Building a disk failure injection framework for Fault-Tolerant systems research

Yathindra Naik[1] *Mike Hibler* *Eric Eide* *Robert Ricci*
School of Computing, University of Utah

Abstract

Storage is one of the most common problematic subsystems in a cloud platform. Disks are getting bigger as storage requirements continues to grow everyday. Building highly fault tolerant system requires the ability to recognize and handle various disk errors. Today's cloud platforms are built with many systems which are composed of heterogeneous storage disks differing in speed and capacity. Infrastructures providing such diversified environment would be ideal for system designers to experiment and test their systems. Such an environment is provided by Emulab testbed. Emulab is a network testbed, giving researchers a wide range of environments in which to develop, debug, and evaluate their systems. Emulab already offers the ability to intentionally degrade hosts and links by turning off the hosts and imposing traffic shaping and drops on the link. In this work, we extend these features to disks. We present a controllable and repeatable disk failure framework for an entire cluster testbed. Device mapper [4] which is a linux driver for creating virtual disk is being used for this purpose. This is still a work in progress.

1 Introduction

Storage forms the backbone of any cloud service. Storage failure can result in downtime and data loss. Dealing with storage failures therefore becomes necessary for building a highly stable system. Enterprise storage systems provided by NetApp typically use different RAID configurations [5] and remote mirroring techniques [6] to protect data from disk failures. Previous studies conducted by Google [7], and others [2] [3] show various disk failure characteristics.

Emulab [8] is widely used by computer science researchers in the fields of networking and distributed systems. Emulab can be used by researchers to test how their systems behave in the face of unstable storage. We present a framework on Emulab called the $disk - agent$ to provide a means for injecting various disk failures. We have made use of device mapper which is a driver in linux as it already provides a simple way to achieve the same.

We have integrated device mapper on Emulab nodes and made use of Emulab's event system to provide a means for experimenters to inject various disk errors. Users will have the ability to script their experiments and also specify the time they want to trigger disk failures. NS is the experiment description language for Emulab, which is based on the language used by the NS simulator. We have added some extensions that allow users to use NS's at syntax to invoke disk failure events on their experimental nodes. Once the users define a disk object and initialize its parameters, they can schedule disk failures with NS at statements. To define a disk object:

```
1. set disk0 [$nodeA disk-agent -name
"bad_disk" -type "linear" -mountpoint
"/mnt"]

2. set disk1 [$nodeA disk-agent -name
"bad_disk" -type "flakey" -mountpoint
"/mnt" params="1 1"]
```

Then in the NS file a set of static events to run these commands:

```
3. $ns at 10 "$disk0 start"

4. $ns at 20 "$disk1 start"
```

In the above example, we would create two disk objects, disk0 and disk1 where both refer to the same virtual disk named bad_disk. In line 1, disk0 is declared to have a direct mapping between the virtual and real disk (the disk agent on the node would figure out and create a partition for the real disk). In line 2, disk1 is declared to be

[1] Students; Yathindra Naik will present

a flaky disk and the extra parameters to flakey type tells device mapper to fail 50% of the disk I/O's. Finally, using the event scheduler, bad_disk starts off being a good disk (line 3) and 20 seconds later starts giving I/O errors (line 4).

This framework will be used on PRObE (Parallel Reconfigurable Observational Environment) supercomputing facility. PRObE is an NSF-sponsored project aimed at providing a large-scale, low-level systems research facility. It will use Emulab to support research in many systems related fields such as Operating Systems, Storage, and High End Computing[1].

2 Design and Implementation

The systems that are hosted in a testbed cluster can support a wide variety of applications and network protocols that eventually talk to the storage backend. It is common to find distributed filesystems and large databases that need raw disk access. Therefore, we needed a fault injection framework that could operate at a block layer rather than the file level. It is also becoming increasingly common to find large distributed applications hosted on commodity hardware. The idea here is to build much of the fault handling features on upper layers of the software stack, realizing disks are most likely to fail.

Disk faults could due to a number of reasons - medium scratches, firmware bugs, mechanical failures, interconnect cable issues, faulty disk adapter etc. Disks are mechanical devices and undergo wear and tear due to friction which is again directly affected by workload, temperature, and other environmental issues [7] [2]. Different classes of disks show varying reliability. For example, Lakshmi N. Bairavasundaram et al, note that nearline disks have an order of magnitude higher probability of developing checksum mismatches than enterprise class disks [2].

Disk failures are indicated by early disk errors which can broadly be classified as:

- Latent sector errors - This kind of error results in a sector or groups of sectors becoming inaccessible. Read or write to those sectors fail and usually indicated as SCSI medium errors.
- Data corruption - Data is inconsistent on disk. They usually show up as checksum errors if the block is protected by checksum.
- Transient errors - Disk read/write fails or times out.
- Slow disk - This is a relatively less known class of failure where the disk I/O's become significantly slow.

2.1 Disk-Agent for Emulab

Emulab is composed of a event-system and a scheduler which dispatches those events at the required time. It also provides a dynamic way to trigger events on various nodes. Disk agent framework will leverage this to provide users with two different modes of operation: one in which users control the disk agent while the experiment is running by dynamically sending disk events. The other is that users can script things all out ahead of time so that they can get repeatable, predictable behaviour.

Users of Emulab typically set up their experiments by specifying the type of topology they are interested in by using NS scripts. They can perform experiments which allows them to inject failures at a later point in time through the NS scripts. So users can inject probable disk errors which are more realistic since disks wear out due to time and usage.

The other part of the idea is to model disk failures. We plan to use some of the disk failure models from literature which gives users the ability to foresee how applications behave when disks starts to degrade. Some of the papers talk about disk failure analysis from a very large disk population for a long duration [2]. By making use of such studies, we can parameterize various factors that affect disk failures such as time, workload, disk class, temperature etc and compress them to shorter timescales.

References

[1] Probe. `http://newmexicoconsortium.org/probe`.

[2] BAIRAVASUNDARAM, L. N., GOODSON, G. R., PASUPATHY, S., AND SCHINDLER, J. An analysis of latent sector errors in disk drives. In *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (New York, NY, USA, 2007), SIGMETRICS '07, ACM, pp. 289–300.

[3] JIANG, W., HU, C., ZHOU, Y., AND KANEVSKY, A. Are disks the dominant contributor for storage failures?: A comprehensive study of storage subsystem failure characteristics. *Trans. Storage 4* (November 2008), 7:1–7:25.

[4] MILAN BROZ. Device Mapper. `http://mbroz.fedorapeople.org/talks/DeviceMapperBasics`.

[5] PATTERSON, D. A., GIBSON, G., AND KATZ, R. H. A case for redundant arrays of inexpensive disks (raid). In *Proceedings of the 1988 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 1988), SIGMOD '88, ACM, pp. 109–116.

[6] PATTERSON, H., MANLEY, S., FEDERWISCH, M., HITZ, D., KLEIMAN, S., AND OWARA, S. Snapmirror®: file system based asynchronous mirroring for disaster recovery. In *Proceedings of the 1st USENIX conference on File and storage technologies* (Berkeley, CA, USA, 2002), FAST'02, USENIX Association, pp. 9–9.

[7] PINHEIRO, E., WEBER, W.-D., AND BARROSO, L. A. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX conference on File and Storage Technologies* (Berkeley, CA, USA, 2007), USENIX Association, pp. 2–2.

[8] White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., and Joglekar, A. An integrated experimental environment for distributed systems and networks. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation* (Boston, MA, Dec. 2002), USENIX Association, pp. 255–270.