A Hierarchical MapReduce Framework

Yuan Luo and Beth Plale
School of Informatics and Computing, Indiana University
Data To Insight Center, Indiana University

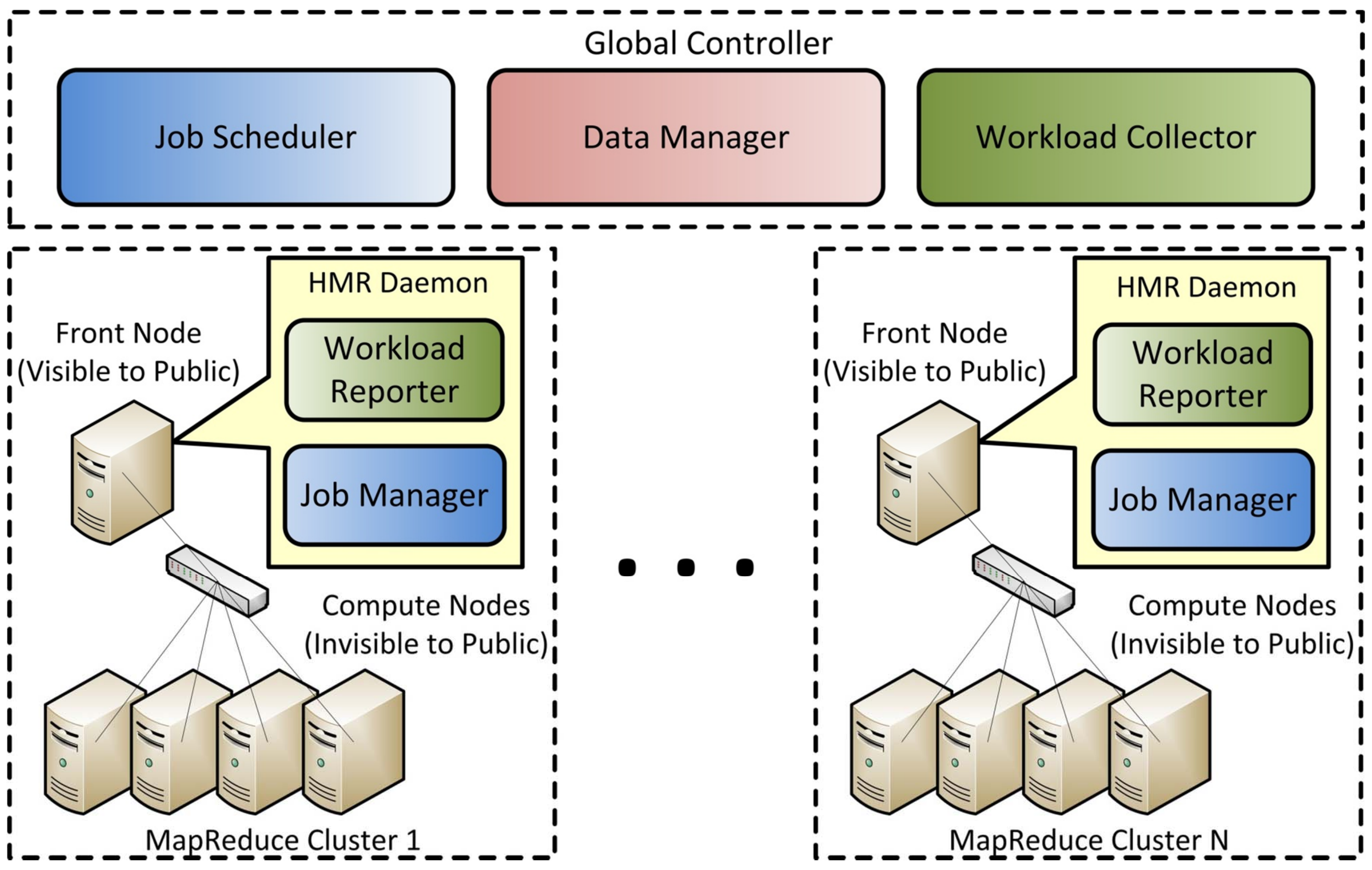
DATA TO INSIGHT CENTER
INDIANA UNIVERSITY
Pervasive Technology Institute

Motivation

- While MapReduce has proven useful on data-intensive high throughput applications, conventional MapReduce model limits itself to scheduling jobs within a single cluster. As job sizes become larger, single-cluster solutions grow increasingly inadequate.
- An input dataset could be very large and widely distributed across multiple clusters. When mapping such data-intensive tasks to compute resources, scheduling algorithms need to determine whether to bring data to computation or bring computation to data.

Architecture

We present a Hierarchical MapReduce (HMR) framework that gathers computation resources from different clusters and runs MapReduce jobs across them.



Global Controller:

- Job Scheduler, Data Manager, and Workload Collector;

Local MapReduce Clusters:

- A HMR daemon on each MapReduce cluster master node
- Compute nodes not public accessible.

Scheduling Algorithms

Compute Capacity Aware Scheduling (CCAS) :

- For compute-intensive jobs.
- Distribution of Map tasks to be scheduled to local clusters is calculated based on the computing power of each cluster, e.g., the Available Mappers, CPU speed, memory size, storage capacity, etc.

Data Location Aware Scheduling (DLAS) :

- For data-intensive jobs.
- Data set is partitioned and replicated among clusters.
- A candidate cluster for processing a data partition requires the physical residence of the data partition.
- Selected clusters have similar ratio of data partition size over cluster compute capacity, or called, balanced processing time.
- No data transfer consideration in the current version.

References

[1] Yuan Luo, Zhenhua Guo, Yiming Sun, Beth Plale, Judy Qiu, Wilfred Li. 2011. A Hierarchical Framework for Cross-Domain MapReduce Execution. In*Proceedings of the second international workshop on Emerging computational methods for the life sciences* (ECMLS '11). ACM, New York, NY, USA, 15-22. DOI=10.1145/1996023.1996026

[2] Yuan Luo and Beth Plale, Hierarchical MapReduce Programming Model and Scheduling Algorithms, *Doctoral Symposium of the 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, Ottawa, Canada, May 13-16, 2012, to appear.

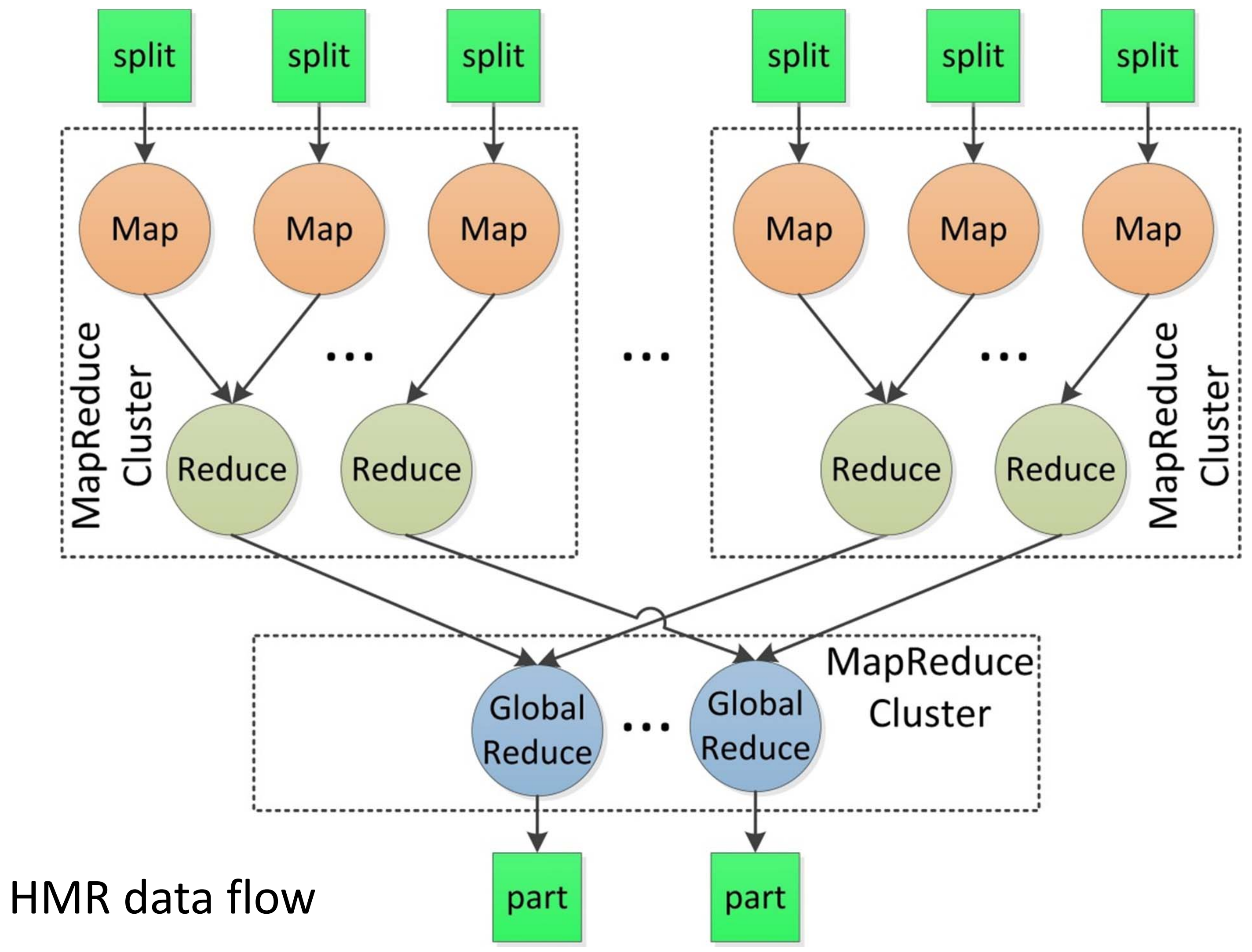
[3] Data To Insight Center, http://d2i.indiana.edu

Programming Model

Map-Reduce-GlobalReduce model:

- ***Map*** takes an input data split and produces an intermediate key/value pair;
- ***Reduce*** takes an intermediate input key and a set of corresponding values produced by the Map task, and outputs a different set of key/value pairs.
- ***GlobalReduce*** takes the output from all the local clusters to outputs the final result.

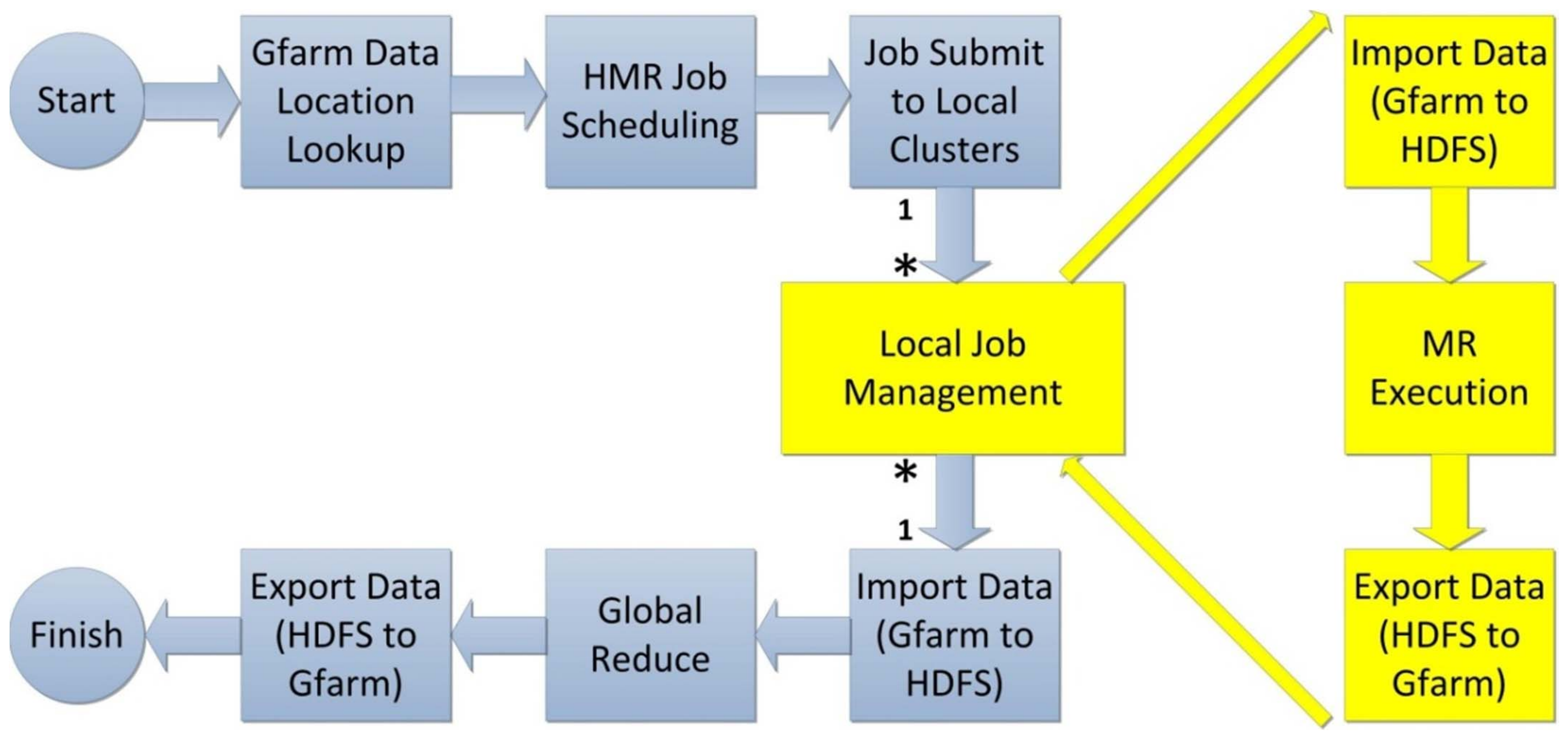
Function Name	Input	Output
Map	(k^i, v^i)	(k^m, v^m)
Reduce	$(k^m, [v_1^m, \dots, v_n^m])$	(k^r, v^r)
GlobalReduce	$(k^r, [v_1^r, \dots, v_n^r])$	(k^o, v^o)



Relevance to PRAGMA

Data-intensive applications

- We used Gfarm to share data sets among the Hadoop clusters using a MapReduce version of grep as a test case. Two 4-nodes virtual clusters (pragma-f0 and pragma-f1) were provisioned on the PRAGMA testbed to test the grep application.
- The steps of a HMR execution with Gfarm shows in the graph below. The blue box steps are executed on the Global Controller and the yellow box steps are executed on local MapReduce clusters.
- Gfarm acts well for data location lookup.


Start
Gfarm Data Location Lookup
HMR Job Scheduling
Job Submit to Local Clusters
1
*
Local Job Management
*
1
Import Data (Gfarm to HDFS)
MR Execution
Export Data (HDFS to Gfarm)
Import Data (Gfarm to HDFS)
Global Reduce
Export Data (HDFS to Gfarm)
Finish

Potential PRAGMA research questions going forward

- Data transfer in the next version of DLAS algorithms; Implementation using Gfarm as data transfer/replicate service.
- HMR on PRAGMA clusters over multiple institutes, heterogeneous network challenges HMR scheduling algorithms.