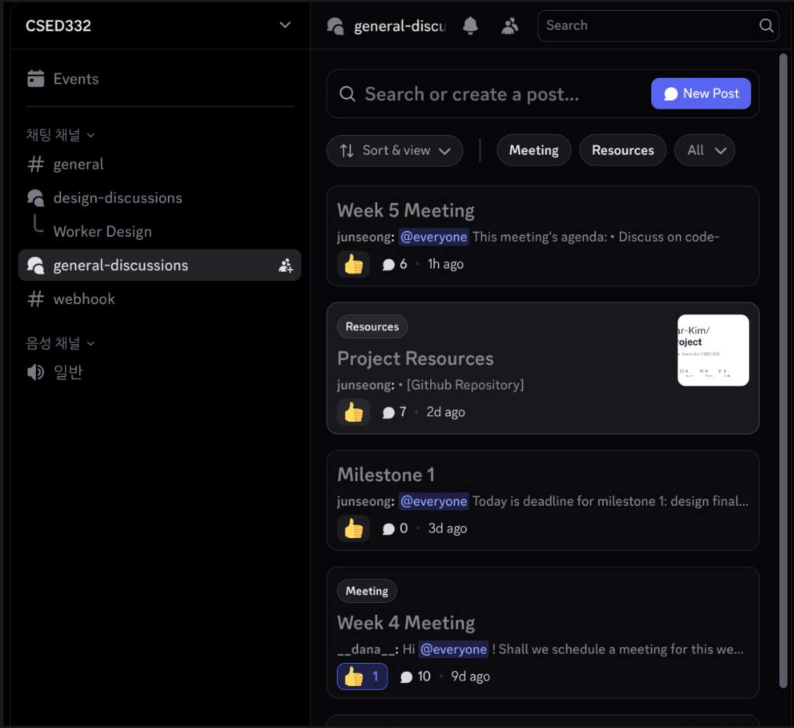
PROGRESS PRESENTATION

Team Red

Team Organization

Communication



CSED332

Events

채팅 채널

general

design-discussions

Worker Design

general-discussions

webhook

음성 채널

일반

general-discu

Search

Search or create a post...

New Post

Sort & view

Meeting

Resources

All

Week 5 Meeting

junseong: @everyone This meeting's agenda: • Discuss on code-

6 · 1h ago

Resources

Project Resources

junseong: • [Github Repository]

7 · 2d ago

Milestone 1

junseong: @everyone Today is deadline for milestone 1: design final...

0 · 3d ago

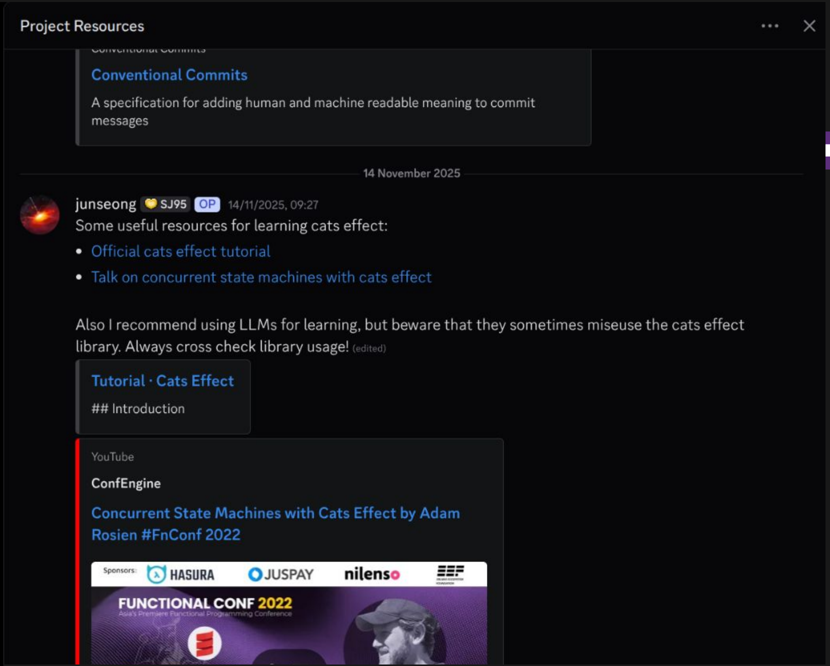
Meeting

Week 4 Meeting

__dana__: Hi @everyone ! Shall we schedule a meeting for this we...

1 · 10 · 9d ago

Separate threads per topics



Project Resources

Conventional Commits

A specification for adding human and machine readable meaning to commit messages

14 November 2025

junseong SJ95 OP 14/11/2025, 09:27

Some useful resources for learning cats effect:

- Official cats effect tutorial
- Talk on concurrent state machines with cats effect

Also I recommend using LLMs for learning, but beware that they sometimes miseuse the cats effect library. Always cross check library usage! (edited)

Tutorial · Cats Effect

Introduction

YouTube

ConfEngine

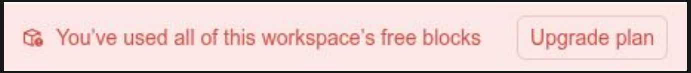
Concurrent State Machines with Cats Effect by Adam Rosien #FnConf 2022

Threads also act as a part of documentation

Communication

Weekly meetings

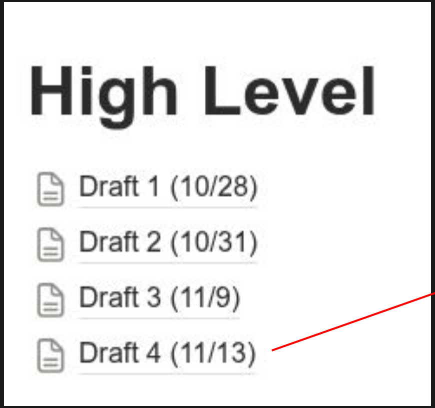
Notion



You've used all of this workspace's free blocks

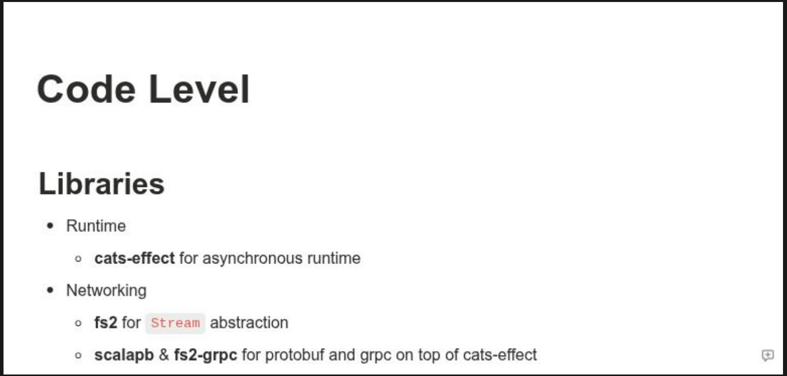
~~Notion~~ → git for documentation

Documentation



High Level

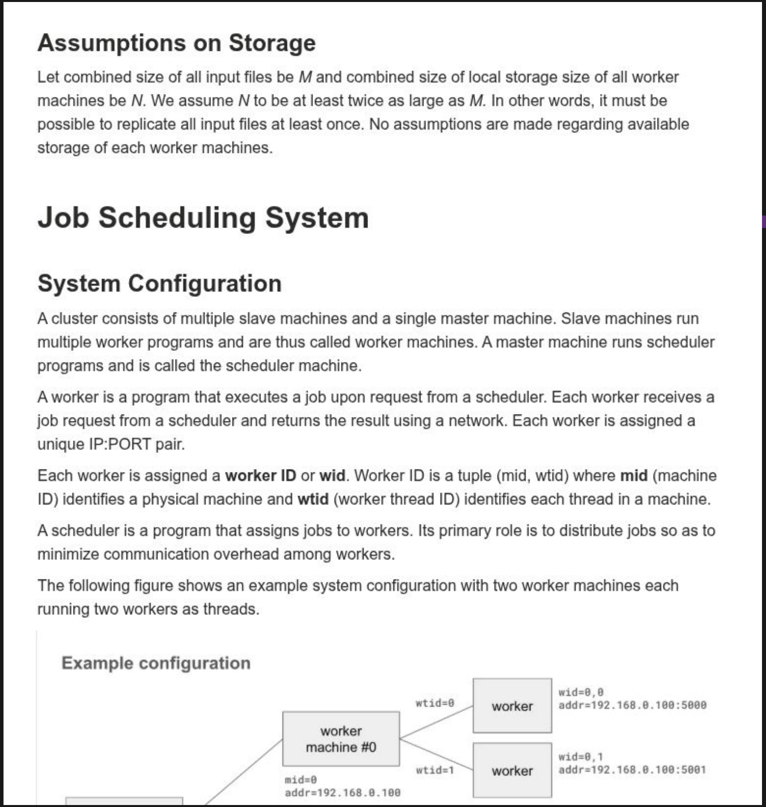
- Draft 1 (10/28)
- Draft 2 (10/31)
- Draft 3 (11/9)
- Draft 4 (11/13)



Code Level

Libraries

- Runtime
  - **cats-effect** for asynchronous runtime
- Networking
  - **fs2** for `Stream` abstraction
  - **scalapb** & **fs2-grpc** for protobuf and grpc on top of cats-effect



Assumptions on Storage

Let combined size of all input files be *M* and combined size of local storage size of all worker machines be *N*. We assume *N* to be at least twice as large as *M*. In other words, it must be possible to replicate all input files at least once. No assumptions are made regarding available storage of each worker machines.

Job Scheduling System

System Configuration

A cluster consists of multiple slave machines and a single master machine. Slave machines run multiple worker programs and are thus called worker machines. A master machine runs scheduler programs and is called the scheduler machine.

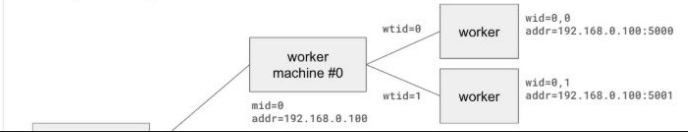
A worker is a program that executes a job upon request from a scheduler. Each worker receives a job request from a scheduler and returns the result using a network. Each worker is assigned a unique IP:PORT pair.

Each worker is assigned a **worker ID** or **wid**. Worker ID is a tuple (mid, wtid) where **mid** (machine ID) identifies a physical machine and **wtid** (worker thread ID) identifies each thread in a machine.

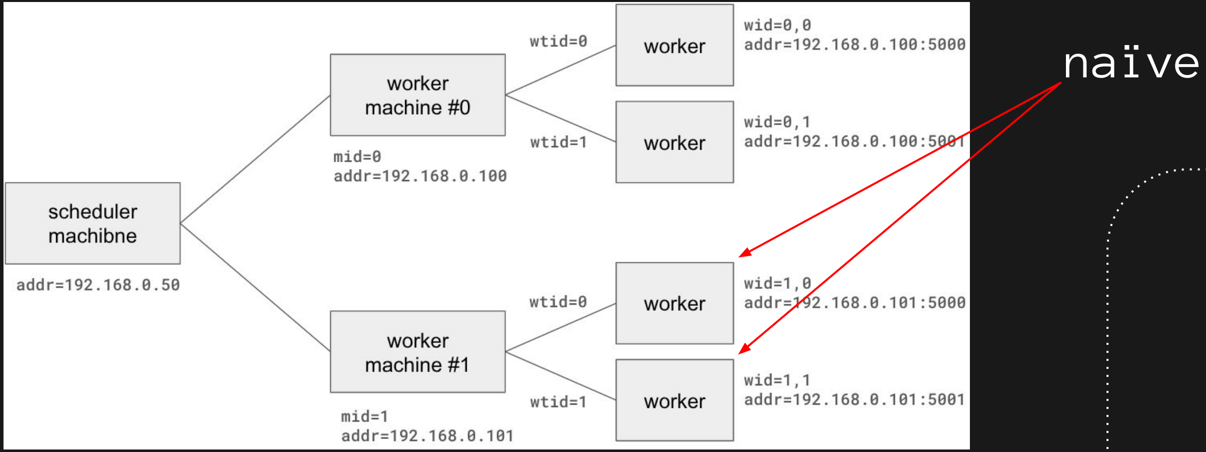
A scheduler is a program that assigns jobs to workers. Its primary role is to distribute jobs so as to minimize communication overhead among workers.

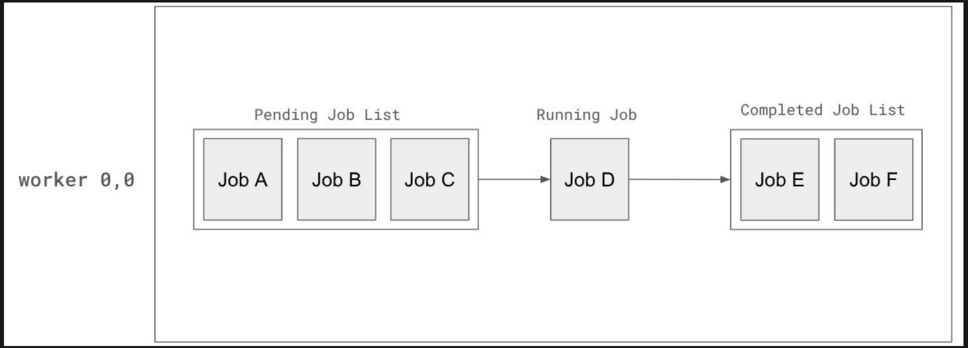
The following figure shows an example system configuration with two worker machines each running two workers as threads.

Example configuration



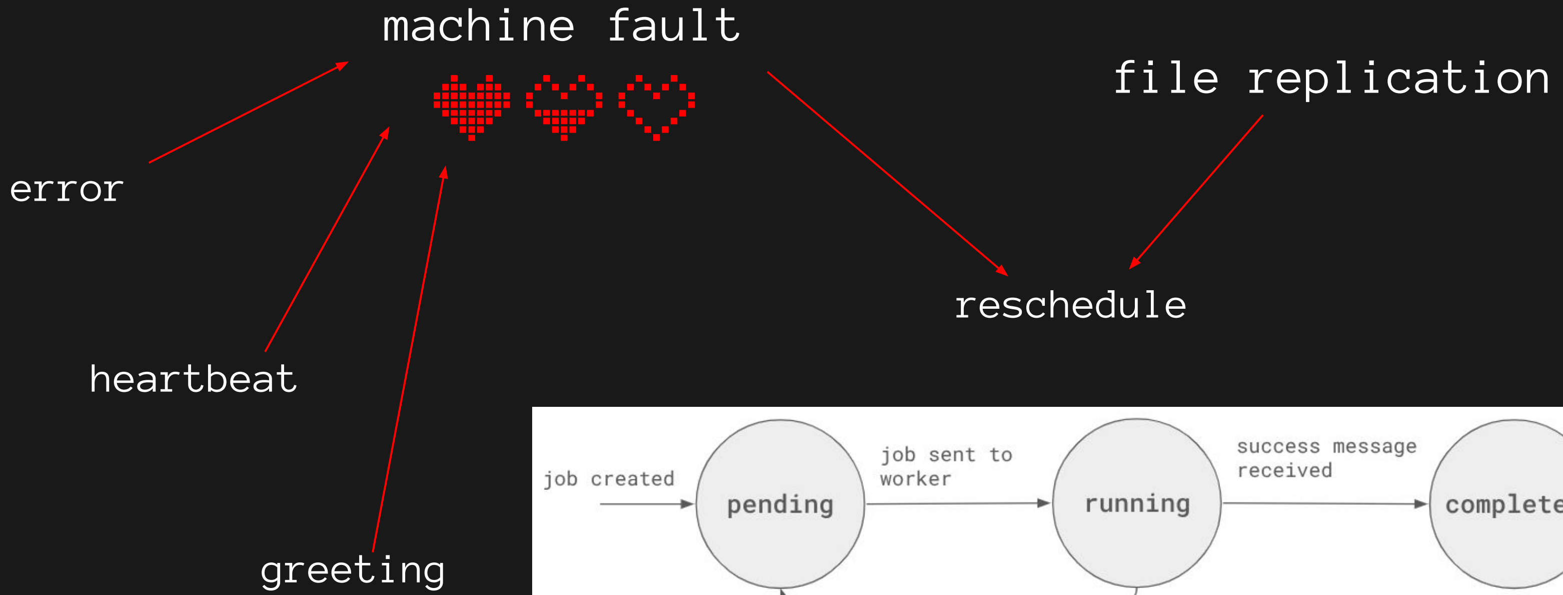
Design - High Level

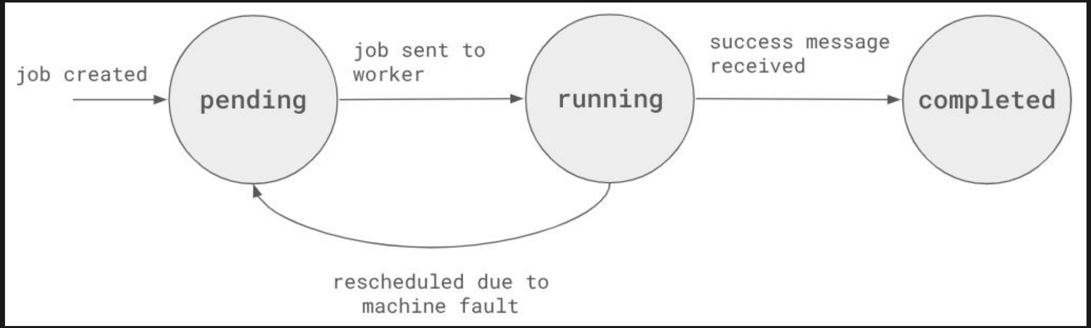




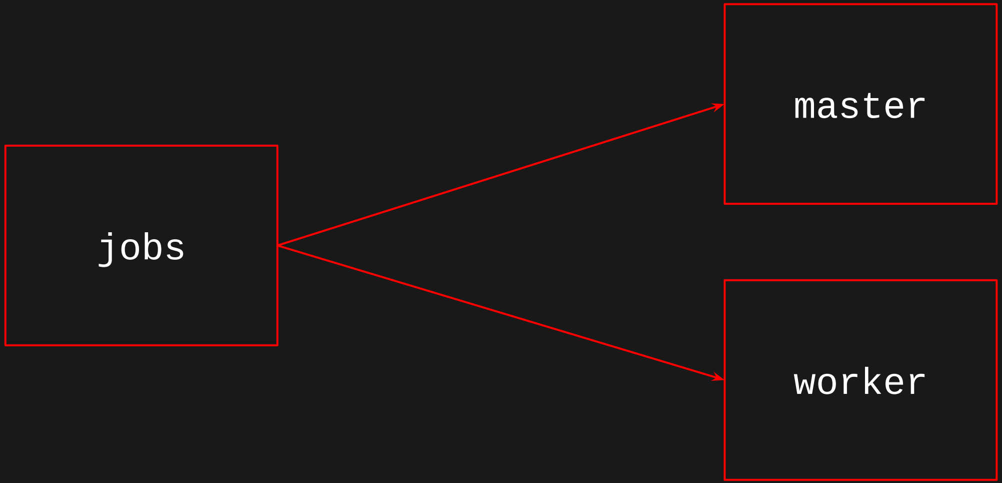
sampling
+
soriting
+
partitioning
+
merging

Design - High Level





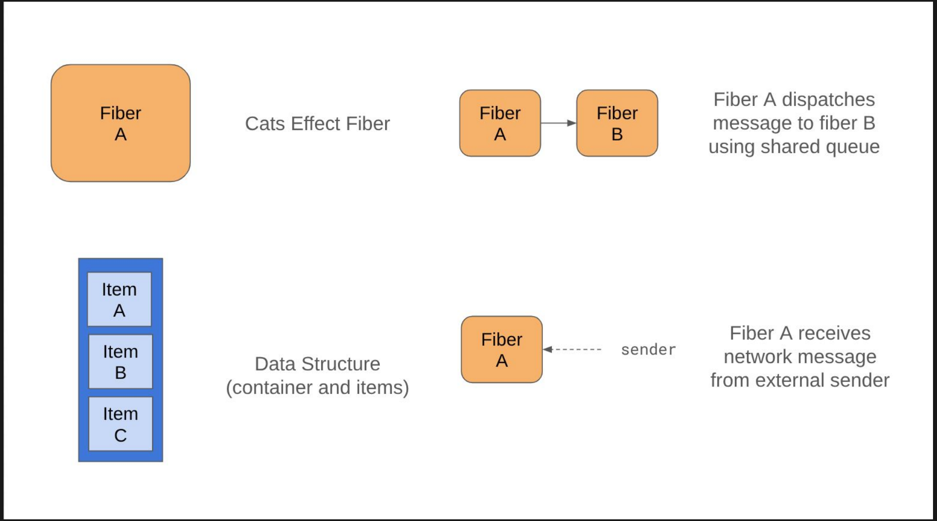
Design - Code Level

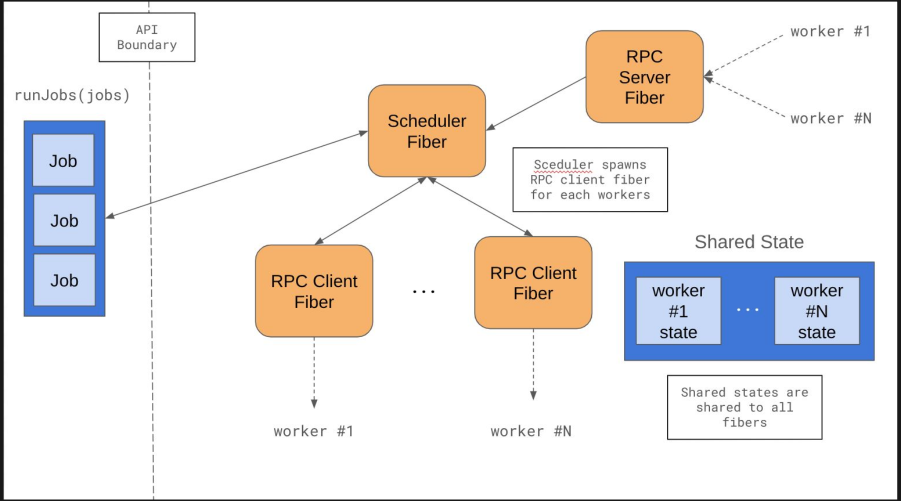


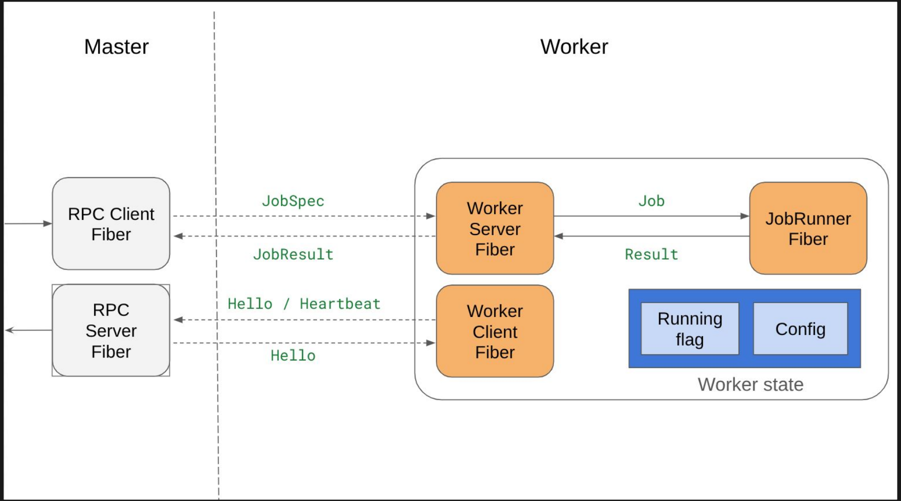
distributed job system
infrastructure

implements distributed
sorting on top of **jobs**

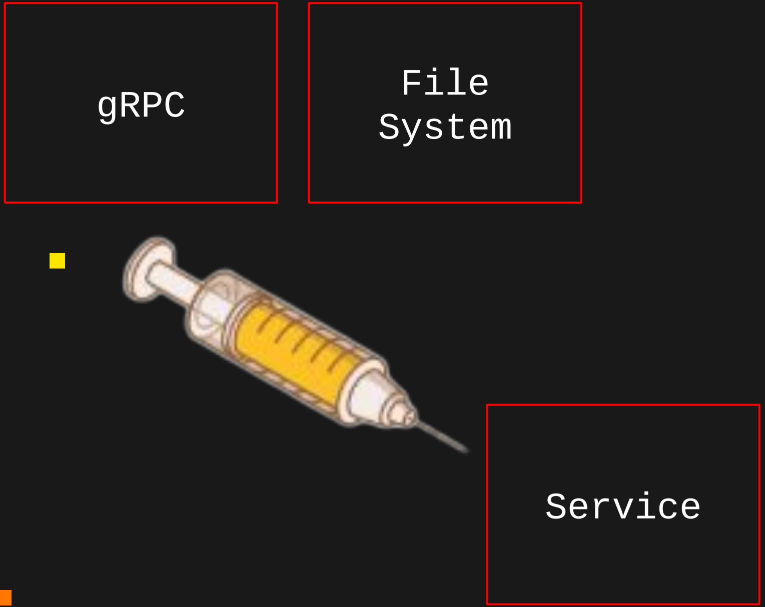
Design - Code Level


Fiber A
Cats Effect Fiber
Fiber A
Fiber B
Fiber A dispatches message to fiber B using shared queue
Item A
Item B
Item C
Data Structure (container and items)
Fiber A
sender
Fiber A receives network message from external sender

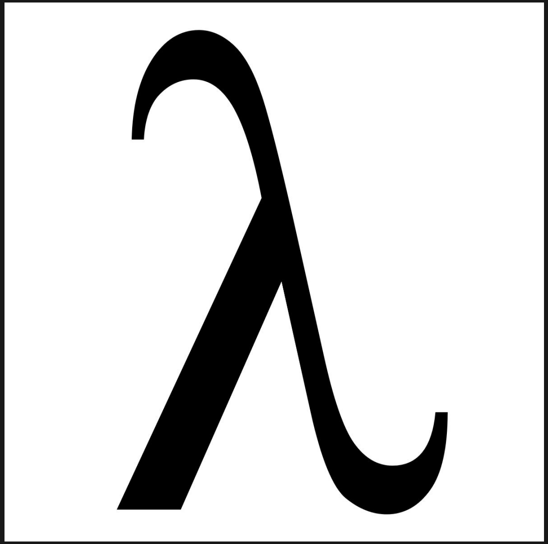




Design - Code Level



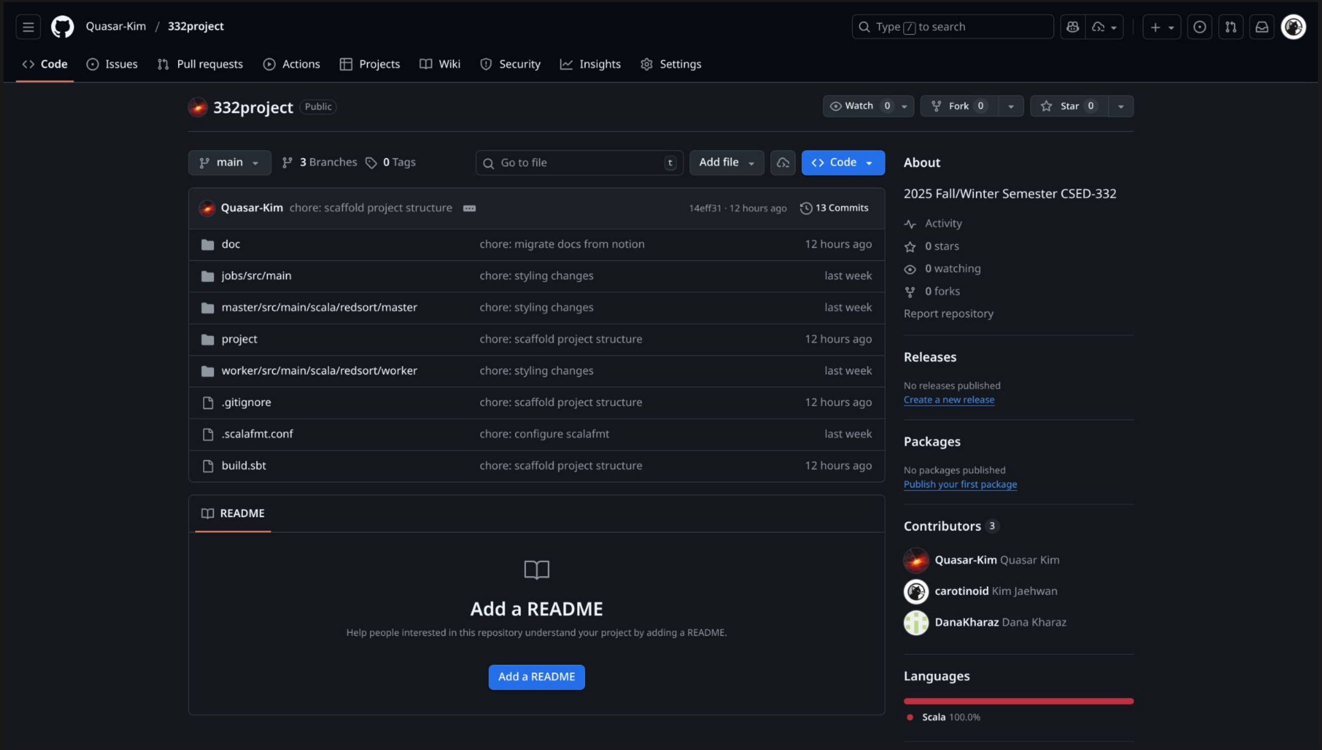
dependency injection



functional

Repository

https://github.com/Quasar-Kim/332project



Quasar-Kim / 332project

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

332project Public

Watch 0 Fork 0 Star 0

main 3 Branches 0 Tags Go to file Add file Code

Quasar-Kim chore: scaffold project structure 14eff31 · 12 hours ago 13 Commits

doc	chore: migrate docs from notion	12 hours ago
jobs/src/main	chore: styling changes	last week
master/src/main/scala/redsort/master	chore: styling changes	last week
project	chore: scaffold project structure	12 hours ago
worker/src/main/scala/redsort/worker	chore: styling changes	last week
.gitignore	chore: scaffold project structure	12 hours ago
.scalafmt.conf	chore: configure scalafmt	last week
build.sbt	chore: scaffold project structure	12 hours ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

About

2025 Fall/Winter Semester CSED-332

Activity
0 stars
0 watching
0 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

Quasar-Kim Quasar Kim
carotinoid Kim Jaehwan
DanaKharaz Dana Kharaz

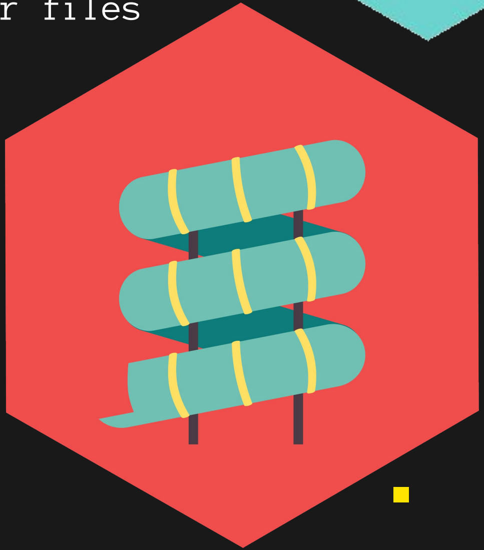
Languages

● Scala 100.0%

Libraries & Minor Details

- Runtime
  - cats-effect for asynchronous runtime
- Networking
  - fs2 for `Stream` abstraction
  - scalapb & fs2-grpc for protobuf and grpc on top of cats-effect
  - (maybe) fs2-ftp for transmission of larger files
  - grpc-netty for grpc transport
  - http4s for web ui
- Logging
  - log4cats for logging
- Testing
  - scalatest as testing framework
  - scalamock for stubbing and mocking
- Programming environment: Scala 2.13.16
- gensort for input data:

```
$ ./gensort -a 500000 ./input.txt # 50MB
```



Milestones

- Project design finalization (~ ~~week 4.5: 11/13 (Thu)~~ → week 5: 11/16 (Sun))
- Being able to sort small data on a local machine (~ week 7: 11/30)
    - sorting in an ideal condition. (~ week 6: 11/23)
    - presence of faults (short or long period / during each stage) (~ week 6.5: 11/27)
    - (optional) worst cases (~ week 7: 11/30)
        - lack of disk spaces
        - skewed input key distribution
        - unbalanced disk usage
        - etc.
- Running the project on actual (~ week 8: 12/7)

expectations vs. reality