



Politecnico di Milano
EECS Dept.
Milan, Italy

User-Friendly Approach to Capacity Planning studies with **Java Modelling Tools**

Marco Bertoli, Giuliano Casale, Giuseppe Serazzi

outline

- the JMT suite of tools
- the JSIM simulator
- Case Study: optimal admission control policy

the JMT open source suite: six tools

JMT - Java Modelling Tools v.0.7.4

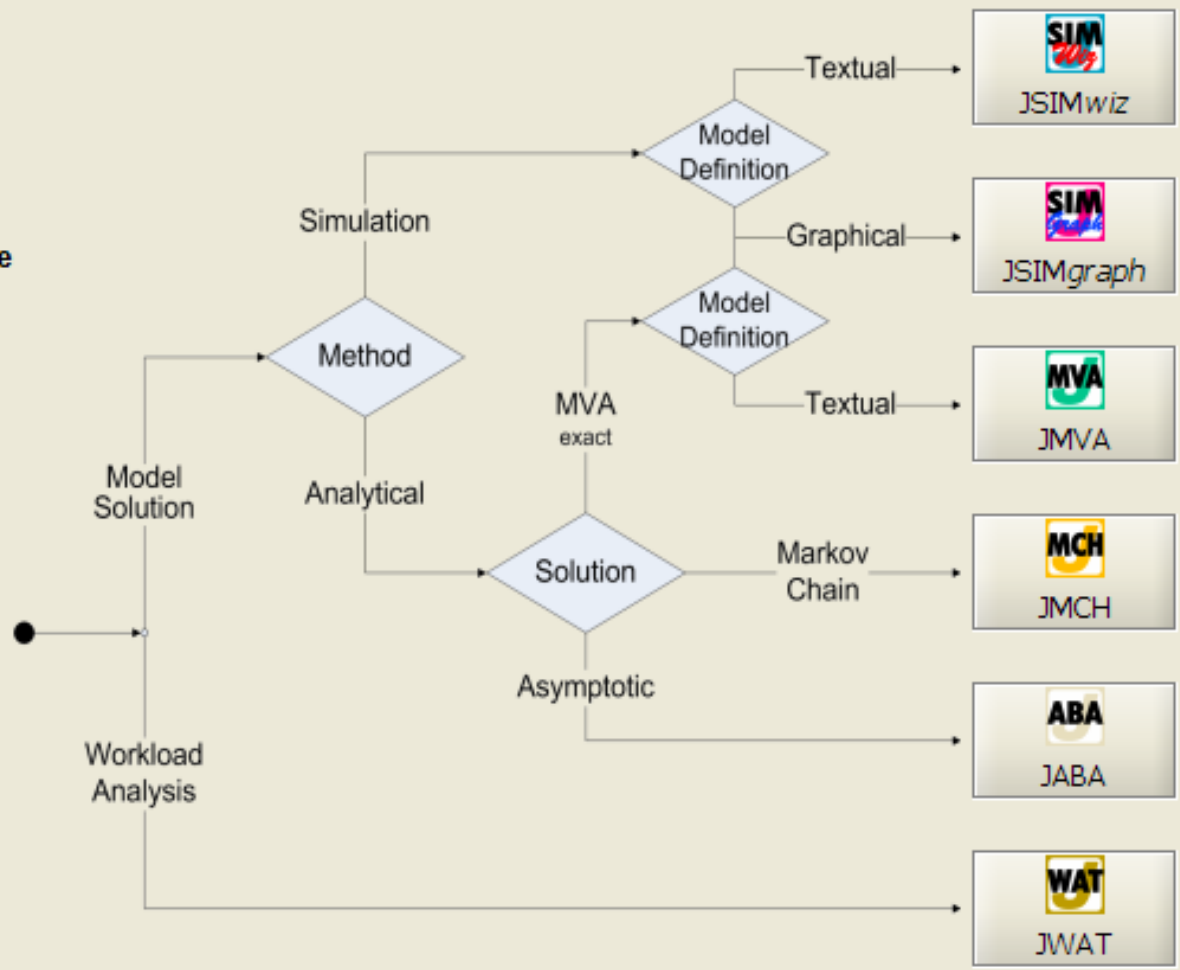
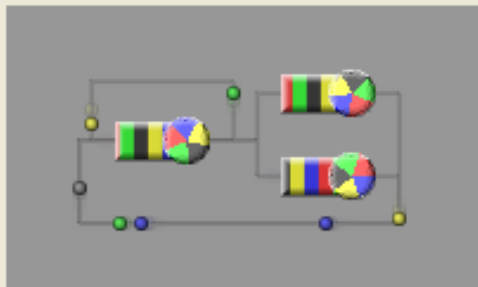


Java Modelling Tools v.0.7.4
Performance Evaluation Lab
Dipartimento di Elettronica e Informazione
Politecnico di Milano - Italy

Project Coordinator: prof. G.Serazzi

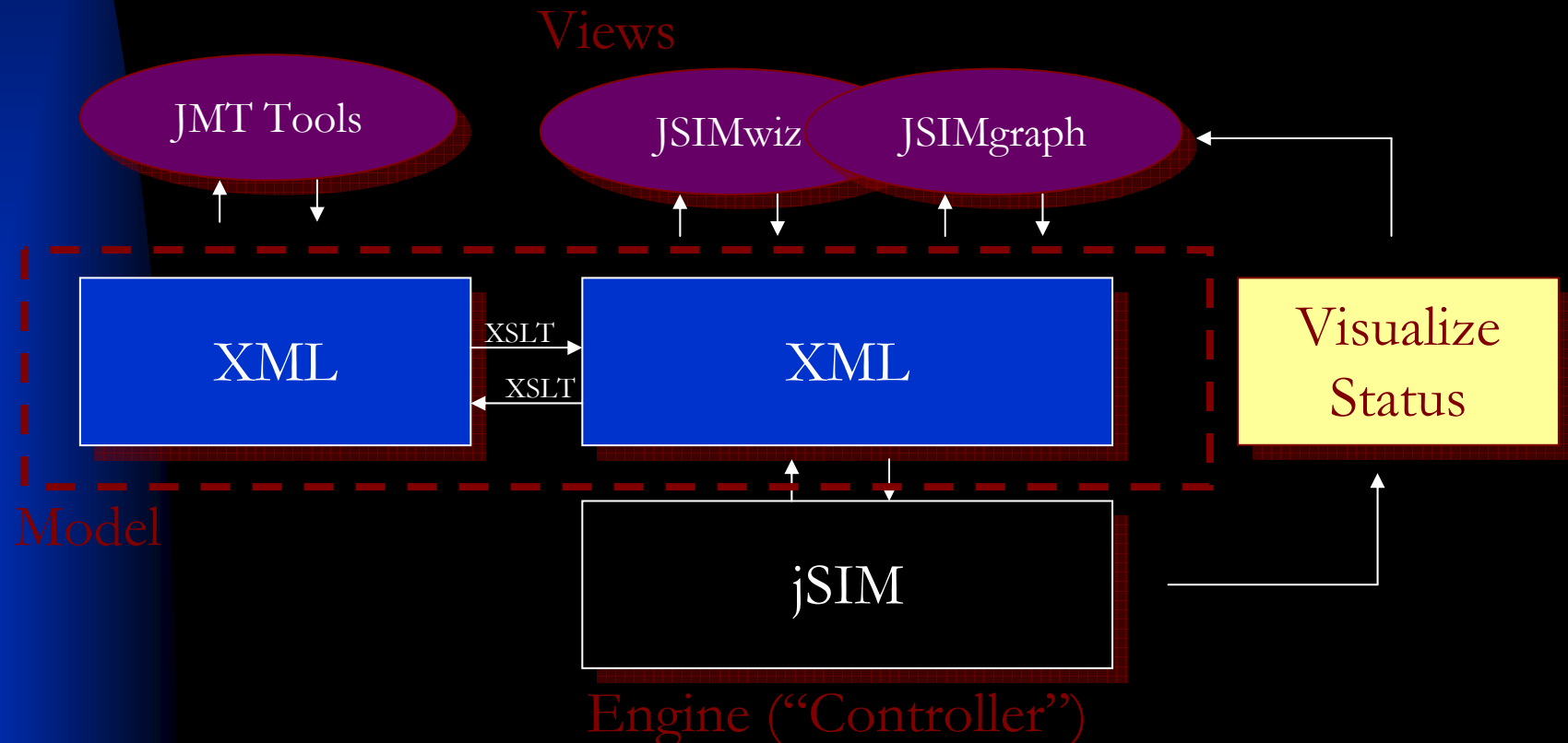
Introduction to JMT

Online Documentation



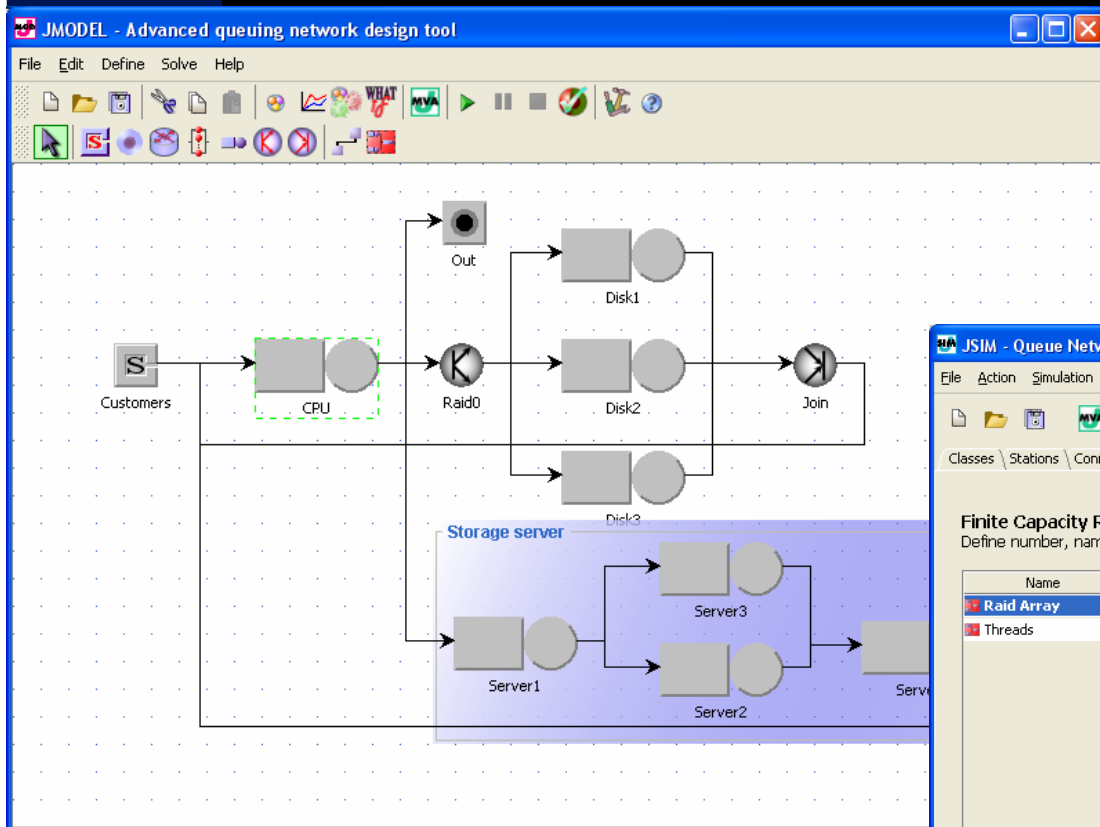
the JMT architecture

- “Model-View-Controller”-like pattern
 - ◆ Better reuse and isolation of components



the JSIM simulator: two graphical interfaces

JSIMgraph



JSIMwiz

The screenshot shows the configuration window for a **Finite Capacity Region** in the JSIM simulator. The window title is "JSIM - Queue Network Models Simulator - WebServer.jmodel".

Finite Capacity Region Characteristics
Define number, name, composition, global and class specific constraints for finite capacity regions.

Regions: 2

| Name | Capacity | ∞ |
|------------|----------|-------------------------------------|
| Raid Array | 20 | <input type="checkbox"/> |
| Threads | ∞ | <input checked="" type="checkbox"/> |

Stations in Raid Array

| Station name | |
|--------------|-------------------------------------|
| Raid disk 2 | <input checked="" type="checkbox"/> |
| Raid disk 1 | <input checked="" type="checkbox"/> |
| Raid disk 0 | <input checked="" type="checkbox"/> |

Stations: 3

Class specific Raid Array Properties

| Class | Capacity | ∞ | Drop |
|--------|----------|-------------------------------------|-------|
| Class0 | 10 | <input type="checkbox"/> | true |
| Class1 | ∞ | <input checked="" type="checkbox"/> | true |
| Class2 | ∞ | <input checked="" type="checkbox"/> | false |

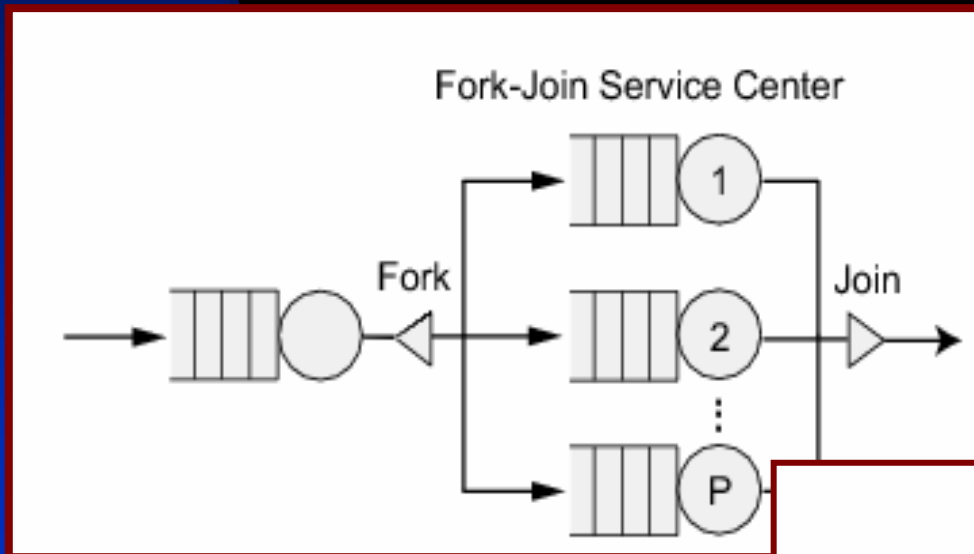
Buttons: < Back, Next >, Solve, Cancel, Help

JSIM Engine

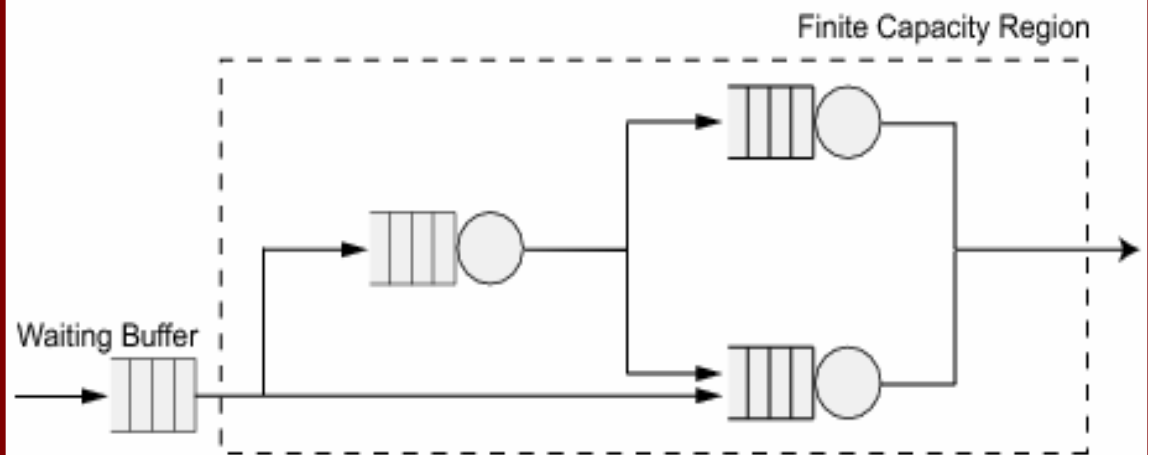
- discrete-event simulator for queueing networks
- **several** distributions (exp, Erlang, Pareto, **burst/MMPP2**, ...)
- support for NPF features:
 - ◆ **general** arrival and service processes
 - ◆ **Fork-Join** centers
 - ◆ **blocking** and **finite capacity** regions
 - ◆ **priority** Classes
 - ◆ **state-dependent** routing:
 - ★ route to least utilized center, to shortest queue
 - ★ route to the center with shortest response time
 - ★ fastest service time, round robin, random
- **Logger** component (debugging, processing of transient data, ...)

Fork-Join and Finite Capacity features

- **Fork and Join** components
 - ◆ fork node: jobs are forked into P tasks
 - ◆ Synchronization at the join node



- a group of queues can be tagged as a region with **finite capacity**
 - ◆ non-admitted jobs can be either in a FCFS waiting buffer or dropped



Statistical Analysis

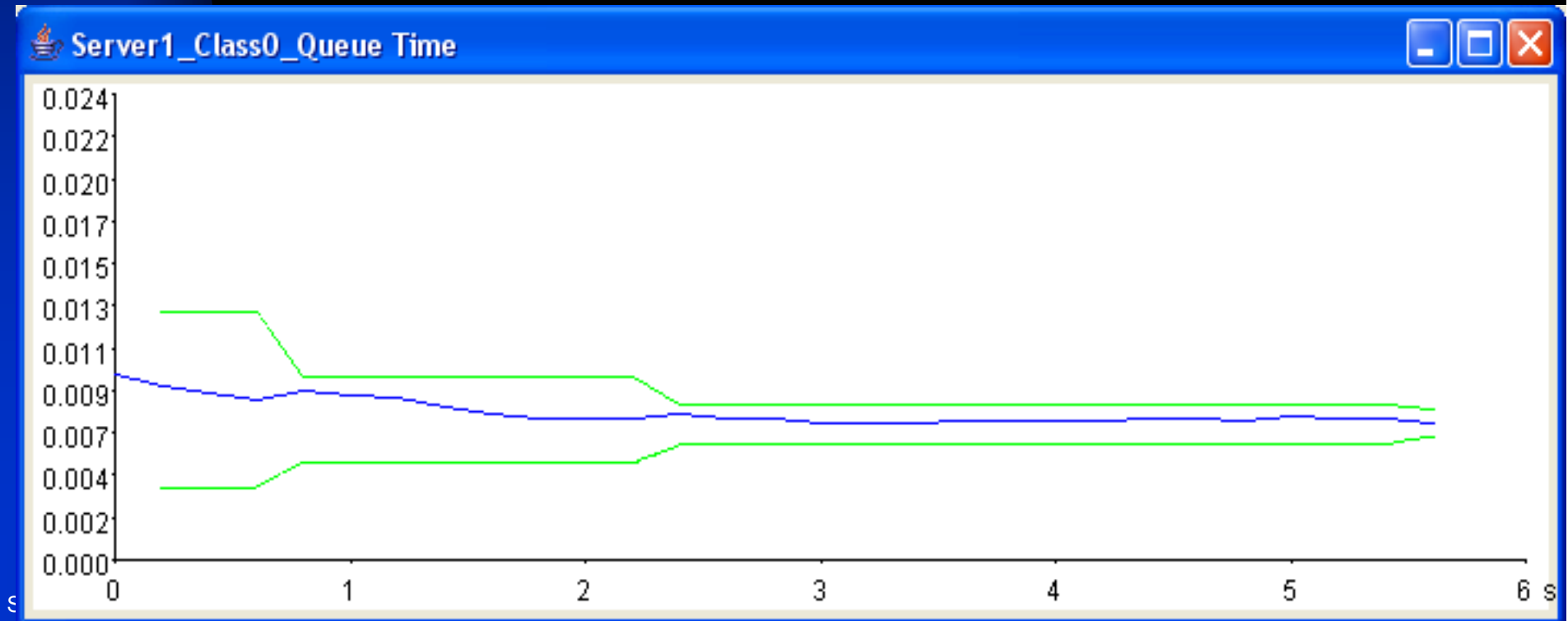
- **Automatic removal of the initial bias**
 - ◆ R-5 Heuristic
 - ◆ MSER-5 Rule (Marginal Standard Error Rule)
- **C.I. generation using spectral methods**
 - ◆ Spectral Analysis [Heidelberger & Welch, 1981]
 - ◆ Used also for run-length control

Arrival and Service Process

- Exponential **insufficient** for many models
 - ◆ Pareto, Hyperexponential, Erlang, Gamma, burst general/MMPP2, ...
 - ◆ **Custom** distribution (external text file, from log, from Logger, future JWAT)
- **Random number** generation
 - ◆ Mersenne Twister
- **Load-dependent** service process
 - ◆ Server speed variable with the current queue-length
 - ◆ Building block for Hierarchical Modeling

simplification of simulation experiments

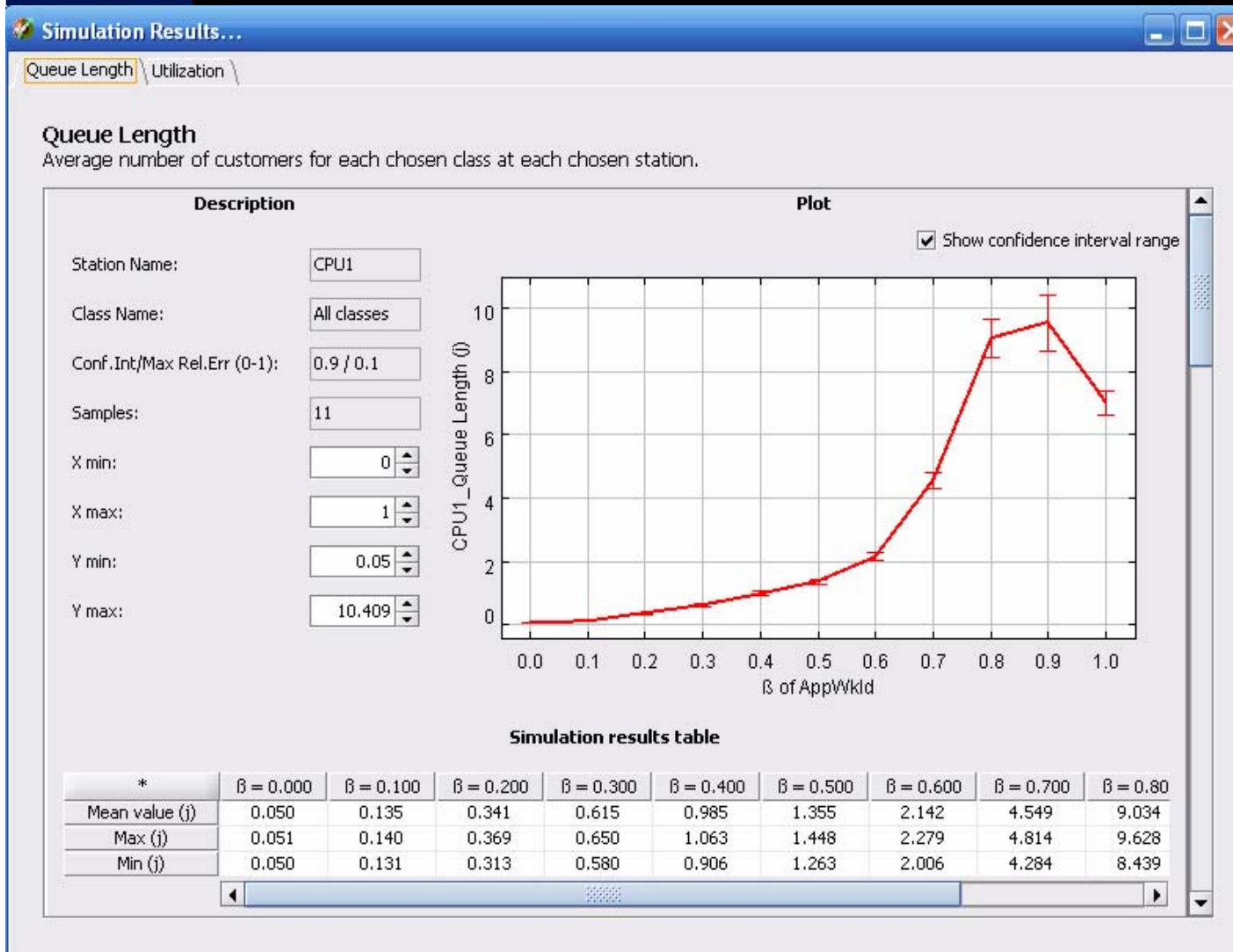
- automatic **maximum relative error** control [Pawlikowski 1990]
 - ◆ ratio half-width marginal CI / estimated mean
- automatic **removal** of the initial bias (transient filtering)
- **max n. of samples** (long run analysis) and simulation time
- **CI generation** using spectral methods



What-if Analysis

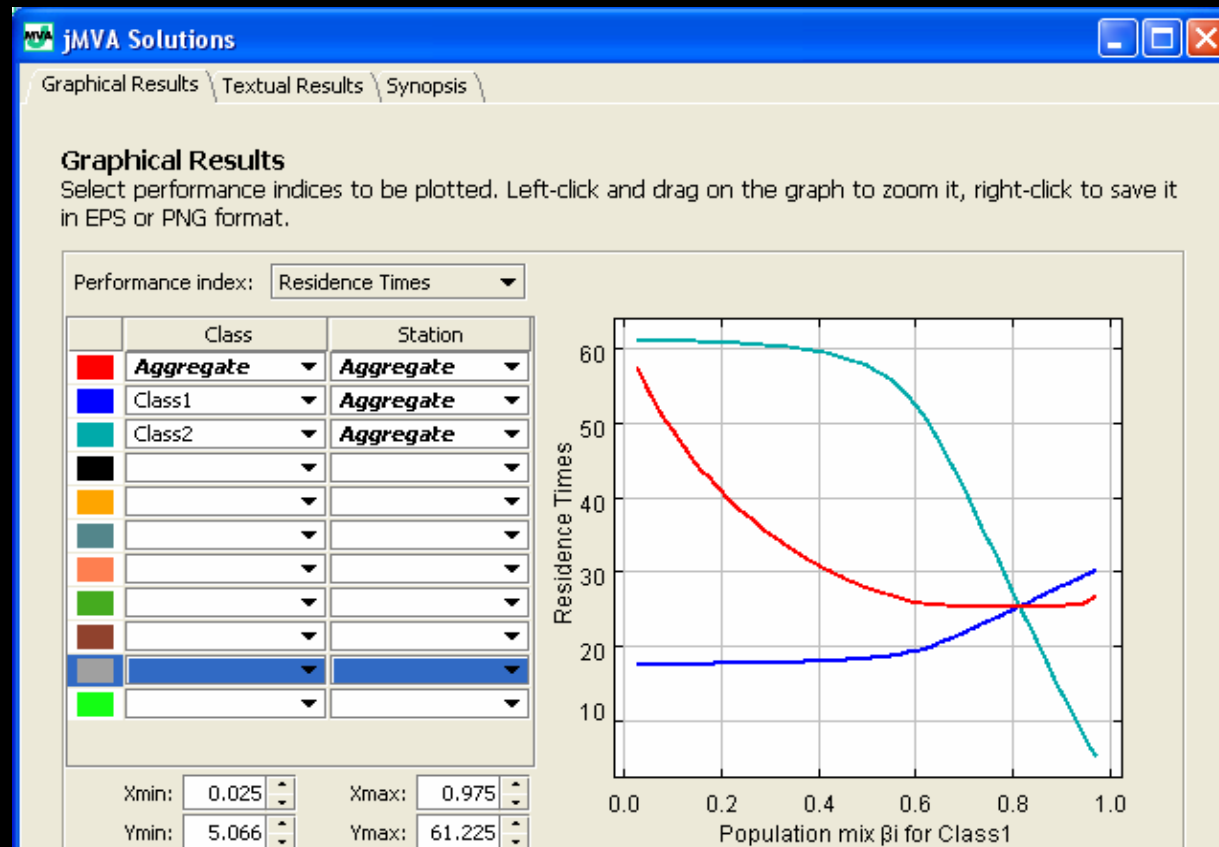
simulations control parameters

- ◆ arrival rate (cl.)
- ◆ customer numbers
- ◆ service demands
- ◆ pop. mix (2 class)

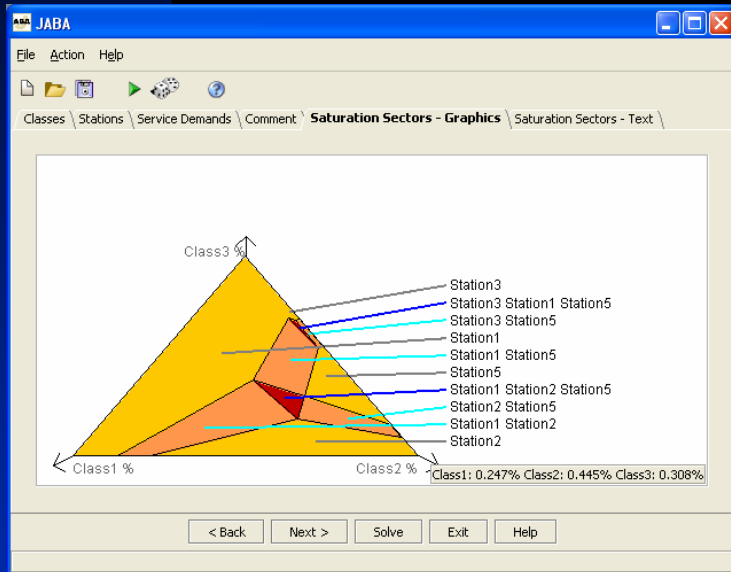


the JMVA analytic solver

- Solve open/closed/mixed BCMP queueing nets
 - ◆ Native support for what-if analyses
 - ◆ Integrated with JSIMgraph (reuse models)

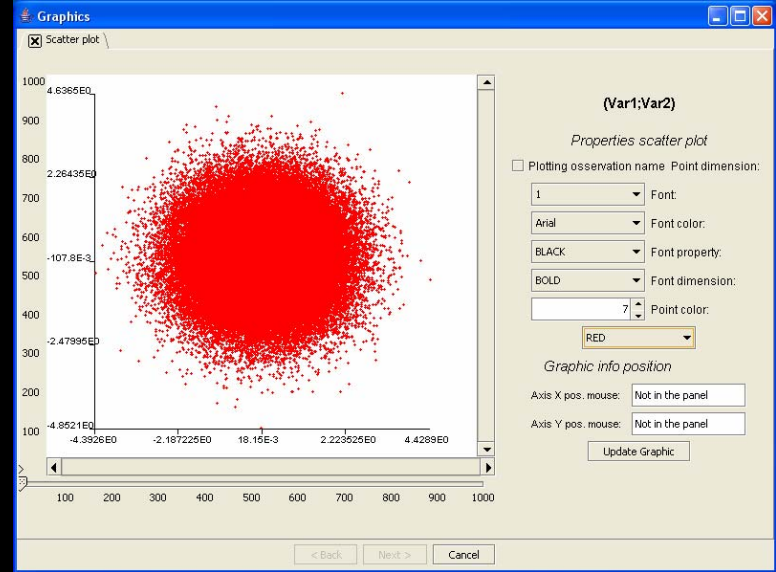


jABA/jMCH/jWAT

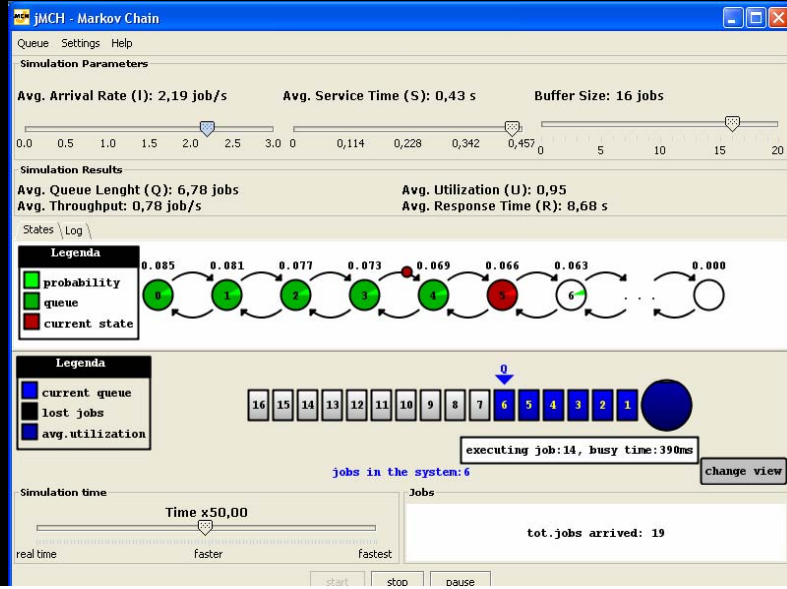


jABA

jMCH



jWAT



Case Study: maximization of throughput

- Multi-tier system: Front Server, Storage Server, Database server
- Workload: two web services WS1 (class 1) and WS2 (class 2)
 - ◆ Finite Capacity Region with **constant** population of requests (N_1, N_2) , $N_1 + N_2 = N = 100$

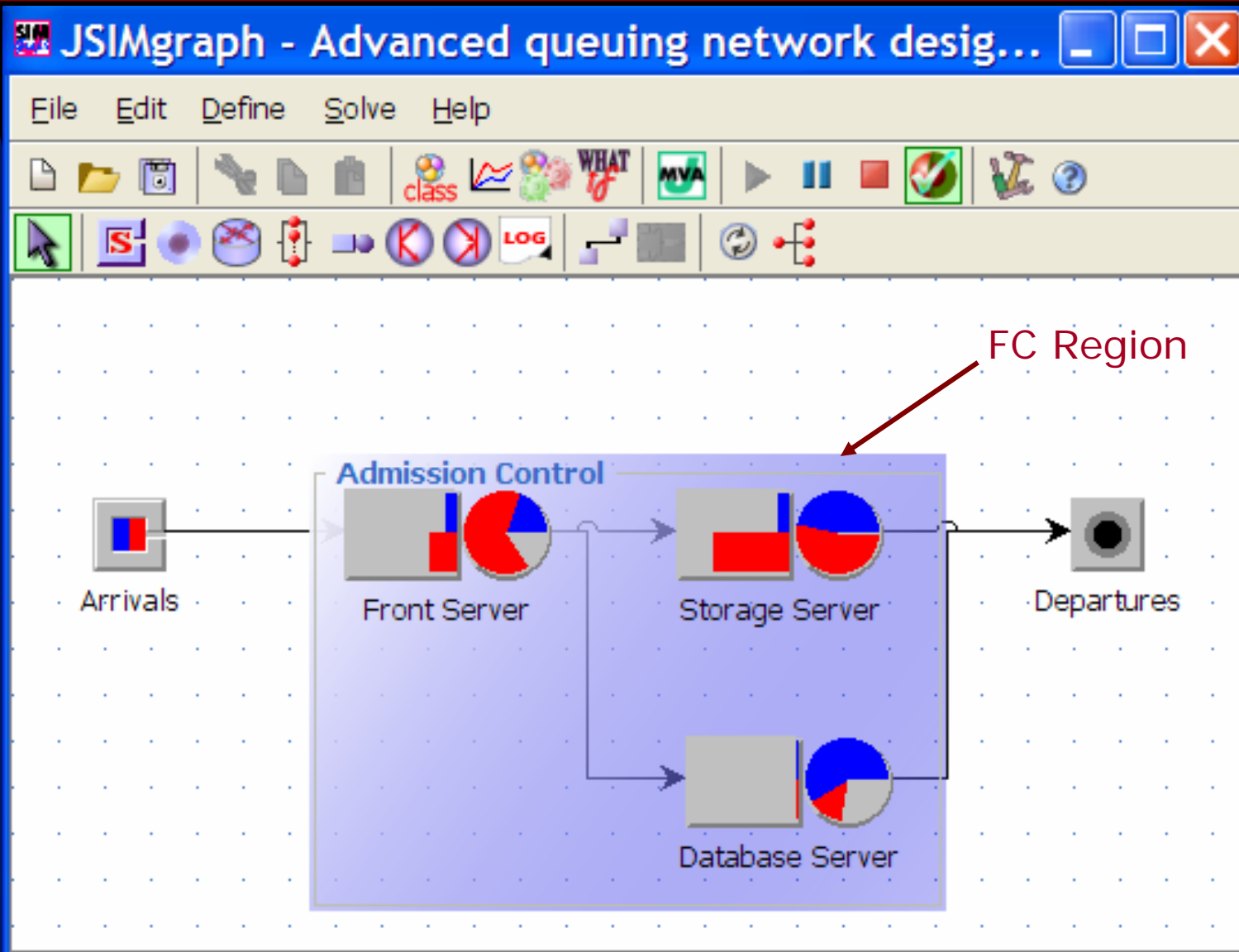
Admission Control algorithm → BEST mix of requests WS1+WS2

| Parameters | | Web service WS1 | Web Service WS2 |
|-----------------|---------------|-----------------|-----------------|
| Front Server | | | |
| service demand | D_{FS} [ms] | 28.48 | 68.07 |
| Storage Server | | | |
| service demand | D_{SS} [ms] | 69.15 | 55.18 |
| Database Server | | | |
| service demand | D_{DB} [ms] | 86.86 | 13.95 |

bottlenecks



Case Study – JSIM Graphical interface

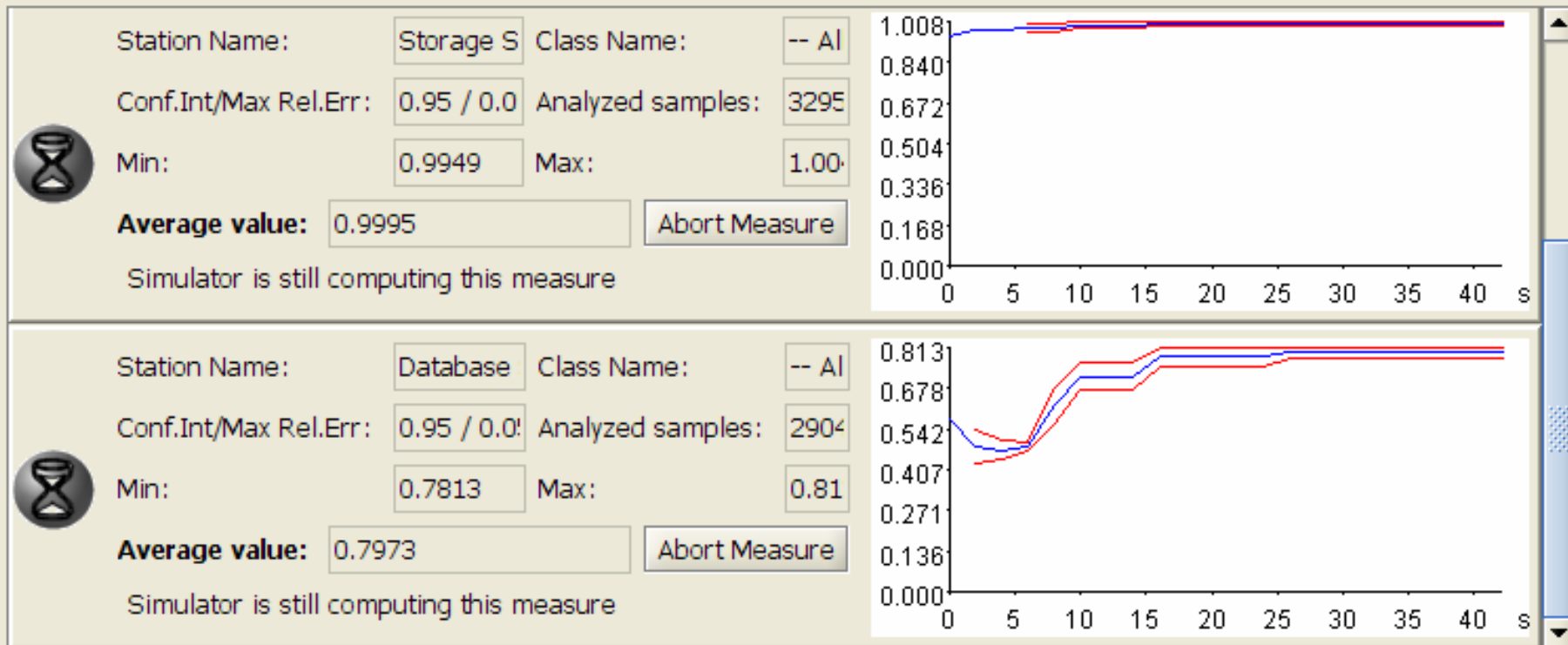


Case Study – JSIM simulation progress

Utilization \ System Response Time \ System Throughput \ Customer Number \

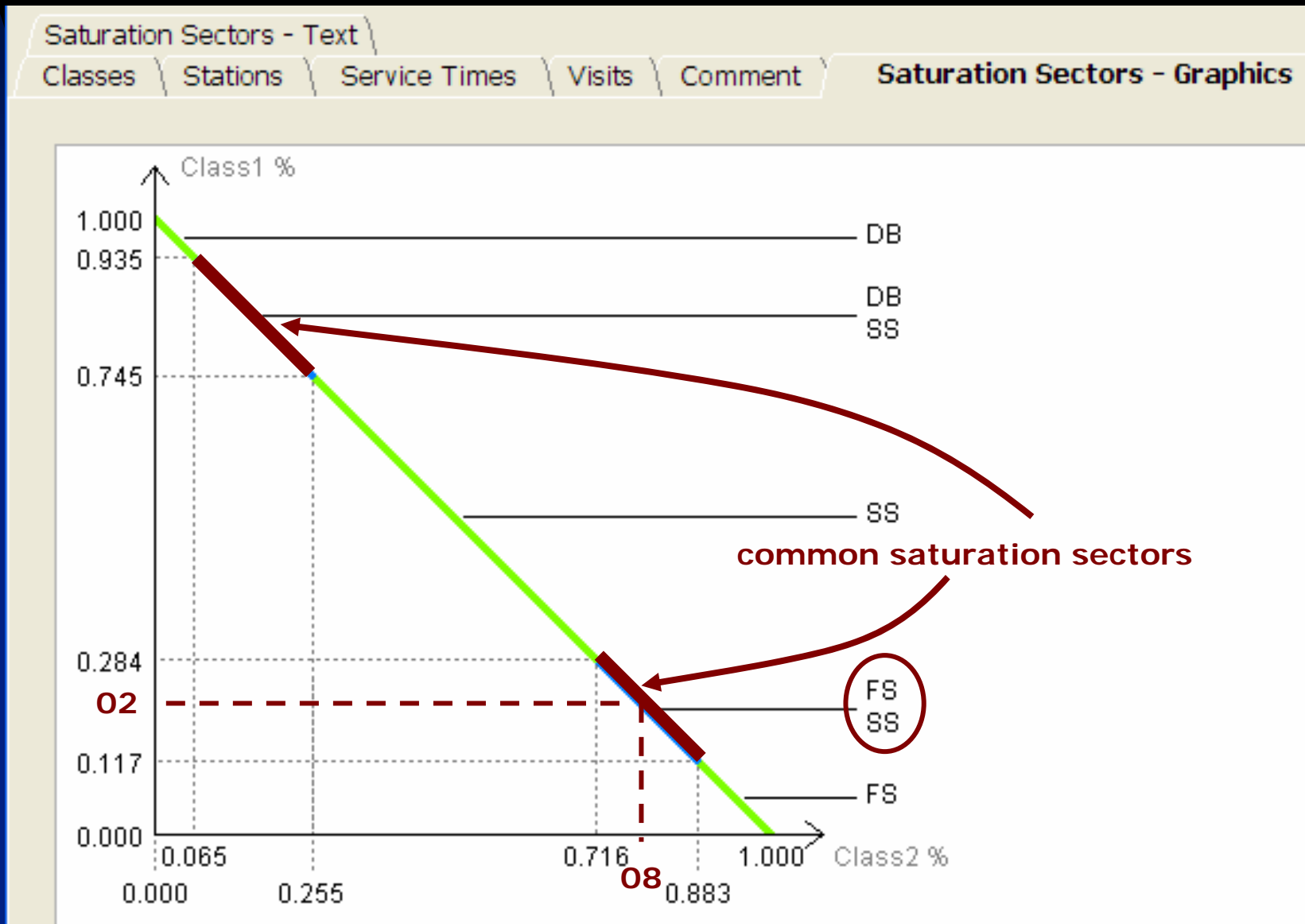
Utilization

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

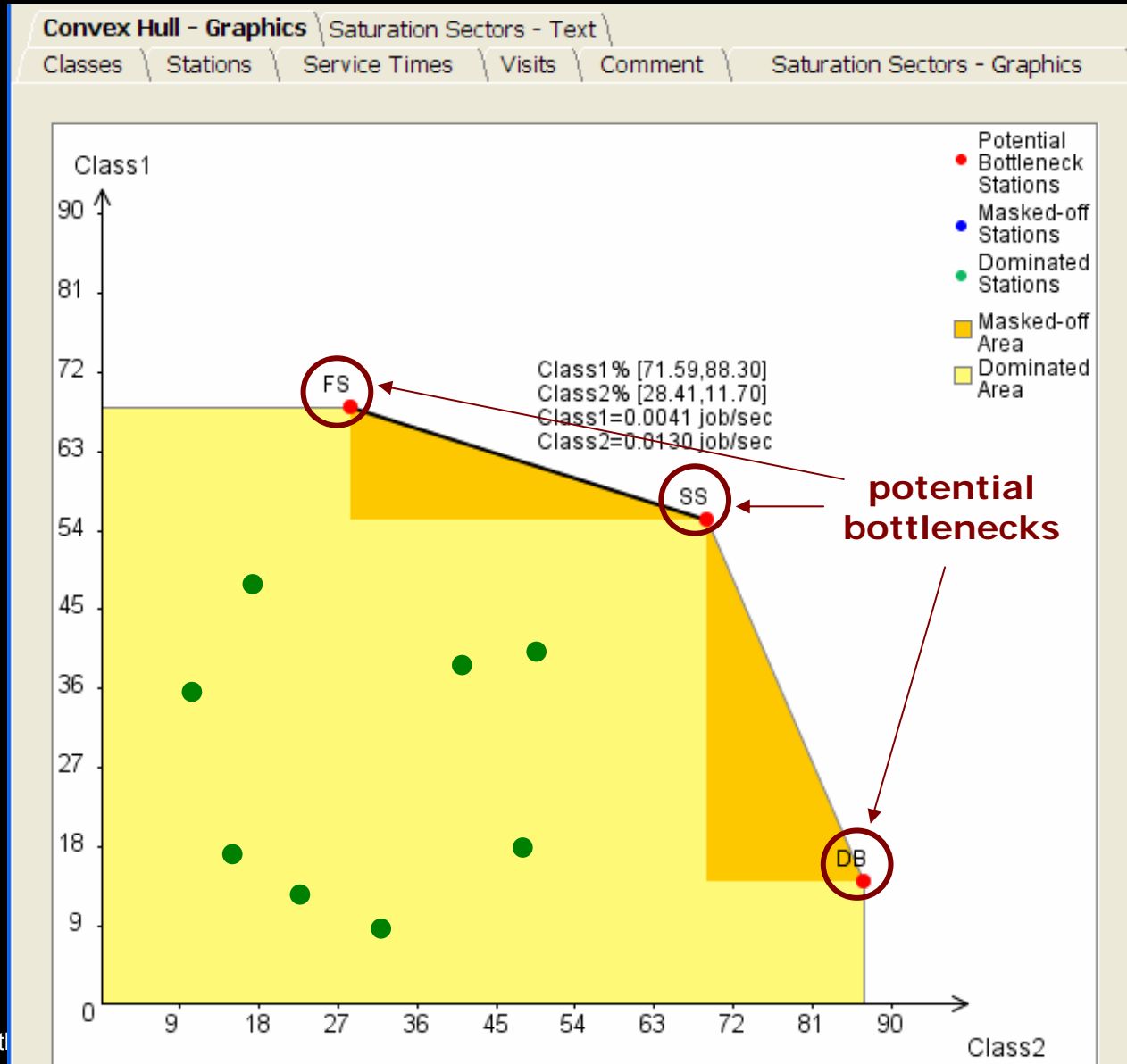


Double click on this graph to zoom

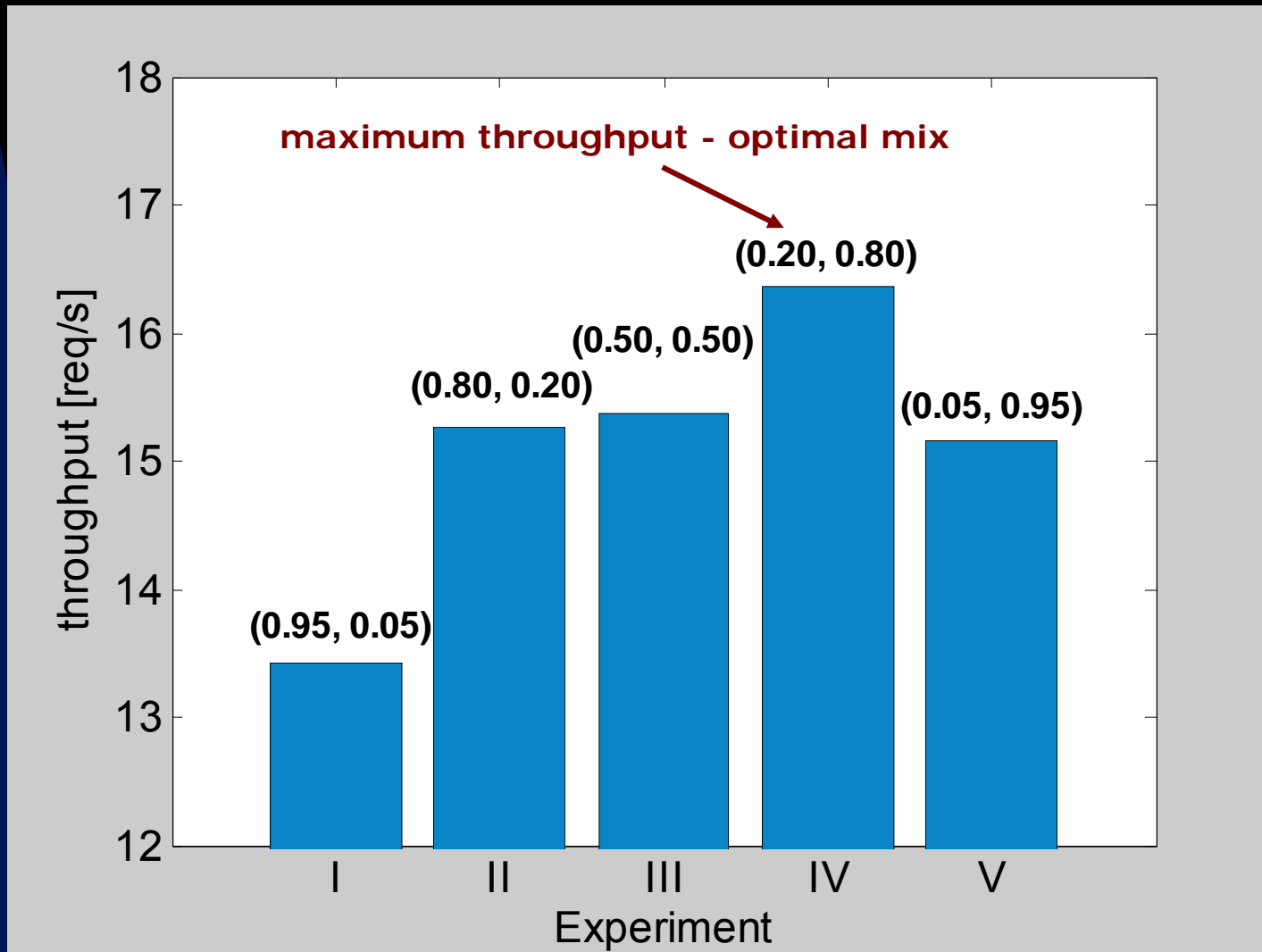
Case Study: JABA Asymptotic Analysis



Case Study: JABA convex hull



Case Study: throughput vs mix of requests



conclusions

- the project

<http://jmt.sourceforge.net>

> 11000 downloads since April 2006