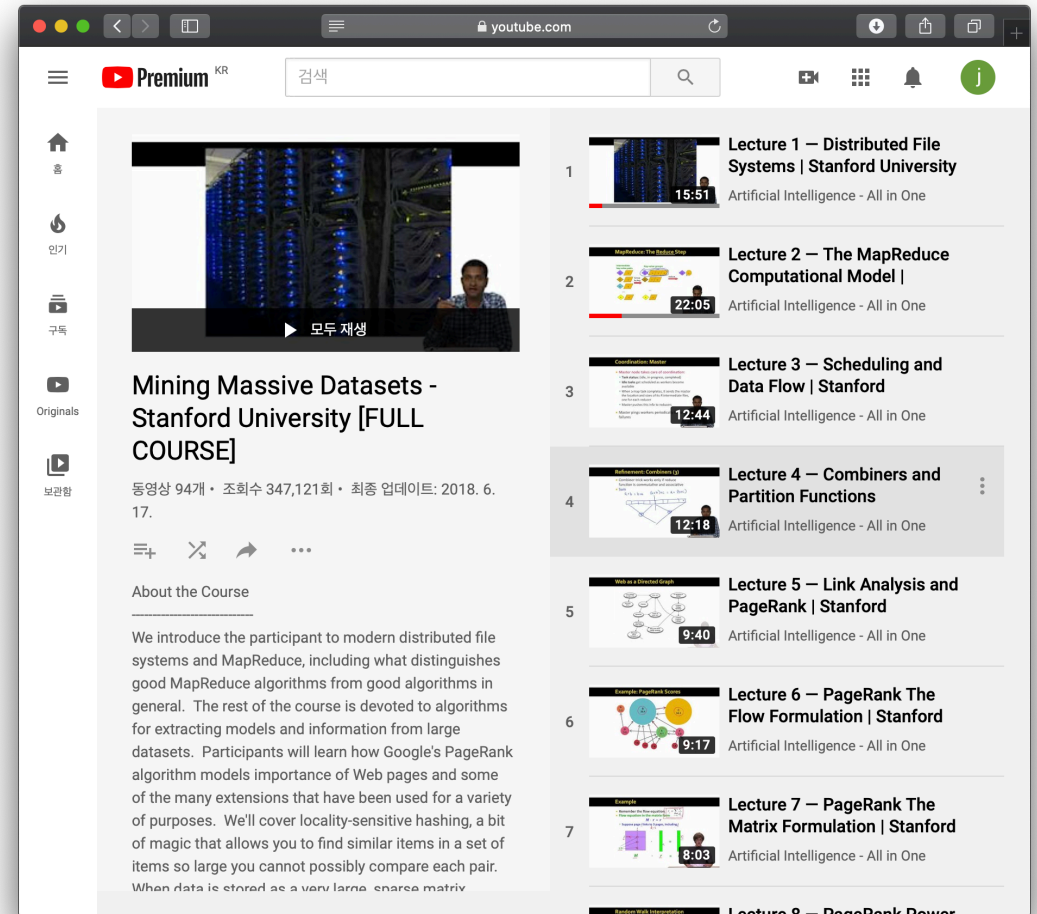# Mining Massive Datasets

Jisung Jeong / jisung0920.github.io

Reference

- Open Course :
  Mining Massive Datasets (Stanford)

- Textbook :
  Mining of Massive Datasets
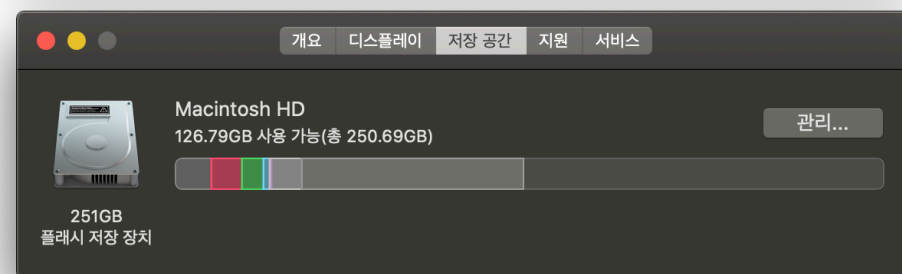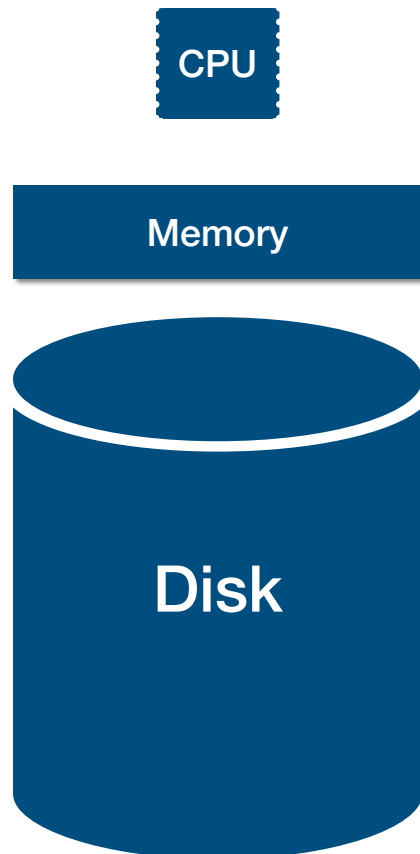 (Jure Leskovec, Anand Rajaraman, Jeff Ullman)

1. MapReduce

2. PageRank

3. Locality-Sensitive Hashing

4. Distance Measures, Nearest Neighbors

5. Frequent Itemsets

6. Communities in Social Networks

7. Stream Algorithms

8. Recommender Systems

9. Dimensionality Reduction

10. Clustering

11. Computational Advertising

12. Machine Learning

13. More About MapReduce

14. More About Locality-Sensitive Hashing

15. More About Link Analysis

MapReduce

Physical Organization of Compute Nodes

Jisung Jeong / jisung0920.github.io

CPU

Memory

Disk

macOS Catalina
버전 10.15.5

MacBook Air (Retina, 13-inch, 2020)
프로세서   1.1 GHz 쿼드 코어 Intel Core i5
메모리   8GB 3733 MHz LPDDR4
그래픽   Intel Iris Plus Graphics 1536 MB
일련 번호   FVFCM6VSMNHX

시스템 리포트...   소프트웨어 업데이트...

™ 및 © 1983-2020 Apple Inc. 모든 권리 보유. 사용권 계약

개요   디스플레이   저장 공간   지원   서비스

Macintosh HD
126.79GB 사용 가능(총 250.69GB)

관리...

251GB
플래시 저장 장치

J Lab

- 웹페이지 수 : 10Billion

- 웹페이지 크기 : 20KB

= 200TB

CPU

Memory

Disk

Web
File

CPU

50m/s

Memory

Disk

Web
File
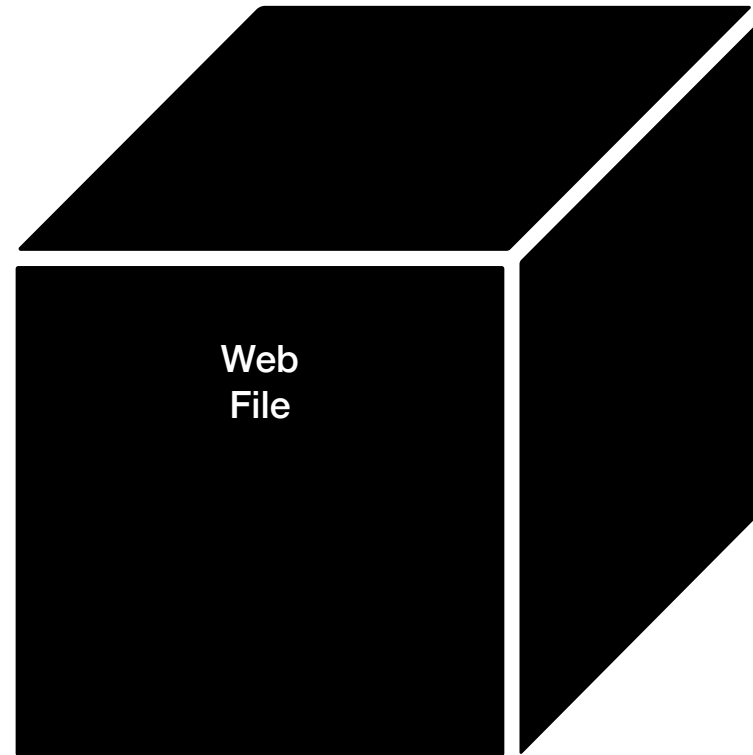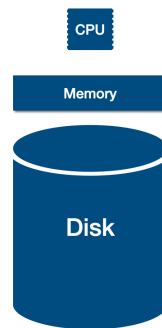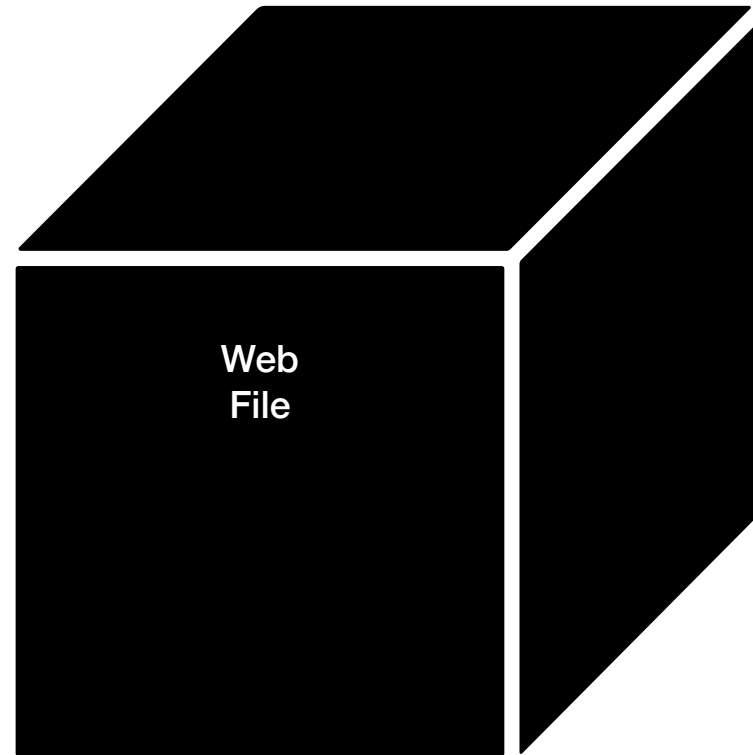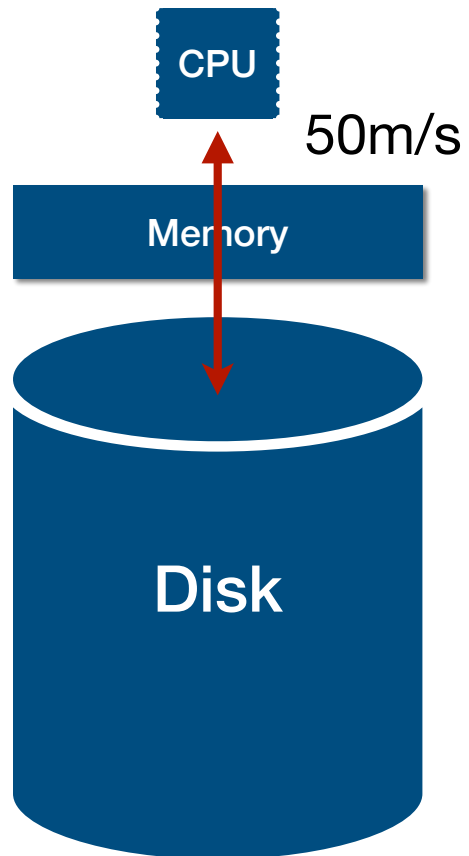
1. Physical Organization of Compute Nodes

2. Distributed File Systems

3. MapReduce

1. Physical Organization of Compute Nodes

Compute Node
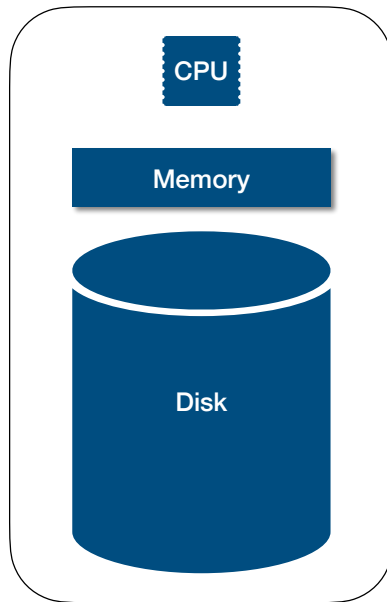
Rack
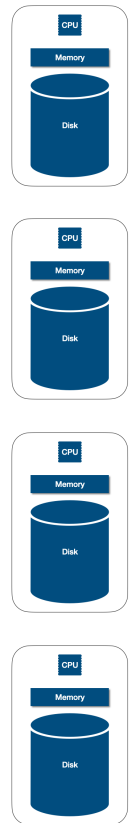
1. Physical Organization of Compute Nodes

1. Physical Organization of Compute Nodes

Backbone Switch >1Gbps

Switch 1Gbps

Switch

Switch

Rack

Rack

. . . .

Rack

1. Physical Organization of Compute Nodes

J Lab
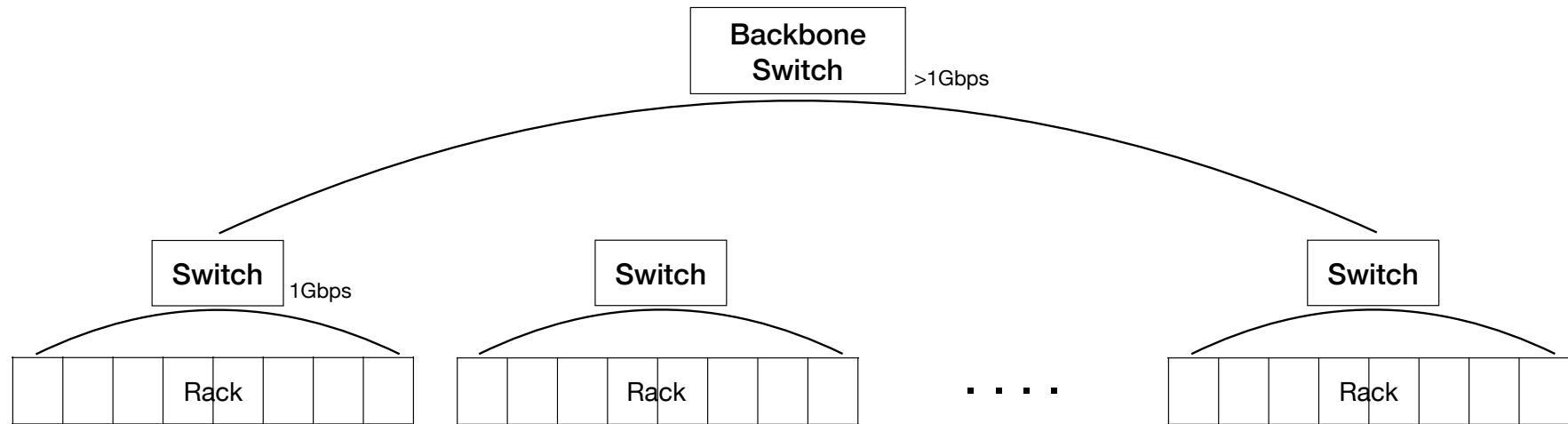
MapReduce

Distributed File Systems

Jisung Jeong / jisung0920@gmail.com

Backbone Switch >1Gbps

Switch 1Gbps

Switch

Switch

Rack

Rack

. . . .

Rack

J Lab
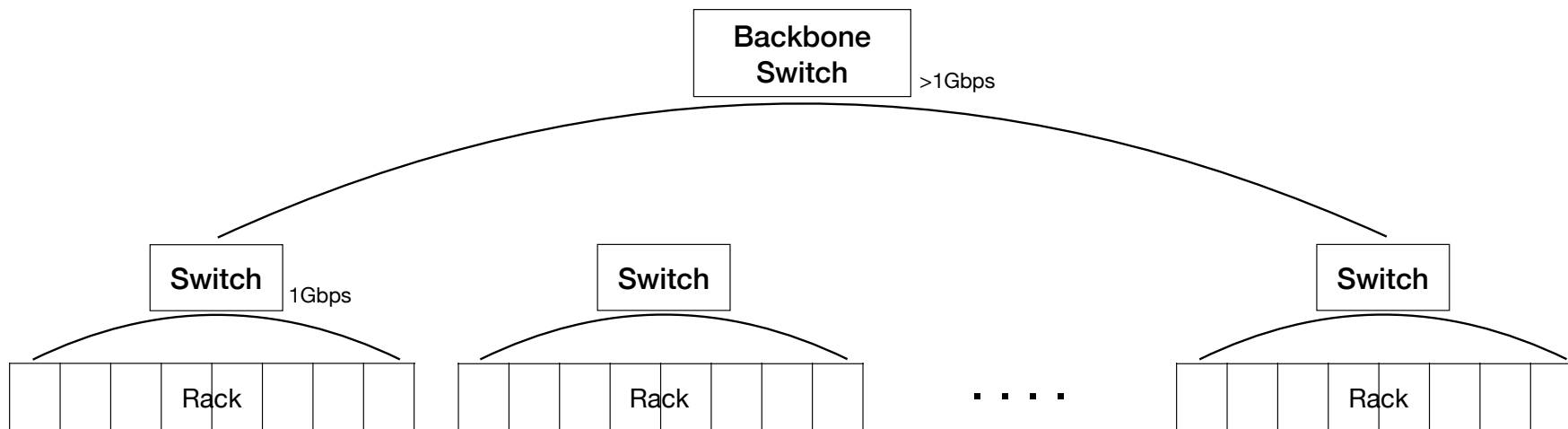
1. Node Failure

2. Network Bottleneck

3. Programming

Node Failure

Backbone Switch

>1Gbps

Switch 1Gbps

Switch

Switch

Rack

Rack

· · · · ·

Rack

Network Bottleneck

Backbone Switch >1Gbps

Switch 1Gbps

Switch

Switch

Rack

Rack

. . . . .
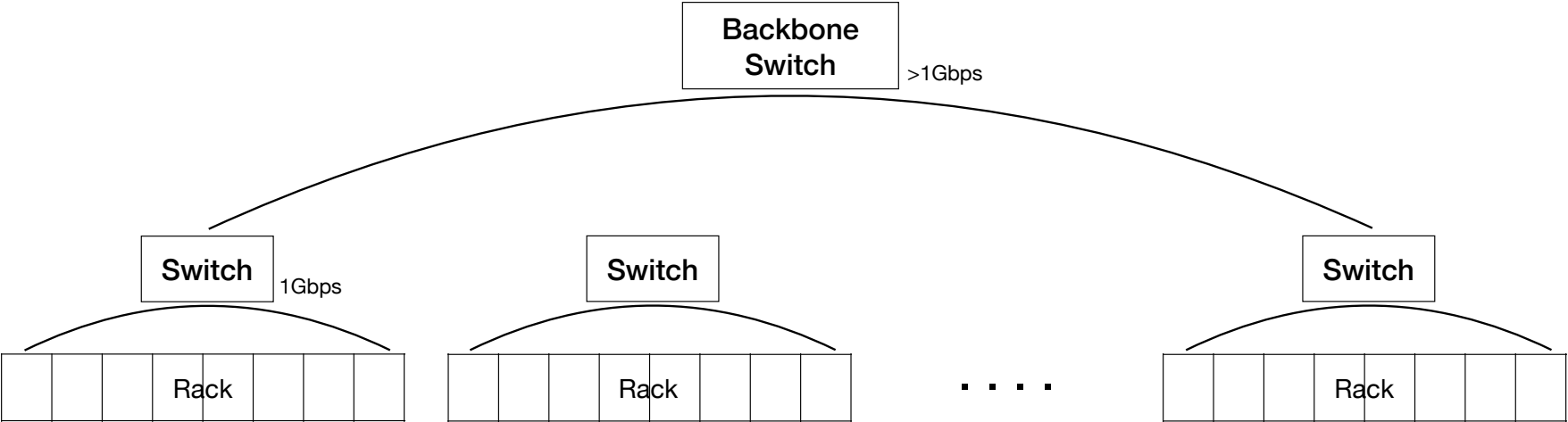
Rack
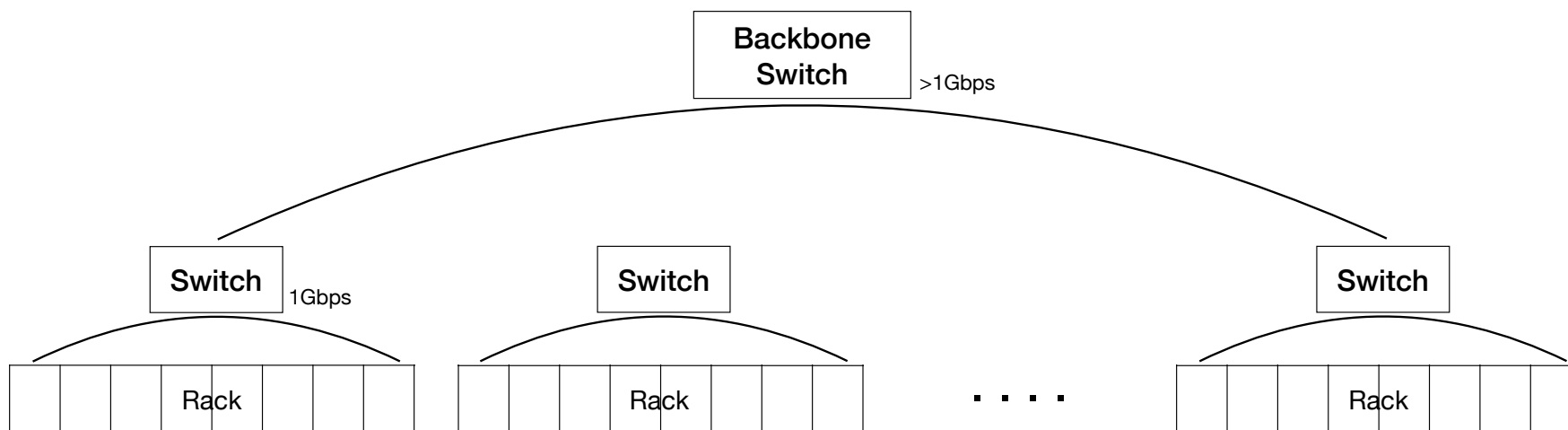
Programming
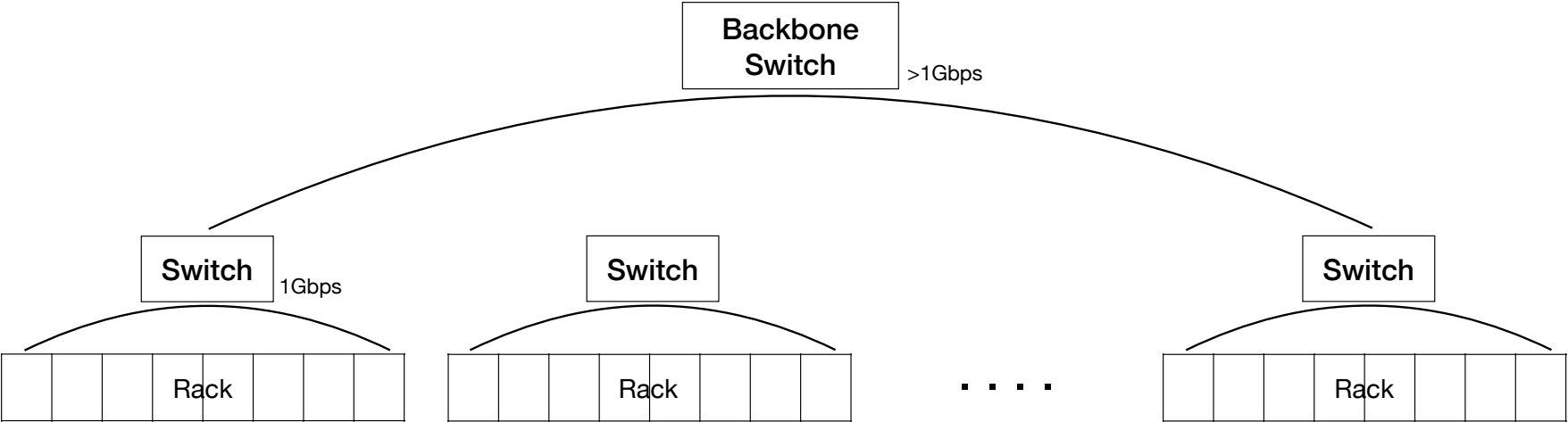
Distributed File System

Distributed File System

file A

A1
A2
A3
.
.
.
.
.

Distributed File System

file A

A1
A2
A3
.
.
.
.
.

file B

B1
B2
B3
.
.
.
.
.

Rack 1

Disk
A1	B1

Disk
A2	B3

Disk
B2	A3

. . .

Rack 2

Disk
A3	B1

Disk
B3	A2

Disk
B2	A1

. . .

J Lab
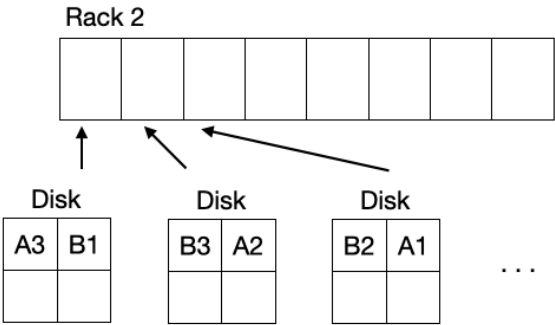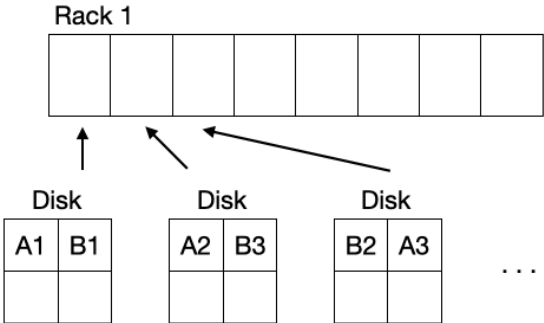
Distributed File System

Master Node

Distributed File System

			Rack			

			Rack			

. . . .

			Rack			

Disk

A3	B1
E1	U2

Disk

B3	A2
C2	Q9

Disk

B2	A1
V5	I8

. . .

Disk
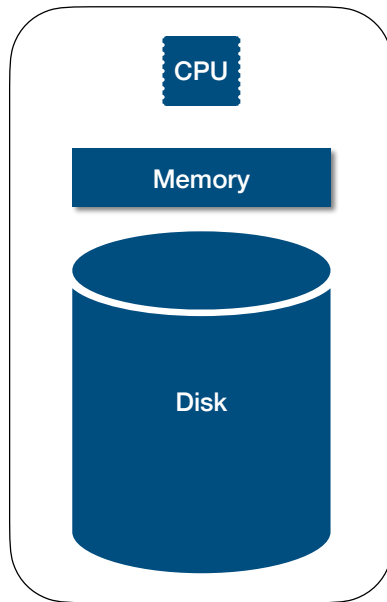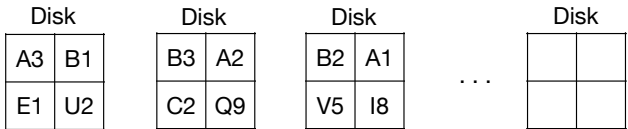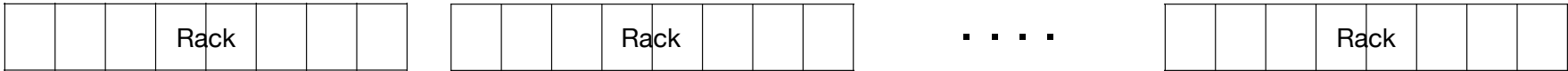
MapReduce

MapReduce

Jisung Jeong / jisung0920@gmail.com

Cluster

File

A1
A2
A3
.
.
.
.
.

B3	A2
C2	Q9

B2	A1
V5	I8

A3	B1
E1	U2

DFS

1. Map

2. Group by Key

3. Reduce

- Worker

- Task

- Map Task

- Reduce Task

- Mapper

- Reducer

Worker 1

Chunk

| A1 | B1 |
| C2 | E4 |

Map Task · Mapper

Worker 2

Chunk

| D2 | B3 |
| F7 | A2 |

Map Task · Mapper

Worker 3

Chunk

| B2 | A3 |
| A4 | D1 |

Map Task · Mapper

Map Task · Mapper

Worker 4

Reduce Task · Reducer

Worker 5

Reduce Task · Reducer

Output File

J Lab

Map

Elements → Map Function → (key,value)*
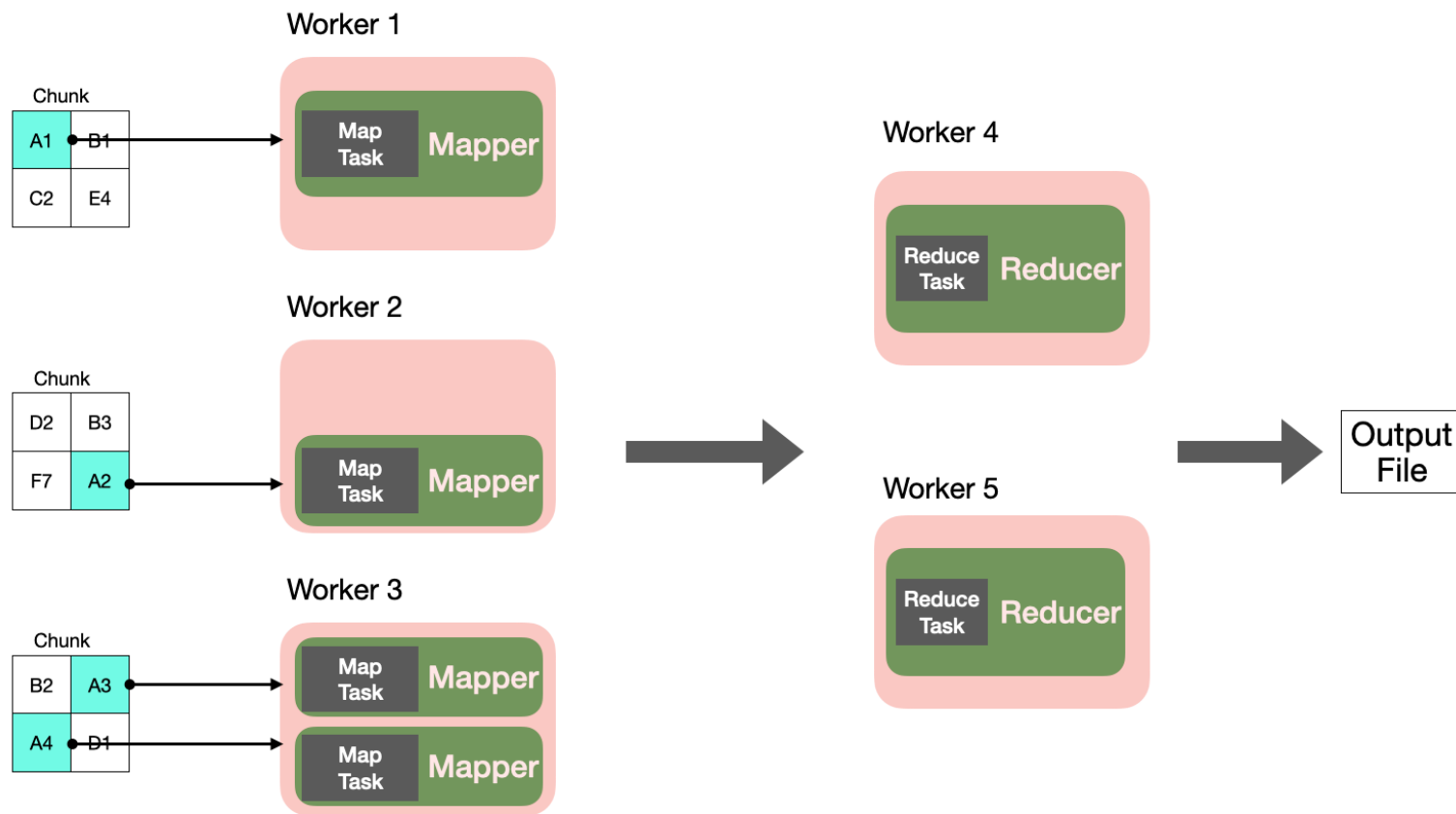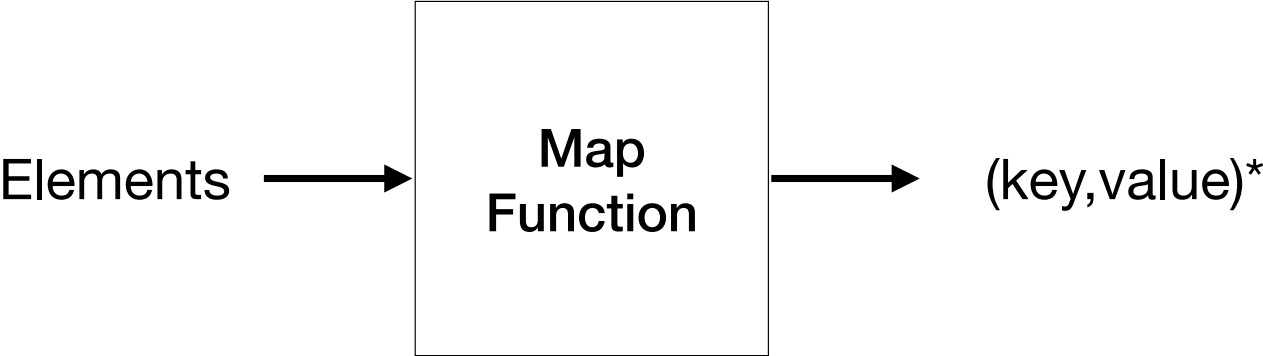
Group by key

$$(\, k_1, \, v_a), \, (\, k_2, \, v_a), \, (\, k_3, \, v_b), \, (\, k_1, \, v_b), \, (\, k_1, \, v_c) \, \ldots$$

$$\downarrow$$

$$(\, k_1, \, [v_a, v_b, v_c] \,) \, , \, (\, k_2, \, [v_a] \,), \ldots$$

Reduce

$$(\text{key}, [v_1, v_2, \ldots, v_n])$$ →
| Reduce Function | → (key, value')*

Example - Word Counter

Chunk 1

노드 장애(=Node failure)란 컴퓨트 노드가 여러 원인들로 인해 정상적인 작동을 못하는 것을 말합니다.

단일 노드구조에도 노드 장애는 일어날 수 있습니다.

이 노드 장애가 발생할 확률은 매우 낮아서 단일 구조인 경우, 이 상황이 큰 문제가 되진 않습니다.

그러나 클러스터 구조에서는 단일 구조에서 보다 큰 문제가 됩니다.

클러스터에 연결된 노드가 많아질수록 노드 장애가 발생할 확률이 높아집니다.

Chunk 2

하나의 컴퓨트 노드에서 노드 장애가 1년에 한번씩 발생한다고 가정해봅시다.

64개의 노드를 갖는 랙과, 6개의 랙으로 구성된 클러스터 컴퓨터에서는

이 노드 장애가 하루에 한번씩 발생하게 됩니다. (64*6)

하나의 파일이 각 노드에서 나눠져서 작동하기 때문에 하나의 노드에서 장애가 발생한다면

Chunk 3

그 노드를 정상 작동 시키기 전까지는 파일 전체를 사용할 수 없게됩니다.

파일 시스템은 Persistence(: 데이터를 저장하면 그것을 다시 read 하도록 하는 것)

Availability(:작업이 중단 없이 수행되는 것) 이 두 속성을 보장해야하지만,

현재 클러스터 컴퓨팅 구조에서는 이를 보장하기 어렵습니다.

Chunk 4

네트워크 병목현상(=Network bottleneck)은 데이터를 이동(=move) 할 때 발생합니다.

노드에서 연산에 필요한 데이터가 다른 노드에 있는 경우,

이를 가져와서 연산을 수행해야합니다.

클러스터 컴퓨팅 구조를 사용하는 경우는 일반적으로 대규모의 데이터(GB,TB)를 처리하는 경우입니다.

그러나 노드와 노드 사이를 연결하는 스위치의 데이터 대역폭은 Gbps 수준입니다.

Map →

(key , value)

('노드',1)

('장애',1)

('컴퓨트', 1)

('노드가', 1)

('여러', 1)

('원인들로',1)

('인해',1)

.

.

.

Example - Word Counter

(key , value)

('노드',1)

('장애',1)

('컴퓨트', 1)

('노드가', 1)

('여러', 1)

('원인들로',1)

('인해',1)

('정상적인',1)

('작동을', 1)

('못하는',1)

('것을',1)

('말합니다',1)

('단일',1)

.

.

.

(key , list)

('노드', [1,1,1,1])

('장애', [1,1,1])

('컴퓨트', [1,1])

('노드가', [1])

('여러', [1,1])

('원인들로',[1])

('인해',[1,1])

('정상적인',[1])

('작동을', [1])

('못하는', [1,1])

.

.

.

Example - Word Counter

(key , list)   (key , value)

('노드', [1,1,1,1])		('노드', 4)
('장애', [1,1,1])	Reduce	('장애', 3)
('컴퓨트', [1,1])	→	('컴퓨트', 2)
('노드가', [1])		('노드가', 1)
('여러', [1,1])		('여러', 2)
('원인들로', [1])		('원인들로', 1)
('인해', [1,1])		('인해', 1)
('정상적인',[1])		('정상적인', 1)
('작동을', [1])		('작동을', 1)
('못하는', [1,1])		('못하는', 2)

.
.
.

Example - Average Price

Chunk 1	
맥주	1900
김치	4000
계란	6000
파	2000
양파	2000
콜라	2000
맥주	2000

Chunk 2	
맥주	2500
계란	4000
과자	1500
과자	1000
후추	2000
라면	4000
삼겹살	30000

Map →

(key , value)

('맥주',1900)

('김치',4000)

('계란', 6000)

('파', 2000)

('양파', 2000)

('콜라',2000)

('맥주',2000)

J Lab

Example - Average Price

(key , value)   (key , list)

('맥주',1900)   ('맥주',[1900,2000])

('김치',4000)   ('김치',[4000])

('계란', 6000)   ('계란', [6000])

('파', 2000)   ('파', [2000])

('양파', 2000)   ('양파', [2000])

('콜라',2000)   ('콜라',[2000])

('맥주',2000)   ('맥주',[2000])

Example - Average Price

(key , list)

(key , value)

('맥주',[1900,2000])

('김치',[4000])

('계란', [6000])

('파', [2000])

('양파', [2000])

('콜라',[2000])

('맥주',[2000])

Reduce

('맥주', 1950)

('김치', 4000)

('계란', 6000)

('파', 2000)

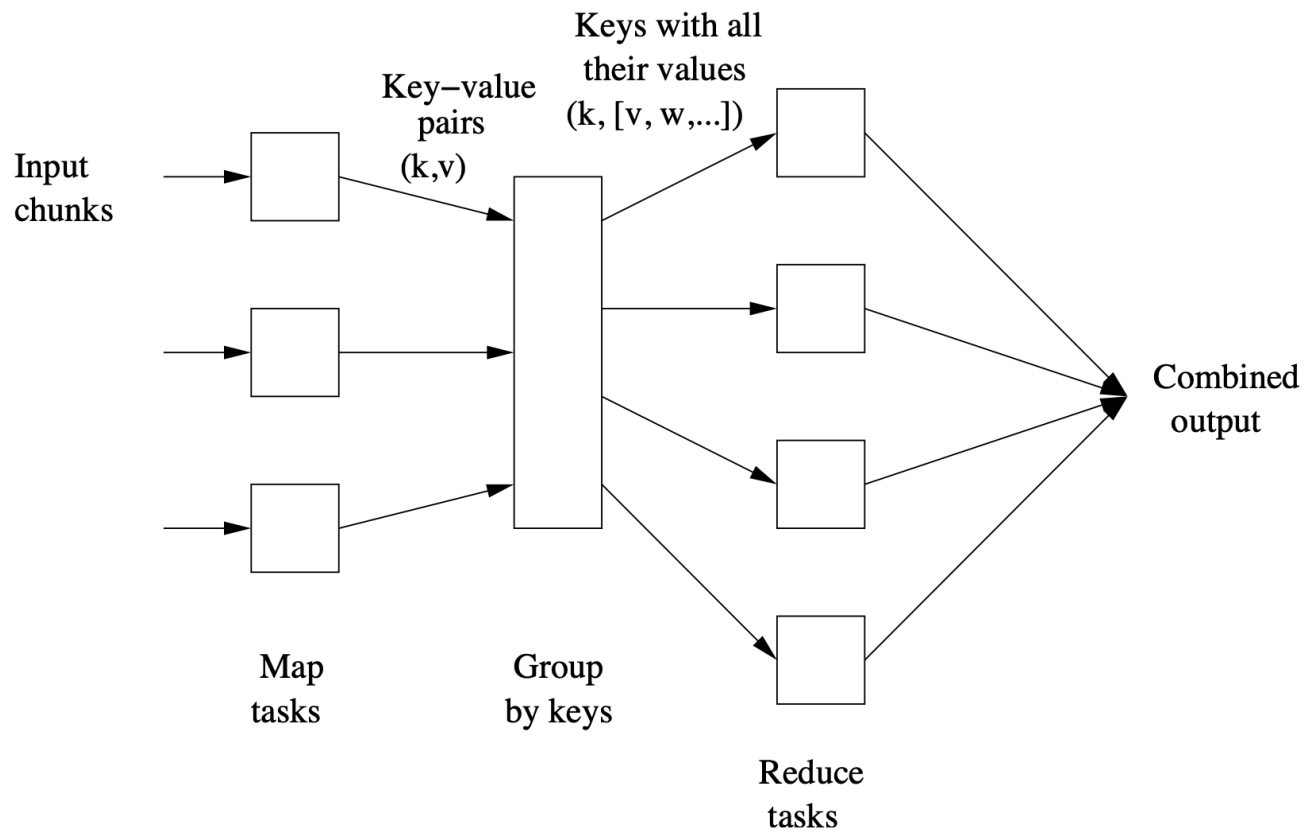('양파', 2000)

('콜라', 2000)

('맥주', 2000)

MapReduce

Figure 2.2: Schematic of a MapReduce computation

Mining of Massive Datasets - Figure 2.2

J Lab

MapReduce

Advanced
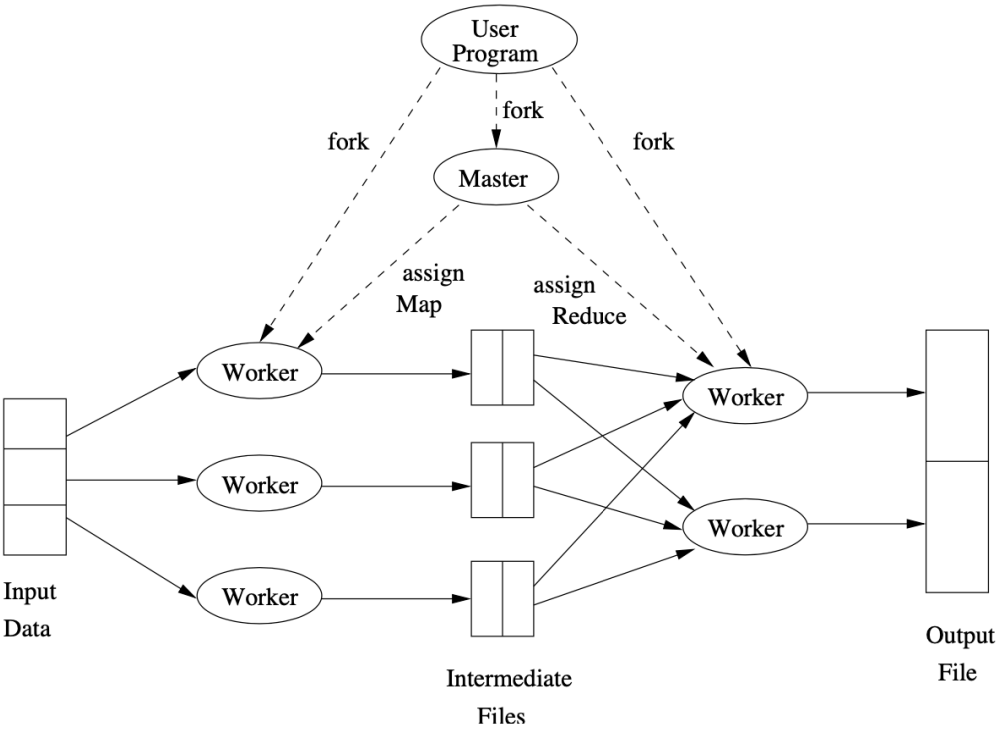
Jisung Jeong / jisung0920@gmail.com

MapReduce

Figure 2.3: Overview of the execution of a MapReduce program

Mining of Massive Datasets - Figure 2.3

J Lab

Node Failure

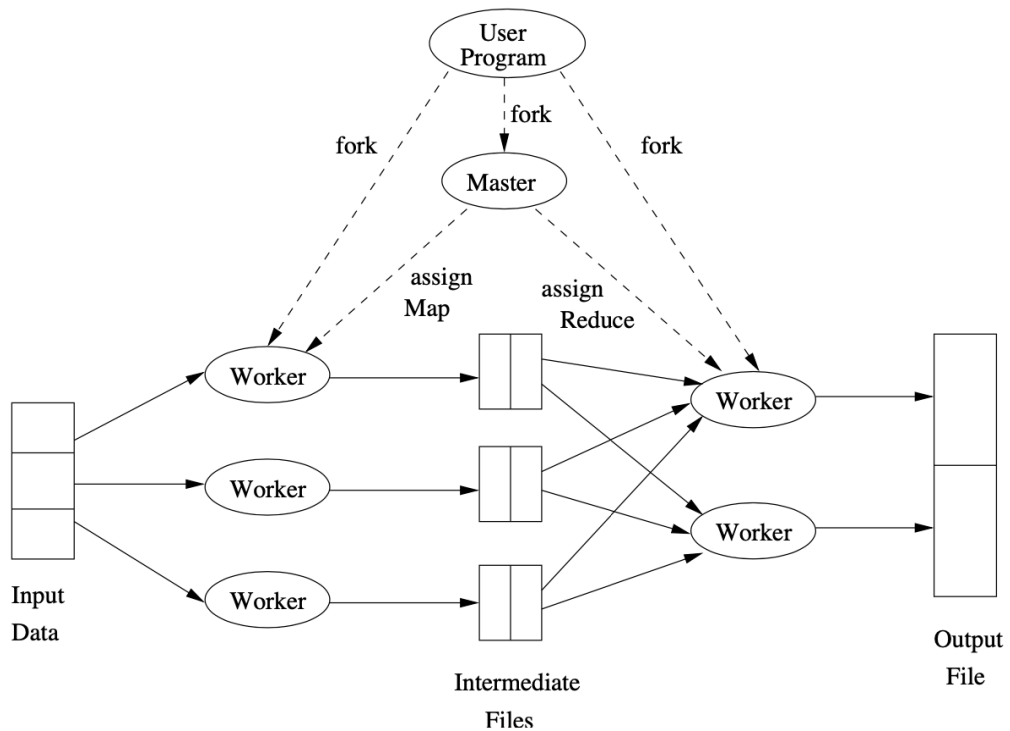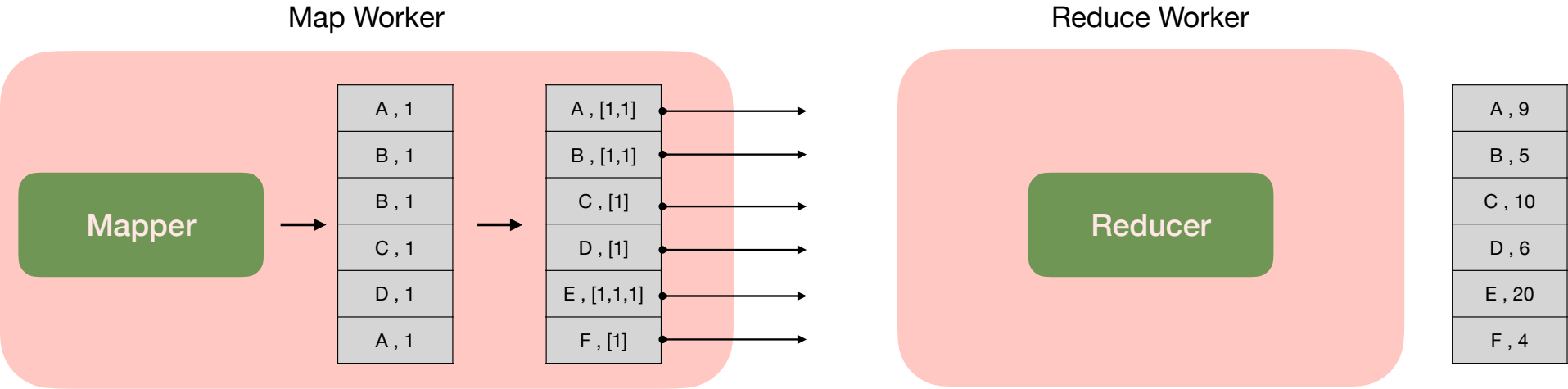

Figure 2.3: Overview of the execution of a MapReduce program

Mining of Massive Datasets - Figure 2.3
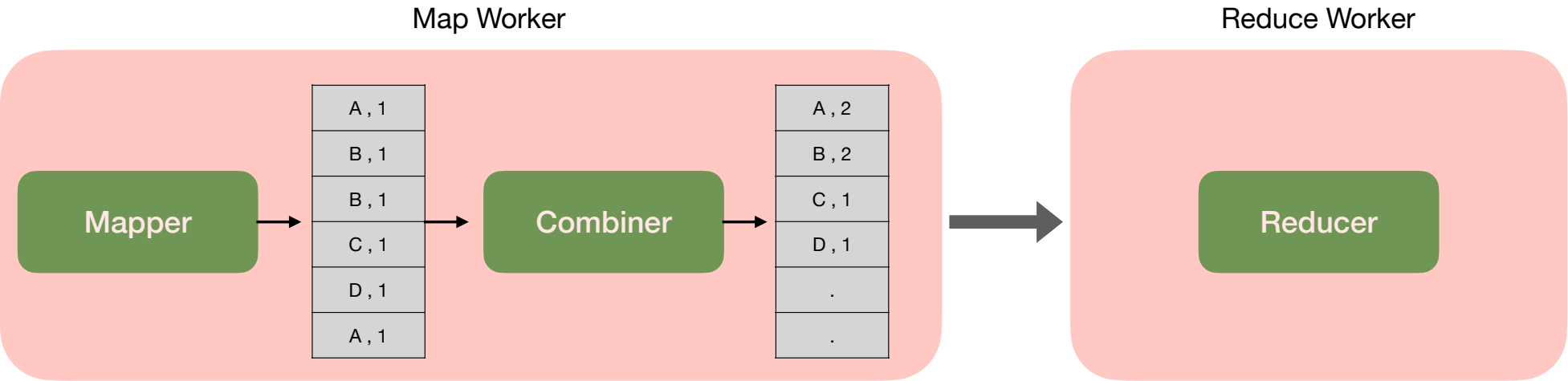
- Master

- Map Worker

- Reduce Worker

J Lab

Combiner

Map Worker

Mapper

A , 1
B , 1
B , 1
C , 1
D , 1
A , 1

A , [1,1]
B , [1,1]
C , [1]
D , [1]
E , [1,1,1]
F , [1]

Reduce Worker

Reducer

A , 9
B , 5
C , 10
D , 6
E , 20
F , 4

J Lab

Combiner

Map Worker

Reduce Worker

Mapper	→	A , 1
		B , 1
		B , 1
		C , 1
		D , 1
		A , 1

Combiner	→	A , 2
		B , 2
		C , 1
		D , 1
		.
		.

Reducer

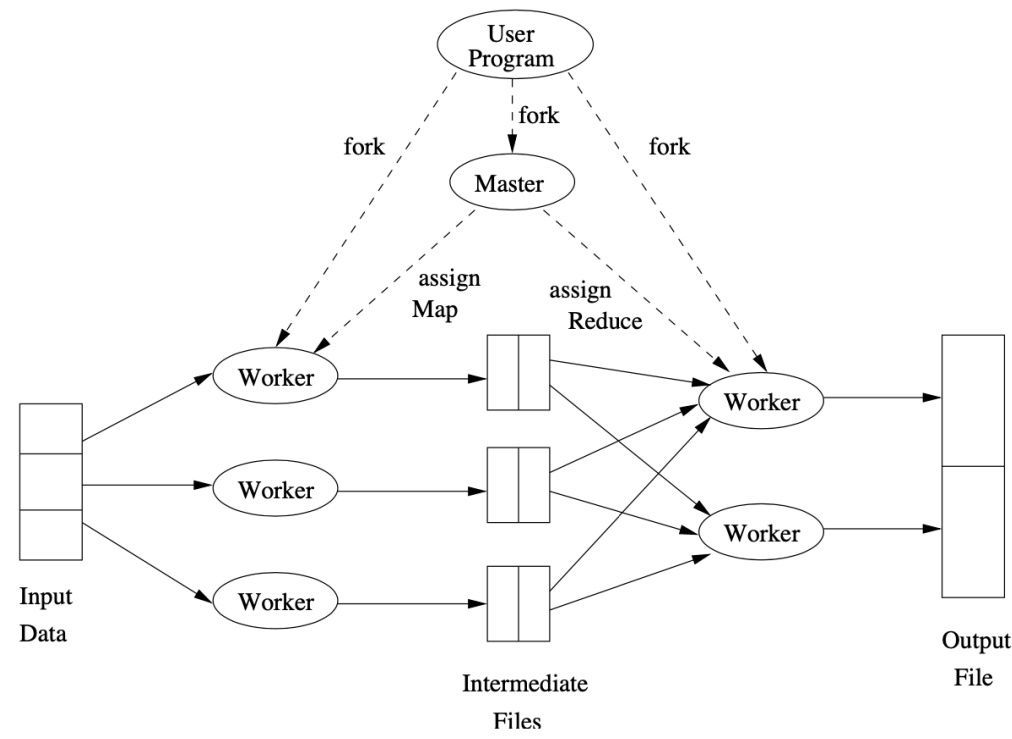J Lab

Partition Function

Figure 2.3: Overview of the execution of a MapReduce program

Mining of Massive Datasets - Figure 2.3

J Lab

Partition Function