

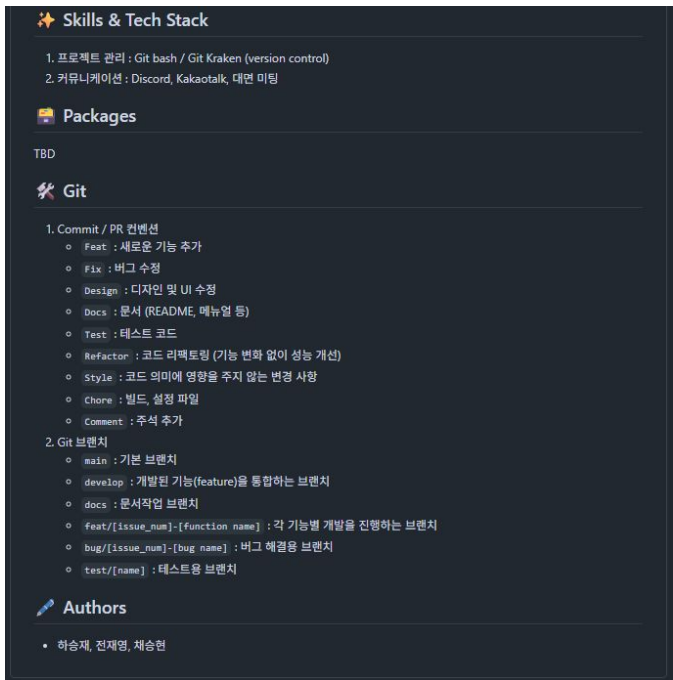
# SD progress presentation

team white

# Contents

1. how we manage the project?
2. master-worker connection
3. sorting algorithm
4. project class structure - class diagram
5. weekly progress & milestone

# 1. Management



**github:** version control system

+ **gitkraken, fork**

**discord:** save file, online meeting

**kakaotalk:** simple communication

**google drive:** make ppt file, report

**boardmix:** make UML

git convention

[illegible]

```
sorted_key.py test_valsort x test_valsort x test_valsort x  
# test_valsort  
1 |_w|EBj0 000000000000000000000000000E4 B8BBD0D0111888888AAAA11144499996666332222DD000  
2 A-wJ[-t] 0000000000000000000000000009B B8B877771118BBBAAs5550000FFf111222444888B4444  
3 cP0`ZvWw 000000000000000000000000000310 AAAA6666FFFFCC9999FFfAAAs3336666DD009996666EE  
4 fTz[D].Dy 0000000000000000000000000003AB FFFf77777777222200006666BBBBCCCCC4444DD0077774444  
5 TTrU--da 0000000000000000000000000001F EEfEBBfCCCCCBBAAff777777CCCAAAffEEfEEf00000000  
6 00`P[skv 0000000000000000000000000001A 666699997777333EEfEEfEEf4444111EEfEEf44499993337777  
7 P3I8 /?> 000000000000000000000000000121 DDD888877775554447f7779999777888844442222226666  
8 [3L;fbLm 000000000000000000000000000385 AAAA444888877778000555111DD00888555999988882222  
9 jPw[sG_1 000000000000000000000000000300 66660000EEf5555BBB84444BBBfCCCC5555DD000111170001111  
10 sA[P]rFf 000000000000000000000000000000  
11 "HVLv..tl 000000000000000000000000000000 Windows PowerShell x Windows PowerShell x Ubuntu x +  
12 rIZ_O0.bq 000000000000000000000000000000  
13 lP(Lc-zL 000000000000000000000000000000 jjong@DESKTOP-0300964:~/test$ valsort test_valsort  
14 unL[oc"- 000000000000000000000000000000 Records: 1000  
15 sJ[L]-E - 000000000000000000000000000000 Checksum: 1f5dfb3631a  
16 f'-EHBS"D 000000000000000000000000000000 Duplicate keys: 0  
17 !bu_pzo[U 000000000000000000000000000000  
18 [CVr|>br$? 000000000000000000000000000000 SUCCESS - all records are in order  
19 tW_dfr,n7 000000000000000000000000000000 jjong@DESKTOP-0300964:~/test$  
20 VUSZ55fqL 000000000000000000000000000000  
21 I[W0E:5Yn 000000000000000000000000000000  
22 [d-ZZU)N; 000000000000000000000000000000  
23 [f' m[Z.B" 000000000000000000000000000000  
24 [FX;-0]68 000000000000000000000000000000  
25 "w"o~>pP 000000000000000000000000000000  
26 "r[uagzcZ 000000000000000000000000000000  
27 "&"gkE]Ml 000000000000000000000000000000  
28 "vud_eFA 000000000000000000000000000000  
29 ".1nkZ-nK 000000000000000000000000000000  
30 "/?CzRlZ 000000000000000000000000000000  
31 "2k0~-fb,y 000000000000000000000000000000  
32 "3tPlp-jH 000000000000000000000000000000  
33 "xub_mZEg 000000000000000000000000000000  
34 "pv_XSPg3 000000000000000000000000000000
```

# valsort

## 2.b. Build Libraries Setting

**OS:** window 11

+ **WSL**(CentOS 7 or Ubuntu)

**Java:** jdk 8 (JDK 8u202)

**Scala:** scala 3.3.7

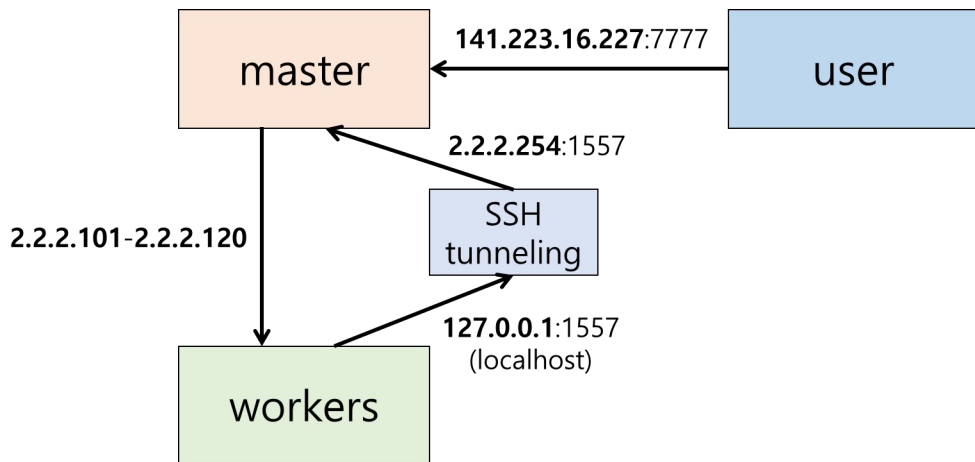
**Libraries:** Scalapb, netty, protobuf, gRPC

```
white@vm-1-master:~  
[white@vm-1-master ~]$ cat /etc/os-release  
NAME="CentOS Linux"  
VERSION="7 (Core)"  
ID="centos"  
ID_LIKE="rhel fedora"  
VERSION_ID="7"  
PRETTY_NAME="CentOS Linux 7 (Core)"
```

```
white@vm01: ~  
white@vm01:~$ cat /etc/os-release  
NAME="Ubuntu"  
VERSION="18.04.3 LTS (Bionic Beaver)"  
ID=ubuntu  
ID_LIKE=debian  
PRETTY_NAME="Ubuntu 18.04.3 LTS"
```

**master** uses CentOS, **worker** uses Ubuntu

## 2.c. server connection test



could be automated with JSch

The image displays two terminal windows. The left window shows the execution of a Java program on the master node, which outputs a message in Korean indicating the start of the gRPC server. The right window shows the execution of the same Java program on a worker node, which outputs a message in Korean indicating the receipt of a connection request from the master. Below these, a third terminal window shows the successful execution of an SSH command from the worker node to the master node, establishing a connection.

```
[white@vm-1-master ~]$ java -cp 332project-assembly-0.1.0-SNAPSHOT.jar com.example.GreeterServer
gRPC 서버가 0.0.0.0:1557 에서 시작되었습니다.
워커 노드의 접속을 대기합니다...
```

```
white@vm01:~$ java -cp 332project-assembly-0.1.0-SNAPSHOT.jar com.example.GreeterClient
마스터 서버 (127.0.0.1:1557)로 요청 전송: Worker No. 1
서버로부터 응답 받음: Hello from Master, Worker No. 1
클라이언트 종료.
white@vm01:~$
```

```
white@vm-1-master ~$ ssh -L 1557:localhost:1557 -p 7777 white@vm01
Last login: Tue Nov 18 16:01:31 2025 from 172.17.1.1
[white@vm-1-master ~]$
```

connection succeeded

## 3.a. sample sort

step	to do
sampling	for each worker, sample random K data (ex. first 1000 of data).
partitioning	master collects and sorts all samples, then finds the pivots (N-1) to divide them into ranges corresponding to the number of workers (N). → informs this pivot information back to all workers.
shuffle	each worker sorts its data and compares this against the pivots provided. then send them to each worker.
merge	each worker bother the sended data and combine them by k-way merge.

in short, **quick sort with multiple pivots!**

### 3.a. sample sort - **if worker is dead**

if worker is dead, worker is going to resurrect soon.

but **input data will be still remained.**

a resurrected worker gets the overall progress of sample sort at the master.

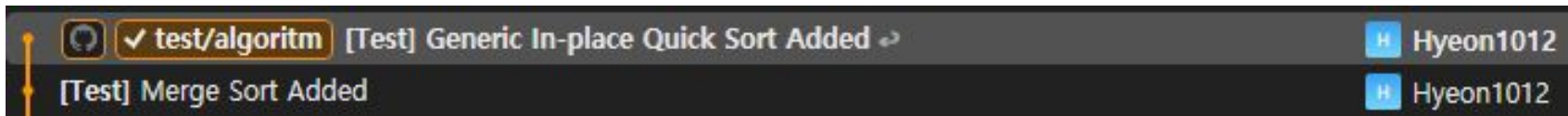
→ **do something** (for step 1 ~~~, for step 2 ~~~ , for step 3/4 ~~~)

**key idea : worker creates and saves temporary files before send data to other worker**

→ **even if one worker died, we can easily resume the shuffle / merge.**



## 3.b. k-way merge sort



```
[Test] Generic In-place Quick Sort Added
```

- merge sort removed
- I think that in-place quick sort is more efficient than merge sort because it can save memory

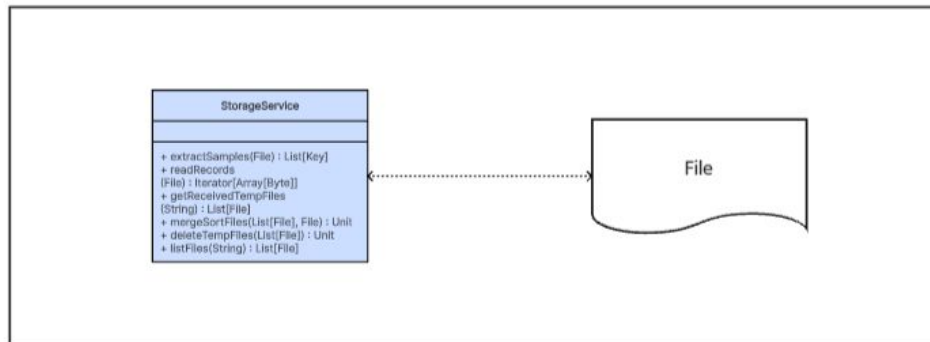
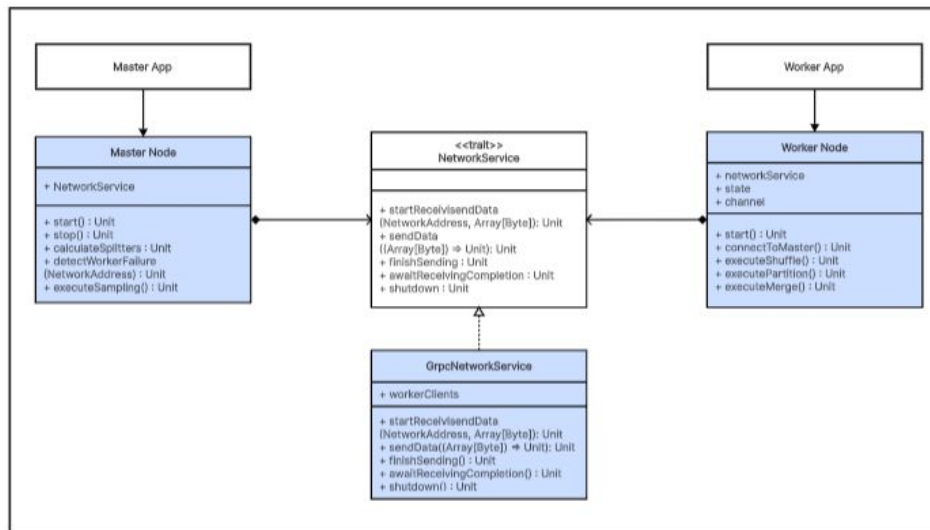
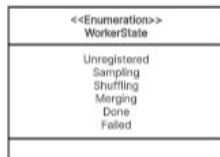
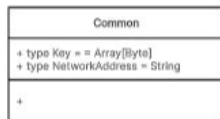
in fact, **quick sort** is more efficient/faster than merge sort for smaller datasets

so, I implemented **generic in-place quick sort**

but I noticed that I can use [java.util.Arrays.sort\(\)](#)

**I have to implement k-way merge by using this**

# Class Diagram



# Review of Weekly Progress

**Week 1 Setup & Planning:** Project initialization, GitHub repository setup, and establishing code conventions and meeting schedules.

**Week 2 System Architecture:** Designed the high-level system architecture and discussed core concepts for distributed sorting and tech stack.

**Week 3 Environment & Tooling:** Configured the development environment, secured VPN/IP access, and set up essential tooling (UML, IntelliJ).

**Week 4 Core Prototyping:** Began implementation with gRPC/protobuf (using ScalaPB), created Master/Worker prototypes, and generated sample data.

**Week 5 Sorting Logic & Integration:** Drafted class diagrams, implemented core sorting logic (QuickSort), and prepared for code integration and testing.

# MileStone

Milestones

New milestone

Open5Closed0

Sort

<u>Test running program on multiple machines</u> <div>⚠ Overdue by 3 day(s) • Due by November 14, 2025</div>	<div></div> <div>0% complete0 open0 closed</div>
<u>Implement distributed Sorting algorithm</u> <div>⚠ Overdue by 3 day(s) • Due by November 14, 2025 • 0/1 issues closed</div>	<div></div> <div>0% complete1 open0 closed</div>
<u>System test, all components in hands</u> <div>Due by December 5, 2025</div>	<div></div> <div>0% complete0 open0 closed</div>
<u>Component test and early system test</u> <div>Due by November 21, 2025</div>	<div></div> <div>0% complete0 open0 closed</div>
<u>Project Deadline</u> <div>Due by December 7, 2025</div>	<div></div> <div>0% complete0 open0 closed</div>

Milestones

New milestone

Open5Closed0

Sort

<u>Implement distributed Sorting algorithm</u> <div>Due by November 25, 2025 • 0/1 issues closed</div>	<div></div> <div>0% complete1 open0 closed</div>
<u>Test running program on multiple machines</u> <div>Due by November 25, 2025</div>	<div></div> <div>0% complete0 open0 closed</div>
<u>Component test and early system test</u> <div>Due by November 28, 2025</div>	<div></div> <div>0% complete0 open0 closed</div>
<u>System test, all components in hands</u> <div>Due by December 5, 2025</div>	<div></div> <div>0% complete0 open0 closed</div>
<u>Project Deadline</u> <div>Due by December 7, 2025</div>	<div></div> <div>0% complete0 open0 closed</div>