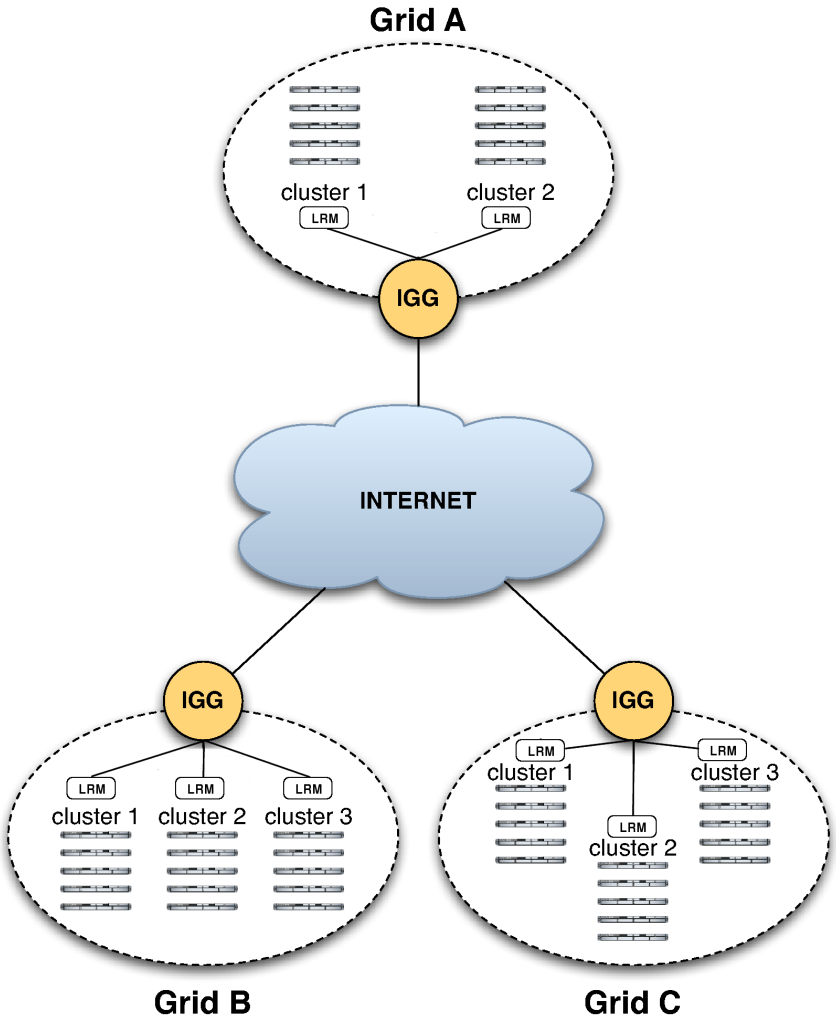
Performance Analysis of Preemption-aware Scheduling in Multi-Cluster Grid Environments

Mohsen Amini Salehi , Bahman Javadi, Rajkumar Buyya
Cloud Computing and Distributed Systems (CLOUDS) Laboratory,
Department of Computer Science and Software Engineering,
The University of Melbourne, Australia
mohsena,bahmanj,raj@csse.unimelb.edu.au

Introduction: InterGrid

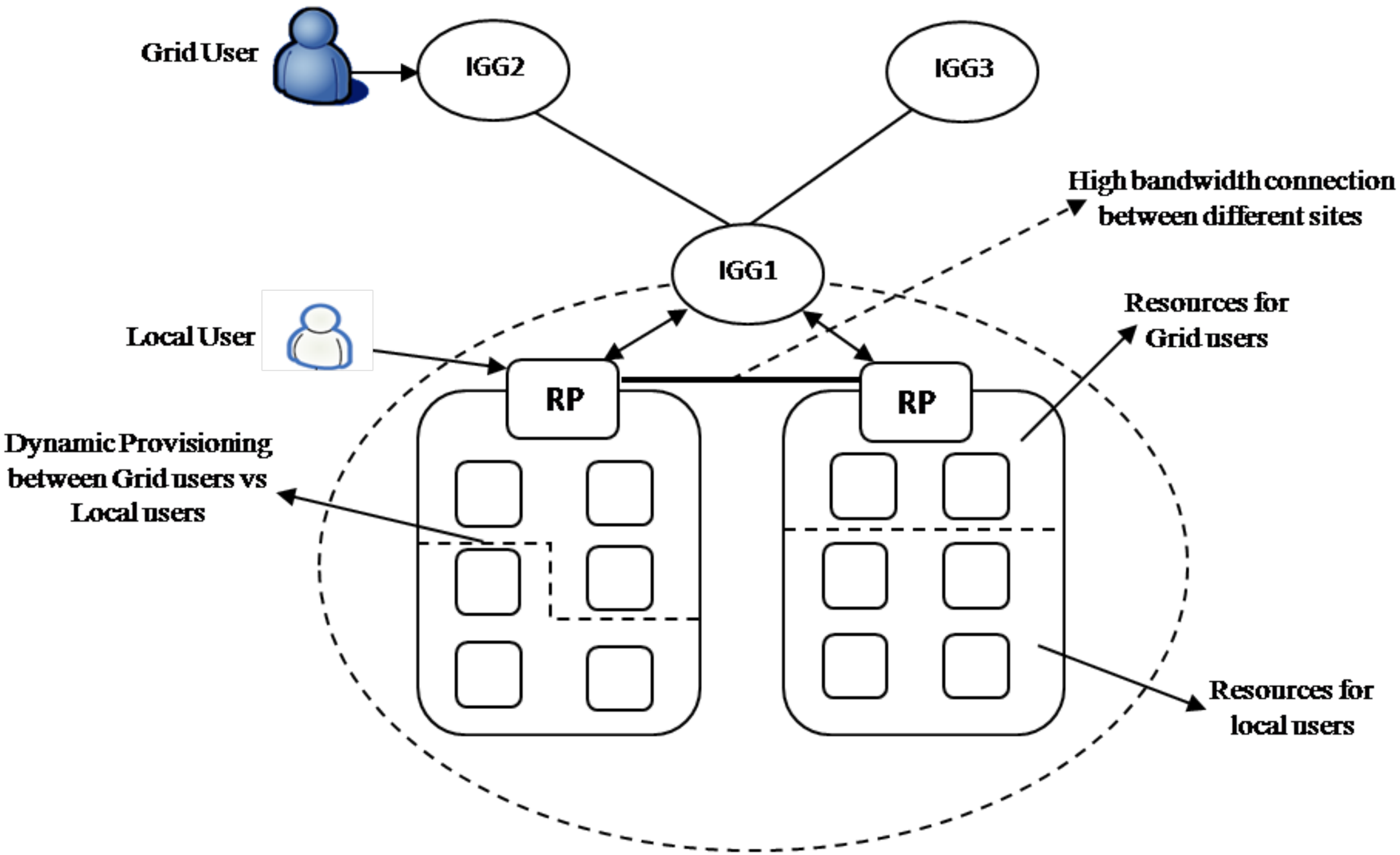


- provides an architecture and policies for inter-connecting different Grids.
- Computational resources in each RP are shared between grid users and local users.
- Provisioning rights of the resources in a Grid are delegated to the InterGrid Gateway (IGG).
- Local users vs External users.
  - Local users have priority!

Contention between Local and External users

- When contention happens?
  - Lack of resource
- Solution for Contention:
  - ***Preemption of Ext. requests in favor of local requests***
- Drawbacks of Preemption:
  - overhead to the underlying system (degrades utilization)
  - increases the response time of the grid requests

Contention Scenario in InterGrid

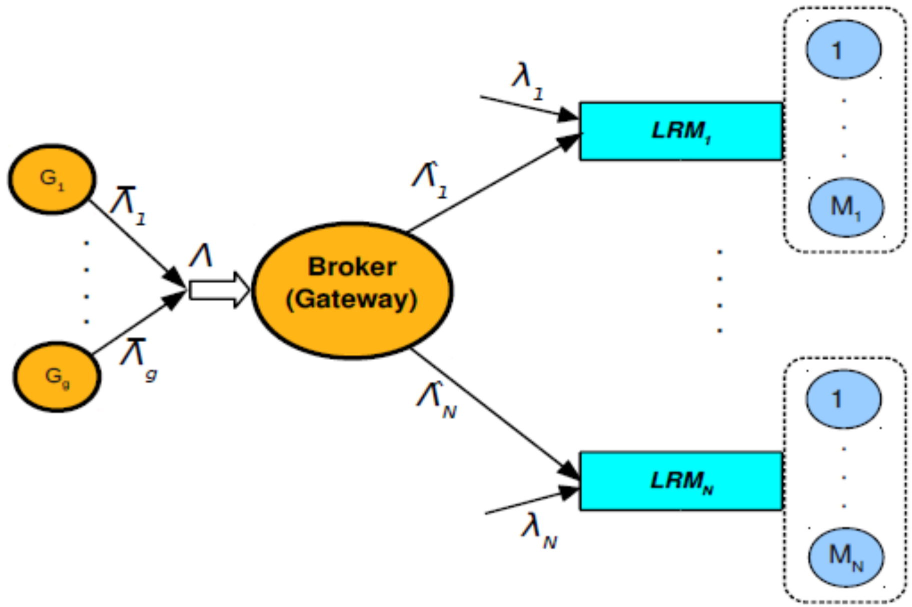


Lease based Resource Provisioning in InterGrid

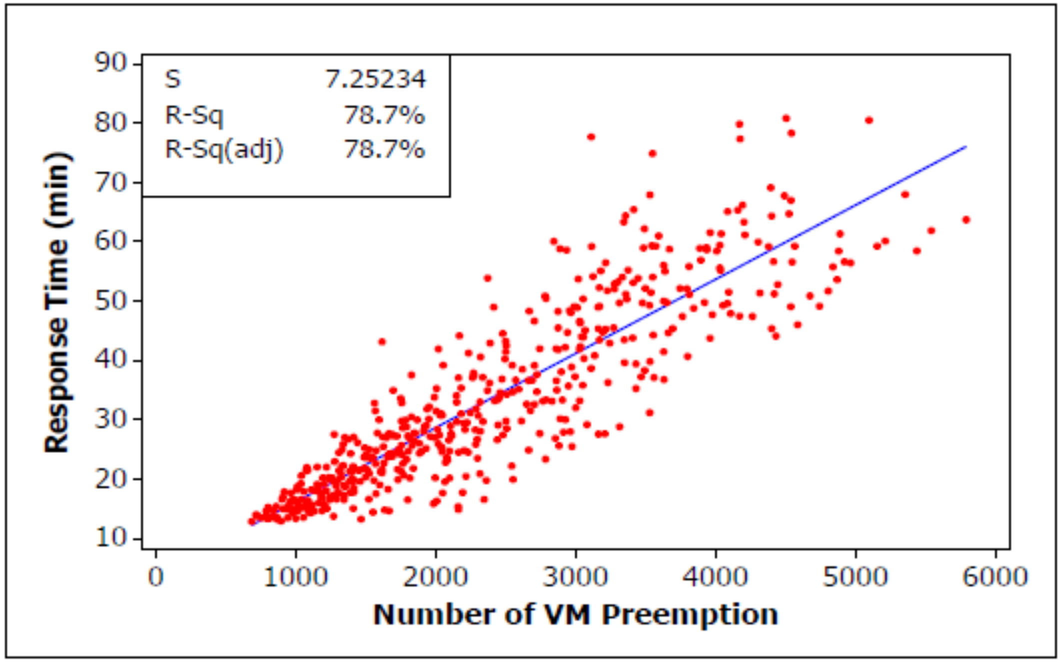
- *A lease is an agreement between resource provider and resource consumer whereby the provider agrees to allocate resources to the consumer according to the lease terms presented.*
- Virtual Machine (VM) technology is a way to implement lease-based resource provisioning.
- VMs are able to get suspended, resumed, stopped, or even migrated.
- InterGrid makes one lease for each user request.
  - Best-Eort (BE)
    - Cancelable
    - Suspendable
  - Deadline-Constraint (DC)
    - Migratable
    - Non-preemptive

Problem Statement

- How to decrease the number of preemptions that take place in a multi-cluster Grid?
  - Analytical modeling of preemption in a multi-cluster Grid environment based on routing in parallel queues



Analysis: Correlation of Response time and Number of preemption



- Therefore, if we decrease response time the number of preemption also decreased!

Analysis: Minimize Response Time

$$T = \frac{1}{\Lambda}\sum_{j=1}^{N}\hat{\Lambda}_j \cdot T_j$$

Where the constrain is:

$$\hat{\Lambda}_1 + \hat{\Lambda}_2 + ... + \hat{\Lambda}_N - \Lambda = 0.$$

Analysis: Optimal arrival Rate to each Cluster

Response time of Ext. requests for each cluster j (M/G/1 queue with preemption):

$$T_j = \frac{1}{1-\rho_j}\left(\theta_j + \frac{\kappa_j m_j}{2(1-u_j)}\right)$$

The input arrival rate to each cluster by using Lagrange multiplier:

$$\hat{\Lambda}_j = \frac{(1-\rho_j)}{\theta_j} - \frac{1}{\theta_j}\sqrt{\frac{(1-\rho_j)(\omega_j(1-\rho_j)) + \theta_j\lambda_j\mu_j}{2\theta_j(1-\rho_j)z + (\omega_j - 2\theta_j^2)}}$$

where z is the Lagrange multiplier

Analysis: Finding out Lagrange multiplier (z)

Since we have:

$$\Lambda = \hat{\Lambda}_1 + \hat{\Lambda}_1 + \ldots + \hat{\Lambda}_N,$$

Z can be worked out from the below equation:

$$\sum_{j=1}^{N} \frac{1}{\theta_j} \sqrt{\frac{(1-\rho_j)(\omega_j(1-\rho_j)) + \theta_j \lambda_j \mu_j}{2\theta_j(1-\rho_j)z + (\omega_j - 2\theta_j^2)}} = \left(\sum_{j=1}^{N} \frac{(1-\rho_j)}{\theta_j} \right) - \Lambda$$

Analysis: Using bisection algorithm for finding z

Since:

$$\hat{\Lambda}_j \geq 0$$

Then:

$$z \geq \frac{\lambda_j \mu_j}{2(1-\rho_j)^2} + \frac{\theta_j}{(1-\rho_j)}$$

$$\sum_{j=1}^{N} \phi_j(lb) \geq \left(\sum_{j=1}^{N} \frac{(1-\rho_j)}{\theta_j} \right) - \Lambda$$

upper bound also can be worked out as follows:

$$\sum_{j=1}^{N} \phi_j(ub) \leq \left(\sum_{j=1}^{N} \frac{(1-\rho_j)}{\theta_j} \right) - \Lambda$$

Algorithm 1: Preemption-Aware Scheduling Policy (PAP).

Input: $\bar{\Lambda}_j, \theta_j, \omega_j, \lambda_j, \tau_j, \mu_j$, for all $1 \leq j \leq N$.
Output: $(\hat{\Lambda}_j)$ load distribution of grid requests to different clusters, for all $1 \leq j \leq N$.

for $j \leftarrow 1$ **to** N **do**
  $\psi_j = \frac{\lambda_j \mu_j}{2(1-\rho_j)^2} + \frac{\theta_j}{(1-\rho_j)}$;
`Sort` (ψ);
$k \leftarrow 1$;
while $k < N$ **do**
  if $\sum_{j=1}^{k} \phi_j(\psi_k) \geq \left(\sum_{j=1}^{k} \frac{(1-\rho_j)}{\theta_j} \right) - \Lambda$ **then**
    break;
  else
    $k \leftarrow k + 1$;
$lb \leftarrow \psi_k$;
$ub = 2 * lb$;
while $\sum_{j=1}^{k} \phi_j(ub) > \left(\sum_{j=1}^{k} \frac{(1-\rho_j)}{\theta_j} \right) - \Lambda$ **do**
  $ub = 2 * ub$;
while $ub - lb > \epsilon$ **do**
  $z \leftarrow (lb + ub)/2$;
  if $\sum_{j=1}^{k} \phi_j(z) \geq \left(\sum_{j=1}^{k} \frac{(1-\rho_j)}{\theta_j} \right) - \Lambda$ **then**
    $lb \leftarrow z$;
  else
    $ub \leftarrow z$;
for $j \leftarrow 1$ **to** k **do**
  $\hat{\Lambda}_j = \frac{(1-\rho_j)}{\theta_j} - \frac{1}{\theta_j} \sqrt{\frac{(1-\rho_j)(\omega_j(1-\rho_j)) + \theta_j \lambda_j \mu_j}{2\theta_j(1-\rho_j)z + (\omega_j - 2\theta_j^2)}}$;
for $j \leftarrow k + 1$ **to** N **do**
  $\hat{\Lambda}_j = 0$;

Preemption-aware Scheduling Policy

- The analysis provided is based on the following assumption:
  - each cluster was an M/G/1 queue. However, in InterGrid we are investigating each cluster as a $G/G/M_j$ queue.
  - all requests needed one VM. InterGrid requests need several VMs for a certain amount of time.
  - each queue is run in FCFS fashion while we consider conservative backfilling.

Implementation details

- To support several VMs, the service time of ext. and local requests on cluster *j* is calculated:

$$\theta_j = \frac{\bar{v}_j \cdot \bar{d}_j}{M_j s_j} \qquad \tau_j = \frac{\bar{\zeta}_j \cdot \bar{\varepsilon}_j}{M_j s_j}$$

- *CV* is used to obtain second moment:

$$\omega_j = (\alpha_j \cdot \theta_j)^2 + \theta_j^2$$

$$\mu_j = (\beta_j \cdot \tau_j)^2 + \tau_j^2$$

Experiment Set up

- GridSim Simulator
- 3 clusters with 32, 64, and 128 nodes
- Conservative backfilling scheduler as LRM
- 100 Mbps network bandwidth
- Different ext. request types:
  - BE-Suspendable:40% and BE-Cancelable:10%
  - DC-Nonpreemptable:40% and DC-Migratable:10%.

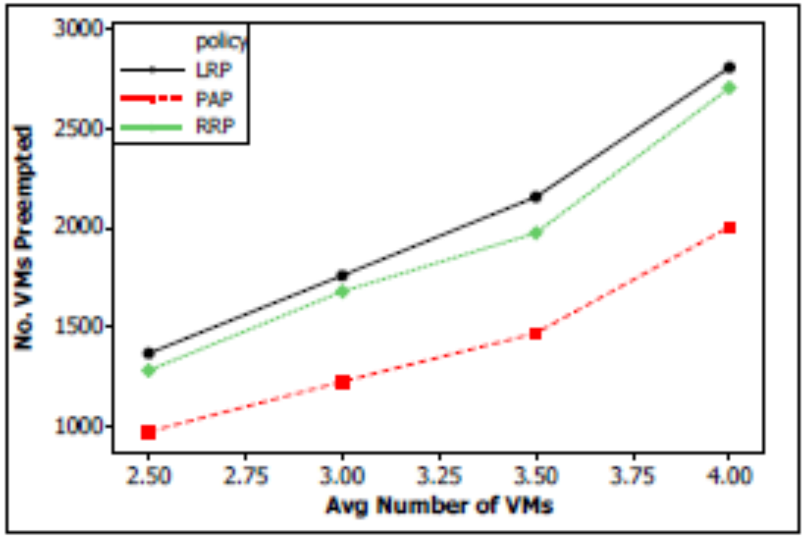
Baseline Policies

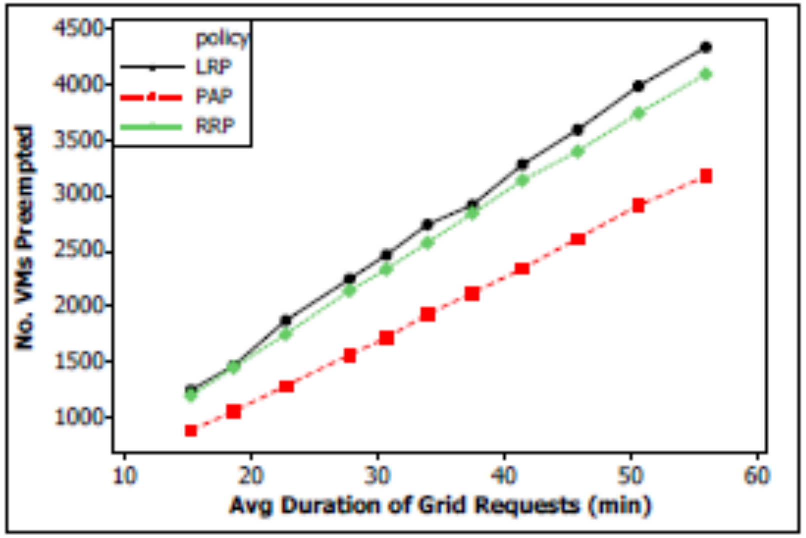
- Round Robin Policy (RRP)
- Least Rate Policy (LRP)

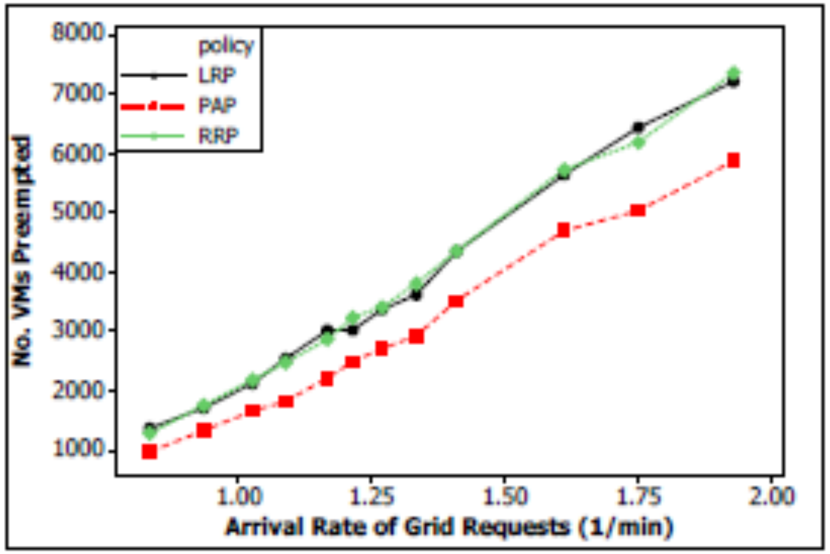
- DAS-2 workload model

Input Parameter	Distribution	Values Grid Requests	Values Local Requests
No. of VMs	Loguniform	$(l = 0.8, 1.5 \leq m \leq 3, h = 5, q = 0.9)$	$(l = 0.8, m = 3, h = 5, q = 0.9)$
Request Duration	Lognormal	$(1.5 \leq a \leq 2.6, b = 1.5)$	$(a = 1.5, b = 1.0)$
Inter-arrival Time	Weibull	$(0.7 \leq \alpha \leq 3, \beta = 0.5)$	$(\alpha = 0.7, \beta = 0.4)$
P_{one}	N/A	0.2	0.3
P_{pow2}	N/A	0.5	0.6

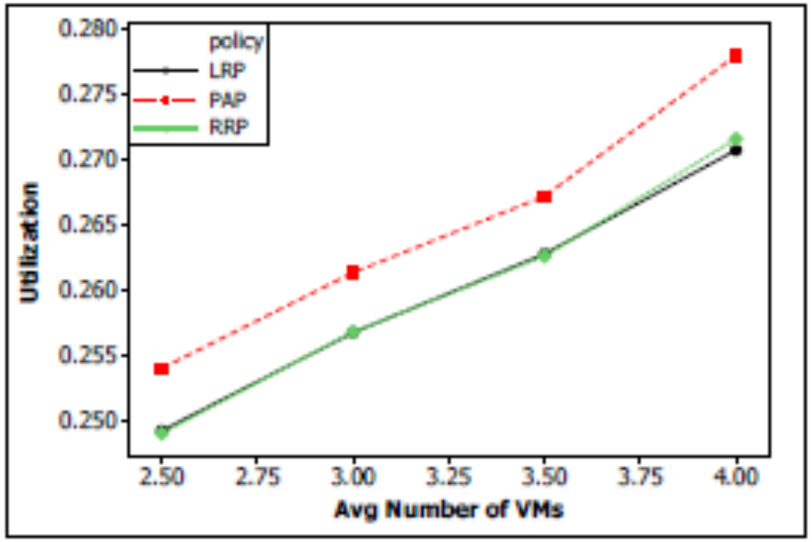
Experimental Results: Number of VM Preemptions

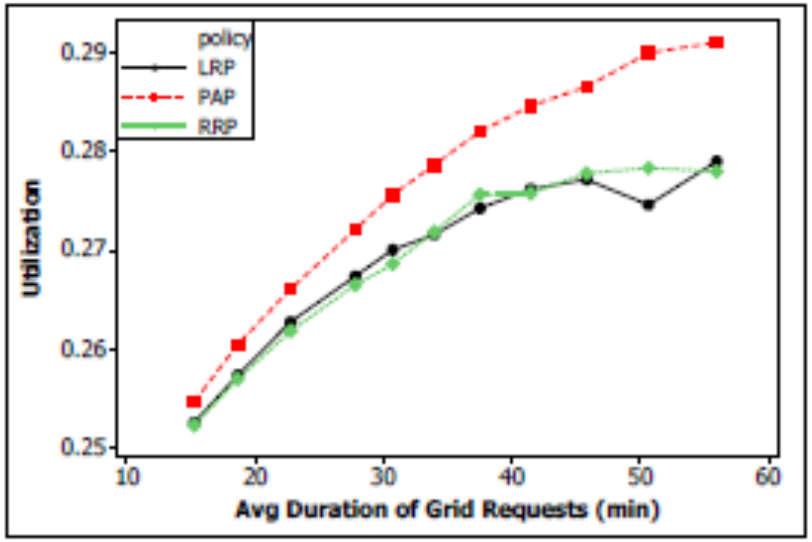


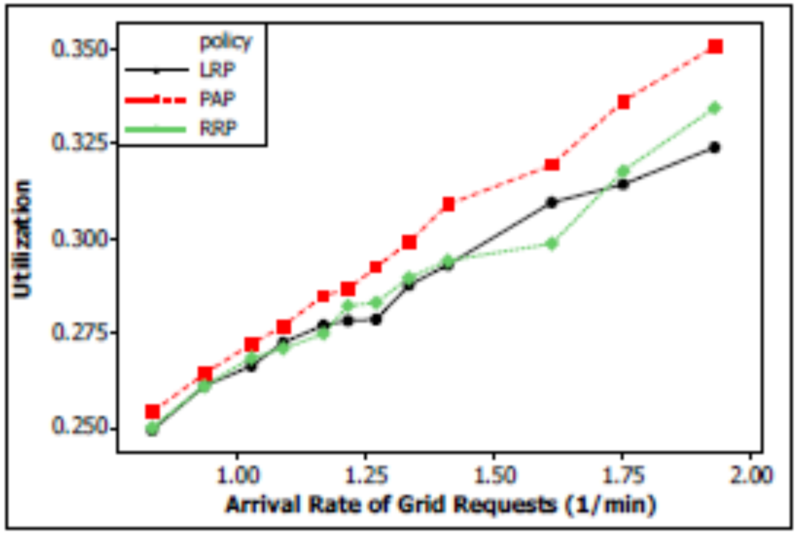




Experimental Results: Resource Utilization

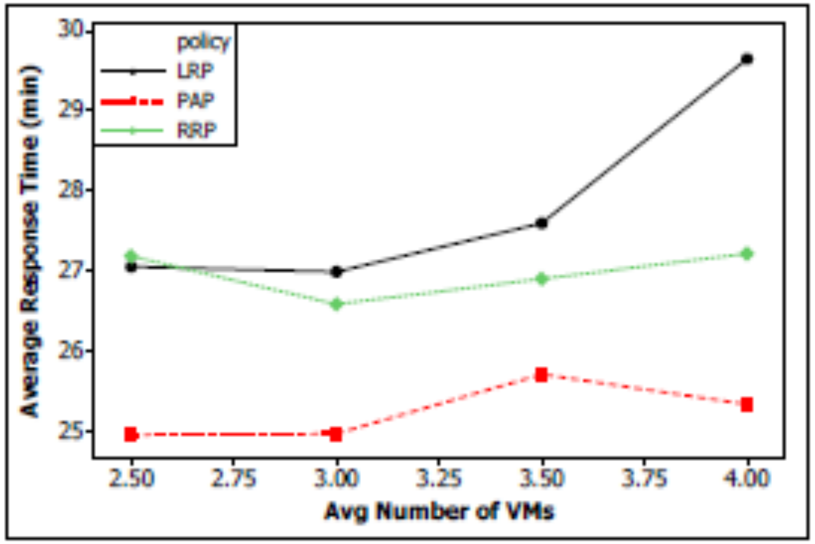


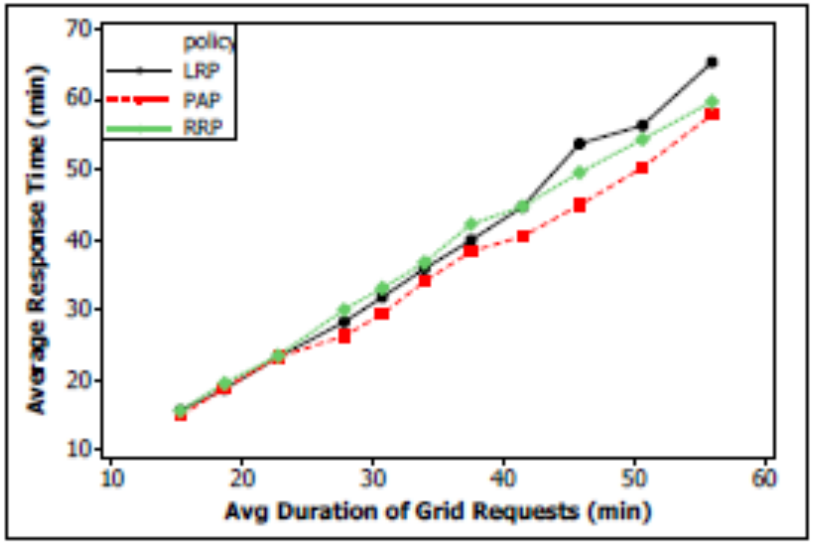


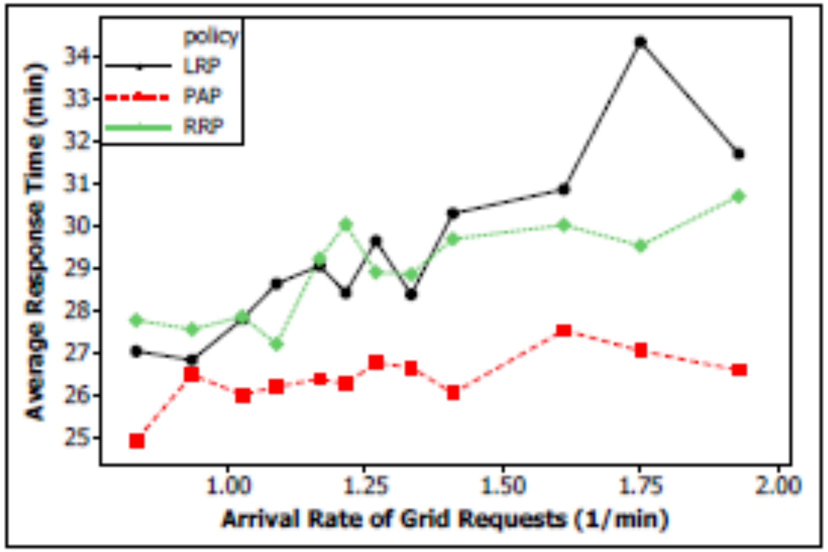


MELBOURNE CLOUDS LAB

Experimental Results: Average Response Time







Conclusions and Future Work

- we explored how we can minimize the number of preemptions in InterGrid.
- We proposed a preemption-aware scheduling policy (PAP)
- Experiments show that PAP resulted in up to 1000 less VM preemptions (22.5% improvement) comparing with other policies.

- Questions?