Borland® InterBase® and MySQL™

A technical comparison

A Borland White Paper

By Bill Todd, The Database Group

March 2004

Borland®

Contents

Executive summary

Borland® 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:

- Triggers
- Stored procedures
- Client-side events
- 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
- N-way replication

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, replication, and the ability to trigger events in the client application from triggers in the database.

InterBase lowers your development and maintenance costs with:

- Stored procedures
- Triggers
- Views
- Domains
- Check constraints
- Roles
- Easier configuration
- Superior performance monitoring tools
- N-way replication

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.

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, declarative referential integrity, stored procedures, or triggers. 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

[1] MySQL Reference Manual, Section 7

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 from InnoBase Oy of Helsinki, Finland. InnoDB provides transaction control, declarative referential integrity, multiple transaction isolation levels, and row-level locking. 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.

Stored procedures

Situation: Your database supports a manufacturing operation and several related applications. Several of the applications need to expand the bill of material for a product.

Because MySQL does not support stored procedures[2], the code to expand the bill of material must be compiled into each application. If you need to change the code, each application must be changed and recompiled.

Using InterBase you can write a stored procedure that takes the part number as a parameter and returns the expanded bill of material. Because this procedure is part of the database you can change it easily without changing the client programs. You can even change the stored procedure while the database is in use. Because all changes made by the stored procedure are part of the transaction that called it, the work done by the stored procedure will commit or roll back with the rest of the transaction.

Triggers

Situation: You are developing a financial application that requires an audit trail. All changes to the major data tables must be logged to corresponding change log tables. Log records must include the original value of each field, the data and time of the change, and the name of the user who made the change.

Using MySQL, the logging code must be built into each client program that uses the database[3]. There is no way to have the database handle the change logging automatically.

[2] MySQL Reference Manual, Section 1.8.4.4
[3] MySQL Reference Manual, Section 1.8.4.4

With InterBase, you can add triggers to the data tables. You can designate whether the trigger fires before or after the INSERT, UPDATE, or DELETE event occurs. 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. With InterBase, it is a snap to add after triggers to each data table. The triggers insert audit records in the change tables automatically. No code is required in any client application, and you can change the triggers without changing the client programs. Like stored procedures, InterBase triggers are part of the transaction that caused the INSERT, UPDATE, or DELETE event. If you roll back the transaction, the changes made by the triggers also will roll back.

Events

Situation: Your order-entry application must relieve the inventory for each item as it is added to the order. If relieving the inventory causes the quantity on hand to drop below the reorder point, the inventory control program, which is running on another computer, needs to be notified.

Because MySQL does not support triggers, there is no way to fire a user-defined event[4].

Using InterBase, you add only the POST_EVENT command to your trigger. In the inventory control application, you add an event handler component and set its properties to register interest in the event. All of the communications and callback mechanisms are built into InterBase. All your team has to code is the business logic. And there's no need to worry about the inventory system getting erroneous information if the user rolls back the order entry transaction after adding several items. InterBase does not fire the events you post until the transaction commits.

[4] MySQL Reference Manual, Section 1.8.4.4

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. To back up InnoDB databases while the database is online, you must purchase the InnoDB Hot Backup utility from Innobase Oy at a cost of $450 for a 1-year license or $1,150 for a perpetual license[5].

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 is 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.

A data type for money

Situation: Your application tracks a large investment portfolio and must perform financial calculations and summarize fields containing dollar amounts.

MySQL does not provide a data type that supports accurate financial calculations. Although MySQL has a DECIMAL data type, which stores the value as a string, all calculations are performed by converting the stored value to double-precision floating point[6]. Because floating point notation cannot represent most decimal fractions accurately, errors can occur when you perform calculations such as summing a large number of values. The only alternative is to store all monetary amounts as pennies in an integer field and convert back to dollars in your client application when the amounts are displayed.

InterBase supports both the NUMERIC and DECIMAL data types to provide accurate calculations with real numbers.

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.

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:

[5] http://www.innodb.com/hotbackup.html
[6] MySQL Reference Manual, Section 1.7.6.2, and private communication from Innobase Oy

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 multimodule 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.

Views

Situation: Users in your human resources department need the ability to run ad hoc queries and reports. You must ensure that users below the level of Director of Executive Compensation are limited to viewing data for employees whose pay grade is 10 or below.

MySQL has no mechanism for restricting access to rows in a table.

Using InterBase, create a view on the employee table that includes only those records with a pay grade less than or equal to 10. Grant the human resources staff role rights to the view but

not to the employee table. Now the HR staff members have a virtual employee table that does not include records for any executives.

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 and VARCHAR is 255 bytes. The four text types have differing maximum sizes from 255 bytes to 4 gigabytes. The MySQL CHAR type does not follow the ANSI standard in that the value is not padded to its full length with spaces[7]. 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.[8] 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 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.

---

[7] MySQL, ISBN 0-7357-1212-3, p. 114
[8] MySQL Reference Manual, Section 6.5.3.1

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.[9] MySQL does not support triggers, 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 2037.[10]

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.

In addition to not having triggers, MySQL does not have check constraints.[11] 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

[9] MySQL Reference Manual, Section 6.5.3
[10] MySQL, ISBN 0-7357-1212-3, p. 128

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.

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.

[11] MySQL Reference Manual, Section 6.5.3

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 114 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, Inno DB buffer, and InnoDB Log buffer. You must decide how to allocate memory among these caches to get the best 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.

Replication

Situation: Your company headquarters is in Dallas. You have sales offices in Houston, Austin, and El Paso. You have inside sales reps entering orders at the headquarters office and field sales reps entering orders in each field office. All of the orders entered in the field offices must be replicated to the headquarters database, and all of the orders entered at the headquarters must by replicated to the field office that entered the order.

MySQL supports asynchronous one-way replication. You must designate one database as the master and all others as slaves. Data can be replicated only from the master to the slaves.

With the InterBase N-way replication, you can replicate the orders from headquarters to the appropriate field office and from the field offices to headquarters.

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® 7.1	MySQL™ 4.1 alpha
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	✓	
Events	✓	
Backup while database is in use	✓	$1,150 option
Declarative referential integrity	✓	✓

Feature	InterBase® 7.1	MySQL™ 4.1 alpha
REPLICATION		
Synchronous	✓	
Asynchronous	✓	✓
Master/slave	✓	✓
N-way	✓	
SMP Support	✓	✓
Are roles supported	✓	
DATA TYPES		
Integer 16	✓	✓
Integer 32	✓	✓
Float	✓	✓
Double	✓	✓
Numeric	✓	
Decimal	✓	
Char	✓	Not ANSI standard
Varchar	✓	✓
Text blob	✓	✓
Blob	✓	✓
Date	✓	✓
Time	✓	✓
TimeStamp	✓	✓
Boolean	✓	

Feature	InterBase® 7.1	MySQL™ 4.1 alpha
NETWORK PROTOCOLS		
TCP/IP	✓	✓
Named pipes (NetBEUI)	✓	✓
IPX/SPX	✓	
Maximum database size	Unlimited	64,000 GB
RESOURCE REQUIREMENTS		
Disk space	40 MB with manuals and examples; 15 MB without	74 MB
Memory	32 MB	Not published

Conclusion

With InterBase you get:

- Triggers
- Stored procedures
- Client-side events
- 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
- N-way replication

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

About the author

Bill Todd is president of The Database Group, Inc., a database consulting and development firm based near Phoenix. He 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.