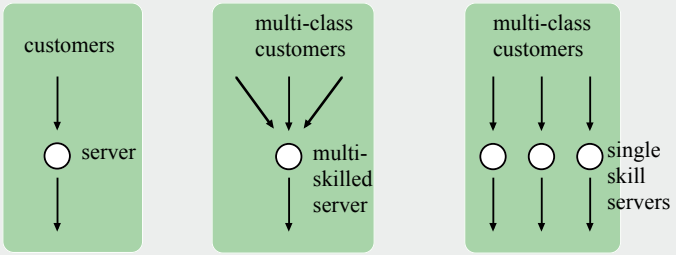
Queues with Skill Based Routing and Resource Pooling under First Come First Served / Assign Longest Idle Server Discipline

EURANDOM 2nd Dutch-Israeli Conference, September 2010

Gideon Weiss
University of Haifa
joint work with and work by:
Ivo Adan, Rene Caldentey, Cor Hurkens, Ed Kaplan,
Rishi Talreja, Jeremy Visschers, Damon Wischik, Ward Whitt

Skill based routing and resource pooling:

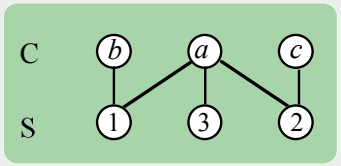


Multi-type servers with overlaps:

Customer types i=a,b,c, . . . S(i) servers of i

Server types j=1,2,. . .,K C(j) customers of j

Bipartite system graph
Custom Tailored Service
as well as Resource Pooling



Motivating Question 1: Infinite matching Kaplan '84,

In assigning housing to entitled families several housing projects are available (Kaplan, '84.'88).

> ". . . Households applying for public housing are allowed to specify those housing projects in which they are willing to live; when a public housing unit becomes newly available, of those households willing to live in the associated housing project, the one that has been waiting the longest is offered the unit. . . ."

FCFS discipline by law

OR question, game theoretic

How many choices should you mark ?

- Mark only 1st choice: long wait
- Mark several: shorter wait but worse allotment

if everyone marks many choices: long wait and bad allotment

Q: How many profile i applicants go to project j ?

Motivating Question 2: Overloaded system (Talreja Whitt)

Skill based routing in a call center type system:

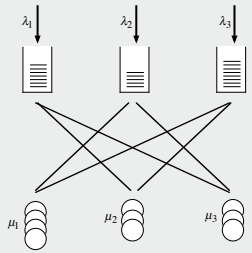
Overloaded systems, arrival rates $\Sigma \lambda_i > \Sigma \mu_j$

Stabilized by abandonment

general arrival streams, general service times,

patience distributions F_i

FCFS discipline



OR Question: How does it perform ?

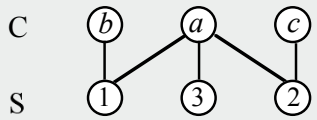
How should we assign skills and staff them ?

Q: How many type i customer get served by server of type j ?

Matching rates

When you use skill based routing,
How many type c customers get served by type j servers ?

EASY ?

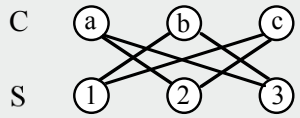


	β_1	β_2	β_3
α_a			
α_b		X	X
α_c	X		X

Matching rates

When you use skill based routing,
How many type c customers get served by type j servers ?

NOT EASY !!



	β_1	β_2	β_3
α_a	X		
α_b		X	
α_c			X

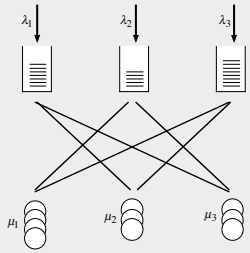
Overloaded system with abandonment (Talreja Whitt, ManSci'08)

Call Centers

Overloaded $\lambda=\Sigma\,\lambda_i > \Sigma\,\mu_j=\mu$

Abandonments: patience F_i

FCFS



Fluid Arguments:

Uniform acceleration - many server scaling:
servers are busy almost all the time
successive customers wait almost same time
on fluid scale we may get GLOBAL FCFS
- everyone with patience <W abandons
- everyone with patience >W waits exactly W.

$$\sum\lambda_i(1-F_i(W))=\sum\mu_j$$

Service rates in overloaded system

Question: find ν_{ij} the rate of services of i by j

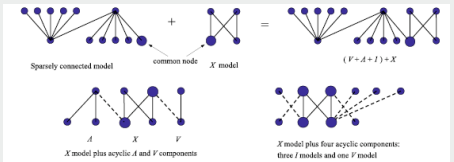
Solve for W from: $\sum\lambda_i(1-F_i(W))=\sum\mu_j$

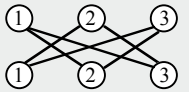
Then solve

$$\sum_{i\in C(j)}\nu_{ij}=\mu_j,\qquad j=1,\ldots,J$$
$$\sum_{j\in S(i)}\nu_{ij}=\lambda_i\bar{F}_i(W),\quad i=1,\ldots,I$$

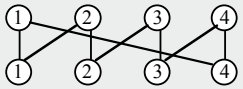
These may have no positive solution, unique solution or many solutions
- Can you solve it ?
- Does the stochastic system converge ?

Solved for trees, complete graphs, hybrids





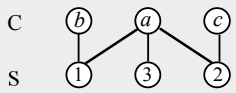
Non-Unique

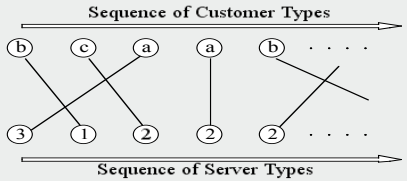


The Housing problem: Infinite matching (Caldentey Kaplan '02, Caldentey, Kaplan, W '09)

This is not a customer server situation ---
Housing units leave with the households and don't come back for many years

Other examples: Taxi-Cabs in the airport
Adoptions,
Kidney Transplants,
Auctions ?



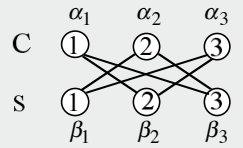


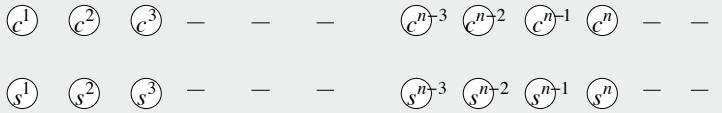
For any ω,
we have a unique
FCFS matching

NO TIMES INVOLVED
NO QUESTION OF LOAD
PURE MATCHING QUESION

FCFS Infinite matching - Markovian description

Data: i.i.d. α,β
graph G
FCFS





Fraction i->j matches in first n : $r^n_{i,j}$ $\qquad r^n_{i,j} \xrightarrow{a.s.} r_{i,j}$

Markovian description

Consider then s^{n+1} he will first look at those c left by $s^1 \ldots s^n$
If he finds a match he will take the first, and leave one less unmatched behind.
Else he will look for a match among those not considered previously,
until he finds a match, adding a geometric number of c from $C(s^{n+1})$.

Can we work with this ?

Define: X_n the Markov chain of: ordered list of unmatched customers
state space is words in alphabet of $i = 1,\ldots,I$

Conjecture: The Markov chain X is ergodic iff for all non-trivial S,C:

$$\alpha(C) < \beta(S(C)), \qquad \beta(S) < \alpha(C(S)), \qquad **$$

Theorem: If the Markov chain is ergodic then $r^n_{i,j} \xrightarrow[n\to\infty \; a.s.]{} r_{i,j}$

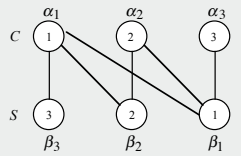
This chain is intractable, we could only do some special cases

Example 1 Almost complete graph --- each node connected to all but one

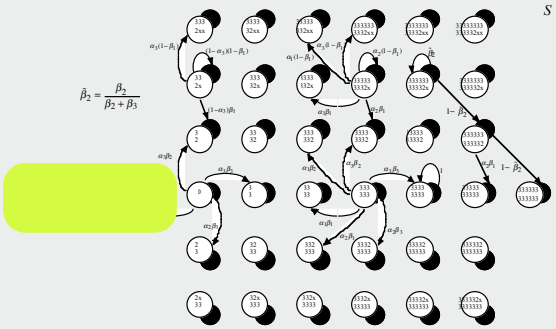
$$r_{i,j} = \frac{\alpha_i\beta_j\left[(1-\alpha_i)(1-\beta_j)-\alpha_j\beta_i\right]}{(1-\alpha_i-\beta_i)(1-\alpha_j-\beta_j)} \Bigg/ \left(1+\sum_{i=1}^{I}\frac{\alpha_i\beta_i}{1-\alpha_i-\beta_i}\right)$$

Example 2: `NN' model

Necessary condition for ergodicity: $\beta_3 < \alpha_1,\ \alpha_3 < \beta_1,$



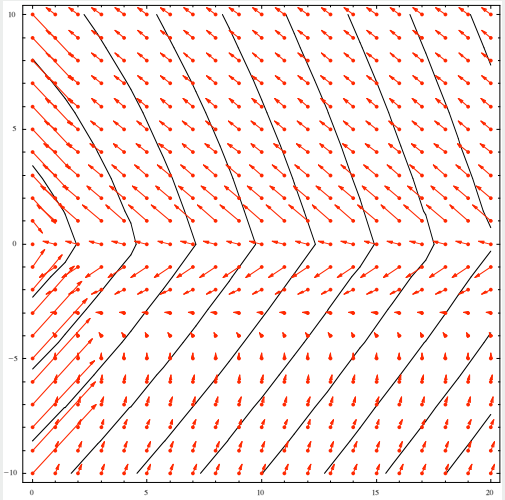
State space and transitions:



A Lyapunov function of Fayolle, Malyshev and Menshikov.

To show positive recurrence we need to show that $E(\Delta f(x,y)) \le -h < 0$

We were able to construct the right f (choice of u,v,w)

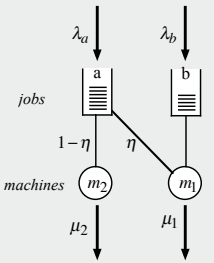


Back to Queueing - Manufacturing system `N' model

Adan, Foley, and Mcdonald, '08 `N' system queueing model:

- Memoryless,
- FCFS.

If system is empty, and type a arrives, machine 1 will take him w.p η



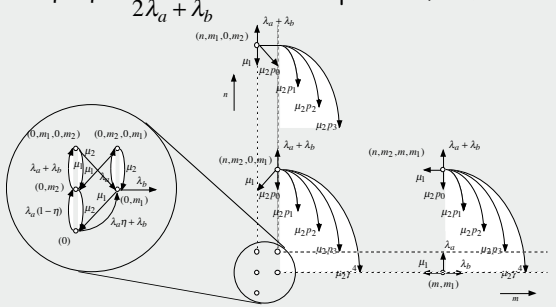
Typical state:
Machine m_1 serves first customer
x type b customers behind it
Machine m_2 serves next type a customer
y customers behind second machine

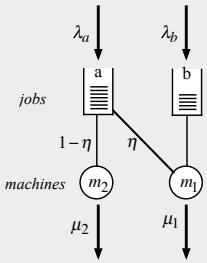
Results: steady state exact asymptotics, for large (x,y)

$$\Pi(x,y) \sim c(x/y)\left(\frac{\lambda_a+\lambda_b}{\mu_1+\mu_2}\right)^y \left(\frac{\lambda_b}{\mu_1}\right)^x$$

Product form solution for the `N' queueing system

When $\eta = \eta^* = \dfrac{\lambda_a}{2\lambda_a+\lambda_b}$ this has a product form solution:



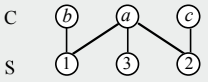


$$\alpha \equiv \frac{\lambda_a+\lambda_b}{\mu_1+\mu_2} < 1, \qquad \beta \equiv \frac{\lambda_b}{\mu_1} < 1,$$

$$\pi^*(x,y) \equiv \begin{cases} c^* & for\ (x,y)=(0,0), \\ c^*\frac{(1-\eta^*)\lambda_a}{\mu_2} & for\ (x,y)=(1,0), \\ c^*\frac{\eta^*\lambda_a+\lambda_b}{\mu_1}\beta^{y-1} & for\ x=0, y>0, \\ c^*\frac{\eta^*\lambda_a(\lambda_a+\lambda_b)^2}{\mu_1\mu_2}\alpha^{x-1} & for\ x>0, y=1, \\ c^*\frac{\lambda_a}{\mu_1+\mu_2}\frac{\eta^*\lambda_a+\lambda_b}{\mu_1}\alpha^{x-1}\beta^{y-1} & for\ x>0, y>1, \end{cases}$$

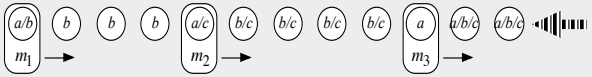
$$c^* \equiv 1/\left(1+\frac{(1-\eta^*)\lambda_a}{\mu_2}+\frac{\eta^*\lambda_a+\lambda_b}{\mu_1(1-\beta)}+\frac{\eta^*\lambda_a(\lambda_a+\lambda_b)^2}{\mu_1\mu_2(1-\alpha)}+\frac{\lambda_a(\eta^*\lambda_a+\lambda_b)\beta}{(\mu_1+\mu_2)\mu_1(1-\alpha)(1-\beta)}\right)$$

What worked - 4 skill based service models

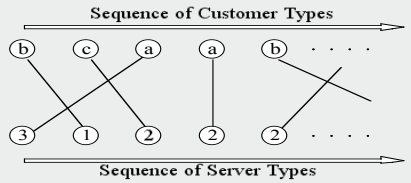


- Loss systems: Servers $m_1, m_2, \ldots, m_K$

- Stable FCFS queues:



- Infinite matching

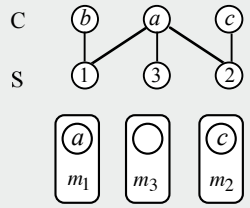


- Overloaded system with abandonment: $F_a, F_b, F_c, \ldots$

A reversible multi-type loss system (Adan, Hurkens, W '10)

Loss system:

State: set of idle machines $\{M_1, M_2, \ldots, M_k\}$



Assume MC reversible. Detailed balance:

out: $\{M_1, M_2, \ldots, M_{k-1}\} \to \{M_1, M_2, \ldots, M_k\}$ rate μ_{Mk}

In: $\{M_1, M_2, \ldots, M_k\} \to \{M_1, M_2, \ldots, M_{k-1}\}$ rate $\lambda_{M_k}\{M_1,\ldots,M_k\}$

$$\lambda_{M_k}\{M_1,\ldots,M_k\} = \sum_{i\in C(M_k)} \lambda_i \, P(i \text{ chooses server } M_k \mid \{M_1,\ldots,M_k\} \text{ idle})$$

$$\pi\{M_1,\ldots,M_{k-1}\}\mu_{M_k} = \pi\{M_1,\ldots,M_k\}\lambda_{M_k}\{M_1,\ldots,M_k\}$$

Solution:

$$\pi\{M_1,\ldots,M_k\} = \pi\{\varnothing\}\frac{\mu_{M_1}\ \mu_{M_2}\ \cdots\ \mu_{M_k}}{\lambda_{M_1}\{M_1\}\lambda_{M_2}\{M_1,M_2\}\cdots\lambda_{M_k}\{M_1,\ldots,M_k\}}$$

Assignment condition

Solution:

$$\pi_{Loss}\{M_1,\ldots,M_k\} = \pi\{\varnothing\}\frac{\mu_{M_1}\ \mu_{M_2}\ \cdots\ \mu_{M_k}}{\lambda_{M_1}\{M_1\}\lambda_{M_2}\{M_1,M_2\}\cdots\lambda_{M_k}\{M_1,\ldots,M_k\}}$$

Makes sense only if for all permutations:

$$\lambda_{M_1}\{M_1\}\lambda_{M_2}\{M_1,M_2\}\cdots\lambda_{M_k}\{M_1,\ldots,M_k\} = \lambda_{\bar{M}_1}\{\bar{M}_1\}\lambda_{\bar{M}_2}\{\bar{M}_1,\bar{M}_2\}\cdots\lambda_{\bar{M}_k}\{\bar{M}_1,\ldots,\bar{M}_k\}$$

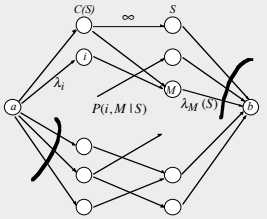
This uniquely determines

$$\lambda_M(S) = \sum_{i\in C(S)} \lambda_i \Big/ \left(1 + \sum_{j\in S} \frac{\lambda_j(S\setminus M)}{\lambda_M(S\setminus j)}\right)$$

Question: Can you find $P(i, M \mid S)$

$$\lambda_M(S) = \sum_{i\in C(M)} \lambda_i \, P(i, M \mid S)$$

Answer: Yes, always, max-flow of $\sum_{i\in C(S)} \lambda_i$ does it



This max-flow exists by monotonicity

$$\lambda_M(R) \ge \lambda_M(S), \quad M \in R \subset S$$

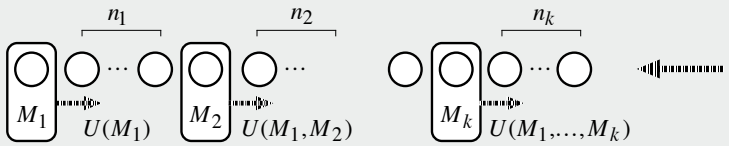
Skill based queues - FCFS (Visschers '00, Visschers, Adan, W '10)

Service exponential μ_j Graph **G**

Arrivals Poisson λ_i Poisson

FCFS service, **random assignment to idle servers**.

State $\mathbf{s} = (M_1, n_1, \cdots, M_k, n_k)$



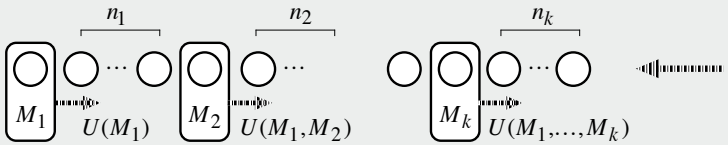
Thm: Use loss system assignment, then: MC obeys partial balance, product form:

$$\pi_{Queue}(\mathbf{s}) = \frac{\lambda_{M_1}\{M_1\cdots M_K\}\lambda_{M2}\{M_2\cdots M_K\}\cdots\lambda_{M_k}\{M_{k+1}\cdots M_K\}}{\mu_{M_1}(\mu_{M_1}+\mu_{M2})\cdots(\mu_{M_1}+\cdots\mu_{M_k})}\rho_1^{n_1}\cdots\rho_k^{n_k}$$

$$\rho_l = \frac{\lambda_{U\{M_1,\ldots,M_l\}}}{\mu_{M_1}+\cdots\mu_{M_l}}$$

Waiting time in queue

Using distributional form of Little's law we get the waiting time from arrival till start of service for a customer of type **c** :



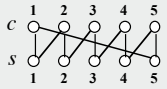
W_c is a mixture of sums of exponentials, each equal to an M/M/1 waiting time, as if customer is going through a tandem queue till he finds a server,

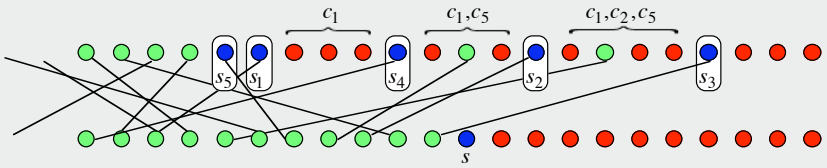
$$W_c \sim \pi_{Queue}\{M_1,\cdot,M_2,\cdot,\ldots,M_k,\cdot\}\exp(\lambda_{U\{M_1\cdots M_k\}} - \mu_{\{M_1\cdots M_k\}})*$$
$$*\exp(\lambda_{U\{M_1\cdots M_{k-1}\}} - \mu_{\{M_1\cdots M_{k-1}\}})*\cdots$$

Back to FCFS ∞−Matching : A new Markov chain

Server s^n sees the last match of customers to each type of server, and counts unmatched customers inbetween them:

$$\mathbf{s} = (s_5, 0, s_1, 3, s_4, 2, s_2, 3, s_3)$$





Stationary distribution:

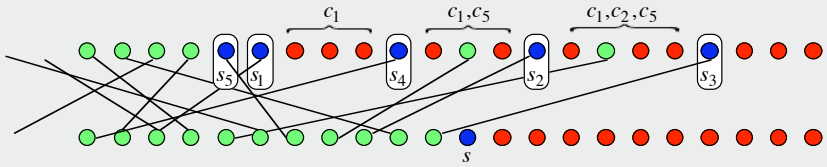
$$\pi_{Match}(\mathbf{s}) = B \prod_{k=1}^{J-1} \frac{\left(\alpha_{U\{M_1,\ldots,M_k\}}\right)^{n_k}}{\left(\beta_{M_1}+\ldots+\beta_{M_k}\right)^{n_k+1}}$$

MATCHING RATES – THE FORMULA

$$r_{c_i s_j} = \beta_{s_j} \sum_P B \left(\prod_{k=1}^{J-1} \frac{1}{\beta_{(k)} - \alpha_{(k)}} \right) \left(\left(\sum_{k=1}^{J-1} \phi_k \frac{\alpha_{(k)}}{\beta_{(k)} - \alpha_{(k)} \chi_k} \prod_{l=1}^{k-1} \frac{\beta_{(l)} - \alpha_{(l)}}{\beta_{(l)} - \alpha_{(l)} \chi_l} \right) + \frac{\phi_J}{\phi_J + \psi_J} \prod_{l=1}^{J-1} \frac{\beta_{(l)} - \alpha_{(l)}}{\beta_{(l)} - \alpha_{(l)} \chi_l} \right)$$

for permutation $(S_1, S_2, \ldots, S_J)$

$\alpha_{(k)} = \alpha_{U(S_1,\ldots,S_k)}$
$\beta_{(k)} = \beta_{S_1} + \cdots + \beta_{S_k}$
ϕ_k probability of i, j match with customer in k interval
ψ_k probability of wrong match with customer in k interval
χ_k probability of no match with customer in k interval



I tried to get expressions for special cases - just as horrible
I programmed it and calculated for I=J=7

SUSPICION: These quantities are #P-hard to evaluate

Symmetry lost ? Symmetry regained:

The Matching model is symmetric between customers and servers.
The Loss and Queueing models are not !

Consider now the loss system, replace random assignment with ALIS:
Assign arrivals to Longest Idle Servers

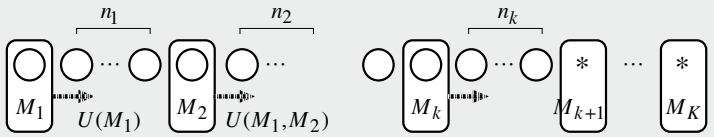
New system is not reversible, but is tractable:
State: $(M_1, M_2, \ldots, M_k)$ idle servers, ordered, M_k has been idle longest

$$\pi_{Loss}(M_1,\ldots,M_k) = B \frac{\mu_{M_1}\ \mu_{M_2}\ \cdots\ \mu_{M_k}}{\lambda_{C\{M_1,M_2\cdots M_k\}} \cdots \lambda_{C\{M_{k-1},M_k\}} \lambda_{C\{M_k\}}}$$

Adding over permutations we get the same as for random assignment

$$\pi_{Loss}\{M_1,\ldots,M_k\} = \pi\{\varnothing\} \frac{\mu_{M_1}\ \mu_{M_2}\ \cdots\ \mu_{M_k}}{\lambda_{M_1}\{M_1,\ldots,M_k\} \cdots \lambda_{M_{k-1}}\{M_{k-1},M_k\} \lambda_{M_k}\{M_k\}}$$

FCFS-ALIS Queueing model



State $\mathbf{s} = (M_1, n_1, \cdots, M_k, n_k, M_{k+1}, \cdots, M_K)$

$$\pi_{Queue}(\mathbf{s}) = B \frac{1}{\mu_{M_1}(\mu_{M_1}+\mu_{M2})\cdots(\mu_{M_1}+\cdots\mu_{M_k})\lambda_{C\{M_{i+1}\cdots M_K\}}\cdots\lambda_{C\{M_K\}}} \rho_1^{n_1}\cdots\rho_k^{n_k}$$

$$\rho_l = \frac{\lambda_{U\{M_1,\ldots,M_l\}}}{\mu_{M_1}+\cdots\mu_{M_l}}$$

Summing over permutations of the idle machines we get the same as for the random assignment model

Resource pooling

Stability of Queue:

$$\lambda(C) < \mu(S(C)) \quad for\ all\ \ C$$

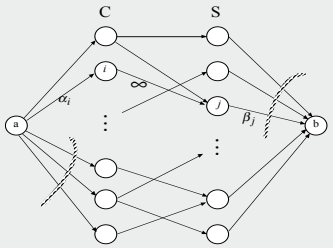
$$\lambda = \sum \lambda_i, \quad \mu = \sum \mu_j, \quad \rho = \frac{\lambda}{\mu}, \quad \alpha_i = \frac{\lambda_i}{\lambda}, \quad \beta_j = \frac{\mu_j}{\mu}$$

Queue is stable if ρ is small enough

complete resource pooling

$$\alpha(C) < \beta(S(C)), \qquad \beta(S) < \alpha(C(S)), \quad all\ non-trivial\ C$$

max-flow / min-cut



Overloaded system:

Conjecture: under complete resource pooling, when ρ > 1
all the servers stay together, just the last queue blows up:

$$\lim_{t\to\infty} P(M_1, n_1, \cdots, n_{K-1}, M_K) = \pi_{Match}(M_1, n_1, \cdots, n_{K-1}, M_K)$$
$$\lim_{t\to\infty} P(n_K < 100000) = 0$$

Overloaded system with abandonments:

Solve for global waiting time $\sum \lambda_i (1 - F_i(W)) = \sum \mu_j = \mu$

$$\alpha_i = \frac{\lambda_i (1 - F(W))}{\mu}, \quad \beta_j = \frac{\mu_j}{\mu}$$

Conjecture: if overloaded system with abandonment has complete resource pooling then under uniform acceleration scaled state is distributed like

$$\pi_{Match}(M_1, n_1, \cdots, n_{K-1}, M_K)$$