Self-Tuning of Job Concurrency for Hadoop Performance Improvement

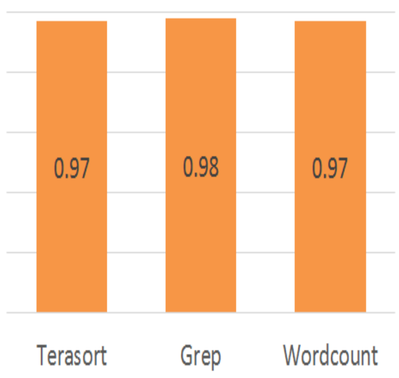
ACIS ADVANCED COMPUTING & INFORMATION SYSTEMS

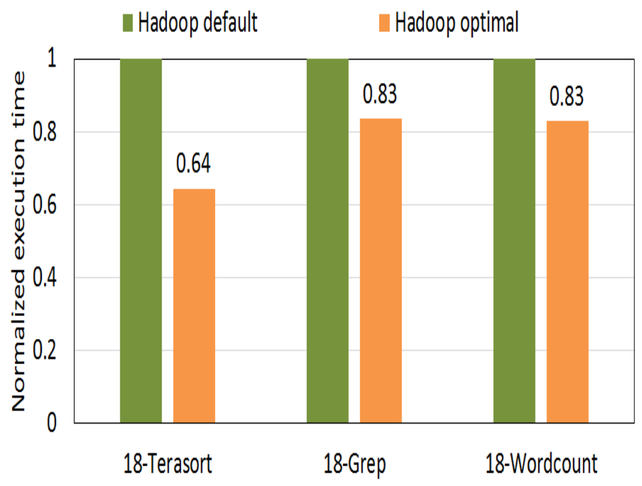
Giljae Lee and José A.B. Fortes

Background and Motivation

- **Performance in multi-tenant data-analytics frameworks**
  - Disparity of performance according to configuration of data-analytics frameworks
  - Minimize execution time of data-analytics jobs (user perspective)
  - Maximize resource utilization of clusters (provider perspective)

- **Tuning challenges for performance in data-analytics frameworks**
  - Too many tunable configuration parameters
  - Parameter interdependencies
  - Domain (application, system) specific knowledge requirement
- **Hadoop behavior**
  - CPU usage vs. MapReduce
  - Makespan vs. job concurrency (MR) tasks





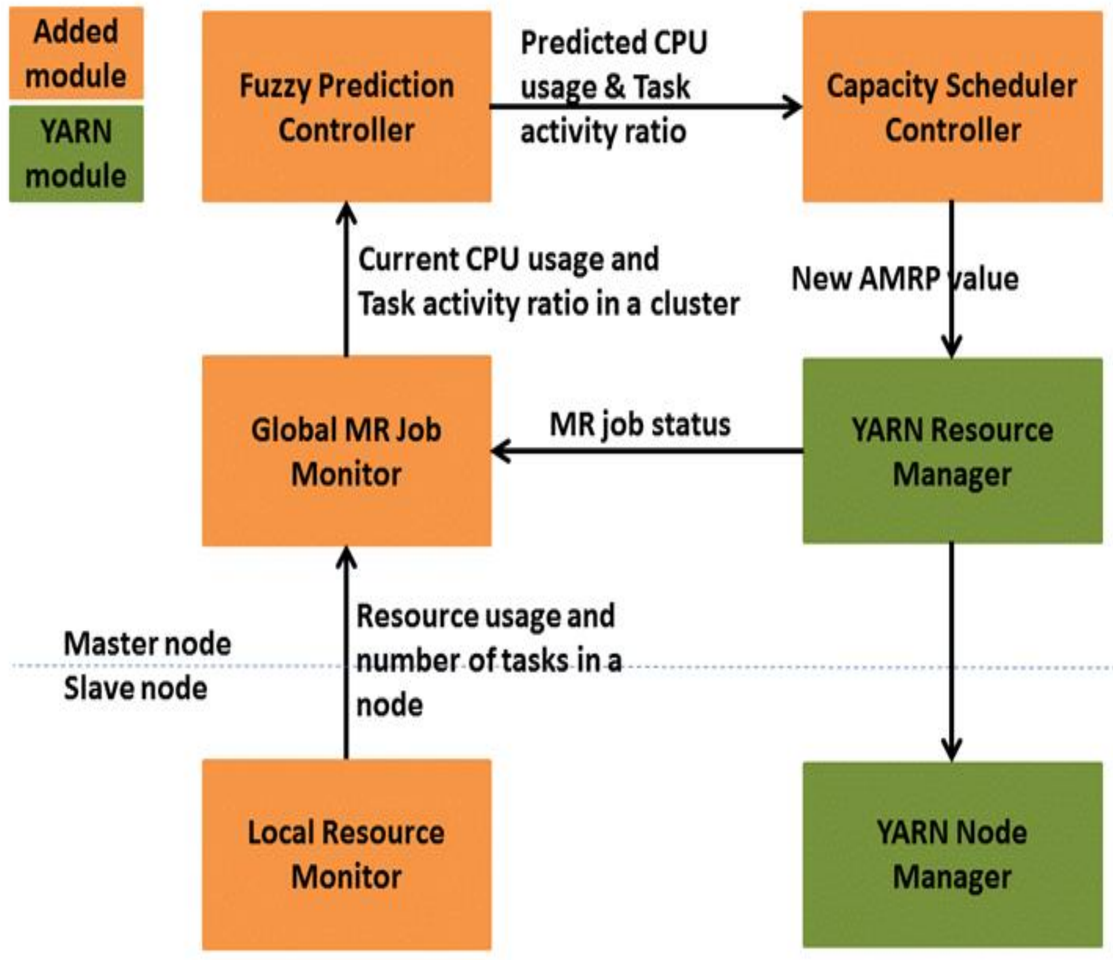
Self-Tuning Approaches and Challenges

Approach	Features	Challenges
Individual MR job	• Performance prediction model • Various techniques (e.g., Machine learning)	• Time-consuming training • Renew or retraining models
MR jobs concurrency	• Dynamic adjustment of concurrent MR jobs using log-based resource usage • Dynamic adjustment of concurrent MR tasks monitoring system-metric resource usage	• Insufficient relationship between workload and resource usage of MR jobs • Modification of existing Hadoop systems
Slot utilization	• Dynamic slot allocation to MR tasks in Hadoop 1	• Inapplicable to recent container-based Hadoop 2 system

Autonomic Control of MR Job Concurrency

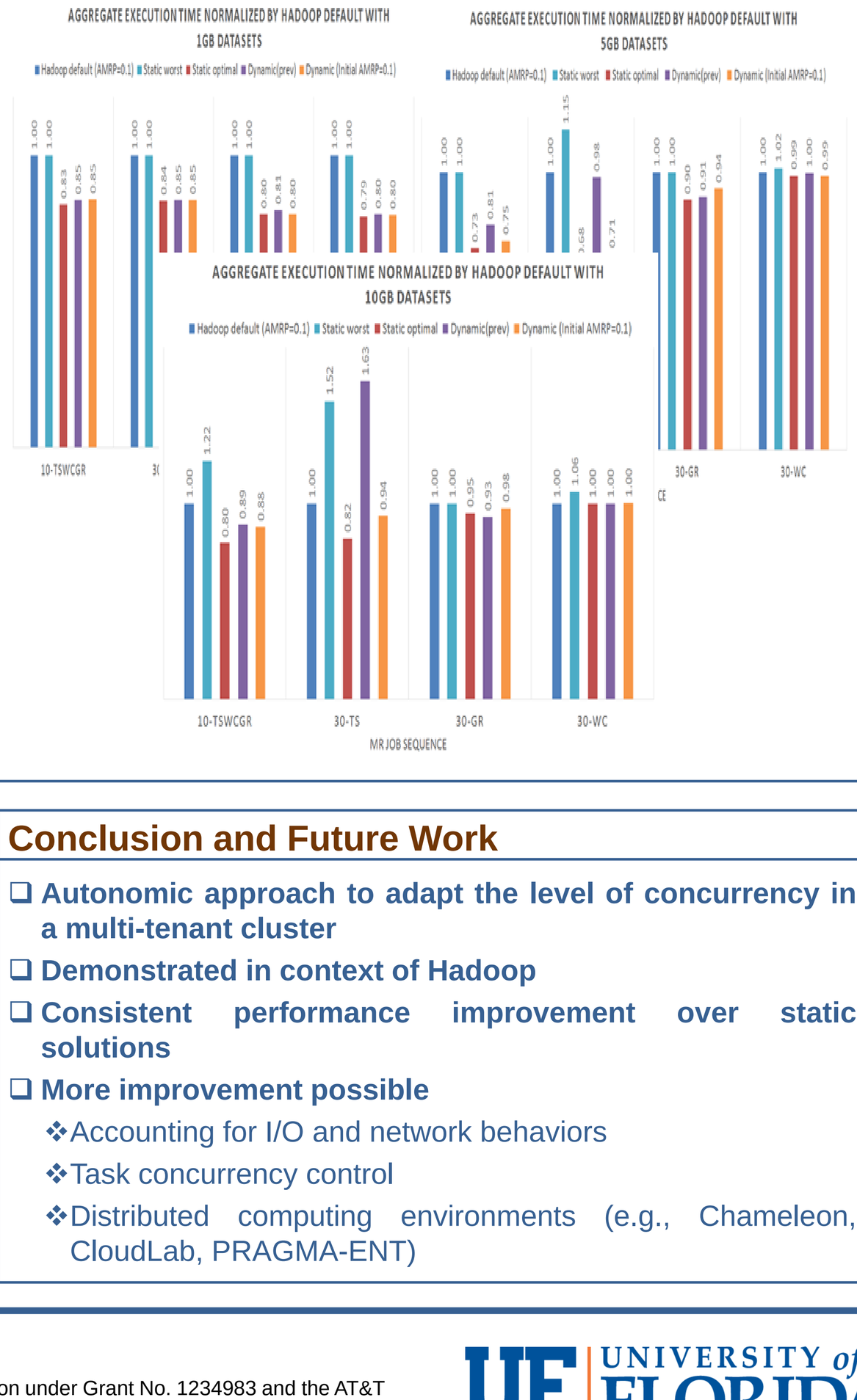
- **Goals**
  - Minimize overall execution time of multiple concurrent MR jobs
  - Use unmodified Hadoop software
- **Scope**
  - Hadoop version 2.7.x
  - Default Hadoop Capacity scheduler
  - Use of available monitoring metrics

Architecture Overview



Results

- **Performance near static optimal**
- **Scalability over size of datasets and types of MR jobs**



Conclusion and Future Work

- **Autonomic approach to adapt the level of concurrency in a multi-tenant cluster**
- **Demonstrated in context of Hadoop**
- **Consistent performance improvement over static solutions**
- **More improvement possible**
  - Accounting for I/O and network behaviors
  - Task concurrency control
  - Distributed computing environments (e.g., Chameleon, CloudLab, PRAGMA-ENT)

PRAGMA 32 : April 12-15, 2017

This material is based in part upon work supported by the National Science Foundation under Grant No. 1234983 and the AT&T Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation and AT&T Foundation.

UF UNIVERSITY of FLORIDA