

jail process limits

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/stable/2008-05/msg00417.html>

- *From:* Vivek Khera <vivek@xxxxxxxxxx>
 - *Date:* Thu, 22 May 2008 15:26:13 -0400
-

While we're on the topic of jail resource limits, I think I'll ask my question again... I asked last month but got no response...

I've got a jail server (FreeBSD 6.3/amd64) which runs a bunch of web site development environments. There is an apache or lighttpd running in each jail as user httpd (same UID on base system and each jail).

On the jail host, I counted 231 processes owned by httpd.

If I try to start an application server (or any process) as user httpd in one of the jails, it exits immediately with "Cannot fork: Resource temporarily unavailable". Even if I "su httpd" I get the same error on any command I try to run such as "ls". If I run the same on the jail host, it has no problems. The jail itself only has 34 processes running.

On the jail host, the following is logged:

```
Apr 22 16:34:38 staging kernel: maxproc limit exceeded by uid 80, please see tuning(7) and login.conf(5).
```

tuning(7) and login.conf(5) have pretty much nothing to say about "maxproc".

The sysctl settings are all default on this box.

```
kern.maxproc: 6164  
kern.maxprocperuid: 5547
```

The user httpd is of login class "daemon". My login.conf is unchanged from the distributed version, which states "unlimited" for max processes.

Why am I getting the resource unavailable when I barely have 230 processes, not even close to the limits.

Apache seems unaffected since the parent is run as root, so it can fork children willy-nilly and not be blocked by any limits.

Can anyone tell me where to look to find out what is limiting user httpd from creating new processes inside the jail, and what exactly that limit is? More importantly, how to increase it.

freebsd-stable@xxxxxxxxxxxxx mailing list

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

To unsubscribe, send any mail to "freebsd-stable-unsubscribe@xxxxxxxxxxxxx"