DiskReduce: Implementation

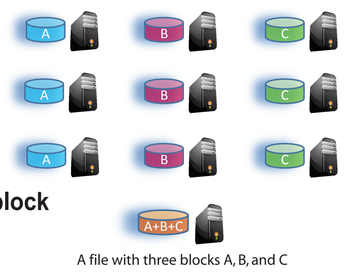
Wittawat Tantisiriroj, Lin Xiao, Bin Fan, Garth Gibson

After Hearing Us Talk: RAID + Mirror

- Hadoop HDFS (0.22.0) implemented a version of DiskReduce v1 (two copies + RAID 5 encoding)
- Thanks to Druba Borthakur & the HDFS team

HDFS RAID

- Start the same: triplicate every data block
- Background encoding
  - Combine third replica of blocks from a single file to create parity block
  - Remove third replica
  - Apache JIRA HDFS-503
- DiskReduce from CMU



A file with three blocks A, B, and C

What We Are Building: RAID6

EXAMPLE:

Step0: N0 picks a codeword (x,N1,N2,N3,N4) randomly

Step1: N0 creates D1 and sends to N1, N2

Step2: N0 creates D2 and sends to N1, N3

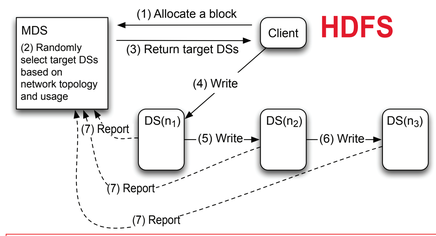
Step3: N0 creates D3 and sends to N1, N4

Step4: N0 and N1 encode D1,D2 and D3 and P1=f1(D1,D2,D3), P2=f2(D1,D2,D3)

Implementation State Machines

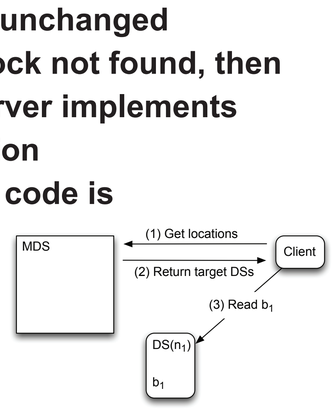
IMPLEMENTATION - WRITE

- Write unchanged
- Except policy for selecting location of replicas
- A key design principle is that initial writing is unchanged, starting with triplication



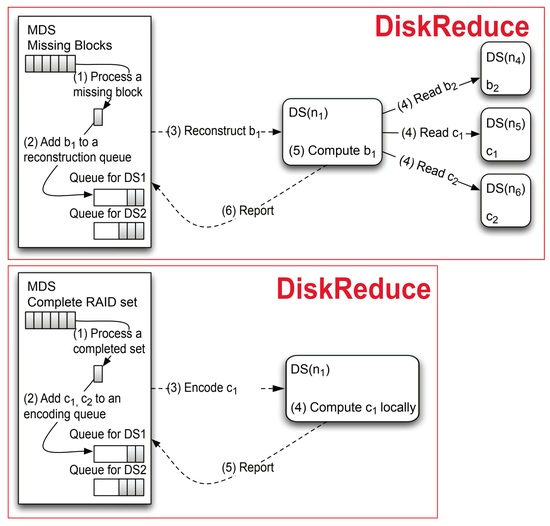
IMPLEMENTATION - READ

- HDFS Read unchanged
- Except if block not found, then 2nd data server implements reconstruction
- HDFS client code is unchanged!



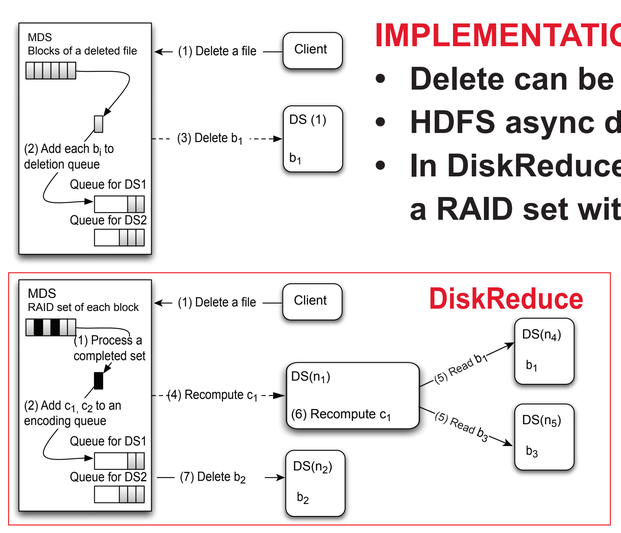
IMPLEMENTATION - RECOVERY & ENCODING

- Recovery extended
- A missing block is queued for recovery as in original but data server does RAID reconstruct
- Encoding is triggered using same queuing but computing check block can be all local if triplication of blocks in RAID set chosen appropriately



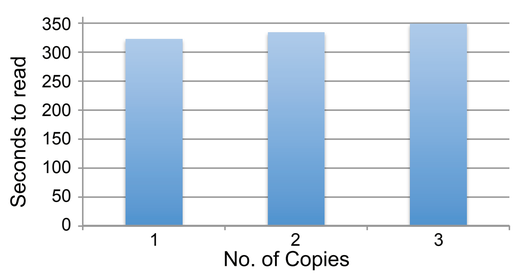
IMPLEMENTATION - DELETE

- Delete can be harder
- HDFS async deletes each block in a file
- In DiskReduce if a deleted block was in a RAID set with blocks that are not deleted check codes become wrong when block is gone – check blocks must be recomputed to recover capacity

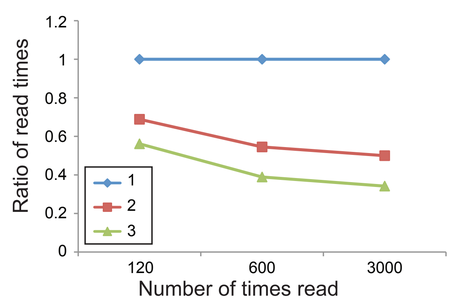


Delaying Encoding/Re-encoding

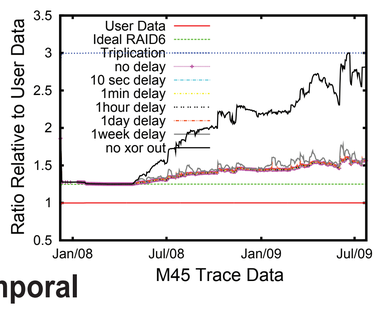
- Delaying encoding – performance advantages from having multiple copies to read from?
- Simple test: 29 nodes, 116 files each 4GB, 64MB blocks, read each byte once via Hadoop in Y seconds
- 3 cases: one, two or three copies of each block
- No significant advantage (useful bytes on every disk)



- Try harder: small hot files: 512MB file in one 512MB block, read redundantly by X maps (30 nodes)
- Two copies faster by 25% - 50%, three copies faster by 40% - 60%
- There are significant performance benefits from replication, but harder to get than we expected



- Deleting a block in a RAID set forces check codes to be recomputed in order to recover block's space
- Delaying delete to avoid recompute (xor) comes with a capacity penalty
- Penalty huge if wait for all blocks in RAID set to die
- Need to recompute to recover space, but can shift to "idle" time
- Interesting choices of which blocks to assign a RAID set to improve temporal locality of deletion



Carnegie Mellon

pdsi

YAHOO!

Parallel Data Laboratory

SVD10