Progress Presentation #1

Distributed Sorting System

Team Green

Retrospective

- Iteration 0
- Iteration 1
- Iteration 2
- Milestones

Logistics

- Iteration Meeting
- Collaboration Tools

DevOps

- Coding Convention
- Git Convention
- Development Environment
- CI

Design

- Overview
- Data
- RPC
- Master, Worker

Retrospective

Review of Weekly Progress

POSTECH

Iteration 0

Retrospective

- Created Git Repository on GitHub
  - https://github.com/betarixm/434project
- Created team ground rules
  - Meeting Day, Git convention, Coding convention, etc.
- Setup collaboration tools
  - Discord, GitHub Project.
- Setup development environments
  - Init SBT project

Iteration 1

Retrospective

- Document architecture proposal (#2)
  - Includes our understanding of this project.
- Setup CI and pre-commit hooks (#3, #4)
  - Applies scalafix, scalafmt, markdownlint.
- Document coding convention (#5)
  - Defines coding convention and git convention.

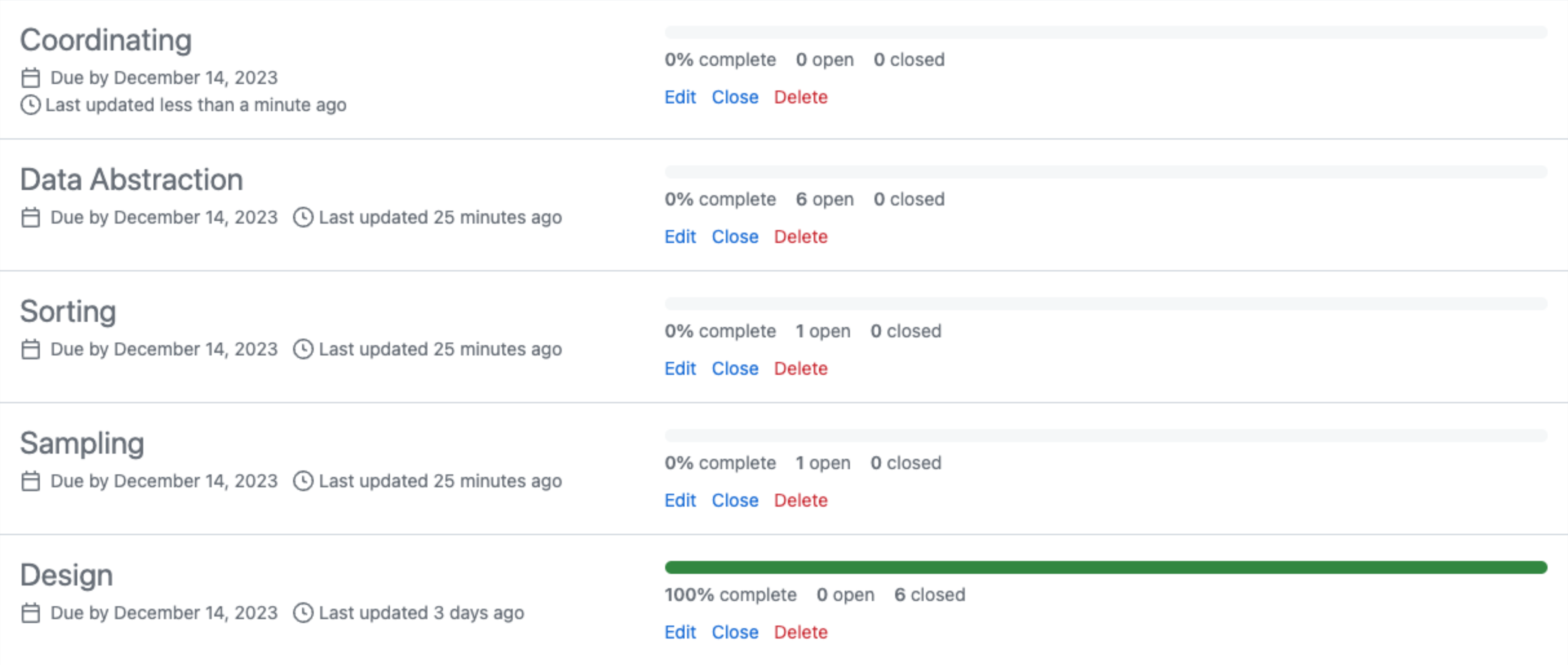
Iteration 2

Retrospective

- Document data structure proposal (#6)
  - Defines terms and data abstractions.
- Document RPC protocol proposal (#7)
  - Defines services and master.
- Document worker proposal (#8)
  - Defines worker related classes and its methods.
- Document master proposal (#9)
  - Defines master related classes and it methods.

Milestones

Retrospective



Coordinating

Due by December 14, 2023

Last updated less than a minute ago

0% complete 0 open 0 closed

Data Abstraction

Due by December 14, 2023 Last updated 25 minutes ago

0% complete 6 open 0 closed

Sorting

Due by December 14, 2023 Last updated 25 minutes ago

0% complete 1 open 0 closed

Sampling

Due by December 14, 2023 Last updated 25 minutes ago

0% complete 1 open 0 closed

Design

Due by December 14, 2023 Last updated 3 days ago

100% complete 0 open 6 closed

https://github.com/betarixm/434project/milestones

Logistics

How We Work?

POSTECH

Iteration Meeting

Logistics

- Every Thursday!
- Add tickets as backlogs.
- Select tickets to be implemented in next iteration.
- Assign tickets to teammates.

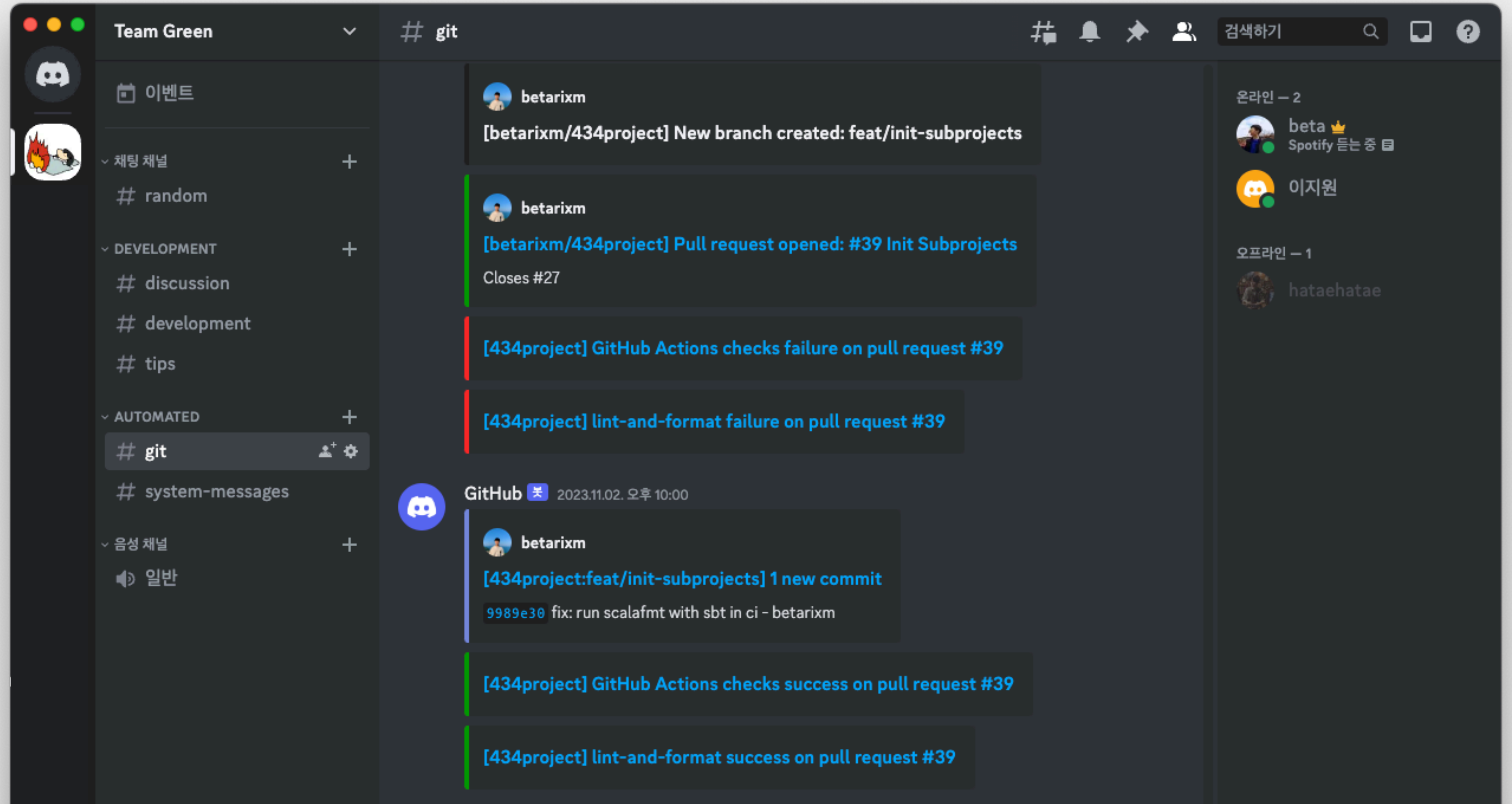
Tools

Logistics

- Discord
  - For communication e.g. chatting and voice calls.
- GitHub Project
  - For managing tickets and tracking issues
  - https://github.com/users/betarixm/projects/2

Tools

Logistics



betarixm

[betarixm/434project] New branch created: feat/init-subprojects

betarixm

[betarixm/434project] Pull request opened: #39 Init Subprojects

Closes #27

[434project] GitHub Actions checks failure on pull request #39

[434project] lint-and-format failure on pull request #39

GitHub 봇 2023.11.02. 오후 10:00

betarixm

[434project:feat/init-subprojects] 1 new commit

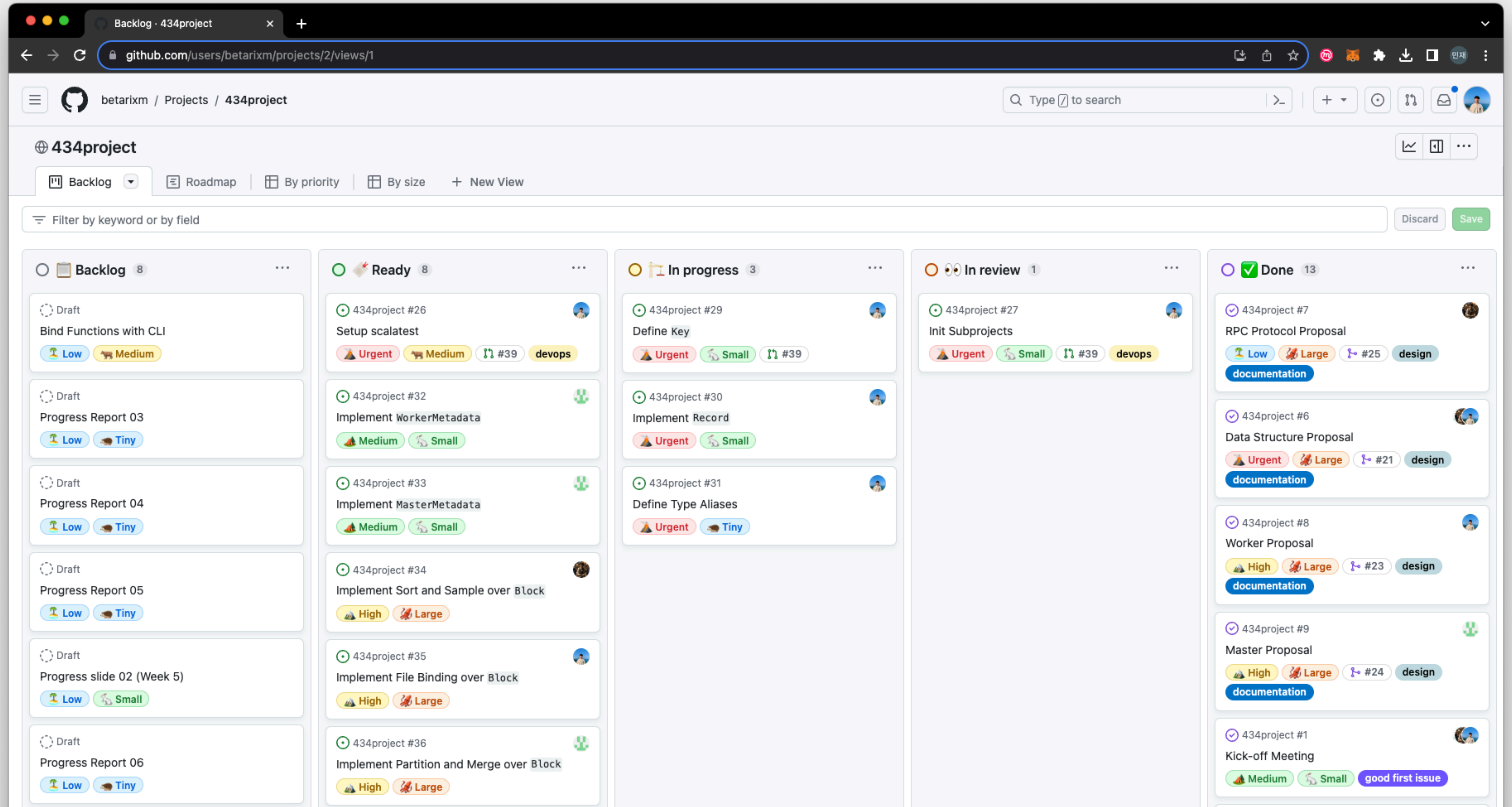
`9989e30` fix: run scalafmt with sbt in ci - betarixm

[434project] GitHub Actions checks success on pull request #39

[434project] lint-and-format success on pull request #39

Tools

Logistics



434project

Filter by keyword or by field

Backlog 8

- Draft
  Bind Functions with CLI
  Low · Medium
- Draft
  Progress Report 03
  Low · Tiny
- Draft
  Progress Report 04
  Low · Tiny
- Draft
  Progress Report 05
  Low · Tiny
- Draft
  Progress slide 02 (Week 5)
  Low · Small
- Draft
  Progress Report 06
  Low · Tiny

Ready 8

- 434project #26
  Setup scalatest
  Urgent · Medium · #39 · devops
- 434project #32
  Implement `WorkerMetadata`
  Medium · Small
- 434project #33
  Implement `MasterMetadata`
  Medium · Small
- 434project #34
  Implement Sort and Sample over `Block`
  High · Large
- 434project #35
  Implement File Binding over `Block`
  High · Large
- 434project #36
  Implement Partition and Merge over `Block`
  High · Large

In progress 3

- 434project #29
  Define `Key`
  Urgent · Small · #39
- 434project #30
  Implement `Record`
  Urgent · Small
- 434project #31
  Define Type Aliases
  Urgent · Tiny

In review 1

- 434project #27
  Init Subprojects
  Urgent · Small · #39 · devops

Done 13

- 434project #7
  RPC Protocol Proposal
  Low · Large · #25 · design · documentation
- 434project #6
  Data Structure Proposal
  Urgent · Large · #21 · design · documentation
- 434project #8
  Worker Proposal
  High · Large · #23 · design · documentation
- 434project #9
  Master Proposal
  High · Large · #24 · design · documentation
- 434project #1
  Kick-off Meeting
  Medium · Small · good first issue

DevOps

Development and Operations

POSTECH

Development Environments

Development and Operations

- Scala 2.13.12 with Java 20
- Scalatest 3.2.17, Scalafix 0.11.1, Scalafmt 2.4.6
- ScalaPB 0.11.14

Coding Convention

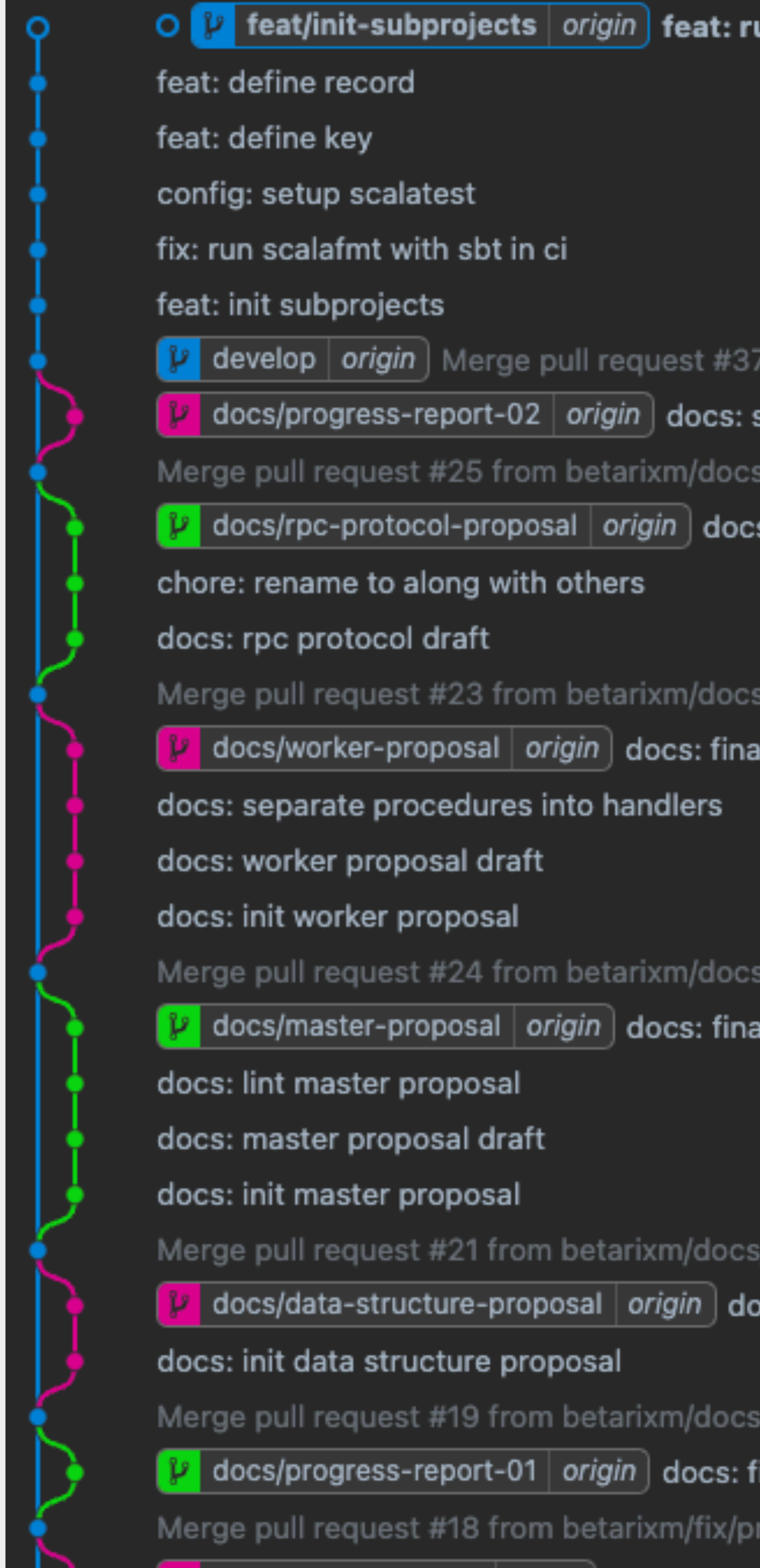
Development and Operations

- Adhere to typical Java style conventions.
- Use scalafmt for formatting.
- Use scalafix with default configurations for linting.

Git Convention

Development and Operations

- Follow the rules of conventional commits.
  - https://www.conventionalcommits.org/en/v1.0.0/
- Employ main-develop branch strategy.
  - Ensure that the main and develop branches can only be pushed with PRs that have received approvals.
  - Merge the develop branch into the main branch when each iteration ends.
- Try to make commits in a semi-linear manner.

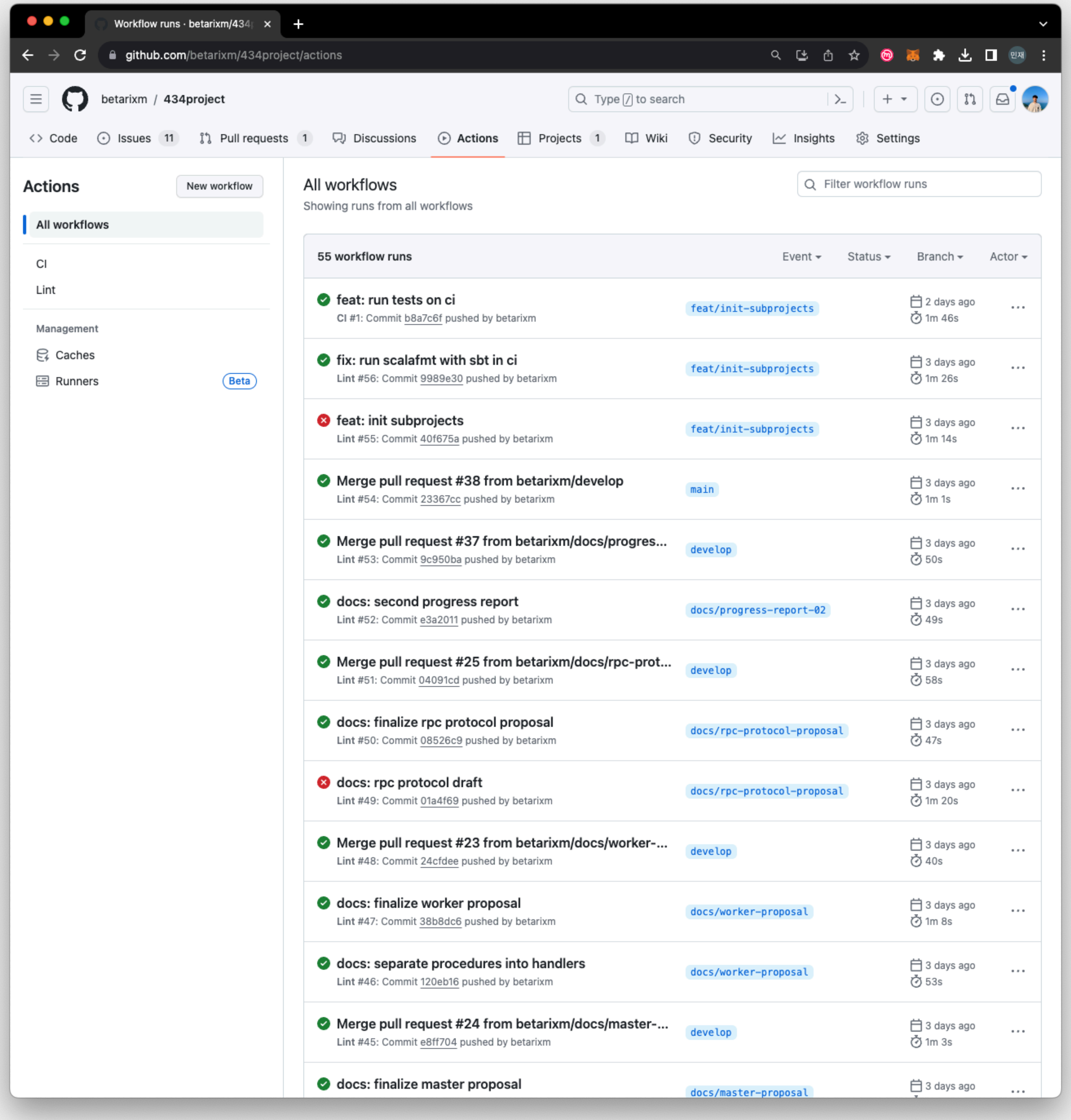


```
feat/init-subprojects  origin  feat: ru
feat: define record
feat: define key
config: setup scalatest
fix: run scalafmt with sbt in ci
feat: init subprojects
develop  origin  Merge pull request #37
docs/progress-report-02  origin  docs: s
Merge pull request #25 from betarixm/docs
docs/rpc-protocol-proposal  origin  doc
chore: rename to along with others
docs: rpc protocol draft
Merge pull request #23 from betarixm/docs
docs/worker-proposal  origin  docs: fina
docs: separate procedures into handlers
docs: worker proposal draft
docs: init worker proposal
Merge pull request #24 from betarixm/docs
docs/master-proposal  origin  docs: fina
docs: lint master proposal
docs: master proposal draft
docs: init master proposal
Merge pull request #21 from betarixm/docs
docs/data-structure-proposal  origin  do
docs: init data structure proposal
Merge pull request #19 from betarixm/docs
docs/progress-report-01  origin  docs: f
Merge pull request #18 from betarixm/fix/p
```

CI

Development and Operations

- Powered by GitHub Actions.
- Check formatting and linting.
- Test via scalatest.



All workflows

Showing runs from all workflows

55 workflow runs

✓	**feat: run tests on ci** CI #1: Commit b8a7c6f pushed by betarixm	`feat/init-subprojects`	2 days ago 1m 46s
✓	**fix: run scalafmt with sbt in ci** Lint #56: Commit 9989e30 pushed by betarixm	`feat/init-subprojects`	3 days ago 1m 26s
✗	**feat: init subprojects** Lint #55: Commit 40f675a pushed by betarixm	`feat/init-subprojects`	3 days ago 1m 14s
✓	**Merge pull request #38 from betarixm/develop** Lint #54: Commit 23367cc pushed by betarixm	`main`	3 days ago 1m 1s
✓	**Merge pull request #37 from betarixm/docs/progres...** Lint #53: Commit 9c950ba pushed by betarixm	`develop`	3 days ago 50s
✓	**docs: second progress report** Lint #52: Commit e3a2011 pushed by betarixm	`docs/progress-report-02`	3 days ago 49s
✓	**Merge pull request #25 from betarixm/docs/rpc-prot...** Lint #51: Commit 04091cd pushed by betarixm	`develop`	3 days ago 58s
✓	**docs: finalize rpc protocol proposal** Lint #50: Commit 08526c9 pushed by betarixm	`docs/rpc-protocol-proposal`	3 days ago 47s
✗	**docs: rpc protocol draft** Lint #49: Commit 01a4f69 pushed by betarixm	`docs/rpc-protocol-proposal`	3 days ago 1m 20s
✓	**Merge pull request #23 from betarixm/docs/worker-...** Lint #48: Commit 24cfdee pushed by betarixm	`develop`	3 days ago 40s
✓	**docs: finalize worker proposal** Lint #47: Commit 38b8dc6 pushed by betarixm	`docs/worker-proposal`	3 days ago 1m 8s
✓	**docs: separate procedures into handlers** Lint #46: Commit 120eb16 pushed by betarixm	`docs/worker-proposal`	3 days ago 53s
✓	**Merge pull request #24 from betarixm/docs/master-...** Lint #45: Commit e8ff704 pushed by betarixm	`develop`	3 days ago 1m 3s
✓	**docs: finalize master proposal**	`docs/master-proposal`	3 days ago

Pre-commit Hooks

Development and Operations

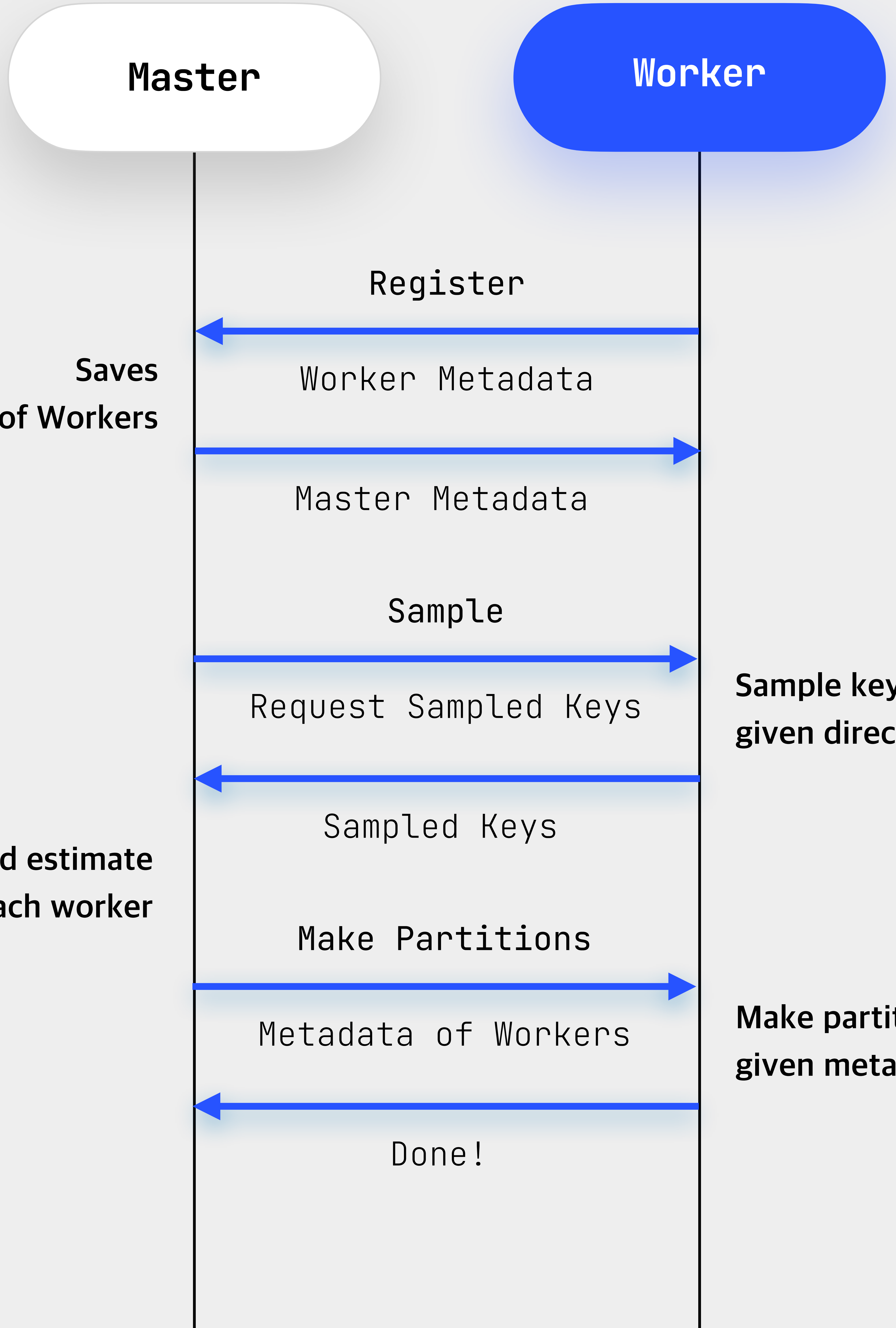
- Powered by Git pre-commit hooks and wrapper package.
- Exposes similar formatting and linting environments as CI.

Design

Data, Protocol, Worker and Master

Overview

Design



Master

Worker

Register

Worker Metadata

Saves
Metadata of Workers

Master Metadata

Sample

Request Sampled Keys

Sample keys from
given directories

Sampled Keys

Gather the keys and estimate
key ranges for each worker

Make Partitions

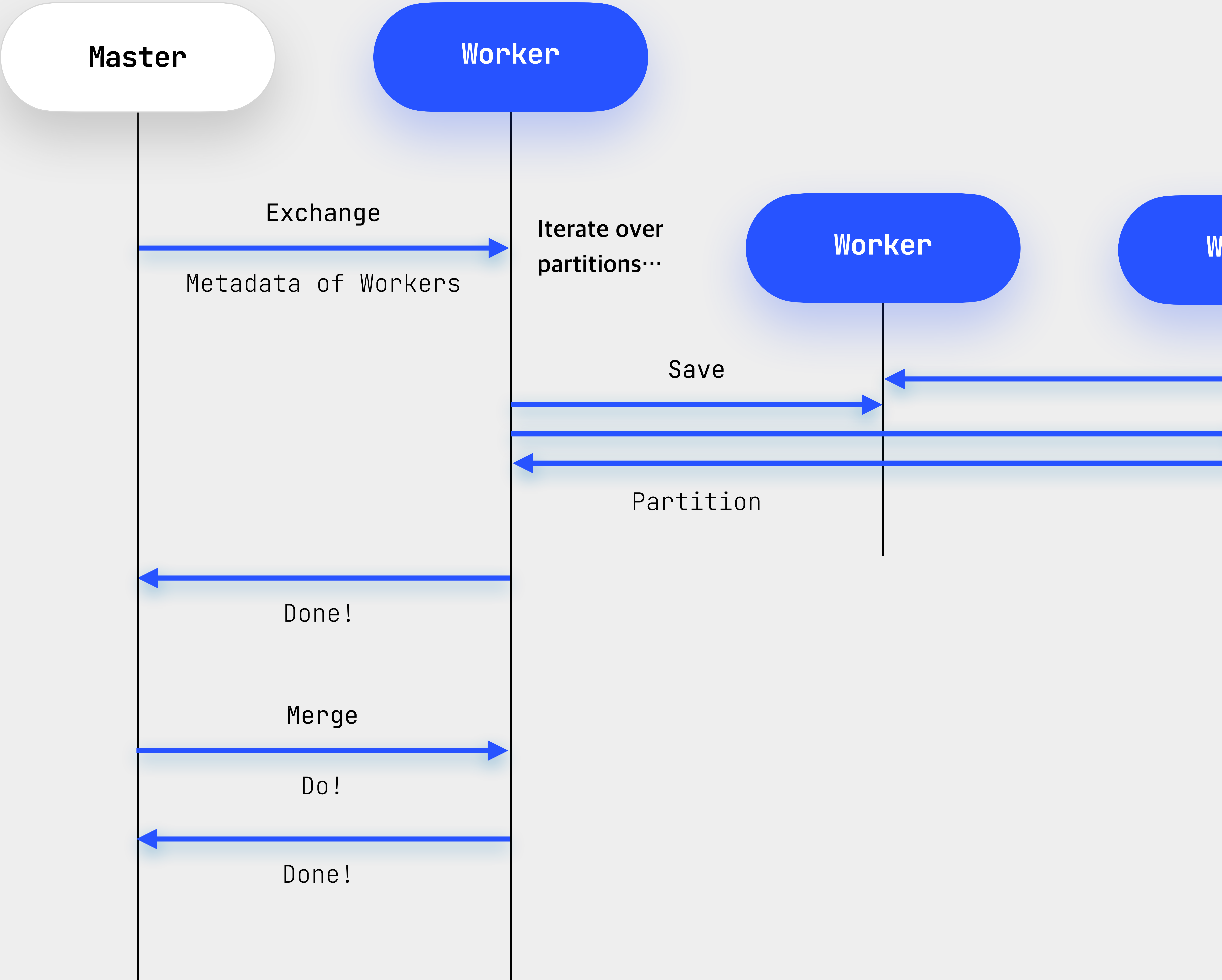
Metadata of Workers

Make partitions based on
given metadata of workers

Done!

Overview

Design



Data Abstraction

Design

- Key
  - Wrap 10 bytes of array.
  - Ordered value class, e.g. comparable between keys.
- Record
  - Wrap Key with 90 bytes of array.
  - Ordered value class, e.g. comparable between records.
  - Support (de)serialization to read and write.

Data Abstraction

Design

- Block
  - Wrap records into stream.
  - Can sample keys from records.
  - Can sort its records.
  - Can make partitions in terms of given metadata of workers.
- Worker Metadata, Master Metadata
  - Saves host, port, key range and etc.

Data Abstraction

Design

- Key Range
  - Tuple of key and key.
- Partition
  - Tuple of worker metadata and block.

RPC: Worker Exchange Service

Design

- Method: Save Block
  - Save given block into filesystem when triggered.

RPC: Worker Service

Design

- Method: Sample
  - Returns sampled keys from blocks.
- Method: Sort
  - Sorts records and saves them as files.
- Method: Partition
  - Makes and saves partitions with given metadata of workers.

RPC: Worker Service

Design

- Method: Exchange
  - Exchanges partitions between workers using exchange service.
- Method: Finalize
  - Merges and saves blocks in sorted form.

RPC: Master Service

Design

- Method: Register
  - Saves metadata of workers.

Worker

Design

- Spawn RPC services
  - e.g. Worker Service and Worker Exchange Service.
- Initiate registration with the master.
- Handle master’s requests.

Master

Design

- Start the master service.
- Wait for all workers to send their metadata.
- Collect sampled keys from workers and estimate key ranges.
- Instruct workers to perform sorting and partitioning.
- Enable block exchange between workers.
- Request workers to perform merging.

Progress Presentation #01

Distributed Sorting System

Gwon Minjae, Dept. of Computer Science & Engineering, POSTECH.

Lee Jiwon, Dept. of Computer Science & Engineering, POSTECH.

Ha Taehyeok, Dept. of Computer Science & Engineering, POSTECH.

Advanced Programming, 2023.