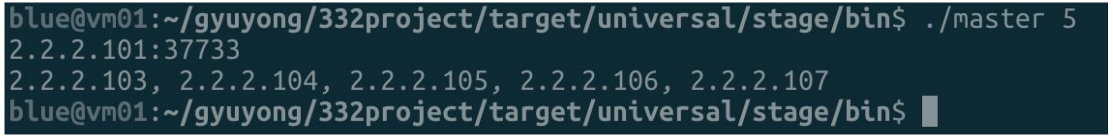
CS332
Final Presentation

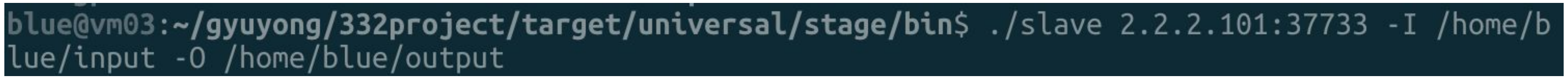
Team Blue

Experiment

- Environments
    - VM Cluster
    - 1 Master, 5 Slaves
    - 4 input blocks per slave
        - each input block: 32MB
        - entire records size: 640MB



```
blue@vm01:~/gyuyong/332project/target/universal/stage/bin$ ./master 5
2.2.2.101:37733
2.2.2.103, 2.2.2.104, 2.2.2.105, 2.2.2.106, 2.2.2.107
blue@vm01:~/gyuyong/332project/target/universal/stage/bin$ 
```



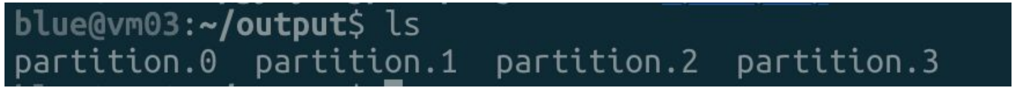
```
blue@vm03:~/gyuyong/332project/target/universal/stage/bin$ ./slave 2.2.2.101:37733 -I /home/b
lue/input -O /home/blue/output
```

Experiment (Results)

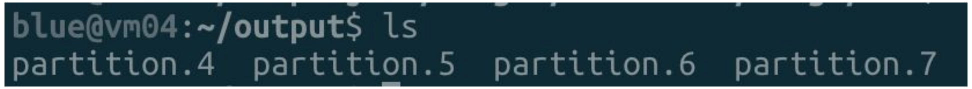
- Correctness
  - Does the master start? Yes
  - Does each worker connect to the master? Yes
  - Does the master collect sample data? Yes
  - Does the master return distribution keys back to workers? Yes
  - Do workers pass intermediate data between each other(during shuffling)? Yes
  - Does the master print a sequence of workers? Yes
  - Is the output sorted in each worker? No. Last partition in each slave has problem.
  - # of records in the input == # of records in the output? No. Last partition in each slave has problem.
- Execution time
  - around 45 seconds

Experiment (Failed Correctness)

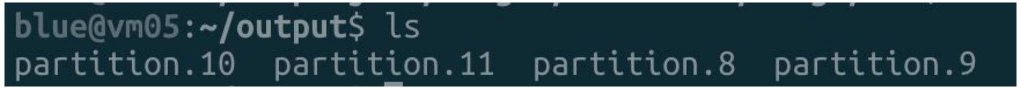
- All the other files except the last partition in each slave are correctly sorted.
- Last partition file has duplicated records from the previous partition file.
- It occurs when the merged file cannot be splitted into exactly same size.



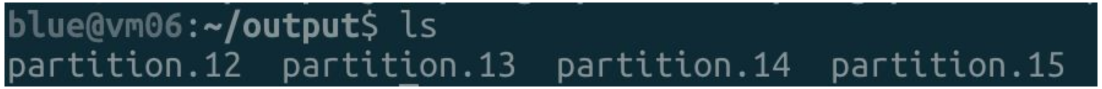
```
blue@vm03:~/output$ ls
partition.0  partition.1  partition.2  partition.3
```



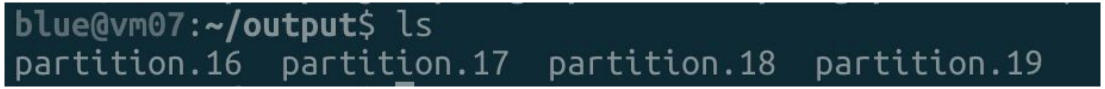
```
blue@vm04:~/output$ ls
partition.4  partition.5  partition.6  partition.7
```



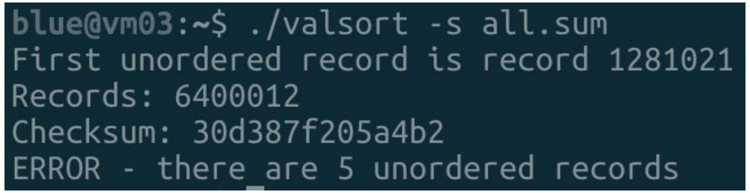
```
blue@vm05:~/output$ ls
partition.10  partition.11  partition.8  partition.9
```



```
blue@vm06:~/output$ ls
partition.12  partition.13  partition.14  partition.15
```



```
blue@vm07:~/output$ ls
partition.16  partition.17  partition.18  partition.19
```



```
blue@vm03:~$ ./valsort -s all.sum
First unordered record is record 1281021
Records: 6400012
Checksum: 30d387f205a4b2
ERROR - there are 5 unordered records
```

Project Management (Milestones)

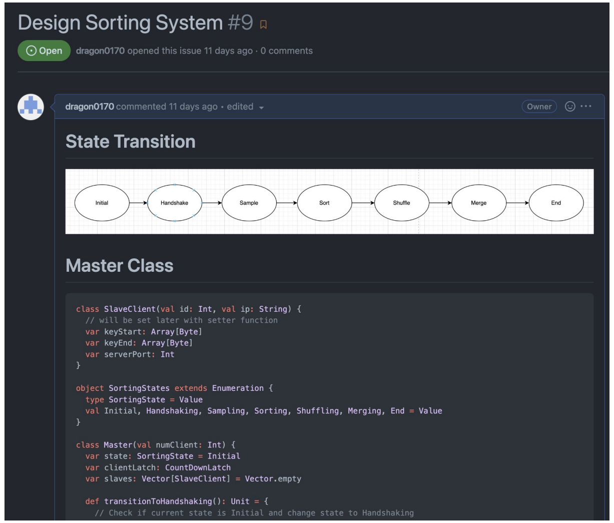
1. Be familiar with **gensort/valsort, grpc, protobuf**. Master can connects to Slave with grpc and **sends ip address and prints the ip address list**.
   - original: ~11/7, actual: 11/13
2. **Sample data** from the file in each worker. Master **determines and broadcasts sorting key ranges for each slaves**.
   - original: ~11/13, actual: 12/10
3. **Sort input files** in each slave and **save the sorted results into partitioned files** with appropriate key ranges.
   - original: ~11/20, actual: 12/10
4. **Shuffle the sorted files** with each other between slaves. **Merge all sorted files** in each slave and **save into partitioned files** with appropriate size.
   - original: ~12/4, actual: 12/11
   - Shuffling and merging is correct, but partitioning is incorrect.

Project Management (Member Responsibility)

- 최규용
  - Manage the team
  - Design the sorting system
  - Distribute tasks
  - Develop
    - Set up integration test with docker
    - Handshake between slave and master
    - Sort unsorted files and save it into partitions with key ranges
    - Review Mathis' code and test/fix bugs and merge into main branch
- Mathis
  - Develop
    - Sample data and create key range partitions
    - Shuffle files between slaves
    - Merge sorted files into one file and split it into several files
- 김수빈
  - Develop
    - Try to print server endpoint and slaves' ip address at Master application
    - Try to use externalsortinginjava library in scala
  - Missing

Design

- Design document: source of truth for coworking
    - https://github.com/dragon0170/332project/issues/9
    - Defines protobuf message
    - Describes Master class and Slave class structure
    - Defines required functions and abstractions for each stage



Design Sorting System #9

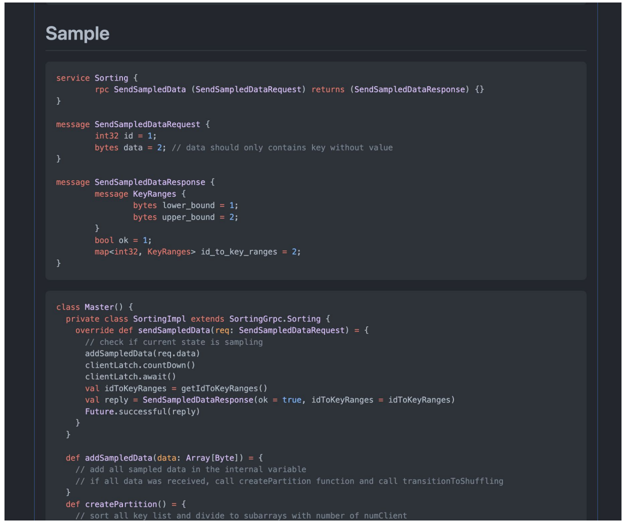
Open dragon0170 opened this issue 11 days ago · 0 comments

dragon0170 commented 11 days ago · edited Owner

State Transition

Master Class

```
class SlaveClient(val id: Int, val ip: String) {
  // will be set later with setter function
  var keyStart: Array[Byte]
  var keyEnd: Array[Byte]
  var serverPort: Int
}

object SortingStates extends Enumeration {
  type SortingState = Value
  val Initial, Handshaking, Sampling, Sorting, Shuffling, Merging, End = Value
}

class Master(val numClient: Int) {
  var state: SortingState = Initial
  var clientLatch: CountDownLatch
  var slaves: Vector[SlaveClient] = Vector.empty

  def transitionToHandshaking(): Unit = {
    // Check if current state is Initial and change state to Handshaking
```



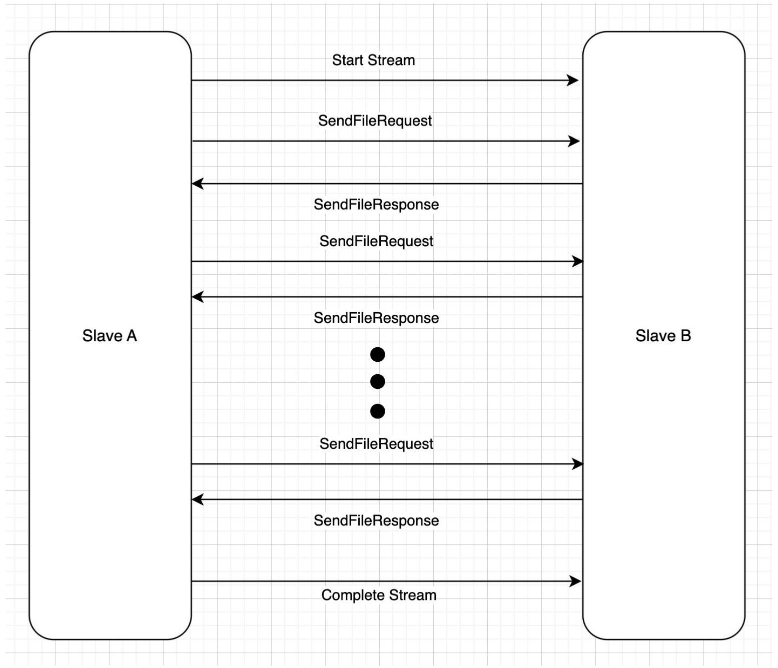
Sample

```
service Sorting {
        rpc SendSampledData (SendSampledDataRequest) returns (SendSampledDataResponse) {}
}

message SendSampledDataRequest {
        int32 id = 1;
        bytes data = 2; // data should only contains key without value
}

message SendSampledDataResponse {
        message KeyRanges {
                bytes lower_bound = 1;
                bytes upper_bound = 2;
        }
        bool ok = 1;
        map<int32, KeyRanges> id_to_key_ranges = 2;
}
```

```
class Master() {
  private class SortingImpl extends SortingGrpc.Sorting {
    override def sendSampledData(req: SendSampledDataRequest) = {
      // check if current state is sampling
      addSampledData(req.data)
      clientLatch.countDown()
      clientLatch.await()
      val idToKeyRanges = getIdToKeyRanges()
      val reply = SendSampledDataResponse(ok = true, idToKeyRanges = idToKeyRanges)
      Future.successful(reply)
    }
  }

  def addSampledData(data: Array[Byte]) = {
    // add all sampled data in the internal variable
    // if all data was received, call createPartition function and call transitionToShuffling
  }
  def createPartition() = {
    // sort all key list and divide to subarrays with number of numClient
```

Design

- externalsortinginjava library
  - https://github.com/lemire/externalsortinginjava
  - used for merging sorted files into one file
  - There was a issue with a line separator
    - Add carriage return character(\r) explicitly
- bidirectional streaming rpc for shuffling
  - Suppose Slave A send all proper files to Slave B



What we learned from the project

- We need to finish the implementation sooner and allocate more time for testing.
  - Implementation completed on the evening of the last day.
- We should have done the design earlier.
  - Detailed design comes out 7 days before the due date.
- It would be better to meet offline regularly for 2-3 hours a week and have time to work on the project.
  - We met offline for three times during the project. Last two days of the project was not enough to complete the project.
- We needs motivation.