



**Design Notes:**

- Because each node (Sun machine) in the cluster has access to the same set of filesystems (which implies multi-mountable Multiple Reader / Writer filesystems), each has access to, and exports the same data. Clients connections can therefore be IP load balanced among these machines (in a round-robin fashion or, if server performance querying is done by the load balancers, more advanced methods).
- If properly designed, downtime can be reduced to almost zero since a failed node's clients will reconnect to another node on reconnect attempt. The usual caveats are in order here: (1) the behavior/robustness of the clients, in particular whether it has been coded to sense a loss of client/server connection or timeout, and then attempt to reconnect and (2) whether the client/server process itself is stateful, and if it is, whether the server application to which the client was once connected is intelligent enough to release and re-normalize the resources associated with "user" that represented that lost client connection (namely its associated files and metadata). This second point is important because the application has to be written so that, if state is important, it must have the logic to prevent client reconnection until the the server that once hosted that client normalizes everything. This can be as simple as a "cleanup" flag file that is checked by the client.
- Upgrading servers (O/S, patches, hardware) is made easier because the rest of the servers will handle the load of the offlined server.
- The addition of a new server is easy... (just edit gabtab and llthosts, llttab),,,, Volume groups are already made as well as the filesystems on them.
- Talk about IDDM