2025-1118

Distributed External Merge Sort System using gRPC

cse4332

Team Orange

20210212 이지민
20220500 서영서
20220963 오상원

System Setup & Team Workflow

Communication

- KakaoTalk
- Discord
- Weekly offline meetings

Git workflow

feature/* → Pull Request → develop
↳ main (release)

Logging

- println-based debugging
- Key events: register / sampling / splitters / shuffle
- Future: Logback migration (Week7)

Environment

OS	Ubuntu 24.04 (WSL2)
JAVA	OpenJDK 17
Scala	2.13.x
sbt	1.9+
RPC	gRPC + ScalaPB

Libraries

- gRPC
- ScalaPB
- Java NIO
- ScalaTest

Project Overview & Milestones

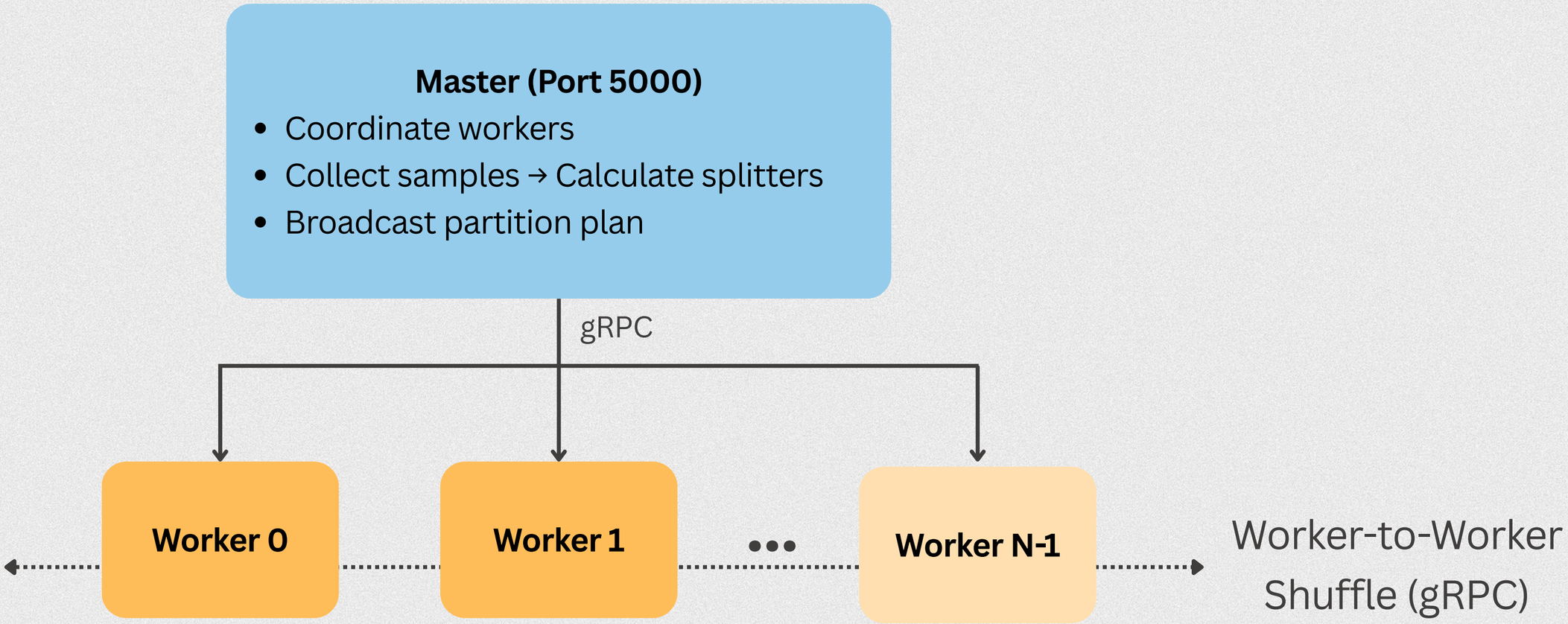
Project Overview

A distributed large-scale sorting system where the Master coordinates sampling and planning, and Workers perform sorting, partitioning, and shuffle.

Milestones

Week 1 | RPC & Protobuf IDL created
Week 2 | Worker registration + heartbeat + WorkerRegistry
Week 3 | Sampling → sample aggregation → splitter calculation
Week 4 | PartitionPlan generation + Master→Worker broadcast
Week 5 | Worker local sort → partition → shuffle (sender & receiver)

System Architecture



Code Structure

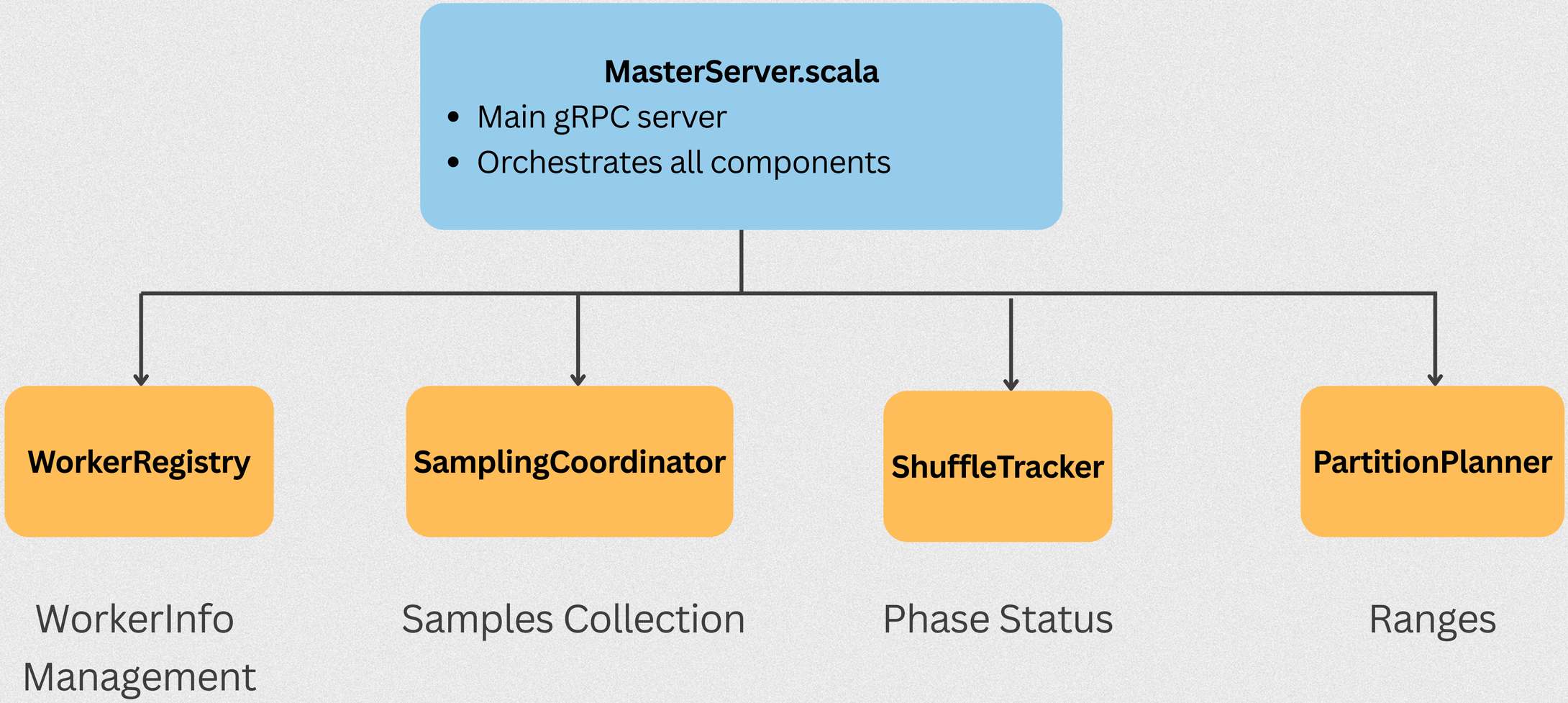
```
src/main/scala/
├── master/
│   ├── MasterServer.scala          ← gRPC server + orchestration
│   ├── WorkerRegistry.scala        ← Worker lifecycle management
│   ├── SamplingCoordinator.scala   ← Sample collection + splitter calc
│   └── PartitionPlanner.scala      ← Splitter → PartitionPlan
│
├── worker/
│   ├── WorkerClient.scala          ← Main entry + coordination
│   ├── WorkerServer.scala          ← gRPC server (shuffle receiver)
│   ├── MasterClient.scala          ← Master communication
│   └── WorkerState.scala           ← Global worker state
│
├── common/
    ├── RecordIO.scala              ← 100-byte record I/O
    └── Sampling.scala              ← Uniform every-N sampling

src/main/protobuf/
└── sort.proto                      ← RPC definitions
```

RPC & Protobuf Design

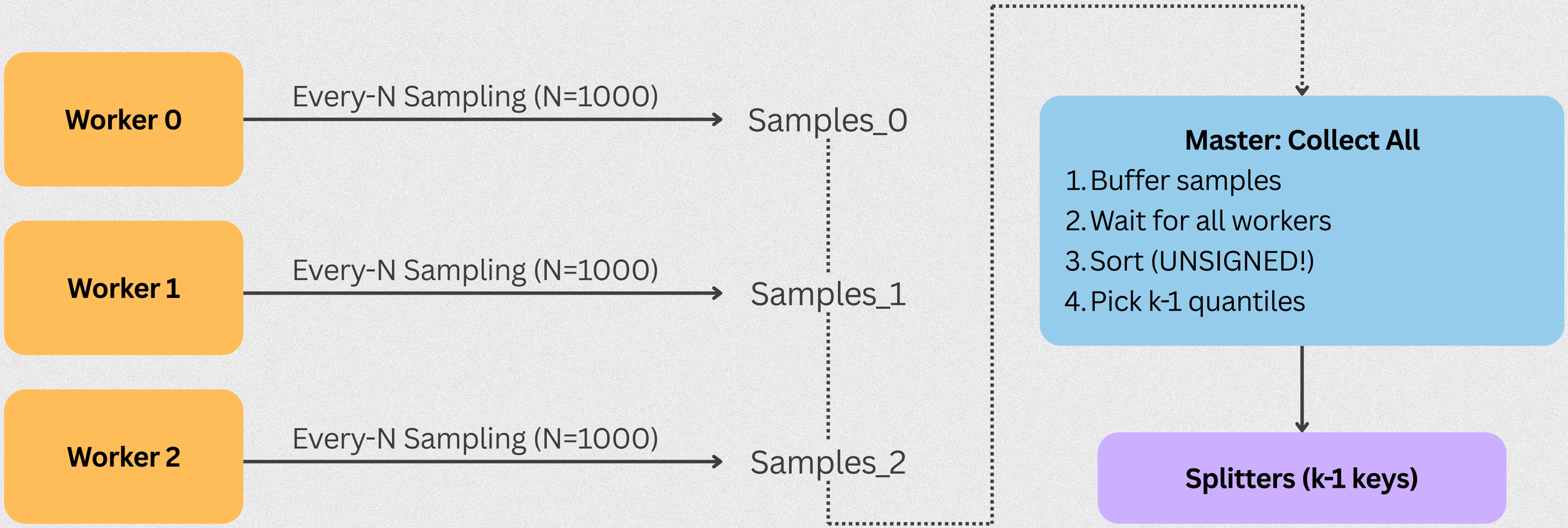
```
service MasterService {
  rpc RegisterWorker(WorkerInfo)
      returns (WorkerAssignment);

  rpc Heartbeat(WorkerInfo)
      returns (Ack);

  rpc SendSamples(stream Sample)
      returns (Splitters);          ← Client streaming!

  rpc ReportShuffleComplete(WorkerStatus)
      returns (Ack);
}

service WorkerService {
  rpc SetPartitionPlan(PartitionPlan)
      returns (Ack);

  rpc PushPartition(stream PartitionChunk)
      returns (Ack);                ← Server streaming!

  rpc StartShuffle(TaskId) returns (Ack);
  rpc FinalizePartitions(TaskId) returns (Ack);
}
```

```
message Sample { bytes key = 1; }
message Splitters { repeated bytes key = 1; }
message PartitionRange {
  bytes lo = 1;          // Inclusive
  bytes hi = 2;          // Exclusive
  int32 target_worker = 3;
}
message PartitionChunk {
  TaskId task = 1;
  string partition_id = 2;
  bytes payload = 3;
  int64 seq = 4;
}
```

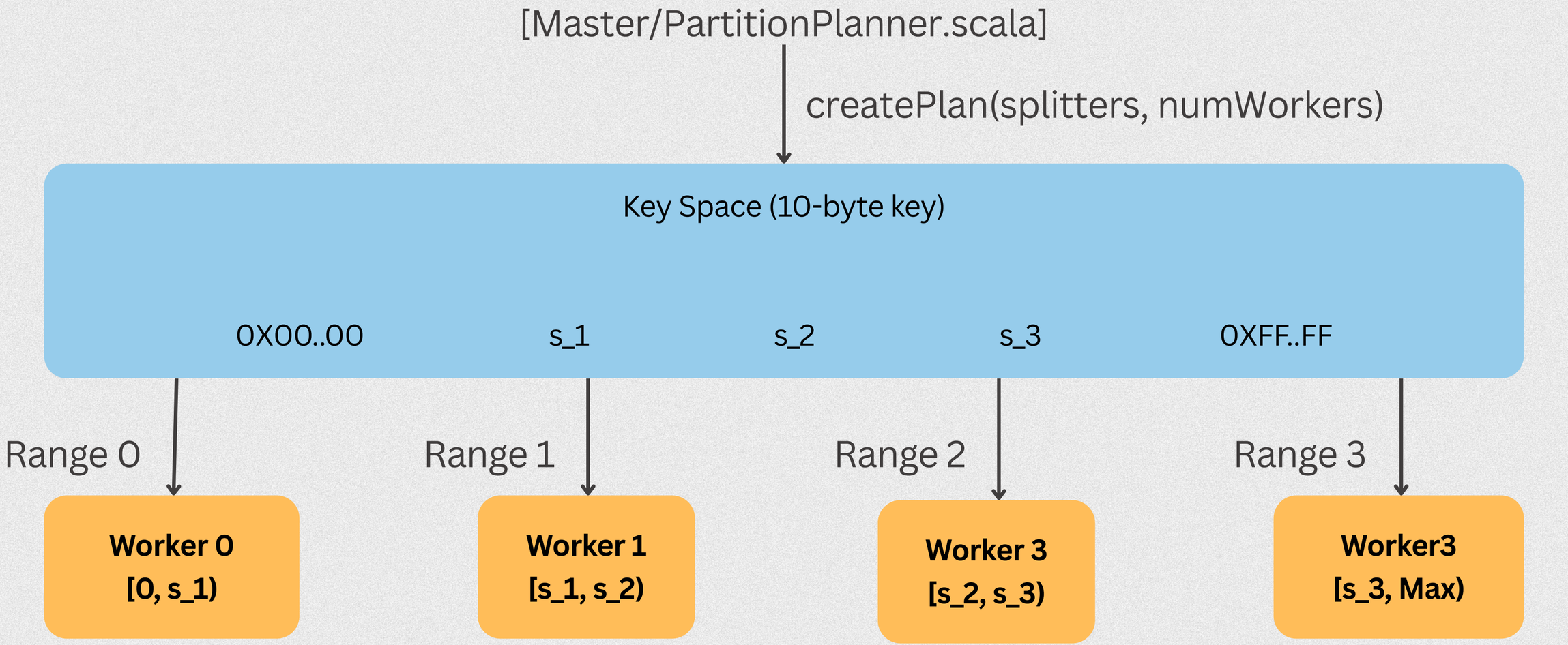
Master-Side Components



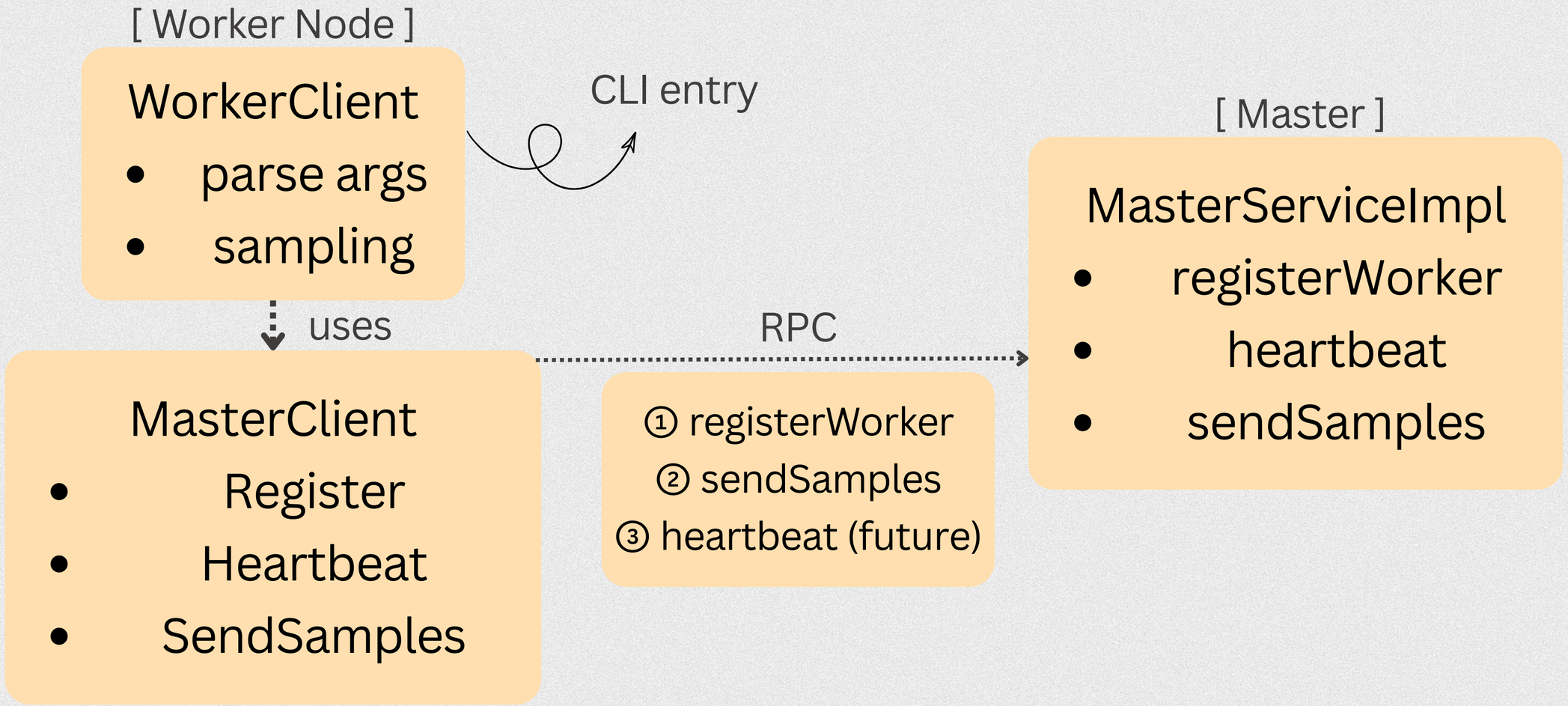
Sampling & Splitter Calculation



Partition Plan Design



Worker-side: WorkerClient & MasterClient



Worker-side: WorkerServer & Shuffle Receiver

[Master]

PartitionPlan

setPartitionPlan

[Other Workers]

PushPartition

WorkerServer (WorkerServiceImpl)

[1] PlanStore : latest PartitionPlan

[2] PartitionStore : partitionId → runs[Array[Array[Byte]]]

[3] K-way merge & writePartitionToFile()

outputDir / partition.N

Data Flow

step 1

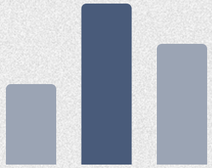
Input Read (Worker)

input .dat 읽기(RecordIO.streamRecords)

step 2

Sampling (Worker → Master)

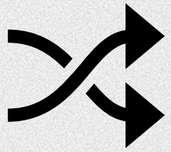
every 1000th key 샘플(Sampling.uniformEveryN)



step 3

Splitters & PartitionPlan (Master)

샘플 정렬 → splitters & PartitionPlan 생성



step 4

Shuffle (Worker <-> Worker)

Partition별 record 전송(PushPartition)

step 5

Merge & Output (Worker)

K-way merge → outputDir/partition.N

Remaining Tasks

(Completed) W1 W2 W3 W4 W5

RPC → Reg/HB → Sampling → Splitters/Plan → Shuffle (send/recv)

(Remaining) W6 W7

Merge I/O (finalize + E2E)

Reliability + Logs (retry, FT, logback)

Integration + Perf (full pipeline)

Core data path completed;

remaining tasks focus on merge finalization, reliability, and full integration