Cluster Storage System, RAID and Solid State Drive Emulation with David

Leo Arulraj, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau
Computer Sciences Department, University of Wisconsin–Madison
{*arulraj,dusseau,remzi*}*@cs.wisc.edu*

Abstract

David (to appear in the proceedings of FAST 2011) is a light-weight in-kernel storage system emulator that allows emulation of large storage devices using a comparatively small amount of physical storage. Our previous work experimentally showed how a 1TB disk can be emulated using a 100GB disk for benchmarking purposes. We also showed simple RAID-1 setups in our previous work. In this poster, we present our ideas on extending David to allow emulation of three new and interesting types of storage systems : a) Cluster Storage Emulation using a desktop b) SSD emulation c) Complex RAID Schemes. We discuss the challenges of each of the above kinds of emulation below.

1 Background

David achieves emulation of large storage device using small amounts of backing storage by storing only the metadata persistently. David throws away writes to data blocks and generates fake content during subsequent reads to the data blocks. Since most benchmarking applications care only about the response times during data reads and not the actual contents of the data blocks, this strategy saves a lot of physical storage. However, writes to metadata blocks are stored persistently on the backing store and reads to metadata blocks are serviced from the backing store.

2 Cluster Storage Emulation

David already allows emulation of a large storage device using a small storage space. We want to explore the limits of David in scaling to emulate numerous disks. How many disks can David emulate in a single Desktop machine ? For example, today David allows emulation of a 200GB disk using 20GB of physical space without using *"compression of metadata"*. In theory, today David should be able to emulate approx 250 disks using a 5TB physical storage space. To do this, we will need smart disk models that allow light weight modeling with *minimal computational resources*.

With such a setup, David can be effectively used as a cost effective tool to answer several questions regarding Cluster Configuration like : a) How many of the disks in a cluster should be SSD drives and how many should be magnetic drives to satisfy latency requirements of a given workload ? b) How to configure a cluster to optimize for power usage: (how many of the magnetic disks should have power saving features like Dynamic Speed Control, what policies should be used to decide which disks to spin down when ?) c) How to reduce the cost of a cluster for a given workload by configuring a fraction of the storage devices with "cheap but slow disks" and the rest with "fast and expensive disks" ? etc.

3 Solid State Drive Emulation

The response times of the backing store plays a big role in the accuracy of the emulation. An SSD drive cannot be emulated with a magnetic disk as a backing store. To illustrate this, let us consider an example. Let us a say read request to metadata is being emulated and the time predicted by the model of the SSD disk is 200 microseconds. If it takes 3 milliseconds to read the metadata content from the backing magnetic storage drive, then the response to the read request can only be sent after 3 milliseconds by the emulator. This leads to inaccurate emulation. We want to explore how to emulate SSD drives using techniques like *compression* to reduce the size of the metadata and to use a combination of "RAM Memory" and "SSD storage drive" as the backing store.

4 Complex RAID Scheme Emulation

We want to explore smart *"data generation techniques"* and *"storage space reduction techniques"* in RAID setups by using the knowledge about the RAID setup. As part of the data generation aspect, we want to explore how to generate the content of the data blocks and parity blocks on the fly in a fast manner. As part of "storage space reduction" aspect, we want to explore techniques like storing only one copy of a block in mirrored RAID setups.