Distributed Operating Systems lecture 2

Process migration

Stateful and Stateless - 17.4

NFS - 17.6.2

Process migration

What do we need?

- location transparency of
  - processes
  - resources used by the process

What defines a process?

- PCB
- Resources
  - files
  - communication channels
  - memory
  - devices - including windows, keyboard, mouse
- Threads

References to resources within the program

- must not be associated with the machine the process is running on

References to the process from outside

other processes communicating with it

Minimise the amount of down time

Process must be stopped at some stage

Stop, copy, notify

How much do we copy?

Only the working set

get the remaining pages by demand paging

Everything, but don't stop the process

then copy pages which were dirtied during the copy

Both approaches only stop the process while the working set is moving.

In reality process migration is not used for load balancing.

- It is too expensive.
- Most processes only run for a few seconds.
- Transferring a process can easily take a few seconds.

It is still useful when a machine needs to be closed down for maintenance and it has running processes which we don't want to kill.

This may require migration transparency of other resources as well.

Common use - idle workstations

Move processes when no longer idle.

Generally load balancing is only done when a process starts

or when it has to move.

Where is the best place to run this process?

How can we implement location transparency?

location servers

centralised or on every machine

(usually files are clustered because there are so many of them)

forwarding addresses when things move

while location servers are being updated

Distributed File Systems - ch 17.6 several different examples

Stateful

Server knows who has the file open for what type of access and where it is in the file etc.

When the client calls open it receives an identifier to be used to access the file.

Looks very similar to traditional local file access to the requesting processor.

More efficient, the needed data may be read ahead by the server. Information about the file is certainly held in memory.

If the server crashes - it is difficult to start again since all the state information is lost.

Server has problems with processes which die - needs to occasionally check.

Stateless

Server has no idea.

Client doesn't call open.

Requesting processor has to pass all the extra information, such as the current file location the process is reading from.

Server doesn't have to worry about processes stopping, it doesn't keep any records and is not taking up any memory space on the server.

NFS - 17.6.2

Remote directories (or entire file systems) can be mounted anywhere in the local directory tree

(limitations of access rights of course)

Works with heterogeneous environment

RPC and XDR - external data representation

Mount protocol

Mount servers on each machine

export table /etc/exports

- The full pathname of the directory to be exported
- The client machines that will have access to the exported directory
- Any access restrictions

Request comes to mount a directory from this machine

returns a file handle to this directory (file system:inode)

Client maintains a list of the directories which are mounted from other systems.

Automounter mounts and unmounts remote directories on demand.

Uses *maps* (files containing links between the mount point and the actual directory)

e.g. Setting up a shared namespace for /home

First an entry is made in auto_master (master configuration file) which associates the mount point /home to a map called auto_home:

```
*/home  auto_home
```

auto_home is a map that associates user names to home directories on their respective servers:

```
*sally   server1:/export/home/sally
*greg   server1:/export/home/greg
*tom   server2:/export/home/tom
*grace   server3:/export/home/grace
```

When the automount command is invoked at system start up time it looks in auto_master and then auto_home and knows to set up /home as directory of mount points. These mount points will become mounted file systems at the time they are referenced by users. Users can be added or deleted from the namespace by adding or subtracting them from the auto_home map. Any changes will be automatically implemented the next time the file system is mounted.

NFS protocol

Clients and servers are identical

normal system call

VFS (virtual file system) determines local or remote

NFS service layer makes the RPC to the remote machine

request gets pumped into VFS on the remote machine

carried out locally

then back goes the result

Support for Replica Servers

Replica servers are servers with duplicate file systems that can be substituted in the event that the server in use becomes unavailable. NFS currently supports replication of read-only file systems. A map entry for a read-only file system may describe several locations for alternative replica servers. At mount time, the automounter mounts the file system on the server that has the nearest proximity to the client. This reduces network routing delays and helps keep NFS traffic off the corporate backbone.