

Re: Relationship between load average and CPU busy or CPU idle

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.solaris/2005-09/0604.html>

From: Logan Shaw (lshaw-usenet_at_austin.rr.com)

Date: 09/09/05

Date: Fri, 09 Sep 2005 08:30:41 GMT

js wrote:

- > *Is there some kind of relationship between the load average figure and CPU*
- > *busy / idle percentage ?*

Yes, there's SOME relationship, but it's not a really simple one.

- > *Thus, what I am concluding so far is that a load average nearing N-CPU's will*
- > *have a very low CPU idle %.*

Every thread on the machine can either be runnable or not. For example, if you call "sleep(100);" in a thread, then for the next 100 seconds, the thread isn't runnable. Likewise if you're waiting on the network. But if your thread is in the middle of doing processing (actually on a processor) or in principle *could* be doing processing if a processor were available, then it's runnable. To make matters a tad bit more complicated, there is a third state on Unix systems, which is that your thread is blocked because of some short-term I/O, like reading from or writing to a disk. Because Unix expects disk I/O to finish really soon (at which point your thread will become runnable again), this is basically counted as "almost runnable".

So, the load average is computed by periodically looking at the state of all the threads and counting the number of ones that are running, runnable but not running, and "almost runnable" (in short-term disk wait). But, the load average is also a decaying average over a certain interval (1 minute, 5 minutes, or 15 minutes), so it doesn't necessarily reflect what the situation is at any given moment.

Meanwhile, idle time is computed by looking at what the processors are doing. The system keeps statistics about what percentage of the time the CPU is running, idle, etc. It does this by periodically waking up (via an interrupt) and gathering statistics about each CPU for a certain instant in time. It puts together lots of these samples to get a more accurate picture of how much time is spent doing what. But the stuff with short-term disk wait makes things complicated again. As I understand it, if there is ANY outstanding short-term disk wait (for ANY processor!),

then at the time that routine takes its sample, it counts ALL processors that are idle as in short-term disk wait instead. So, in a 4 processor system, one processor waiting for short-term disk I/O and 3 processors idle will be counted just the same as all if all 4 processors were really waiting for short-term disk I/O.

So, to recap:

- * load average means periodically sample the run queue and determine how many threads are running, runnable, or "almost runnable" (short term disk wait), then make a decaying average of this.
- * CPU idle/system/user means periodically sample all the CPUs and determine what percentage of the time they're running and not, but if a CPU is idle and it or ANY OTHER CPU has a short-term disk wait going on, count that as disk wait (non-idle).

So yes, your conclusion is basically correct. If you have a load average of N, and if you have N processors, you are going to have a low idle percentage. This assumes you don't have a processor set defined that makes things more complicated: you could have 1000 CPU-bound threads that do no I/O at all, but using a processor set, put all 1000 of them on a single processor in a 4 processor system, and run nothing else so that the other 3 processors are idle. Then your load average would be 1000, but your idle percentage would be 75%.

One more note: the "any other CPU" thing means something a bit funny for a multiprocessor system that does I/O. If you have one single thread that is doing nothing but disk I/O on a 4 processor system, you might expect to see 75% idle, but you won't. You'll see more like 0% idle and 100% iowait. That's because one of the CPUs is basically perpetually in iowait, and though the others are perpetually idle, they are all getting counted as in iowait by the routine that collects the statistics. I guess if you didn't know this, you could get an inaccurate sense of the I/O load on a multiprocessor system. (A better measure would probably be whether I/O service times tend to increase as the server's load grows.)

By the way, I've based a lot of the above on the Solaris Internals book and on <http://sunsite.uakom.sk/sunworldonline/swol-08-1997/swol-08-insidesolaris.html> . Both of these are several years old, so it's possible something has changed in more recent versions of Solaris. (It's also possible I've totally misunderstood everything...)

- Logan