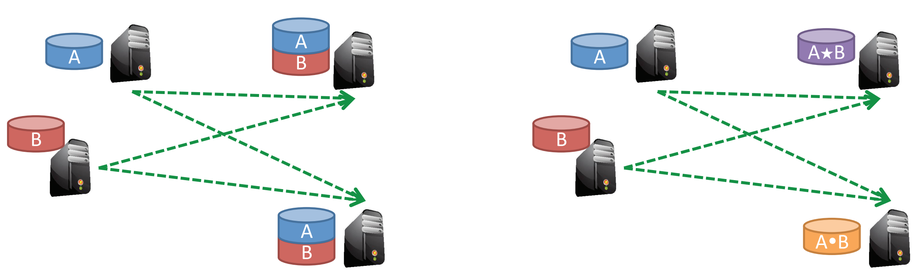
DiskReduce: RAIDing the Cloud

Bin Fan, Wittawat Tantisiriroj, Lin Xiao, Garth Gibson

Overview

Google FS/ HDFS on Data Intensive Scalable Computers

- **Triplication can recover from 2 failures but it trades 200% extra storage for this redundancy**
- **Parity saves storage and tolerates the loss of any two nodes**



GFS / HDFS DiskReduce

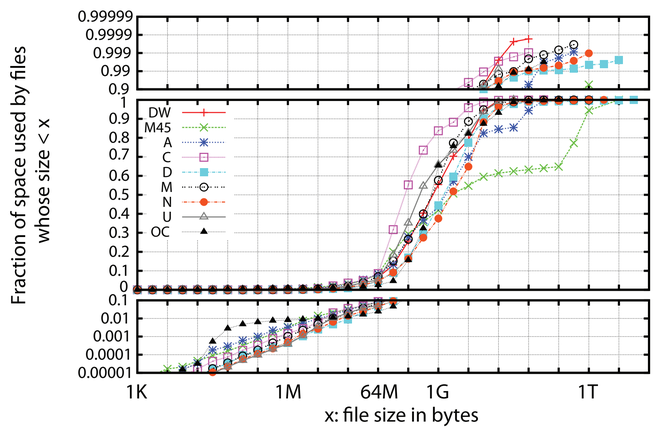
RAID Per-File vs. RAID Across-Files

RAID Per-file: blocks in a RAID set are from the same file

+ Simple

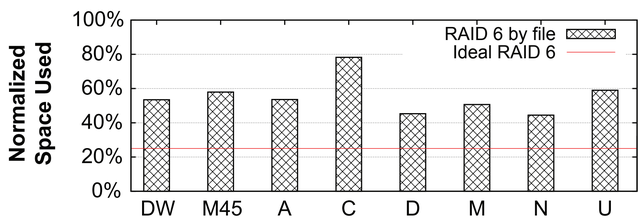
- Too much overhead

Cloud file size distribution



- **Across 9 file systems (1.5 - 21 PB), 30 - 80% of the storage used by files smaller than 1 GB (size of 16 blocks, 64 MB each)**
- **Since each block is large (64 MB by default), small files tend to form short RAID sets**

Space overhead



- **When group size w = 8, per-file RAID 6 requires about 50%overhead while ideal RAID 6 requires only 25% overhead**

RAID Across-files: blocks in a RAID set can be from different files

+ Per-directory RAID 6 can archieve much lower overhead

- Small write problem - potential read-modify-write to update parity blocks on single file deletion

- **To reduce extra work**
  1. group blocks by directory (likely to be deleted together)
  2. defer deletion
  3. after awhile, replace deleted blocks with new blocks

Immediate vs. Background Encoding

Immediate encoding:

+ Efficient

- Complex: Handling failures on critical path

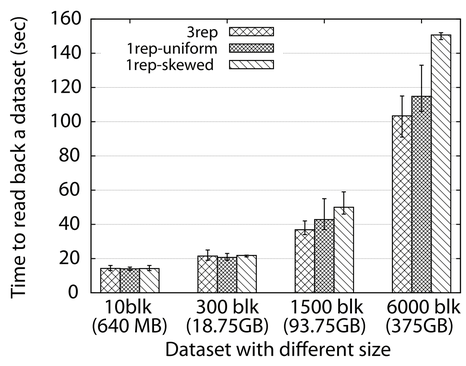
Background encoding:

+ Simple & no change in client code

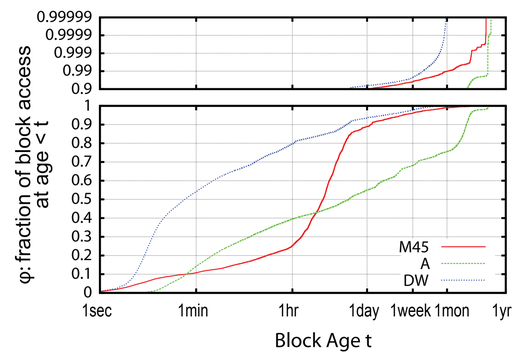
+ Cache young data for higher read bandwidth

- Less efficient

Reading performance

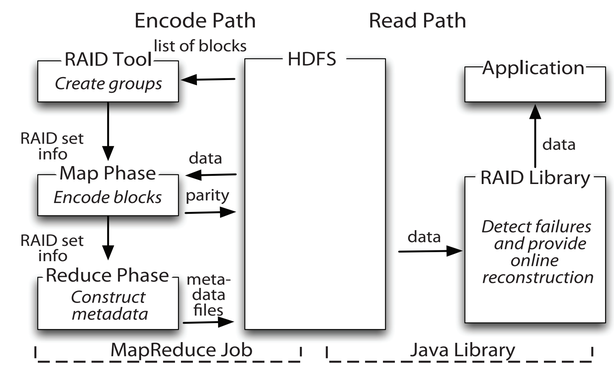


Cloud data access pattern



- **Read performance:**
  - **RAID encoded data can be read as fast as if triplicated**
  - **If data access very skewed, more copies helps performance**
- **Treat triplicated data as in cache:**
  - **Locality: over 90% of data blocks accessed within 1st day after creation in M45 & DW and 50% in cluster A**

Prototype



- **The prototype is built as a tool and a client library**
  - **Tool (Mapreduce): encode a directory into RAID sets or repair corrupted files**
  - **Library: detect and correct missing data while reading**
- **Released as Mapreduce-2036 patch for HDFS 0.22.0 @ http://issues.apache.org/jira/browse/MAPREDUCE-2036**

- **60 nodes (two quad-core 2.83GHz Xeon, 16GB memory, four 7200 rpm SATA 1TB disks, 10 Gigabit Ethernet)**
- **Dataset: 240GB (3,840 files, each 64MB in size)**

Operation	Throughput GB/s(stdev)	Disk I/O GB/s (stdev)
Write(Triplication)	1.93(0.06)	5.80(0.18)
Encode(RAID6 8+2)	3.69(0.34)	4.61(0.43)
Repair	0.23(0.02)	2.09(0.19)

- **Encoding is fast but reconstruction needs tuning**

Carnegie Mellon YAHOO! pdsi facebook Parallel Data Laboratory PDLR10