Performance Modelling of Computer Systems

Mirco Tribastone

Institut für Informatik
Ludwig-Maximilians-Universität München

Queueing Networks

More on Poisson Processes (1/2)

Theorem (Superposition of Poisson processes)

Let A_1, A_2, ..., A_n be independent Poisson processes with rates λ_1, λ_2, ..., λ_n. Then $A = A_1 + A_2 + \ldots + A_n$ is a Poisson process with rate $\sum_{i=1}^{n} \lambda_i$.

Case $n = 2$ (in general, proven by induction).

Let K be the r.v. which gives the number of arrivals of $A = A_1 + A_2$ in an interval of length t. Define similarly K_1 and K_2.

$$
\begin{aligned}
\mathbb{P}(K = k) &= \mathbb{P}(K_1 = k, K_2 = 0) + \mathbb{P}(K_1 = k-1, K_2 = 1) + \ldots + \mathbb{P}(K_1 = 0, K_2 = k) \\
&= \sum_{i=0}^{k} \mathbb{P}(K_1 = i, K_2 = k - i) = \sum_{i=0}^{K} \mathbb{P}(K_1 = i)\mathbb{P}(K_2 = k - i) \\
&= \sum_{i=0}^{k} \frac{(\lambda_1 t)^i}{i!} e^{-\lambda_1 t} \frac{(\lambda_2 t)^{(k-i)}}{(k-i)!} e^{-\lambda_2 t} = e^{-(\lambda_1 + \lambda_2)t} \sum_{i=0}^{k} \frac{(\lambda_1 t)^i}{i!} \frac{(\lambda_2 t)^{(k-i)}}{(k-i)!} \\
&= \frac{e^{-(\lambda_1+\lambda_2)t}}{k!} \sum_{i=0}^{k} \binom{k}{i} (\lambda_1 t)^i (\lambda_2 t)^{(k-i)} = e^{-(\lambda_1+\lambda_2)t} \frac{[(\lambda_1 + \lambda_2)t]^k}{k!}.
\end{aligned}
$$

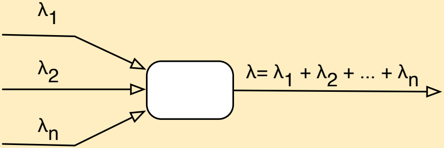
□

More on Poisson Processes (2/2)

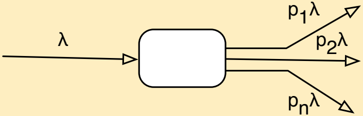
Theorem (Decomposition of Poisson Processes)

Let A be a Poisson process with rate λ. Define $B_1, B_2, \ldots, B_n$ as the processes which are assigned an arrival from A with probability $p_1, p_2, \ldots, p_n$ such that $\sum_{i=1}^{n} p_i = 1$. Then B_i is a Poisson process with rate $p_i\lambda$, for all $1 \leq i \leq n$.

Superposition of Poisson processes



Decomposition of Poisson processes



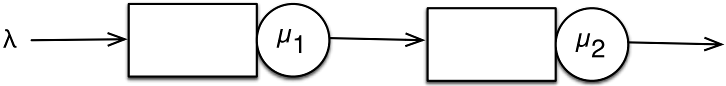
Toward Queueing Networks

Theorem (Burke)

In a $M/M/1$, $M/M/m$, or $M/M/\infty$ queue in the steady state with arrival rate λ, the following hold:

- *The departure rate is Poisson with rate λ;*
- *At any time t the queue length process is independent of the departure process prior to time t.*

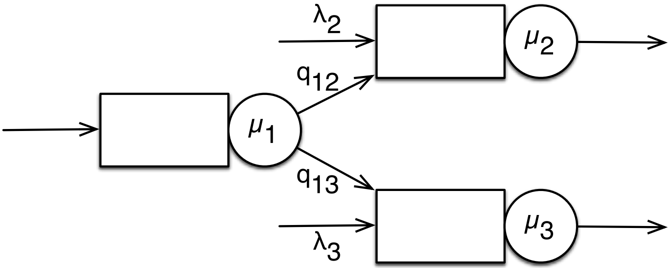
Application: Tandem Network of $M/M/1$ Queues



Let $\pi(k_1, k_2) = \mathbb{P}(K_1 = k_1, K_2 = k_2)$. (Similarly for the marginal steady-steady probabilities.) Let $\rho_i = \lambda/\mu_i$, $i = 1, 2$.

$$\pi(k_1, k_2) = \rho_1^{k_1}(1 - \rho_1)\rho_2^{k_2}(1 - \rho_2) = \underbrace{\pi_1(k_1)\,\pi_2(k_2)}_{\text{product form solution}} .$$

Feedforward Networks



Assume that all stations are $M/M/1$ queues and that $q_{12} + q_{13} = 1$.

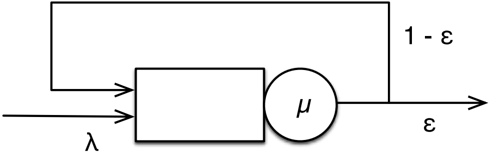
$$\pi(k_1, k_2, k_3) = \left[\rho_1^{k_1}(1 - \rho_1)\right]\left[\rho_2^{k_2}(1 - \rho_2)\right]\left[\rho_3^{k_3}(1 - \rho_3)\right],$$

where

$$\rho_1 = \lambda_1/\mu_1,$$
$$\rho_2 = (\lambda_2 + p_{12}\lambda_1)/\mu_2,$$
$$\rho_3 = (\lambda_3 + p_{13}\lambda_3)/\mu_3,$$

with $\rho_i < 1$ for $i = 1, 2, 3$.

A Queue with Feedback



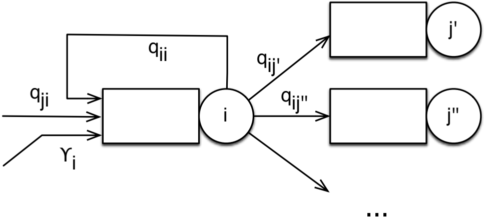
- Suppose that $\lambda \ll \mu$, i.e., arrivals are relatively infrequent with respect to service times. Suppose also that ε is very small.
- One exogenous arrival will give rise to a burst of arrivals because:
  - The customer will be served before the next exogenous arrival with high probability;
  - With high probability, the customer will be re-routed back into the queue, and will be served fast;
  - That is, an exogenous arrival may lead to that customer circulating in the network very often.

Jackson Networks

A queueing network consisting of M nodes, ranged over by i, j, such that:

- The network is open, i.e., there is at least one node with exogenous arrivals and there is at least one node where customers leave the system;
- For all i, node i is an infinite queue with m_i exponential servers with rate μ_i;
- External arrivals at node i are Poisson with rate γ_i;
- After completing service at node i, the customer moves to node j with probability q_{ij}, independently from the past history.
  - The matrix $Q = [q_{ij}]$ is called the routing matrix.
  - It is allowed that $q_{ii} > 0$ in order to model, for instance, the system shown in the previous slide.
  - If $\sum_j q_{ij} < 1$ then a customer departs from the system from station i with probability $1 - \sum_{j=1}^{M} q_{ij}$.

Traffic Equations



Define λ_i as the aggregate arrival rate at node i from the external source as well as from all other nodes:

$$\lambda_i = \gamma_i + \sum_{j=1}^{M} q_{ji}\lambda_j, \qquad i = 1, 2, \ldots, M.$$

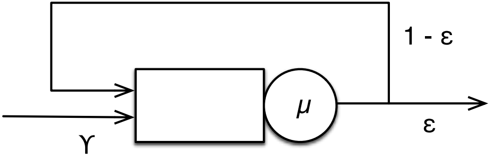
In matrix notation, let: $\vec{\lambda} = \begin{bmatrix}\lambda_1 & \lambda_2 & \ldots & \lambda_M\end{bmatrix}$ and $\vec{\gamma} = \begin{bmatrix}\gamma_1 & \gamma_2 & \ldots & \gamma_M\end{bmatrix}$. Solve the (standard) non-homogeneous system of linear equations:

$$\vec{\lambda} = \vec{\gamma} + Q\vec{\lambda} \quad \Longrightarrow \quad (I - Q)\vec{\lambda} = \vec{\gamma}.$$

Jackson Theorem

Consider a Jackson network with effective arrival rate λ_i at node i, with $i = 1, 2, \ldots, M$. If $\lambda_i < m_i \mu_i$ for all i then the following hold in steady state:

- Node i behaves as if it received Poisson arrivals with rate λ_i;
- The queue length at any node is independent from the queue length at every other node.



$$\lambda = \gamma + (1 - \varepsilon)\lambda \quad \Longrightarrow \quad \lambda = \gamma/\varepsilon.$$

If $\rho \equiv \gamma/(\varepsilon\mu) < 1$ then

$$\pi_k = \rho^k (1 - \rho).$$

Application

- Consider a Jackson network of M single-server nodes with rate μ_i.
- The steady-state probability distribution is calculated as follows:

$$\begin{aligned}\pi(k_1, k_2, \ldots, k_n) &= \pi_1(k_1)\pi_2(k_2)\cdots\pi_M(k_M) \\ &= \prod_{i=1}^{M} \left(\frac{\lambda_i}{\mu_i}\right)^{k_i} \left(1 - \frac{\lambda_i}{\mu_i}\right),\end{aligned}$$

  where λ_i, $1 \leq i \leq M$ are the solutions to the traffic equations.
- The network enjoys a product-form solution, i.e., each queue may be studied in isolation (provided that λ is obtained) and then the overall network behaviour is the product of the marginal distributions.
- This is similar to the case of feedforward networks, but in general here the arrivals are not Poisson.

Performance of Single-Server Jackson Networks

Let $\rho_i = \lambda_i/\mu_i$, with $1 \leq i \leq M$.

- Mean number of customers:

$$L_i = \frac{\rho_i}{1-\rho_i} \quad \Longrightarrow \quad L = L_1 + L_2 + \ldots + L_M = \sum_{i=1}^{M} \frac{\rho_i}{1-\rho_i}.$$

- Mean time spent in the network: We apply Little's law, where the system under consideration is taken to be the entire network, which is subjected to an overall (Poisson) arrival rate $\gamma = \sum_{i=1}^{M} \gamma_i$ with L customers in the steady-state. Therefore

$$W = \frac{L}{\gamma}.$$

- Average response time at a node: computed in a similar manner as

$$W_i = \frac{L_i}{\lambda_i}, \qquad 1 \leq i \leq M.$$

In general $W \neq W_1 + W_2 + \ldots + W_M$. **When is it the case?**

Closed Queueing Networks

- The network handles a fixed population N of customers cycling through the M nodes. There are no sources nor sinks, i.e.,

$$\sum_{j=1}^{M} q_{ij} = 1, \qquad 1 \leq i \leq M.$$

- The state space of the CTMC is given by

$$\mathcal{S}(N, M) = \left\{ (n_1, n_2, \ldots, n_M) \mid n_i \geq 0 \text{ for all } i, \sum_{i=1}^{M} n_i = N \right\},$$

  which has $\binom{N+M-1}{M-1}$ states.
- Define the traffic equations to be

$$\lambda_i = \sum_{j=1}^{M} \lambda_j q_{ji}, \qquad 1 \leq i \leq M \qquad \iff \qquad \vec{\lambda} = \vec{\lambda} Q.$$

Product Form for Closed Queueing Networks (1/2)

- The linear system of equations $\vec{\lambda} = \vec{\lambda} Q$ admits an infinity of solutions.
- Recall the similarity with the DTMC problem $\pi = \pi P$.
  - In that case, one linearly dependent equation was replaced with the normalising condition $\sum_i \pi_i = 1$.
  - Here, one fixes arbitrarily one solution $\vec{e} = \begin{bmatrix} e_1 & e_2 & \dots & e_M \end{bmatrix}$ by imposing, e.g., $e_1 = 1$.
  - The arrival rate λ_i will be proportional to e_i, i.e., $e_i = c\lambda_i$ for some positive constant c.
  - Similarly,

$$\rho_i = \lambda_i / \mu_i = e_i / (c\mu_i),$$

    that is,

$$e_i / \mu_i = c\rho_i.$$

  - Arrival rates and utilisations are normalised with respect to the arrival rate at station 1, which is taken to be equal to one.

Product Form for Closed Queueing Networks (2/2)

In a closed Jackson network (also known as *Gordon and Newell* network) with N circulating customers the equilibrium distribution is computed as:

$$\pi(\vec{n}) = \pi(n_1, n_2, \ldots, n_m) = \frac{1}{G(N)} \prod_{i=1}^{M} f_i(n_i), \quad \text{with}$$

$$f_i(n_i) = \begin{cases} \left(\dfrac{e_i}{\mu_i}\right)^{n_i}, & \text{if node } i \text{ is load independent,} \\ \dfrac{e_i^{n_i}}{\prod_{k=1}^{n_i} \mu_i(k)}, & \text{if node } i \text{ is load dependent,} \end{cases}$$

where $\mu_i(k)$ gives the service rate at station i when the queue length is k. $G(N)$ is the normalisation constant obtained by requiring that

$$\sum_{\vec{n} \in \mathcal{S}} \pi(n_1, n_2, \ldots, n_M) = 1, \qquad \text{from which}$$

$$G(N) = \sum_{\vec{n} \in \mathcal{S}(N,M)} \left[\prod_{i=1}^{M} f_i(n_i) \right].$$

State-Space Explosion Problem

- The computation of the normalisation constant

$$G(N) = \sum_{\vec{n} \in \mathcal{S}(N,M)} \left[\prod_{i=1}^{M} f_i(n_i) \right]$$

  is computationally difficult as it requires a summation over potentially many states.
- This phenomenon arises in most discrete-state models of concurrent systems, and is called state-space explosion.[1]

N	M	Size of $\mathcal{S}(N, M)$
1	2	2
10	2	11
100	3	5151
100	5	4598126
200	5	70058751

[1] Another example will be seen later on in the course.

Buzen's Algorithm (1/3)

- It is an iterative procedure for the computation of $G(N)$.
- The iterations are over N, the total customer population in the network.
- In the case of a network with load-independent servers, the algorithm requires $2MN$ operations in total.
- Many network characteristics (throughput, utilisation, ...) can be obtained as a function of $G(0)$, $G(1)$, ..., $G(N)$.

Buzen's Algorithm (2/3)

Define the auxiliary function

$$g_m(n) = \sum_{\vec{n} \in \mathcal{S}(n,m)} \left[\prod_{i=1}^{m} f_i(n_i) \right].$$

and observe that $G(N) = g_M(N)$.
Rewrite the equation as follows:

$$g_m(n) = \sum_{\vec{n} \in \mathcal{S}(n,m)} \left[\prod_{i=1}^{m} f_i(n_i) \right] = \sum_{k=0}^{n} \left[\sum_{\substack{\vec{n} \in \mathcal{S}(n,m) \\ n_m = k}} \prod_{i=1}^{m} f_i(n_i) \right]$$

$$= \sum_{k=0}^{n} f_m(k) \left[\sum_{\vec{n} \in \mathcal{S}(n-k,m-1)} \prod_{i=1}^{m-1} f_i(n_i) \right] = \sum_{k=0}^{n} f_m(k) g_{m-1}(n-k).$$

Buzen's Algorithm (3/3)

$$g_m(n) = \sum_{\vec{n} \in \mathcal{S}(n,m)} \left[\prod_{i=1}^{m} f_i(n_i) \right] = \sum_{k=0}^{n} f_m(k) g_{m-1}(n-k).$$

Start the iteration with

$$g_1(n) = \sum_{\vec{n} \in \mathcal{S}(n,1)} \prod_{i=1}^{1} f_i(n_i) = f_1(n), \qquad n = 0, 1, 2, \ldots, N,$$

$$g_m(0) = \sum_{\vec{n} \in \mathcal{S}(0,m)} \prod_{i=1}^{m} f_i(n_i) = f_1(0) f_2(0) \cdots f_m(0).$$

In fact, it is a double iteration:

for all m such that $1 \leq m \leq M$ **do**
  for all n such that $0 \leq n \leq N$ **do**
    compute $g_m(n) = \sum_{k=0}^{n} f_m(k) g_{m-1}(n-k)$
  end for
end for

Performance Measures and the Normalisation Constant

Let $\pi_i(n, N)$ be the steady-state probability that the i-th node has n customers when the total population is n.

- The marginal queue length probability distribution for a load independent node i is

$$\pi_i(n, N) = \frac{f_i(n)}{G(N)} \left[G(N-n) - \frac{e^i}{\mu_i} G(N-1-n) \right], \; G(j) = 0 \text{ if } j < 0.$$

- The throughput at node i, i.e., the rate at which customers leave node i is

$$X_i(N) = e^i \frac{G(N-1)}{G(N)}.$$

- The utilisation of a load independent node i is

$$U_i(N) = \frac{e^i}{\mu_i} \frac{G(N-1)}{G(N)}.$$

Mean Value Analysis

- For large models, the computation of $G(N)$ may be computationally challenging.
- It can be numerically unstable.
- In many practical situations, it may carry much more information than actually required.
  - Often, one is interested in the average performance measures of the system under scrutiny, not the complete distribution.
  - Mean value analysis (MVA) is an exact iterative algorithm which provides the average performance measures without computing the normalisation constant.
  - It is based on the following.

Theorem (Arrival theorem)

In a closed queueing network, the stationary distribution observed by a customer arriving at a node is equal to the stationary distribution of customers in the same network with one fewer customer.

Mean Value Analysis: Assumptions and Notation

The arrival theorem will be used in conjunction with Little's result by MVA. Consider a network with M single-server nodes with exponentially distributed service rates μ_i, $i = 1, 2, \ldots, M$ and FCFS scheduling policy.

Notation

- $A_i(k)$ is the average steady-state measure A concerning node i when the network population is k, where A is be one of the following:
  - L is the queue length, the customer population including the one in service, if any;
  - R is the response time, the sum of the time spent in the queue and during service;
  - X is the throughput, the rate at which customers depart from a node;
- ν_i is the visit ratio for node i, i.e., the number of times node i is visited between two successive visits to node 1, i.e., the solution of the system of linear equations $\vec{\nu} = \vec{\nu} Q$, with $\nu_1 = 1$.

Mean Value Analysis

Consider a network with k customers.

- By the arrival theorem, an arriving customer at node i in the steady state will see $L_i(k-1)$ customers.
- Since the queue is FCFS, the customer will wait on average $1/\mu_i$ time units for each other customer to clear, plus its own $1/\mu_i$, that is

$$R_i(k) = \frac{1}{\mu}\left[L_i(k-1)+1\right].$$

- By Little's law (applied to the entire network) and by properties of $\vec{\nu}$,

$$X_i(k) = \nu_i \frac{k}{\sum_{i=1}^{M} R_i(k)}, \qquad i = 1, 2, \ldots, M.$$

- Finally, by Little's law applied to each node in isolation

$$L_i(k) = X_i(k) R_i(k), \qquad i = 1, 2, \ldots, M.$$

- The iterations start with $L_i(0) = 0$, $i = 1, 2, \ldots, M$.

Closing Remarks

- This is a much simplified version of the original MVA algorithm, which applies to more general networks with different kinds of nodes
  - Exponentially distributed multi-server FCFS nodes;
  - Processor sharing nodes (not necessarily exponentially distributed);
  - Last-come first-served preemptive resume nodes (not necessarily exponentially distributed);
  - Infinite-server nodes (not necessarily exponentially distributed);
- . . . and multiple classes of customers!
- For further details, see the following references:

  - F. Baskett, K.M. Chandy, R. Muntz, and F. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *J. ACM*, vol. 22, no. 2, 1975.
  - M. Reiser, S. Lavemberg, "Mean-Value Analysis of Closed Multichain Queuing Networks," *J. ACM*, vol. 27, no. 2, 1980.