PostgreSQL as a benchmarking tool

How it was used to check and improve the scalability of the DragonFly operating system

François Tigeot
ftigeot@wolfpond.org

About Me

- Independent consultant, sysadmin
- Former ccTLD system engineer
- *BSD user since ~= 1999
- Also a PostgreSQL user since ~= 1999
- Introduced FreeBSD in the .fr registry
- DragonFly developer since 2011

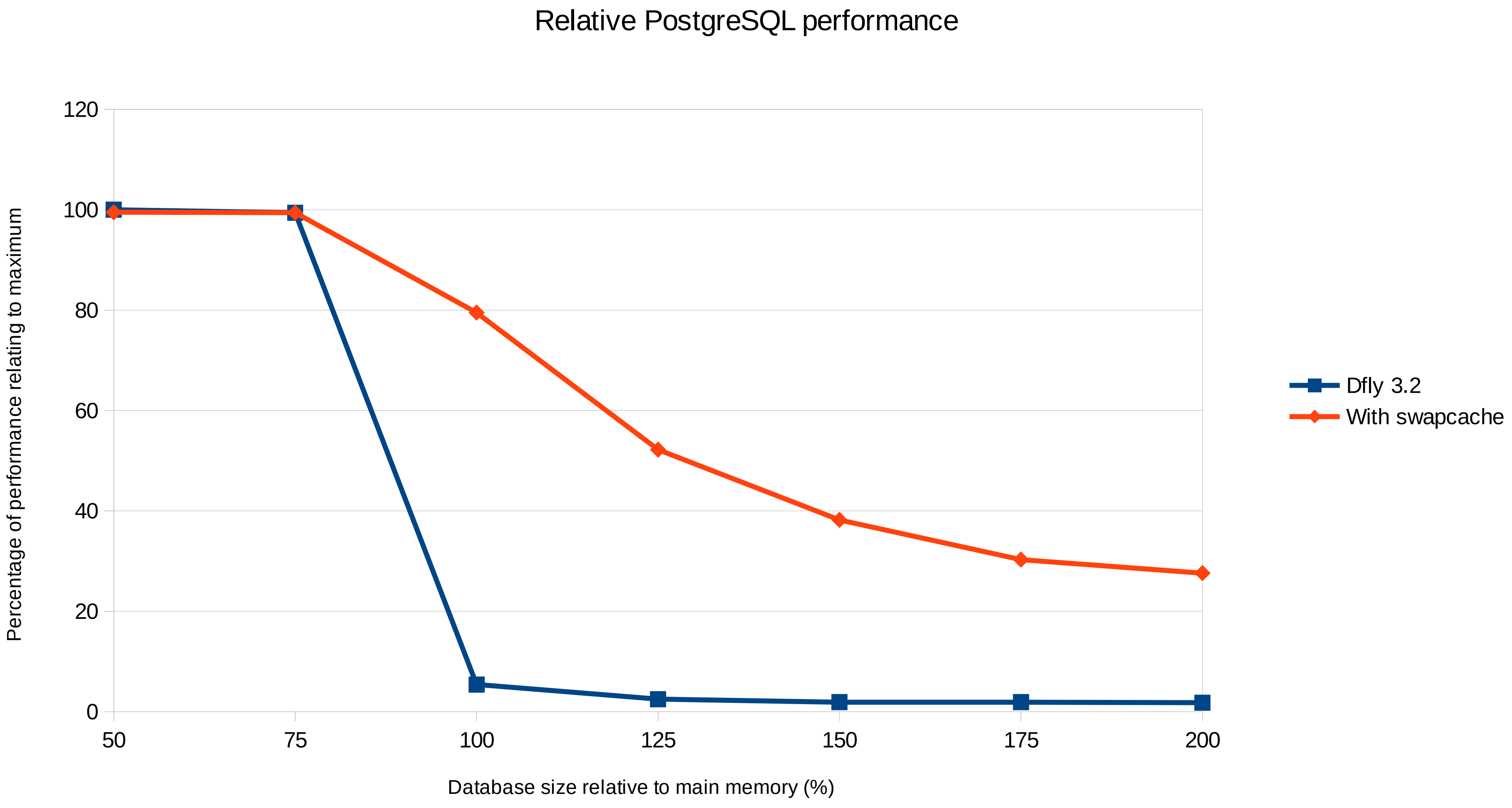
About DragonFly

- Unix-like Operating System
- Forked from FreeBSD 4.8 in 2003
- By Matthew Dillon (not the actor)
- Aims to be high-performance
- Uses per-core replicated resources and messaging
- Many operations are naturally lockless

About DragonFly (2)

- Innovative features very useful for some workloads
- Swapcache: second-level file cache
- Uses existing swap infrastructure
- Optimized for SSDs

Swapcache


Relative PostgreSQL performance
Percentage of performance relating to maximum
120
100
80
60
40
20
0
50
75
100
125
150
175
200
Database size relative to main memory (%)
Dfly 3.2
With swapcache

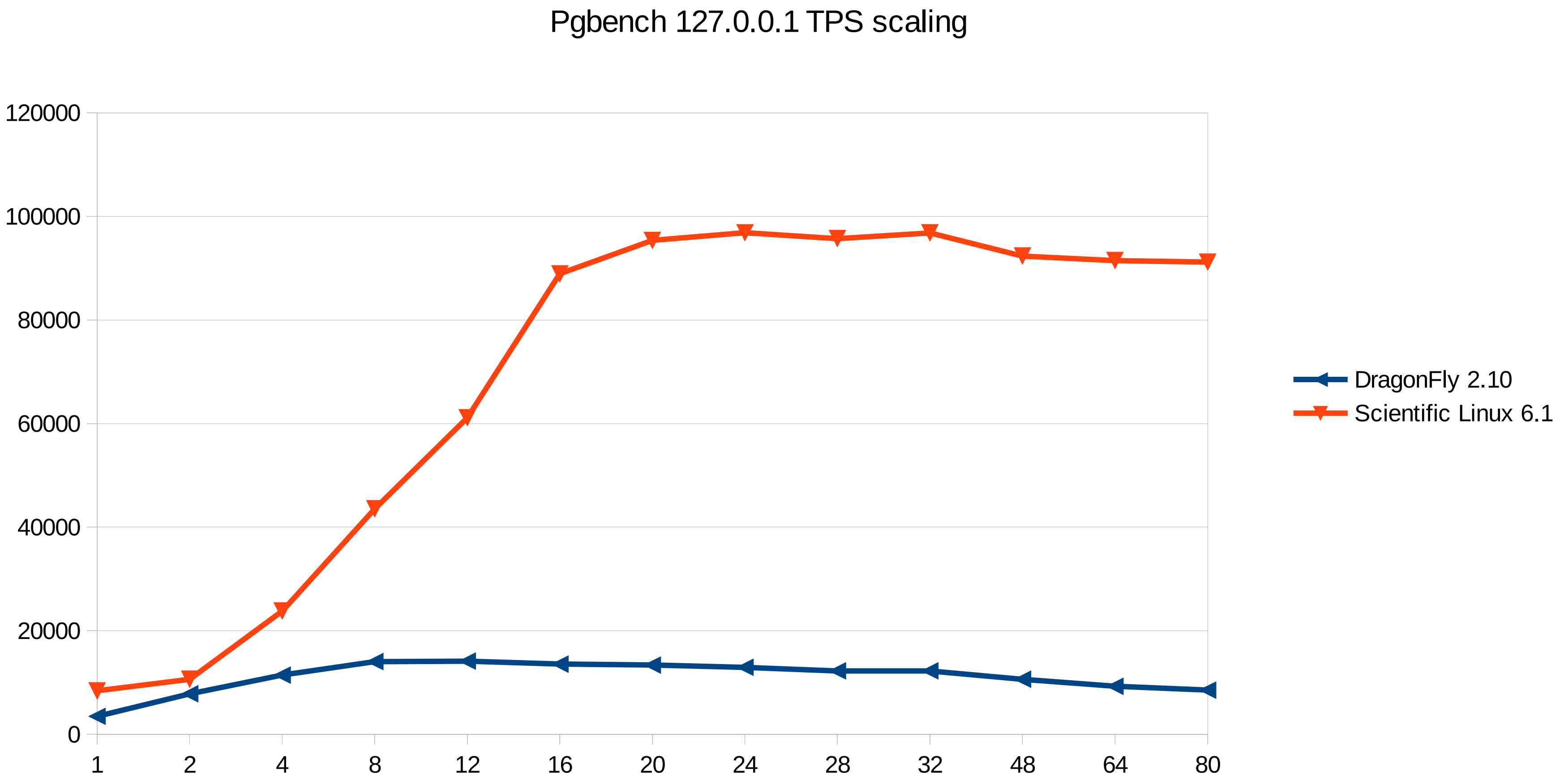
November 2011

- PostgreSQL 9.1
- DragonFly 2.10 and 2.13 (development version)
- Dual Xeon, 24 threads, 96GB RAM
- Global MP lock removed from the kernel after version 2.10
- Was looking for benchmarks showing CPU scalability
- PGbench (read-only) was a good fit

November 2011 (2)

- Some crashes and bugs with high PGbench loads
- Quickly fixed (generally in less than a day)
- Deadlocks in the VM subsystem
- Overflows and races in zalloc()
- Races in the SysV shared memory subsystem
- Etc...

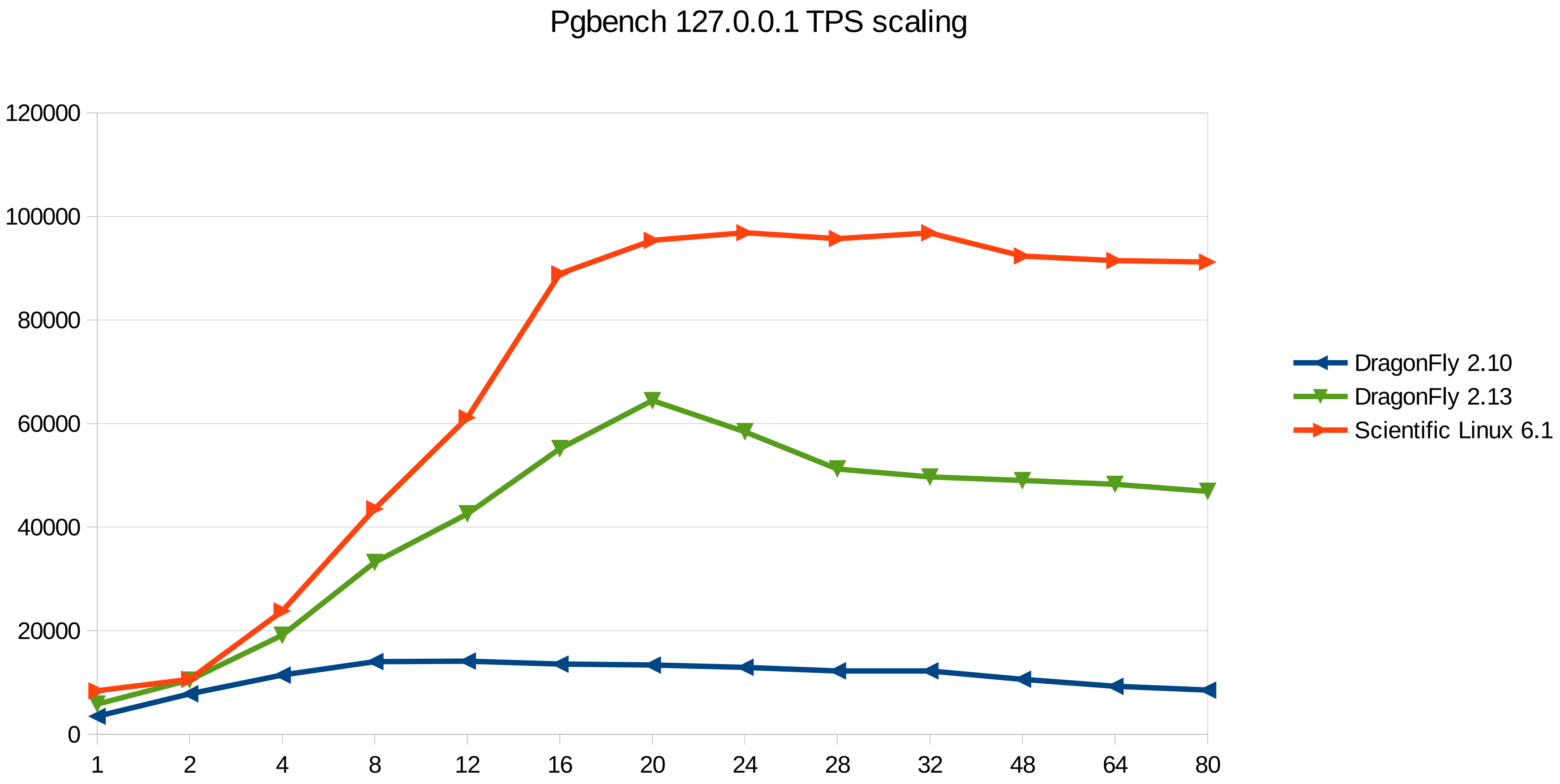
November 2011 (3)



November 2011 (4)

- System changes to improve performance
- Remove MP lock from SysV semaphore code
- Improve select() and poll()
- New “dmalloc” lockless memory allocator in libc
- Improve other memory allocation code paths
- Make it possible to concurrently process huge numbers of page faults

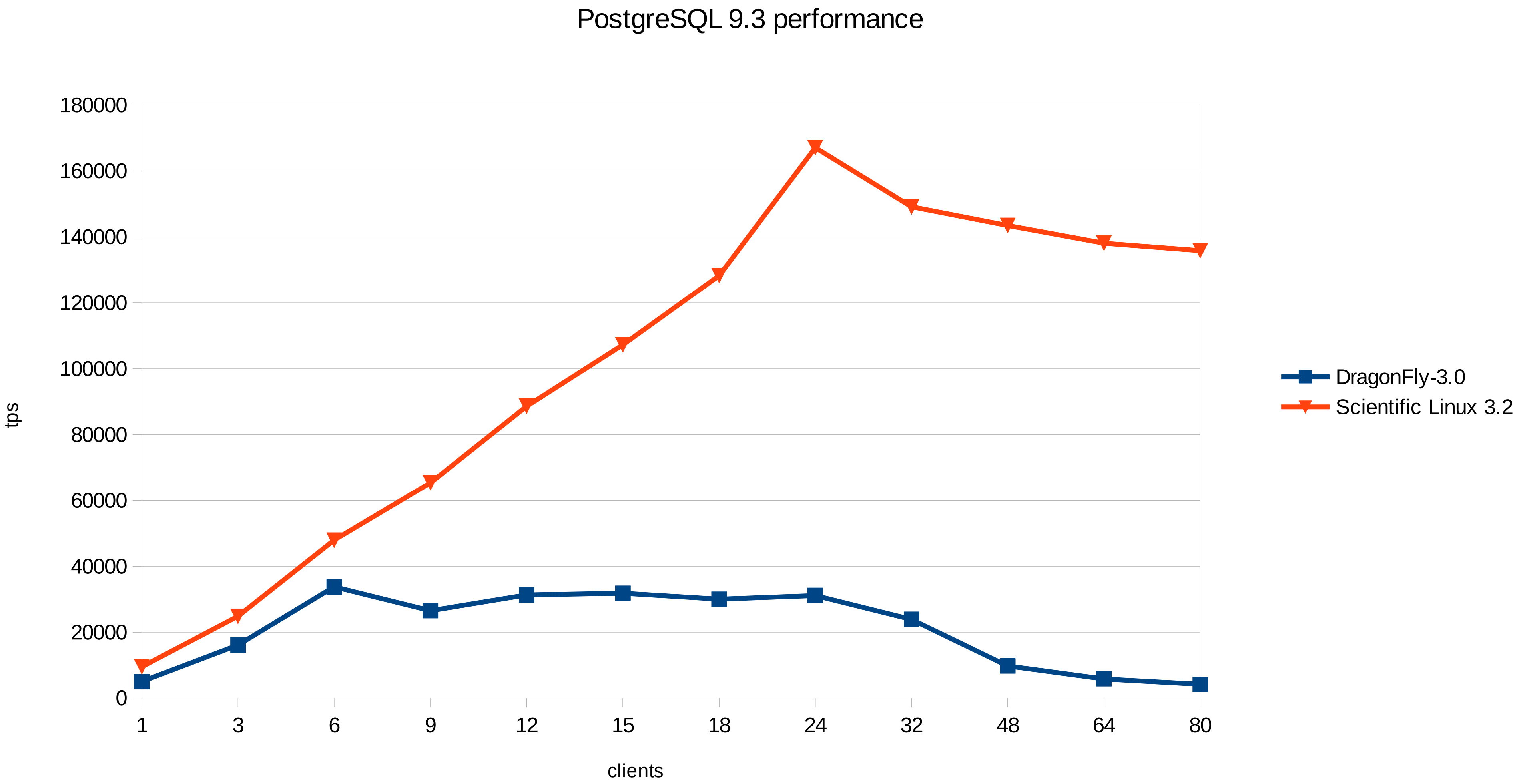
November 2011 (5)



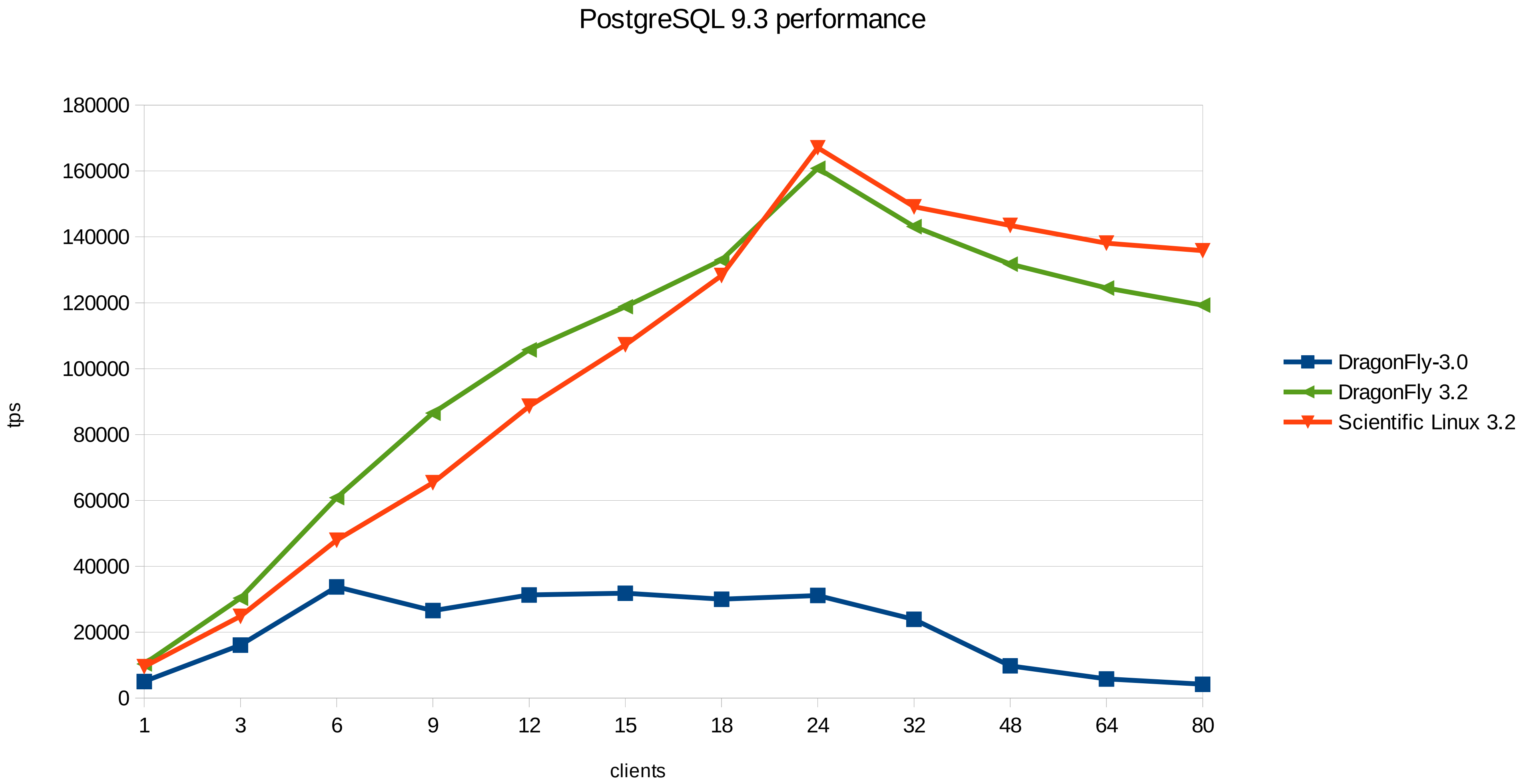
September-October 2012

- PostgreSQL 9.3 (development branch using mmap)
- DragonFly 3.0 and 3.1 (development, future 3.2)
- Dual-Xeon, 24 threads, 24GB RAM
- Benchmark
- Find bottleneck
- Fix or tweak
- Repeat
- Very time consuming

September-October 2012 (2)



September-October 2012 (3)



September-October 2012 (4)

- Mihai Carabas added a CPU topology framework to the kernel (work sponsored by Google)
- Old BSD scheduler changed to take this information into account
- Still significant limitations due to the original design
- The scheduler itself was single-threaded

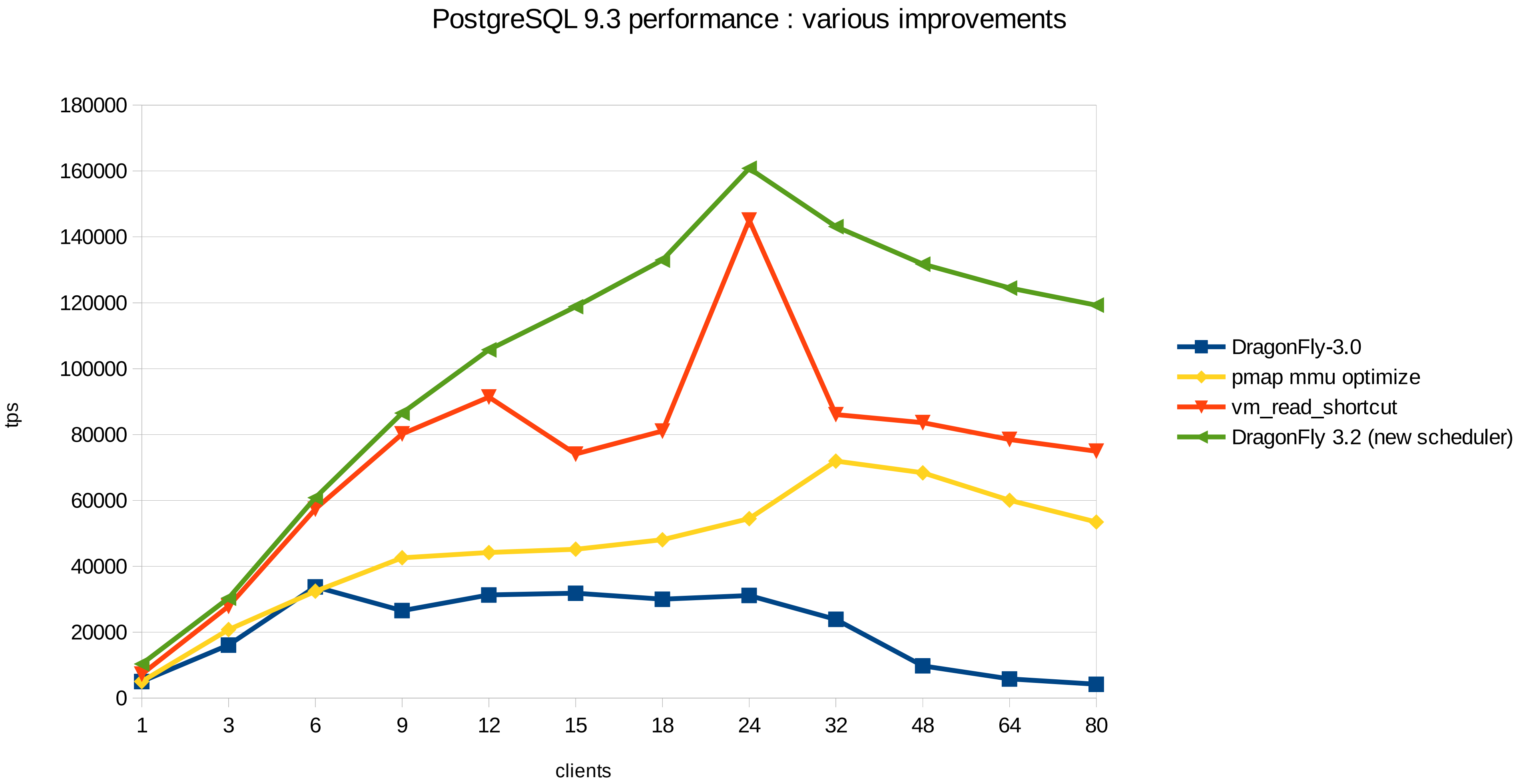
September-October 2012 (5)

- Matt Dillon wrote a new scheduler
- Schedules processes as close as possible to the place they were last run on
- Avoids unnecessary competition for resources
- Doesn't use different hardware threads from the same CPU core at the same time if possible
- Globally balances the load and takes the machine topology into account to avoid hot spots

September-October 2012 (6)

- Other improvements
- Many default values tuned for new 64-bit machines (buffer cache)
- PMAP MMU optimizations. Avoids having to fault huge amounts of pages for processes using shared memory.
- Read shortcuts through the VM subsystem

September-October 2012 (7)



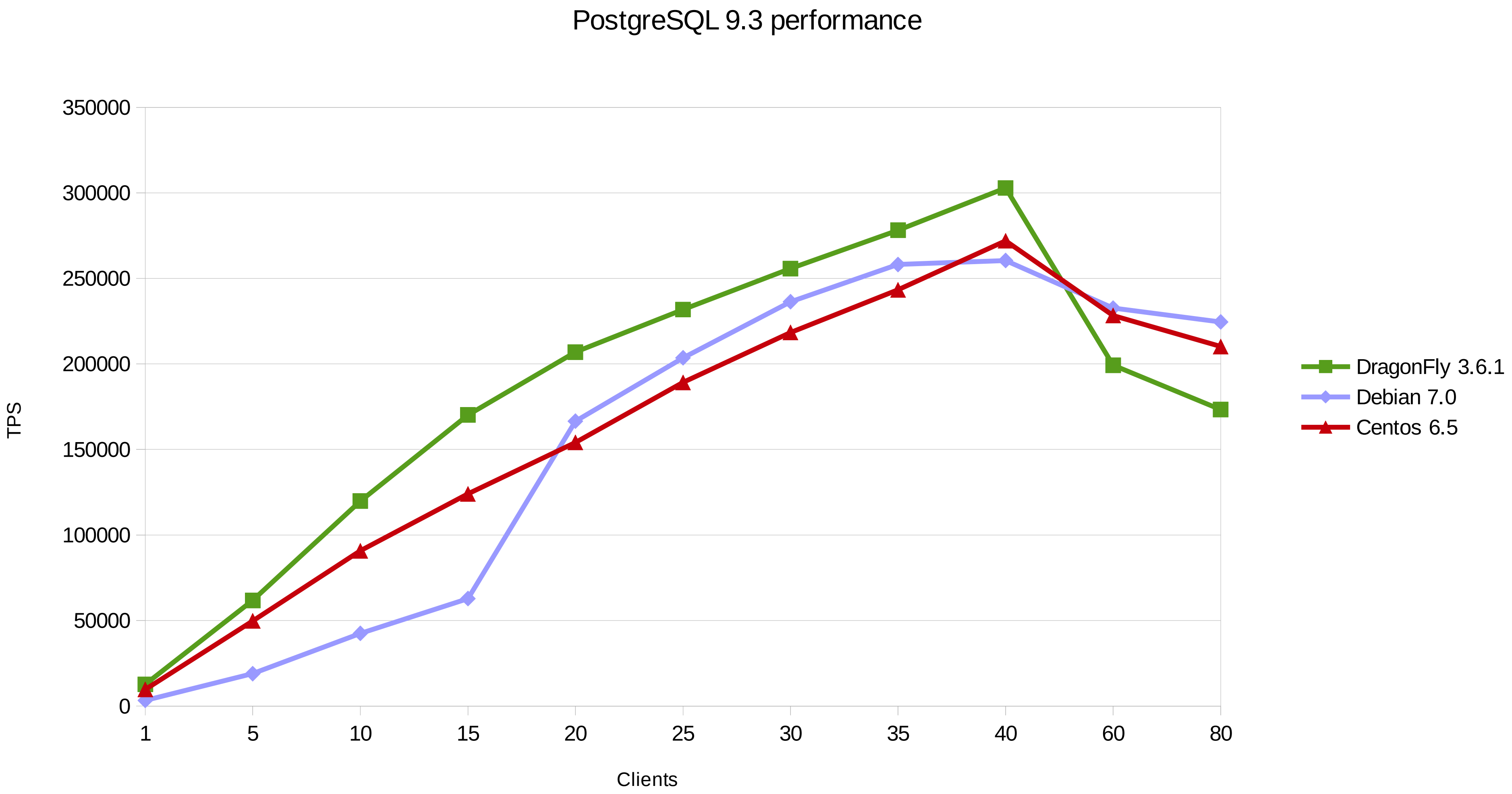
September-October 2012 (8)

- Performance improvements not PostgreSQL-specific
- Number of MP MMU invalidations globally reduced
- read() performance globally improved
- Across the board improvements of performance under load

March 2014

- PostgreSQL 9.3, DragonFly 3.6.1
- Dual-Xeon, 40 threads, 128GB RAM
- No PostgreSQL-specific performance work this time
- Improvements wrt DragonFly 3.2 likely caused by analysis of Poudrière runs in 2013
- Poudrière = package building tool originally from FreeBSD
- Very CPU + fork/exec + I/O intensive

March 2014 (2)


PostgreSQL 9.3 performance
TPS
350000
300000
250000
200000
150000
100000
50000
0
1
5
10
15
20
25
30
35
40
60
80
Clients
DragonFly 3.6.1
Debian 7.0
Centos 6.5

Thank you

- Questions ?