A Novel Scheduling Mechanism For Hybrid Cloud Systems

Haoran Wei [1] Fanchao Meng [2]

[1]Department of Computer and Information Sciences
University of Delaware
nancywhr@udel.edu

[2]Department of Computer and Information Sciences
University of Delaware
fcmeng@udel.edu

Hybrid Cloud Systems

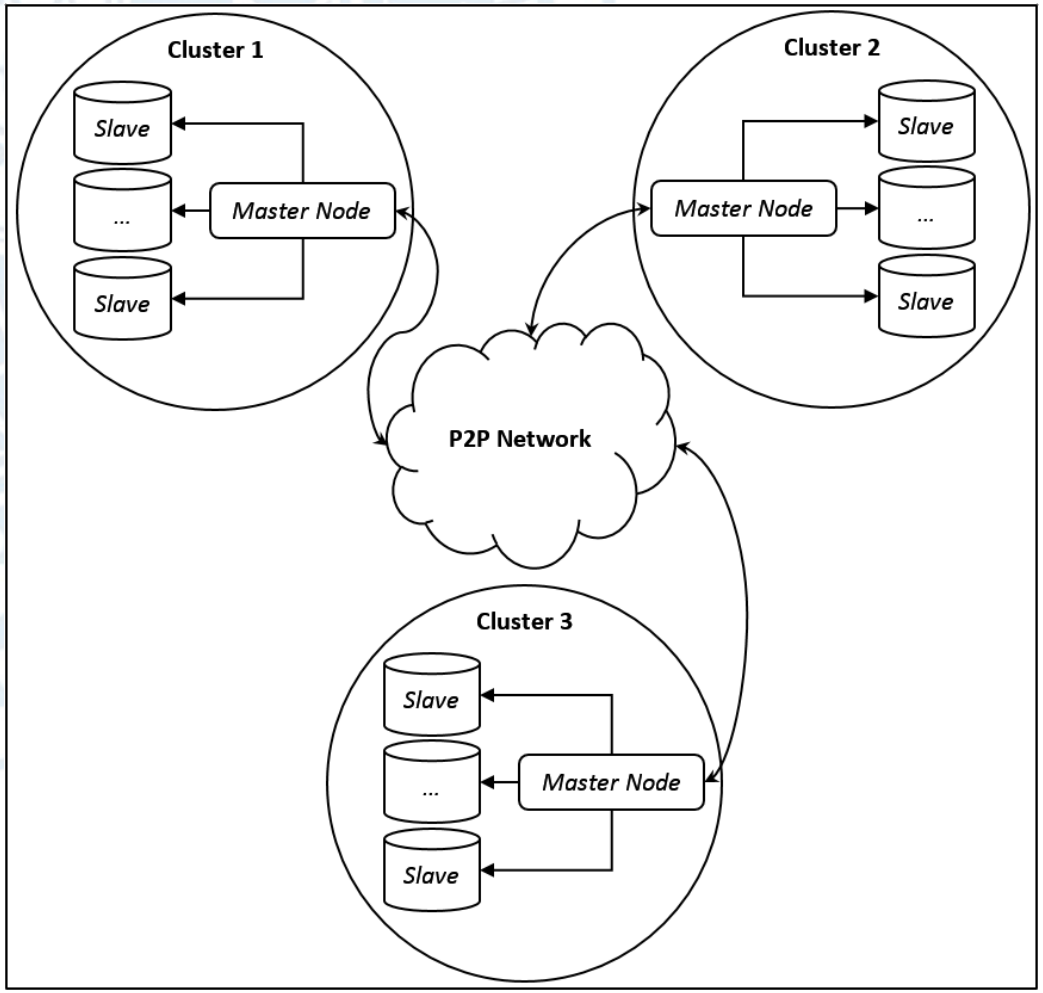


Figure 1: Hybrid Cloud Systems

System Overview

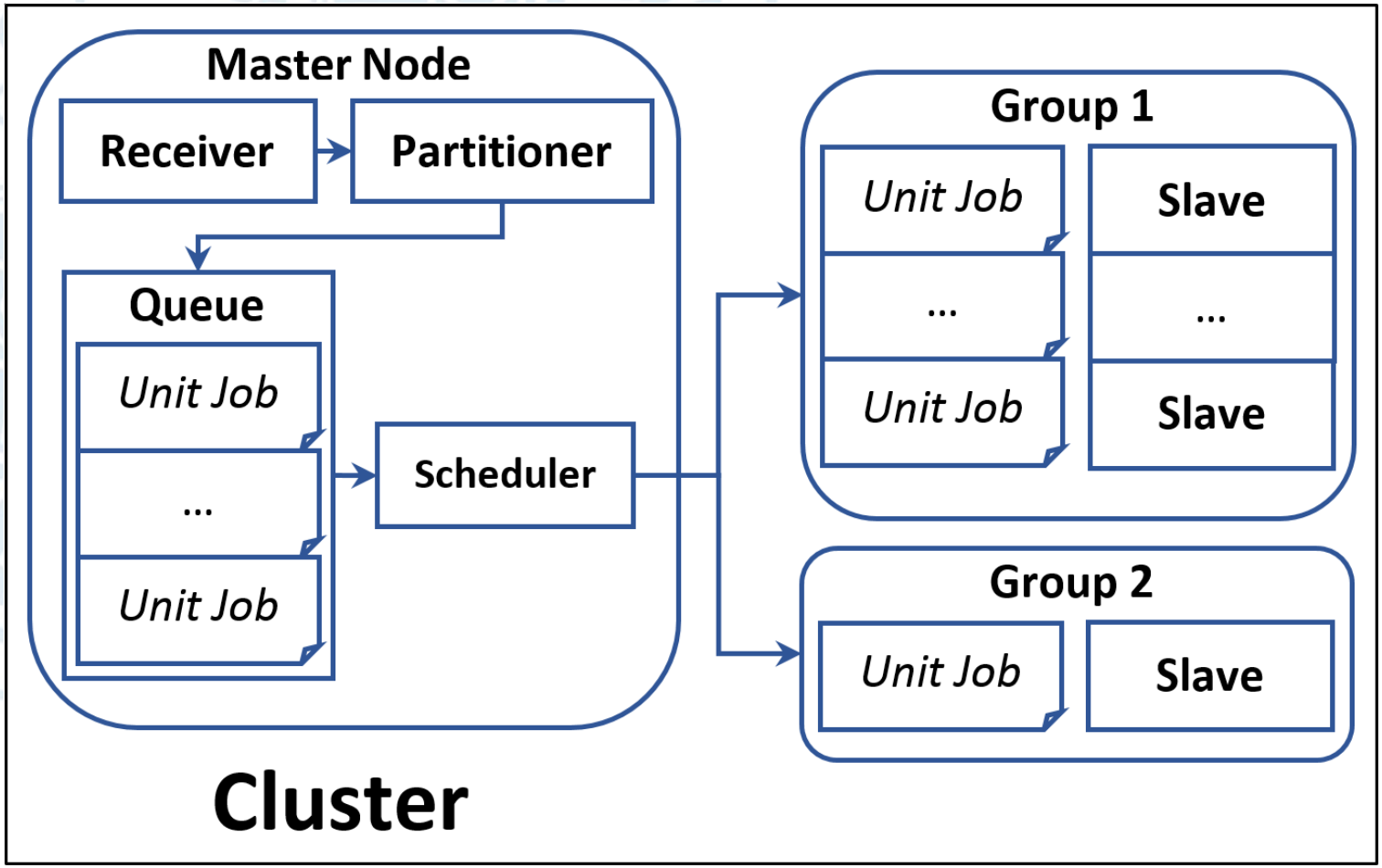


Figure 2: The framework of cluster.

System Overview (Cont.)

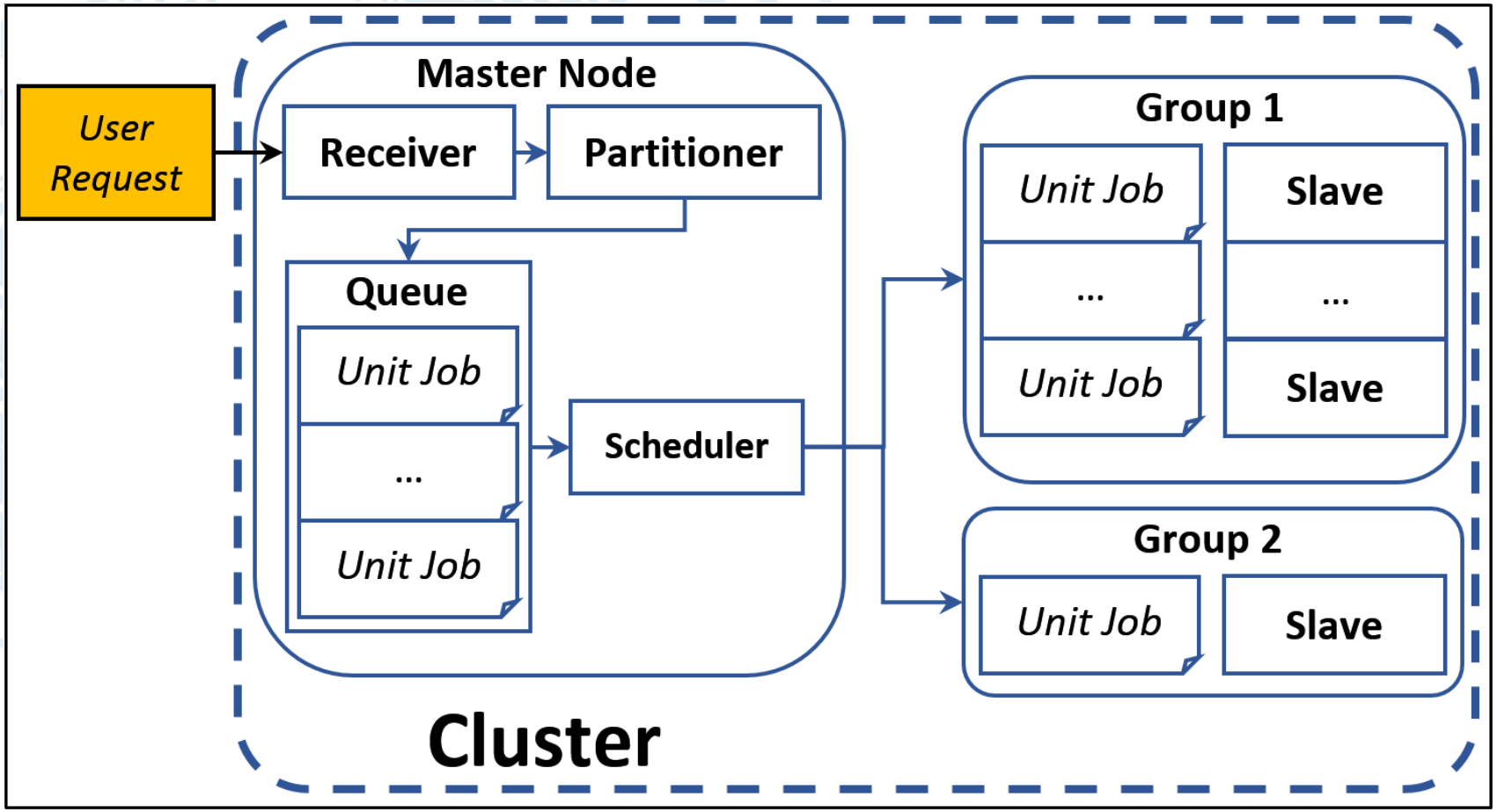


Figure 3: User Request.

System Overview (Cont.)

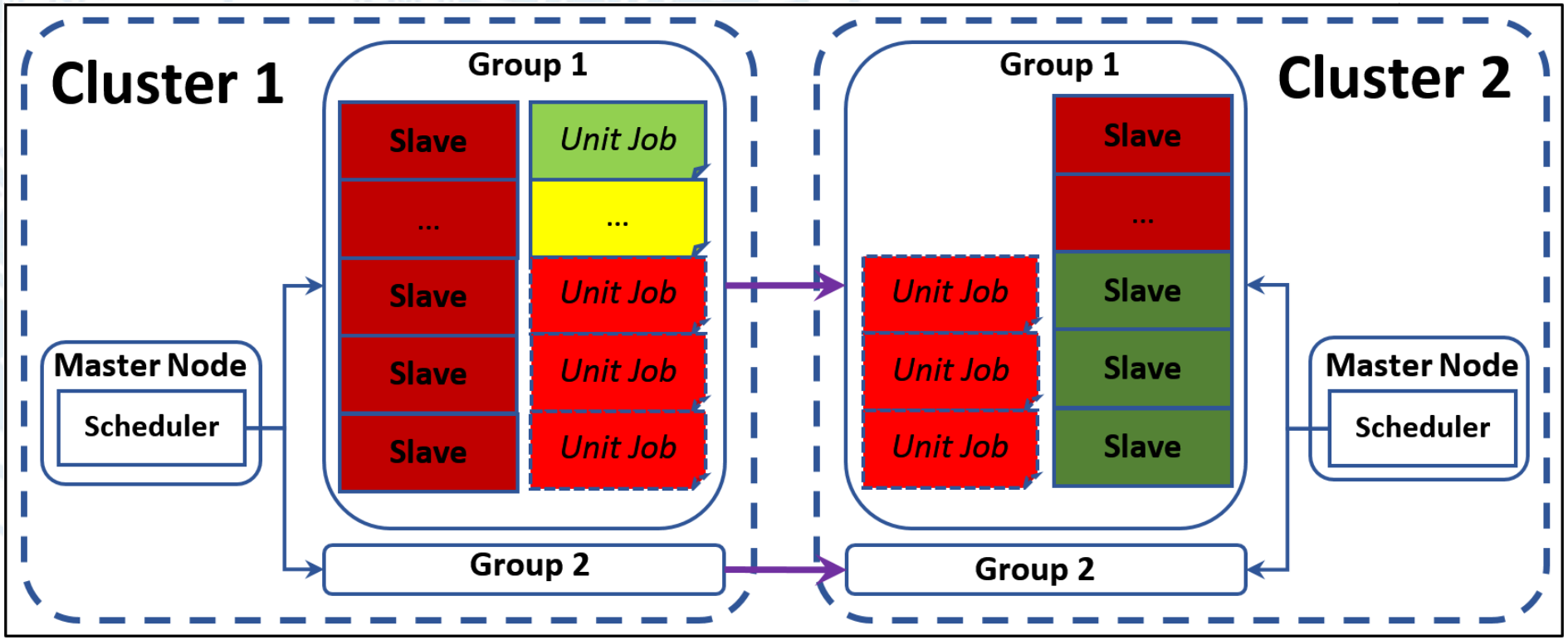


Figure 4: Unit job migration between clusters.

Measures

Makespan

Total elapsed time from entering the system to finishing/leaving.

Expected (Ideal) Makespan

Actual Makespan

Estimated Makespan

Elapsed time already spent in the system + estimated time remaining in the system.

Measures (Cont.)

Actual Makespan Ratio

The ratio of actual makespan to ideal makespan. It measures how efficiently a given unit job/user request is processed by the system.

Estimated Makespan Ratio

The ratio of estimated makespan to ideal makespan.

Baseline (Average) Makespan Ratio

Given a sequence of user request completions in a cluster, this ratio is the weighted average of their actual makespan ratios, where the weights are normalized lengths.

Measures (Cont.)

Estimated Baseline Makespan Ratio Elevation (EBMRE)

The weighted average of the baseline makespan ratio and the sum of the estimated makespan ratios of queued unit jobs in a given group. It measures how much the baseline makespan ratio would rise if all unit jobs in this group are completed here (i.e. no job is moved out).

System Performance & Objective

System Performance

Average actual user request makespan.

System Performance & Objective

System Performance

Average actual user request makespan.

One Principle

Less waiting $\Longrightarrow$ Better performance

System Performance & Objective

System Performance

Average actual user request makespan.

One Principle

Less waiting $\Longrightarrow$ Better performance

Objective of the Scheduling Mechanism

Minimize the amount of waiting.

System Performance & Objective

System Performance

Average actual user request makespan.

One Principle

Less waiting $\Longrightarrow$ Better performance

Objective of the Scheduling Mechanism

Minimize the amount of waiting.

General Idea

Migrate some unit jobs to somewhere to make them wait less.

Core Problem

When should we migrate **whom** to **where**?

Analysis

Why Might Waiting Be Reduced?

- Groups in the same cluster are not load-balanced relatively to each other.

Analysis

Why Might Waiting Be Reduced?

- Groups in the same cluster are not load-balanced relatively to each other.

Solution

- Intra-Cluster Load Balancing.
- Single Unit Job Scheduling (for high-priority unit jobs).

Analysis

Why Might Waiting Be Reduced?

- Groups in the same cluster are not load-balanced relatively to each other.

Solution

- Intra-Cluster Load Balancing.
- Single Unit Job Scheduling (for high-priority unit jobs).

- The cluster and its neighbor clusters are not load-balanced.

Analysis

Why Might Waiting Be Reduced?

- Groups in the same cluster are not load-balanced relatively to each other.

Solution

- Intra-Cluster Load Balancing.
- Single Unit Job Scheduling (for high-priority unit jobs).

- The cluster and its neighbor clusters are not load-balanced.

Solution

- Inter-Cluster Load Balancing.

Analysis (Cont.)

Key Fact

The increasing of unit jobs' waiting time will worsen the system performance.

Analysis (Cont.)

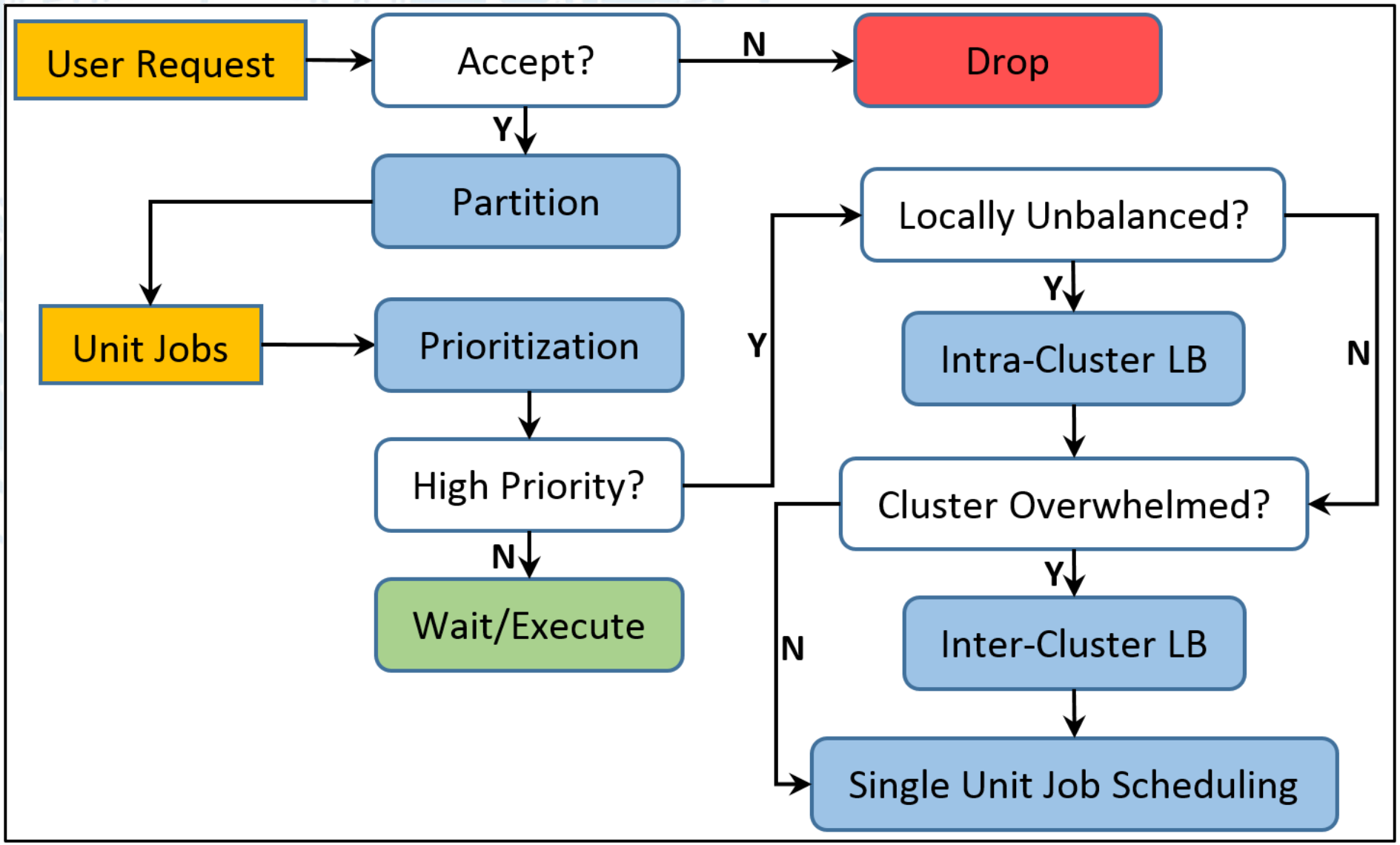
Key Fact

The increasing of unit jobs' waiting time will worsen the system performance.

Prioritization

The priority of a unit job measures how much its waiting would worsen the system performance. It is dynamically updated, (i.e. the longer the waiting time the higher the priority). Furthermore, high priority is telling that the corresponding group might have been overwhelmed.

General Scheduling Mechanism



Two Important Indicators

$\hat{N}^k_{mig}$ and $\hat{N}^k_{acc}$

$\hat{N}^k_{mig}$: the estimated number of unit jobs could be migrated out of group k.

$\hat{N}^k_{acc}$: the estimated number of unit jobs at most group k could accept from other groups.

Two Important Indicators

$\hat{N}^k_{mig}$ and $\hat{N}^k_{acc}$

$\hat{N}^k_{mig}$: the estimated number of unit jobs could be migrated out of group k.

$\hat{N}^k_{acc}$: the estimated number of unit jobs at most group k could accept from other groups.

How to calculate?

Calculated from the EBMRE value for the group.

Conditions in General Scheduling Mechanism

*"**When** should we migrate whom to where?"*

High Priority?

The priority of a given unit job is higher or equal to Γ_{pri}

Conditions in General Scheduling Mechanism

*"**When** should we migrate whom to where?"*

High Priority?

The priority of a given unit job is higher or equal to Γ_{pri}

Locally Unbalanced?

High-priority unit job exists and $\sum_k \hat{N}^k_{mig} + \sum_k \hat{N}^k_{acc} \geq \Gamma_{ll}$

Conditions in General Scheduling Mechanism

*"**When** should we migrate whom to where?"*

High Priority?

The priority of a given unit job is higher or equal to Γ_{pri}

Locally Unbalanced?

High-priority unit job exists and $\sum_k \hat{N}^k_{mig} + \sum_k \hat{N}^k_{acc} \geq \Gamma_{ll}$

Cluster Overwhelmed?

Locally balanced and $\sum_k \hat{N}^k_{mig} - \sum_k \hat{N}^k_{acc} \geq \Gamma_{gl}$

Intra-Cluster Load Balancing

"When should we migrate whom to **Where***?"*

Migrate unit jobs from $\hat{N}^k_{mig} > 0$ groups to $\hat{N}^k_{acc} > 0$ groups to minimize the sum of EBMREs.

Intra-Cluster Load Balancing

"When should we migrate whom to **Where***?"*

Migrate unit jobs from $\hat{N}^k_{mig} > 0$ groups to $\hat{N}^k_{acc} > 0$ groups to minimize the sum of EBMREs.

"When should we migrate **Whom** *to Where?"*

Unit jobs are waiting and of low priorities.

Inter-Cluster Load Balancing

"When should we migrate whom to **Where***?"*

A cluster having its number of queued unit jobs in a certain group greater than average among its neighbors is considered as *heavy*, otherwise *light*. It migrates unit jobs to its light neighbors if any. The source and the destination groups are the same.

Inter-Cluster Load Balancing

"When should we migrate whom to **Where***?"*

A cluster having its number of queued unit jobs in a certain group greater than average among its neighbors is considered as *heavy*, otherwise *light*. It migrates unit jobs to its light neighbors if any. The source and the destination groups are the same.

"When should we migrate **Whom** *to Where?"*

Unit jobs are waiting and of low priorities.

Single Unit Job Scheduling

"When should we migrate **Whom** *to* **Where***?"*

Place this high priority unit job to a group who will have the minimum increase in its cumulative estimated makespan for adding this job in.

Additional Considerations

Implementation of Inter-Cluster Load Balancing

It is using a Gossip-based P2P communication. A gossip table is maintained by each cluster, and it is used to select neighbors to migrate unit jobs.

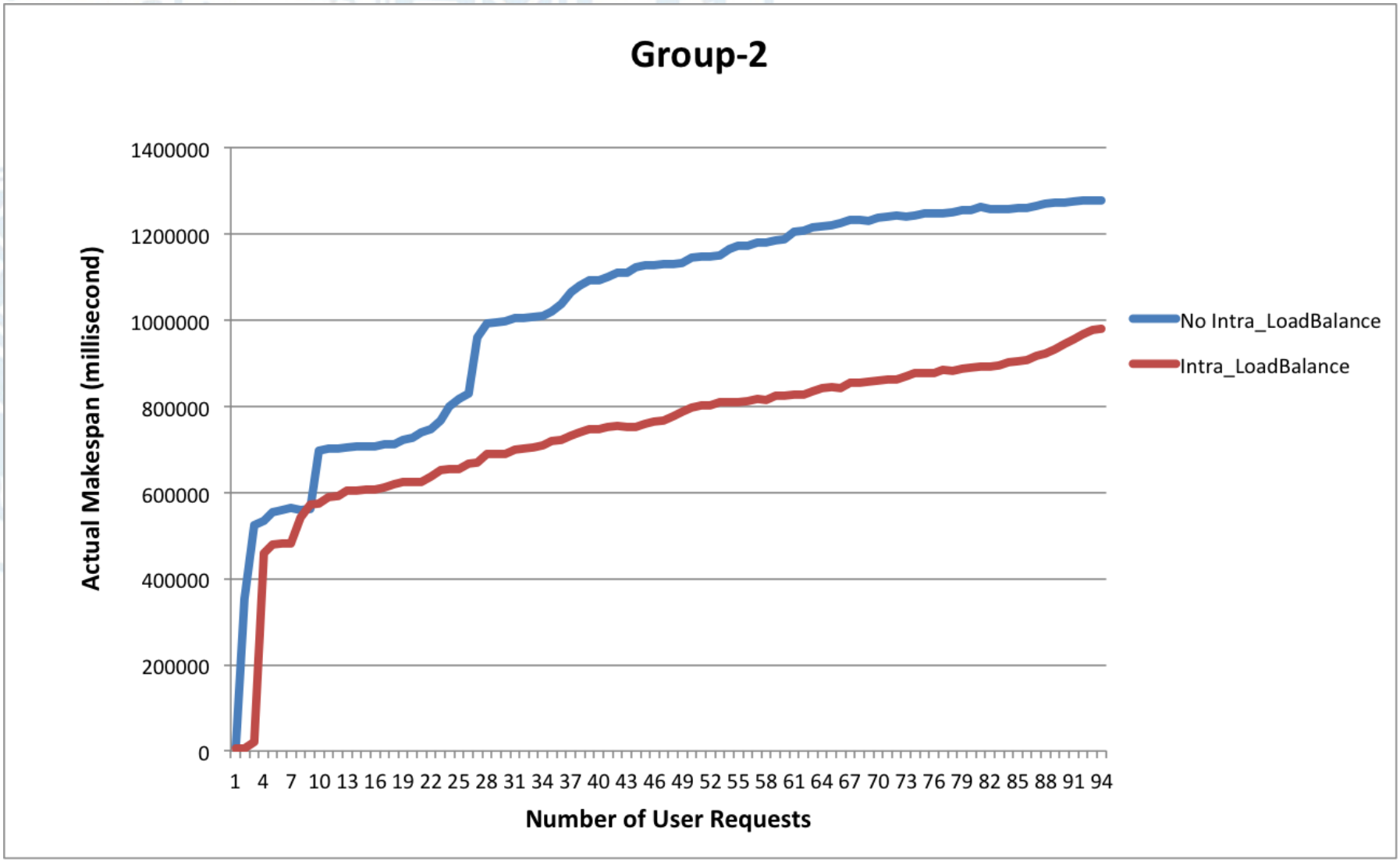
Settings

- Four clusters;
- Four groups in each cluster;

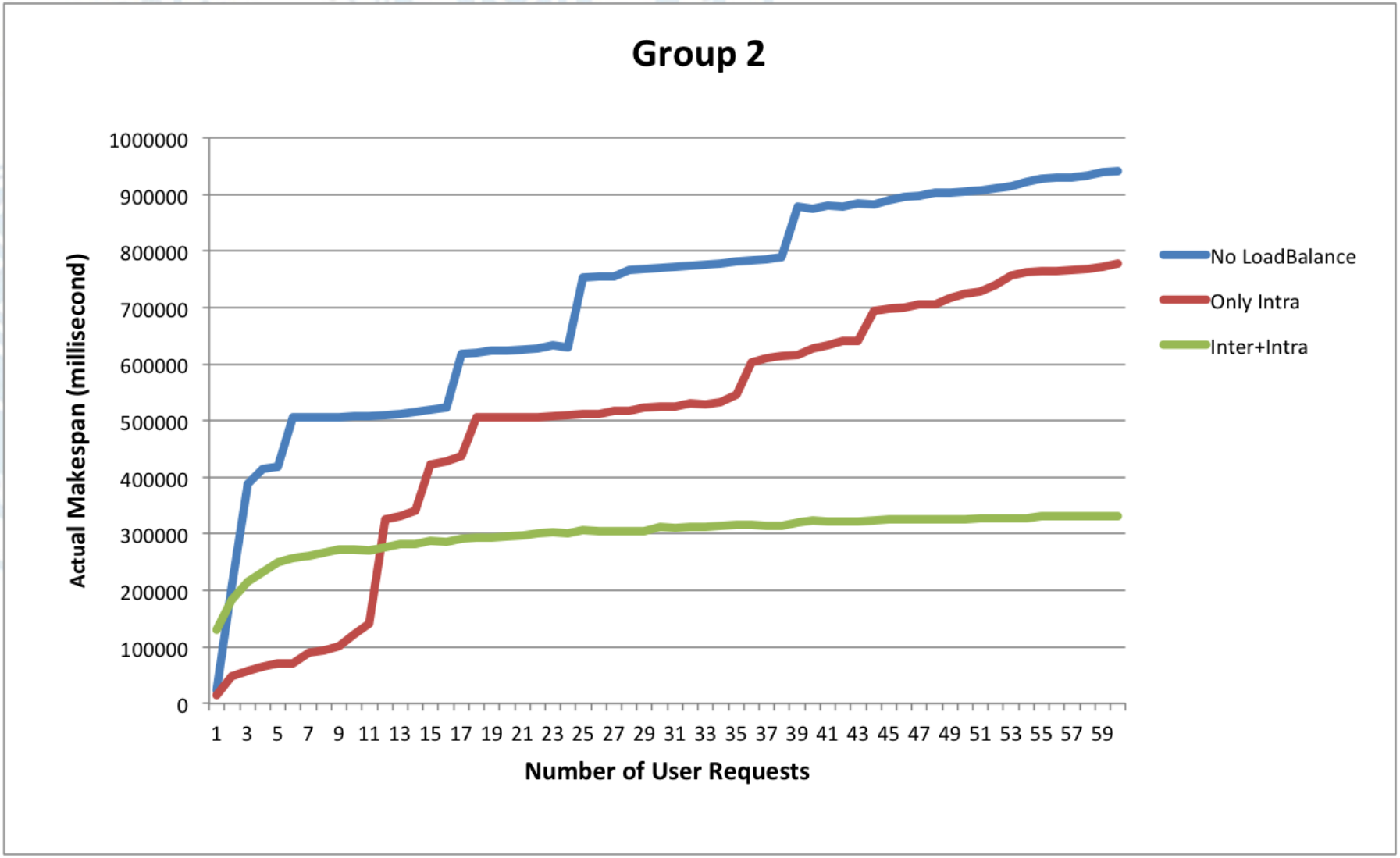
Group	Execution Time(millisecond)
$\mathcal{G}_1$	50
$\mathcal{G}_2$	250
$\mathcal{G}_3$	1250
$\mathcal{G}_4$	6250

- Arrival of user requests follows the Poisson distribution, $\lambda = 20$;
- Number of unit jobs in each user request is randomly picked in range of $[1 \sim 20]$;
- Performance measure is average actual user request makespan.

Experimental Results (Intra-Cluster LB)



Experimental Results (Intra- and Inter-Cluster LB)



Future Work

Pricing & Costs

If the pricing and the costs of different types of computing resources are taken into account, then the problem would become more interesting.

The End