Service Engineering

Class 9

Stochastic Markovian Service Station in Steady State - Part I: Classical Queueing Birth & Death Models

- Service Engineering: Starting to Close the Circle.
- Workforce Management (WFM): Hierarchical Operational View.
- 4CallCenters – A Personal Tool for Workforce Management.
- Markov Jump Processes (MJP): Ergodicity; The Method of Cuts; Reversibility.
- The M/M/1 Queue.
- Infinite-Server Queues (M/M/∞).
- The Erlang-C (M/M/n) model.
- Pooling.
- Queueing Models with Blocking.

Service Engineering: A Subjective View

Goal (Subjective):
Develop scientifically-based design principles (**rules-of-thumb**) and tools (**software**) that support the balance of service **quality**, process **efficiency** and business **profitability**, from the (often conflicting) views of customers, servers and managers.

Contrast/Complement the traditional and prevalent

- Service Management (U.S. Business Schools)
- Industrial Engineering (European/Japanese Engineering Schools)

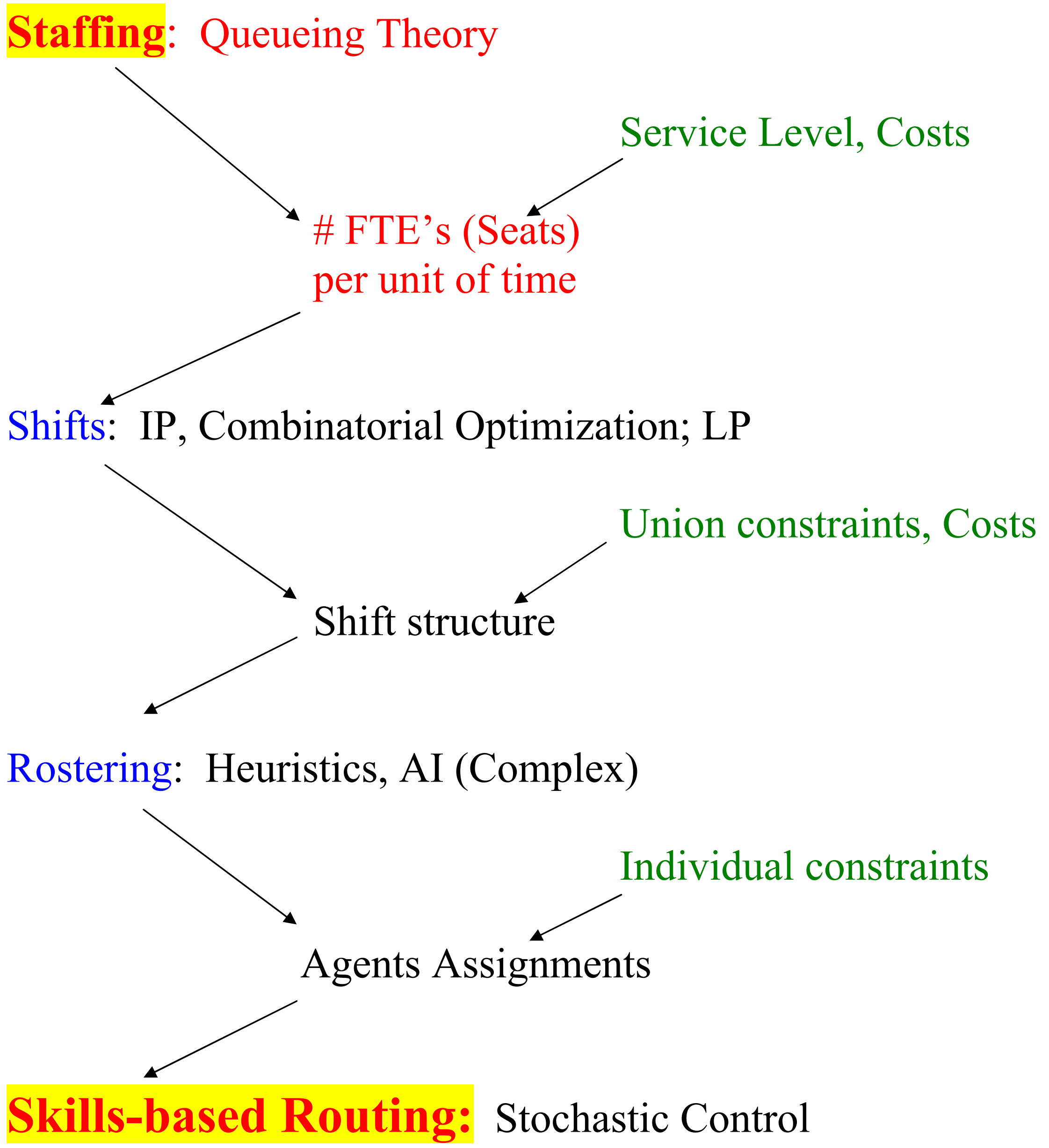
Examples:

- **Staffing** - How many agents required for balancing service-quality with operational efficiency (or, for maximizing profit).
- **Skills-Based Routing (SBR)** - Platinum and Gold and Silver customers, all seeking Information or Purchase or Technical Support, via Telephone or IVR or e.mail of Chat.
- Service Process **Design** + Staffing + SBR.

Recipe for Progress in Research, Teaching, Applications:
Simple Models at the Service of Complex Realities, with a pinch of a Multidisciplinary View (Operations, HRM, Marketing, MIS) = **Service Engineering**.

Workforce Management (WFM): Hierarchical Operational View

Forecasting Customers: Statistics, Time-Series
Agents : HRM (Hire, Train; Incentives, Careers)



Staffing: Queueing Theory

Service Level, Costs

# FTE's (Seats)
per unit of time

Shifts: IP, Combinatorial Optimization; LP

Union constraints, Costs

Shift structure

Rostering: Heuristics, AI (Complex)

Individual constraints

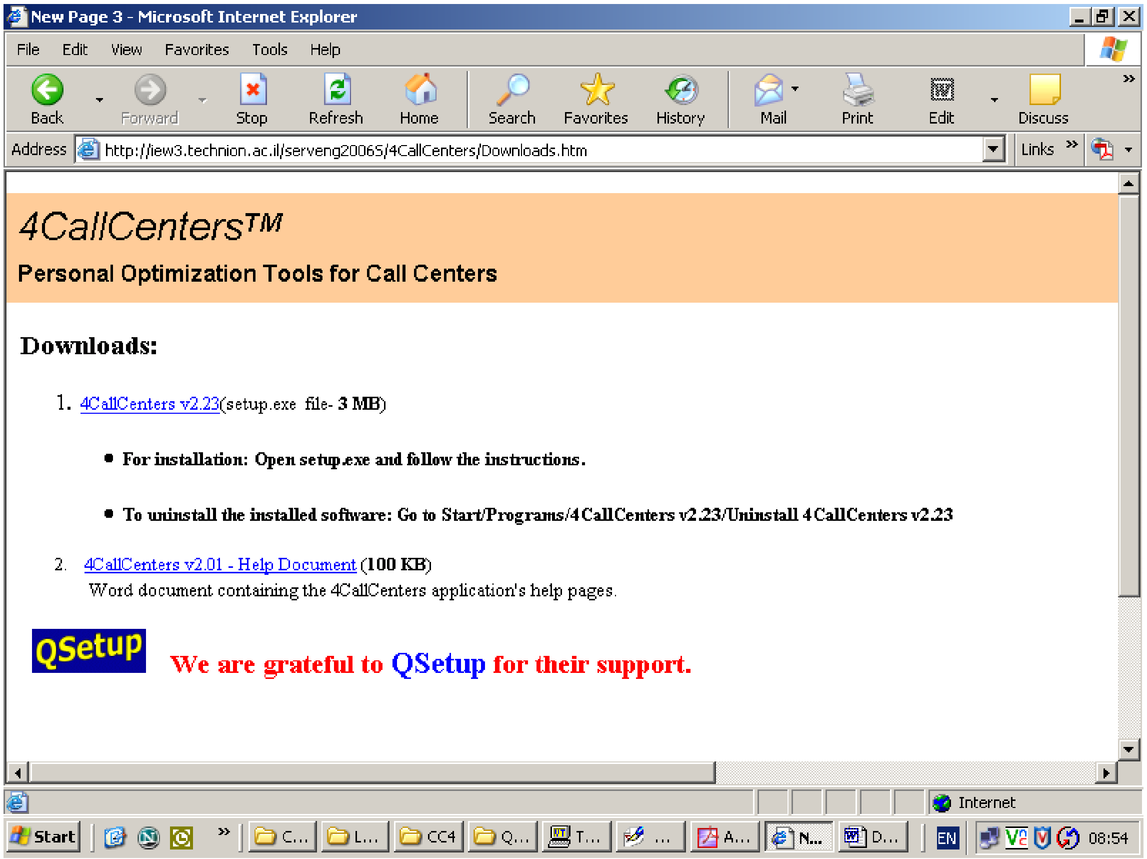
Agents Assignments

Skills-based Routing: Stochastic Control

Software Tool: 4CallCenters (4CC)

- Mathematical Engine: Technion M.Sc. thesis of Ofer Garnett.
- Used in Germany, India, Brazil, ..., Israel.
- Important tool in our course.
- Free download from

http://iew3.technion.ac.il/serveng/4CallCenters/Downloads.htm


New Page 3 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address
Links
Internet
Start
C... L... CC4 Q... T... ... A... N... D...
EN
08:54

http://iew3.technion.ac.il/serveng2006S/4CallCenters/Downloads.htm

4CallCenters™

Personal Optimization Tools for Call Centers

Downloads:

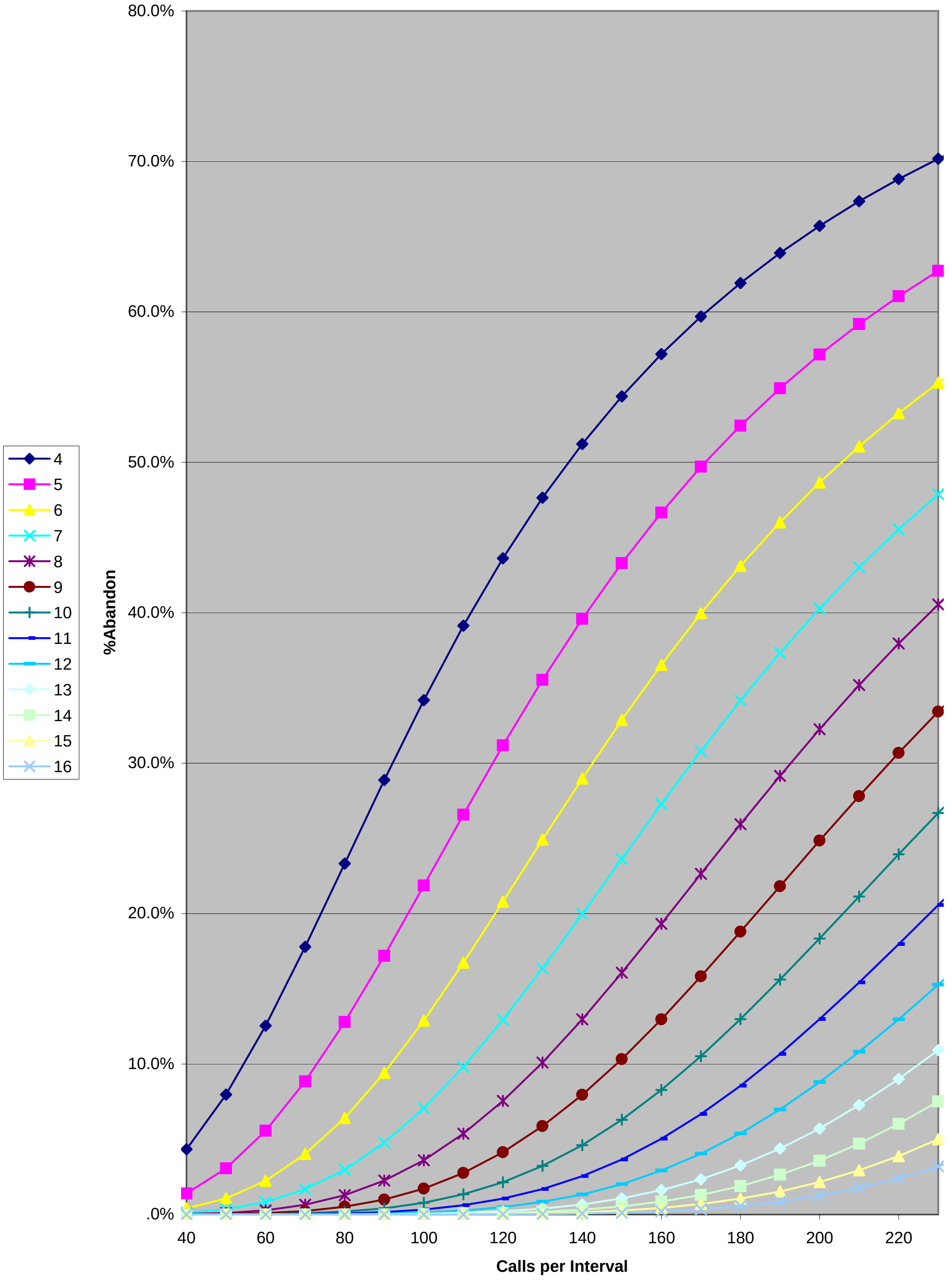
1. 4CallCenters v2.23(setup.exe file- **3 MB**)
   - **For installation: Open setup.exe and follow the instructions.**
   - **To uninstall the installed software: Go to Start/Programs/4CallCenters v2.23/Uninstall 4CallCenters v2.23**
2. 4CallCenters v2.01 - Help Document (**100 KB**)
   Word document containing the 4CallCenters application's help pages.

QSetup **We are grateful to QSetup for their support.**

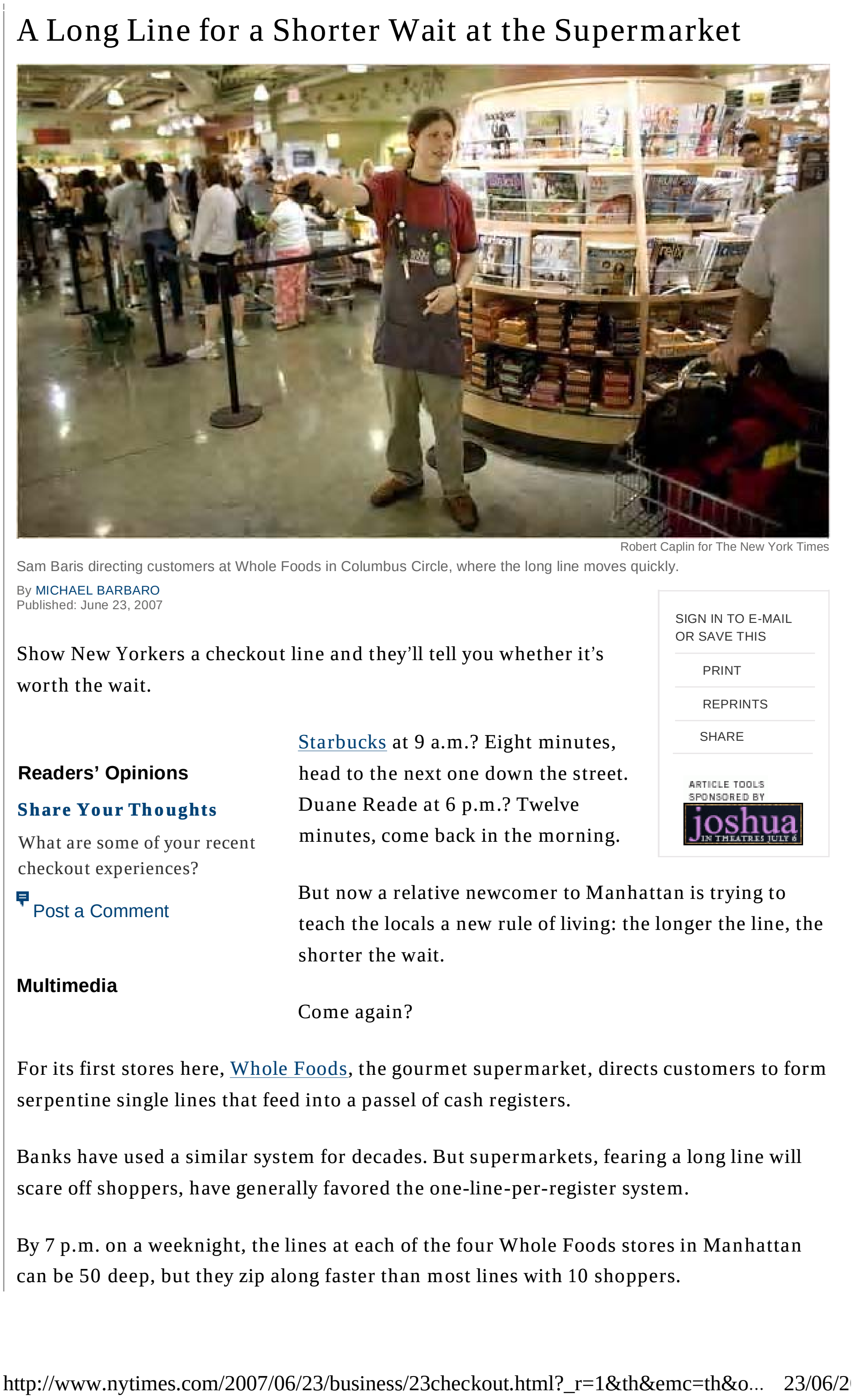
Example of 4CC Output: Congestion Curves

% Abandon vs. Calls/Hour for various Number of Agents

(E[Service] = 3:30 min, E[(Im)Patience] = 6:00 min.)



Pooling Queues at a NYC Supermarket



A Long Line for a Shorter Wait at the Supermarket

Robert Caplin for The New York Times

Sam Baris directing customers at Whole Foods in Columbus Circle, where the long line moves quickly.

By MICHAEL BARBARO
Published: June 23, 2007

Show New Yorkers a checkout line and they'll tell you whether it's worth the wait.

Starbucks at 9 a.m.? Eight minutes, head to the next one down the street. Duane Reade at 6 p.m.? Twelve minutes, come back in the morning.

But now a relative newcomer to Manhattan is trying to teach the locals a new rule of living: the longer the line, the shorter the wait.

Come again?

For its first stores here, Whole Foods, the gourmet supermarket, directs customers to form serpentine single lines that feed into a passel of cash registers.

Banks have used a similar system for decades. But supermarkets, fearing a long line will scare off shoppers, have generally favored the one-line-per-register system.

By 7 p.m. on a weeknight, the lines at each of the four Whole Foods stores in Manhattan can be 50 deep, but they zip along faster than most lines with 10 shoppers.

Readers' Opinions

Share Your Thoughts

What are some of your recent checkout experiences?

Post a Comment

Multimedia

SIGN IN TO E-MAIL OR SAVE THIS

PRINT

REPRINTS

SHARE

http://www.nytimes.com/2007/06/23/business/23checkout.html?_r=1&th&emc=th&o... 23/06/2

Markov Jump Processes: Brief Review

Characterization: $i, j \in S$;
$\pi^0 = (\pi_i^0)$ - initial distribution, $Q = [q_{ij}]$ - generator matrix.

Steady-state equations: $\left\{ \begin{array}{l} 0 = \pi Q \\ \Sigma_i \pi_i = 1, \ \pi_i \geq 0 \end{array} \right\}$

Ergodic Theorem: Let X be *irreducible* ($i \leftrightarrow j$). Assume that there exists a solution π to its steady-state equations. Then π is its (unique) stationary and limit-distribution.

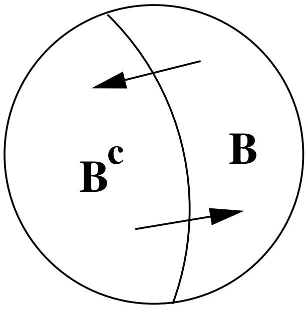
Transition rates: $\pi_i q_{ij}$ = long-run (steady-state) transition-rate (number of transitions per unit of time) from i to j.

Balance equations: For each state $j \in S$, its long-run (steady-state) entry-rate equals its exit-rate. Formally,

$$\sum_{i \neq j} \pi_i q_{ij} = -\pi_j q_{jj} = \sum_{i \neq j} \pi_j q_{ji}, \ \forall j.$$

Cuts: $\forall B \subset \mathcal{S}$, the long-run (steady-state) transition rate from B to B^c equals that from B^c to B. Formally,

$$\Sigma_{i \in B} \Sigma_{j \in B^c} \pi_i q_{ij} = \Sigma_{i \in B^c} \Sigma_{j \in B} \pi_i q_{ij}.$$



Markov Jump Processes: Time-Reversibility, Birth & Death.

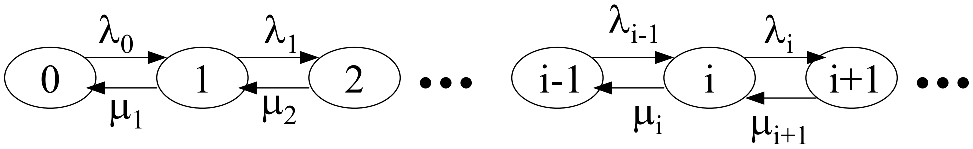
Reversibility: A stochastic process $X = \{X_t,\ -\infty < t < \infty\}$ is called *reversible* if for any r

$$\{X_t,\ 0 \le t \le r\} \overset{d}{=} \{X_{r-t},\ 0 \le t \le r\}\,.$$

Fact. Ergodic MJP in steady-state is reversible if and only if the *detailed-balance* equations hold:

$$\pi_i q_{ij} \;=\; \pi_j q_{ji}\,, \qquad \forall\, i,j \in \mathcal{S}\,.$$

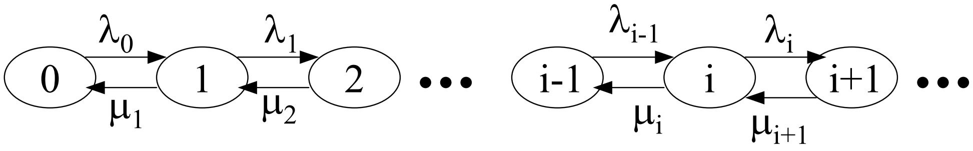
Birth & Death process: MJP on $S = \{0,1,2,\ldots\}$, with jumps only between adjacent states: $q_{ij} = 0$ if $|i-j| > 1$.



Cuts: $\pi_i\lambda_i = \pi_{i+1}\mu_{i+1} \qquad (\pi_i q_{i,i+1} = \pi_{i+1} q_{i+1,i})$.
(Take $B = \{0,1,\ldots,i\}$ and $B^c = \{i+1, i+2, \ldots\}$.)

Corollary. Every ergodic Birth & Death process is reversible. (Follows from the cut-equations.)

Service Station: Birth & Death Animation



- i – number-in-system;
- λ_i – arrival rate, with i customers in system;
- μ_i – service rate, with i customers in system.

Cuts at $i \leftrightarrow i+1$: $\pi_i \lambda_i = \pi_{i+1} \mu_{i+1},\ i \geq 0$,
which yields

$$\pi_{i+1} = \frac{\lambda_i}{\mu_{i+1}}\,\pi_i = \frac{\lambda_i \lambda_{i-1}}{\mu_{i+1}\mu_i}\,\pi_{i-1} = \cdots = \frac{\lambda_0 \lambda_1 \ldots \lambda_i}{\mu_1 \mu_2 \ldots \mu_{i+1}}\,\pi_0 \ .$$

Stability: Steady-state (Limit) distribution exists if and only if

$$\sum_{i=0}^{\infty} \frac{\lambda_0 \ldots \lambda_i}{\mu_1 \ldots \mu_{i+1}} < \infty \ ,$$

in which case it is given by

$$\begin{cases} \pi_i & = \frac{\lambda_0 \ldots \lambda_{i-1}}{\mu_1 \ldots \mu_i}\,\pi_0 \ , \ i \geq 0, \\ \pi_0 & = \left[\sum_{i\geq 0} \ \frac{\lambda_0 \ldots \lambda_i}{\mu_1 \ldots \mu_{i+1}}\right]^{-1} \end{cases}$$

Classical Markovian Queues

Assumptions (from now on):

- n **statistically identical independent (iid) servers**;
- **FCFS discipline** – First Come First Served;
- **Work conservation**: a server does not go idle if there are customers in need of service;
- Arriving customers all join and remain till end of service **(do not abandon**).

Queueing Notations: $G_a/G_s/n/K$, where

- G_a: General Arrivals (M for Poisson arrivals),
- G_s: General Services (M for Exponential service times),
- n = number of servers,
- K = maximal number in system.

Next: $M/M/1, M/M/\infty, M/M/n, M/M/n/k, M/M/n/n.$

Measures of Performance (Steady-State, Long-Run)

- $\lambda = \Sigma_{i \geq 0}\, \pi_i \lambda_i$ - arrival rate = service rate - $\mu = \Sigma_{i \geq 1}\, \pi_i \mu_i$
- L - number of customers in system (sometimes L_s);
- L_q - number of customers in queue;
- W - **sojourn time** through the system (W_s);
- W_q - **waiting time** in the queue.

In steady state (in the long run),

$$\mathrm{E}[L] \;=\; \sum_{k \geq 0} k \cdot \pi_k \;=\; \lim_{T \to \infty} \frac{1}{T} \cdot \int_0^T L(t) dt\,.$$

$$\mathrm{E}[L_q] = \sum_{k=n+1}^{\infty} (k-n) \cdot \pi_k\,.$$

Little's formula yields average times:

$$\mathrm{E}[L] = \lambda \cdot \mathrm{E}[W]; \qquad \mathrm{E}[L_q] = \lambda \cdot \mathrm{E}[W_q]\,.$$

Average service time, E[S], must satisfy:

$$\mathrm{E}[W] \;=\; \mathrm{E}[W_q] + \mathrm{E}[S]\,.$$

Review: MJP 1

MJP $X = \{X_t,\ t \geq 0\}$ on $\mathcal{S} = \{i, j, \ldots\}$ countable.
Markov property: $P_r\{X_t = j | X_r,\ r < s;\ X_s = i\} = P_{ij}(s,t), \quad \forall\, s < t,\ \forall\, i, j \in \mathcal{S}$.
Time homogeneity: $P_r\{X_{s+t} = j | X_s = i\} = P_{ij}(t), \quad \forall\, s, t,\ i, j,$ transition probabilities.

Characterization: π^0 = initial distribution and $P(t) = [P_{ij}(t)],\ t \geq 0$, stochastic.
Finite-dimensional distributions:
$P_r\{X_0 = i_0,\ X_{t_1} = i_1, \ldots, X_{t_n} = i_n\} = \pi^0(i_0) P_{i_0,i_1}(t_1) \ldots P_{i_{n-1},i_n}(t_n - t_{n-1})$.

$P(t)$: stochastic ; $P(s+t) = P(s)P(t), \quad \forall\, s, t$ (Chapman Kolmogorov);
$\exists\, P(0) = I$; $\exists\, \dot{P}(0) = Q = [q_{ij}]$, infinitesimal generator $\left(\sum_{j \in \mathcal{S}} q_{ij} = 0\right)$.

Micro to Macro : $\dot{P}(t) = P(t)Q \ \ (= QP(t))$ and $P(0) = I$
Forward (Backward) equations.

Solution : $P(t) = \exp[tQ] = \sum_{n=0}^{\infty} \frac{t^n}{n!} Q^n$, $t \geq 0$.

Animation: $i \xrightarrow{q_{ij}} j; \quad \forall\, i, j \in \mathcal{S}\ \exists$ exponential clock at rate q_{ij}, call it (i, j).
Given i, consider clocks $(i, j),\ j \in \mathcal{S}$; move to the "winner" when rings.
Thus: stay at $i \sim \exp(q_i = \sum_{j \neq i} q_{ij})$ and switch to j with probability $P_{ij} = q_{ij}/q_i$ $(q_{ij} = q_i P_{ij}, i \neq j; q_{ii} = -q_i)$.

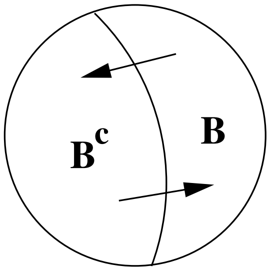
Transient analysis vs. long-run/limit stability/steady-state
$\exists \lim_{t \uparrow \infty} P_{ij}(t) = \pi_j,\ \forall\, i;$ $\pi = \pi P(t),\ \forall\, t$.

Calculation via **steady-state equations:** $\dot{P}(\infty) = P(\infty)Q \Rightarrow \left\{ \begin{array}{l} 0 = \pi Q \\ \sum_i \pi_i = 1,\ \pi_i \geq 0 \end{array} \right\}$

or balance equations: $\sum_{i \neq j} \pi_i q_{ij} = -\pi_j q_{jj} = \sum_{i \neq j} \pi_j q_{ji},\ \forall\, j$.

Transition rates: $\pi_i q_{ij}$ = long-run average number of switches from i to j.

Cuts: $\sum_{i \in B} \sum_{j \in B^c} \pi_i q_{ij} = \sum_{i \in B^c} \sum_{j \in B} \pi_i q_{ij},\ \forall\, B \subset \mathcal{S}$.



Review: MJP 2

Ergodic Theorem: Let X be *irreducible* ($i \leftrightarrow j$). Assume that there exists a solution π to its steady-state equations. Then, X must be "unexplosive" and π must be its stationary distribution, its limit distribution and

SLLN •$\lim\limits_{T\uparrow\infty} \frac{1}{T} \int_0^T f(X_t)dt = \sum\limits_i \pi_i f(i)$ ("="$Ef(X_\infty)$) ; eg. $f(x) = 1_B(x)$.

•$\lim\limits_{T\uparrow\infty} \frac{1}{T} \sum\limits_{t\le T} g(X_{t-}, X_t) = \sum\limits_i \pi_i \sum\limits_j q_{ij} g(i,j)$, for $g(x,x) = 0$, $\forall\, x$; e.g. $g(x,y) = 1_C(x,y)$.

Birth-and-death process: MJP on $S = \{0,1,2,\ldots\}$, where all jumps are between adjacent states: $q_{ij} = 0$ if $|i-j| > 1$.

Cuts: $\pi_i q_{i,i+1} = \pi_{i+1} q_{i+1,i}$.
(Take $B = \{0,1,\ldots,i\}$ and $B^c = \{i+1, i+2, \ldots\}$.)

Reversibility: A stochastic process $X = \{X_t,\ -\infty < t < \infty\}$ is called *reversible* if for any τ

$$\{X_t,\ -\infty < t < \infty\} \overset{d}{=} \{X_{\tau - t},\ -\infty < t < \infty\}\,.$$

Fact. Ergodic MJP in steady-state is reversible if and only if the *detailed balance equations* hold:

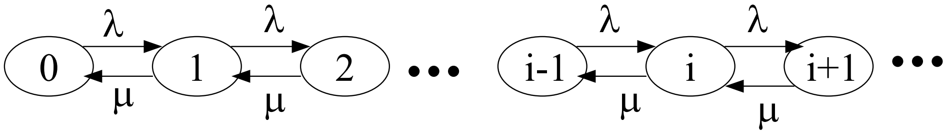
$$\pi_i q_{ij} \;=\; \pi_j q_{ji}\,, \qquad \forall\, i,j \in \mathcal{S}\,.$$

Corollary. Every ergodic birth-and-death process is reversible.

(Follows from the cut equations.)

The M/M/1 Queue

- Poisson arrivals, at rate λ;
- Single exponential server, at rate μ ($\mathrm{E}[S] = 1/\mu$).



Transition rates: $\lambda_i \equiv \lambda,\ i \geq 0; \quad \mu_i \equiv \mu,\ i \geq 1.$

Cut equations: $\lambda \pi_i = \mu \pi_{i+1}\,, \qquad i \geq 0\,.$

Traffic intensity $\rho = \frac{\lambda}{\mu} < 1$ (iff stability).

Steady-state distribution:

$L \overset{d}{=}$ **Geometric**$(p = 1 - \rho)$ (from 0):

$$\pi_i = (1 - \rho)\rho^i, \qquad i \geq 0.$$

Hence, can calculate:

$\mathrm{E}[L] = \rho/(1-\rho)$, then $\mathrm{E}[W]$, then $\mathrm{E}[W_q]$, finally $\mathrm{E}[L_q]$.

Insightful calculations:

Effective (actual) service rate =

$\sum_{n \geq 1} \pi_n \cdot \mu = \mu(1 - \pi_0) = \mu \cdot \rho = \mu \cdot \frac{\lambda}{\mu} = \lambda.$

Contrast with

Service Capacity (potential service rate) = μ.

M/M/1: Sojourn Time

Sojourn time is **exponentially** distributed:

$$W \overset{d}{=} \exp\left(\text{mean} = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu}\left[1 + \frac{\rho}{1-\rho}\right]\right).$$

Proof: By PASTA and the memoryless-properly of Exponentials,

$$W \overset{d}{=} \sum_{i=1}^{N} X_i\,, \qquad X_i \sim \exp(\mu) \;\; \text{i.i.d.}\,,$$

$N \triangleq L + 1 \overset{d}{=} \text{Geom}(1-\rho)$ (from 1);
N and $\{X_i\}$ are all independent.

Conclude by recalling: Geometric sum of iid Exponentials is Exponentially distributed. (The parameter is calculated via Wald.)

Aside: The latter property is essentially Poisson-Splitting. A self-contained proof can be give via *Moment generating functions:*

$$\begin{aligned}
\phi_W(t) &\triangleq \mathrm{E}\left[\exp\{tW\}\right] = \mathrm{E}\left[\exp\{t \cdot \sum_{i=1}^{N} X_i\}\right] \\
&= \mathrm{E}\left[\mathrm{E}\left[\exp\{t \cdot \sum_{i=1}^{N} X_i\}\middle|\, N\right]\right] \\
&\quad (X_i \text{ independent with } \mathrm{E}\left[\exp\{tX_i\}\right] = \tfrac{\mu}{\mu-t}) \\
&= \mathrm{E}\left[\left(\frac{\mu}{\mu-t}\right)^{N}\right] = \sum_{k=1}^{\infty}(1-\rho)\rho^{k-1}\left(\frac{\mu}{\mu-t}\right)^{k} \\
&= \frac{\mu(1-\rho)}{\mu-t} \cdot \sum_{k=0}^{\infty}\left(\frac{\mu\rho}{\mu-t}\right)^{k} = \frac{\mu(1-\rho)}{\mu(1-\rho)-t} \\
&= \phi_{\exp(\mu(1-\rho))}(t)\,.
\end{aligned}$$

M/M/1: Further Properties

- **Delay probability** (PASTA):

$$\mathrm{P}\{W_q > 0\} \;=\; \rho\,.$$

- **Waiting time in queue** (given delay, it is exp):

$$\frac{W_q}{1/\mu} \overset{d}{=} \begin{cases} 0 & \text{wp } \; 1-\rho \\ \exp\left(\text{mean} = \frac{1}{1-\rho}\right) & \text{wp } \; \rho \end{cases}$$

Note: $\mathrm{E}[\frac{W_q}{1/\mu}] = 0 \times (1-\rho) + \frac{1}{1-\rho} \times \rho = \frac{\rho}{1-\rho}$.

- **Number-in-system/queue:**

$$\mathrm{E}[L] = \frac{\rho}{1-\rho}\,; \qquad \mathrm{E}[L_q] = \frac{\rho^2}{1-\rho}\,.$$

- **Server's utilization** (occupancy) is $\rho = \lambda/\mu$.

Via Little's formula, applied to "system = server":

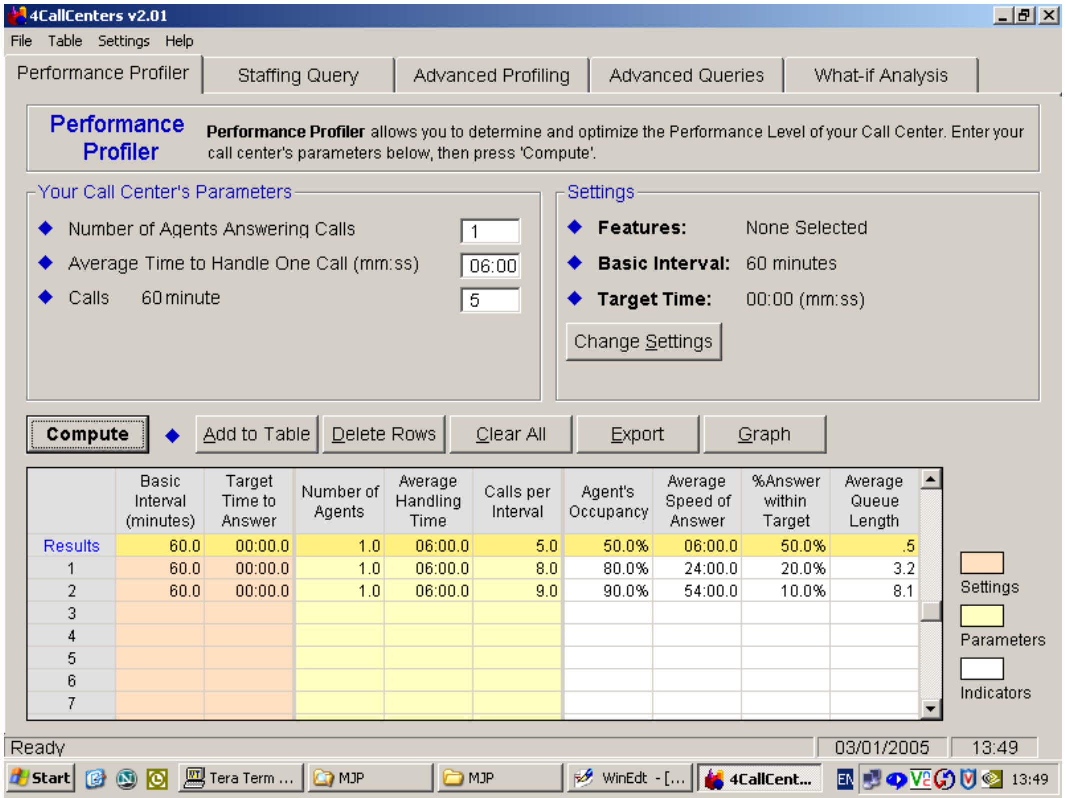
$$\rho = \lim_{T\to\infty} \frac{1}{T} \int_0^T L_{Server}(u)\, du = \lambda \times \frac{1}{\mu}\,.$$

- **Departure process** in steady state is Poisson (λ) (Burke's theorem) – useful in queueing networks.

Support: Reversibility implies that the departure process equals (in distribution) the arrival process. Furthermore,
Average inter-departure time =

$$\frac{1}{\mu} \cdot \rho + \left(\frac{1}{\lambda} + \frac{1}{\mu}\right) \cdot (1-\rho) \;=\; \frac{1}{\lambda}\,.$$

M/M/1 Queue: 4CallCenters



Performance Profiler

Performance Profiler allows you to determine and optimize the Performance Level of your Call Center. Enter your call center's parameters below, then press 'Compute'.

Your Call Center's Parameters

- Number of Agents Answering Calls: 1
- Average Time to Handle One Call (mm:ss): 06:00
- Calls 60 minute: 5

Settings

- **Features:** None Selected
- **Basic Interval:** 60 minutes
- **Target Time:** 00:00 (mm:ss)

	Basic Interval (minutes)	Target Time to Answer	Number of Agents	Average Handling Time	Calls per Interval	Agent's Occupancy	Average Speed of Answer	%Answer within Target	Average Queue Length
Results	60.0	00:00.0	1.0	06:00.0	5.0	50.0%	06:00.0	50.0%	.5
1	60.0	00:00.0	1.0	06:00.0	8.0	80.0%	24:00.0	20.0%	3.2
2	60.0	00:00.0	1.0	06:00.0	9.0	90.0%	54:00.0	10.0%	8.1
3									
4									
5									
6									
7									

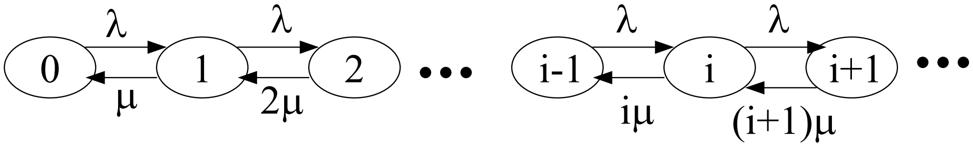
Average Waiting Times =
$E[S]$, for $\rho = 50\%$; $4 \cdot \mathrm{E}[S]$, for $\rho = 80\%$;
$9 \cdot \mathrm{E}[S]$, for $\rho = 90\%$; $\ldots$, $19 \cdot \mathrm{E}[S]$, for $\rho = 95\%$ (via model).

4CallCenters, performance measures:

- Average Speed of Answer (ASA) = $\mathrm{E}[W_q]$
  (will be very different with abandonment);
- % Answer within Target = $\mathrm{P}\{W_q \le T\}$;
  ($T = 0$ important, as we'll learn later, but hardly used.)
- Average Queue Length = $\mathrm{E}[L_q]$.

M/M/∞ Queue (Ample Servers)

- Poisson arrivals, rate λ;
- Infinite number of exponential servers, rate μ $(E(S) = 1/\mu)$.



$\lambda_i \equiv \lambda, \quad i \geq 0; \qquad \mu_i = i \cdot \mu, \quad i \geq 1.$

Cut equations:

$$\lambda \cdot \pi_i = (i+1)\mu \cdot \pi_{i+1}\,, \qquad i \geq 0\,.$$

Always stable.

Steady-state distribution is **Poisson**:

$$\pi_i = e^{-R} \cdot \frac{R^i}{i!}\;,\; i \geq 0\,;$$

$R = \dfrac{\lambda}{\mu}$ is the **offered load**, (measured in units of **Erlangs**)
= Average amount of work-units (time-units of service) that arrives to the system per time-unit
= Average number of customers in steady-state (via Little's Law).

Offered-Load Extension to a **time-varying environment**:
Consider $M_t/M/\infty$ (time-inhomogeneous Poisson arrivals, at rate $\lambda(t), t \geq 0$):

$$R(t) = E\int_{t-S}^{t} \lambda(u)\, du = E\lambda(t-S)\;, \quad t \geq 0\,.$$

M/M/∞ Queue: Continued

Very *useful*: ∞-server models provide bounds (ideal):
Recall, DS-PERT's with Infinite-Servers.
Later, queues with abandonment.

Average **number-in-system** in steady-state
(via Little's formula):

$$\mathrm{E}[L] \;=\; \mathrm{E}(\#\text{ Busy Servers}) \;=\; \lambda \times \frac{1}{\mu} \;=\; R\,.$$

All the steady-state results are valid, as is, also for $M/G/\infty$ – **generally distributed service times**.
(Insensitivity of performance to the service-time distribution.)

Time-Varying Arrivals: The observation $R = \mathrm{E}[L]$ paves the way to the definition of **offered-load** in a time-varying environment: In $M_t/G/\infty$, with a time-inhomogeneous Poisson arrival process at rate $\{\lambda(t), t \geq 0\}$), the average number-in-system at time t is:

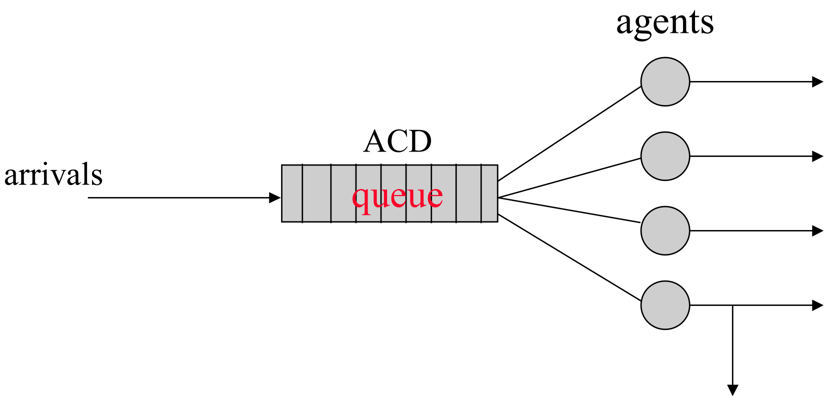
$$R(t) = E\int_{t-S}^{t} \lambda(u)\,du = E\lambda(t - S_e)\;, \quad t \geq 0\;.$$

Here S denotes a service-time and S_e a *residual service-time* (as in biased-sampling).
In the special case of S exponential, S_e has the same exponential distribution as S, due to the memoryless property.

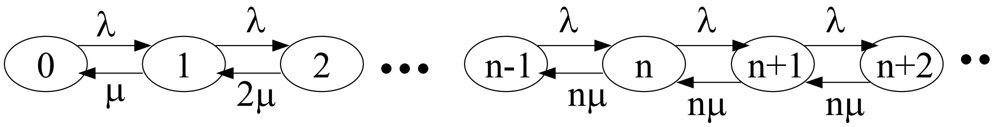
The Erlang-C (M/M/n) Queue

- Poisson arrivals, rate λ;
- n exponential servers, each at rate μ.

Widely used in call centers - the workhorse of WFM.



Transition-rate diagram:



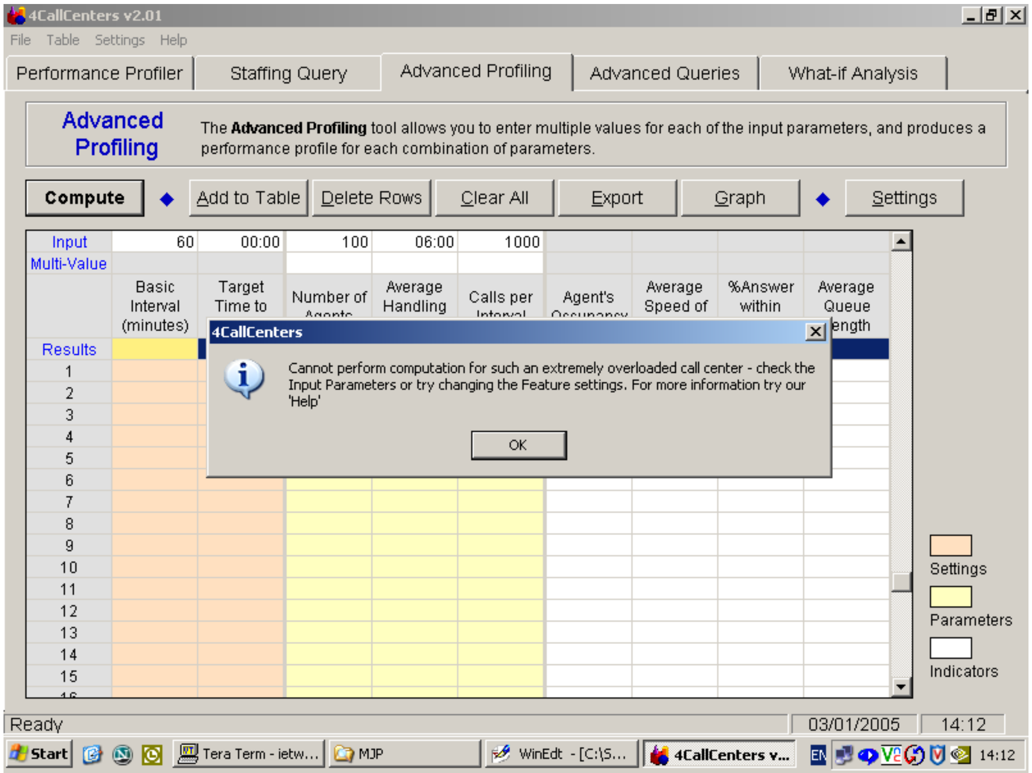
$$\begin{aligned} \lambda_j &\equiv \lambda, \qquad\quad j \geq 0, \\ \mu_j &= (j \wedge n)\mu, \;\; j \geq 1. \end{aligned}$$

Agents' utilization

$$\rho = \frac{\lambda}{n\mu}\,.$$

Assume $\rho < 1$ ($R < n$) to ensure stability (in analogy to M/M/1).

Example of Instability, via 4CallCenters



The **Advanced Profiling** tool allows you to enter multiple values for each of the input parameters, and produces a performance profile for each combination of parameters.

	Basic Interval (minutes)	Target Time to	Number of	Average Handling	Calls per	Agent's	Average Speed of	%Answer within	Average Queue ength
Input	60	00:00	100	06:00	1000				
Multi-Value									
Results									
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									

4CallCenters

Cannot perform computation for such an extremely overloaded call center - check the Input Parameters or try changing the Feature settings. For more information try our 'Help'

OK

Ready 03/01/2005 14:12

Steady-state distribution:

$$
\begin{aligned}
\pi_i &= \frac{R^i}{i!}\,\pi_0, && i \le n,\\
&= \frac{n^n \rho^i}{n!}\,\pi_0, && i \ge n,\\
\pi_0 &= \left[\sum_{j=0}^{n-1} \frac{R^j}{j!} + \frac{R^n}{n!(1-\rho)}\right]^{-1},
\end{aligned}
$$

where $R = \lambda/\mu$ is the offered load.

M/M/n Queue: Properties

Erlang-C Formula (1917) for the Delay probability:

$$\mathrm{P}\{W_q > 0\} \triangleq E_{2,n} = \sum_{i \geq n} \pi_i = \frac{R^n}{n!} \frac{1}{1-\rho} \cdot \pi_0 \,.$$

Erlang-C computation: via recursion, see Erlang-B below.

Erlang-C approximations: important later in course.

Number-in-queue:

$$\mathrm{P}\{L_q = i\} = E_{2,n} \cdot (1-\rho)\rho^i \,, \qquad i > 0,$$

or

$$L_q = \begin{cases} 0 & \text{wp } 1 - E_{2,n} \\ \mathrm{Geom}_{\geq 0}(1-\rho) & \text{wp } E_{2,n} \end{cases}$$

Waiting time distribution:

$$\frac{W_q}{1/\mu} = \begin{cases} 0 & \text{wp } 1 - E_{2,n} \\ Exp\left(\text{mean} = \frac{1}{n} \cdot \frac{1}{1-\rho}\right) & \text{wp } E_{2,n} \end{cases}$$

Compare with M/M/1!

Departure process: Poisson(λ) in steady-state.
Proof via reversibility, as with M/M/1.

M/M/n Queue: Waiting-Time Distribution

In analogy to M/M/1,

$$\frac{1}{\mathrm{E}[S]} \cdot W_q | W_q > 0 \overset{d}{=} \exp(n(1-\rho)),$$

$$\mathrm{P}\left\{ \frac{1}{\mathrm{E}[S]} \cdot W_q > t | W_q > 0 \right\} = e^{-n(1-\rho)t}.$$

Formally:

$$\mathrm{P}\{W_q > t\} = \sum_{k=1}^{\infty} \mathrm{P}\{L_q = k-1\} \cdot \mathrm{P}\{E_k > t\}$$

(where $E_k \overset{d}{=} \mathrm{Erlang}(k, n\mu)$)

$$= E_{2,n} \cdot \sum_{k=1}^{\infty} \left[(1-\rho)\rho^{k-1} \cdot \int_t^{\infty} \frac{n\mu(n\mu x)^{k-1}}{(k-1)!} e^{-n\mu x} dx \right]$$

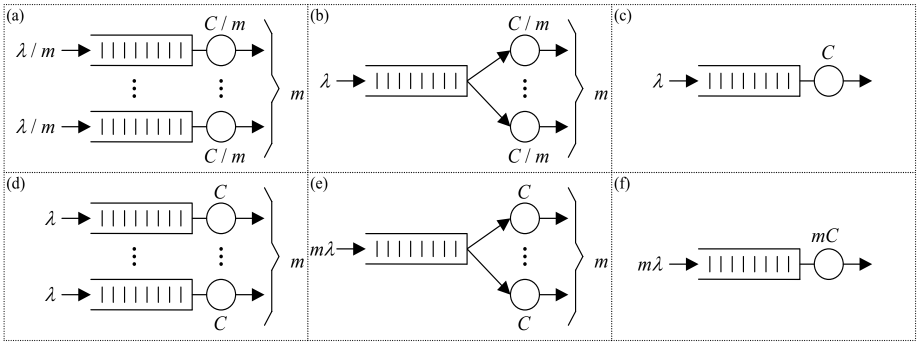
$$= E_{2,n} \cdot n\mu(1-\rho) \cdot \int_t^{\infty} \left(e^{-n\mu x} \sum_{k=1}^{\infty} \frac{(n\mu\rho x)^{k-1}}{(k-1)!} \right) dx$$

$$= E_{2,n} \cdot n\mu(1-\rho) \cdot \int_t^{\infty} e^{-n\mu(1-\rho)x} dx$$

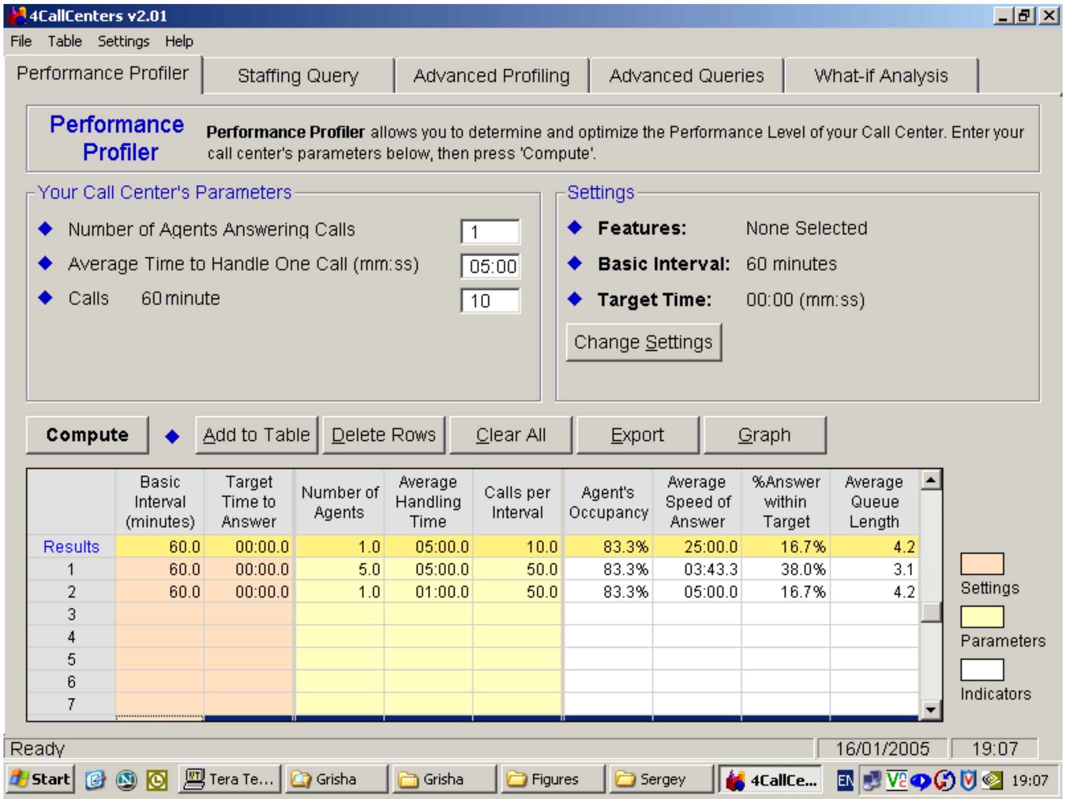
$$= E_{2,n} \cdot e^{-n\mu(1-\rho)t}.$$

Pooling: Economies of Scale

Example: Kleinrock, L. Vol.II, Chapter 5 (1976)



4CallCenters output



Performance Profiler allows you to determine and optimize the Performance Level of your Call Center. Enter your call center's parameters below, then press 'Compute'.

Your Call Center's Parameters

- Number of Agents Answering Calls: 1
- Average Time to Handle One Call (mm:ss): 05:00
- Calls 60 minute: 10

Settings

- **Features:** None Selected
- **Basic Interval:** 60 minutes
- **Target Time:** 00:00 (mm:ss)

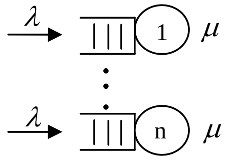
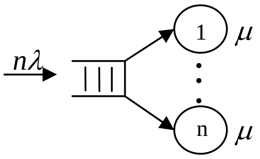
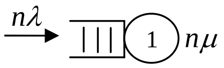
	Basic Interval (minutes)	Target Time to Answer	Number of Agents	Average Handling Time	Calls per Interval	Agent's Occupancy	Average Speed of Answer	%Answer within Target	Average Queue Length
Results	60.0	00:00.0	1.0	05:00.0	10.0	83.3%	25:00.0	16.7%	4.2
1	60.0	00:00.0	5.0	05:00.0	50.0	83.3%	03:43.3	38.0%	3.1
2	60.0	00:00.0	1.0	01:00.0	50.0	83.3%	05:00.0	16.7%	4.2
3									
4									
5									
6									
7									

Ready 16/01/2005 19:07

Pooling M/M/1 to M/M/n

Pooling Queues (one vs. many) and Servers (slow vs. fast)

Erlang-C

	1 $n \times M \mid M \mid 1$ $\lambda,\ \mu$		**2** $M \mid M \mid n$ $n\lambda,\ \mu$		**3** $M \mid M \mid 1$ $n\lambda,\ n\mu$
	Multiple queues Multiple Servers		Single queue Multiple Servers		Single queue Single server
					
Tradeoff	Separate vs. single queue		Slow vs. fast server		
Enabler	Process design		Technology		
Gain	Load balancing $\forall t$ (avoid a long queue & idle servers)		Maximal capacity $\forall t$ $\le n\mu$ vs. $n\mu$		
Utilization	$\frac{\lambda}{\mu}=\rho$		$\frac{(n\lambda)}{(n)\mu}=\rho$		$\frac{(n\lambda)}{(n\mu)}=\rho$
$P\{W_q>0\}$	ρ	②	$E_{2,n}(\rho)$	②	ρ
$E\left[W_q \mid W_q>0\right]$	$\frac{1}{\mu}\cdot\frac{1}{1-\rho}$	①	$\frac{1}{n\mu}\cdot\frac{1}{1-\rho}$	=	$\frac{1}{n\mu}\cdot\frac{1}{1-\rho}$
$E\left[W_q\right]=$ $E\left[W_q \mid W_q>0\right]\cdot P\{W_q>0\}$	$\frac{1}{\mu}\cdot\frac{\rho}{1-\rho}$	④	$\frac{1}{\mu}\cdot\frac{E_{2,n}}{n(1-\rho)}$	③	$\frac{1}{\mu}\cdot\frac{\rho}{n(1-\rho)}$
$E[S]$	$\frac{1}{\mu}$	=	$\frac{1}{\mu}$	>	$\frac{1}{n\mu}$
$E\left[W_{sys}\right]=$ $=E[W_q]+E[S]$	$\frac{1}{\mu}\cdot\frac{1}{1-\rho}$	⑤	$\frac{1}{\mu}\cdot\left[\frac{E_{2,n}}{n(1-\rho)}+1\right]$	⑥	$\frac{1}{n\mu}\cdot\frac{1}{1-\rho}$

<u>Basic Relations</u>:

$$1 \le \frac{1-E_{2,n}}{1-\rho} \le n \qquad\qquad \left(\begin{array}{c} \Rightarrow\; E_{2,n}(\rho) \le \rho \\ \text{Intuition ?} \end{array} \right)$$

<u>Proof</u>:

$$1-E_{2,n} = P(\text{at least one server idle}) =$$

$$= P\left(\bigcup_{i=1}^{n} \{\text{server } i \text{ idle}\} \right) \quad \begin{cases} \le \sum_{i=1}^{n} P\{\text{server } i \text{ idle}\} = n(1-\rho) \\ \ge P\{\text{server } i \text{ idle}\} = 1-\rho \end{cases}$$

$q.e.d.$

Corollary: $1-\rho \le 1-E_{2,n} \quad \Rightarrow \quad E_{2,n} \le \rho \quad \Rightarrow \qquad \textcircled{2} \; \textcircled{3}$

Corollary: $1-E_{2,n} \le n(1-\rho) \quad \Rightarrow \quad \dfrac{1}{n(1-\rho)} \le \dfrac{E_{2,n}}{n(1-\rho)} + 1 \quad \Rightarrow \qquad \textcircled{6}$

$\textcircled{2} \; \textcircled{3}$

$$\Updownarrow$$

$$EW_q(n, n\lambda, \mu) \;\le\; EW_q(1, n\lambda, n\mu)$$

Multiple-slow $\le$ Single-fast

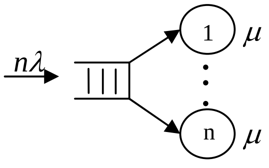
$\textcircled{6}$

$$\Updownarrow$$

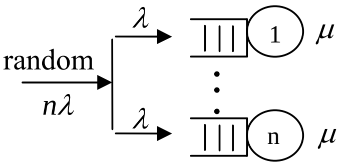
$$EW_{sys}(n, n\lambda, \mu) \;\ge\; EW_{sys}(1, n\lambda, n\mu)$$

Multiple-slow $\ge$ Single-fast

<u>Intuition</u>: $E_{2,n}(\rho) \le \rho$



vs.



$$E_{2,n}(\rho) \qquad\qquad \le \qquad\qquad \rho$$

since have an earlier commitment

Pooling M/M/1 to M/M/n: Conclusions

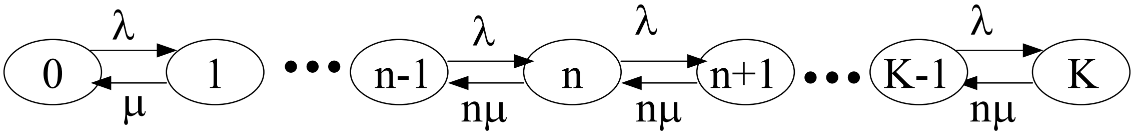
$1 \longrightarrow 2$: Pooling yields $\mathrm{E}[W_q]$ decrease by more than factor n;

$1 \longrightarrow 3$: Fast server yields $\mathrm{E}[W]$ and $\mathrm{E}[W_q]$ decrease by factor n;

$2 \longrightarrow 3$: Fast server better for $\mathrm{E}[W]$;
Pooling better for $\mathrm{E}[W_q]$.

M/M/n/K Queue

- Poisson arrivals, rate λ;
- n exponential servers, rate μ;
- K trunks ($K \geq n$);
- If all trunks busy, arriving customer blocked (busy signal).

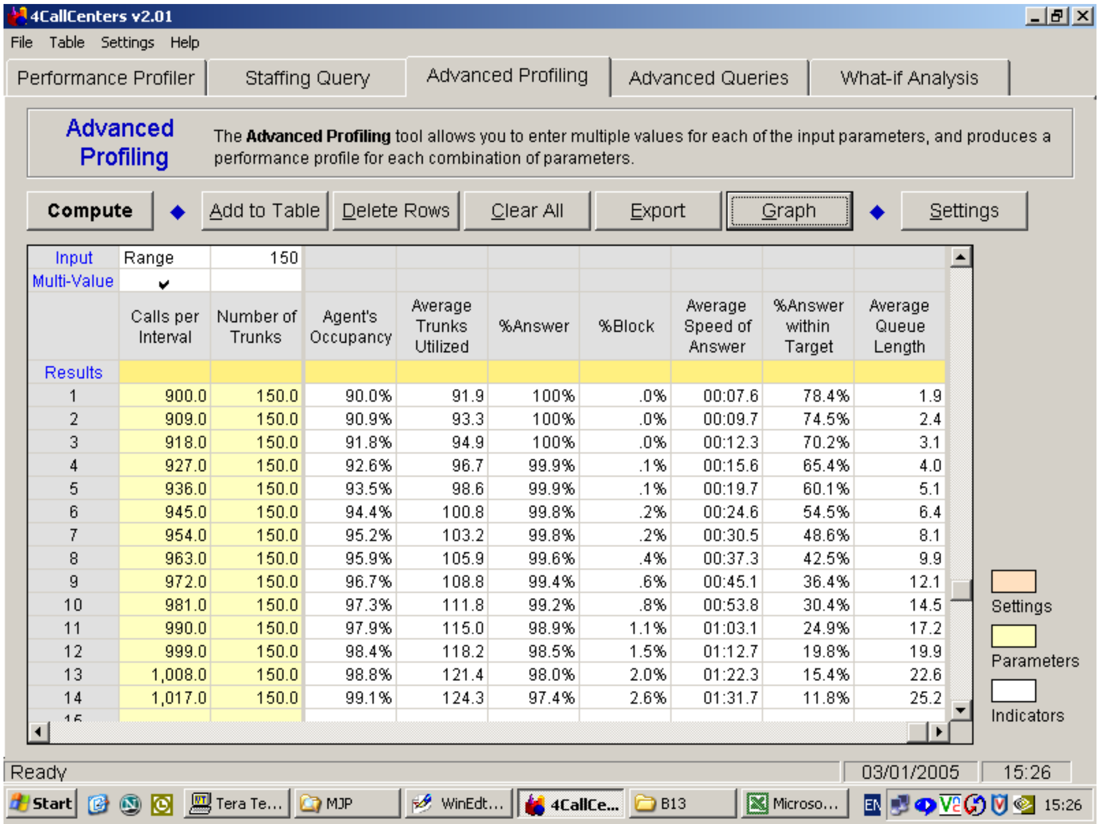


$$\begin{aligned} \lambda_j &= \lambda, & 0 \leq j \leq K-1, \\ \mu_j &= (j \wedge n)\mu, & 1 \leq j \leq K. \end{aligned}$$

Formulae straightforward but cumbersome
(simply truncate M/M/n).

Always reaches steady state.

M/M/n/K Queue: 4CallCenters



The **Advanced Profiling** tool allows you to enter multiple values for each of the input parameters, and produces a performance profile for each combination of parameters.

Input	Range	150							
Multi-Value	✓								
	Calls per Interval	Number of Trunks	Agent's Occupancy	Average Trunks Utilized	%Answer	%Block	Average Speed of Answer	%Answer within Target	Average Queue Length
Results									
1	900.0	150.0	90.0%	91.9	100%	.0%	00:07.6	78.4%	1.9
2	909.0	150.0	90.9%	93.3	100%	.0%	00:09.7	74.5%	2.4
3	918.0	150.0	91.8%	94.9	100%	.0%	00:12.3	70.2%	3.1
4	927.0	150.0	92.6%	96.7	99.9%	.1%	00:15.6	65.4%	4.0
5	936.0	150.0	93.5%	98.6	99.9%	.1%	00:19.7	60.1%	5.1
6	945.0	150.0	94.4%	100.8	99.8%	.2%	00:24.6	54.5%	6.4
7	954.0	150.0	95.2%	103.2	99.8%	.2%	00:30.5	48.6%	8.1
8	963.0	150.0	95.9%	105.9	99.6%	.4%	00:37.3	42.5%	9.9
9	972.0	150.0	96.7%	108.8	99.4%	.6%	00:45.1	36.4%	12.1
10	981.0	150.0	97.3%	111.8	99.2%	.8%	00:53.8	30.4%	14.5
11	990.0	150.0	97.9%	115.0	98.9%	1.1%	01:03.1	24.9%	17.2
12	999.0	150.0	98.4%	118.2	98.5%	1.5%	01:12.7	19.8%	19.9
13	1,008.0	150.0	98.8%	121.4	98.0%	2.0%	01:22.3	15.4%	22.6
14	1,017.0	150.0	99.1%	124.3	97.4%	2.6%	01:31.7	11.8%	25.2
15									

Use `Change Settings` ⟹ `Features` ⟹ `Trunks`.

Note new indicators:

`Average Trunks Utilized` and `%Blocked`.

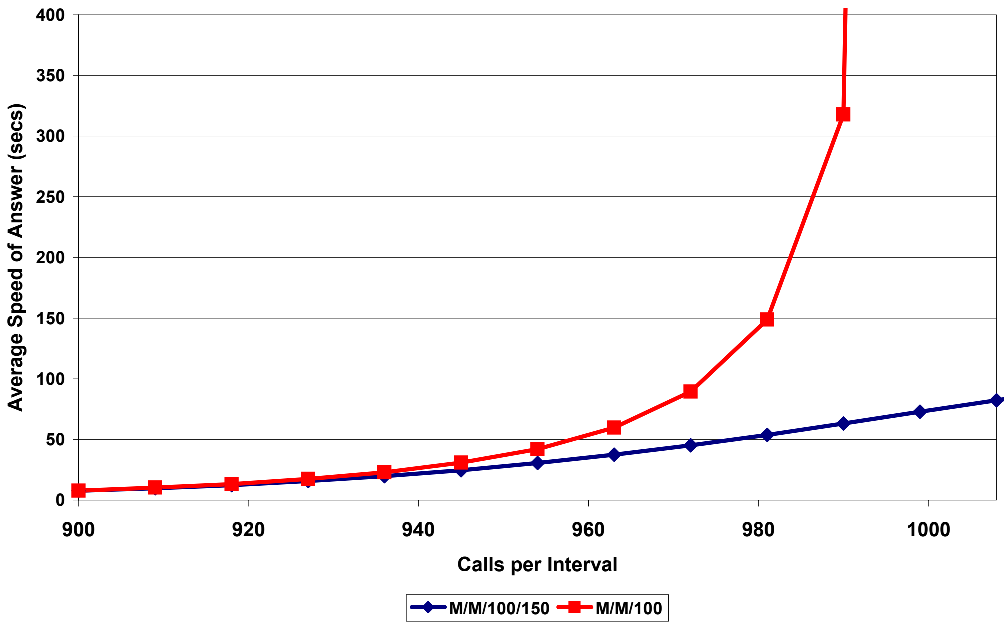
4CallCenters: Advanced Profiling

Arrival rate varied from 900 to 1017 per hour, in step 9.

Excel interface: graphs and spreadsheets.

M/M/n/K vs. Erlang-C

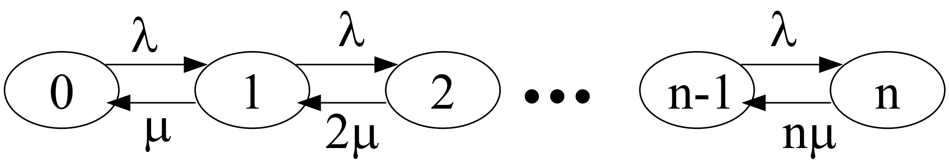
Average service time = 6 min, 100 agents, 150 trunks.



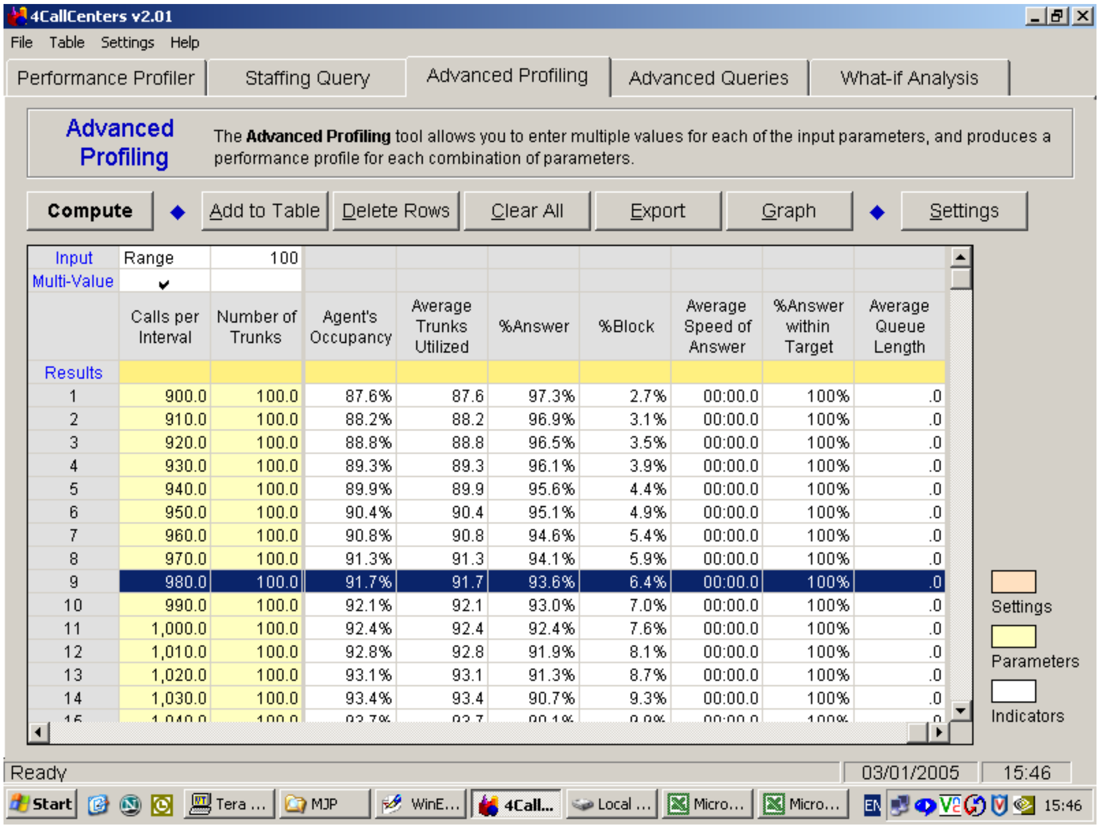
Similar performance for light loads.

Erlang-C "explodes" as $\rho = \dfrac{\lambda}{n\mu} \uparrow 1$.

The Erlang-B (M/M/n/n) Queue



$$\lambda_i \equiv \lambda, \qquad 0 \le i \le n-1\,,$$
$$\mu_i = i \cdot \mu, \qquad 1 \le i \le n\,.$$



Advanced Profiling

The **Advanced Profiling** tool allows you to enter multiple values for each of the input parameters, and produces a performance profile for each combination of parameters.

Input	Range	100							
Multi-Value	✓								
	Calls per Interval	Number of Trunks	Agent's Occupancy	Average Trunks Utilized	%Answer	%Block	Average Speed of Answer	%Answer within Target	Average Queue Length
Results									
1	900.0	100.0	87.6%	87.6	97.3%	2.7%	00:00.0	100%	.0
2	910.0	100.0	88.2%	88.2	96.9%	3.1%	00:00.0	100%	.0
3	920.0	100.0	88.8%	88.8	96.5%	3.5%	00:00.0	100%	.0
4	930.0	100.0	89.3%	89.3	96.1%	3.9%	00:00.0	100%	.0
5	940.0	100.0	89.9%	89.9	95.6%	4.4%	00:00.0	100%	.0
6	950.0	100.0	90.4%	90.4	95.1%	4.9%	00:00.0	100%	.0
7	960.0	100.0	90.8%	90.8	94.6%	5.4%	00:00.0	100%	.0
8	970.0	100.0	91.3%	91.3	94.1%	5.9%	00:00.0	100%	.0
9	980.0	100.0	91.7%	91.7	93.6%	6.4%	00:00.0	100%	.0
10	990.0	100.0	92.1%	92.1	93.0%	7.0%	00:00.0	100%	.0
11	1,000.0	100.0	92.4%	92.4	92.4%	7.6%	00:00.0	100%	.0
12	1,010.0	100.0	92.8%	92.8	91.9%	8.1%	00:00.0	100%	.0
13	1,020.0	100.0	93.1%	93.1	91.3%	8.7%	00:00.0	100%	.0
14	1,030.0	100.0	93.4%	93.4	90.7%	9.3%	00:00.0	100%	.0
15	1,040.0	100.0	93.7%	93.7	90.1%	9.9%	00:00.0	100%	.0

🟧	Settings
🟨	Parameters
□	Indicators

Ready 03/01/2005 15:46

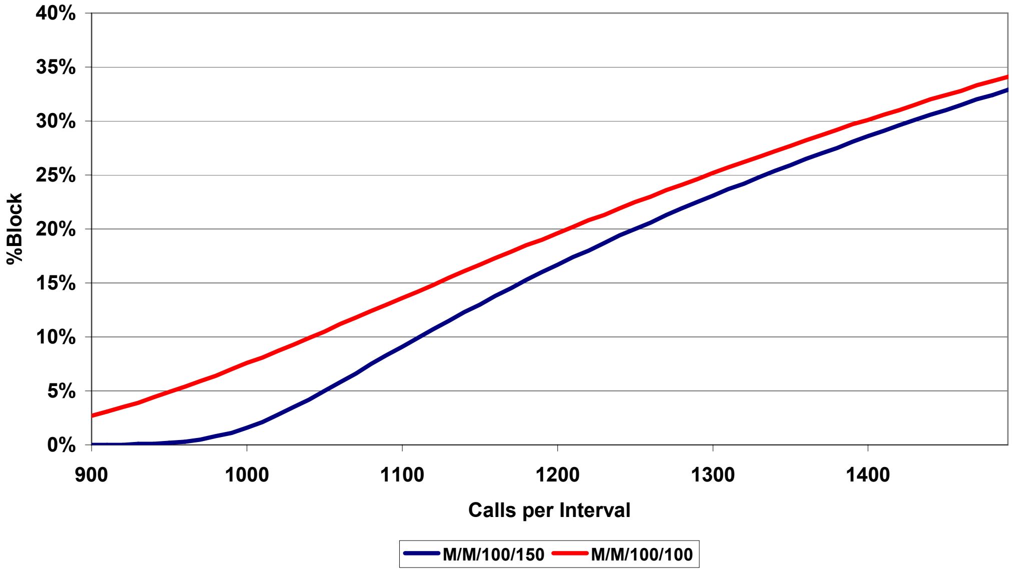
Steady-State:

$$\pi_i = \frac{R^i}{i!} \Big/ \sum_{j=0}^{n} \frac{R^j}{j!}, \qquad 0 \le i \le n.$$

Note: The above applies to M/G/n/n - again, insensitivity.

M/M/n/K vs. Erlang-B

Average service time = 6 min, 100 agents



Moderate load: additional trunks prevent blocking.

Heavy load: % blocking $\approx 1 - 1/\rho$ (*"fluid limit"*).

Erlang-B Formula (1917)

Loss probability:

$$E_{1,n} \triangleq \pi_n = \frac{R^n}{n!} \Big/ \sum_{j=0}^{n} \frac{R^j}{j!} \ .$$

Follows from PASTA.
Recall: Erlang-B valid for M/G/n/n (General services.)

$\lambda\pi_n$ – rate of lost customers,
$\lambda(1-\pi_n)$ – effective throughput.

Erlang-B Computation via recursion:

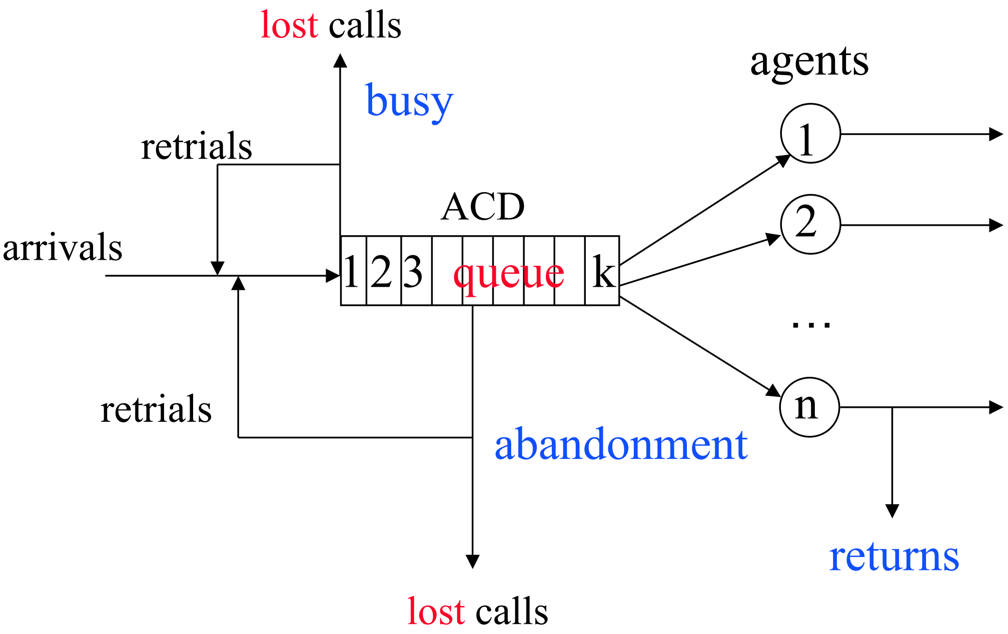
$$E_{1,n} = \frac{RE_{1,n-1}}{n + RE_{1,n-1}} = \frac{\rho E_{1,n-1}}{1+\rho E_{1,n-1}} \qquad E_{1,0} = 1 \, .$$

Erlang-B Computation:

$$E_{1,n} = \frac{(n-R)E_{2,n}}{n - RE_{2,n}} \ ; \qquad E_{2,n} = \frac{E_{1,n}}{(1-\rho)+\rho E_{1,n}} \ ;$$

$E_{2,n} > E_{1,n}$, as expected: why?

Telephone Call Center



Two customer-centric (subjective) operational measures of performance:

- **Abandonment** due to (im)patience, or need;
- **Retrials/Redials**, which can often be absorbed into the Poisson arrival process).

How to model Abandonment?
The Palm / Erlang-A model.