

MOHA: Many-Task Computing meets the Big Data Platform

Jik-Soo Kim, Cao Nguyen, Soonwook Hwang National Institute of Supercomputing and Networking at KISTI



Distributed/Parallel computing systems to support various types of challenging applications

 HTC (High-Throughput Computing) for relatively long running applications consisting of *loosely-coupled* tasks

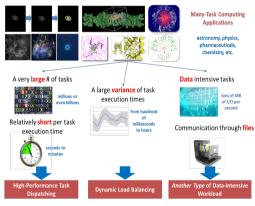
BUILD HICONDOR SET SHOME

- HPC (High-Periormance Computing) targets efficiently processing tightly-coupled parallel tasks
- DIC (Data-intensive Computing) mainly provide the effectively leveraging distributed storage systems and parallel processing frameworks



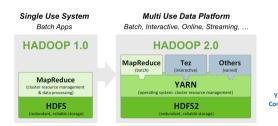
Many-Task Computing (MTC) as a new computing paradigm [I. Raicu, I. Foster, and Y. Zhao, MTAGS'08]

- •A very large number of tasks (millions or even billions) •Relatively short per task execution times (sec to min) •Data intensive tasks (i.e., tens of MB of I/O per second)
- •A large variance of task execution times (i.e., ranging from hundreds of milliseconds to hours)
- Communication-intensive, however, not based on message passing interface but through files



Hadoop, the *de facto standard* "Big Data" store and processing infrastructure

with the advent of Apache Hadoop YARN, Hadoop 2.0 is evolving into *multi-use data platform*harness various types of data processing workflows
decouple application-level scheduling and resource management



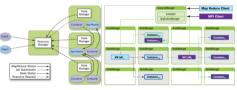
Main Contribution of this Research Work

- •MOHA (Many-task computing On HAdoop)
- framework which can effectively *combine* Many-Task Computing technologies with the existing Big Data platform Hadoop
- developed as one of *Hadoop YARN applications* transparently *cohost* existing MTC applications with other Big Data processing frameworks in a single Hadoop cluster



Hadoop YARN Execution Model

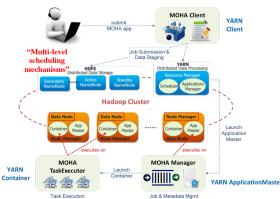
- *platform layer* is responsible for resource management (*first-level* scheduling)
- •Resource Manager, Node Manager
- framework layer coordinates application execution (second-level scheduling)
- •ApplicationMaster → New MOHA Framework !



MOHA System Architecture

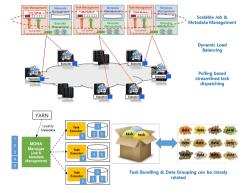
•MOHA Client (YARN Client)

- submit a MOHA job and performs data staging
 A MOHA job is a bag of tasks (i.e., a collection of multiple tasks)
- A MOHA job is a bag of tasks (i.e., a collection of multiple tasks)
 upload required data into the HDFS
- •MOHA Manager (YARN ApplicationMaster) •create and launch MOHA job queues
- •split a MOHA job into multiple tasks a ge Apache Kafka •get containers allocated and launch MOHA TASKEXECULAR
- •MOHA TaskExecutor (YARN Container)
- •pull the tasks from the MOHA job queues and process them



Hadoop can be a viable choice for addressing challenging MTC applications

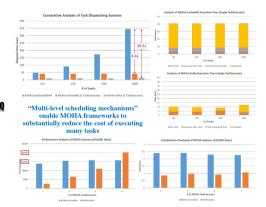
- technologies from MTC community should be effectively converged into the ecosystem
- seamless integration of various techniques such as highperformance task dispatching, effective dynamic load balancing, data-intensive workload support



Performance Evaluation

- •Comparison Models
- •YARN Distributed-Shell, MOHA-ActiveMQ, MOHA-Kafka
 •Workload

•Microbenchmark based on "sleep 0"



Conclusion and Future Work

- Design and implementation of MOHA (Many-task computing On HAdoop) framework
 MOHA prototype as a Proof-of-Concept
- MOHA prototype as a Prototo-Concept
 MOHA can bring many interesting research issues
 related to data grouping & declustering on HDFS, scalable job/metadata management, dynamic load balancing, etc.
 considering applying a new type of high-performance storage system in HPC area such as Lustre on top of Hadoop
 applying real MTC applications

•ultimately contributing to a new data processing framework for MTC applications in Hadoop 2.0 ecosystem