



# 2차 중간발표

KUORA

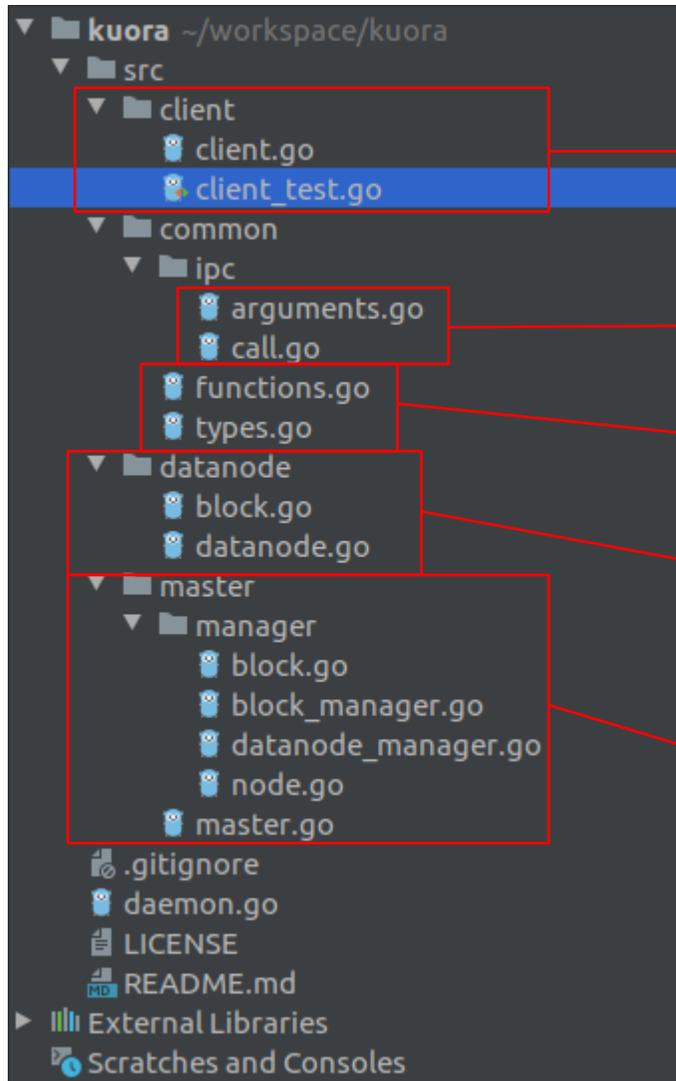
9 팀 (서지원, 홍나리) | 졸업프로젝트 1 | 유준범 교수님

2020.06.04



# Packages

KUORA



Client Package

- Client 기능 / Client Test Code

IPC Package

- RPC Call 기능 / RPC Data Structure

common Package

- User Defined Type / 공용 함수

datanode Package

- DataNode의 기능 및 Data Structure

master Package

- Master 기능 및 Data Structure

# TEST LOG

KUORA

```
INFO[0000] Run Master Node - - -
INFO[0000] INIT NEW BLOCK MANAGER
INFO[0000] INIT NEW DATANODE MANAGER
2020/06/04 00:14:41 rpc.Register: method "ExpireCheck" has 1 input parameters; needs exactly three
INFO[0002] New DataNode 127.0.0.1:40002
INFO[0005] New DataNode 127.0.0.1:40001
INFO[0019] Create File Operation
INFO[0019] map[127.0.0.1:40001:{0} 127.0.0.1:40002:{0}]
INFO[0019] map[]
INFO[0019] Create Block Info - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.724629909 +0900 KST m=+199.117396830 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Hot Key : 0_0
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.724629909 +0900 KST m=+199.117396830 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Hot Key : 0_0
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.724629909 +0900 KST m=+199.117396830 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Create File Operation
INFO[0019] map[127.0.0.1:40001:{0} 127.0.0.1:40002:{0}]
INFO[0019] map[]
INFO[0019] Create Block Info - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.759382461 +0900 KST m=+199.152149380 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Create Block Info - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.759413993 +0900 KST m=+199.152180913 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Create Block Info - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.75942652 +0900 KST m=+199.152193441 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Hot Key : 1_2_3_0
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.759382461 +0900 KST m=+199.152149380 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.759413993 +0900 KST m=+199.152180913 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.75942652 +0900 KST m=+199.152193441 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Hot Key : 1_2_3_0
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.759382461 +0900 KST m=+199.152149380 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Hot Key : 0_0
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.724629909 +0900 KST m=+199.117396830 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Hot Key : 1_2_3_0
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.759382461 +0900 KST m=+199.152149380 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Hot Key : 0_0
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.724629909 +0900 KST m=+199.117396830 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0019] Hot Key : 1_2_3_0
INFO[0019] Block Information - &{[0 0] 0 0 0 0} 2020-06-04 00:18:00.759413993 +0900 KST m=+199.152180913 [127.0.0.1:40001 127.0.0.1:40002] 127.0.0.1:40001}
INFO[0032] Remove Node - 127.0.0.1:40001
INFO[0032] Sweep Block Phase
INFO[0040] Remove Node - 127.0.0.1:40002
INFO[0040] Sweep Block Phase
```

```
INFO[0013] - TASK: Heartbeat
INFO[0013] - TASK: Heartbeat
INFO[0014] - TASK: Heartbeat
INFO[0014] - TASK: Heartbeat
INFO[0014] Create Block RPC Call
INFO[0014] Format: /home/wessup/testd1/Block_0.blk
INFO[0014] Create Block RPC Call [00][00][00]
INFO[0014] Format: /home/wessup/testd1/Block_1.blk
INFO[0014] Format: /home/wessup/testd1/Block_2.blk
INFO[0014] Format: /home/wessup/testd1/Block_3.blk
INFO[0015] - TASK: Heartbeat
INFO[0015] - TASK: Heartbeat
INFO[0016] - TASK: Heartbeat
```

```
INFO[0029] - TASK: Heartbeat
INFO[0029] - TASK: Heartbeat
INFO[0030] - TASK: Heartbeat
INFO[0030] Garbage Collection For 0 blocks
INFO[0030] - TASK: GarbageCollection
INFO[0030] - TASK: Heartbeat
INFO[0031] - TASK: Heartbeat
```

# Test

KUORA

✘ Tests failed: 2, passed: 3 of 5 tests – 290 ms

## Test Flow

- Generate Random Bytes
- Create File Blocks
- Write Bytes to Blocks
- Read Operation
- Compare Original Bytes and Read Bytes

### 2<sup>nd</sup> Test

- **Big File Create/Write/Read Operations**
- **Read File with Random Offset/Length**

**BUG T^T**

# Test Cases

Traceability Matrix

Basic Test		High-Level		1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	2.5	2.6	2.7	3.1	3.2	3.3	3.4	3.5	3.6
Write File	Exchange Metadata	→	○													○					
	Create	→						○	○										○		
	Write	→																			
	Replication	→			○							○									
Read File	Exchange Metadata	→										○		○		○					
	Read	→	○																		
Delete File	Delete	→		○							○						○				
		→				○							○		○						
		→								○									○		
System Test		High-Level								○										○	
Heartbeat	Heartbeat	→																			
Expire/Migration	Scheduled Expire	→							○	○		○		○							○
	Migration Instruction	→								○										○	
Re-Replication	Health Check	→							○			○		○							○
	Re Replication	→											○		○					○	
Garbage Collection	Scheduled GC	→											○		○					○	

# Test Case

KUORA

No.	Name	Description
1.1	Read File	Client에서 Key를 통해 Read를 요청하였을 때 해당 데이터를 읽어오는가 여부를 확인
1.2	Write File	Client에서 새로운 파일을 만들고 Write 요청을 할 때 해당 데이터가 DataServer에 올바르게 저장되는가 여부를 확인
1.3	Delete File	Client에서 Key를 통해 Delete 요청을 처리 후 해당 데이터가 Garbage Collection 되는지 확인
2.1	Heartbeat	Master가 지속적으로 DataServer로부터 요청을 받아 DataServer가 정상임을 확인할 수 있는지 여부를 확인
2.2	Expire/Migration	Master의 지시에 따라 DataServer가 Expire에 의한 Delete 혹은 Migration을 수행할 수 있는지 여부를 확인
2.3	Garbage Collection	Configure된 주기에 따라 DataServer 내의 실제 Chunk 데이터가 삭제되는지 여부를 확인
2.4	Replication	Client의 Write 후 DataServer에 Chunk 데이터가 복제가 되었는지 확인
2.5	Re-Replication	DataServer 하나를 중지시킨 후 해당 DataServer가 가진 데이터가 다른 서버에 복제되었는지 확인
Q1	Scalability	Master가 새로운 DataNode를 시스템에 포함시킬 수 있다.
Q2	Multiple Client Test	Client를 다수 실행하여 쓰기 및 읽기를 실행하였을 때 데이터의 동일성 및 Key의 유일성을 확인



구현 완료



미구현



일부 구현 / 진행중

# 3<sup>rd</sup> Iteration

KUORA

---

## 1. Client Side Bug Fix

### 1. Read Big File Operation (Multiple Blocks)

## 2. System Side Implementation

Client 측 Big File을 읽을 때 Read EOF 버그 수정

### 1. Re-Replication

### 2. Block Sweep

### 3. Expire/Migration

Fault-Tolerance를 위한 Re-Replication 및 Block Sweep 기능

+ Migration 기능 구현

# 3<sup>rd</sup> Iteration

KUORA

No.	Name	Description
1.1	Read File	Client에서 Key를 통해 Read를 요청하였을 때 해당 데이터를 읽어오는가 여부를 확인
1.2	Write File	Client에서 새로운 파일을 만들고 Write 요청을 할 때 해당 데이터가 DataServer에 올바르게 저장되는가 여부를 확인
1.3	Delete File	Client에서 Key를 통해 Delete 요청을 처리 후 해당 데이터가 Garbage Collection 되는지 확인
2.1	Heartbeat	Master가 지속적으로 DataServer로부터 요청을 받아 DataServer가 정상임을 확인할 수 있는지 여부를 확인
2.2	Expire/Migration	Master의 지시에 따라 DataServer가 Expire에 의한 Delete 혹은 Migration을 수행할 수 있는지 여부를 확인
2.3	Garbage Collection	Configure된 주기에 따라 DataServer 내의 실제 Chunk 데이터가 삭제되는지 여부를 확인
2.4	Replication	Client의 Write 후 DataServer에 Chunk 데이터가 복제가 되었는지 확인
2.5	Re-Replication	DataServer 하나를 중지시킨 후 해당 DataServer가 가진 데이터가 다른 서버에 복제되었는지 확인
Q1	Scalability	Master가 새로운 DataNode를 시스템에 포함시킬 수 있다.
Q2	Multiple Client Test	Client를 다수 실행하여 쓰기 및 읽기를 실행하였을 때 데이터의 동일성 및 Key의 유일성을 확인



구현 완료



미구현



일부 구현 / 진행중