

Re: Postgresql performance profiling

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2006-06/msg00005.html>

- *From:* Kris Kennaway <kris@xxxxxxxxxxxxxxxx>
 - *Date:* Sun, 11 Jun 2006 20:30:34 -0400
-

On Sun, Jun 11, 2006 at 04:31:44PM -0400, Kris Kennaway wrote:

On Sun, Jun 11, 2006 at 01:45:28PM -0400, Kris Kennaway wrote:

I set up supersmack against postgresql 8.1 from ports (default config) on a 12 CPU E4500. It scales and performs somewhat better than mysql on this machine (which is heavily limited by contention between threads in a process), but there are a number of obvious performance bottlenecks:

FYI, on a dual p4 + HTT, mysql significantly outperforms pgsq (by

55% peak performance, probably more if I was using libthr which I

cannot on this machine for technical reasons) on select-key.smack when configured the same way (i.e. transport over IPv4 instead of local socket, which supersmack prefers for mysql).

Contention is still a big issue here (only listing mutexes contended more than 10% of acquisitions):

```
0 0 142969 0 1996 14458 .101 kern/kern_synch.c:218 (Giant)
0 0 199028 0 11649 27944 .140 kern/kern_condvar.c:208 (selck)
0 0 400103 0 111216 91336 .228 kern/kern_sysctl.c:1317 (Giant)
0 0 303147 0 108735 131237 .432 i386/i386/trap.c:1005 (Giant)
```

I turned off process title setting and got an 8% performance boost.

Contention is now a bit better but still serious:

```
0 0 22952 0 2067 2521 .109 vm/vm_fault.c:987 (vm object)
0 0 199153 0 12589 31512 .158 kern/kern_condvar.c:208 (selck)
0 0 361305 0 124766 130901 .362 i386/i386/trap.c:1005 (Giant)
```

i.e. semop() (the Giant-locked syscall) is contending with itself a lot, and select() is a secondary problem.

Actually rwatson noticed that semop() is marked MPSAFE, so it's not

Re: Postgresql performance profiling

clear (but nevertheless true) why Giant is acquired here. OK, pjd worked out that it's because SYSCALL_MODULE_HELPER() *never* sets the mpsafe flag, so all such syscalls registered that way (i.e. those which are part of subsystems that may be loaded from kld) are Giant-locked regardless of what syscalls.master says.

I removed the SYSCALL_MODULE_HELPERS from sysv_sem.c but now postgresql hangs when trying to start; possibly the locking in sysv_sem.c is just broken since it was never in fact tested.

That was my mistake, the syscalls weren't getting registered. I made SYSCALL_MODULE_HELPER add the SYF_MPSAFE flag to work around it instead. The new mutex contention looks like:

```
0 0 199118 0 12134 30704 .154 kern/kern_condvar.c:208 (sellck)
0 0 354890 0 100749 110295 .310 kern/sysv_sem.c:1011 (semid)
```

i.e. semaphores are still contending with themselves. It didn't make any performance difference on this workload, as expected since it was only contending with itself and still is (but in mixed workloads with other Giant activity it will help, of course).

Kris

Attachment: pgpyYJ6iEok1e.pgp

Description: PGP signature