Distributed Sorting

DEVAUCHELLE Alex
49004567

RENAUD Thomas
49004578

Details - Communication Libraries

- Communication Method: Basic Socket and Object Stream

- Reasoning: We opted for a simple and effective communication method using basic socket and object stream for our Distributed Sorting project in Scala.

- Advantages:
    - Simplicity: Straightforward implementation and easy to understand.
    - Flexibility: Allows for custom serialization and deserialization of objects.
    - Language Agnostic: Can communicate between different programming languages if needed.

Communication Methods Comparison: gRPC vs. Socket and Stream

gRPC

Advantages:

- **Protocol Buffers**: Efficient serialization using Protocol Buffers for data interchange.
- **Bidirectional Streaming**: Supports streaming requests and responses simultaneously.
- **Interceptors**: Supports middleware for handling cross-cutting concerns.

Disadvantages:

- **Complexity**: May introduce complexity in understanding due to its feature-rich nature.
- **Learning Curve**: Developers might need time to grasp the concepts and usage.
- **Heavy Dependencies**: Requires additional libraries and tools.

Basic Socket with Object Stream

Advantages:

- **Simplicity**: Straightforward implementation, especially for smaller projects.
- **Custom Serialization**: Allows for custom serialization/deserialization methods.
- **Platform Independence**: Works well across different platforms.

Disadvantages:

- **Limited Features**: Lacks some advanced features like built-in authentication, load balancing, and middleware.
- **Potential for Bottlenecks**: Depending on implementation, may face performance challenges with large-scale data.

Details - Logging in the Project

- Logging Library Used: Log4j
- Implementation Steps:
    - Add library dependency in build.sbt
    - Create a log4j2.xml configuration file. Define loggers, appenders, and layout patterns
    - Implementation in Code : Import Logging, Extend my class or object with Logging and use logger.{info;debug,error}
- Exemple :

```
[info] running com.cs434.sortnet.master.Master 2
2023-11-22 20:48:34.381  INFO --- [bt-bg-threads-1] c.c.s.m.Master$                    : 2.2.2.142:9999
2023-11-22 20:48:34.389  INFO --- [bt-bg-threads-1] c.c.s.m.Master$                    : Stage start : Register
2023-11-22 20:48:34.392  INFO --- [bt-bg-threads-1] c.c.s.m.Master$                    : Master listening on 9999
```

```
[info] running com.cs434.sortnet.master.Master
2023-11-22 20:52:08.245 ERROR --- [bt-bg-threads-1] c.c.s.m.Master$                    : Usage : master <# of workers>
```

Overall Design

Data structure

Network data structure

Milestones

Design **Deadline : 10/16 - 11/12**

- ✓ Generate input data
- ✓ Overall flow chart design
- ✓ Choosing communication library, programming environment, logging system
- ✓ Choosing data structure for sampling, sorting, partitioning, shuffling and merging
- ✓ Documentation report : Data structure design
- ✓ Documentation report : Network data structure
- ✓ Sequence diagram on communication protocol

Milestones

Implementation 1/2 **Deadline : 11/13 - 11/26**

- ✓ Connection Master-Worker
- ✓ Setup logger (log4j)
- ✓ Communication sampling task
- ✓ Sample input, send and received from worker to master
- ✓ Key range attribution and broadcasted, partition plan
- ✓ Communication sorting task
- ✓ Sorting file
- ✓ Partition each file given partition plan
- ✓ Testing partial solution

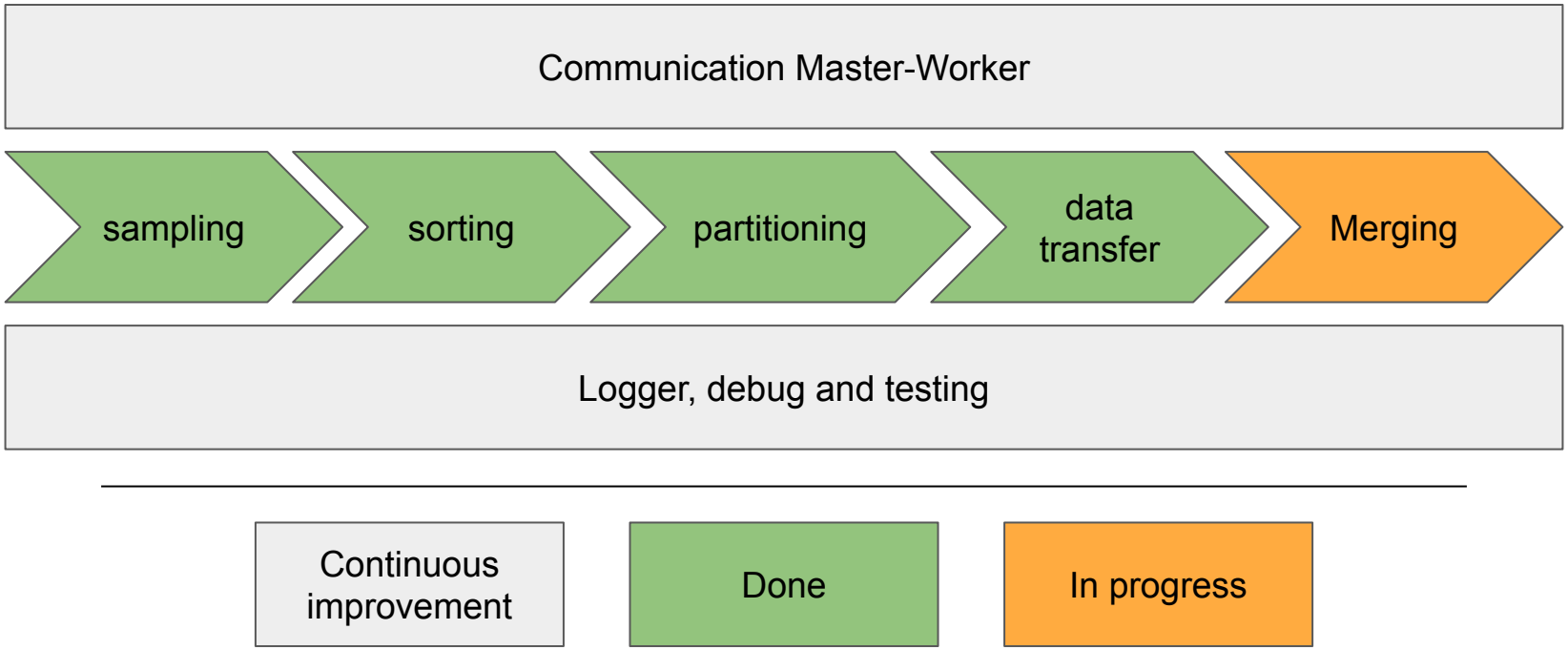
Milestones

Implementation 2/2

Deadline : 11/27 - 12/03

- ✓ Communication shuffling task
- ✓ File transfer between worker
- ❑ Communication merging task
- ❑ Merging method
- ❑ Validating output result (valsort)
- ❑ Testing overall solution

Summary


Communication Master-Worker
sampling
sorting
partitioning
data transfer
Merging
Logger, debug and testing
Continuous improvement
Done
In progress