Seattle University File System

CPSC 4910 Cloud Computing
Winter 2020, v1.0
CheckPoint #1
Team 7

Design Description:

The following is design choices we will be making regarding the implementation of the SUFS. This section will be updated anytime a new choice may be made.

- The reads and writes of our system will follow the normal HDFS process, but we will have minor checks in our writes to prevent over replication.
- We will consider a node to have failed if it does not respond to two consecutive heartbeats. We will also include a 5 second buffer along with them in case there is a delay in the NameNodes reception.
- When we write to the file system there is the concern that heartbeats immediately following the write may enter a race condition with writes on the DataNode. To ensure we give the write time to complete on each DataNode before checking appropriate replication via heartbeat, we will not consider replication of that file in the following heartbeat.
- When a failure in a DataNode occurs, we will contact AWS API and restart a DataNode. We will have that DataNode contact the NameNode to figure out which DataNode was most recently updated/up to date. It will then use that DataNode data to copy to it's new one.

RESTful API:

- Block Report/Heartbeat
- Writes

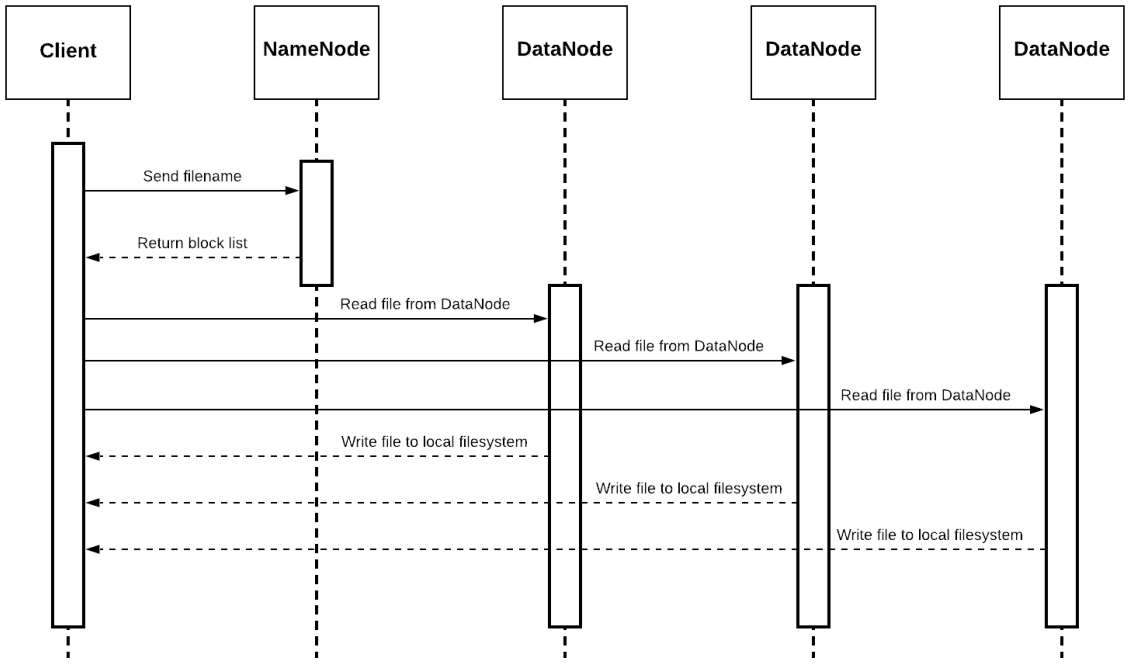
Technologies/Tools:

- Technologies:
  - Language: Java
  - Development: AWS SDK for Java 2.0, OpenJDK 13
- Development tools:
  - Repository: AWS CodeCommit
  - Platforms: AWS EC2
  - REST Framework: Spring
  - Testing: Postman

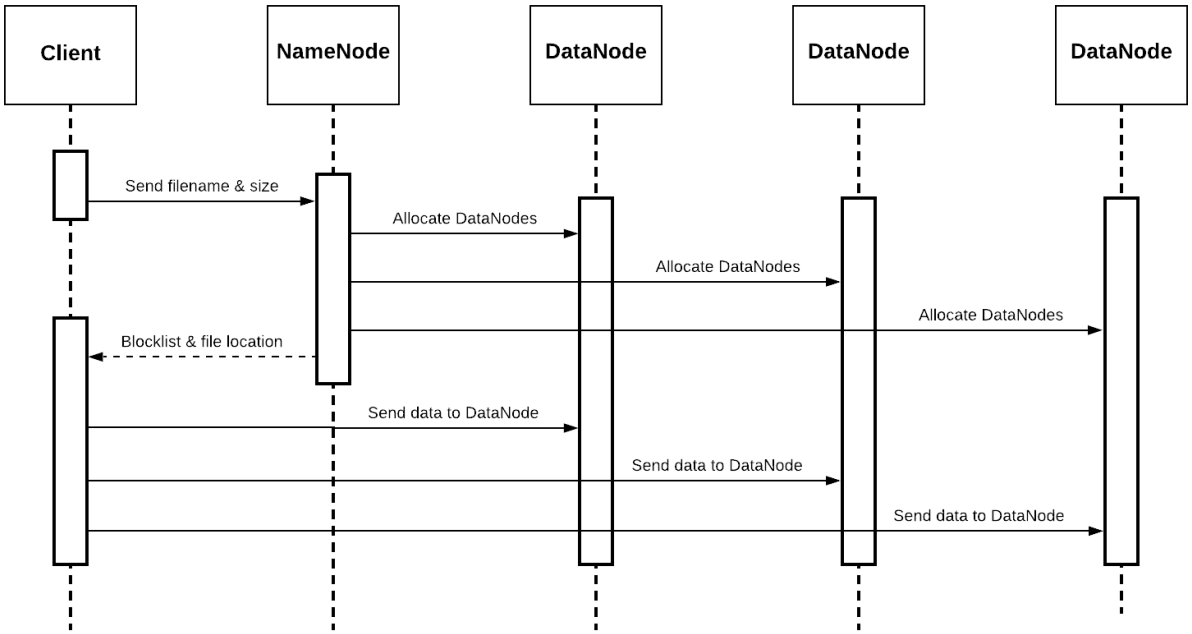
System Parameters:

- Block size: 128 MB
- Replication factor: N = 3
- Block report frequency: Every 30 seconds
- Heartbeat: Not needed, block report is going to carry on the job of the heartbeat

Sequence Diagrams:
Reads:



Writes:



DataNode Failure: