Spark on Ceph at UPSud/LAL

1. What Spark is about
2. Why Spark on Ceph?
3. Implementation ideas

1. What Spark is about

- Spark is a computing framework
  - Siminar to Hadoop MapReduce… from afar
- Many more use cases
  - Machine Learning, Bioinformatics, …
- Key concept : Resilient Distributed Dataset
  - Tries to fit the dataset into RAM

1. What Spark is about

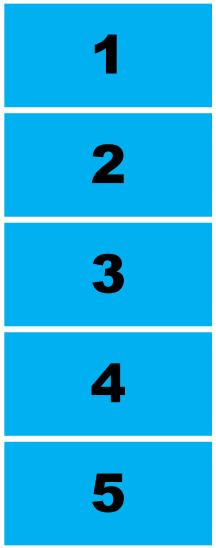
- Spark runs on a cluster
  - Uses YARN, MESOS, or standalone
- Reads from/writes to distributed filesystems
  - HDFS, S3, …
  - Not to Ceph (yet)
- Preferably uses HDFS
  - Data locality – but doesn't make sense in VMs
  - Uses rename on writes – possible problem

1. Experiments at UPSud

- Life Sciences
  - DNA/RNA Sequence alignment
  - Galaxy on Spark
  - Simulating turtle embryos growth
- Astrophysics
  - Image coaddition
  - Cross matching catalogs (CDS Strasbourg)

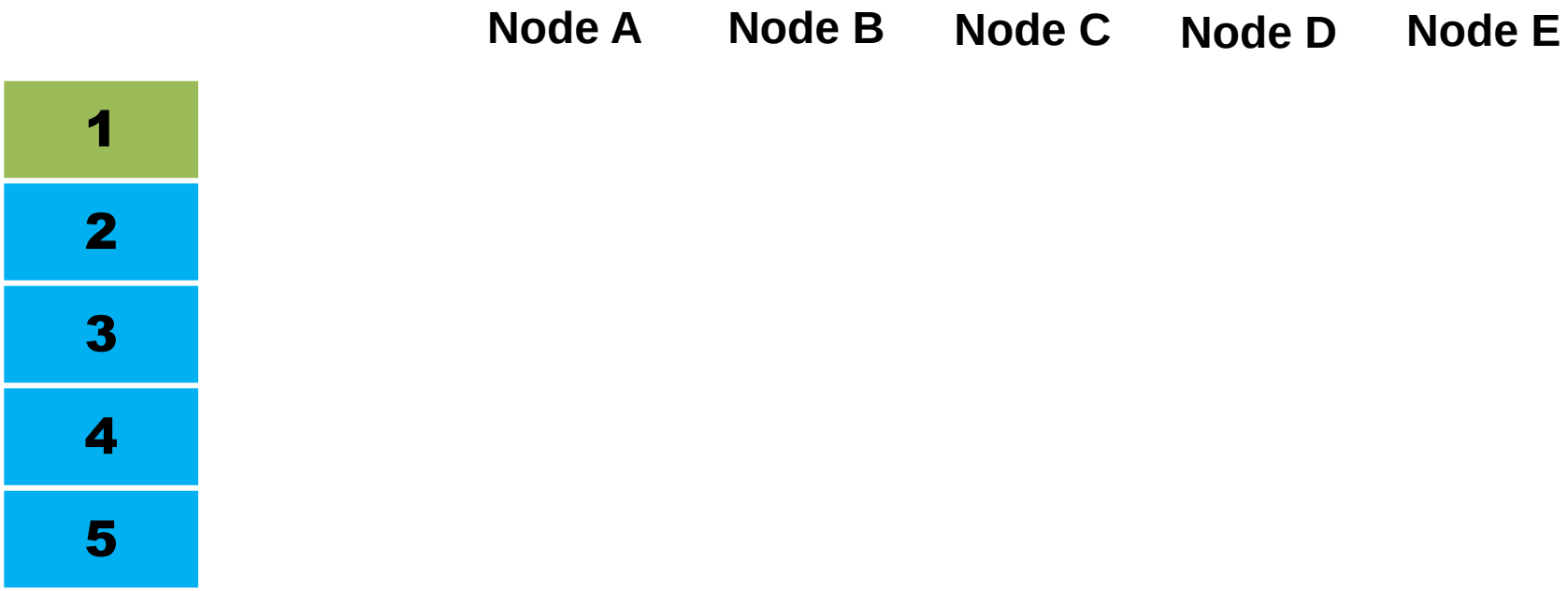
How HDFS works

1. Split files into blocks
   - Split on data structure boundaries (e.g. line)
   - Indicative size : 128MB



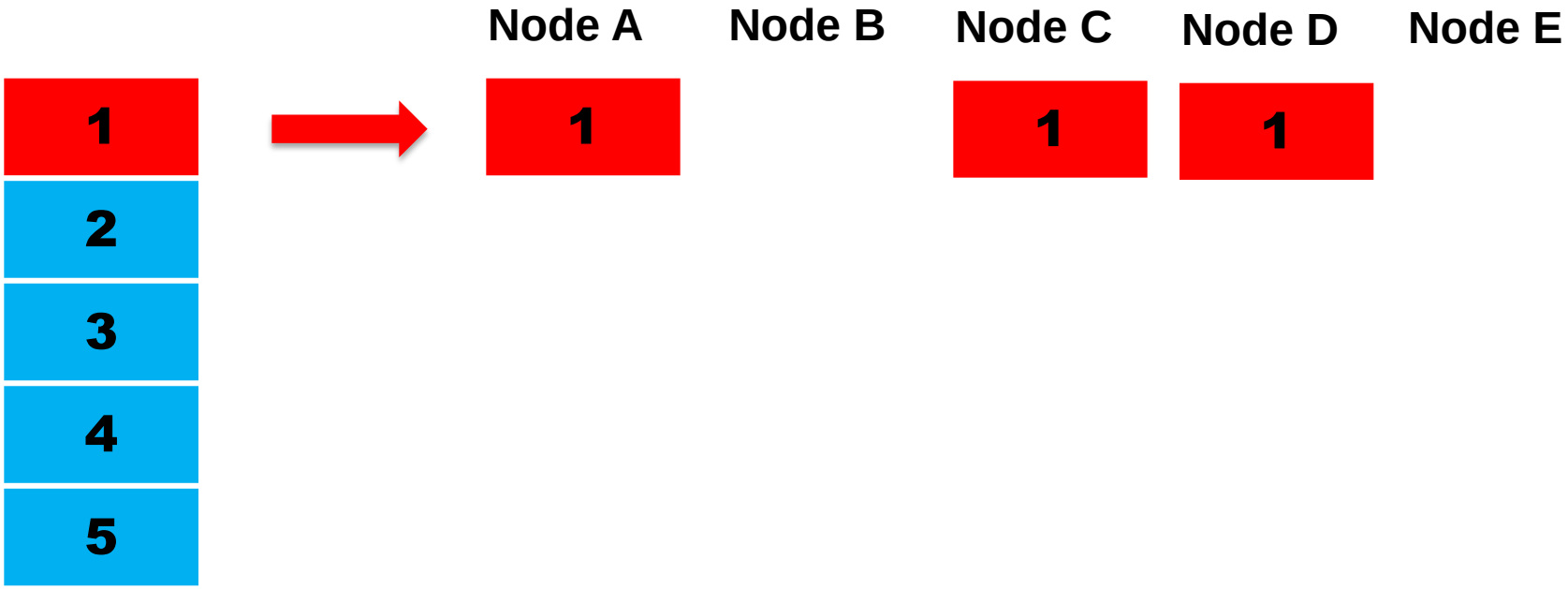
How HDFS Works

2. Copy each block on multiple nodes



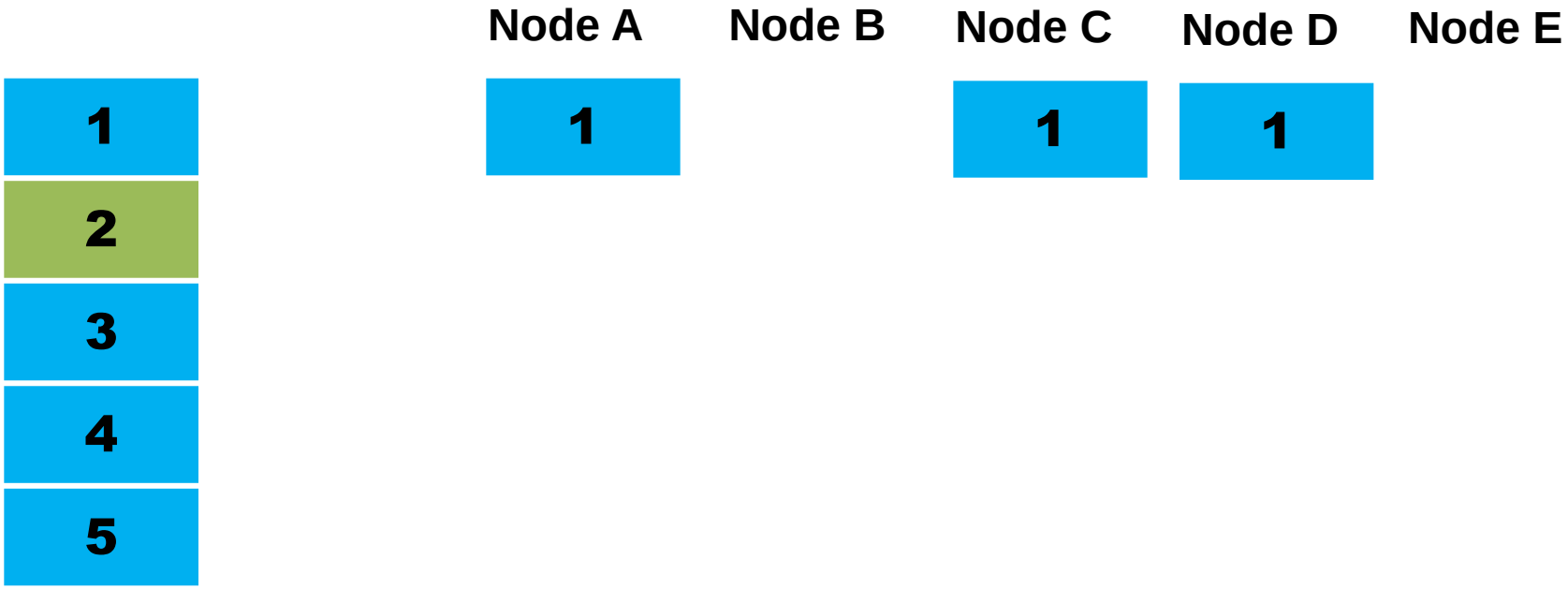
How HDFS Works

2. Copy each block on multiple nodes
   - In general, 3 copies



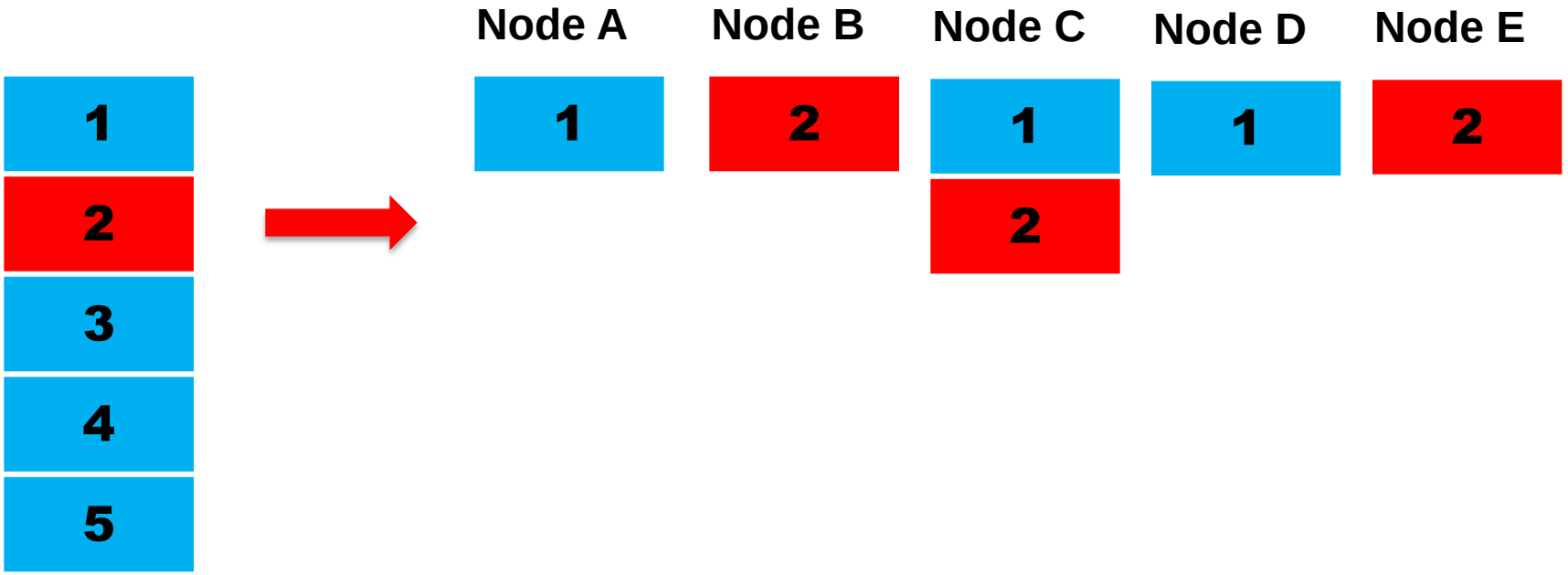
How HDFS Works

2. Copy each block on multiple nodes
   - In general, 3 copies



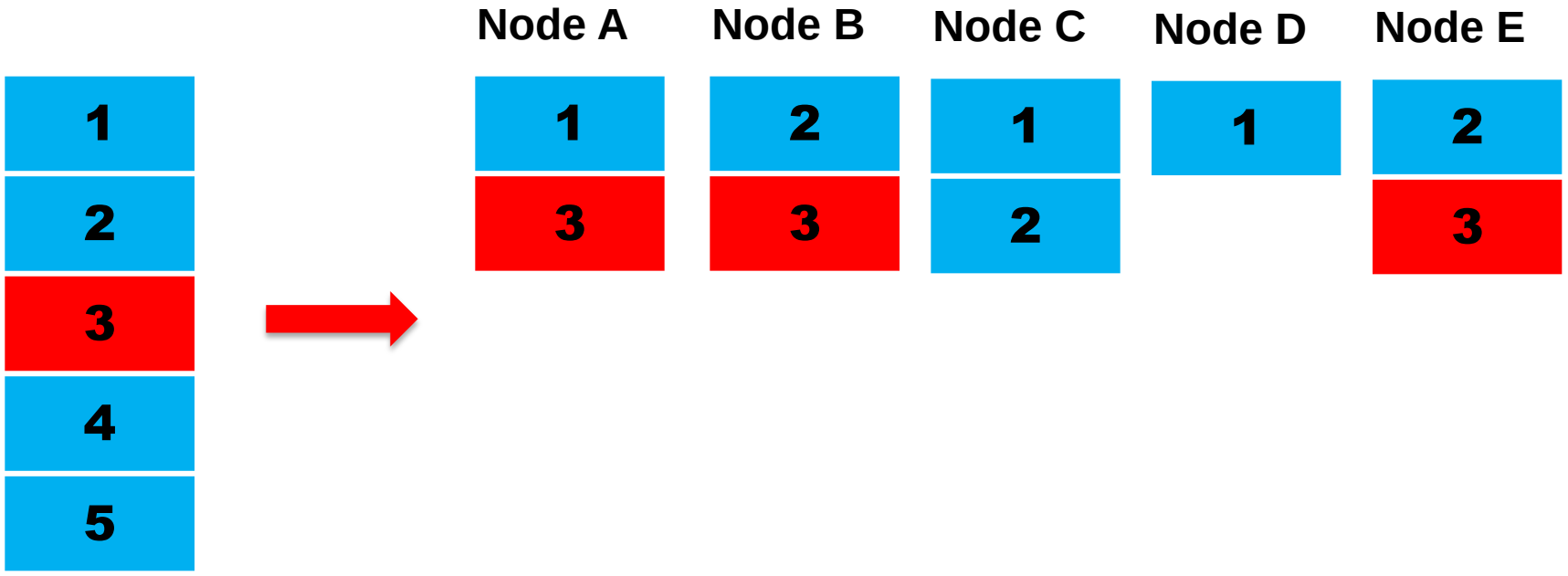
How HDFS Works

2. Copy each block on multiple nodes
   - In general, 3 copies



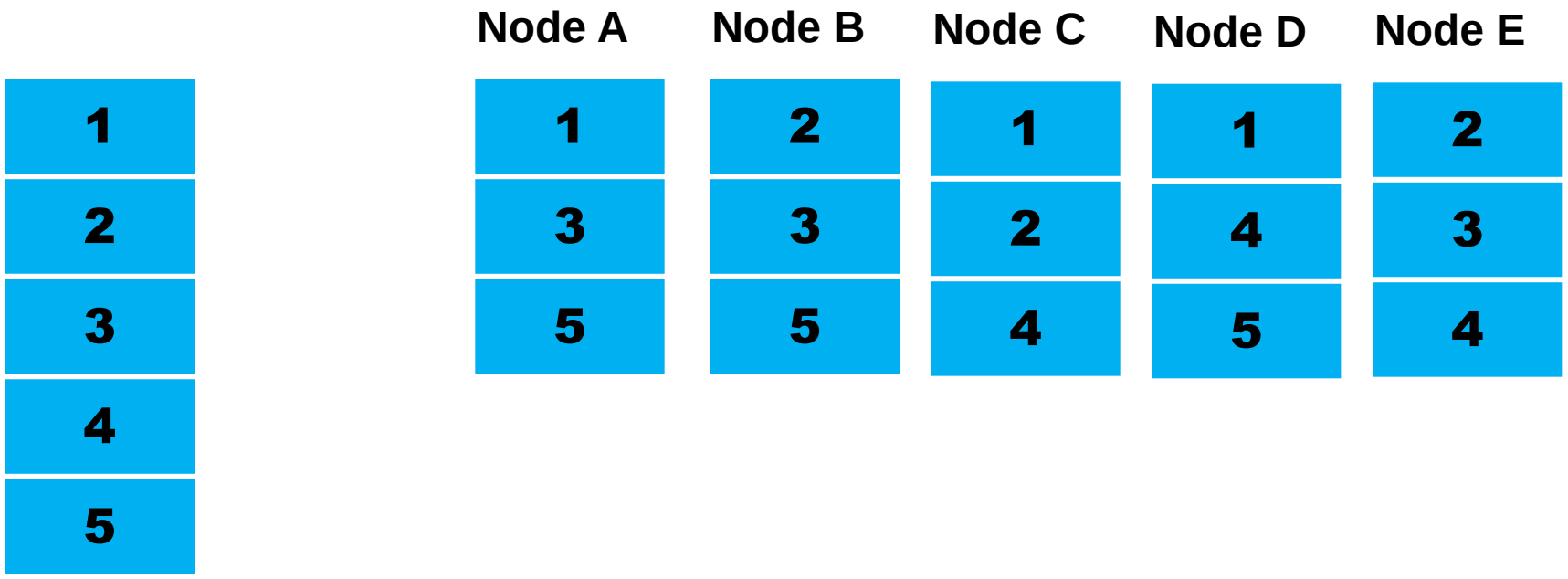
Fonctionnement de HDFS

2. Copy each block on multiple nodes
   - In general, 3 copies



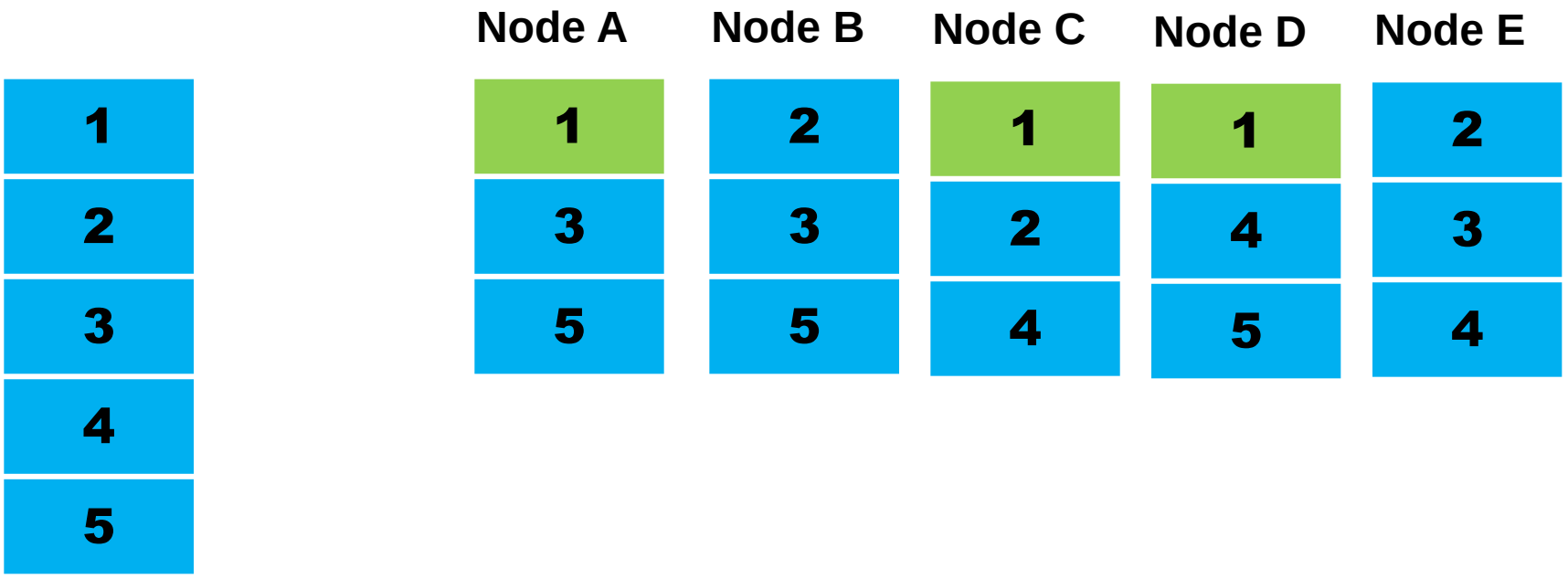
How HDFS Works

2. Copy each block on multiple nodes
   - In general, 3 copies



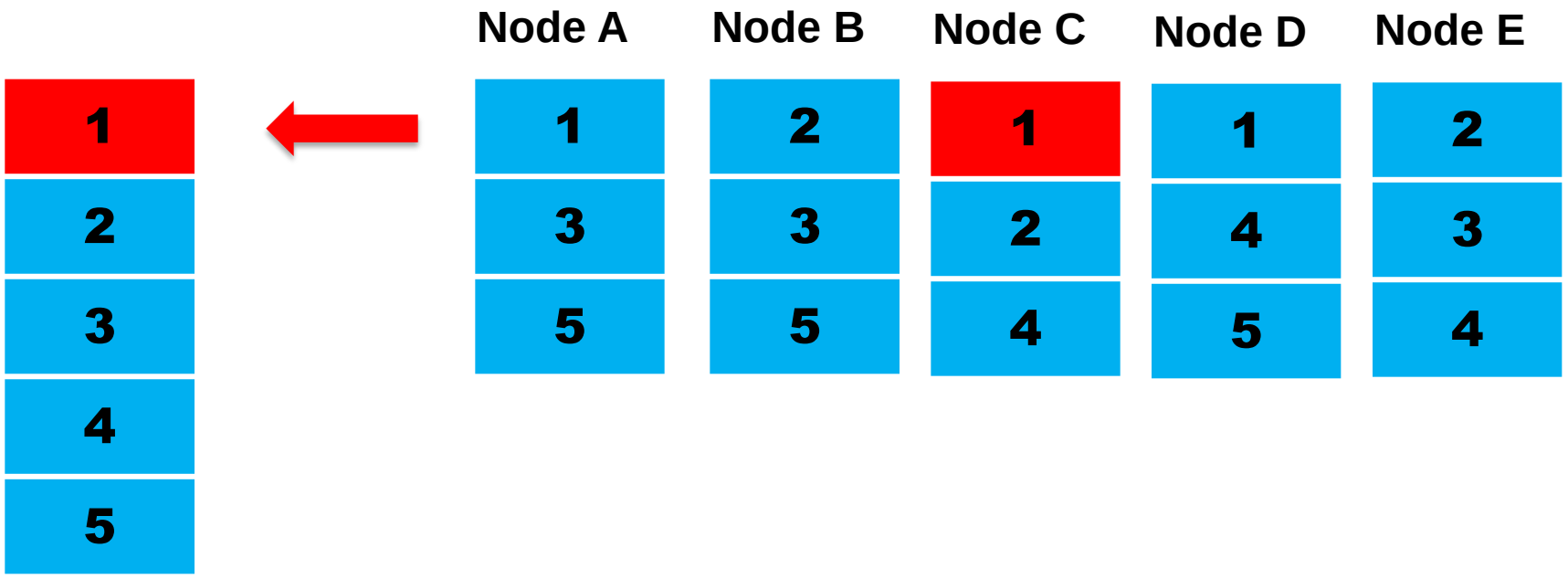
How MapReduce Works

1. Select nodes on which to run computations
   - Data has to be node-local (if possible)



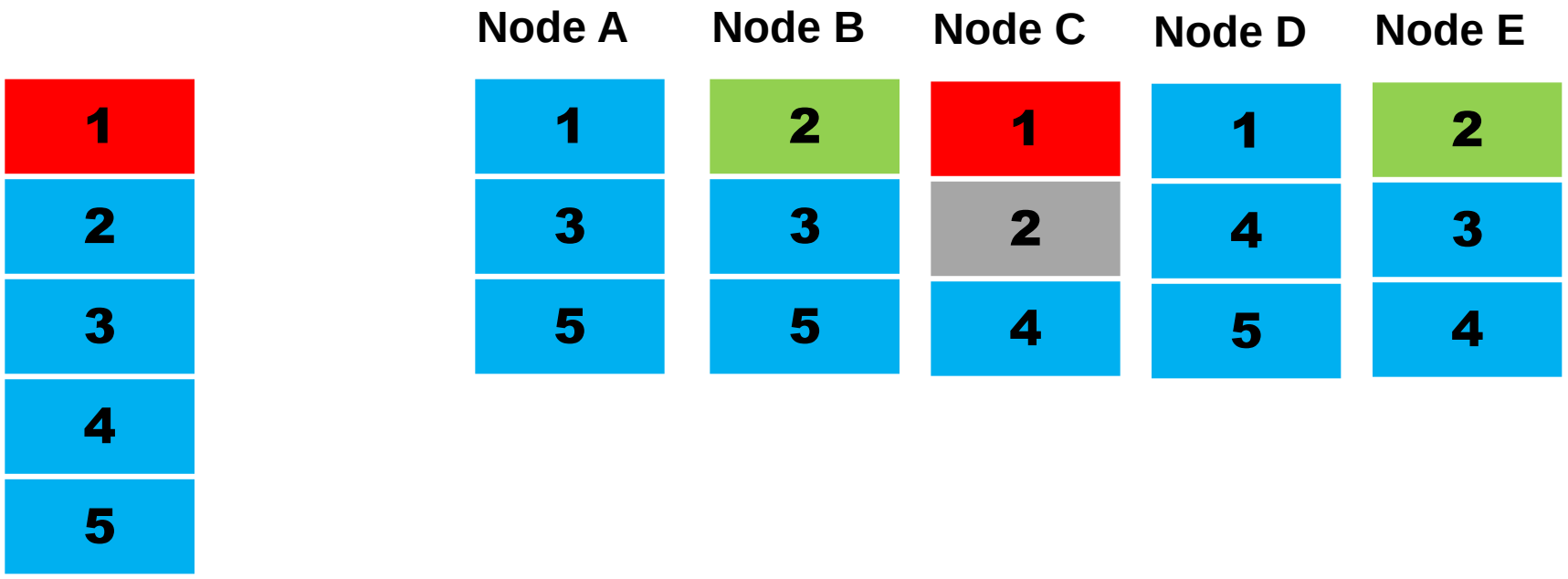
How MapReduce works

1. Select nodes on which to run computations
   - Data has to be node-local (if possible)



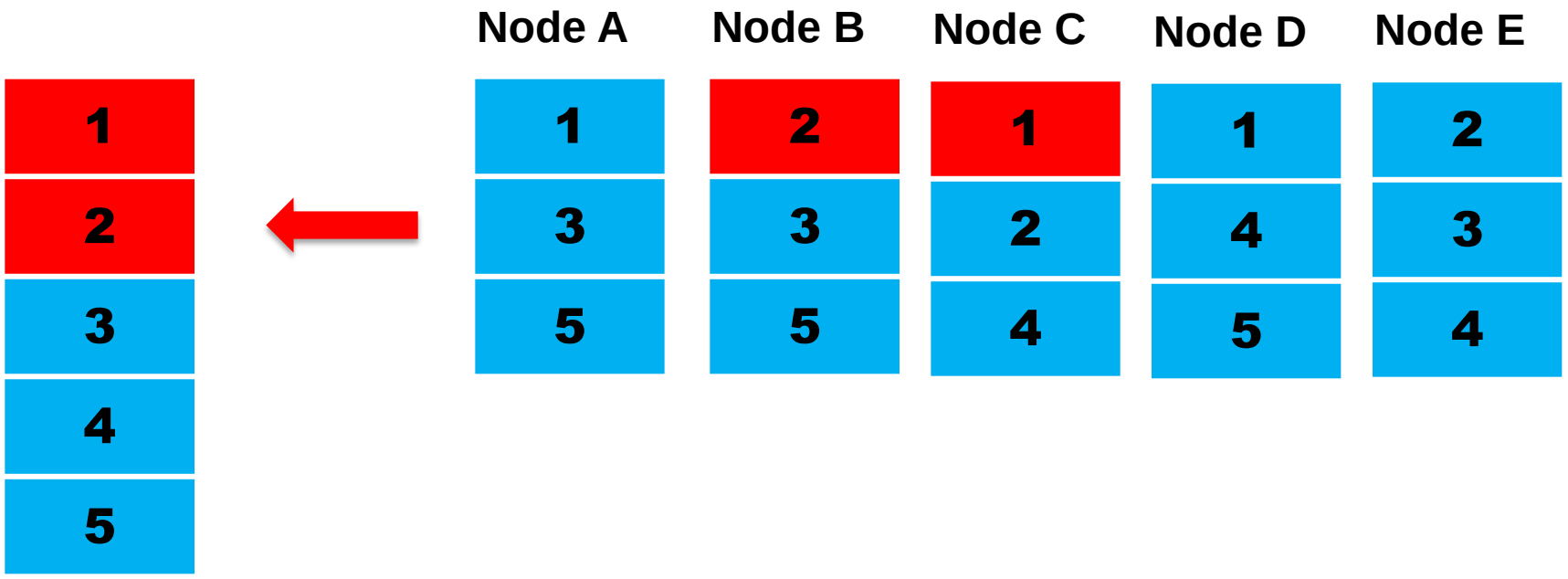
How MapReduce works

1. Sélection des nœuds portant les calculs
   - The node must not be busy



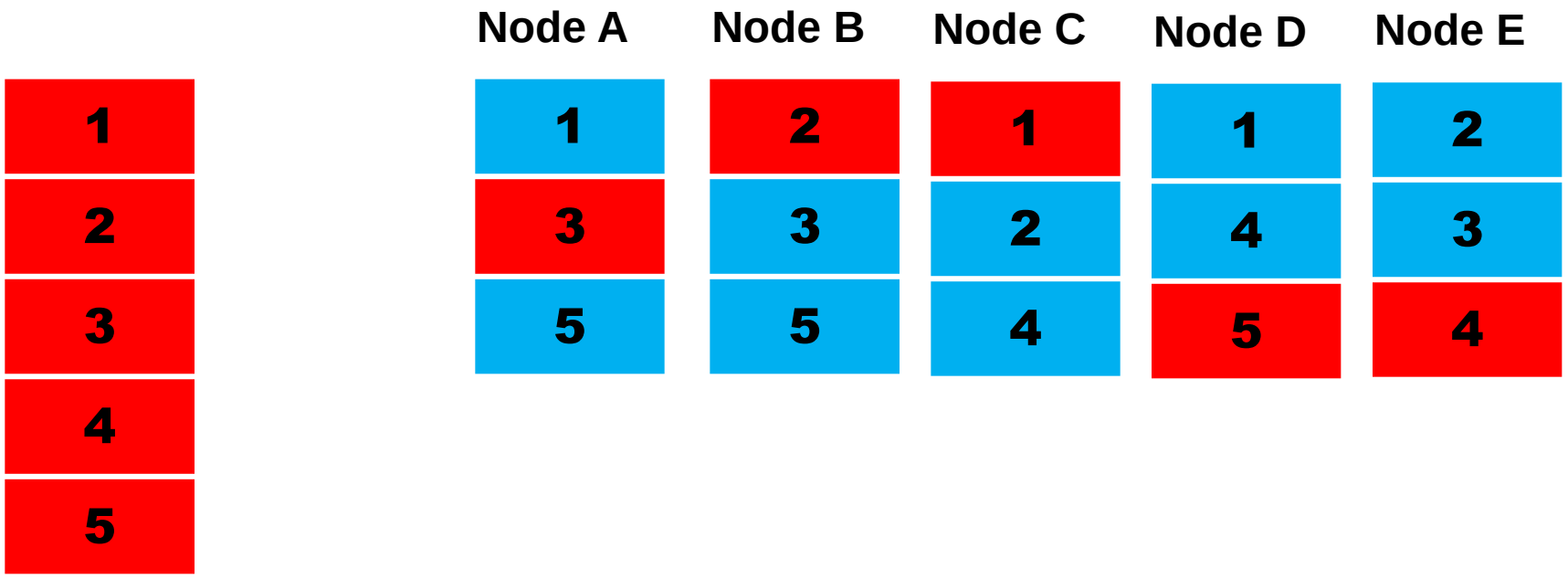
How MapReduce works

1. Sélection des nœuds portant les calculs



How MapReduce works

1. Sélection des nœuds portant les calculs



2. Why Spark on Ceph?

- Spark clusters in VM works great
  - For computations at least
  - Main usage of Spark (public clouds)
- Spark requires a distributed storage
  - HDFS, S3, NFS …
  - HDFS in a VM will not solve the problem
    - HDFS over Ceph = double penalty
    - Data locality doesn't make sense in VMs

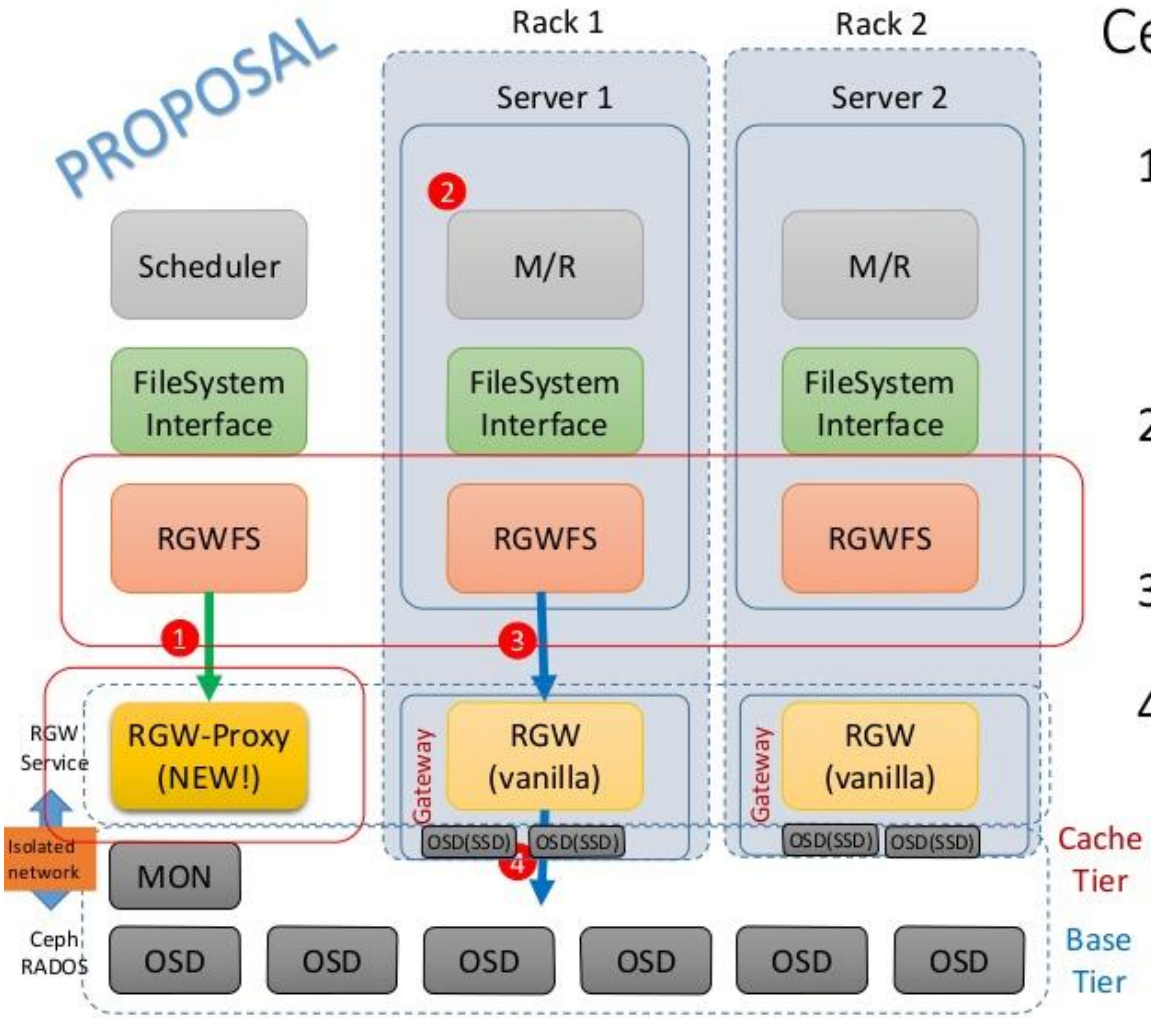
2. Why Spark on Ceph?

- Ceph is coupled with our OpenStack cluster
  - Local expertise
- HDFS is not an option
  - Problems with data locality
  - Computing and storage not paired in our cloud

3. Spark on Ceph – ideas

1. Using RGWFS
2. Using CephFS-Hadoop
3. Using a gateway with an S3 endpoint

3.1 - RGWFS



Ceph RGW with SSD cache

1. Scheduler ask RGW service where a particular block locates (control path)
   - RGW-Proxy returns the closest active RGW instance(s)
2. Scheduler allocates a task on the server that is near to the data
3. Task access data from nearby (data path)
4. RGW get/put data from the CT, and CT would get/put data from BT if necessary (data path)

3.1 - RGWFS

- http://www.slideshare.net/zhouyuan/hadoop-over-rgw
- Pros
  - Should ntegrate well with Spark through rgw://
- Cons
  - Git repo doesn't exist anymore
  - Cannot find more info – vaporware?

3.2 - CephFS-Hadoop

- https://github.com/ceph/cephfs-hadoop
- http://noahdesu.github.io/2015/07/12/hadoop-ceph-diving-in.html
- Pros
  - Transparent for Spark through hdfs://
- Cons
  - VMs have to be within the OSD network
  - Perfs?
  - Hadoop 1.X or doc not updated?

3.2 – S3 Gateway

- http://docs.ceph.com/docs/master/radosgw/s3/
- Pros
  - Hadoop supports the S3 protocol
  - VMS outside of the OSD network
- Cons
  - Another layer of indirection?
  - Perfs depending on the number of gateways?

Which solution is best suited?

- discussion