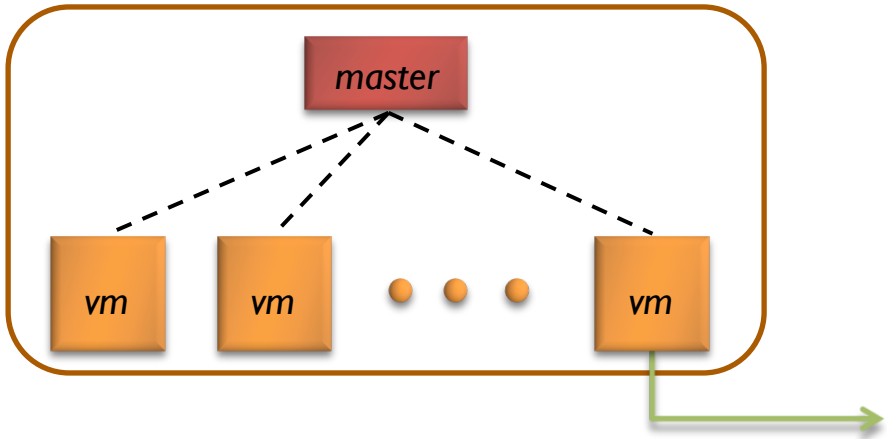
Locality-Aware Dynamic VM Reconfiguration on MapReduce Clouds

Jongse Park, Daewoo Lee, Bokyeong Kim,
Jaehyuk Huh, Seungryoul Maeng

Virtual Clusters on Cloud

- Private cluster on public cloud
  - Distributed computing platforms
    - MapReduce [OSDI '04], Hadoop, Dryad [Eurosys '07]
    - New York Times used 100 nodes on *Amazon EC2* [OSDI '08]
  - Each VM in a virtual cluster has static configuration

virtual cluster



e.g. Amazon EC2 VM instance types

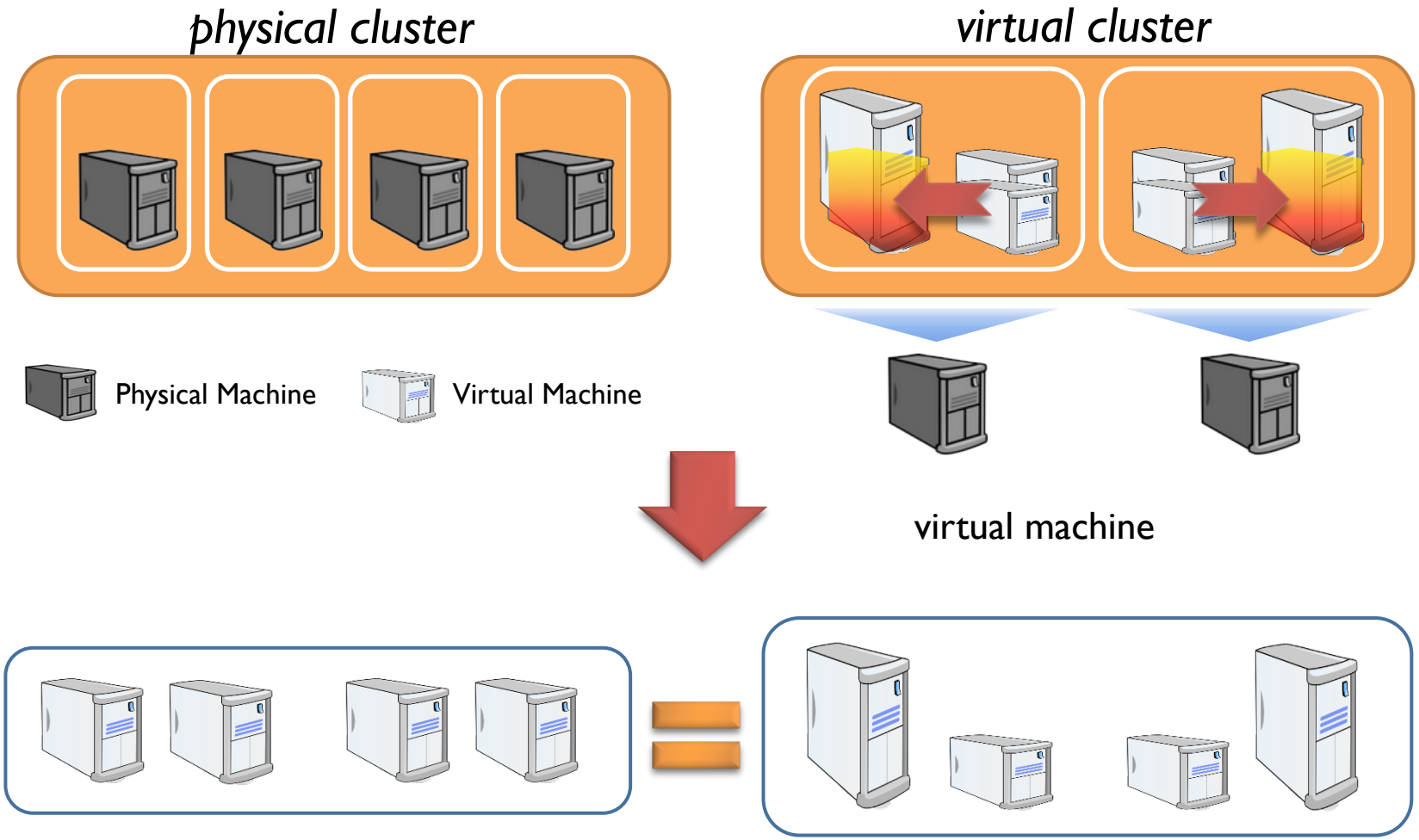
Instance types	Configuration
Small	1 virtual core, 1.7GB memory
Large	2 virtual cores, 7.5GB memory
Extra Large	4 virtual cores, 15GB memory

Resource Utilization Management

- Physical cluster
  - Load balancing is the only mechanism for higher utilizati on
- Virtual cluster
  - Dynamic resource management is also possible
    - With using resource *hot-plug* technique
    - Possible resource types: ***core*** and ***memory***

We focus on ***core*** hot-plugging in this work

Dynamic Resource Management

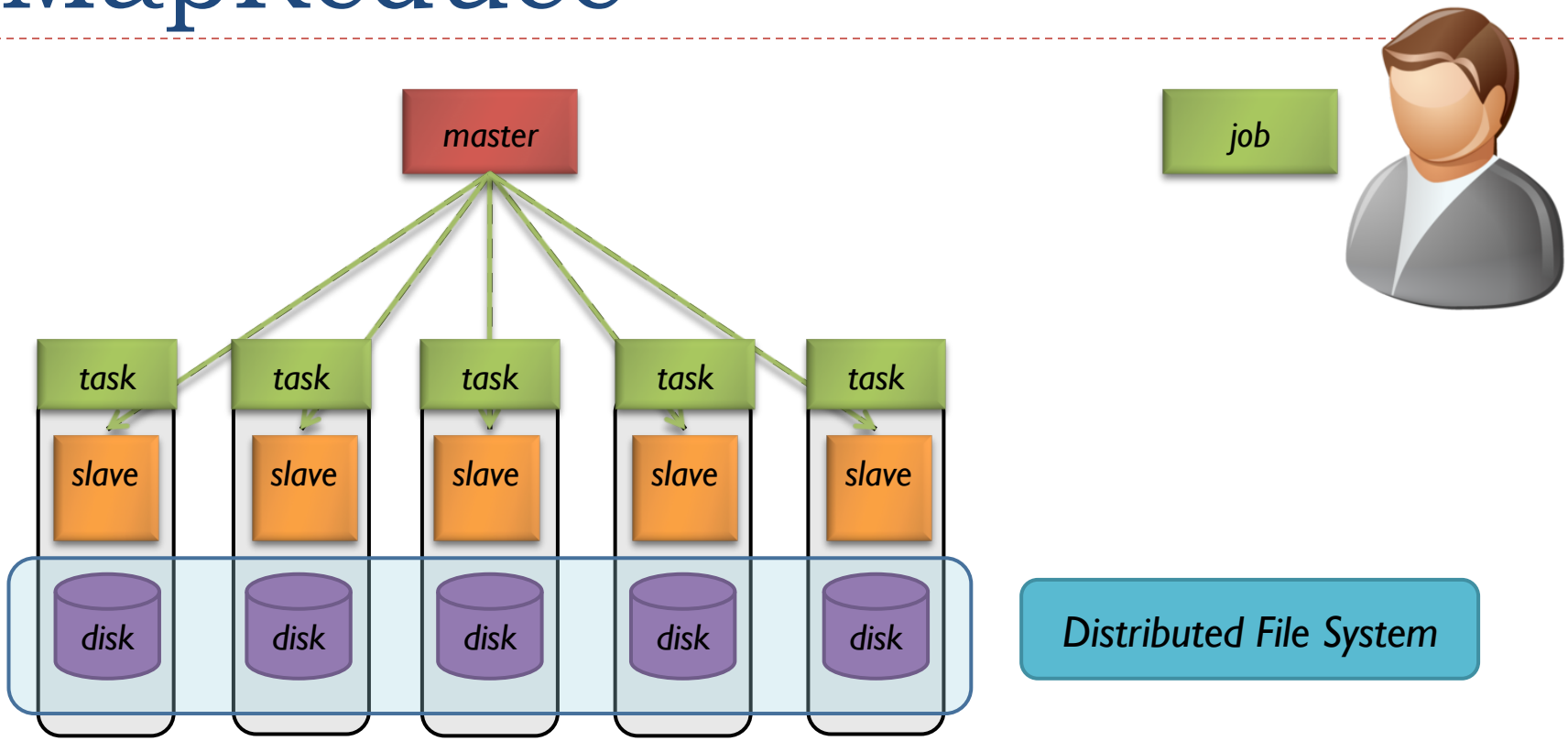


Management by Whom?

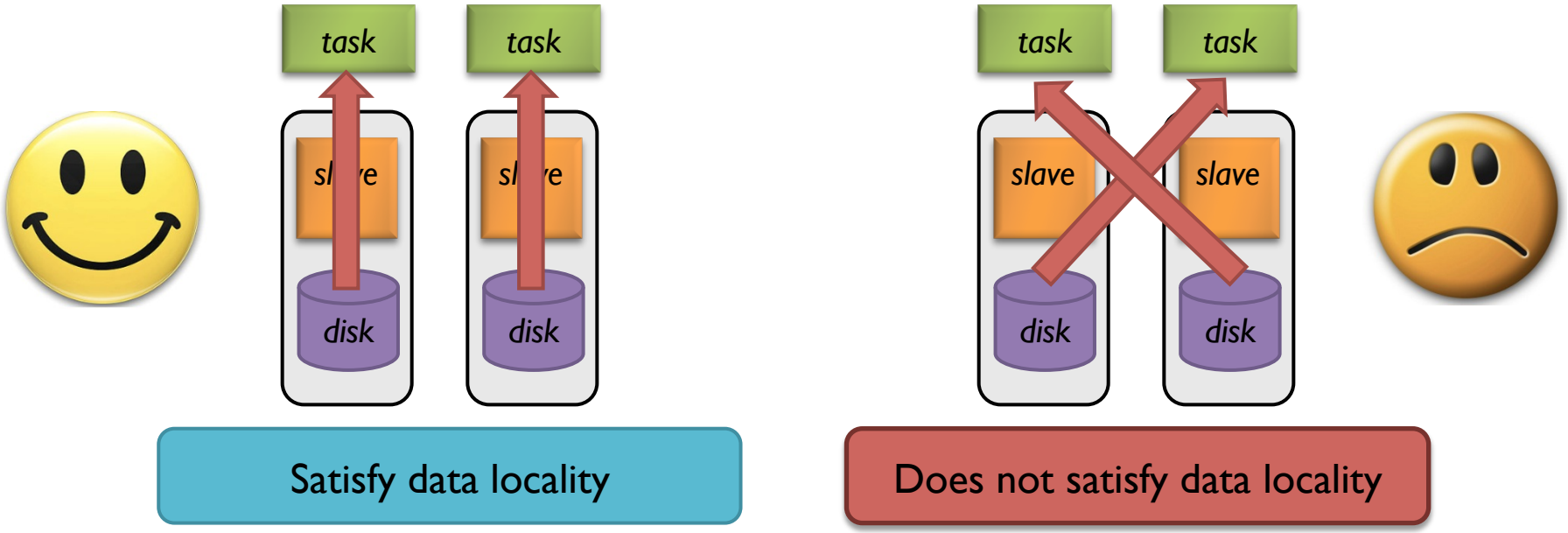
- Requirements
  1. Current resource utilization monitoring
  2. Platform-level information
  3. Privileged permission to hot-plug resource
  4. Support management for multiple users

- Resource management as Platform-as-a-Service (PaaS) service
  - Provider offers platform with dynamic resource management for various users
  - e.g. Amazon Elastic MapReduce

MapReduce


master
job
task
task
task
task
task
slave
slave
slave
slave
slave
disk
disk
disk
disk
disk
Distributed File System

Data Locality on MapReduce

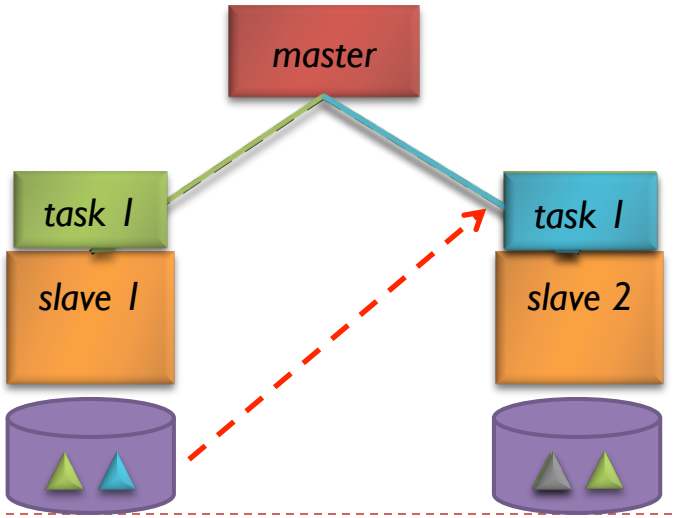


- Disadvantages from low data locality
  1. Network performance degradation because of netwo rk bottleneck
  2. Under-utilization of computing resource

Hadoop Fair Scheduler

- Hadoop
  - Open source implementation of MapReduce
- Hadoop Fair Scheduler
  - Generally used scheduler
  - Guarantee fairness between submitted jobs on Hadoop

Job	# Running tasks
job A	2
job B	0

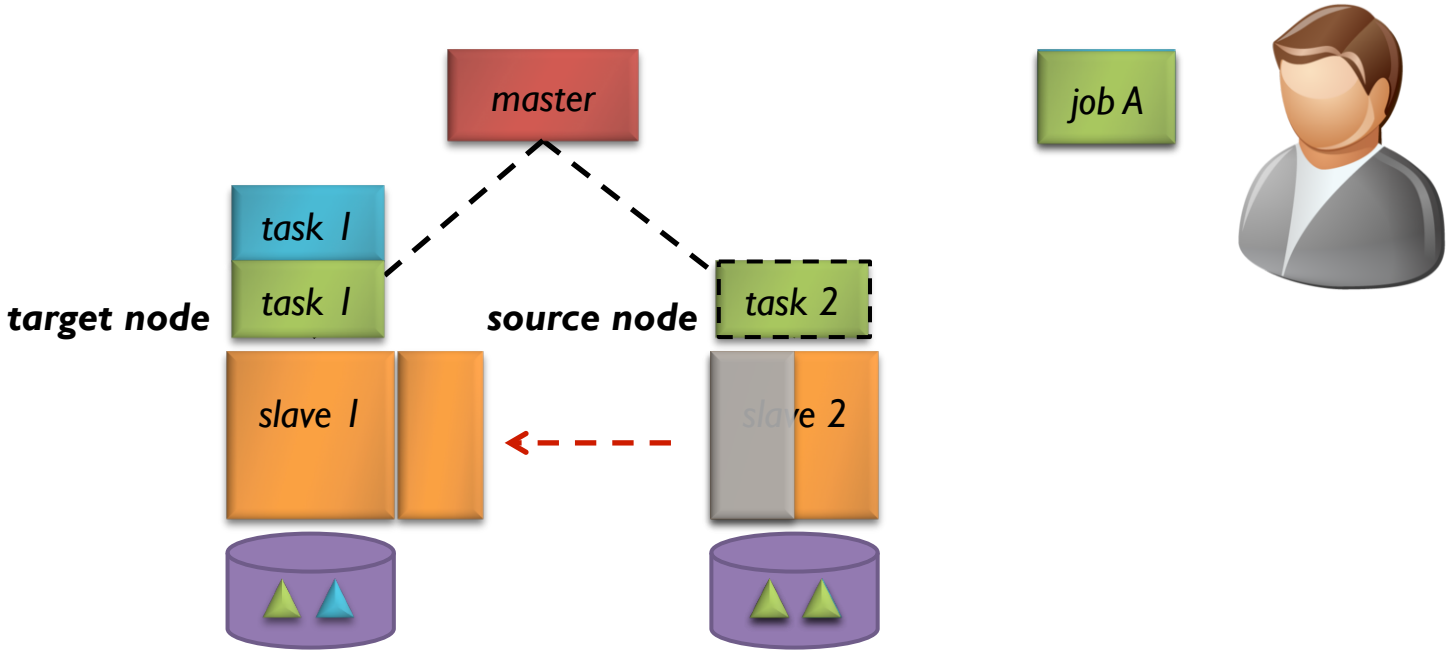


Main Idea

- Approach
  - Move available resource to a node satisfying data localit y and assign a task to the node

Dynamic Resource Reconfiguration

Dynamic Resource Reconfiguration



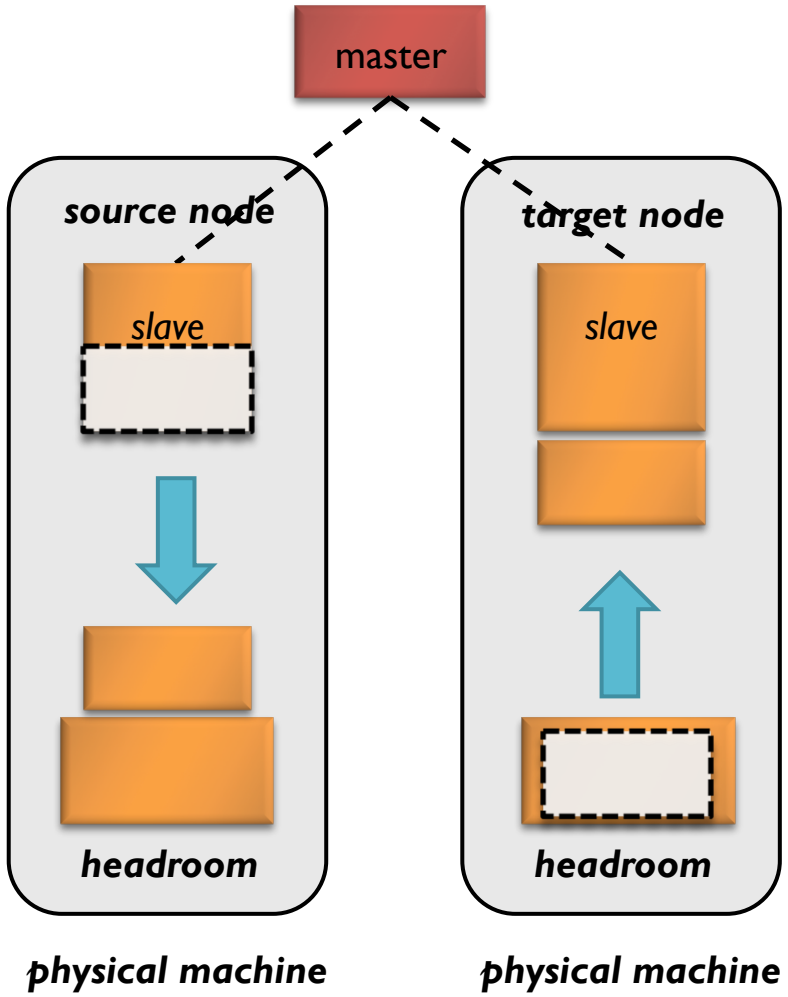
1. A node(***source node***) does not satisfy data locality
2. Master schedule to another node(***target node***) satisfying data locality
3. Reconfigure both source and target nodes

Dynamic Resource Reconfiguration

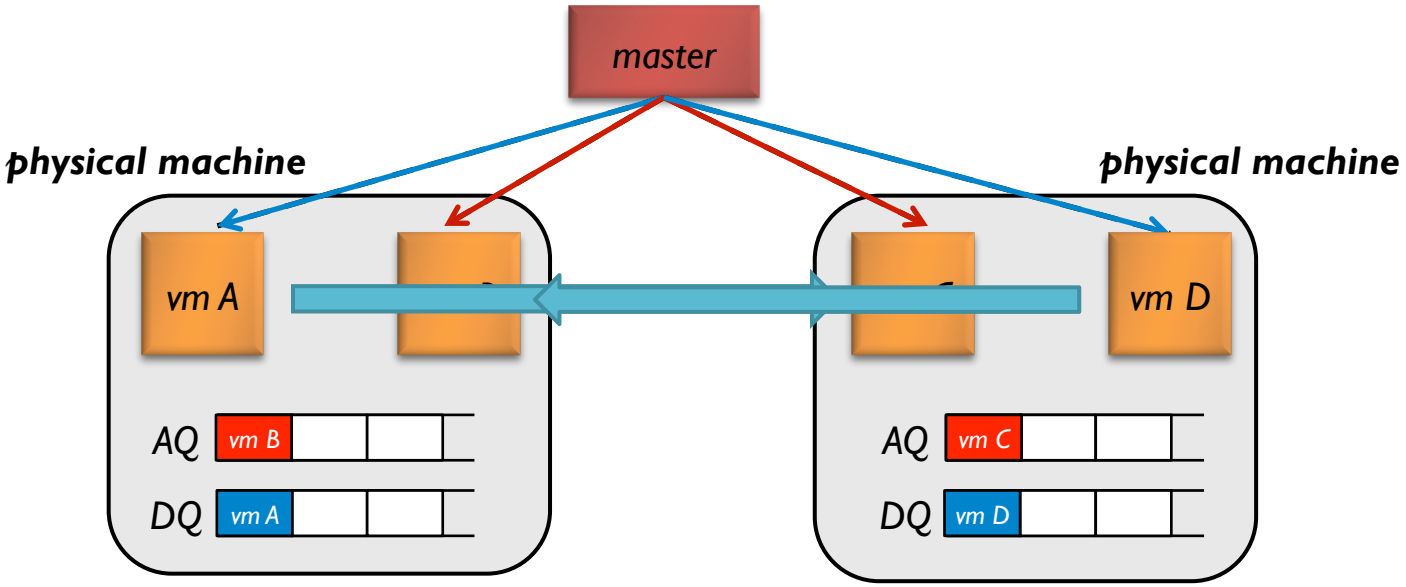
- Resource hot-plugging
  - De-allocation
    - Giving up and giving back resource to provider
    - Always possible
  - Allocation
    - Taking new resource from provider
    - Not always possible
- Two solutions
  - Synchronous DRR
  - Queue-based DRR

Synchronous DRR

- Headroom
  - Remained by provider
  - Idle and available resource on each physical machine
  - Shared by all VMs on a physical machine

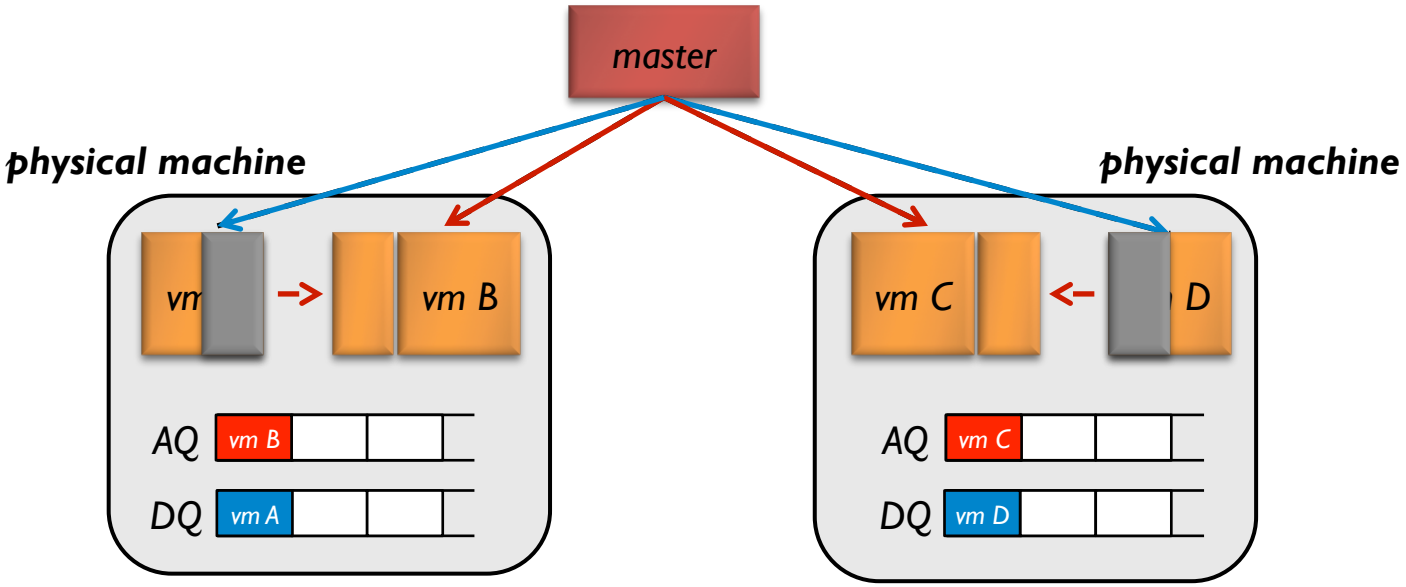


Queue-based DRR



1. Reconfiguration from *vm A* to *vm C*
2. Reconfiguration from *vm D* to *vm B*

Queue-based DRR



1. Reconfiguration from *vm A* to *vm C*
2. Reconfiguration from *vm D* to *vm B*
3. Reconfigure (*vm A*, *vm B*) and (*vm C*, *vm D*)

Synchronous vs. Queue-based

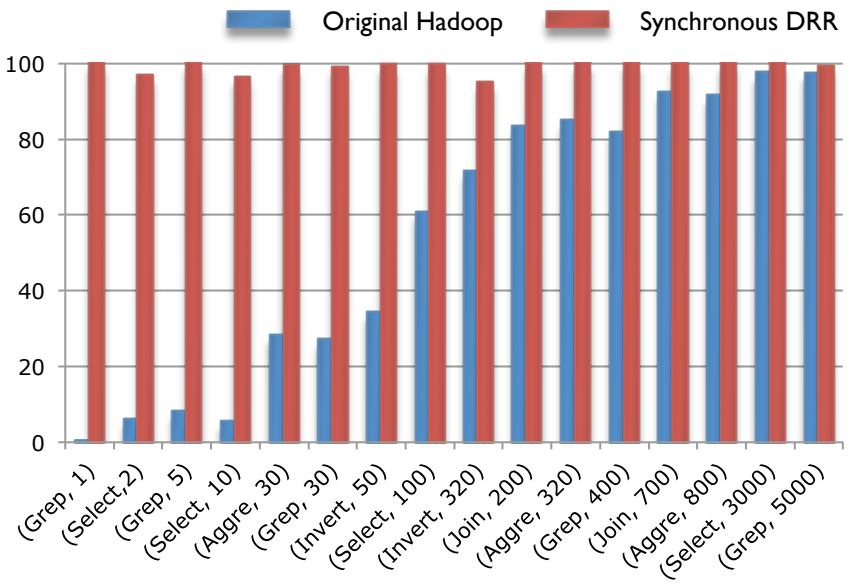
- Synchronous DRR
  - No waiting time until reconfiguration
  - Synchronously executed allocation and deallocation
  - Overall resource under-utilization because of headroom
- Queue-based DRR
  - Realistic and industry-applicable mechanism
  - Performance degradation if queuing delay is large

Evaluation

- Environment
  - EC2 cluster: 100 VM instances
    - 8 virtual cores, 7 GB memory (High-CPU Extra Large Instance)
    - Synchronous DRR only
  - Private cluster: 30 VMs on 6 physical machines
    - 6 cores, 16GB memory
    - Synchronous DRR + Queue-based DRR
- Workloads
  - Hive performance benchmark
    - grep, select, join, aggregation, inverted index
- Job schedule
  - Randomly generated schedule based on the trace of the ind ustry [Eurosys'10]

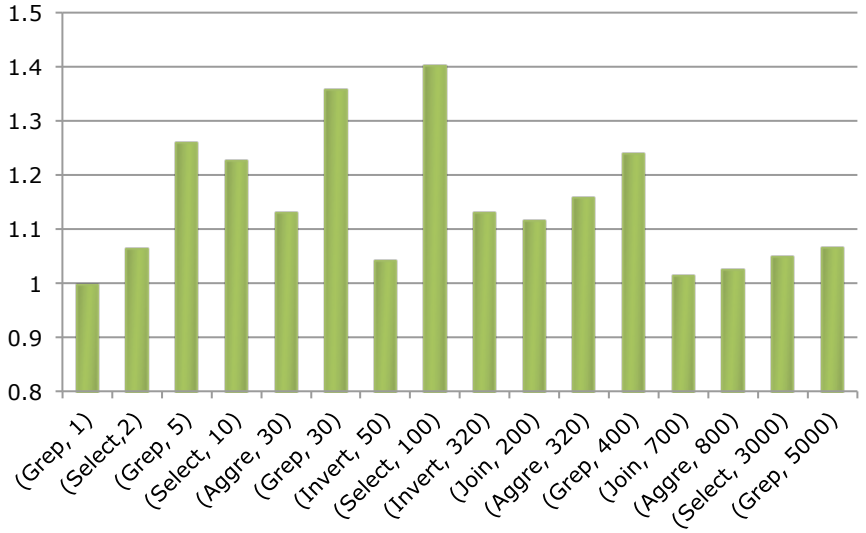
Large-scale Evaluation

Locality (%)



(Workloads, # of map tasks)

Speed-up

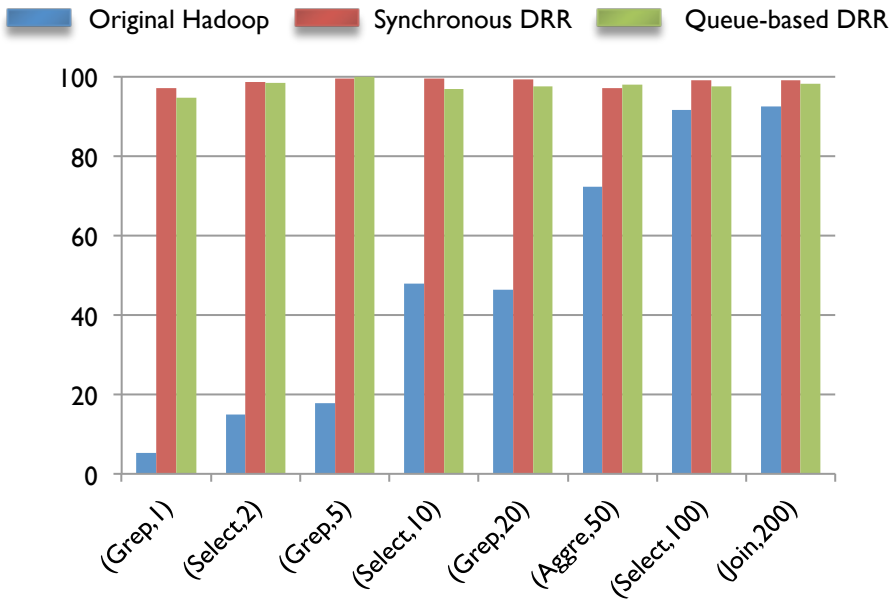


(Workloads, # of map tasks)

- Overall speedup : 15%

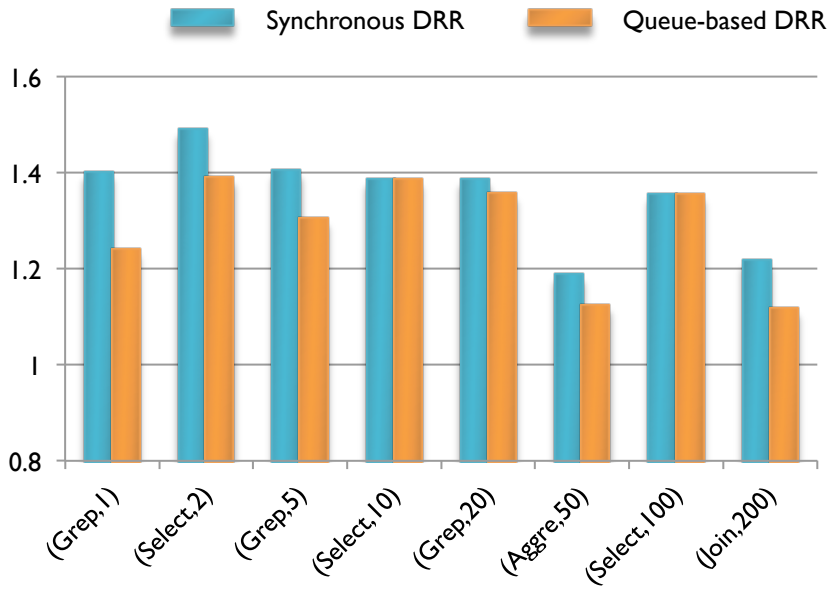
Evaluation on the Private Cluster

Locality (%)



(Workloads, # of map tasks)

Speed-up



(Workloads, # of map tasks)

- Overall speedup
  - Synchronous DRR : 41%
  - Queue-based DRR : 35%

Conclusion

- Propose a dynamic VM reconfiguration mechanism for distributed data-intensive platforms on virtual ized cloud environment

- Improve the input data locality of a virtual MapRe duce cluster, by temporarily increasing cores to V Ms to run local tasks, and it is called Dynamic Res ource Reconfiguration (DRR)