6.826 Principles of Computer Systems **Fall Semester, 1992**

Lecture 22: December 2, 1992

Lecturer: William Weihl *Scribe: Quinton Zondervan*

1 Handouts

- Handout 46: Solutions to Problem Set 7
- Handout 47: Replication Techniques
- Handout 48: Problem Set 9
- Handout 49: Gray, Jim, "A Census of Tandem System Availability Between 1985 and 1990," *IEEE Transactions on Reliability*, volume 39, number 4, pg 409-418, October, 1990

2 Overview

The first part of this lecture deals with the issue of availability. How can we built systems that continue to provide service to their users, even when pieces of the system fail? The primary mechanism for achieving availability is to introduce redundancy into the system. The second part of this lecture was a discussion of handout 47, and is not covered by these notes.

3 Availability

There are two types of availability:

1. The availability of the data stored in the system.
2. The availability of active services provided by the system. This includes not only processes that respond to user requests, but autonomous processes as well.

This lecture will focus on techniques for tolerating *stochastic* hardware failures, as opposed to deterministic failures caused, for example, by design faults, which cannot be solved through replication.

4 Replication methods

Replication methods can be roughly divided into two categories:

1. In strongly consistent methods, the collection of replicas that provide a given service create the illusion that there is only one single copy of the service. In other words, the specification does not have to change significantly, except to allow for failure of certain operations.

2. In loosely consistent methods, the fact that replicas exist is visible to the client. In other words, the operations invoked on different replicas are not serialized.

4.1 Strongly consistent methods

Strongly consistent methods usually make use of one of the following two mechanisms :

1. Primary copy, also known as master-slave or primary-backup. One replica, the primary, processes clients' requests and communicates with the rest to inform them of the operation, so that other replicas keep their state up-to-date. In the event the primary fails, one of the other replicas becomes the primary. This scheme has the advantage that the order of the incoming requests is easily maintained, because they all come in through one replica — the primary, except for the duration when there is a change in primaries.

2. Voting mechanisms. The client communicates with some subset of the replicas (for example a majority of replicas). One example of this is TMR (Triple Modular Redundancy) which is a hardware technique in which a piece of hardware is triplicated, and some additional voting hardware (which must also be reliable) is added. Such a module will tolerate any single failure of those three pieces.

4.2 Static and Dynamic Redundancy

A primary copy scheme can make use of replicas in two ways:

1. Static redundancy: If the primary fails, one of the backups is designated as the primary. Until the failed primary is repaired, the system keeps working with one less replica. The situation is similar when a backup goes down. When the failed replica recovers, it is reintegrated into the system.

2. Dynamic redundancy: When one of the replicas fails, primary or backup, a new one is started, so that the system is immediately back to the full complement of replicas. To this end a pool of reserve replicas is maintained that camn be used to replace the failed ones. When the failed replica is repaired, it is returned to the reserve pool. This scheme effectively reduces the repair time seen by the system, thus increasing availability.

5 Evaluating availability

For this analysis, we will assume that we have one primary and one backup. First we need some terminology:

- **MTTF** The Mean Time Between Failures (for each component, and for the system as a whole). We assume that the failures occur stochastically, according to some Poisson distribution (i.e. the failures happen in a memoryless way).
- **MTTR** The Mean Time To Repair. This indicates how long it takes to repair a failed component (on average). In static redundancy, for example, this time will be fairly long, because the component has to be physically repaired. In dynamic redundancy, this time is relatively short, because it is simply the time it takes to start up a new replica process (on a different processor). The failed hardware can then be repaired off line.

Given these two parameters, one estimate of the availability of the system is:

$$MTTF/(MTTF + MTTR)$$

This approximates the percentage of time during which the system can provide service to the users. This assumes that MTTF is significantly larger than the time that is required to process a single operation. If the system keeps failing before it is done with a single operation, it is effectively totally unavailable.

Given identical primary and backup components, the MTTF for the pair working together, assuming that a faulty component gets repaired, is given by :

MTTF$_{pair} = MTTF_{one}^2/MTTR_{one}$

This means we can improve the MTTF$_{pair}$ by either increasing the MTTF$_{one}$, or by decreasing the MTTR$_{one}$.

5.1 Duplex disks

As an example we consider a duplex disk system, where a read corresponds to read from either disk, and a write corresponds to write to both disks. We also assume that failures are detectable (this is not an entirely realistic assumption). This avoids the problem of latent failures, which would skew the analysis, since we won't repair it unless we know that it has failed. We often assume that failures are fail-stop, i.e. when the disk fails it stops (as opposed to causing damage, known as Byzantine failure), and the failure can be detected (this can be achieved with redundant hardware for self-checking).

In 1987, Tandem reported an MTTF of 50,000 hours (6 years) for a single disk (today that is about 5 times better) and an MTTR of 5 hours for a single disk. According to our formula, the MTTF for a pair should equal about 60,000 years! The observed value of the MTTF however was about 1300 years. One of the main reasons for the discrepancy is that failures were not in fact independent. For example, when one disk fails, the repair person would come in and accidentally fix the working disk! The problem was fixed by putting indicator lights on the disk units.