

# Re: More threads

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2005-12/msg00012.html>

---

- *From:* David Xu <[davidxu@xxxxxxxxxxxxx](mailto:davidxu@xxxxxxxxxxxxx)>
  - *Date:* Tue, 06 Dec 2005 18:33:31 +0800
- 

Michael Vince wrote:

Hi All,

I been benchmarking a Java servlet I have created and I want to be able to handle a massive amount of simultaneous connections.

So far using benchmarking tools I have been able to get around 2,565 threads on the Tomcat 5.5 Java process (with native 1.4.2 Java) according to `ps -auxwH | grep -c 'java'`

But I just can't seem to get past that mark, I have a lot of memory currently the Tomcat is allocated 2gigs of memory.

When I max it out with my benchmarks I get this in catalina Tomcat 5.5 but there is still plenty of free memory to the Tomcat process.

```
SEVERE: Caught exception (java.lang.OutOfMemoryError: unable to create new native thread) executing org.apache.jk.common.SocketAcceptor@f36e59, terminatingthread
```

I have been testing with my libmap.conf with these different types of implementations.

```
[/usr/local/jdk1.4.2/]  
#libpthread.so          libc_r.so  
#libpthread.so.2       libc_r.so.6  
libpthread.so.2        libthr.so.2  
libpthread.so          libthr.so  
libc_r.so.6            libthr.so.2  
libc_r.so              libthr.so
```

I have been getting the best performance with libthr and have been using the above libthr and with these settings below.

I have set my max\_threads in /etc/sysctl.conf to a massive amount.

## Re: More threads

```
kern.threads.max_threads_per_proc=40000
kern.threads.max_groups_per_proc=40000
```

```
ps -auxwH | grep -c 'java'
2565
```

I am using Apache2.2 with the new built in AJP module which has been a great addition to Apache 2.2.  
I have been able to get the setup performing most stable (no 503 status errors) but with less performance / threads with the libmap.conf below.  
[/usr/local/jdk1.4.2/]  
libpthread.so libc\_r.so  
libpthread.so.2 libc\_r.so.6

I am on 6.0 Release i386, dual Intel P4, generic SMP kernel.

With Tomcat on Windows XP I have been able to get it running better.  
Does any one know of some other sysctls that increase the amount of threads in libthr which I am assuming utilizes the above sysctls. Does any one know how many threads can be created in Java on FreeBSD?

Cheers,  
Mike

Number of threads you can create if you use libthr is limited by following factors:

1) sysctl:

```
kern.threads.max_threads_per_proc
kern.threads.max_groups_per_proc
```

2) stack

per-thread userland stack, default number on 64 bits platform is 2M, on 32 bits platform, it is 1M. I don't know whether java supports adjusting default per-thread stack size. if can not, we may add an environment variable to thread libraries, for example:  
LIBPTHREAD\_THREAD\_STACKSIZE allows user to set default stack size.

## Re: More threads

Thread also needs a kernel mode stack when it is in kernel, if I am right, it is 16K bytes per-thread. if you create too many threads, make sure both side won't exhaust address space, and because kernel stack can not be swapped out when process is running, you'd make sure physical memory can not be exhausted.

3) check memory limits  
type limits command:

```
Resource limits (current):
cputime          infinity secs
filesize         infinity kB
datasize         524288 kB
stacksize        65536 kB
coredumpsize     infinity kB
memoryuse        infinity kB
memorylocked     infinity kB
maxprocesses     5547
openfiles        11095
sbsize           infinity bytes
vmemoryuse       infinity kB
```

if address space is not large enough, you have to reconfigure kernel to allow larger space. Fix me if I am wrong.

David Xu

---

freebsd-performance@xxxxxxxxxxxxx mailing list  
<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>  
To unsubscribe, send any mail to "freebsd-performance-unsubscribe@xxxxxxxxxxxxx"