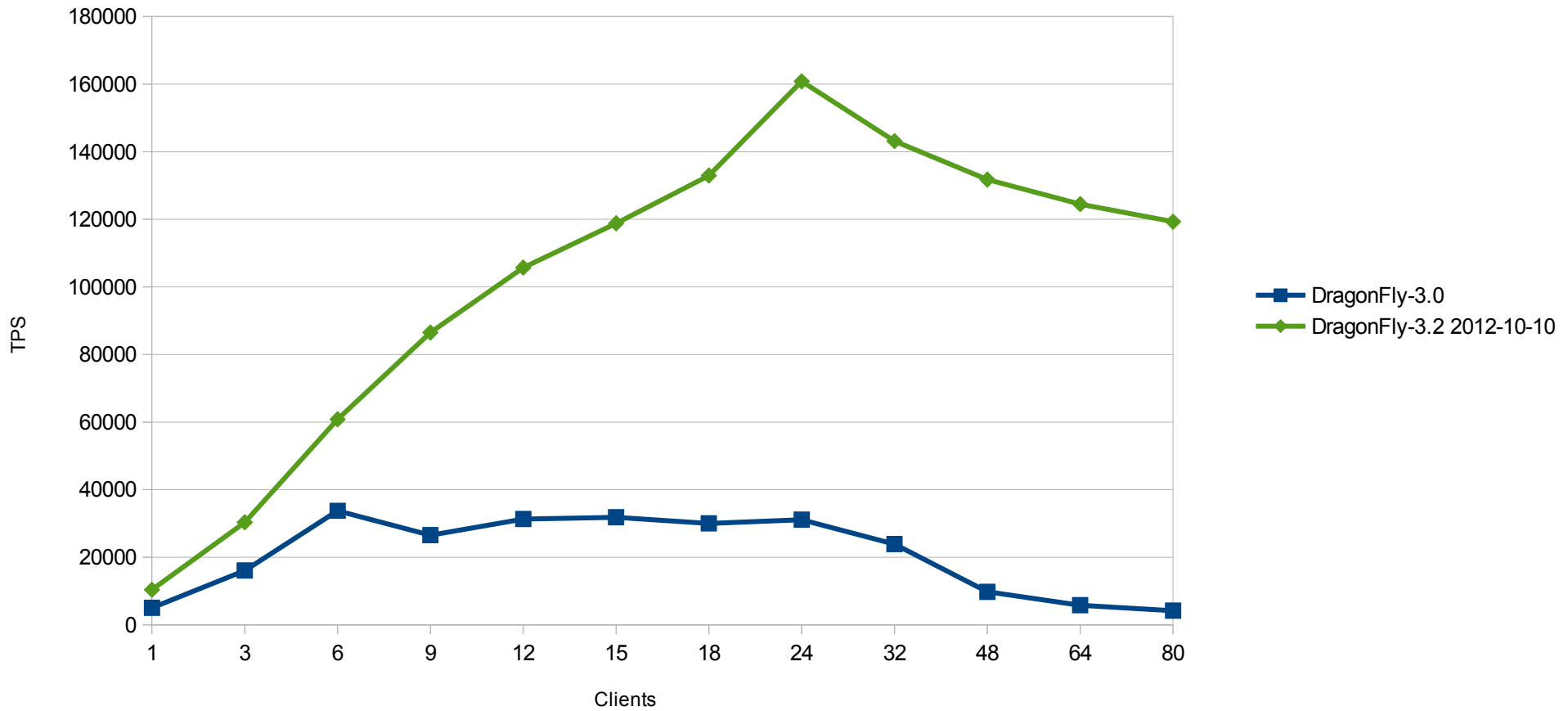


Results

PostgreSQL 9.3 Pgbench Transactions Per Second, 2x Xeon X5650 / 24GB (24 threads), Unix socket												
Clients	1	3	6	9	12	15	18	24	32	48	64	80
DragonFly-3.0	5024	16077	33735	26530	31304	31825	30002	31132	23898	9769	5786	4193
DragonFly-3.2 2012-10-10	10363	30333	60809	86492	105704	118805	132948	160813	143131	131742	124481	119276

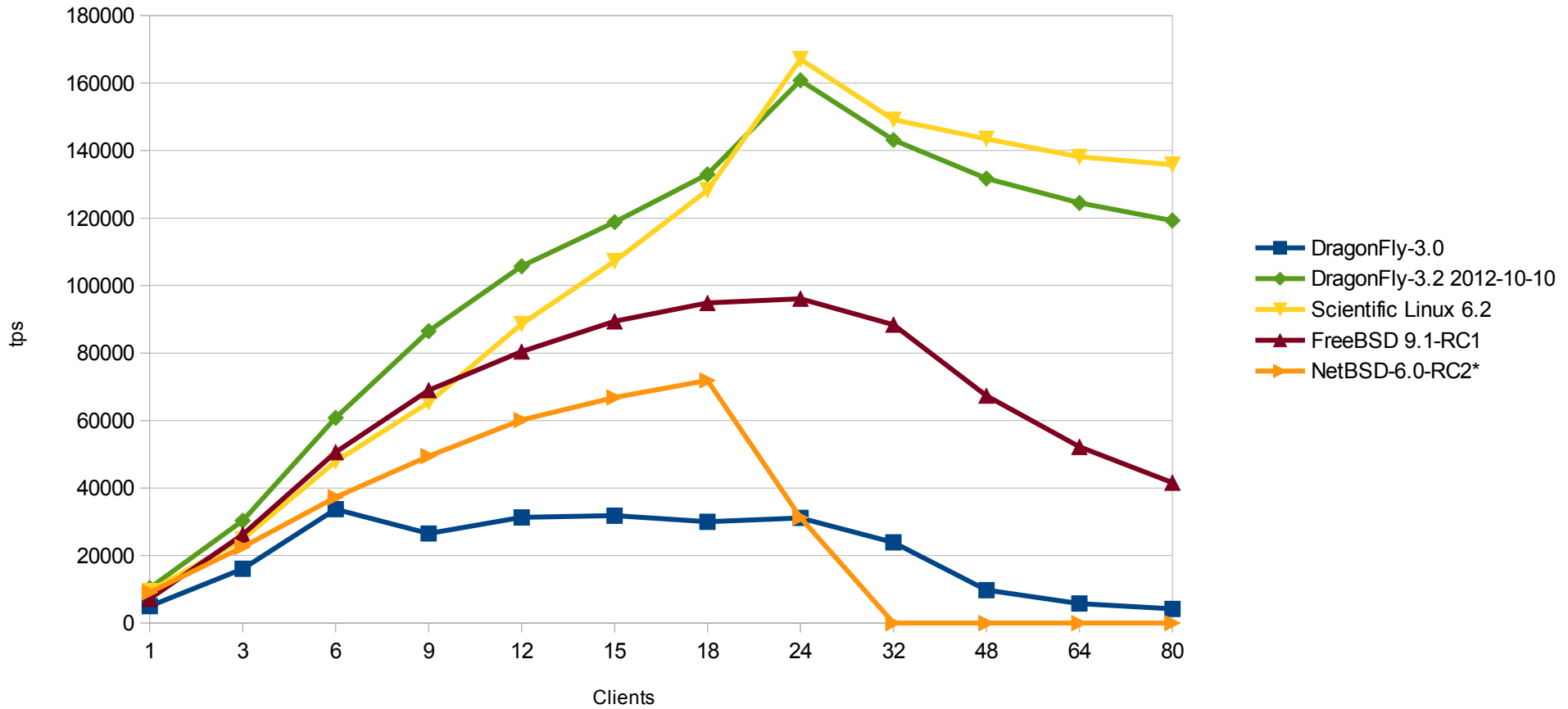
PostgreSQL 9.3 mmap performance



Results

PostgreSQL 9.3 Pgbench Transactions Per Second, 2x Xeon X5650 / 24GB (24 threads), Unix socket												
Clients	1	3	6	9	12	15	18	24	32	48	64	80
DragonFly-3.0	5024	16077	33735	26530	31304	31825	30002	31132	23898	9769	5786	4193
DragonFly-3.2 2012-10-10	10363	30333	60809	86492	105704	118805	132948	160813	143131	131742	124481	119276
Scientific Linux 6.2	9518	24822	47921	65354	88592	107255	128261	167080	149119	143467	138124	135827
FreeBSD 9.1-RC1	7156	26175	50613	69014	80443	89398	94858	96098	88420	67356	52214	41586
NetBSD-6.0-RC2*	9073	22543	37297	49454	60146	66837	71892	31231	0	0	0	0

PostgreSQL 9.3 mmap performance



\* NetBSD-6.0-RC2 has many issues and is unable to complete a full benchmark run.

Raw data

Detailed TPS results excluding connections establishing

All tests were done with Postgres 9.3-devel and the new mmap shared memory facility

Clients (and threads)	1	3	6	9	12	15	18	24	32	48	64	80
DragonFly 3.0 mmap	5,659	17,199	35,552	31,191	30,928	32,297	28,600	29,556	26,920	10,299	5,935	4,150
DragonFly 3.0 mmap	4,202	14,198	33,562	29,122	31,558	31,368	30,392	30,383	22,474	9,940	5,702	4,271
DragonFly 3.0 mmap	5,210	16,833	32,091	19,276	31,426	31,809	31,014	33,456	22,301	9,070	5,721	4,157
DragonFly 3.2 2012-10-10	10,341	30,592	60,763	86,238	105,599	118,814	132,905	160,846	143,184	131,872	124,484	119,415
DragonFly 3.2 2012-10-10	10,345	30,194	60,752	86,619	105,723	118,836	132,987	160,792	143,138	131,853	124,384	119,219
DragonFly 3.2 2012-10-10	10,402	30,214	60,911	86,619	105,789	118,765	132,952	160,801	143,071	131,502	124,574	119,195
Scientific Linux 6.2 mmap	9,883	23,825	49,273	65,392	88,944	106,995	128,606	167,035	149,162	143,405	138,247	135,839
Scientific Linux 6.2 mmap	9,422	26,043	47,051	65,398	89,032	107,528	127,081	167,194	149,207	143,690	137,909	136,024
Scientific Linux 6.2 mmap	9,248	24,597	47,438	65,273	87,800	107,241	129,094	167,011	148,989	143,305	138,218	135,617
FreeBSD 9.1-RC2	7,142	26,271	50,678	69,010	80,437	89,420	94,843	95,413	88,354	67,096	52,222	41,602
FreeBSD 9.1-RC2	7,169	26,450	50,608	69,028	80,482	89,326	94,837	95,496	88,447	67,470	52,090	41,653
FreeBSD 9.1-RC2	7,156	25,804	50,552	69,004	80,409	89,446	94,894	97,384	88,458	67,501	52,331	41,504
NetBSD-6.0-RC2	9,679	22,457	37,040	49,395	60,396	66,938	75,170	93,693				
NetBSD-6.0-RC2	9,332	22,788	37,367	49,579	60,019	66,368	66,342					
NetBSD-6.0-RC2	8,207	22,383	37,483	49,387	60,022	67,205	74,163					

NetBSD-6.0-RC2 shuts down under load with this error :

/etc/powerd/scripts/sensor\_temperature: CRITICAL TEMPERATURE! SHUTTING DOWN.

This issue has been reported in PR #46833

Even after disabling powerd, NetBSD is unable to complete a full benchmark serie. After the first 24-clients run, the system becomes unresponsive, some « [system] » named process permanently consuming cpu time.

## Setup

### Hardware:

- 2x Xeon X5650 (24 threads total)
- 24 GB RAM

### Software

- PostgreSQL 9.3-devel, just after the patch changing setup of the shared memory segment to mmap
- Various operating systems

### Goal:

Test scalability with pgbench, show improvements, compare with other operating systems

### postgresql.conf :

```
max_connections = 100
update_process_title = off
autovacuum = off
shared_buffers = 6GB
effective_cache_size = 12GB
```

### Initialize database cluster :

```
/usr/local/postgres-9.3.mmap/bin/initdb -D /usr/local/pgdata.93
```

### Run Postgres :

```
/usr/local/postgres-9.3.mmap/bin/postgres -D /usr/local/pgdata.93
```

### Create test database

```
psql template1
create database bench;
pgbench -i -s 800 bench
```

Scaling factor 800 => ~= 11GB database

### Running tests :

```
pgbench -j 6 -c 6 -T 1800 -S bench
```

# Dummy run to warm up caches for 30 mn

```
#!/bin/sh
```

```
for clients in 1 3 6 9 12 15 18 24 32 48 64 80
do
    THREADS=${clients}
    ./pgbench -j ${THREADS} -c ${clients} -T 300 -S bench > result_${clients}.txt
done
```

Repeat three times and average the individual results

### NetBSD special configuration:

```
/etc/rc.d/powerd stop
```