Notes on Diskfull and Diskless Performance

Dan Walsh

Diskless is not Cost-Effective

There are many advantages to Sun customers that come from running diskless workstations. Among these are ease of administration, easy sharing of user files, and quiet operation in an office environment. But lowest cost/seat is not among then. A glance at the current Sun price list illustrates this. A Sun-3/52 lists at $13.9K. A Sun-3/50 lists at $7.9K, but you need a Sun-3/180 server ($21.9K), two Eagle disks ($19.9K) and a 6250 BPI tape drive ($16.9K) to support ten clients. This gives a diskless cost/seat of $7.9K + $78.6K/10 = $15.76K.

This is not just an anomoly brought on by current pricing. The reason is the the cost per megabyte of disk is about the same for small and large disks. So the diskless user has to pay not only for the disk space, but also for some proportion of a server. It seems that diskless can never be cheaper than diskfull.

What is Disk Space Used For?

Using my own server, speed, as a guideline, it seems that the average diskless machine uses 5 Mbytes for root, 20 Mbytes for swap and 25 Mbytes for user files, for a total of about 50 Mbytes. In addition, there are 50 Mbytes used for the system utilities for both 68010 and 68020 clients, and for speed's own root partition. From this we see that the amount of disk space a server needs is 50 Mbytes + (50 Mbytes * number of clients).

The space for just the 68020 system utilities is about 42 Mbytes, so a diskfull 3/52 would need 42 + 50 = 92 Mbytes of (formatted) disk space to hold the equivalent amount of data. This illustrates why the 71 Mbyte disks Sun currently ships are inadequate.

Dynamically, what do Sun workstations access disk storage for? In the CASE environment in engineering, most of the traffic is paging and swapping and reading system utilities. Surprisingly, accessing user files is only a small proportion of the traffic. For example, on speed, the Eagle with the paging, swapping and system utilities gets 90% of all the disk requests, while the Eagle with all the user files gets only 10%. I don't know if this is generally true or not.

How to Make Diskless Cheaper

The key to making diskless cheaper is to serve more clients from a server, thus driving down the cost/seat of the server. There are two approaches to this: (1) modifying clients to put less demands on the server and (2) modifying servers to handle requests more efficiently.

Client Modifications

The key to reducing client requests is to provide the client machines with more memory or a local disk.

More Memory

More memory on client machines would reduce paging and swapping traffic to nothing. Also, frequently used system utilities would remain in client memory given the current code in the kernel. No software changes would be necessary to implement this.

Let's look at cost. Sun charges $1K/Meg or more for memory, so if we assume another 4 Meg of memory would cost at least $4K/seat. Let us assume this would reduce the server load by a factor of 4, which is a wild guess that may be unrealistic. A glance at Table 1 shows that the result would be a cost/seat of $15.86K. Slightly more expensive than today.

If/when memory gets cheaper, which means at least a couple of years from now, this might be effective. However, there is also the problem of the psychological resistance in the market to larger memories, which the sales force says makes Sun look bad (the software is such a pig it needs more memory than competitors' software does).

	No. Clients	No. Disks	Server Cost	Server cost per seat	Client Cost	Client cost per seat	Software effort
Current	10	1.49	78.60	7.86	7.90	15.76	
8 meg client	40	5.54	158.20	3.95	11.90	15.86	none
50 meg disk	100	8.24	217.90	2.18	12.40	14.58	modest
Faster server	20	2.84	98.50	4.92	7.90	12.82	large
Diskfull					13.90	13.90	
Best case				2.69	7.90	10.59	

Table 1. Cost/seat under various assumptions.

Adding a Disk to Client Machines

Suppose, instead, we add a local disk to the client machine. Let's assume we can get by with only 50 Meg of disk instead of the current 71 Meg, and that we can delete the ¼ inch tape since no backups are needed. Let's assume that as a result the cost drops by $1.5K to $4.5K for the disk. If we use the disk for all paging and swapping, and also modify NFS to use the disk as a cache for system utilities, we might be able to drop the load on the server by a factor of 10. This is a fairly wild guess that would have to be substantiated with hard evidence. Also, since the client would have its own swap area, the server would only need to have 30 Mbytes of disk for each client. We see from table 1 that the cost/seat drops to $14.58K. An improvement over today, but not a major one.

Making the Server Faster

Each server needs enough disk to support the clients' data, and a tape drive for backups. The key to improving server efficienty is reducing the CPU time to handle requests. It is clear that this can be done. The current implementation of the NFS code is layered into many levels for ease of implementation and maintenance. There has been a significant effort to make it efficient, but much more could be done. It is hard to predict exactly how much improvement would result from how much effort, but I think it reasonable to suppose that a couple of man-years invested might lower the CPU time to serve a request by half. This is only a guess, and would need to be refined if it is decided that such a project might be worthwhile. We see in table 1 that the result would be a cost/seat of $12.82K, which is actually better than diskfull. The reason is that sharing the system utilities has finally paid off enough to outweigh the cost of the server. This advantage could be reduced if the utilities are made smaller by shared libraries. It could also be reduced if a server supporting 20 clients needed more memory to cache the utilities, so that it could serve them to the clients without bottlenecking the disk they are stored on.

Raw Disk Speed

(*The following sections are on related, though distinct, topics.*) Since Sun workstations do lots of paging and loading of system utilities, the ability to read large amounts of data into and out of memory quickly is important. An Eagle disk, with a nominal bit rate of 15 Mbits/sec., actually presents to the CPU through the controller 1.52 Megabytes per second. Since we currently use 8K blocks, we calculate that the Eagle can present an 8K block every 5.26 milliseconds. It takes 6 msec. of software overhead to process a disk block (running the controller, handling the interrupt, etc.) and 2 msec. of DVMA time for the data to come into main memory, for a total of 8 msec. of overhead. Using one out of two blocks, the best we can do, gives us a file system throughput of 77.5 Kbytes per second. In fact, we observe only about 720 Kbytes per second due to occasional seeks and other random factors.

The thing to notice here is that any disk with a slower bit-rate is going to slow down performance. For example, the 5¼ in. disks we currently ship have a nominal bit rate of 5 Mbits/sec., and actually present the CPU with 522 Kbytes/sec. Since the controller can't react to back-to-back requests, the CPU is forced to interleave the 8K blocks, for a theoretical maximum rate of 261 Kbytes/sec. Actual measurements show about 240 Kbytes/sec. due to occasional seeking and other random factors. Since a Carrera client driving a Carrera NFS server with an Eagle can get about the same, a local 5¼ inch disk does not improve performance if the server is not busy doing anything else. The key point here is that the 5 Mbit/second ST-506 drives are already a performance bottleneck with Carrera class machines, which can easily handle 15 Mbit/second disk drives.

When Sirius and Sunrise come along, they will have less DVMA and CPU overhead than Carrera class machines, so they will want greater bit-rates from their disks. It would be a serious mistake to plan on using 5 Mbit/sec. disks on these machines since these disks are already a bottleneck on the slower machines we have today.

Heavily Loaded Servers

Recently some people in Steve Saperstein's organization tried to run some benchmakrs for Lawrence Livermore Laboratories. They concluded that a Sun-3 server could not support more than 3 Sun-3 clients. There are some problems with this conclusion. One is that the load they were presenting to the server was extremely heavy, and not at all typical of what we usually see. For example, in engineering we support eight to ten Sun-3 clients per Sun-3 server without any problems of this sort. The larger problem is that Sun-3 server machines do not degrade gracefully under load. The reason is that we have never studied this aspect of performance. If this is a real concern for Sun, some resources need to be devoted to understanding the effects of heavy load on Sun server machines, and modifying the software to degrade gracefully in these cases.

Need for More Study

Performance has never had a high priority at Sun, with the result that little is known about many of these issues. This memo is filled with many guesses and little hard data. If performance is deemed important to the future of the company, more resources need to be devoted to it.