NetFlow Analysis with MapReduce

Wonchul Kang, Yeonhee Lee, Youngseok Lee

Chungnam National University

{teshi85, yhlee06, lee}@cnu.ac.kr

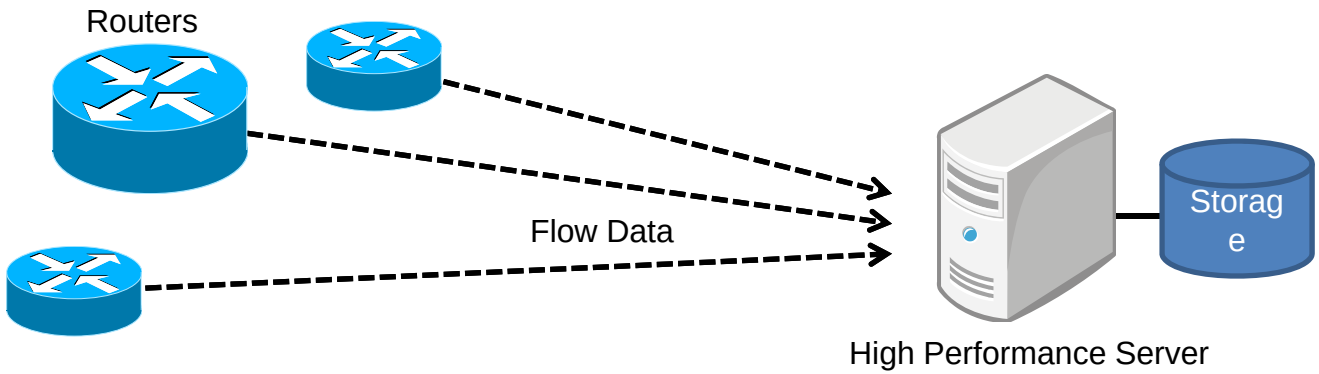
2010.04.24(Sat)

based on "An Internet Traffic Analysis Method with MapReduce", Cloudman workshop, April 2010

Introduction

- Flow-based traffic monitoring
  - Volume of processed data is reduced
  - Popular flow statistics tools : Cisco NetFlow [1]

- Traditional flow-based traffic monitoring
  - Run on a high performance central server



Motivation

- A huge amount of flow data
  - Long-term collection of flow data

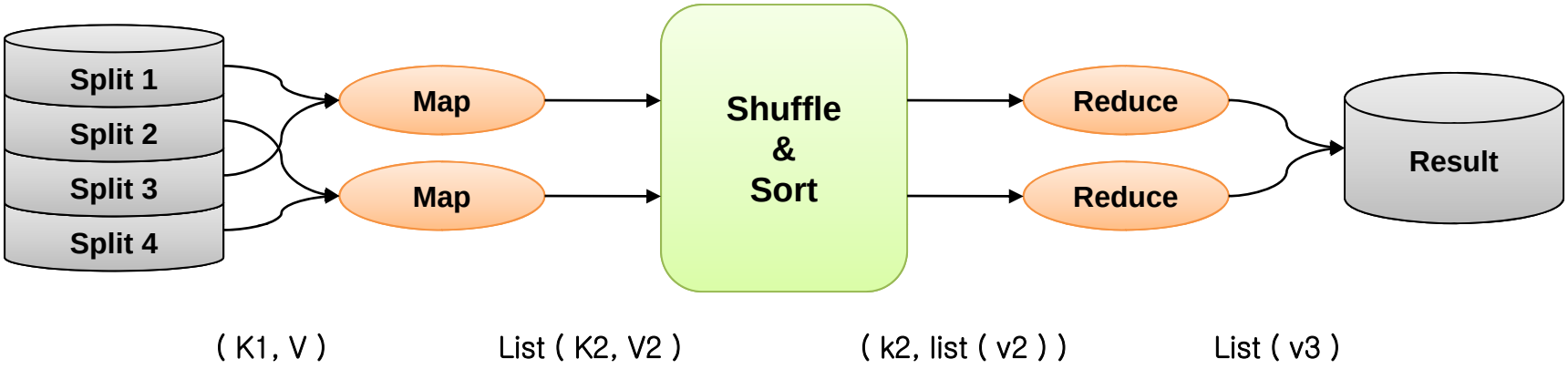
Flow data in our campus network (/16 prefix)			
# of Routers	1 Day	1 Month	1 Year
1	1.2 GB	13 GB	156 GB
5	6 GB	65 GB	780 GB
10	12 GB	130 GB	1.5 TB
200	240 GB	2.6 TB	30 TB

  - Short-term period of flow data
    - Massive flow data from anomaly traffic data of Internet worm and DDoS
- Cluster file system and cloud computing platform
  - Google's programming model, MapReduce, big table [8]
  - Open-source system, Hadoop [9]

MapReduce

- MapReduce is a programming model for large data set
- First suggested by Google
  - *J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Cluster," OSDI, 2004 [8]*
- User only specify a map and a reduce function
  - Automatically parallelized and executed on a large cluster

MapReduce



- Map : return a list containing zero or more (k, v) pair
  - Output can be a different key from the input
  - Output can have same key

- Reduce : return a new list of reduced output from input

Hadoop

- Open-source framework for running applications on large clusters built of commodity hardware
- Implementation of MapReduce and HDFS
  - MapReduce : computational paradigm
  - HDFS : distributed file system
- Node failures are automatically handled by framework
- Hadoop
  - Amazon : EC2, S3 service
  - Facebook : analyze the web log data

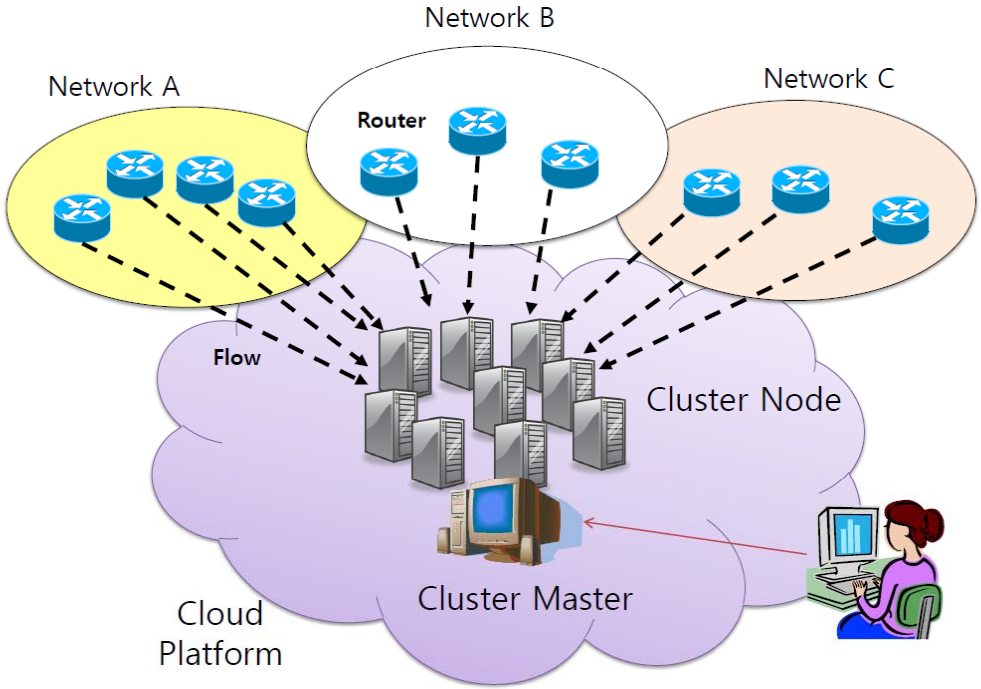
Related Work

- Widely used tools for flow statistics
  - Flow-tools, flowscan or CoralReef[5]
- P2P-based distributed analysis of flow data
  - DIPStorage : each storage tank associated with a rule [11]
- MapReduce software
  - Snort log analysis : NCHC cloud computing research group [16]

Contribution

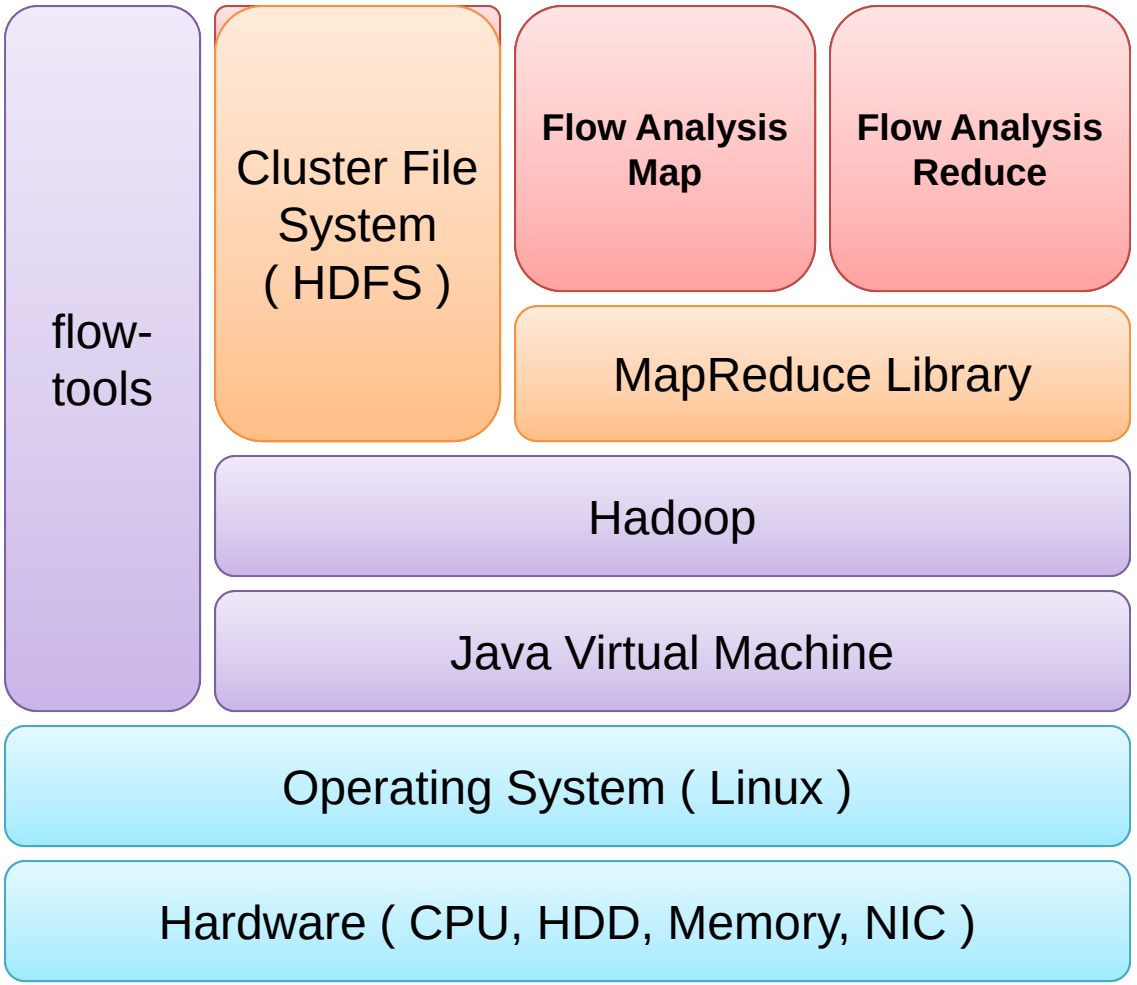
- A flow analysis method with MapReduce
  - Process flow data in a cloud computing platform, hadoop
- Implementation of flow analysis programs with Hadoop
  - Decrease flow computation time
  - Enhance fault-tolerant of flow analysis jobs

Architecture of Flow Measurement and Analysis System



- Each router exports flow data to cluster node
- Cluster master manages cluster nodes

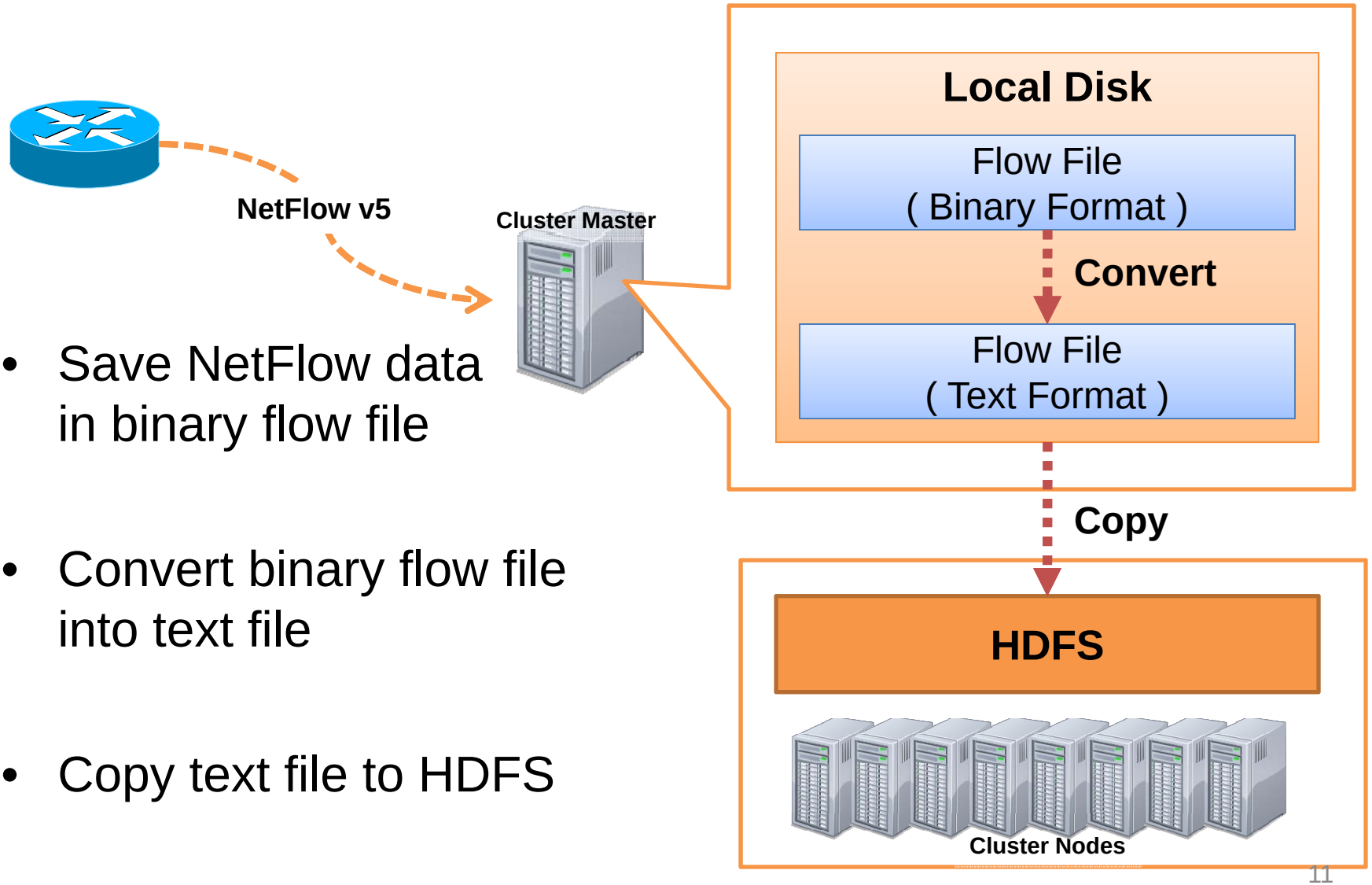
Components of Cluster Node



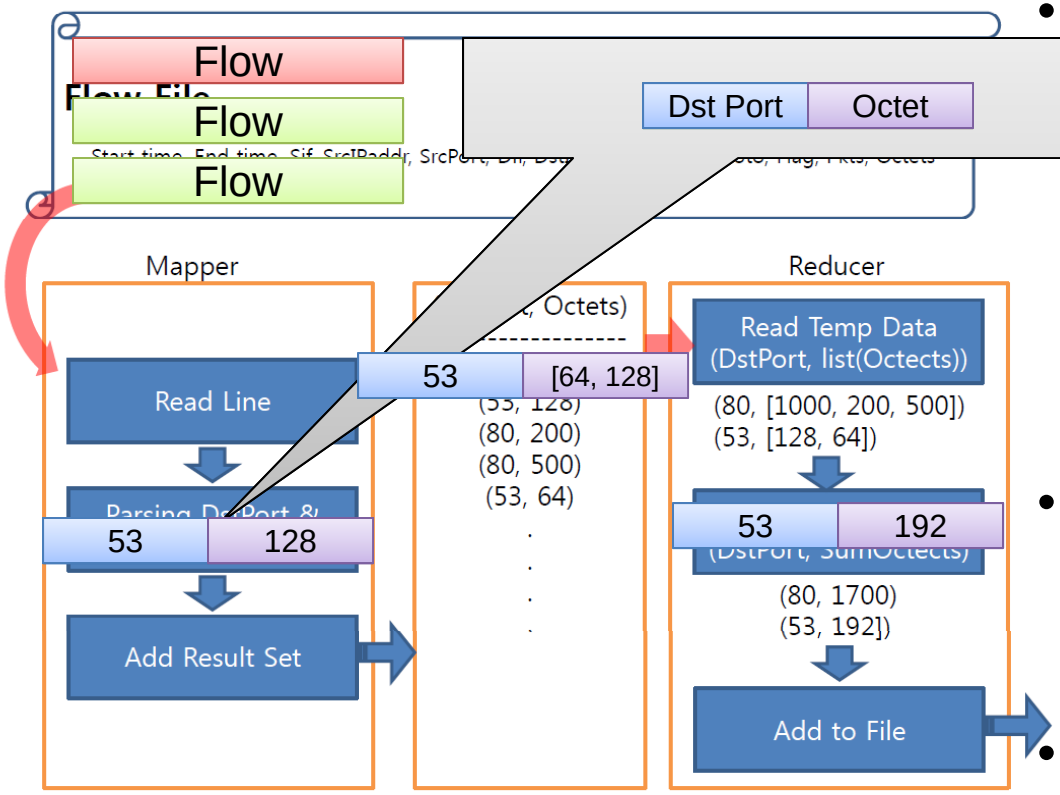
- Flow file input processor
- Flow analysis map/reduce
- Flow-tools
- Hadoop
  - HDFS
  - MapReduce
- Java VM
- OS : Linux

Flow File Input Processor

- Save NetFlow data in binary flow file
- Convert binary flow file into text file
- Copy text file to HDFS



Flow Analysis Map/Reduce



- Read text flow files
- Run map tasks
  - Read each line (Validation Check)
  - Parsing flow data
  - Save result into temporary files (key, value)
- Run reduce tasks
  - Read temporary files (Key, List[Value])
  - Run sum process
- Write results to a file

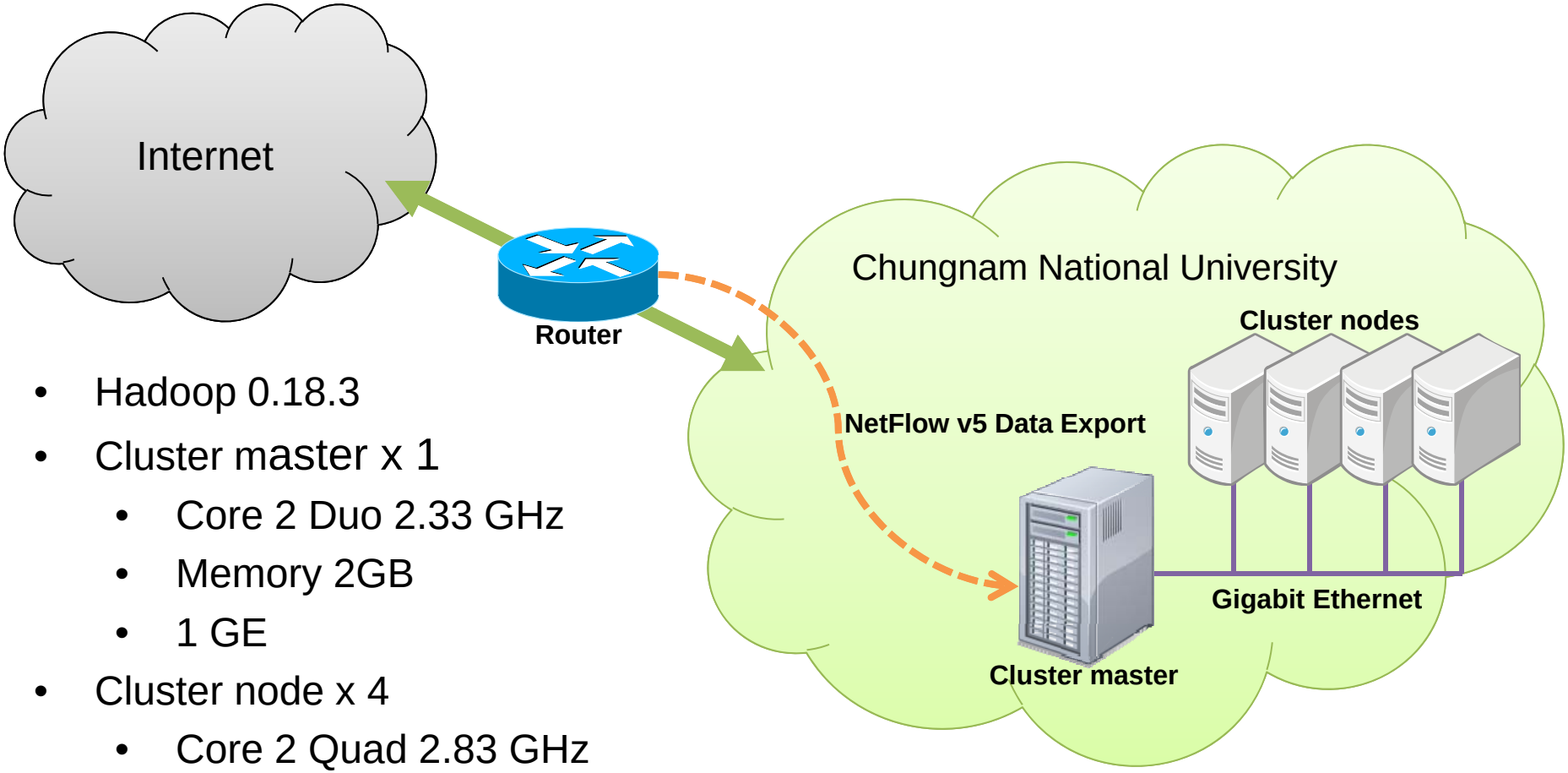
Performance Evaluation Environment

- Data: flow data from /24 subnet

Duration	Flow count (million)	Flow file count	Total binary file size (GB)	Total text file size (GB)
1 day	3.2	228	0.2	1.2
1 week	19.0	1596	0.3	2.3
1 month	109.1	7068	2.0	13.1

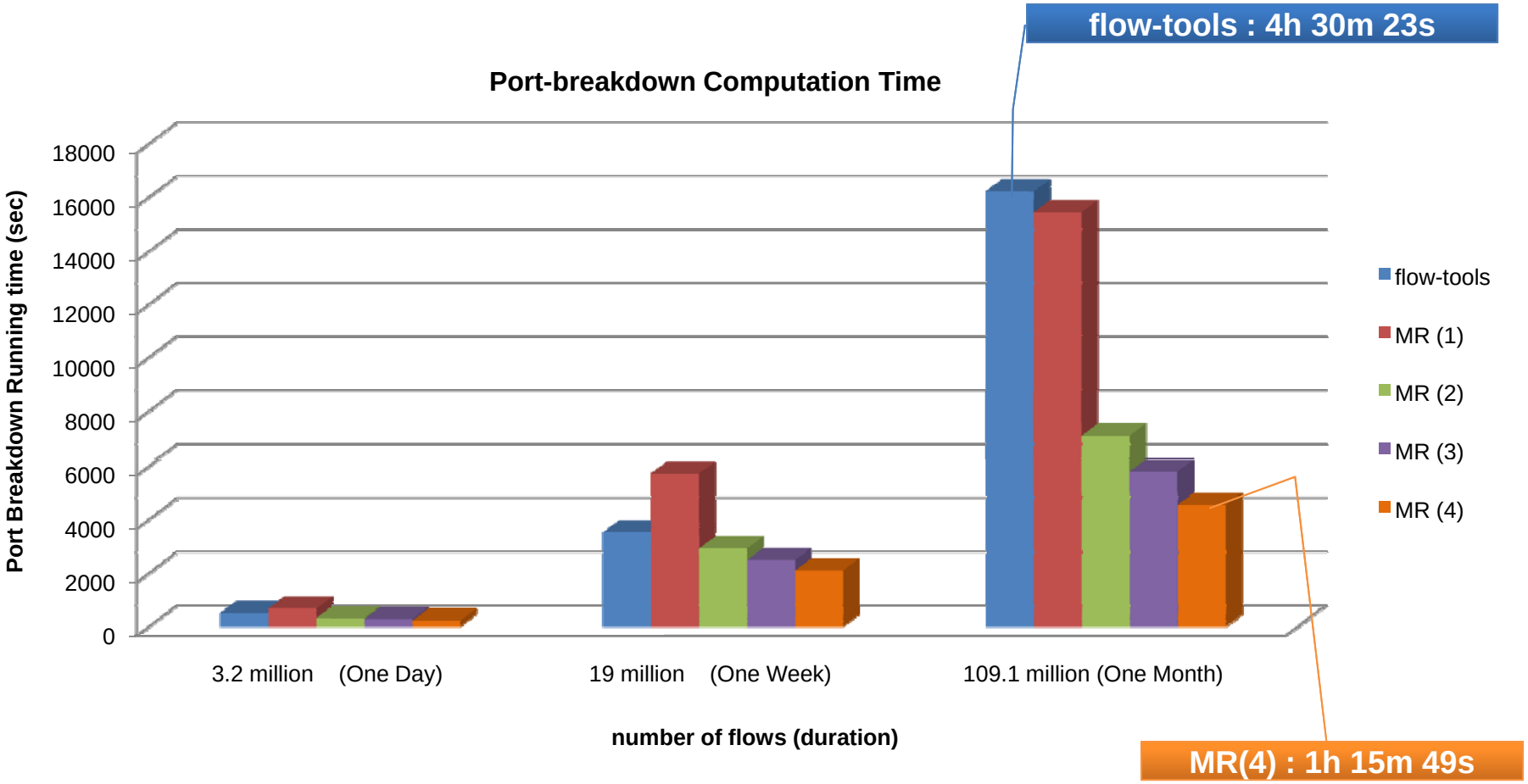
- Compared methods : computing byte count per destination port
  - flow-tools : flow-cat [flow data folder] | flow-stat –f 5
  - Our implementation with Hadoop
- Performance metric
  - flow statistics computation time
- Fault recovery against map/reduce tasks

Our Testbed



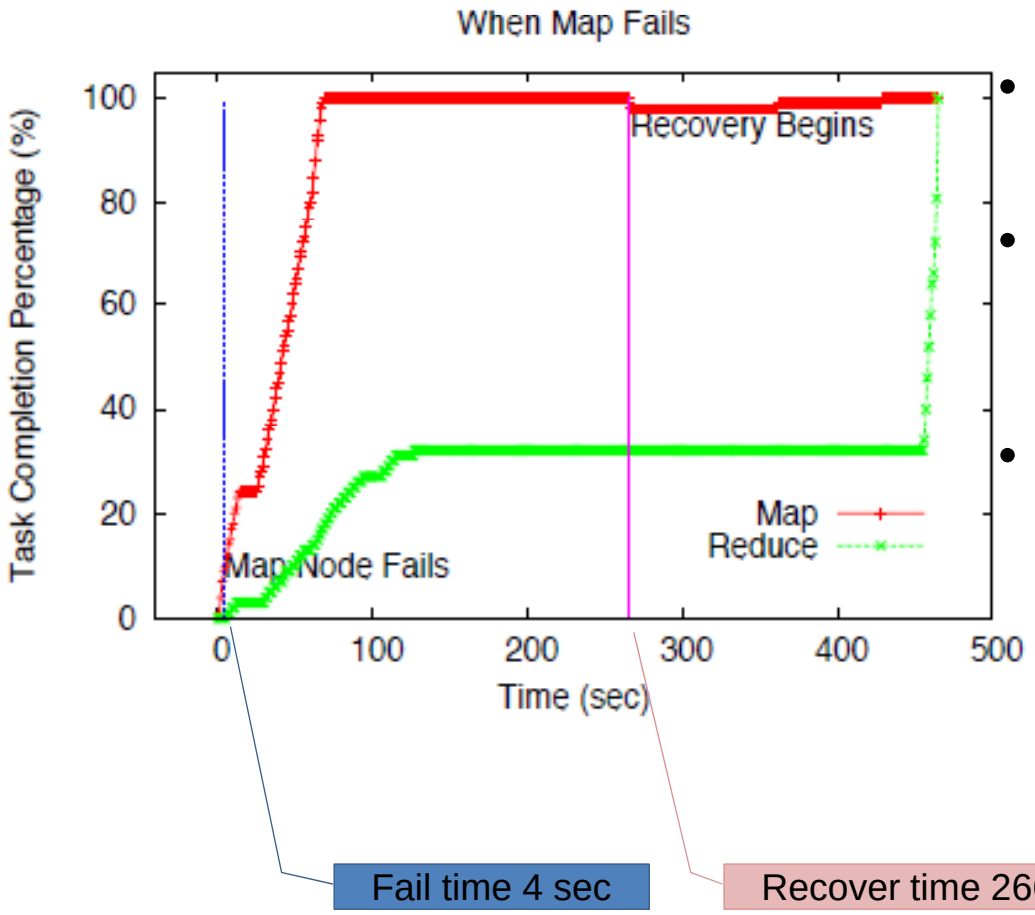
- Hadoop 0.18.3
- Cluster master x 1
  - Core 2 Duo 2.33 GHz
  - Memory 2GB
  - 1 GE
- Cluster node x 4
  - Core 2 Quad 2.83 GHz
  - Memory 4GB
  - HDD 1.5 TB
  - 1 GE

Flow Statistics Computation Time



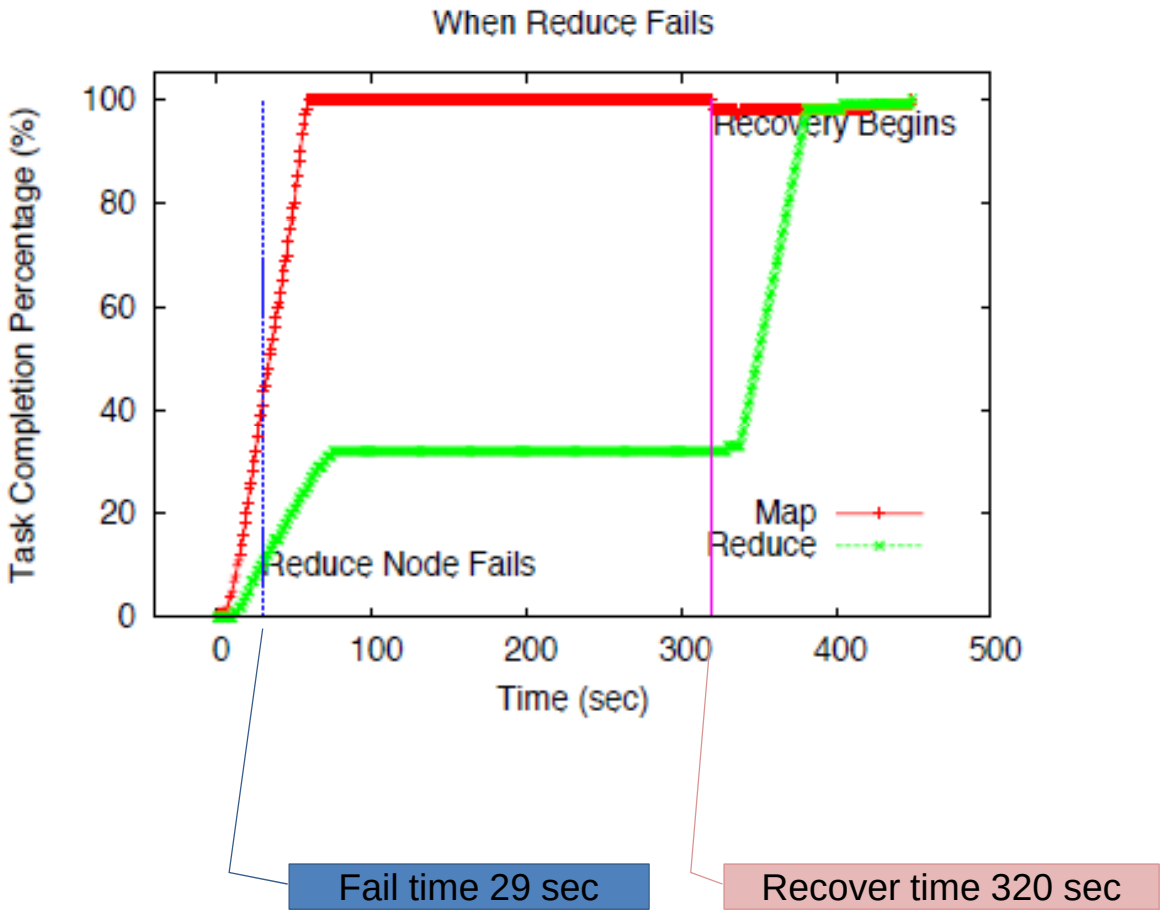
- Port breakdown computation time
  - 72% decrease with MR(4) on Hadoop

Single Node Failure : Map Task



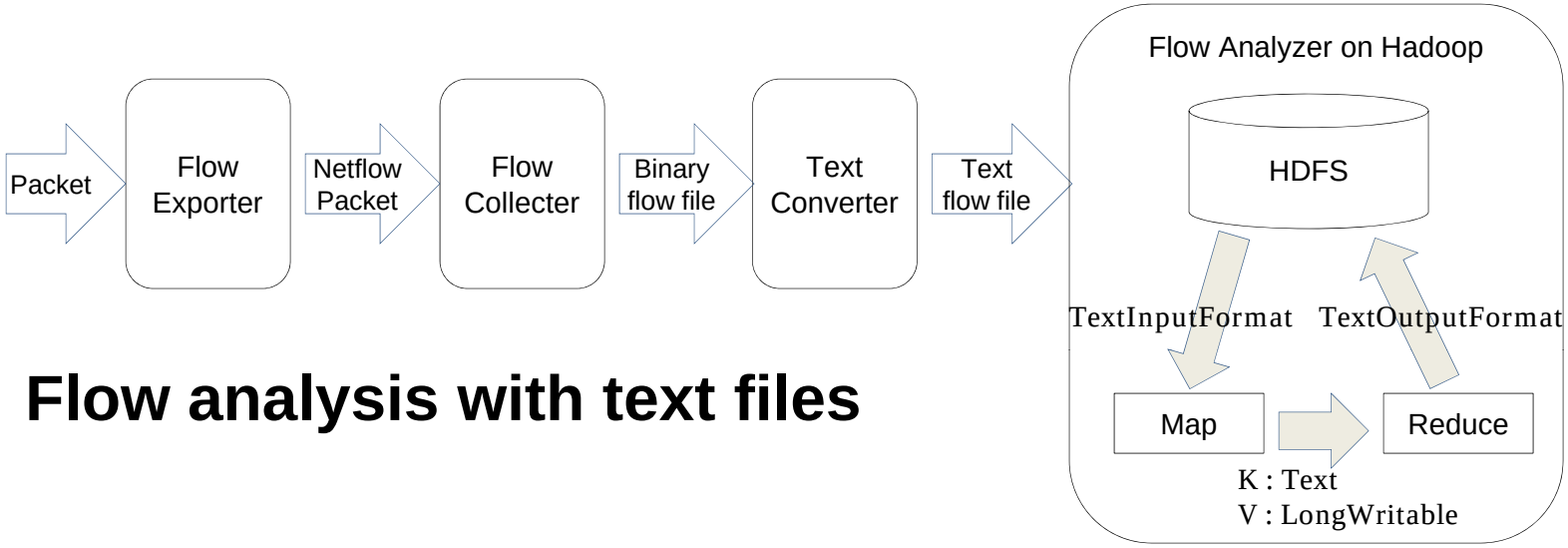
- Under 4 cluster nodes
- Map task fail time
  - 4 sec (M : 9% R : 0%)
- Map task recover time
  - 266 sec (M : 99% R : 32%)

Single Node Failure : Reduce Task

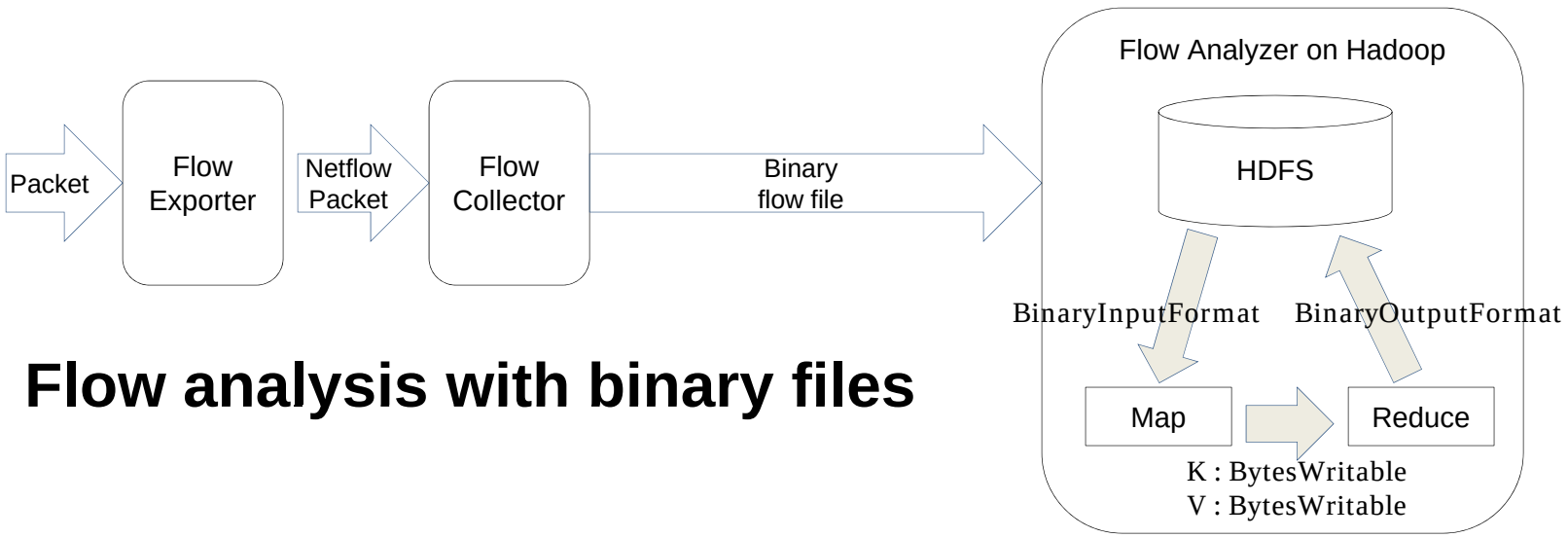


- Under 4 cluster nodes
- Reduce task fail time
  - 29 sec (M : 41% R : 10%)
- Reduce task recover time
  - 320 sec (M : 99% R : 32%)

Text vs. Binary NetFlow Files



Flow analysis with text files

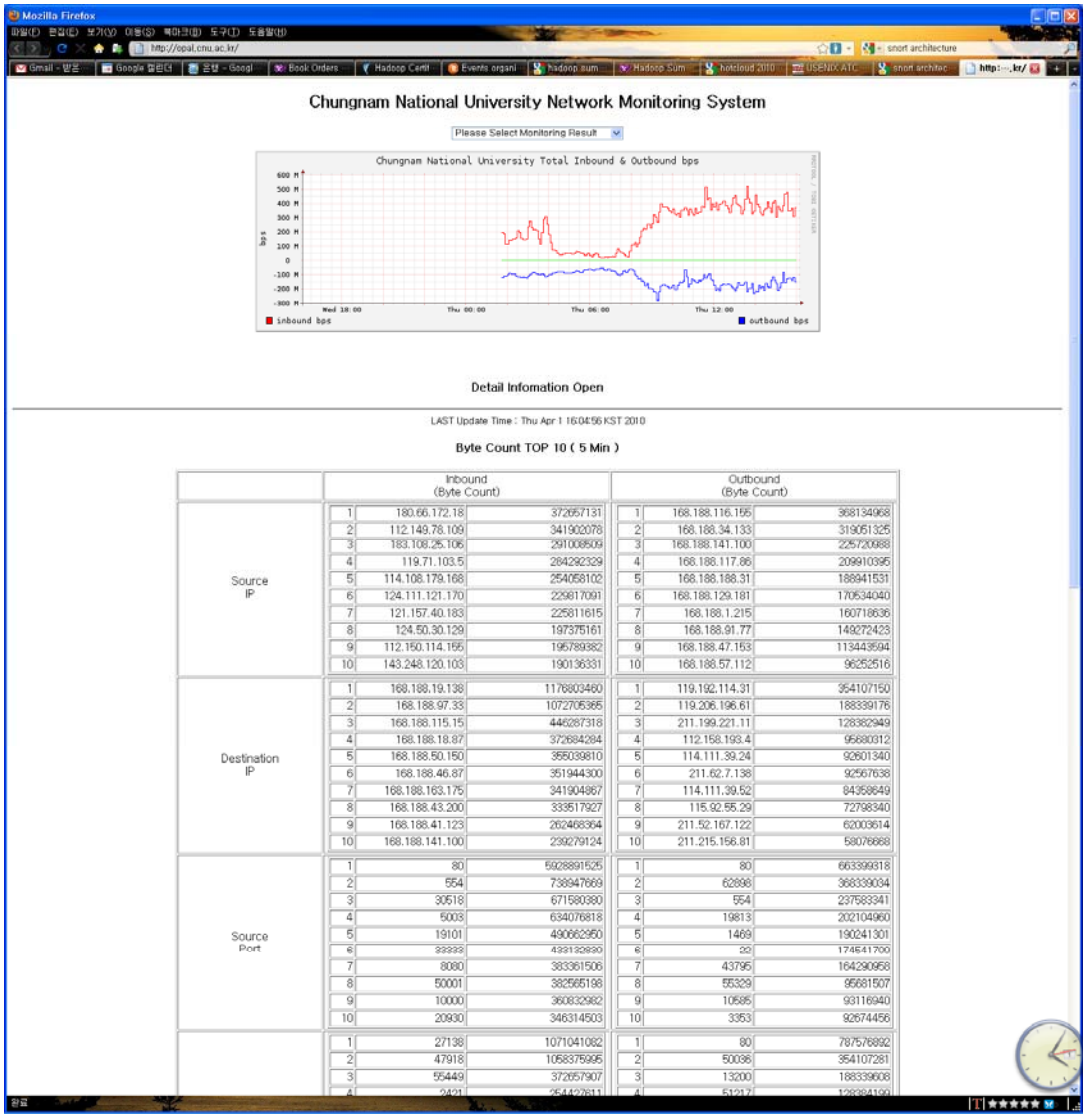


Flow analysis with binary files

Binary Input in Hadoop

- Currently developing BinaryInputFormat module for Hadoop
- Small storage by binary NetFlow files
  - Reduces # of Map tasks → increasing performance
- Decreasing computation time
  - By 18% ~ 55% for a single flow analysis job
  - By 58% ~ 75% for two flow analysis jobs

Prototype


Mozilla Firefox
Chungnam National University Network Monitoring System
Please Select Monitoring Result
Chungnam National University Total Inbound & Outbound bps
inbound bps
outbound bps
Detail Infomation Open
LAST Update Time : Thu Apr 1 16:04:56 KST 2010
Byte Count TOP 10 (5 Min)
Inbound (Byte Count)
Outbound (Byte Count)
Source IP
Destination IP
Source Port

Summary

- NetFlow data analysis with MapReduce
  - Easy management of big flow data
  - Decreasing computation time
  - Fault-tolerant service against a single machine failure
- Ongoing work
  - Supporting binary NetFlow files
  - Enhancing fast processing of NetFlow files

References

[1] Cisco NetFlow, http://www.cisco.com/web/go/netflow.

[2] L. Deri, nProbe: an Open Source NetFlow Probe for Gigabit Networks, TERENA Networking Conference, May 2003.

[3] J. Quittek, T. Zseby, B. Claise, and S. Zander, Requirements for IP Flow Information Export (IPFIX), IETF RFC 3917, October 2004.

[4] tcpdump, http://www.tcpdump.org.

[5] CAIDA CoralReef Software Suite, http://www.caida.org/tools/measurement/coralreef.

[6] M. Fullmer and S. Romig, The OSU Flow-tools Package and Cisco NetFlow Logs, USENIX LISA, 2000.

[7] D. Plonka, FlowScan: a Network Traffic Flow Reporting and Visualizing Tool, USENIX Conference on System Administration, 2000.

[8] J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on Large Cluster, OSDI, 2004.

[9] Hadoop, http://hadoop.apache.org/.

[10] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices, ACM CoNEXT, 2008.

[11] C. Morariu, T. Kramis, B. Stiller DIPStorage: Distributed Architecture for Storage of IP Flow Records., 16thWorkshop on Local and Metropolitan Area Networks, September 2008.

[12] M. Roesch, Snort - Lightweight Intrusion Detection for Networks, USENIX LISA, 1999.

[13] W. Chen and J. Wang, Building a Cloud Computing Analysis System for Intrusion Detection System, CloudSlam 2009.

[14] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, Raghotham Murthy Hive: a warehousing solution over a map-reduce framework., Proceedings of the VLDB Endowment Volume 2 , Issue 2 (August 2009) Pages: 1626-1629

[15] HBase, http://hadoop.apache.org/hbase

[16] Wei-Yu Chen and Jazz Wang. Building a Cloud Computing Analysis System for Intrusion Detection System, CloudSlam'09