Building Native Erasure Coding Support in HDFS

Zhe Zhang[+], Kai Zheng[+], Bo Li[+], Andrew Wang[+], Vinayakumar B[+], Uma Gangumalla[+], Todd Lipcon[+], Yi Liu[+], Weihua Jiang[+], Aaron Myers[+] & Silvius Rus[+]

[+]Cloudera, [*]Intel, zhezhang@cloudera.com, kai.zheng@intel.com

Problem Statement

Benefits of triplication

- Fault tolerance
- Better locality
- Load balancing

200% overhead

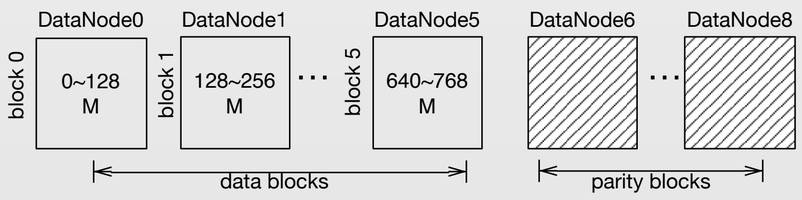
Secondary replicas rarely accessed

Erasure coding?

- Same or better fault tolerance
- < 50% overhead in a typical setup

Data Layouts

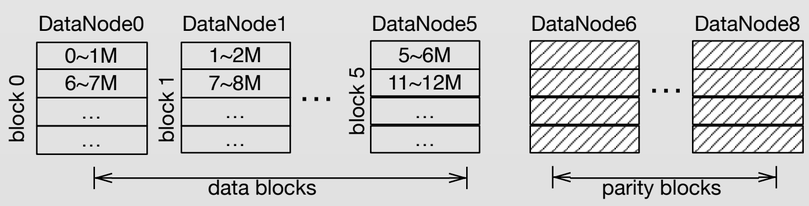
Contiguous



Good compatibility with locality-sensitive applications

Poor handling of small files

Striping



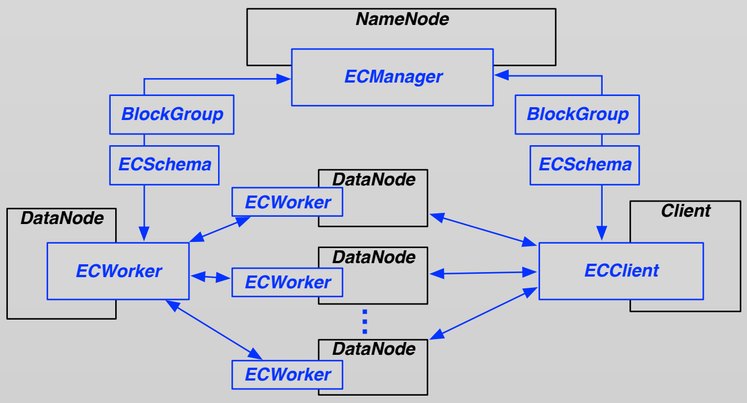
Improved I/O performance with high speed networking

Heavier memory and CPU overhead on NameNode

	Replication	Erasure Coding
Striping	Ceph (before firefly) Lustre	Ceph (optional w/ firefly) QFS
Contiguous	HDFS	Facebook f4 Azure

HDFS-EC aims to enable all 4 forms to support heterogenous workloads

HDFS-EC Architecture



BlockGroup: data and parity blocks in an erasure coding group

ECSchema: e.g., 6 data + 3 parity blocks, with Reed-Solomon

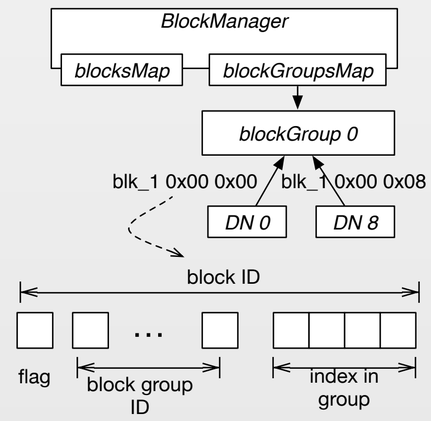
ECManager: group allocation, placement, monitoring

ECWorker/ECClient: codec calculation and striped read/write logics

Unique Research Challenges

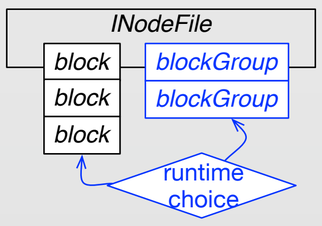
Reduce NameNode overhead

- Hierarchical block naming protocol
- Fixed placement groups
- Peer monitoring and recovery in a group

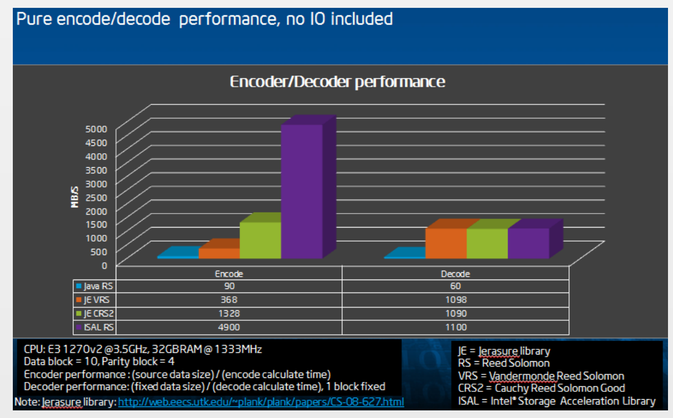


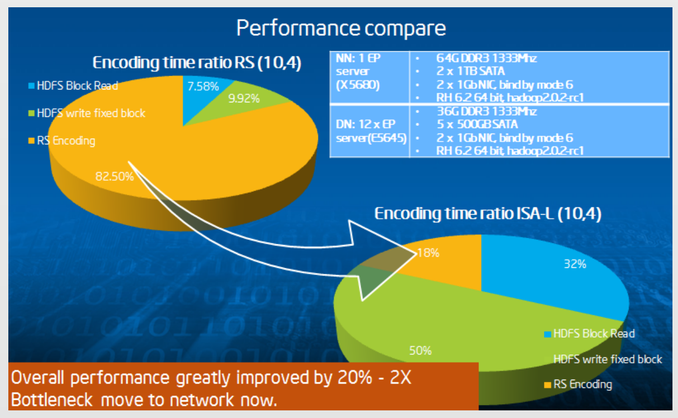
Preserve data locality

- *Hybrid* storage forms for individual files



Faster codec calculation





Preliminary Results

File categorization

- Assuming (6,3) coding schema
- Small files: < 1 block,
- Medium files: 1~6 blocks
- Large files: > 6 blocks (1 group)

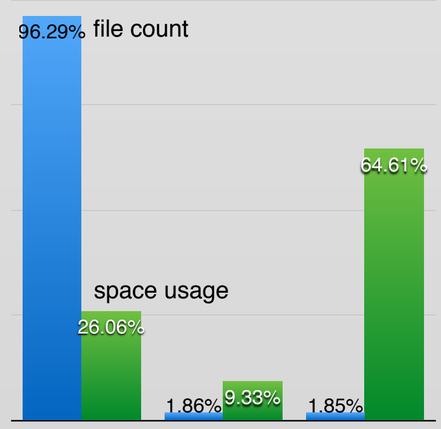
Storage usage simulation

- *Contiguous* skips a file if parity data is larger than secondary replicas

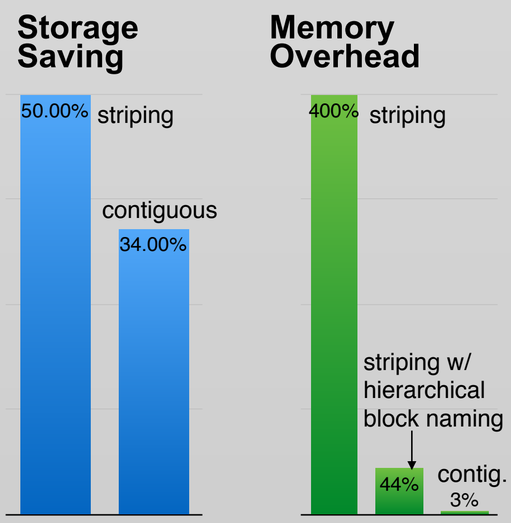
Memory usage calculation

- Each block uses ~78 bytes
- Each additional replica location uses ~16 bytes

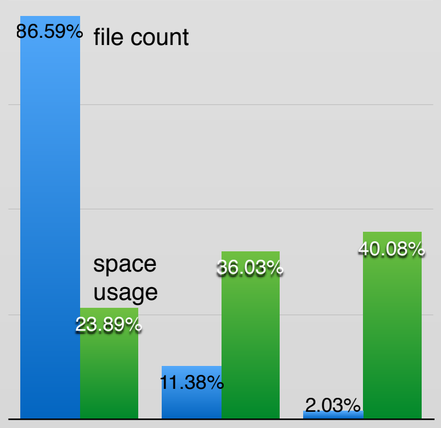
Cluster A Profile



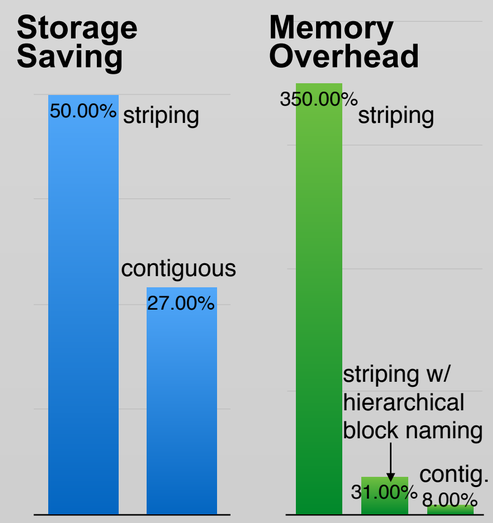
Top **2%** files occupy **~65%** space



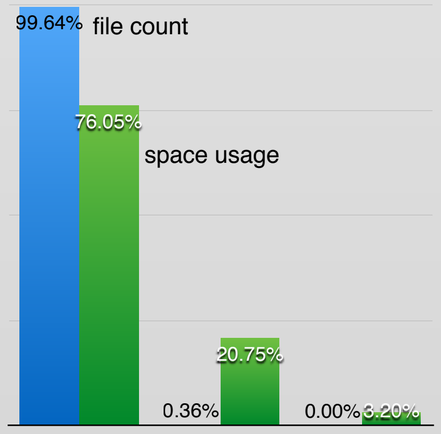
Cluster B Profile



Top **2%** files occupy **~40%** space



Cluster C Profile



Dominated by small files