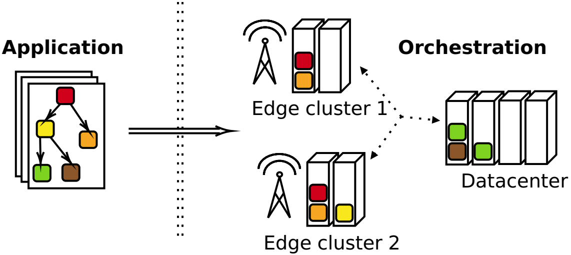
LUNDS UNIVERSITET

Cloud application modeling

using mean-field fluid models

Johan Ruuskanen

Introduction, managing cloud applications



Trends in modern cloud computing

- Applications split into graphs of smaller services
- Clouds of multiple clusters

Complex service graphs and dynamic environments

Introduction, managing cloud applications

Problem, how to deploy/manage an application such that

a) users receive a good QoS (e.g. low latency, robustness)
b) the costs are minimized (e.g. allocated resources)

Automatic adaption of resources and scheduling

Popular research topic considering single service application, and recently more considering service-graph applications.

Good decisions necessitates good models

Introduction, managing cloud applications

Problem, how to deploy/manage an application such that

a) users receive a good QoS (e.g. low latency, robustness)
b) the costs are minimized (e.g. allocated resources)

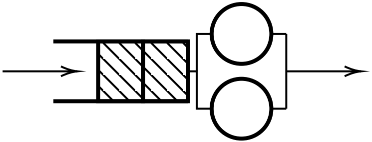
Automatic adaption of resources and scheduling

Popular research topic considering single service application, and recently more considering service-graph applications.

Good decisions necessitates good models

Applications as queuing networks

Common to model a service[1] as a queue.



Lifetime of a request: (i) arrives, (ii) is assigned a service time from G_s, (iii) processed according to *discipline* and (iv) departs once completed.

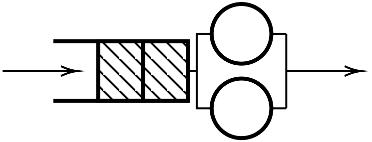
Queuing disciplines

- First come, first served (FCFS)
- Processor sharing (PS)
- Pure delay (INF)

[1] i.e. server, but not necessarily a physical computer

Applications as queuing networks

Common to model a service[1] as a queue.



Lifetime of a request: (i) arrives, (ii) is assigned a service time from G_s, (iii) processed according to *discipline* and (iv) departs once completed.

Queuing disciplines

- First come, first served (FCFS)
- Processor sharing (PS)
- Pure delay (INF)

[1] i.e. server, but not necessarily a physical computer

Applications as queuing networks

Applications of many stages, use many queues in a network

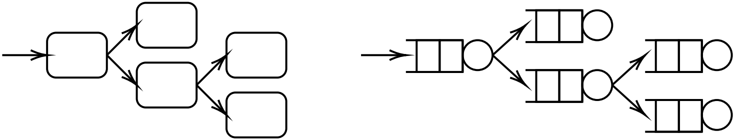


Figure: Simple example, where each stage is a service in a service graph.

Exists many extensions, one important is

Multi-class queues; Each queue has a set of *classes*, each request is assigned to one. Each class has its own G_s, and destination once completed.

$P_{i,j}^{r,s}$ - the probability that a completed request of class r in queue i gets routed to class s in queue j.

Applications as queuing networks

Applications of many stages, use many queues in a network

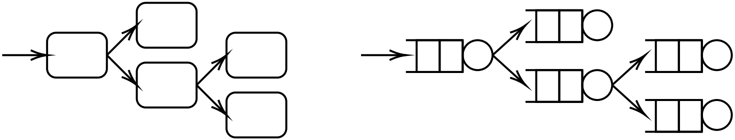


Figure: Simple example, where each stage is a service in a service graph.

Exists many extensions, one important is

Multi-class queues; Each queue has a set of *classes*, each request is assigned to one. Each class has its own G_s, and destination once completed.

$P_{i,j}^{r,s}$ - the probability that a completed request of class r in queue i gets routed to class s in queue j.

Evaluating a queuing network

$X_{i,r}(t)$ - population of requests of class r in queue i at time t.

Always possible to estimate the PMF of $X_{i,r}(t) \forall i, r, t \geq 0$ using MC simulations.

Very computationally intensive, not suitable for most cases.

Instead, approximate important metrics (e.g. mean queue length, response time)

Exists many methods

- Stationary, product-form -> methods utilizing the BCMP theorem
- Transient, non-product-from -> fluid models

Evaluating a queuing network

$X_{i,r}(t)$ - population of requests of class r in queue i at time t.

Always possible to estimate the PMF of $X_{i,r}(t) \forall i, r, t \geq 0$ using MC simulations.

Very computationally intensive, not suitable for most cases.

Instead, approximate important metrics (e.g. mean queue length, response time)

Exists many methods

- Stationary, product-form -> methods utilizing the BCMP theorem
- Transient, non-product-from -> fluid models

Fluid model of a queuing network

Model $\mathbb{E}\left[X_{i,r}(t)\right]$ as $x_{i,r}(t)$, where $\boldsymbol{X}(0) = \boldsymbol{x}(0)$ and

$$\dot{x}_{i,r}(t) = f_{i,r}^{in}(\boldsymbol{x}(t)) - f_{i,r}^{out}(\boldsymbol{x}(t))$$

Difficult to find f^{in}, f^{out} such that $\boldsymbol{x}(t)$ is a good approximation.

Much research has been done considering the single-queue/single-class case.

Queuing networks trickier, for some types the *mean-field approximation* gives one way.

Fluid model of a queuing network

Model $\mathbb{E}\left[X_{i,r}(t)\right]$ as $x_{i,r}(t)$, where $\boldsymbol{X}(0) = \boldsymbol{x}(0)$ and

$$\dot{x}_{i,r}(t) = f_{i,r}^{in}(\boldsymbol{x}(t)) - f_{i,r}^{out}(\boldsymbol{x}(t))$$

Difficult to find f^{in}, f^{out} such that $\boldsymbol{x}(t)$ is a good approximation.

Much research has been done considering the single-queue/single-class case.

Queuing networks trickier, for some types the *mean-field approximation* gives one way.

Fluid model of a queuing network

Mean-field approximation

Let $\boldsymbol{X}$ be a vector of populations in a *density-dependent population process* (special type of CTMC).

Transition l such that at an event $\boldsymbol{X}\left(t^{+}\right) = \boldsymbol{X}(t) + l$ with rate function $f(\boldsymbol{X}, l)$. The drift then becomes $F(\boldsymbol{X}) = \sum_{l \in \mathscr{L}} l f(\boldsymbol{X}, l)$

Mean-field approximation; $\dot{\boldsymbol{x}} = F(\boldsymbol{x})$, certain conditions $\upsilon^{-1}\boldsymbol{X}^{(\upsilon)} \to \boldsymbol{x}$ at all t when $\upsilon \to \infty$ (Kurtz's theorem).

Mean-field fluid model

For some queuing networks, possible to translate to such a CTMC.

Applies to multi-class queuing networks of PS and INF queues where G_s has a *Phase-type* distribution [2] [3].

[2] **Closed networks:** F. Pérez and G. Casale, *Line: Evaluating Software Applications in Unreliable Environments*, IEEE Transactions on Reliability (2017)

[3] **Open/mixed networks:** (allowing arrivals/departures) pre-print available

Fluid model of a queuing network

Mean-field approximation

Let $\boldsymbol{X}$ be a vector of populations in a *density-dependent population process* (special type of CTMC).

Transition l such that at an event $\boldsymbol{X}\left(t^{+}\right) = \boldsymbol{X}(t) + l$ with rate function $f(\boldsymbol{X}, l)$. The drift then becomes $F(\boldsymbol{X}) = \sum_{l \in \mathscr{L}} l f(\boldsymbol{X}, l)$

Mean-field approximation; $\dot{\boldsymbol{x}} = F(\boldsymbol{x})$, certain conditions $\upsilon^{-1}\boldsymbol{X}^{(\upsilon)} \to \boldsymbol{x}$ at all t when $\upsilon \to \infty$ (Kurtz's theorem).

Mean-field fluid model

For some queuing networks, possible to translate to such a CTMC.

Applies to multi-class queuing networks of PS and INF queues where G_s has a *Phase-type* distribution [2] [3].

[2] **Closed networks:** F. Pérez and G. Casale, *Line: Evaluating Software Applications in Unreliable Environments*, IEEE Transactions on Reliability (2017)

[3] **Open/mixed networks:** (allowing arrivals/departures) pre-print available

Fluid model of a queuing network

Mean-field approximation

Let $\boldsymbol{X}$ be a vector of populations in a *density-dependent population process* (special type of CTMC).

Transition l such that at an event $\boldsymbol{X}\left(t^{+}\right) = \boldsymbol{X}(t) + l$ with rate function $f(\boldsymbol{X}, l)$. The drift then becomes $F(\boldsymbol{X}) = \sum_{l \in \mathscr{L}} l f(\boldsymbol{X}, l)$

Mean-field approximation; $\dot{\boldsymbol{x}} = F(\boldsymbol{x})$, certain conditions $v^{-1}\boldsymbol{X}^{(v)} \to \boldsymbol{x}$ at all t when $v \to \infty$ (Kurtz's theorem).

Mean-field fluid model

For some queuing networks, possible to translate to such a CTMC.

Applies to multi-class queuing networks of PS and INF queues where G_s has a *Phase-type* distribution [2] [3].

[2] **Closed networks:** F. Pérez and G. Casale, *Line: Evaluating Software Applications in Unreliable Environments*, IEEE Transactions on Reliability (2017)

[3] **Open/mixed networks:** (allowing arrivals/departures) pre-print available

Mean-field fluid model

Phase-type distribution

Represent a distribution as the *time to absorption* in a single-sink CT Markov random walk across some graph.

Parameterized (for every class r in every queue i)

- $\alpha \in \mathbb{R}^{S_{i,r}}$, prob. vector of starting transient state
- $\Psi \in \mathbb{R}^{S_{i,r} \times S_{i,r}}$, matrix of transition rates between transient states
- $\psi \in \mathbb{R}^{S_{i,r}}$, transition rates between transient states and the sink

We can now introduce $X_{i,r,a}$

- population of requests in state a, in class r, in queue i.

Mean-field fluid model

Phase-type distribution

Represent a distribution as the *time to absorption* in a single-sink CT Markov random walk across some graph.

Parameterized (for every class r in every queue i)

- $\alpha \in \mathbb{R}^{S_{i,r}}$, prob. vector of starting transient state
- $\Psi \in \mathbb{R}^{S_{i,r} \times S_{i,r}}$, matrix of transition rates between transient states
- $\psi \in \mathbb{R}^{S_{i,r}}$, transition rates between transient states and the sink

We can now introduce $X_{i,r,a}$

- population of requests in state a, in class r, in queue i.

Mean-field fluid model

Assume a multi-class queuing network of PS and INF queues under Poisson arrivals

Nice thing with PS and INF queues, order does not matter.

- $\theta_{i,r,a}(\boldsymbol{X}) = X_{i,r,a} \frac{\min(k_i, \sum_{s,b} X_{i,s,b})}{\sum_{s,b} X_{i,s,b}}$
  Requests in i, r, a times the share of each request in queue i

then with PH distributions, the evolution of $\boldsymbol{X}$ is a CTMC.

Mean-field fluid model

Exists four types of transitions (l_1 and l_2 from Peréz & Casale)

- $\boldsymbol{e}_{i,r,a}$, zero vector with 1 on position i, r, a.

between phases: $l_1 = \boldsymbol{e}_{i,r,b} - \boldsymbol{e}_{i,r,a}$

$$f^n(\boldsymbol{X}, l_1) = \Psi^{i,r}_{a,b} \theta_{i,r,a}(\boldsymbol{X})$$

between classes: $l_2 = \boldsymbol{e}_{j,s,b} - \boldsymbol{e}_{i,r,a}$

$$f^c(\boldsymbol{X}, l_2) = \psi^{i,r}_{a} \alpha^{j,s}_{b} P^{r,s}_{i,j} \theta_{i,r,a}(\boldsymbol{X})$$

arrivals: $l_3 = \boldsymbol{e}_{i,r,a}$

$$f^a(\boldsymbol{X}, l_3) = \alpha^{i,r}_{a} \lambda^{i,r}$$

departures: $l_4 = -\boldsymbol{e}_{i,r,a}$

$$f^d(\boldsymbol{X}, l_4) = \psi^{i,r}_{a} \left(1 - \sum_{j,s} P^{r,s}_{i,j}\right) \theta_{i,r,a}(\boldsymbol{X})$$

Mean-field fluid model

Drift in each i, r, a

$$F_{i,r,a}(\boldsymbol{X}) = \sum_b \Psi^{i,r}_{b,a} \theta_{i,r,b}(\boldsymbol{X}) + \alpha^{i,r}_a \sum_{j,s,b} \psi^{j,s}_b P^{s,r}_{j,i} \theta_{j,s,b}(\boldsymbol{X}) + \alpha^{i,r}_a \lambda^{i,r}$$

Assuming $\boldsymbol{X}$ subsequently ordered in phases/classes/queues

$$\begin{aligned}
\boldsymbol{\Psi} &= \mathrm{diag}\left(\Psi^{1,1}, \Psi^{1,2}, \Psi^{1,3}, \ldots\right) \\
\boldsymbol{A} &= \mathrm{diag}\left(\alpha^{1,1}, \alpha^{1,2}, \alpha^{1,3}, \ldots\right) \\
\boldsymbol{B} &= \mathrm{diag}\left(\psi^{1,1}, \psi^{1,2}, \psi^{1,3}, \ldots\right) \\
\boldsymbol{P} &= \begin{bmatrix} P^{\cdot,\cdot}_{1,1} & P^{\cdot,\cdot}_{1,2} & \cdots \\ P^{\cdot,\cdot}_{2,2} & P^{\cdot,\cdot}_{2,2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}
\end{aligned}$$

We can create $\boldsymbol{W} = \boldsymbol{\Psi} + \boldsymbol{B}\boldsymbol{P}\boldsymbol{A}^T$ and

$$F(\boldsymbol{X}) = \boldsymbol{W}^T \theta(\boldsymbol{X}) + \boldsymbol{A}\boldsymbol{\lambda}$$

Mean-field fluid model

Drift in each i, r, a

$$F_{i,r,a}(\boldsymbol{X}) = \sum_{b} \Psi^{i,r}_{b,a} \theta_{i,r,b}(\boldsymbol{X}) + \alpha^{i,r}_{a} \sum_{j,s,b} \psi^{j,s}_{b} P^{s,r}_{j,i} \theta_{j,s,b}(\boldsymbol{X}) + \alpha^{i,r}_{a} \lambda^{i,r}$$

Assuming $\boldsymbol{X}$ subsequently ordered in phases/classes/queues

$$\begin{aligned}
\boldsymbol{\Psi} &= \mathrm{diag}\left(\Psi^{1,1}, \Psi^{1,2}, \Psi^{1,3}, \ldots\right) \\
\boldsymbol{A} &= \mathrm{diag}\left(\alpha^{1,1}, \alpha^{1,2}, \alpha^{1,3}, \ldots\right) \\
\boldsymbol{B} &= \mathrm{diag}\left(\psi^{1,1}, \psi^{1,2}, \psi^{1,3}, \ldots\right) \\
\boldsymbol{P} &= \begin{bmatrix} P^{\cdot,\cdot}_{1,1} & P^{\cdot,\cdot}_{1,2} & \cdots \\ P^{\cdot,\cdot}_{2,2} & P^{\cdot,\cdot}_{2,2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}
\end{aligned}$$

We can create $\boldsymbol{W} = \boldsymbol{\Psi} + \boldsymbol{B}\boldsymbol{P}\boldsymbol{A}^{T}$ and

$$F(\boldsymbol{X}) = \boldsymbol{W}^{T} \theta(\boldsymbol{X}) + \boldsymbol{A}\boldsymbol{\lambda}$$

Mean-field fluid model

The entire mean-field fluid model can then be expressed as

$$\dot{\boldsymbol{x}} = \boldsymbol{W}^T \theta(\boldsymbol{x}) + \boldsymbol{A}\boldsymbol{\lambda}$$
$$\boldsymbol{x}(0) = \boldsymbol{X}(0)$$

then $\lim_{\upsilon \to \infty} \upsilon^{-1} \boldsymbol{X}^{(\upsilon)} = \boldsymbol{x}$ at all t,
where $\boldsymbol{X}^{(\upsilon)}$ is $\boldsymbol{X}$ with k, $\boldsymbol{X}(0)$ and $\boldsymbol{\lambda}$ scaled with υ.

However, can give poor performance for smaller system sizes

Mean-field fluid model

The entire mean-field fluid model can then be expressed as

$$\dot{\boldsymbol{x}} = \boldsymbol{W}^T \theta(\boldsymbol{x}) + \boldsymbol{A}\boldsymbol{\lambda}$$
$$\boldsymbol{x}(0) = \boldsymbol{X}(0)$$

then $\lim_{v\to\infty} v^{-1}\boldsymbol{X}^{(v)} = \boldsymbol{x}$ at all t,
where $\boldsymbol{X}^{(v)}$ is $\boldsymbol{X}$ with k, $\boldsymbol{X}(0)$ and $\boldsymbol{\lambda}$ scaled with v.

However, can give poor performance for smaller system sizes

Improving the mean-field fluid model

Why is this?

Want $\boldsymbol{x}$ to approximate $\mathbb{E}(\boldsymbol{X})$ but

$$\frac{d}{dt}\mathbb{E}[\boldsymbol{X}] = \mathbb{E}[F(\boldsymbol{X})] \neq F(\mathbb{E}[\boldsymbol{X}]) = \frac{d}{dt}\boldsymbol{x}$$

the queuing network case

$$\mathbb{E}\left[\boldsymbol{W}^T\theta(\boldsymbol{X}) + \boldsymbol{A}\boldsymbol{\lambda}\right] = \boldsymbol{W}^T\mathbb{E}[\theta(\boldsymbol{X})] + \boldsymbol{A}\boldsymbol{\lambda} \neq \boldsymbol{W}^T\theta(\mathbb{E}[\boldsymbol{X}]) + \boldsymbol{A}\boldsymbol{\lambda}$$

Can we find another $\hat{\theta}(\mathbb{E}[\boldsymbol{X}])$ that improves accuracy?

Improving the mean-field fluid model

Why is this?

Want $\boldsymbol{x}$ to approximate $\mathbb{E}(\boldsymbol{X})$ but

$$\frac{d}{dt}\mathbb{E}[\boldsymbol{X}] = \mathbb{E}[F(\boldsymbol{X})] \neq F(\mathbb{E}[\boldsymbol{X}]) = \frac{d}{dt}\boldsymbol{x}$$

the queuing network case

$$\mathbb{E}\left[\boldsymbol{W}^T\theta(\boldsymbol{X}) + \boldsymbol{A}\boldsymbol{\lambda}\right] = \boldsymbol{W}^T\mathbb{E}[\theta(\boldsymbol{X})] + \boldsymbol{A}\boldsymbol{\lambda} \neq \boldsymbol{W}^T\theta(\mathbb{E}[\boldsymbol{X}]) + \boldsymbol{A}\boldsymbol{\lambda}$$

Can we find another $\hat{\theta}(\mathbb{E}[\boldsymbol{X}])$ that improves accuracy?

Improving the mean-field fluid model

Problem,

$$\mathbb{E}\left[\theta_{i,r,a}(\boldsymbol{X})\right] = \sum_{\boldsymbol{z}} \mathbb{P}\left(\boldsymbol{X} = \boldsymbol{z}\right) z_{i,r,a} \frac{\min(k_i, \sum_{s,b} z_{i,s,b})}{\sum_{s,b} z_{i,r,a}}$$

$$\theta_{i,r,a}\left(\mathbb{E}\left[\boldsymbol{X}\right]\right) = \mathbb{E}\left[X_{i,r,a}\right] \frac{\min\left(k_i, \sum_{s,b} \mathbb{E}\left[X_{i,s,b}\right]\right)}{\sum_{s,b} \mathbb{E}\left[X_{i,s,b}\right]}$$

First, let $\theta_{i,r,a}(\boldsymbol{X}) = X_{i,r,a} g_{i,r,a}(\boldsymbol{X})$,

$g_{i,r,a}(\boldsymbol{X})$ is the processor share of queue i and $g_{i,r,a}(\boldsymbol{X}) = g_{i,s,b}(\boldsymbol{X})$

Let $\hat{\theta}_{i,r,a}\left(\mathbb{E}\left[\boldsymbol{X}\right]\right) = \mathbb{E}\left[X_{i,r,a}\right] \hat{g}_{i,r,a}\left(\mathbb{E}\left[\boldsymbol{X}\right]\right)$, then by summing over all states/classes in queue i

$$\hat{g}_{i,r,a}\left(\mathbb{E}\left[\boldsymbol{X}\right]\right) = \frac{\sum_c \mathbb{P}\left(\sum_{s,b} X_{i,r,a} = c\right) \min\left(k_i, c\right)}{\sum_{s,b} \mathbb{E}\left[X_{i,s,b}\right]} = \frac{k_i \rho_i(\boldsymbol{X})}{\sum_{s,b} \mathbb{E}\left[X_{i,s,b}\right]}$$

Dependence on the PMF of $\boldsymbol{X}$, we need to allow $\hat{g}$ to change

Improving the mean-field fluid model

Problem,

$$\mathbb{E}\left[\theta_{i,r,a}(\boldsymbol{X})\right] = \sum_{\boldsymbol{z}} \mathbb{P}\left(\boldsymbol{X} = \boldsymbol{z}\right) z_{i,r,a} \frac{\min(k_i, \sum_{s,b} z_{i,s,b})}{\sum_{s,b} z_{i,r,a}}$$

$$\theta_{i,r,a}\left(\mathbb{E}\left[\boldsymbol{X}\right]\right) = \mathbb{E}\left[X_{i,r,a}\right] \frac{\min\left(k_i, \sum_{s,b} \mathbb{E}\left[X_{i,s,b}\right]\right)}{\sum_{s,b} \mathbb{E}\left[X_{i,s,b}\right]}$$

First, let $\theta_{i,r,a}(\boldsymbol{X}) = X_{i,r,a} g_{i,r,a}(\boldsymbol{X})$,

$g_{i,r,a}(\boldsymbol{X})$ is the processor share of queue i and $g_{i,r,a}(\boldsymbol{X}) = g_{i,s,b}(\boldsymbol{X})$

Let $\hat{\theta}_{i,r,a}\left(\mathbb{E}\left[\boldsymbol{X}\right]\right) = \mathbb{E}\left[X_{i,r,a}\right] \hat{g}_{i,r,a}\left(\mathbb{E}\left[\boldsymbol{X}\right]\right)$, then by summing over all states/classes in queue i

$$\hat{g}_{i,r,a}\left(\mathbb{E}\left[\boldsymbol{X}\right]\right) = \frac{\sum_c \mathbb{P}\left(\sum_{s,b} X_{i,r,a} = c\right) \min\left(k_i, c\right)}{\sum_{s,b} \mathbb{E}\left[X_{i,s,b}\right]} = \frac{k_i \rho_i(\boldsymbol{X})}{\sum_{s,b} \mathbb{E}\left[X_{i,s,b}\right]}$$

Dependence on the PMF of $\boldsymbol{X}$, we need to allow $\hat{g}$ to change

Improving the mean-field fluid model

One such possible function is

$$\hat{g}_{i,r,a}\left(\boldsymbol{x} \mid p_i\right) = \frac{1}{\left(1 + \left(k_i^{-1} \sum_{s,b} \boldsymbol{x}_{i,s,b}\right)^{p_i}\right)^{1/p_i}}$$

The inverse p-norm, can be seen as a smoothing of $g_{i,r,a}(\boldsymbol{X})$ with parameter p_i.

$p_i \to \infty$ gives back $g_{i,r,a}(\boldsymbol{X})$.

Nice because of monotonicity:

- given data at stationarity, "optimal" $\boldsymbol{p}^*$ can be found

Evaluation, M/M/1 queue

First considering the most simplistic queuing network,

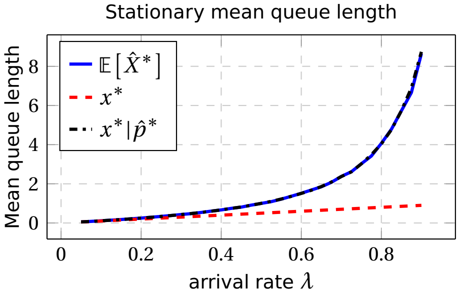
- a single queue with 1 server, 1 class and 1 phase.

The mean field model then gives $\dot{x} = -\mu \min(1, x) + \lambda$

stationary point: $x = \lambda/\mu = \rho \leq 1$,

However, true mean is well-known: $\mathbb{E}[X] = \frac{\rho}{1-\rho}$.

Evaluation, M/M/1 queue

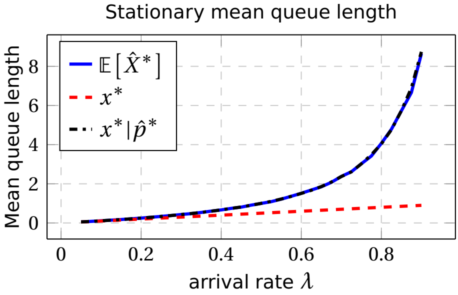


The p^* found is consistently around 1, which gives

$$\dot{x} = x \cdot \hat{g}\left(x \mid p = 1\right) + \lambda = \frac{x}{x+1} + \lambda$$

known as the Tipper model

Evaluation, M/M/1 queue

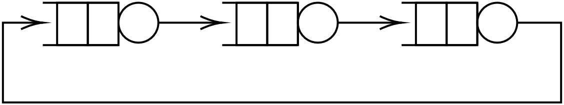


The p^* found is consistently around 1, which gives

$$\dot{x} = x \cdot \hat{g}\left(x \mid p = 1\right) + \lambda = \frac{x}{x+1} + \lambda$$

known as the Tipper model

Evaluation, three tandem queues

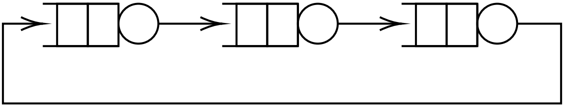


Three single class queues, queue 1 (INF) and queue 2 & 3 (PS)

$$W = \begin{bmatrix} -\mu_1 & \mu_1 & 0 & 0 & 0 \\ 0 & -4.0 & 4.0 & 0 & 0 \\ 0 & 0 & -4.0 & 4.0 & 0 \\ 1.9 & 0 & 0 & -2.0 & 0.1 \\ 0.1 & 0 & 0 & 0 & -0.1 \end{bmatrix}$$

$$\theta(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_2 \cdot \min(4, x_2 + x_3)/(x_2 + x_3) \\ x_3 \cdot \min(4, x_2 + x_3)/(x_2 + x_3) \\ x_4 \cdot \min(8, x_4 + x_5)/(x_4 + x_5) \\ x_5 \cdot \min(8, x_4 + x_5)/(x_4 + x_5) \end{bmatrix}$$

Evaluation, three tandem queues



Three single class queues, queue 1 (INF) and queue 2 & 3 (PS)

$$\boldsymbol{W} = \begin{bmatrix} -\mu_1 & \mu_1 & 0 & 0 & 0 \\ 0 & -4.0 & 4.0 & 0 & 0 \\ 0 & 0 & -4.0 & 4.0 & 0 \\ 1.9 & 0 & 0 & -2.0 & 0.1 \\ 0.1 & 0 & 0 & 0 & -0.1 \end{bmatrix}$$

$$\theta(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_2 \cdot \min(4, x_2 + x_3)/(x_2 + x_3) \\ x_3 \cdot \min(4, x_2 + x_3)/(x_2 + x_3) \\ x_4 \cdot \min(8, x_4 + x_5)/(x_4 + x_5) \\ x_5 \cdot \min(8, x_4 + x_5)/(x_4 + x_5) \end{bmatrix}$$

Evaluation, three tandem queues

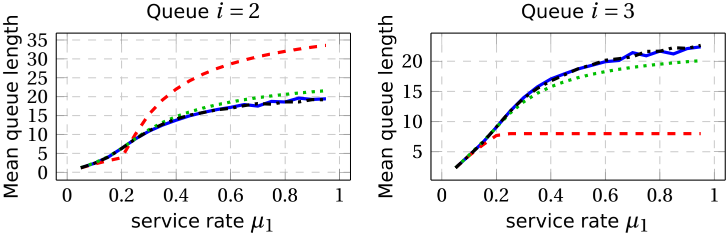


Figure: (Blue) queue length from simulation, (Red) mean-field model, (Black) smoothed model with $\boldsymbol{p}^*$ estimated at every μ_1, (Green) smoothed model with $\boldsymbol{p}$ estimated at $\mu_1 = 0.2$

Conclusion & future work

Conclusion

- Managing applications in the cloud is tricky
- Model using queuing networks, evaluate using fluid models
- Mean-field approximation for networks of PS queues
- Not necessarily good, consider using smoothed model

Next steps

- Test on a real system.
- How to construct a fluid model that tracks a running application.

Conclusion & future work

Conclusion

- Managing applications in the cloud is tricky
- Model using queuing networks, evaluate using fluid models
- Mean-field approximation for networks of PS queues
- Not necessarily good, consider using smoothed model

Next steps

- Test on a real system.
- How to construct a fluid model that tracks a running application.