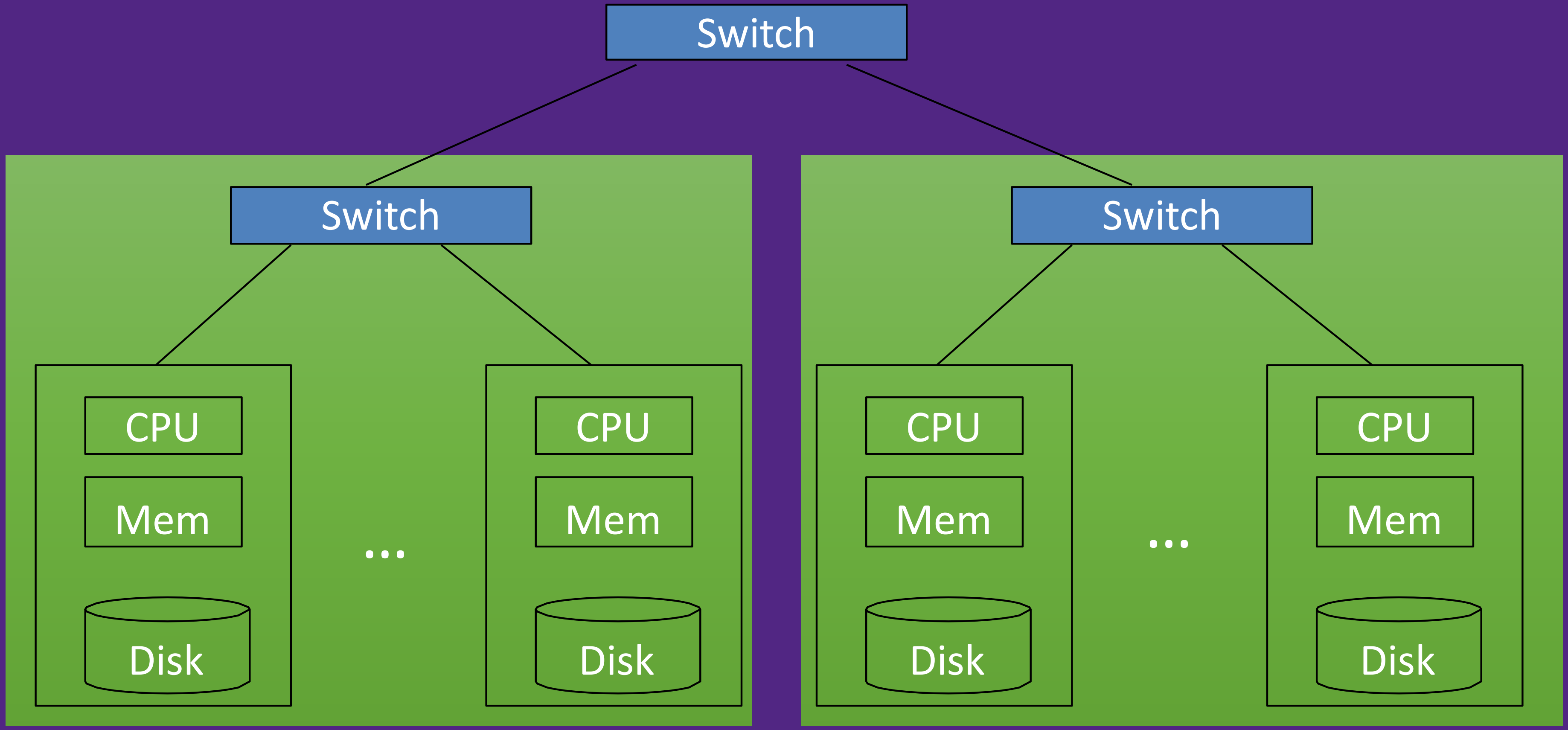
Cluster Architectures

Nodes, Shards, and Replicas

CS 4417B

The University of Western Ontario

Cluster Architecture



Large-scale clusters consist of several racks

Racks consist of nodes

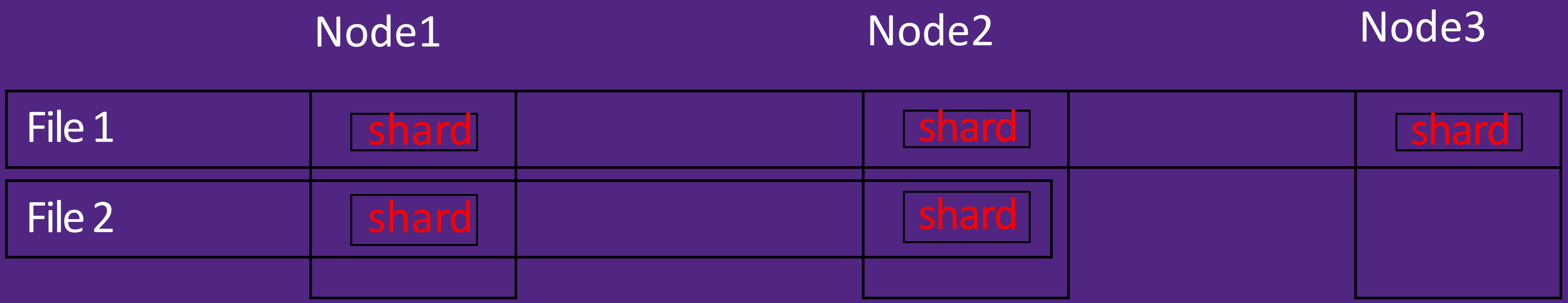
Communication within racks is faster than between racks

Cluster

- Cluster
  - Racks
    - Nodes (computers)

- A *shard* is a partition of data
  - Each shard is on a different computer (or node)
  - An index or file may be partitioned into shards

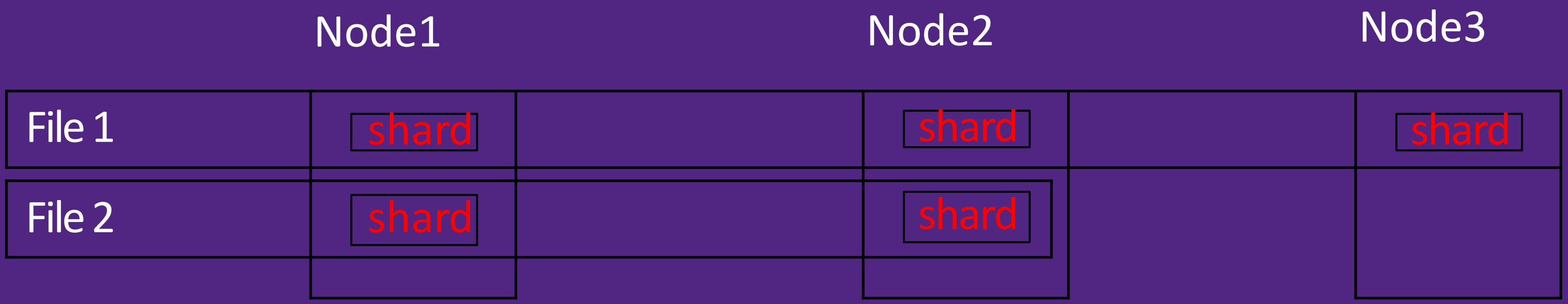
Shards



	Node1	Node2	Node3
File 1	shard	shard	shard
File 2	shard	shard	

- In the above example we see that file 1 consists of three shards and file 2 consists of two shards

Why Shards?



Node1

Node2

Node3

File 1

shard

shard

shard

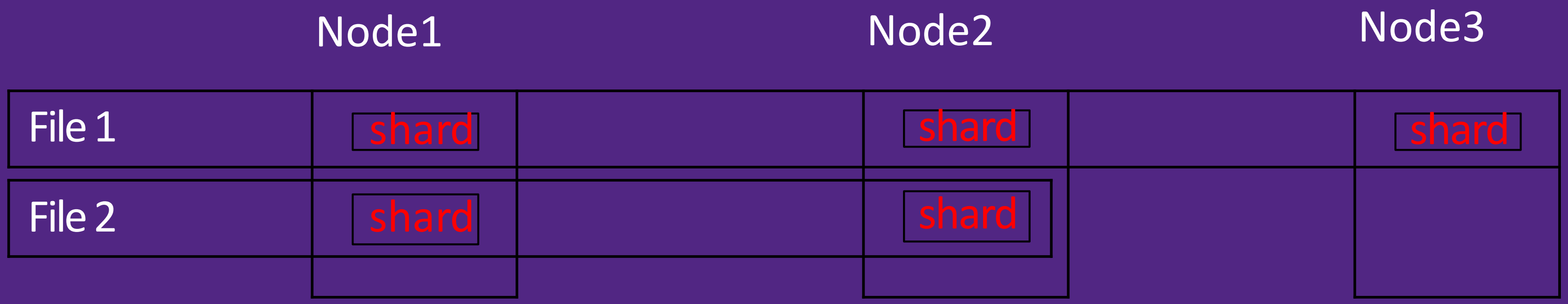
File 2

shard

shard

- Files may not fit on one disk
- Multiple users may want to access the same file
- Shards enable the use of multiple computers to handle requests
- The load is balanced better if shard access is uniform
- Elasticsearch, MongoDB split indices into shards; Hadoop File System splits files into "blocks"

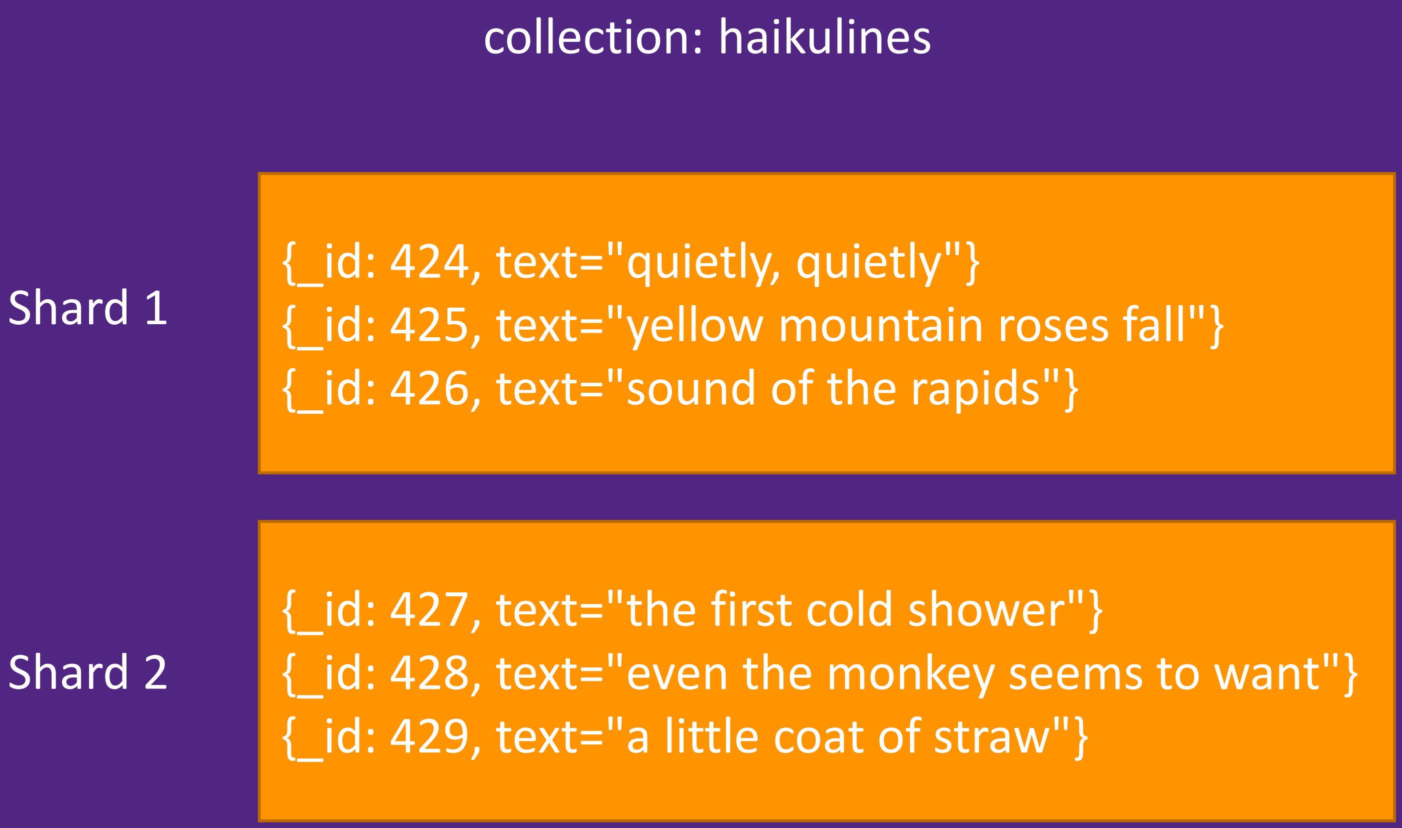
Why Not Shards?



	Node1	Node2	Node3
File 1	shard	shard	shard
File 2	shard	shard	

Sharding issues – how to divide?

- Elasticsearch, MongoDB shard at the document level



collection: haikulines

Shard 1

{_id: 424, text="quietly, quietly"}
{_id: 425, text="yellow mountain roses fall"}
{_id: 426, text="sound of the rapids"}

Shard 2

{_id: 427, text="the first cold shower"}
{_id: 428, text="even the monkey seems to want"}
{_id: 429, text="a little coat of straw"}

Sharding issues – how to divide?

- Hadoop FS shards (blocks) at the byte level

file: haikulines.json

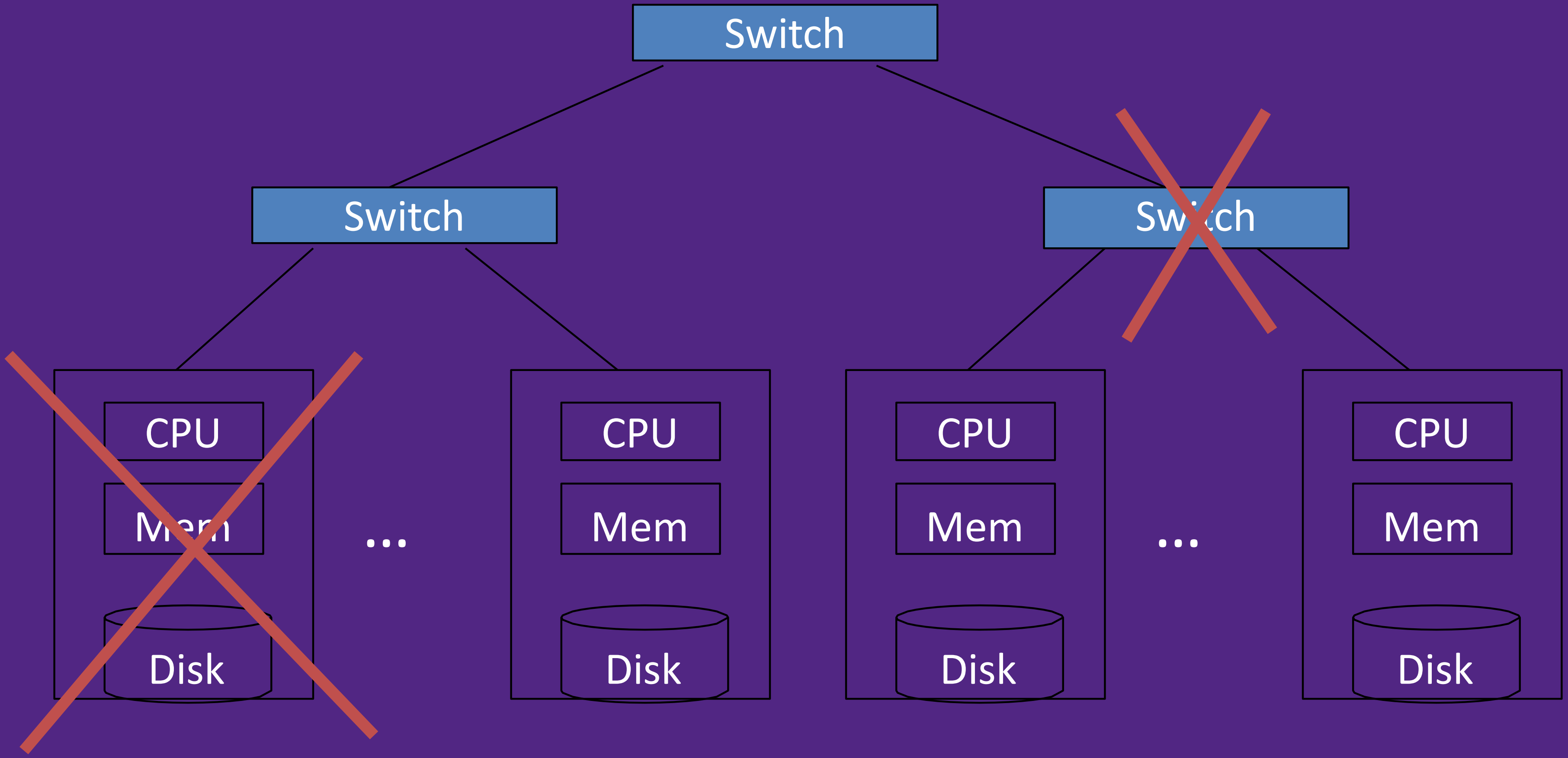
Shard (block) 1
128 bytes

```
{_id: 424, text="quietly, quietly"}
{_id: 425, text="yellow mountain roses fall"}
{_id: 426, text="sound of the rapids"}
{_id: 4
```

Shard (block) 2
124 bytes

```
27, text="the first cold shower"}
{_id: 428, text="even the monkey seems to want"}
{_id: 429, text="a little coat of straw"}
```

Cluster Architecture



- Failures can occur during computation
- Do not want to restart computation every time there is a failure

Replicas

- What if a node goes down?
- This means that a shard (and hence part of an index is lost).
- We may want to create at least one replica of a shard to ensure availability

Sharding and Replicas

- Is replication faster for data that is mostly read or mostly written?

Cluster Architecture

- If nodes or switches fail, how can we store data persistently and maintain availability?

- There are multiple file systems that can shard, replicate, and maintain files so that hardware failures can be tolerated
    - Google GFS
    - Hadoop HDFS
    - Kosmix KFS

Cluster Underlying Operating System

- Today companies like Google, Amazon, Facebook have clusters that use the Linux Operating system
- Why?
  - Linux is open source
  - You can modify Linux to suit your needs
- Google developed MapReduce assuming Linux as a base

Tools for Cluster Computing

- These run on top of the underlying operating system, facilitate data access and computation

- File systems already mentioned: Google GFS, Hadoop HDFS, Kosmix KFS

- https://en.wikipedia.org/wiki/List_of_cluster_management_software

Summary

- Cluster
- Racks
- Nodes
- Shards
- Blocks
- Replicas