Brent

Learning from Andrew

A Low Cost Per Seat Strategy for Sun

David S. H. Rosenthal
James A. Gosling

ABSTRACT

We examine the potential market for low-cost Sun workstations, distinguishing between the requirements of small and large installations. We make the business case for concentrating on the larger installations, and examine Sun's current strategy for doing so, contrasting it with that developed by the C-MU/IBM Andrew project. Sun's strategy is no longer cost-effective, and we examine two ways of correcting this; a hardware strategy based on reducing server cost, and a coordinated software/hardware strategy based on the Andrew experience. The second offers not merely low cost per seat, but the possibility of using Sun workstations over the phone, and advantages in system adminstration.

1. Introduction

We examine the potential market for low-cost Sun workstations, distinguishing between the requirements of small and large installations. We make the business case for concentrating on the larger corporate or University installations. Examining Sun's current strategy for doing so, and contrasting it with that developed by the C-MU/IBM Andrew project, we establish that diskless workstations and NFS are no longer cost-effective. It is almost certainly cheaper to equip each machine with a shoebox than to use servers. This has two effects:

- Sun's lowest effective cost per seat is too high to be competitive.
- System administration of large configurations of machines each with a local file system is a nightmare.

We discuss two ways of correcting this; a hardware strategy based on reducing server cost, and a coordinated software/hardware strategy based on the Andrew experience. The second offers not merely low cost per seat, but the possibility of using Sun workstations over the phone, and advantages in system adminstration.

2. The Markets for Low-Cost Suns

We can distinguish two very different markets for low-cost Sun workstations:

- Small installations, which buy in small quantities (<5) typically from local dealers,
- Large installations, which buy in substantial quatities (>5) typically direct from Sun.

2.1. Small Installations

The small customers do not currently buy from Sun. They require an entirely new distribution mechanism, based on developing a local dealer network. The costs of developing this will be considerable. We will be competing with IBM and DEC in areas where they are unassailable; being able to spread these costs across a much wider product line.

Further, existing Sun hardware and software is completely unsuitable for this distribution channel. The reason why Sun can offer much better hardware price/performance than IBM has a lot to do with IBM's engineering for reliability and servicability as against Sun's engineering for raw performance. Sun software is complex, feature-laden, and space-intensive; quite unsuitable for a market in which the difference between a 40M and a 70M disk is a critical price point.

2.2. Larger Installations

The larger customers are already in many cases Sun sites, and can be sold machines through our existing distribution channels. The unique feature of these sites is their stress on networking, an area in which Sun has a major technological lead over IBM. The fact that the RT was announced with no network support is not accidental, it is a result of IBM's corporate commitment to SNA and token-ring LANs. An IBM machines network support will either be:

- Inefficient, because it has to be based on SNA.
- Late, because it had to fight long corporate battles to avoid being based on SNA.

The problems outlined above are not so serious in this market. The fragility of Sun hardware is not a major problem for a large site, spare machines will normally be available to cover for failures. The size and complexity of the software are less of a problem too, since it will normally reside on a file server, and experts will normally be available in the user community.

In all respects the larger installation market seems a better match for Sun's strengths.

3. Current Sun Low Cost/Seat Strategy

Sun's current strategy for providing the lowest cost/seat is diskless workstations supported over a high bandwidth network by one or more servers. The software is NFS†, which provides each workstation with a small (say 400Kbyte) evanescent‡ cache of file blocks.

This strategy presents four problems:

- A proportion of the cost of the server must be added to the cost of the diskless workstation to arrive at the final cost/seat. The decline in the cost of the basic diskless node has been much faster than the decline in the cost of the server. The greater CPU performance of the Sun/3 means that fewer clients can be supported by a single server. These factors mean that the contribution to the server is now a major part of the cost/seat.
- The diskless workstation is useless when disconnected from the high bandwidth net. It is difficult to persuade students, for example, to buy a machine they cannot use off-campus. A machine which could be used at home, even with some sacrifice in performance, would have a major selling edge.
- The small size of the disk block cache, its evanescent nature, and its block nature, all result in high transaction rates at the server, reducing the client/server ratio.
- Low client/server ratios mean that commonly used files are replicated many times, eating up disk space. We estimate that Sun, for example, currently devotes almost an entire Eagle to Emacs alone.

Based on Sun's current list prices, we can compute the cost/seat in configurations of varying numbers of seats for various client/server ratios, and thus the proportion of the cost of a seat represented by file support. Averaging these over the range of configurations having 9 to 250 seats, we generate the following table, showing that even for the wildly optimistic case of 20 clients/server, the file support is a quarter of the cost of a seat.

---

† Projecting a little into the future.

‡ The cache is evanescent in the sense that it is flushed when the machine is re-booted.

NFS averaged over 9 to 250 seats		
Clients/server	Cost/seat	Files as % cost
0	12800	38
3	22923	61
8	13533	41
10	12407	36
15	10988	28
20	10286	23

Note that the experience at Lawrence Livermore shows that the current client/server ratio for Sun/3 machines may be only 3. Sun claims 8. For disklessness to be cost-effective, the ratio must be at least 9.

A further problem is that the current strategy is encouraging large configurations of diskfull machines. Administering the large number of independent file systems this involves is a nightmare. While NFS makes an excellent job of making a file system abstraction visible to the network it provides few if any tools for managing large numbers of independent file systems to form a coherent and consistent large-scale network file name space.

4. Andrew Low Cost/Seat Strategy

The VICE‡ file system was designed at C-MU with a target of supporting a very large total number of clients (>5000). The plan was for the workstations to be purchased by the students, and the network infrastructure including the servers by the University. This resulted in a subsiduary goal of a client/server ratio of at least 100, less than this led to unacceptable economics.

The VICE file system is currently in production with a client/server ratio of about 60. The clients are Sun/2, RT PC and MicroVAX II workstations; the servers are 4M Sun/2s and VAX/750s with Eagles. Profiling the server code suggests that the system will saturate at a ratio somewhere between 100 and 150. Assuming that we could achieve similar ratios, we can repeat the cost/seat computations:

VICE averaged over 9 to 250 seats		
Clients/server	Cost/seat	Files as % cost
50	9054	12
100	8624	8
150	8575	7

These ratios are achieved by a file system devoted to reducing the transaction load on the servers as much as possible. It uses about 20M of local disk for:

- A small (<1M) root file system.
- Swap space (~8M).
- A large (~10M) persistent cache of files.

The VICE caching strategy achieves the goal of reducing the transaction load on the server in two ways:

- The server is involved only in file `open()` (if there is a cache miss), `close()` (if the file was written), and directory write operations. These are infrequent compared with I/O operations.
- If a cached file is written, the server is notified on `close()`. It then notifies all other workstations holding cached copies of the file that they are invalid. Although this is expensive, it is infrequent and the number of notifications per close is small. The result is that a workstation can open a file in its cache without involving the server.

‡ For detailed information on the VICE file system see: <XXX> Comm. ACM (March 1986).

The load is so low that the system has been used experimentally over a 9600-baud asynchronous line. Although the experiment was not wholly successful†, we believe that a VICE workstation would remain usable while temporarily disconnected from the high-bandwidth LAN if:

- It was connected to a server by a 1200-baud or faster link.
- The usage pattern of the system remained fairly constant. Cache misses for medium and large files would be very slow.

The VICE strategy of caching whole files can be viewed both:

- as a means for dynamically tailoring a stripped-down UNIX† to the user's needs without requiring user intervention, and
- as a means for eliminating local system administration entirely.

On the other hand, it has problems with large files. The largest file that can be accessed is the size of the file cache, and the delay while the whole of a large file is fetched can be excessive.

VICE simplifies the administration of a large-scale network file system by allowing multiple servers to cooperate to provide clients with the illusion of one single enormous UNIX file system. Administrative operations on the file system, such as the allocation of files to servers, replication of files between servers, backup and archiving of files, and so on are invisible to the clients. No Andrew client (or user) need ever know the name of any file server.

5. New Low Cost/Seat Strategies

The per-seat cost of file system support can be reduced in two ways, by reducing the cost of the servers or by spreading the cost over a larger number of seats.

The approach of reducing the cost of the servers has already been suggested. The idea is to build special-purpose low-cost servers; in effect a shoe-box presenting an Ethernet rather than a SCSI interface. Assuming a target 10% file support per seat cost, the relationship between the client/server raio and the cost of this box is:

Cost Target for Cheap Server	
Client/Server	Cost
3	2370
6	4740
9	7110
12	9480

It would seem difficult to achieve a high enough client/server ratio to make the box cost-effective, and the strategy fails to address either:

- the system administration issues of having a large number of independent file servers, or
- the costs of replicating shared files among the large number of small servers, or
- the importance of remote use for low-cost machines.

Based on the results of the Andrew experiements, we can state a set of goals for an alternative low cost/seat strategy:

- A client/server ratio of at least 100.
- Workstations able to operate with 20M of disk.
- Workstations able to operate while connected to their server by a phone line.

---

† It took place before the VICE file cache had been made persistent. The file system was, therefore, operating entirely in a low cache-hit-rate mode as the cache filled. The delays while essential large files such as the window manager were transferred made the system unusable.

† UNIX is a trademark of Bell Laboratories.

Achieving these goals calls for coordinated developments in both software and hardware.

5.1. Software

The software developments required are a combination of the Virtual Memory re-write and a revised NFS using a large persistent cache. Contrasting this approach with existing NFS and VICE we have:

Comparison of three strategies				
Name	Cache size	Cache in	Persistent	Cache is
NFS	400K	RAM	No	blocks
VICE	10M	disk	Yes	files
NFS/$	20M	RAM/disk	Yes	blocks

The virtual memory re-write allows us to manage the whole of the local storage, both RAM and disk, as a cache of disk blocks. By pushing copies of disk blocks fetched from the server to the local disk we achieve both local file caching and the UNIX swap mechanism. The server will be involved only in:

- Cache misses. Given the size of the cache these will be infrequent.
- Directory operations.
- Writes to shared files‡. VICE shows that these are relatively infrequent, and can be batched until the file is closed.

The servers would cache a description of the set of writable files for which a workstation had at least some blocks cached. When another workstation closed a shared file it had written to, the server would be notified and would send messages to all other clients holding blocks from this file to invalidate their cache entries. While this operation is expensive for the server, VICE experience shows that it is relatively infrequent and that the number of notifications per close is small.

Although NFS/$ operates at a block level, the larger cache and better integration into the system should give it performance advantages over VICE. Similar client/server ratios should be achievable, though detailed simulation work would be required to support these claims.

5.2. Hardware

The hardware required for this environment would resemble a current 3/50 with a small (3.5"?) internal hard disk and a smaller monitor, to make taking it home easier. It would also require an internal modem and auto-dialler, to permit the file system daemon to contact the mounted file servers as required.

In the future, this strategy can effectively use a number of evolving hardware technologies:

- It seems possible to build large, low-power, slow RAMs. With battery back-up, these would make an ideal replacement for the disk.
- If CD-ROMs become very cheap, they can be used to pre-load the local cache with a large amount of fairly recent information from the file system. At start-up the client tells the server the contents of the cache, and if any are out-of-date the server invalidates them. Thus the read-only nature of the CD is not a problem, files that have been overwritten will be invisibly fetched from the file server. Over time, this traffic will increase to the point at which a new CD will be desirable, but the performance degradation will be gradual.
- When the local phone net goes digital and 56Kb service is available from the RJ-11 jack in homes, a local cache-based file system for workstations will be really effective. The available band-width will be capable of providing very good performance at typical cache miss rates.

‡ /tmp can be special-cased out to avoid the cost of sharing uninteresting writable files.

6. Conclusion

Sun's current low cost per seat strategy has been invalidated by cost and performance trends.

A hardware-only strategy based on reducing server cost offers some improvement in the short term. As CPU performance rises, it can only survive by using higher band-width LANs, such a optical fibers. Sites that have invested the immense sums needed to wire large sites (C-MU's campus wiring is a $10M project) will not easily be persuaded to pull new wire.

A coordinated hardware and software development effort requiring no untried technology can produce an innovative product offering medium and large installations both low cost/seat, and the convenience of being able to use the same environment both at home and in the office.