



# Connector/J Performance Gems

Mark Matthews - Sun Microsystems

Innovation Everywhere

# This Talk Is About Performance

- It's not about scalability (for the most part)
- Can't have scalability without performance
- The converse is not necessarily true



# Easy Wins

- Mark's #1 Goal with Connector/J - Compliant out of the box - no unexpected behaviors
- Stay within the "bounds" of the JDBC API
- Are usually a configuration change



# Cache Server Configuration

- “cacheServerConfiguration=true”
- Requires restart of application if you change these MySQL variables: *language*, *wait\_timeout*, *interactive\_timeout*, *net\_write\_timeout*, *character\_set\_\**, *\*timezone*, *query\_cache\**
- Causes *some* contention when creating new connections

# Using Local Connection State

- “useLocalSessionState=true”
- Transaction isolation, auto commit and catalog (database)
- “useLocalTransactionState=true” (5.1.7)
- Saves commit/rollback
- Doesn't work with Query Cache < 6.0



# Rewriting Batches

- “rewriteBatchedStatements=true”
- Affects (Prepared)Statement.add/executeBatch()
- Core concept - remove latency
- Special treatment for prepared INSERT statements



# A Rewritten Plain Statement

- `INSERT INTO foo VALUES (...); INSERT INTO FOO VALUES (...); ...`
- Works even if multi-statements are not enabled by default
- Turns on multi-statements for the batch, when batch count > 3

# A Rewritten INSERT

- INSERT INTO foo VALUES (...),(...),(...) up to max\_allowed\_packet
- Starting in 5.1.8, ON DUPLICATE KEY UPDATE rewritten into multi-value as well!
- except for case where you use LAST\_INSERT\_ID()



# Treat Bulk UPDATE as INSERT?

- Use MySQL's "INSERT ... ON DUPLICATE KEY UPDATE" functionality
- Re-written into a "multi insert" in C/J 5.1.8
- (as long as you don't use auto-incs)
- Hibernate - see @SQLUpdate or `<sql-update check="none" />`

# Using Server-Side Prepared

- “useServerPrepStmts=true”
- Less parsing - “native” on-wire format
- Binding and Execution have compact on-wire format
- Sketchy on early versions of MySQL-5.0
- Type conversions can be more costly
- Less memory pressure for result sets with numeric data

# Caching Prepared Statements

- “cachePrepStmts=true”
- “prepStmtCacheSize=..” and “prepStmtCacheSqlLimit=...”
- Saves parsing cost (even more in version 5.1.8)
- Reduces memory footprint for non-server-side statements
- Reduces latency for server-side statements prepareStatement() phase

25 statements, 256  
bytes statement limits

# Maintaining Timed Statistics

- “maintainTimeStats=false”
- Some platforms have expensive getTimeOfDay()
- We have “friendly” error messages that include elapsed times
- Trade ease of use for performance?
- 3% (or so) throughput increase

# Try different I/O Strategies

- “useUnbufferedIO=false” and “useReadAheadInput=false”
- Can reduce system calls 3-5x
- More useful when MySQLd can't fill the pipe fast enough





# Can't Remember All of This Stuff?



- Shortcut!
- “useConfigs=...”
- Bundled sets of configuration options
- maxPerformance, solarisMaxPerformance
- fullDebug
- 3-0-Compat, 5-0-Compat



# Living On The Edge

- Usually Safe
- Caveats
- Not always JDBC Compliant



# Caching ResultSet Metadata

- You *must* know your application
- Only works when “shape” of result sets is always the same
- Saves parsing and memory allocation
- 3-5% performance gain generally
- More for result sets with large column counts



# Use LOCAL INFILE for Batch Loads

- SQL: “LOAD DATA LOCAL INFILE ...”
- Google the above text for the manual page
- It doesn't need to be a file!
- Custom Input Stream with Connector/J:  
`com.mysql.jdbc.Statement.setLocalInfileInputStream(  
InputStream)`
- Use a URL  
LOAD DATA LOCAL INFILE ‘http://....’  
“allowUrlInLocalInfile=true”

# Take Aways...

- Performance improvement can lead to scalability improvements
- Not enabled out-of-box because of Mark's #1 Rule
- There are some *easy* things you can do
- We even help you cheat! (e.g. "useConfigs=...")
- If you really know your application there's more you can try *safely*



# Resources

- This presentation - posted to the Conference Website
- Java BoF
- MySQL Enterprise Monitor BoF
- [markm@sun.com](mailto:markm@sun.com) (or [mark@mysql.com](mailto:mark@mysql.com))
- <http://forums.mysql.com/>



**Questions?**



**Thanks!**