Performance Comparison of Postgres 8.4 Vs Postgres 8.3

Jignesh Shah
Staff Engineer,
ISV Engineering,
Sun Microsystems Inc

PGCon 2009

About Me

- Working with Sun Microsystems for about 8 1/2 years
  - All of them in ISV Engineering (MDE)
  - Primarily responsibility at Sun is to make ISV and Open Source Community Software applications run best on Sun
- Worked with various databases (Postgres, MySQL, DB2 UDB, Progress OpenEdge, Oracle)
- Prior to Sun worked as ERP Consultant
- Worked with various ERP (QAD, Lawson) and CRM (Dispatch-1), etc
- Previous responsibilities also included : Low Cost Data Warehousing
- Blog: http://blogs.sun.com/jkshah

Goal / Agenda

- Goal: Regression test of 8.4 Vs 8.3
- Performance Changes in Postgres 8.4
- Test Setup
- Benchmarks
- Summary

Performance Changes in Postgres 8.4

Postgres 8.4 Changes

- Feature release
- Not a performance release but
  - Has features which may impact performance
- New Free Space Map Implementation
  - Gets rid of max_fsm_pages & max_fsm_relations
- New Visibility Maps
  - Tracks changed tuples from last VACUUM
  - Improves VACUUM Performance
- Multi process restore with pg_restore -j option
  - Trick used : Remove constraints, load tables in parallel and then re-apply constraints

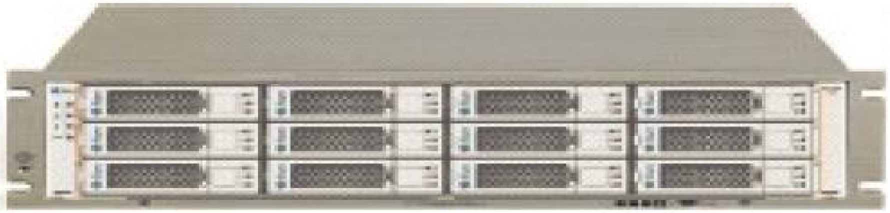
Other Postgres 8.4 Tweaks/Changes

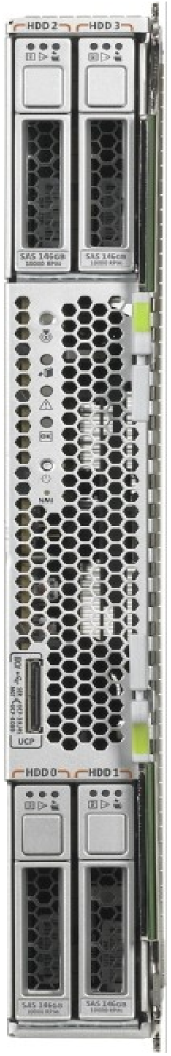
- And many other...
  - **Constraint exclusion = 'partition' (it was “off” in PG 8.3)**
  - **default_statistics_target = 100 (it was 10 in PG 8.3)**
  - New GUC stats_temp_directory
  - Enhanced Optimizer statistics calculations, full-text columns
  - New semi-join and anti-join executor methods
  - concurrent_io_reads for readaheads of bitmap index scans
    - Not supported on OpenSolaris
  - cursor_tuple_fraction
  - Hash index for DISTINCT/UNION/INTERSECT/EXCEPTION
  - Improvement in build speed for Hash index
  - Reduction in memory footprint for trigger handling
  - suppress_redundant_updates_trigger() added

Test Setup

Test Setup

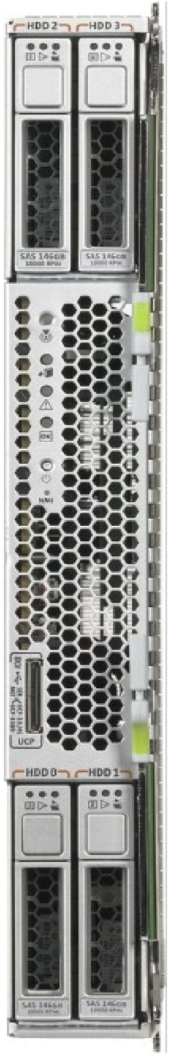
- Sun Fire X6270
  - > 2x Quad Core Intel Xeon 5500 (Nehalem) Series
  - > 8 Cores, 16 threads (with Hyperthreads ON)
- OpenSolaris 2009.06 preview (build 111a)
- Sun StorageTek 2540
  - > 12 x 146 GB Drives (FC)
- Workloads used:
  - > Pgbench
  - > Sysbench – OLTP (read only & read write)
  - > dbt2 with W 100
  - > dbt3 with 5GB sca
  - > iGen v1.6

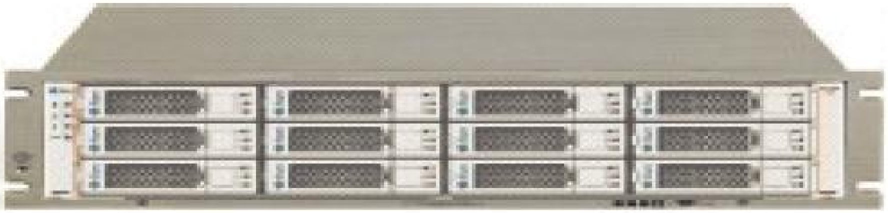




Test Layout

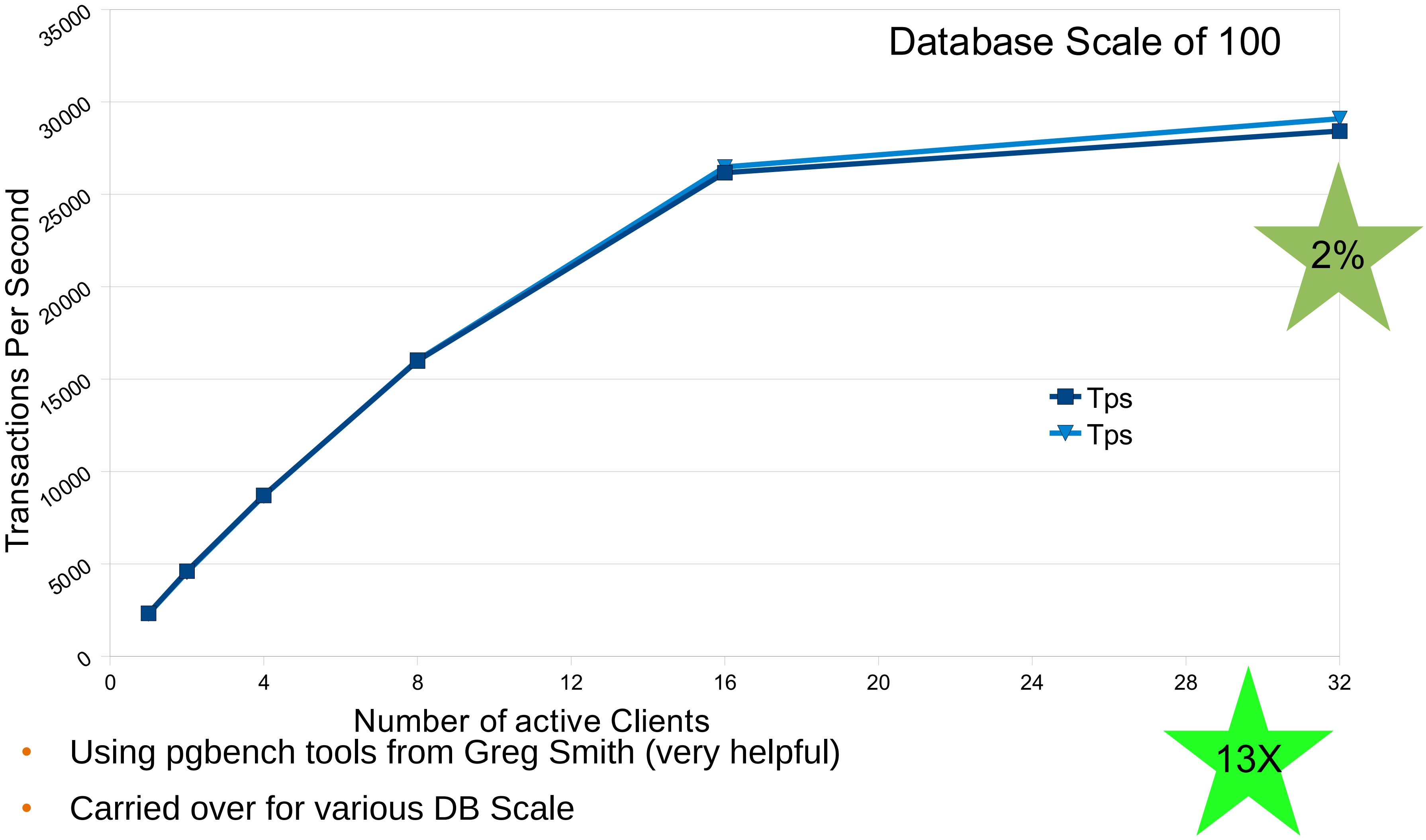
- PGDATA on 4-disk RAID0 ZFS
- PGLOG on 4-disk RAID0 UFS (forcedirectio)
- For each test
  - Database dropped, recreated, reloaded
  - Pretty much same procedures were used for both versions
  - Multiple runs were done to check that it wasn't a fluke





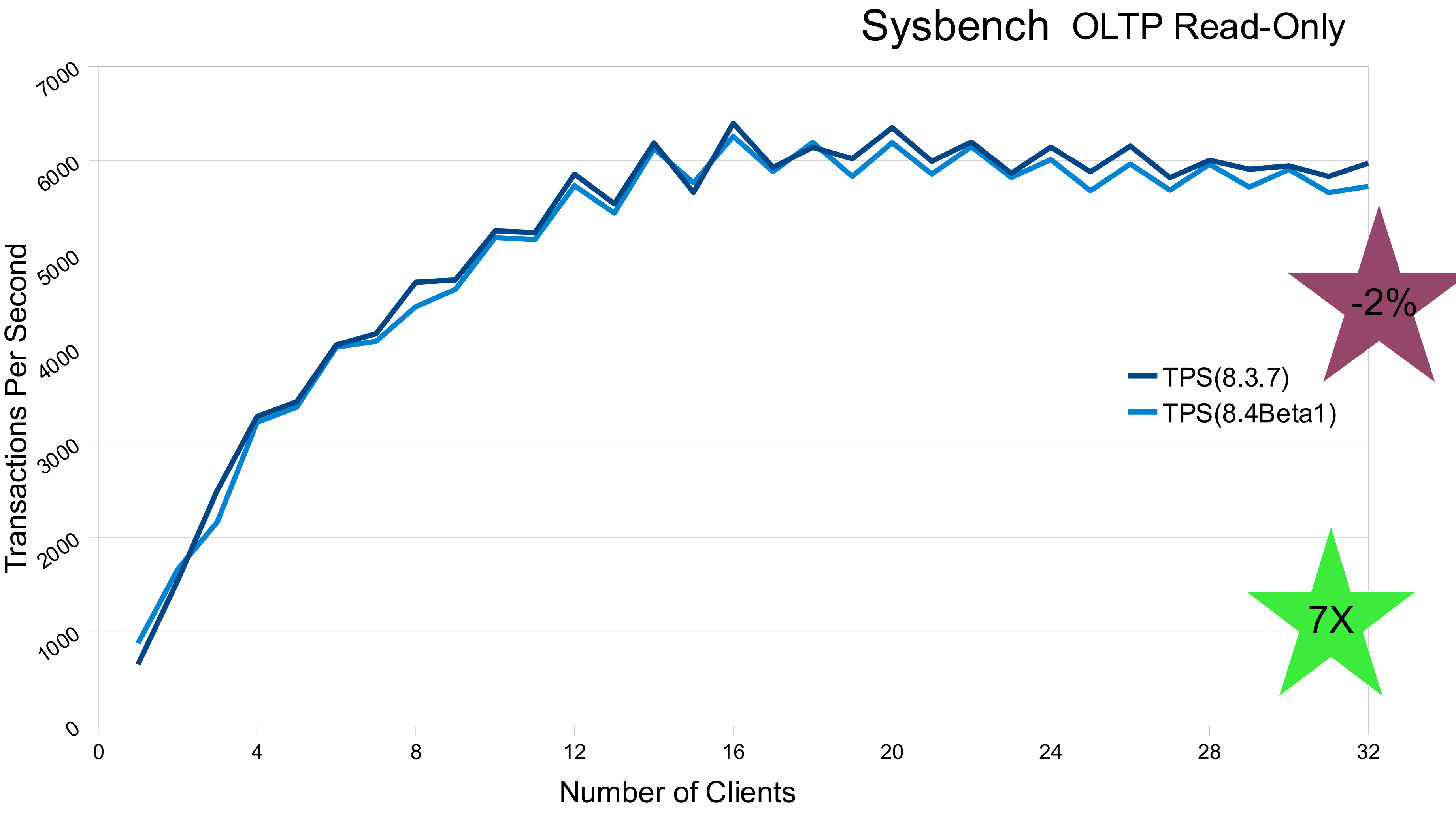
Benchmark Comparisons

PgBench (Select) Comparison

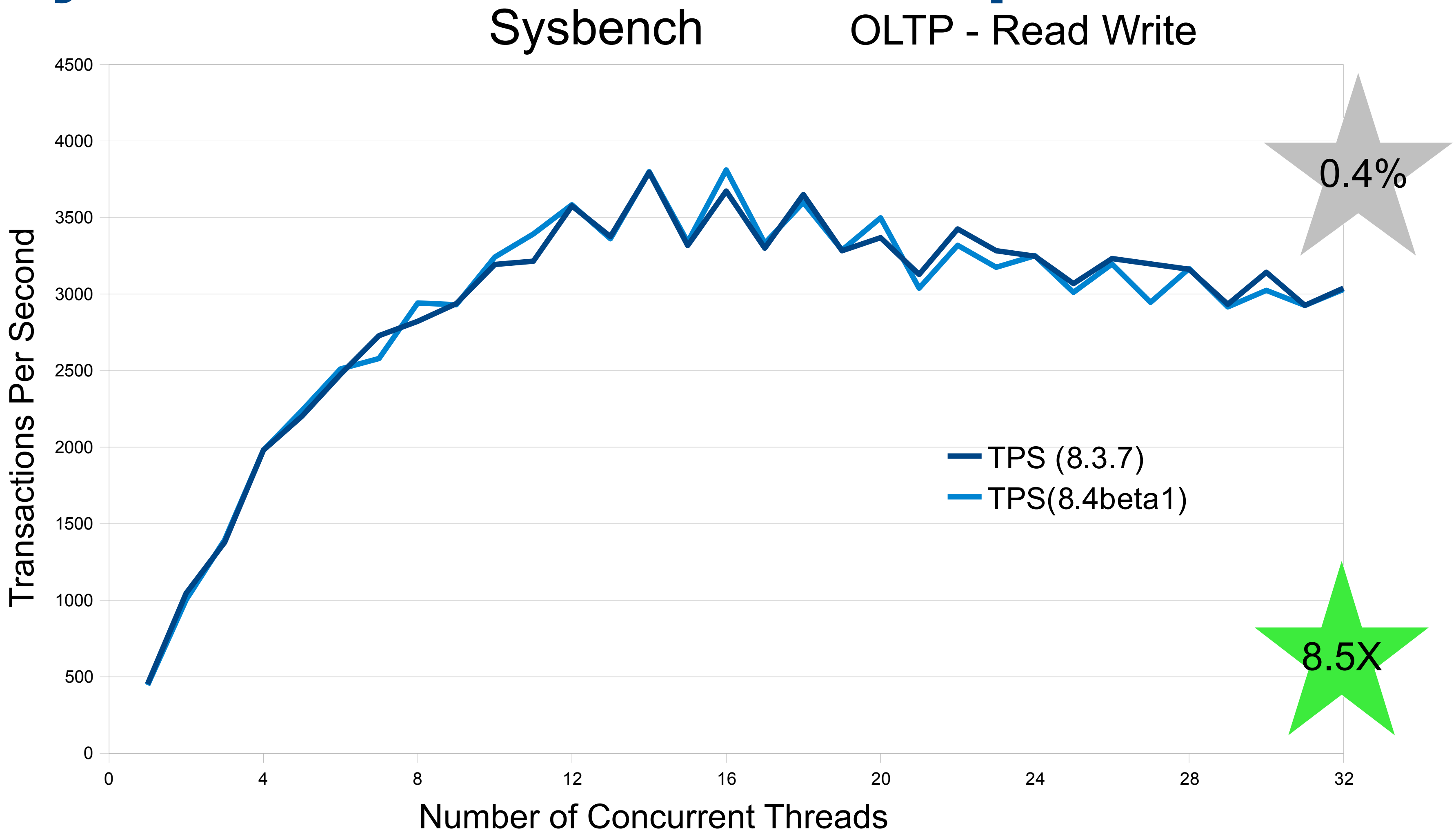


- Using pgbench tools from Greg Smith (very helpful)
- Carried over for various DB Scale

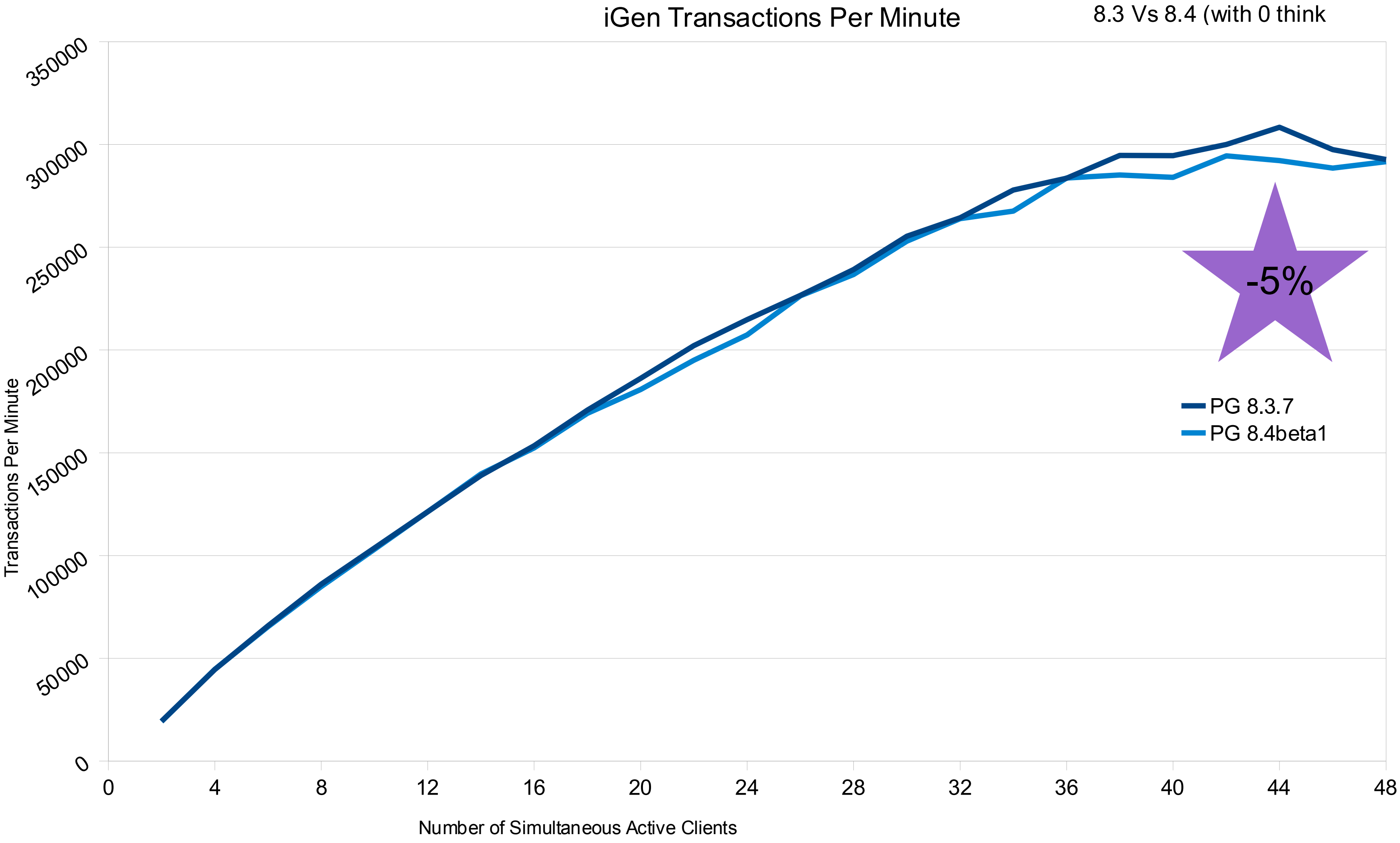
Sysbench OLTP RO Comparison



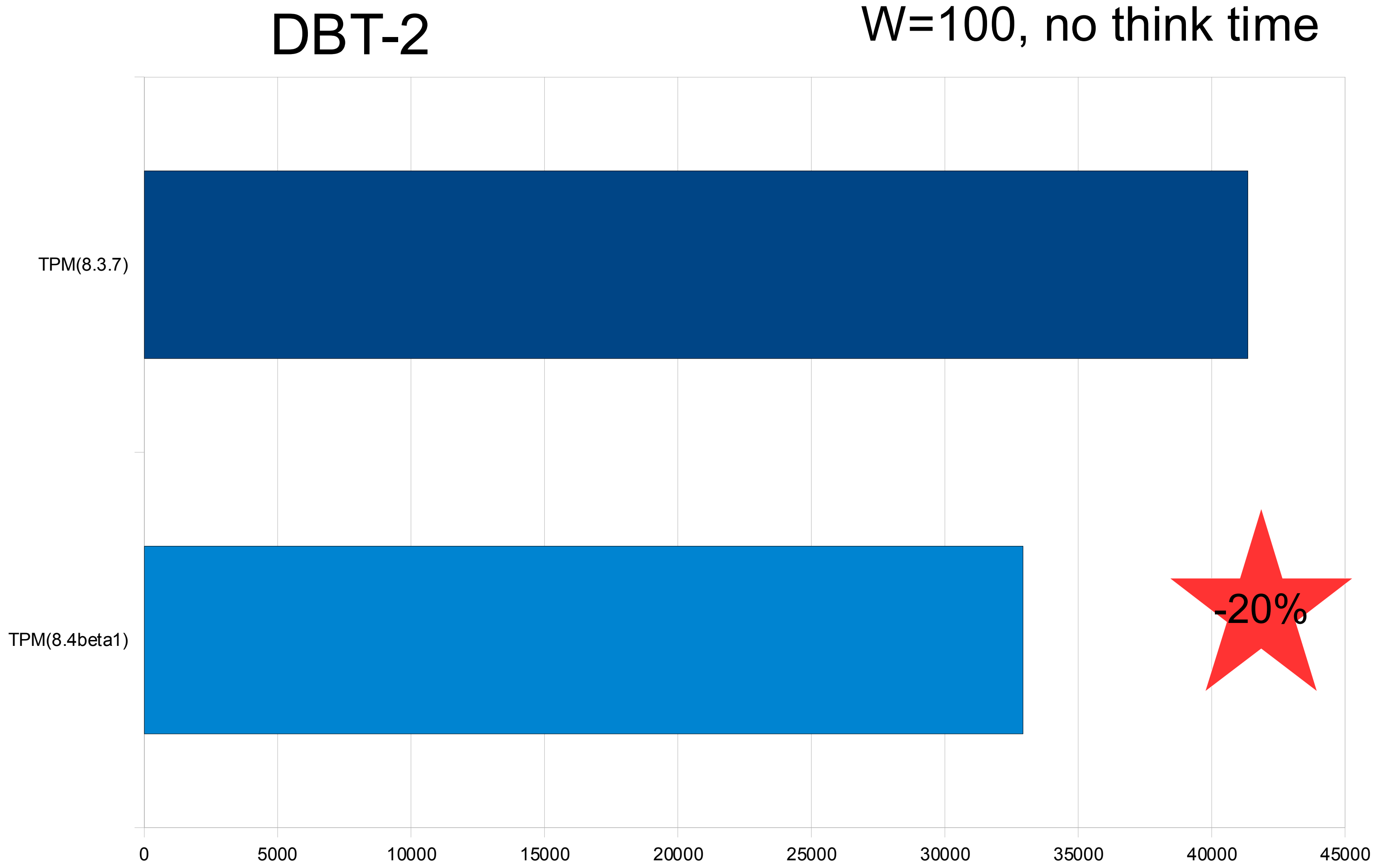
Sysbench -OLTP RW Comparison



iGen Comparison



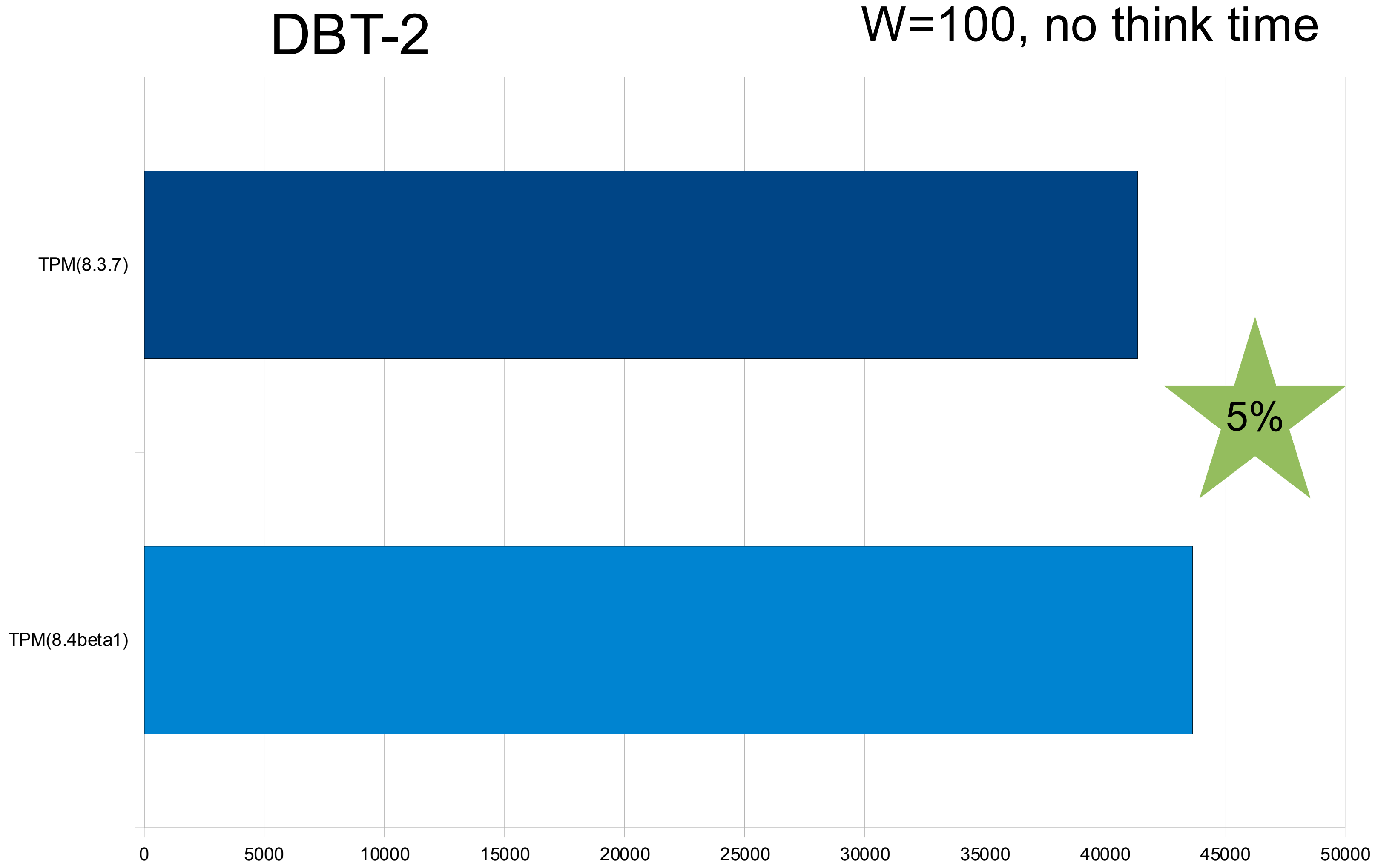
DBT-2 Comparison (First Set of tests)



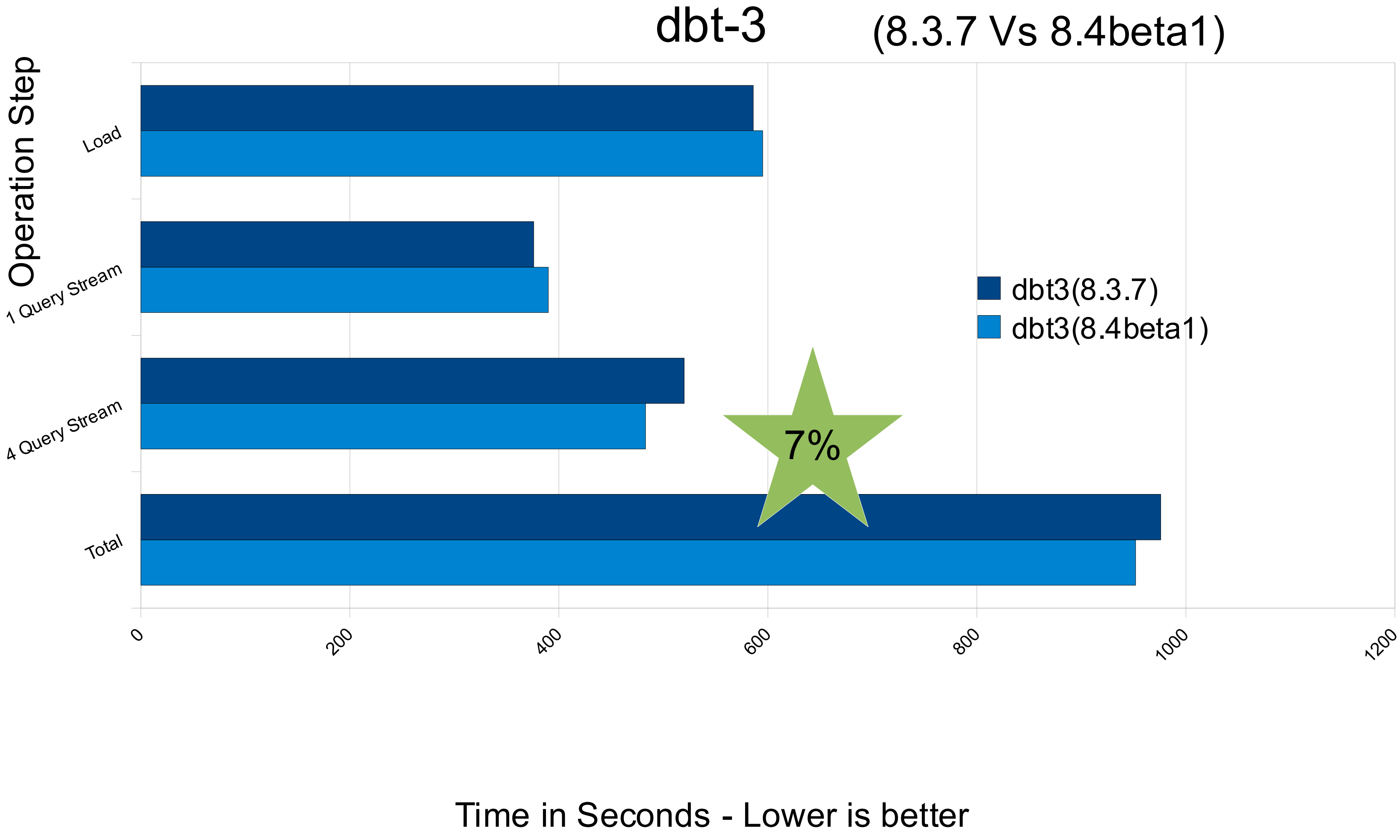
DBT-2: Glitches Observed

Tunable	PG 8.3	PG 8.4beta1	Comments
default_statistics_target	10	100	Dbt2 like the value at 10
constraints_exclude	off	partition	Dbt2 seems to like the value to off
			With new defaults dbt2 take about 15-20% hit

DBT-2 Comparison



DBT-3 Comparison



8.3 Vs 8.4 Summary Results

Workload	Performance Gain/Loss in %
PGBench	2.37
Sysbench RO	-2.17
Sysbench RW	0.4
IGEN V1.6	-4.50
DBT-2 (W=100)	5.54*
DBT-3 (SF=5)	6.83
	* default_statistics_target = 10 constraints_exclude = off

Results Interpretations

- 8.4 Performance seems to be within +/- 5% for most workloads
- Anything +/- 2% is potentially error margin in tests itself
- Long running simultaneous queries may see improvements
- Some workloads may be impacted by default_statistics_target and constraint_exclude

PostgreSQL Lock Waits by Workload

-For Developers!

pglockwait_84.d Tool

- Prints out a summary every 10 second
- Can be used on single backend using $PID or all backends using '*'
- Output is Lock Type, wait time for backends spent in wait state for locks and counts for waits
- Useful to figure out how much time was wasted in wait states for acquiring locks

PGBench Select.sql- Hot Locks Waits

Lock-Id	Mode	Wait-Time(ms)	Count
LockMgrLocks	Exclusive	7006	94922

- About 40 backends in this snapshot
- LockMgrLocks are aggregated for all NUM_LOCK_PARTITIONS
- Wait-times and Count to be taken with a grain of salt
- Important data is top lock waits types

Sysbench RO- Hot Locks Waits

Lock-Id	Mode	Wait-Time(ms)	Count
LockMgrLocks	Exclusive	35	601

Sysbench RW- Hot Locks Waits

Lock-Id	Mode	Wait-Time(ms)	Count
WALInsertLock	Exclusive	105	9117
LockMgrLocks	Exclusive	72	5858
ProcArrayLock	Exclusive	50	1422
DynamicLocks	Shared	32	1767
CLogControlLock	Shared	17	1433
ProcArrayLock	Shared	11	912
DynamicLocks	Exclusive	8	388
CLogControlLock	Exclusive	2	234

Igen Hot Locks Waits Observed

Lock-Id	Mode	Time(ms)	Count
LockMgrLocks	Exclusive	53837	80247
ProcArrayLock	Exclusive	40130	89132
WALInsertLock	Exclusive	4895	26081
ProcArrayLock	Shared	1028	5360
XidGenLock	Exclusive	520	1394
CLogControlLock	Exclusive	60	1226
CLogControlLock	Shared	30	976
BufMappingLocks	Exclusive	7	14
DynamicLocks	Shared	1	11

DBT2 Hot Locks Waits Observed

Lock-Id	Mode	Time(ms)	Count
BufMappingLocks	Shared	8534	28207
LockMgrLocks	Exclusive	4186	11019
BufMappingLocks	Exclusive	1604	9213
WALInsertLock	Exclusive	1373	25942
ProcArrayLock	Exclusive	300	1246
DynamicLocks	Shared	163	597
CLogControlLock	Exclusive	136	759
CLogControlLock	Shared	52	991
BufFreelistLock	Exclusive	22	790
DynamicLocks	Exclusive	12	29
ProcArrayLock	Shared	4	263

Useful Dtrace Scripts

- pgtps.d – Transactions Per Second
- pglockwait_84.d – Lock Wait Statistics
- PostgreSQL Dtrace Toolkit on pgfoundry
  - > http://pgfoundry.org/projects/dtrace/

- Available from http://blogs.sun.com/jkshah

Acknowledgements

- Greg Smith – For pgbench tools
- Mark Wong - DBT-2
- Sun Microsystems Inc – Hardware, Time

Q & A

- Blog: http://blogs.sun.com/jkshah