

# Project Progress

Team Grey

# Review of Weekly Progress

# Weekly Progress

Progress Category: **Communication** / **Setting** / **Design** / **Implementation**

Week1	<b>Communication</b> <ul style="list-style-type: none"><li>- Github repository 생성</li><li>- Documentation 공유 목적의 Notion 생성</li></ul>
Week2	<b>Design</b> <ul style="list-style-type: none"><li>- Project에서 요구하는 기능의 전반적 흐름 이해</li><li>- 각 단계별로 Master와 Worker 사이에서 수행되어야 하는 일과 module 간의 관계 이해</li><li>- 여러 Shuffling methods에 대한 논의</li><li>- 이를 바탕으로 프로젝트 전반의 milestone 구체화</li></ul>

# Weekly Progress

Progress Category: **Communication** / **Setting** / **Design** / **Implementation**

Week3	<p><b>Communication</b></p> <ul style="list-style-type: none"><li>- 용어 통일: Master, Worker</li><li>- github commit convention 결정</li><li>- scala coding style convention 결정</li></ul> <p><b>Setting</b></p> <ul style="list-style-type: none"><li>- 개발 환경 통일: Scala 2.13.8</li><li>- Test 환경으로 Docker 사용 결정</li></ul> <p><b>Design</b></p> <ul style="list-style-type: none"><li>- Sampling method 검증 및 전체적인 flow에서 sampling 순서 결정</li><li>- 전체적인 flow 도식화</li></ul> <p><b>Implementation</b></p> <ul style="list-style-type: none"><li>- gRPC의 전반적인 작동 방식 및 demo 진행 상황 공유</li></ul>
-------	---

# Weekly Progress

Progress Category: **Communication** / **Setting** / **Design** / **Implementation**

Week4	<p><b>Setting</b></p> <ul style="list-style-type: none"><li>- 가상환경 세팅: Docker</li></ul> <p><b>Design</b></p> <ul style="list-style-type: none"><li>- Flow chart 정리</li></ul> <p><b>Implementation</b></p> <ul style="list-style-type: none"><li>- gensort를 통한 dataset generation</li><li>- gRPC를 통한 Master, Worker 간 connection 구현</li><li>- Java library를 이용하여 external merge sort 구현</li><li>- partitioning 구현</li></ul>
-------	--

# Logistics

# Communication

- 매주 일요일 오후, **GSR**에서 **2시간** 이상 대면 미팅 진행
- 미팅에서 각자의 진행 상황 공유
- 공유된 진행 상황을 바탕으로 프로젝트 관련 중요 사안 결정
  - 팀원 모두의 동의를 기반으로 함
- 다음 미팅까지 진행할 각자의 **Milestone** 설정
- 한 명이 미팅 내용 기반으로 **github README.md**에 **Weekly progress** 업로드

순번	분관	유형	공간	동반이용자수	사용시간	상태
1	Library	GSR	4F 화상회의실-B	2명	2022/11/13 14:00 ~ 18:00	사용종료
7	Library	GSR	4F 화상회의실-B	2명	2022/11/06 16:00 ~ 19:00	사용종료
9	Library	GSR	4F 화상회의실-B	2명	2022/10/30 14:30 ~ 16:30	사용종료

# Documentation

## Notion

- 각 주차의 상세 Meeting Log
- Milestone, Scala Coding Convention을 비롯한 Documents

### 22.10.21(금) Week 1



332project  
ijeongone

#### TODOs

- ☐ README에 어느 정도의 내용을 담을지, documentation을 어떻게 관리할지에 대한 논의
- ☐ 끝지막한 milestone에 대한 구체화 (다음시간)
  - ☐ 이를 기반으로 역할 분담 (다음시간)
- ☐ 다음에 언제 만날지에 대해 이야기하기: 10/30 14:00~ / GSR

README.md 에 대한 고찰



## [CSED322] SD Team Project

2022 POSTECH fall-semester Software Design Methods



332project  
ijeongone

Document

Task List

### ▼ Meeting Log

22.10.21(금) Week 1

22.10.30(일) Week 2

22.11.06(일) Week 3

22.11.13(일) Week 4



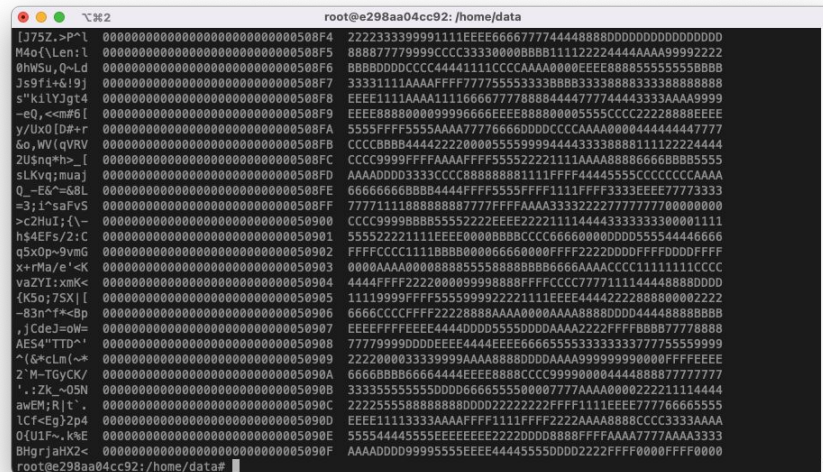
# Details

# Programming Environment

- Network: gRPC, ScalaPB
- Logger: java.util.logger

## Input Dataset

```
root@8c34382543f9:/home/64# ./gensort -a 330000 partition1
root@8c34382543f9:/home/64# ls -l -h
total 32M
-rwxrwxr-x 1 500 500 138K Mar 17 2013 gensort
-rwxr-xr-x 1 root root 32M Nov 12 17:08 partition1
-rwxrwxr-x 1 500 500 132K Mar 17 2013 valsort
```



# Library for External Merge Sort

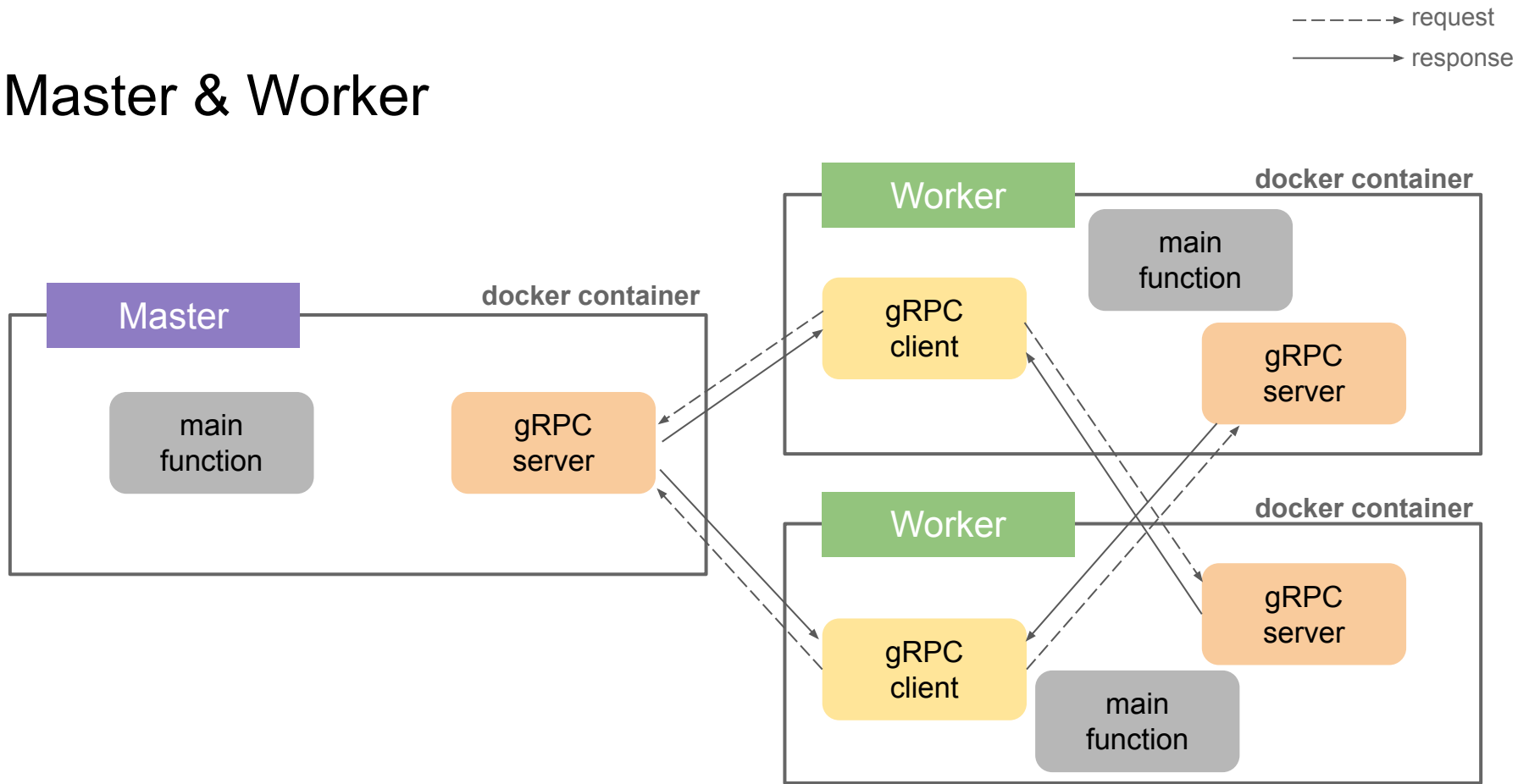
## externalsortinginjava 0.6.1 API

- [API Document](#)
- Dependency added on build.sbt
- Useful Methods

Sorting	<b>ExternalSort.sortInBatch</b> (BufferedReader fbr, long datalength, Comparator<String> cmp, int maxtmpfiles, long maxMemory, Charset cs, File tmpdirectory, boolean distinct, int numHeader, boolean usegzip, boolean parallel)
Sorting & Merging	<b>ExternalSort.mergeSortedFiles</b> (List<File> file, File outputfile)

Design

# Master & Worker

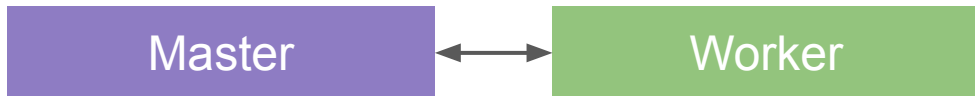


# Message



def workerRegistration (address: String): Boolean

When	worker 자신의 IP address를 파악, 현재 master와 registration이 되어있는지 여부 확인 후
Request	worker IP address
Response	success/fail
Functionality	<ul style="list-style-type: none"><li>- 현재 master에 등록된 worker 개수와 프로그램을 실행함에 있어 필요한 총 worker 개수를 비교</li><li>- 필요한 worker 개수보다 적을 경우, master의 worker list에 추가하고 True를 반환</li><li>- 이미 필요한 worker가 모두 모인 경우 False를 반환</li><li>- 이 때 master에 등록하는 순서대로 worker가 ordering 됨</li></ul>



# Message

```
def getWorkerPivots (address: String, sample: File): List[Worker]
```

When	master와 connection이 완료되고 worker 내부적인 external merge sort가 완료되었을 때
Request	worker에서 sampling을 수행한 file
Response	각 worker의 IP address, pivot 정보
Functionality	<ul style="list-style-type: none"><li>- 각 worker별로 getSample 함수를 통해 sample file을 생성</li><li>- 각 worker별로 IP address와 sample file을 받음</li><li>- 전체 worker에서 sample을 받았으면 setPivot 함수를 통해 각 worker에 해당하는 pivot을 계산</li><li>- 전체 worker에 worker information이 담긴 list를 반환</li></ul>



# Message

```
def getPartitionInfo (address: String, partition: Map[Worker, Int]): Map[Worker, Int]
```

When	partitionByPivot이 완료된 후
Request	해당 worker에서 가지고 있는 pivot(worker)에 따른 partition file 개수에 대한 정보
Response	다른 worker로부터 받아야 하는 file 개수
Functionality	<ul style="list-style-type: none"><li>- getWorkerPivot에서 받은 worker 정보를 기반으로 external merge sort가 수행된 file에 대한 partitioning을 수행</li><li>- pivot에 따른 partition file의 개수를 저장</li></ul>

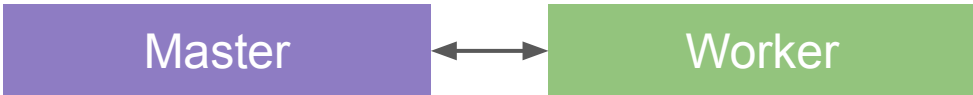




# Message

def shuffling (address: String, count: Int): List[File]

When	getPartitionInfo message response를 받은 후
Request	받아야 하는 <b>file</b> 의 개수를 보냄
Response	<b>request</b> 를 보낸 <b>worker</b> 의 <b>pivot range</b> 에 해당하는 <b>file list</b> 를 보냄
Functionality	- <b>request</b> 를 보낸 <b>worker</b> 에서 요구한 <b>file</b> 의 개수와 실제 <b>machine</b> 에 존재하는 <b>file</b> 의 개수가 맞는지 확인 후 <b>file</b> 을 반환



# Message

def mergeDone (address: String, range:[Long, Long]): Boolean

When	merge가 다 되었을 때 이를 master에 알림
Request	worker 내부의 file 정보와 함께 master에게 merge 완료를 알림
Response	worker의 request를 잘 받았음을 반환
Functionality	<ul style="list-style-type: none"><li>- mergeIntoSortedFile을 통해 worker 내부에서 sorting을 수행</li><li>- 해당 worker에 저장된 file의 min key값과 max key 값을 master에 전송</li></ul>

# Functions

def externalMergeSort (files: List[File]): File

Purpose	worker 내부적인 <b>sorting</b> 을 수행할 때
Input	<b>sorting</b> 을 수행해야 하는 <b>file list</b>
Output	<b>sorting</b> 이 완료된 하나의 <b>file</b>
Functionality	<ul style="list-style-type: none"><li>- external merge sort를 통해 worker 내부적인 <b>sorting</b>을 수행</li><li>- externalsortinginjava library</li></ul>

# Functions

def sampling (file: File): File

Purpose	worker 내부적으로 sampling을 수행
Input	sorting이 완료된 file
Output	sampling된 값들을 저장하는 하나의 file
Functionality	- 주어진 sampling factor를 통해 sorting된 file에 대한 sampling을 수행

# Functions

```
def setPivot (files: List[File]): List[Worker]
```

Purpose	worker로부터 받은 sample을 이용하여 pivot을 계산할 때
Input	worker로부터 받은 sample file list
Output	계산된 pivot값을 가지는 worker list
Functionality	<ul style="list-style-type: none"><li>- worker로부터 받은 sample을 이용하여 각 worker에 해당하는 pivot 값을 계산</li><li>- master에 저장되어 있는 worker information에 pivot 정보 저장</li></ul>

# Functions

```
def partitionByPivot (file: File, workers: List[Worker]):Map[Worker, List[File]]
```

Purpose	worker 내부에서 pivot에 따른 partitioning 수행
Input	- partitioning을 수행할 파일 - 각 worker에 대한 정보
Output	각 worker에 따른 partitioned file list
Functionality	- 각 worker에 따른 pivot을 이용하여 worker 내부 file에 대한 partitioning을 수행

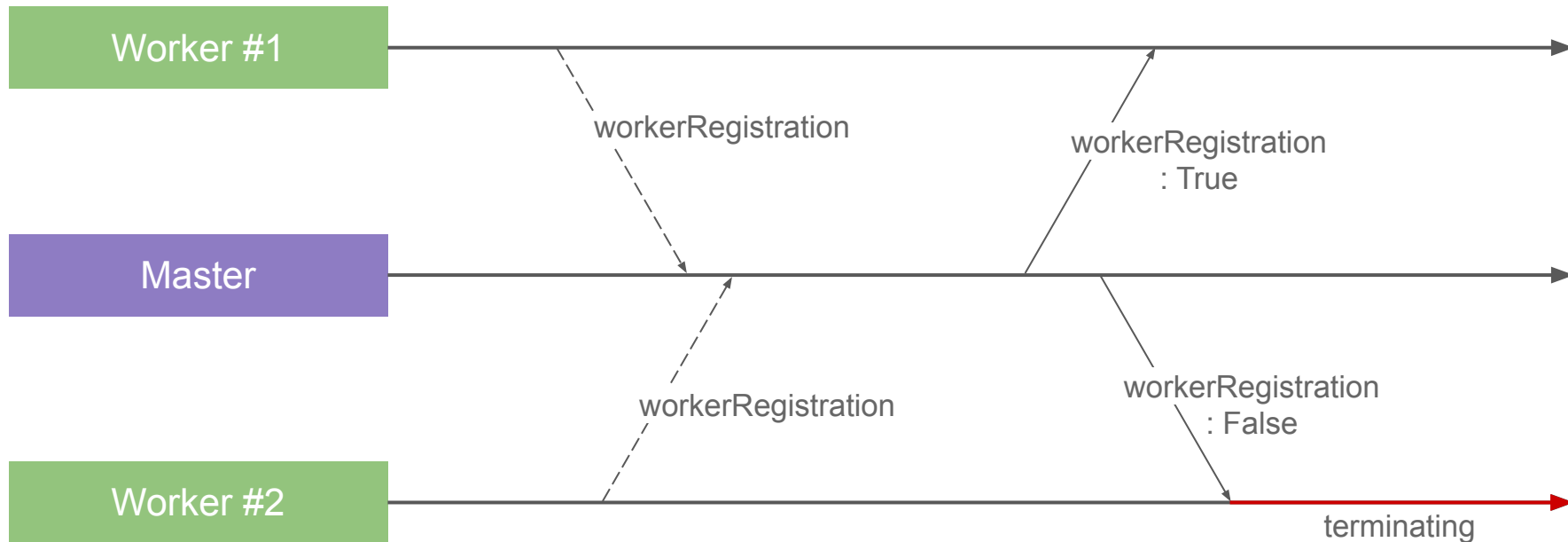
# Functions

def mergeIntoSortedFile(files: List[File]):File

Purpose	shuffling이 완료된 후 worker 내부에서 merging 수행
Input	merge해야 하는 file list
Output	merging이 완료된 하나의 file
Functionality	<ul style="list-style-type: none"><li>- worker들 간 shuffling이 모두 완료되었을 때 실행</li><li>- externalsortinginjava library: ExternalSort.mergeSortedFiles</li></ul>

# Connection

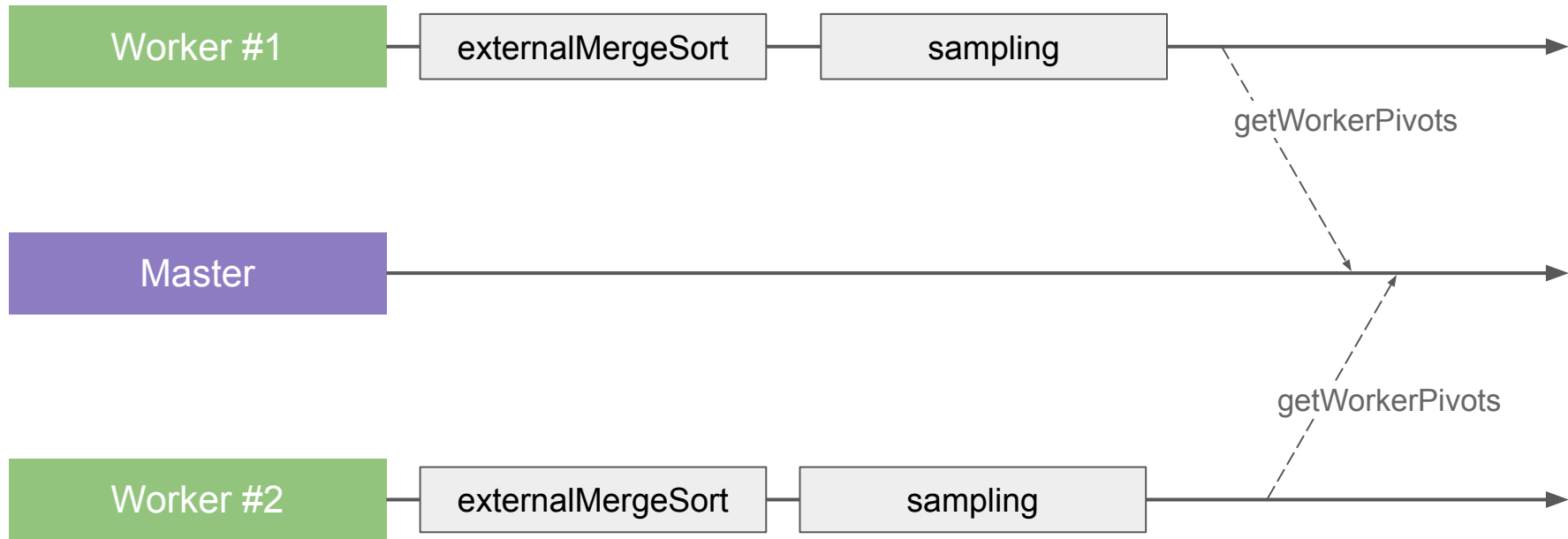
-----> request  
-----> response





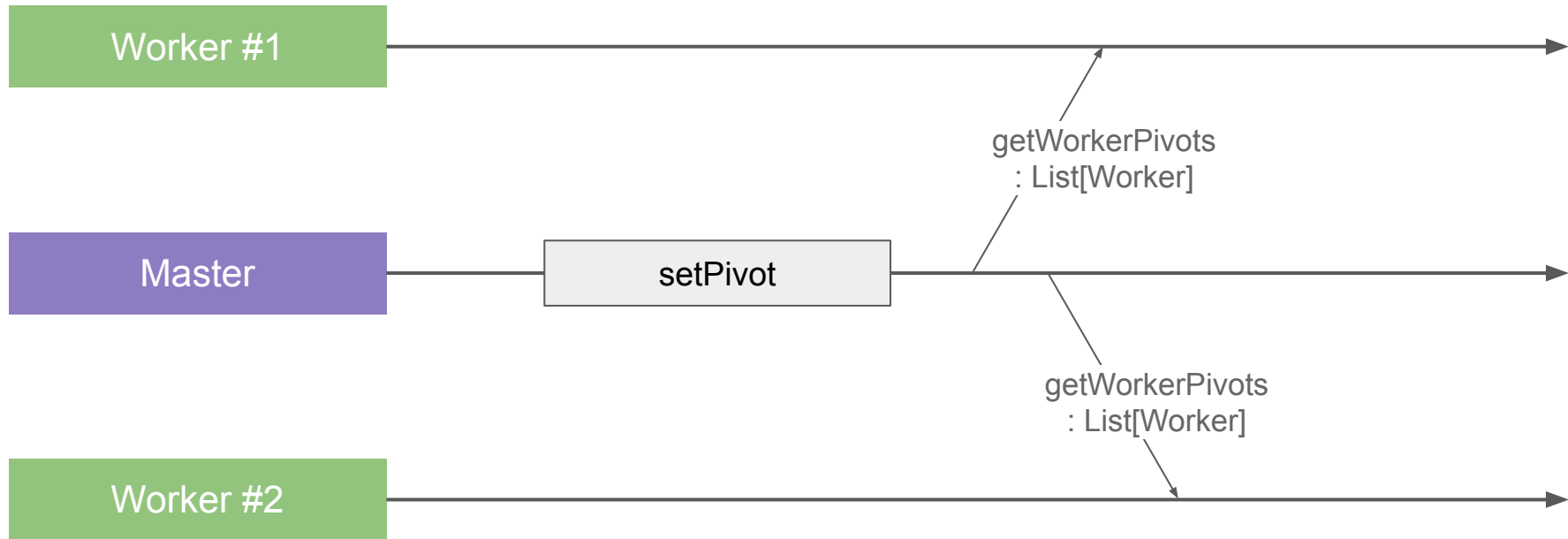
# Sampling

-----> request  
-----> response



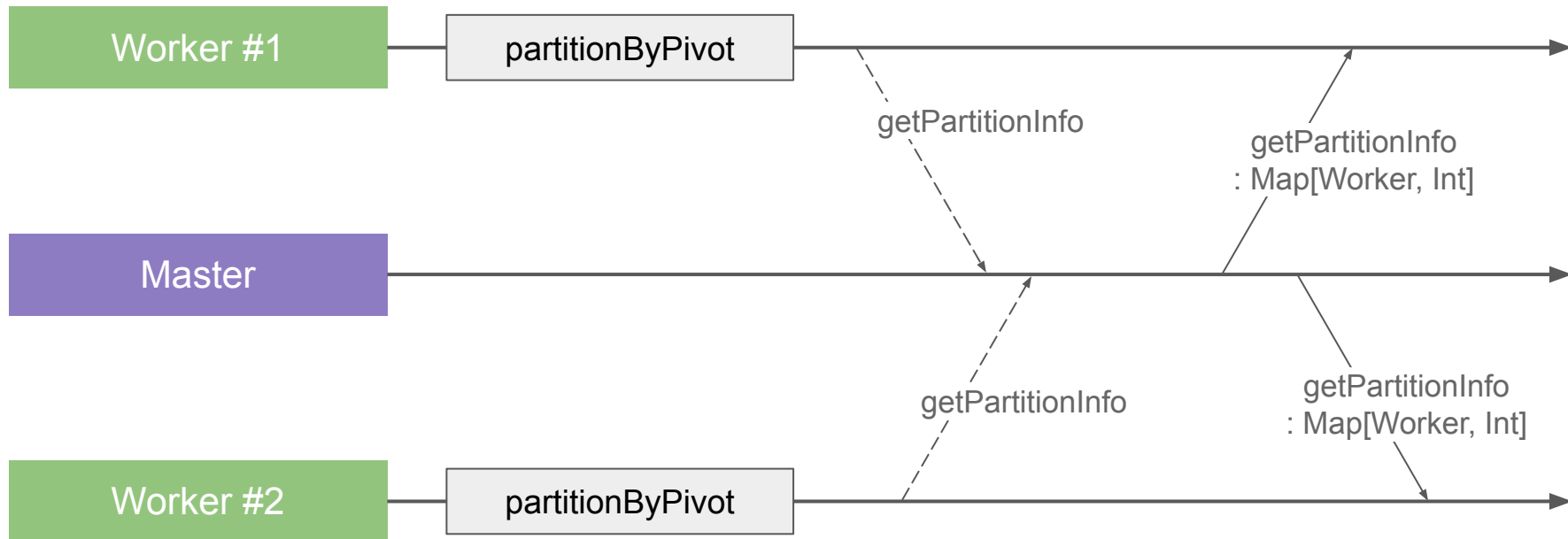
# Pivoting

-----> request  
-----> response



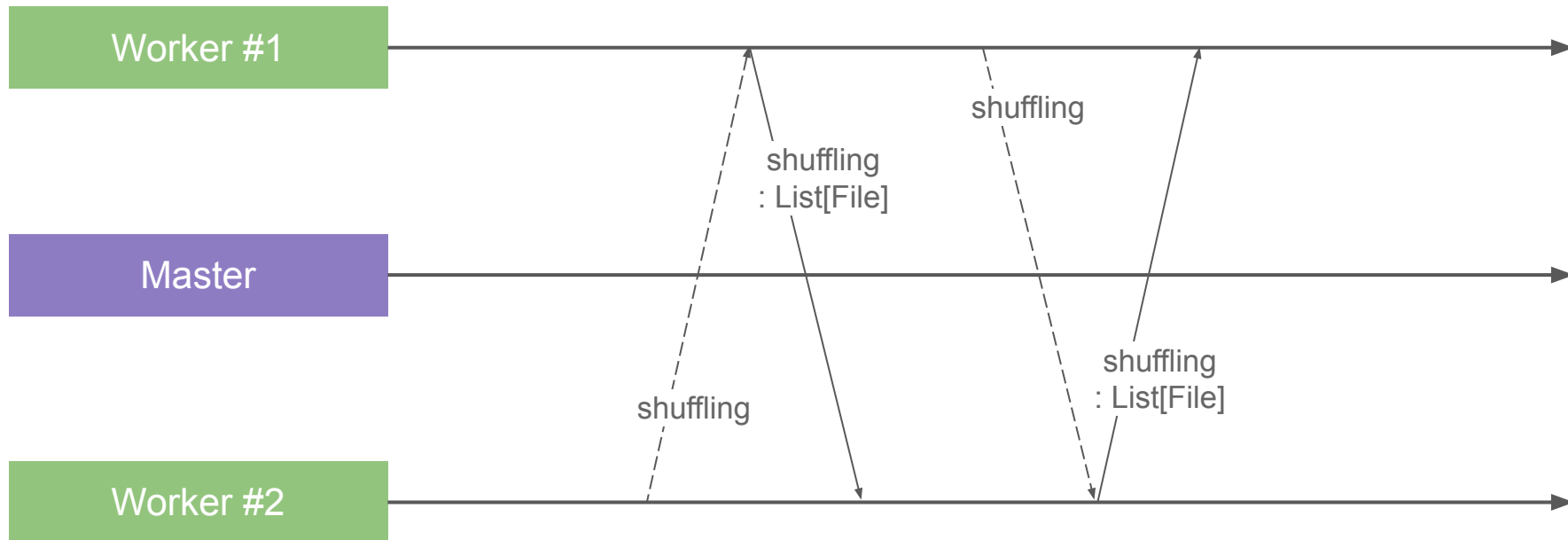
# Partitioning

-----> request  
-----> response



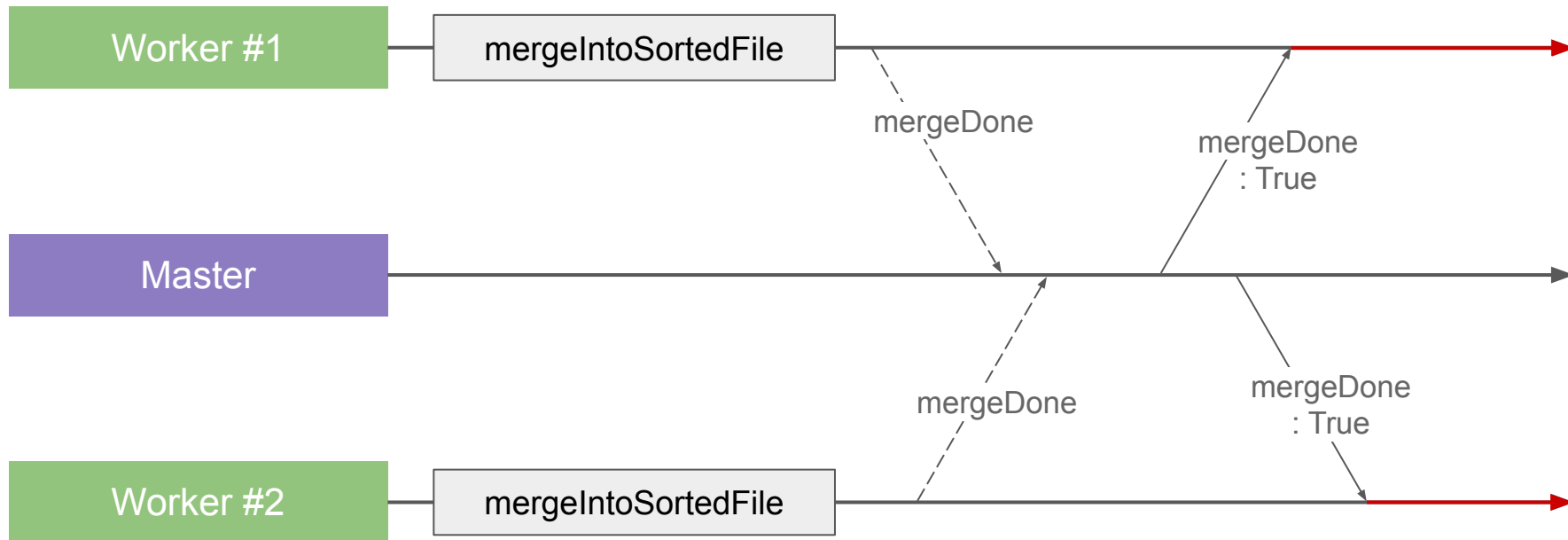
# Shuffling

-----> request  
-----> response



# Merging

-----> request  
-----> response



Progress

# Milestone By Week

**Week 1** 진행 방향 및 공부할 내용



**Week 2** Midterm



**Week 3** Body Design(1) + Testing Environment Setting (1)



**Week 4** Body Design(2) + Testing Environment Setting (2)



**Week 5** Coding 시작 + Design 수정



**Week 6** Coding 마무리 + Unit Test 설계



**Week 7** Debugging + Testing



**Week 8** 마무리



# gRPC Network Progress

gRPC and ScalaPB environment setting



Simple Request-Response Demo



Send Tuple, Array Message



Send File Message



Send List of Files



Register Multiple Services





# gRPC Network Progress

Master가 Client의 Request를 모두 받고 Response를 처리하도록 하기

workerRegistration



getWorkerPivot



Shuffling



mergeDone



## output (partitioned Files)

## externalMergeSort 구현

- |    | test0 ×     | test1 ×                           | test2 ×   |  |
|----|-------------|-----------------------------------|---|--|
| 1  | WH!         | (umONZ-                           | 000000000000000000000000000028FFB                     | 44448888222222444BBB55555553338888AAA2224444 |
| 2  | xL\$!~ GEfw | 000000000000000000000000000028FFC | 11112222DDDBBB1111BBB777711112222FFF3333BBB5555       |  |
| 3  | ts5(qcHZ^m  | 000000000000000000000000000028FFD | 7777BBBBDDD1111EEEE4444DDDAAA7777BBBDDDDCCCCAAAA      |  |
| 4  | Rt0y7u6=#!  | 000000000000000000000000000028FFE | DDDD666611115555DDDDFFFF1111AAAAEEEEE66633377773333   |  |
| 5  | <+>[dXo]-{A | 000000000000000000000000000028FFF | 22229999111155557777888444455555557777000000000000000 |  |
| 6  | c@CS mOf~   | 000000000000000000000000000029000 | 9999AAAAAAAA888866662224444CCCC77775555000000001111   |  |
| 7  | t01W~+xA+_3 | 000000000000000000000000000029001 | 666633366688887777FEFF000088885555AAAA666644446666    |  |
| 8  | I~!w@hOQ)   | 000000000000000000000000000029002 | 44440000EEEE00002225556BBBBBB555533334444DDDDFFFF     |  |
| 9  | xblKL48-6c  | 000000000000000000000000000029003 | 6666BBB33333333BBB9999BBB77779999AAAA11111111CCCC     |  |
| 10 | #efxj19Kl@u | 000000000000000000000000000029004 | BBBB00003332222222EFFF3333BBB1111EEEE11118888DDDD     |  |
| 11 | )YIvI~+r6D  | 000000000000000000000000000029005 | BBB5555AAAA3333EEEEBBBDDDD2222BBB6666999900002222     |  |
| 12 | ChU\$~4)os< | 000000000000000000000000000029006 | CCC3333444466668888BBBFFF5555333355599998888BBB       |  |
| 13 | 7IZM\$YfAXk | 000000000000000000000000000029007 | AAAA3332228888AAAA7777999966666666DDDD444477778888    |  |
| 14 | /J\$Vsy#W   | 000000000000000000000000000029008 | 7777FEFF2222EFFF33338880000QA662222BBB444455559999    |  |
| 15 | vKzk~T&k0   | 000000000000000000000000000029009 | FFFFBBBEEEE7777DDDD22233333333344444441111FFFFEEEE    |  |

```

32 J6,;1,19Xg 000000000000000000000000000000000028FE1 EEEEEEE11111111EEEE1110000EEEEBBB1111111166666666
33 Q$|bz:~aFy 00000000000000000000000000000000002900D 00008888999900001111DDDD777788889999AAAA0003333AAAA
34 Pyw/u16:/g 00000000000000000000000000000000002900F AAAABBBB33337777BBB8BFFFFFFFFFF1111BBB8AAAA9999FFFF0000
35 R18y7u6~#[] 000000000000000000000000000000000028FFE DDD666611115555DDDD0000FFFF1111AAAAEEEE66663333777773333
36 S:a_e~ytL 0000000000000000000000000000000000028FDC BBBB8888FFFF555566668888BBB8FFFF66666666EEEE55555555
37 Ta2_~]3~a~ 000000000000000000000000000000000028FDE AAAAAAAA7777BBB8CCCCAAA6666FFFFBBB86666DDDD11113333
38 VJ1/y_~hEk 0000000000000000000000000000000000028FE3 77772222DDDD7777BBB8AAA2222EEEE6666777700003333CCCC
39 W#1(um0~N2 000000000000000000000000000000000028FBB 44448888222222224444BBB5555555533338888AAAA22224444
40 W|]unrw)b/ 0000000000000000000000000000000000028FDB 2222FFFFFFFFFF444477774444DDDD8888DDDDCC888822220000
41 [tz:~S'M/AH 000000000000000000000000000000000028FF2 2222FFFF88885555222266661111AAAAEEEEEEEEAAAAAAFFFF
42 #efx19K1@u 000000000000000000000000000000000029004 BBBB0000333322222222EEEE3333BBB1111EEEE11118888DDDD
43 ]dr_~W!A:~# 000000000000000000000000000000000028FFA EEEE2222EEEECCCC5555CCCC6666AAAAAAFFFF999944447777
44 _va 15]4J 0000000000000000000000000000000000029013 CCCC00002222444477770000555566668888CCCCDDDD0000CCCC
45 aW#G17F:~cY 000000000000000000000000000000000028FD9 44445555A222222222222DDDEEEE1111888800002222AAAAEEEE
46 cD|B+SFR:~ 000000000000000000000000000000000028FF6 777744447777CCCCDDDD6666FFFF999966664444AAAA5555BBB8

```

## output (a sorted File)

# Merging Progress

## mergeIntoSortedFile 구현

- gensort로 생성된 일부 데이터로 Local 테스트
- Docker 환경에서 확인 필요

```
0 x 1 x 2 x
1 |"I*wh@_.) 000000000000000000000000000029002 44440000EEEE000022225555BBB8BB8B555533334444DDDDFFFF
2 |)YIV!&*r6Q 000000000000000000000000000029005 BBB85555AAAA3333EEEEBB8BDDDD2222BB8B6666999900002222
3 |, ]/'sYvB# 000000000000000000000000000029008 7777FFFF2222EEEE333388880000AAAA2222BB8B444455559999
4 |,qNs3 1Ye: 00000000000000000000000000002900B BBB8BB8B333366660000DDDD4444AAAA8BB8CCCCBB8B11114444
5 |1s5(cqHZ"m 000000000000000000000000000028FDD 7777BB8BDDDD1111EEEE4444DDDDAAAA7777BB8BDDDDCCCCAAAA
6 |5}|B0]"sQ0 00000000000000000000000000002900C CCCC777711116666DDDDCCCC7777888833331111444466665555
7 |7IZM$VfAxK 000000000000000000000000000029007 AAAA333322228888AAAA7777999966666666DDDD444477778888
8 |<+dU6xo|~A 000000000000000000000000000028FFF 2222999911115555777788884444555555557777000000000000
9 |B$MI ZzD95 00000000000000000000000000002900A CCCC44443333888888882222DDDD0000007777FFFFDDDD77777777
10 |ChU$^4)os< 000000000000000000000000000029006 CCCC3333444466668888BB8BFFFF55553333555599998888BB8B
11 |OS|bz:<aPy 00000000000000000000000000002900D 00008888999900001111DDDD777788889999AAAA00003333AAAA
12 |Pyw/uIG:/g 00000000000000000000000000002900F AAAABBB833337777BB8BFFFFF1111BB8BAAA9999FFFF0000
13 |R1@y7u6=#( 000000000000000000000000000028FFE DDD666611115555DDDDFFFF1111AAAAEEEE6666333377773333
14 |WHI(umONZ~ 000000000000000000000000000028FFB 444488882222224444BB8B5555555533338888AAAA22224444
15 |#efx!9K1@u 000000000000000000000000000029004 BBB800003333222222EEEE3333BB8B1111EEEE11118888DDDD
```

input (sorted Files)

```
testOutput x
32 |JG,:1,|9Xg 000000000000000000000000000028FE1 EEEEEEE11111111EEEE11110000EEEE888B1111111166666666
33 |OS|bz:<aPy 00000000000000000000000000002900D 00008888999900001111DDDD777788889999AAAA00003333AAAA
34 |Pyw/uIG:/g 00000000000000000000000000002900F AAAABBB833337777BB8BFFFFF1111BB8BAAA9999FFFF0000
35 |R1@y7u6=#( 000000000000000000000000000028FFE DDD666611115555DDDDFFFF1111AAAAEEEE6666333377773333
36 |S:a_e~yt_8 000000000000000000000000000028FDC BBB8FFFFF555566669999BB8BFFFF66666666EEEE55555555
37 |Ta?_]S^a* 000000000000000000000000000028FDE AAAAAAA7777BB8BCCCCAAA6666FFFFBB8B6666DDDD11113333
38 |VJ|/y_>hEk 000000000000000000000000000028FE3 77772222DDDD7777BB8BAAA2222EEEE6666777700003333CCCC
39 |WHI(umONZ~ 000000000000000000000000000028FFB 444488882222224444BB8B5555555533338888AAAA22224444
40 |W|]vnrw)b/ 000000000000000000000000000028FDF 2222FFFFFFFFFF444477774444DDDD8888CCCC888822220000
41 |[tz:s'M/aH 000000000000000000000000000028FF2 2222FFFF88885555222266661111AAAAEEEEEEEEAAAAAAFFFF
42 |#efx!9K1@u 000000000000000000000000000029004 BBB800003333222222EEEE3333BB8B1111EEEE11118888DDDD
43 |[dr_!IA[z# 000000000000000000000000000028FFA EEEE2222EEEECCCC5555CCCC6666AAAAAAFFFF999944447777
44 |_~va l5|4J 000000000000000000000000000029013 CCCC00002222444477770000555566669999CCCCDDDD0000CCCC
45 |aWwG!7f|cY 000000000000000000000000000028FD9 44445555AAAA8888EEEEDDDDEEEE1111888800002222AAAAEEEE
46 |cD|B+sFR_~ 000000000000000000000000000028FF6 777744447777CCCCDDDD6666FFFF999966664444AAAA5555BB8B
```

output (a sorted File)