

## Re: Tuning for PostgreSQL Database

**Source:** <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2003-07/0090.html>

---

**From:** Terry Lambert ([tlambert2\\_at\\_mindspring.com](mailto:tlambert2_at_mindspring.com))

**Date:** 07/26/03

Date: Sat, 26 Jul 2003 00:18:15 -0700

To: Tom Samplonius <[tom@sdf.com](mailto:tom@sdf.com)>

Tom Samplonius wrote:

> *On Fri, 25 Jul 2003, Terry Lambert wrote:*

> > *Ideally, you would use memory mapped files for this, and not System V  
> > shared memory, so that the OS could implement swapping policies as it  
> > saw fit, and could actually swap the data, if neccessary, instead of  
> > it sucking up huge amounts of wired memory.*

>

> *PostgreSQL is from the good old days of RDMSES when they would  
> System V shared memory for everything, and store databases on raw devices  
> in an effort to utilize as little of the OS as possible, in effort to be  
> fast and reliable.*

>

> *But it does give PostgreSQL the advantage of working with large tables  
> and databases. Mmapping a file over 4GB in size would likely exhaust the  
> VM on a x86. Or, is it possible to map 4+GB with PAE?*

It's not possible. PAE only provides the ability to utilize a larger-than-4G amount of RAM consecutively in different processes.

It doesn't buy you the ability to have 4+GB System V shared memory segments, either. The limitation is based on pointer size in user space, which is, in turn, based on the size of the hardware registers.

PAE is basically good for allowing you to utilize large amounts of RAM for separate process instances and/or RAM disks (not even cache), and is most often used merely to run a lot of processes without getting bottlenecked by disk I/O doing swapping.

To get better than that, you need 64 bit pointers, which means a 64 bit architecture.

System V shared memory segments are basically \*subtracted\* from your available memory.

Memory mapped files don't benefit you (or hurt you) in terms of available address space for your process. But they \*are\* able to be swapped, and as long as you use `madvise()` and/or can live

freebsd-performance: Re: Tuning for PostGreSQL Database

with "LRU" as your paging policy, they \*will\* give you more physical RAM to play with for other things, if it gets down to it, than using System V shared memory will.

Maybe you could reimplement the code to allocate pageable RAM for shared memory segments; I don't really see how, without an almost total rewrite of the System V shared memory code, though.

-- Terry

---

freebsd-performance@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>

To unsubscribe, send any mail to "freebsd-performance-unsubscribe@freebsd.org"