Re: dump(8) performance

Source: http://unix.derkeiler.com/Mailing−Lists/FreeBSD/hackers/2006−05/msg00375.html

- *From*: Peter Jeremy <peterjeremy@xxxxxxxxxxxxxxx>
- *Date*: Thu, 1 Jun 2006 05:22:51 +1000

On Wed, 2006−May−31 08:05:21 −0700, Eugene M. Kim wrote:

> While watching the output of iostat −dxz −w10 −n100 to monitor the
> progress/performance of a dump(8) process straight to a tape, I found
> out something interesting and disappointing at the same time: The disk
> read throughput was exactly twice as high as the tape write throughput,
> throughout the entire dump phases 4 and 5, i.e. dumping actual inodes.

I looked into dump performance many years ago (as part of a thread which resulted in Matt Dillon adding the existing caching code) but can't find my notes right now. From memory, the main problems were that dump would re−read the inode multiple times and the physical read sizes were (pretty much) limited to the filesystem blocksize. The caching code increases the read blocksize but this also means that data read from the disk may not be needed. Dumping small files is the worst case.

If you remove the '−C' parameter, you'll probably find that your disk throughput drops to only slightly more than the tape throughput, though the tape utilisation will probably drop further.

If your concern is tape drive utilisation (because you want it for other tasks) or the tape drive dropping out of streaming mode, your only real solution is staging to disk. I wrote a tool similar to ports/misc/buffer that supported hysteresis to minimise the number of start/stop cycles but it only marginally speeds up the total dump time (sometimes dump runs faster than the tape and so a buffer helps here).

There probably is more scope for enhancing dump throughput.

−−
Peter Jeremy