6.826 Principles of Computer Systems	**Fall Semester, 1993**

Lecture 21: November 24, 1993

Lecturers: Bill Weihl and Butler Lampson *Scribe: Ira Scharf*

1 Overview

In the first part of this lecture we will finish up the discussion on replication techniques. In particular we will discuss an optimization to the primary/copy scheme, and then talk about an alternative approach called voting. The second part of the lecture will cover a new topic – system management. We will discuss the various issues involved in administering and maintaining a large system and the specific role a system manager plays in these functions.

2 Primary/Copy Optimization

In order for the primary/copy scheme to be robust there must be at least three copies – a primary and two replicas. This insures that there can be a majority if one of the copies fails. It is possible, however, to have only two copies of the data if there are three servers voting. This is called the **witness scheme**.

The witness serves to break ties when one of the other servers fails. In the case of a failure a majority is allowed to be formed with the witness. The amount of storage necessary for the witness is quite small. The witness keeps up on information only about recent votes. This amounts to maintaining the most recent part of the log. Once something has been committed the witness can forget about it.

3 Voting

Another approach besides the primary/copy scheme is to use voting – involving *quorum* and *consensus* amongst the sites.

Under this scheme there are a fixed number of replicas and each replica is assigned a fixed number of votes, for instance one each. A quorum in this system is some set of sites which defines the votes you need to do an operation. For example:

- Read - requires r votes
- Write - requires w votes

Each replica has a value and version number for each piece of data. To do a read you talk to each site and get its value and version number. You then choose the value with the highest version number. To do a write you first do a read quorum to determine the current value. A write is then done by incrementing the highest version number and writing out the new value.

3.1 Constraints on the Voting Scheme

1. When you read you want to make sure you see the latest write. For this reason every read quorum must intersect with every write quorum. This is the only way to insure that a read quorum will be guaranteed to return the most recent write.

2. In the handout reads and writes are treated as atomic actions. In a real system it is necessary to deal with synchronization issues to make sure there are no concurrency conflicts. For this reason a write should be treated as a form of 2-phased commit.

3. Writes require talking to a majority of sites, therefore writes can never be incredibly available. Read quorums, however, can be quite small. Since writes are less available, if any site goes down it may not be possible to write. Read quorums are highly available; a read quorum can consist of only one site.

If you wanted to build a replicated service with the semantics of a non-replicated service it is possible to build a **loosely consistent system**. A loosely consistent system is one in which it is not considered critical to system performance to maintain identical replicas of the data at all times. In such systems a client does an update with one copy and the changes then get propagated out at a later time. Examples of such systems include:

- Athena kerbros - updates get propagated once a day
- Phone book - updates get propagated once a year

The tradeoff in choosing to implement a loosely consistent system is between performance and functionality.

4 System Management — Overview

The role of a system manager is to this day not a coherent or well understood subject. A system manager, for example, is faced with dealing with countless failures and inconsistencies which may arise in large systems. As difficult as the job is to define, a system manager can be an invaluable asset in keeping a large system operating and reliable. A recent Wall Street Journal study found that for every PC purchased companies spend on average $6000 per year on people “futzing” with the machine. “Futzing” they define as people having to spend time doing something which is outside their job description.

A system manager’s role in a large system can be broken down into 2 main areas: setting policy, and keeping it working.

4.1 Setting Policy

Setting policy involves getting inputs to the system from the outside on such things as:

- disk usage
- list of users
- resource allocation

A system manager acts as an authorized agent responsible for registering users in the system and setting up parameters associated with each user. The system manager is also responsible for determining the appropriate allocation of resources (printers, disks, tape drives, etc.) to meet the requirements of the user community.

4.2 Keeping it Working

In an ideal system this component does not exist at all. It is an engineering task to design into the system some level of fault tolerance and error recovery. For example, Internet routers are designed to handle certain failures and faults on the network automatically. This is an example of an **internal** fault recovery algorithm.

By contrast, an **external** fault recovery involves the class of events which the system was not specifically designed to handle. External fault recovery can be either manual, involving operator assistance, or automatic. Certain systems, for example, are designed to detect faults and send repair notices to the vendor automatically. The goal in building system fault recovery algorithms is to move from

$$\textit{manual external} \Longrightarrow \textit{automatic external} \Longrightarrow \textit{internal}$$

5 System Management — Issues

5.1 Naming and Accessing

One important responsibility of the system manager is to develop a mechanism to reach each of the components in a large system. A naming scheme must be implemented which allows access to each of the components and the ability to determine if they are still working. Often it is also a requirement that components be accessible from remote sites over the network.

A hierarchical name space is usually developed to provide access to the following functions from each component:

- Read
- Write
- Enumerate

The Simple Network Management Protocol (SNMP) is one particular convention for structuring a namespace and for implementing the Read, Write, and Enumerate functions. The Internet Protocol and IP addresses are structured in this manner.

5.2 Configuration Control

There are two major issues involved in managing the configuration of a large system: assignment of names and installation/repair.

1. **Assignment of Names**

   Naming in a system can be either based on an ideal model of the world or one which actually reflects the real world. In a sense the ideal naming model can be used to drive the real, or one can built a model more directly based on reality. Setting up naming a scheme for a large system can involve many different facets. IP addresses, for example, can be assigned locally by the system manager. By contrast, ethernet addresses, which are 48-bit sequences, are assigned globally. They must be bought directly from Xerox by the hardware manufacturer for roughly $1000 per block.

2. **Installation / Repair**

   Probably the most involved aspect of a system manager's role in to handle the installation and repair of the variety of devices and components which make up the entire computing environment. In this area the system manager is generally responsible for:

   - Detecting Failures
   - Reporting Failures
   - Fixing Failures

   It is a well known rule that systems which were designed to be fault tolerant often tend to become non-fault tolerant.

5.3 Performance Monitoring

1. **Measure**

   In any large system it is often critical to be able to measure the performance of the system. In general we need to be able to measure all parts of the system in order to diagnose specific performance problems.

2. **Model**

   A model of the system is usually important to develop in order to define what are normal operating specs. This provides a basis to compare measurements to. Without a realistic model of how a system should perform, measurements taken from the system can provide little information. The goal is to localize problems by identifying a component of the system which is not performing up to the model.