Cider

A Case for Block Level Variable Redundancy on a Distributed Flash Array

Sharath Chandrashekhara, *Madhusudhan R. Kumar, Mahesh Venkataramaiah, and Vipin Chaudhary**

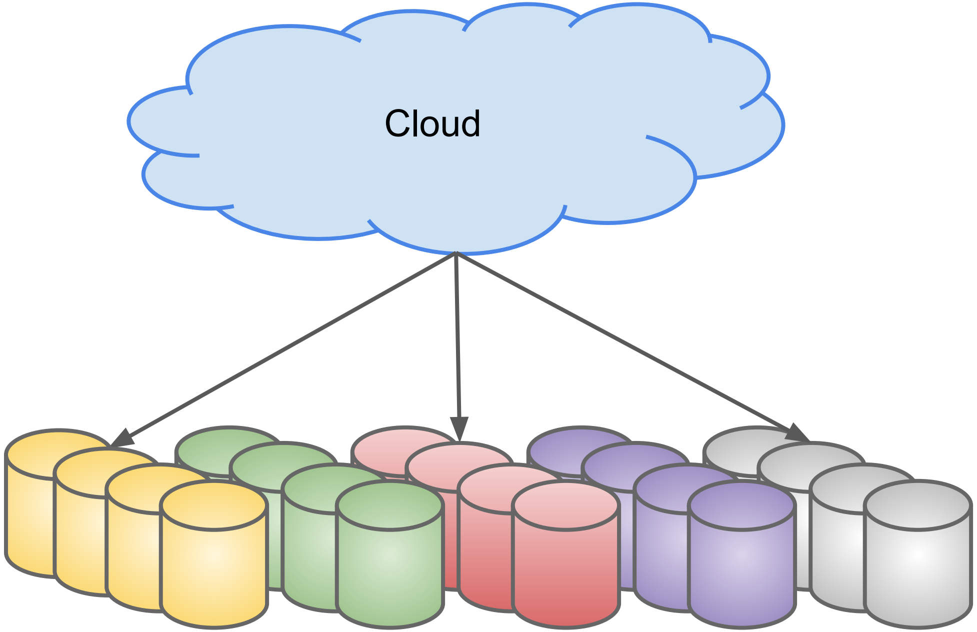
University at Buffalo, The State University of New York

**This work does not reflect the policy of NSF in any way*

Welcome to the Petabyte World

- Data is exploding and hoarded
- To large for traditional data centers
- Enterprise data migrating to cloud
- Cloud: Lower costs & higher reliability

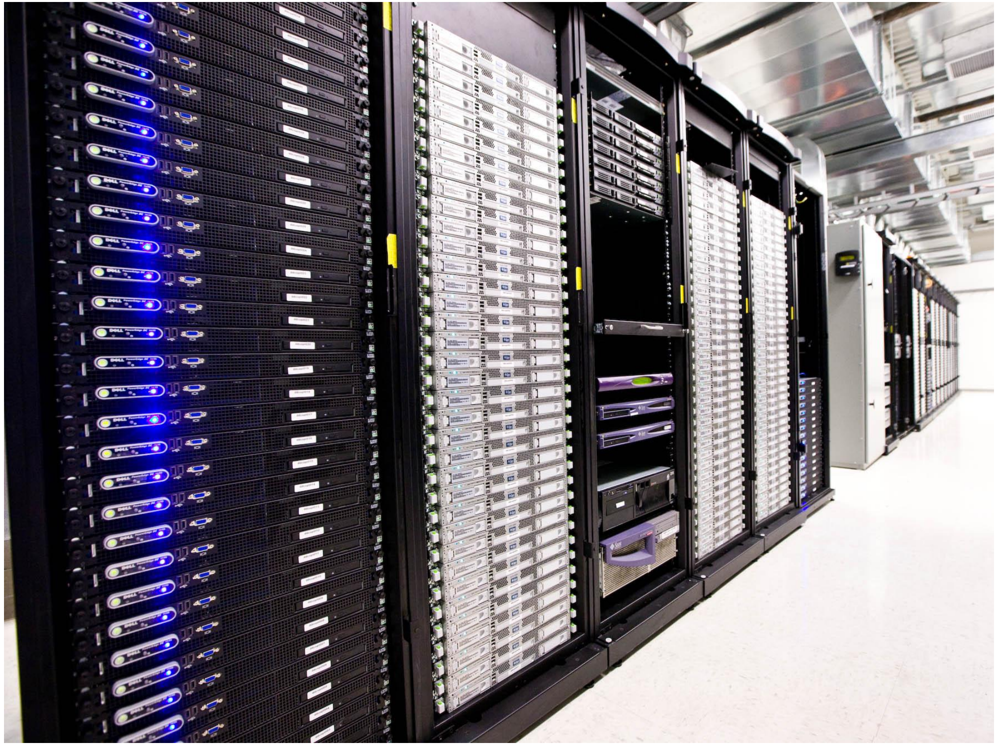
So, is cloud just a large data center?



Storage Arrays

Cloud Data Centres

- Different from traditional data centers
- Much larger capacity
- Infrastructure shared by many clients with different needs
  - QoS, Security, etc.
- Need high flexibility



Traditional Solutions: Features

- RAID for redundancy
- SAN based block storage or NAS servers
- Interface used for traditional POSIX applications
- Replication across data centers or storage racks

RAID Based Storage

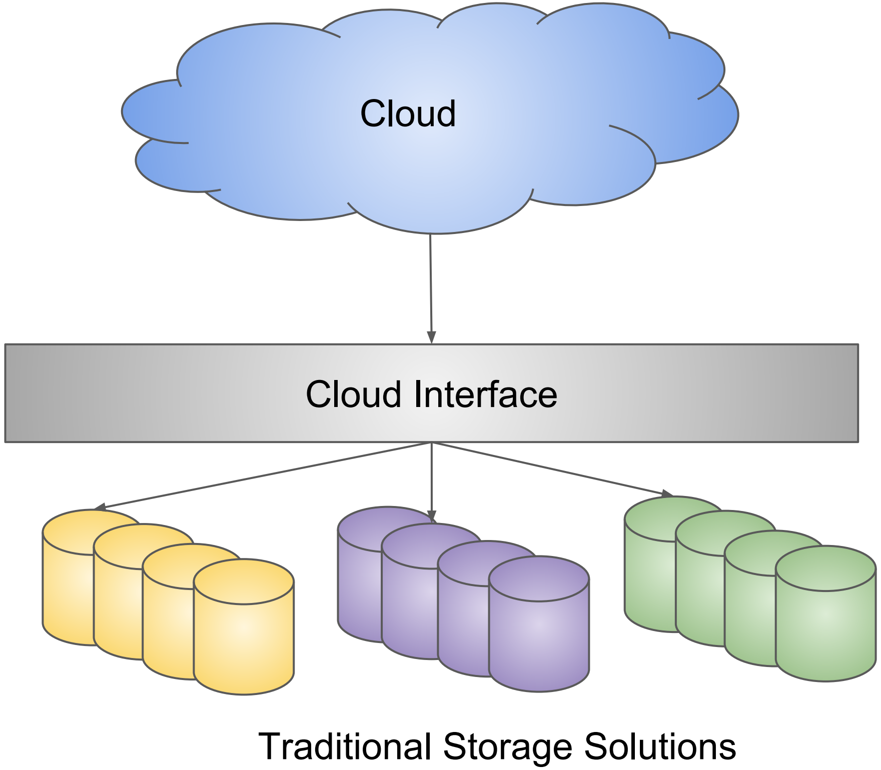
Traditional Solutions: Limitations

- Rigid: Cannot change the redundancy level once created
- Failure domain cannot span across multiple data centers
- High reconstruction time
- Unrecoverable Read Error

RAID Based Storage

Cloud Storage

- Redesign entire stack
- Often built on top of traditional data centers addressing some of the shortcomings
- Erasure coding and full replication
- Cost/Byte increases with reliability guarantee

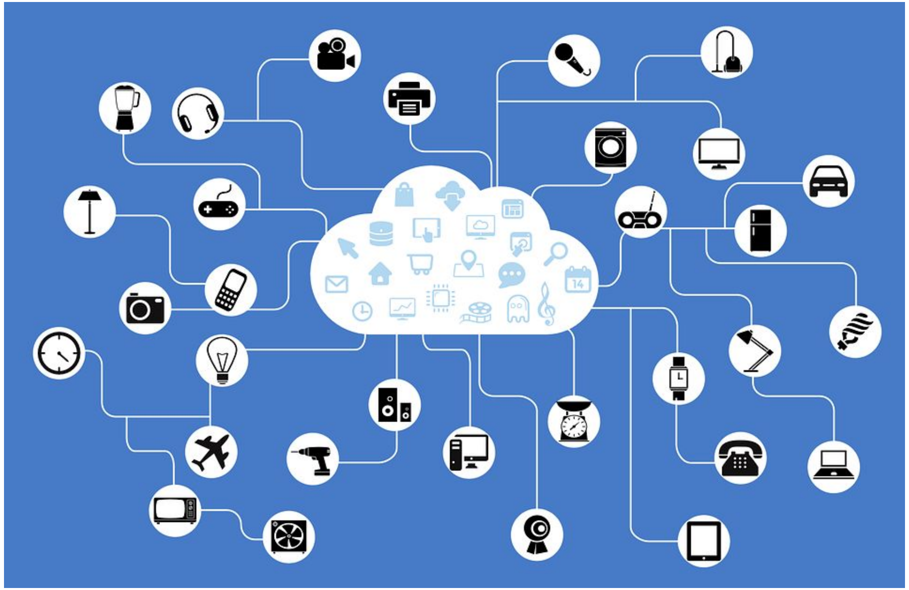


User/Application’s View of Data

Example: Financial Data

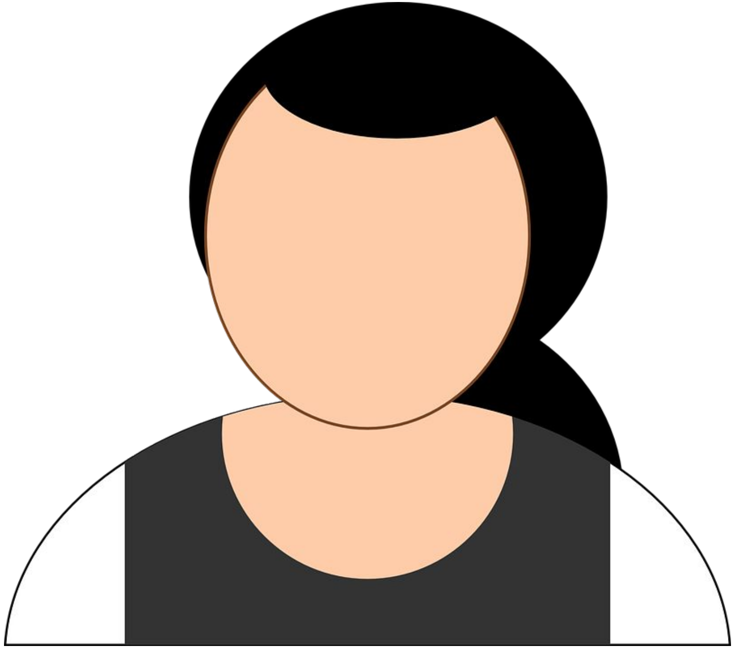
- Maximum reliability for 7 years (Legal requirement)
- Medium reliability for 7-10 years
- Low reliability for 10+ years

Example: Data from Internet of Things



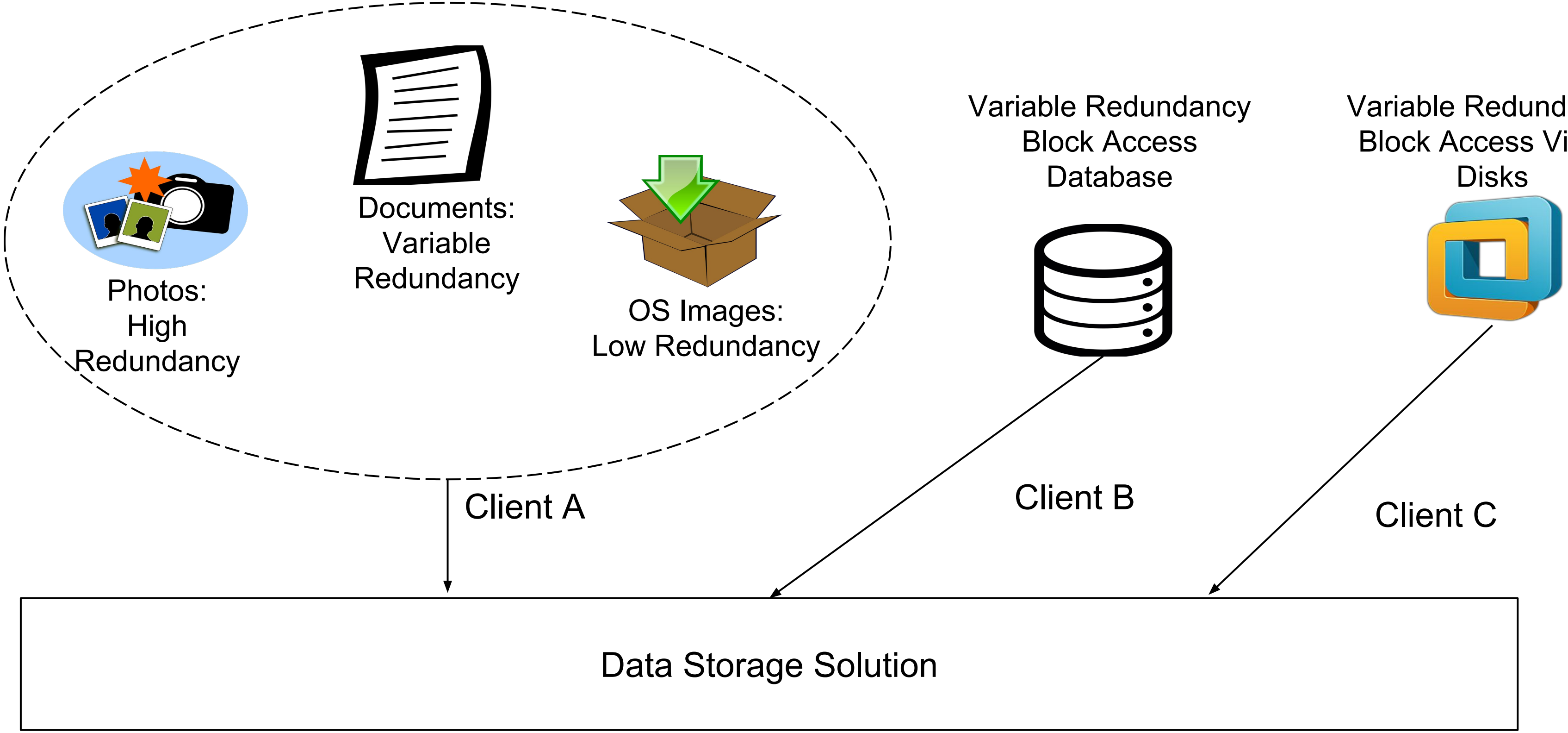
- High reliability to start with
- Reliability decreases temporally

Example: Regular Users



- Health records: Forever
- Personal photos: Highest reliability
- Other documents: Medium priority
- Downloaded files: Low priority

Multiple Clients



Photos: High Redundancy

Documents: Variable Redundancy

OS Images: Low Redundancy

Variable Redundancy Block Access Database

Variable Redundancy, Block Access Virtual Disks

Client A

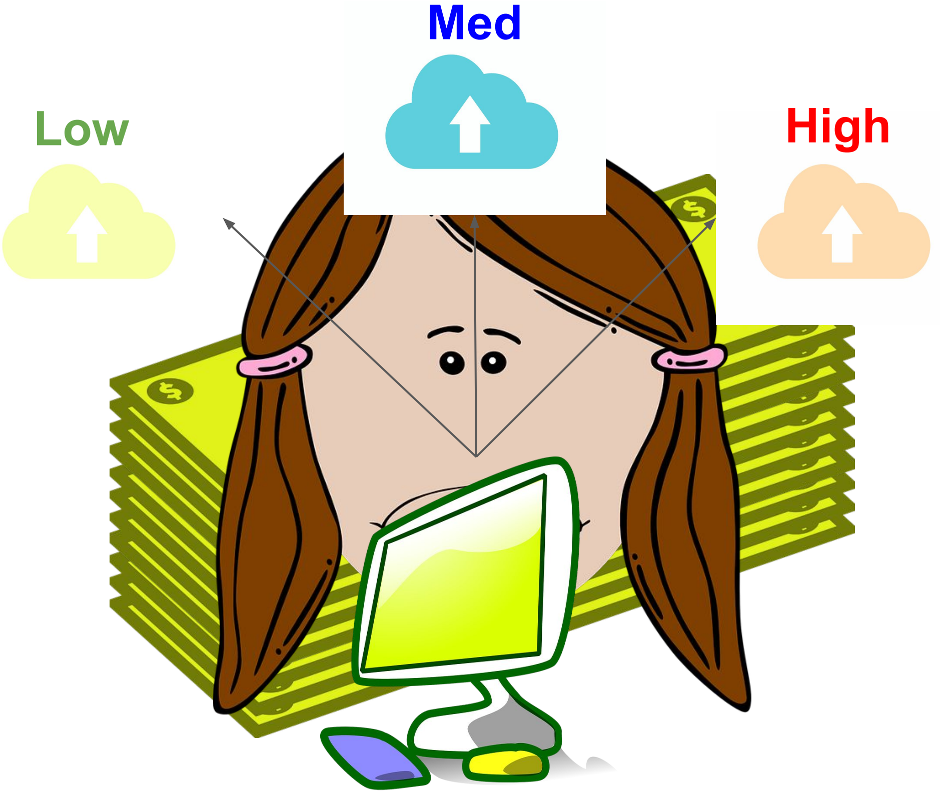
Client B

Client C

Data Storage Solution

User’s Choice

- All data with the highest reliability!
- Different sets of data in different cloud stores
- Less than ideal



Problem Statement

Can we create a more flexible storage backend that the cloud can readily use?

Cider: Goals

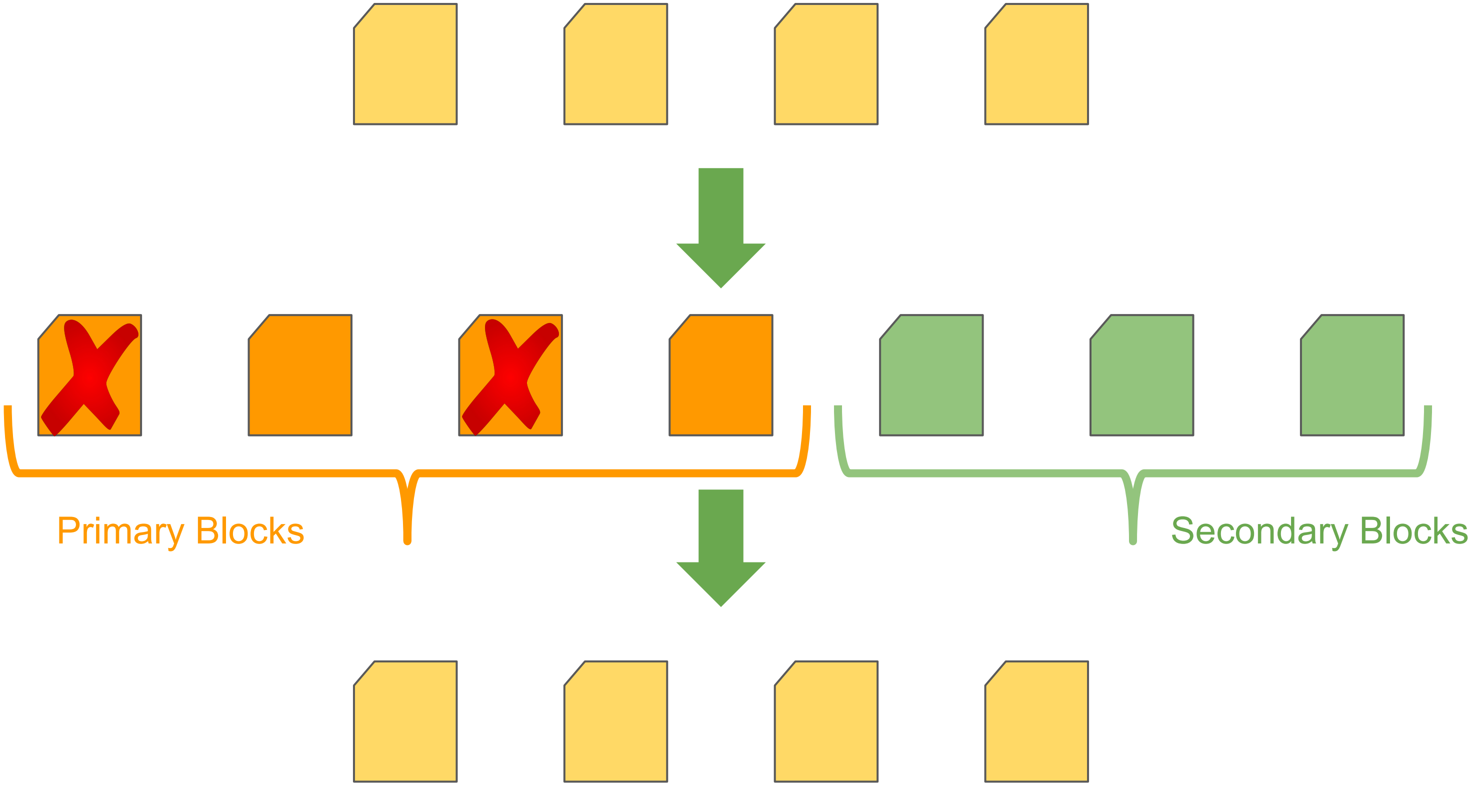
Build a highly flexible storage backend suitable for the cloud.

- Similar semantics to existing backend (RAID, etc.)
- Variable redundancy
- Temporally variable redundancy
- Scalable
- Good performance

...Rest of The Talk

- Recap on erasure coding in storage
- Variable redundancy blocks
- Adopting variable redundancy to flash array
- Prototype implementation
- Future work

Erasure Coding

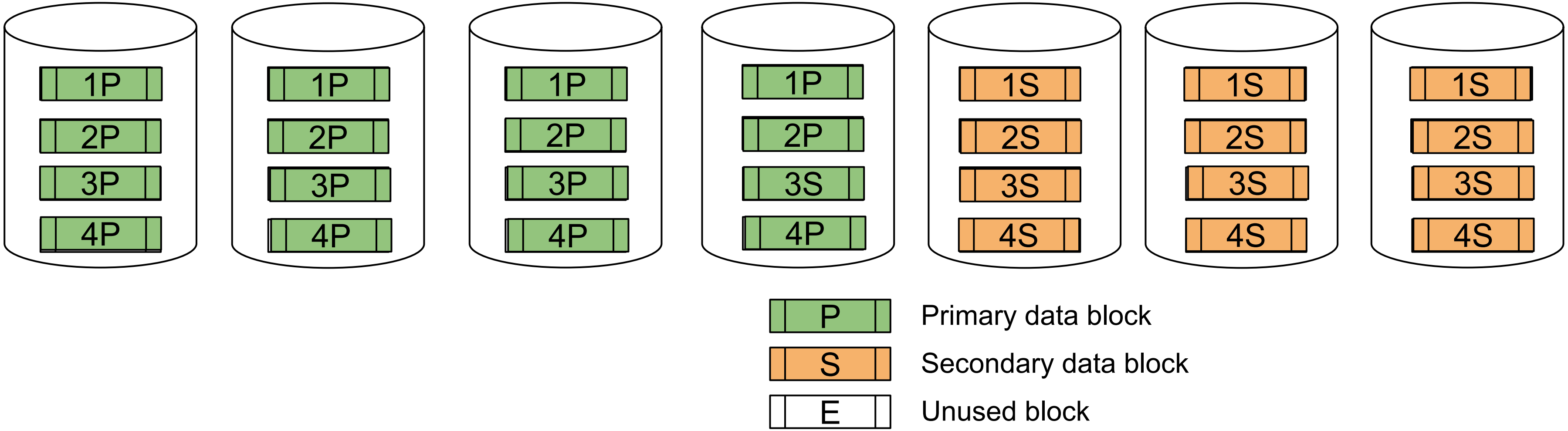


k = 4 Blocks

m = 7 Blocks

Erasure Coding with Constant Scheme

Scheme: 4 primary + 3 secondary



Constant Redundancy Distribution

- Translation from *Virtual Block numbers* to *Physical Block numbers* is very straightforward
- Virtual to Physical address map can be computed statically
- Examples of systems using this technique at file level and less commonly at block level.

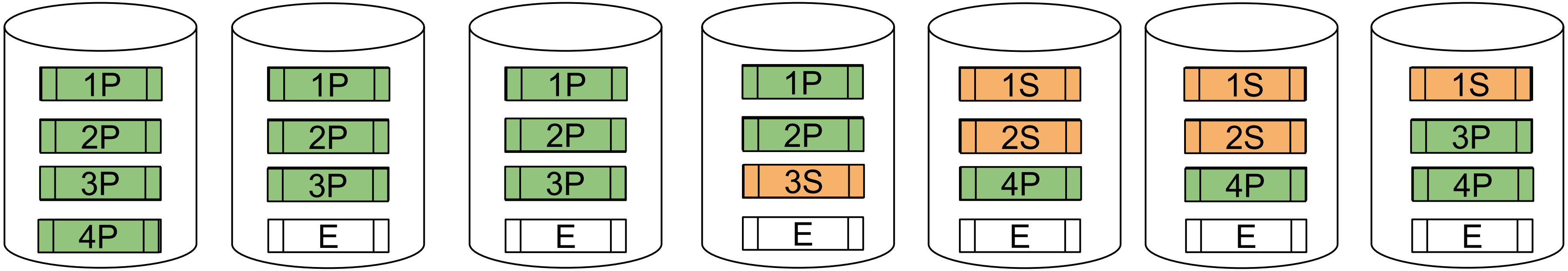
Variable Redundancy Distribution

1st Block: 4 + 3

2nd Block: 4 + 2

3rd Block: 4 + 1

4th Block: 4 + 0



P Primary data block

S Secondary data block

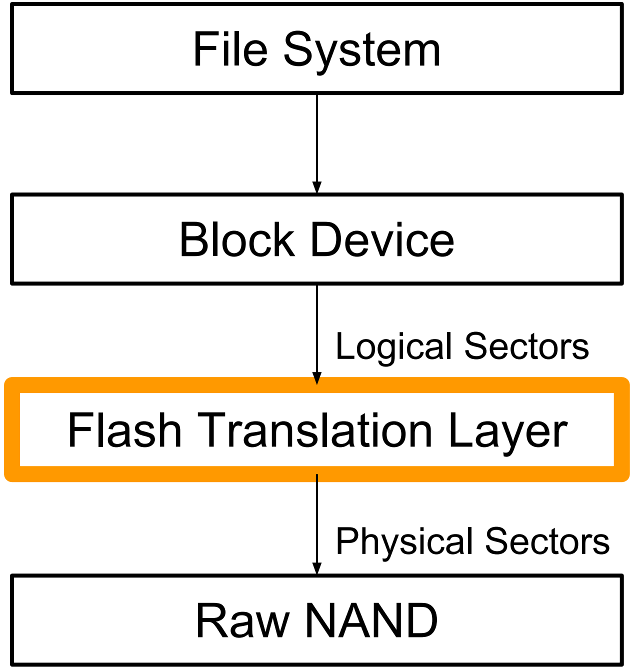
E Unused block

Variable Redundancy Distribution

- Features
  - Allows heterogenous data to be stored with different redundancy
  - Maximum freedom to configure granularity of redundancy
  - Erasure coding allows failure domains can span across multiple servers
  - Flexibility with reconstruction times
- Challenges
  - Flexibility comes at an extra cost of a lookup or address translation
  - Additional storage overhead for metadata associated with mapping
  - Block allocations have to carefully managed
- How do we mitigate the extra overhead?

Rethinking Variable Redundancy for Flash

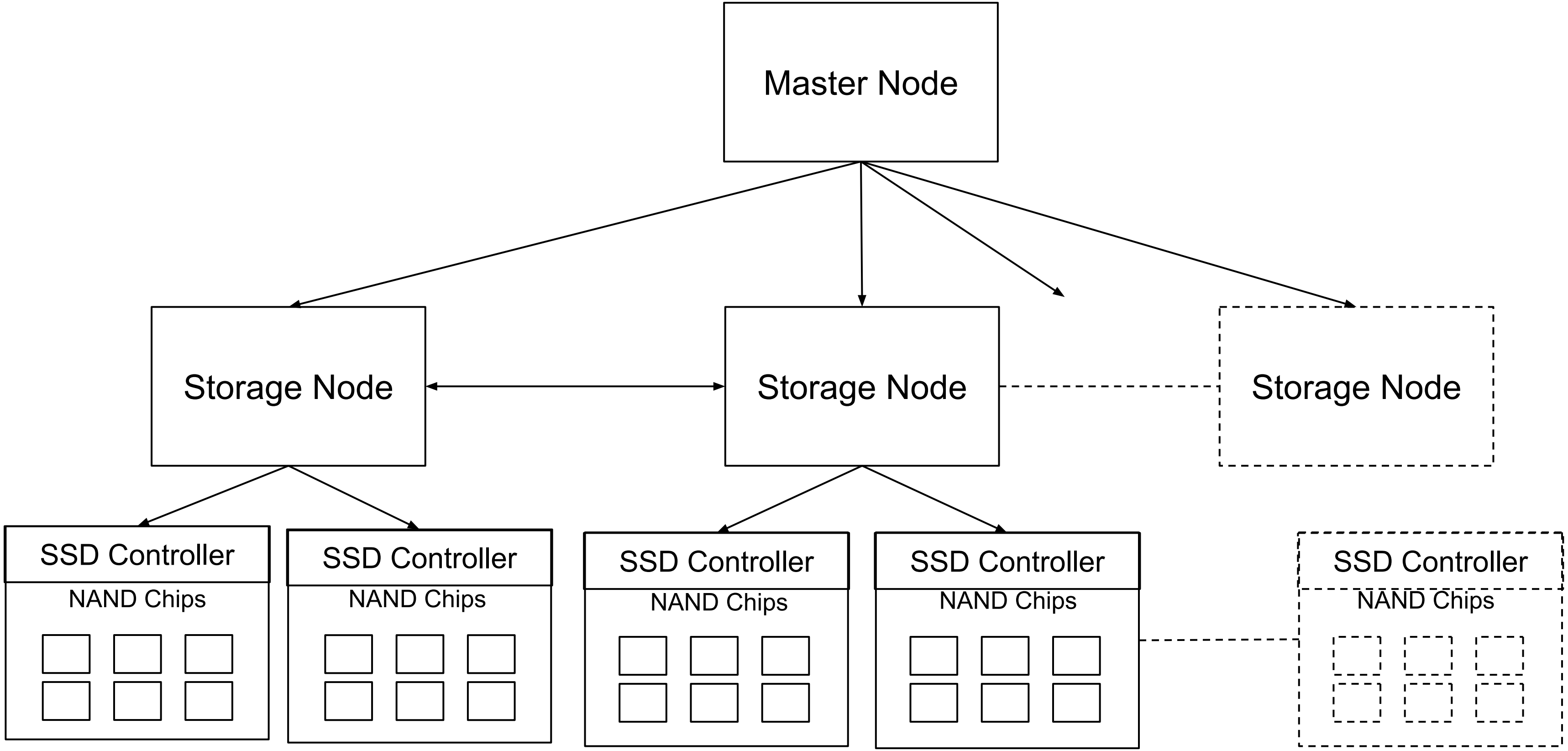
- Flash Translation Layer
  - Address translation
    - Block allocation
  - Garbage collection
  - Wear-Leveling
- Blocks are converted from Virtual block to physical blocks



Case Study: Application Managed Flash

- SSD based arrays v/s HDD based arrays
- FTL: Highly restrictive, unpredictable
- Application managed FTL
  - Combining with deduplication, etc.
  - Higher control
- Challenges: Extracting internal parallelism
  - NAND chips, dies, channels, planes
- Cider with FTL

Case Study: Globally Managed Flash Array



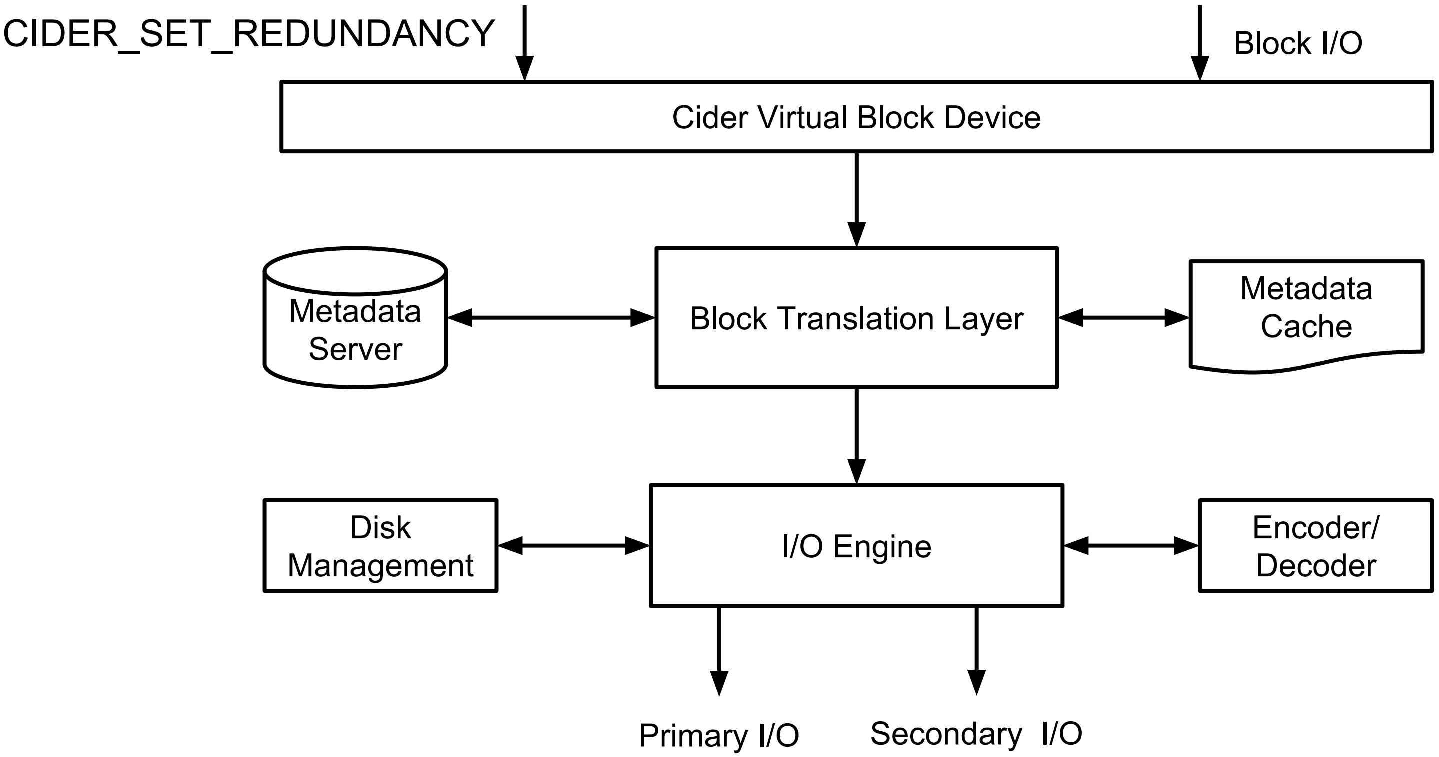
Use Case Study: SSD Simulator

- Modified FlashSim to enable a globally managed FTL
- 5 SSDs in a RAID Class
- Local translation, garbage collection and block allocation is disabled
- Global FTL as pluggable module

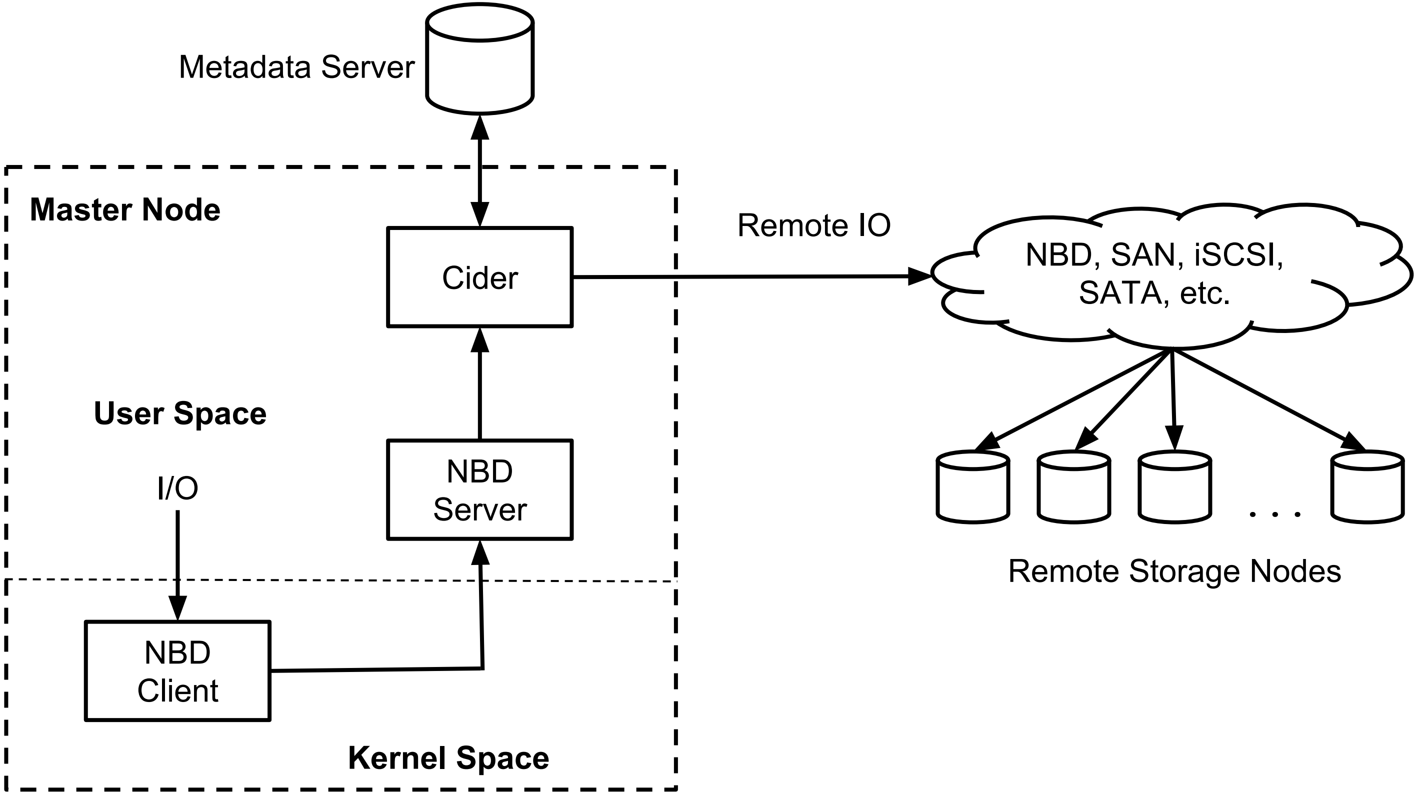
Cider: Architecture

- Virtual Block Device: Backed up off shelf physical storage devices
- `ioctl(CIDER_SET_REDUNDANCY, redundancy)`
- Writes
    - Uses the previously set redundancy level
    - ioctl/write should be automatically called to ensure correct redundancy
    - Changing redundancy through write with no data
- Reads
    - Uses the previously saved metadata
    - Parallel reads from backing disks

Cider Architecture



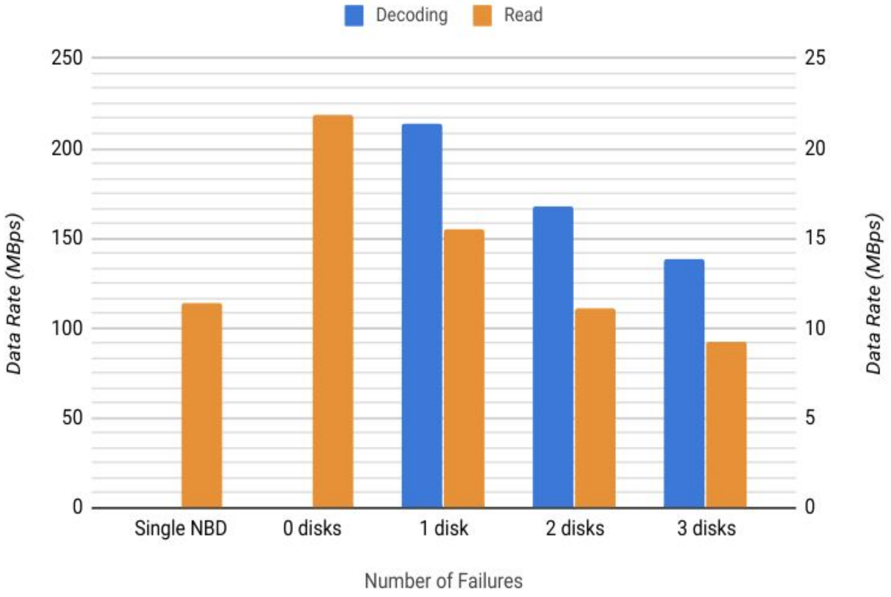
Prototype Implementation

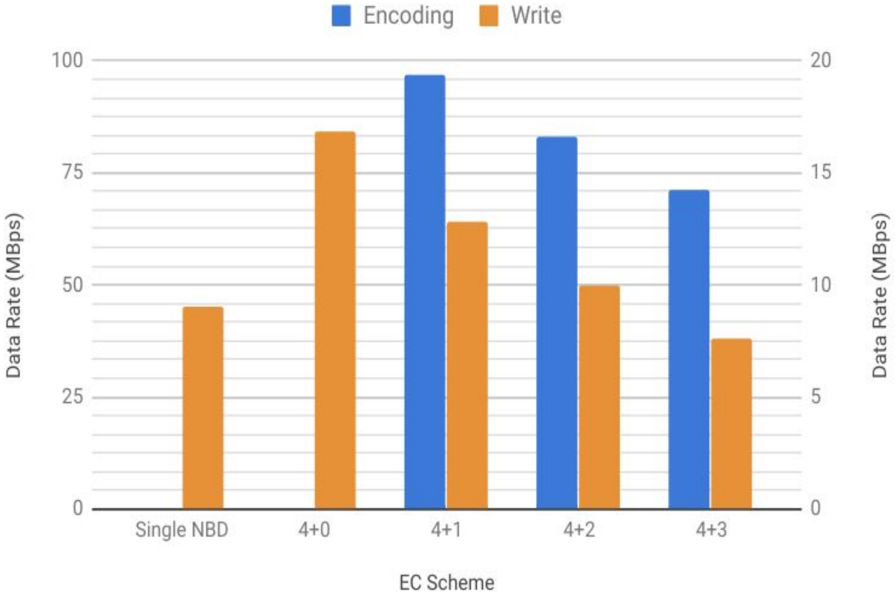


Evaluation: Preliminary Results

- Microbenchmarks on off-the-shelf hardware:
- 7 hard disks connected through NBD
- Writes throughputs with different EC schemes
- Reads throughputs with different failures
- Raw NBD device as a base performance

Microbenchmarks: Preliminary Results





Discussions and Future Work

- Disk recovery and handling failures
- Disk defragmentation
- Extracting full parallelism

Related Work

- Distributed and Block storage
  - Petal, Blizzard, Network RAID
- Erasure Coding
  - Placement of chunks
  - Faster computations
- Flash and FTL
  - MinFlash, NoFTL

University at Buffalo

Conclusion

- Classify and store data based on their criticality
- Limitations of existing solutions
- Cider as an alternative for a block level

Thank You!

A Case for Block Level Variable Redundancy on a Distributed Flash Array

(Patent Pending PCT/US2015/026267)

Presented at The 26th International Conference on Computer Communications and Networks (ICCCN 2017)
July 31 -August 3, 2017, Vancouver, Canada

Sharath Chandrashekhara
sc296@buffalo.edu
https://cse.buffalo.edu/~sc296/cider/