

Sun and MySQL: How It Stacks Up for Developers

Most business analysts have applauded the announced acquisition by Sun Microsystems of open-source database provider MySQL AB. Although the business case for bringing MySQL under the Sun umbrella is clear, the acquisition leaves many developers wondering just what is in store for them.

Developers want products that are easy to use during development, that are easy to scale to enterprise levels, and that will last long enough to repay the time investment required to master the technologies. Some MySQL community members wonder if they will continue to enjoy superior open-source products from the combined companies, and if the community will remain strong in the future.

MySQL, the Leading Open-Source Option

MySQL leads the open-source database market, and by every indication, the future of MySQL at Sun is with the open-source community. According to Mårten Mickos, CEO of MySQL, it was Sun's commitment to free and open-source software that helped seal the deal. During a four-year courtship, which Jonathan Schwartz chronicles in his [blog](#), it became apparent to the MySQL leadership that Sun's historic commitment to open systems had extended to open-source software, and that Sun would be a home in which the MySQL community could thrive. Both Java technology and MySQL are licensed under GPL Version 2 with an eye toward GPL Version 3 as that license matures.

For MySQL developers, little will change. But the MySQL Enterprise customer can expect an even greater level of service and support, backed by a Fortune 500 company. And as MySQL is adopted more broadly, all users can expect faster product innovation, more rigorous testing, and a better MySQL product overall. MySQL and the Sun Software Stack

Skeptics of Sun's commitment to open source might fear being coerced, however subtly, into adopting other Sun products — in particular the Solaris OS, which is itself undergoing an open-source renaissance. Events at Sun, including partnerships with Intel, Microsoft, Oracle, and other competitors, indicate that Sun will adapt to the marketplace rather than try to enforce any type of software regime on developers.

The acronym LAMP (Linux operating system, Apache web server, MySQL database server, PHP/Perl/Python languages) describes the software stack favored by open-source developers. Each element in the stack has alternatives, both open source and proprietary, making LAMP not very useful as a descriptive acronym. For example, of the 50,000 daily downloads of MySQL, the majority of binaries are for Windows operating systems — for WAMP developers, perhaps? Similar numbers apply to other open-source tools and products sponsored by Sun, notably the GlassFish application server. While determined to compete aggressively for market share among LAMP developers, Sun is porting its tools and software to every popular platform.

Developers who are committed to Sun software may likewise be wondering if MySQL will supplant database alternatives such as PostgreSQL and JavaDB in its software stack. Again, Sun has voiced its intention of supporting any popular management software that developers want to insert into the database slot, including Apache Derby. In the words of Josh Berkus of the PostgreSQL core team, "Sun is not dropping PostgreSQL support. It's Sun's goal to be the number one data center vendor, not the number one MySQL vendor.... In the world of databases, there is no 'one size fits all.' "

If you're already using MySQL in your web application software stack, you can expect MySQL to be the same free, community-based database you've been using, regardless of operating system, web server, or language. When you want to scale out for enterprise-level performance, Sun will provide the support and services you need. If you're committed to another database, expect it to cooperate well with Sun products. [MySQL Details](#)

If you haven't tried MySQL or other Sun offerings in developing web applications, it may be time to reevaluate.

Although there is no such thing as a one-size-fits-all database, the popularity of MySQL shows just how close it has come to that ideal. Its adaptability and ease of use is largely due to its architecture and ability to scale. This architecture helps explain why MySQL is the choice for more than 11 million active installations worldwide with more than 100 million copies downloaded or distributed to date. At this writing, a new copy of MySQL is being downloaded every two seconds. [Scale-out Capabilities](#)

The strategy that many MySQL users employ for success is known as scale-out. The term refers to an architectural design that separates and spreads a database's workload across commodity hardware by replicating data across different physical servers. This strategy is in contrast to adding more CPU power, storage, and memory to a smaller number of monolithic servers — a strategy sometimes called scale-up. The scale-out strategy typically provides better fault tolerance and overall performance for the money, as evidenced by the many heavily trafficked MySQL web

sites that have achieved high levels of uptime with this approach.

Figure 1 illustrates the scale-out architecture. [Click here for a larger image.](#) **Figure 1. Scale-out Architecture**

In a typical situation, users initiate sessions on customer-facing web or application servers. These servers send requests as required into the MySQL database servers through load balancers or other application logic that directs the requests. When a request causes a server to write into the database, the server replicates its data to other MySQL servers to keep them up-to-date for subsequent queries.

In high-availability, high-volume settings, a failure in any MySQL server is typically corrected by taking the failed server offline and swapping it for another server. In a few minutes, the database can be replicated on the new server, and query service can proceed as before. The low cost of commodity servers makes this swap-out strategy cost-effective. **Storage Engine Architecture**

The storage engine architecture of MySQL makes it unique in the database world. Of special interest is the pluggable storage engines layer, as shown in Figure 2. [Click here for a larger image.](#) **Figure 2. Storage Engine Architecture**

Above the storage engine layer, MySQL presents itself as any other database management system would. A group of connectors is available through which to query the database. A connection pool in the server provides authentication and manages threads, connections, memory, and caches. Modules in the server parse queries, optimize access paths, and so on.

Beneath this layer, multiple storage engines can be plugged into the architecture. The storage engines are the components of the database server that perform actions on the underlying data being maintained at the physical server level. The pluggable storage engine architecture enables a standard set of management and support services to apply to all underlying storage engines. The result is like having many databases under the umbrella of a single manager.

Some storage engines, such as those used for archiving, are nontransactional in nature. These storage engines can insert and read data very efficiently. Other storage engines are tuned for efficiency when transactional operations are required, and still others provide high availability through clustering. These storage engines avoid the natural overhead of proprietary systems, which typically must live with the transactional overhead of their table architectures even when it isn't needed.

For example, a site that primarily needs to look up data can avoid all of the transactional overhead by using the MyISAM storage engine, which is the default engine for MySQL. The MyISAM engine offers high-speed query and insert capability, is nontransactional, provides table-level locking, and supports indexes. It is a good engine for traditional data warehouses.

The MySQL database server comes supplied with a group of internal storage engines. External storage engines are tuned to optimize performance for specific products and situations, and are supplied both by independent software vendors and by the MySQL community.

Having multiple storage engines provides you with the capability of multiple databases. If you migrate to MySQL from another database, you don't need to change the way your applications interact with the database server — the SQL code is the same. But internal to the MySQL server, you can pick and choose storage engines that meet your application's needs.

For example, if you discover that you need a transactional database, you can use a single statement to transform your database from a nontransactional system to a transactional one. If you then decide that you need a data warehouse, a single command can put you in possession of a powerful column-oriented database tuned for data retrieval. If your main activity is to capture and store data, the Archive storage engine is designed to efficiently handle large volumes of inserts and compress data to a small footprint. **High Availability**

ISPs and mainstream businesses may find that the MySQL server's replication feature is sufficient for their high availability needs.

To increase availability, customers can add storage area network (SAN) technology or Distributed Replicated Block Device (DRBD), a Linux kernel module from LINBIT that distributes block device replication across the network. The highest availability — "five nines," or 99.999 percent — is provided by MySQL Cluster, which performs synchronous replication between data nodes, automatic failover, and replication between clusters.

The ability of MySQL to provide the highest levels of performance for the most heavily trafficked sites, combined with its ease of use in product development, helps account for its broad popularity. If It's Free, Why Pay for It?

Twenty-five percent of MySQL users are developers. In many small companies, especially startups, the developer is also the database administrator. As the company matures and volume and profits grow, key individuals discover that their time is better spent in activities other than database administration. High reliability and staffing expenses become increasingly important. At this point, purchasing MySQL Enterprise makes economic sense because of the monitoring and support capabilities it supplies.

Note that the community version of MySQL has features identical to the MySQL Enterprise version — although the Enterprise version undergoes more frequent updates. These features include the GUI management tools, the migration toolkits for migrating data from other servers, connectors for JDBC and ODBC, and so on. Access to documentation and user forums is also free to the community. You can develop for free with MySQL, and there will be no surprises when you begin to scale out your deployment and move to a paid subscription version of MySQL. Conclusion

Embracing an open-source model for Java technology, the Solaris Operating System, and other properties has made Sun Microsystems the world's largest corporate contributor to the open-source community. MySQL, already the world's most popular open-source database, fills an important niche in Sun's software stack. Backed by Sun's reach and resources, MySQL is poised for even wider adoption. The result should benefit Sun, MySQL, the developer community, and enterprise customers both big and small.