Imperial College London

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

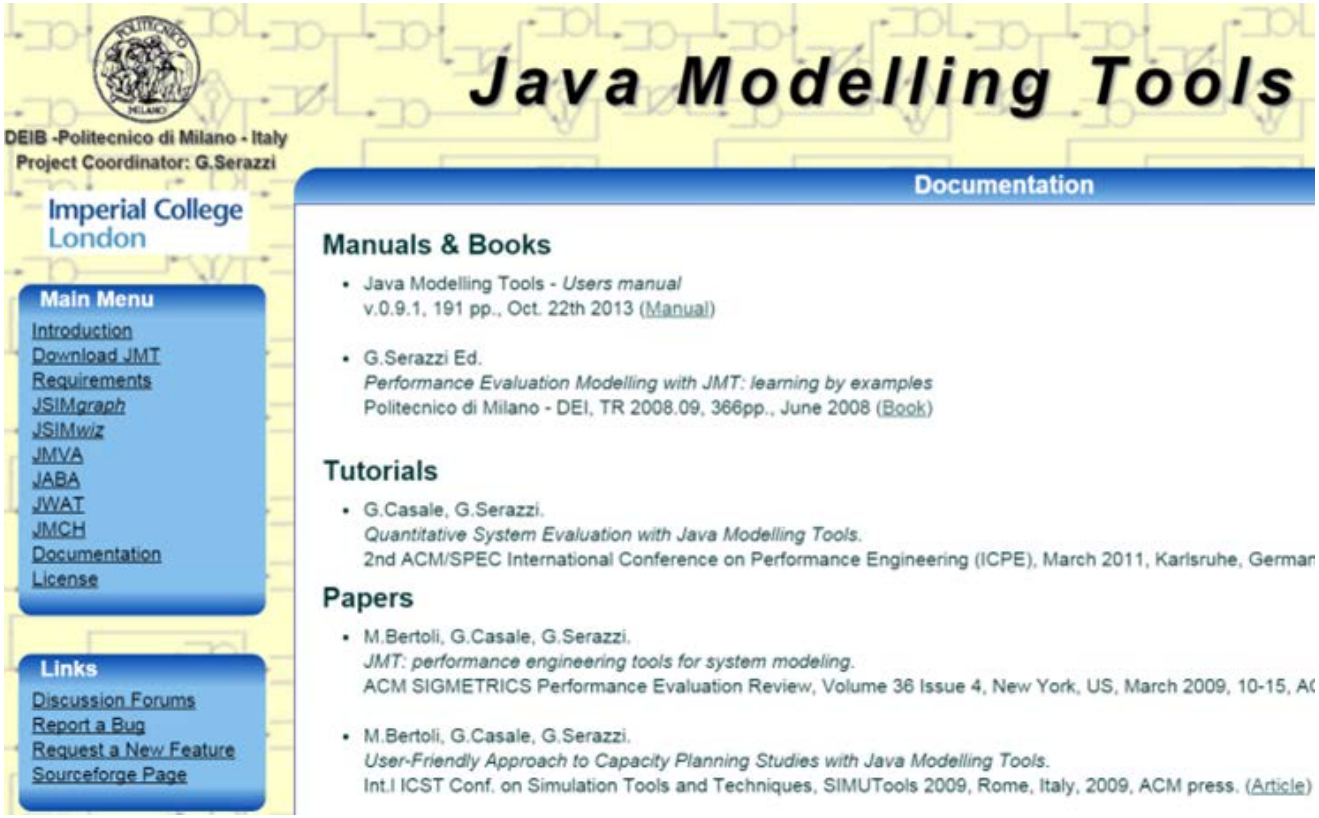
- Simulation and analysis of queueing networks.
- Project started in 2002 at Politecnico di Milano, since 2010 co-developed at Imperial.
- JMT is open source: GPL v2,
  - Medium-size project: ~1,000 classes
  - JAR, source code and maven build files (*pom.xml*)
    http://jmt.sourceforge.net/Download.html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

Supported models

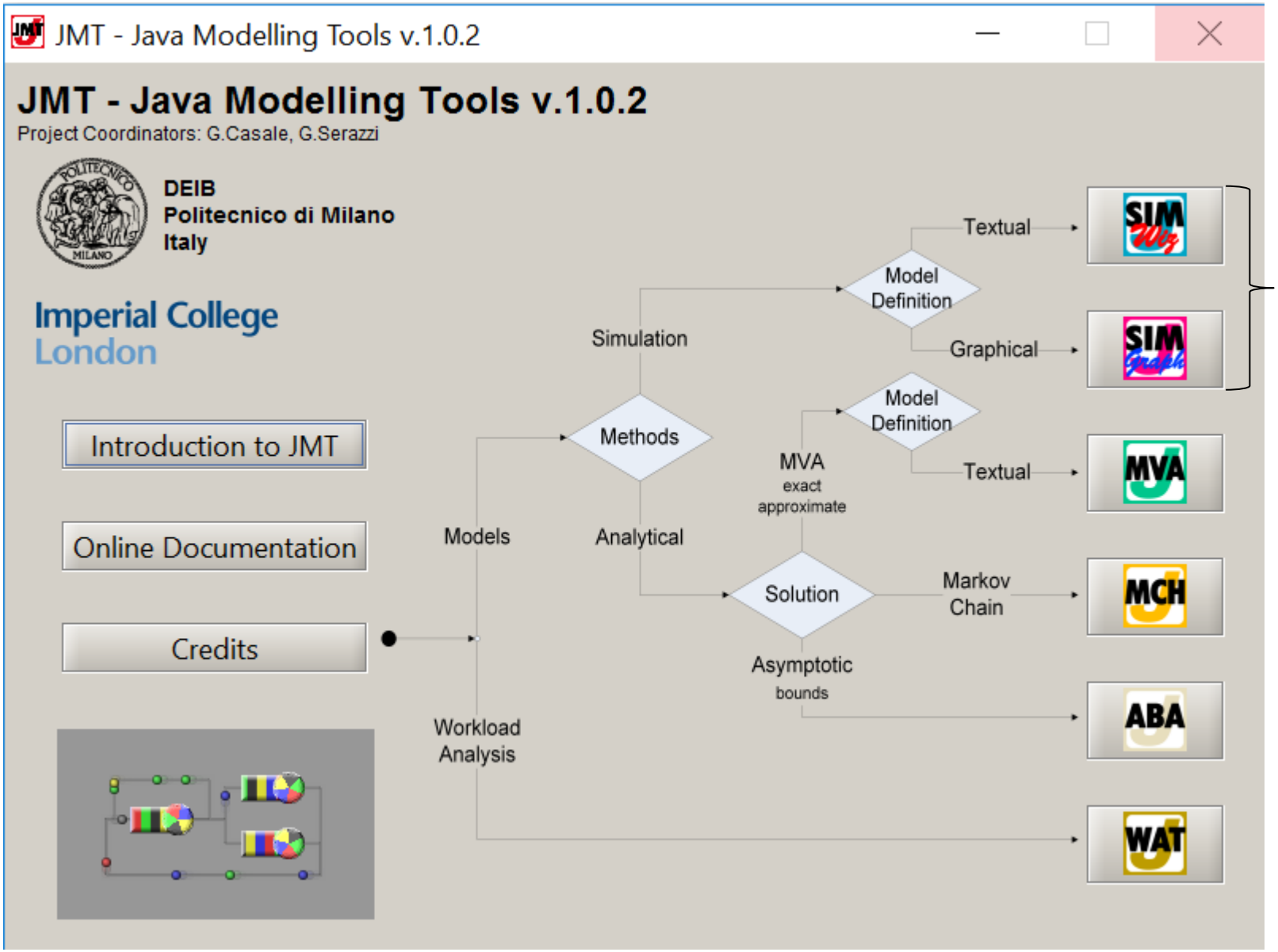
- Queueing Systems
- Queueing Networks (QN)
  - Product-form
  - Extended (fork/join, blocking, priorities, …)
- Petri Nets (PN)
  - Stochastic Petri Nets (SPN)
  - Generalized Stochastic Petri Nets (GSPN)
  - Coloured Petri Nets (CPN)
- Queueing Petri Nets (QPNs)

Who uses JMT?

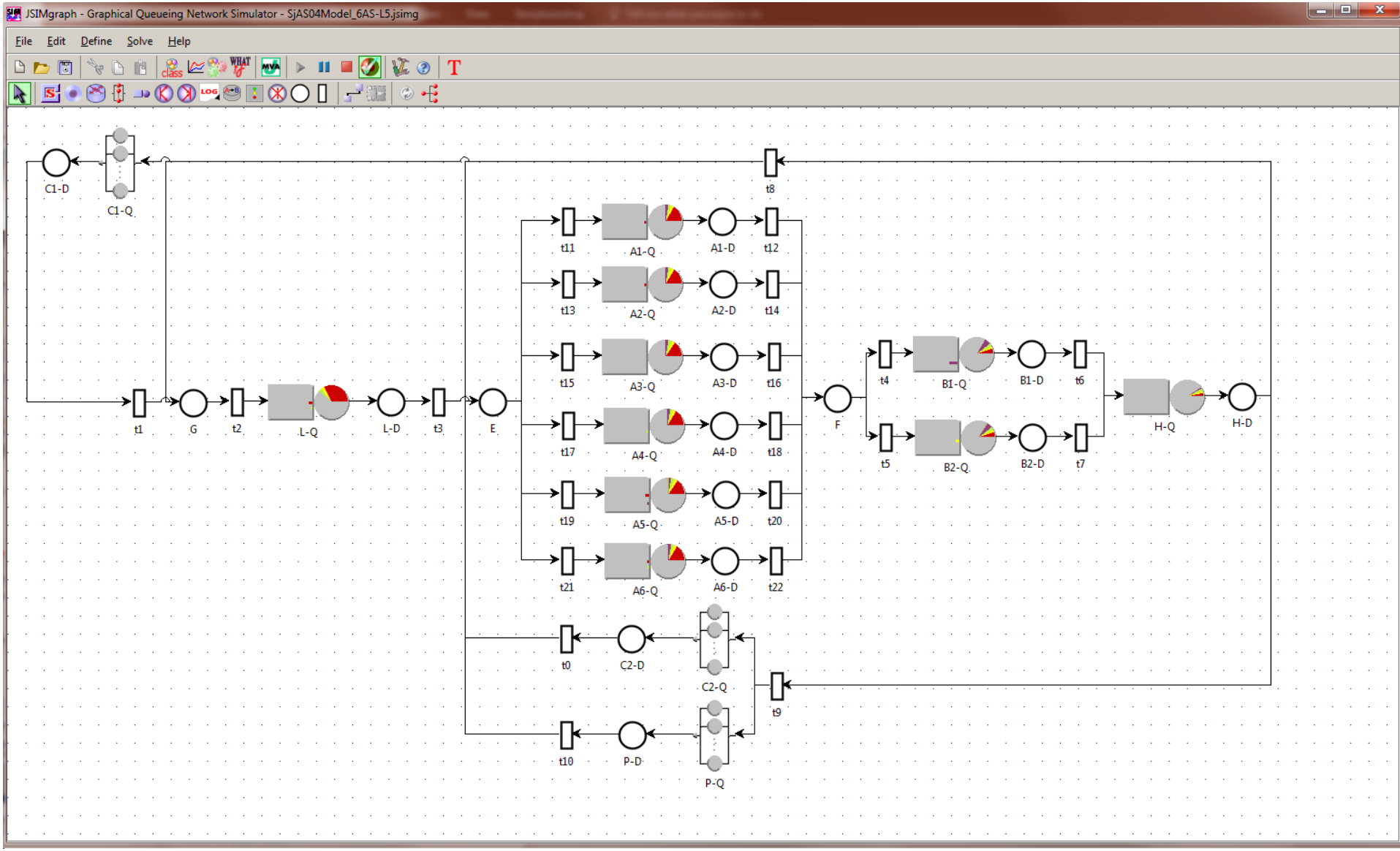
- JMT is for PE practice, teaching, and research
- Several university courses worldwide (tell us!)
- Supporting materials available on website



JMT Start Screen



JSIMgraph: QN & PN simulation



Templates

Add/See Available Templates

Site: JMT Off

Name

- JMT Official Site
  - Intranet
    - Three Tier Intrane
  - Parallel computatior
    - Parallel
  - System Componen.
    - Ceph
    - RAID

This template generates an intranet wi
(for the data management).
The number of service stations (that r
If the classes of workload have alread
A limit to the number of customers in
may change all the parameters as need

Add/Use templates

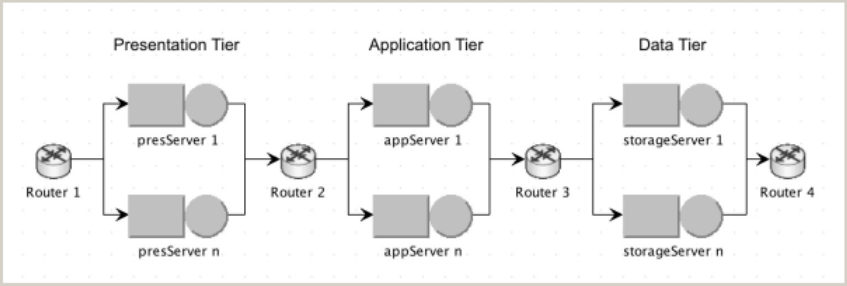
Add/See a

Template	Version
Parallel	1.06
Three Tier Intranet	1.09

3tier_Intranet

The routing probabilities are the same for all stations of each tier

Model layout



Presentation tier

Name of the service centers: presServer

Number of service centers: 1

Number of processors per service center: 1

Mean service time per class (same for all stations):

Class1: 1

Application tier

Name of the service centers: appServer

Number of service centers: 1

Number of processors per service center: 1

Mean service time per class (same for all stations):

Class1: 1

Data tier

Name of the service centers: storageServer

Number of service centers: 1

Number of processors per service center: 1

Mean service time per class (same for all stations):

Class1: 1

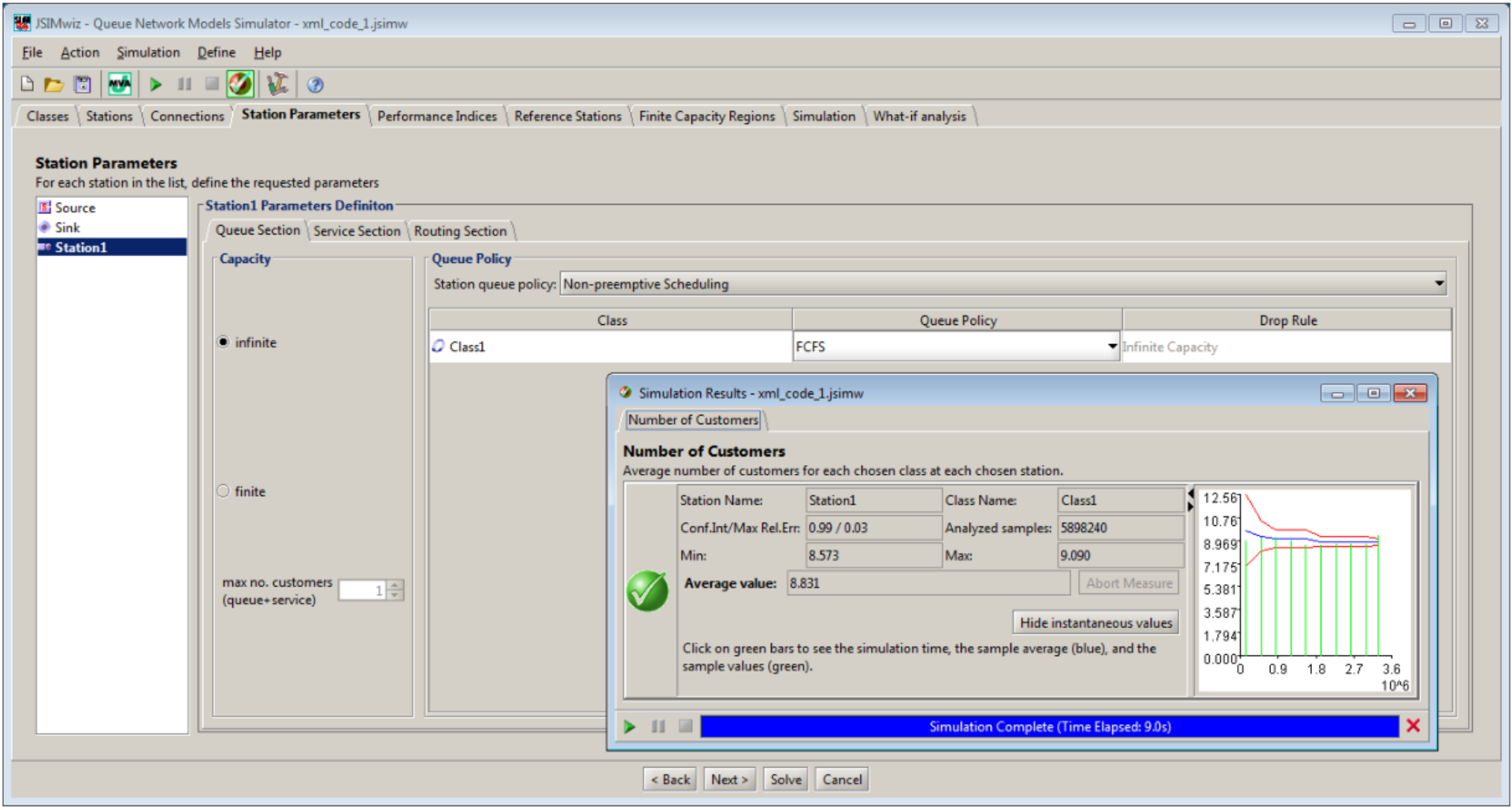
Finite Capacity Region

Max number of customers in the Intranet (-1 = infinite)

-1

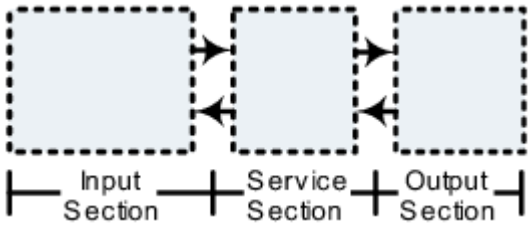
Insert

JSIMwiz: wizard-based user interface

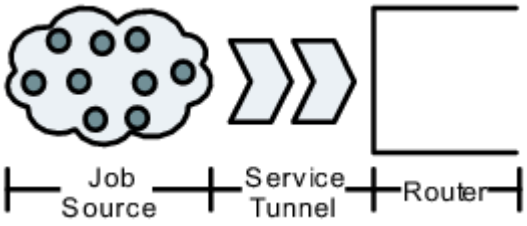


JSIMengine: discrete-event simulator

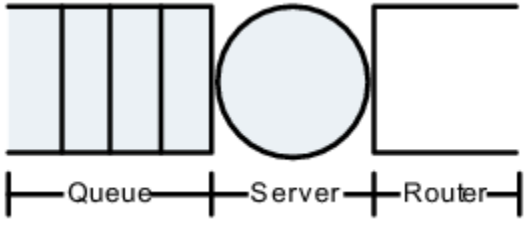
- Simulation components defined by 3 sections



component sections

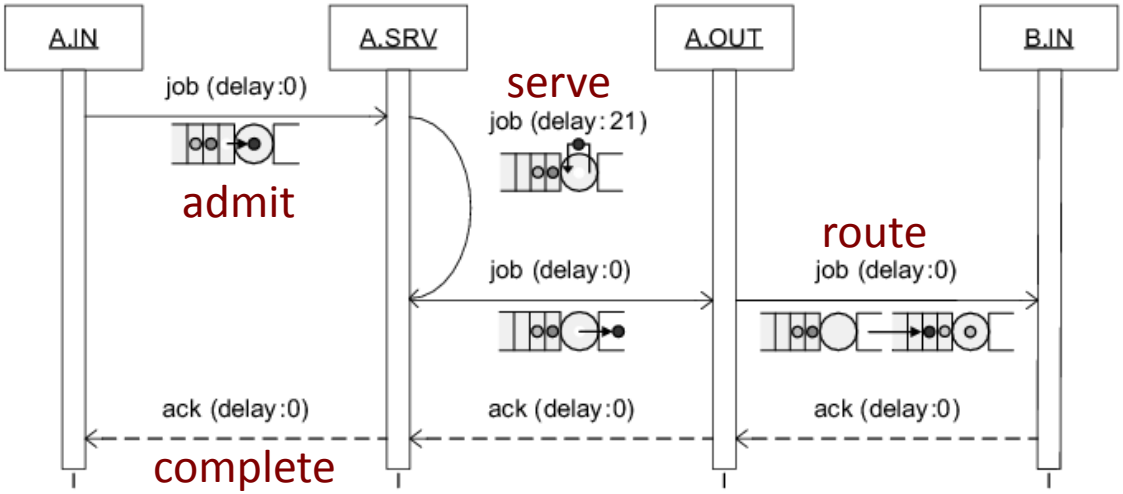


external arrivals (open class)

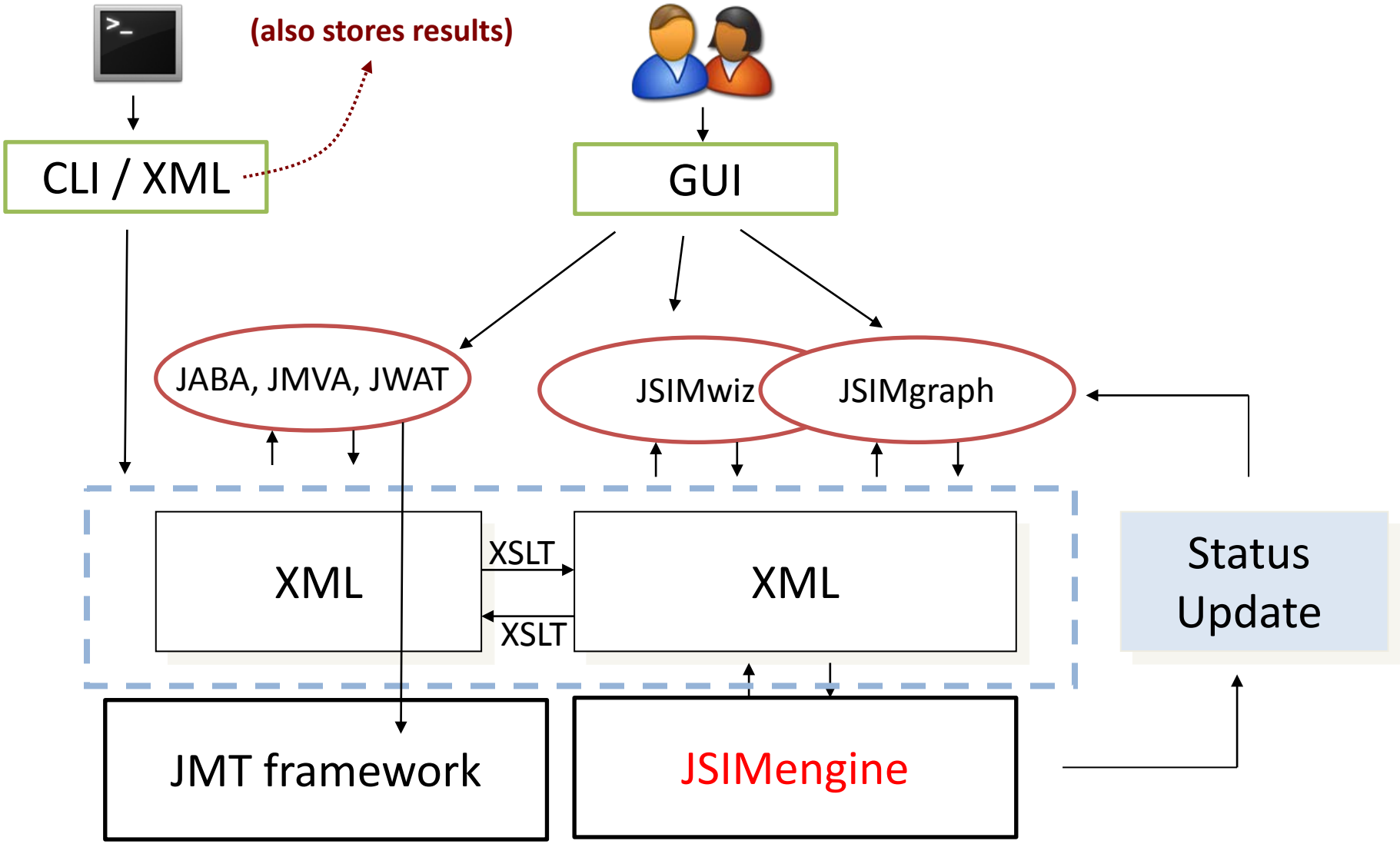


queueing station

- Discrete-event simulation of section messaging



JSIMengine: JMT architecture



JMVA: analytical solver

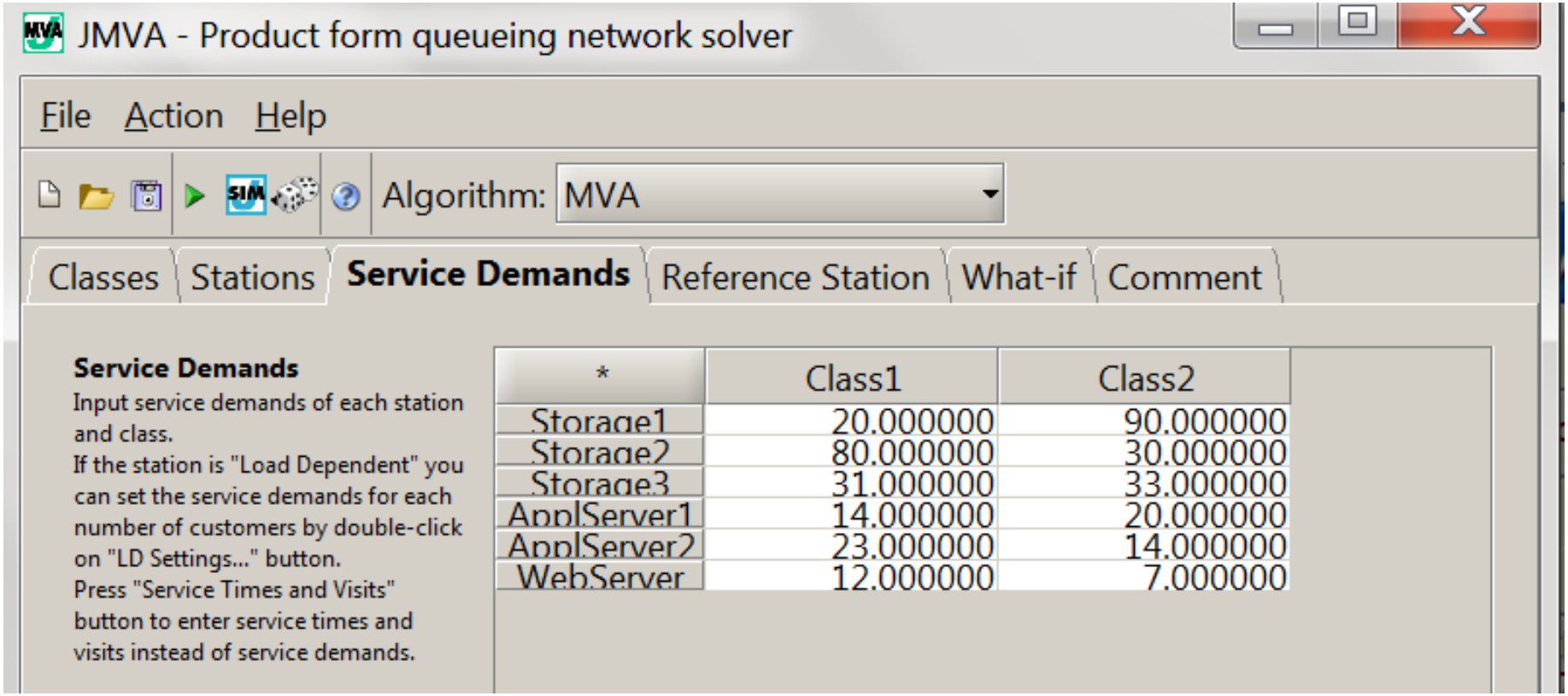
- Analysis of product-form queueing networks
- Several exact and approximate algorithms
  - *Exact MVA*
    - Reiser & Lavenberg $O(N^R)$
    - Load-dependent $O(N^{2R})$
  - *Approximate MVA* $O(1)$
    - Chow
    - Bard-Schweitzer
    - AQL
    - Linearizer
    - De Souza-Muntz
  - *Normalizing constant*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

N: jobs

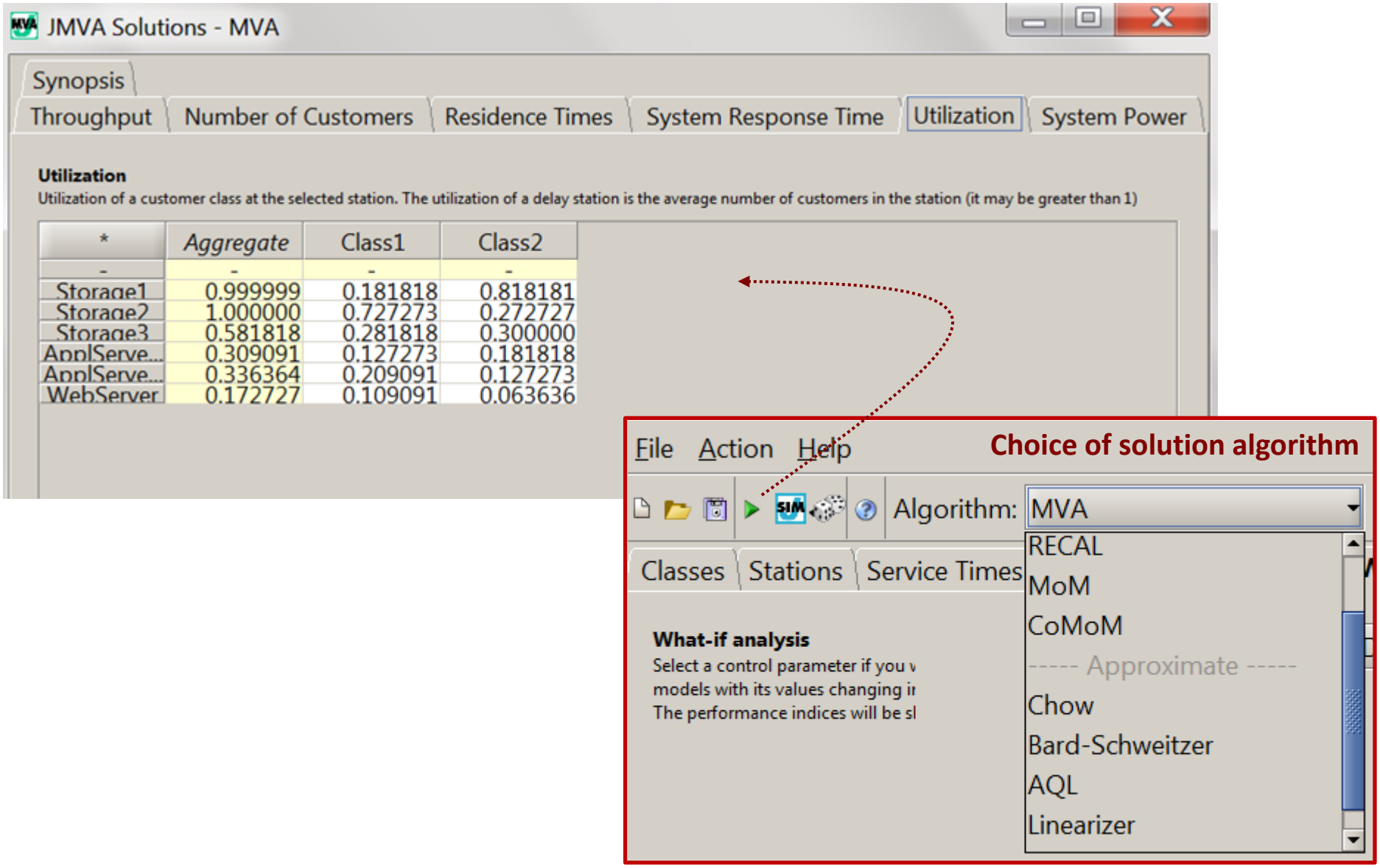
M: stations

R: classes

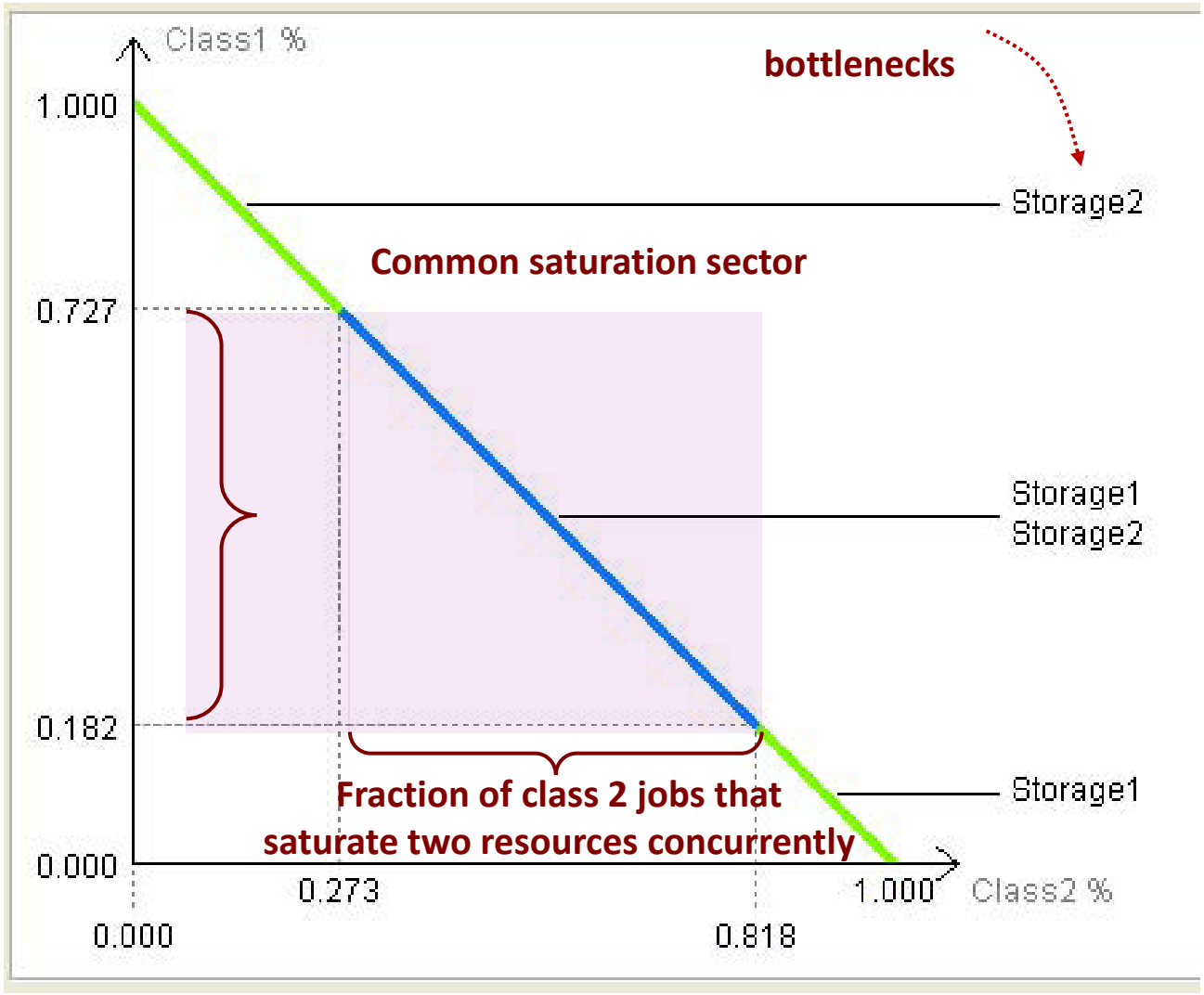
JMVA: model parameterization



JMVA: solutions

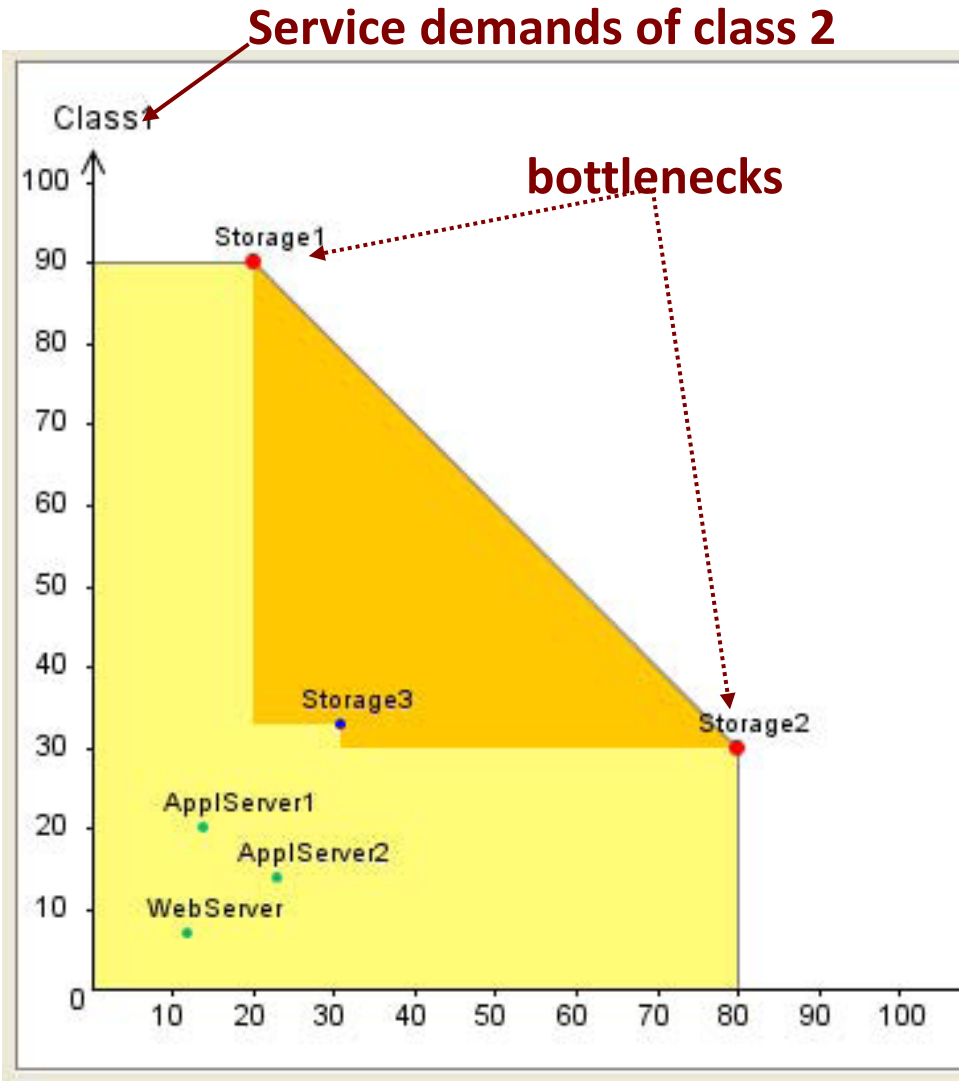


JABA: bottleneck identification

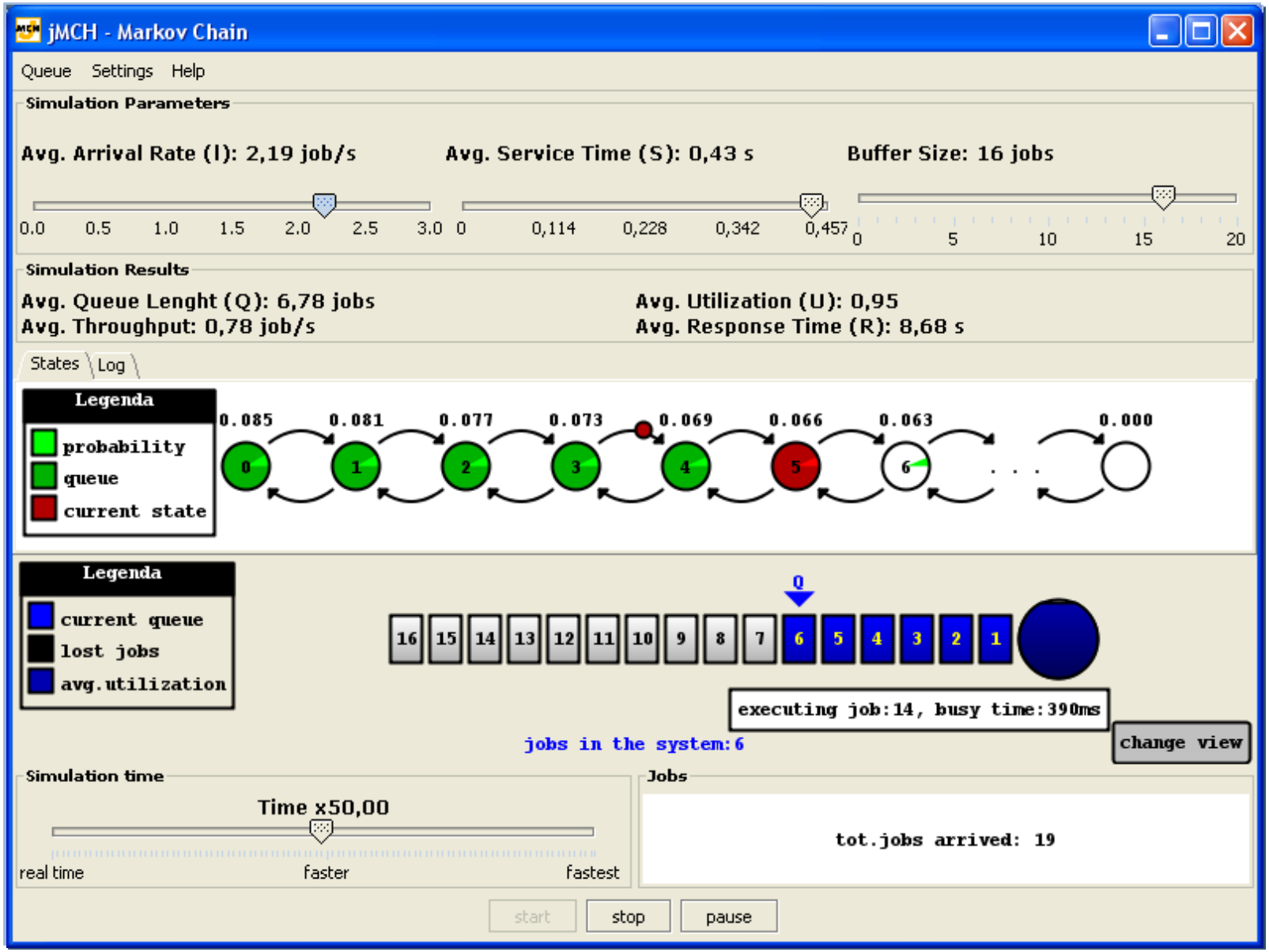


JABA: bottleneck identification

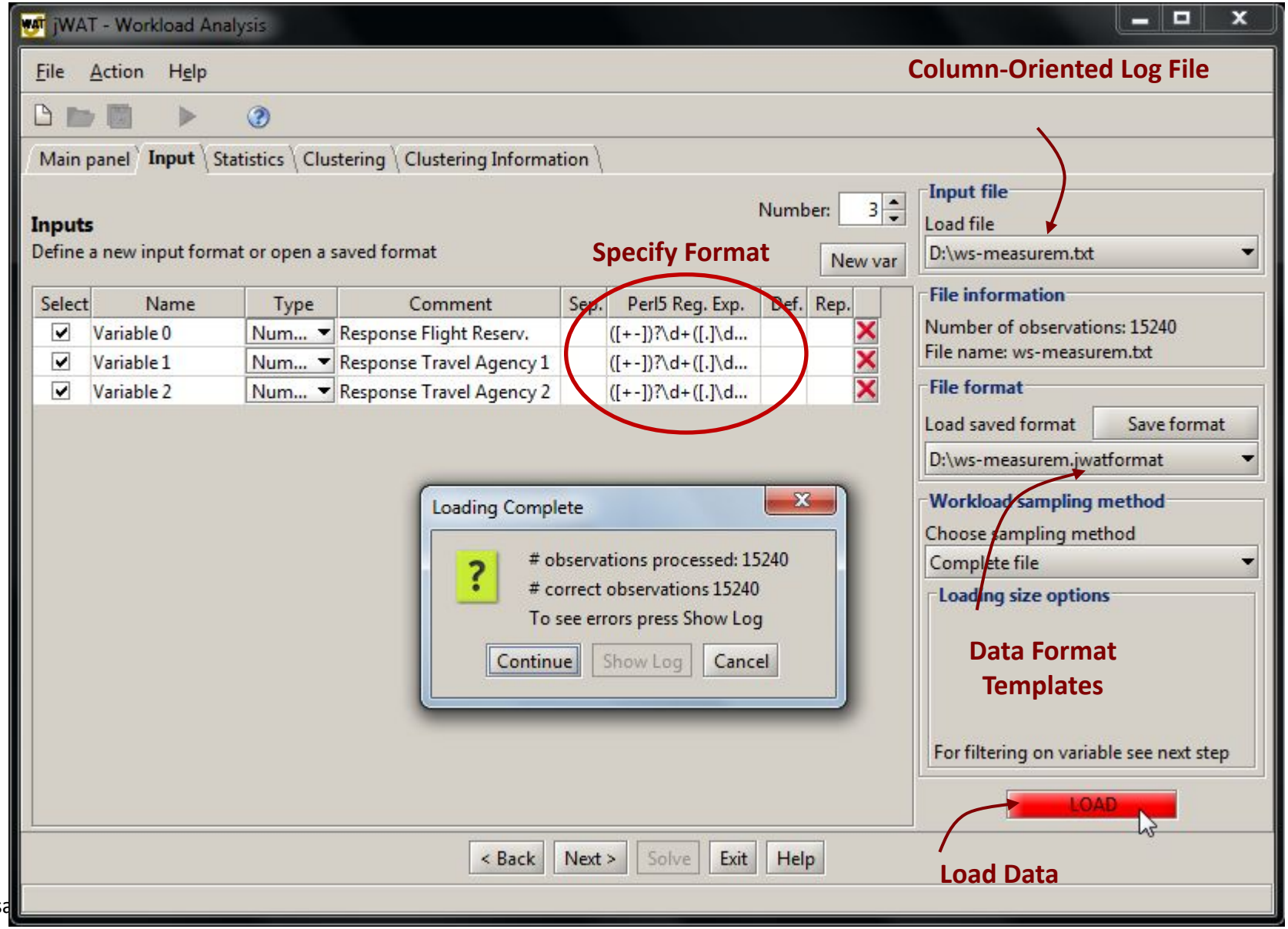
- 3-class model



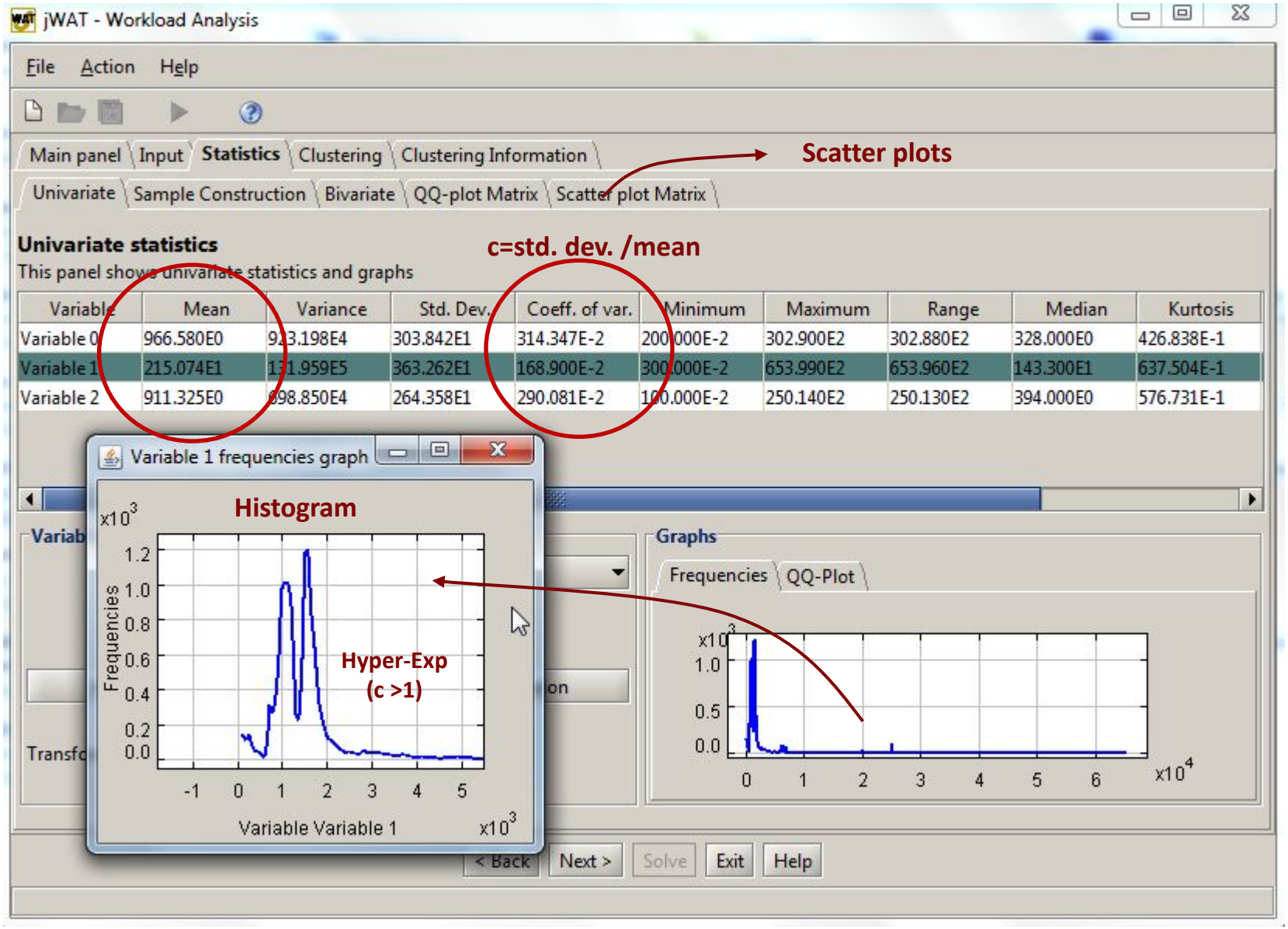
JMCH: Markov chain animation



JWAT: workload characterization



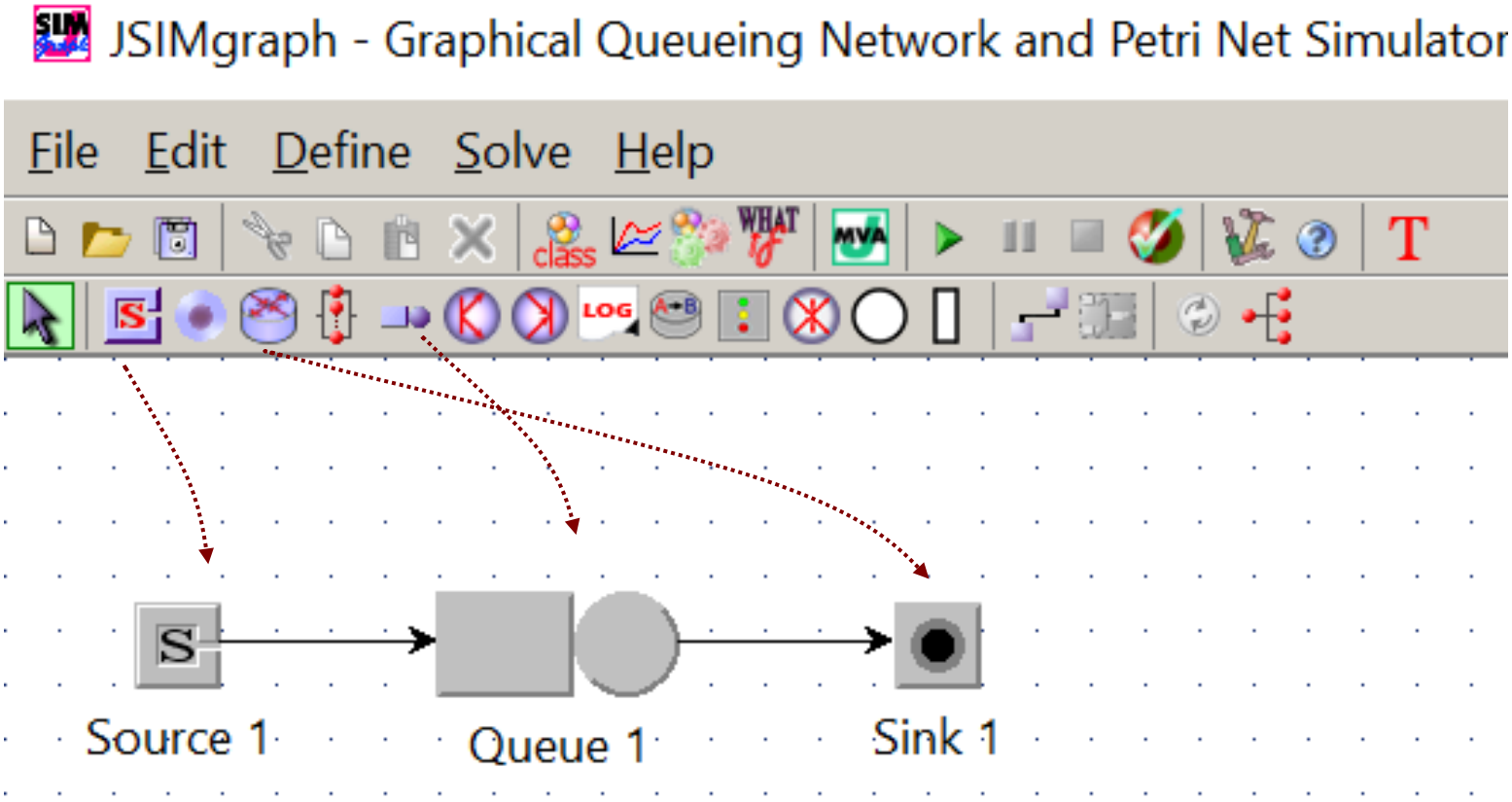
JWAT: workload characterization



Activity 1: getting started

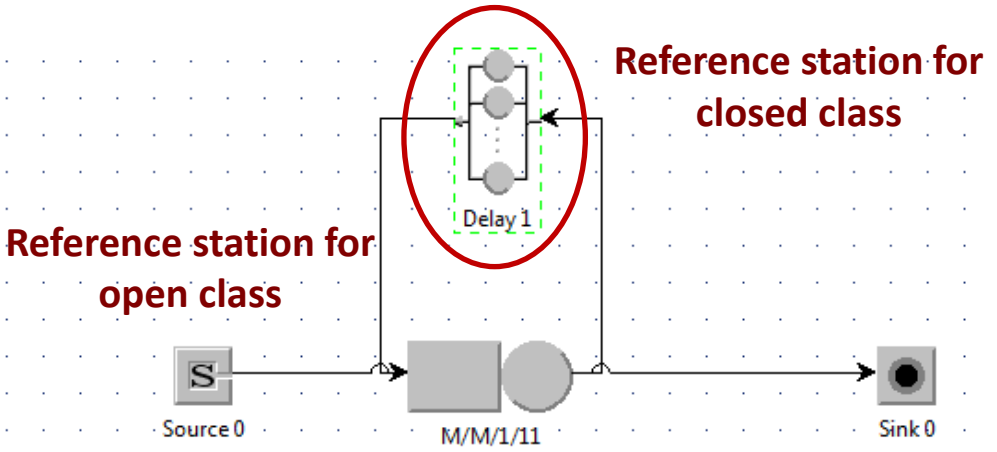
Hands-on activity: M/M/1

- Arrival rate: λ=0.5 job/s (Exponential)
- Service rate: μ=1.00 job/s (Exponential)
- Goal: verify μ=> λ property

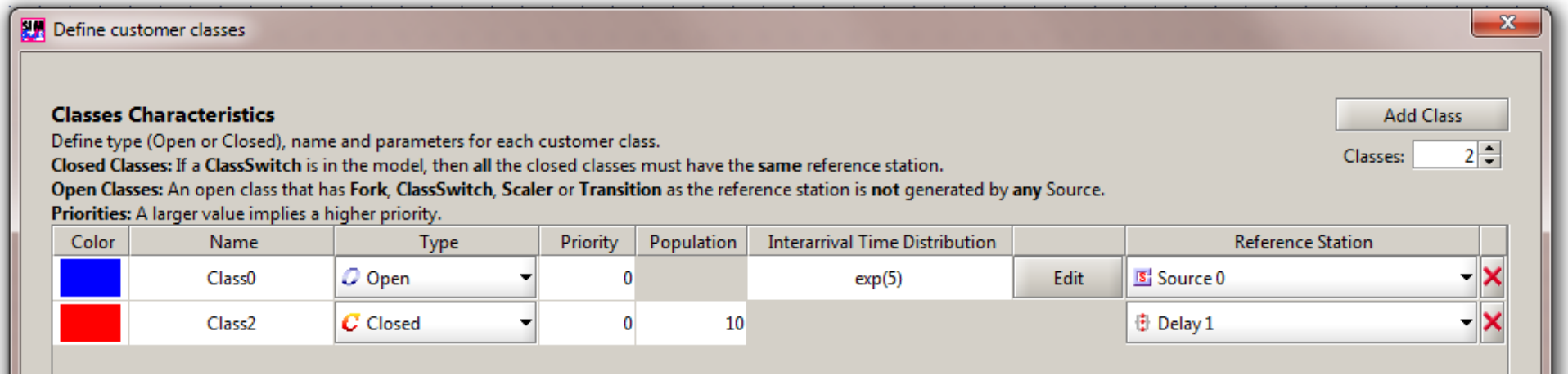


Class definition

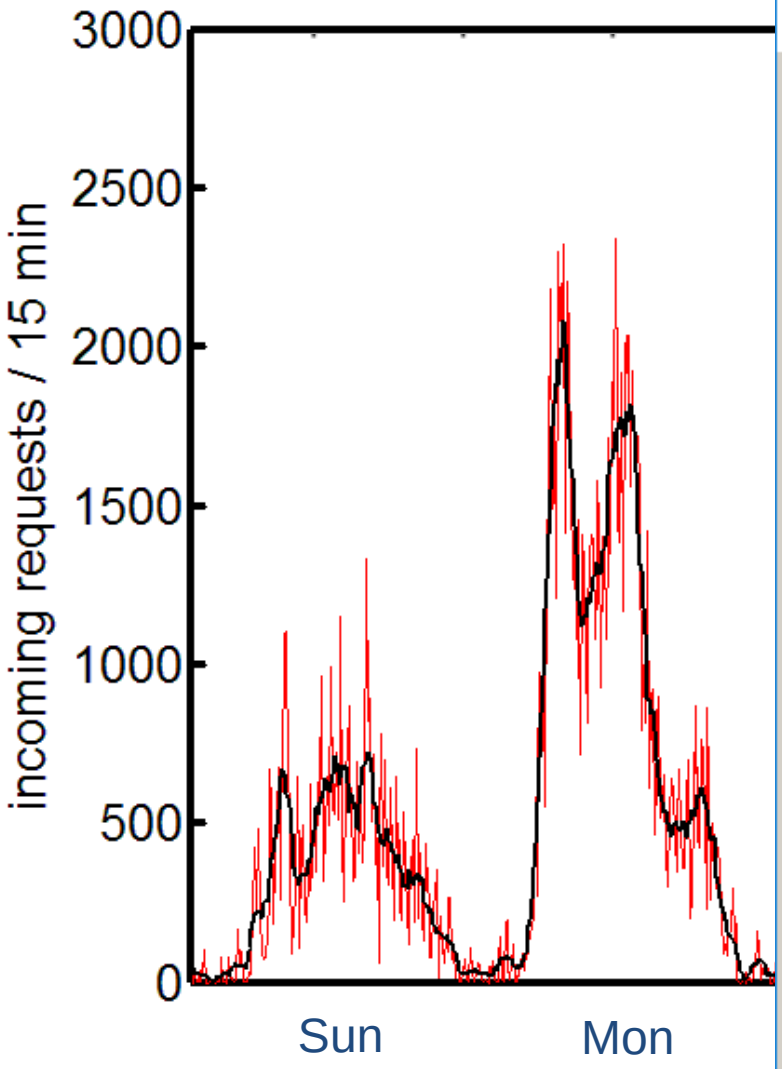
- Open, closed, and mixed workloads
- Priorities and reference stations

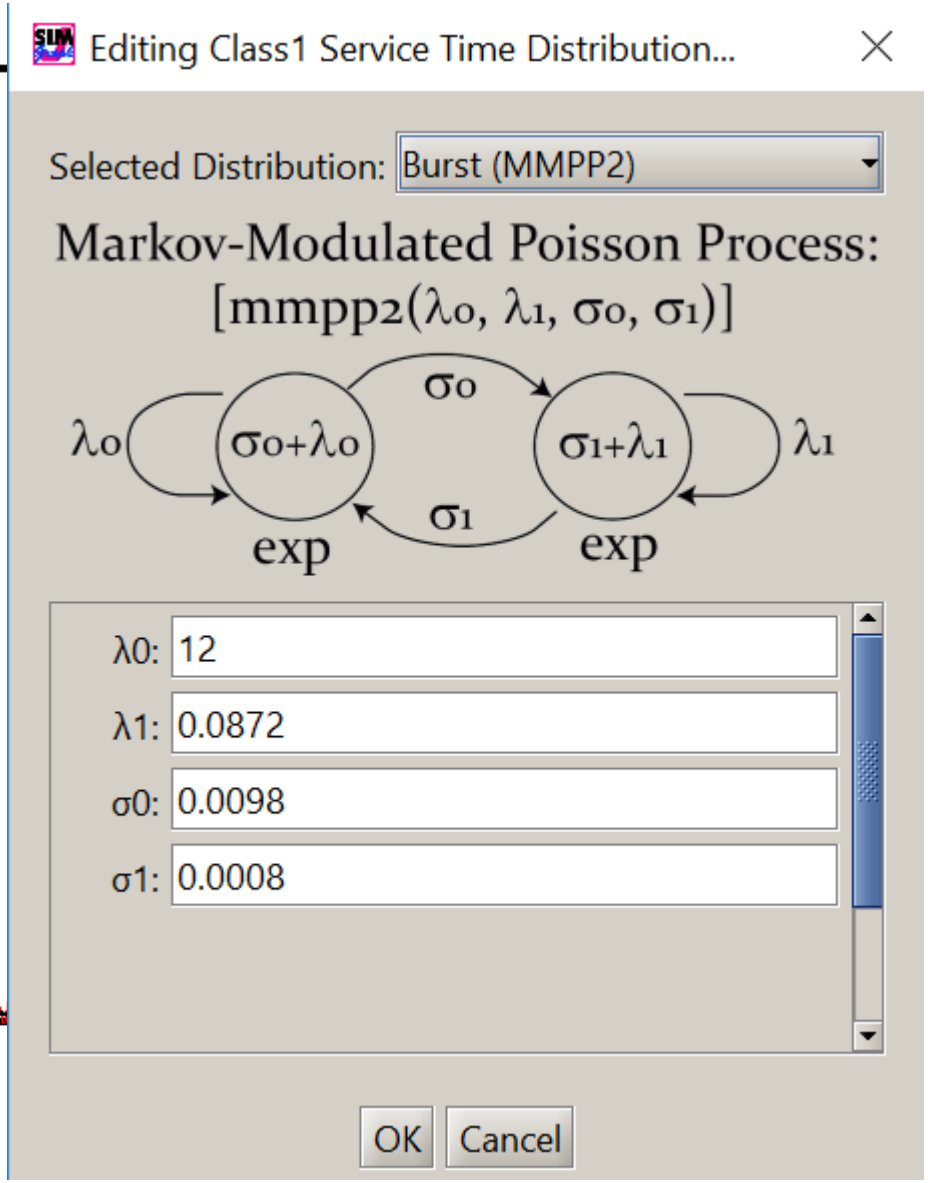


- Reference stations used for system metrics
- Visits at node = TPUT*node*/TPUT*ref*
- *ref* is the reference station



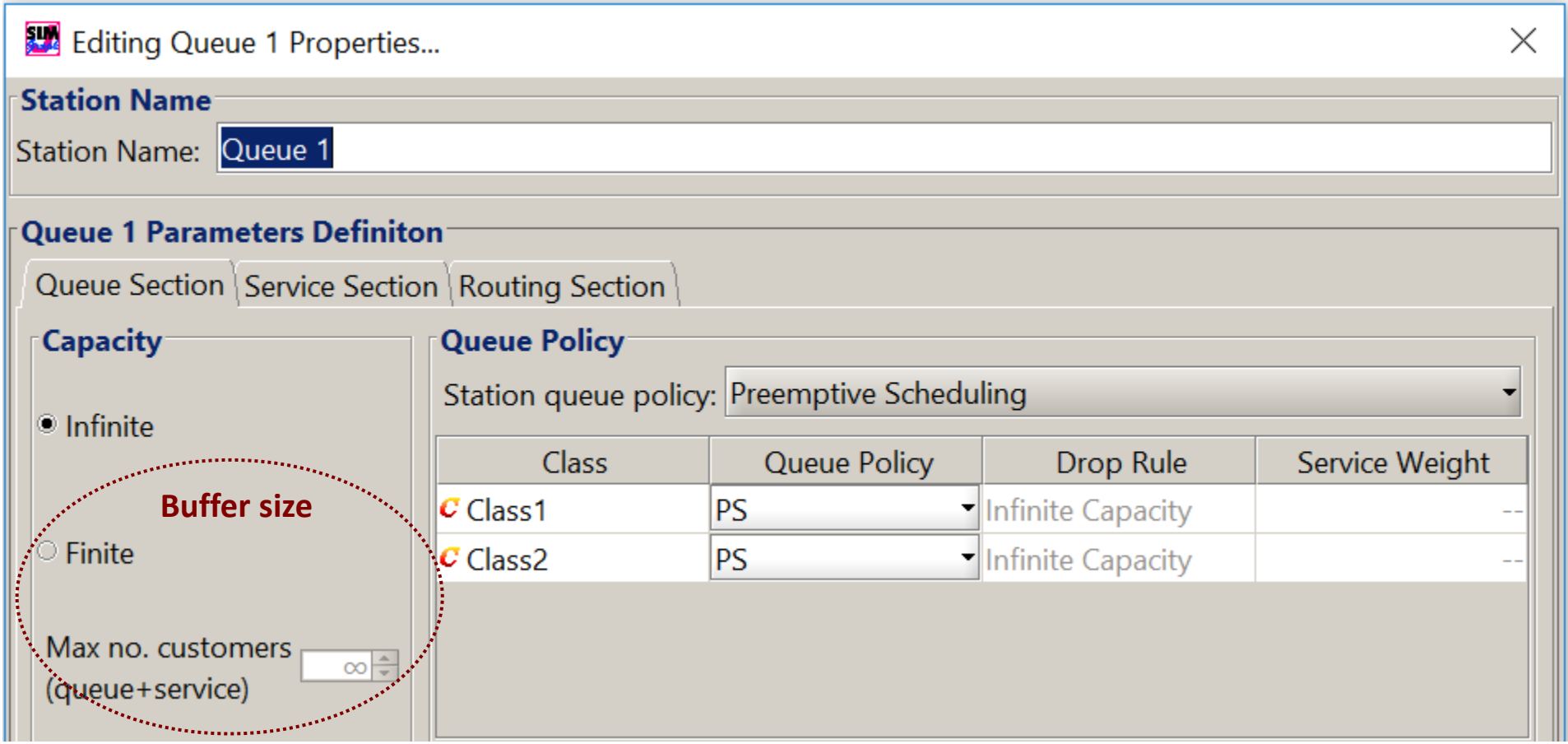
Arrival distribution



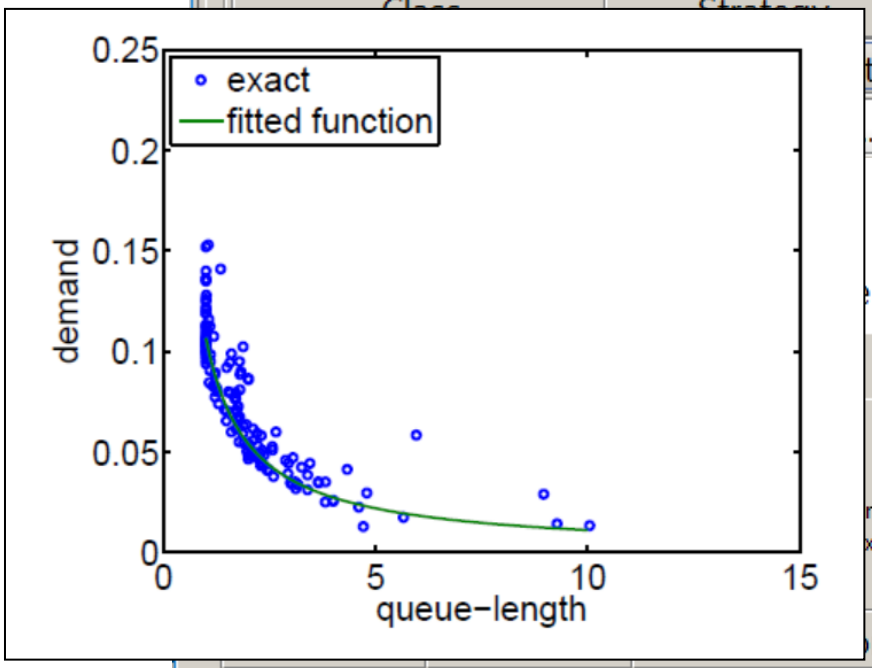


Queue section

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



Service section



Queue Section | Service Section | Routing Section

Number of Servers

Number: 1

number of servers

exponential rate = 0.5

Service Time Distributions

Service Time Distribution	
Load Dependent Service Stra...	Edit
exp(0.5)	Edit

Strategy...

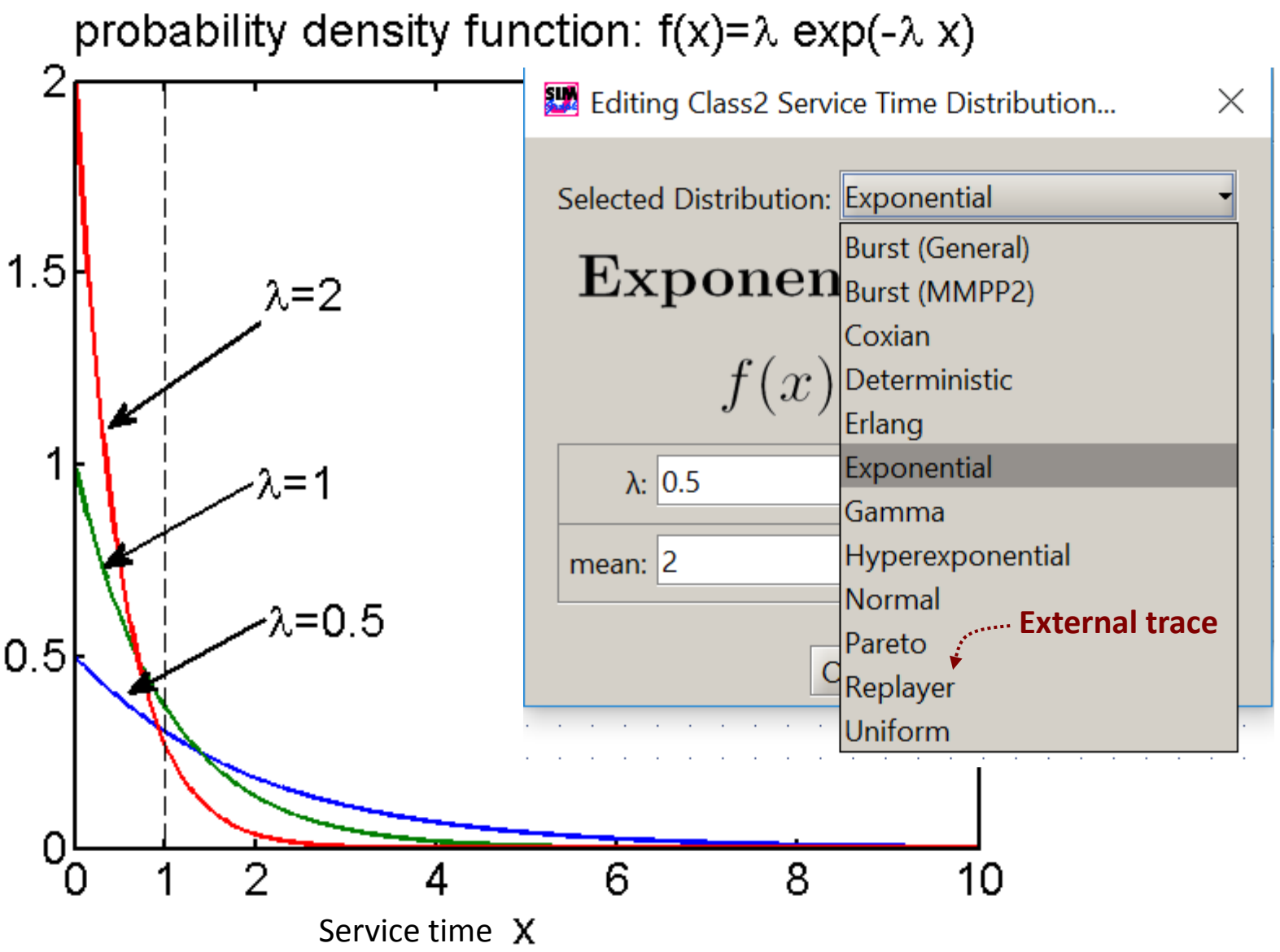
Add Range

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

Ranges: 2

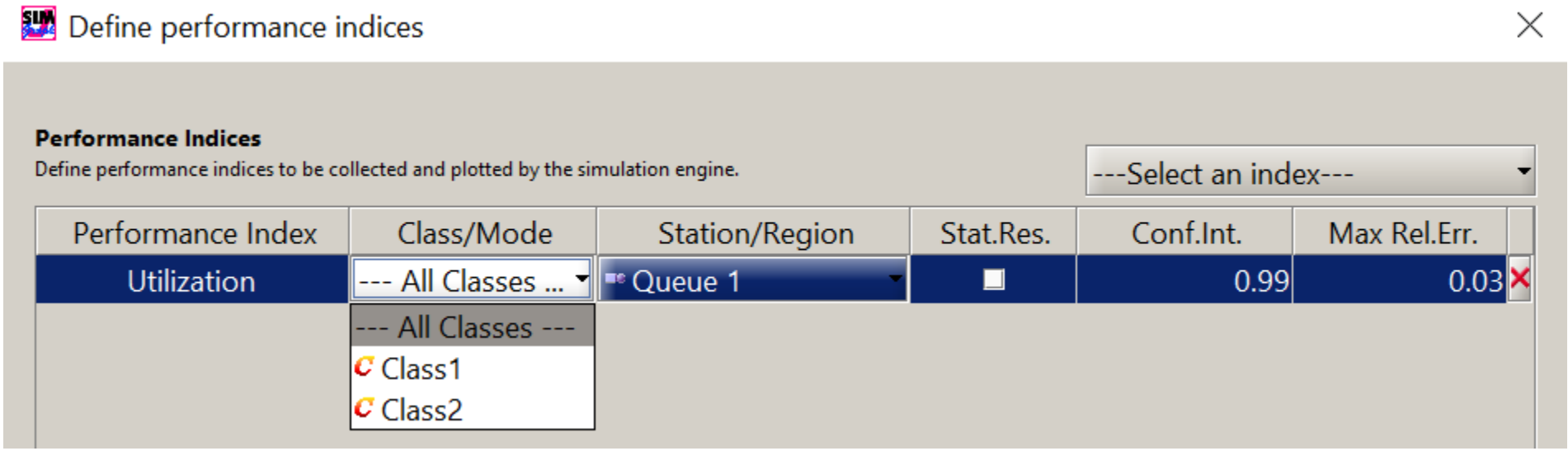
			Mean	C
1	10	Exponential	2	
11	∞	Exponential	1	

Service time distribution

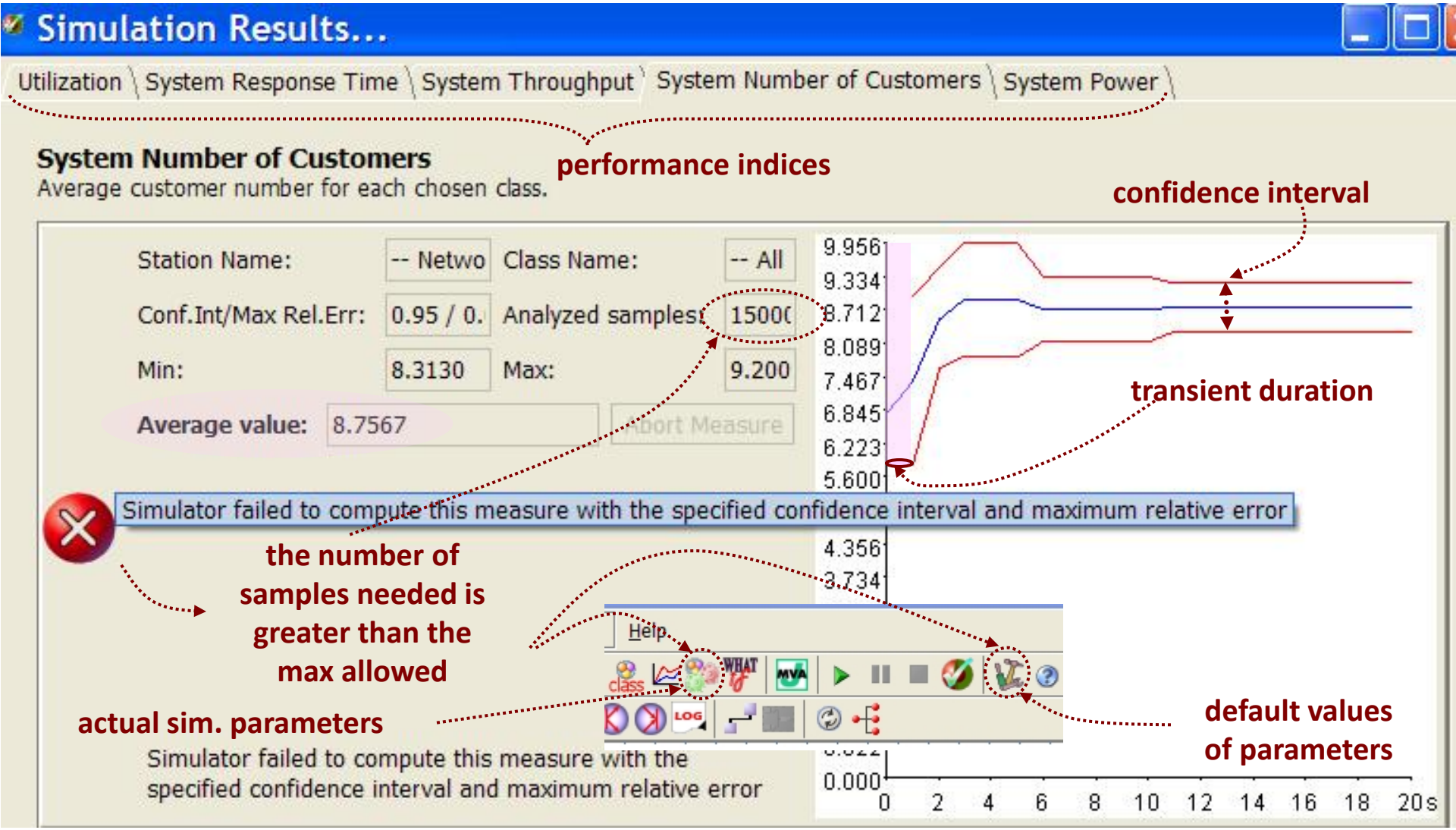


Perf. Indices

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



Sim. Results


Simulation Results...
Utilization
System Response Time
System Throughput
System Number of Customers
System Power
System Number of Customers
Average customer number for each chosen class.
performance indices
confidence interval
Station Name:
-- Netwo
Class Name:
-- All
Conf.Int/Max Rel.Err:
0.95 / 0.
Analyzed samples:
1500(
Min:
8.3130
Max:
9.200
Average value:
8.7567
Abort Measure
9.956
9.334
8.712
8.089
7.467
6.845
6.223
5.600
4.356
3.734
0.000
0
2
4
6
8
10
12
14
16
18
20s
transient duration
Simulator failed to compute this measure with the specified confidence interval and maximum relative error
the number of samples needed is greater than the max allowed
Help
actual sim. parameters
default values of parameters
Simulator failed to compute this measure with the specified confidence interval and maximum relative error

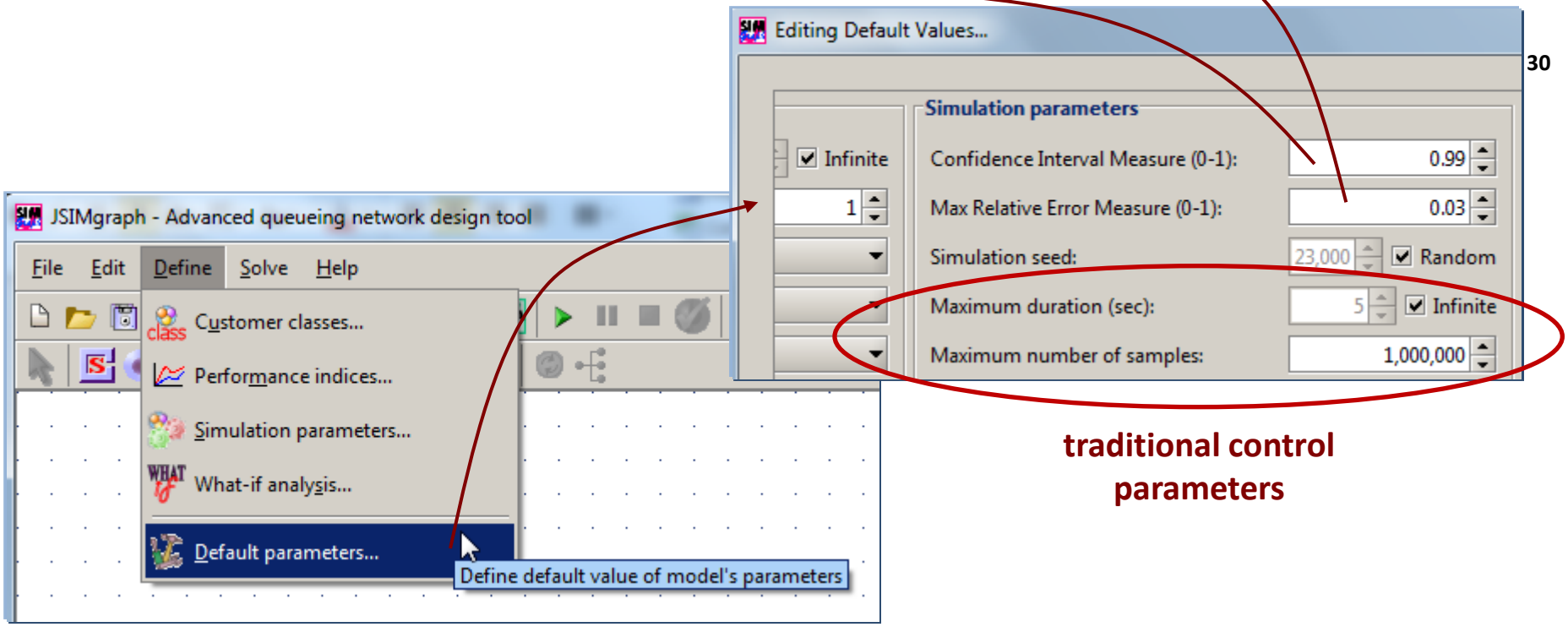
Statistical analysis

- Automated (overridable) simulation stop

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

maximum relative error

confidence level

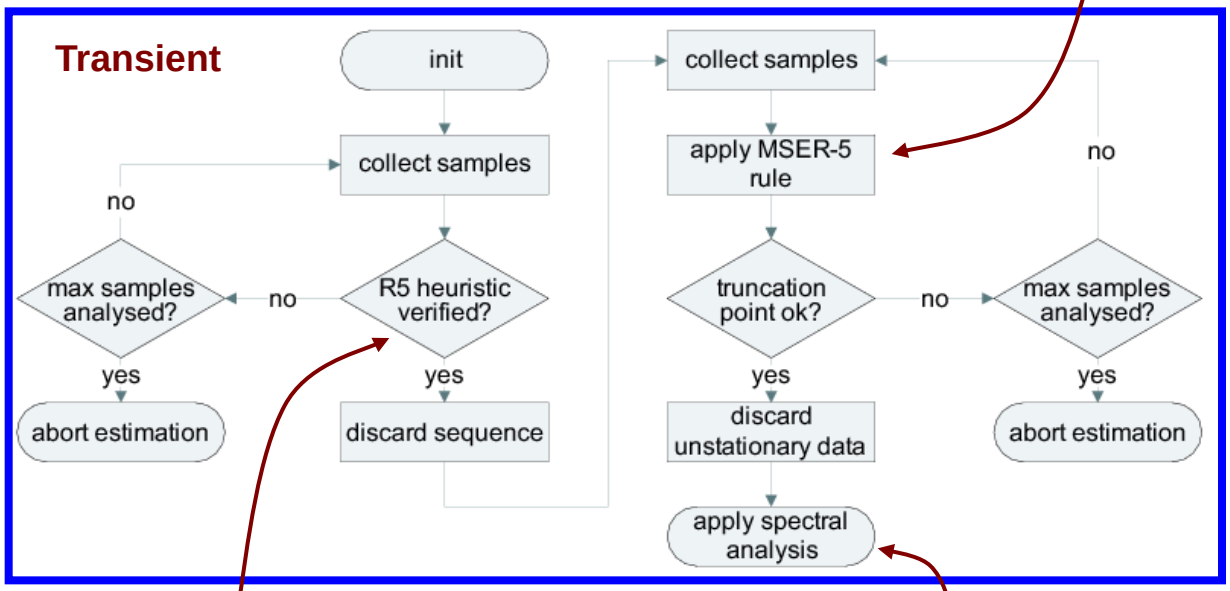


traditional control parameters

Transient filtering

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

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



[Pawlikowski, *CSUR*, 1990]

(Steady State)

[Heidelberger&Welch, *CACM*, 1981]

Detailed statistical results

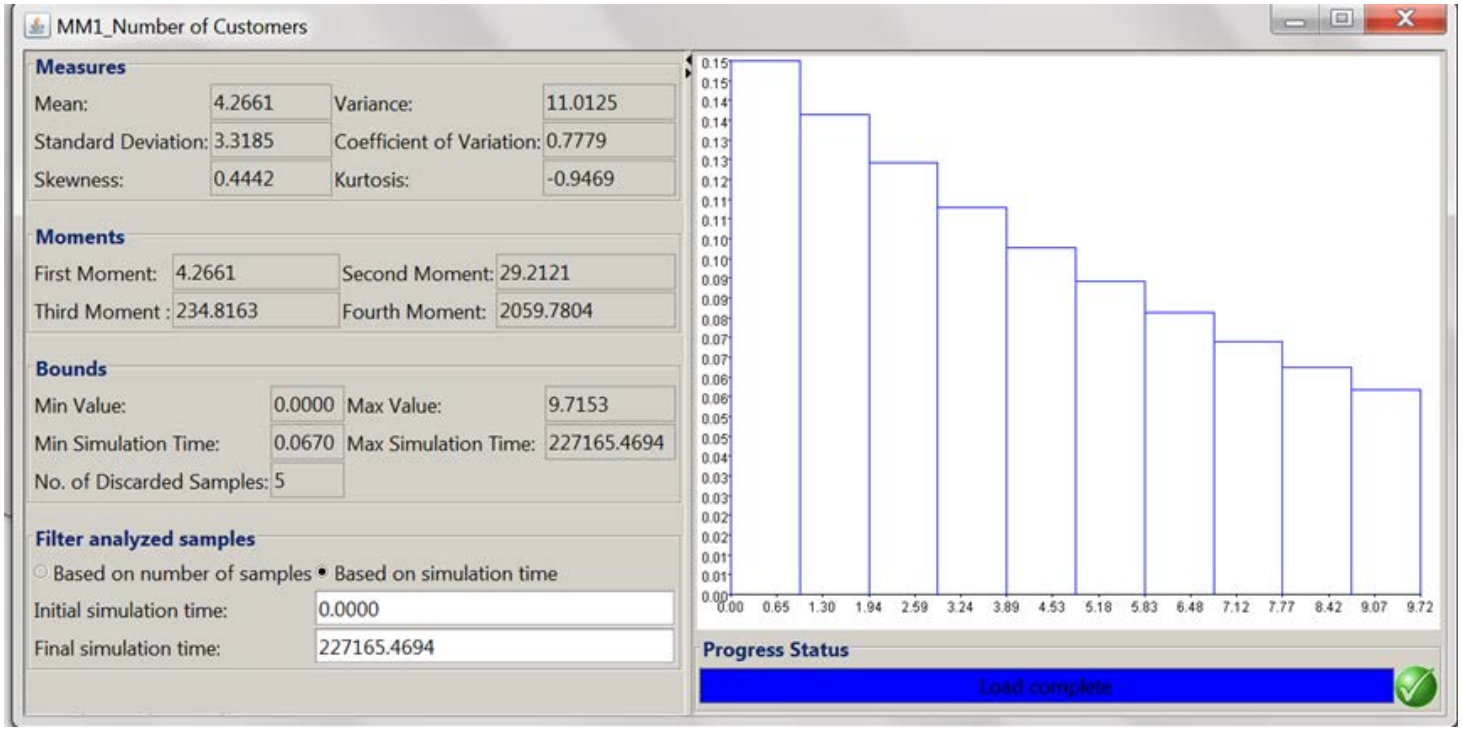
- Response time percentiles, buffer overflow probability, departure process moments, …

Performance Indices

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

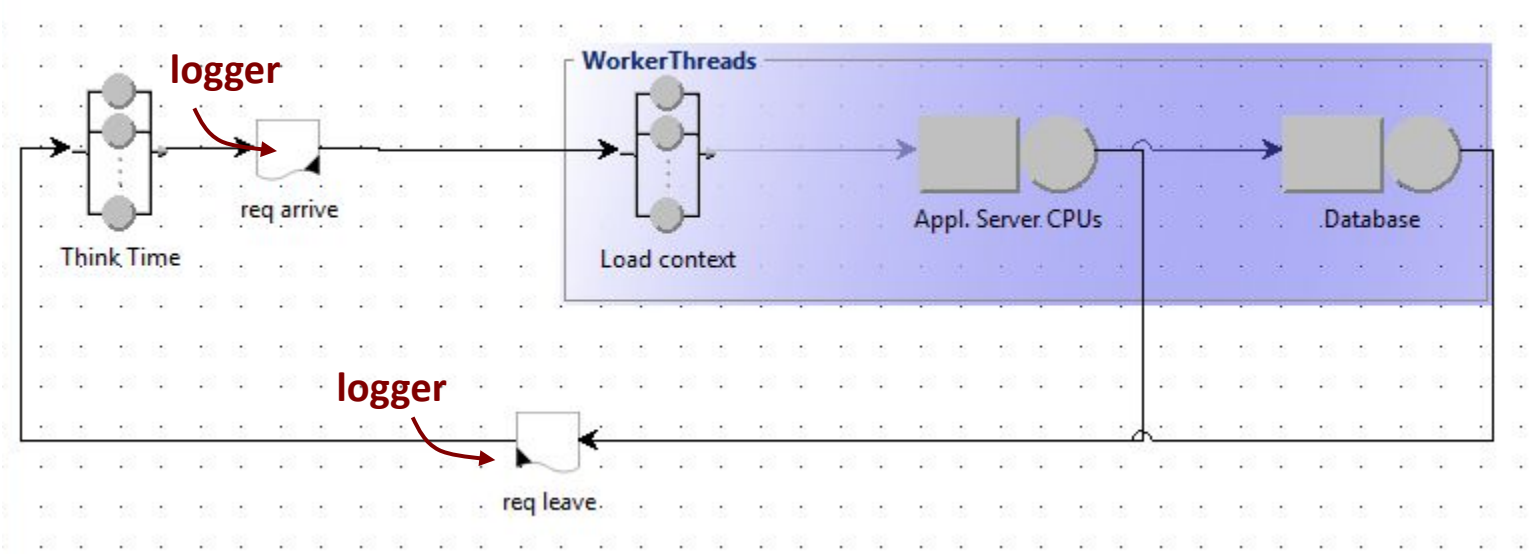
---Select an index---

Performance Index	Class/Mode	Station/Region	Stat.Res.	Conf.Int.	Max Rel.Err.
Response Time	--- All Classes ...	Queue 1	☑	0.99	0.03



Loggers

- Simulation events can be traced in CSV files



global.csv

job id (same throughout simulation)

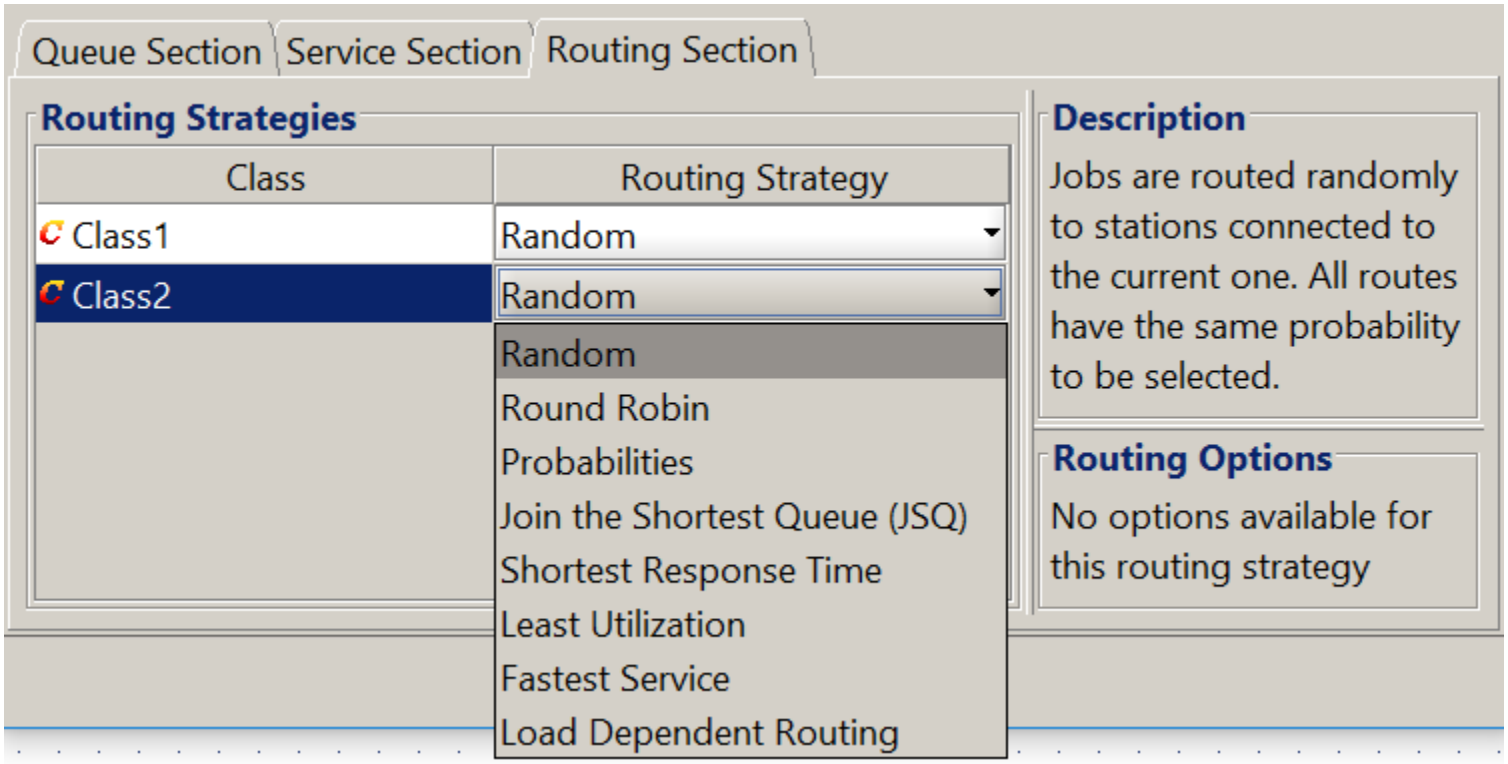
```
LOGGERNAME;TIMESTAMP;JOB_ID;CLASS_ID;INTERARRIVAL_SAMECLASS;INTERARRIVAL_ANYCLASS;SIMUL_START_TIME
req arrive;0.009420010041266342;253625;Transactional workload;;;
req leave;0.0217557654334812;253625;Transactional workload;;;
req arrive;0.031032734664243056;253498;Transactional workload;;;
req leave;0.04915995332909814;253498;Transactional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

job class

Activity 2:
Load balancing

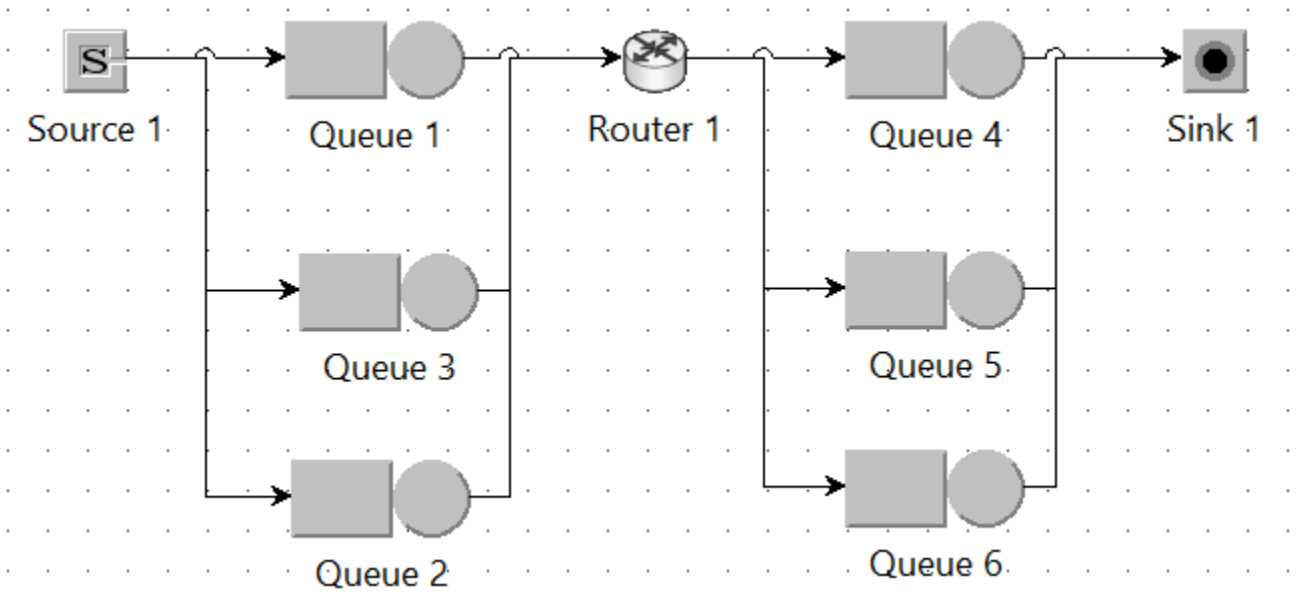
Routing section or ….

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



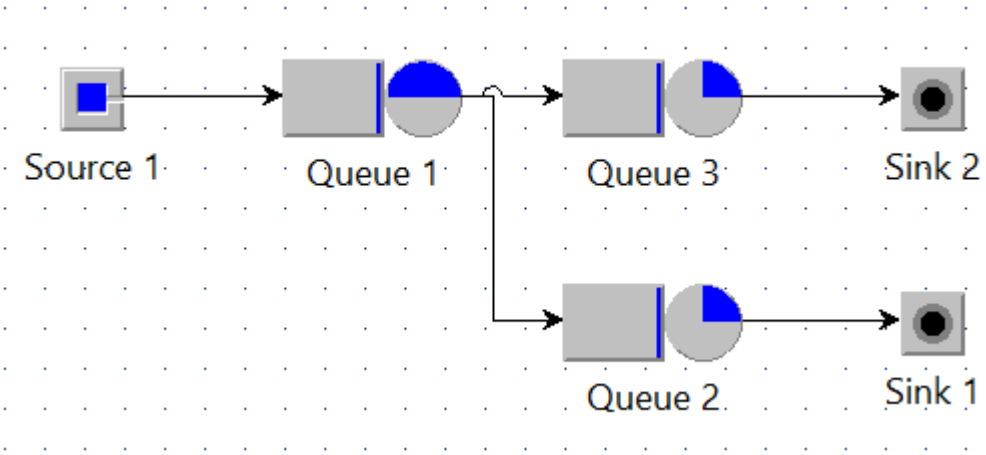
… router node

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

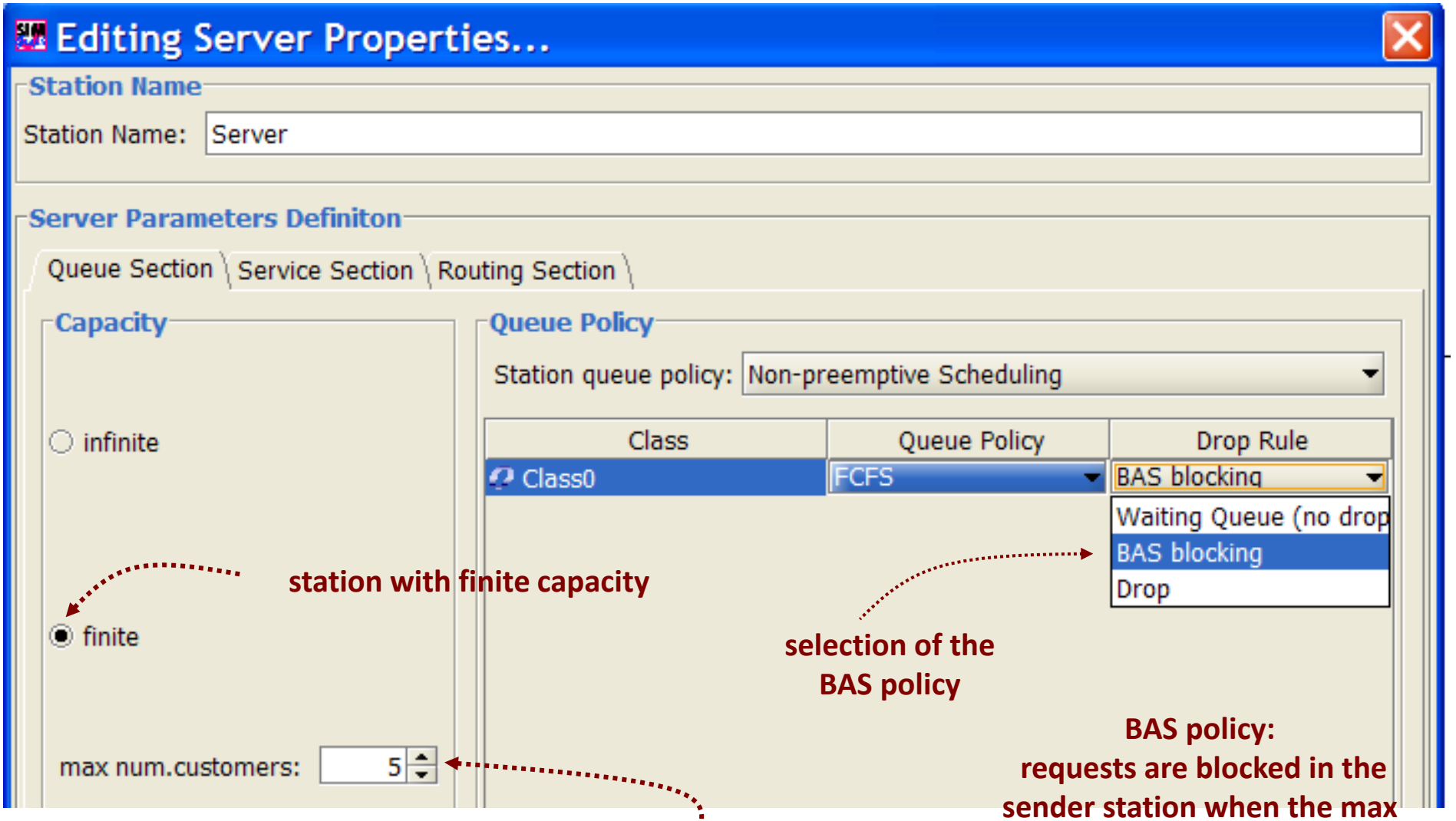


Hands-on activity: load balancing

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



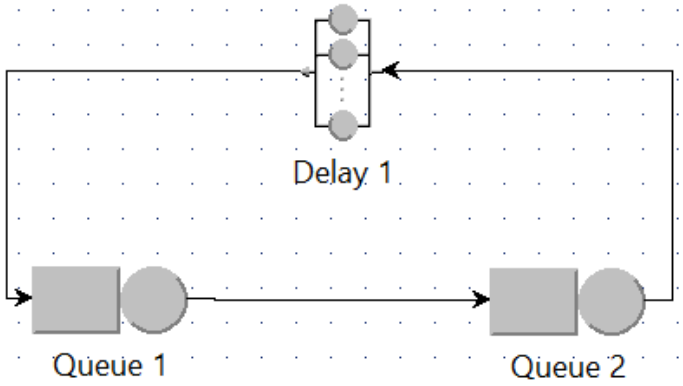
Blocking after service

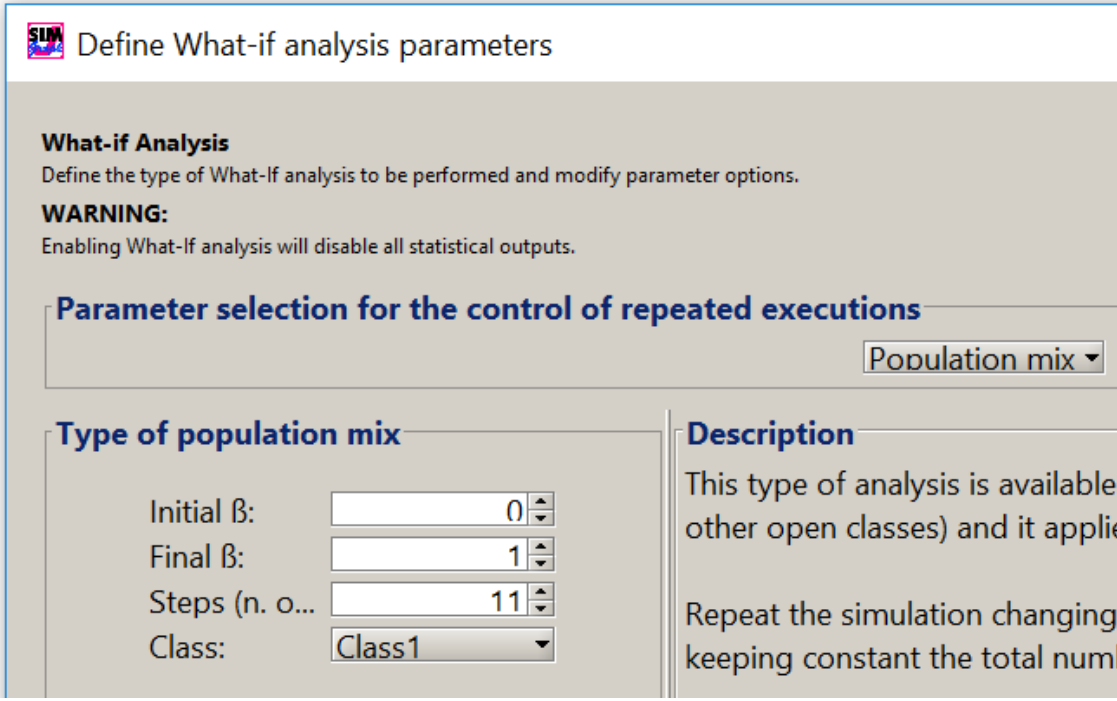


Activity 3:
Parameter sweeping

Hands-on activity: bottleneck switch

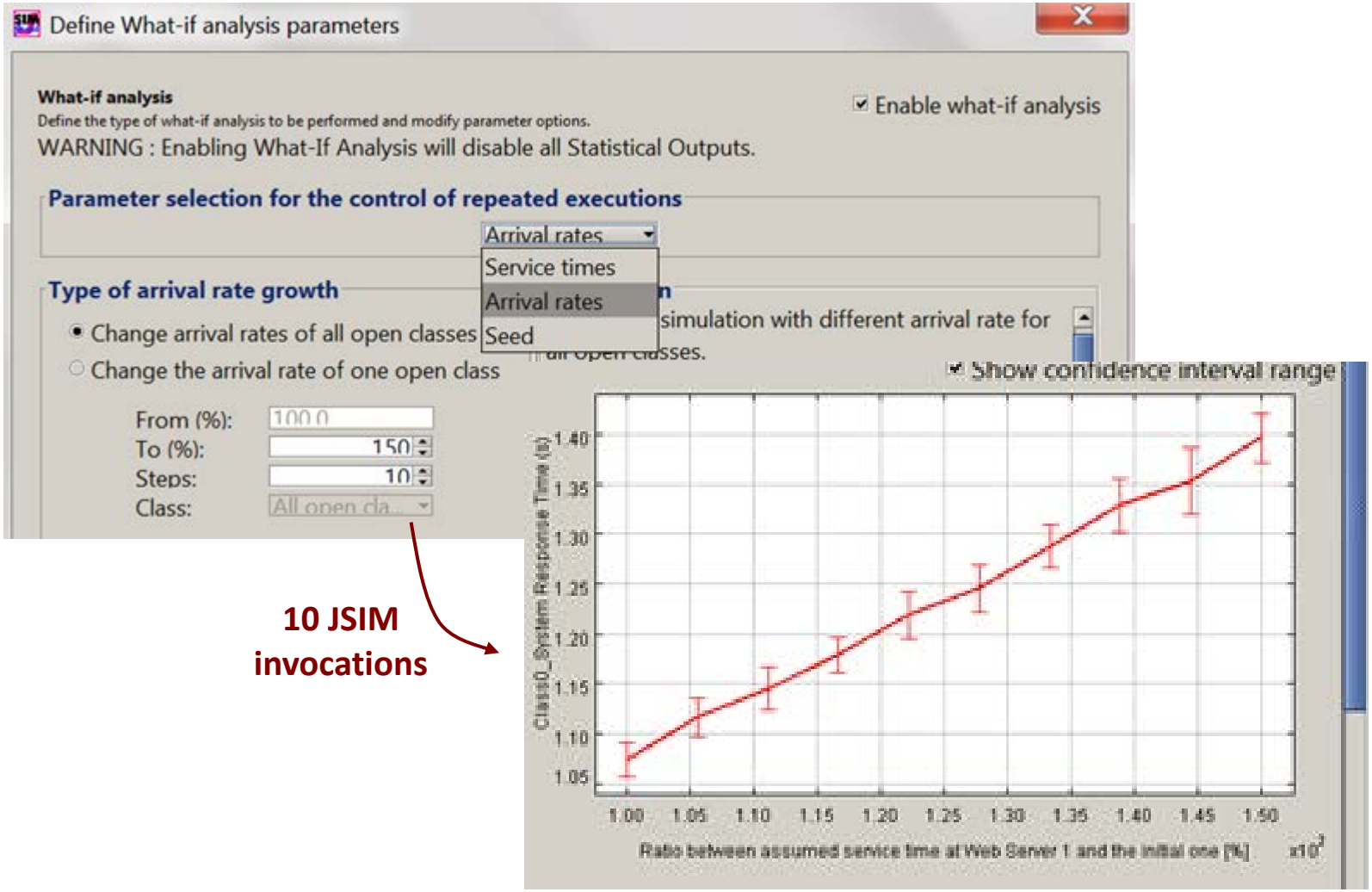
- Analysis of bottleneck switch
- Measure: *Number of Customers*
- Demands: *Queue 1*: 10 , 5; *Queue 2*: 5 , 9





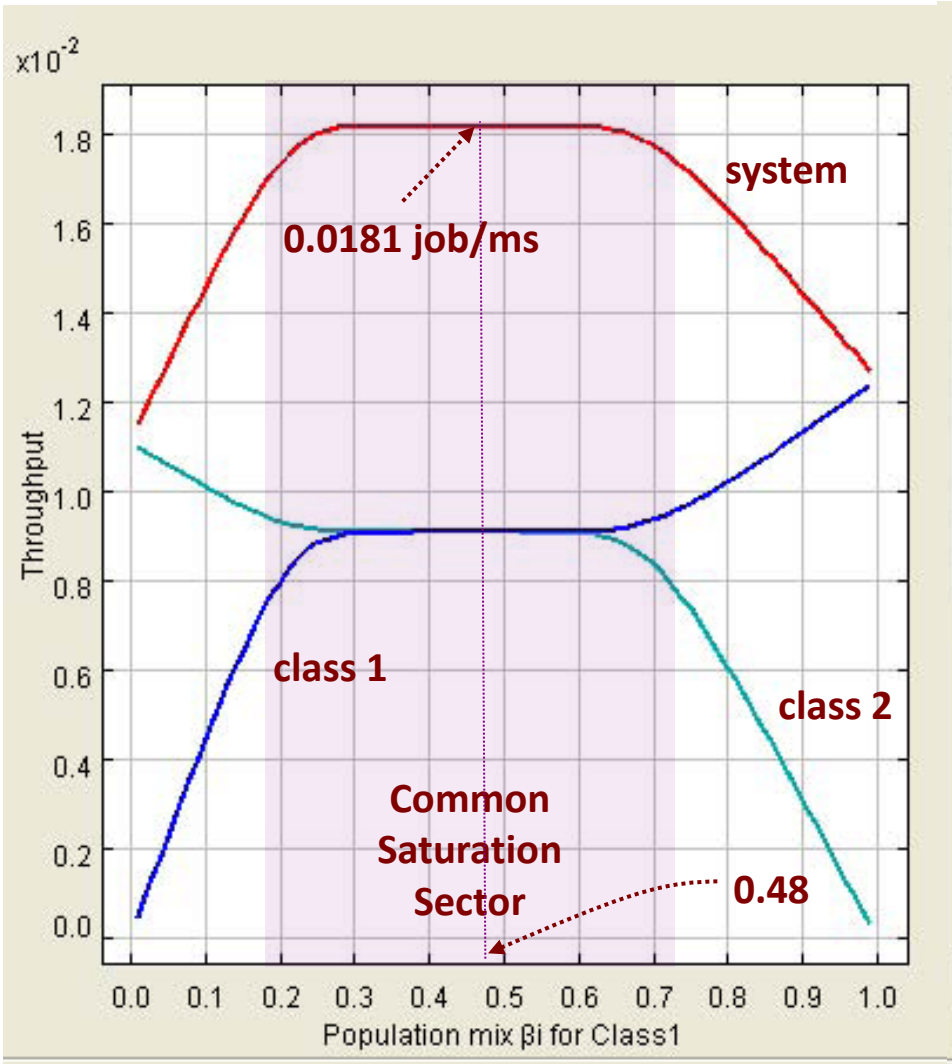
What-If analysis

- Perform repeated executions automatically

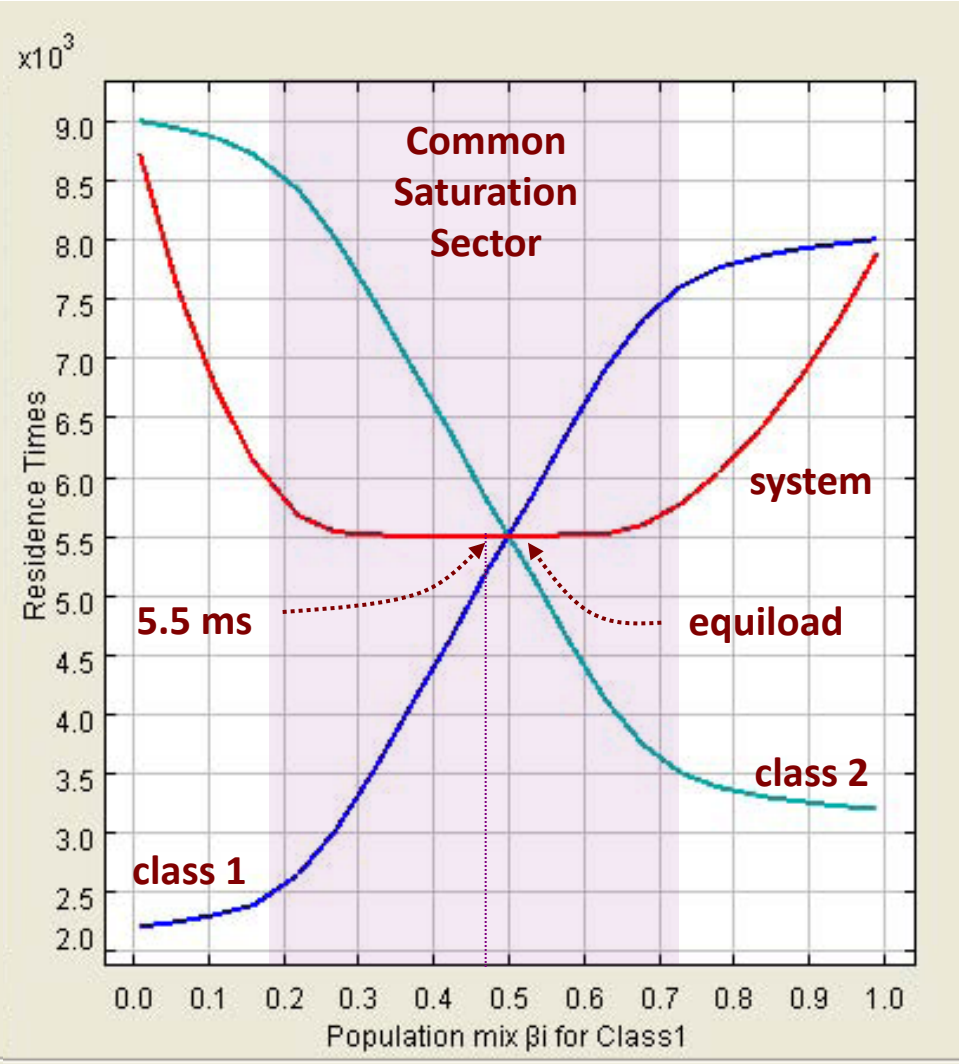


10 JSIM invocations

JMVA: What-If



Throughput

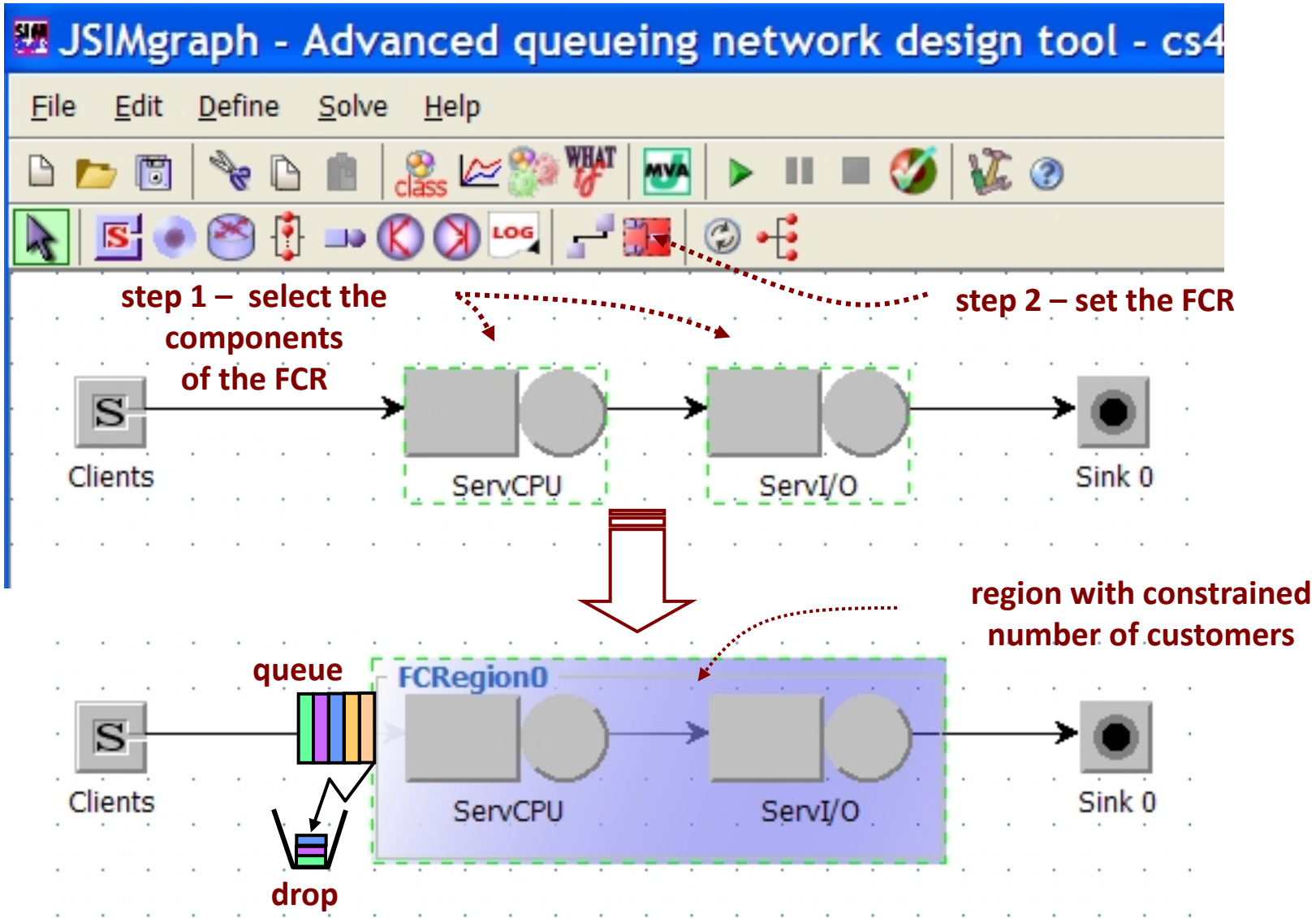


Response times

Activity 4:
Capacity constraints

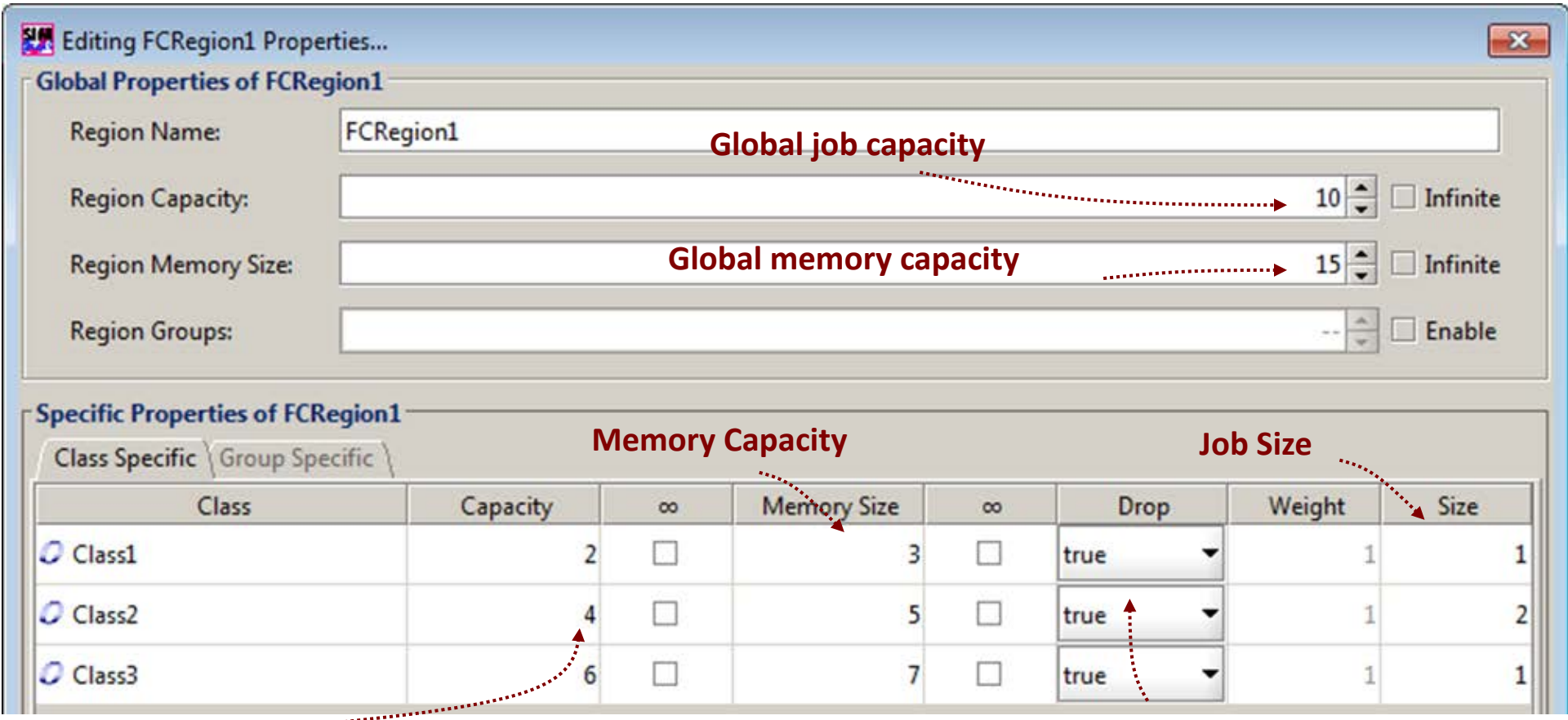
FCR definition

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



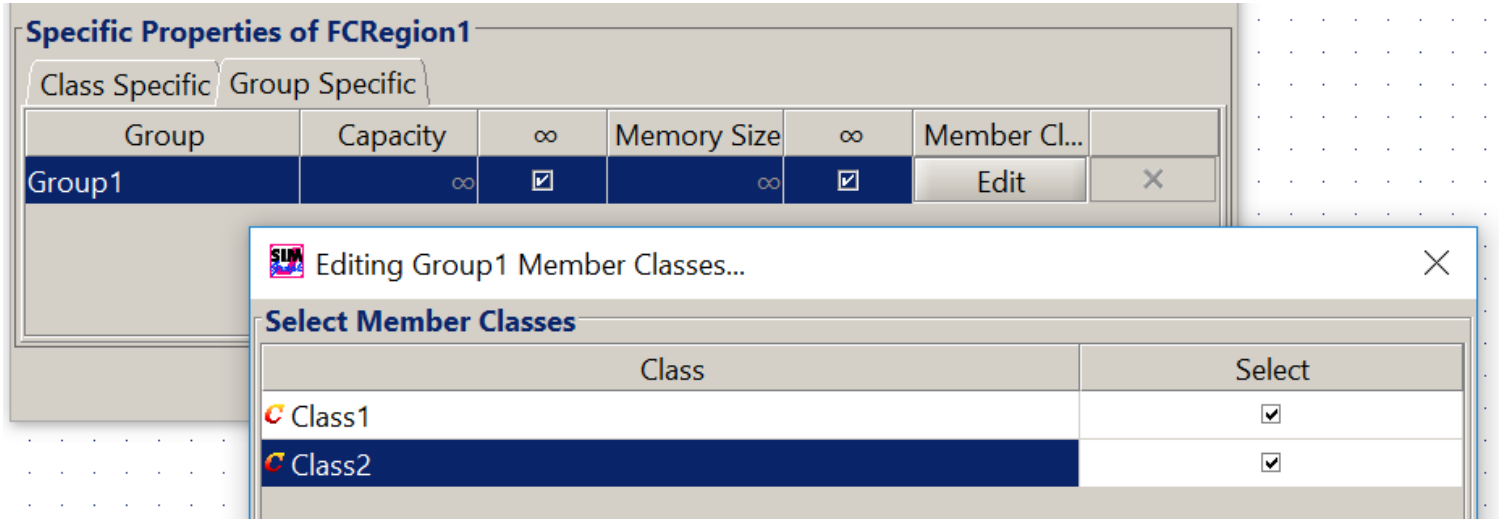
FCR parameters

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

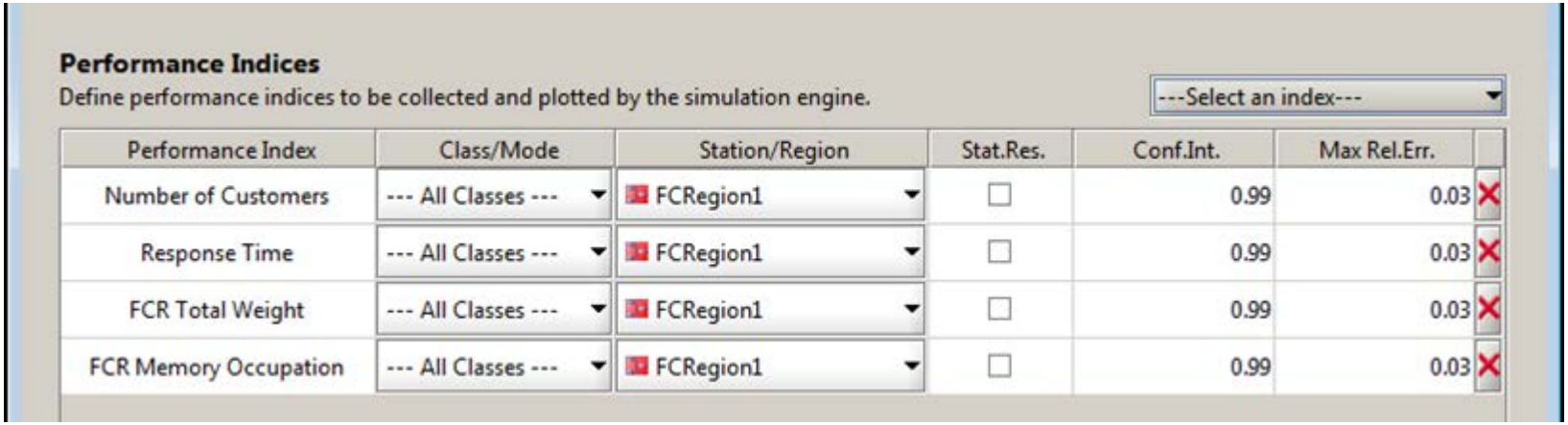


FCR groups and indices

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

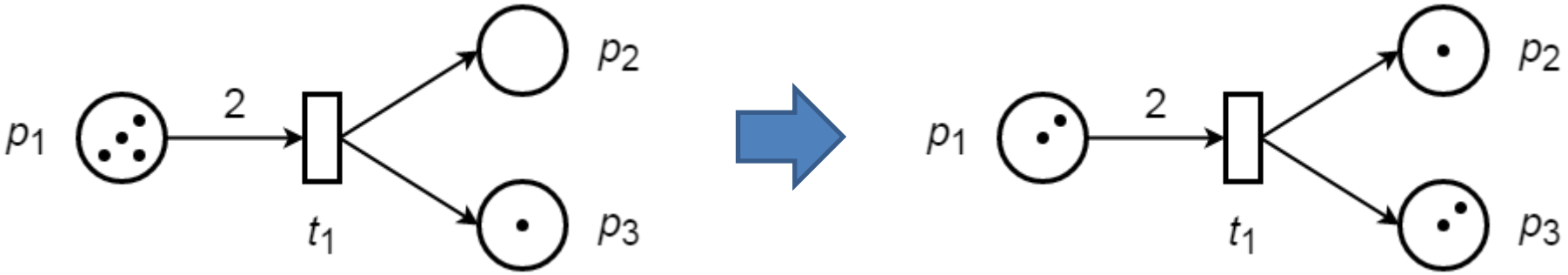


- Dedicated performance indices

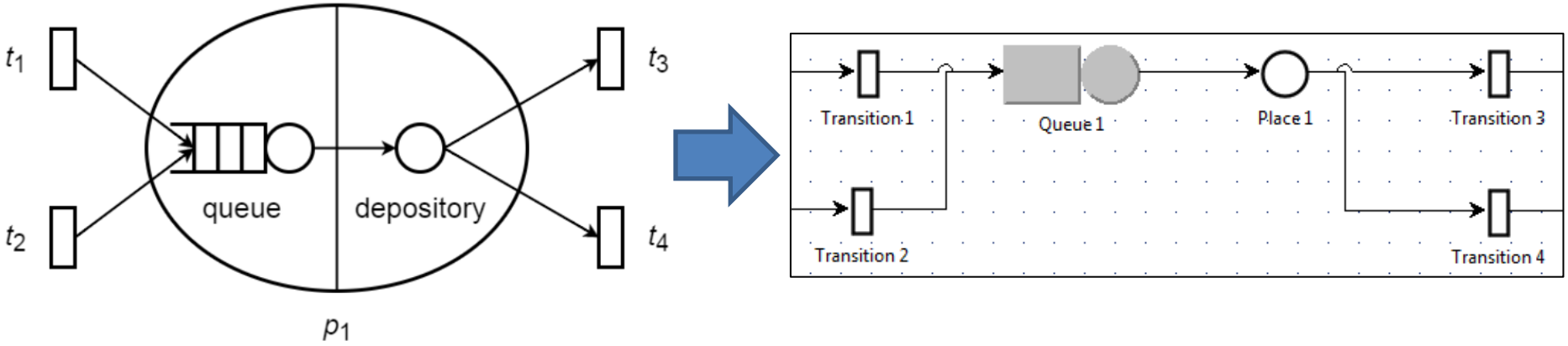


Support for PN elements

- Places and transitions

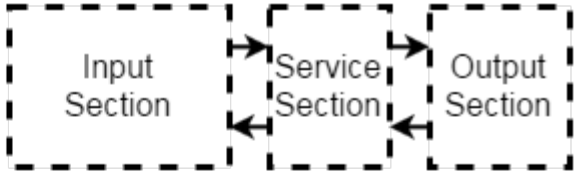


- Queueing Petri nets



PN sections & modes

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

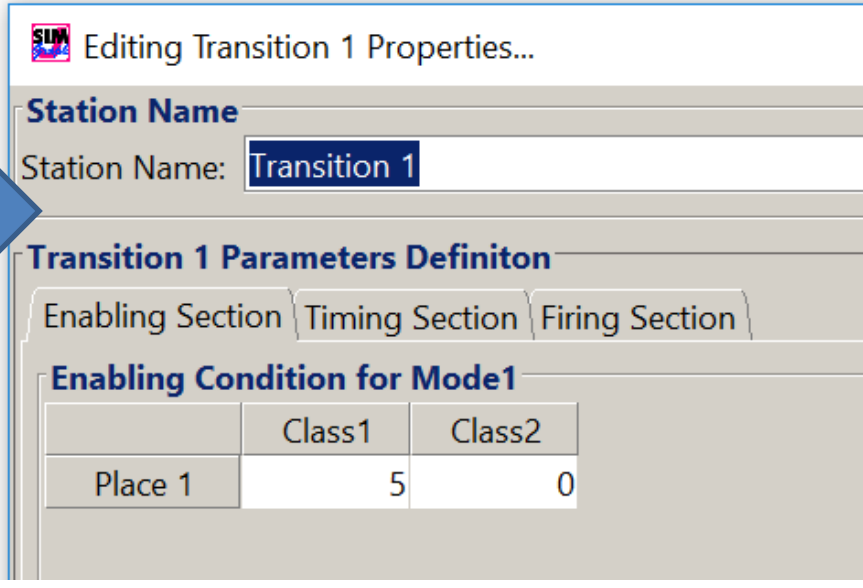


Place Station

Store Section | Tunnel Section | Link Section

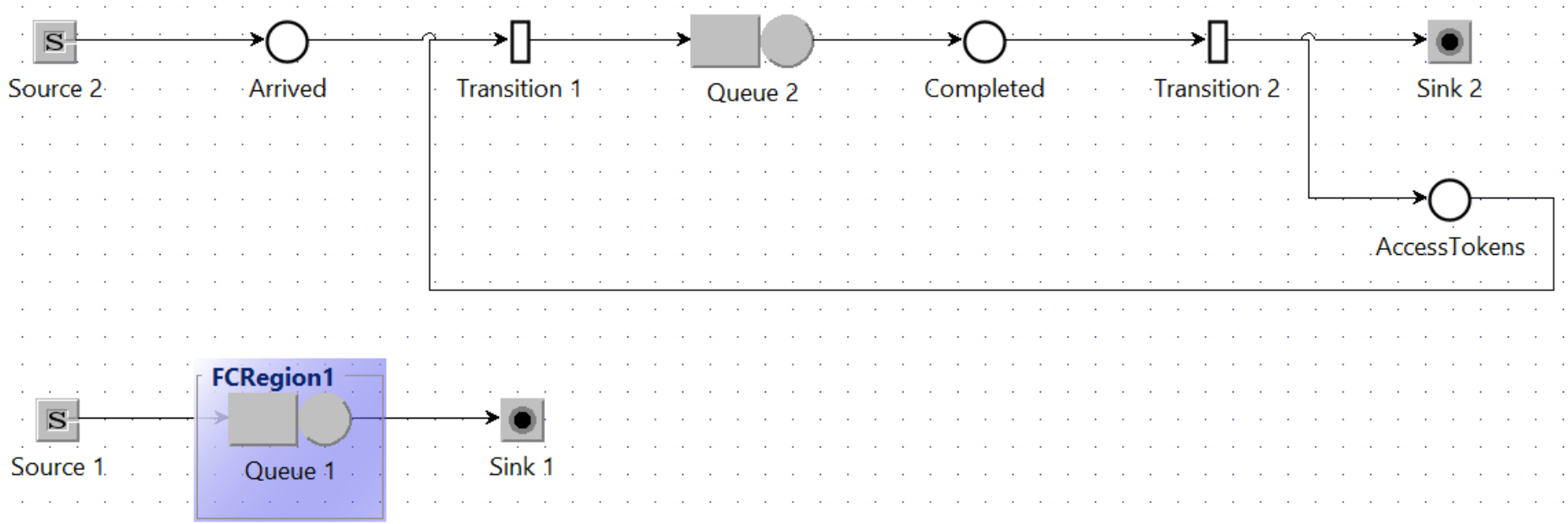
Transition Station

Enabling Section | Timing Section | Firing Section



Hand-on activity: FCRs vs QPNs

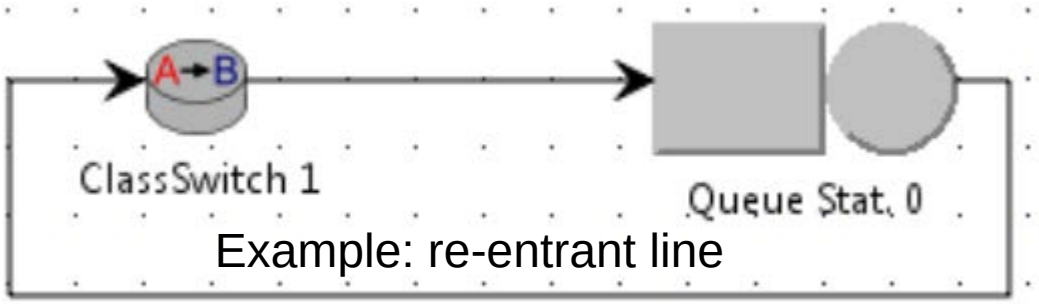
- Arrival rate: λ=0.99 job/s
- Service rate: μ=1.00 job/s
- Goal: restrict max 1 job inside queue



Activity 5:
Workflows & fork-join

Class-switching

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



Example: re-entrant line

Editing ClassSwitch 1 Properties...

Station Name

Station Name: ClassSwitch 1

ClassSwitch 1 Parameters Definiton

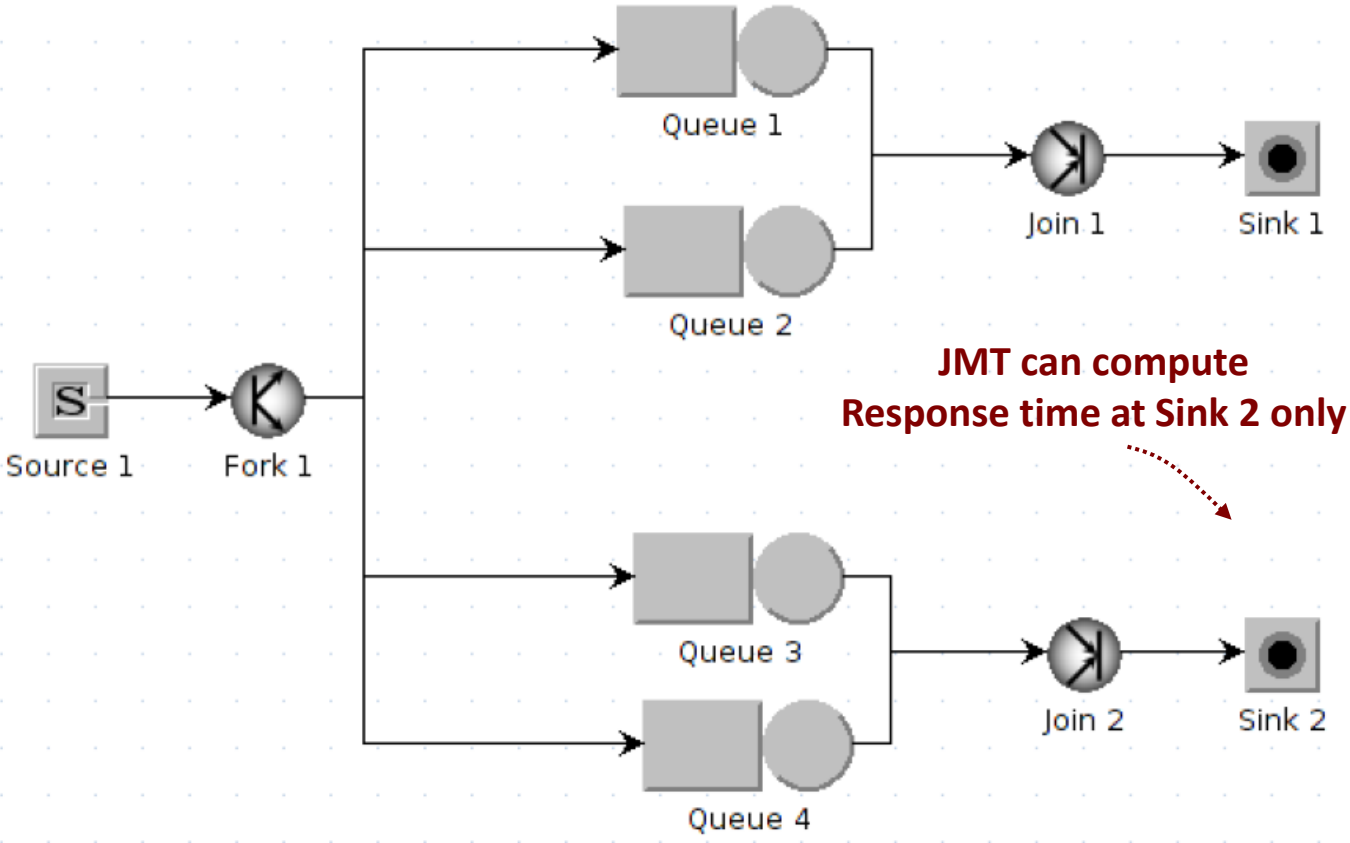
Class Switch Matrix | Routing Section

CS Strategies

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

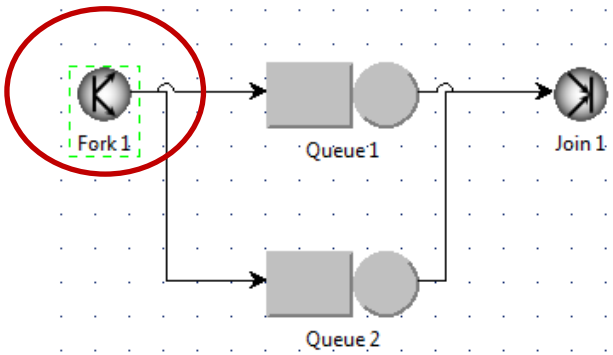
Fork-Join elements

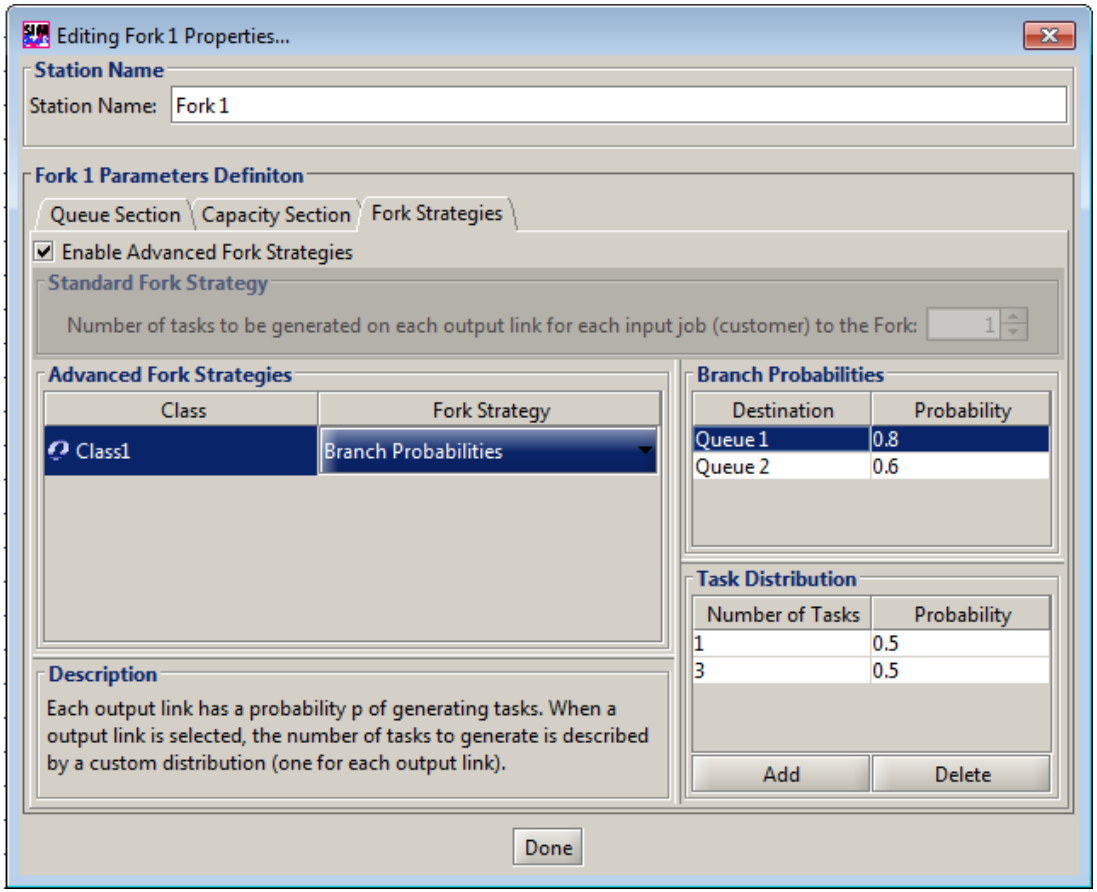
- Jobs split into *tasks* carrying *id* of the parent job
- Support for:
  - nested fork-joins
  - multiple join points
  - finite capacity between fork-join
  - advanced policies (e.g., quorum)



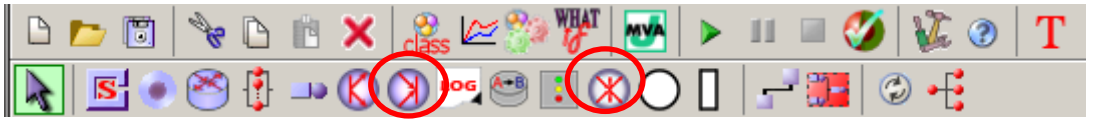
Advanced fork

- 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

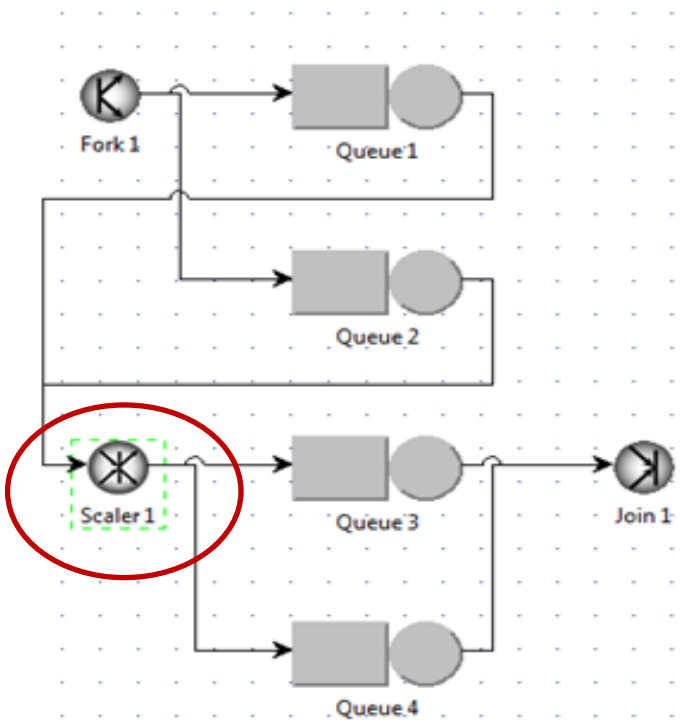


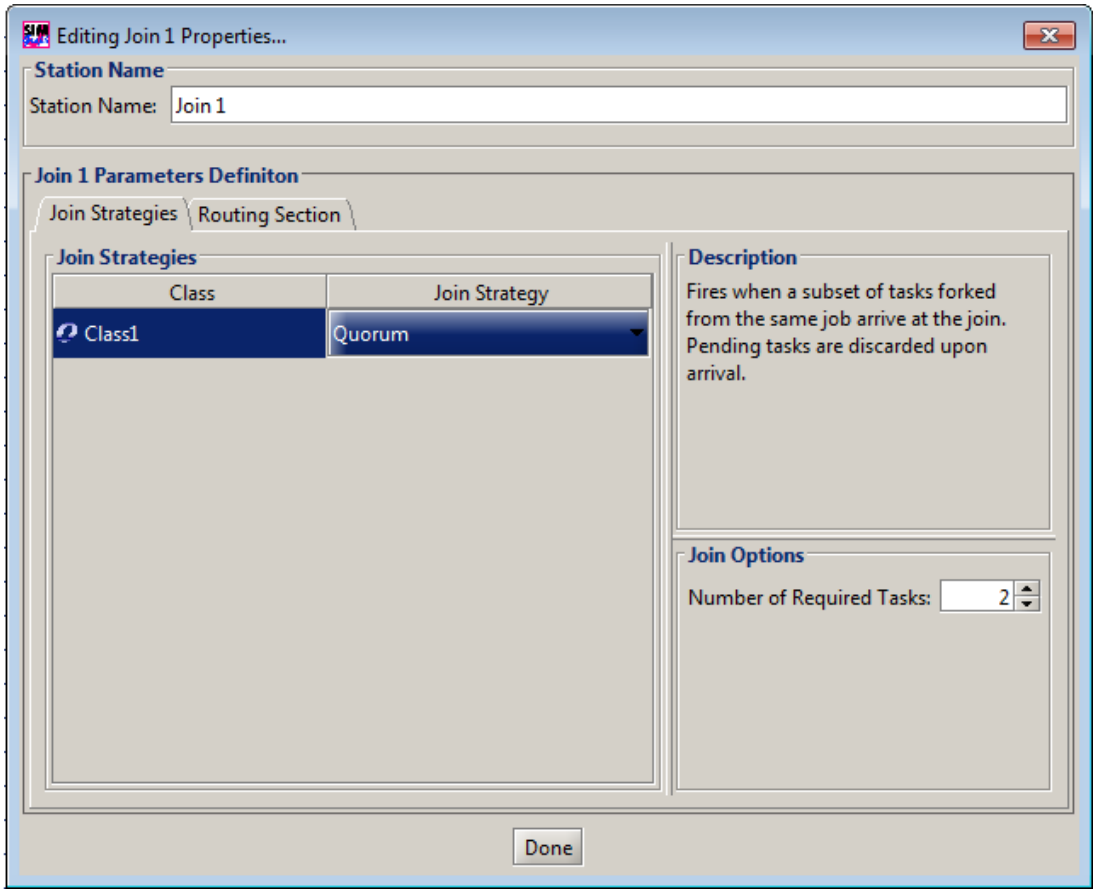


Advanced join



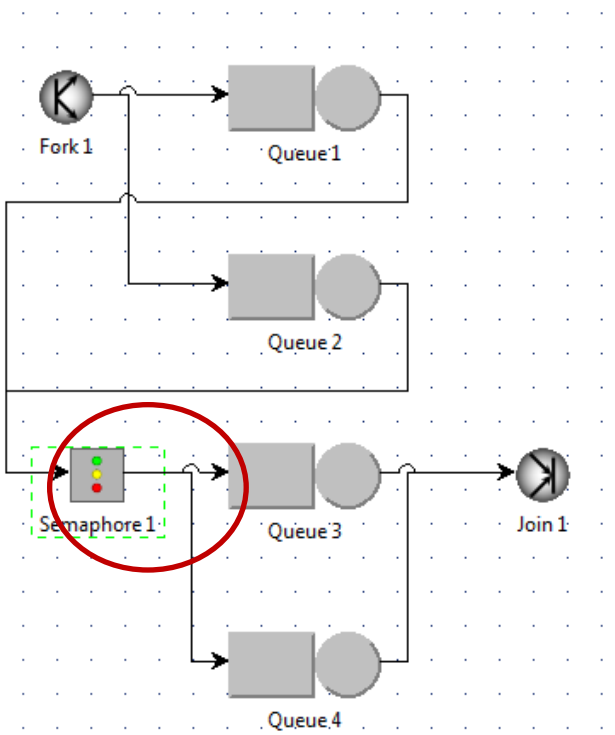
- Quorum: wait a subset of tasks (of the same job)
- Guard: like quorum but requires given class mix
- Scaler: join then fork again

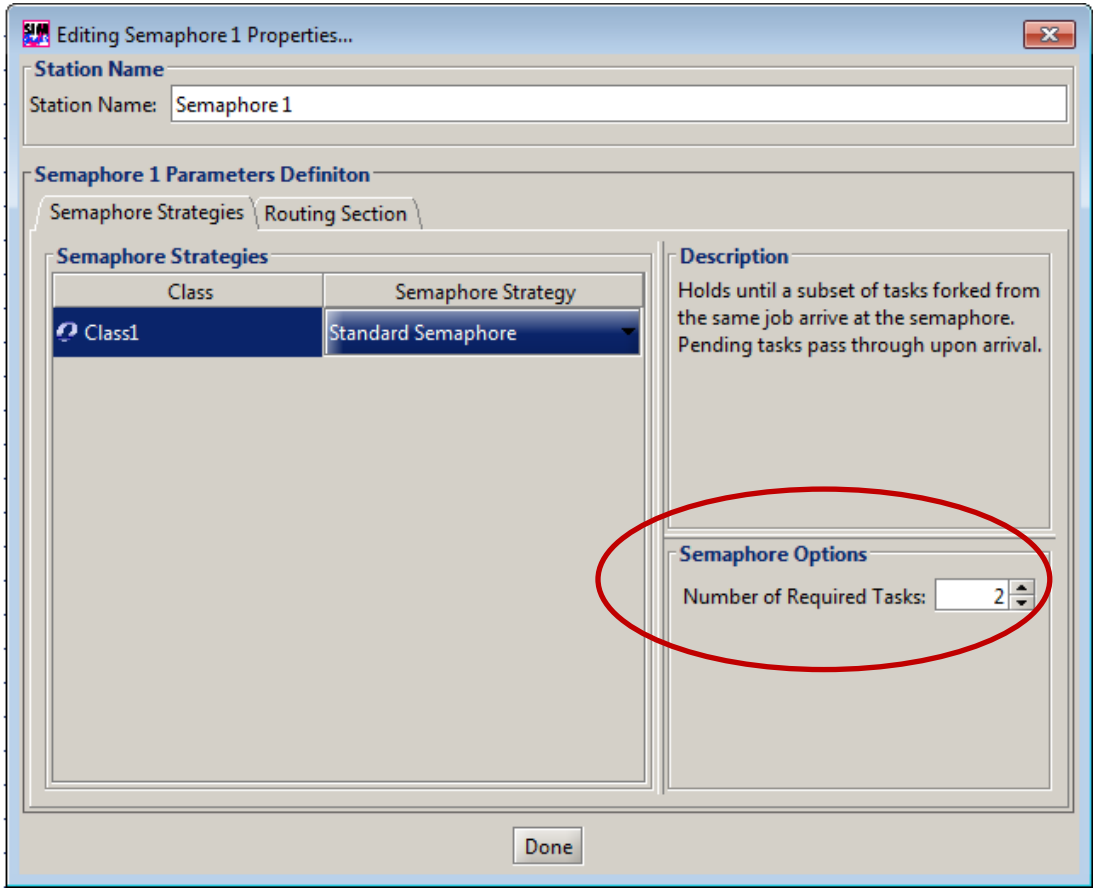




Semaphore

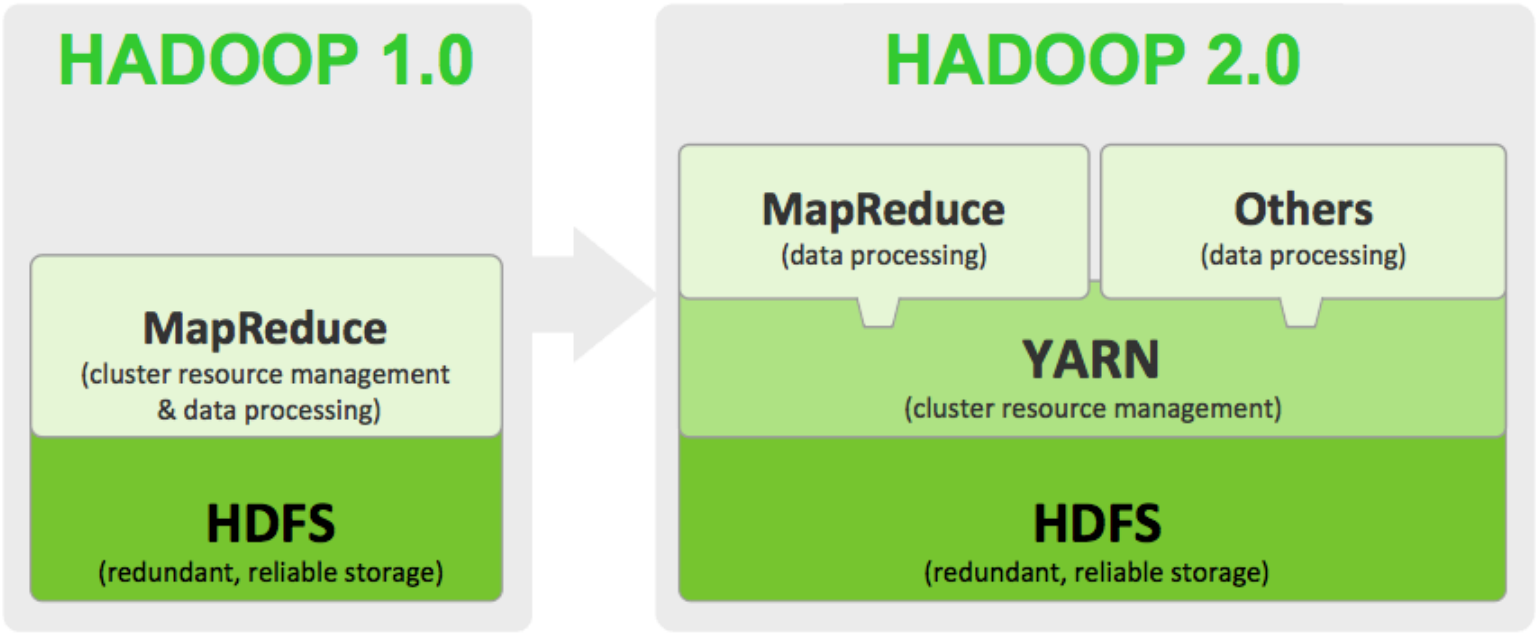
- Block first N tasks forked from the same job
- Upon arrival of the Nth, unblock and let the other pass





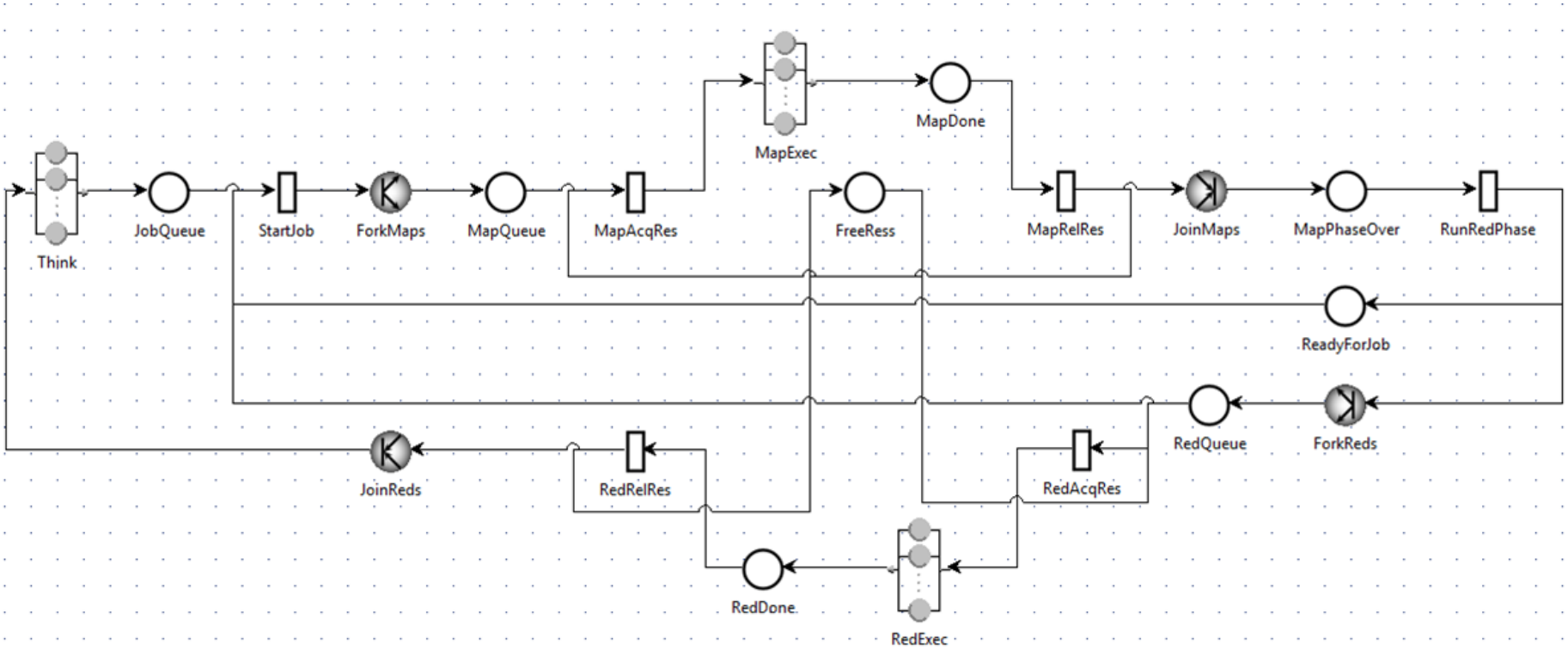
Case Study: YARN Capacity Scheduler

- YARN — Yet Another Resource Negotiator



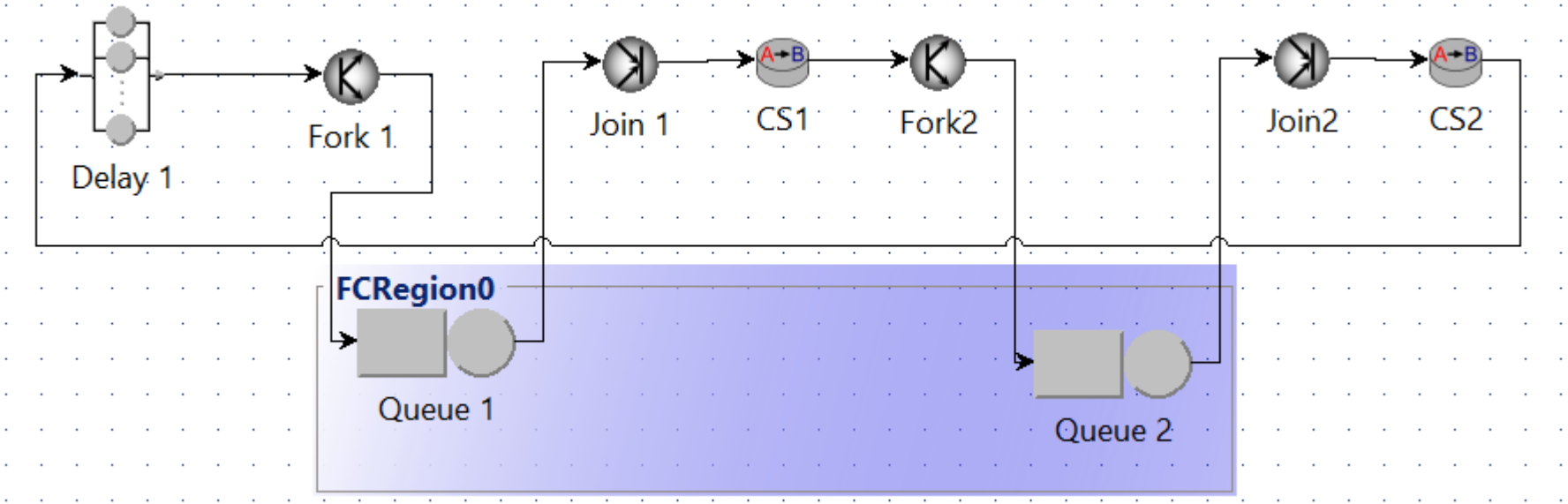
Case Study: YARN Capacity Scheduler

- 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 (>= 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

Tutorial supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644869.