Fault Tolerance Analysis for Hadoop MapReduce on Gfarm Distributed File System

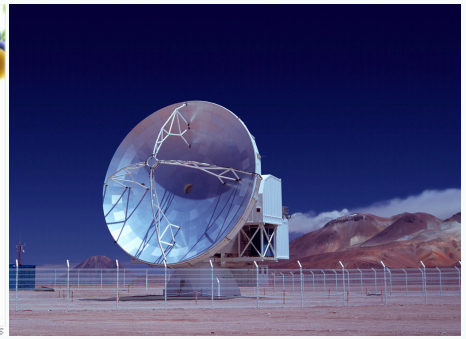
Graduate School of System and Information engineering
Dept. of Computer Science 1st Year
201120781 Marilia Melo
Supervisor: Osamu Tatebe

Outline

- Introduction
- Research Target
- Background
  - MapReduce
    - Hadoop
  - Distributed File Systems
    - HDFS
    - Gfarm
  - Hadoop-Gfarm plugin
- Proposed System Analysis
- Implementation Issues
- Related Work
- Conclusion and Work Proposal

Introduction

- The society has been generating more and more data
  - Companies
    - User transactions
    - System log data
  - Research
    - Genome project
    - Astronomical Data Analysis – TB ~ PB/year/telescope
    - Petabyte, Exabyte scale data intensive computing



Introduction

- Data Analysis of Large Scale data is necessary
  - Computation is distributed across computers
  - Cluster of Computers
  - Clouds
- In order to process data in clusters and clouds:
  - Distributed Data Process
    - MPI (Message Passing Interface)
    - MapReduce
  - Distributed File System
    - Google File System, HDFS, Gfarm

Research Target

- Ability to process large scale data by using a simple programming framework executed over a reliable and powerful distributed computing system.
- Scientific Large Scale Data: requires POSIX API
- Programming Framework: Hadoop
- Distributed computing system: Gfarm
- **Provide fault tolerance capabilities to Hadoop-Gfarm plugin**

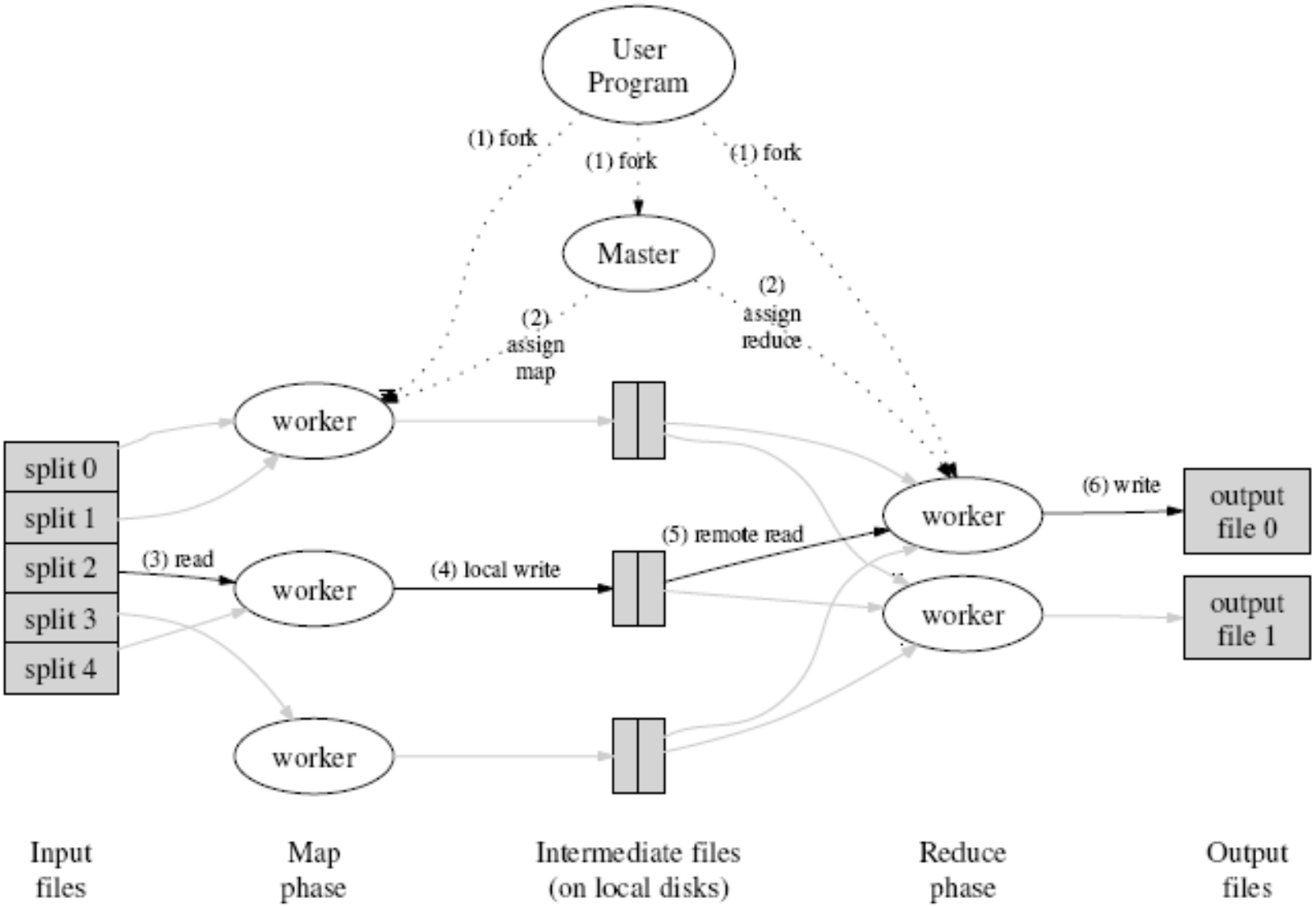
Background

- MapReduce
  - Hadoop
- Distributed File System
  - HDFS
  - Gfarm
- Gfarm-Hadoop plugin

MapReduce

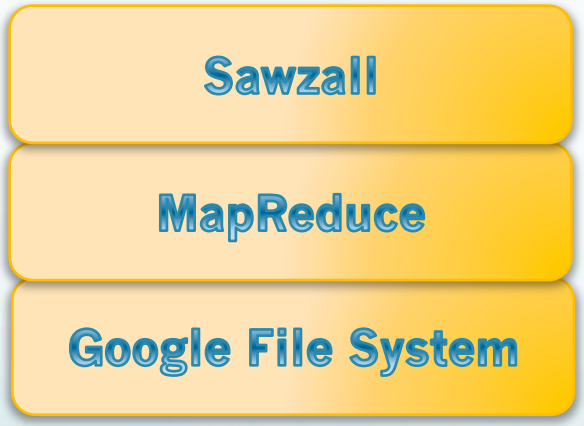
- MapReduce [2004 Dean]
  - Large scale data process framework
  - Map, Shuffle and Reduce phase
  - Parallel implementation details are hidden
  - Google's implementation uses Google File System as Data Storage

MapReduce Work Flow



Hadoop

- Open source implementation of Google MapReduce
- Uses Hadoop Distributed File System (HDFS) as data input/output



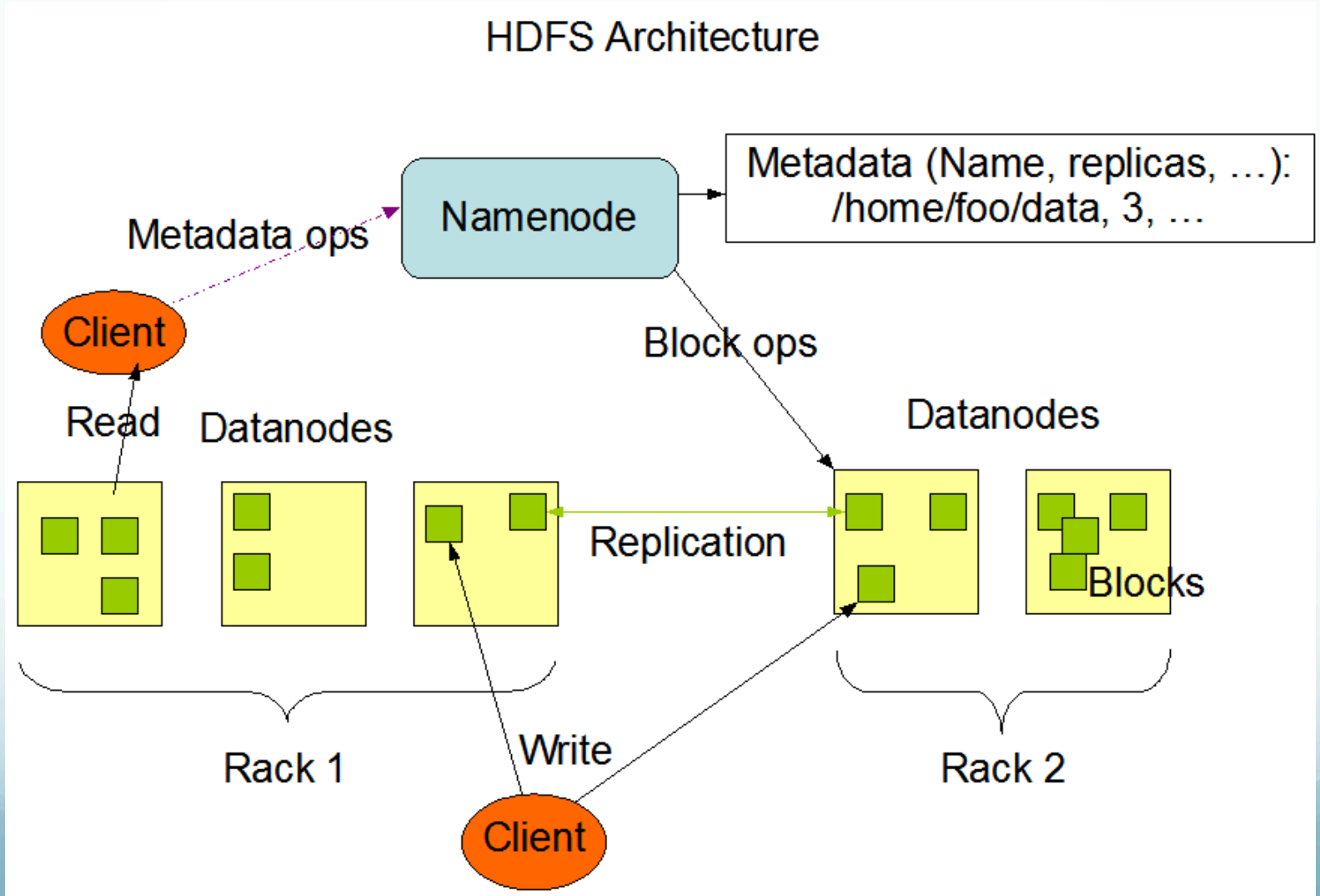
Google's MapReduce structure

Open Source Technology structure

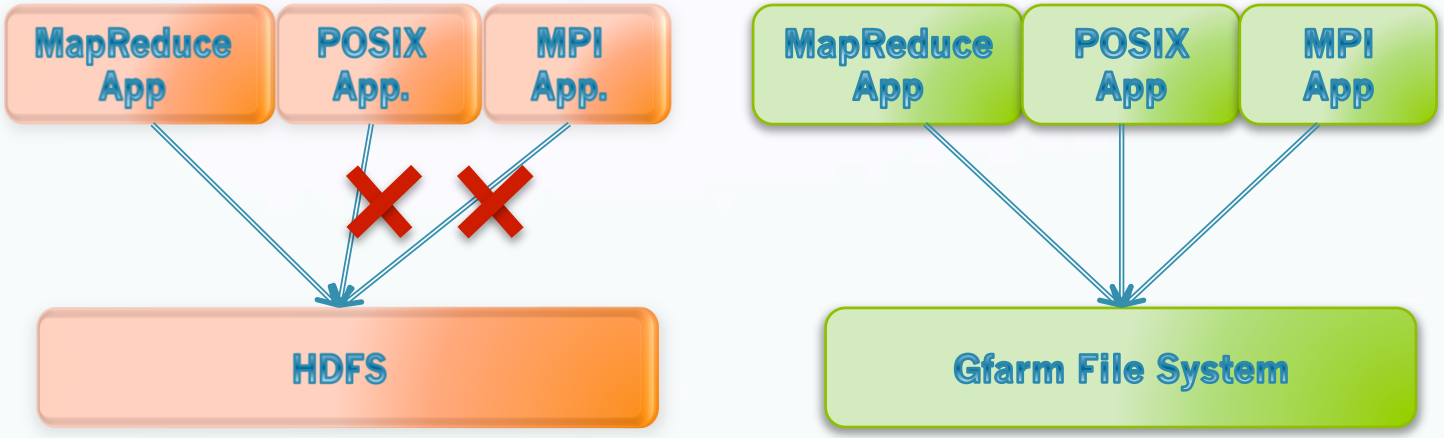
Distributed File System 1/2

- HDFS
  - Master
    - Manages file system namespace
    - Regulate access to files
  - Slave
    - Stores data
  - Data I/O occurs directly to slaves
  - File blocks (default 64MB)
  - Reliability by replication
  - Data locality

HDFS architecture



HDFS Problems



- Difficult to run jobs other than MapReduce
  - Not a POSIX compliant file system
  - No file modification other than append
  - No concurrent writes to a single file from multiple clients

HDFS Problems

- Data import/export is required
  - Time consuming
  - Waste of storage

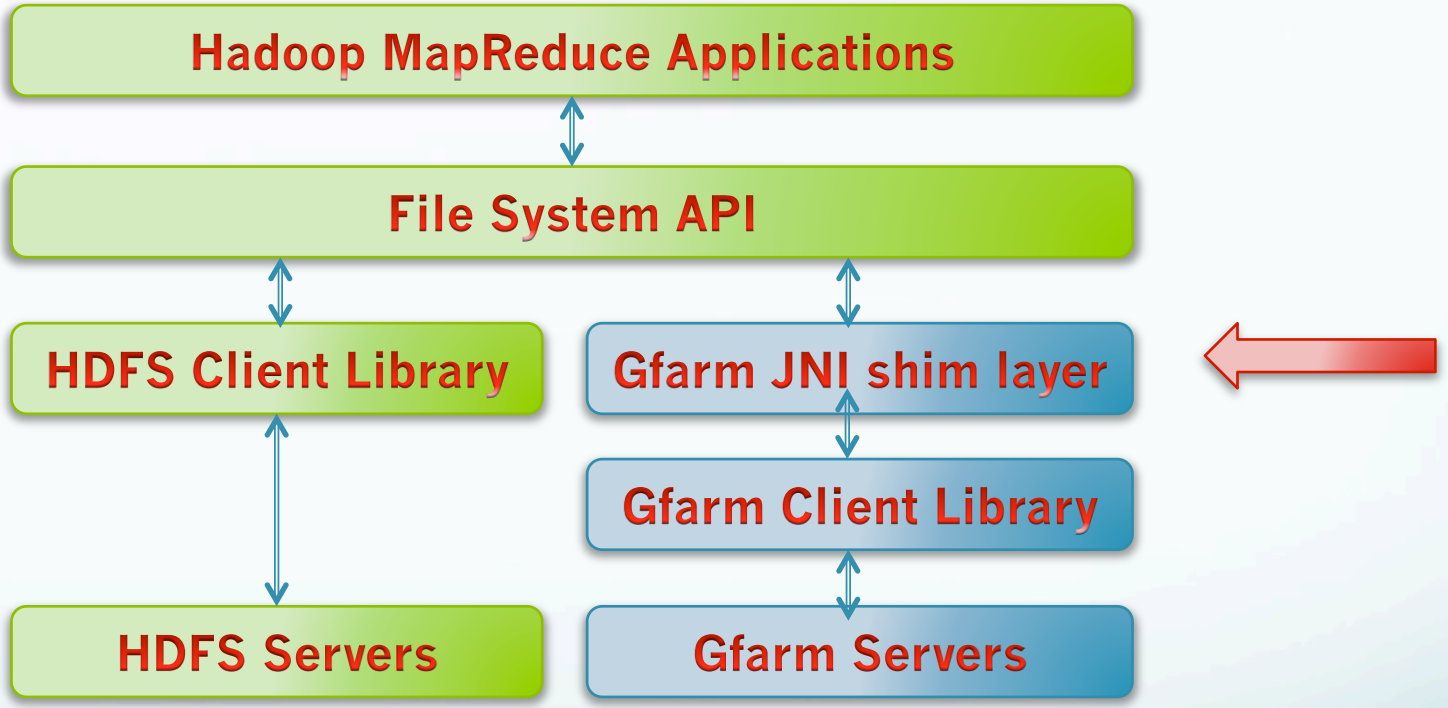
Distributed File System 2/2

- Gfarm [2002 Tatebe]
  - Grid File System
  - Master
    - Storages metadata and provides namespace
  - Slave
    - Storages data
  - Explores data locality
  - Provide fault tolerance by transparent replica access
  - Storages by file unit
  - POSIX compliant API
  - Can be accessed by local software using FUSE

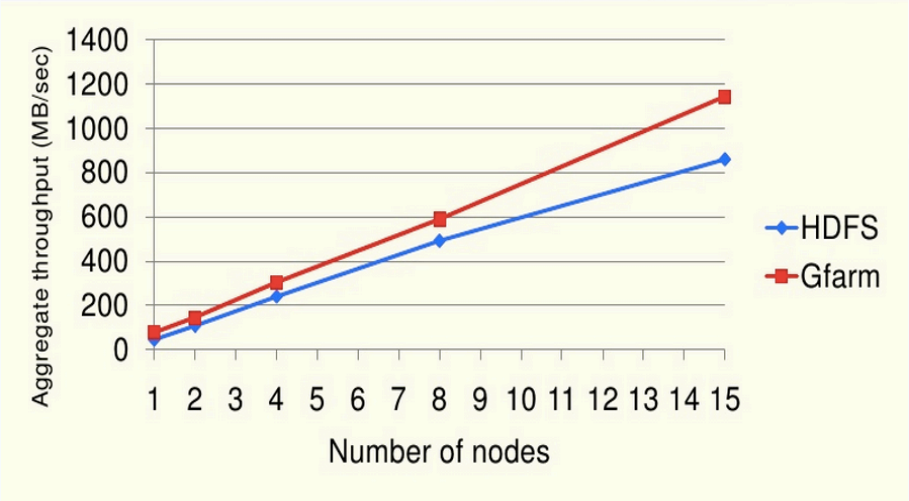
Hadoop-Gfarm plugin

- *Using the Gfarm File System as a POSIX compatible storage platform for Hadoop MapReduce applications* [2010 Mikami, Ohta, Tatebe]
- Plugin to enable Hadoop to directly interact with Gfarm
- Use Hadoop's provided File System API to access
  - org.apache.hadoop.fs.FileSystem
- Explores locality by using *getFileBlockLocations* function
- Implemented using Java Native Interface (JNI)

Hadoop-Gfarm Architecture



Hadoop-Gfarm plugin Analysis

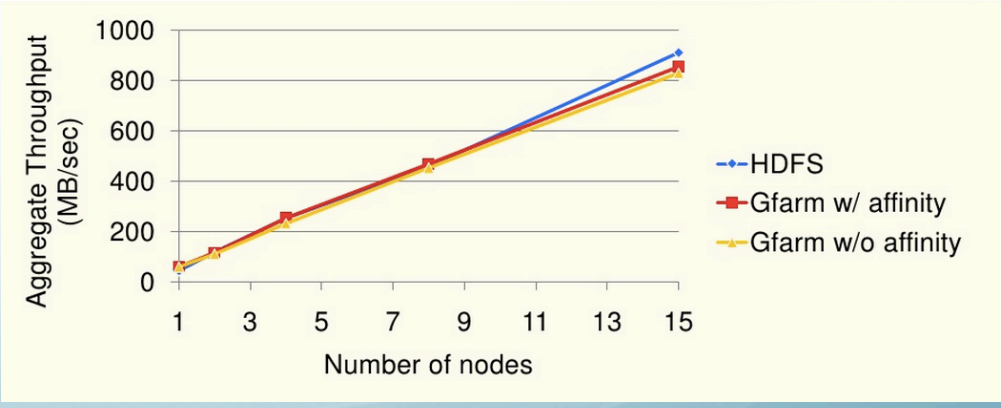


Write Performance

- Hadoop provided TestDFSIO benchmark
- Writes 50GB data (no replication)
- Gfarm presents around 30% more performance

Read Performance

- TestDFSIO read
- Each nodes read 5GB files
- Almost same performance as HDFS
- Affinity refers to data locality



* Data provided by Mikami

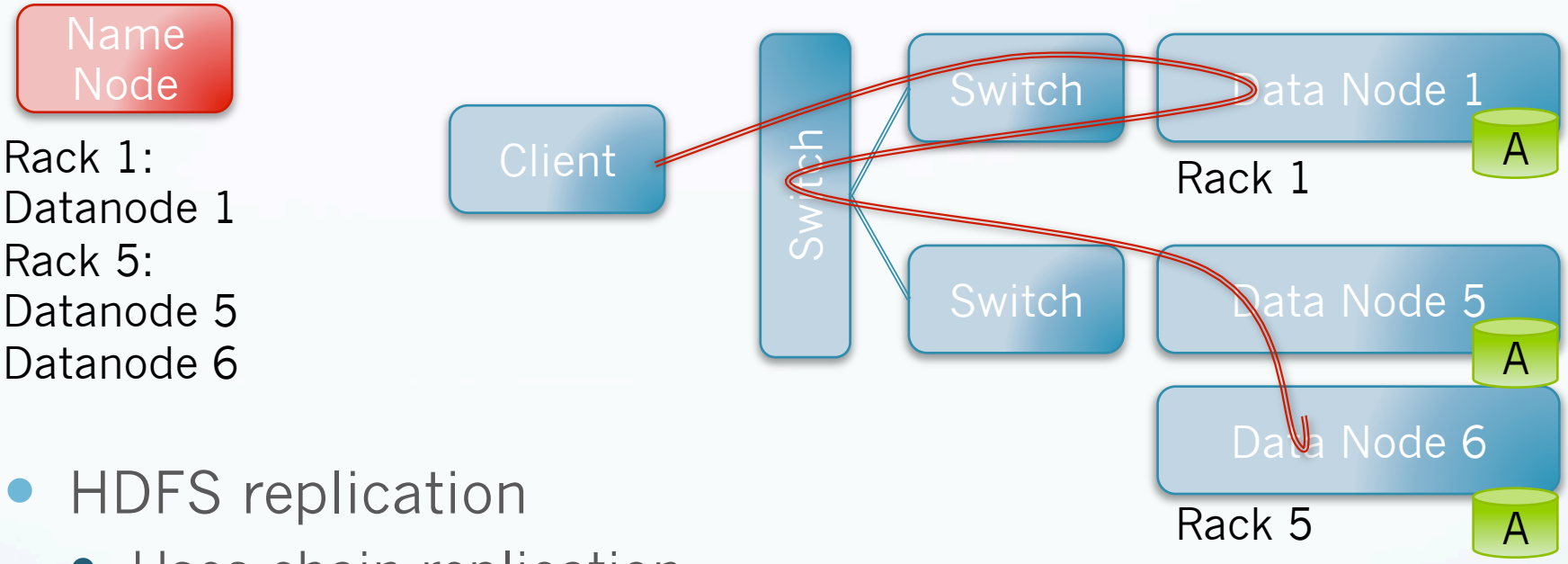
Hadoop-Gfarm reliability issue

- Distributed Parallel Computing requires Reliability
  - A single computer might fail once a year
  - With 365 computers, one will fail everyday
  - 36.500 computers, every few hours
  - To be useful, scalable system requires fault tolerance
- Hadoop-Gfarm plugin has been implemented without replication features

Proposed System Analysis

- Provide fault tolerance to Hadoop-Gfarm plugin by implementing replication
- Analysis of system performance by applying default benchmarks
- Experiment fault tolerant capability to systems like Impala, which runs on HDFS but does not have reliability

Replication in HDFS



- HDFS replication
  - Uses chain replication
  - By the time the client finishes writing the data, the replicas have already been created
  - The master Namenode is a Single Point of Failure, since it can't be replicated

Replication in Gfarm

- Gfarm replication
  - Create replicas in the background
  - After the client has finished writing data
  - Replicas should not change write performance
- Gfarm can also replicate the master node
  - Solves HDFS Single Point of Failure problem

Implementation Issues

- MapReduce has been intensively researched
  - System need exists
- Many alternatives to HDFS with POSIX compliant requirements
  - Efficient solution has not been provided yet
- Systems to implement:
  - Hadoop
  - Gfarm
  - Hadoop-Gfarm plugin

Related Work

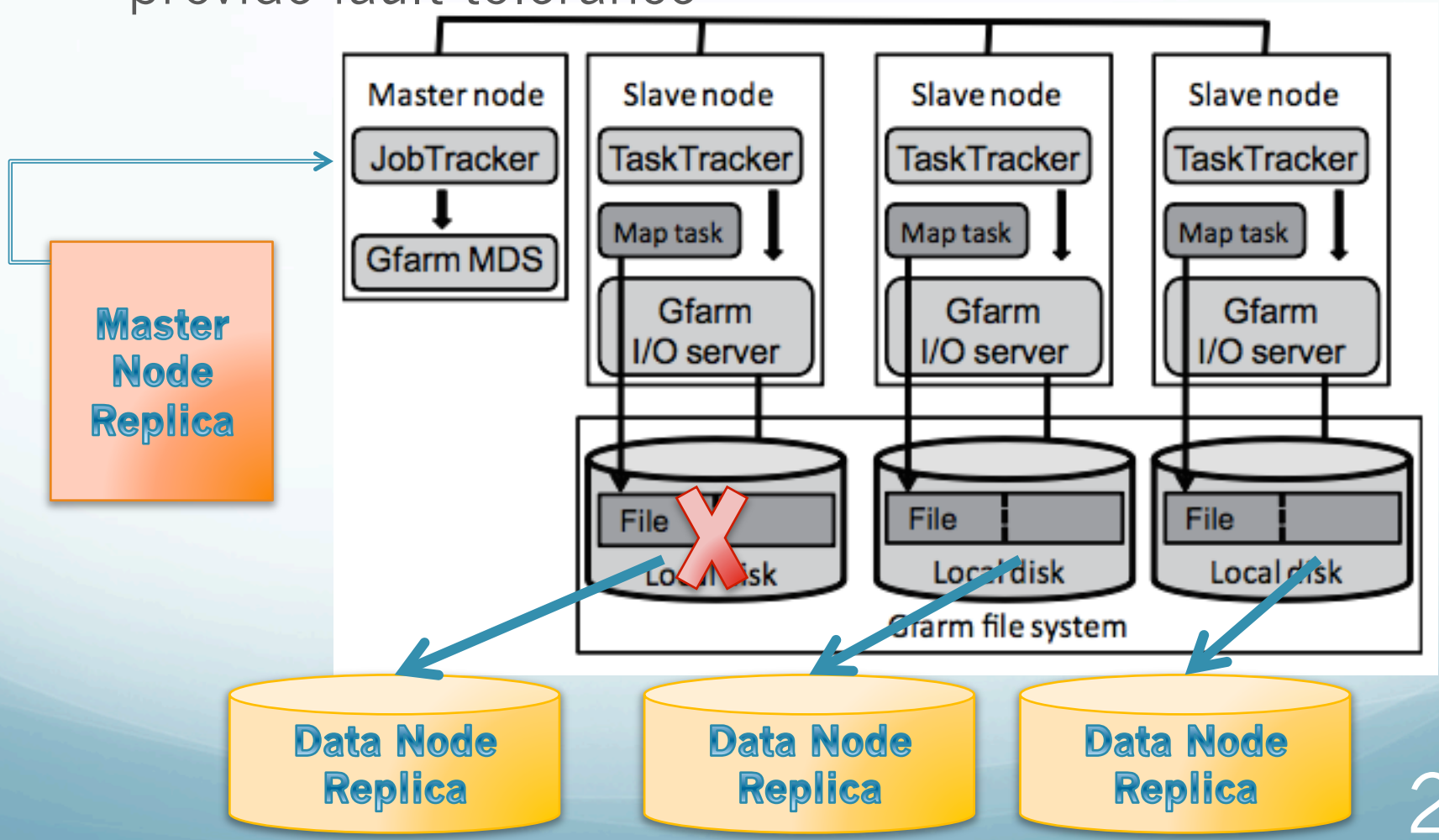
- CloudStore
  - Master, Slave, Block split, like HDFS
  - POSIX-like C++ API, FUSE
  - There is no authentication or file permission features
- Hadoop-PVFS
  - Striping cluster file system
  - The I/O bandwidth is limited by network

Conclusion

- In research we analyzed the current mainly used solutions for High Performance Cloud Computing
- MapReduce framework has been more and more utilized for data analysis processing
- Distributed File System with POSIX compliance is required for scientific research
- Hadoop-Gfarm plugin enables Hadoop MapReduce to execute on Gfarm File System
- However, this plugin does not provide fault tolerance capabilities

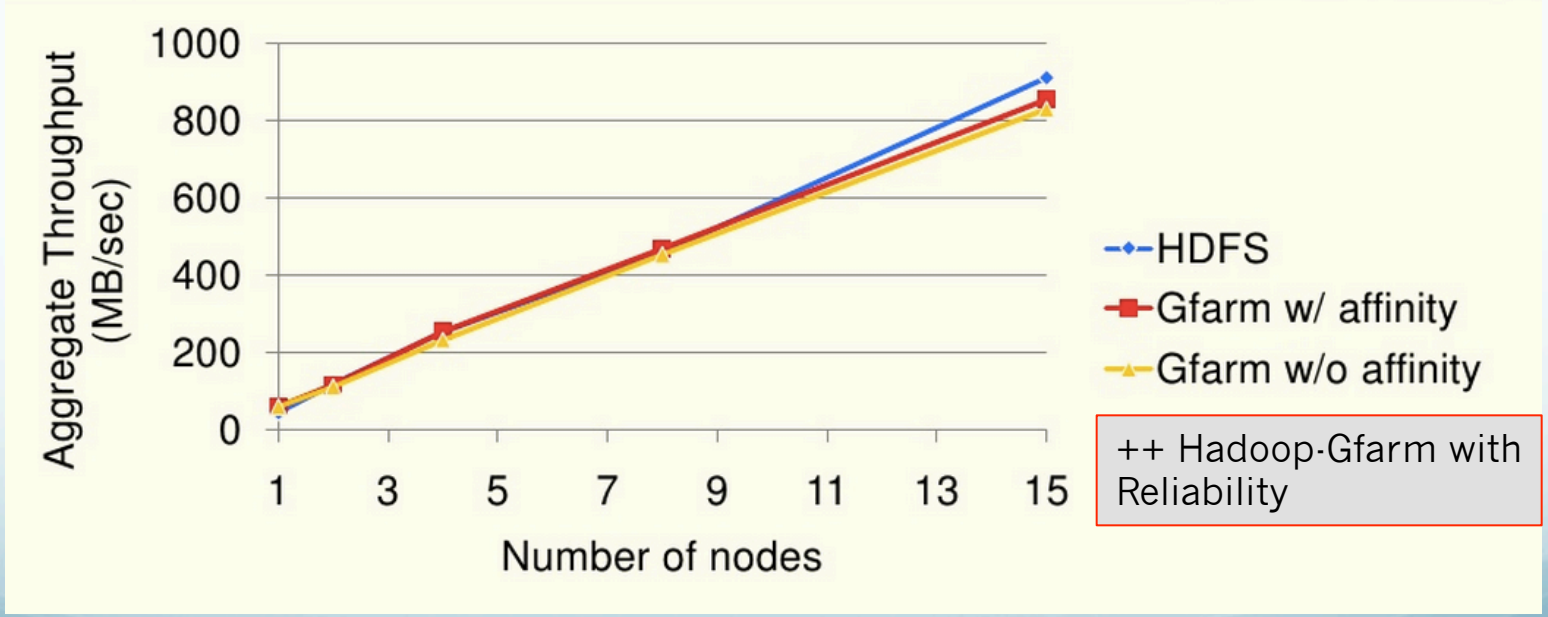
Work proposal 1/3

- Deploy Hadoop-Gfarm plugin using replication to provide fault tolerance



Work Proposal 2/3

- Benchmark the plugin's performance with this new reliability layer
- Should not impact in performance results



Work Proposal 3/3

- Application use: Implement Impala and other applications with Hadoop-Gfarm plugin providing fault tolerance

Thank You