

(This file is pumpkinseed:~blyon/dfs/memos/meeting.840125.)

Sun Distributed File Service Meeting Notes

Date: Jan 25, 1984

Attendees:

wnj (Bill Joy),
schmidt (Eric S Schmidt)
pugs (Tom Lyon)
shannon (Bill Shannon)
dan (Dan Walsh)
blyon (Bob Lyon)
dg (David Goldberg)
rusty (Russel Sandberg)
srk (Steve Kleiman)
(Jskud (Joseph P. Skudlarek) was not able to attend)

Administrative

1. blyon was appointed secretary, and assuming that he can write will continue in this position.
2. The next meeting of the dfs group will be Wed, Feb 1, 1984; time is not currently known.

Minutes

1. wnj stated what Sun's goals / philosophy are in the networking area in general. They are
 - Sun wants to build network services; it does NOT wish to attempt to build a distributed operating system.
 - Since it is Unix that sells Suns, we must provide good bridges from the Sun network services (back) to Unix.
 - As with any good package, Sun network services and their interfaces must be fast.
2. wnj then stated the goals / philosophy / tensions concerning the Sun dfs, in particular. They are
 - (goal) It should be possible to build networks that only consist of diskless workstations and file services; however, the dfs should not exclude the use of private disks.
 - (performance) The diskless Sun workstation should have a through-put of about 400 to 500 K-bytes per second (3.6 - 4.0 Mbit/sec). (pugs pointed out that these rates are faster than what the (local) Eagle disks deliver in release 1.0. blyon pointed out that ND's best rate was about 100 Kbytes / sec.)
 - (administrative) It must be easy to add (change, delete) a new (old) [workstation | user | file server] to the system.
 - (UNIX transparency) *Most* Unix users should not have to know that they are accessing ->multiple<- file servers during their normal activities.
 - *Most* Unix based programs should work either on a normal-Unix system or on a diskless-Unix system.
3. wnj presented an overview of his proposed dfs. (The proposal is documented in a set of 3 memos (1)). In short, fs inodes should evolve into vnodes - vnodes clearly separate the concrete implementation (of inodes) from the inode abstraction. The separation allows implementation of virtual file systems, one of which turns inode operations into network level messages (or remote procedure calls). Another virtual file

system would replace the current inode implementation.

Sub-proposals and clarifications (all kernel level) presented were:

- Consistency at the vnode level is achieved via introduction of file version numbers (time stamps???) and the following actions:
 - a) a client Close causes the server to flush all dirty pages of the file and update the file's version number.
 - b) a client Open causes the client machine to verify the client cache by comparing the cached file version number with the (true) file server's file version number.
- (User level file consistency (for example, preventing two users from modifying the same file concurrently) is outside the scope of the dfs. This group of functions can be handled by a co-resident file-consistency-service. This service is NOT deemed important enough for inclusion in the first release of the dfs.)
- Client machines keep caches of interior nodes of the fs. The caches map [directory-handle, sub-name] to (sub)directory-handle. This does not introduce any new problems; it merely compounds an existing problem.
- The client (diskless) machine does not keep a superblock. Since superblocks are owned by the implementation of a file system, the networked fs does not need one.
- There is one root directory per file server, NOT per client machine as in nd.
- File access control would be implemented via Unix user and group ids. This implies that Unix uids and gids are constant on all file servers and all fs client machines.
- Various devices on remote machines are accessed via other network services and not via entries in a /dev. Examples of these devices and services are tape drives and modems (and disks, of course).

4. Various objections and concerns were voiced concerning item 3.

- dg pointed out the administrative disaster that would occur if two (previously) disjoint internets were plugged together. This is due to the use of (flat) uid & gids on the network.
- A discussion between shannon (on the freedom of choice side) and wnj (on the freedom from choice side) developed over the implication of only one "/" per fs (rather than one per client machine). In particular, the workings of /dev become quite murky. wnj stated that *most* client workstations could use the dfs-defaulted /dev and that the users do not want to be bothered with /dev. shannon pointed out that there will always be special devices (or private disks) on some machines that also wish to access the dfs.
- Other problems with the "/" were pointed out and an action item was made to investigate them (see #1 under Action Items).
(The was general agreement that one "/" per fs was the right thing to do, although accomplishing it may be very painful.)
It appeared that a lot of the problems could be solved by "mount" commands in the rc file.

5. one more hard issue was raised.

- The relationship between machine and user is not well understood in the Unix based networked workstation environment. What does login and logout really mean on a workstation. Should they be replaced by or augmented by checkin and checkout? What implications does this have on the future compatibilities of ws Unix and ts Unix?
pugs pointed out that ws Unix and ts Unix could continue to be compatible if each user process was a virtual Unix machine. He said

a relatively easy modification to the kernel would be necessary. The mod involves moving global kernel data into each process record.

Action Items

1) wnj, rusty and srk will provide a list which enumerates the problems associated with only one / per fs. (They currently do not have an action item to suggest solutions to these problems.) A partial list is as follows:

- /
- /dev
- /tmp
- /etc/rc
- setuid
- stat
- su
- ioctl
- swap space
- login
- crontab

Action Items Not Taken

1. Investigation / resolution of the user - machine relationship. Does each user have a "home" machine or can she log into any idle workstation and have the right things happen?

2. Investigation of a new "mount" command. Is it possible to mount directories instead of file systems? Who may do this? Are there considerable security holes?

3. Are there (will there be) any other serious proposals to the dfs? Is so, they should at be identified ASAP.

Footnotes:

(1) All by Bill Joy: "Design of the Sun Network File System", "Sun UNIX Modifications to use the Sun Network File Service", and "Sun Network File Protocol Design Considerations".