Technical Comparison of DB2 and MySQL

Martin Hubel
Martin Hubel Consulting Inc.
October 2004

Contents

1. Executive Summary

MySQL has long been known as the database of choice for the open source web developer. Its growing popularity and use for commercial applications has increased to a point where support is now available from MySQL AB. Until recently, MySQL was not considered a commercial product like those from "real" relational database vendors such as IBM's DB2, Oracle, and Microsoft's SQL Server. However, the growing popularity of the open-source community, as well as improvement in the MySQL feature set has helped it gain acceptance from some developers and corporate IT staff.

The transformation of MySQL from an open source into a commercial product is a big undertaking, and while MySQL AB provides support, that does not mean that these facilities are as robust as DB2 and the other commercial offerings. Enterprises require rich functionality to ensure service levels, performance, scalability, and data integrity. While the basic functionality is certainly available in all commercially available database products, only over time can the functionality be enhanced to better suit the requirements of businesses.

Integrity encompasses the areas of operability, security, and recovery. These aspects of database systems are important to support high volume commercial use. Tools may also be supplied with the database, or be available from other vendors, to supplement the capabilities and improve the viability of the product. Tools are often available to support administration, monitoring, tuning, productivity and the safe use of the product. The tools for MySQL are not as developed, nor readily available as those available for DB2.

MySQL is attempting to add many of the features that can be found in the more mature commercial offerings. An example of a promised, but yet unavailable feature in MySQL is the addition of clustering technology. MySQL Cluster will partition data across various storage nodes and keep redundant copies of the data synchronized using a two-phase commit protocol. However, this approach introduces performance bottlenecks and has limited scalability (up to 4 nodes). This valuable feature has been available in DB2 since Version 1, and can support up to 1000 nodes. DB2 using the Database Partitioning Feature (DPF) is an ideal choice for clustered databases.

As DB2 has matured, IBM has added advanced autonomic features to DB2 such as self-managing computing solutions that automate and simplify time consuming and complex database tasks. Some examples include performance optimization, online maintenance capabilities, and system health monitoring capabilities. These capabilities do not exist in MySQL.

DB2 also makes migration to new versions simple through tools like db2updv8 for upgrading databases. Conversely, upgrading MySQL versions still requires data unload and reload. The migration of data from other database products to DB2 is simple with free of charge toolkit from IBM, while MySQL does not have this capability.

DB2 is still the industry leader with best return on investment from entry-level database to high-end clustered solutions. Both DB2 and MySQL have entry level solutions priced at under $500. DB2 offers some of the best manageability and autonomic features in the engine that not only reduce

cost of setup, etc. but also help protect the investment by reducing the overall downtime of the system.

Even though MySQL has grown in popularity, it still lacks the breadth of features and depth of technical ability that an advanced database system such as DB2 offers. Combined with extremely competitive pricing strategy and virtually limitless performance and scalability, DB2 is still the ideal choice for database systems. This fact is underlined by Gartner Research's 2003 report for database vendors that placed IBM as the leader in worldwide database market share.

This white paper compares DB2 and MySQL in technical detail and focuses on many key areas including scalability, performance, maintainability, cost of ownership, and self-tuning features.

2. Introduction

The database server is a standard part of almost every business these days. Common commercial databases, such as DB2, Oracle, and Microsoft SQL Server include many features that people have come to rely on to make their database servers "enterprise worthy". These features include advanced database utilities, management tools, replication, disaster recovery, and performance monitoring.

During the past ten years, the open-source community has improved the quality of its software, making it ready for the enterprise. This popularity in open-source software has also initiated a new trend: databases are quickly becoming an integral part of the open-source middleware stack such as LAMP. The functionality of MySQL now makes it a competitor to commercial products such as DB2 for some applications.

Over the last few years, MySQL has traditionally been a natural choice for PHP or Perl/DBI for web-enabled database sites that use open-source databases. Until recently, MySQL AB, the software vendor behind MySQL, has been the champion of users whose database requirements are simple; make it fast, reliable, and cheap. For all but a handful of users, cost was the main factor in their selection criteria.

This Whitepaper provides an in-depth technical comparison MySQL with DB2. The DB2 and MySQL product editions and platform support will be discussed as well as other topics that include engine capabilities, clustering, availability, and scalability.

3. Packaging

Editions:

MySQL offers two General Public License (GPL) based databases and two commercially licensed database flavors:

1. MySQL Standard – GPL based, full transactional support database.
2. MySQL Max – GPL based, early adopters' database with advanced features.
3. MySQL Pro – Commercially licensed version of MySQL Standard.
4. MySQL Classic – Commercially licensed, basic in-memory database.

DB2 has a whole range of products that support all types of applications and platforms. They can be categorized broadly as follows:

1. DB2 Cloudscape – A Java-based lightweight embeddable database
2. DB2 UDB (for distributed platforms – Linux, UNIX, and Windows):
    a. DB2 Express Edition – 2 Processor limit, also memory limit suited for the small and medium businesses
    b. DB2 Workgroup Edition – 4 Processor limit, for larger departmental applications
    c. DB2 Enterprise Server Edition – Highly mission critical applications & clusters
    d. DB2 Enterprise Server Edition – Highly mission critical applications & clusters
    e. DB2 Personal Edition – Full function database for non commercial use
3. DB2 z/OS – database for IBM zSeries mainframes
4. DB2 for iSeries – database for IBM iSeries (AS400) platforms
5. DB2 Everyplace – For mobile and other pervasive devices

MySQL database is not a single database engine, but a combination of four table handlers: InnoDB - storage, MyISAM – read-only, HEAP in-memory, BerkleyDB extended features. These engines have overlapping functionality and use a different table structure, a different file format on disk and offer different features. It is not just the data access and concurrency control features that are different; the administrative requirements are also different. The distinct separations between these storage engines and the core server might seem flexible but can potentially cause manageability overhead and data inconsistency.

DB2 has the same engine, code-base as well as administration mechanism on all distributed platforms, across the board. In DB2, storage and core server capabilities are all part of the database engine itself and no separate handlers are required.

Install:

Being an open source database MySQL install could be binary-based or source-based. However, DB2 is a binary-based only install. Binary-based installation methods supported by both MySQL and DB2 are similar and both products support features such as silent installs/de-installs, etc. The install times for both products can vary based on the edition, and the server your are installing on.

Embedded Capabilities:

Both MySQL and DB2 offer versions that can be embedded in other applications. These versions have features such as a small disk and memory footprint with lightweight database functionality. DB2 Cloudscape also includes some advanced offerings for easy embeddability including near zero management requirements and easy upgrades. We will discuss more about these features in the following sections.

Positioning Statements:

In commercial applications that require full transactional support including logging or read-intensive applications, MySQL Pro would be the edition of choice. Open-source projects can choose from the MySQL standard or MySQL Max flavors for development. In the rare situations where a commercial application can function with just an in-memory storage engine (data consistency is not important) MySQL Classic may be used. Clusters support may be included in the MySQL Pro edition starting v5.0 (ETA mid 2005) however; this has not been announced yet.

The majority of customers that are looking for a database such as MySQL should evaluate DB2 Cloudscape, DB2 Express, or DB2 Workgroup Edition as comparable offerings. DB2 Cloudscape would be best suited where lightweight embeddable database is needed with limited transaction load. Typical applications that would use DB2 Cloudscape are read only applications such as Java based web applications that can benefit from JDBC and SQL-92 compliance, full transactional support, as well as near zero maintenance features. Once the needs of the application environment grows and a database is needed that can handle industry leading transactional loads and advanced features such as replication, high availability, complex SQL, etc. DB2 Express would provide an easy to use solution with one of the best total costs of ownership and return on investment in the industry.

Product Roadmap:

The latest production release of MySQL is v4.0.7.The main features of this release were inclusion of InnoDB in the standard offerings to enable transactional support capabilities. MySQL v4.1, which is in beta at the moment, will add much needed capabilities such as sub-queries to the MySQL engine. The alpha release of MySQL v5.0 is also available which likely will add clustering capability to MySQL.

DB2 UDB on the distributed platforms is currently at v8.2 which includes amongst other advanced offerings high availability features such as HADR, Client Reroute, etc. More information on these features is available in the following sections. All the following sections discuss feature sets that are available with DB2 on distributed platforms and the MySQL Pro edition.

4. Platform Support Comparison

The figure below lists the supported platforms by both MySQL and DB2. Both databases have similar support models for the distributed platforms. However, DB2 offers a more complete list of widely used platforms from mobile computing to mainframes.

Platform	DB2 V8.2	MySQL 4.0.7
MS Windows (Intel/AMD) XP, 2000/3, NT	X	X
AIX (4.33, 5L)	X	X
Sun Solaris (7,8,9)	X	X
HP UX (RISC, Itanium)	X	X
Linux (Intel/AMD, various distributions)	X	X
zLinux and Power Linux	X	N/A
Mac OS X	N/A	X
AS400	X	N/A
z/OS	X	N/A

Figure 1: Supported Platforms for DB2 and MySQL

As of October, 2004.

This table is based on platforms supported by MySQL Pro 4.0.7 and DB2 V8.2.

Apart from these platforms DB2 is also available (DB2 Everyplace edition) to be used with mobile platforms and applications such as hand-held devices, cell-phones etc.

MySQL database currently supports 64 bit versions of the OS for the Linux (IA64 and AMD) and HP-UX (IA64). DB2 Supports 64 bit versions available on all supported distributed platforms

5. Engine Capabilities

Database Objects:

MySQL lacks many database objects that are available in other commercial and non-commercial vendors. These features are important for data integrity, development productivity, and performance. Some of these are discussed below:

Triggers:

Triggers provide valuable functionality for data validation and performing updates to other related tables following an insert, update or delete. A trigger is "fired" (activated) either before or after the INSERT, UPDATE or DELETE statement is executed. You can have multiple before and after triggers on the same event and you have complete control of the order of execution. Both before and after triggers have access to the old and new values in every field for the row being changed/added. The action taken by a trigger is part of the same unit of work as the triggering statement; if a transaction rollback occurs, the changes made by triggers will rollback too.

For example, a financial application is under development, and the audit department recommends adding more information to the audit trail. All changes to the major data tables must be logged to a corresponding audit log in DB2 tables. The audit log must include the old and new data values, the data and time of the change, and the name of the user that made the change.

To implement this without triggers, such as in MySQL, application code must be written within each application program that performs updates. Additional programming time is needed to build and test this application code. In contrast, by using triggers DB2, it is possible to build and capture the audit trail without changing application programs.

Trigger support is not currently available in MySQL. It is planned for v5.0 in mid 2005).

Views:

Views are an important part of value-based data security and data independence in DB2 and other database products. Views allow user access to be restricted to only certain columns and/or rows of data within a table. Views are considered to be "tables" to a user, without the need to store the information again. View also provide program and data independence, because if a program uses a view, the columns in a table not used by the view can be changed without the program being affected.

MySQL has no mechanism for restricting access to rows in a table. The logic must be coded in the application layer.

For example, users in the human resources department need the ability to run ad hoc queries and reports. It is necessary to ensure that only the director of executive compensation can view

employee information whose pay grade is greater than 10. Using DB2, it is easy to create a view on the employee table that includes only those records with a pay grade less than or equal to 10. The human resource staff can be granted access to the employee through the view. The HR staff members have a virtual employee table that does not include records for any executives. In addition, another view could be created to exclude the salary column for anyone not authorized to view salary information.

Indexes:

Indexes have two main uses: to improve data retrieval performance and to enforce uniqueness of column values. Simply put, one of the most important aspects of database performance is the ability to build and use indexes that the database can use efficiently to retrieve data.

DB2 is known for the amount of tweaking it allows for databases, especially when it comes to indexing. Overall, experienced DB2 users often find the indexing strategies employed by these open-source databases such as MySQL quite primitive.

MySQL supports indexing techniques such as single column, multi-column, unique, and primary key indexes. DB2's wide array of indexing support includes clustered indexes, indexes on MQTs, multi-dimensional indexes, unique indexes with include columns, indexes for reverse scans, etc.

Materialized Query Tables (MQTs)/ Materialized Views:

An MQT is a table whose definition and its data is based on the result of a query. The data in an MQT is in the form of pre-computed results from the query in its definition, and the database saves time when the MQT can be used to process subsequent queries. A common form of an MQT is the automated summary table.

Complex workloads, such as business intelligence applications, have large volumes of data that can benefit from MQTs since applications and users typically read and re-read the same data over and over. DB2 provides a powerful MQT mechanism to improve response time for these complex queries, especially queries that might require some of operations like:

- Aggregated data over one or more dimensions
- Joins and aggregates data over a group of tables

MySQL doesn't support MQTs or materialized views.

Multi-Dimensional Clustering (MDC):

MDC provides an elegant method for flexible, continuous, and automatic clustering of data in your tables along multiple dimensions eliminating the need for reorganization. MDC enables a table to be physically clustered on more than one key (or dimension) simultaneously. DB2 v8 and beyond supports multi-dimensional clustering of data, as well has indexing support for these MDCs.

MySQL has no concept of clustering indexes, or clustering on multiple dimensions. MySQL doesn't support MDCs.

Concurrency & Locking:

DB2 provides row-level locking, transaction support, declarative referential integrity and multiple transaction isolation levels that guarantee a consistent snapshot of data at any point in time. DB2 provides (utilizes might be a better word) a facility called “lock escalation” that helps manage locks for long-running transactions and jobs that acquire many locks. Lock escalation consolidates many row-level locks into a single lock at the table level and in doing so it reduces locking overhead and improves elapsed time.

MySQL as well provides row-level locking. However, MySQL does not support lock escalation. This can lead to severe performance problems. As readers and writers increase in number with transactional loads lock wait times will increase exponentially as the database will not be able to process all the incoming locking requirements.

SQL Standards Compliance:

In a mixed database environment, the use of SQL that conforms to industry standards allows programmers to more easily transfer between projects. It also allows applications to be run on different platforms with fewer modifications.

MySQL has limited support for advanced SQL functionality and adherence to standards such as ANSI and ODBC SQL. Some examples are:

- Dashes in SQL (“- -”) are only a comment if followed by a white space
- For VARCHAR columns, trailing spaces are removed when the value is stored, which does not conform to the SQL standards
- CHAR columns are silently changed to VARCHAR columns if they exceed limits
- Privileges for a table are not automatically revoked when you delete a table. You must explicitly issue a REVOKE to revoke privileges for a Table GRANT

DB2 has been an industry leader in compliance to standards as well as a contributor to these standards. Many of DB2’s built-in capabilities frequently become part of the industry standard.

Data types:

While DB2 supports a long list of ANSI standard data types as well as many advanced types, MySQL comes short with no support for key data types such as NUMERIC, BOOLEAN, CLOB, etc. and has various extensions to standard data types that make porting to other databases difficult. Figure 2 provides a list of comparable data types.

DB2 Data Type	MySQL Data Type
CHAR VARCHAR LONGVARCHAR CLOB BLOB	CHAR VARCHAR LONGVARCHAR N/A BLOB TINYBLOB MEDIUMBLOB LOGNBLOB TINYTEXT MEDIUMTEXT LONGTEXT
CHAR FOR BIT DATA VARCHAR FOR BIT DATA LONGVARCHAR FOR BIT DATA BLOB FOR BIT DATA	N/A
SMALLINT	SMALLINT TINYINT MEDIUMINT
BIGINT	BIGINT
INTEGER	INT
DOUBLE	DOUBLE
REAL FLOAT	N/A FLOAT
DATE	DATE
NUMERIC DECIMAL	N/A DECIMAL
TIMESTAMP TIME	TIMESTAMP TIME
GRAPHIC VARGRAPHIC LONGVARGRAPHIC	N/A

Figure 2: Comparison of DB2 and MySQL data types

6. Application Development

Application development involves support for run-time environments and interfaces to popular development platforms and tools.

Application Access:

There is a wealth of drivers available for accessing data in the DB2 and MySQL databases ranging from native ODBC to open-source scripting languages based drivers. Here is a brief discussion of some accessibility options and some of the main features that they provide:

ODBC Based Drivers:

MySQL Connector/ODBC:

MySQL Connector/ODBC is based on MyODBC, an ODBC driver that is used to access MySQL server. This is also extended to provide a .Net Data Provider that wraps an existing connection. MySQL does not have a Managed Provider, however, there are third party drivers available. MySQL .Net capability is rather primitive at the moment.

DB2 CLI Driver / .Net Data Provider:

The ODBC .NET Data Provider makes ODBC calls to a DB2 data source using the DB2 CLI Driver. Therefore, the connection string keywords supported by the ODBC .NET Data Provider are the same as those supported by the DB2 CLI driver. DB2 .Net Data Provider is fully .Net capable and excellent integration with development environments such Visual Studio .Net.

DB2 also provides .Net Managed Providers (Native .Net) that doesn't require ODBC bridges. DB2 supports advanced .Net functionality such as CLR stored procedures and UDFs.

Java Based Drivers:

MySQL Connector/J:

MySQL Connector/J is based on Sun's JDBC 3.0 API for connecting to the MySQL database server. However, the driver is non-compliant to the JDBC 3.0 specification due to several limitations such as lack of stored procedure support, etc. Connector/J is also limited in advanced functionality such as lack of in-place large object modification, no prepared statement support in the database, no save points, no updateable resultsets, etc.

MySQL Connector/J only comes as a JDBC type IV implementation and does not support open Java standards such as SQLJ.

DB2 Universal Driver:

DB2 Universal Driver is an architecture-neutral JDBC driver for distributed and local DB2 access. Driver architecture is independent of any particular JDBC driver-type connectivity or target platform, which allows for both Type IV and Type II connectivity in a single driver instance to DB2 platforms. In addition, platform specifics are abstracted to the lowest layers, so that the driver differences among the various DB2 platforms are minimized.

DB2 Universal driver is performance optimized and fully supports the JDBC 3.0 specification. The driver also has various key problem determination enhancements such as advanced tracing support, etc. This driver supports the SQLJ standard that enhances application performance through the use of static SQL as well as improves developer productivity.

Other Script-Based Drivers:

PHP/Perl DBI:

Perl and PHP can both be used to write dynamic Web pages that can connect and communicate with a database server such as MySQL. PHP has built-in functions for the task, while Perl has an open standard module called DBI that defines set of functions, variable and conversions for interfacing with databases. MySQL has database drivers (DBDs) available for Perl and PHP.

DB2 has support for most leading scripting interfaces that are popular in the open source community. The DBD::DB2 driver works with DBI and a DB2 client to access databases using Perl. PHP can be used as a method to access DB2 from web-based applications. PHP supports DB2 access using the Unified-ODBC access method, in which the user-level PHP communicates to DB2 using ODBC calls. Unlike standard ODBC, with the Unified-ODBC method, communication is directly to the DB2 CLI layer, not through the ODBC layer. DB2 drivers for REXX and Fortran are also available.

Web Services Support:

Current application development trends call for platform and end-point neutral access to information and service invocation in a heterogeneous environment. Web Services is an open-standard framework based on XML that can achieve high levels of loosely coupled yet well integrated applications.

Both DB2 and MySQL support the producing and consuming of web services and have features such as XML RPC support, etc. DB2 XML enhancements such as fast document shredding and retrieval as well as integration with development tools such as WebSphere Studio Application Developer make it an ideal choice for Web Services developers.

IDE Integration:

Both DB2 and MySQL developers can choose from a list of IDE environments that can be used to enable rapid application development using these databases as back-end.

However, DB2 delivers more plug-ins and advanced features for Visual Studio .NET, Eclipse, WebSphere Studio and Rational tools, and has more partner tools and solutions from which to choose from. Plug-ins for Eclipse provides visual tools and wizards that enable easy manageability of DB2 schema operations such as the creation of nicknames, tables, indexes, views and other database objects. Plug-ins support the DB2 type-4 JDBC driver and help integration by providing developers with combined online assistance for developing and debugging applications. DB2 supports Rational XDE Developer for Java. Rational XDE developer for Java plugs into the Eclipse development environment and provides an intuitive visual data-modeling interface for Eclipse.

Database Subroutines:

Subroutines are an important tool in code abstraction, logic encapsulation, code reuse, developer productivity, administration, as well as performance and security. MySQL does not support creating database sub-routines such as stored procedures. This support is being to be added to the MySQL database in v4.1.

DB2 supports advanced stored procedures in the engine from SQL to Java. Creating stored procedures does not require any external compilers in DB2 (starting v8.2) as the code is compiled and kept in the database itself. DB2 Development Center supports creating, running, and debugging stored procedures from SQL to JDBC (dynamic SQL using JDBC) as well as SQLJ (static SQL using SQLJ) stored procedures.

7. Database Clustering

A Database cluster, whereby multiple database servers are linked together, are used to improve query performance, system throughput, scalabiity as well as overall availability in case of failure. It is important to understand clustering as handled by both DB2 and MySQL in order to discuss scalability and availability characteristics of these products.

MySQL Clusters:

MySQL Cluster is a technology that MySQL AB purchased from Ericsson, and will be available in MySQL v5.0.1 (ETA mid 2005). In MySQL the SQL parsing engine and the storage or I/O features of the database are encapsulated in separate table handlers like MyISAM, InnoDB, etc. MySQL Cluster is yet another type of table handler which means that in order for designers to take advantage of MySQL Cluster, they could not utilize the other MySQL available table handlers. MySQL Cluster would appear to be similar to DB2's implementation of clusters. MySQL Cluster requires that the DBA define node groups that contain up to four storage nodes. Each storage node owns its own disk and memory. Table data is partitioned (hashed) across all the storage nodes.

Typical Cluster architecture contains applications connecting to multiple MySQL servers that parse the incoming queries. Then pass on the requests to the DB nodes that handle data storage. Each database node has a primary partition of data that it owns and could have secondary partitions that are kept transitionally consistent to the original partition using a two-phase commit protocol. This architecture not only leads to scale out bottlenecks but also will be wasteful due to excessive disk consumption. You can have a management node for administration. The following figure shows this relationship graphically.

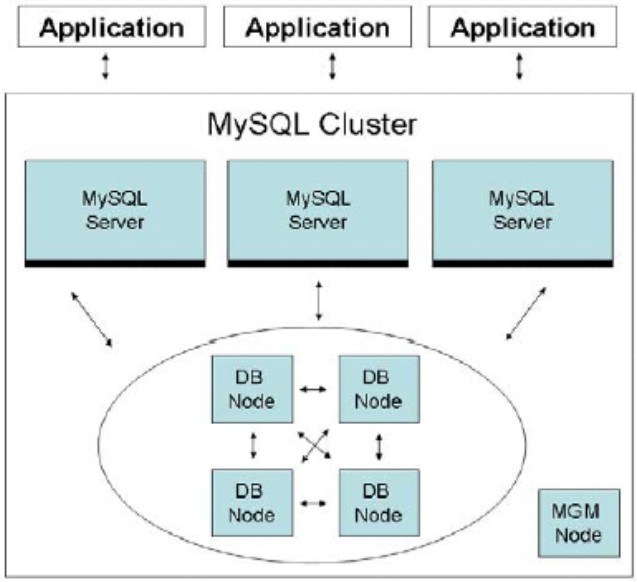

Figure 3: MySQL Cluster overview

DB2 Clusters:

DB2 clusters are also based on a partitioned instance. The feature of DB2 ESE to work with a partitioned instance is called the database-partitioning feature (DPF).

A database partition is part of a database that consists of its own data, indexes, configuration files, and transaction logs. A partitioned database is a database with two or more partitions. Tables can then be located in one or more database partitions. Processors associated (processors need not be associated with the partitions in an SMP server) with each database partition are used to satisfy database requests. Data retrieval and update requests are decomposed automatically into sub-requests, and executed in parallel among the applicable database partitions. Applications connecting to the database can connect to any of the database partitions, and this acts as the coordinator partition that then sprays the SQL functions to the corresponding data partition.

DB2 clustering using DPF is a shared-nothing architecture that utilizes all the computing resources of the servers that contain the partition. These partitions could be on physically distinct machines (cheap commodity based hardware) or on logical nodes of a large multiprocessing capable machine.

DB2 Clustering

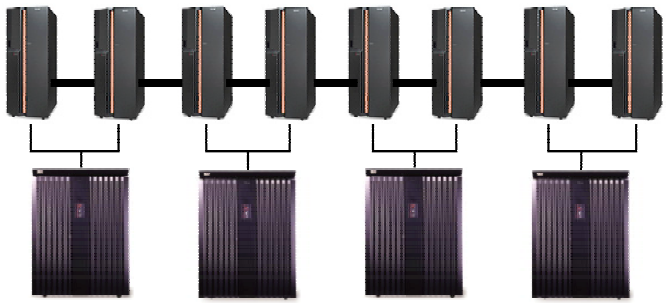

Figure 4: DB2 DPF overview

8. Availability

In the age of eBusiness, web sites must be available at all hours of the day and night. Planned outages should be masked such that they appear invisible to users. Unplanned outages due to failures can happen, but their impact on availability should be reduced or eliminated through features in hardware, the operating system and database software.

Unplanned Outage in Clusters:

MySQL Clusters are designed around availability requirements and seem to ignore scalability and performance considerations of a cluster. MySQL storage nodes in a node group replicate their respective changes synchronously to the other storage nodes in its node group using a two-phase commit protocol. Since storage nodes can run on different hosts, if a host machine or storage node fails, one of the remaining storage nodes in the node group can assume the duties of the failed storage node as it contains a synchronous copy of all the storage node data in memory for the storage nodes defined in its node group. However, this design makes it consume n-times as much disk capacity and slows the update to the cluster due to its synchronous update requirements.

DB2 can utilize active-active, active-passive or a cascading availability solution using the DPF. DB2 can also use hardware clustering technologies such as HACMP technology to create mutual failover pairs. Database partitions must have a dual path to the underlying disk so that if a partition fails, the partition can be restarted on its partner node.

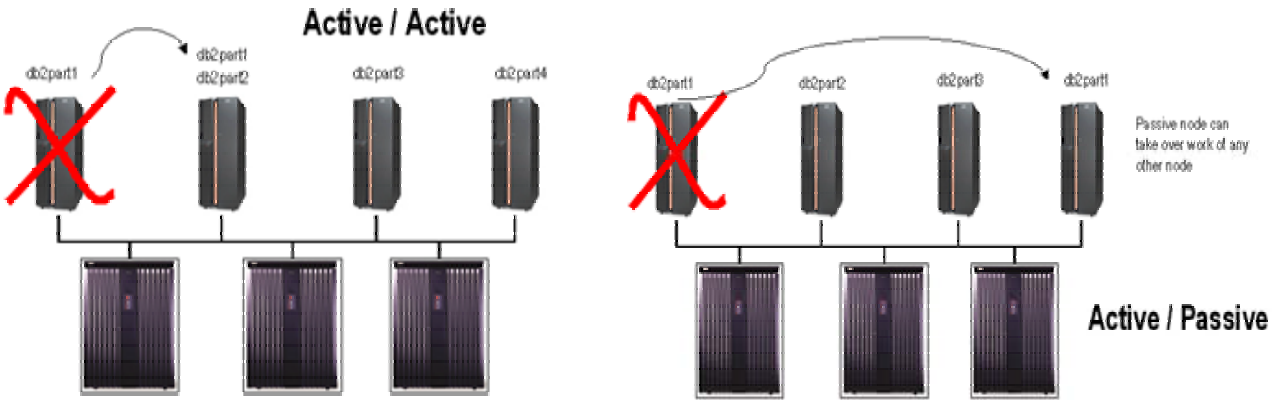

Figure 5: DB2 failover solutions

DB2 on Linux Integrated Cluster Environment (ICE) bundles Tivoli System Automation (TSA) software that uses a policy-based self-healing approach to increase system operations efficiency and improved applications availability. TSA provides fast detection of outages, sophisticated

knowledge about application components and their relationships, quick and consistent recovery of failed and affected resources and applications either in the same server or on another system within the cluster. This can also include activities like shutting down the remainder of an application or even a whole failed system, moving a TCP/IP address, or recovering data from a log.

Unplanned Outage in a Non-Clustered Environment:

MySQL has basic replication to be used for unplanned outages in a non-clustered environment. This is a traditional log-shipping based solution that has a single master, many target paradigm.

DB2 has had replication-based high availability solution support in the product since v6. DB2's latest release V8.2 has several high availability improvements that help recover from planned and unplanned outages.

DB2 HADR is a high availability solution for both partial and complete site failures. HADR is a next-generation log shipping based replication feature that protects against data loss by replicating data changes from a source database, called the primary, to a target database, called the standby. When a failure occurs on the primary, you can fail over to the standby. The standby then becomes the new primary. Since the standby database server is already online, failover can be accomplished very quickly, resulting in minimal down time.

Client ReRoute feature allows client applications to recover from a loss of communication with the server so that they can continue to work with minimal interruption. After a loss of communication, the client application attempts to reconnect to the server. If this fails, the client is then rerouted to a different server. You can use automatic client reroute with high availability disaster recovery (HADR) to make client applications connect to the new primary database after a takeover operation.

Q-Replication allows you to replicate committed transactional data from DB2 sources to targets by using two programs: Q Capture and Q Apply. The Q Capture program runs on the source system. The Q Capture program reads DB2 recovery logs for changed source data and writes the changes to WebSphere MQ queues and offers an elegant asynchronous replication solution.

Planned outage:

DB2 V8.2 also has several autonomic features and improvements that reduce overall planned outage time for system maintenance. Some of these features are online maintenance and execution of utilities such as online reorganization of data, online index rebuild, online backup as well as several throttleable utilities that limit utilization of system resources while maintenance proceeds in the background. Many of the traditional utilities such as load and import have also been performance optimized lower downtime. These features are discussed more in the Utilities section below.

MySQL has no such features as of yet.

9. Scalability

Successful applications get used more, and they need to handle the increased activity load. This may mean moving to more or faster processors on the same platform or moving to a different platform that allows for further growth. For large applications, it might be necessary to partition the data across multiple database servers.

Scaling up:

DB2 has both inter partition and intra partition parallelism to fully exploit both an SMP and a cluster of shared memory multiprocessor machines (SMP). Processing of queries and utilities can parallelize using these two methods. The unique partitioning architecture in DB2 allows you to run DPF on a single large SMP. By breaking the SMP up into logical database partitions (not LPAR or Domains so there is still only one version of the OS running) we can exploit all of the CPUs on the server even with a very low concurrent query environment. It is difficult to get a large 32/ 64 way to run at max capacity with a few queries (difficult to break a query up into many parts). But with logical partitioning in DPF we can exploit the server even with low numbers of end user queries.

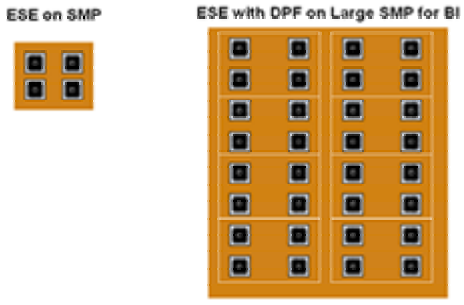

Figure 6: DB2 Scale up architecture

MySQL is multi-threaded and supports SMP machines. However, threading in MySQL is meant to handle one connection/query per thread. Thus, better results will be obtained only if simultaneous queries are executed. MySQL Clusters haven't been extensively tested on SMPs yet.

Scaling out:

Clustering multiple large SMPs together and you can get extreme scalability in a massively parallel processing architecture (MPP). DB2 led the Industry by introducing the first database to support Linux clusters where customers could start small & grow as their needs grow. DB2 ICE is an integrated, performance, optimized, pre-tested solution in which it is easy to add nodes to the cluster. Customers can get high-end clustering for web and data warehouse apps for low TCO using commodity hardware.

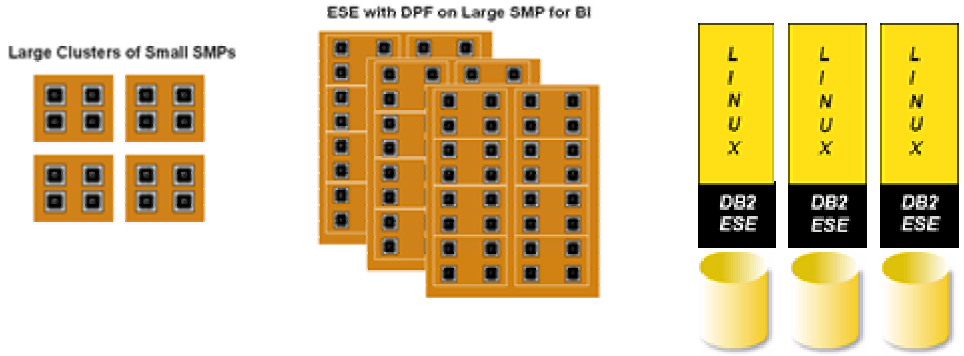

Figure 7: DB2 Scale out architecture

MySQL clusters use multiple data servers. Currently MySQL Cluster supports four nodes.

Data Scalability:

On the whole, only 2 Exabytes of information was created in the entire world in 1999 including paper, film, optical, and magnetic devices. To download 1 Yottabyte off of the Internet would take you 86 trillion years. The universe is only 20 billion years old.

In terms of support for largest databases possible, MySQL databases can contain maximum of 8 Million Terabytes in all, whereas, DB2 v8 can support up to 1 Million Exabytes database.

User Scalability:

DB2 has various features to handle multiple user sessions simultaneously. DB2's Connection Concentrator allows the server to handle large end user populations without overloading the server with too many processes. DB2 Connection Concentrator is a transaction-pooling agent shares the same physical connection to the database amongst incoming clients. DB2's memory footprint per agent for an incoming connection is much lower than most commercial vendors to handle high transaction volumes.

10. Performance

Performance is a key attribute of any database application. If performance is unsatisfactory, users will be dissatisfied, and more hardware may be required to process the same workload. Performance is related to scalability in that good performing applications use fewer resources and can handle more work using less hardware.

Benchmarking:

Benchmarking is a key metric for measuring performance for a database system.

DB2 has been focused on performance benchmarks that provide real value to customers some of these include: SAP, Siebel, PeopleSoft, Baan, etc. DB2 continues to deliver leading results in industry standard benchmarks such as: TPC-H, TPC-C, ECPerf, SpecjAppServer2004, etc.

MySQL has not delivered any significant industry benchmarks.

In-house testing using the Darkstar2 benchmark suite tested MySQL v4.1.1 alpha against DB2 v8.2 on similar hardware. These tests showed that on any database of significant size (4GB or greater) DB2 outperformed MySQL in all query tests, running the test suite over 350% faster than MySQL.

Optimizer Strength:

MySQL query optimizer is simplistic at best, as bigger queries are submitted, the time spent in query optimization may easily become the major bottleneck in the server performance.

Since 1970, IBM has been actively involved in research on relational query optimization. This presentation deals with some of the "Industrial strength" features of query optimizers in IBM DB2 family of products that evolved from IBM's research. This research has led to commercial DB2 optimizer that supports advanced features such as: Query transformations, access path evaluation, support for MQTs, etc.

Along with all the optimizer improvements DB2 has also added various other performance improvement features in the product. Some of them are: asynchronous I/O for better performance of page cleaners (for cleaning data out of buffer pools) and Pre-fetchers (for getting data from disk into memory), Linux kernel 2.6 feature exploitation, multiple buffer pools for various data pages, etc.

Parallelism:

Components of a task, such as a database query, can be run in parallel to dramatically enhance performance. MySQL parallelism is limited to a multi-threaded architecture. However, The nature of the task, the database configuration, and the hardware environment, all determine how DB2 will perform a task in parallel. The following advanced types of parallelism are supported by DB2: I/O with many OS channels and parallel container access, Query parallelism with inter and intra partition and inter and intra query options, and Utility parallelism exploiting multiple processors.

11. Manageability

Manageability refers to the time and cost of administering the database environment. The activities of the database administrator (DBA) include tuning, administration and problem determination.

Many smaller organizations do not have a DBA. For these organizations, an important requirement is to have a product that is easily administered, and as much as possible, be self-administered and self-tuning. This section looks at these aspects.

Tuning:

DB2 is ready to use out of the box. The install scripts can create an instance for you, and in a matter of minutes you can start working. The fun comes in the tuning of the database to meet your specifications. The default database parameters will work fine, but if you really want to get the most out of your system, you can tune the database to handle the type of workload it will experience. There are thee main levels of database tuning in DB2; Registry level, Instance level, and Database level. Each level affects the system differently, and can all work together to get the best performance out of the database. One of the new features in DB2 v8.1 is the much-improved Configuration Advisor. This tool is accessed from the Control Center, from a command line, or during the create index command and provides a very user-friendly way to get the most out of you system. It uses a combination of user input and current database characteristics to properly tune the system for the required workload.

With MySQL, right from the database design stage, you need to start thinking about future performance. During table creation, you must specify what type of storage structure will be used, with the default being InnoDB. There are various other types that can be used (BerkeleyDB, Memory (heap), MyISAM), each with its own advantages and disadvantages. Once the database server is up and running, MySQL can be configured through various command line parameters. To fully get optimal performance, the user needs to be very knowledgeable in system and database design, as there is no advisor functionality in the product.

Administration:

Database administration on the two platforms differs quite a bit as well. DB2 has a comprehensive set of integrated tools to handle all type of administrative tasks. GUI, command line, and web tools all come standard with the DB2 package. The main GUI administrative tool is the DB2 Control Center. This tool allows for easy manageability of all DB2 instances, nodes, and databases, and database objects.

MySQL has a basic command line interface that comes standard with it. This interface can be used, as well as the mysqladmin tool to handle all administrative tasks. There are two separately downloadable GUI administration products available (MySQL Control Centre and MySQL Administrator). These two products allow for graphical database and table management and server administration.

Tooling:

DB2 has with it a number of tools that make all types of database tasks easier. Most prominently is the Control Center for general administration mentioned above, but along with that, there are a number of other tools for specialized tasks. A few of them include: the DB2 Configuration Advisor (mentioned in the above Tuning section), DB2 Health Monitor to view overall database health and system status, DB2 Memory Visualizer to see how much memory is being used by the database, agents, and DB2 instance at any point in time, DB2 Design Advisor to simplify the optimization of workloads by advising on the creation of indexes, MQTs, MDCs, and table partitionin keys, DB2 Command Editor to run SQL and CLP commands, DB2 Visual Explain to graphically analyze access paths that DB2 chooses for your SQL statements, etc.

MySQL does not come with a set of integrated tools to ease the database administration process; however, it does include some useful capabilities in its MySQL Administrator product. Some of these features include: Single view dynamic health monitoring, performance optimization, simple backup and restore operations, and centralized log viewing.

Problem Determination:

DB2 provides the user with many problem determination features built into the product. The server allows for complete logging and tracing capabilities, and some built in tooling for quickly determining and fixing any problems that might occur. The admin notification log (notify.xxx) , diagnostic log (*db2diag.log)*, combined with the *db2support, db2dart, db2look, and db2trc* tools all make finding and correcting database issues an easy task for any DBA.

MySQL offers various log files for use in problem determination. These logs (Error log, Binary log, General Query log, Slow Query log, Update log, ISAM log) must be hand-searched to find underlying causes of any problems that might occur.

Autonomics:

DBA's can reap benefits from the broadened autonomic computing solutions offered in DB2 v8. These solutions automate and simplify potentially time consuming and complex database tasks.

Some of the autonomic capabilities available in DB2 include:

- DB2 Design Advisor - assist in making optimal and comprehensive database design decisions
- Automatic Maintenance - automate database maintenance such as backup, reorg, and runstats
- Health Center Recommendation Advisor – help resolve DB2 health alerts
- Self-tuned and throttled backup and restore operations

MySQL has no such autonomic functionality.

12. Security

Security mechanisms must prevent unauthorized access. They should also ensure the ease of safe use of the product such that appropriate security levels can be achieved with a small amount of effort and confidence that resources are secured properly.

Users Security:

Access to an instance or a database first requires that the user be authenticated. The authentication type for each instance determines how and where a user will be verified. DB2 and MySQL support OS authenticated users and has limited authorization in the database. MySQL does not support writing of external routines or integration of external authentication engines with the database.

DB2 also support user exits for integration with external or custom authentication mechanisms such as Kerberos. Kerberos is used when both the DB2 client and server are on operating systems that support the Kerberos security protocol. Using the Kerberos security protocol enables the use of a single sign-on to a remote DB2 UDB server.

Both MySQL and DB2 have similar authorization mechanisms in the database based on GRANT, REVOKE and user Roles. DB2 also supports external authorization mechanisms such as Tivoli Access Manager, etc.

Encryption:

MySQL provides server-side encryption.

DB2 supports 128-bit server, client and on-wire encryption. DB2 has improved the security of user data with authentication types that allow you to manage the encryption of user data. One new authentication type, *SQL_AUTHENTICATION_DATAENC*, requires connections to use the data encryption. The other new authentication type, *SQL_AUTHENTICATION_DATAENC_CMP*, allows for a compatibility mode with down-level products that do not support the new authentication type. There is also column level encryption in DB2.

Certifications:

DB2 has the Common Criteria (CC) certification compliant with evaluation level EAL4 certification. MySQL has no industry standard certifications for security and since the database binaries are generally available, it is prone to attacks from the hacker community. Code bugs have also led to various security vulnerabilities and attacks in the recent past.

13. Utilities

Database tools and utilities play a critical role in overall ease of administration as well as cost of ownership of a system. These utilities can help improve overall productivity as well as robustness of the database system.

Backup and Restore:

MySQL has limited functionality available for any production environment to setup efficient backup and restore policy for the database systems.

mysqldump:

The main methodology available for database backups and restore in a MySQL database is mssqldump.The utility dumps existing data and table structures into an ASCII based text file. This data and table structures can then be used to repopulate another MySQL server.

mysqlhotcopy:

The mysqlhotcopy utility takes incremental backup of the database and restores it on a local target. More specifically, it will lock the tables, flush the tables, make a copy, and unlock the tables. Although it is the fastest method available for backing up a MySQL database, it is limited to backing up only those databases residing on the same machine as where it is executed.

DB2, on the other hand, has a wide variety of built in Backup and Restore utilities. These range from text-based to binary-based ultra high-speed backup and restore tools as well as wizards and advisors. Some of the advanced features offered are policy based backup support, log file inclusion in the backup images, online backup, automatic backup, performance optimized self-tuning backup, status of utilities being executed, etc.

db2 backup db:

This DB2 command takes a backup of the database at various levels, database, tablespace, and etc. based on high speed binary format dump from the engine and is the primary method of developing a backup strategy. The Backup process is automatically tuned in V8.2, so the DBA does not need to worry about buffers and parallelism.

db2 import / export / load:

DB2 offers utilities to export data and object definitions out of the database and then recreate them to a target database. These utilities are a less efficient method used for database backup and restore and are similar to what MySQL offers. Import and export are SQL based utilities and offer a huge range of customizable options such as online options, optimization options, as well as

concurrency and constraint options. Load is a not SQL based and a much faster option for data import to DB2 server.

db2 split mirror backup:

Hardware facilities exist for making a second “mirror” copy of a database. DB2 provides a set of utilities and commands for splitting the mirrored copies, such that the primary and secondary are separated. This saves downtimes for backups, allows for fast cloning of data, is a fast implementation of standby failover. This involves suspending I/O to a database, splitting the mirror, and resuming I/O to the database.

Other Utilities:

Database statistics:

Information on database table, indexes, etc. such as the number of rows, the number of pages used, the sizes of indexes, the value distribution within columns or indexes, and so on can be used by the SQL Optimizer to determine the best strategy for executing SQL statements. MySQL does not have a utility to update the statistics of the system for optimal performance of the engine. DB2 offers the *runstats* utility that can help the optimizer with latest statistics on the system.

Data Organization:

The table and index data in the system often needs reorganization by redistributing the data/index into unfragmented, physically contiguous pages. This cleanup is very helpful in efficiency and reduction in I/O time for a SQL query. DB2 offers *reorg* and *reorgchk* utilities to help with scattered data. MySQL does not have a comparable option.

14. Code Refresh

Patches/Upgrades:

DB2 releases a set of fixpacks every few month to address bug fixes and functionality enhancements. The fixpacks can be installed on top of the old releases, with nothing beside a database restart to implement the changes. Every so often, DB2 will come out with a major release that introduces new features and concepts to the database.

With new version releases, DB2 provides a toolset for easy database migration between versions. To transparently update specific databases, tools such as *db2updv8* can be used.

MySQL has a version point release cycle, with minor fixes and updates being released every few months, and with major releases introducing new functionality every year or so. The current production version of MySQL is 4.0, with a 4.1 in beta, and 5.0 in alpha. The process to upgrade a MySQL server is also fairly straightforward, but to take advantage of all the new features, many additional steps need to be taken. The server itself can be installed overtop the old version, but then a variety of scripts and command parameters need to be changed to take advantage of new features. The process though is being streamlined, as the upgrade path from 3.23 to 4.0 is much more complex than the planned 4.1 to 5.0 migration.

Migration from other RDBMS's:

DB2 has a very comprehensive set of migration tools available to port database schema and data from different vendor's databases to DB2. There are 3 separate migration toolkit products that help facilitate database migration from Oracle, SyBase, Microsoft SQL Server, Informix, and of course, MySQL. These toolkits are augmented by the vast amounts of papers and books written on the subject. IBM Redbook, DB2 Developer Domain Tutorials, and various other industry articles and publications can help with this task.

The toolkits are available from the DB2 Migrate Now website at:
http://www-306.ibm.com/software/data/db2/migration/.

There is also an IBM Redbook about specifically migrating MySQL databases to DB2 located at http://publib-b.boulder.ibm.com/redbooks.nsf/RedbookAbstracts/sg247093.html?Open.

The only integrated migration product MySQL offers is one to perform the conversion of mSQL C API function calls to their MySQL equivalents. No native software is available to facilitate the move from different database vendors to MySQL.

15. Total Cost of Ownership

License Cost:

DB2 has a simple licensing structure: per-user or per-CPU based agreement. For a small number of users, DB2 is very competitive in licensing costs. Where the true TCO benefits of DB2 comes to the forefront are on larger, more complex systems. The license costs of DB2 versus other enterprise databases are significantly less. DB2 Express licensing starts at $499 per server and moves up based on configuration.

MySQL runs two separate pricing schemes:

- **GNU General Public License (GPL)** This license is valid only if your project is 100% GPL. This means that your source code must be shipped with your product.
- **Commercial License** MySQL sells commercial licensees for its product on a per server basis. The cost allows for unlimited users to connect, and unlimited processors on the server.

For the commercial licensing scheme, MySQL charges $495 per server for the full transaction-capable database (MySQL Pro).

Support/Maintenance:

DB2 offers world-class software maintenance packages to support its products. The standard IBM Software Group maintenance package costs 25% of the base license fee per year, and includes all access to software upgrades and fixes, and skilled telephone support for its products.

MySQL charges for support of its licensed products. This is based on level of service and length of support. These services range in price from $1500 per-year for the basic entry-level technical support to $48000 per-year for the advanced premium support.

16. Conclusions

Even though MySQL has grown in popularity, it still lacks the breadth of features and depth of technical ability that an advanced database system such as DB2 offers.

DB2 has stable and proven clustering support. MySQL Cluster, which will be released sometime in 2005, will partition data across various storage nodes and keep redundant copies of the data synchronized using a two-phase commit protocol. This would provide a similar availability solution as DB2 using DPF. However, this introduces performance bottlenecks and has limited scalability (up to 4 nodes). DB2 can support up to 1000 nodes using DPF is an ideal choice for clustered databases.

MySQL has no comparable offerings to advanced autonomic features in DB2 such as self-managing computing solutions that automate and simplify potentially time consuming and complex database tasks. Some examples include performance optimization, online maintenance capabilities, system health monitoring capabilities, etc.

DB2 also leads in maintainability of the systems and has simple tools for upgrading databases from earlier versions to the newer ones such as d*b2updv8*. Upgrading MySQL versions still requires data extraction and recreation. Also, migration from other database products such as MySQL to DB2 is simplified with the migration toolkits available free of charge.

In terms of cost of ownership, DB2 is still an industry leader with best return on investment from entry-level database to high-end clustered solutions. DB2 offers some of the best manageability and autonomic features in the engine that not only reduce cost of setup, etc. but also help protect the investment by drastically reducing the overall downtime of the system.

All the technical features combined with an extremely competitive pricing strategy and virtually limitless performance and scalability makes DB2 the ideal choice for database systems from small and mid-sized business to enterprise level consumers.