Performance Evaluation of Dynamic MapReduce Clusters

19th Conference of the Advanced School for Computing and Imaging
Eindhoven, The Netherlands

Bogdan Ghit, Alexandru Iosup, and Dick Epema

Parallel and Distributed Systems Group
Delft University of Technology
Delft, The Netherlands

COMMIT/

TU Delft
Delft University of Technology

Big Data Today

Batch processing

- Convert 11 mil. articles (1851-1922) to PDFs

Complex algorithms and workflows

- Track terrorist activity from credit-card receipts, hotel records, travel data
- How does the legal bans and tracker take-downs impact BitTorrent?

Small, very fast queries

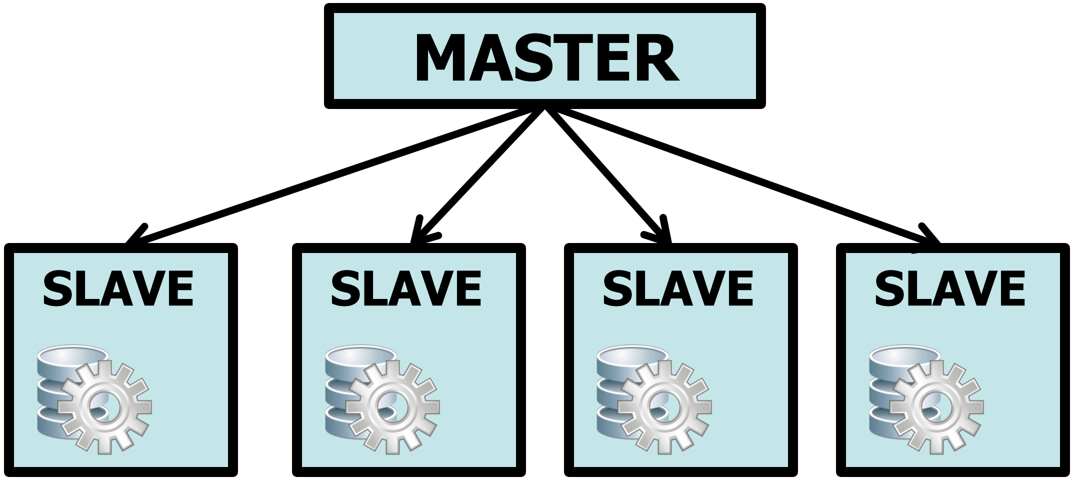
- Very popular at Facebok, Cloudera, Yahoo!

So, *different* data sets, *different* applications, *different* characteristics and performance, and… different frameworks!

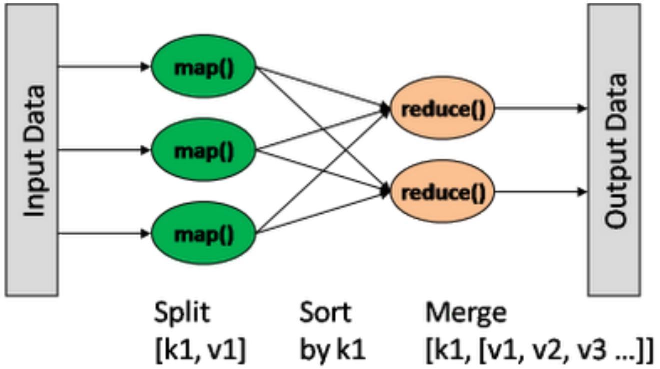
Yong Guo, **Bogdan Ghit**, Mihai Capota, Alexandru Iosup. *Survey on Big Data Use Cases.*

MapReduce Overview

- **The framework**
  - Distributed file system
  - Master-slave architecture



- **The computation**
  - Relatively small and independent processing units
  - Pipeline execution



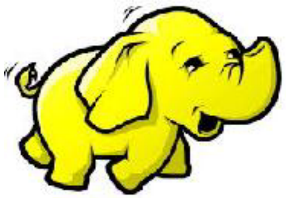
Why Multiple Frameworks?

- **Performance Isolation**
  - Scheduling artifacts from mixing long and short jobs
  - No one-size-fits-all policy: specific policies for different workloads

- **Data Isolation**
  - Secure data sets and protect users privacy
  - Configurations may be suboptimal for certain formats

- **Failure Isolation**
  - Hide the failures of a framework from the users of the others
  - Extend from single physical clusters to multicluster deployments

- **Version Isolation**
  - Different production and testing frameworks
  - Run different versions/releases simultaneously

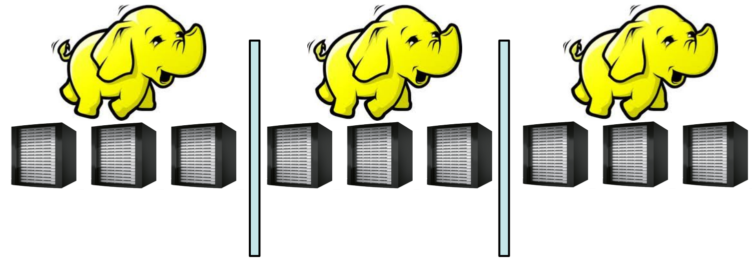


Bogdan Ghit, Nezih Yigitbasi, Dick Epema. *Resource Management for Dynamic MapReduce Clusters in Multicluster Systems (Best Paper Award)*, MTAGS' 12 (with SC).

TU Delft

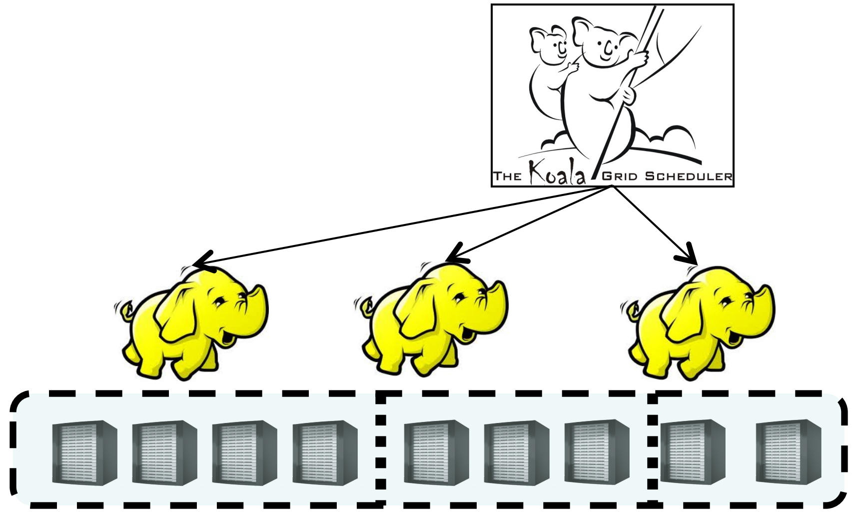
How to Provision Multiple Frameworks?

- *Static Partitioning*
  - Frameworks have complete control over a set of resources
  - Fragmentation and suboptimal resource utilization



- *Two-level Scheduling*
  - Control delegated to frameworks
  - Fine-grained resource multiplexing
  - No preemption nor specific policies
  - *Suboptimal for long tasks and large jobs (e.g., Mesos)*

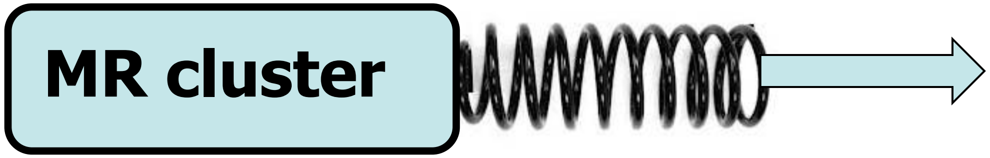
- **Dynamic Partitioning**
  - Course-grained resource multiplexing
  - Isolate data in separate DFS
  - Explicit policies for fair-sharing
  - Hint: dynamic MapReduce



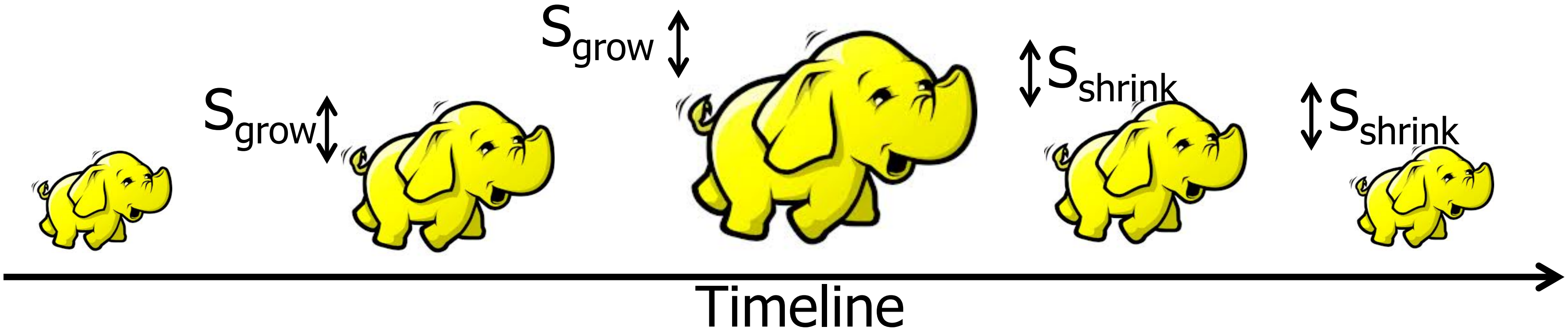
Goal: Balance the allocations to converge to similar levels of service

Dynamic MapReduce Cluster

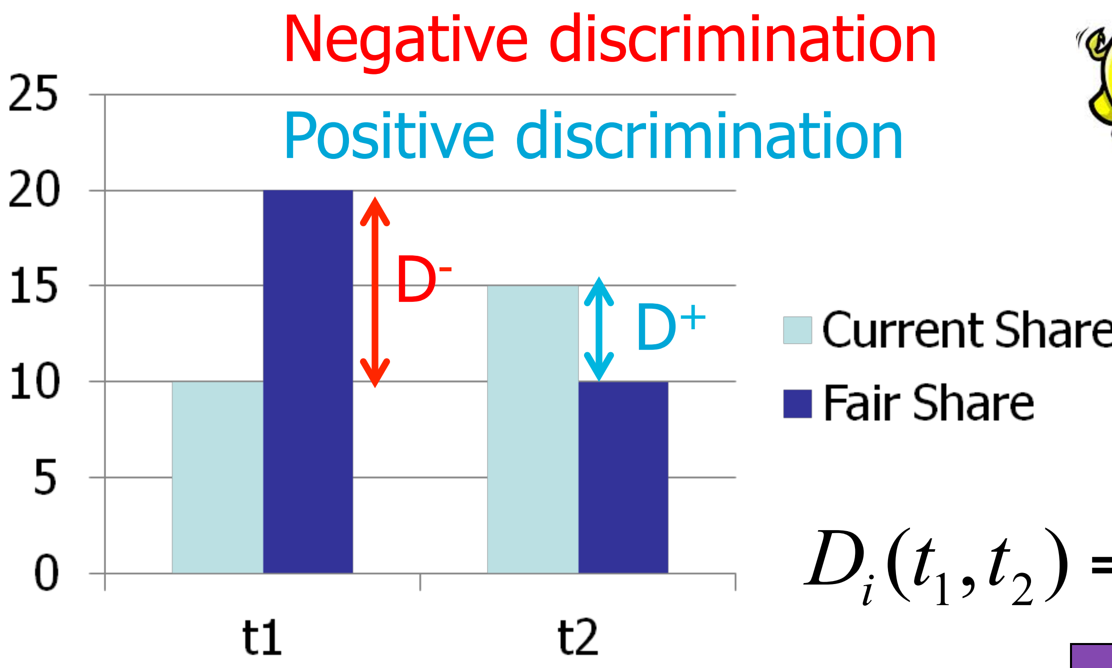
- **The tradeoffs**
  - Reliable data management through **replication**
  - Fast reconfigurations by relaxing the data locality model

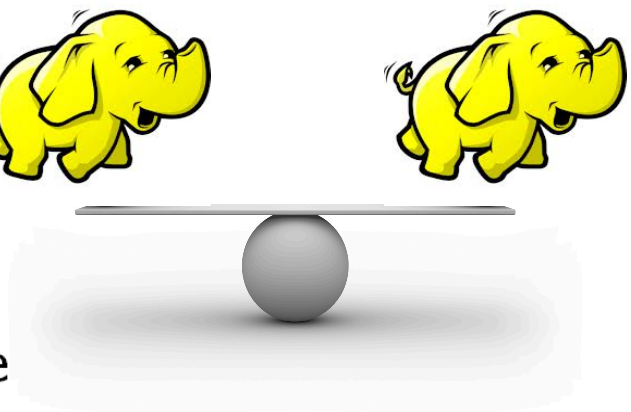


- **MR-cluster**
  - Core nodes: computations, storage **with** input data
  - Transient nodes: **only** computations
  - Transient-core nodes: computations, storage **without** input data



Fair or Unfair Allocations





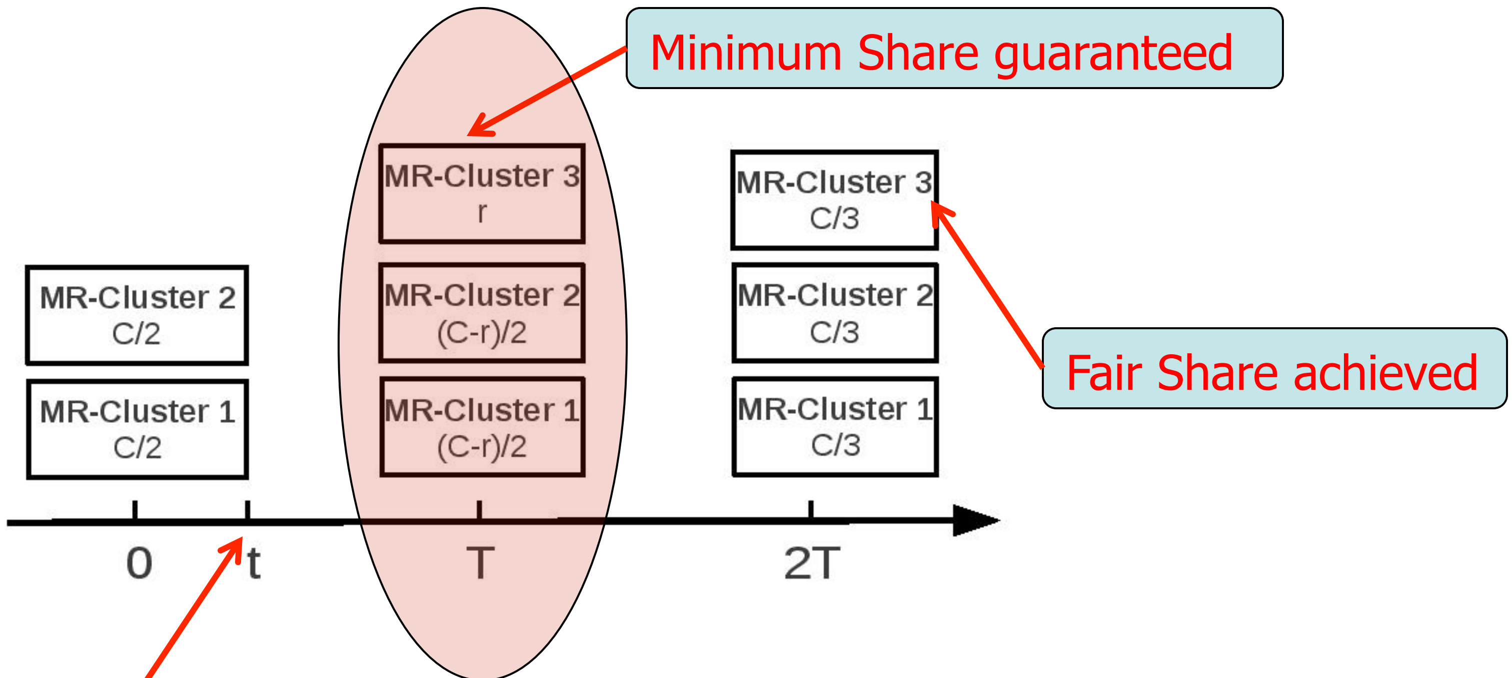
$$D_i(t_1,t_2) = \int_{t_1}^{t_2} (c_i(t) - w_i(t))dt$$

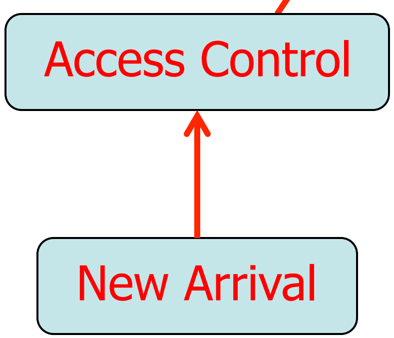
$$\sum D^+ = \sum D^-$$

- **Measure of imbalance:**

$$Var(D) = E[D^2] - E[D]^2 = E[D^2] > \tau$$

Admission Policy





- **Global view**
  - Take snapshots periodically
  - Gather samples of system operation
  - Use the averages y_i over the last interval
  - Adapt the weights

$$w_i = \frac{y_i}{\sum_{k=1}^{n} y_k}$$

TU Delft

Changing Shares

- **Differentiate** the MR-clusters
  - Demand-based weighting (e.g., queue size: jobs, data, tasks)
  - *Usage-based weighting* (e.g., processor, disk, both)
  - *Performance-based weighting* (e.g., job slowdown, throughput)

- **Resize** the MR-clusters to their fair shares
  - Shrink MR-clusters in D^+
  - Grow MR-clusters in D^-

Growing	Transient Nodes (TR)	Transient-Core Nodes (TC)
Shrinking	Instant Preemption (IP) • Kill tasks and reschedule	Delayed Preemption (DP) • Kill tasks and reschedule • Replicate data

Empirical Approach

- Popular MapReduce Benchmarks
  - Wordcount, Sort, PageRank, Kmeans
- **Real-world applications**
  - BTWorld use case: data collected from BitTorrent over 4 years.

Meet production workloads characteristics

Job	Type	Data	Input	Output
WC	compute	Random	200 GB	5.5 MB
ST	disk	Random	200 GB	200 GB
PR	compute	Random	50 GB	1.5 MB
KM	compute,disk	Random	70 GB	72 GB
TT	compute	BitTorrent	100 GB	3.9 MB
AH	disk,compute	BitTorrent	100 GB	90 KB

1) Many short jobs

2) Low selectivity

Tim Hegeman, **Bogdan Ghit**, Mihai Capota, Jan Hidders, Dick Epema, Alexandru Iosup. *The BTWorld Use Case for Big Data Analytics,* IEEE BigData, 2013

TU Delft

DAS-4 Infrastructure

- Research in systems for over a decade
  - 200 machines
  - 1,600 cores (quad cores)
  - 2.4 GHz CPUs, GPUs
  - 180 TB storage
  - 10 Gbps WAN / 20 Gbps Infiniband

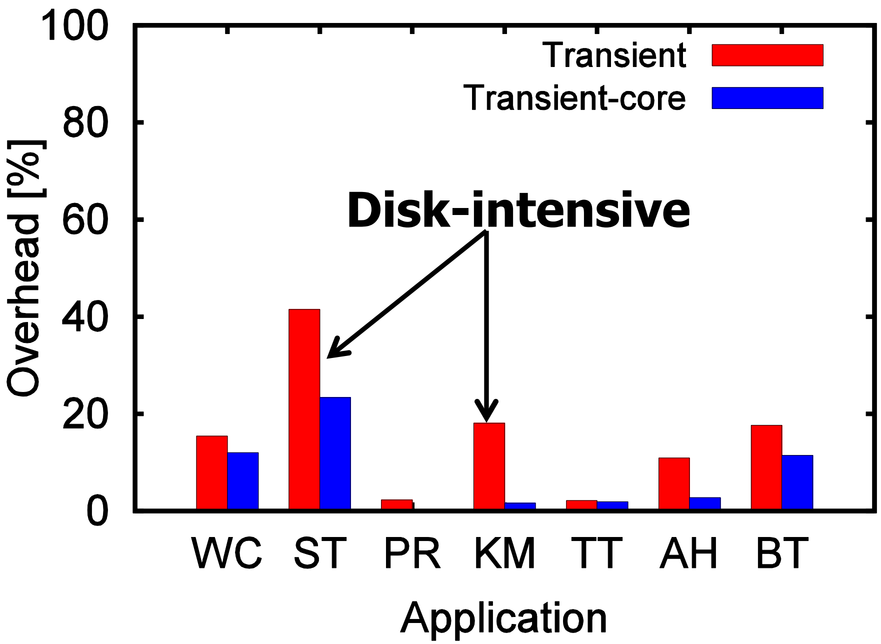
- Meta-scheduler, transparent for local schedulers
  - Specific modules for different types of jobs
  - MapReduce, Workflows, Bags-of-Tasks, etc.
  - Now extended to cloud interfaces

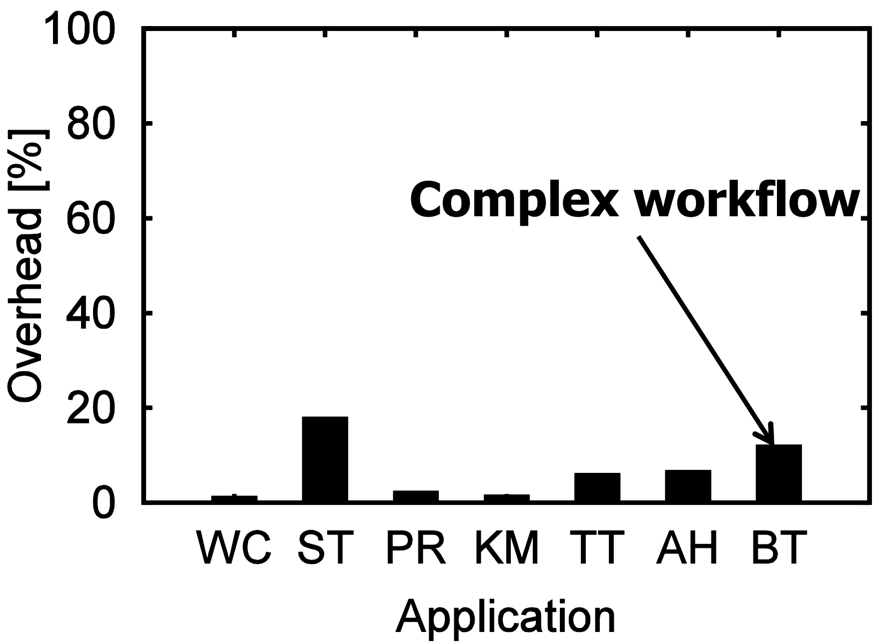
Lipu Fei, **Bogdan Ghit**, Alexandru Iosup, Dick Epema. *KOALA-C: A Task Allocator for Integrated Multicluster and Multicloud Environments*.

Bogdan Ghit, Nezih Yigitbasi, Dick Epema. *Resource Management for Dynamic MapReduce Clusters in Multicluster Systems (Best Paper Award)*, MTAGS' 12 (with SC).

TU Delft

Impact of Data Locality

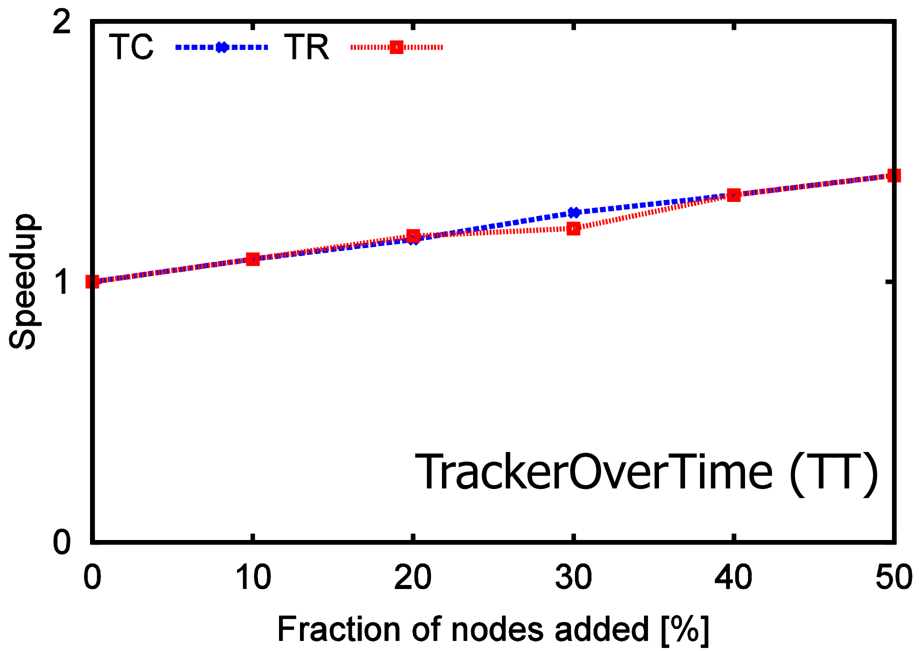


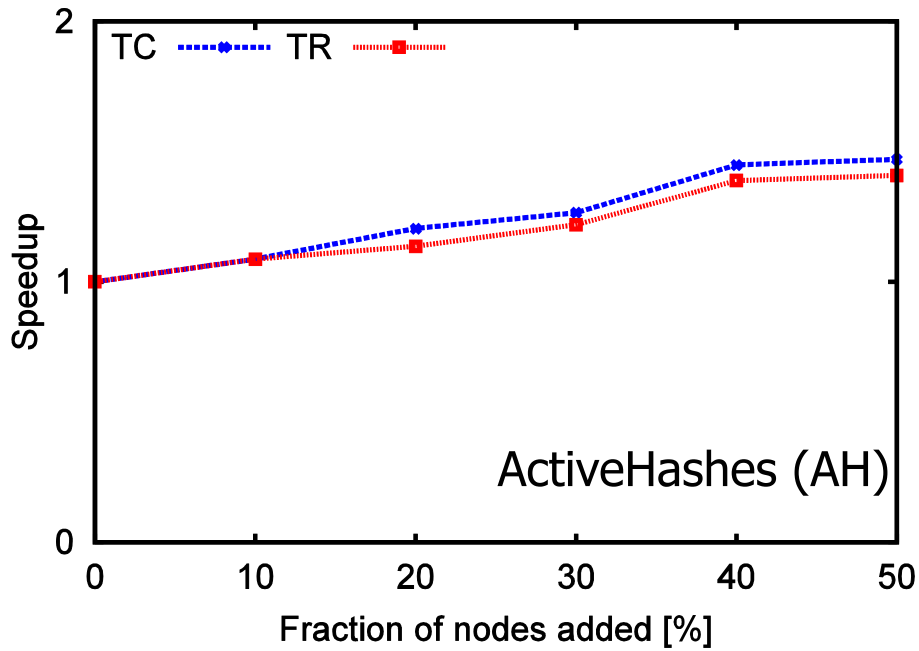


- 10 core + 10 TR/TC
- **TC nodes reduce overhead of disk-intensive jobs**

- 20 core in 2 physical clusters
- **Low overhead in co-allocation settings**

Growing MR-clusters



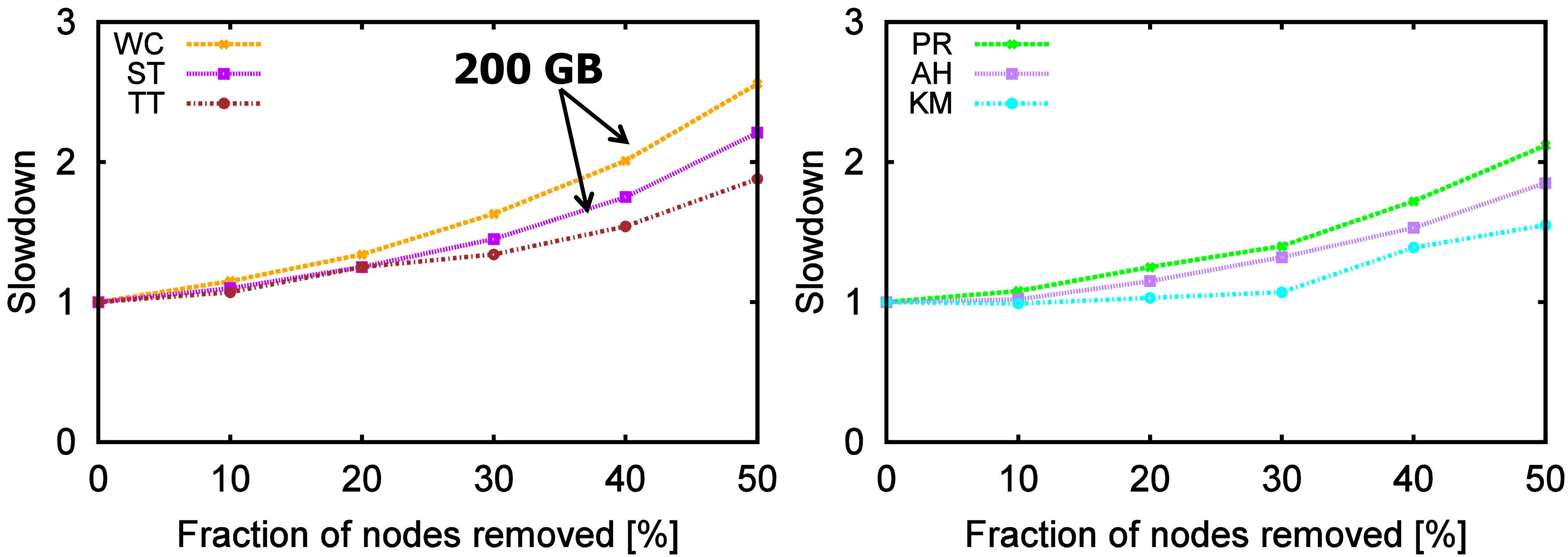


- 20 core nodes + TR/TC

- **Transient and transient-core nodes significantly improve the performance of both processor and disk intensive jobs**

TUDelft

Shrinking MR-clusters

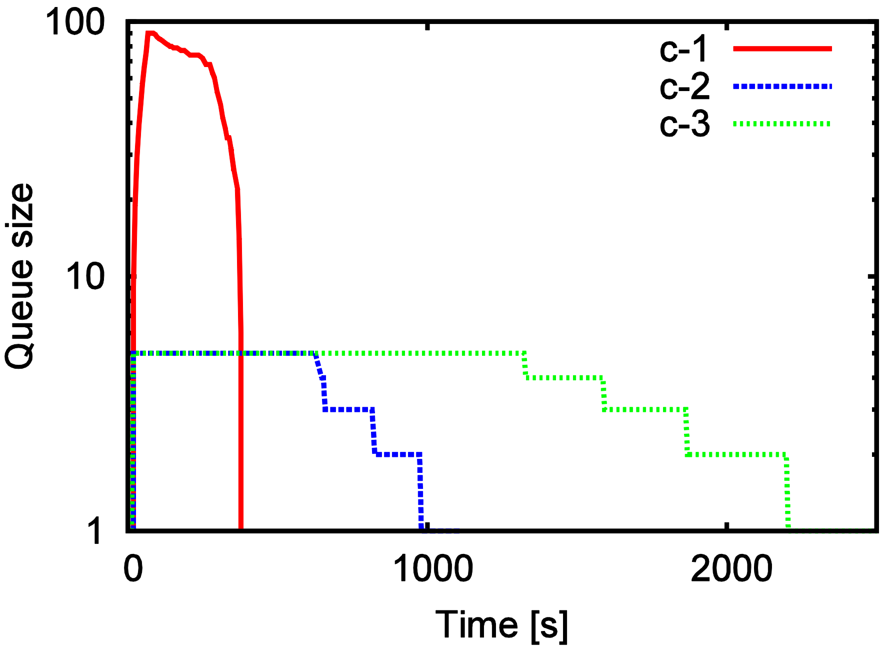


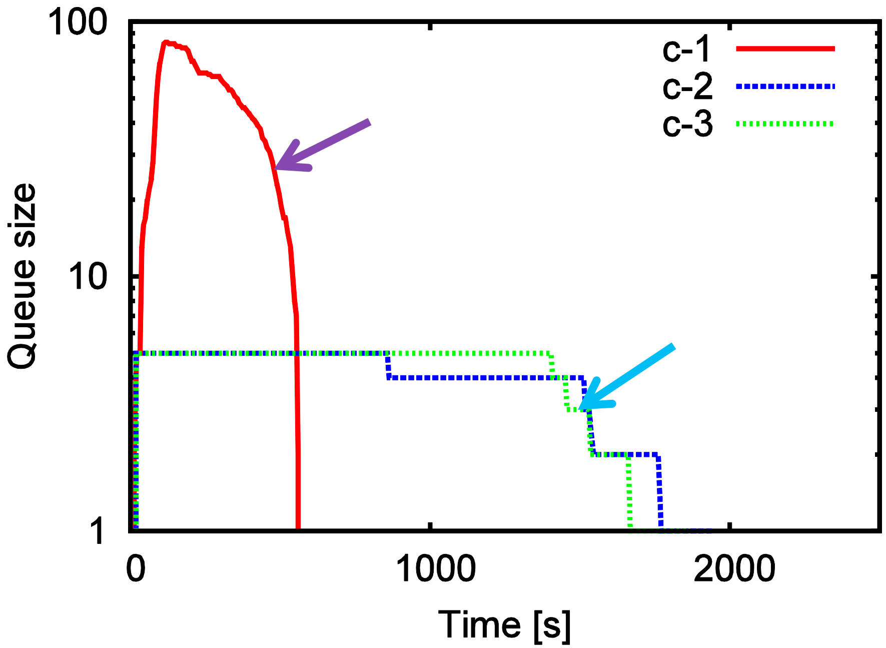
- 20 core nodes

- **Less compute-intensive jobs may have higher runtime due to input data size**

TU Delft

Fairness of Weighting





- c-1: 90 small jobs (1 GB)
- c-2: 5 medium jobs (50 GB)
- c-3: 5 large jobs (100 GB)

- 60 resources and 100 Sort jobs in total
- Weighting: number of tasks in queue
- TC growing, DP shrinking

- **Preserves performance of small workloads**
- **Achieves balanced resource allocations for heavy workloads**

TU Delft

Conclusions

- **New abstraction** for dynamic MapReduce clusters
  - Relaxed data locality model, with two types of growing/shrinking
  - Experiments with synthetic and real-world single applications
  - MR-clusters may benefit from weak data locality!
- **Grow and shrink** mechanism to provision multiple MR-clusters
  - Measure the fairness or the imbalance
  - Weighted proportional allocations to balance
  - Experiments with workloads mixing different job types
  - Balanced allocations for heavy workloads, without impact on small workloads!
- **Future Work**
  - *Explore the full design space of policies*

More Information

- **Home pages**
  - www.pds.ewi.tudeltf.nl/ghit
  - www.pds.ewi.tudelft.nl/~iosup
  - www.pds.ewi.tudelft.nl/epema

- **KOALA**
  - www.st.ewi.tudelft.nl/koala

THE Koala GRID SCHEDULER

TU Delft