Work-in-Progress Session 2

Interference-aware Scheduling for Data-processing Frameworks in Container-based Clusters

Miguel G. Xavier

miguel.xavier@acad.pucrs.br

Advisor: Prof. César A. F. De Rose

Faculty of Informatics, PUCRS
Porto Alegre, Brazil

1st Joint International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems

Data-processing frameworks

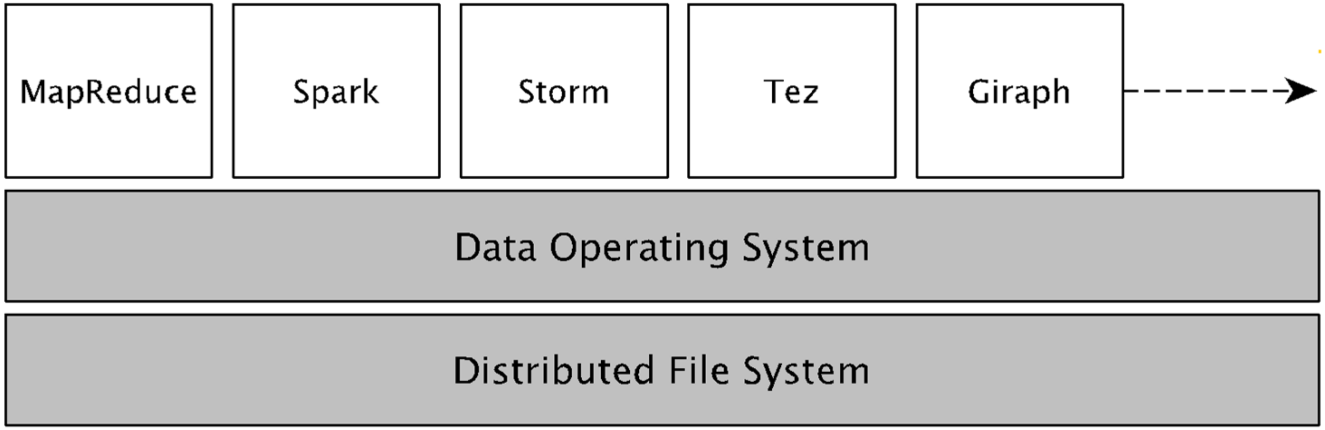
As the popularity of large-scale data analysis increases, the emergence of new data-processing frameworks and programming models beyond just MapReduce-centric also grows

To process data with different applications in multiple ways:

- real-time event processing (Storm);
- human-interactive SQL queries (Hive);
- batch processing (Java Apps);
- graph processing (Giraph);
- in-memory processing (Spark);
- machine learning (Mahout), and so on.

Cluster resource manager

Orchestrates multiple frameworks in a cluster of computers and allows applications to access the same data set independent of the framework



Cluster resource managers

Most popular solutions:

MESOS

- Shares a cluster between multiple different frameworks
- Creates another level of resource management
- Management is taken away from cluster's RMS

hadoop

- YARN - Hadoop Next Generation
- Better job scheduling/monitoring
- Uses virtualization to share a cluster among different frameworks

Problem Statement:

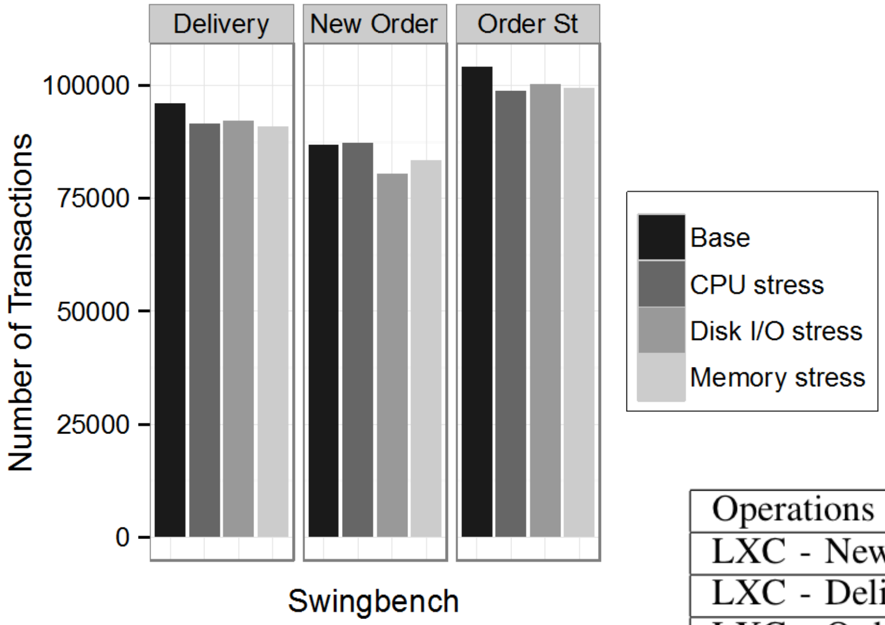
Interference-related performance degradation in resource-sharing clusters

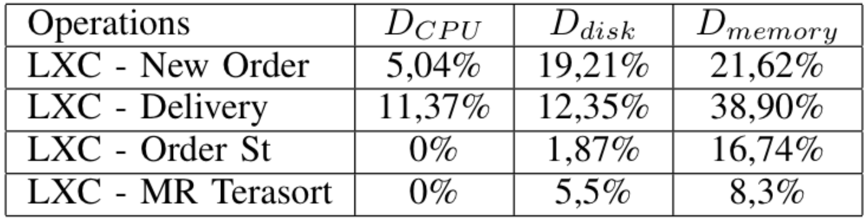
An application might interfere the performance of another co-located application in two ways:

- Resource Contention: when multiple applications compete for the same resource (CPU, disk, memory, network);
- Resource Isolation Weakness: when multiple co-located applications with allocated resources independently interfere each other.

Understanding contention-related performance overheads in resource-sharing clusters

Performance variations of co-located data-intensive applications in container-based clusters





Operations	D_{CPU}	D_{disk}	D_{memory}
LXC - New Order	5,04%	19,21%	21,62%
LXC - Delivery	11,37%	12,35%	38,90%
LXC - Order St	0%	1,87%	16,74%
LXC - MR Terasort	0%	5,5%	8,3%

On-going work

We have proposed an interference-aware scheduling for BigData frameworks, aiming at:

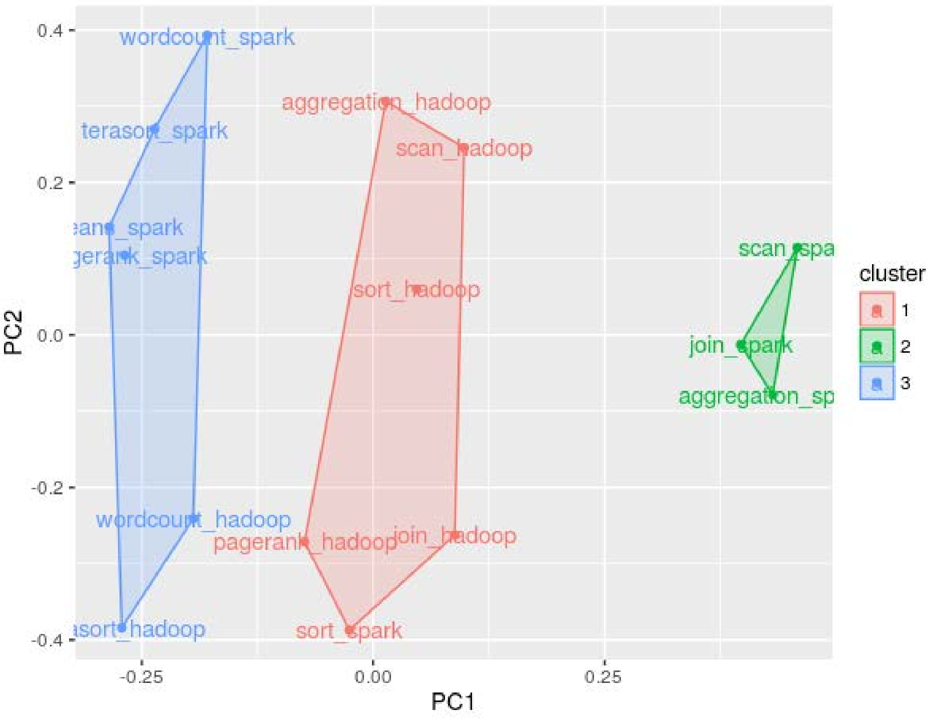
- Scheduling tasks to clusters in a way that minimizing the performance interference effect from co-located applications
- Characterize the performance interference impact and mitigate it whenever possible during the task scheduling/resource provisioning

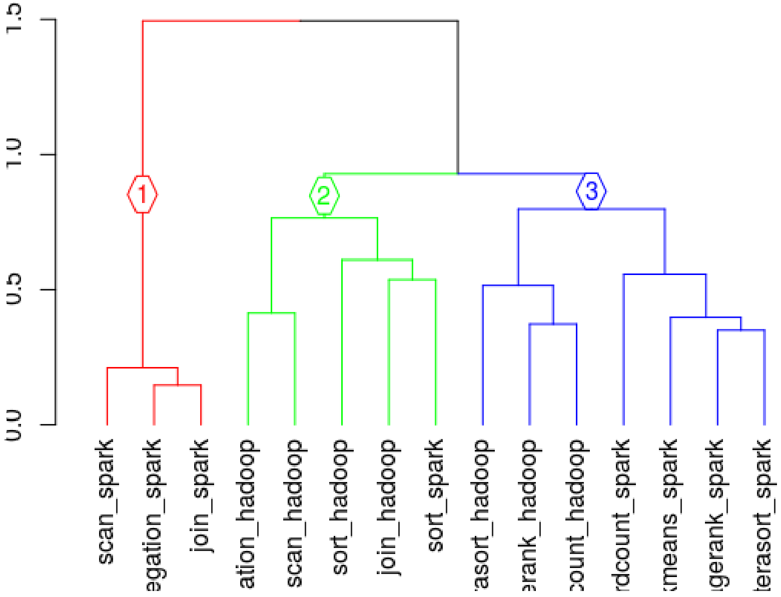
How to get there?

1. profiling queued applications to map resource contention effects
2. clustering applications per their similarity in terms of contention effects
3. scheduling applications' tasks on the best-suited nodes—the nodes that cause the lowest performance interference effects

Preliminary clustering analysis

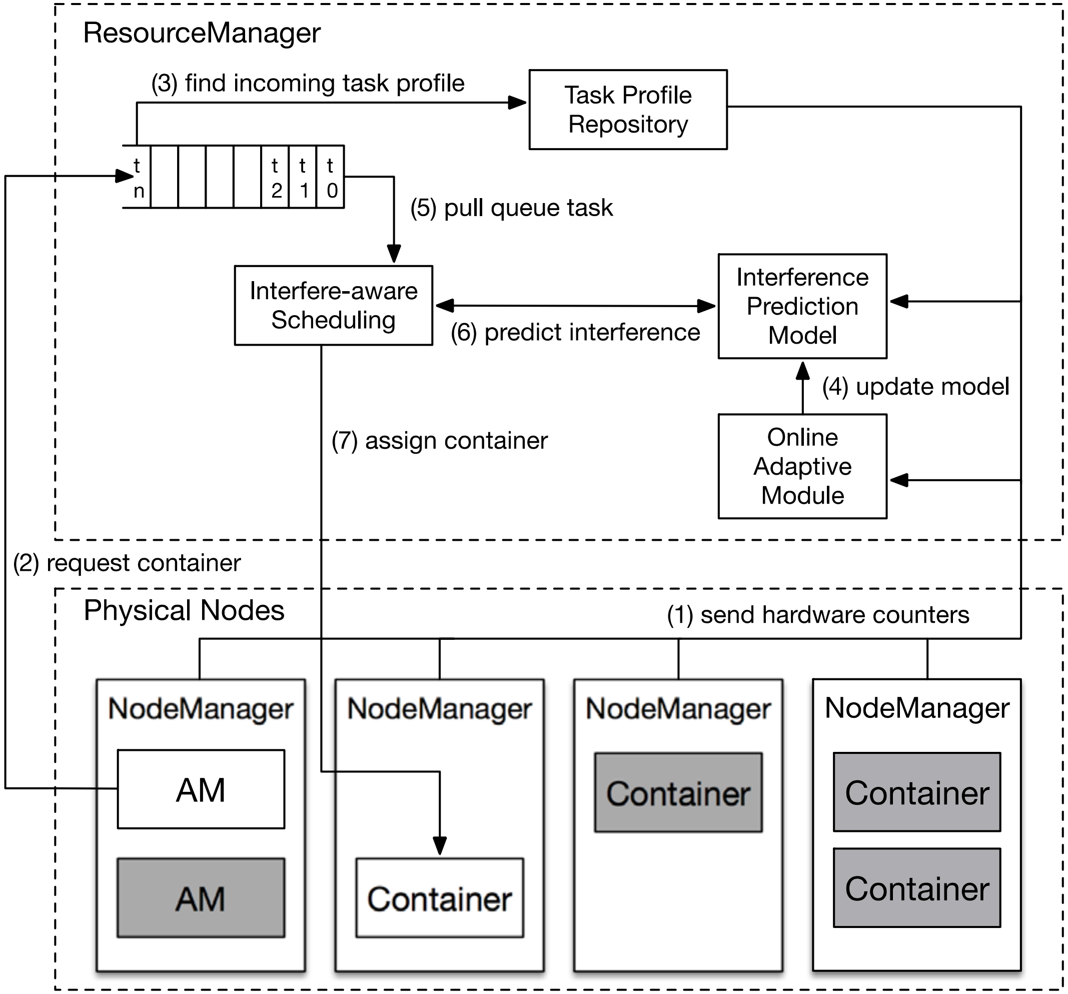
Applications are grouped per their similarity prior the scheduling process





Next Directions...

Interference-aware scheduling design in Yarn



Thank you for your attention !!