Flexible Time-Windows for Advance Reservation in LambdaGrids

Neena Kaushik
Silvia Figueira
Stephen Chiappari

Santa Clara University
Santa Clara, California, USA
Advance reservation

- A LambdaGrid provides the computational, storage, visualization, and the optical-network resources to the user as schedulable resources.

- Resources can be reserved in advance and requests may be rejected when the resources are not available.
Flexibility

- **GOAL** → Use flexibility to increase acceptance rate and decrease blocking probability.

Should I be flexible or wait in the queue?
Hypothesis

The value of the *average window time* for all user requests, which will *theoretically lower the blocking probability* in the advance-reservation scheduling domain to 0, is the same as the *mean waiting time* of an equivalent queue-based on-demand scheduler, when the traffic intensity is less than 1.
Results

- Window size that brings the blocking probability to zero
  - M/M/1
- Represented in the results are window sizes for:
  - Queuing model
  - Simulation with no time-slots
  - Simulation with time-slots
Simulation parameters

- Traffic intensity $\rho = 0.2$ to $0.8$
- 50 different traces
  - For each trace, calculate the average window size.
  - Take the average window size over the 50 values obtained.
- Maximum possible window size = 1,000
Window size that brings the blocking probability to zero

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>WS (in hours)</th>
<th>Other Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>$\lambda = 0.2$, $\mu = 1$</td>
</tr>
<tr>
<td>0.4</td>
<td>1.33</td>
<td>$\lambda = 0.2$, $\mu = 0.5$</td>
</tr>
<tr>
<td>0.6</td>
<td>4.5</td>
<td>$\lambda = 0.2$, $\mu = 0.333$</td>
</tr>
<tr>
<td>0.8</td>
<td>16</td>
<td>$\lambda = 0.2$, $\mu = 0.25$</td>
</tr>
</tbody>
</table>
Flexibility

- How much flexibility will help to decrease the blocking probability of advance reservations?

How flexible should I be?
Simulation parameters

- A single trace was generated.
- Calculate the blocking probability and utilization for increasing window sizes.
- The window size is increased until the blocking probability drops to zero or close to zero.
Decreasing blocking probability with increasing window size (1 of 3)

![Graph showing decreasing blocking probability with increasing window size for different load factors. The graph includes lines for load factors 0.2, 0.4, 0.6, and 0.8, with window size on the x-axis and block probability on the y-axis. The M/M/1 notation is placed at the bottom right of the graph.]
Decreasing blocking probability with increasing window size (2 of 3)

\[ f(x) = \left( \frac{\alpha k^\alpha}{1 - (k/p)^\alpha} \right) x^{(-\alpha - 1)} , \quad k \leq x \leq p \]

\( M/B/1 : \alpha = 1.7, \ k = 1, \ p = 1000 \)
Decreasing blocking probability with increasing window size (3 of 3)

\[ f(x) = \frac{\alpha k^\alpha}{(1-(k/p)^\alpha)} x^{(-\alpha-1)} , \quad k \leq x \leq p \]

B/M/1 : \( \alpha = 0.9 \), \( k = 1 \), \( p = 1000 \)
Conclusion

- Flexible time-windows can improve resource utilization in advance reservation scheduling.
- Flexible window size is equal to mean waiting time in the queue.