DiskReduce: RAIDing the Cloud

Lin Xiao

Bin Fan, Wittawat Tantisiriroj, Garth Gibson

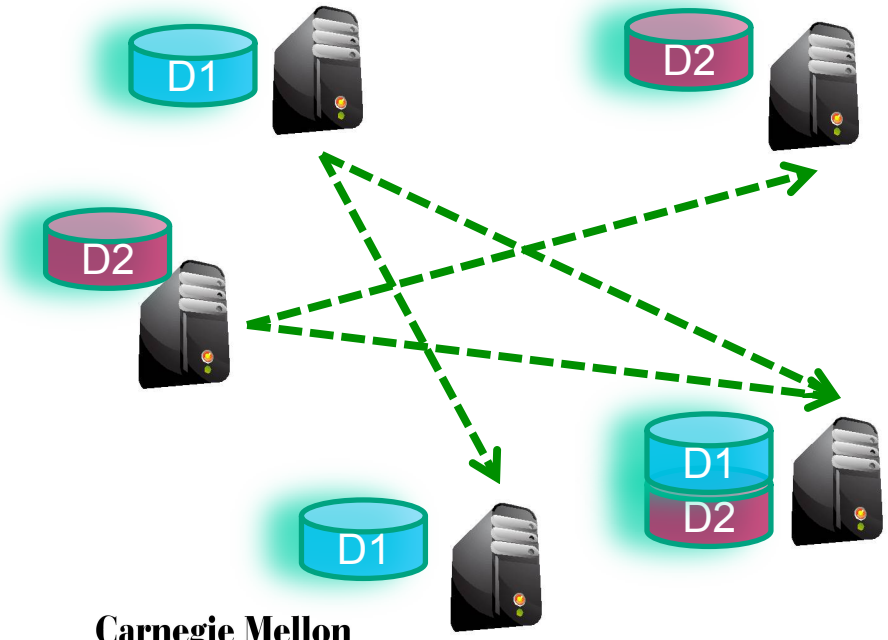
Carnegie Mellon University

Carnegie Mellon
Parallel Data Laboratory

YAHOO!
pdsi
Office of Science

Motivation: Save Space

- **GFS & HDFS triplicate every data block**
  - Triplication: one local + two remote copies
  - But 200% space overhead

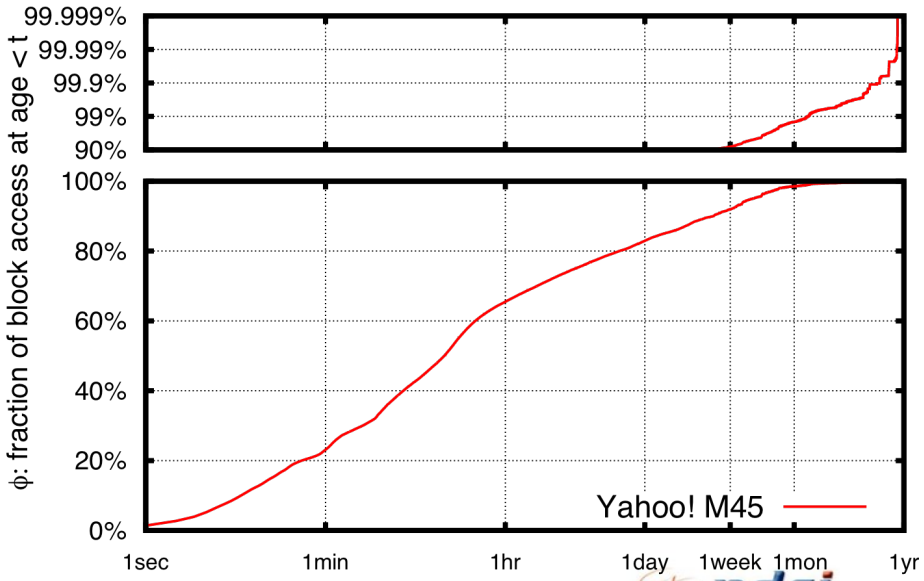


- **RAID technique can lower overhead**
  - Overhead of RAID 6: 2/n
    - n: # of data blocks in a group
  - But sync error handling at client side is hard*
    - Complex logic, code

* Panasas does Object RAID over servers [Welch08]

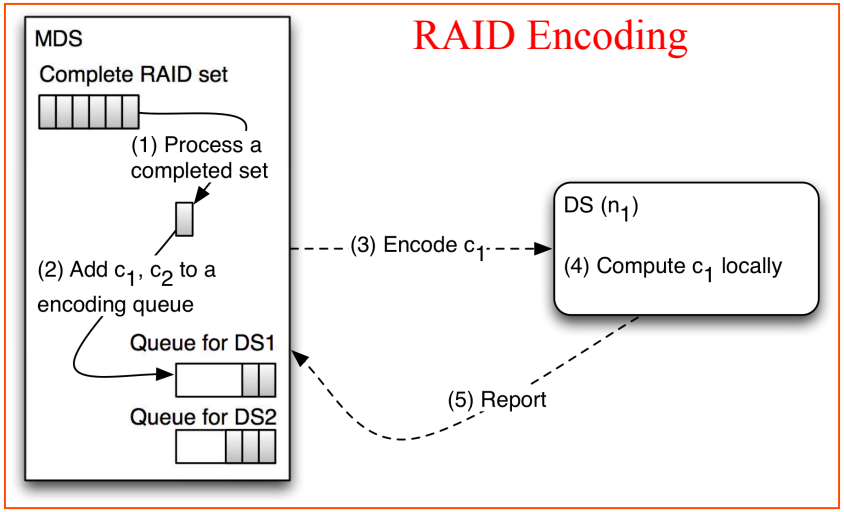
Basic Idea: Async. Encoding

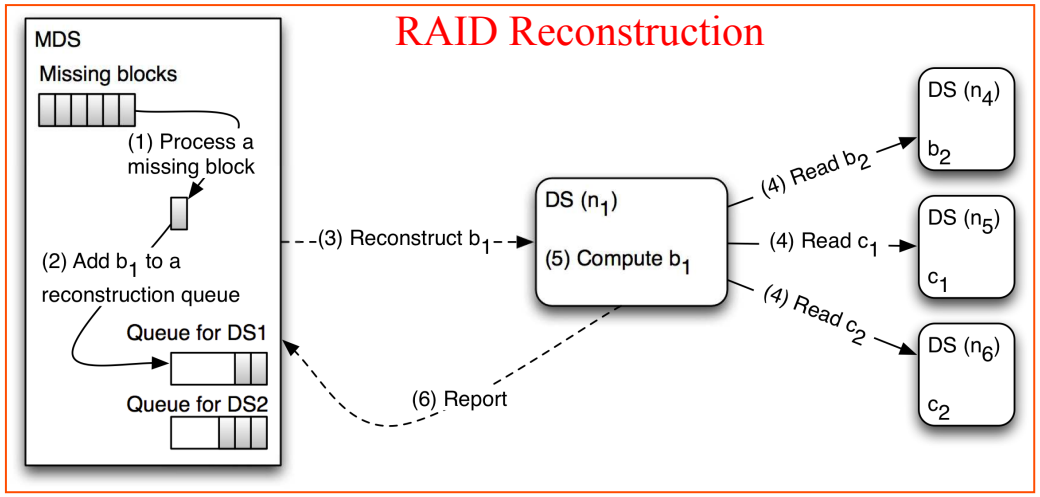
- **Triplicate data blocks initially**
- **Defer RAID encoding, similar to AFRAID** [Savage96]
  - Async encoding modeled after HDFS recovery process
    - GFS & HDFS defer repair in background tasks to repair missing copies
    - Notably less scary to developers
  - Hide encoding cost
    - encode when idle
  - Benefit read performance
    - 80% of reads can be served when file has 3 copies w/ 1 day delay



Encoding & Reconstruct

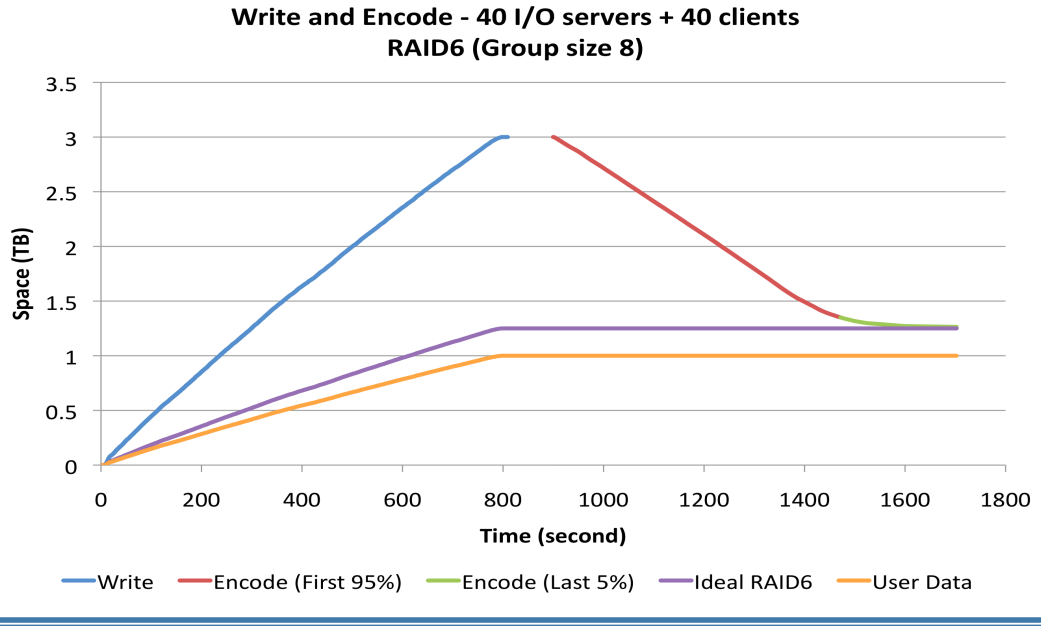
- An encoding task is scheduled at MDS and queued for each data server. Computation can be local w/ proper initial placement.
- Recovery of a missing block is queued as in original but data server does RAID reconstruct





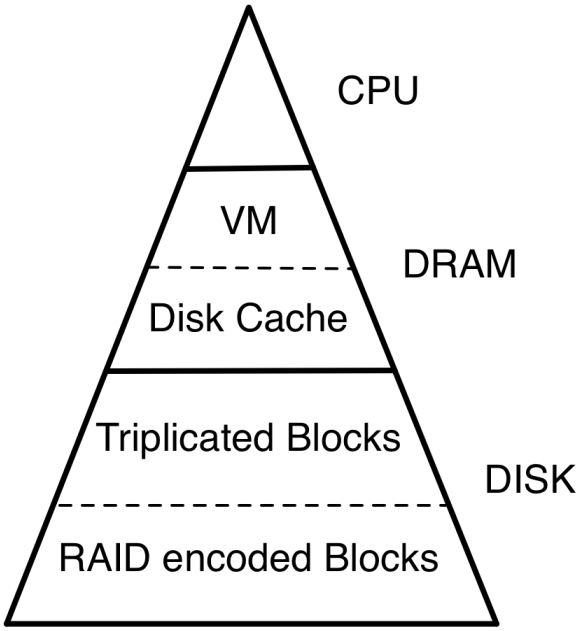
Prototype: It is Working!

- **Prototype implementation based on HDFS 0.20.0**
  - Write total 1TB on 40 nodes
    - 1.25 user GB/s
  - Achieve ideal space overhead 25% after encoding
  - Encoding is expensive
    - delay encoding to idle time



Lets Cache!

- **Triplication may benefit performance**
  - Treat data in disk as in two-layer cache [Cate91] [Wikes96]:
    - Triplication layer v.s. RAID layer
- **"Cache" Replacement**
  - Triplicate all data recently written
  - Apply LRU to decide when to turn data in triplication into RAID



Closing

- DiskReduce for HDFS
  - Give users ~3X more stored data
  - Exploit async encode
    - shift encoding to idle period
    - benefit read performance
- Not covered in this talk:
  - Async deletion
  - Fragmentation, the never beaten annoyance

Related Work

- [Welch08] Brent Welch and et al., Scalable Performance of the Panasas Parallel File System
- [Savage96] Stefan Savage and et al., AFRAID: A Frequently Redundant Array of Independent Disks.
- [Cate91] Vincent Cate and et al., Integration of Compression and Caching for a Two-Level File System.
- [Wilkes96] John Wilkes and et al., The HP AutoRAID hierarchical storage system