

Performance Evaluation with Java Modelling Tools

Giuliano Casale

Department of Computing
Imperial College London, UK

Joint work with:

Giuseppe Serazzi (Politecnico di Milano, Italy)

Lulai Zhu (Imperial College London, UK)

Outline

- Introduction
- Activity 1: getting started
- Activity 2: load balancing
- Activity 3: parameter sweeping
- Activity 4: capacity constraints
- Activity 5: workflows & fork-join

- Please download the latest JMT (v1.0.2) here:
 - <http://jmt.sf.net/Download.html>

Introduction

Java Modelling Tools


- Simulation and analysis of queueing networks.
- Project started in 2002 at Politecnico di Milano, since 2010 co-developed at Imperial.
- JMT is open source: GPL v2,
 - Medium-size project: ~1,000 classes
 - JAR, source code and maven build files (*pom.xml*)
<http://jmt.sourceforge.net/Download.html>
- Good diffusion (59k downloads, mostly from the US)
- Community interaction mainly through
 - Bug reports
 - Feature requests
 - Templates

Supported models

- Queueing Systems
- Queueing Networks (QN)
 - Product-form
 - Extended (fork/join, blocking, priorities, ...)
- Petri Nets (PN)
 - Stochastic Petri Nets (SPN)
 - Generalized Stochastic Petri Nets (GSPN)
 - Coloured Petri Nets (CPN)
- Queueing Petri Nets (QPNs)

Who uses JMT?

- JMT is for PE practice, teaching, and research
- Several university courses worldwide (tell us!)
- Supporting materials available on website


DEIB - Politecnico di Milano - Italy
Project Coordinator: G.Serazzi

**Imperial College
London**

Main Menu
[Introduction](#)
[Download JMT](#)
[Requirements](#)
[JSIMgraph](#)
[JSIMwiz](#)
[JMVA](#)
[JABA](#)
[JWAT](#)
[JMCH](#)
[Documentation](#)
[License](#)

Links
[Discussion Forums](#)
[Report a Bug](#)
[Request a New Feature](#)
[Sourceforge Page](#)

Java Modelling Tools

Documentation

Manuals & Books

- Java Modelling Tools - *Users manual*
v.0.9.1, 191 pp., Oct. 22th 2013 ([Manual](#))
- G.Serazzi Ed.
Performance Evaluation Modelling with JMT: learning by examples
Politecnico di Milano - DEI, TR 2008.09, 366pp., June 2008 ([Book](#))

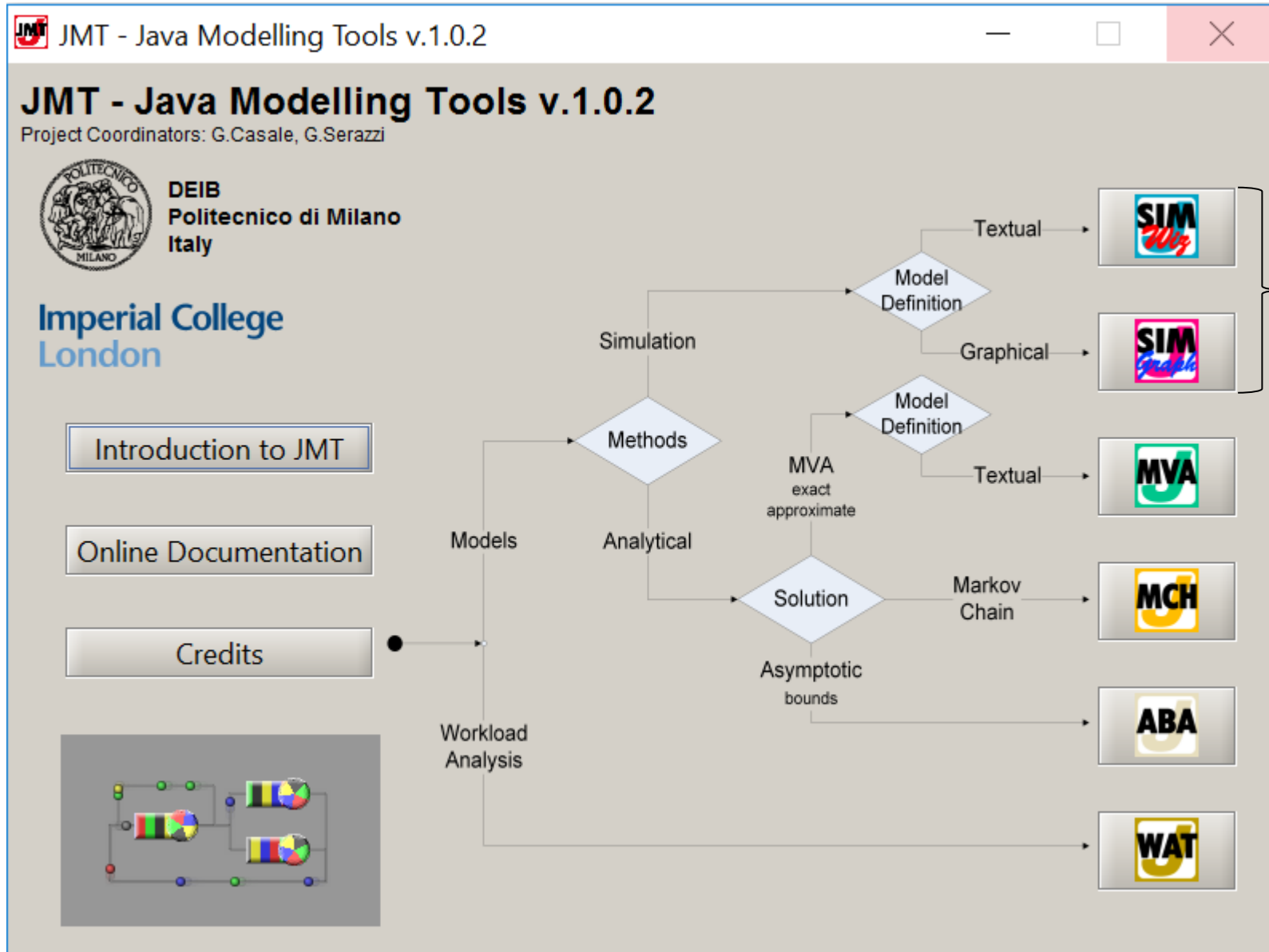
Tutorials

- G.Casale, G.Serazzi.
Quantitative System Evaluation with Java Modelling Tools.
2nd ACM/SPEC International Conference on Performance Engineering (ICPE), March 2011, Karlsruhe, Germany

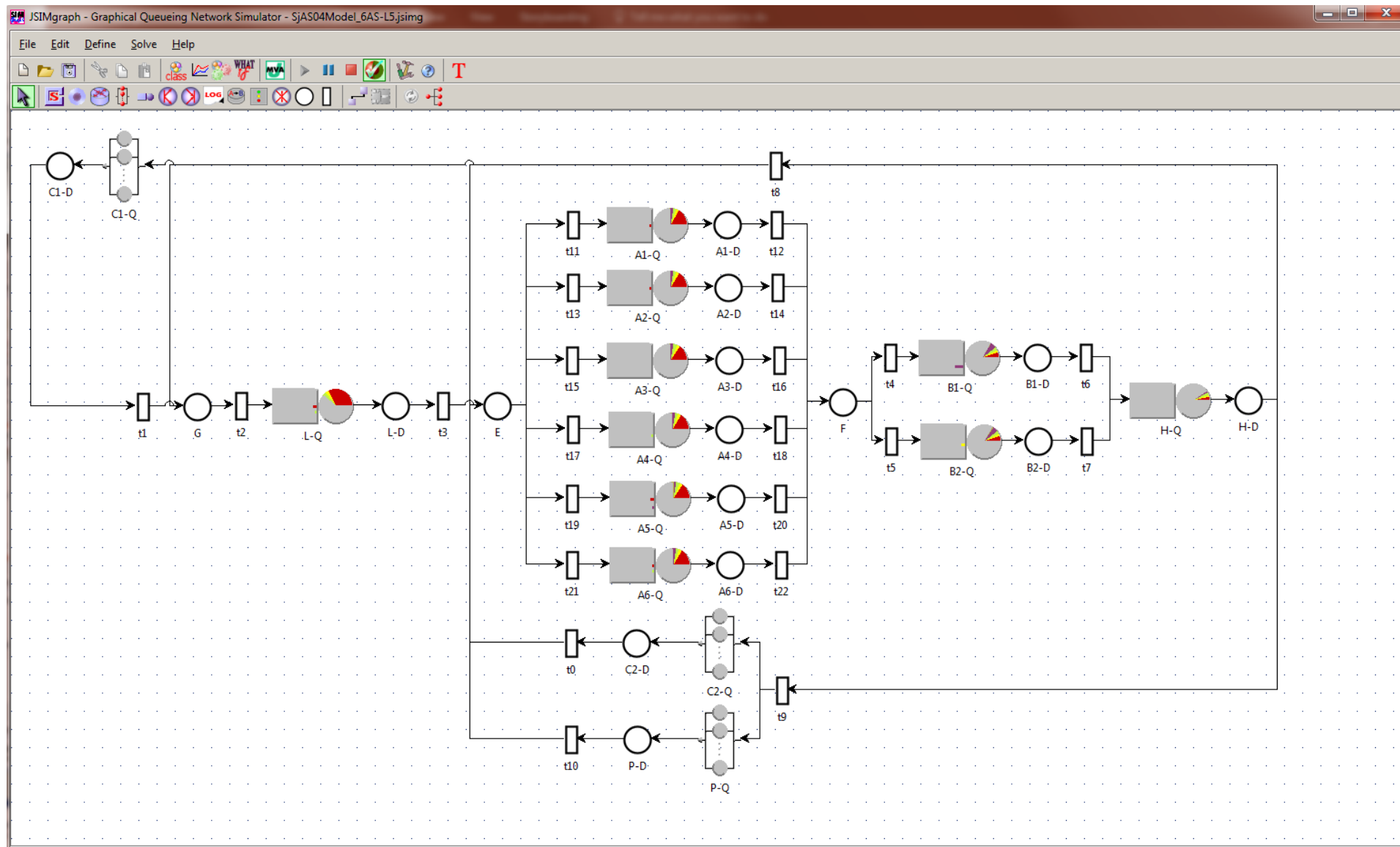
Papers

- M.Bertoli, G.Casale, G.Serazzi.
JMT: performance engineering tools for system modeling.
ACM SIGMETRICS Performance Evaluation Review, Volume 36 Issue 4, New York, US, March 2009, 10-15, AC
- M.Bertoli, G.Casale, G.Serazzi.
User-Friendly Approach to Capacity Planning Studies with Java Modelling Tools.
Int.I ICST Conf. on Simulation Tools and Techniques, SIMUTools 2009, Rome, Italy, 2009, ACM press. ([Article](#))

JMT Start Screen



JSIMgraph: QN & PN simulation



9

JSIMwiz: wizard-based user interface

The screenshot displays the JSIMwiz Queue Network Models Simulator interface. The main window is titled "JSIMwiz - Queue Network Models Simulator - xml_code_1.jsimw". The menu bar includes File, Action, Simulation, Define, and Help. The toolbar contains icons for file operations and simulation control. The main menu bar includes Classes, Stations, Connections, Station Parameters, Performance Indices, Reference Stations, Finite Capacity Regions, Simulation, and What-if analysis.

The **Station Parameters** dialog is open, showing the **Station1 Parameters Definition** tab. The **Capacity** section has the **infinite** radio button selected. The **Queue Policy** section shows the **Station queue policy** set to **Non-preemptive Scheduling**. A table lists the queue policy for **Class1** as **FCFS** with a **Drop Rule** of **Infinite Capacity**.

The **Simulation Results - xml_code_1.jsimw** window is also open, showing the **Number of Customers** results. The **Average value** is 8.831. The **Simulation Complete (Time Elapsed: 9.0s)** status bar is visible at the bottom.

Station Parameters
For each station in the list, define the requested parameters

Station1 Parameters Definition

Capacity

- ☒ infinite
- ☐ finite

max no. customers (queue+service)

Queue Policy

Station queue policy: Non-preemptive Scheduling

Class	Queue Policy	Drop Rule
Class1	FCFS	Infinite Capacity

Simulation Results - xml_code_1.jsimw

Number of Customers
Average number of customers for each chosen class at each chosen station.

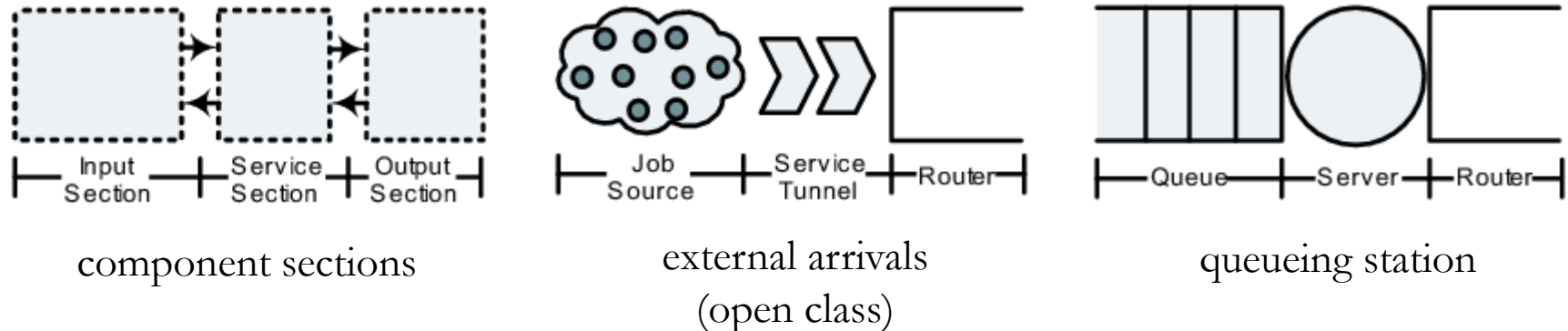
Station Name: Station1 Class Name: Class1
Conf.Int/Max Rel.Err: 0.99 / 0.03 Analyzed samples: 5898240
Min: 8.573 Max: 9.090
Average value: 8.831

Click on green bars to see the simulation time, the sample average (blue), and the sample values (green).

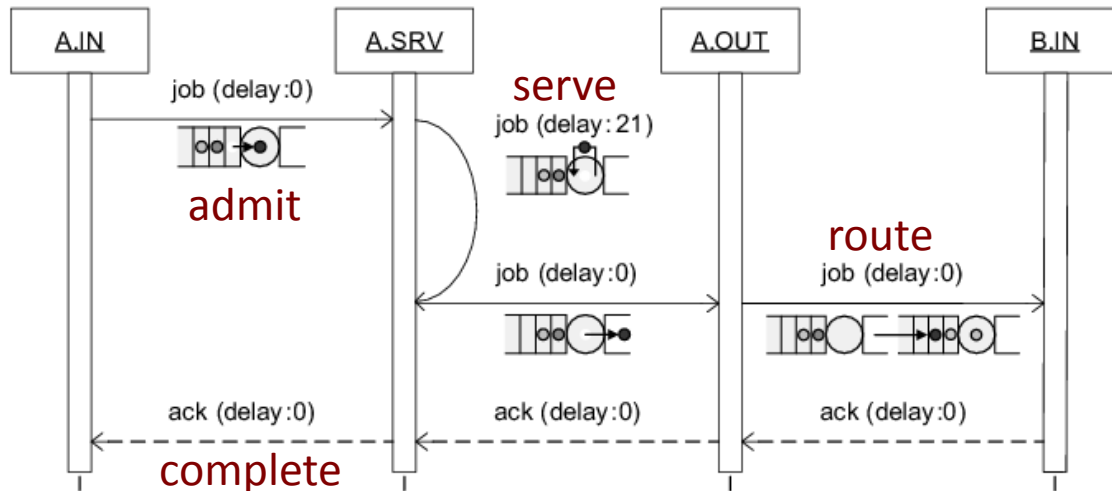
Simulation Complete (Time Elapsed: 9.0s)

JSIMEngine: discrete-event simulator

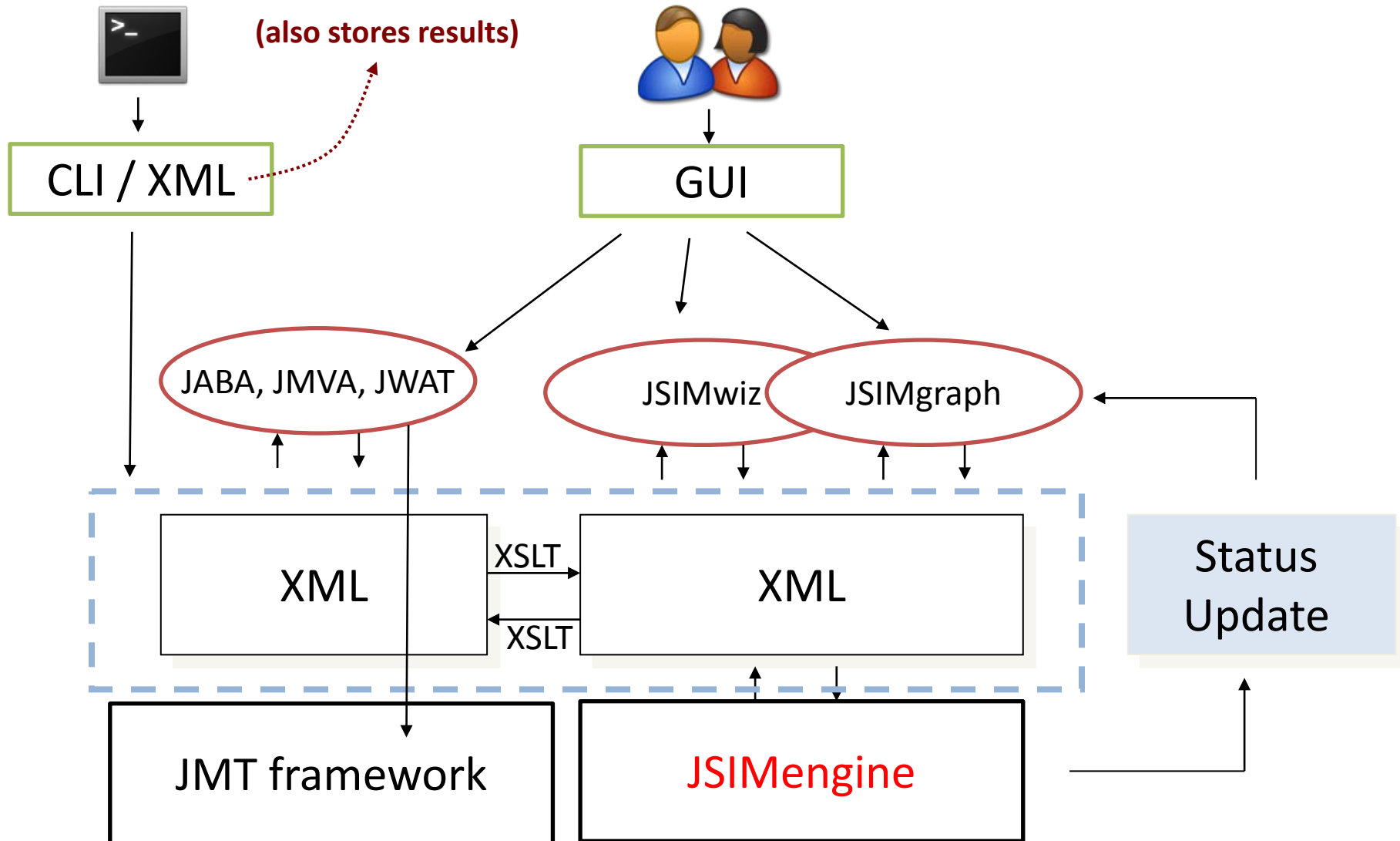
- Simulation components defined by 3 sections



- Discrete-event simulation of section messaging



JSIMEngine: JMT architecture



JMVA: analytical solver

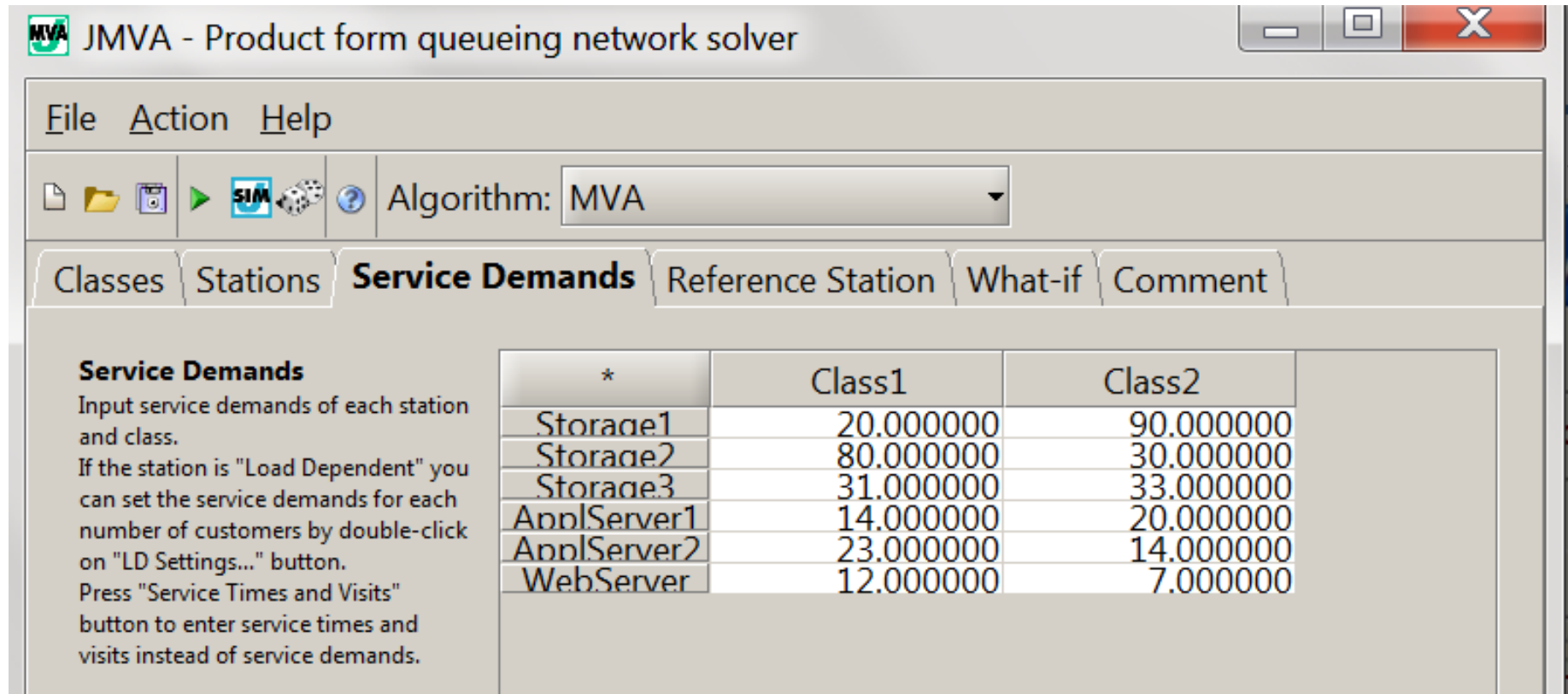
- Analysis of product-form queueing networks
- Several exact and approximate algorithms
 - *Exact MVA*
 - Reiser & Lavenberg $O(N^R)$
 - Load-dependent $O(N^{2R})$
 - *Approximate MVA* $O(1)$
 - Chow
 - Bard-Schweitzer
 - AQL
 - Linearizer
 - De Souza-Muntz
 - *Normalizing constant*
 - RECAL $O(N^M)$
 - CoMoM $O(N \log N)$

N : jobs

M : stations

R : classes

JMVA: model parameterization



JMVA - Product form queueing network solver

File Action Help

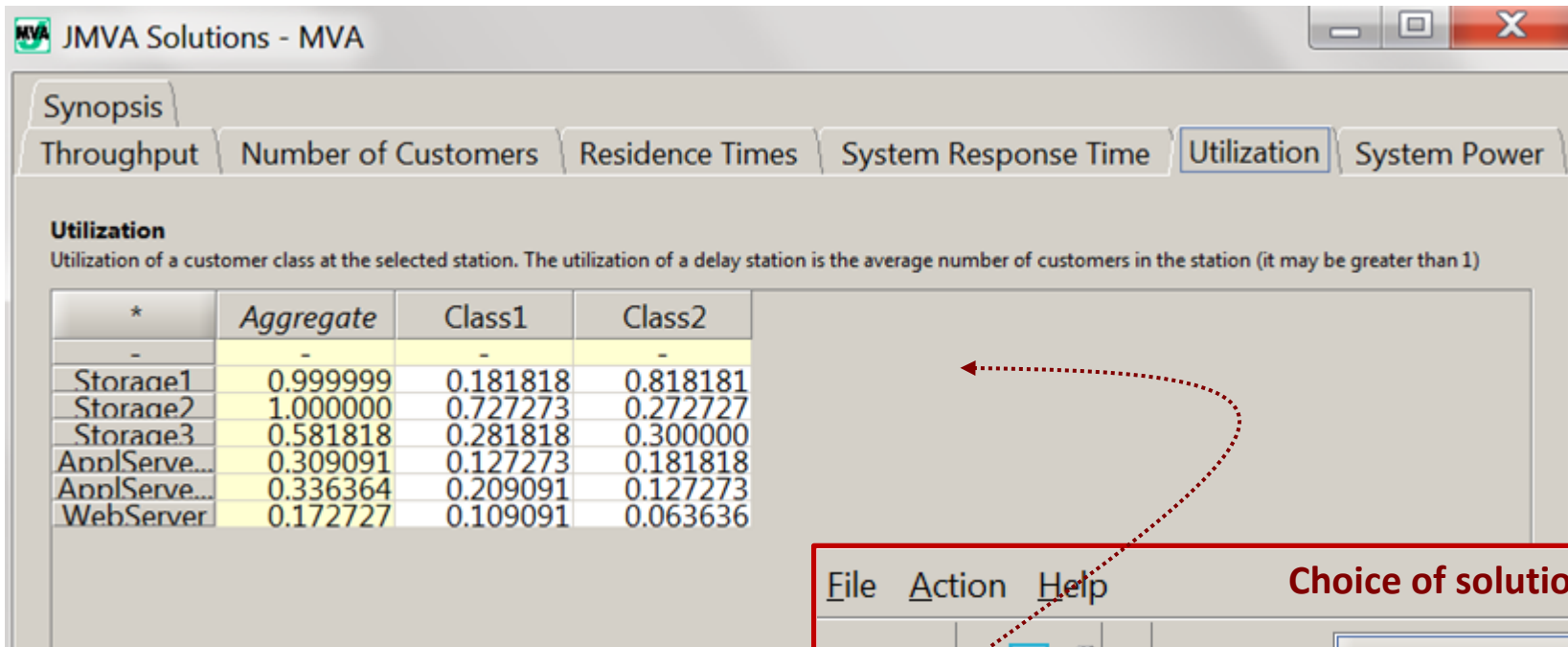
Algorithm: MVA

Classes Stations **Service Demands** Reference Station What-if Comment

Service Demands
Input service demands of each station and class.
If the station is "Load Dependent" you can set the service demands for each number of customers by double-click on "LD Settings..." button.
Press "Service Times and Visits" button to enter service times and visits instead of service demands.

*	Class1	Class2
Storage1	20.000000	90.000000
Storage2	80.000000	30.000000
Storage3	31.000000	33.000000
App1Server1	14.000000	20.000000
App1Server2	23.000000	14.000000
WebServer	12.000000	7.000000

JMVA: solutions



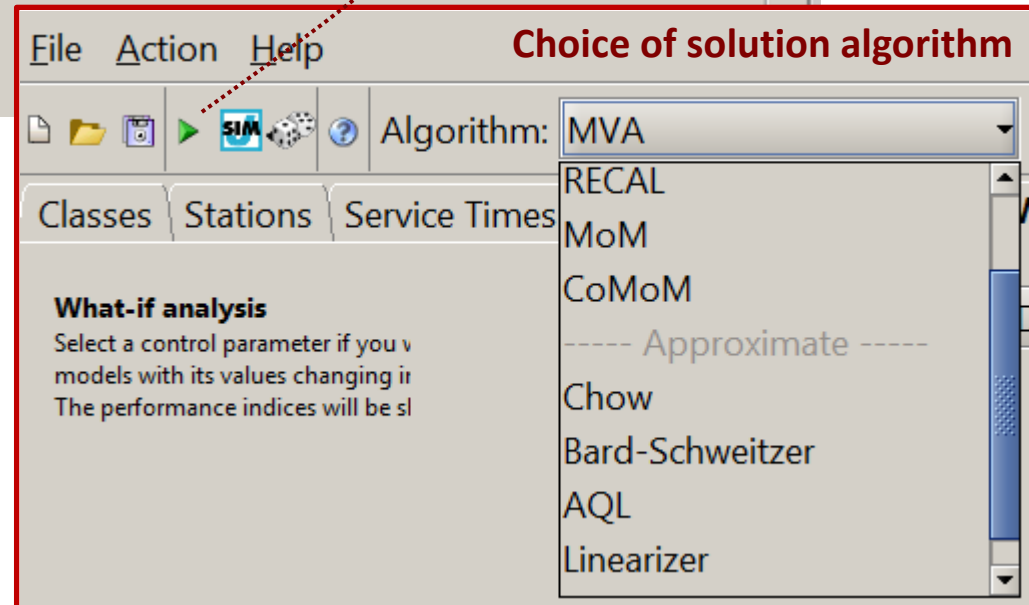
JMVA Solutions - MVA

Synopsis | Throughput | Number of Customers | Residence Times | System Response Time | **Utilization** | System Power

Utilization
Utilization of a customer class at the selected station. The utilization of a delay station is the average number of customers in the station (it may be greater than 1)

*	Aggregate	Class1	Class2
-	-	-	-
Storage1	0.999999	0.181818	0.818181
Storage2	1.000000	0.727273	0.272727
Storage3	0.581818	0.281818	0.300000
App1Serve...	0.309091	0.127273	0.181818
App1Serve...	0.336364	0.209091	0.127273
WebServer	0.172727	0.109091	0.063636

A red dotted arrow points from the 'Utilization' tab to the 'Choice of solution algorithm' dialog box.



Choice of solution algorithm

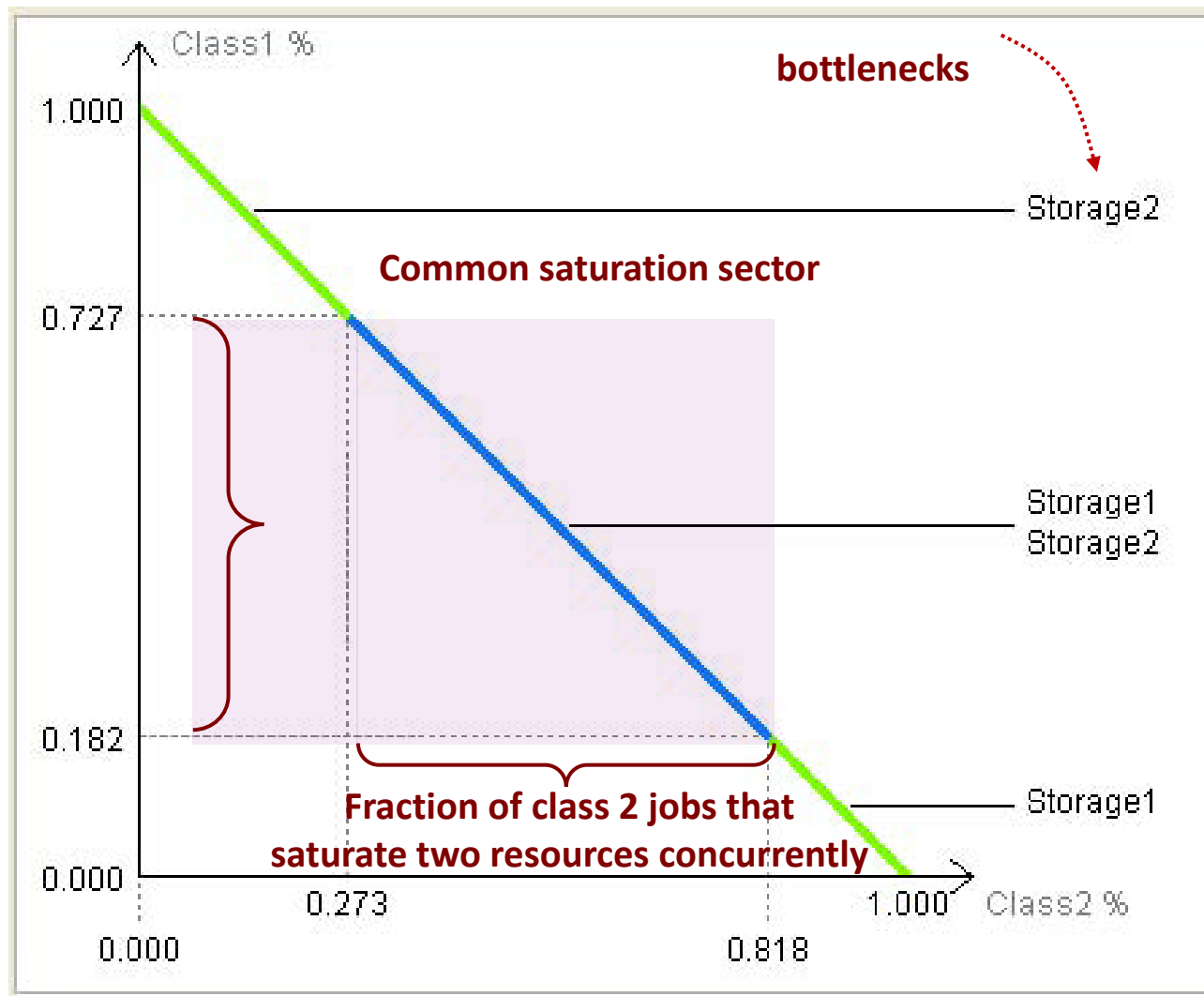
File Action Help

Algorithm: MVA

- RECAL
- MoM
- CoMoM
- Approximate -----
- Chow
- Bard-Schweitzer
- AQL
- Linearizer

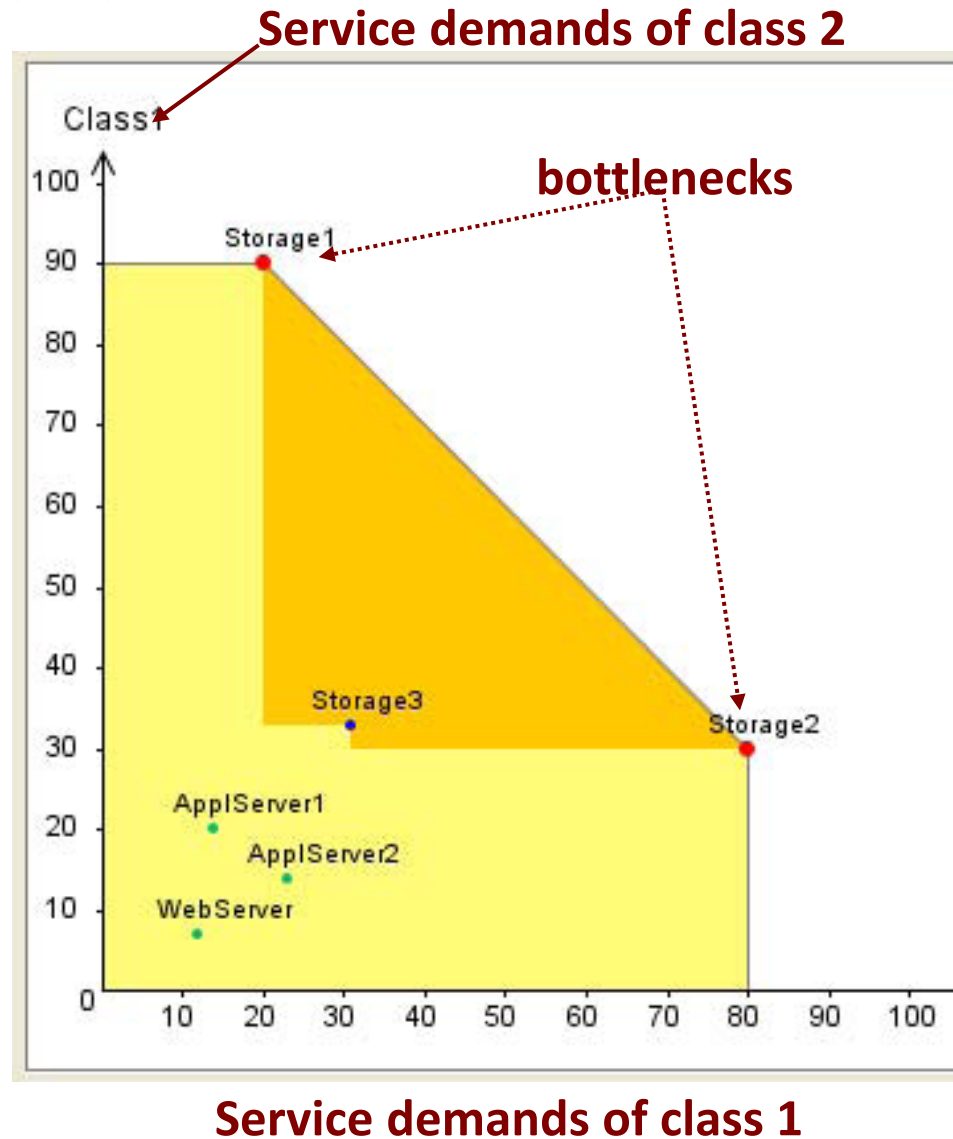
What-if analysis
Select a control parameter if you v
models with its values changing in
The performance indices will be sl

JABA: bottleneck identification

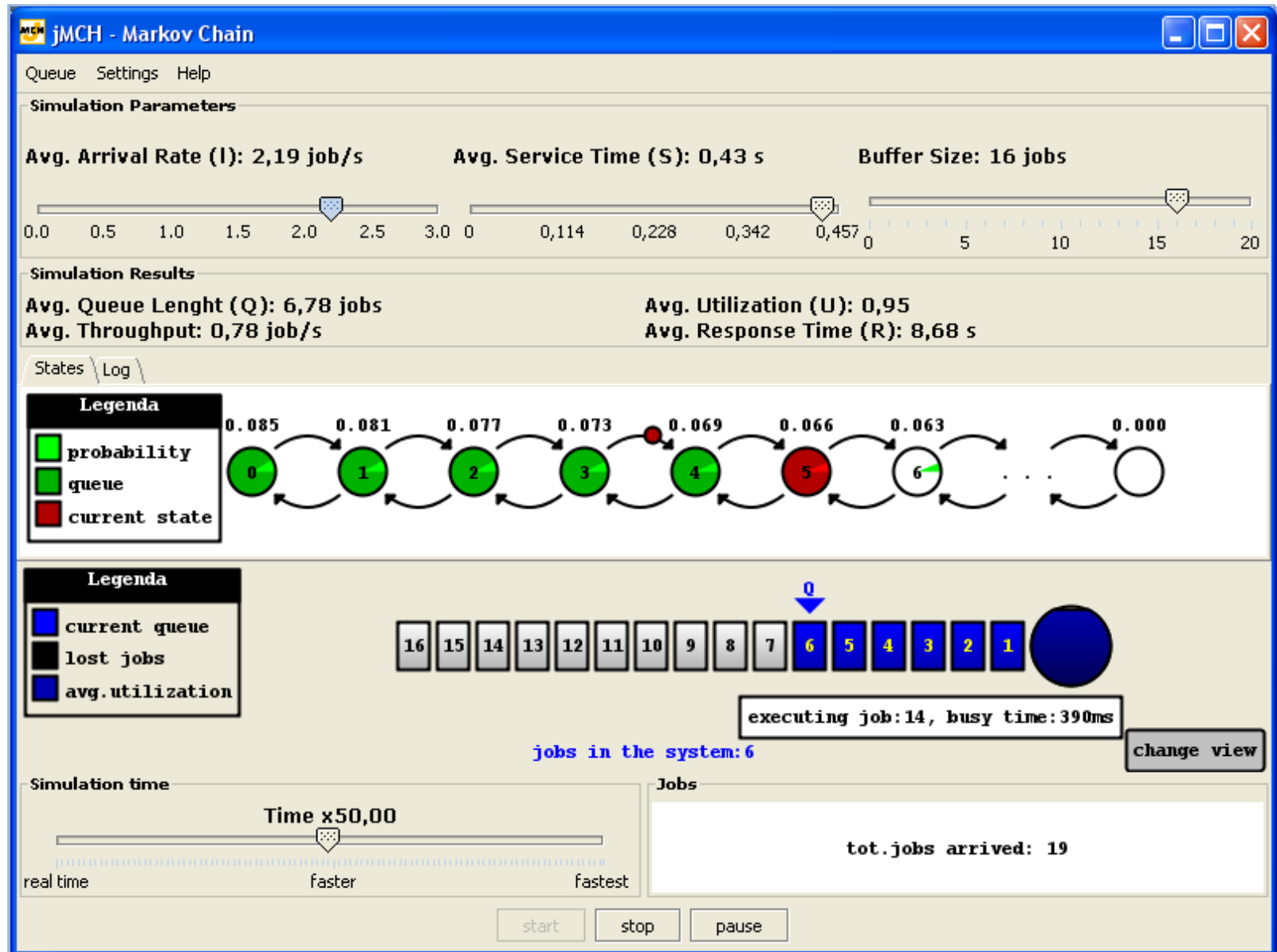


JABA: bottleneck identification

- 3-class model



JMCH: Markov chain animation



JWAT: workload characterization

Column-Oriented Log File

File Action Help

Main panel Input Statistics Clustering Clustering Information

Inputs
Define a new input format or open a saved format

Number: 3

Specify Format

Select	Name	Type	Comment	Sep.	Perl5 Reg. Exp.	Def.	Rep.
<input checked="" type="checkbox"/>	Variable 0	Num...	Response Flight Reserv.		((+-)?)\d+([.]\d...		✗
<input checked="" type="checkbox"/>	Variable 1	Num...	Response Travel Agency 1		((+-)?)\d+([.]\d...		✗
<input checked="" type="checkbox"/>	Variable 2	Num...	Response Travel Agency 2		((+-)?)\d+([.]\d...		✗

Loading Complete

? # observations processed: 15240
correct observations 15240
To see errors press Show Log

Continue Show Log Cancel

Input file
Load file
D:\ws-measurem.txt

File information
Number of observations: 15240
File name: ws-measurem.txt

File format
Load saved format Save format
D:\ws-measurem.jwatformat

Workload sampling method
Choose sampling method
Complete file

Loading size options

Data Format Templates

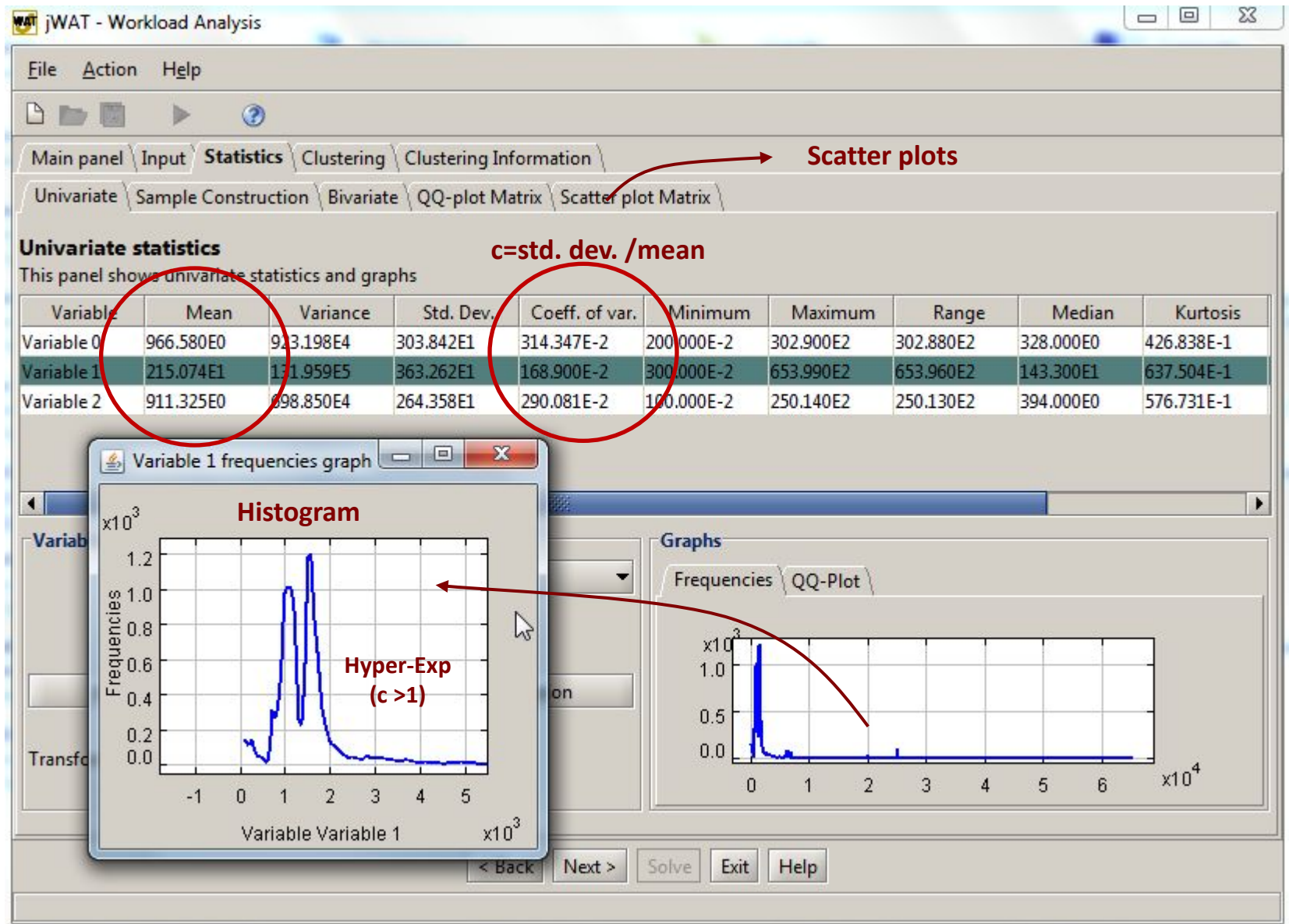
For filtering on variable see next step

LOAD

Load Data

< Back Next > Solve Exit Help

JWAT: workload characterization



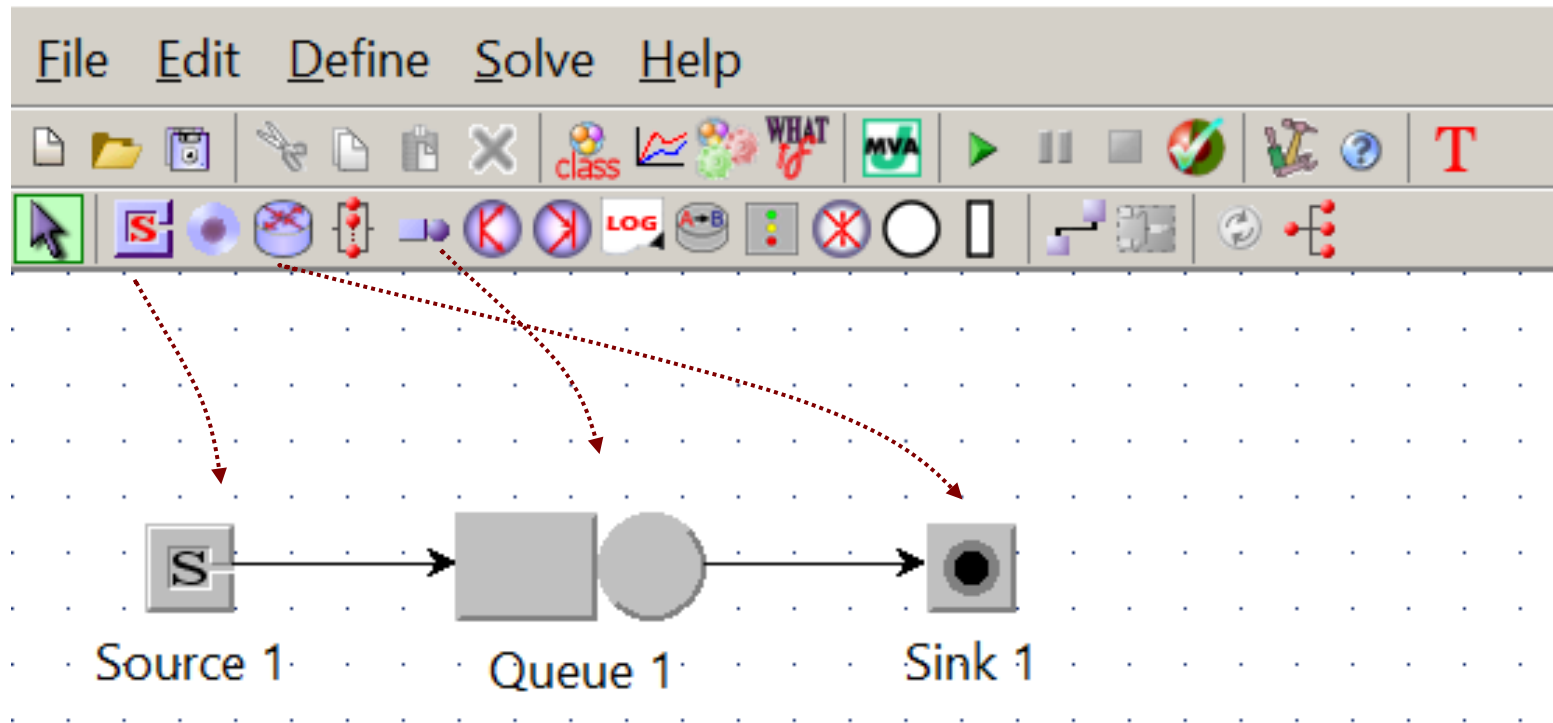
Activity 1: getting started

Hands-on activity: M/M/1

- Arrival rate: $\lambda=0.5$ job/s (Exponential)
- Service rate: $\mu=1.00$ job/s (Exponential)
- Goal: verify $\mu \Rightarrow \lambda$ property



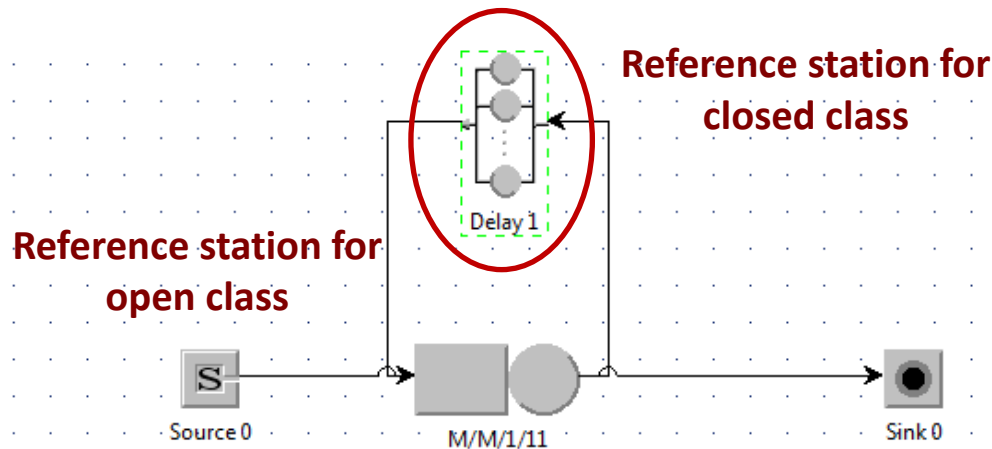
JSIMgraph - Graphical Queueing Network and Petri Net Simulator



Class definition



- Open, closed, and mixed workloads
- Priorities and reference stations



- Reference stations used for system metrics
- Visits at node = $TPUT_{node}/TPUT_{ref}$
- *ref* is the reference station

Define customer classes

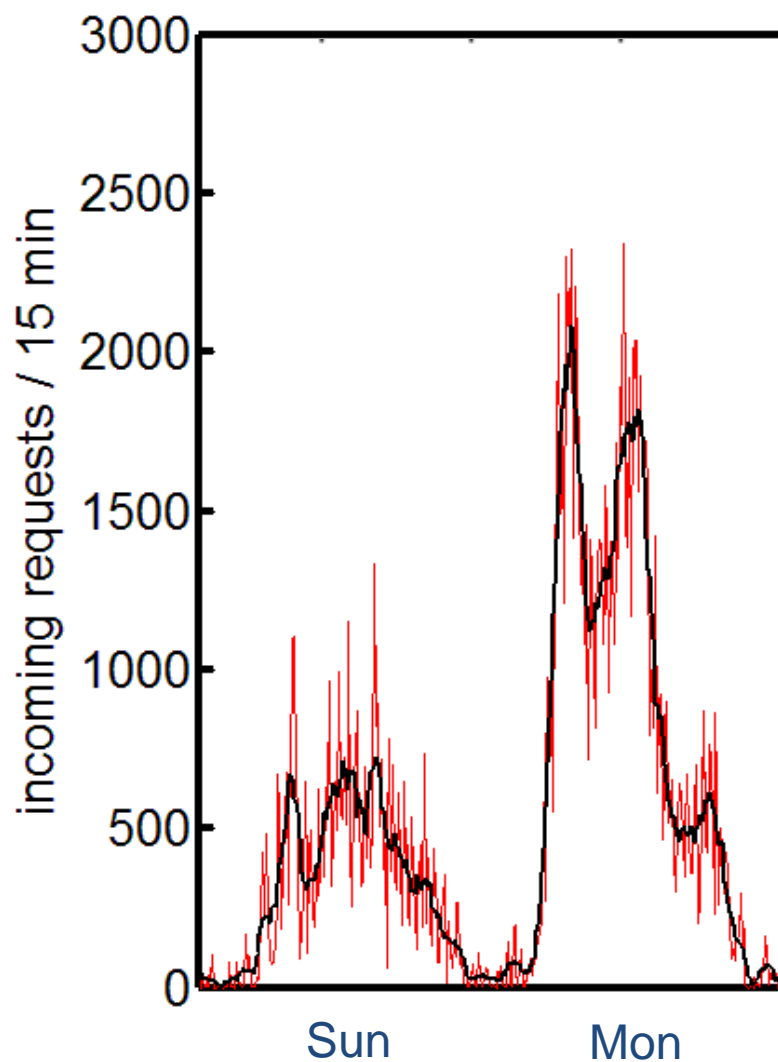
Classes Characteristics
Define type (Open or Closed), name and parameters for each customer class.
Closed Classes: If a **ClassSwitch** is in the model, then **all** the closed classes must have the **same** reference station.
Open Classes: An open class that has **Fork**, **ClassSwitch**, **Scaler** or **Transition** as the reference station is **not** generated by **any** Source.
Priorities: A larger value implies a higher priority.


Add Class

Classes: 2

Color	Name	Type	Priority	Population	Interarrival Time Distribution		Reference Station	
Blue	Class0	Open	0		exp(5)	Edit	Source 0	X
Red	Class2	Closed	0	10			Delay 1	X

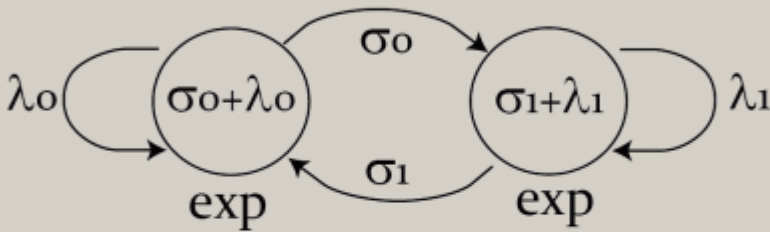
Arrival distribution



 Editing Class1 Service Time Distribution... ✕

Selected Distribution:

Markov-Modulated Poisson Process:
 $[mmpp2(\lambda_0, \lambda_1, \sigma_0, \sigma_1)]$



λ_0 :


λ_1 :

σ_0 :

σ_1 :

Queue section

- Non-preemptive scheduling: FCFS, LCFS, RAND, SJF, LJF, SEPT, LEPT, HOL (FCFS priority)
- Preemptive scheduling: PS, GPS, DPS

 Editing Queue 1 Properties... ✕

Station Name

Station Name:

Queue 1 Parameters Definition

Queue Section | Service Section | Routing Section

Capacity

☒ Infinite



☐ Finite

Buffer size

Max no. customers (queue+service)

Queue Policy

Station queue policy:

Class	Queue Policy	Drop Rule	Service Weight
 Class1	PS	Infinite Capacity	--
 Class2	PS	Infinite Capacity	--

Service section

Queue Section **Service Section** Routing Section

Number of Servers

Number: 1

Service Time Distributions

Service Time Distribution

Load Dependent Service Stra... Edit

exp(0.5) Edit

Strategy...

Add Range

Range of number of jobs inside the station. Mean expression, as a function of the current value of 'n'

Ranges: 2

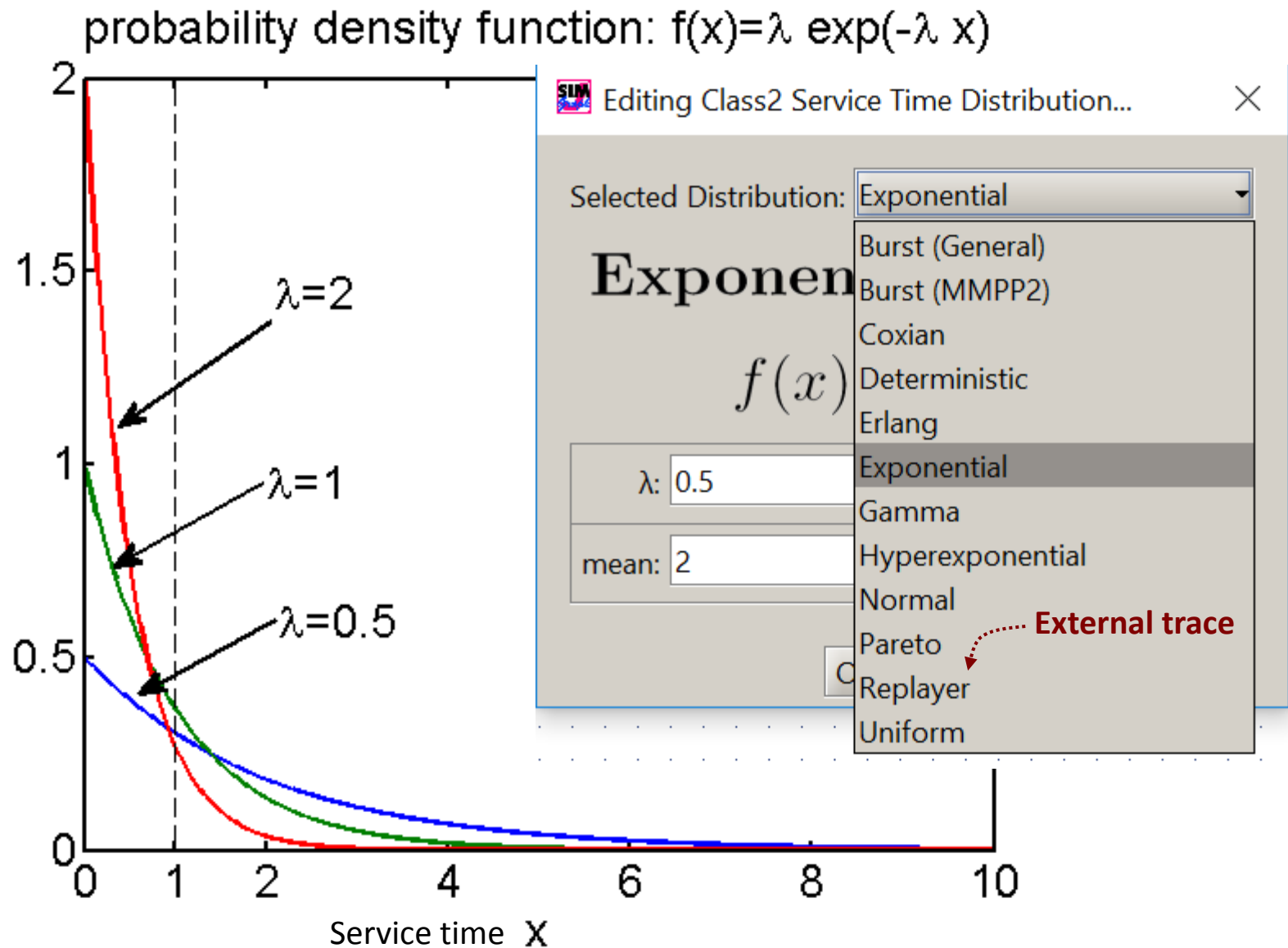
Mean C

1 10 Exponential 2

11 ∞ Exponential 1

queue-length	demand (exact)
1	0.15
2	0.10
3	0.07
4	0.05
5	0.04
6	0.03
7	0.02
8	0.02
9	0.01
10	0.01

Service time distribution



Perf. Indices



- 19 types of performance indices
 - Utilization, residence time, response time
 - Throughput, firing rates, drop rates, ...
- Granularity: system, station, class, mode, sink



Define performance indices



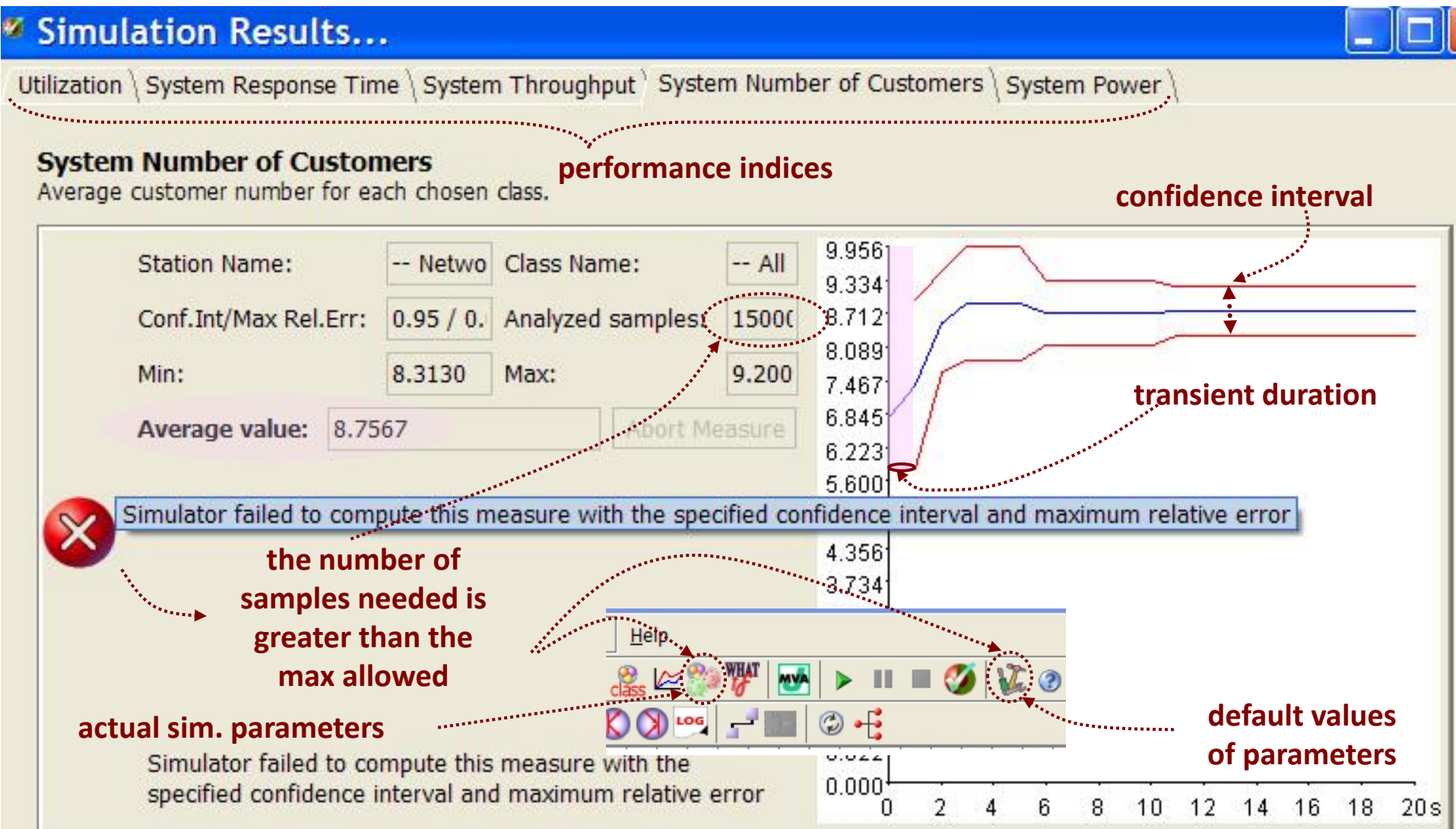
Performance Indices

Define performance indices to be collected and plotted by the simulation engine.

---Select an index---

Performance Index	Class/Mode	Station/Region	Stat.Res.	Conf.Int.	Max Rel.Err.
Utilization	--- All Classes ...	Queue 1	<input type="checkbox"/>	0.99	0.03
	--- All Classes ---				
	Class1				
	Class2				

Sim. Results



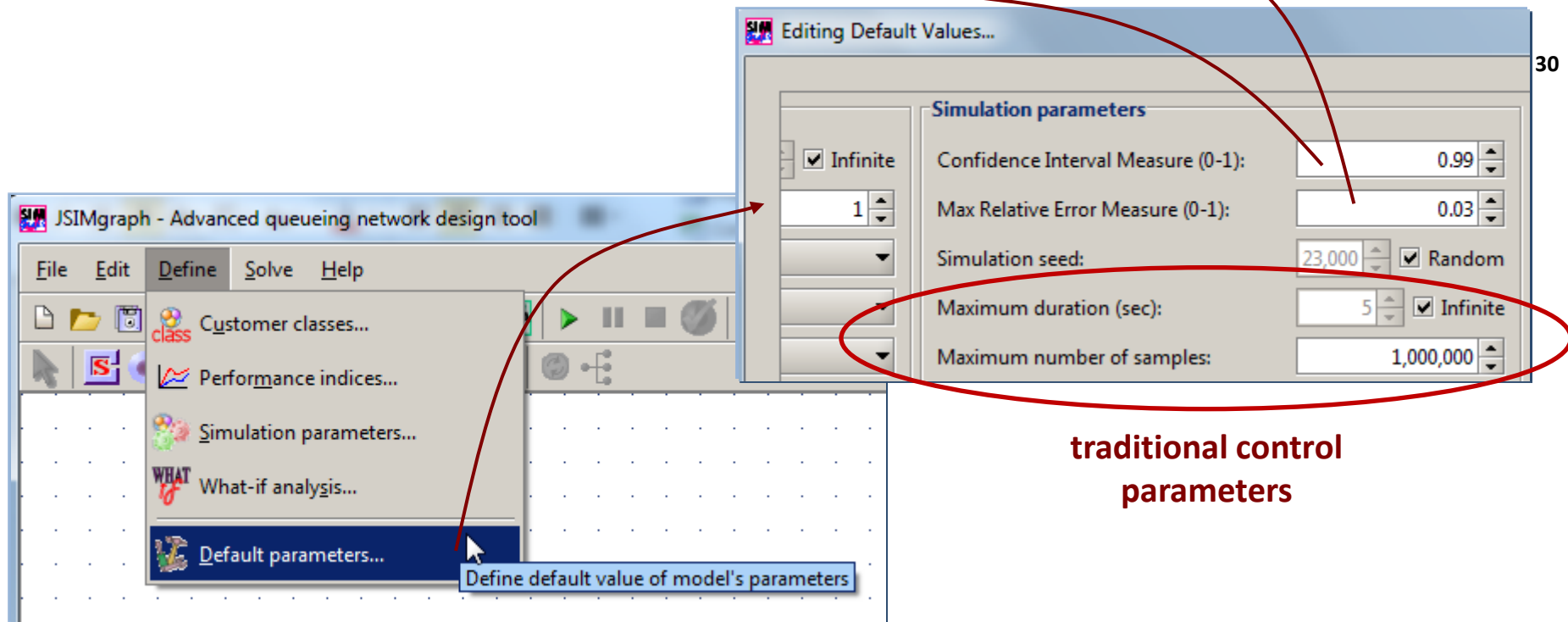
Statistical analysis

- Automated (overridable) simulation stop

$$1 - \alpha \leq P[\text{relative error} \leq \frac{\varepsilon}{1 - \varepsilon}]$$

confidence level

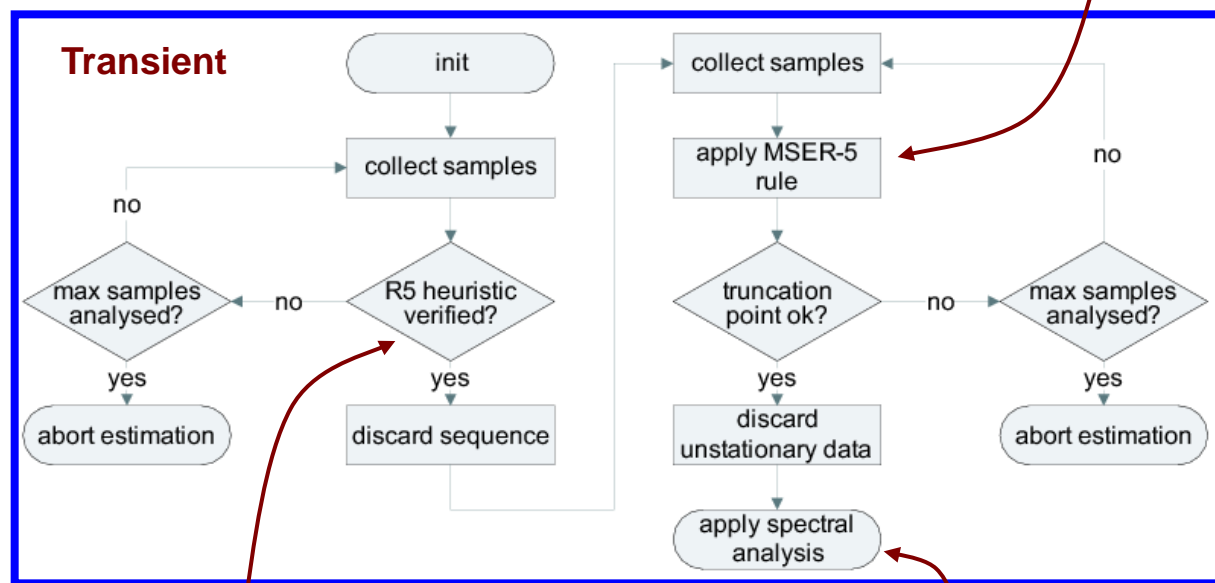
maximum relative error



Transient filtering

- Intelligent filtering of simulation data
 - R5 heuristics, spectral analysis, MSER-5 rule, ...

[Spratt, *M.S. Thesis*, 1998]



(Steady State)

[Pawlikowski, *CSUR*, 1990]

[Heidelberger&Welch, *CACM*, 1981]

Detailed statistical results

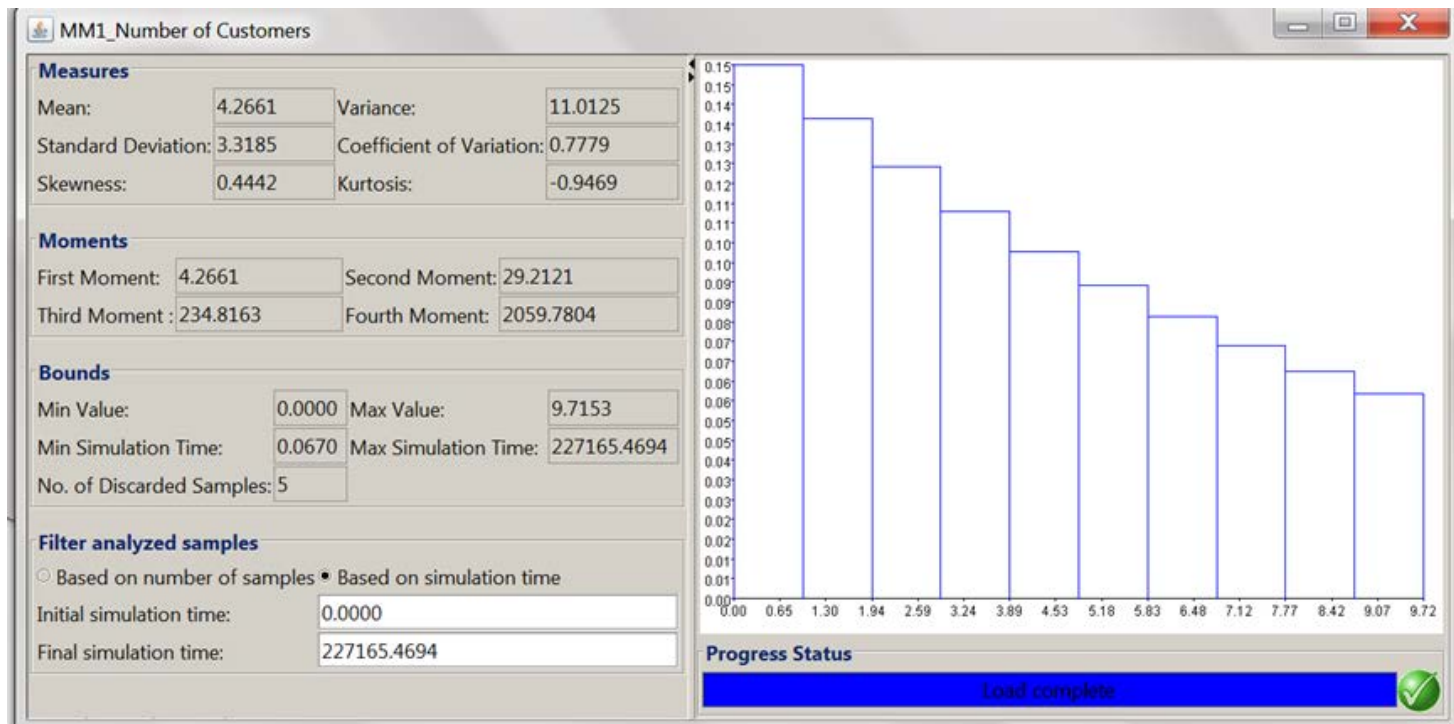
- Response time percentiles, buffer overflow probability, departure process moments, ...

Performance Indices

Define performance indices to be collected and plotted by the simulation engine.

---Select an index---

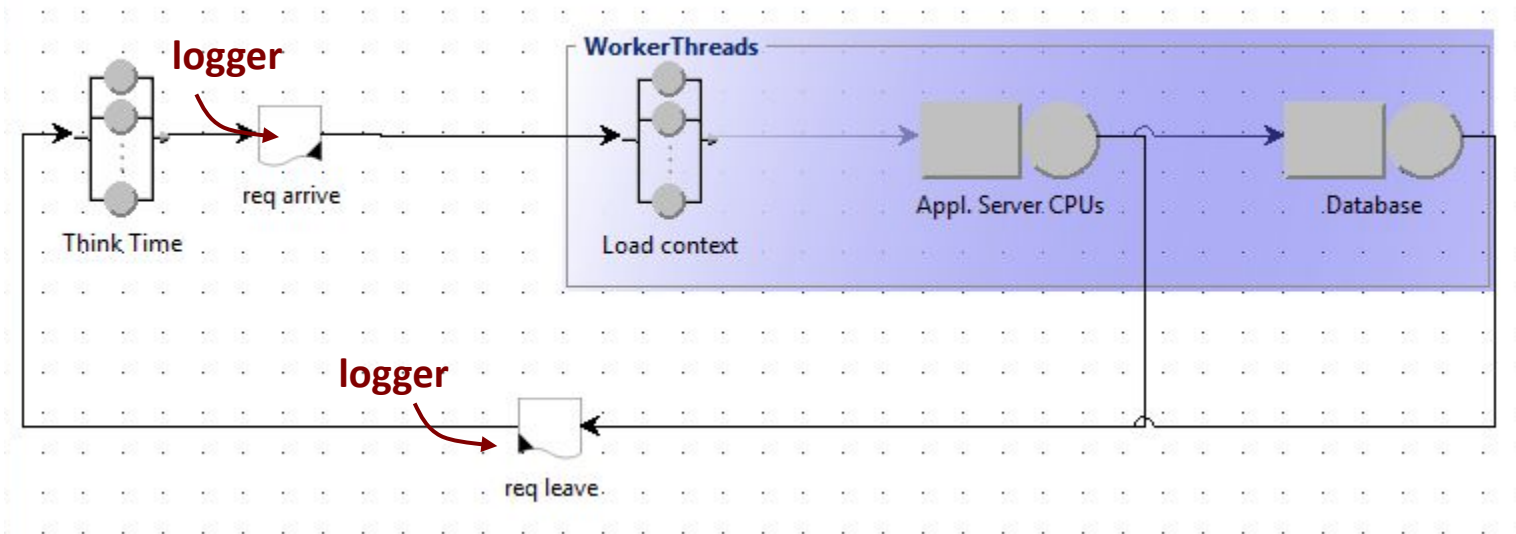
Performance Index	Class/Mode	Station/Region	Stat.Res.	Conf.Int.	Max Rel.Err.
Response Time	--- All Classes ...	Queue 1	<input checked="" type="checkbox"/>	0.99	0.03



Loggers



- Simulation events can be traced in CSV files



global.csv

job id (same throughout simulation)

```
LOGGERNAME;TIMESTAMP;JOB_ID;CLASS_ID;INTERARRIVAL_SAMECLASS;INTERARRIVAL_ANYCLASS;SIMUL_START_TIME
req arrive;0.009420010041266342;253625;Transactional workload;;;
req leave;0.0217557654334812;253625;Transactional workload;;;
req arrive;0.031032734664243056;253498;Transactional workload;;;
req leave;0.04915995332909814;253498;Transactional workload;;;
req arrive;0.07727161520772474;253542;Transactional workload;;;

```

job class

Activity 2: Load balancing

Routing section or

- Probabilistic routing
- State-dependent routing: JSQ, SRT, LU, FS
- Load-dependent probabilistic routing

The screenshot shows a software interface with three tabs: Queue Section, Service Section, and Routing Section. The Routing Section is active, displaying a table of Routing Strategies. The table has two columns: Class and Routing Strategy. Two classes are listed: Class1 and Class2. Both are currently set to 'Random'. A dropdown menu is open for Class2, showing a list of routing strategies: Random, Round Robin, Probabilities, Join the Shortest Queue (JSQ), Shortest Response Time, Least Utilization, Fastest Service, and Load Dependent Routing. To the right of the table, there is a 'Description' section stating: 'Jobs are routed randomly to stations connected to the current one. All routes have the same probability to be selected.' Below this is a 'Routing Options' section stating: 'No options available for this routing strategy'.

Class	Routing Strategy
Class1	Random
Class2	Random

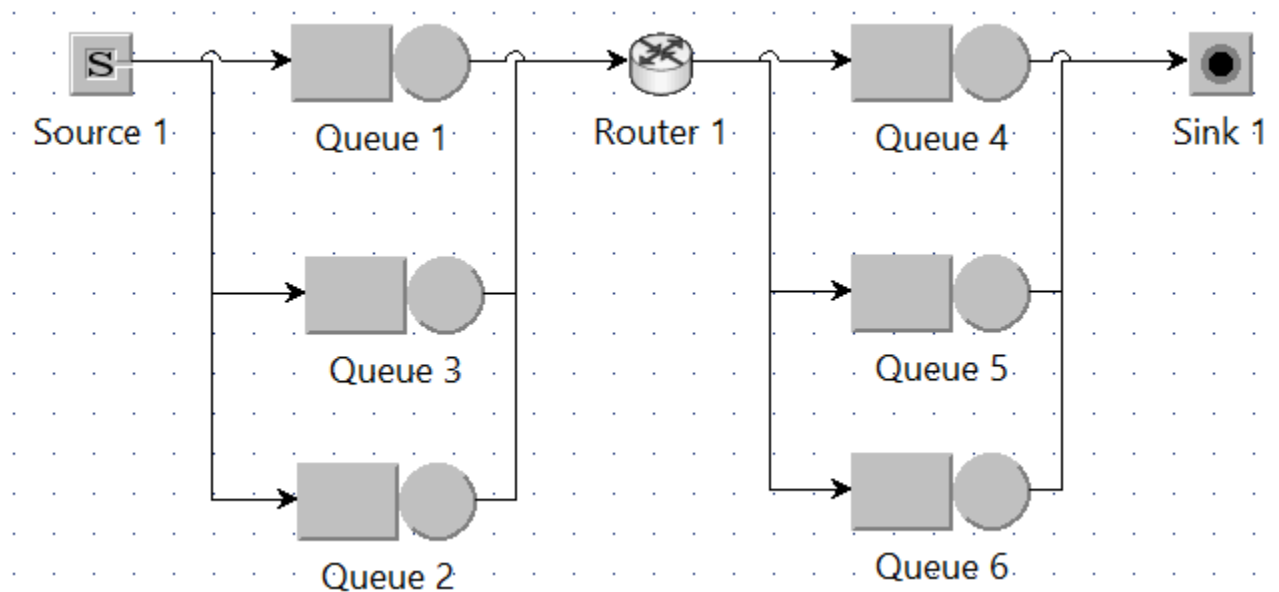
Description
Jobs are routed randomly to stations connected to the current one. All routes have the same probability to be selected.

Routing Options
No options available for this routing strategy

... router node

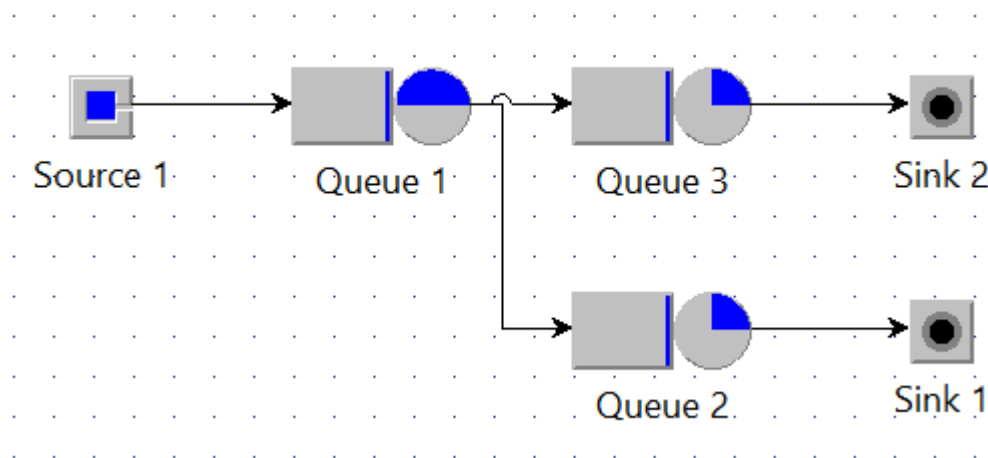


- Router node also allows to specify routing
 - Applies policy across multiple input queues
 - Same policies as routing section



Hands-on activity: load balancing

- We add two queues to the M/M/1 model.
- Goal: compare *round-robin* and *probabilistic* load-balancing



Blocking after service

Editing Server Properties...

Station Name
Station Name:

Server Parameters Definon

Queue Section | Service Section | Routing Section

Capacity

☐ infinite

☒ finite

max num.customers:

Queue Policy

Station queue policy:

Class	Queue Policy	Drop Rule
Class0	FCFS	BAS blocking

Waiting Queue (no drop)
BAS blocking
Drop

station with finite capacity

selection of the
BAS policy

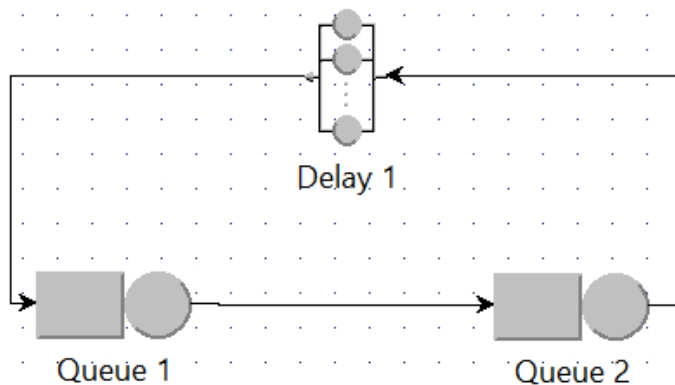
max number of requests
in the station

BAS policy:
requests are blocked in the
sender station when the max
capacity of the receiver
is reached

Activity 3: Parameter sweeping

Hands-on activity: bottleneck switch

- Analysis of bottleneck switch
- Measure: *Number of Customers*
- Demands: *Queue 1: 10 , 5; Queue 2: 5 , 9*



Define What-if analysis parameters

What-if Analysis

Define the type of What-If analysis to be performed and modify parameter options.

WARNING:

Enabling What-If analysis will disable all statistical outputs.

Parameter selection for the control of repeated executions

Population mix ▾

Type of population mix

Initial B:

Final B:

Steps (n. o...

Class:

Description

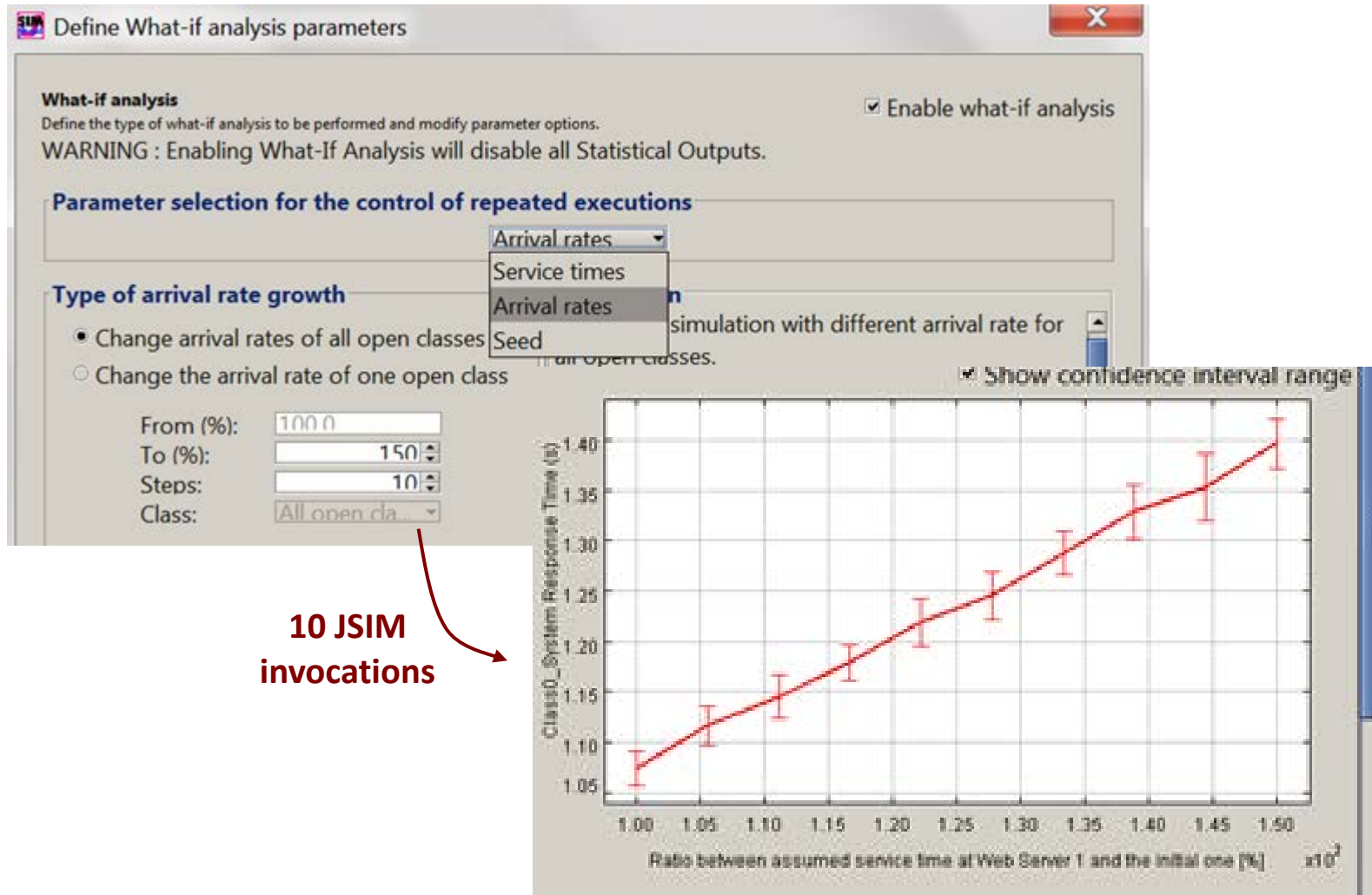
This type of analysis is available (other open classes) and it applies

Repeat the simulation changing (keeping constant the total number

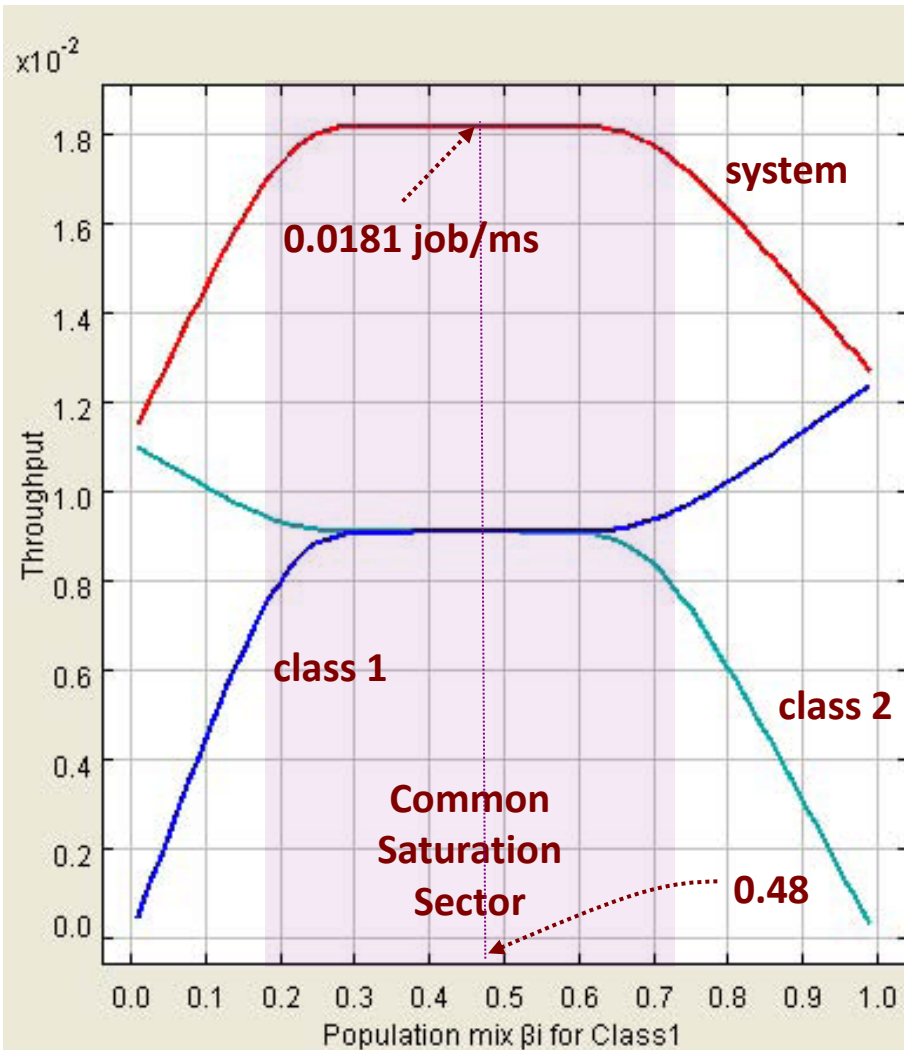
What-If analysis



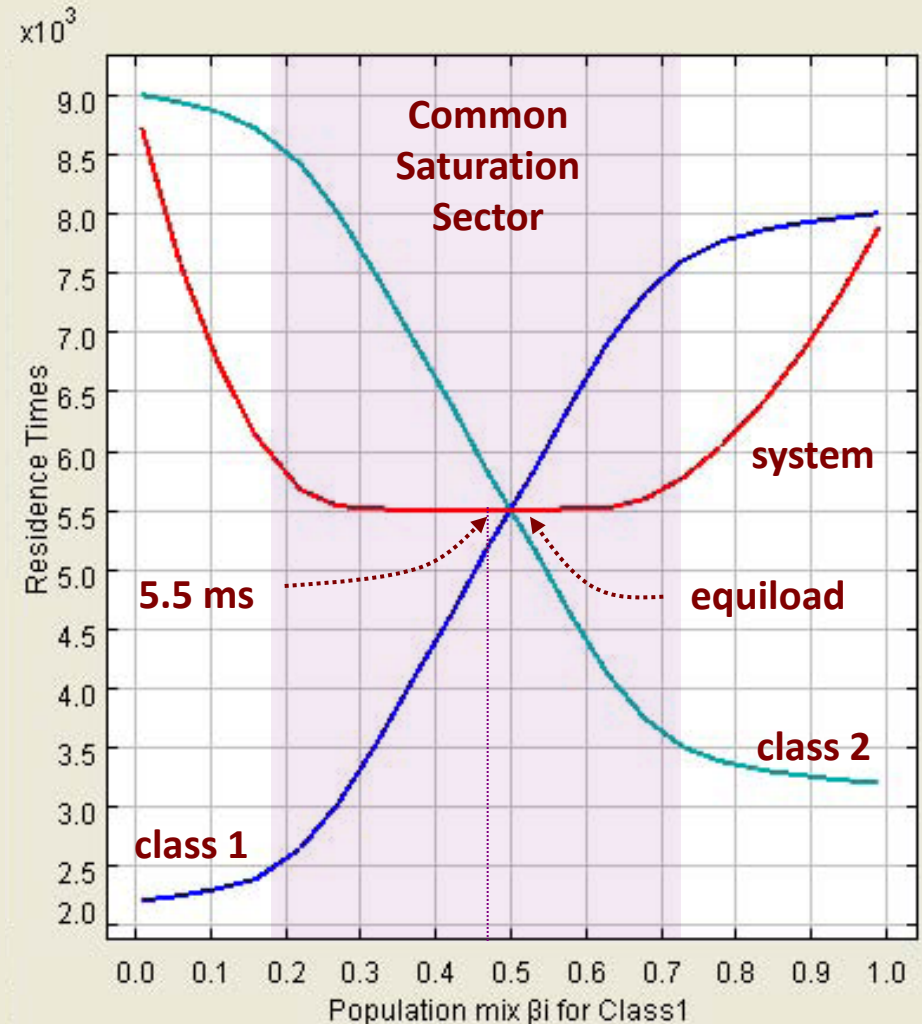
- Perform repeated executions automatically



JMVA: What-If



Throughput

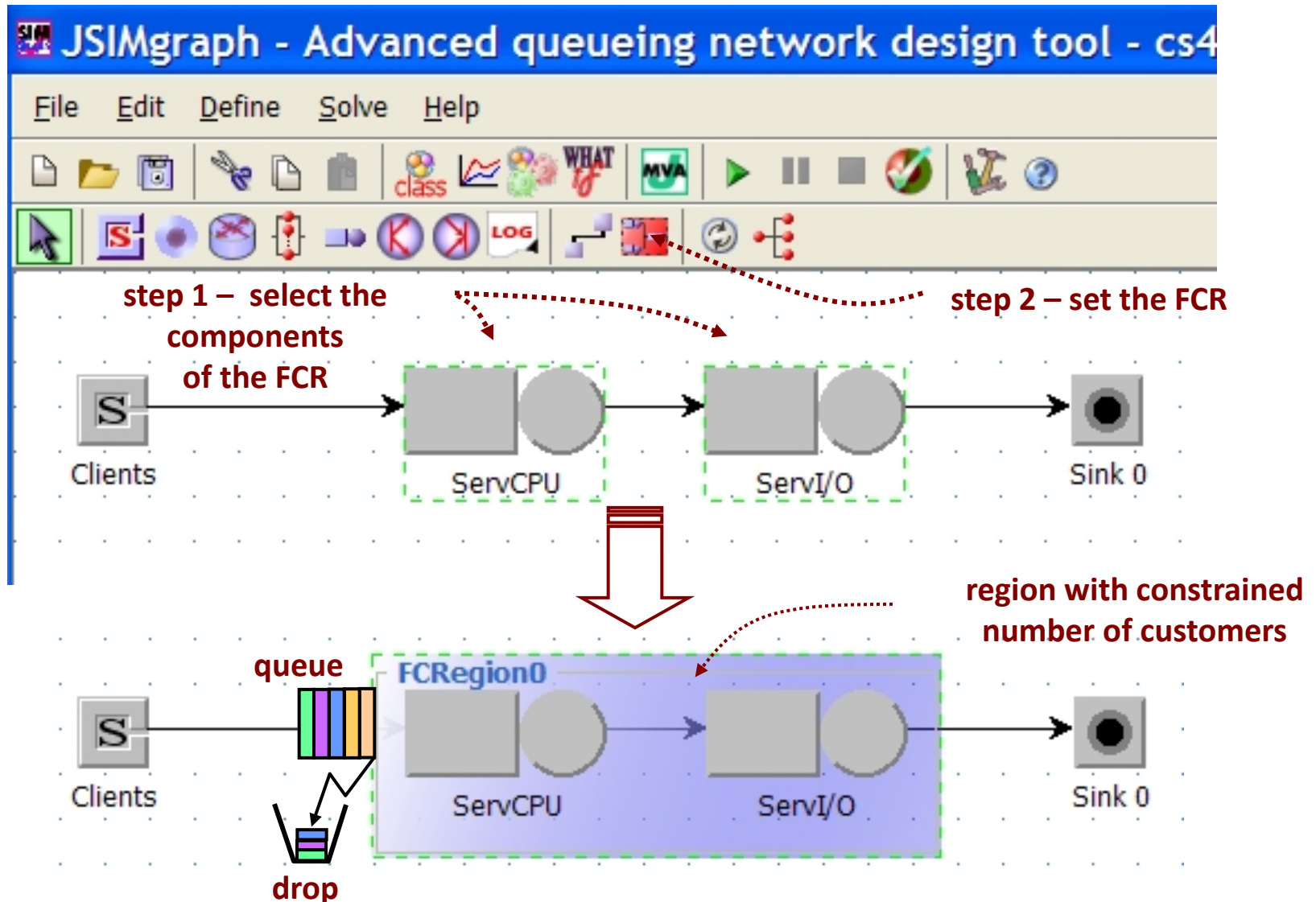


Response times

Activity 4: Capacity constraints

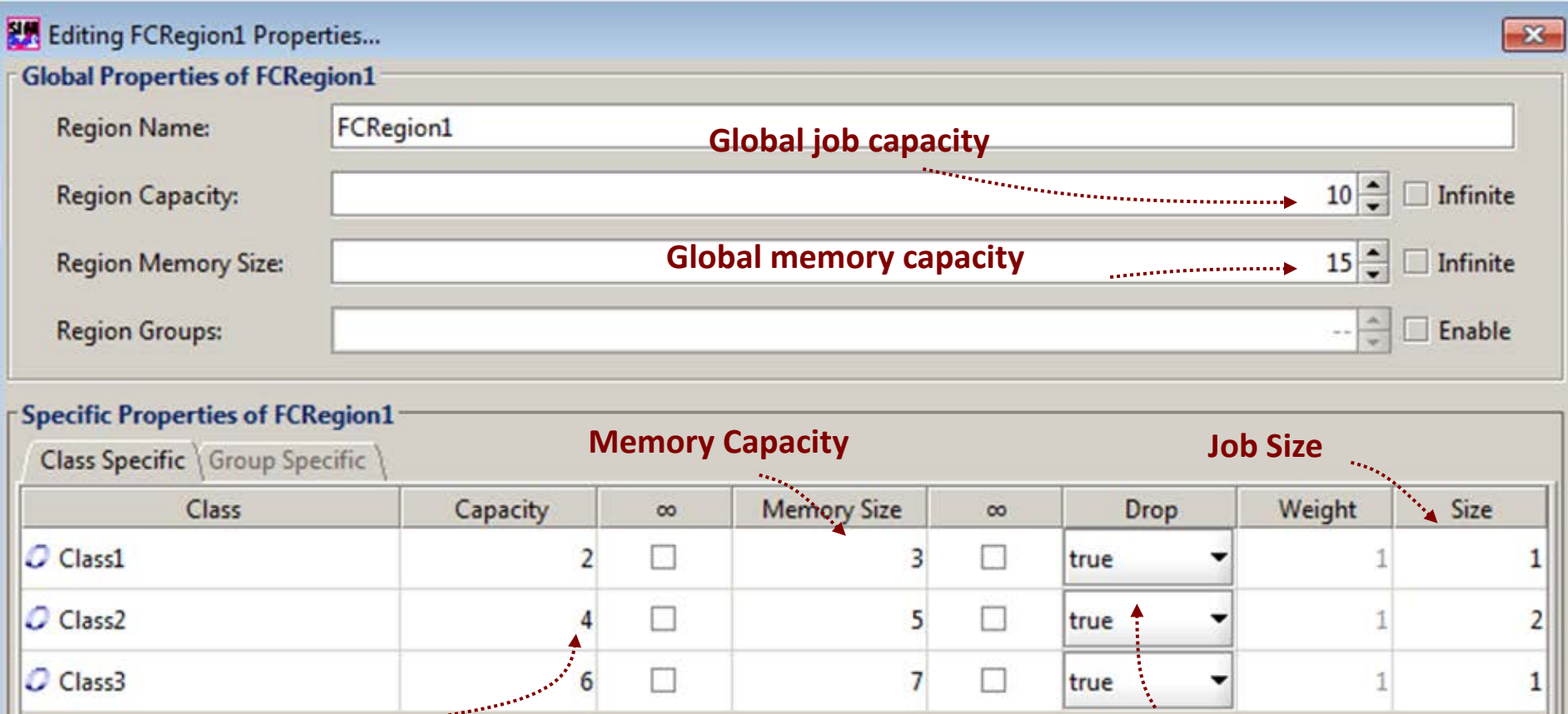
FCR definition

- Thread limits via *finite capacity regions* (FCRs)



FCR parameters

- Capacity constraints: total, per-class, per-group
- Memory constraints: jobs have sizes and must fit



Editing FCRegion1 Properties...

Global Properties of FCRegion1

Region Name: FCRegion1

Region Capacity: 10 ☐ Infinite

Region Memory Size: 15 ☐ Infinite

Region Groups: -- ☐ Enable

Specific Properties of FCRegion1

Class Specific \ Group Specific \

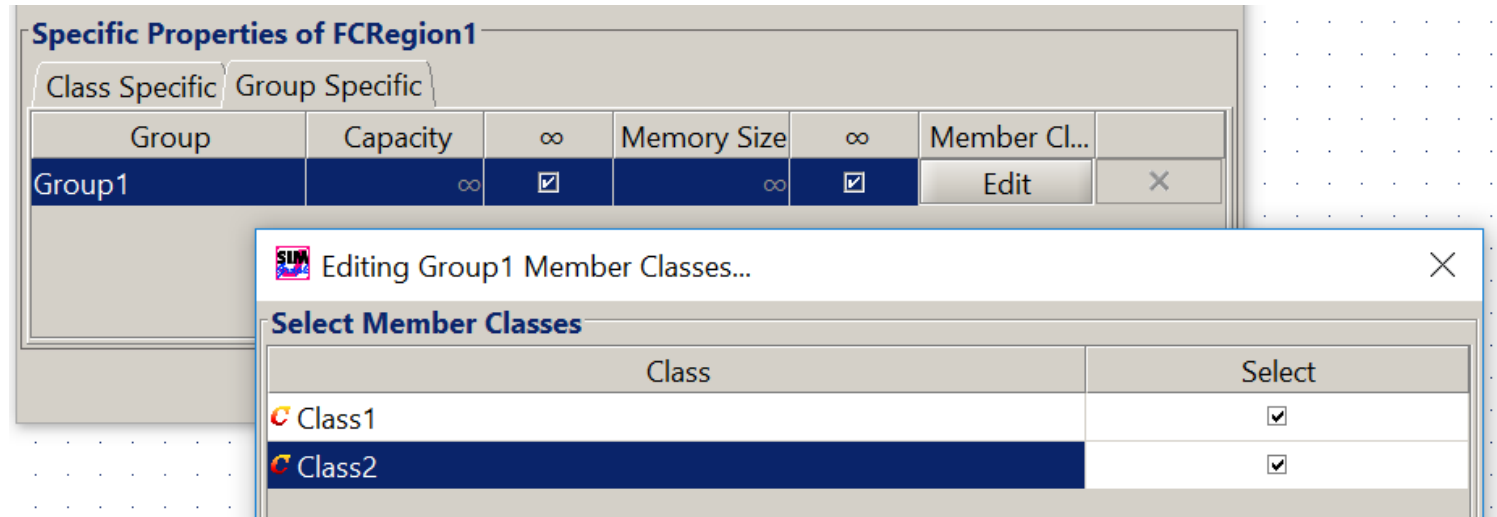
Class	Capacity	∞	Memory Size	∞	Drop	Weight	Size
Class1	2	<input type="checkbox"/>	3	<input type="checkbox"/>	true	1	1
Class2	4	<input type="checkbox"/>	5	<input type="checkbox"/>	true	1	2
Class3	6	<input type="checkbox"/>	7	<input type="checkbox"/>	true	1	1

max number of requests per class in the FCR

drop the requests when the region capacity is saturated

FCR groups and indices

- Group-specific constraints (*i.e.*, for subset of classes)



- Dedicated performance indices

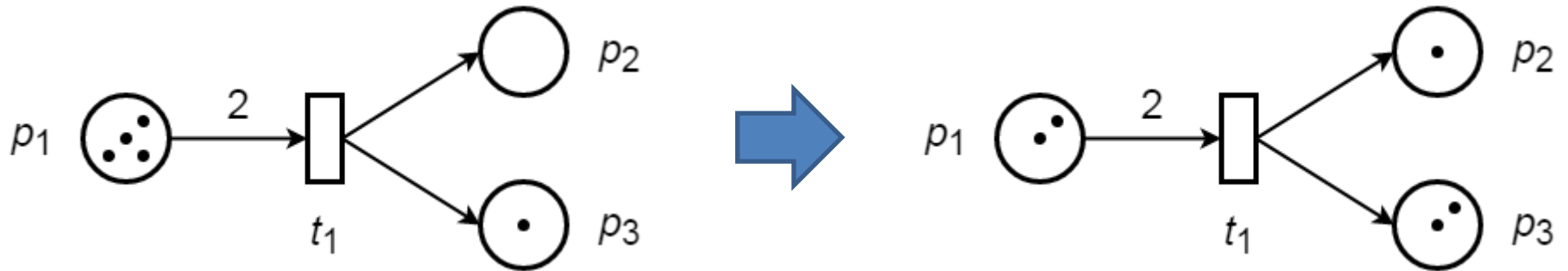
Performance Indices
Define performance indices to be collected and plotted by the simulation engine.

---Select an index---

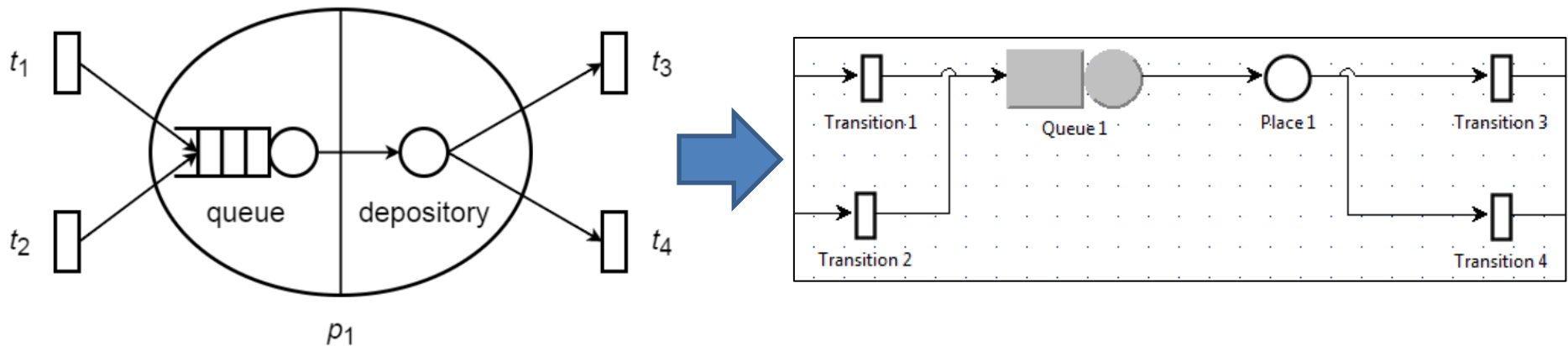
Performance Index	Class/Mode	Station/Region	Stat.Res.	Conf.Int.	Max Rel.Err.
Number of Customers	--- All Classes ---	FCRegion1	<input type="checkbox"/>	0.99	0.03
Response Time	--- All Classes ---	FCRegion1	<input type="checkbox"/>	0.99	0.03
FCR Total Weight	--- All Classes ---	FCRegion1	<input type="checkbox"/>	0.99	0.03
FCR Memory Occupation	--- All Classes ---	FCRegion1	<input type="checkbox"/>	0.99	0.03

Support for PN elements

- Places and transitions



- Queueing Petri nets



PN sections & modes

- JMT design paradigm extends to PN elements
- Mode: a rule to activate and fire a transition



Place Station

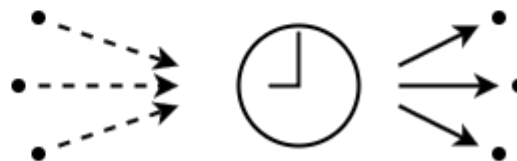


Store
Section

Tunnel
Section

Link
Section

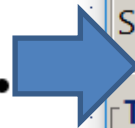
Transition Station



Enabling
Section

Timing
Section

Firing
Section



Editing Transition 1 Properties...

Station Name
Station Name: **Transition 1**

Transition 1 Parameters Definition

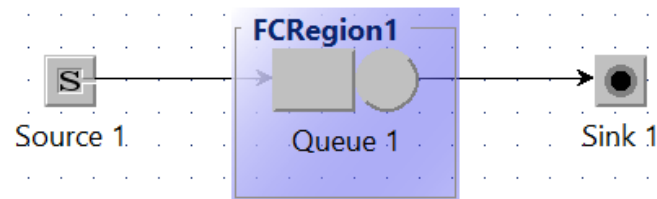
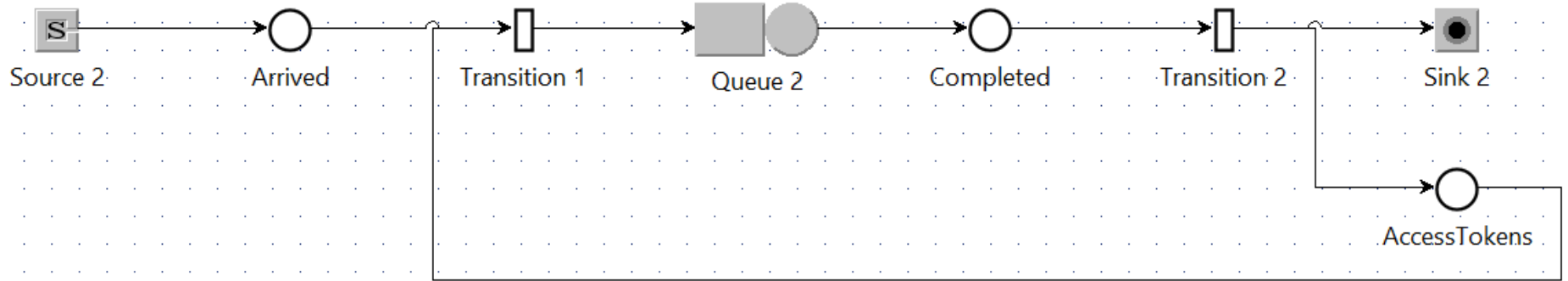
Enabling Section | Timing Section | Firing Section

Enabling Condition for Mode1

	Class1	Class2
Place 1	5	0

Hand-on activity: FCRs vs QPNs

- Arrival rate: $\lambda=0.99$ job/s
- Service rate: $\mu=1.00$ job/s
- Goal: restrict max 1 job inside queue

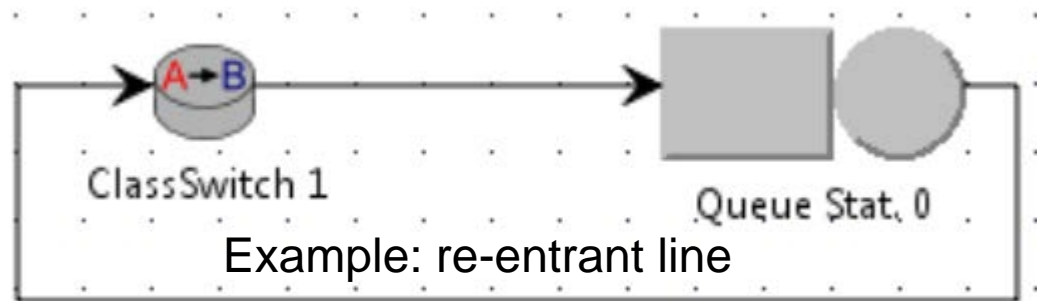


Activity 5: Workflows & fork-join

Class-switching



- A job can change its class during the simulation
 - Workflows, re-entrant lines, track path-wise perf., ...



Editing ClassSwitch 1 Properties...

Station Name

Station Name: ClassSwitch 1

ClassSwitch 1 Parameters Definition

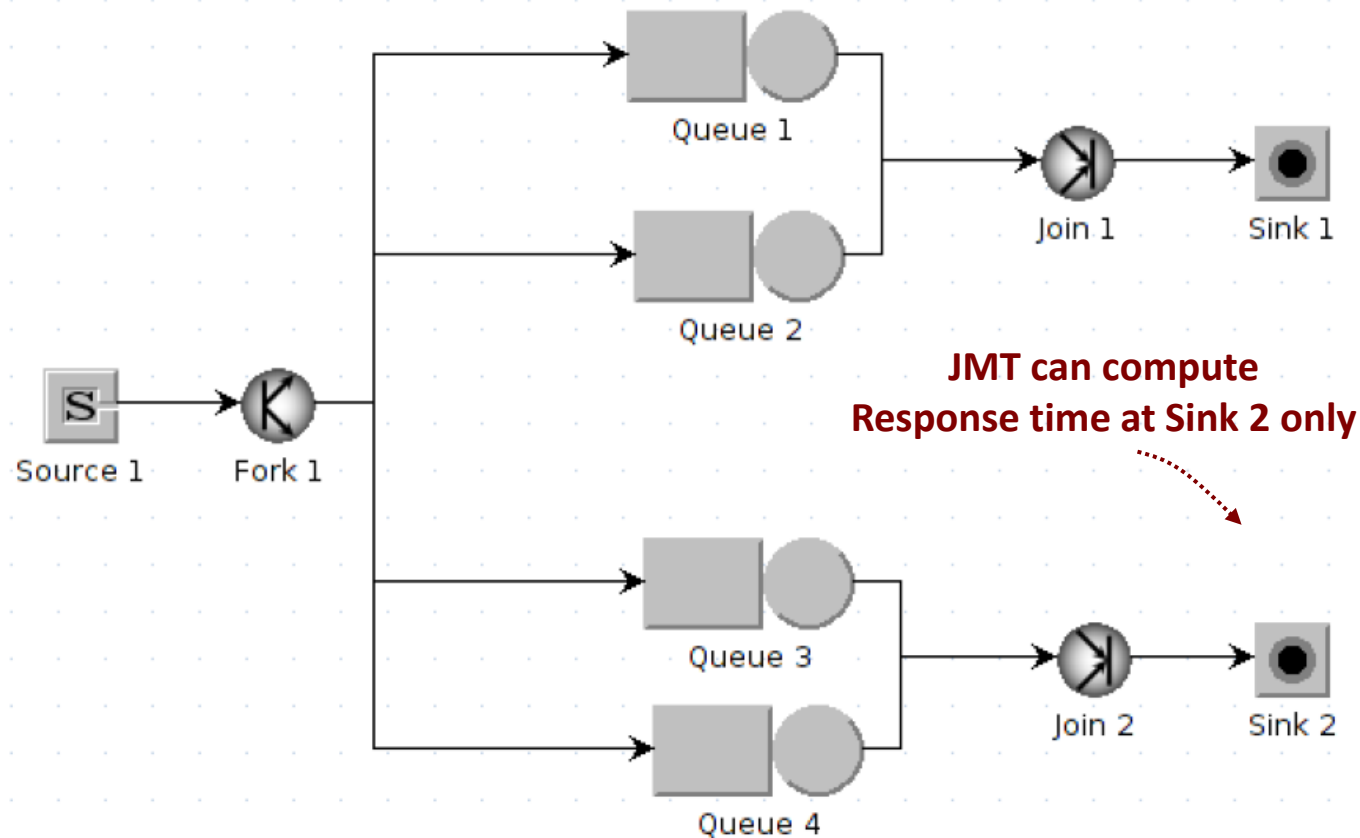
Class Switch Matrix Routing Section

CS Strategies

*	Class0	Class1
Class0	0.6 (60%)	0.4 (40%)
Class1	0.8 (80%)	0.2 (20%)

Fork-Join elements

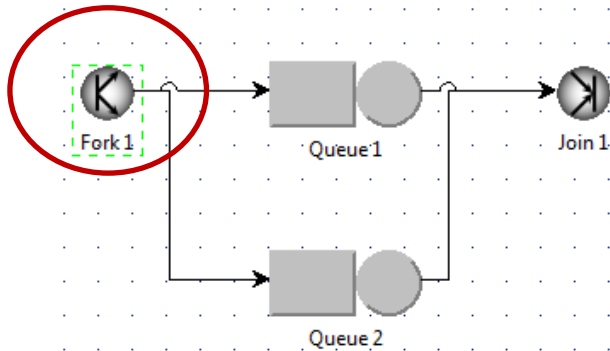
- Jobs split into *tasks* carrying *id* of the parent job
- Support for:
 - nested fork-joins
 - multiple join points
 - finite capacity between fork-join
 - advanced policies (e.g., quorum)



Advanced fork



- Branch prob.: randomize no. tasks and output links
- Random subset: choose n-out-of-k output links
- Class Switch: assign new class to forked tasks



Editing Fork1 Properties...

Station Name
Station Name: Fork1

Fork 1 Parameters Definition

Queue Section Capacity Section Fork Strategies

☒ Enable Advanced Fork Strategies

Standard Fork Strategy

Number of tasks to be generated on each output link for each input job (customer) to the Fork: 1

Advanced Fork Strategies

Class	Fork Strategy
Class1	Branch Probabilities

Description

Each output link has a probability p of generating tasks. When a output link is selected, the number of tasks to generate is described by a custom distribution (one for each output link).

Branch Probabilities

Destination	Probability
Queue 1	0.8
Queue 2	0.6

Task Distribution

Number of Tasks	Probability
1	0.5
3	0.5

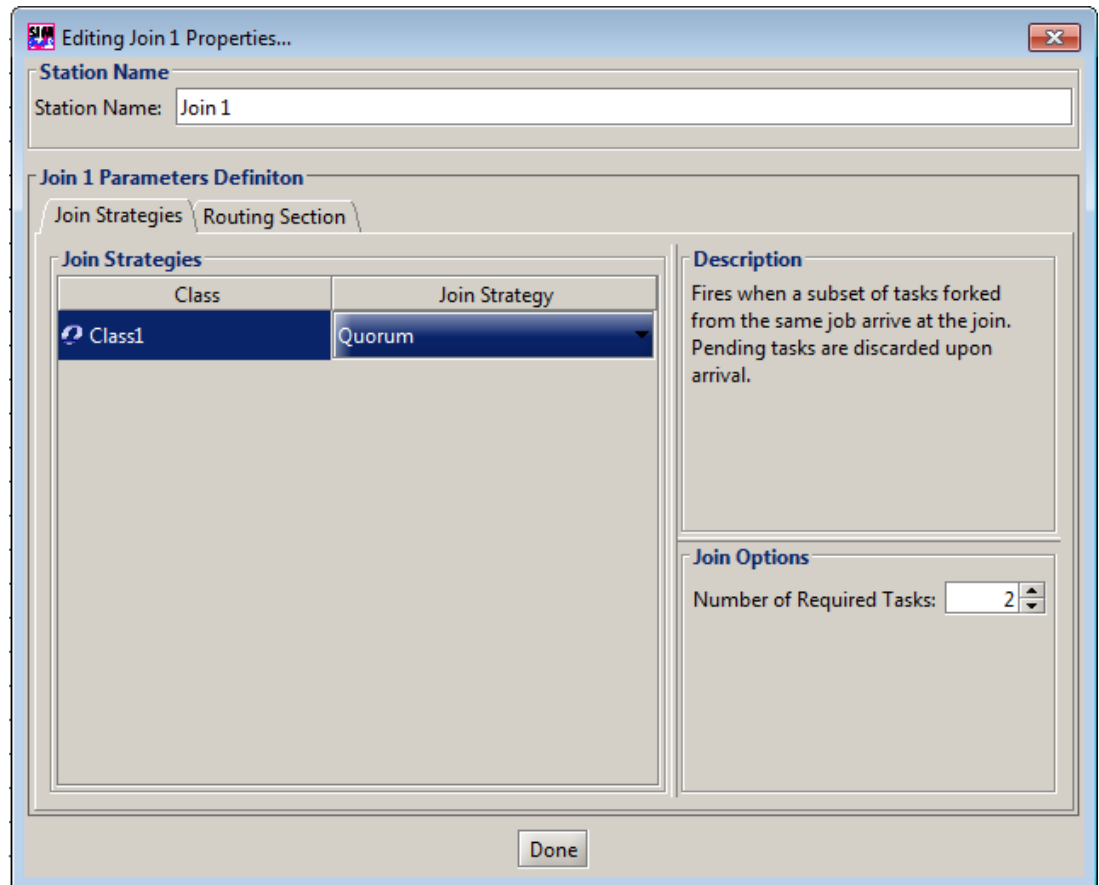
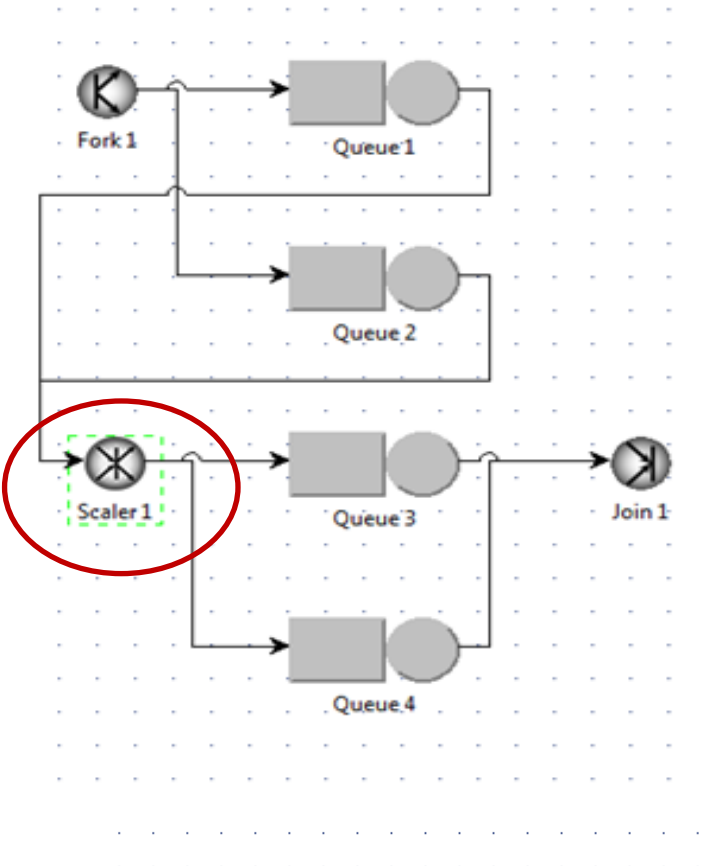
Add Delete

Done

Advanced join



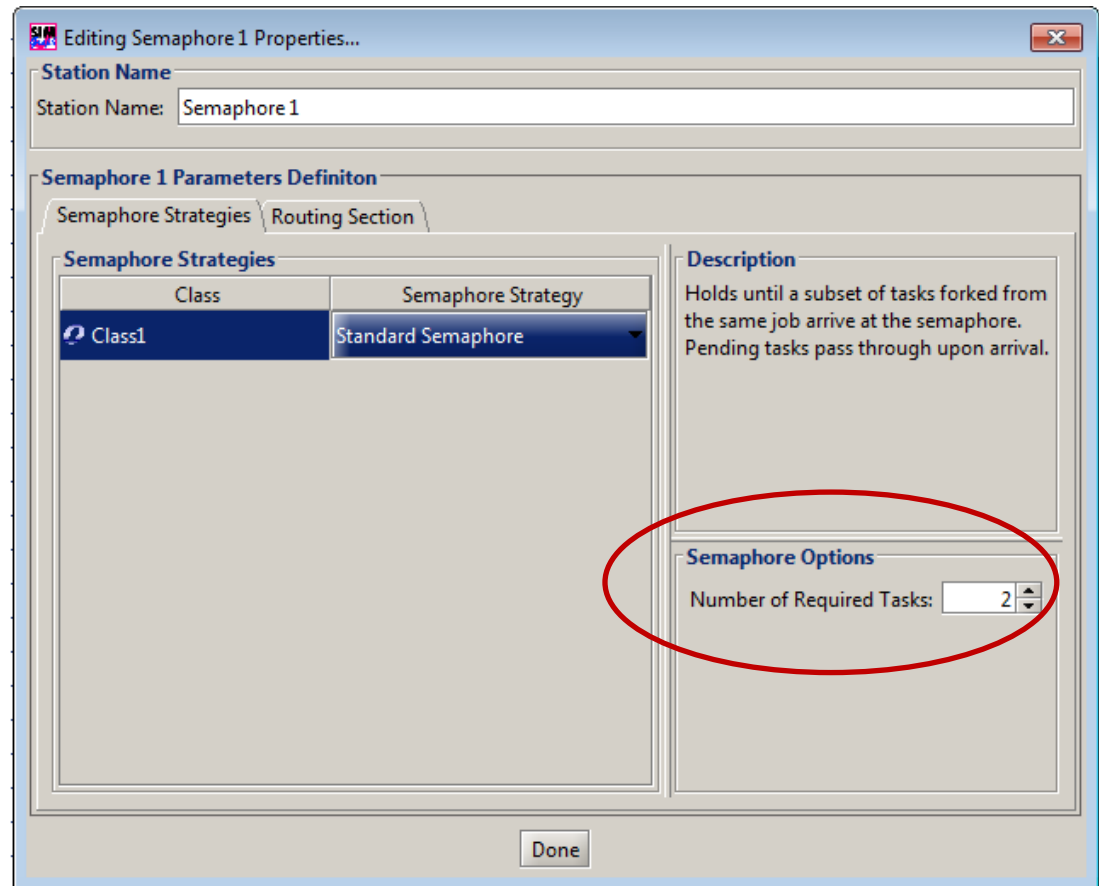
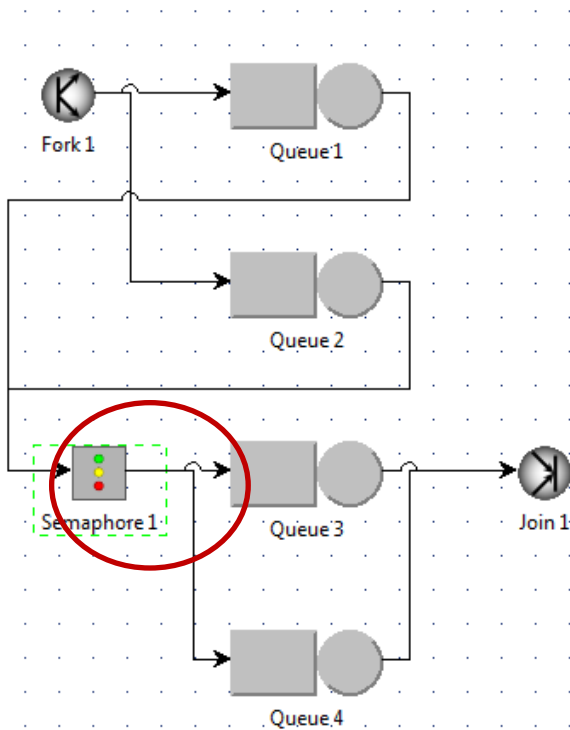
- Quorum: wait a subset of tasks (of the same job)
- Guard: like quorum but requires given class mix
- Scaler: join then fork again



Semaphore

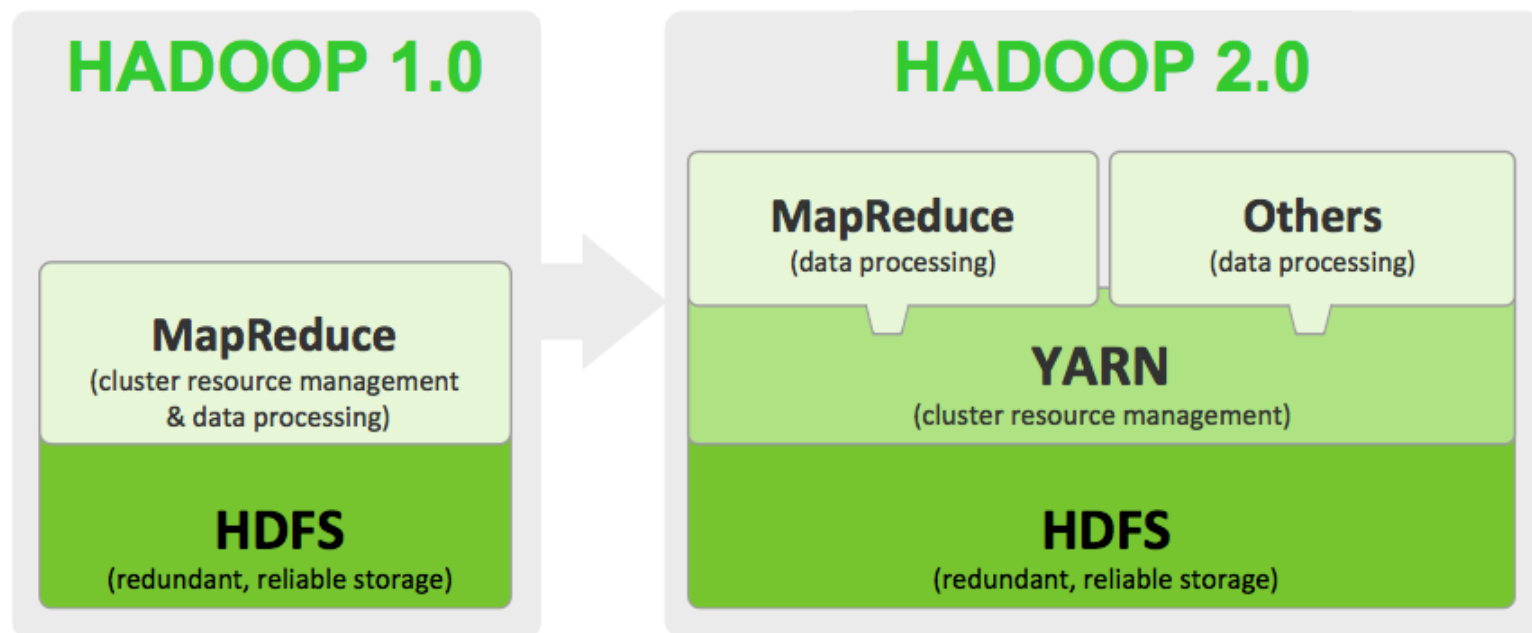


- Block first N tasks forked from the same job
- Upon arrival of the Nth, unblock and let the other pass



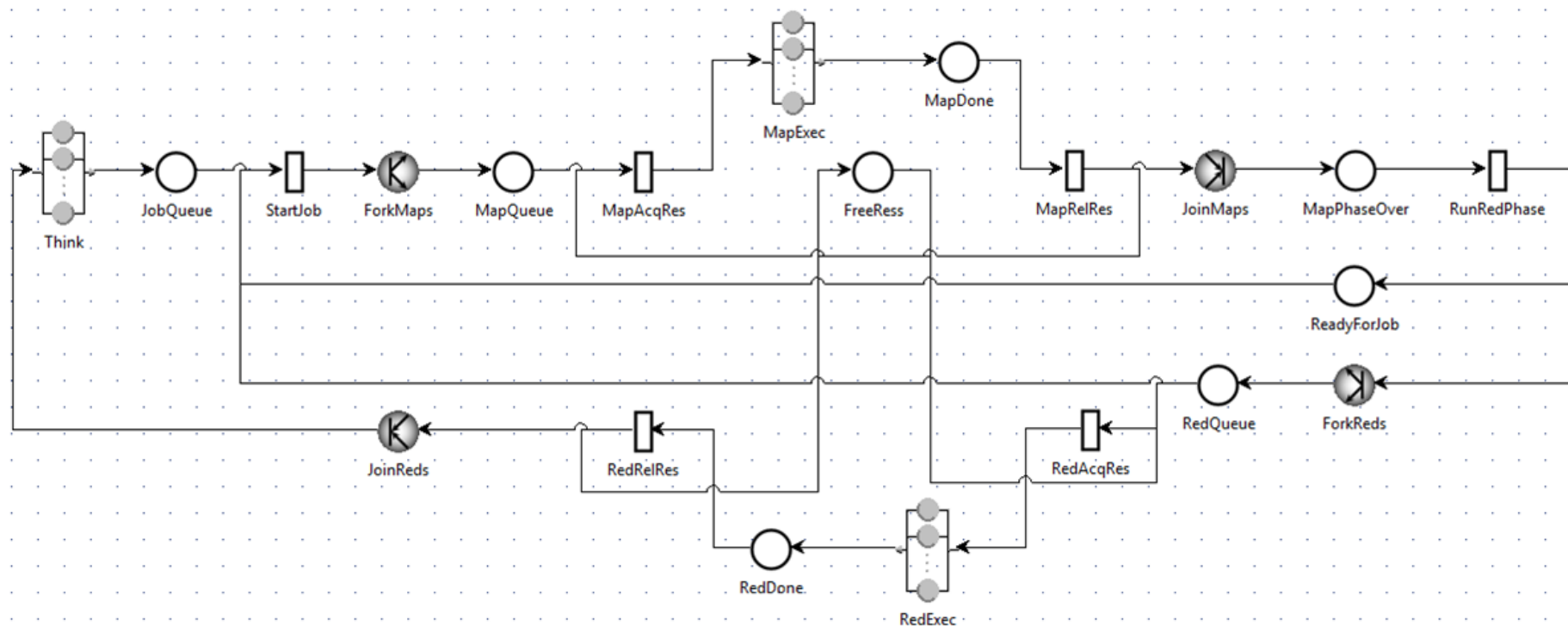
Case Study: YARN Capacity Scheduler

- YARN — Yet Another Resource Negotiator



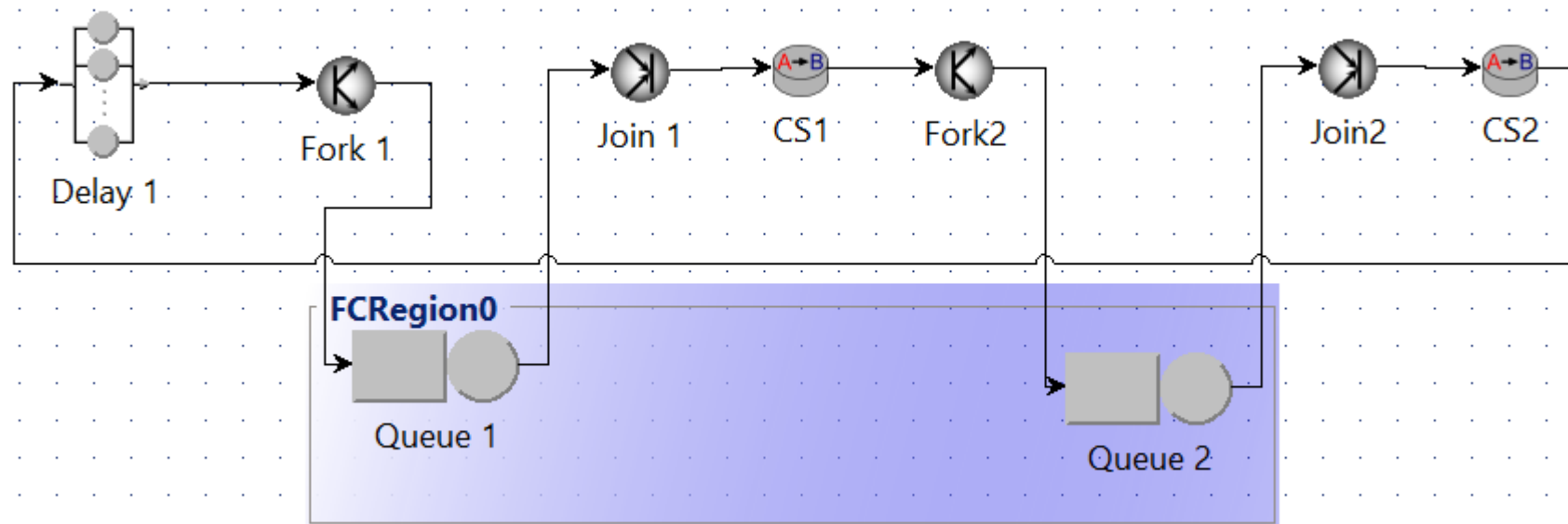
Case Study: YARN Capacity Scheduler

- Detailed model using QPN
 - Nested FCRs (JobQueue, MapQueue, RedQueue)
 - 14.13% error in trace-driven simulation [D. Ardagna *et al.*, ICA3PP'16]



Case Study: YARN Capacity Scheduler

- Simplified model using QN
 - Class switching between Map tasks and Reduce tasks



Conclusion

Coming Soon (\geq version 1.0.3)

- Customer impatience
- Ability to parallelize JMT on multiple cores
 - Collect samples or run what-ifs in parallel
 - Internal simulation remains single-threaded
- New load-balancing policies
 - Power of k choices
 - SITA
 - ...
- TreeMVA in JMVA for sparse networks

