

MySQL Performance 4.1 vs 5.0



Peter Zaitsev,
MySQL AB

MySQL Users Conference 2006
Santa Clara, CA April 24-27

Introduction

- Software life cycle performance evolution
 - What do you think ?
- MySQL 5.0 vs 4.1 Performance
 - Using Well optimized MySQL 4.1 features
 - Implicit usage of MySQL 5.0 new features
 - Explicit usage of MySQL 5.0 new features

What is new version ?

- Bugs are fixed
 - Often meaning more checks are being added, optimizations restricted
- New features are added
 - Requiring more hooks
 - Increasing code base, data structure sizes, indirection
 - Algorithms changes required
- Local code optimizations
 - Same algorithms implemented better
- Global optimizations
 - New algorithms and data structures are implemented. More efficient

What is about MySQL 5.0 ?

- New Features:
 - Stored Procedures, Triggers, Views
 - Significantly increase parser complexity
 - More complex data structures, increased level of redirection
 - Many checks to handle these features in other code
- Local Code optimizations
 - Some. But it is hard to beat Monty
- Global optimizations
 - Greedy Join Optimizer, new execution method
- Performance bugs fixed
 - Optimizer
 - Mutexes

Beware of Optimizer

- What we call Optimizer ?
 - The piece of code which decides how query should be executed
 - Creates query «plan»
 - In MySQL practically merged with «executioner»
- Optimizer is extremely algorithmically complex
 - Developed by elite crew of PhDs
- Have to often make decision having limited information
- Best plan selected by optimizer might not be the fastest
- Optimizer changes from version to version always break some queries
 - But many other queries are executed much faster

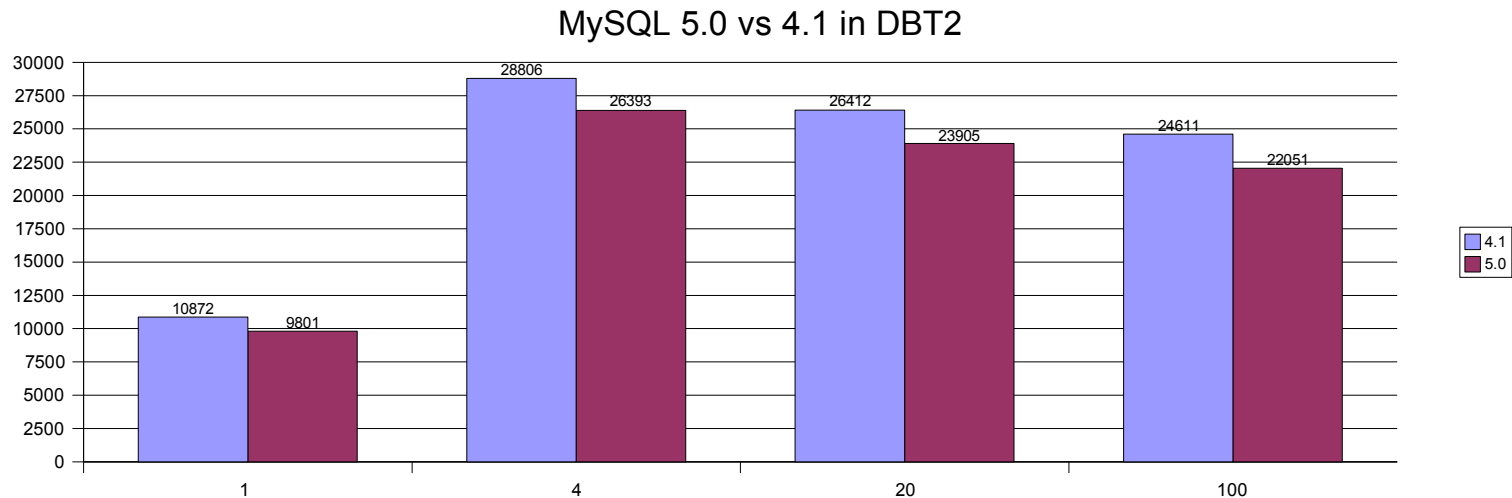
Well Optimized MySQL 4.1

MySQL 5.0 with 4.1 application

- If your application uses only features which were **well optimized** in MySQL 4.1 you might see performance degradation
- Does this mean I should stay with 4.1 ?
 - 5.0 has a lot of neat features for administration and management
 - Information schema, per thread status variables etc
 - You might have a chance to improve application by using 5.0 specific features
 - MySQL 4.1 will be getting only critical bug fixes
 - So you might not get fixes for bugs you find

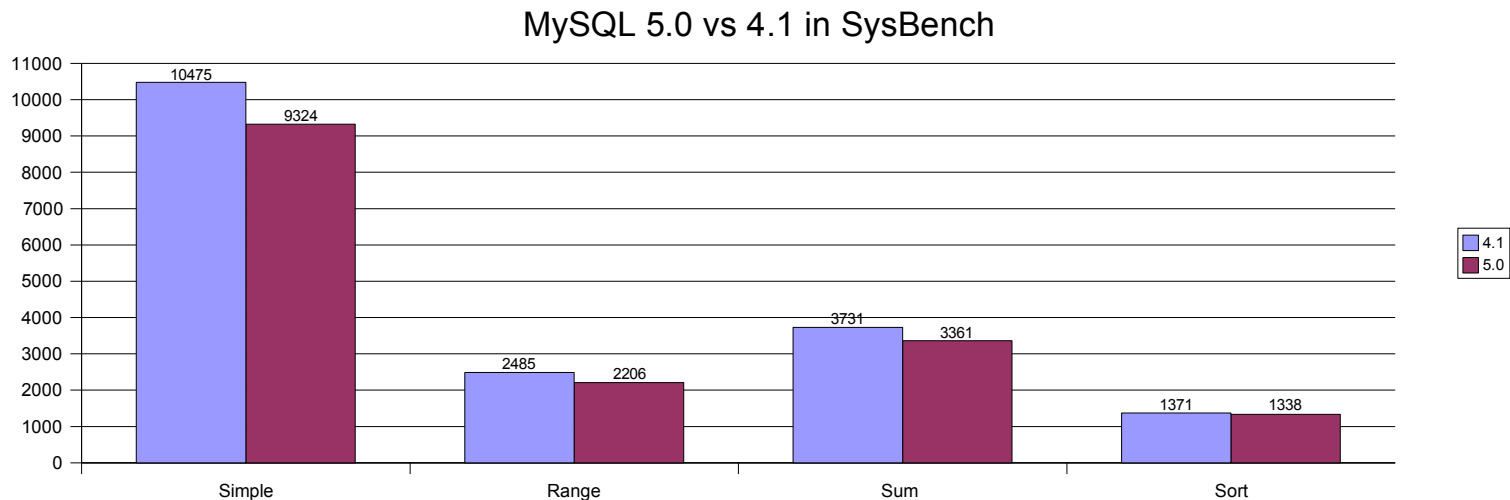
Benchmark Results: DBT2

- DBT2 – TPC-C Like benchmark developed by OSDL
- 4CPU Opteron, CPU Bound workload
- Results – Transactions per minute



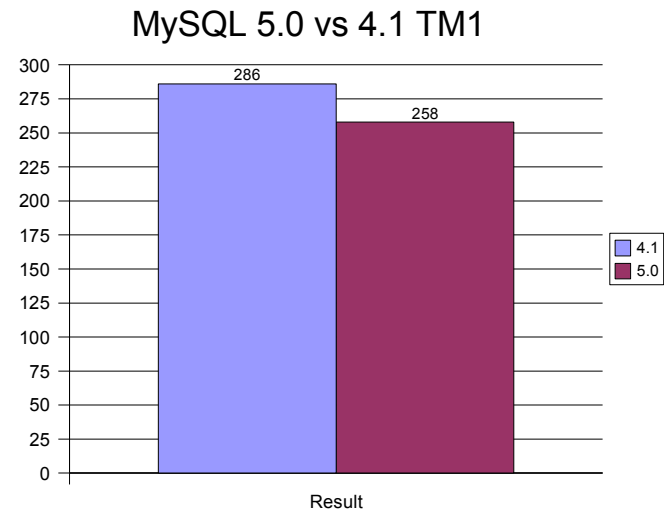
Benchmarks Results SysBench

- Sysbench – simple benchmark for MySQL
 - Available sysbench.sourceforge.net
- 4 CPU Xeon 2.0Ghz, RH AS 3.0, Single Thread



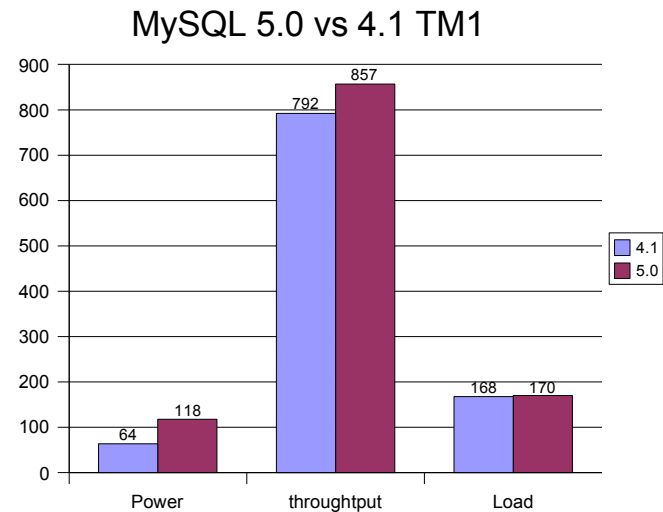
Benchmark Results: TM1

- TM1 – TeleCom One benchmark
 - Typical Home Location Register (HLR) scenario
 - Designed by Solid Database Developers
- Single CPU P4, Linux
- Database fits in memory
- MyISAM



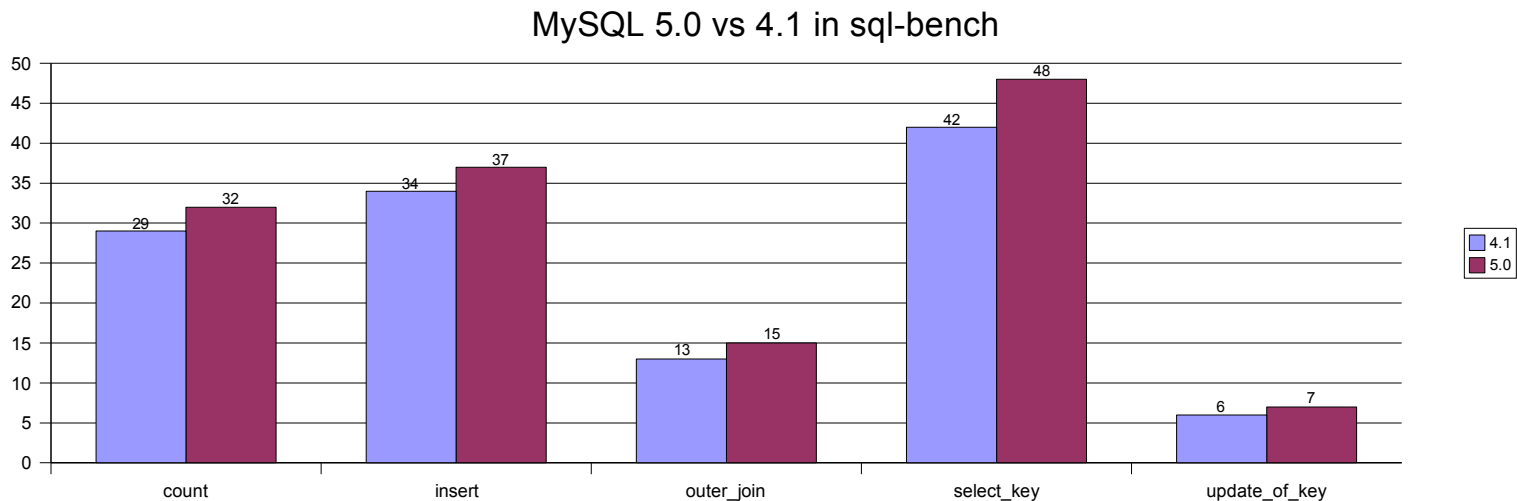
Benchmark Results DBT-3

- TPC-H Like benchmark developed by OSDL
 - Uses same schema and queries
- 0.1 scale size, sized to fit in memory
- Results are in **seconds**. Smaller better
- Difference is normally small
 - One query broken in 5.0



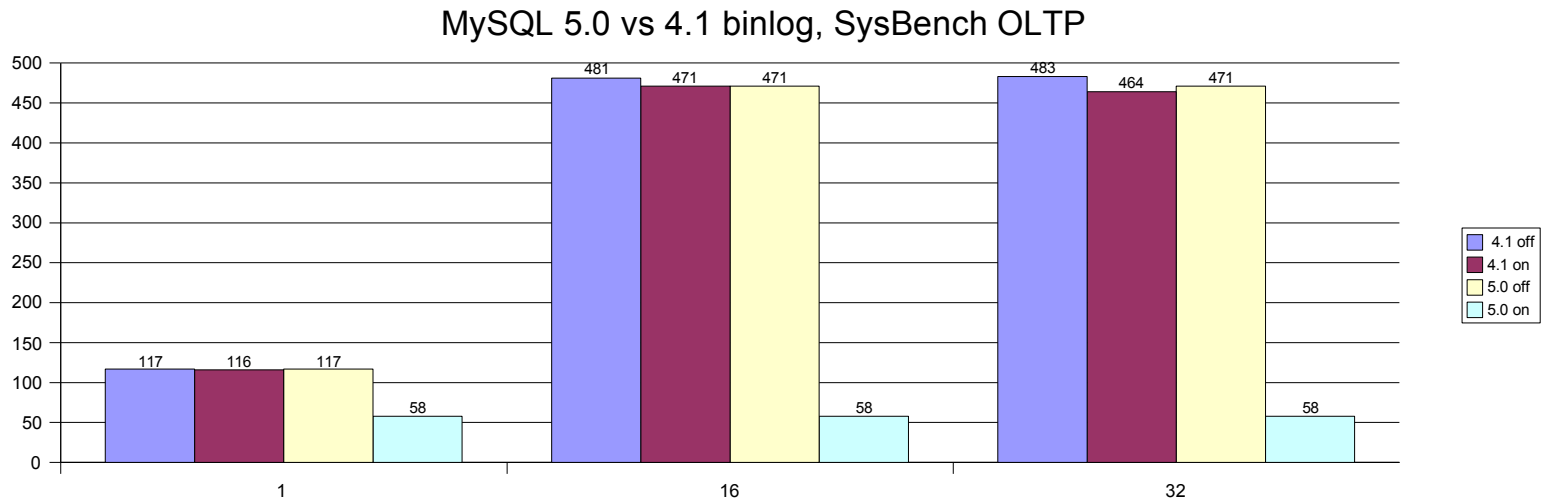
Benchmark Results: sql-bench

- Single user traditional MySQL benchmark
- Works with synthetic queries, small database size
- Mostly used for quality assurance and regression testing
- Results are in **Seconds**.



Benchmarks: Group Commit

- In MySQL 5.0 Group Commit does not work if binary log is enabled
 - Caused by introduction of XA (distributed transactions)
- Most visible if drive cache is flushable or disabled
 - Effect is mild with battery backed up cache on RAID



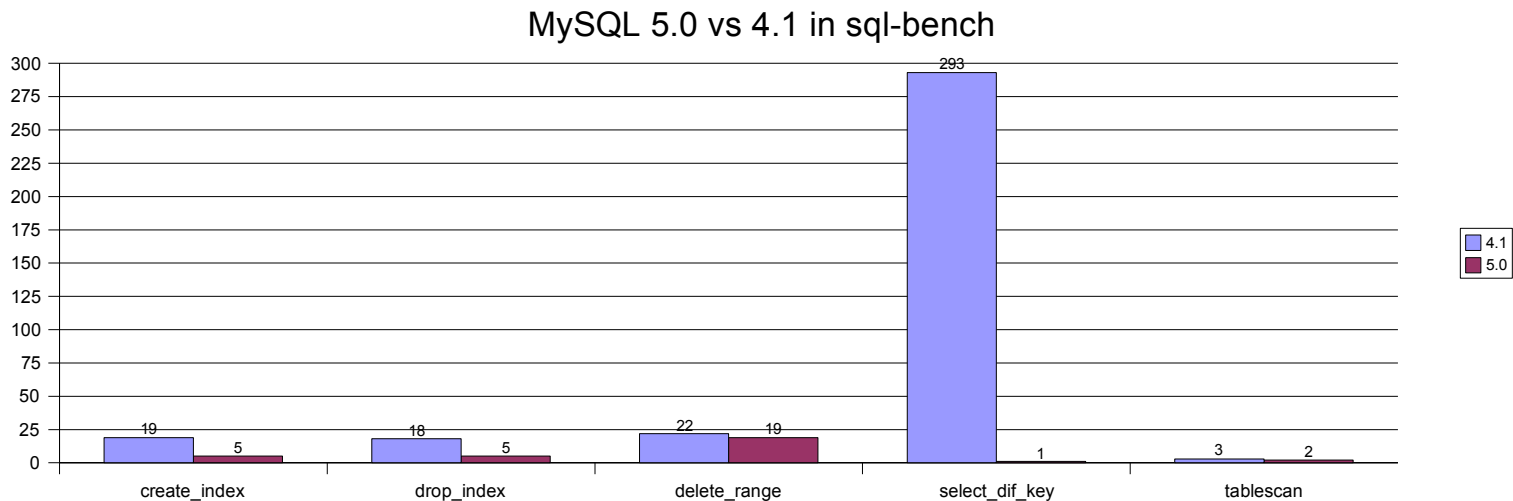
Implicit New Features

Implicit New Features

- Many new optimizations are implemented in MySQL 5.0
- Some just speed up things a bit while other may change performance order of magnitude
- Examples:
 - Greedy join optimizer
 - Equity propagation
 - Index merge data access
 - DISTINCT with aggregates optimization
 - COMPACT storage format for Innodb

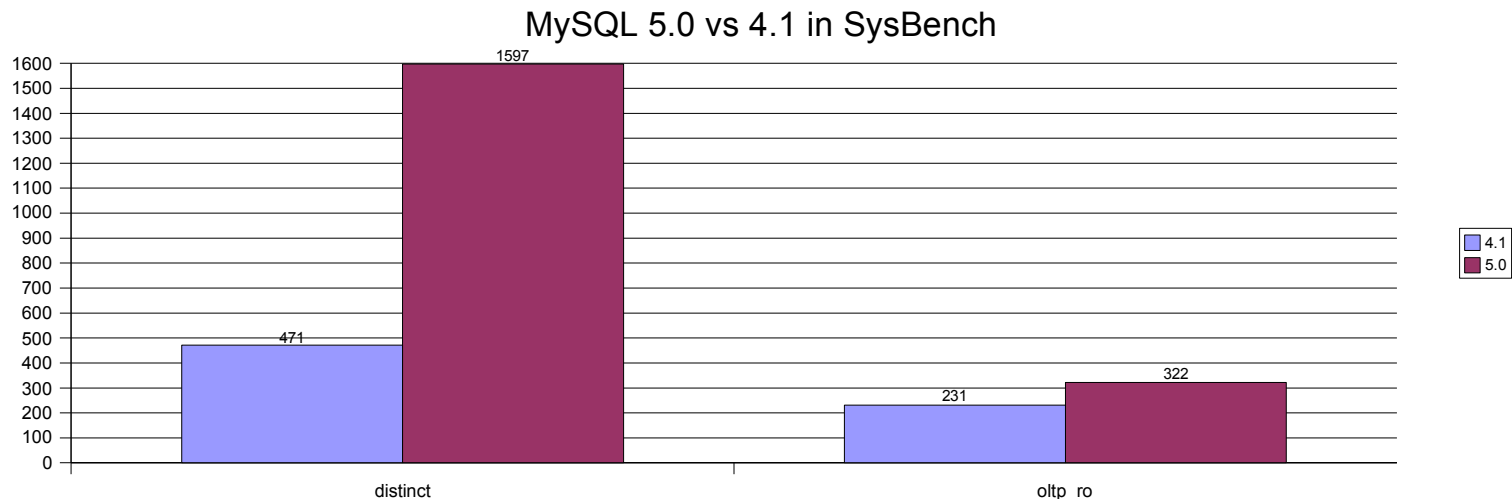
Benchmark Results: sql-bench

- Same sql-bench results, just different data points
- Results are in Seconds
- Results of serious improvements in MySQL 5.0



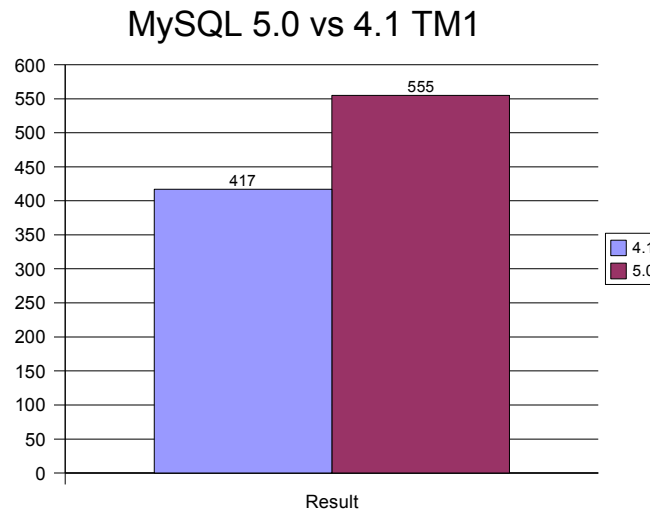
Benchmarks Result: SysBench

- Query: **SELECT DISTINCT c FROM sbtest WHERE id BETWEEN N and M ORDER BY c**
- OLTP RO – mix of the queries, including this one
- Results in Transactions/Second – larger means better



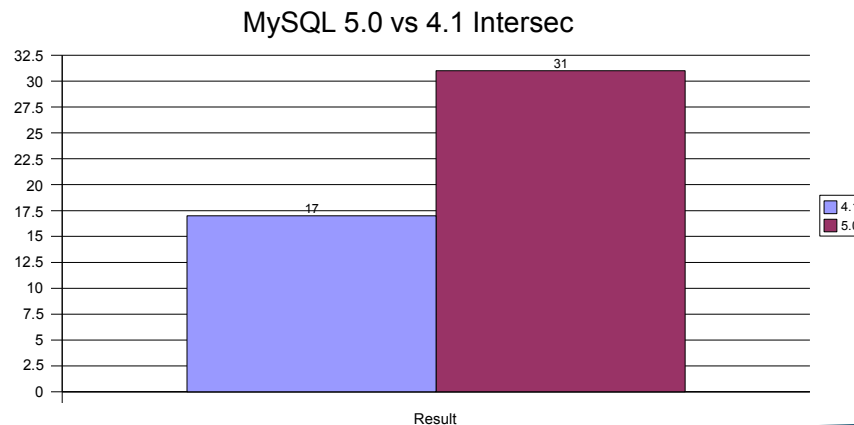
Benchmark Results: TM1

- Same TeleCom One Benchmark
- Innodb tables this time
- MySQL 5.0 is faster due to COMPACT storage
 - Database is close to memory size in this case



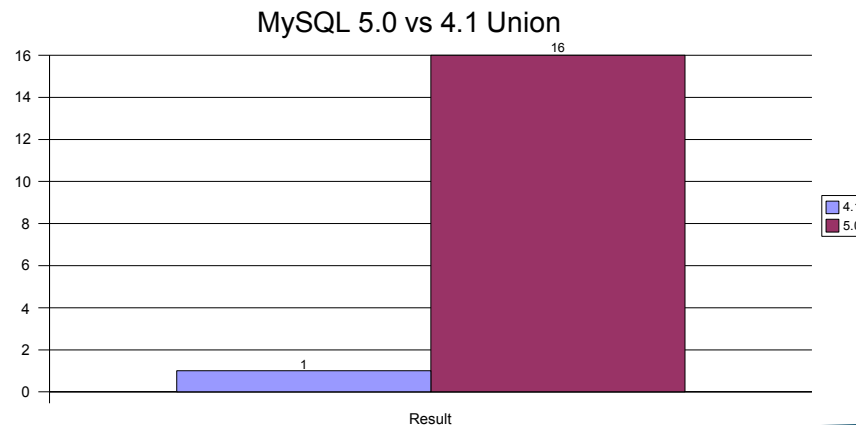
Index Merge Intersec: AND

- Helpful if you have index for each of conditions
 - But have no single index covering both of them
- `SELECT * FROM sbtest WHERE id > 10 AND k=206`
- MySQL 4.1: type: ref, key: k, rows: 2196
- MySQL 5.0: type: index_merge, key: k,PRIMARY
 - Rows: 296, Using intersect(k,PRIMARY);



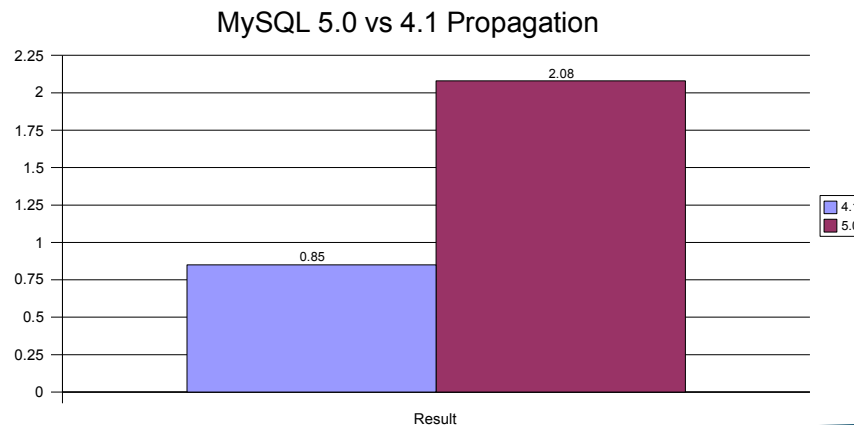
Index Merge Union: OR

- `SELECT c FROM sbtest WHERE id=2567 OR k=124`
- Was serious problem for MySQL 4.1
 - Resulted in full table scan
 - Workaround using UNION – needs temporary table



Equity Propagation

- **SELECT * FROM t1,t2 WHERE t1.id=t2.id AND t1.k=t2.k AND t2.k BETWEEN 498 AND 597**
- T1 – 1000 rows, T2 – 1.000.000 rows
- MySQL 4.1 can't discover T1.k also should be in the same range, so it puts table T2 first.
- Difference can be much more significant



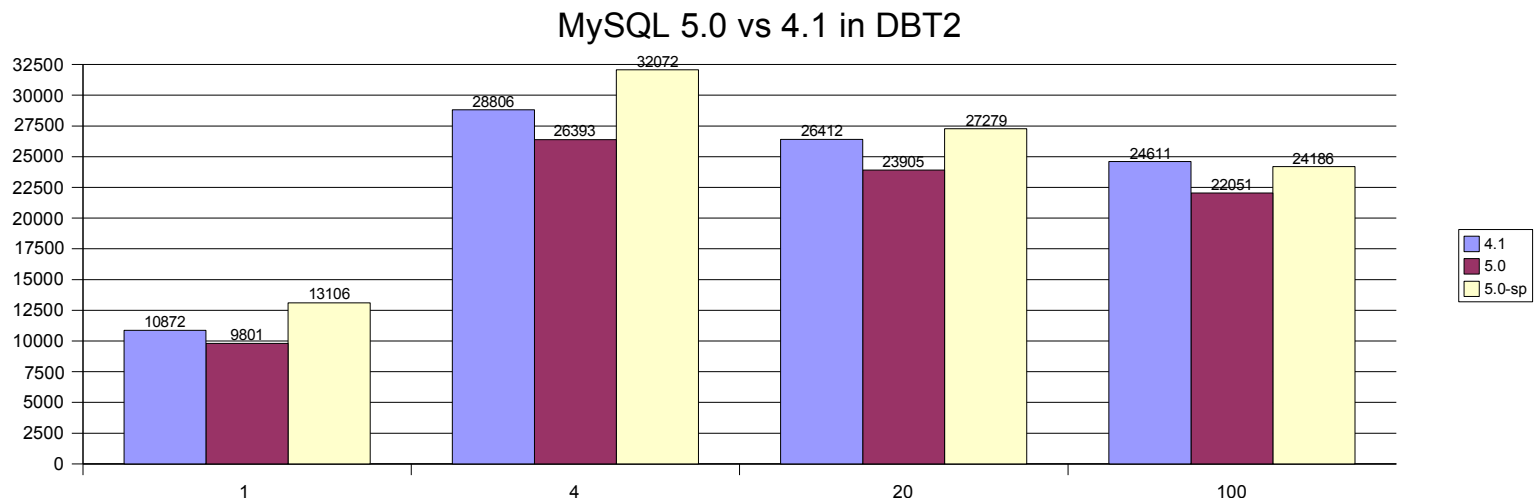
Explicit Optimizations

Explicit Optimizations

- Use MySQL 5.0 specific features to optimize performance
 - Stored Procedures
 - Triggers
 - Views

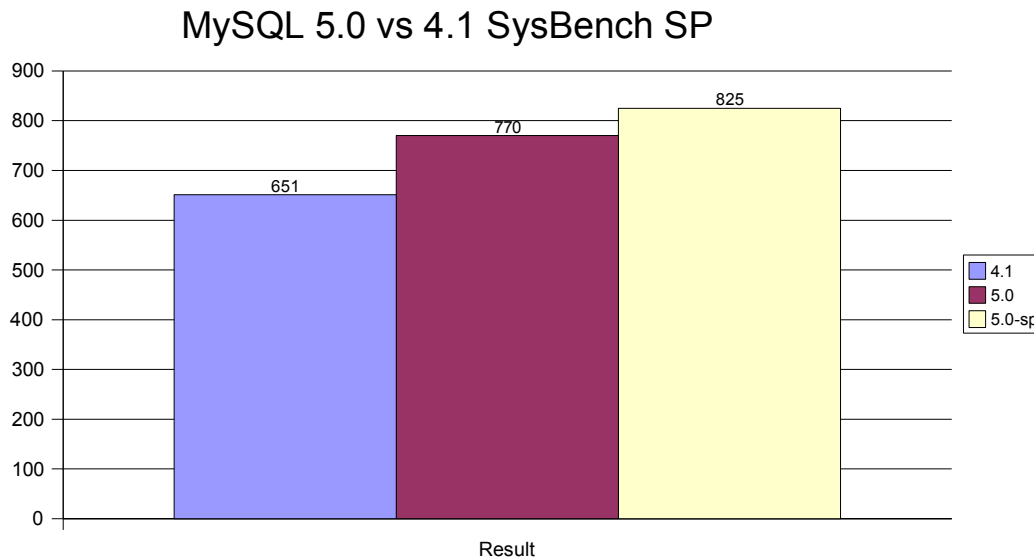
Benchmark: Stored Procedures

- DBT2 – Same DBT2 benchmark implemented with SP
- Stored Procedures deliver savings in
 - Network communications
 - Parsing, various internal works



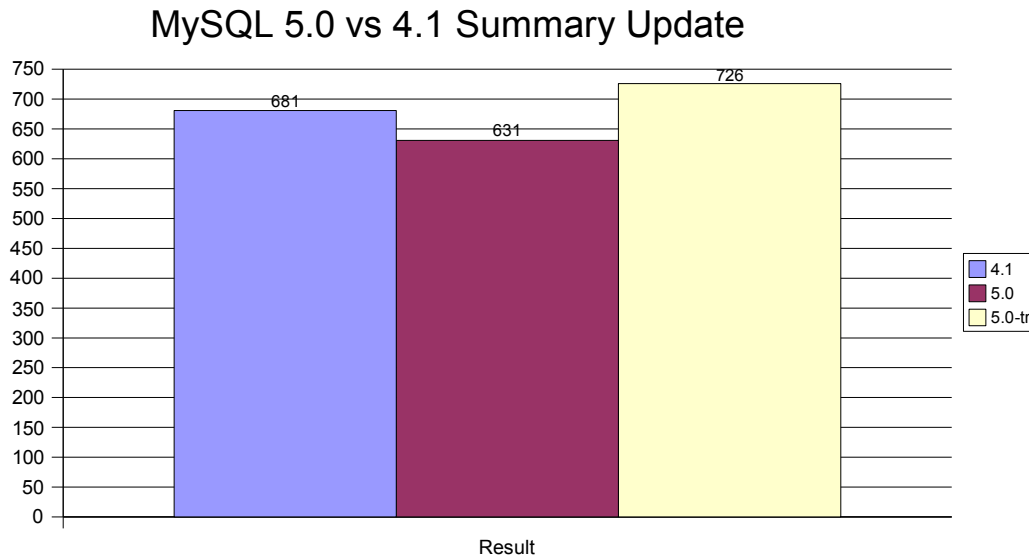
Benchmark: Stored Procedures

- Sysbench OLTP read only implemented in Stored procedures
- 4 concurrent connections
- 5.0 gains increases by using stored procedures



Benchmark: Triggers

- Triggers are great to optimize
 - Updating summary tables
 - Audit logs
 - Other actions triggered by data modification



Views

- Are not directly used to get performance improvement
 - Query against view is not faster than against base tables
 - MySQL does not have materialized views yet
- Views can be used for performance optimization
 - Changing data structure without changing the queries

Summary

- MySQL 5.0 may be faster or slower than 4.1
 - Depending on your workload
- Good to perform benchmark before upgrading
 - To avoid surprises
- Investigate if some of new MySQL 5.0 optimization apply to your application
 - Or if you can change it to benefit from it

Thank You For Comming

- I'm happy to answer your questions while on the conference
- Send questions by email - peter@mysql.com
- Special Thanks to High Performance Group helping to prepare this presentation
 - Vadim Tkachenko
 - Alexey Kopytov
 - Alexey Kishkin