Distributed Data Storage on CubeSat Clusters

iCubeSat

By Obulapathi Challa, Dr. Janise McNair

Author reachable at: obulpathi@gmail.com, obulpathi@ufl.edu

Why distributed?

- Large amounts of sensor data:
  - Images of size 100 MB each
  - Remote sensing data of same size
- Small CubeSats:
  - 1 GHz Processing
  - 1 - 2 GB of RAM
  - Few 10's of GB of flash memory
  - 9.6 kbps downlink communication speed
- A single CubeSat cannot process / downlink 100's of MB of data in reasonable time
- Distributed is the way to go!

A distributed solution requires

1. Distributed sensing, storage, processing and communication
2. Distributed Processing: CubeSat MapReduce
3. Distributed Communication: CubeSat Torrent
4. Distributed Storage: ?
5. Distributed Sensing: ?

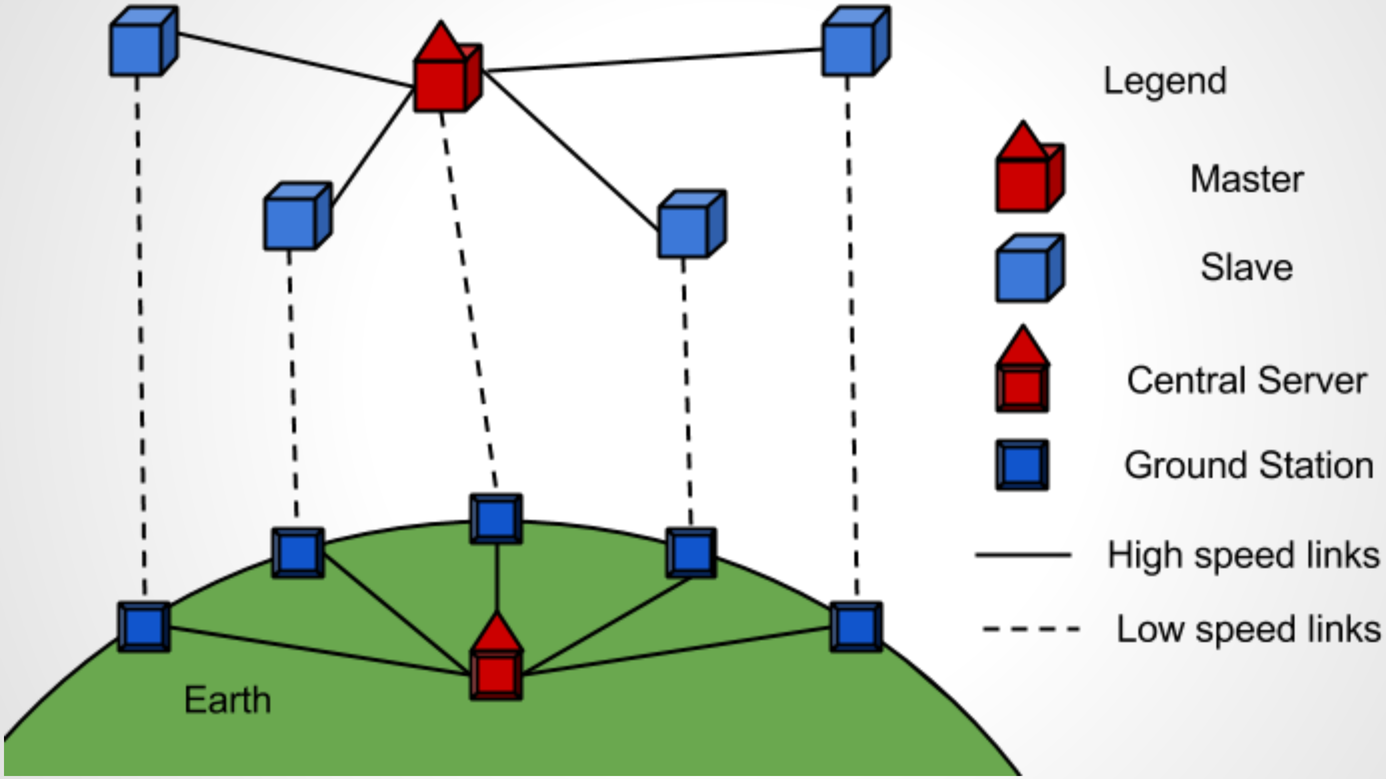
Existing solutions

1. GFS and HDFS: For storing large files in a distributed fashion on a cluster of nodes
2. BigTable and HBase: Non-relational, distributed database for storing large amounts of structured data on cluster of nodes
3. Pros: Scalable and well tested
4. Cons: Too complex for CubeSat clusters, require large amounts of bandwidth and power, scarce resources for CubeSats.

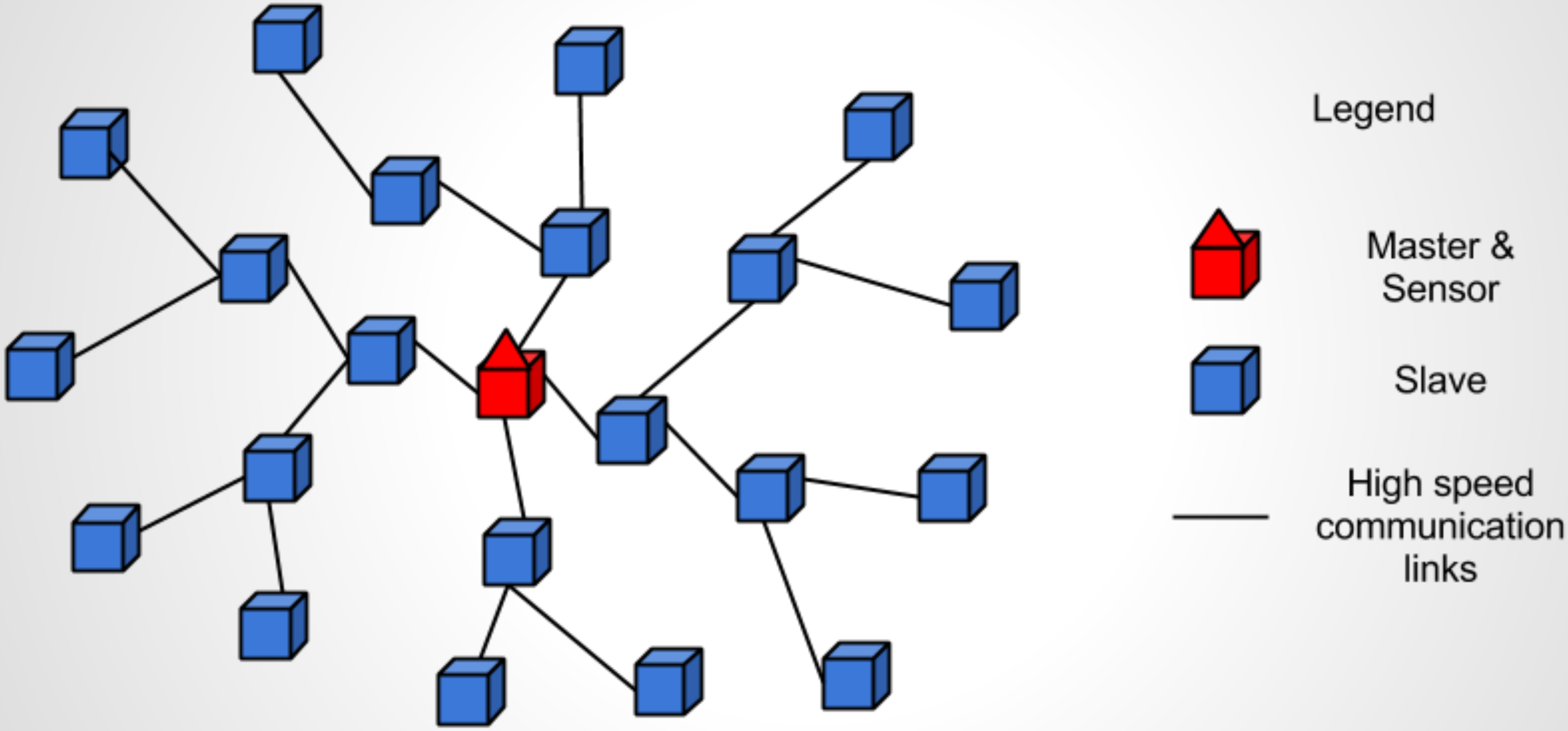
CubeSat Distributed File System

- Goal:
  - Distribute data to the CubeSats in a cluster
- Desirable properties:
  - Bandwidth and power efficient
  - Support distributed processing, communication and sensing
  - Scalability, performance and reliability

Network architecture



CubeSat cluster



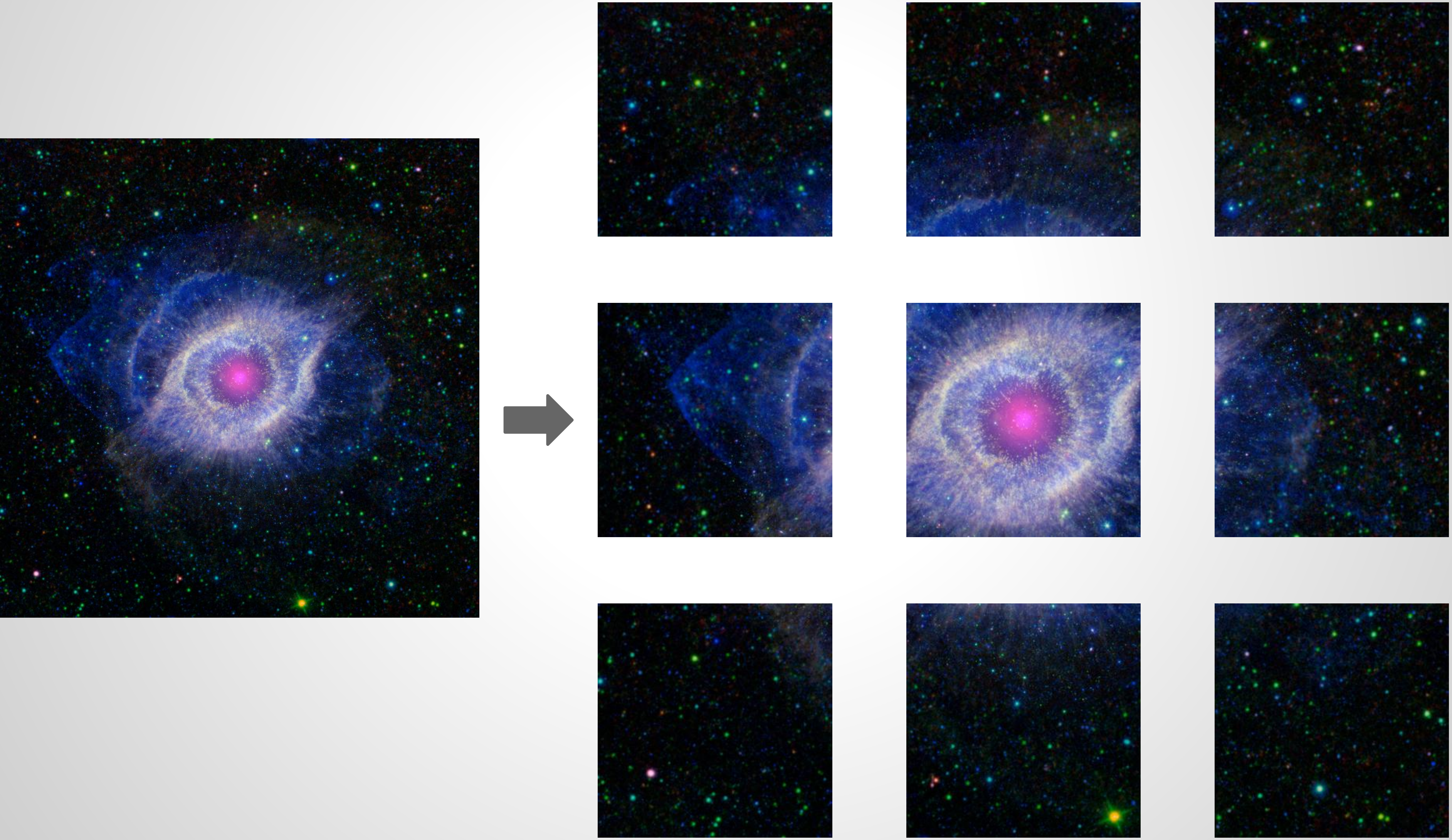
RelNAV

- RF based relative navigation sensor for CubeSat spacecraft
- Intersatellite Communication and timing synchronization

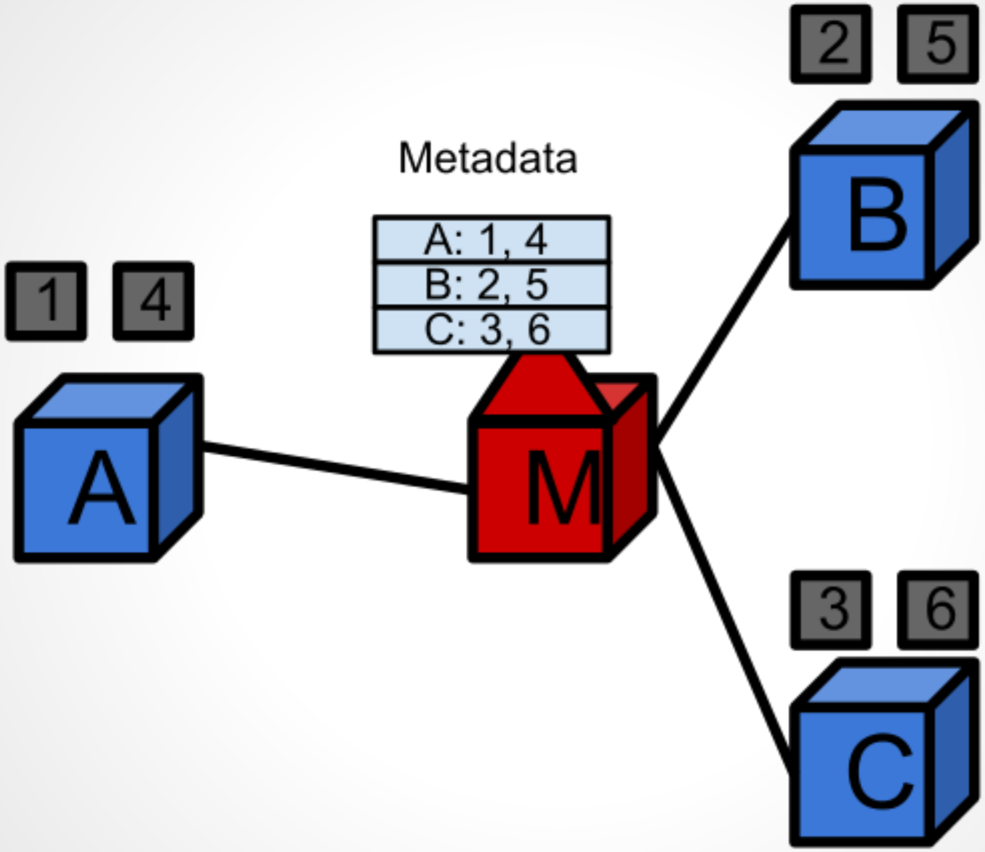
Capabilities

- ±1° altitude accuracy
- ±0.1m range accuracy
- ≤ 0.3 nanosec timing synchronization
- 0.5 to 10,000 meter operating range
- 10 Mbps data communications channel

Data partitioning



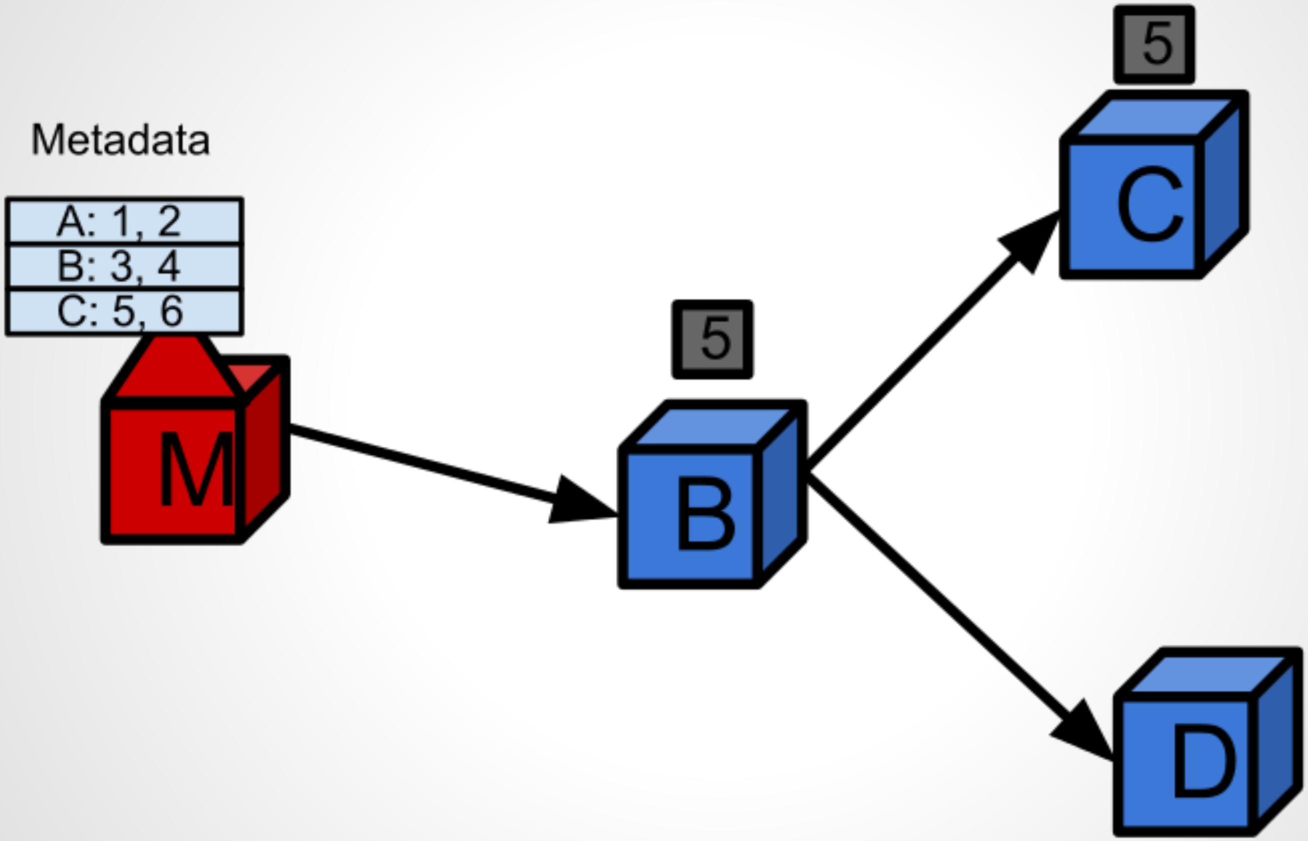
Data allocation



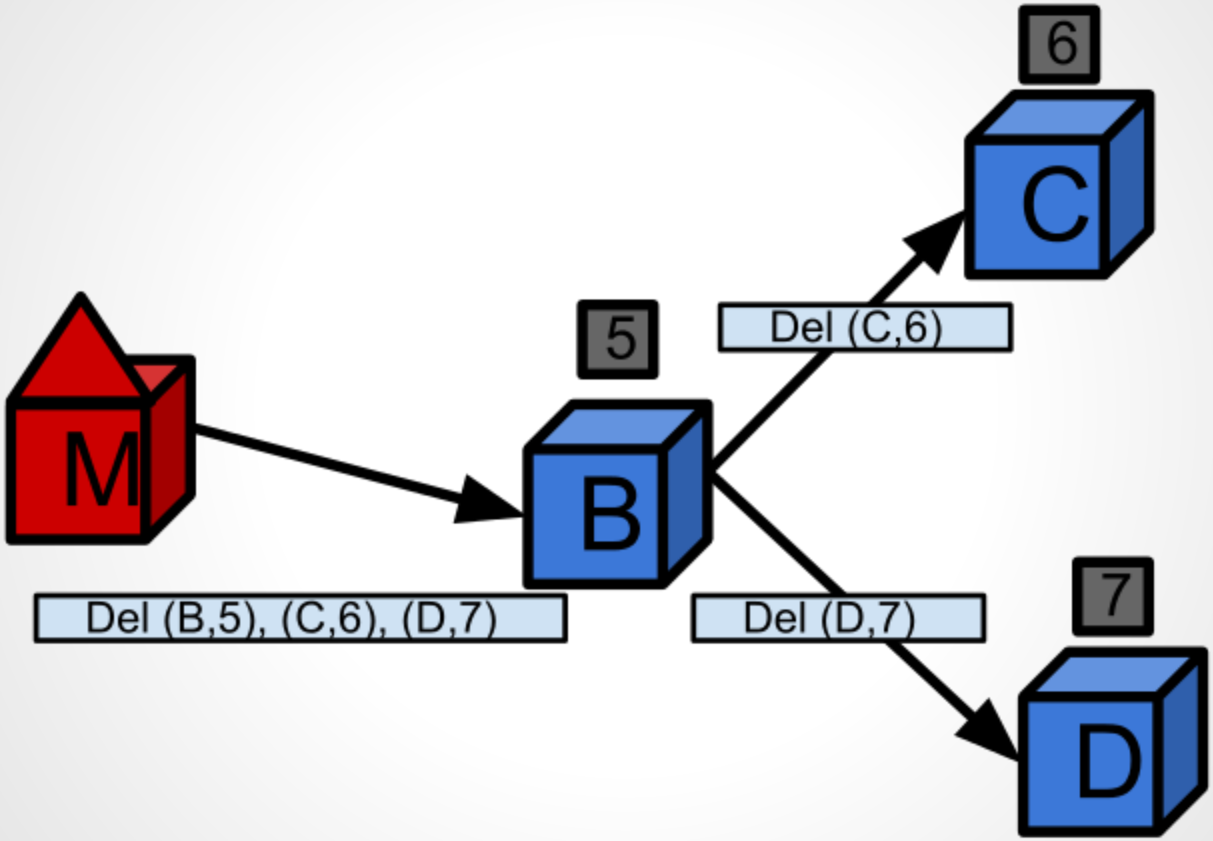
Metadata

A: 1, 4
B: 2, 5
C: 3, 6

Data distribution and replication



Message aggregation and tree routing

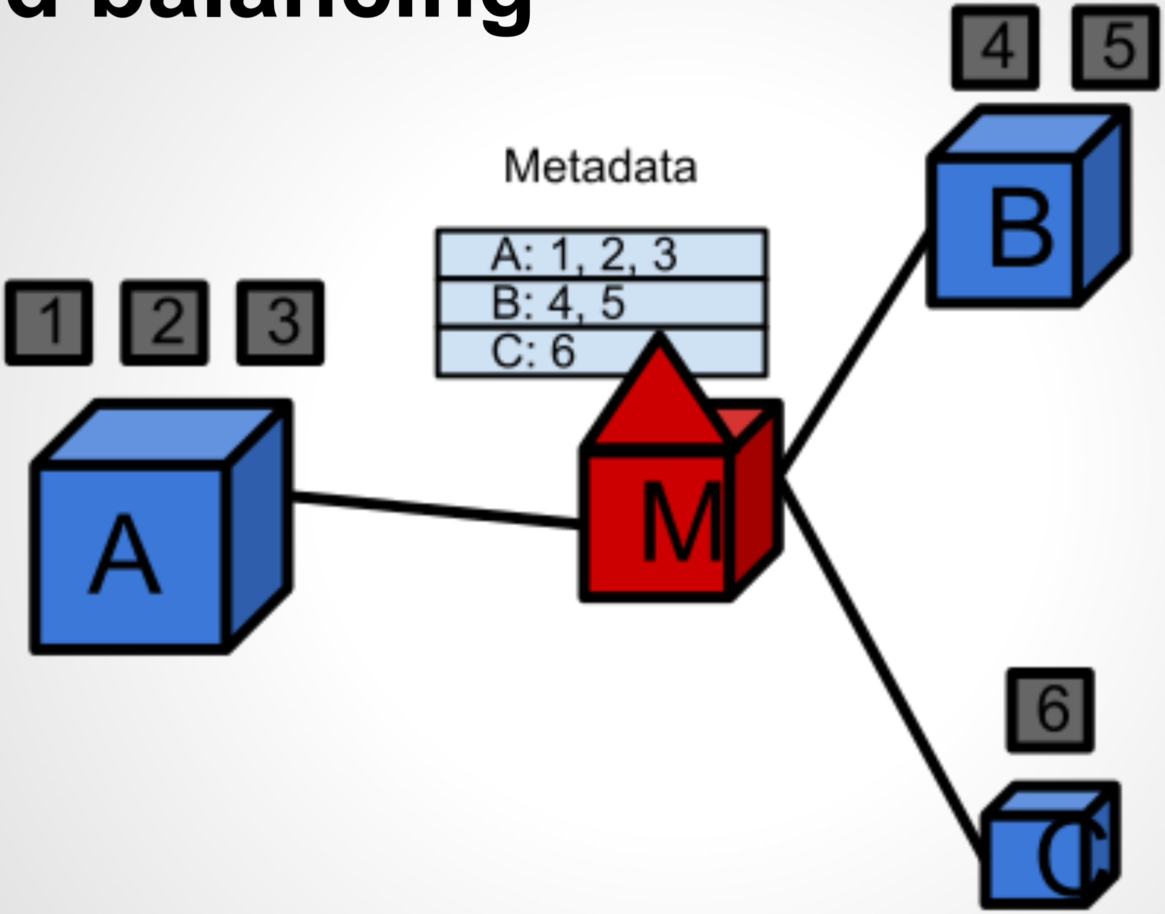


Del (B,5), (C,6), (D,7)

Del (C,6)

Del (D,7)

Load balancing



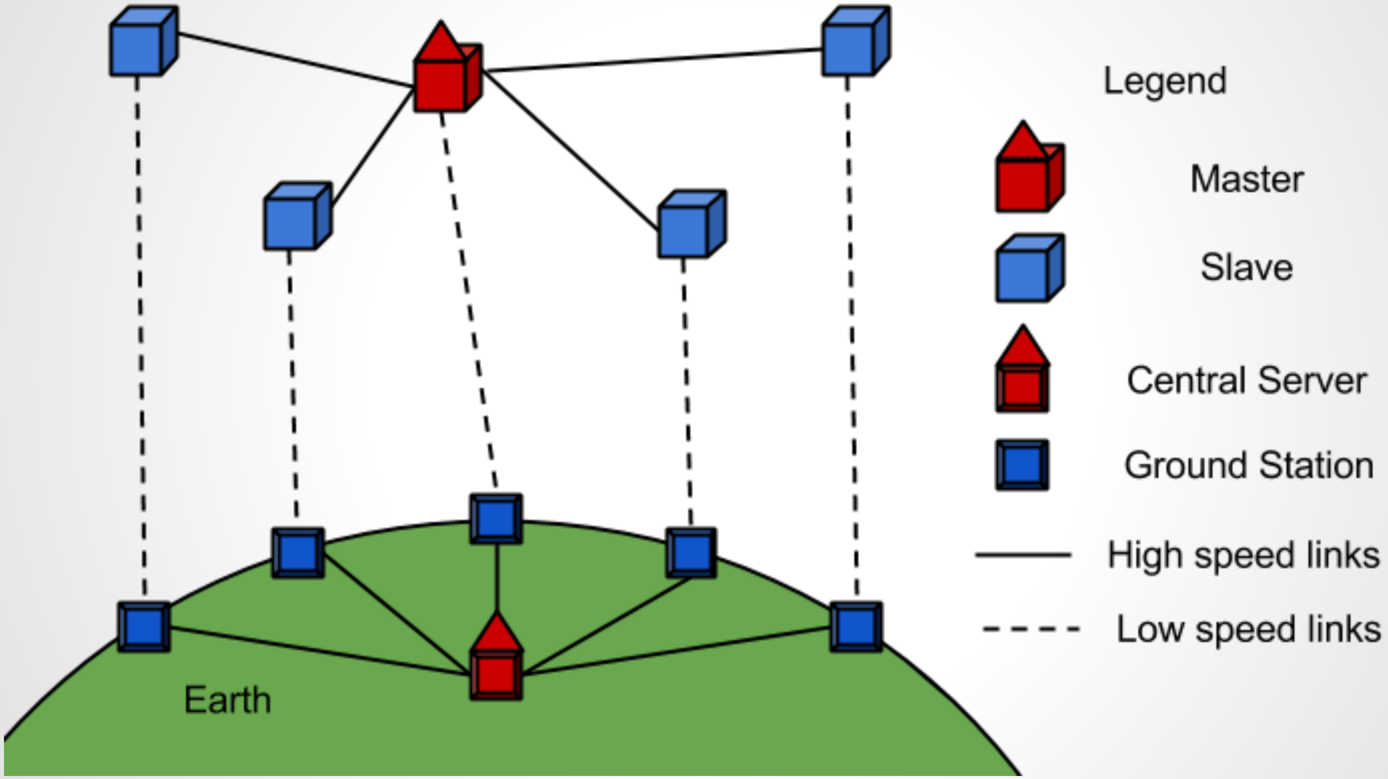
Metadata

A: 1, 2, 3
B: 4, 5
C: 6

Workflow and file operations

- Sensing: Master nodes performs sensing
- Local write: Stores the data to local file
- Create: Creates distributed file to store the sensor data
- Write (aka distribute): Distributes the data to the CubeSats in the Cluster
- Process or Downlink: Perform distributed processing or downlinking of the sensor data
- Delete: Delete the data

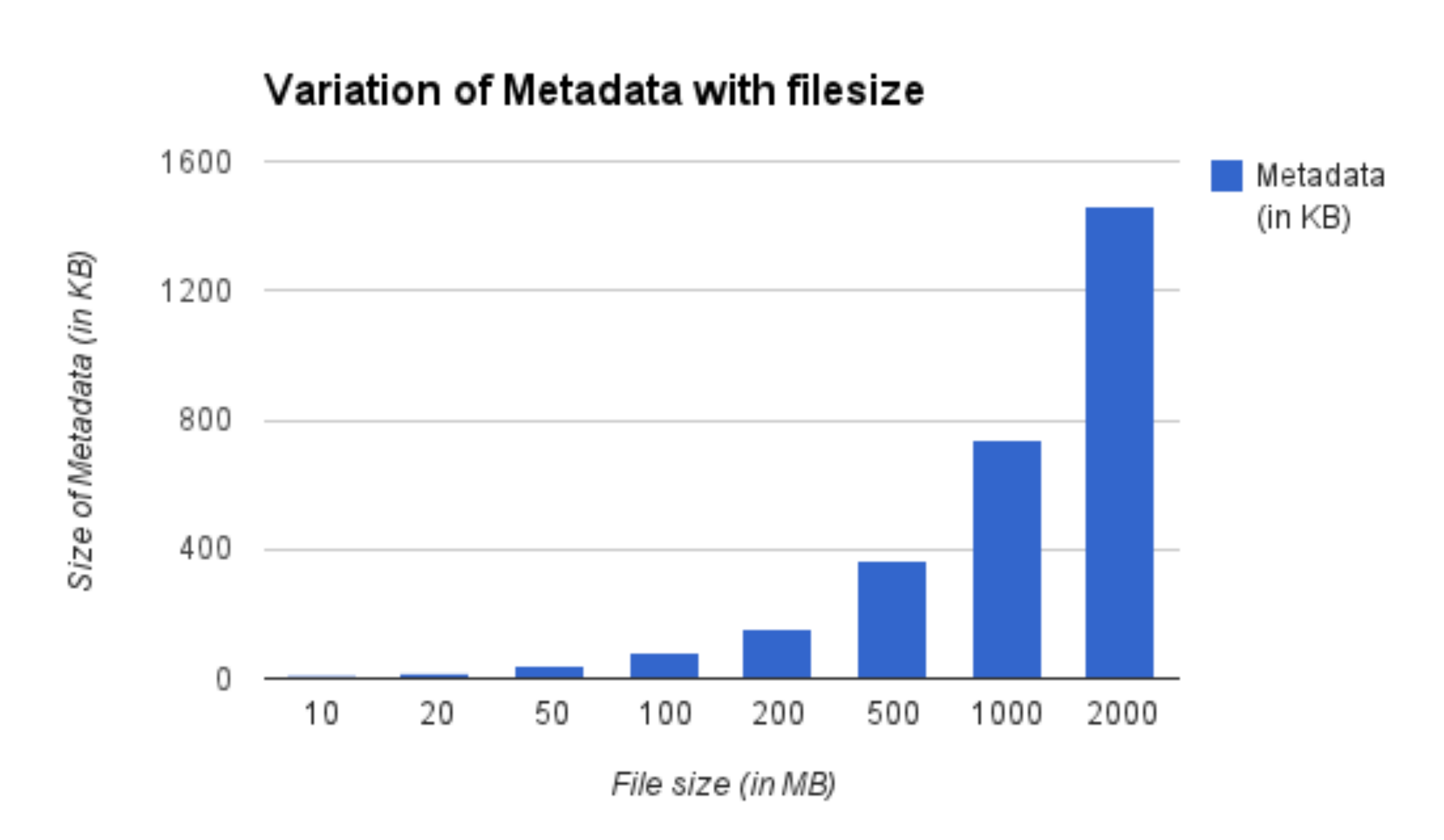
Network architecture



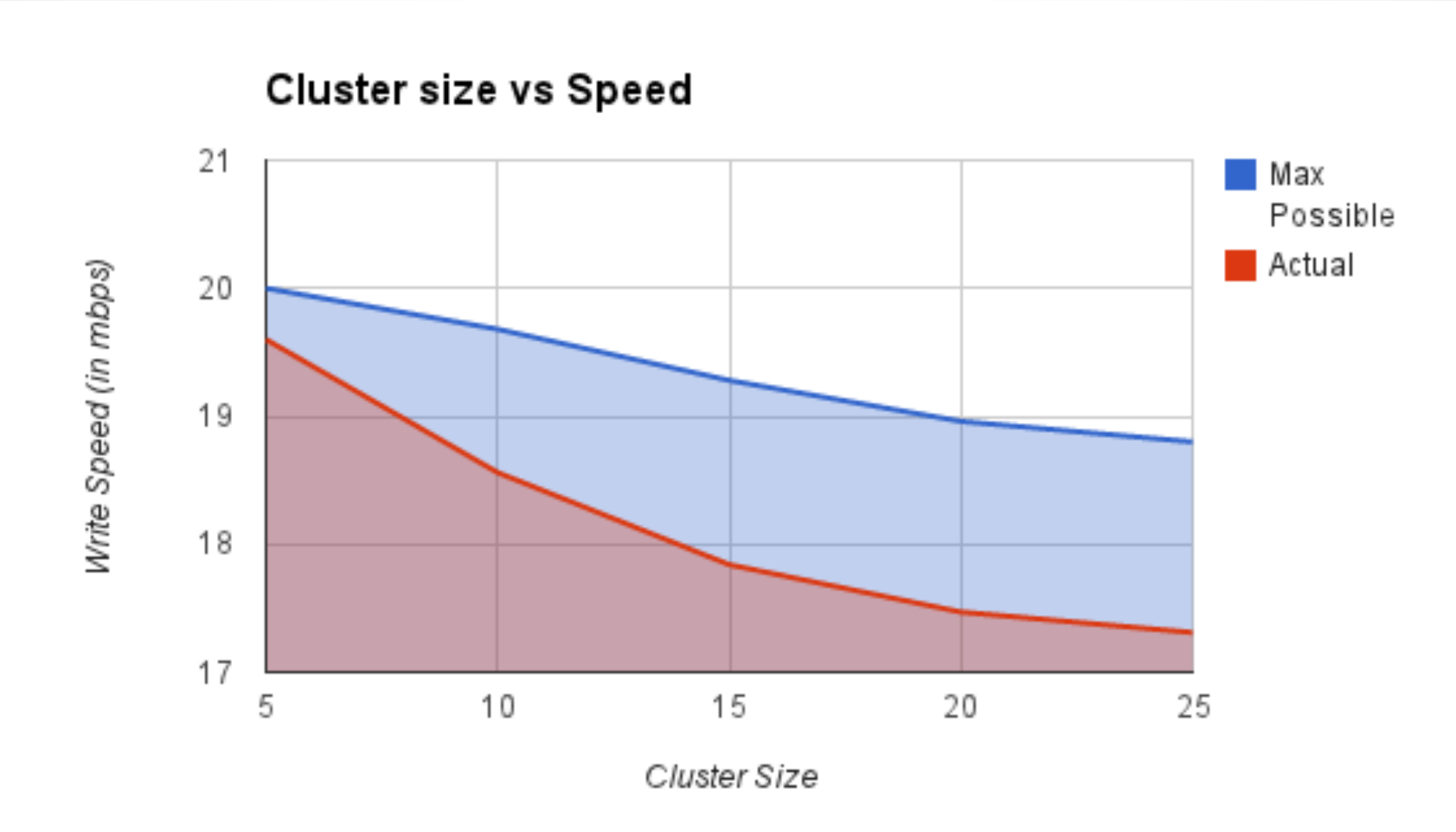
Simulation setup

- 4 - 24 slaves and 1 master
- 1 GHz processing power
- 1 GB of RAM
- Few 10's of GB of flash memory
- 10 Mbps inter cluster communication
- 9.6 kbps ground communication

Simulation results: Metadata



Simulation results: Read and write speed



Conclusions

- Very simple distributed storage system
- Tolerant to component and system failures
- High availability through replication of data
- Metadata is replicated at several nodes
- Processing and downlinking speedup of with 80% efficiency
- Enables storing large files on CubeSat Clusters.

Questions?