Preemption-aware Admission Control in a Virtualized Grid Federation

Mohsen Amini Salehi, Bahman Javadi, Rajkumar Buyya

CLOUDS Laboratory
Department of Computing and Information Systems
The University of Melbourne, Australia
{mohsena,bahmanj,raj}@csse.unimelb.edu.au
Introduction: InterGrid

- Provides an architecture and policies for interconnecting different Grids.
- Computational resources in each Grid are shared between grid (External) users and local users.
- Local users have preemptive priority over external users!
Contention between Local and External (Ext.) users

• Why contention happens?
  – Lack of resource (oversubscription of resources)

• Solution for Contention:
  – Preemption of Ext. requests in favor of local requests

• Preemption increases the response time and leads to deadline violation for Ext. requests.
Research Question

• Deadline violations is because of over-subscription to the ext. requests.
• Resource owners tend to accept as many ext. requests as possible.
• The question that arises is:
  – What is the ideal number of ext. requests a cluster can accept in a way that:
    • The number of accepted ext. requests is maximized
    • Deadline violation is avoided
Our approach: Using Admission Control.
Problem Statement

• What is the optimal queue length \( (K_j) \) for ext. requests for in cluster \( j \)?
  – Analytical modeling of preemption for ext. requests in a cluster.
Assumptions

• Local requests are immediate and must be processed as soon as they are submitted.
• We assume an M/G/1 queue to model the service time of local requests.
• External requests are submitted to a queue in each cluster that can be modeled as an M/G/1/K queue model.
• All requests are moldable applications which can spread over all available resources in a cluster.
Analytical Model

- Our primary objective function is:
  \[ E(R_j) = E(W_j) + E(T_j) \leq D \]

- Assume that overall run time of an ext. request is \( \omega \), and encounters \( n \) preemptions before getting completed, then service time is:
  \[ T_j = e_1^j + l_1^j + e_2^j + l_2^j + \ldots + e_n^j + l_n^j + \epsilon \]

- Arrival rate of local requests (\( \lambda_j \)) follows Poisson distribution, so \( n \) follows Gamma distribution:
  \[ E(n) = \lambda_j \omega \]
Analytical Model (2)

\[ E(T_j) = E(E(T_j|n)) = \omega + \lambda_j \omega E(l_1^j) \]

• We assume that local requests follow M/G/1 model, then:

\[ E(T_j) = \frac{\mu_l^j \cdot \omega}{\mu_l^j - \lambda_j} = \frac{\omega}{1 - \rho_l^j} \]

• The average waiting time of external requests in the M/G/1/K queue is:

\[ E(W_j) = \frac{1}{\Lambda_j} \sum_{k=0}^{K_j-1} k \cdot P_{d,k}^j + \frac{K_j}{\Lambda_j} (P_{d,0}^j + \rho_e^j - 1) - E(T_j) \]

We have to figure out \( \rho_e^j \) and \( P_{d,k}^j \)

• \( \rho_e^j \) is the queue utilization for external requests:

\[ \rho_e^j = \Lambda_j \cdot E(T_j) = \frac{\omega \cdot \Lambda_j}{1 - \rho_l^j} \]
Analytical Model (3)

- $P_{d,k}^j$ is the probability that a newly arriving external request encounters $k$ requests waiting in the queue of cluster $j$:

$$P_{d,k}^j = \frac{P_{\infty,k}^j}{\sum_{i=0}^{K_j-1} P_{\infty,i}^j}, \quad k = 0, 1, ..., K_j - 1$$

$$P_{\infty,k}^j = \frac{1}{\mu_e^j} \left( a_{k-1} \cdot P_{\infty,0}^j + \sum_{i=1}^{k-1} a_{K_j-i} \cdot P_{\infty,i}^j \right)$$

$$a_k^j = \int_0^\infty \frac{(t\lambda_j)^k}{k!} \cdot e^{-t\lambda_j} \cdot b_j(t) \cdot dt$$
Analytical Model(4)

• $b_j(t)$ is the probability density function (PDF) of service time for ext. requests.

• Gong et al.¹ prove the service time of ext. requests with preemption follows the Gamma distribution.

• Based on Gamma distribution:

$$b_j(t) = \frac{(t/\alpha)^{\beta-1}.e^{-t/\alpha}}{\alpha \cdot \Gamma(\beta)}$$

Preemption-aware Admission Control Policy (PACP) for cluster $j$

**Algorithm 1**: Preemption-aware Admission Control Policy (PACP) in cluster $j$.

**Input**: $\Lambda_j, \theta_j, \omega, \lambda_j, \mu_c^j, \mu_l^j, \text{rate}_l, u_l, u_h$

**Output**: $K_j$ (Queue length)

1. $D \leftarrow (\text{rate}_l \times u_l \times \omega) + ((1 - \text{rate}_l) \times u_h \times \omega)$;
2. $K_j \leftarrow 0$;
3. $\text{ExpectedResponse}_j \leftarrow 0$;
4. while $\text{ExpectedResponse}_j < D$ do
   5. /*calculating $E(R)$ for a queue with length $K_j$ in cluster $j$*/
   6. $\sigma \leftarrow 0$;
   7. for $N_q^j \leftarrow 0$ to $K_j - 1$ do
      8. $\sigma + = N_q^j \cdot P_{d,N_q^j}^j$;
   9. $\text{ExpectedResponse}_j \leftarrow \frac{1}{\Lambda_j} \cdot \sigma_j + \frac{K_j}{\Lambda_j} (P_d^j,0 + \rho_e^j - 1)$;
10. $K_j \leftarrow K_j + 1$;
Performance Metrics

• We define $D$ (average deadline of ext. requests) as:

$$D = (rate_l \cdot u_l \cdot \omega) + ((1 - rate_l) \cdot u_h \cdot \omega)$$

  – $rate_l$ is the proportion of low-urgency ext. requests and $u_l, u_h$ are the deadline ratios.

• Deadline Violation Rate (DVR):

$$DVR = \frac{(a \cdot v) + r}{a + r} \cdot 100$$

  – $a$ and $r$ are percentage of accepted and rejected requests. $v$ is the deadline violation ratio.

• Completed External Requests.
Experimental Setup

• We use GridSim for simulation
• 3 clusters with 64, 128, and 256 nodes and different computing speeds (2000, s2=3000, s3=2100 MIPS)
• Conservative Backfilling for cluster scheduling.
• Grid Workload Archive (GWA) is used to generate 2 days of bag-of-tasks requests.
Baseline Policies

- **Conservative Admission Control Policy (CACP):**
  - Admits as many requests as assigned by the IGG (queue length is infinite).

- **Aggressive Admission Control Policy (AACP):**
  - Each cluster accepts one external request at any time and tries to meet the deadline.

- **Rate-based Admission Control Policy (RACP):**
  - Queue length is determined based on the service rate for external requests and local request arrival rate in a cluster.
Deadline Violation Rate (DVR)
Completed External Requests

- 4 graphs showing the completion rate of external requests under varying task duration, external scale, and local scale. Each graph compares different policies (AACP, CACP, PACP, RACP).
- The x-axis represents the task duration or scale, while the y-axis shows the completion rate of external requests.
Conclusion and Future Work

• We explored the ideal number of ext. requests that a cluster can accept without violating deadlines in a federated Grid.
• We developed a performance model based on queuing.
• Experimental results indicate that the PACP decreases the deadline violation rate up to 20%.
• PACP leads to completing more ext. requests (up to 25%).
• In future, we plan to relax the assumption of moldable applications and solve the problem for all types of parallel requests.
• Any Question?