УДК 681.32

Seyedeh Melina Meraji, Yuri Ladyzhensky
Donetsk National Technical University, Donetsk, Ukraine
Department of Computer Engineering
E-mail: melina_meraji@yahoo.com

RESEARCH OF JAVA MODELING TOOLS USING FOR MARKOV SIMULATIONS

Abstract

Seyedeh Melina Meraji, YuriLadyzhensky.Research of Java Modeling Tools UsingFor Markov Simulations.The Java Modeling Tools (JMT) is an open source software for performance evaluation, capacity planning and modeling of computer and communication systems. The software implements algorithms for the exact, asymptotic and simulation analysis of queuing networkmodels. Modelscan be described wizard dialogs or witha graphical user-friendly interface. The software includes alsoa workload analysis tool based on clustering techniques.

Introduction

The availability of several simulation packages, either commercial or free, makes simulation one of the most commonly used techniques for performance evaluation of computer systems and networks. In general, evaluation techniques, that are methods by which performance evaluation indices are obtained, can be subdivided into two main categories: measurement (or empirical) techniques and modeling techniques.

Empirical techniques require that the system or network to be evaluated exists and direct measurements of the evaluation target have to be taken. On the other hand, modeling techniques only require a model of the system. Modeling techniques are of two types: simulation and analytic.Among them, simulationis the most popular, as it applies to a wider variety of systems and does not require restrictiveassumptions. However, in spite of their generality and ease of use, simulation models may fail or producenon-accurate results.

The suite incorporates an XML data layer that enables full reusability of the computational engines.The JMT suite is composed by the following tools:

JSIMwiz, JSIMgraph, JMVA, JMCH, JABA, JWAT.

***JMT* Suite**

The JMT suite is composed of five tools that support different analyses frequently used in capacity planning studies.The organization of the suite is depicted in Figure 1. The main features of each tool follows.

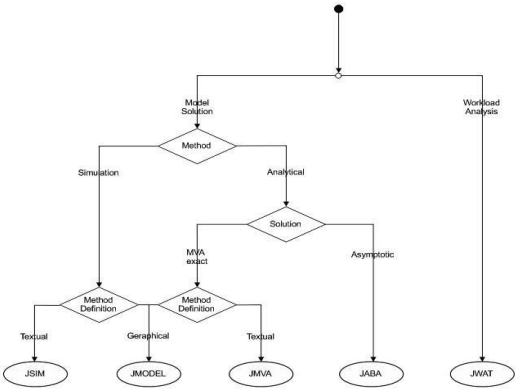


Fig. 1- JMT Suite Organization

Simulator Architecture

A graphical illustration of simulator architecture is given in Figure 2.

The underlying design principle is to obtain a complete separation of the presentation and computational layers using XML.

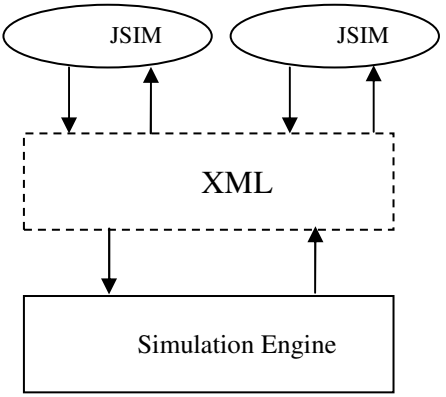


Fig. 2-Simulator architecture

This approach has several benefits. For instance, it is possible to reuse the simulation engine within external applications; further, it simplifies the implementation of different graphical user interfaces. Concerning the last point, the JMT suite offers two simulation interfaces, called JSIMwiz and JSIM graph.

Analysis of Simulation Results

JSIMengine supports the estimation of several reward measures that describe the performance of the simulation models. It supports system-level aggregates like

the total number of jobs in the simulation model or within a finite capacity region. Simulation results are analyzed using transient detection and confidence interval estimation algorithms.After completing the transient detection phase, the statistical analyzer runs the spectral method of Heidelberger and Welch , which is a stable and computationally efficient method for computing simulation confidence intervals using variable batch sizes and a fixed amount of memory.

The spectral analysis proceeds as follows. Given a stationary sequence of samplesX(1), . . . , X(N).

Of a performance measure X, the spectral analysis obtains an estimate of the power spectrum p(f) at the frequency = 0. The value p(0)/N is an unbiased estimator of the sample variance, which can be used to generate confidence intervals on X also if the sequence of sample is correlated.

Control of Simulation ExperimentsHB

JSIMengine offers several options for controlling simulation experiments. Besides defining long-run simulations, the most interesting features are the possibility of executing an automatic stop as a function of the quality of confidence interval estimates and to run parametrically a sequence of "what-if" experiments.

The automatic stop is based on the concept of maximum relative error of the confidence interval and stands for maximum acceptable ratio ε between the half-width of the confidence interval measure α and the estimated mean.

For instance, setting ε = 0.05 and α = 0.05 imposes a simulation stop when the half-width of the 95% confidence interval is no more than 5% of the sample mean.

Parametrical experiments can be controlled by changing the value of the input traffic rate, of the number of jobs for closed classes, of the percentage of jobs belonging to a closed class, of the mean service times, or for different initialization seeds of the random number generator.A screenshot of awhat-if plot is given in the right dialog in Figure 3.

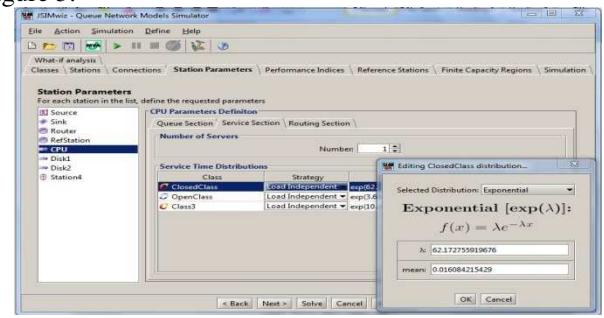


Fig.3-JSIMwiz - Wizard-guided analysis

When an analytical model saved with JMVA is opened in JSIMgraph,

The graphical user interface creates a graphical arrangement of the model based on automatic layout techniques and arbitrarily assigns service distributions to

the stations consistently with the product-form assumptions and with the mean service demands specified in JMVA (see Figure 4).

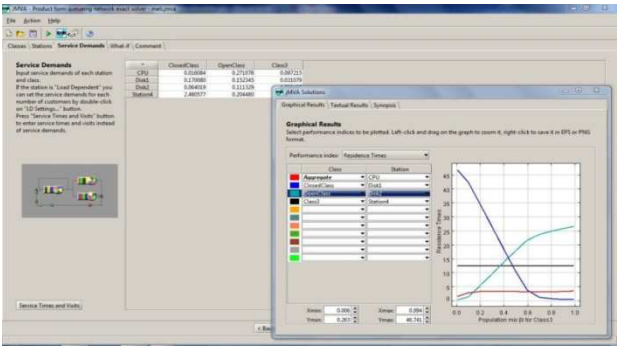


Fig.4- JMVA - What if analysis

JWAT

The Java Workload Analyzer Tool (JWAT) is an application for exploratory analysis of performance data and for generation of static and dynamic workload characterizations that can be used within JSIM, JABA, and JMVA models.

This starts with a static analysis phase, and then moves to a dynamic analysis in which time-varying properties of the trace, such as burstiness, are assessed.After selecting the metrics of interests, the user can apply further filtering and transformations to the data guided by graphical aids such as cross-correlation plots, histograms, scatter-plots, and QQ-plot diagrams.

This stage of the analysis is followed by cluster analysis to partition the workload into representative classes. These classes can be used directly in JSIM, JABA and JMVA models as work-load classes. Clustering results can be inspected for both k-means and fuzzy k-means using the graphical interface, (see Figure 5).

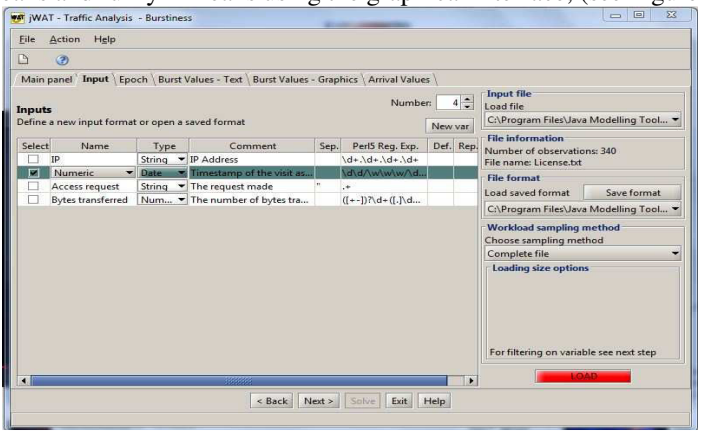


Fig5- JWAT - Traffic Burstiness Analysis

JMCH (Markov Chain)

JMCH provides a graphical representation of the states of the Markov Chain corresponding to a single station models, with limited (M/M/1/k) or infinite queue size (M/M/1), and multiple server stations with n servers (M/M/c) and limited queue size k (M/M/c/k). It is possible to change the arrival rate λ and service time S of the station at simulation run time .If the model is of limited capacity, the size k of the queue can also be changed dynamically. Users may have the visual perception of the bursts of traffic and their effects on the queue length and on the utilization of the server's. The exact analytical results are also shown for comparison purposes.

Output Results

According to the parameters selected (type of station, number of servers, max capacity, arrival rate, service times) the following performance indices are analytically computed and displayed:

- mean number of customers in the station (either in queue and in the servers)
- throughput (that coincides with the arrival rate)
- utilization
- mean response time
- probability of the states of the Markov Chain.

The pane Simulation Results consists of two possible views States and Log. The first view, called States displays the probabilities of the statesand the migration among them with animation obtained through simulation.The second view is the Log screen which shows the data:

- Cust.ID: the unique id assigned to each customer by a counter.
- Arrival Time: shows the instant of time of the creation of the customer.
- Start Execution: the instant of time in which the customer start to be executed from one of the servers.
- Server ID: it shows the id of the server if there are more than one.
- Exit system: the instant of time in which the customer exits from the system.

If the simulation is set with limited number of customers, then this area shows the customers remaining. The user should set the parameters before starting the simulation |(see Figure 6).

Logger

Logging of events can also be obtained from JMCH Simulation. If a user wants to see the arrival time, the execution time, the server id where the customer is being served for each customer should be provided. The Figure 7 shows a logger location and also you can select how many customer you want to simulate and the Figure 8 shows a sample of log output. The log file can be found in the specified location.

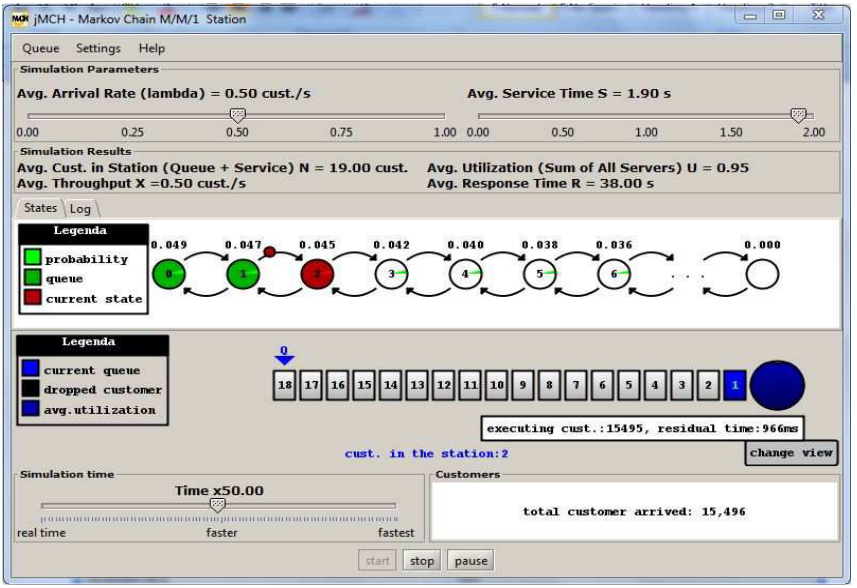


Fig. 6- JMCH Simulation running

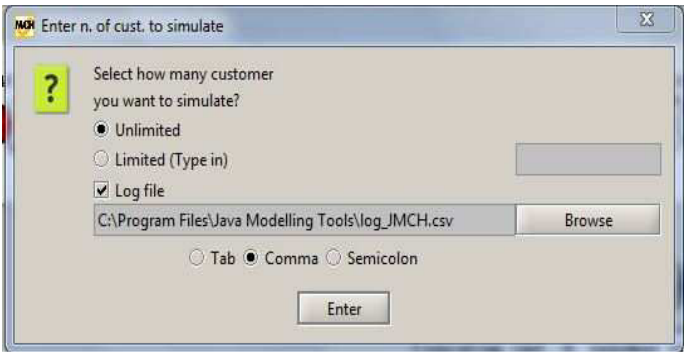
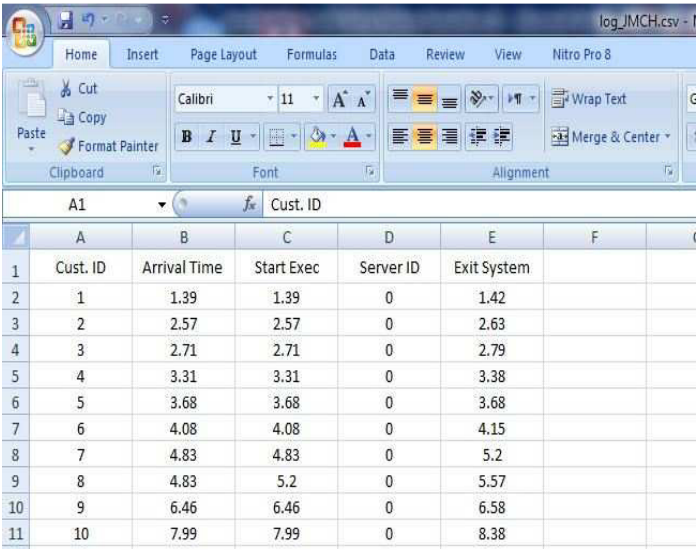


Fig. 7- JMCH Logger location



log_JMCH.csv

Home | Insert | Page Layout | Formulas | Data | Review | View | Nitro Pro 8

Cut, Copy, Format Painter, Paste — Clipboard
Calibri, 11 — Font
Wrap Text, Merge & Center — Alignment

A1 | Cust. ID

	A	B	C	D	E	F
1	Cust. ID	Arrival Time	Start Exec	Server ID	Exit System	
2	1	1.39	1.39	0	1.42	
3	2	2.57	2.57	0	2.63	
4	3	2.71	2.71	0	2.79	
5	4	3.31	3.31	0	3.38	
6	5	3.68	3.68	0	3.68	
7	6	4.08	4.08	0	4.15	
8	7	4.83	4.83	0	5.2	
9	8	4.83	5.2	0	5.57	
10	9	6.46	6.46	0	6.58	
11	10	7.99	7.99	0	8.38	

Fig.8- JMCH Log output

Matrix Inversion Algorithm

Markov chain can be applied in matrix inversion algorithm.Real parallel applications are analyzed by means of a queuing network model (QN).This application addresses the problem of inverting diagonally-dominant matrices (see Figure 9).

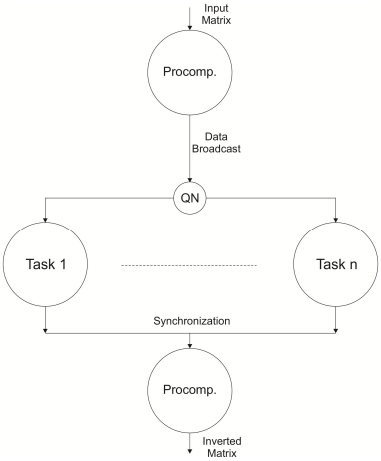


Fig.9- Task graph model of the application.

Conclusions

Java Modeling Tools (JMT) is an integrated environment for workload characterization and performance evaluation based on queuing models. This paper has summarized the features of the main applications that compose the suite, andhas provided intuition on the versatility of JMT in dealing with the different aspects of the performance evaluation and optimization process.

References

1. G. Balbo and G. Serazzi. Asymptotic analysis of multiclass closedqueuingnetworks: Multiple bottlenecks.Perf. Eval.,30(3):115{152, 1997.

2. J. Banks and J. Carson.Discrete-Event System Simulation. Prentice-Hall, Inc., New Jersey, USA, 1984.

3. M. Bertoli, G. Casale, and G. Serazzi. Java modeling tools: an opensourcesuite for queuing network modeling andworkload analysis.In Proceedings of QEST 2006 Conference, pages 119{120, Riverside, US, Sep 2006. IEEE Press.

4. R. Braden, D. Clark, and S. Shenker. Integrated services in the Internetarchitecture: an overview. Internet Request forComment RFC 1633, InternetEngineering Task Force, June 1994.

5. J. C. Browne, K. M. Chandy, R. M. Brown, T. W. Keller, D. F. Towsley, and C. W. Dissly. Hierarchical techniques for the development of realisticmodels of complex computer systems. In Proceedings of the IEEE, volume 63, pages 966{975, 1975.

6. K. M. Chandy, U. Herzog, and L. Woo. Parametric analysis of queuing networks. IBM J. Res. Dev,19(1):36{42, 1975.

7. P. Cremonesi and G. Serazzi.End-to-end performance web services. In Performance 2002. LCNS 1786, Springer, 2002.

8. G. S. Fishman. Statistical analysis for queuing simulations. Management Science, Series A (Theory), 20, 3:363{369,1973.

9. D. Flanagan. Java in a Nutshell. O'Reilly and Associates, Sebastopol, CA, 1997.

10. A. V. Gafarian, C. J. Ancker, and T. Morisaku. Evaluation of commonly used rules for detecting "steady state" incomputer simulation. Naval Research Logistics Quarterly, vol. 25, 511-530, 1978.