Empirical Study on File Systems and its influence on HDFS Performance

04/04/2019

Presented by: Sushant Raikar, Tuhin Tiwari

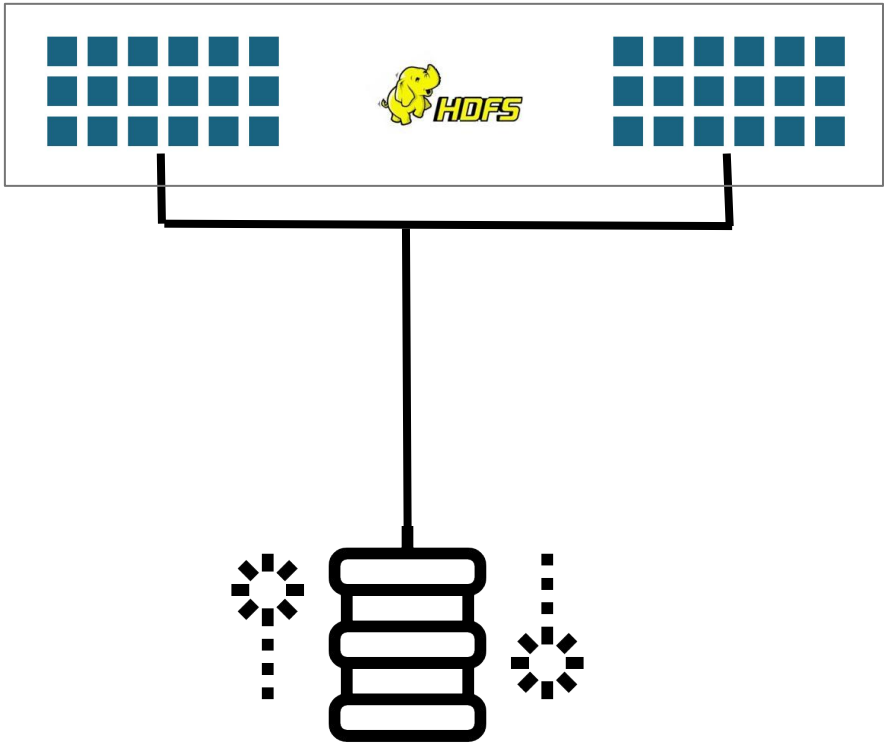
Department of Mathematics, Computer Science faculty

UNIVERSITY OF WATERLOO

Agenda

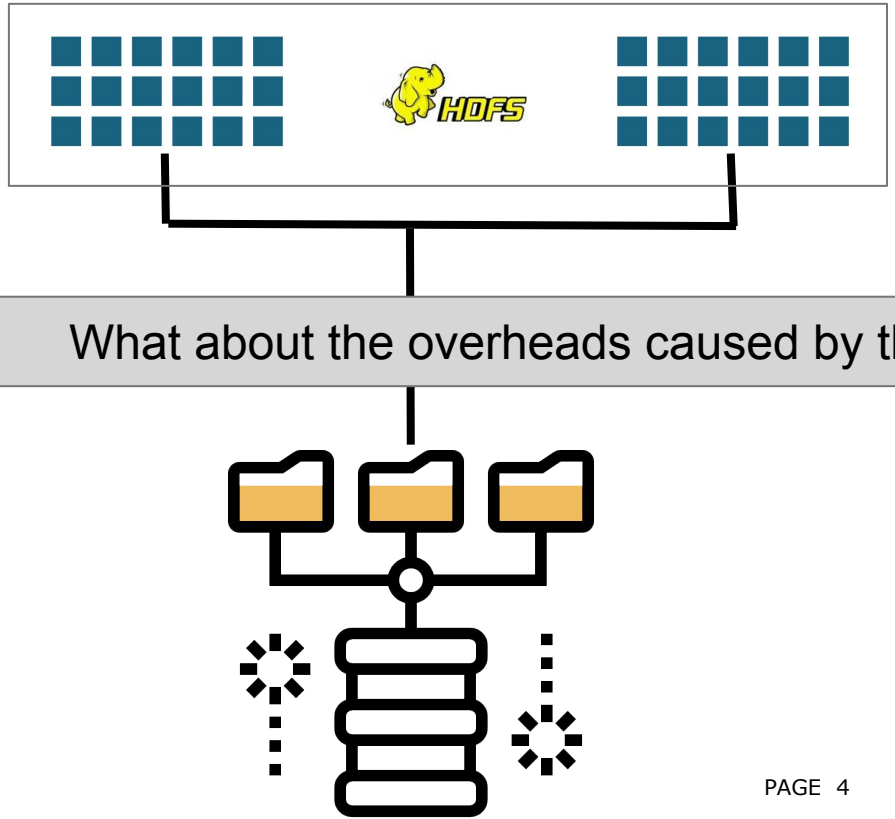
1. Problem Statement
2. Motivation
3. What has been done?
4. Background
5. Experimental Setup
6. Evaluation
7. Interesting Finding
8. Future Work

Problem Statement



- HDFS - Hadoop's File System that enables highly efficient parallel processing.
- **Data Chunking**

Problem Statement



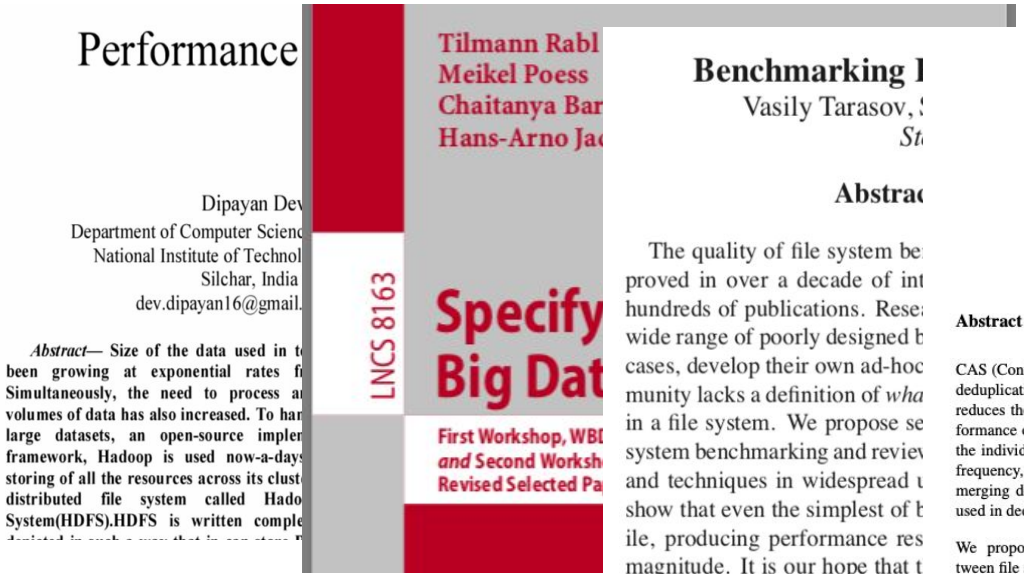
- HDFS - Hadoop's File System that enables highly efficient parallel processing.

What about the overheads caused by the underlying OS File system?

Motivation

We care about these underlying File systems and wanted to test how they behave differently when given different workload and also speculate the possible reason of their behavior.

What has been done?



Performance

Dipayan Dev
Department of Computer Scienc
National Institute of Technol
Silchar, India
dev.dipayan16@gmail.

Abstract— Size of the data used in t
been growing at exponential rates f
Simultaneously, the need to process a
volumes of data has also increased. To har
large datasets, an open-source implen
framework, Hadoop is used now-a-days
storing of all the resources across its clust
distributed file system called Hado
System(HDFS).HDFS is written comple

Tilmann Rabl
Meikel Poess
Chaitanya Bar
Hans-Arno Jac

LNCS 8163

Specify
Big Dat

First Workshop, WBD
and Second Worksh
Revised Selected Pa

Benchmarking I

Vasily Tarasov, S
St

Abstrac

The quality of file system be
proved in over a decade of int
hundreds of publications. Rese
wide range of poorly designed b
cases, develop their own ad-hoc
munity lacks a definition of *wha*
in a file system. We propose se
system benchmarking and reviev
and techniques in widespread u
show that even the simplest of b
ile, producing performance res
magnitude. It is our hope that t

Analysis of Disk Access Patterns on File Systems for Content Addressable Storage

Kuniyasu Suzaki, Kengo Iijima, Toshiki Yagi, and Cyrille Artho
National Institute of Advanced Industrial Science and Technology
{ k.suzaki | k-iijima | yagi-toshiki | c.artho } @aist.go.jp

Abstract

CAS (Content Addressable Storage) is virtual disk with deduplication, which merges same-content chunks and reduces the consumption of physical storage. The performance of CAS depends on the allocation strategy of the individual file system and its access patterns (size, frequency, and locality of reference) since the effect of merging depends on the size of a chunk (access unit) used in deduplication.

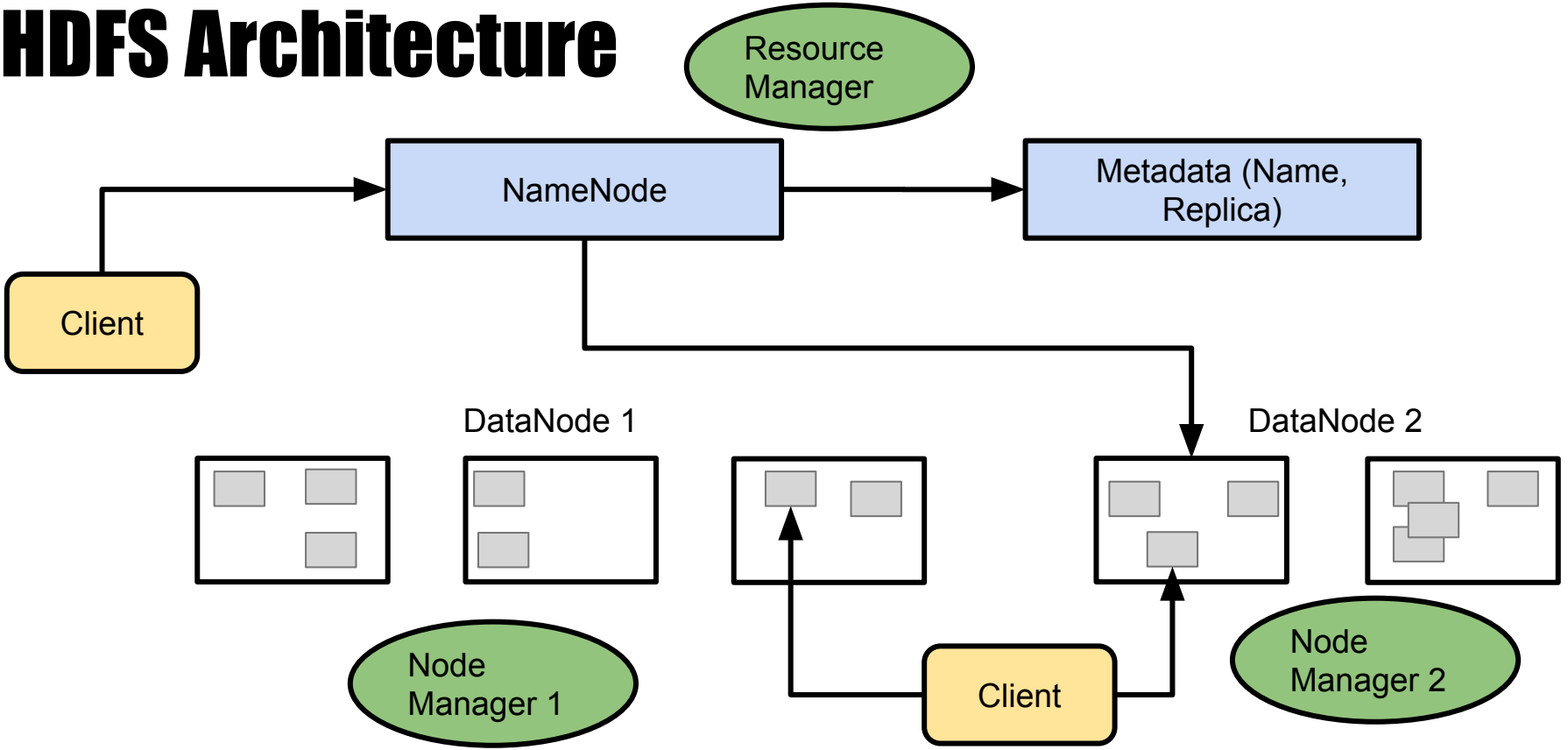
We propose a method to evaluate the affinity between file system and CAS, which compares the degree

CAS provides a universal virtual block device and accepts any file system on it. The performance depends on data allocations and their access patterns through the file system, because each file system has techniques to optimize space usage and I/O performance. The optimizations include data alignment, contiguous allocation, disk prefetching, lazy evaluation, and so on. These factors make the file system a key factor for the performance of CAS.

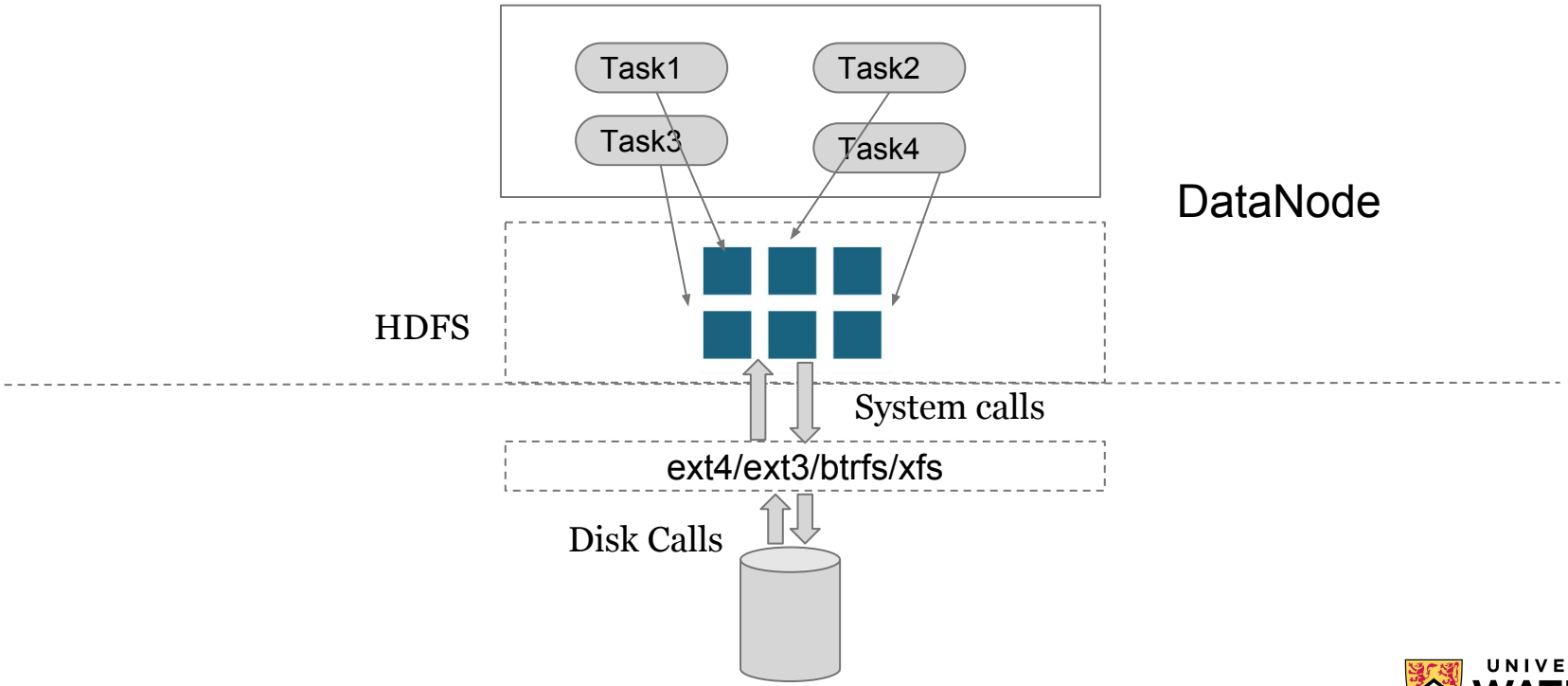
From the view of the disk, a file system works as a "filter" to allocate data. Even if the same contents are saved, access patterns differ between file systems. Espe-

UNIVERSITY OF WATERLOO

HDFS Architecture



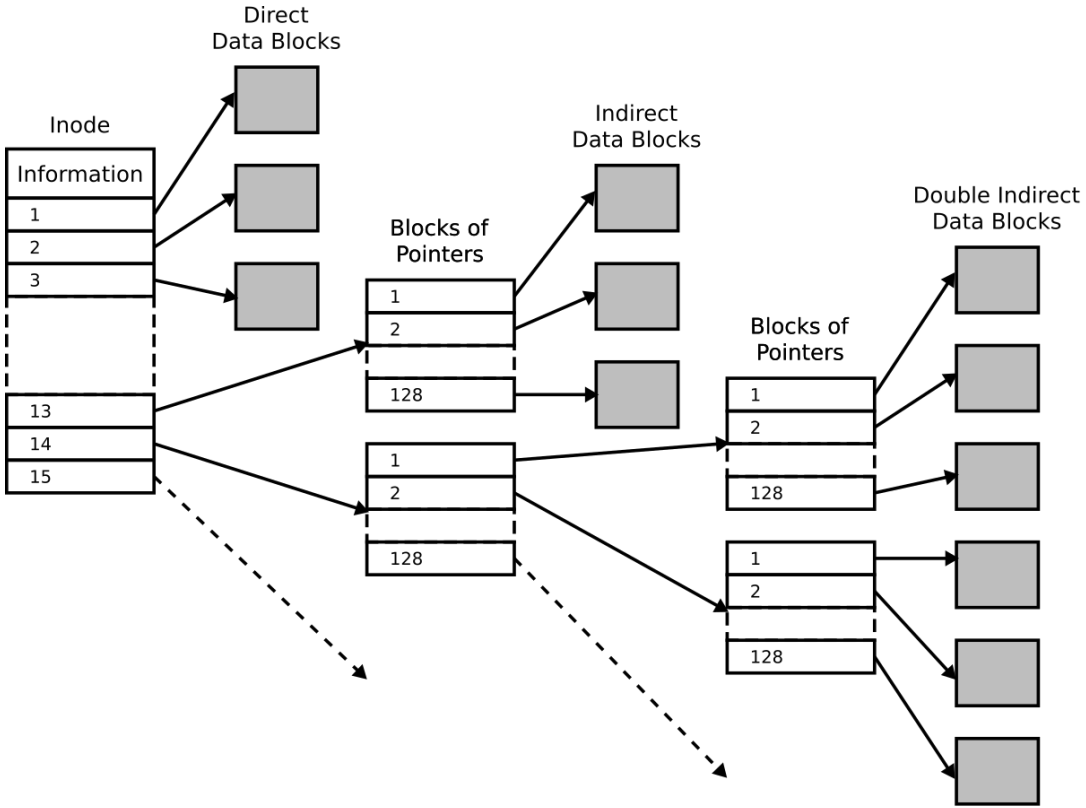
Interaction between HDFS and Linux file system



Linux File Systems

- EXT3 - Third extended filesystem used by Linux kernel
- EXT4 - Fourth extended filesystem. Default used by Hadoop
- BtrFS - Butter FS is a modern CoW(Copy on Write) filesystem, initially implemented by Oracle.

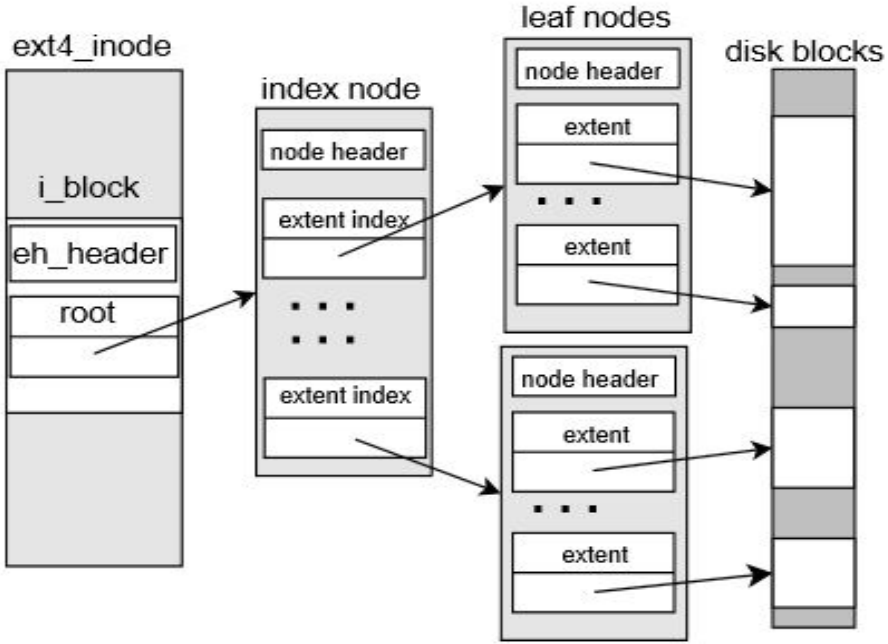
Directory Structure of EXT3



Directory Structure of EXT3

- More fragmented data
- Issues with large files because of multiple pointers
- Traversing through the structure to find a filename takes longer

Directory Structure of EXT4

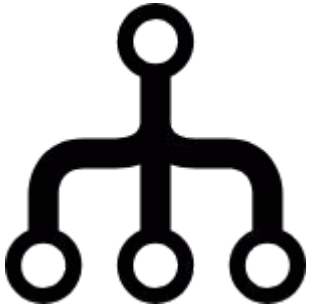


Directory Structure of EXT4

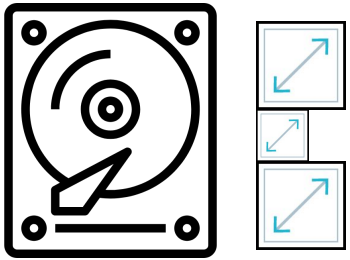
- Fragmentation is prevented by extents and delayed block allocation
- Saves the extent tree as a hashed B-Tree after the number of extents become more than 3
- Deletion of files does not take as long as ext3
- Due to hash structure, fast access to a file name by computing hash code.

Directory Structure of BtrFS

Copy-on-write

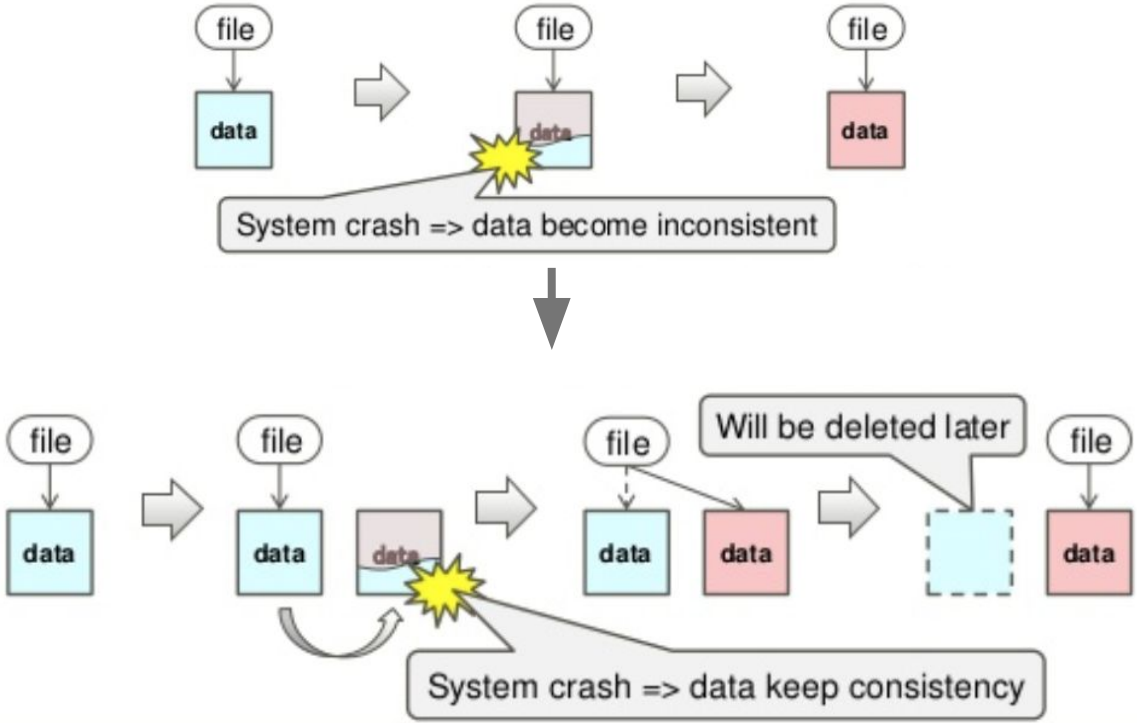


BTree



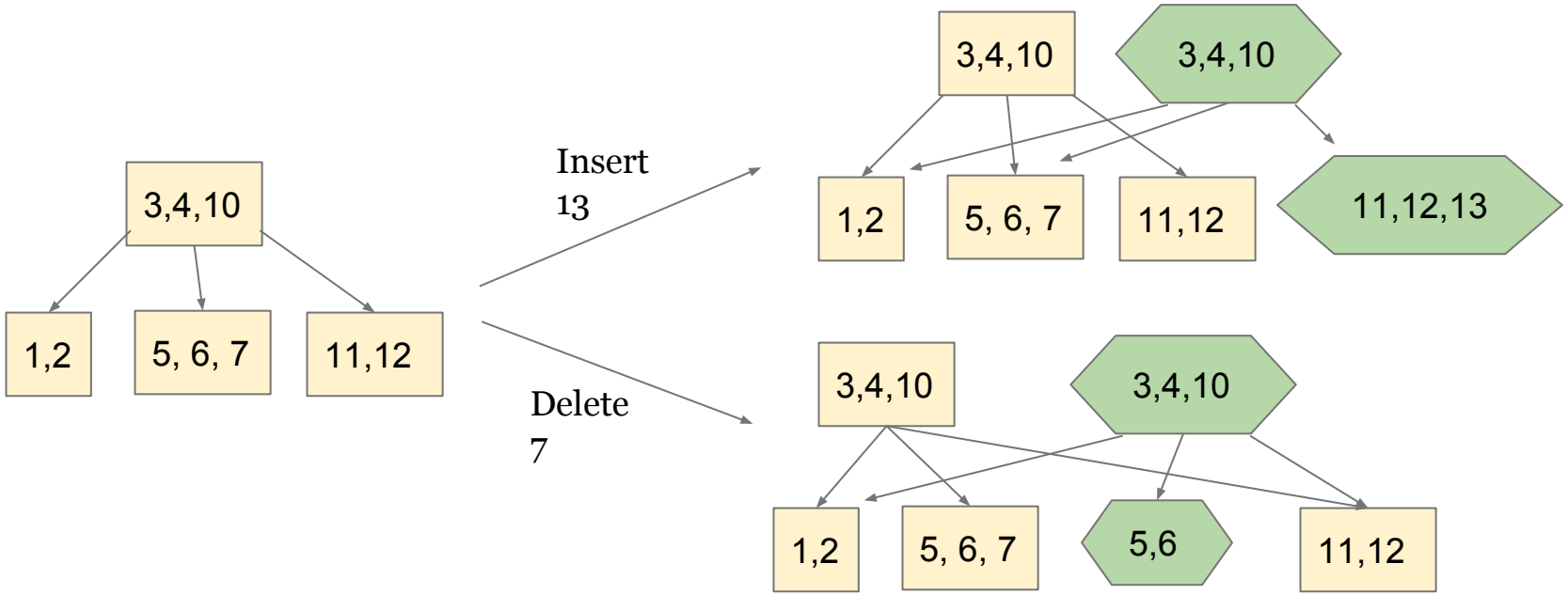
Extents

BtrFS - Copy on Write

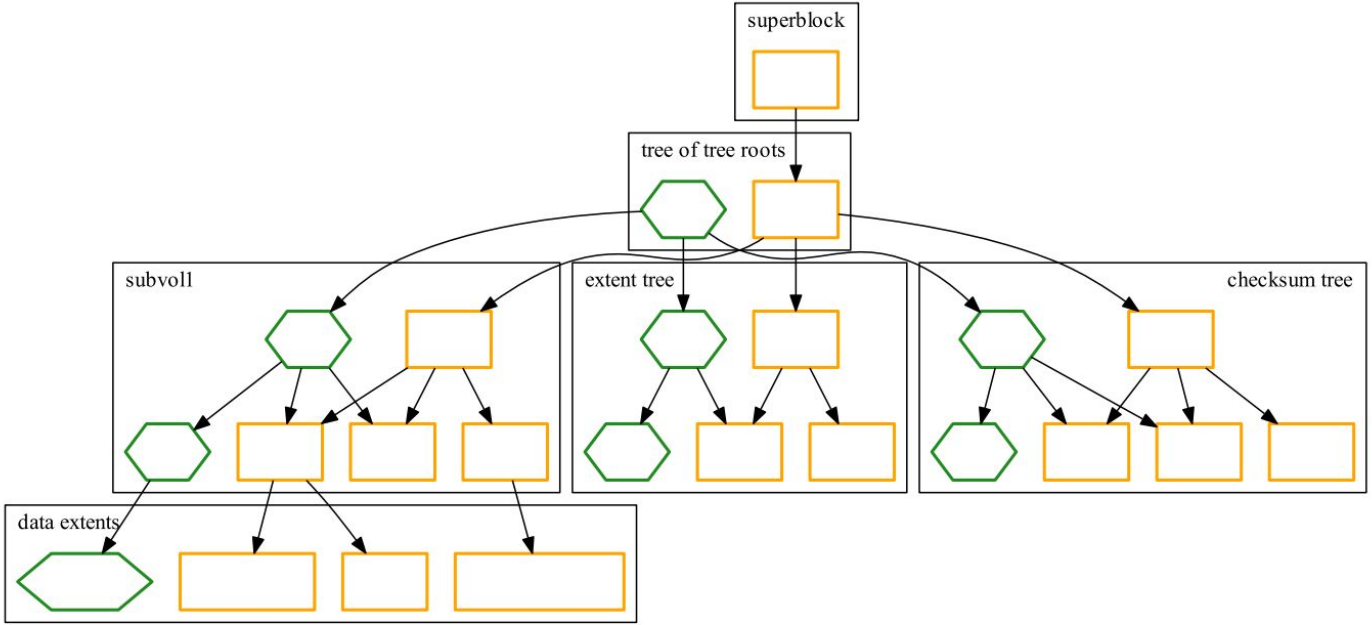


UNIVERSITY OF WATERLOO

BtrFS - COW Btrees



BtrFS - COW Btrees



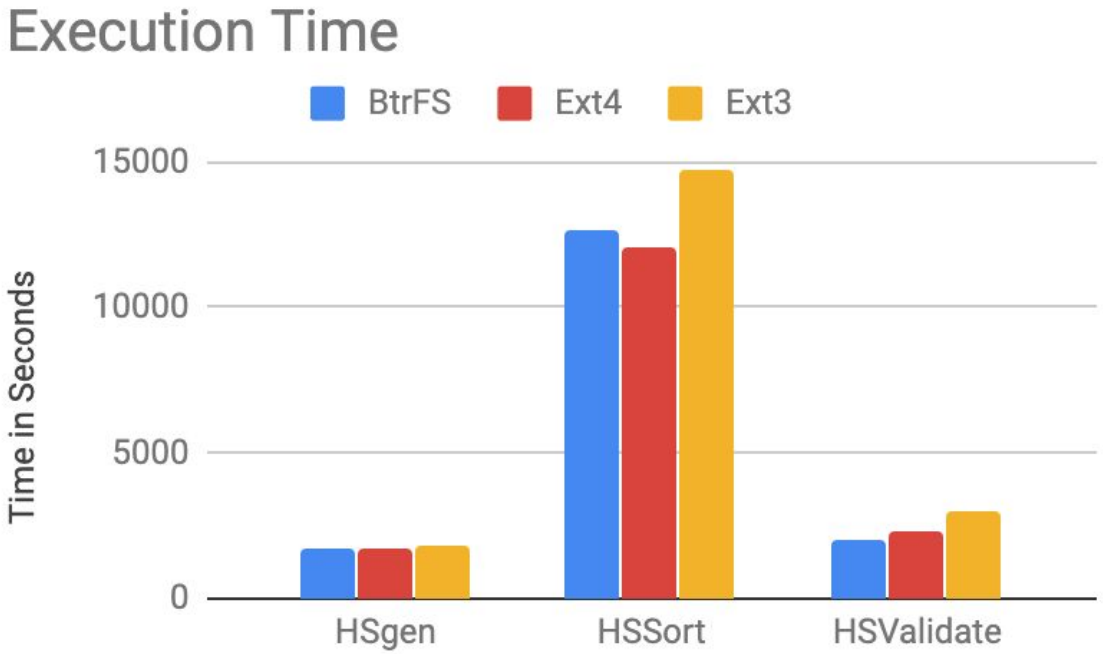
Experimental Setup

- 2 hadoop clusters with 1 NameNode and 2 DataNodes each
- File systems mounted on 512GB HDD partition
- On each cluster, generated 100GB data with two benchmarks: TPC-H & TPC-HS
- TPC-H is run on Apache Hive

RQ1: How do different Filesystems affect Hadoop applications?

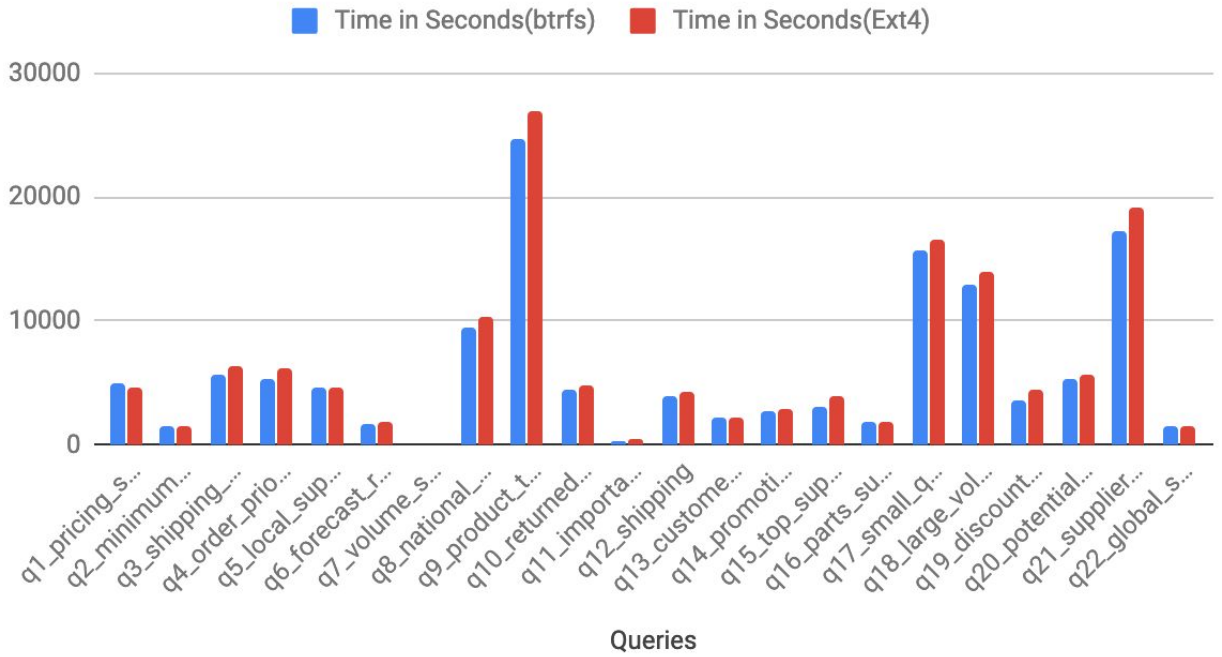
- Two application based benchmarks
    - TPC-H
        - Database queries
        - Tests sequential I/O throughput
    - TPC-HS
        - Sorts Big Data
        - Benchmark meant for Big Data Hadoop

Results - TPC-HS



- Btrfs performs better for Gen and Validate.
- Ext4 performs better for Sort.
- Ext3 underperforms amongst the three.

Results - TPC-H



- Btrfs performs slightly better than ext4 in 90% of the queries
- Q9: complex join as well as aggregation inside the join.
- Q21: selecting from two different tables and checking if column exists or doesn't exist

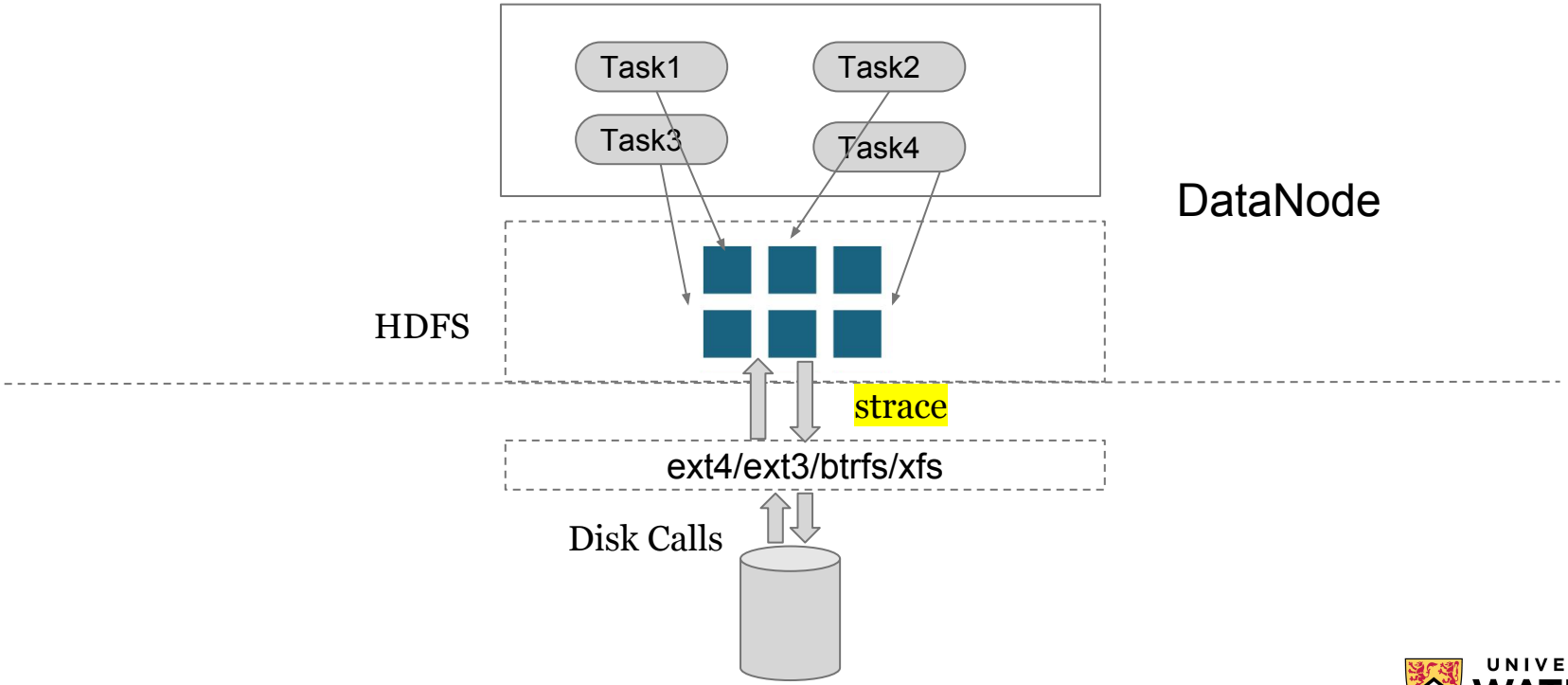
RQ2: How do different Filesystems perform with read-write operations on Hadoop?

- Microbenchmarking Hadoop with TestDFSIO
    - Map tasks to Read or Write data
    - Reduce tasks to collect statistics

Results - DFSIO

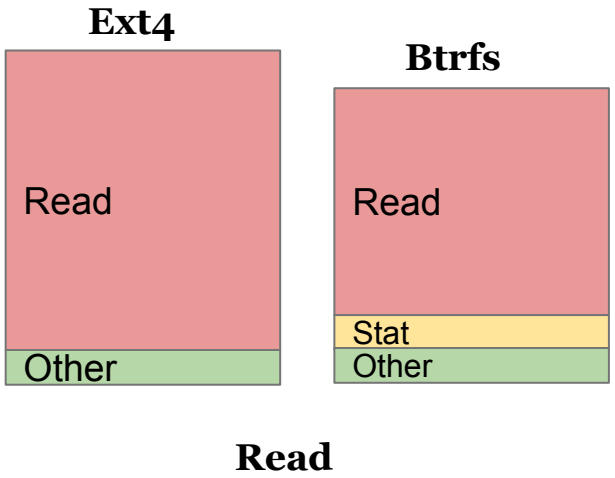
UNIVERSITY OF WATERLOO

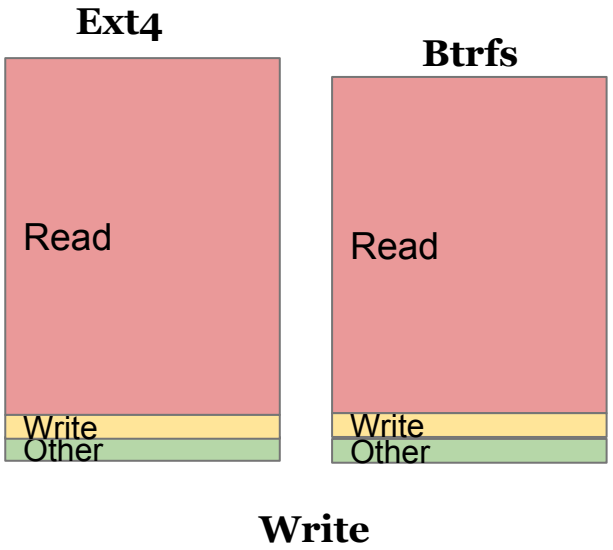
RQ3: Which system calls take longer in the File System?



Results

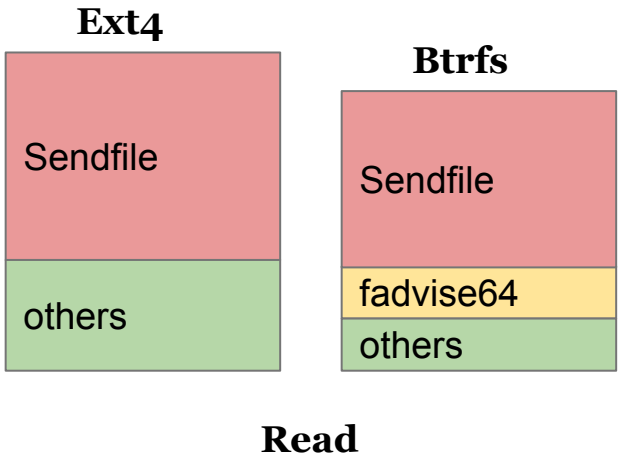
- Strace findings from Node Manager

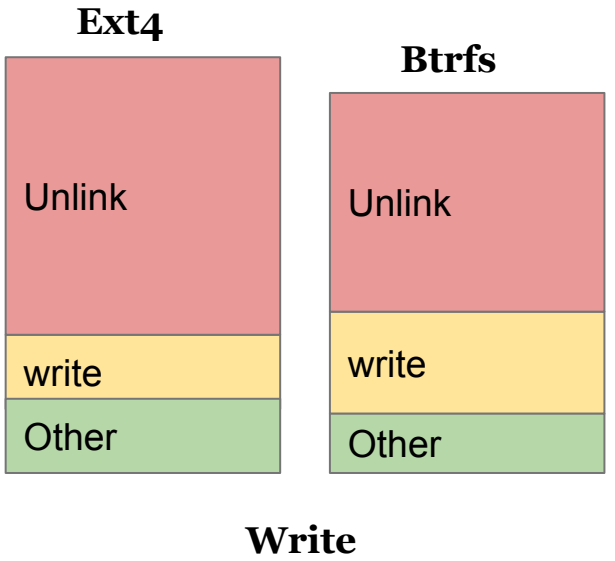




Results

- Strace findings from DataNode





Future Work

- Further exploration of the reason behind the behavioral difference between file systems using traces further down the OS stack.
- Expanding the scope of filesystems to other linux FS such as xfs.
- Configurations tuning for EXT4, BTRFS for performance improvement.

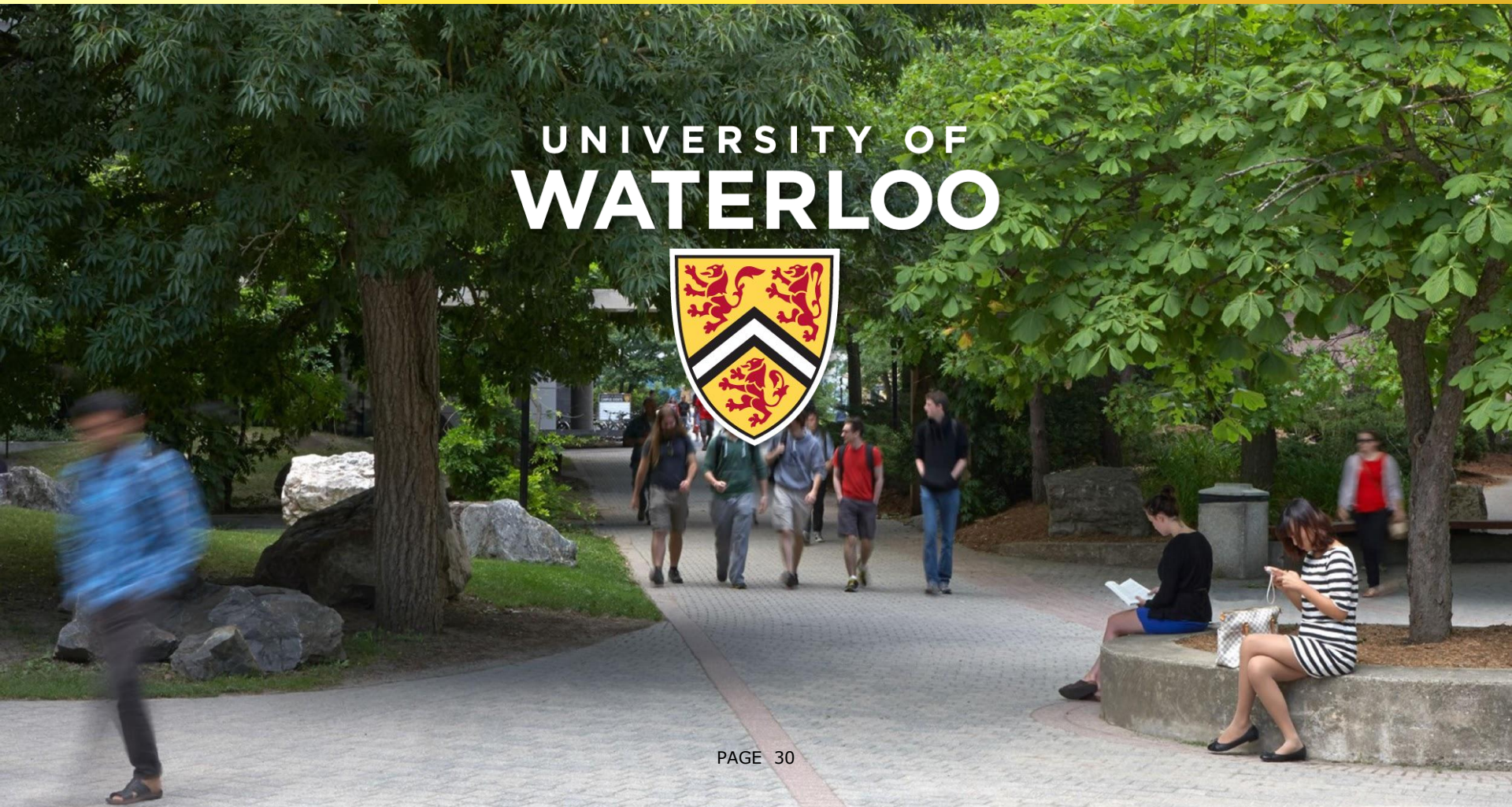
References

[1] J. Li, Q. Wang, D. Jayasinghe, J. Park, T. Zhu and C. Pu, "Performance Overhead among Three Hypervisors: An Experimental Study Using Hadoop Benchmarks," *2013 IEEE International Congress on Big Data*, Santa Clara, CA, 2013, pp. 9-16.
doi: 10.1109/BigData.Congress.2013.11

[2] Islam N.S., Lu X., Wasi-ur-Rahman M., Jose J., Panda D.K.. (2014) A Micro-benchmark Suite for Evaluating HDFS Operations on Modern Clusters. In: Rabl T., Poess M., Baru C., Jacobsen HA. (eds) Specifying Big Data Benchmarks. WBDB 2012, WBDB 2012. Lecture Notes in Computer Science, vol 8163. Springer, Berlin, Heidelberg

[3] Kuniyasu Suzaki, Kengo Iijima, Toshiki Yagi, and Cyrille Artho. Analysis of disk access patterns on file systems for content addressable storage. In 2011 Linux Symposium, pages 23–36, 2011.

THANK YOU


UNIVERSITY OF
WATERLOO