Project #4: Fault Tolerance

Due: June 2, 2006 11:59pm

Overview

For this assignment you will implement a read-only fail-over replicated distributed file system. The primary goal is to allow clients to continue executing (i.e., reading) in the event of a server failure. This will be possible as long as a replica server is still running. In the interest of time, the servers will provide only read access.

As with the previous project, some requirements are discussed below, but the design of the system is primarily up to you. In particular, you are free to use distributed objects or sockets for the network communication.

Server

For this project, each file will be accessible to all clients and will allow only read access. The server will store all files in the system in a single directory named `FILES`.

The Server, named `FaultServer`, must allow users to "lookup" files and to read from files (this is very similar to the functionality supported in the previous assignment). The server must support multiple simultaneous connections. Since the clients will only read, access to files need not be synchronized.

Client Module

The Client Module, `FaultClientModule`, will, again, provide a bridge between the client code and the file servers. The client module must provide a (static) method, `initialize`, that takes an array of `Replica` (see below) objects. The module then forwards all read requests from the client interface to the first replica.

If, at any point, communication with a replica server fails, the client module must switch to another replica. The client interface will have no knowledge of such a change. A read operation will fail if all replicas have failed.

Client Interface

The Client interface to the system is defined through `FaultInputStream`. The `FaultInputStream` implements the functionality described in (and is a subclass of) `java.io.InputStream`. As in the last project, this implementation is supported by the Client Module described above. Your specific implementation must define the `read` methods from `java.io.InputStream`. The constructor takes the name of the file from which to read as its single `String` argument.

Replica Class

```
public class Replica
{
   private String _host;
   private int _port;

   public Replica(String host, int port)
   {
      _host = host;
      _port = port;
   }

   public String getHost()
   {
      return _host;
   }

   public int getPort()
   {
      return _port;
   }
}
```

Handin

To submit on falcon/hornet: **handin akeen proj4 <your files here>**