A Multilevel Secure MapReduce Framework for Cross-Domain Information Sharing in the Cloud

Thuy D. Nguyen, Cynthia E. Irvine, Jean Khosalim
Department of Computer Science

Ground System Architectures Workshop
March 18-21, 2013

Published by The Aerospace Corporation with permission.

Introduction

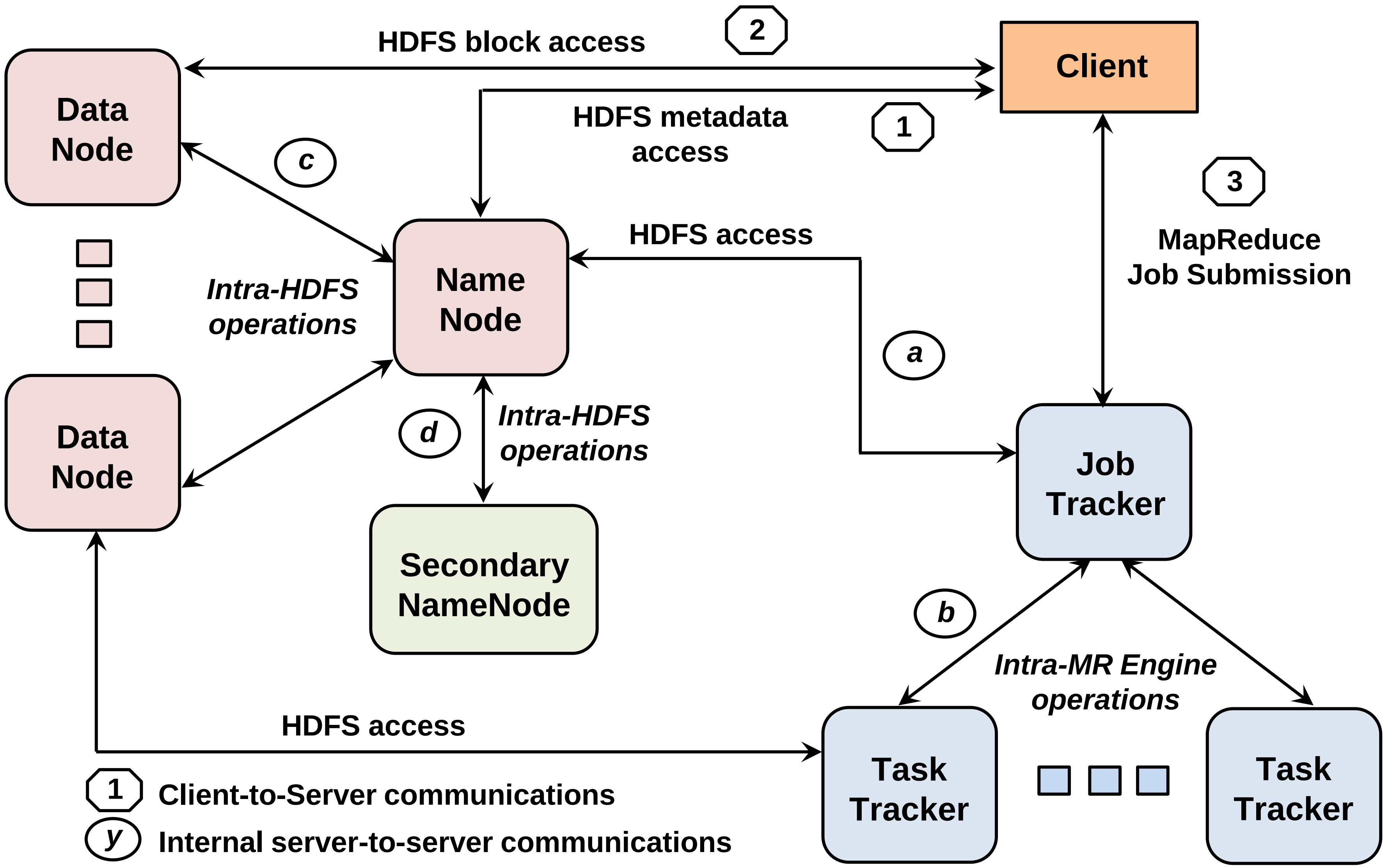
- Motivation
  - Develop a cross-domain MapReduce framework for a multilevel secure (MLS) cloud, allowing users to analyze data at different security classifications

- Topics
  - Apache Hadoop framework
  - MLS-aware Hadoop Distributed File System
    - Concept of operations
    - Requirements, design, implementation
  - Future work and conclusion

Apache Hadoop

- Open source software framework for reliable, scalable, distributed computing
- Inspired by Google's MapReduce computational paradigm and Google File System (GFS)
- Two main subprojects:
  - Hadoop Distributed File System, Hadoop MapReduce
- Support distributed computing on massive data sets on clusters of commodity computers
- Common usage patterns
  - ETL (Extract → Transform → Load) replacement
  - Data analytics, machine learning
  - Parallel processing platforms (Map without Reduce)

Hadoop Architecture



MLS-Aware: A Definition

A component is considered ***MLS-aware*** if it executes without privileges in an MLS environment, and yet takes advantage of that environment to provide useful functionality.

Examples:

- Reading from resources labeled at the same or lower security levels
- Making access decisions based on the security level of the data
- Returning the security level of the data

Objective and Approach

- Objective
  - Extend Hadoop to provide a cross-domain read-down capability without requiring the Hadoop server components to be trustworthy
- Approach
  - Modify Hadoop to run on a trusted platform that enforces an MLS policy on local file system
    - Use Security Enhanced Linux (SELinux) for initial prototype
  - Modify HDFS to be MLS-aware
    - Multiple single-level HDFS instances – each is cognizant of HDFS namespaces at lower security levels
    - HDFS servers running at a security level can access file objects at lower levels as permitted by underlying trusted computing base (TCB)
  - No trusted processes outside TCB boundary

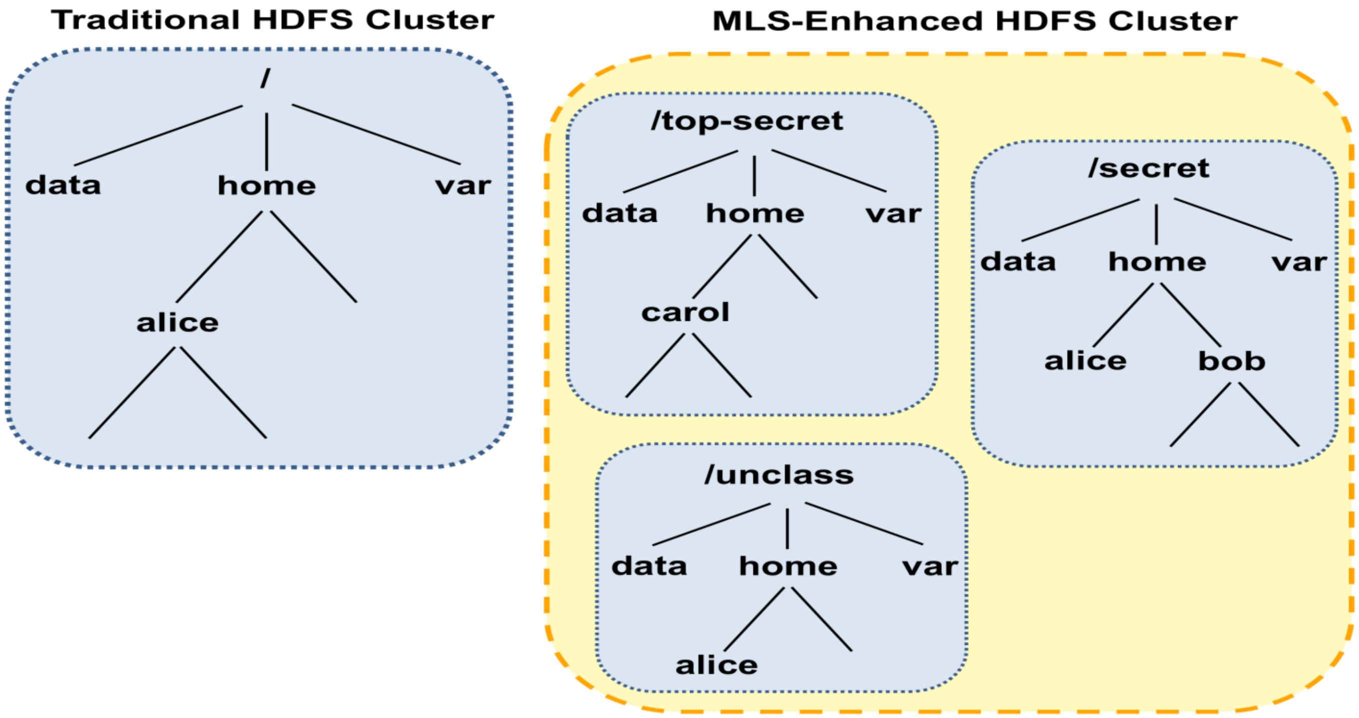
HDFS Concept of Operation

- User session level
  - Implicitly established by security level of receiving network interface and TCP/IP ports
- File access policy rules
  - A user can read and write file objects at user's session level
  - A user can read file objects if the user's session level dominates the level of the requested object
- File system abstraction
  - HDFS interface is similar to UNIX file system
  - Traditional Hadoop cluster: one file system
  - MLS-enhanced cluster: multiple file systems, one per security level

HDFS File Organization

- Root directory at a particular level is expressed as

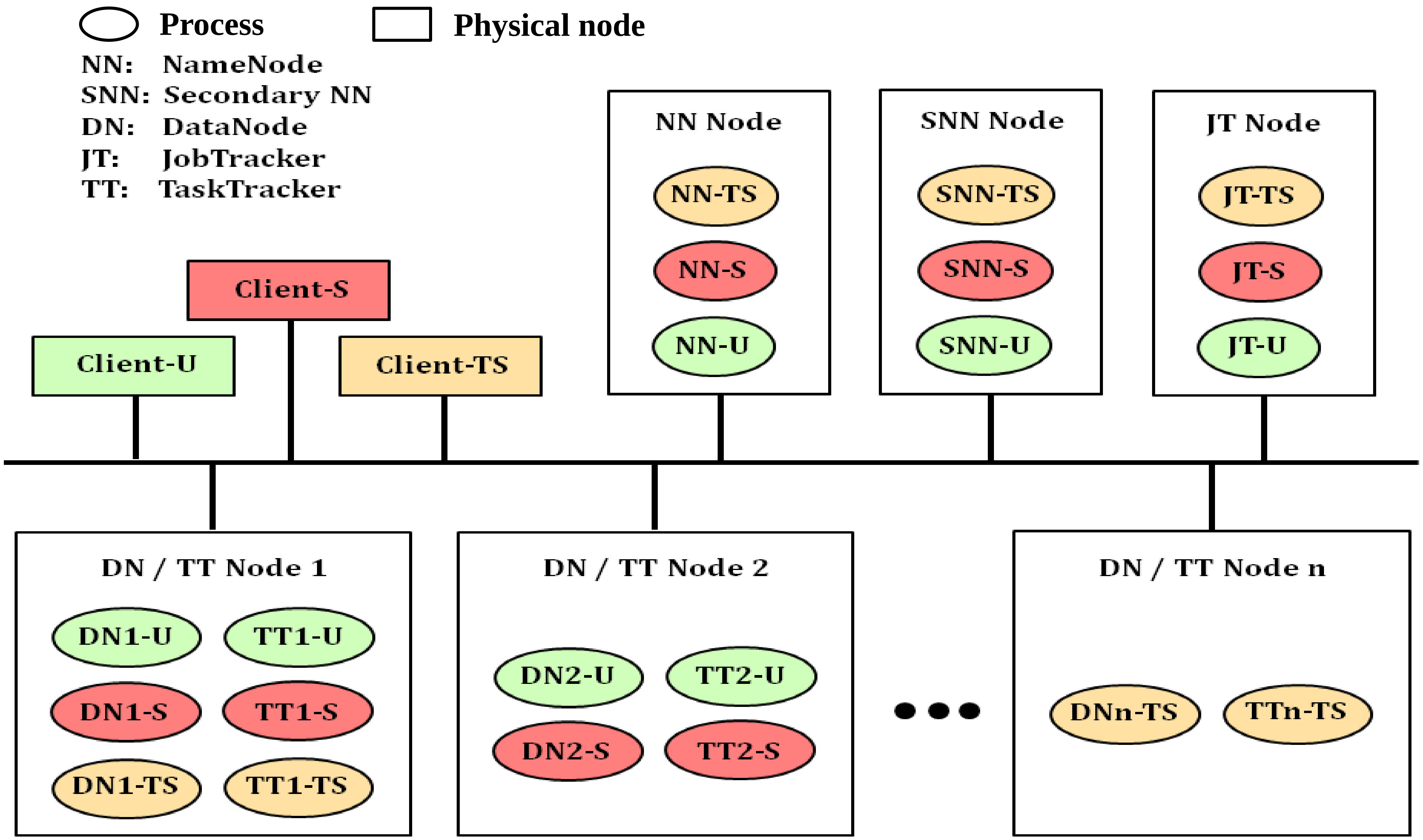
  /<user-defined ***security-level-indicator***>
- Security-level-indicator is administratively assigned to an SELinux sensitivity level
- Traditional root directory (/) is root at the user's session level



MLS-aware Hadoop Design

- Multiple single-level HDFS server instances co-locate on same physical node
- All NameNode instances run on same physical node
- DataNode instances are distributed across multiple physical nodes
  - *Authoritative DataNode* instance: owner of local files used to store HDFS blocks
  - *Surrogate DataNode* instance: handles read-down requests on behalf of an authoritative DataNode instance running at a lower level
- Configuration file defines allocation of authoritative and surrogate DataNode instances on different physical nodes
- Design does not impact MapReduce subsystem
  - JobTracker and TaskTracker only interact with NameNode and DataNode as HDFS clients

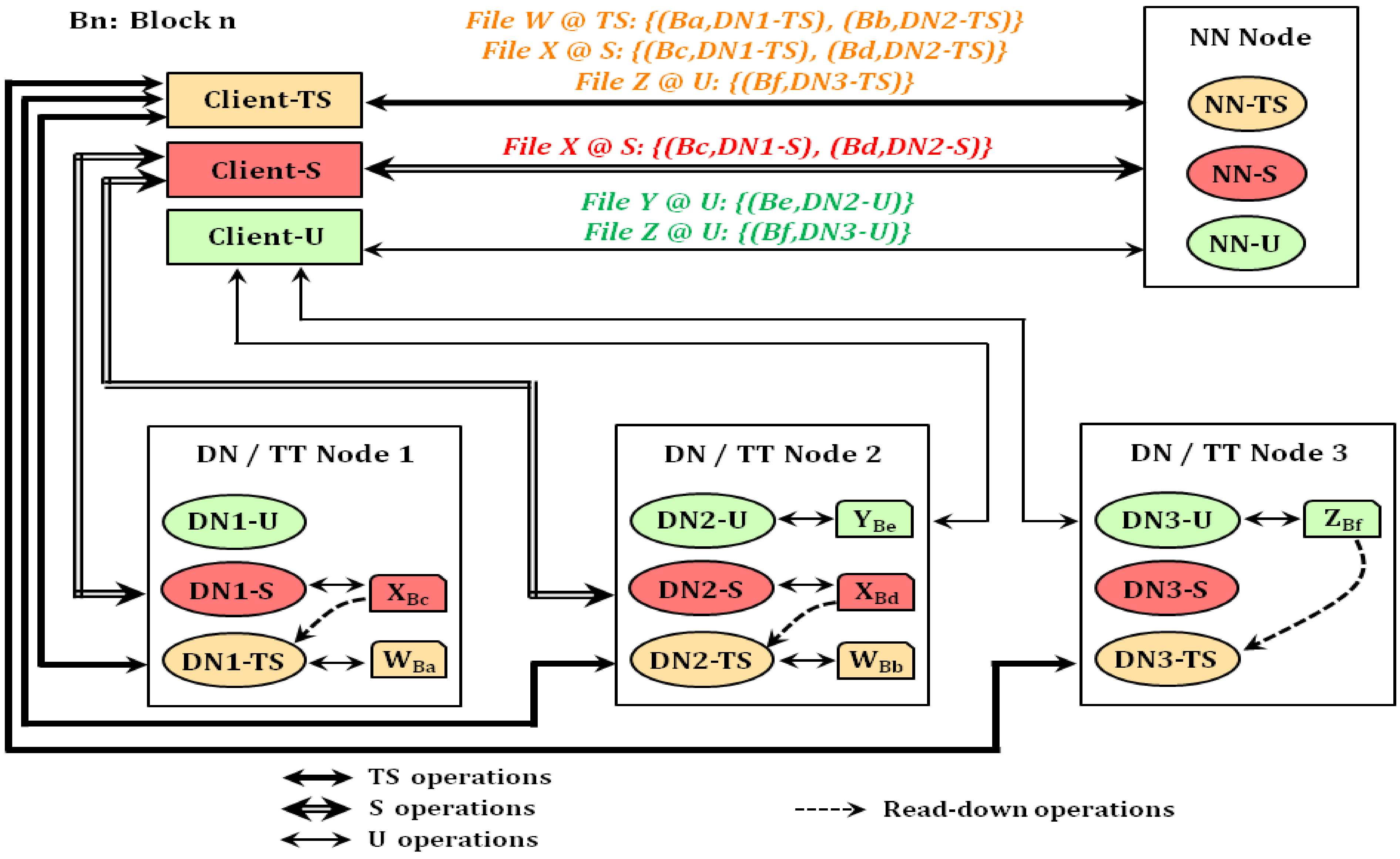
MLS-enhanced Hadoop Cluster



Cross-domain Read-down

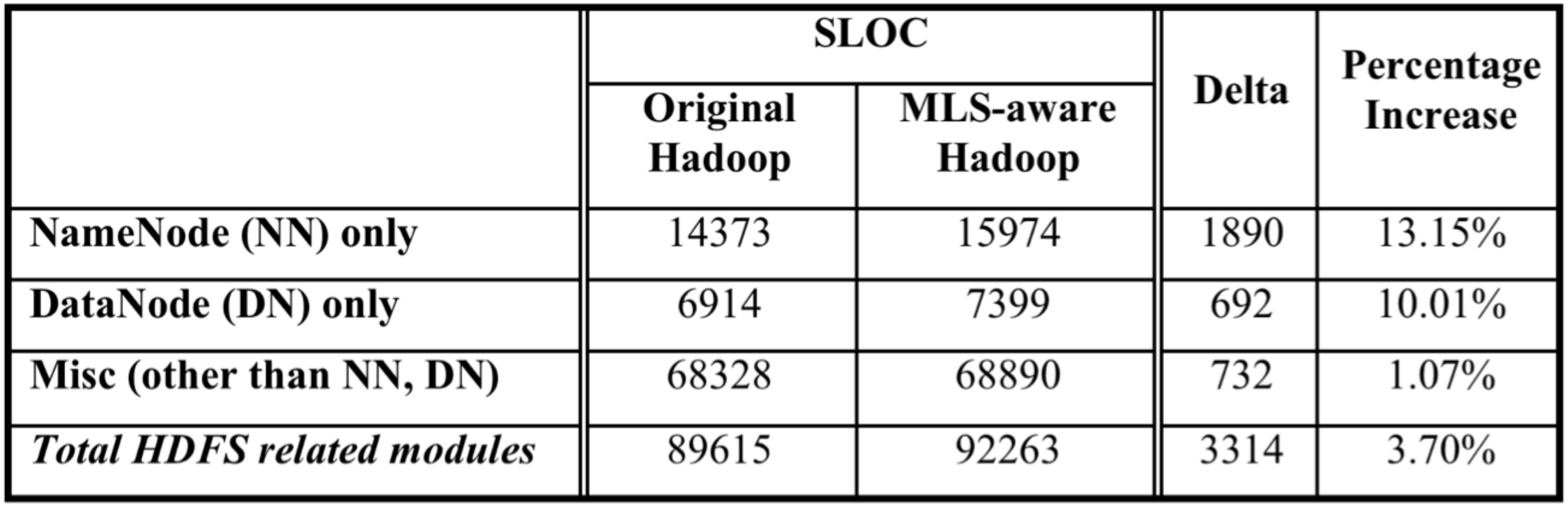
- Client running at user's session level
  - Contact NameNode at same level to request a file at a lower level
- NameNode instance at session level
  - Obtain metadata of requested file and storage locations of associated blocks from NameNode instance running at lower security level
  - Direct client to contact surrogate DataNode instances that co-locate with the file's primary DataNode instances
- Surrogate DataNode instance at session level
  - Look up locations of local files used to store requested blocks
  - Read local files and return requested blocks
    - Security level of local files is lower than session level

Read-down Example



Source Lines of Code (SLOC) Metric

- Use open source Count Lines of Code (CLOC) tool
  - Can calculate differences in blank, comment, and source lines
- Summary of code modification
  - Delta value is the sum of addition, removal, and modification of source lines
  - Overall change is less than 5%



	SLOC		Delta	Percentage Increase
	Original Hadoop	**MLS-aware Hadoop**		
NameNode (NN) only	14373	15974	1890	13.15%
DataNode (DN) only	6914	7399	692	10.01%
Misc (other than NN, DN)	68328	68890	732	1.07%
Total HDFS related modules	89615	92263	3314	3.70%

Future Work and Conclusions

- Future work
  - Adding read-down support to HDFS Federation
  - Implementing an external Cache Manager
  - Investigating Hadoop’s use of Kerberos for establishing user sessions at different security levels
  - Performing benchmark testing with larger datasets
- Prototype is the first step towards developing a highly secure MapReduce platform
  - Does not introduce any trusted processes outside the pre-existing TCB boundary
  - Only affects HDFS servers

Contact

Cynthia E. Irvine, PhD, irvine@nps.edu
Thuy D. Nguyen, tdnguyen@nps.edu

Center for Information Systems Security Studies and Research
Department of Computer Science
Naval Postgraduate School, Monterey, CA 93943 U.S.A

http://www.cisr.us