Some studies in Firebird Performance

Paul Reeves
IBPhoenix

mail: preeves at ibphoenix.com

IBPhoenix
THE POWER WITHIN

About the speaker

I work for IBPhoenix providing technical support.

I maintain the windows installer for Firebird and do the Windows builds.

IBPhoenix
THE POWER WITHIN

Introduction

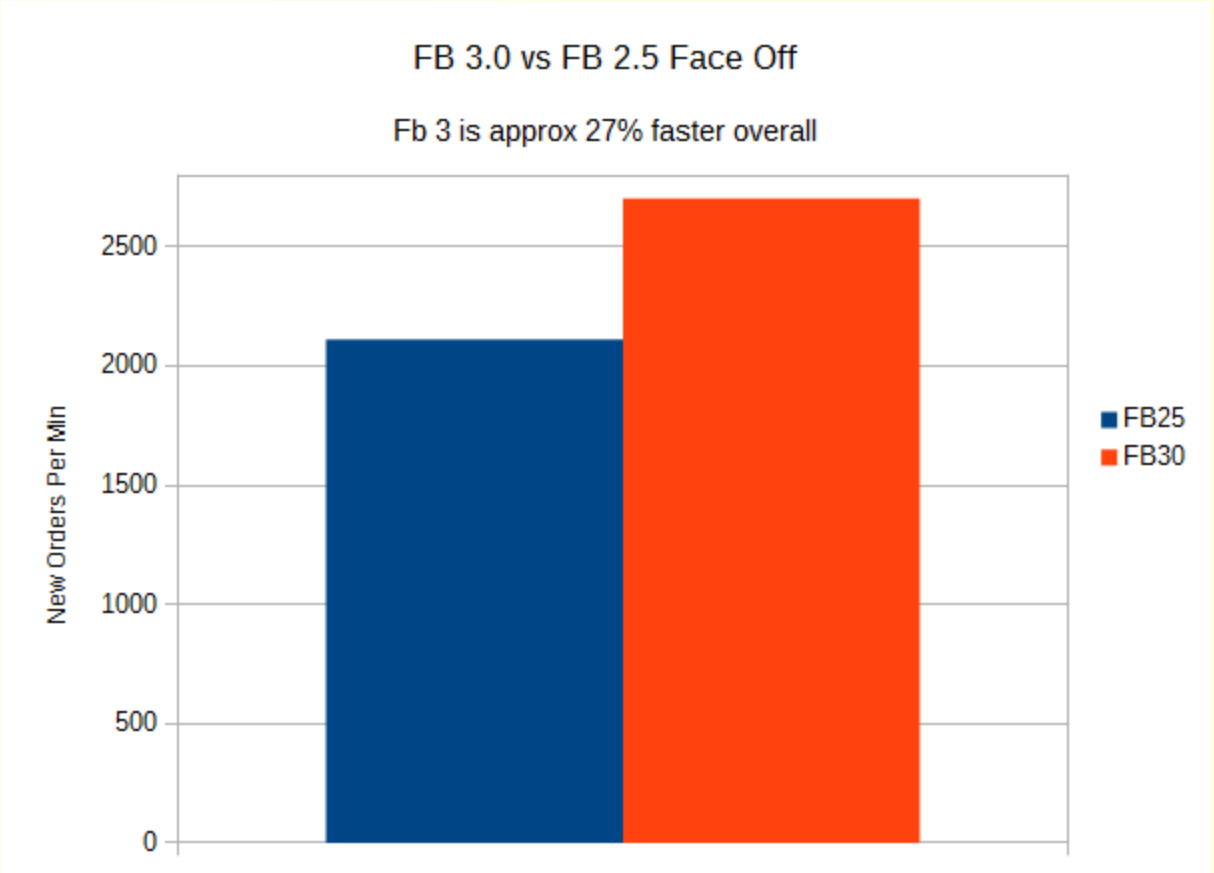
- Using benchmarking to study performance in a multi-user environment.
- Does the improved multi-threading of Firebird 3 bring benefits ?
- How do the different architectures compare in Firebird 3?
- Does database encryption sacrifice speed for security ?
- Building a test harness

About the test harness

- Modest hardware.
- Largely isolated from outside interference
- It provides multi-user activity to exercise fb locking, cpu synchronisation, and memory and disc access.
- It isn't designed to test internal features of Firebird.
- The main aim is to provide a stable platform. By changing one parameter at a time we can build a database of performance statistics to understand how different configurations impact on performance.

So, is FB3 faster than FB2.5 ?

Is FB 3.0 faster than Fb 2.5?



We seem to have a clear winner

IBPhoenix
THE POWER WITHIN

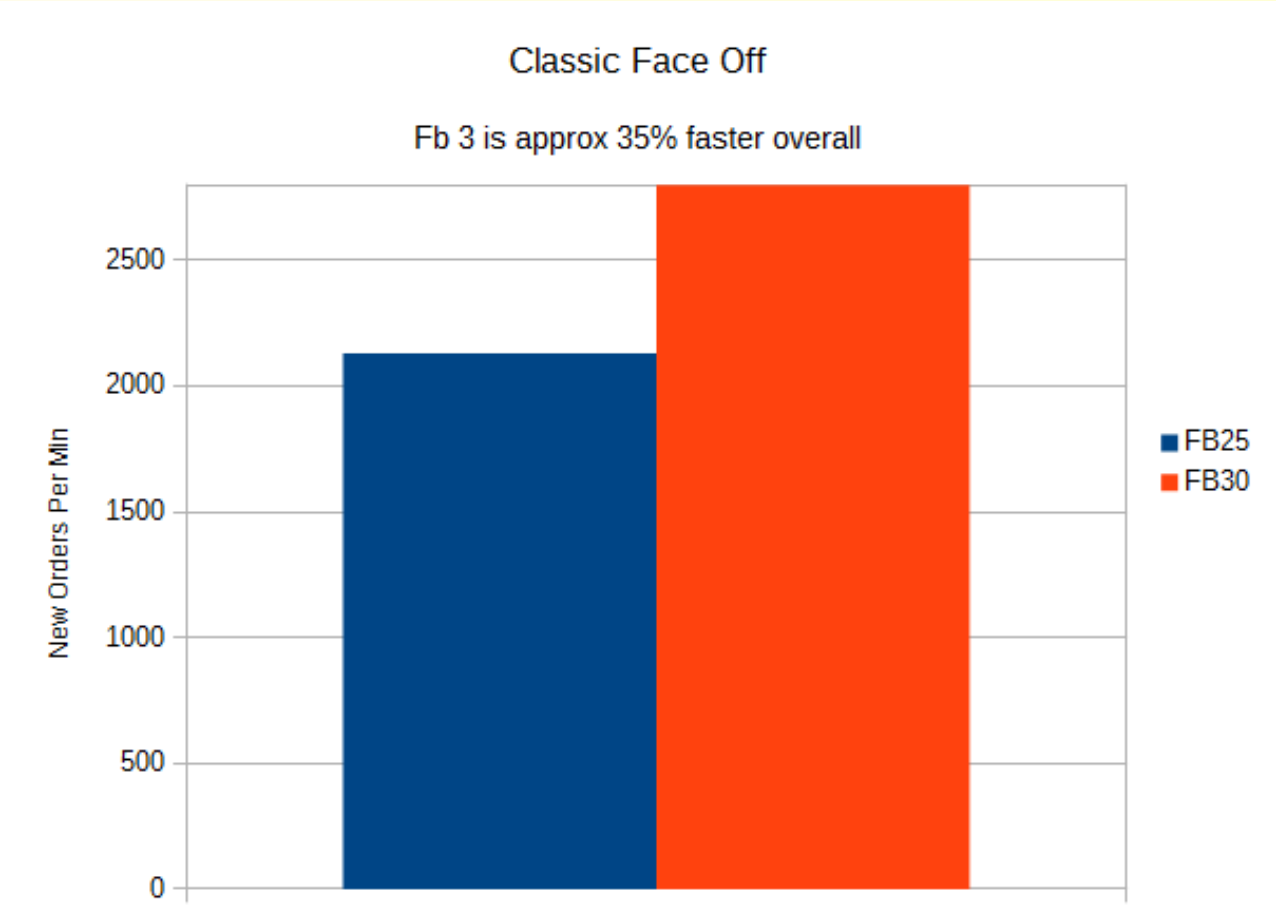
Firebird Configuration

- Each arch has had buffers optimised
  - SS uses 16K or 32K buffers
  - SC uses 2K buffers
  - CS uses 1K buffers
- Hash Slots for FB 2.5 were increased to 8191
- Page sizes of 8KB and 16KB were tested.
- Connection pools consisted of 5 to 15 users

Other Config Notes

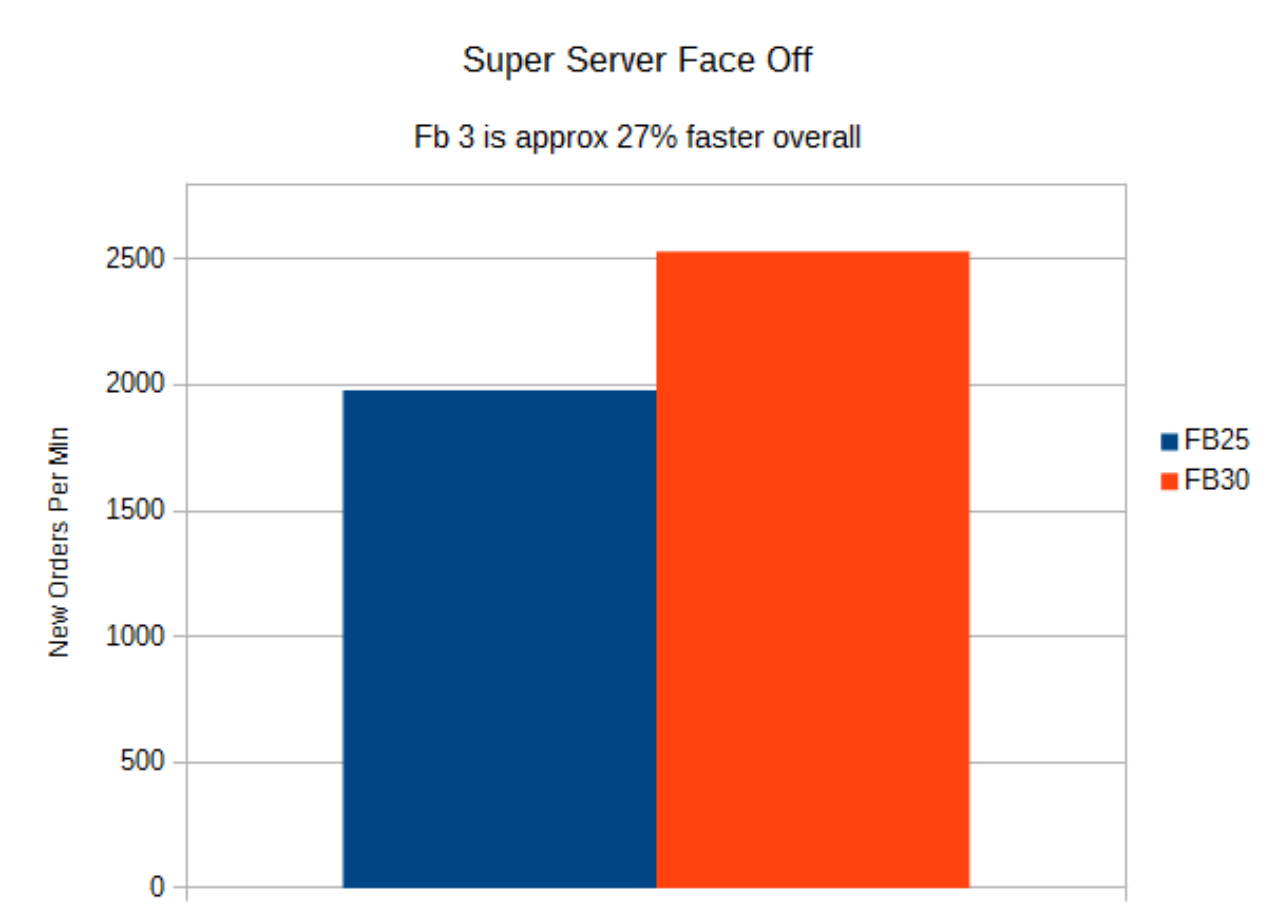
- Test Harness configured for durability, not performance
- No attempts made to overload the system
- All tests are multi-user, but connections de-liberately kept low.
- Defaults are used, except where specified.

Let's look at each architecture



IBPhoenix
THE POWER WITHIN

And for Super Server



IBPhoenix
THE POWER WITHIN

SuperClassic

IBPhoenix
THE POWER WITHIN

About the tests

- The tests are built around tpc-c which has some limitations
- Can the benchmark be trusted ?
- Can any benchmark be trusted ?
- Important to understand limitations of a benchmark implementation
- Building a test harness and benchmark is quite expensive in development time and time required for subsequent analysis.

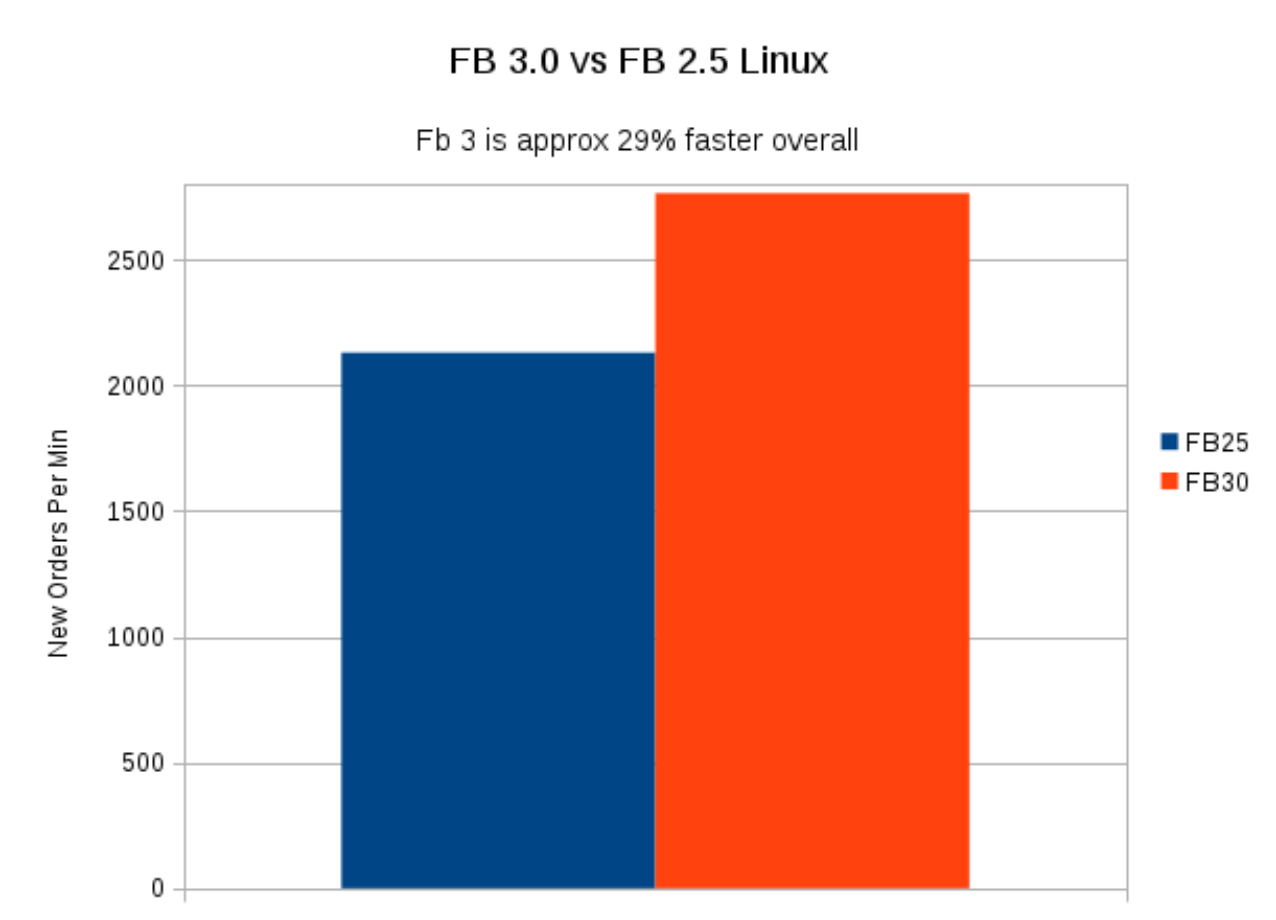
The limitations of TPC-C

- Too simple.
  Most real world applications are a lot more complex.
- Small records, no blobs, SPs, triggers
- Bad design (like a lot of databases :-))
- Locking anomalies which distort consistency of results.
- Not representative of a typical firebird database (But what is?)

But it is not all bad….

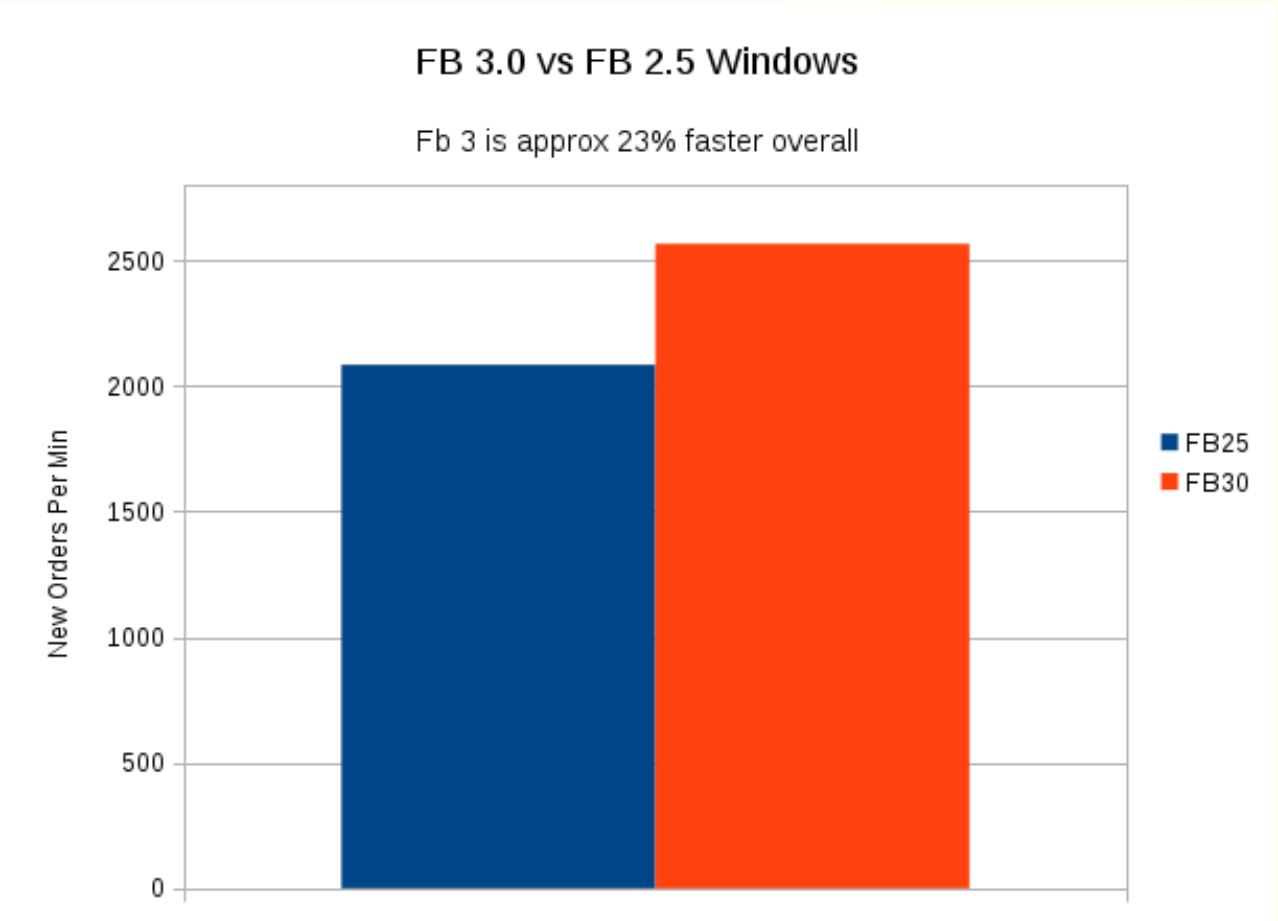
- Tests, although imperfect, do place stress on Firebird and the underlying hardware.
- A benchmark app is only a small part of the test harness.
- More importantly, multiple tests can be run and the results stored for analysis
- Hundreds of tests are executed. Thereby limiting the errors caused by anomalous test runs.
- Analysis of trends in the data should be valid. (I hope.)

Performance under Linux



IBPhoenix
THE POWER WITHIN

And what about windows?



- Hmmm…. Slightly worse than Linux
- But maybe insufficient data?

IBPhoenix
THE POWER WITHIN

OK, we get the picture.

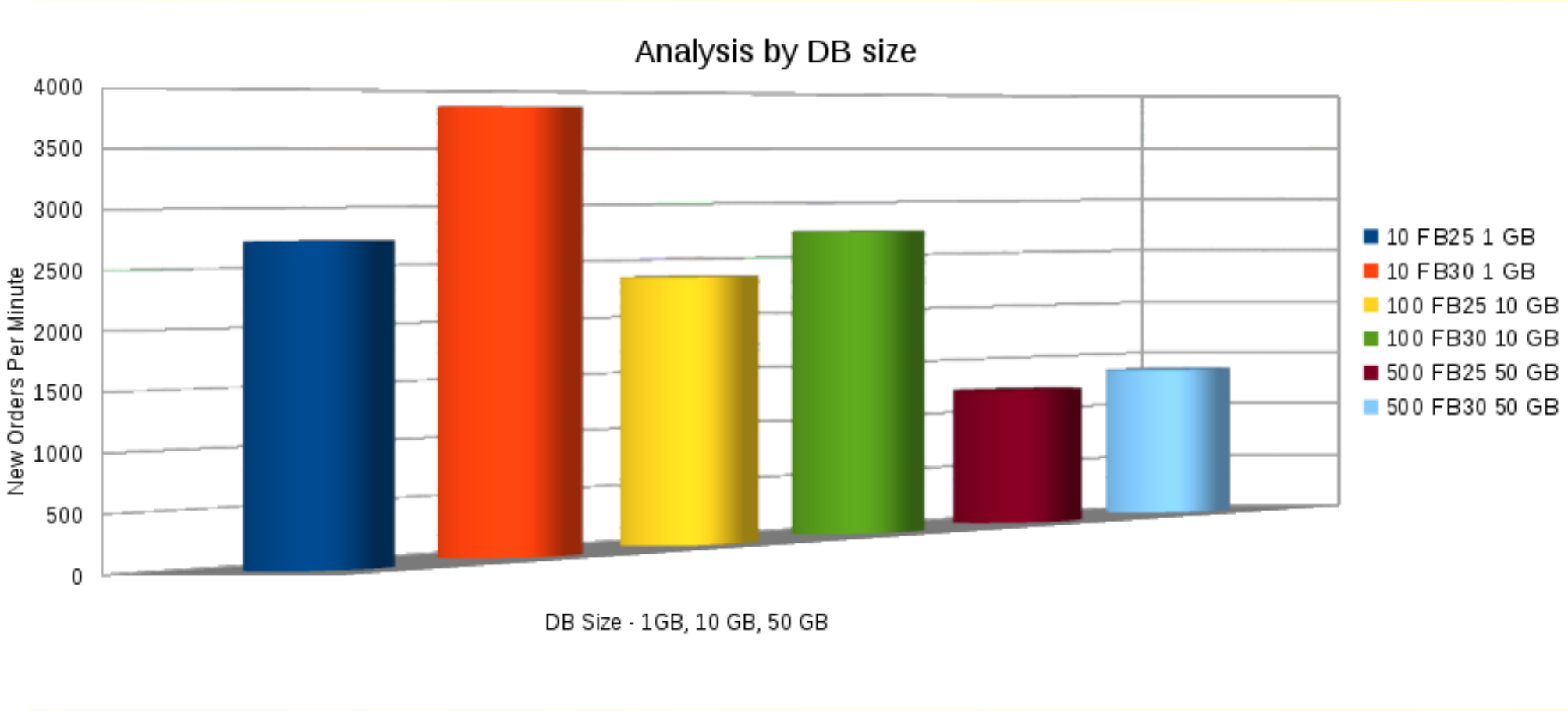
- FB 3 is faster than FB 2.5
- Before you all die of boredom, let's dig deeper

3 different database sizes have been tested

- Small DB – fits easily into available ram. Takes full advantage of database and file system cache.
- Medium DB – too big to fit into memory, but lots of data is served from cache.
- Large DB – lots of cache misses.

- These sizes are relative. For the tests:
  Small DB - ~ 1 GB
  Medium DB - ~10 GB
  Large DB - ~50 GB

Breakdown By Database Size

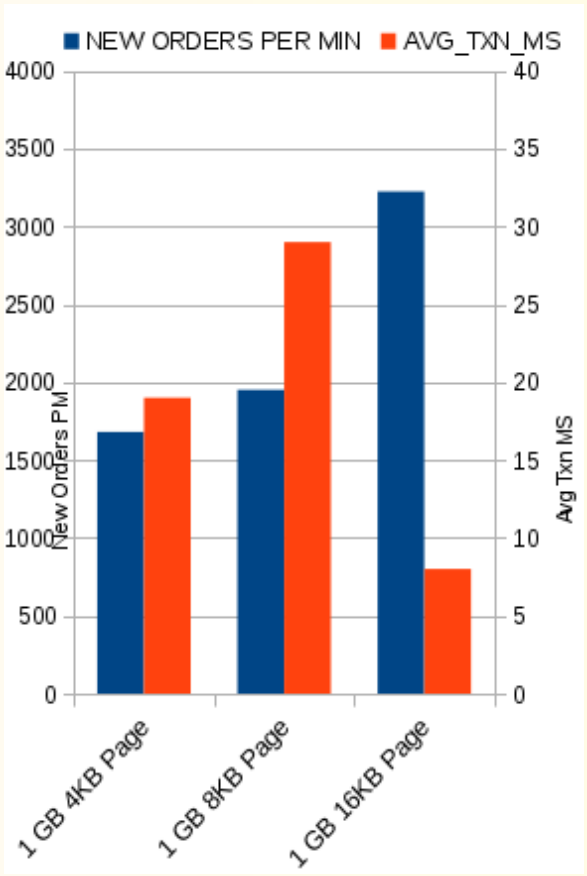


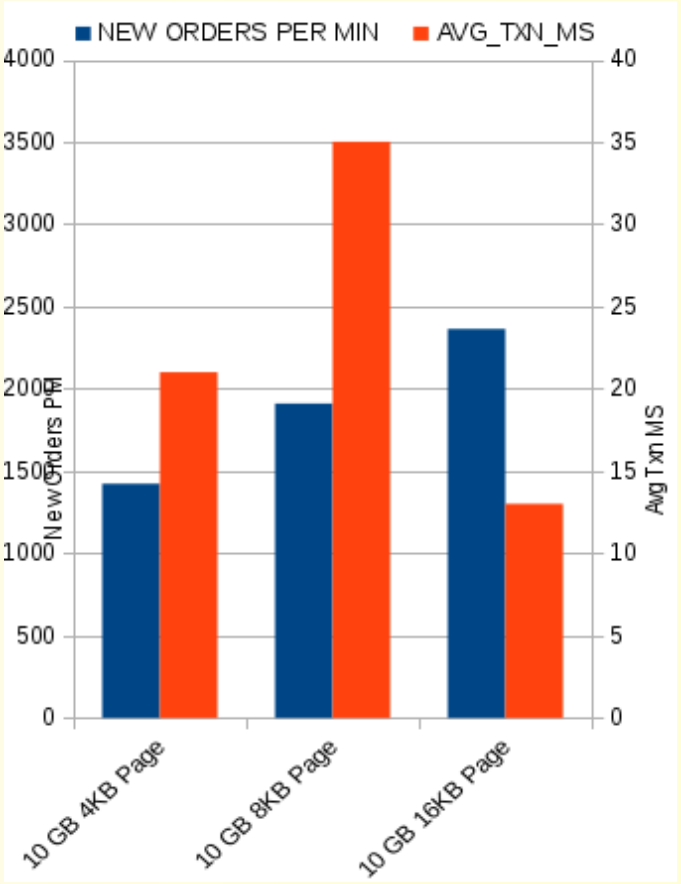
IBPhoenix
THE POWER WITHIN

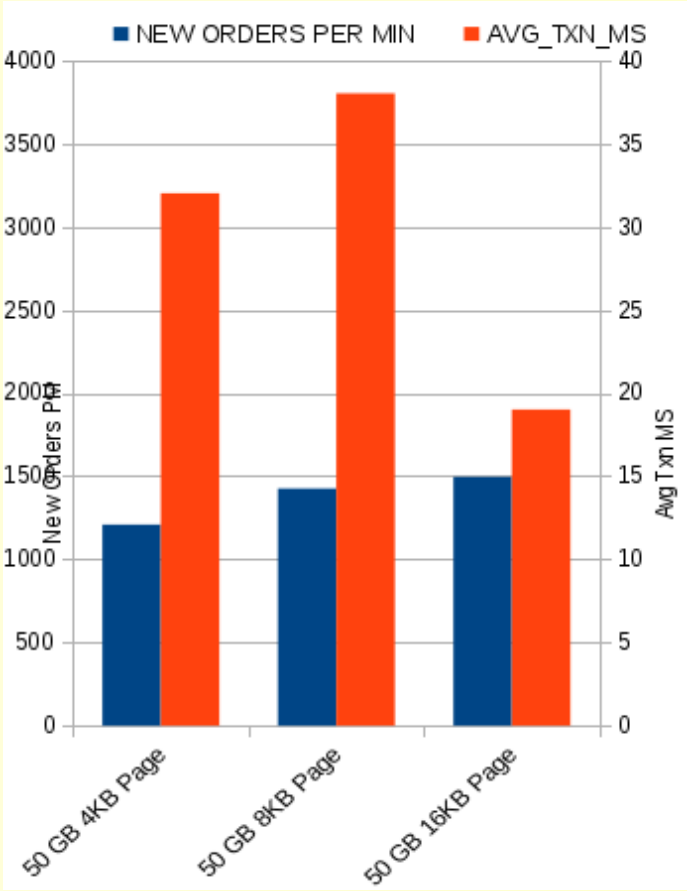
Performance drops heavily as data-base size increases

- What can be done to alleviate this?

What is the influence of Page Size on Database Size?



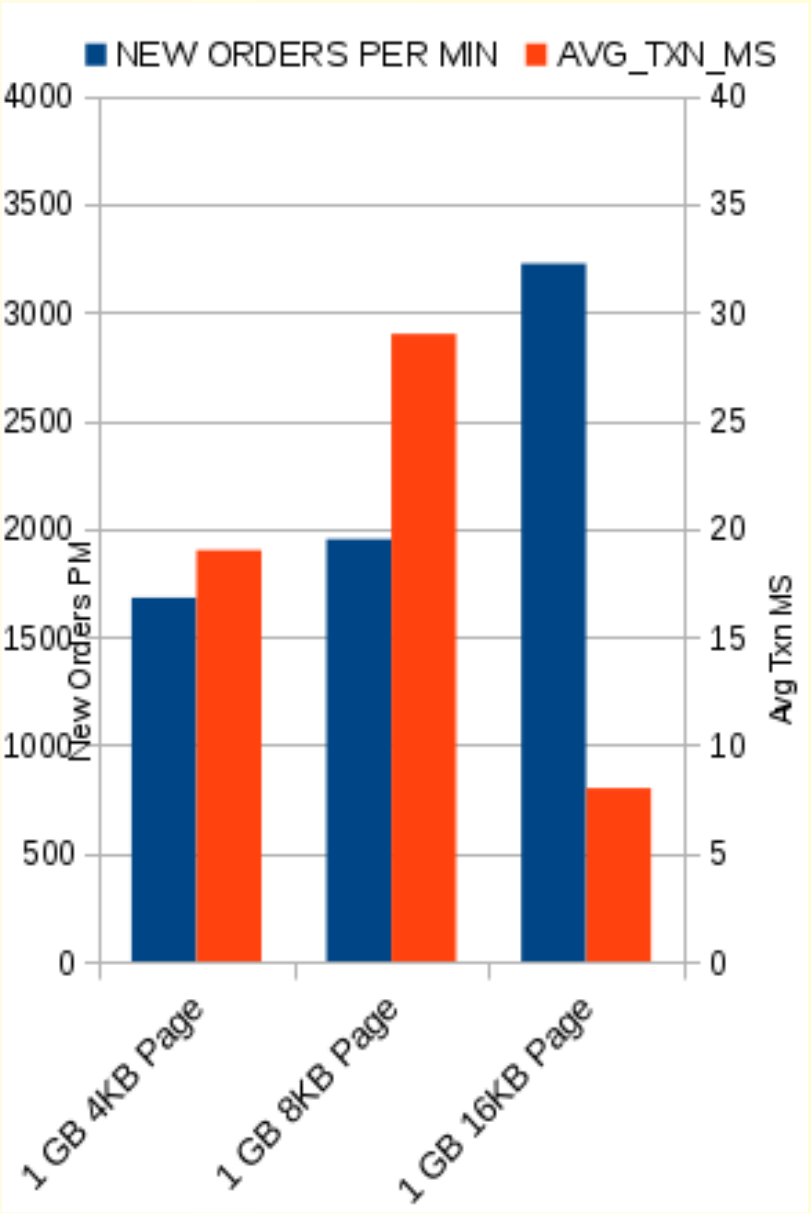




- Note: Data derived from a meta analysis from FB2.5 tests

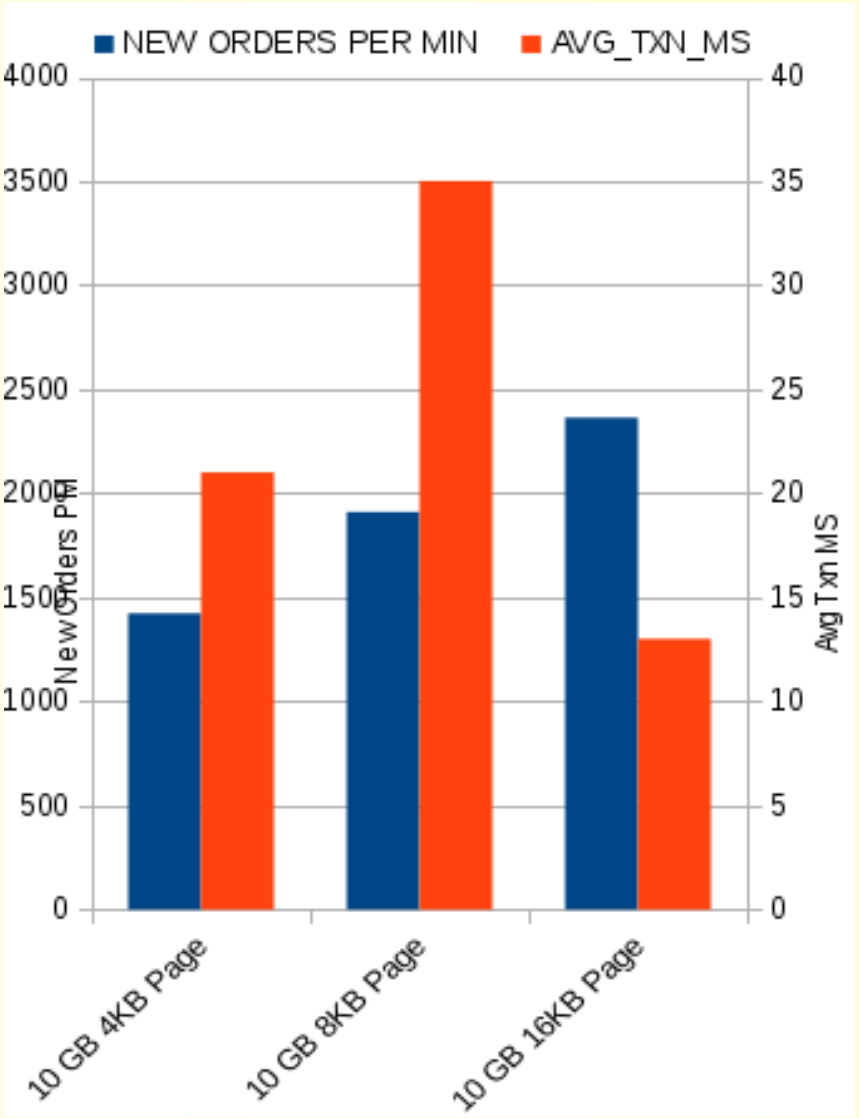
IBPhoenix
THE POWER WITHIN

Page Size and Small databases



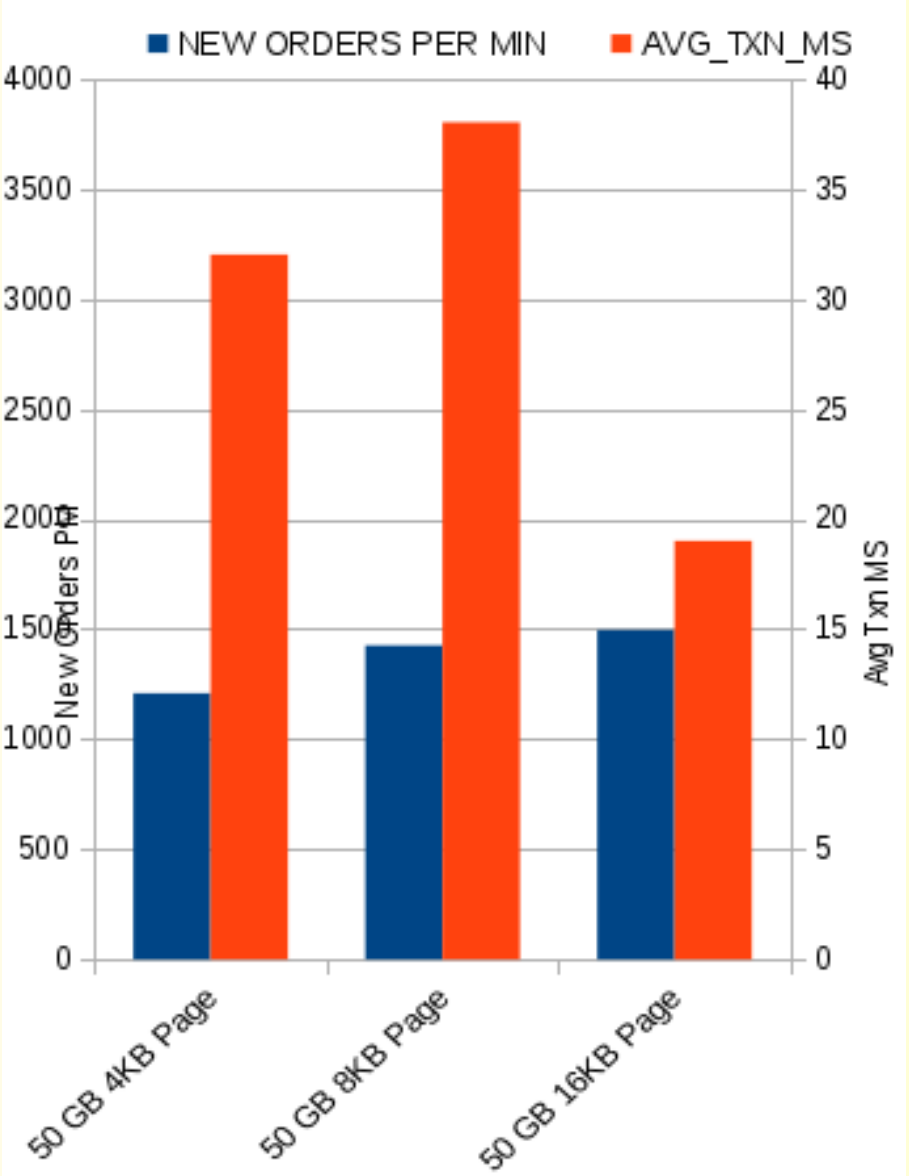
IBPhoenix
THE POWER WITHIN

Page Size and Medium databases



IBPhoenix
THE POWER WITHIN

Page Size and Large databases



IBPhoenix
THE POWER WITHIN

Page Size - summary

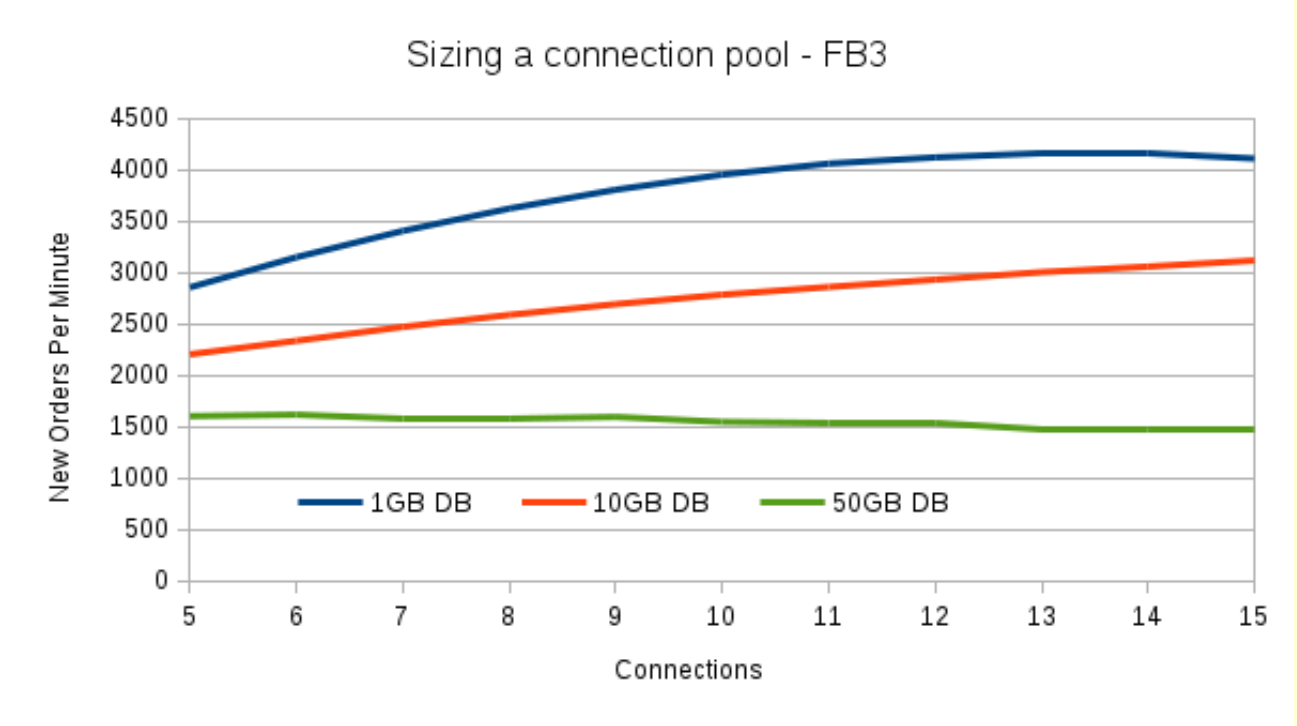
- Page size can make a big difference
- avg txn times don’t match changes in page size

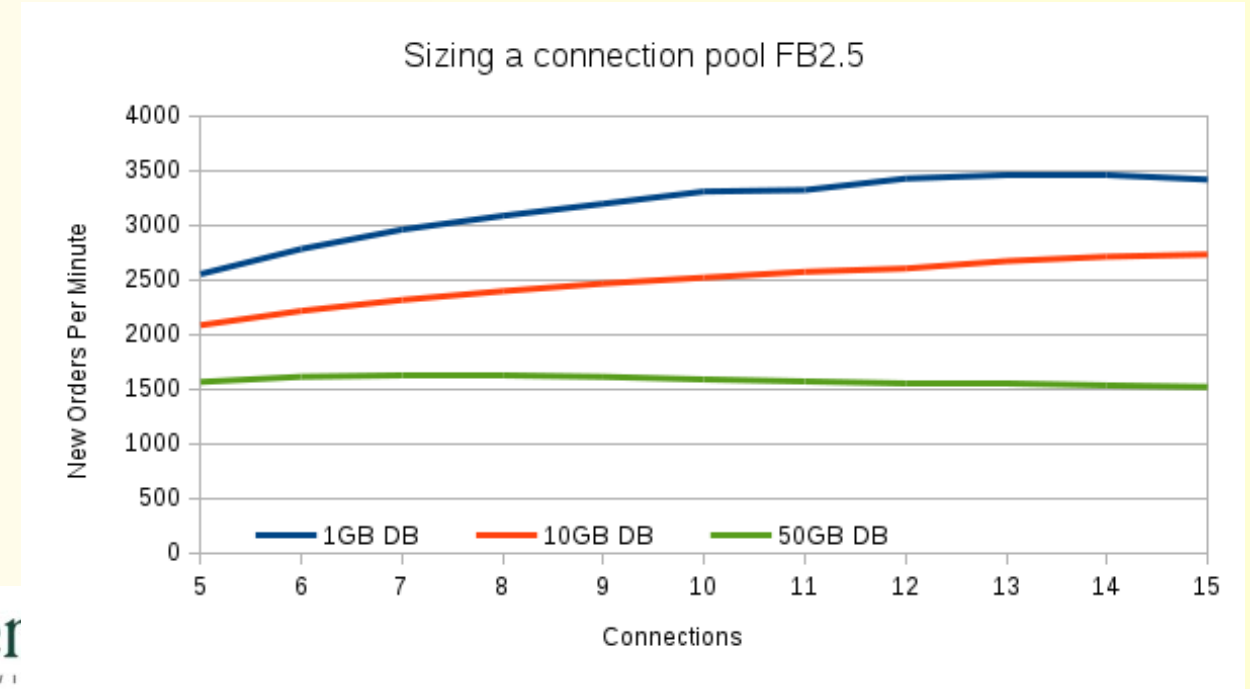
Using Benchmarking as a guide to server provisioning

Connections and user equivalents

- Connections in a multi-user benchmark are not equivalent to users.
- It is difficult to gauge equivalent number of con-current users
- It is more accurate to think of a connection pool
- The pool must be sized for the hardware
- Overloading the system kills performance
- But SuperServer seems to support heavier loads in an overloaded system
- My current test harness can only support a pool of 15 to 20 users

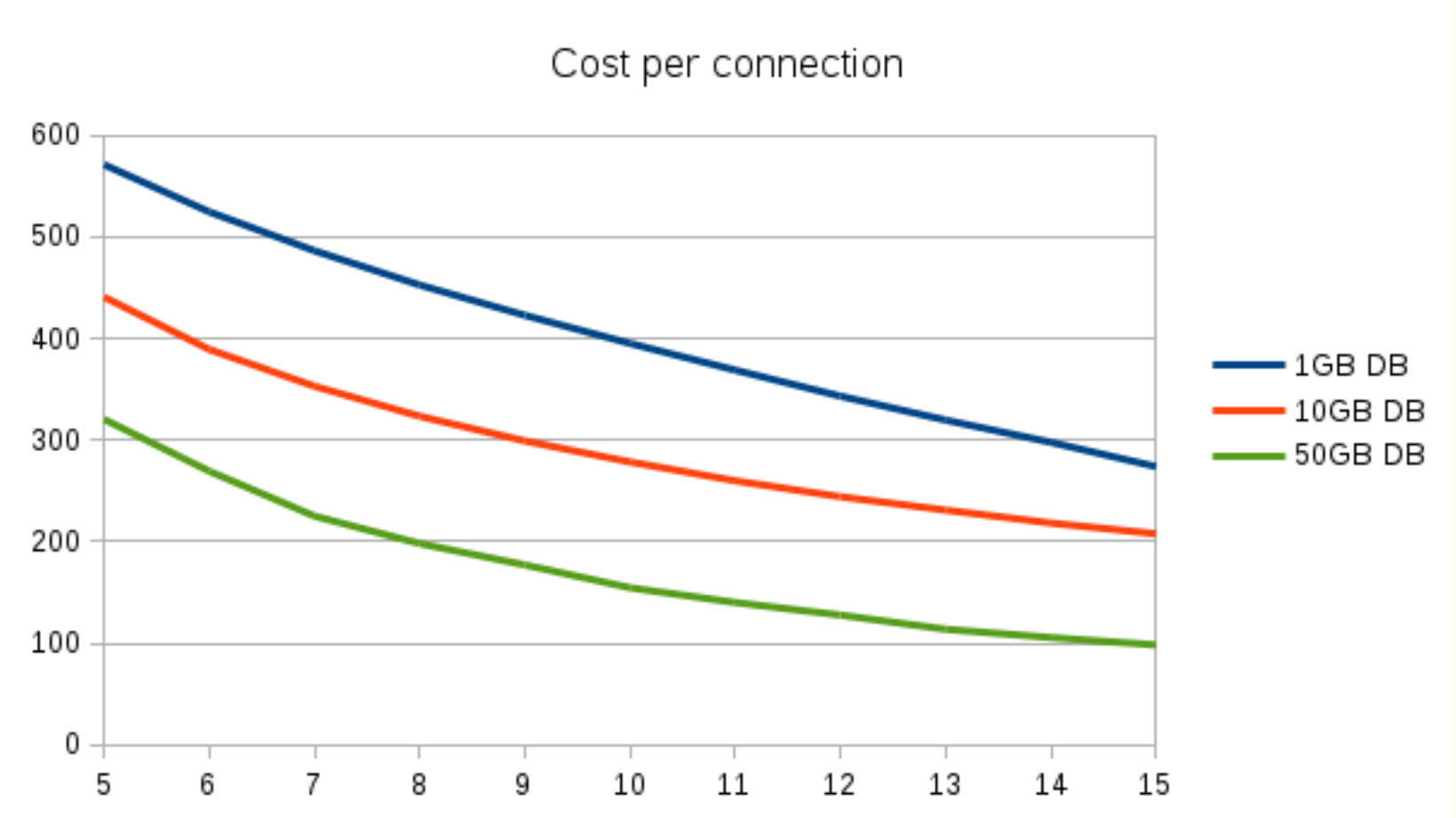
Performance impact of different pool sizes





IBPhoer
THE POWER WI

Cost per connection



IBPhoenix
THE POWER WITHIN

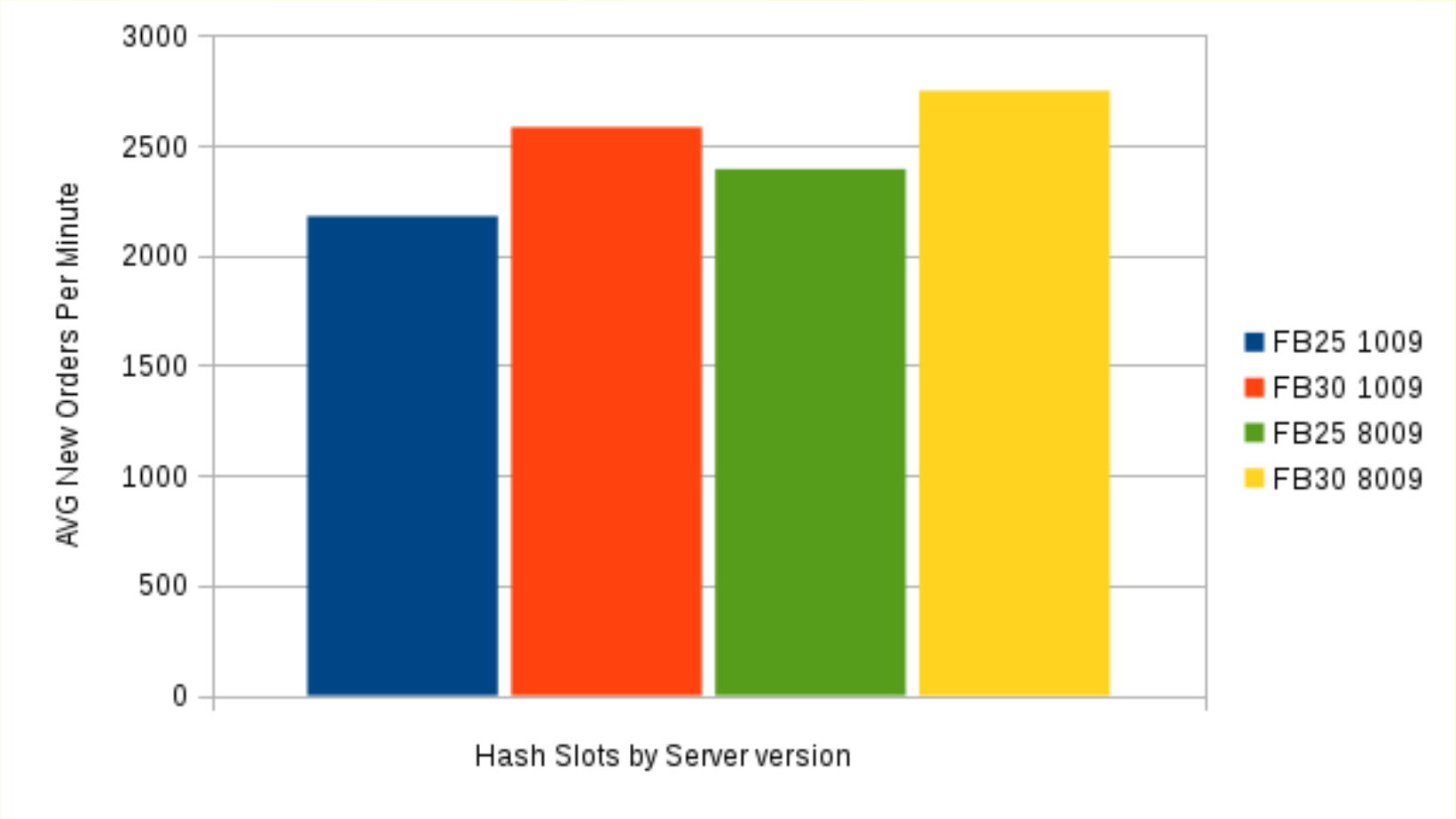
Pool sizing summary

- Choosing a smaller pool size for Large Dbs would make sense
- It would also be interesting to try much larger buffer sizes for a smaller number of connections

Reducing contention with LockHashSlots

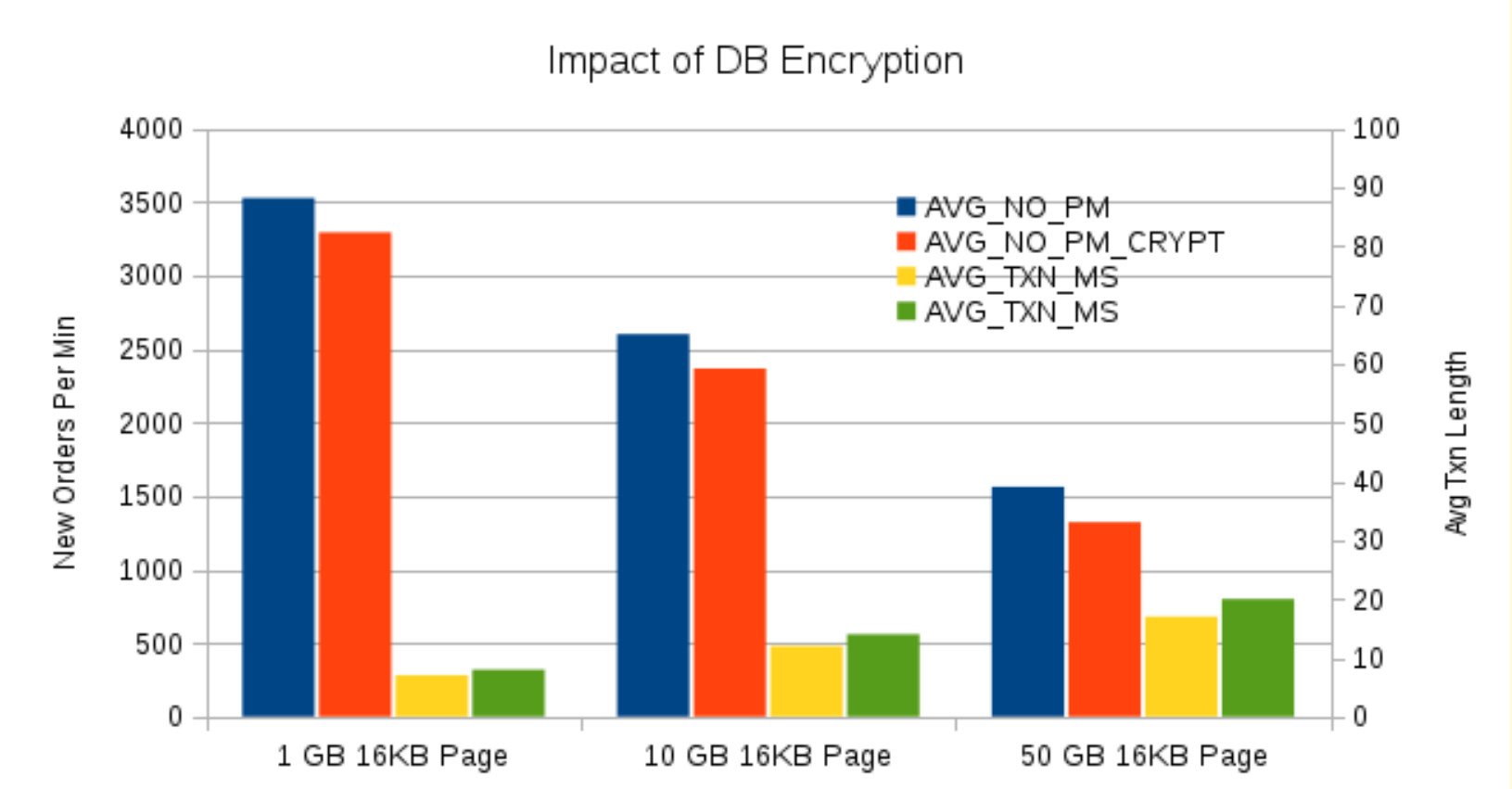
- The LockHashSlots parameter can improve queueing for locks under heavy concurrent load.
- Default for LHS increased to 8191 in FB3, from 1009 in FB2.5
- How has this affected performance?

Hash Slots



IBPhoenix
THE POWER WITHIN

Database encryption



- Yep, there is a cost
- Avg txn times are ~15% slower for the encrypted LDB
- This cost is outweighed by overall perf improvement in FB3

IBPhoenix
THE POWER WITHIN

Nbackup performance

- No hard data available. Testing Nbackup broke the server.
- Again db size is a major factor
- Small dbs are no problem

- I had hoped to test the following :

  - Difference between internal and external backup
  - Impact of backup on active Large Dbs
  - Is it quicker to take the database off-line to perform the backup?

Building a Test Harness

- All benchmarks are artificial.
- The only way to reliably benchmark your application is to build your own test harness.
- At least 2 computers required.
- Ideally no other activity should be running on the server, unless they are part of the test.
- The test app runs on the client and generates threads to simulate multi-user activity.
- The results are stored in a separate database.

Identify your workloads

- For example, in TPC-C there are five, related to new orders, payments, delivery, order and stock checking.
- Each workload is weighted for frequency. Roughly a third each for orders, payments and delivery (which all include writes). Just a few percent each for the status checks.
- These weights are configurable.
- Jobs are assigned to available connections according to the weighting.
- A certain percentage of workloads must fail (rollback).

Only test ONE thing at a time

- Each of the following is a single test
- for OS in linux, windows
  for FBVER in fb25, fb30
    for ARCH in superclassic, classic, superserver
      for DISC in ssd, hdd
        for PAGE_SIZE in 4,8,16
          for TERMINALS in 5,10,15,20 etc
- Obviously some of the above can be elim-inated.

Before a test session starts

- Database is swept.
- Gstat -r is run (and results saved.)

Track the results

As each txn completes some info is stored

- Type of workload
- Success or failure
- Update of min, average and max txn times

When the session finishes

- Lots of data is saved into the results database…
  - Benchmark configuration
  - Stats from MON$DATABASE
  - Number of TXNs executed
  - App specific stats such as New Orders placed.
  - Gstat -r is run to see how much garbage has built up in the db. In the event of anomalies we can compare with the gstat -r from before the session starts.

About the results database

- Obviously some views and stored procedures are added to aid analysis
- Important to store as much data about a test run as possible, so that trends can be identified and questions answered that hadn't been thought of when the project started.

Conclusions

- Benchmarks are about as reliable as the weather forecast.
- Sometimes they are correct.
- Firebird 3 would seem to be faster than Firebird 2.5 if we can believe this benchmark.
- In a multi-user environment changing one parameter has side effects if other factors change.
- The only reliable benchmark is your own app running on your own test harness. (And even then...)

Questions?