Queueing Theory

M/M/1 and M/M/m Queues
Why queuing theory?

- Reason about the system
- Analyze different components
- Determine bottlenecks
- Queues are a core aspect of complex systems

**FIGURE 30.2** Common random variables used in analyzing a queue.

Art of Computer Systems
Performance Analysis - p. 511
Little’s Law

Number of requests in the system = arrival rate * mean response time

- Relates the number of requests to the response time
- Valid for any type of queueing system
- Valid for systems in its entirety or for parts of the system

Number of requests in the queue = arrival rate * mean waiting time in the queue
What you are trying to predict?

- Be clear what the input to the model is. This depends on what you have measured!
- Be clear what elements you want to predict
- Explain which aspects of your system correspond to the model and which ones do not
- Consider all the data you have gathered

There are many options to proceed!

<table>
<thead>
<tr>
<th>Input</th>
<th>Anything that has been measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Arrival rate, service rate</td>
</tr>
<tr>
<td>Output</td>
<td>Length of the queues, number of request in the system, graphs, behaviour, ...</td>
</tr>
</tbody>
</table>
How can we determine this?

Measured

Given by the model

$\lambda$

$\mu$

$m=1$
M/M/1 Queue

How can we determine this?

\[ \lambda \rightarrow \mu \rightarrow m=1 \]
M/M/1 Queue

How can we determine this?

Input to the model: $\lambda$, $\mu$

How to measure the service rate $\mu$?

- There are many approaches, depending on what aspect of your system you want to model!
- E.g., use configuration that gives maximum throughput
**M/M/1 Queue**

How can we determine this?

- Input to the model: $\lambda$, $\mu$
- How to measure the service rate $\mu$?
  - There are many approaches, depending on what aspect of your system you want to model!
  - E.g., use configuration that gives maximum throughput

Service rate $> \text{arrival rate. } \rho < 1$. 

Find the maximum observed throughput

- Service rate $> \text{arrival rate. } \rho < 1$. 

- Input to the model: $\lambda$, $\mu$ 

- How to measure the service rate $\mu$?
  - There are many approaches, depending on what aspect of your system you want to model!
  - E.g., use configuration that gives maximum throughput
Example of computing the output of M/M/m models
Example 1: M/M/2

- Input: $\lambda=9 \text{ req/s}$, $m=2$, $\mu=5 \text{ req/s}$
- Utilization:
- Average service time per worker:
- Average number of requests in queue:
- Average waiting time in queue:
- Average response time:
Example 1: M/M/2

- **Input:** \( \lambda=9 \text{ req/s}, \ m=2, \ \mu=5 \text{ req/s} \)
- **Utilization:** \( \rho = \lambda/(m\times\mu) = 90\% \)
- **Average service time per worker:**
- **Average number of requests in queue:**
- **Average waiting time in queue:**
- **Average response time:**
Example 1: M/M/2

- Input: $\lambda=9$ req/s, $m=2$, $\mu=5$ req/s
- Utilization: $\rho = \frac{\lambda}{m \times \mu} = 90\%$
- Average service time per worker: $E[s] = 0.20$ s
- Average number of requests in queue: $E[nq] = 7.674$
- Average waiting time in queue:
- Average response time:
Example 1: M/M/2

- **Input:** $\lambda = 9 \text{ req/s}$, $m = 2$, $\mu = 5 \text{ req/s}$
- **Utilization:** $\rho = \frac{\lambda}{m \cdot \mu} = 90\%$
- **Average service time per worker:** $E[s] = 0.20 \text{ s}$
- **Average number of requests in queue:** $E[nq] = 7.674$
- **Average waiting time in queue:** $E[wq] = 0.853 \text{ s}$
- **Average response time:**

For the queue: $E[nq] = \lambda \cdot E[wq]$

Little's Law!
Example 1: M/M/2

- Input: $\lambda = 9$ req/s, $m = 2$, $\mu = 5$ req/s
- Utilization: $\rho = \lambda / (m \cdot \mu) = 90\%$
- Average service time per worker: $E[s] = 0.20$ s
- Average number of requests in queue: $E[nq] = 7.674$
- Average waiting time in queue: $E[wq] = 0.853$ s
- Average response time: $E[w] = 0.853 + 0.20 = 1.053$ s

Mean number of req. in the system = arrival rate * mean response time
For the queue: $E[nq] = \lambda \cdot E[wq]$

Little's Law!

Total response time = queuing time + service time
Can also be computed as a function of $\lambda$, $\mu$, $m$, $\rho$. (see book)
Example 1: M/M/2

- Input: \( \lambda = 9 \) req/s, \( m = 2 \), \( \mu = 5 \) req/s
- Utilization: \( \rho = \frac{\lambda}{m \times \mu} = 90\% \)
- Average service time per worker: \( E[s] = 0.20 \) s
- Average number of requests in queue: \( E[nq] = 7.674 \)
- Average waiting time in queue: \( E[wq] = 0.853 \) s
- Average response time: \( E[w] = E[wq] + E[s] = 1.053 \) s

Mean number of req. in the system
= arrival rate * mean response time
For the queue: \( E[nq] = \lambda \times E[wq] \)

\[ \text{Little's Law!} \]

Total response time
= queuing time + service time

Can also be computed as a function of \( \lambda, \mu, m, \rho \ldots \) (see book)

In your report, you need to compare the output of the model to your measurements. Explain the difference!
Example 2: $M/M/4$

- Input: $\lambda=9$ req/s, $m=4$, $\mu=5$ req/s
- Utilization:
- Average service time per worker:
- Average number of requests in queue:
- Average waiting time in queue:
- Average response time:

What changes for $m=4$?
Example 2: M/M/4

- Input: $\lambda = 9 \text{ req/s}$, $m=4$, $\mu=5 \text{ req/s}$
- Utilization: $\rho = \lambda / (m \times \mu) = 45\%$
- Average service time per worker: $E[s] = 0.20 \text{ s}$
- Average number of requests in queue:
- Average waiting time in queue:
- Average response time:
Example 2: M/M/4

- Input: $\lambda=9 \text{ req/s}$, $m=4$, $\mu=5 \text{ req/s}$
- Utilization: $\rho = \frac{\lambda}{m \times \mu} = 45\%$
- Average service time per worker: $E[s] = 0.20 \text{ s}$
- Average number of requests in queue: $E[nq] = 0.105$
- Average waiting time in queue: $E[wq] = 0.012 \text{ s}$
- Average response time: $E[w] = 0.012 + 0.20 = 0.212 \text{ s}$

What changes for $m=4$?
Example 2: M/M/4

- Input: $\lambda = 9 \text{ req/s}$, $m = 4$, $\mu = 5 \text{ req/s}$
- Utilization: $\rho = \frac{\lambda}{m \cdot \mu} = 45\%$
- Average service time per worker: $E[s] = 0.20 \text{ s}$
- Average number of requests in queue: $E[nq] = 0.105$
- Average waiting time in queue: $E[wq] = 0.012 \text{ s}$
- Average response time: $E[w] = 0.012 + 0.20 = 0.212 \text{ s}$

The system is not saturated in this configuration. No queueing.

Queueing can be a big part of the response time (see comparison to M/M/2)
Example 3: How do these systems differ?

<table>
<thead>
<tr>
<th></th>
<th>$\lambda=9 \text{ req/s, } m=2, \mu=5 \text{ req/s}$</th>
<th>$\lambda=18 \text{ req/s, } m=4, \mu=5 \text{ req/s}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Utilization</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Service time</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Queue Length</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Queuing Time</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Response Time</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 3: How do these systems differ?

Both systems have a 90% utilization

<table>
<thead>
<tr>
<th></th>
<th>( \lambda = 9 \text{ req/s}, m=2, \mu = 5 \text{ req/s} )</th>
<th>( \lambda = 18 \text{ req/s}, m=4, \mu = 5 \text{ req/s} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilization</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td>Service time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queue Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queuing Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 3: How do these systems differ?

<table>
<thead>
<tr>
<th></th>
<th>$\lambda=9$ req/s, $m=2$, $\mu=5$ req/s</th>
<th>$\lambda=18$ req/s, $m=4$, $\mu=5$ req/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilization</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td>Service time</td>
<td>0.200 s</td>
<td>0.200 s</td>
</tr>
<tr>
<td>Queue Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queuing Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 3: How do these systems differ?

<table>
<thead>
<tr>
<th></th>
<th>$\lambda=9 \text{ req/s, } m=2, \mu=5 \text{ req/s}$</th>
<th>$\lambda=18 \text{ req/s, } m=4, \mu=5 \text{ req/s}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Utilization</strong></td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td><strong>Service time</strong></td>
<td>0.200 s</td>
<td>0.200 s</td>
</tr>
<tr>
<td><strong>Queue Length</strong></td>
<td>7.67 requests</td>
<td>7.10 requests</td>
</tr>
<tr>
<td><strong>Queuing Time</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Response Time</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 3: How do these systems differ?

<table>
<thead>
<tr>
<th></th>
<th>( \lambda=9 \text{ req/s}, m=2, \mu=5 \text{ req/s} )</th>
<th>( \lambda=18 \text{ req/s}, m=4, \mu=5 \text{ req/s} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilization</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td>Service time</td>
<td>0.200 s</td>
<td>0.200 s</td>
</tr>
<tr>
<td>Queue Length</td>
<td>7.67 requests</td>
<td>7.10 requests</td>
</tr>
<tr>
<td>Queuing Time</td>
<td>0.853 s</td>
<td>0.394 s</td>
</tr>
<tr>
<td>Response Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 3: How do these systems differ?

<table>
<thead>
<tr>
<th></th>
<th>λ=9 req/s, m=2, μ=5 req/s</th>
<th>λ=18 req/s, m=4, μ=5 req/s</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Utilization</strong></td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td><strong>Service time</strong></td>
<td>0.200 s</td>
<td>0.200 s</td>
</tr>
<tr>
<td><strong>Queue Length</strong></td>
<td>7.67 requests</td>
<td>7.10 requests</td>
</tr>
<tr>
<td><strong>Queuing Time</strong></td>
<td>0.853 s</td>
<td>0.394 s</td>
</tr>
<tr>
<td><strong>Response Time</strong></td>
<td>1.053 s</td>
<td>0.594 s</td>
</tr>
</tbody>
</table>
Example 3: How do these systems differ?

What would be better? 2x M/M/2 or 1x M/M/4

<table>
<thead>
<tr>
<th></th>
<th>$\lambda=9$ req/s, $m=2$, $\mu=5$ req/s</th>
<th>$\lambda=18$ req/s, $m=4$, $\mu=5$ req/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilization</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td>Service time</td>
<td>0.200 s</td>
<td>0.200 s</td>
</tr>
<tr>
<td>Queue Length</td>
<td>7.67 requests</td>
<td>7.10 requests</td>
</tr>
<tr>
<td>Queuing Time</td>
<td>0.853 s</td>
<td>0.394 s</td>
</tr>
<tr>
<td>Response Time</td>
<td>1.053 s</td>
<td>0.594 s</td>
</tr>
</tbody>
</table>
Example 3: How do these systems differ?

What would be better?
2x M/M/2 or 1x M/M/4
M/M/4 has less queueing!

<table>
<thead>
<tr>
<th></th>
<th>$\lambda=9\text{ req/s}, m=2, \mu=5\text{ req/s}$</th>
<th>$\lambda=18\text{ req/s}, m=4, \mu=5\text{ req/s}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Utilization</strong></td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td><strong>Service time</strong></td>
<td>0.200 s</td>
<td>0.200 s</td>
</tr>
<tr>
<td><strong>Queue Length</strong></td>
<td>7.67 requests</td>
<td>7.10 requests</td>
</tr>
<tr>
<td><strong>Queueing Time</strong></td>
<td>0.853 s</td>
<td>0.394 s</td>
</tr>
<tr>
<td><strong>Response Time</strong></td>
<td>1.053 s</td>
<td>0.594 s</td>
</tr>
</tbody>
</table>
• We have updated the FAQ page. Please check before sending emails.

• Look at the report in its entirety
  – Analysis is about the same system in every section
  – Make a list of all the measurements you need
  – Write down all the configurations you need to test

• Next week: Network of queues
  – Afterwards: Q&A sessions
  – Office hours (Details to be announced next week)