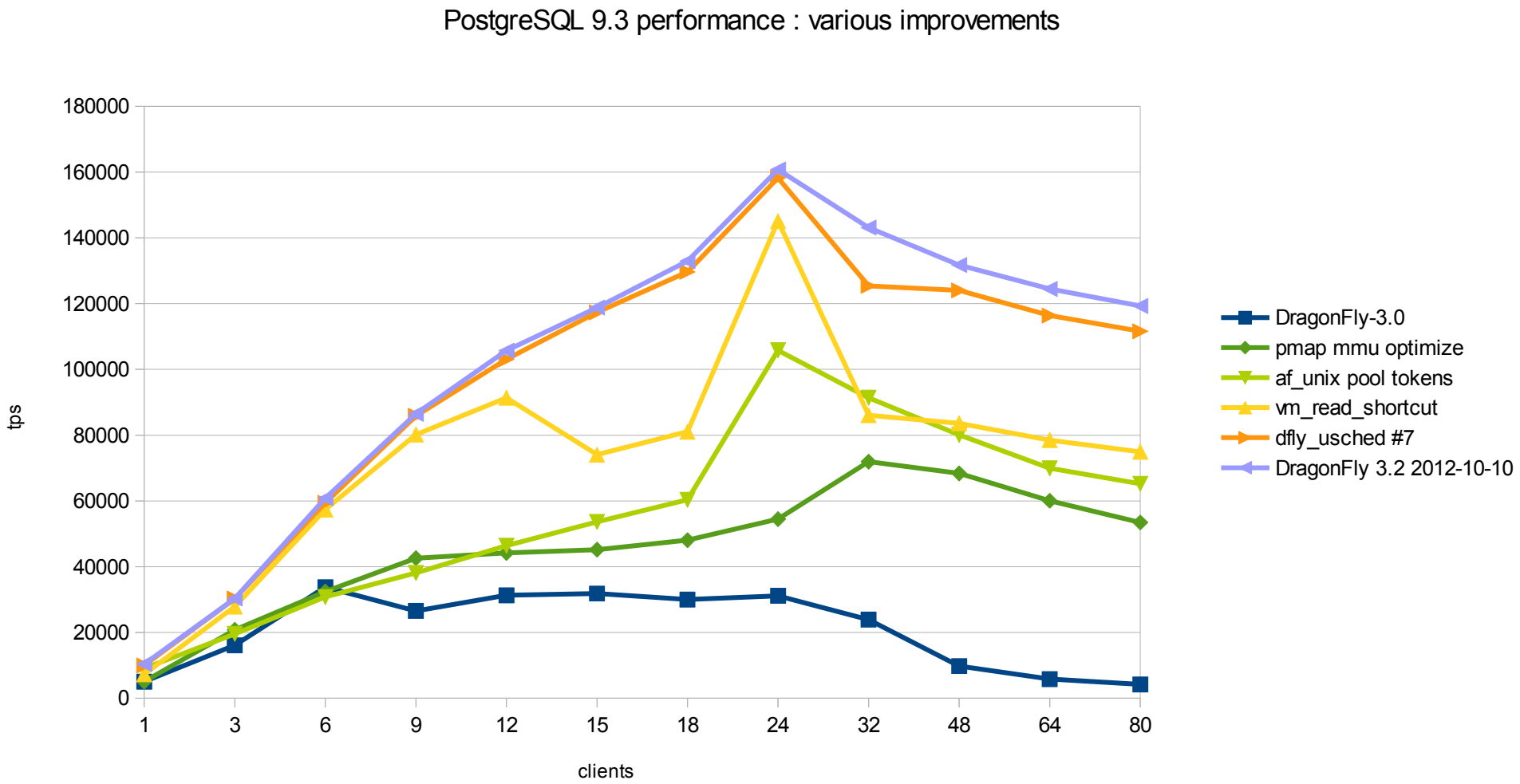
PostgreSQL 9.3 Pgbench Transactions Per Second, 2x Xeon X5650 / 24GB (24 threads), Unix socket												
Clients	1	3	6	9	12	15	18	24	32	48	64	80
DragonFly-3.0	5024	16077	33735	26530	31304	31825	30002	31132	23898	9769	5786	4193
pmap mmu optimize	5021	20775	32410	42580	44168	45187	48058	54461	71923	68360	60039	53435
af_unix pool tokens	9040	19535	30767	38092	46382	53612	60343	105798	91381	80088	69926	65207
vm_read_shortcut	7247	27854	57355	80185	91381	74050	81059	145103	86050	83601	78503	74943
dfly_usched #7	9942	30282	59416	85868	103060	117332	129681	158239	125404	124024	116369	111581
DragonFly 3.2 2012-10-10	10363	30333	60809	86492	105704	118805	132948	160813	143131	131742	124481	119276



Detailled TPS results excluding connections establishing

Clients (and threads)	1	3	6	9	12	15	18	24	32	48	64	80
Initial state												
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 3.1 improvements												
sysctl machdep.pmap_mmu_optimize=1	Commit 921c891ecf560602acfc7540df7a760f171e389[e]											
pmap mmu optimize	4,599	20,102	32,310	45,293	44,180	45,212	48,300	54,362	70,183	68,244	59,751	53,640
pmap mmu optimize	5,443	21,449	32,510	39,866	44,155	45,161	47,815	54,561	73,663	68,476	60,328	53,230
Commit 524d0e3c7c3041d91aaa8a6f0d1efa4623c0f413												
af_unix pool tokens	9,040	19,535	30,206	37,594	46,673	53,527	60,328	105,963	91,154	79,586	72,063	64,964
af_unix pool tokens			29,752	38,970	46,509	53,501	60,428	105,633	91,608	80,590	70,998	64,475
af_unix pool tokens			32,344	37,714	45,965	53,808	60,273				66,716	66,182
vm.read_shortcut_enable=1	Commit 68ad14555b583c75f3d3605fd3b5aee7328a8af0											
vm_read_shortcut	9,148	29,284	56,252	80,612	97,020	73,812	81,237	144,680	85,950	84,009	78,447	75,330
vm_read_shortcut	6,752	23,112	58,949	81,080	86,990	74,234	80,755	145,646	85,312	82,964	79,624	75,707
vm_read_shortcut	5,841	31,166	56,864	78,864	90,134	74,104	81,186	144,983	86,889	83,831	77,439	73,792
New scheduler, 7th tuning iteration.	Commit e3e6be1f3ada3078bf270c3a65637a84a95c4585											
dfly_usched #7	10,321	30,325	60,107	85,191	104,550	117,374	129,791	158,301	125,755	124,321	116,613	111,642
dfly_usched #7	10,314	29,919	59,597	86,267	103,702	117,669	129,632	158,587	125,367	123,984	116,405	111,607
dfly_usched #7	9,191	30,602	58,542	86,145	100,930	116,952	129,620	157,829	125,091	123,767	116,089	111,493
End result												
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

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

Software

PostgreSQL 9.3-devel, after the patch changing setup of the shared memory segment from SYSV shared memory to mmap

DragonFly from 3.0 to 3.2, including various stages of improvement in the 3.1 developement version

Goal:

Test scalability with pgbench, see if the use of mmap() doesn't cause performance problems

Improve performance if needed

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 ${clients} -T 600 -S bench > result_${clients}.txt
done
```

Repeat three times and average the individual results