Distributed Systems, continued

Daniel Andresen

CIS520 – Operating Systems

RPC Failure Semantics

What happens if the server crashes? The client stub should:

- *Use UNIX Semantics*: Hang forever waiting for a reply that will never come (e.g., put the burden on the application programmer).
- *Time out and raise an exception* or report failure to the client.
- *Time out and retransmit* the request - only satisfactory if the operation is **idempotent** - it does not matter how many times it is executed by the server.

Semantics for duplicate requests:

- *Exactly once* - unachievable due to server crash. - At most once -the operation has been performed either zero or one times.
- *At least once* - the client stub tries over and over until it gets a proper reply (e.g., this is only ok for idempotent operations). The client stub may use the last-of –many replies by attaching a transaction id to each request.

What happens if the client crashes?

A running server with no waiting client is called an **orphan**.

Orphans may cause problems by:

- using up cpu cycles,
- holding locks on files, or
- sending results that cause confusion when a client machine is rebooted.

Ways of dealing with orphans:

- *Extermination*: when a machine recovers, it checks if it had any RPC in progress. If so, it asks the server to kill any process running on its behalf. Extermination is performed recursively.
- *Expiration*: the server is given a fixed amount of time to complete a call. When a client machine recovers, any client must wait until all orphans have expired before submitting new requests.
- *Reincarnation*: kill all remote activity in the network, use epochs. Reincarnate only if the client cannot be located; that is, only orphans whose clients cannot be found are killed.

Types of Dist. OS support: Network Operating Systems

- users are aware of multiplicity of machines.
- Access to resources of various machines is done explicitly by remote logging (telnet or rlogin) into the appropriate remote machine.
- Networking software creates two-way link.
- Process on remote machine handles interactions – acts as proxy for user.
- Access file on remote machine
- Compute just as any local user.
- Transferring data from remote machines to local machines, via the File Transfer Protocol (FTP) mechanism.
- Each computer has its own local file system.
- No real file sharing

Distributed Operating Systems

- Users are not aware of multiplicity of machines.
- Access to remote resources similar to access to local resources.
- *Data Migration* - transfer data by transferring entire file, or transferring only those portions of the file necessary for the immediate task.
- *Computation Migration* - transfers the computation, rather than the data, across the system.
- *Process Migration* - executes an entire process, or parts of it, at different sites.

Distributed Operating Systems Features

- *Load balancing* - distribute processes across network to even the workload.
- *Computation speedup* - subprocesses can run concurrently on different sites.
- *Hardware preferences* - process execution may require specialised processor.
- *Software preferences* - required software might be available at only a particular site.
- *Data access* - run process remotely, rather than transfer all data locally.
- Examples: GLUnix (NOW project), Amoeba, Legion (automated migration/scheduling)

Robustness

- A distributed system may suffer from various types of hardware failure.
- To ensure that a system robust, you need to **detect** failures, to **reconfigure** the system to continue and to **recover** when site or link is repaired

Failure Detection

It is difficult to differentiate between link failure, site failure and message loss.

Handshaking

- Used to detect link/site failure.
- Intermittent sends of "I-am-up" message between sites. If the message is not received, then there is a problem.
- If message is not received, send "Are-you-up" message.

  Site failure – no answer; link failure – send by another site.

- message has set time wait for response.
  - If no reply, direct link down/site dawn/alt. Path down/message lost.

Reconfiguration

- If failure has occurred, the discovering site must initiate a procedure to allow the system to reconfigure to continue its normal mode of operation.
- if link is down, every site needs re-routing information.
- If site is down, every site needs to know, so it doesn't try to use services of the failed site.

Recovery from failure

- When a failed link or site is repaired, it must be integrated into the system gracefully and smoothly.
- if link fail, re-establish handshaking between linked sites.
- If site fail, re-establish site, notify others, update local table to current network sites.