Team Pink
Progress

20200445 박은하
20200927 장유진
20200220 오상윤

Review of your weekly progress

1 week >

milestone 설정

Distribution sorting 방법 이해

2 week >

sorting 에서 master와 worker 사이의 networking process이해

Network message design

3 week >

grpc 사용법 이해 및

simple server client program 구현

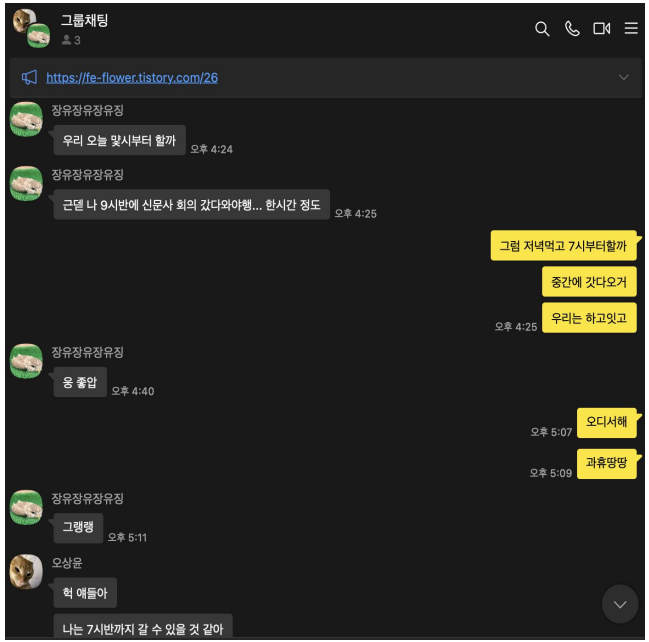
4 week >

grpc master and worker connection

network message design modification

module architecture design

Communication method



server를 만드는 코드는 다음을 참고하면 될 것 같다.

https://github.com/xuwei-k/grpc-scala-sample/blob/master/grpc-scala/src/main/scala/io/grpc/examples/helloworld/HelloWorldServer.scala

client >

client를 만드는 코드는 다음을 참고하면 될 것 이다.

https://github.com/xuwei-k/grpc-scala-sample/blob/master/grpc-scala/src/main/scala/io/grpc/examples/helloworld/HelloWorldClient.scala

gRPC 예시 관련 참고 링크

gRPC | ScalaPB

grpc-scala-sample/build.sbt at master · xuwei-k/grpc-scala-sample (github.com)

Communication method

- Rules for git Commits
  - git commit -m “[Keyword]: contents(what I modified)”
  - Keywords: Fix, Add, Remove, Simplify, Update, Implement, Prevent, Move, Rename

- Examples
  - Fix: myFunction to return something
  - Add: myFile.scala for something
  - Update: myFunction to use someFunction for something

Communication method

! 구현 나눠서 하기로 결정 !

sangyoon : worker module implementation

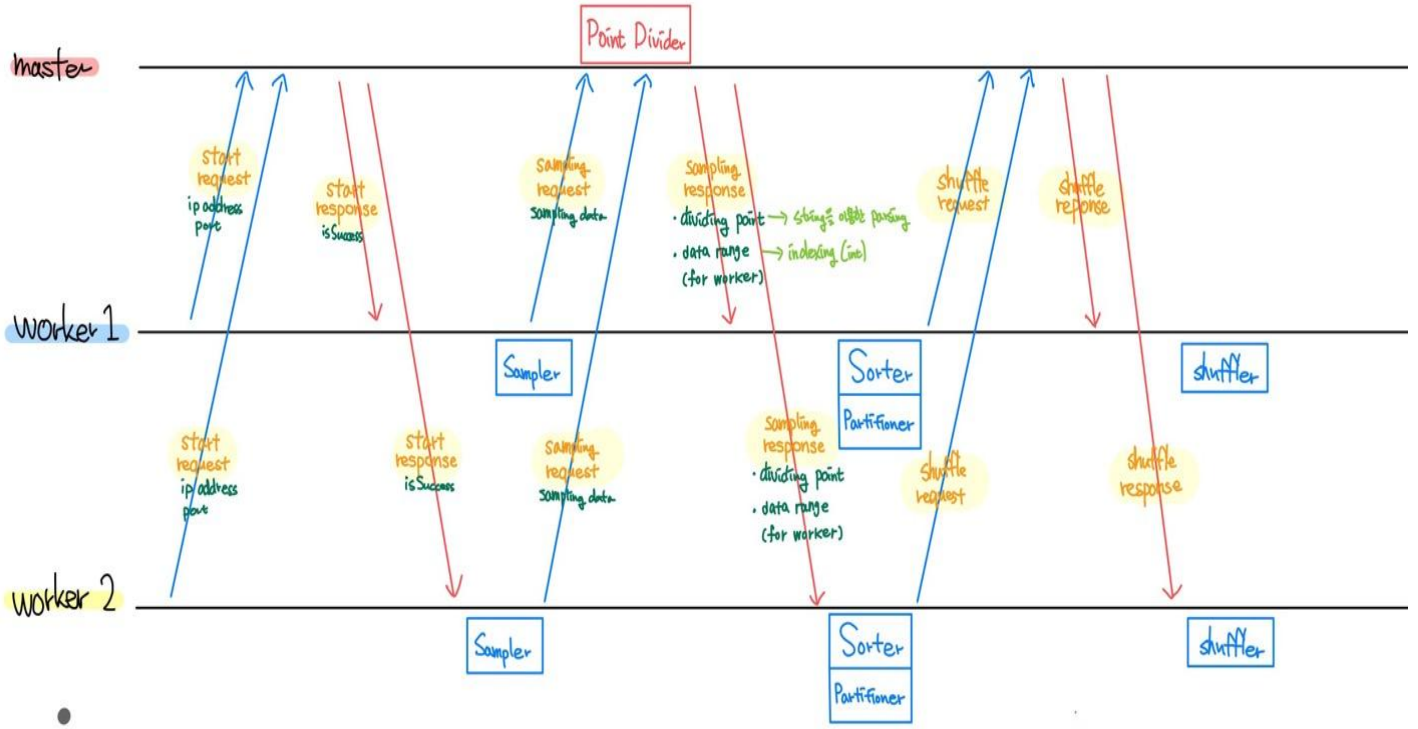
yoojin : sampler(worker) and master module implementation

eunha : network package implementation

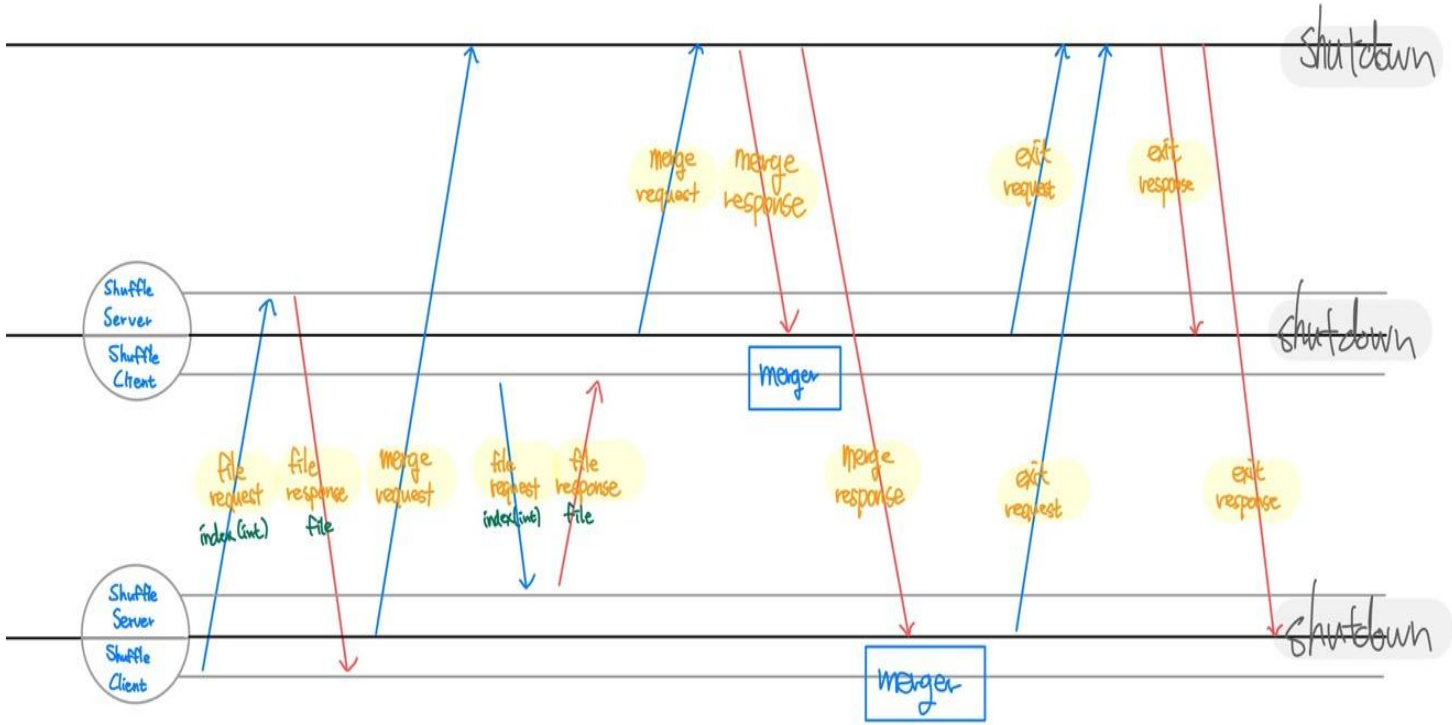
Environment

- Programming environment
  - Windows
  - JDK 1.8.0
  - Scala 2.13.10
  - SBT 1.8.0
  - Logger :java.util.logging.Logger
- Libraries
  - gRPC with ScalaPB
- generate input data
  - gensort 사용법을 익히기 위해 sample data를 만들어보았다.

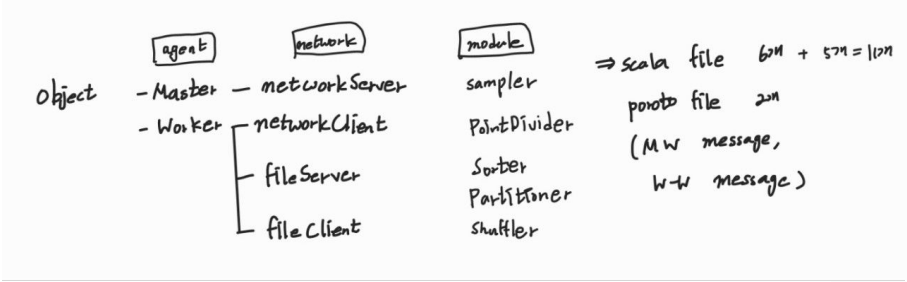
Design - Flow Diagram

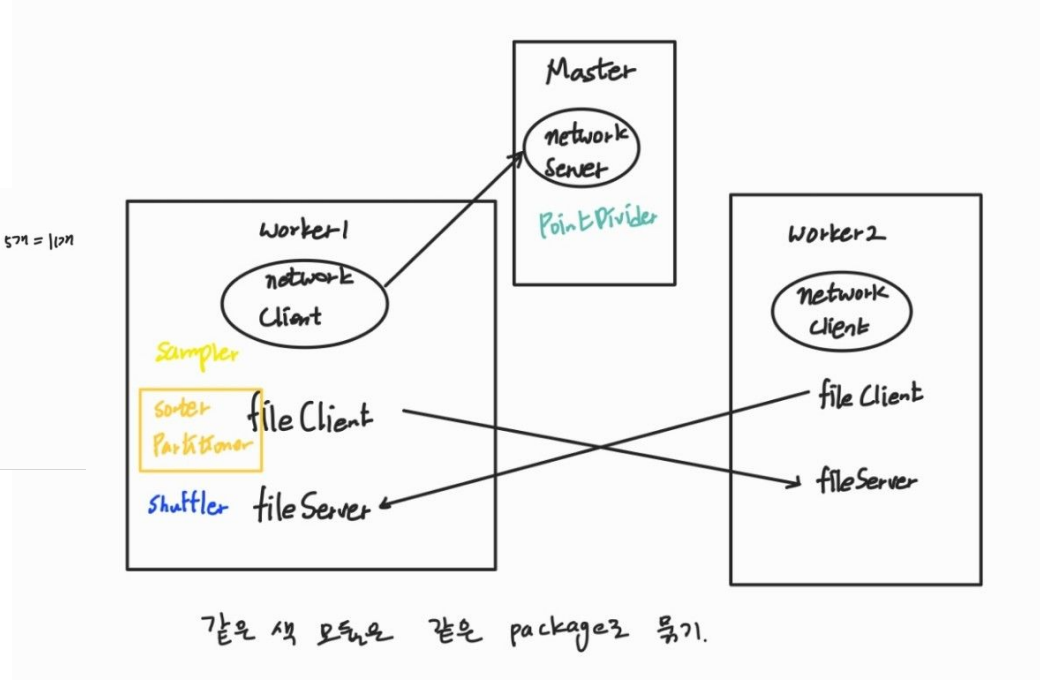


Design - Flow Diagram



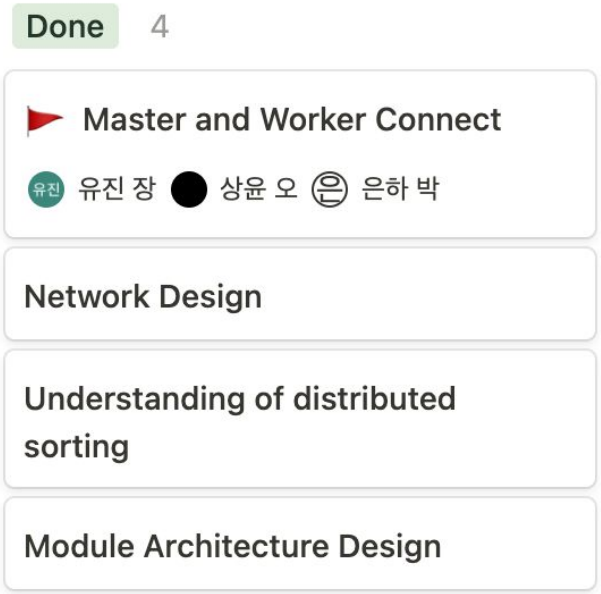
Design - Relations between modules





Progress

Finished milestone>



We had only

- worker sorting module implementation
- master sorting module implementation

as a **remaining milestone**

We realized that

Milestone was too simplified…

Modules and components are not implemented yet….

But we designed them, so according to it we will implement **as quickly as possible.**

Milestone of remained weeks

5 week >

[Network] Master and Workers Connection with command line

[Network] Sending data type design

[Network] protobuf and master and worker network connector implementation

[Master] Point Divider module implementation

[Worker] Sampler module implementation

Milestone of remained weeks

6 week >

[Network] network between workers implementation (shuffle client and shuffle server)

[Worker] module implementation

7 week >

[Test] Two Worker and one file test

[Test] Multi Worker and multi file test

Milestone of remained weeks

8 week>

[Test] Test bigger input with multiple Workers using Docker

Seek Advice

- How do we check the IP address and port number of the machines

- Is there extra memory in machines? We worry about the shuffling because we don’t sure that sampling is exactly correct.

- How can we divide testing jobs?

Thank You

Team Pink