate more information regarding the identified best practice violation.

Replication Specific Monitoring

The Replication Monitor provides a consolidated, real-time view into the health, performance and availability of all master/slave topologies. Working with the Replication Advisor Rules, the Replication Monitor helps the DBA to proactively identify and correct Replication related problems before they can become costly outages. As the Replication Advisor identifies a problem and sends out an alert, the DBA can use the alert content along with the new Replication Monitor to drill into the status of the affected master and/or slave. Using the Replication Monitor and the expert advice from the Replication Advisor they can review the current master/slave status and view metrics (such as Slave I/O, Slave SQL thread, seconds behind master, master binlog position, last error, etc.) that are relevant to diagnosing and correcting the problem. The Replication Monitor is designed and implemented to save DBAs time writing and maintaining scripts that collect, consolidate and monitor similar MySQL Replication status and diagnostic data.



Replication Monitoring

Servers	Type	Slave IO	Slave SQL	Seconds Behind	Binlog	Binlog Pos	Master Binlog	Master Binlog Pos	Last Error	
⊟ Basic (2)	TREE	Running	Running							rename group
master:10101	master				MysqldResource-bin.000002	1,274				
slave:10100	slave	Running	Running	00:00:00			MysqldResource-bin.000002	1,274		
⊟ Ringlet (2)	RING	Running	Running							rename group
Yang:10120	master/slave	Running	Running	00:00:00	MysqldResource-bin.000002	272	MysqldResource-bin.000002	446		
Yin:10121	master/slave	Running	Running	00:00:00	MysqldResource-bin.000002	446	MysqldResource-bin.000002	272		
⊟ RingSpoke (4)	MIXED	Running	Running							rename group
ring1:10183	master/slave	Running	Running	00:00:00	MysqldResource-bin.000002	446	MysqldResource-bin.000002	272		
ring2:10182	master/slave	Running	Running	00:00:00	MysqldResource-bin.000002	272	MysqldResource-bin.000002	446		
ring3:10181	master/slave	Running	Running	00:00:00	MysqldResource-bin.000002	272	MysqldResource-bin.000002	272		
ring3slave:10180	slave	Running	Running	00:00:00			MysqldResource-bin.000002	272		
⊟ Tree 3 (5)	TREE	Running	Running							rename group
master:10153	master				MysqldResource-bin.000002	272				
slave1:10150	slave	Running	Running	00:00:00			MysqldResource-bin.000002	272		
slave2master:10152	master/slave	Running	Running	00:00:00	MysqldResource-bin.000002	272	MysqldResource-bin.000002	272		
slave2slave:10151	slave	Running	Running	00:00:00			MysqldResource-bin.000002	272		
slave3:10154	slave	Running	Running	00:00:00			MysqldResource-bin.000002	272		

MySQL Enterprise © 2005-2008 MySQL AB. All rights reserved. Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as "**admin**" (Oct 7, 2008 10:13:51 AM)
Monitoring 5 instances on 2 hosts (98 hosts remaining). Subscription is up-to-date. More info....

Figure 8: MySQL Enterprise Replication Monitor

Monitoring Application Code - MySQL Query Analyzer

Integrated into the Enterprise Monitor the Query Analyzer is designed to help Developers and DBAs accurately pinpoint SQL code that is causing a slowdown, quickly diagnose the inefficiencies and get the affected application back to peak performance. To this end, the Query Analyzer leverages the MySQL Proxy technology to extend the Service Agent to listen on a user defined port for application queries, collect the SQL code and performance metrics, and then report the results back to the Service Manager for monitoring and presentation. A typical implementation is depicted in Figure 9:

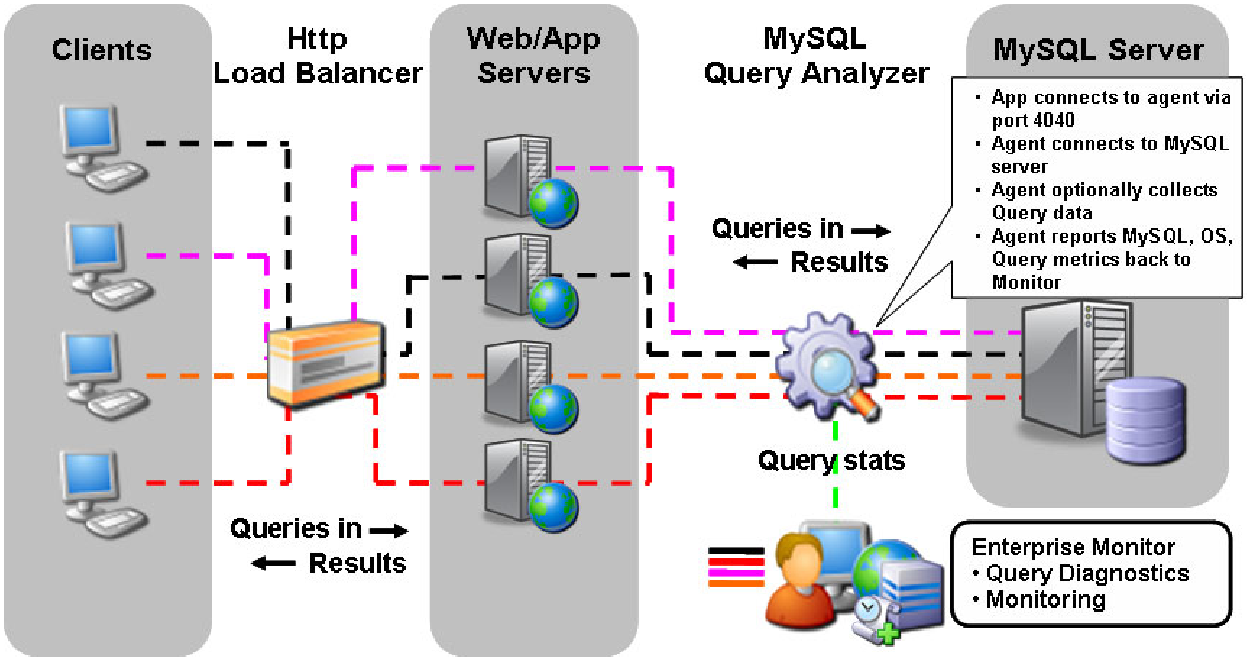


Figure 9: MySQL Query Analyzer Deployment

In this example, the agent is installed on the monitored MySQL server and is configured to either collect queries (the default) or to serve as pass-thru until collection is enabled. The configuration is straightforward:

- The application is configured to connect to the Service Agent on a user defined port (4040 by default)
- Service Agent connects to MySQL server (single server, master, read slave)
- Service Agent is enabled to collect queries or to serve as a pass-thru until needed
- Agent reports MySQL, OS and query metrics back to Service Manager for monitoring

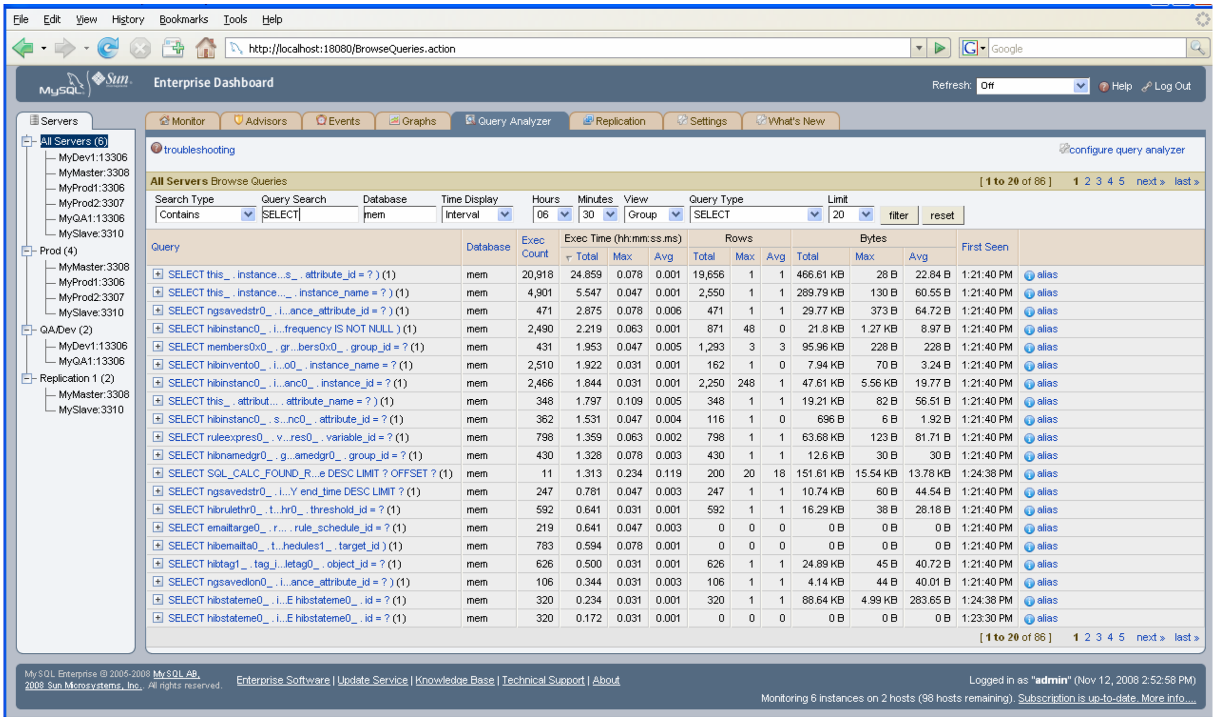
Enabling active query collection with Service Agent is controlled using the Enterprise Dashboard and allows you to enable query collection for individual servers or complete applications.

How the Query Analyzer Helps the Developer and DBA

The Query Analyzer saves you time and effort in monitoring your MySQL servers for problem queries by providing:

- An integrated monitoring solution for all supported versions of MySQL (4.1 and higher).
- Aggregated query content and performance stats in real time with no reliance on MySQL logs or SHOW PROCESSLIST.
- A consolidated view into query activity across all MySQL servers, no user parsing required.
- Historical browsing/analysis of queries across all MySQL servers.
- Aggregated roll ups of all queries in canonical form (no variables) with total executions, total execution time, total data size and date/time of when “first seen”.
- Drill downs into query details, number of executions, execution stats and visual EXPLAIN plan.

The Query Analyzer summary of all queries and aggregated execution statistics is shown here:



- Servers
  - All Servers (6)
    - MyDev1:13306
    - MyMaster:3308
    - MyProd1:3306
    - MyProd2:3307
    - MyQA1:13306
    - MySlave:3310
  - Prod (4)
    - MyMaster:3308
    - MyProd1:3306
    - MyProd2:3307
    - MySlave:3310
  - QA/Dev (2)
    - MyDev1:13306
    - MyQA1:13306
  - Replication 1 (2)
    - MyMaster:3308
    - MySlave:3310

All Servers Browse Queries [**1 to 20** of 86] 1 2 3 4 5 next » last »

Search Type	Query Search	Database	Time Display	Hours	Minutes	View	Query Type	Limit
Contains	SELECT	mem	Interval	06	30	Group	SELECT	20

Query	Database	Exec Count	Exec Time (hh:mm:ss.ms)			Rows			Bytes			First Seen	
			Total	Max	Avg	Total	Max	Avg	Total	Max	Avg		
SELECT this_ . instance...s_ . attribute_id = ?) (1)	mem	20,918	24.859	0.078	0.001	19,656	1	1	466.61 KB	28 B	22.84 B	1:21:40 PM	alias
SELECT this_ . instance..._ . instance_name = ?) (1)	mem	4,901	5.547	0.047	0.001	2,550	1	1	289.79 KB	130 B	60.55 B	1:21:40 PM	alias
SELECT ngsavedstr0_ . i...ance_attribute_id = ?) (1)	mem	471	2.875	0.078	0.006	471	1	1	29.77 KB	373 B	64.72 B	1:21:40 PM	alias
SELECT hibinstanc0_ . i...frequency IS NOT NULL) (1)	mem	2,490	2.219	0.063	0.001	871	48	0	21.8 KB	1.27 KB	8.97 B	1:21:40 PM	alias
SELECT members0x0_ . gr...bers0x0_ . group_id = ? (1)	mem	431	1.953	0.047	0.005	1,293	3	3	95.96 KB	228 B	228 B	1:21:40 PM	alias
SELECT hibinvento0_ . i...o0_ . instance_name = ? (1)	mem	2,510	1.922	0.031	0.001	162	1	0	7.94 KB	70 B	3.24 B	1:21:40 PM	alias
SELECT hibinstanc0_ . i...anc0_ . instance_id = ? (1)	mem	2,466	1.844	0.031	0.001	2,250	248	1	47.61 KB	5.56 KB	19.77 B	1:21:40 PM	alias
SELECT this_ . attribut... . attribute_name = ?) (1)	mem	348	1.797	0.109	0.005	348	1	1	19.21 KB	82 B	56.51 B	1:21:40 PM	alias
SELECT hibinstanc0_ . s...nc0_ . attribute_id = ? (1)	mem	362	1.531	0.047	0.004	116	1	0	696 B	6 B	1.92 B	1:21:40 PM	alias
SELECT ruleexpres0_ . v...res0_ . variable_id = ? (1)	mem	798	1.359	0.063	0.002	798	1	1	63.68 KB	123 B	81.71 B	1:21:40 PM	alias
SELECT hibnamedgr0_ . g...amedgr0_ . group_id = ? (1)	mem	430	1.328	0.078	0.003	430	1	1	12.6 KB	30 B	30 B	1:21:40 PM	alias
SELECT SQL_CALC_FOUND_R...e DESC LIMIT ? OFFSET ? (1)	mem	11	1.313	0.234	0.119	200	20	18	151.61 KB	15.54 KB	13.78 KB	1:24:38 PM	alias
SELECT ngsavedstr0_ . i...Y end_time DESC LIMIT ? (1)	mem	247	0.781	0.047	0.003	247	1	1	10.74 KB	60 B	44.54 B	1:21:40 PM	alias
SELECT hibrulethr0_ . t...hr0_ . threshold_id = ? (1)	mem	592	0.641	0.031	0.001	592	1	1	16.29 KB	38 B	28.18 B	1:21:40 PM	alias
SELECT emailtarge0_ . r... . rule_schedule_id = ? (1)	mem	219	0.641	0.047	0.003	0	0	0	0 B	0 B	0 B	1:21:40 PM	alias
SELECT hibemailta0_ . t...hedules1_ . target_id) (1)	mem	783	0.594	0.078	0.001	0	0	0	0 B	0 B	0 B	1:21:40 PM	alias
SELECT hibtag1_ . tag_i...letag0_ . object_id = ? (1)	mem	626	0.500	0.031	0.001	626	1	1	24.89 KB	45 B	40.72 B	1:21:40 PM	alias
SELECT ngsavedlon0_ . i...ance_attribute_id = ?) (1)	mem	106	0.344	0.031	0.003	106	1	1	4.14 KB	44 B	40.01 B	1:21:40 PM	alias
SELECT hibstateme0_ . i...E hibstateme0_ . id = ? (1)	mem	320	0.234	0.031	0.001	320	1	1	88.64 KB	4.99 KB	283.65 B	1:24:38 PM	alias
SELECT hibstateme0_ . i...E hibstateme0_ . id = ? (1)	mem	320	0.172	0.031	0.001	0	0	0	0 B	0 B	0 B	1:23:30 PM	alias

[**1 to 20** of 86] 1 2 3 4 5 next » last »

MySQL Enterprise © 2005-2008 MySQL AB, 2008 Sun Microsystems, Inc.. All rights reserved.

Enterprise Software | Update Service | Knowledge Base | Technical Support | About

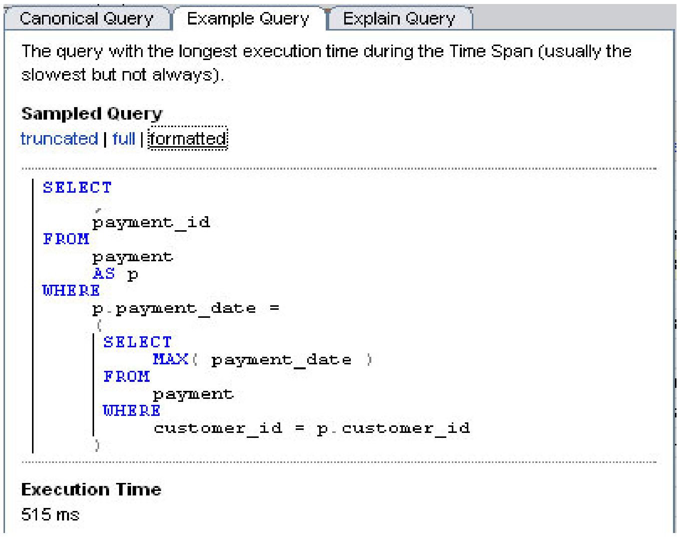
Logged in as "**admin**" (Nov 12, 2008 2:52:58 PM)

Monitoring 6 instances on 2 hosts (98 hosts remaining). Subscription is up-to-date. More info....

The worst performing queries are presented will full drill down details so you can analyze what variable combinations are leading to performance degradations.

Details include:

- Fully qualified (variable substitution) views of worst performing statements.
- Execution time for specific occurrence
- Date/time of execution
- Originating session/user/thread id
- Source/destination host and MySQL server
- Integrated EXPLAIN of worst performing query for analysis of execution path.



The query with the longest execution time during the Time Span (usually the slowest but not always).

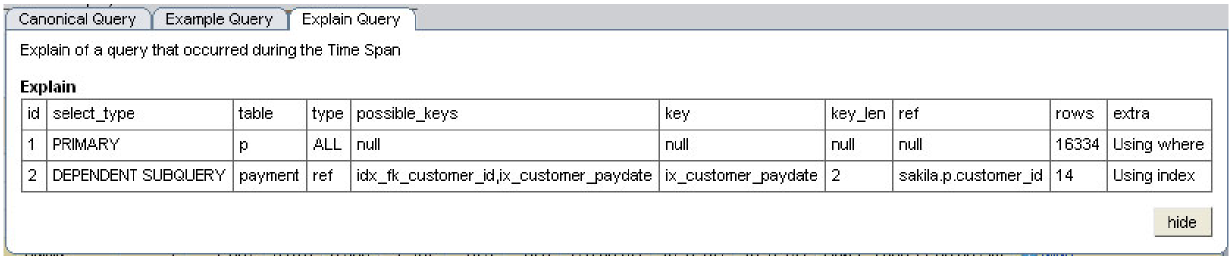
Sampled Query

truncated | full | formatted

```
SELECT
        payment_id
FROM
        payment
        AS p
WHERE
        p.payment_date =
        (
        SELECT
                MAX( payment_date )
        FROM
                payment
        WHERE
                customer_id = p.customer_id
        )
```

Execution Time

515 ms



Explain of a query that occurred during the Time Span

Explain

id	select_type	table	type	possible_keys	key	key_len	ref	rows	extra
1	PRIMARY	p	ALL	null	null	null	null	16334	Using where
2	DEPENDENT SUBQUERY	payment	ref	idx_fk_customer_id,ix_customer_paydate	ix_customer_paydate	2	sakila.p.customer_id	14	Using index

Using the Enterprise Monitor Query Analyzer you can:

- Tune your SQL code during active development before it is promoted to production.

- Quickly identify queries that are negatively affecting the throughput of your production environments
    - by query type, content, server or application user.
- Once problematic code is surfaced you can:
    - View an EXPLAIN to determine the current access path and index usage and tune the offending code for better performance.

In addition to the code monitoring and diagnostic benefits noted above, the Query Analyzer:

- Shares common, core components with other MySQL Enterprise scalability and performance solutions.
- Adds little pass-thru overhead to your monitored environments (benchmarked at 400 micro seconds per packet).
- Is fully supported by MySQL production support services.
- Is included as part of a MySQL Enterprise Gold or Platinum subscription.

Real Time Savings Value for the DBA

As noted earlier, operational and application DBAs with production support responsibilities spend an average of 5% of their time writing and maintaining scripts to monitor MySQL performance and availability metrics. Regardless of the size or make up of the MySQL environment being monitored, all MySQL DBAs are challenged with ensuring their MySQL servers are up and running. The following example analyzes how a DBA can save almost 2 working days on initial set up and on average of 20 minutes a day using the MySQL Enterprise Monitor to monitor this metric instead of writing and maintaining homegrown scripts for the same purpose.

Time Estimate for Monitoring the Up/Down status of 10 MySQL servers – Initial Set up				
Requirement	**With Script**	**Time**	**With MySQL Enterprise**	**Time**
Monitor up/down status of all MySQL Servers	Research Windows or Linux specific commands and syntax	8 hours	Not required	0
	Develop/test Windows and/or Linux "ping" scripts for all servers	30 minutes/server = 5 hours	Automatically monitored	0
	Deploy scripts to all MySQL servers	3 minutes/server = .5 hour	Not required	0
	Schedule/test scripts via Task Scheduler, CRON, etc.	12 minutes/server = 2 hours	Automatically scheduled	0
Total set up time		**15.5 hours**		**0**

As seen in the grid, using scripts to monitor this one key metric requires several steps. Although this example involves only 10 MySQL servers, it is easy to see that it involves work and testing across multiple technologies before any actual monitoring can begin.

Time Estimate for Monitoring the Up/Down status of 10 MySQL servers – Ongoing Expense				
Requirement	**With Script**	**Time**	**With MySQL Enterprise**	**Time**
Monitor up/down status of all MySQL Servers	Monitor "ping" script results for all MySQL servers/slaves	2 minutes/server = 20 minutes	View up/down status for all servers at one time	< 1 minute
	Develop/test Windows and/or Linux "ping" scripts for new servers	*As needed	Not Required	0
	Deploy scripts to all new MySQL servers/slaves	*As needed	Not required	0

	Schedule/test scripts for new servers via Task Scheduler, CRON, etc.	*As needed	Automatically scheduled	0
Total daily time		**20 minutes**		**< 1 minute**
*** more on days when new servers/slaves are added.**				

Once the above monitoring solution is implemented using scripts the DBA must spend time each day compiling and monitoring the script execution results, creating new scripts for each new MySQL servers as they are added to this environment and revisiting existing scripts as new versions of the MySQL server are implemented.

DBAs using the MySQL Enterprise Monitor need only to visit the Enterprise Dashboard to monitor this key metric for all of their MySQL Servers. In addition, the MySQL Enterprise Monitor auto-detects new MySQL servers so there is no set up involved when a new server is added to the environment. Versioning is automatically handled so DBAs can begin monitoring the most current versions of MySQL without worrying about changes to the collected metrics.

This example demonstrates how this one key feature of the MySQL Enterprise Monitor can save an over extended DBA almost 2 hours each week.

Conclusion

As MySQL continues to expand into the corporate data center, DBAs are challenged to maintain production level SLAs for uptime and performance, while managing an ever-growing number of servers and MySQL powered applications. The MySQL Enterprise Monitor helps DBAs manage more MySQL servers with less and effort and to remain proactive in addressing these challenges by providing a consolidated view into the health, availability, security and performance of all of their MySQL servers. Serving as a “Virtual MySQL DBA” assistant to the DBA, the MySQL Enterprise Monitor proactively monitors all of the MySQL servers across the enterprise, and empowers the DBA to address specific problems and tuning opportunities before they become performance degradations or costly outages. The combination of enterprise visibility, proactive monitoring and expert advice and guidance in problem identification and resolution makes the MySQL Enterprise Monitor the perfect addition to any DBA staff.

Learn More About Sun’s MySQL Portfolio

The MySQL product portfolio is the most popular open source database software in the world. Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems and packaged software -- including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube and Zappos.com. For more information on how Sun provides corporate users with commercial subscriptions and services, and actively supports the large MySQL open source developer community visit http://www.mysql.com/ or one the links noted below:

MySQL Enterprise, the Enterprise Monitor, Query Analyzer, Production Support

http://mysql.com/products/enterprise/

MySQL Replication and Scale-Out

http://mysql.com/products/enterprise/high_availability.html

MySQL Professional Services and Consulting

http://mysql.com/consulting/

Customer Success Stories and Case Studies

http://www.mysql.com/why-mysql/case-studies/

Try MySQL Enterprise and the Enterprise Monitor Yourself

http://www.mysql.com/trials/