Hadoop Acceleration in an OpenFlow-based cluster

Sandhya Narayan, Stu Bailey
InfoBlox Inc.

Anand Daga
University of Houston

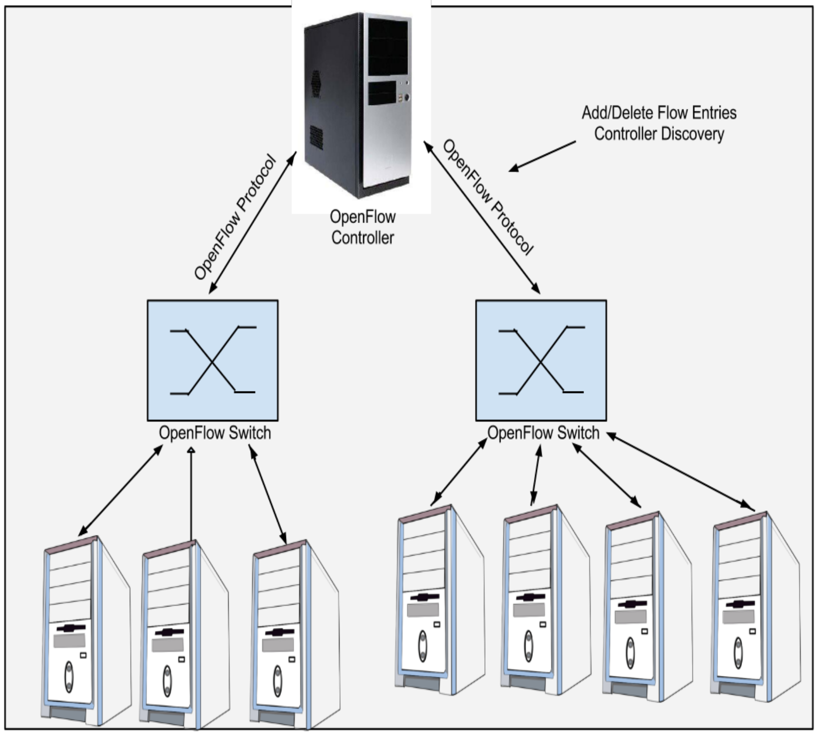
Infoblox Inc. is the market leader in automated network control providing scalable resilient network services, with failover, recovery and seamless maintenance. Our products for DNS, DHCP, IPAM and network change management automate business critical network services allowing our customers to deliver massively scalable, always-on services.

Presentation Outline

- OpenFlow
- Hadoop
- OpenFlow and Hadoop
- Experimental Setup
- Experiments
- Results and Conclusions

OpenFlow Overview

Software-Defined Networking is an emerging architecture that provides an open interface to control forwarding decisions of network devices.

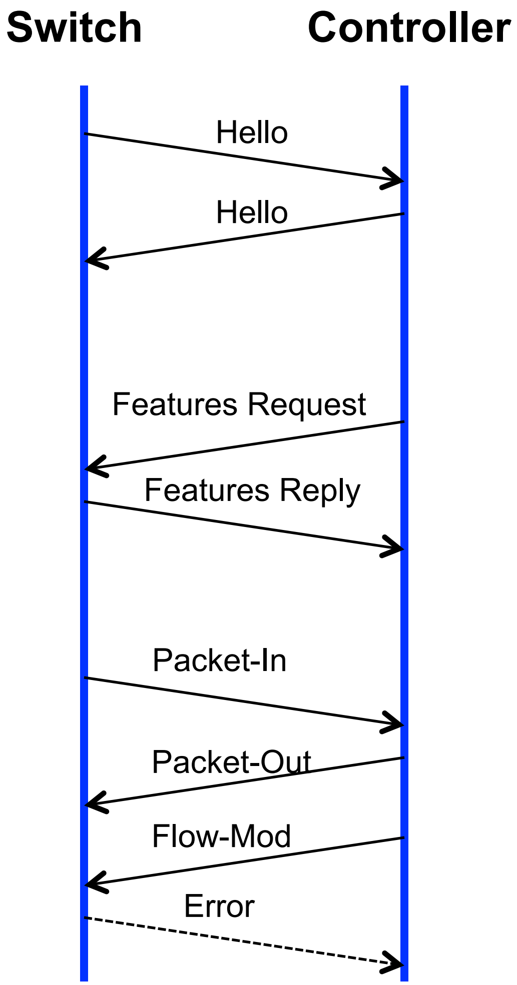


OpenFlow:

- Leading standard for SDN.
- Being developed and promoted by Open Networking Foundation.
- The OpenFlow protocol defines a standard interface for installing flexible packet-handling rules in network switches by a separate centralized program called the OpenFlow controller.

How OpenFlow works

- OpenFlow rules are sent by the controller and specify how packets should be processed by the switch.
- A rule consists of
  - pattern that identifies a set of packets
  - list of actions that specifies how the packets should be processed.
- Examples:
  - Path Selection: If dst-ip = 10.70.33.6, output packet on port 1
  - Optional Quality-of-Service (QoS) via queuing mechanism: If dst-ip = 10.70.33.6, enqueue packet on queue 1 of port 1



Interactions between switch and controller

OpenFlow Use Cases

- Variety of applications that introduce new network functionality or improve existing ones.
  - Flexible access control
  - Web server load balancing
  - Energy-efficient networking
  - Automated data center provisioning
  - Seamless virtual-machine migration
  - WAN Traffic Engineering
  - ….
- **Our Use Case: Hadoop Job Acceleration**

Hadoop Overview

- **The leading platform for Big Data Analytics**
- **Provides a reliable storage and analysis system that is scalable and built from commodity hardware.**
- **Core Hadoop**
  - HDFS: Distributed storage system called HDFS
  - MapReduce framework: Distributed compute system that deliver scalable, reliable parallel processing services
- **Open source, written in Java, many distributions available**
- **More nodes-> more compute power, more storage and more I/O bandwidth**
- **Scale: Thousands of nodes, petabytes of data**
- **Hadoop Ecosystem**

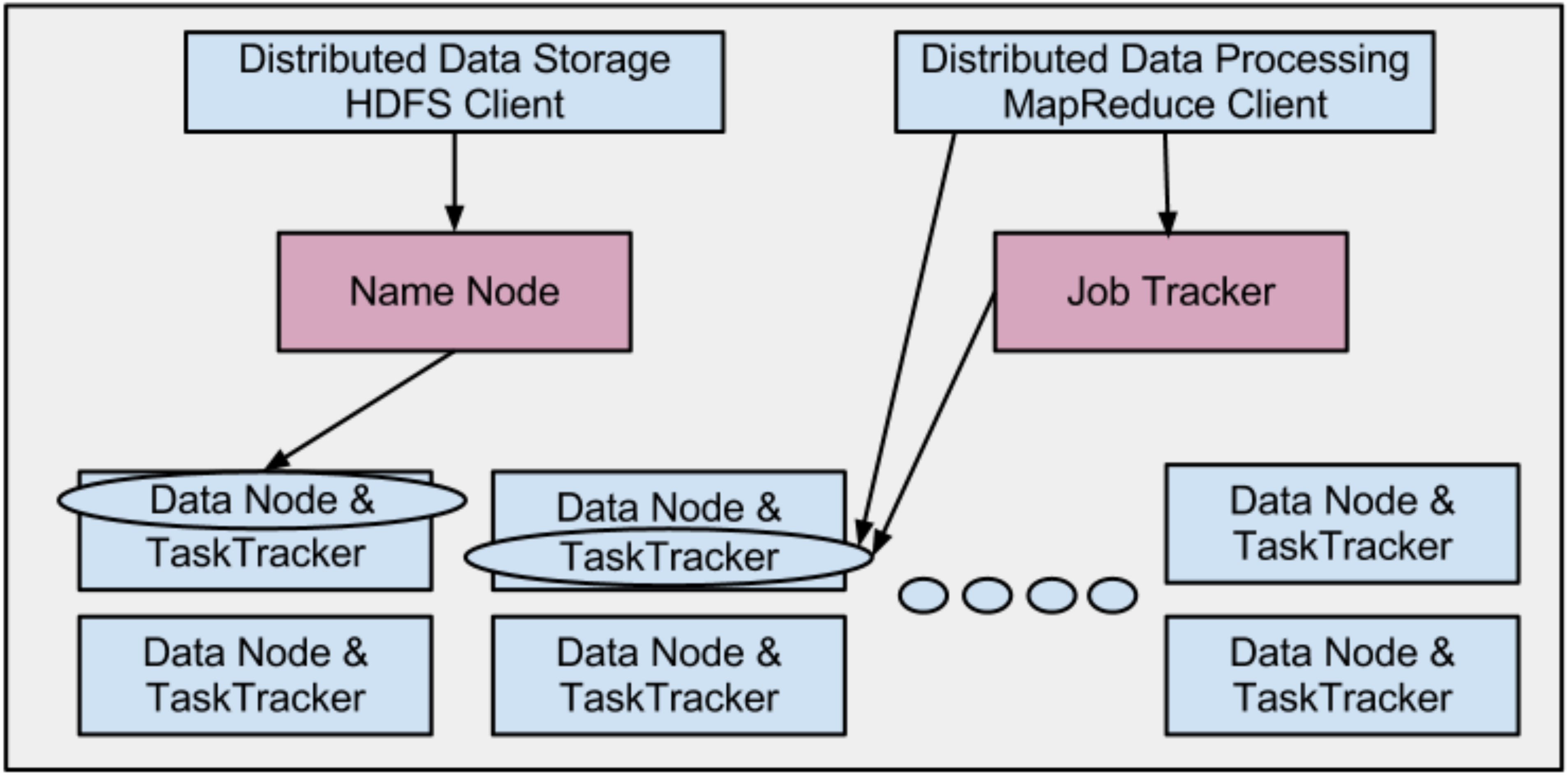
Hadoop System Characteristics

- Cluster Size- number of nodes
- Node Characteristics – CPU, Memory, Storage, Disk I/O
- Network Characteristics – Link bandwidth, latency, topology
- Job mix- Type, data size, frequency
- Ability to parallelize the computation into MapReduce execution model
- Ability to process data locally where it is resident
- Background activity in the cluster

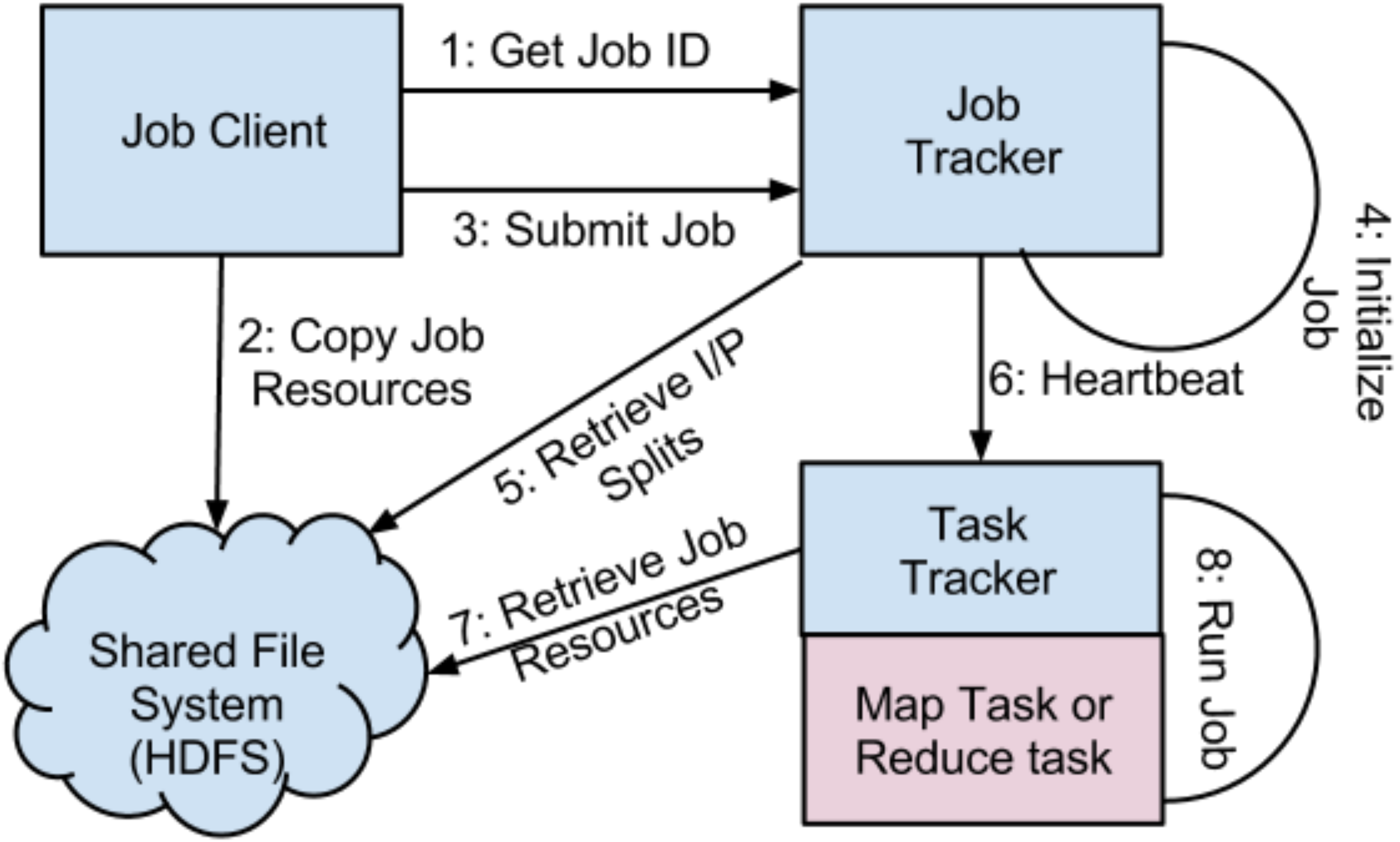
Hadoop Acceleration Approaches

- Throw more hardware at the problem – Hadoop does this well
- Throw more powerful systems at the problem – NOT Hadoop approach!
- Improve caching
- Improve disk I/O
- Improve scheduling
- Improve network – use faster Interconnect – 10G, Infiniband
- **Our approach – explore OpenFlow to setup the network characteristics from the application**

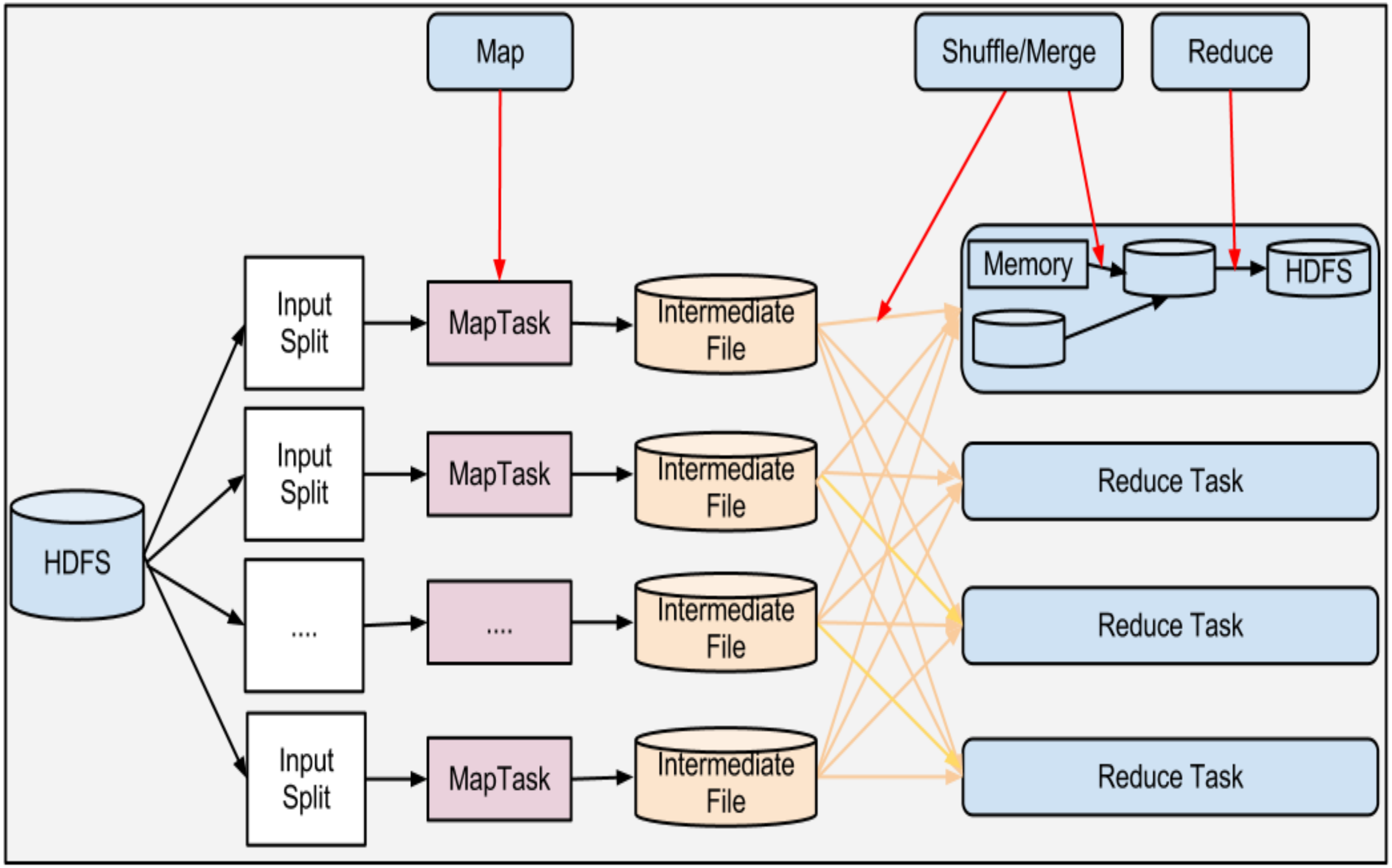
HDFS and MapReduce



Life Cycle of a Hadoop Job



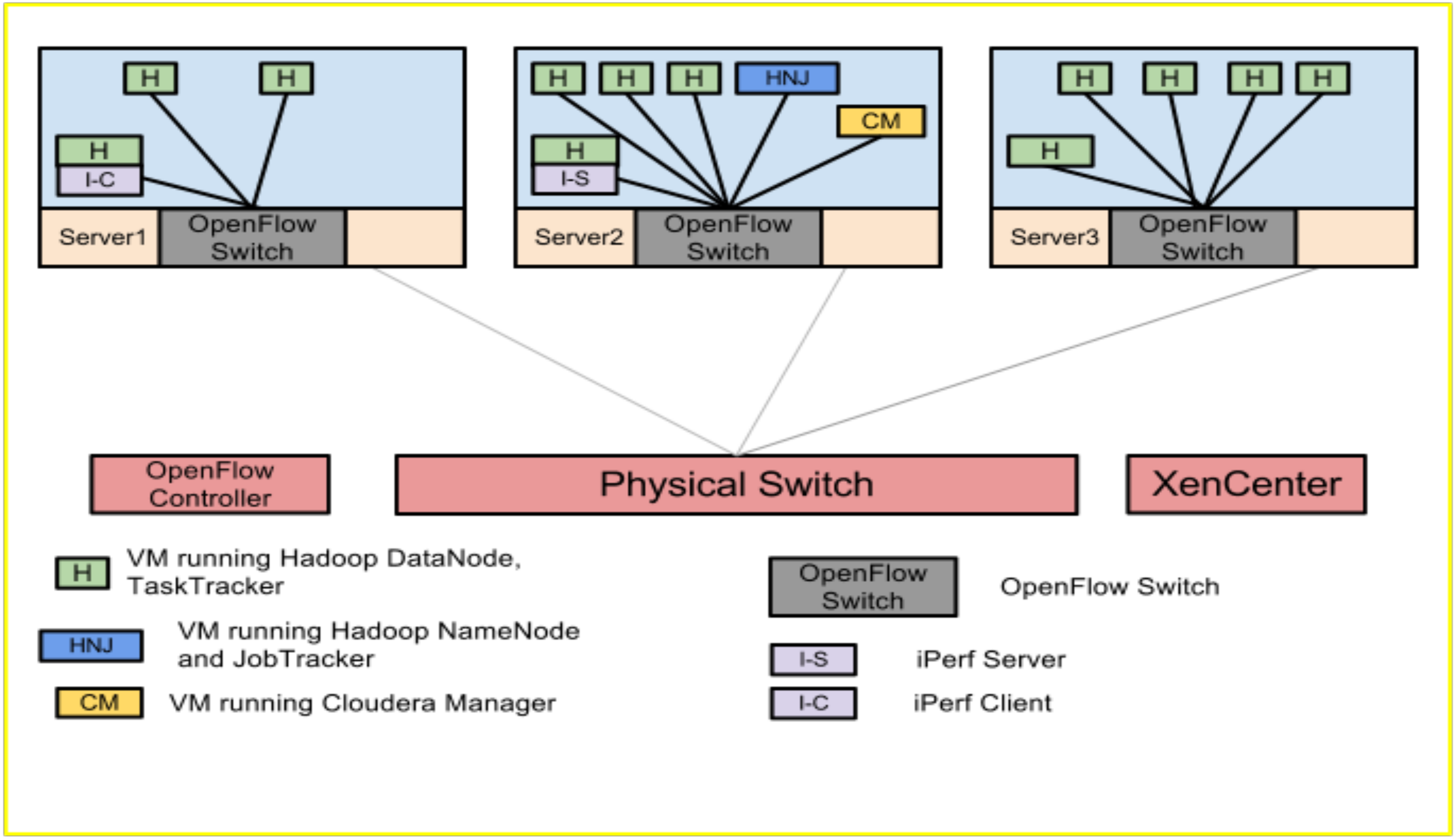
MapReduce Pipeline



Network Traffic Hadoop

- **Categories of network traffic in Hadoop :**
  - HDFS read and write by clients
  - HDFS replication
  - Hadoop split traffic between HDFS and TaskTrackers
  - Hadoop results stored in HDFS
  - Interactions between NameNode and DataNode
  - Interactions between JobTracker and TaskTracker
  - **Our Focus: Interaction between TaskTrackers which includes Hadoop shuffle traffic**

OpenFlow based Hadoop Cluster



Shuffle Traffic Acceleration

- Using queues
  - Set flows statically to use higher maximum rate queue for shuffle traffic
  - Set flows statically to provide lower maximum rate queue to extraneous traffic
  - Set flows dynamically to accelerate some shuffle traffic over others
- Using different paths
  - Provide higher bandwidth paths to higher priority jobs
  - Provide less congested paths to higher priority jobs
- Our current approach: use queues so we explore the use of OpenFlow in a small cluster with just three systems.

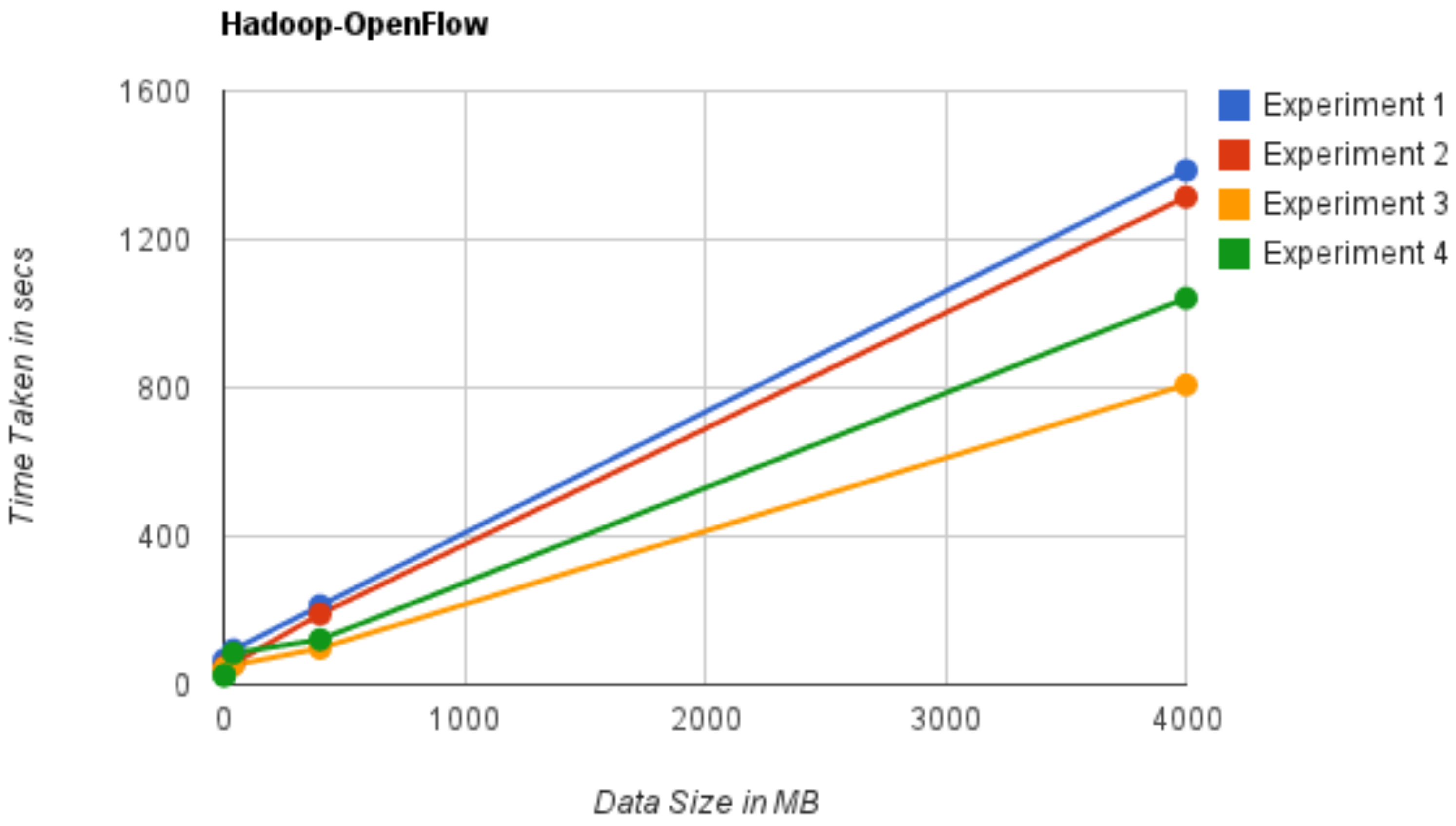
Experiment

- **Job under Test: Hadoop Sort Program**
  - A MapReduce program that is part of Hadoop distribution.
  - It performs a partial sort of its input.
  - It is very useful for benchmarking the whole MapReduce system, as the full input dataset is transferred through the intermediate shuffle phase.
  - Steps of the Hadoop Sort program: generate random data, perform sort, and validate the results.

Experiment Setup

- Setup queues on the outbound port of the switches with different maximum rates:
  - q0 (50 Mbps), q1 (200 Mbps), q2 (5 Mbps), q3 (250 Mbps)
- Used low maximum rates to artificially create congestion and investigate the benefit of OpenFlow switches under such situations.
- Generated additional traffic in the cluster using a utility called Iperf. Ran Iperf server on one VM and Iperf client on another VM on a different physical system.
- Ran Hadoop Sort jobs for the different input sizes(0.4 MB, 4 MB, 40 MB, 400 MB and 4GB).
- Experiment 1: All traffic on q0
- Experiment 2: Iperf traffic on q2, rest of q0.
- Experiment 3: Traffic between the Task Trackers to q1, and Iperf traffic on q2 and the rest on q0.
- Experiment 4: All traffic on q3

Results


Hadoop-OpenFlow
Time Taken in secs
1600
1200
800
400
0
0
1000
2000
3000
4000
Data Size in MB
Experiment 1
Experiment 2
Experiment 3
Experiment 4

Further Work

- Dynamically set flows from within Hadoop
- Provide better paths to some traffic categories
- Provide priority to some jobs
- Provide higher level abstractions
- Explore other applications

Thank You

Please visit

www.flowforwarding.org