White Paper

embarcadero

InterBase® and MySQL™
A Technical Comparison

By Bill Todd

Embarcadero Technologies

November/2010

Americas Headquarters
100 California Street, 12th Floor
San Francisco, California 94111

EMEA Headquarters
York House
18 York Road
Maidenhead, Berkshire
SL6 1SF, United Kingdom

Asia-Pacific Headquarters
L7. 313 La Trobe Street
Melbourne VIC 3000
Australia

CONTENTS

EXECUTIVE SUMMARY

Embarcadero Technology InterBase® is a powerful, SQL-compliant database that is often considered for embedding in applications and for application-specific uses. Savvy developers and application architects who take the time to examine InterBase closely find that it offers substantial advantages over MySQL.™ Those advantages include:

- Online backup
- Faster crash recovery
- Journaling
- Domains for easier design and maintenance
- Easier access control with roles
- Views to provide virtual tables and row-level access control
- Simpler string data types that conform to the ANSI standard
- More powerful and flexible default values
- Easier data validation using check constraints
- Superior performance-monitoring tools
- Streamlined configuration options

This white paper discusses these advantages in detail.

INTRODUCTION

Many database applications being developed today require a database that can be embedded in the application or a database that can easily be deployed to and supported at remote sites. InterBase has all of these attributes, plus the high performance, reliability, and sophisticated feature set that developers would expect of a mature database that has been in production for many years at millions of sites around the world.

InterBase is easy to deploy, easy to learn and use, and requires virtually no maintenance or care in production environments. InterBase is one-fifth the installation size for MySQL, yet it provides SMP support, cross-platform support, a sophisticated cost-based query optimizer, row-level locking, instantaneous recovery from a server crash, transaction support with isolation levels that guarantee a consistent snapshot of your data at a point in time, stored procedures, triggers, views, check constraints, a rich SQL dialect, online backups, online metadata changes, journaling and the ability to trigger events in the client application from triggers in the database.

InterBase lowers your development and maintenance costs with:

- Domains
- Check constraints
- Roles
- Easier configuration
- Superior performance monitoring tools

- Journaling

Without these features, you would spend more time writing client-side code to perform functions that should be centralized in the database and more time on maintenance tasks such as controlling access, recompiling and redeploying client applications, and configuring the database for site-specific load requirements.

CHOOSING THE RIGHT DATABASE FOR YOUR PROJECT

It is not difficult to create a checklist of attributes that your database project needs. However, evaluating databases against a simple feature list is not enough. The most critical step in choosing the right database for your application is to compare the behavior of the databases you are considering in real-world situations your application will encounter. This paper examines the behavior of InterBase and MySQL in a number of situations you are likely to face in today's business environment.

MYSQL™: TOO MANY CHOICES?

Situation: You are developing an application that will deliver substantial savings to your company. The application must be deployed quickly, so minimizing learning and development time is a major consideration.

MySQL is not one database but four, so the first decision you have to make is which of the four available database engines to use[1]. Each of the engines uses a different table structure, a different file format on disk, and offers different features. It is not just the data access and concurrency control features that are different; the administrative requirements are also different. For example, assume you know that the database may have to be moved to a different directory on the server as hardware and operating systems are updated. The BerkeleyDB database engine embeds the full path to the table in the table header, so you cannot move the table files to a new directory. You must dump the data, create a new database in the new location, and reload the data. InnoDB and MyISAM databases, however, can be moved to a new directory.

Because the capabilities of database engines vary widely, you must carefully compare your application's requirements to the capabilities of each engine before you make a decision. This evaluation adds to the time and cost of your project.

The following paragraphs provide a brief overview of the different database engines. The ISAM engine was the original database engine for MySQL. It has been deprecated and replaced by the MyISAM engine, so ISAM can be ignored.

[1] MySQL 5.5 Reference Manual, Chapter 13.

The MyISAM engine was designed to provide high performance for SELECT queries and is the only table format that supports full text indexing. MyISAM also offers the most flexible implementation of auto-incrementing fields. However, MyISAM does not provide transaction control or declarative referential integrity. The MyISAM engine processes queries sequentially, giving INSERT, UPDATE, and DELETE statements priority over SELECT statements. Concurrent access is provided by exclusive table locks. When an INSERT, UPDATE, or DELETE is being executed, no other user can read or write any row in the table that is being changed.

Today's database applications most often consist of a mixture of reads and writes. If your application needs to analyze data or produce reports on large tables, the SELECT statements likely will run for a relatively long time. Suppose you have an order-entry system. Printing invoices requires that you query the Customer table to get name, address, and other customer information. You also need to join the Customer table to the Order table, the Item table, and the Part table to get information about the orders, the line items on the order, and the price and description of each part. Because MyISAM does not allow updates to tables that are being read, no user can update the Customer, Order, Item, or Part table until the invoice query has finished.

If your application requires a lot of INSERT, UPDATE, and DELETE statements, users may have trouble getting SELECT statements to execute. Because MyISAM processes statements sequentially and gives SQL statements that change the data priority over SELECT statements, users running SELECTs may see poor performance when these are competing with large numbers of updates.

The BerkeleyDB (BDB) database engine provides transaction support, but the only transaction isolation level is read-committed. This means the only way to get a consistent view of data is to lock tables so no updates can occur. Because BDB uses page-level locking, it offers a higher level of concurrency than MyISAM tables, but it still locks an entire page of data to update a single row. BDB does not provide declarative referential integrity or full text indexing. In addition, the full path to each BDB table file is included in the file. This makes it impossible to move the files to a new directory. If you need to move your BDB database, you must dump all of the data, create a new database in the new location, and reload the data.

The most capable database engine for MySQL is the InnoDB engine originally developed by InnoBase Oy of Helsinki, Finland and later purchased by Oracle Corporation. InnoDB provides transaction control, declarative referential integrity, multiple transaction isolation levels, and row-level locking. With MySQL 5.5, InnoDB will be the default engine for MySQL. Because the InnoDB engine is the most feature-rich, this paper assumes that MySQL is being used with the InnoDB database engine unless otherwise noted.

InterBase provides a single, integrated SQL database server with a single consistent set of features. Comparing InterBase to your requirements is easy.

ONLINE BACKUP

Situation: You are creating a retail point of sale application. Each store must back up its database at least once per day. The store manager takes a copy of the backup files home as part of your disaster-recovery plan. Many of your retail outlets are open 24 hours a day.

The only MySQL database engine that supports online backups is InnoDB[2]. To back up InnoDB databases while the database is online, you can purchase Enterprise ZRM for MySQL from Zmanda at a cost of $300 for a 1-year license. You could also select MySQL Enterprise Edition for $5,000/year.

Online backup is built into the InterBase engine. You can back up the database anytime, and the backup will give you a logically consistent copy of your data as it was at the instant the backup started. You can add backup capability to your application by calling an InterBase API function, run backups manually using the IBConsole (the InterBase GUI development tool), or use the command-line backup tool to add backups to batch files or scripts. The batch files or scripts can be run at a set time using the operating system's scheduler.

RECOVERY SPEED

Situation: The database server in your regional office in Dallas crashes because of a power failure. When the server is restarted, the database must automatically and quickly recover to a consistent state.

The time to recover on restart of MySQL depends on the size of the transaction log that must be processed to roll back the active transactions. This might be a few minutes or longer, depending on how the database administrator has configured the server.

Recovery on restart is instantaneous with InterBase, because no changes to the database are required to roll back active transactions. Instead, InterBase simply changes the status bits for each active transaction rolled back and leaves the record of versions created by the rolled back transaction in the database. The versioning engine automatically ignores these record versions, which are automatically removed by the garbage collector as the records are visited during normal database use.

DOMAINS

Situation: Your company has decided to convert to a new industry-standard part number system that is being adopted by most of your customers. The new part numbers are larger than the ones you are currently using. The part number field appears in many tables throughout your database.

With MySQL, you must change the part number field in each table.

[2] MySql 5.5 Reference Manual, Chapter 6.

InterBase supports domains. A domain is a custom data type that you define based on one of the native data types supported by InterBase. For the original part number field you might have used:

CREATE DOMAIN PART_NO_TYPE VARCHAR(10) NOT NULL;

Once you have added the domain to your database, you use it as the data type for the part number in each table. Suppose the new part number is 14 characters long. All you have to do is:

ALTER DOMAIN PART_NO_TYPE TYPE VARCHAR(14)

and you are done.

ROLES

Situation: You are implementing the first phase of a multi-module application that will support 200 simultaneous users. You can divide the users into three groups based on their access rights to the data. You know that these rights will change as more tables are added to the database when the next phase of the project is implemented.

MySQL supports access control at the user level. If you need to change access rights, you must make the change for each individual user.

InterBase supports roles. Using InterBase, you can create three roles and grant the necessary rights to each role. Next, grant the appropriate role to each user. To change the rights for all users in one group, just change the rights for the role.

STRINGS

Situation: You need to add a string field to a table to store some descriptive text. Users estimate that the text will never be longer than one or two lines, so you define the field as VARCHAR(150). Later, changing requirements dictate that the field be expanded to handle 500 characters.

MySQL has six string field types; CHAR, VARCHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT. The maximum size for CHAR is 255 bytes. The four text types have differing maximum sizes from 255 bytes to 4 gigabytes. While CHAR and VARCHAR fields can be indexed, the four text field types cannot. The best solution with MySQL is the TEXT type, which can store up to 64 kilobytes.

To add to the confusion, in some cases, MySQL changes the data type[4]. For example, you cannot have a VARCHAR field with a length greater than three and a CHAR field with a length greater than three in the same table. If you try, MySQL changes the CHAR to a VARCHAR. The same thing happens if you try to have a CHAR with a length greater than three and a text

[4] MySQL 5.5 Reference Manual, Section 10.1.3

field in the same table. The exception is that CHAR columns whose length is less than four will not be changed to VARCHAR. In fact, VARCHAR columns with a size less than four will be converted to CHAR.

With InterBase, your choices are easier. InterBase provides just three types, CHAR, VARCHAR, and text blob, and the type you specify is the type you get. CHAR and VARCHAR can hold up to 32 kilobytes of data, and the CHAR data type correctly preserves trailing spaces.

DEFAULT VALUES

Situation: You need to log the date and time each record in a table is created.

Although MySQL lets you specify a default value for a column, the default is limited to a literal value. Because you cannot call a function, there is no way to make the default in a DATETIME column the current date and time.[5] MySQL so you must assign the current date and time in the client application.

MySQL has another field type, called TIMESTAMP, for storing a date and time. If you do not specify a default value for the first TIMESTAMP column in a table, the default will be the current date and time. However, the TIMESTAMP type has a limited range that ends in the year 2038.[6] With MySQL 5.5, the TIMESTAMP format that was used prior to MySQL 4.1 is no longer supported. This may mean that applications written for earlier versions of MySQL may need to be modified to work correctly with MySQL 5.5 adding unnecessary cost to your application development.

In addition to literal values, InterBase lets you specify the current date and time as the default for any TIMESTAMP column. You can also specify the current user name as the default for a VARCHAR or CHAR column.

CHECK CONSTRAINTS

Situation: You want to centralize business rules in the database to ensure consistent enforcement across all applications and to make maintenance easier.

MySQL does not have check constraints.[7] The only column constraint you can define in MySQL is NOT NULL, so you must do all data validation in your client applications.

To make data validation easy, InterBase provides both triggers and check constraints. Using a check constraint as part of a column's definition, you can determine whether the column's value is between two values, is like a value, is in a list of values, contains a value, or starts with a value. You can also combine multiple conditions using AND and OR. InterBase makes enforcing business rules in the database a snap.

[5] MySQL 5.5 Reference Manual, Section 10.1.4
[6] MySQL 5.5 Reference Manual, Section 10.1.2
[7] MySQL 5.5 Reference Manual, Appendix C, Section C.6.3.4

MONITORING

Situation: Users complain at irregular intervals that system performance is poor. You suspect a user is doing something that causes a very long-running, CPU-intensive transaction. You must identify the user and the SQL statement that causes the problem.

MySQL provides very limited monitoring tools. There is no way to see the transactions for a connection or the statements for a transaction. You cannot see the start time, the total number of inserts, updates, deletes, the number of reads that came from the cache versus disk, or other important statistics at the connection, transaction, or statement level.

InterBase provides a complete suite of performance-monitoring tables. You can view this information using the interactive performance monitor in IBConsole, or you can construct your own queries to get only the information you need. The performance monitoring tables provide detailed information about connections, transactions, and individual statements as well as open tables and stored procedures. The available information includes the start time, CPU time, total rows selected, inserted, updated, and deleted as well as the number of page reads from disk, writes to disk, and fetches from the cache.

Using these tables, you can easily identify the statements using the most CPU cycles and determine the IP address of the connection that is executing the statement. You can also view the SQL statement that is being executed. Everything you need to know for capacity planning and analyzing system use is immediately available.

CONFIGURATION

Situation: Your application will be deployed to sites where the number of users and volume of data varies widely. You need a database that does not have to be tuned for varying loads and does not require a highly trained database administrator (DBA) to establish the initial configuration settings.

MySQL includes over 300 system variables plus numerous configuration parameters that can be set to control various aspects of its operation. For example, MySQL has a query cache, key buffer (index cache), table cache, InnoDB buffer, and InnoDB Log buffer. You must decide how to allocate memory among these caches to get the best performance. While this may seem to offer a great deal of flexibility, consider the nightmare of deploying an application to hundreds of end customers where system variables could be set to any number of configurations which may adversely affect application performance.

InterBase is almost completely self-tuning. InterBase uses a single, unified cache and dynamically allocates space in the cache for data pages, index pages, query caching, sorting, and other purposes based on the mix of statements being executed at the time. Although the InterBase configuration file contains 31 settings, there are only three that you may need to change in any but the most unusual circumstances. One is the cache size, and the others are CPU affinity and whether hyperthreading support is enabled. CPU affinity lets you specify which processors on a multiprocessor machine InterBase uses, and hyperthreading enables InterBase's support for hyperthreaded processors.

JOURNALING

Situation: Your disk drive containing your InterBase database has crashed. You've replaced the disk and now need to recover your data.

MySQL maintains a General Query Log that contains statements received from clients and a Binary Log that contains all statements that change data. These statements can be used to 'replay' the database activity and attempt to recreate a database. However, because these are simply logs of SQL activity, differences in the environment, environment settings and timing of replayed transactions mean that there is no guarantee the original database can be recreated.

With the InterBase, journals can be maintained which log changed data pages. When enabled, journal archiving allows databases to recover from a complete loss of an InterBase server machine to within a few minutes of when the disaster occurred, or to a specific point in time. Because InterBase journals tracked the changed pages, an archived journal file can be used to perform point-in-time recovery. Point-in-time recovery uses the internal timestamp that is recorded on each transaction in the file, which allows you to, if desired, recover to a specific date and time. There is no dependency on the environment, environment settings or timing.

FEATURE COMPARISON

This table compares the features of InterBase and MySQL. This list is not exhaustive, but instead focuses on features important to embedded applications that are deployed to remote sites and applications that consist of a mixture of update and long read transactions.

Feature	InterBase® 10.0	MySQL™ 5.5 Beta
Cross-platform support	✓	✓
Consistent snapshot of data without blocking updaters	✓	✓
No conversion deadlocks	✓	✓
Cyclic deadlocks only at row level	✓	✓
Locks are not escalated above row level	✓	✓
Row-level locking	✓	✓
Does not escalate locks	✓	✓
Supports transactions	✓	✓
Supports savepoints	✓	✓
Automatically generates sequential keys	✓	✓
User-defined functions	✓	✓

Performance-monitoring tools	✓	Limited
Stored procedures	✓	✓
Triggers	✓	✓
Views	✓	✓
Check constraints	✓	No
Events	✓	✓
Backup while database is in use	✓	$300 option
Declarative referential integrity	✓	✓
Journaling	✓	No
SMP Support	✓	✓
Are roles supported	✓	No
DATA TYPES		
Integer 16	✓	✓
Integer 32	✓	✓
Float	✓	✓
Double	✓	✓
Numeric	✓	No
Decimal	✓	✓
Char	✓	✓
Varchar	✓	✓
Text blob	✓	✓
Blob	✓	✓
Date	✓	✓
Time	✓	✓
TimeStamp	✓	✓
Boolean	✓	No
NETWORK PROTOCOLS		
TCP/IP	✓	✓
Named pipes (NetBEUI)	✓	✓
IPX/SPX	✓	No
RESOURCE REQUIREMENTS		
Disk space	40 MB / 15MB	260 MB
Memory	32 MB	2GB

CONCLUSION

With InterBase you get:

- Online backup
- Faster crash recovery
- Data types for accurate financial calculations
- Domains for easier design and maintenance
- Easier access control with roles
- Views to provide virtual tables and row-level access control
- Simpler string data types that conform to the ANSI standard
- More powerful and flexible default values
- Easier data validation using check constraints
- Superior performance monitoring tools
- Far fewer configuration options
-

InterBase gives you the features you need to create a robust application at minimum total cost in minimum time.

ABOUT THE AUTHOR

Bill Todd has co-authored four database programming books and is the author of more than 100 articles. He has presented more than two dozen papers at developer conferences in the U.S. and Europe.

Embarcadero Technologies, Inc. is the leading provider of software tools that empower application developers and data management professionals to design, build, and run applications and databases more efficiently in heterogeneous IT environments. Over 90 of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero's award-winning products to optimize costs, streamline compliance, and accelerate development and innovation. Founded in 1993, Embarcadero is headquartered in San Francisco with offices located around the world. Embarcadero is online at www.embarcadero.com.