

# Mercury: An Integrated Environment for Performance and Dependability Evaluation of General Systems

Bruno Silva<sup>\*</sup>, Rubens Matos<sup>†</sup>, Gustavo Callou<sup>‡</sup>, Jair Figueiredo<sup>†</sup>, Danilo Oliveira<sup>†</sup>, Joao Ferreira<sup>†</sup>, Jamilson Dantas<sup>†</sup>, Aleciano Lobo<sup>†</sup>, Vandi Alves<sup>†</sup> and Paulo Maciel<sup>†</sup>

<sup>\*</sup>Academic Unit of Serra Talhada, Federal Rural University of Pernambuco,  
Serra Talhada, Brazil, Email: bs@cin.ufpe.br

<sup>‡</sup>Informatics Center, Federal University of Pernambuco, Recife, Brazil  
{rsmj, jjcf, dmo4, jfsj3, jrd, aflj, valn, prmm}@cin.ufpe.br

<sup>†</sup>Department of Statistics and Informatics, Federal Rural University of Pernambuco,  
Recife, Brazil, Email: gustavo@deinfo.ufrpe.br

**Abstract**—The evaluation of dependability or performance of general systems is not a trivial task. Therefore, the assistance of software tools to obtain the wanted metrics is of utmost importance. This paper introduces the Mercury environment, which is an integrated software that enables creating and evaluating Reliability Block Diagrams, Stochastic Petri Nets, Continuous Time Markov Chains, and Energy Flow Models. Mercury provides graphical user interface for these modeling formalisms and a script language that allows using it through command-line interface and also integration with external applications. The set of features available in the Mercury tool make it helpful for dependability and performance evaluation of various systems in both academy and industry scenarios.

## I. INTRODUCTION

Software packages for creation and evaluation of dependability and performance models are essential tools for infrastructure designers, since they provide support to dealing with the complexity of each formalism. Generally they provide a user-friendly graphical user interface (GUI), and an evaluation engine to obtain metrics from the model. Each software has its strengths and drawbacks, and can be specialized in a specific formalism, or support different types of models.

Mercury tool [1] was developed by MODCS research group to allow the creation and evaluation of performance and dependability models. The proposed environment can be adopted as a modeling tool for the following formalisms: Continuous Time Markov Chains (CTMC) [2], Reliability Block Diagrams (RBD) [3], Energy Flow Models (EFM) [4], and Stochastic Petri Nets (SPN) [5]. Mercury offers useful features that are not easily found in other tools, such as: more than 25 probability distributions supported in SPN simulation, sensitivity analysis of CTMC and RBD models, computation of reliability importance indices, and moment matching of empirical data. The proposed environment has been adopted by MODCS Research Group. The tool's web [1] page presents the papers (e.g., [6]) that adopted Mercury as evaluation engine.

There are many academic and commercial software tools for dependability and performance evaluation of general systems. BlockSim [7] allows system modeling by using RBD and Fault Trees [2], or even combination of both modeling

techniques, but do not use state-based models (e.g. SPNs) which makes it harder to deal with dependencies. Relex [8] and Sharpe [9] provide dependability and performance evaluation through the adoption of RBD, SPN, and CTMC models, just like the proposed tool. However, the later tool obtains SPN metrics only by numerical analysis. This method is unsuitable when the modeled system is too complex and large [10]. The proposed solution also provides simulation as a more comprehensive technique for finding metrics of SPNs. Different from previous works, Mercury presents RBD, SPN, CTMC, and EFM modelling for calculation of dependability and performance in a single environment.

This paper explores the characteristics of Mercury tool, and is organized as follows. Section II describes the features for each modeling formalism supported by Mercury. Section III presents the scripting language which enable textual representation and evaluation of models as well as interaction through command-line interface (CLI). Section IV describes a case study that uses Mercury tool to solve an availability model for a cloud system. Section V draws the final remarks on this tool.

## II. TOOL OVERVIEW

This section presents the main features available on Mercury for all supported modeling formalisms.

### *SPN editor and evaluator.*

Regarding SPN models, Mercury environment allows dependability/performance evaluation utilizing simulation and numerical analysis techniques [10], [11]. Time-dependent metrics are obtained through transient evaluation, while steady-state metrics are result of stationary evaluation. Figure 1 depicts the SPN editor. To assist the validation of SPN models, the editor has a feature, namely, token game, which makes possible the firing of transitions graphically and interactively according to the current net marking. By adopting Mercury tool, users can evaluate structural properties of the adopted stochastic Petri net through the analysis of place invariants, siphons and traps [12].

### *CTMC editor and evaluator.*

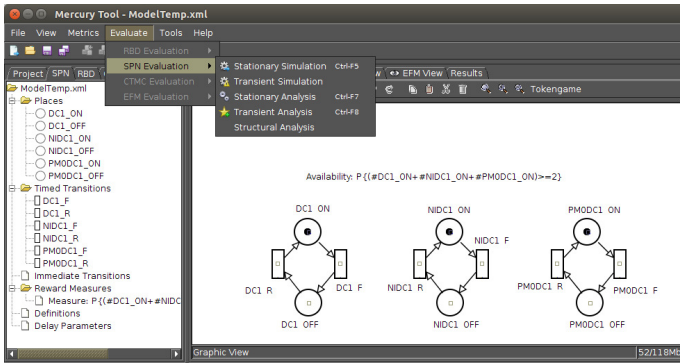


Fig. 1. Mercury environment - SPN editor and evaluator

Mercury provides a panel in which users can create, edit, and evaluate CTMC models (Figure 2). Representative numerical techniques are available for stationary analysis: GTH [13] and Gauss-Seidel [2]) as well as transient analysis (i.e., Uniformization, Runge-kutta and Trapezoid-Euler [2]). For models that have absorbing states, Mercury also enables computing probability of absorption and mean time to absorption. The user may create custom metrics with expressions that include state probabilities. Transition rates can be defined by means of algebraic expressions using symbolic parameters. Parameter names can include greek letters. Parameters and metrics are easily viewed and modified in the CTMC editor. Mercury is also able to compute the sensitivity of state probabilities to each input parameter.

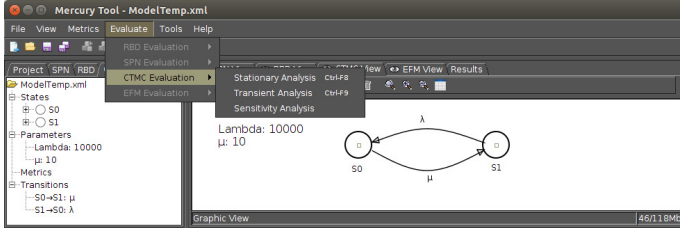


Fig. 2. Mercury environment - CTMC editor and evaluator

### RBD editor and evaluator.

The Mercury tool RBD editor and evaluator allows reliability and availability analysis using block diagrams. The types of reliability configurations supported by the RBD editor and evaluator are: Series, Parallel, K out of N, and Bridge. In this editor, a diagram contains only one input and one output node. RBDs provide closed-form equations, so the results are usually obtained faster than using other methods, such as SPN simulation. However, there are many situations (e.g., dependency among components) in which modelling using RBD is harder than adopting SPN. Figure 3 shows an RBD view of the tool, in this configuration some components are in series and parallel arrangement.

Additionally, the RBD evaluator allows the calculation of reliability importance, structural and logical functions as well as bounds of dependability measures. Reliability Importance (RI) is a technique that allows to quantify the improvement in system reliability due to a component, when the respective reliability is increased by one unit. Structural and logical functions

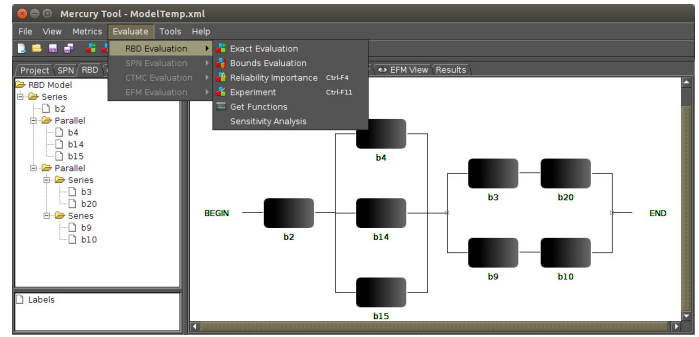


Fig. 3. Mercury environment - RBD editor and evaluator

are alternative ways of representing the system mathematically, in which the former adopts algebraic function and the latter adopts logical expression. RBD evaluation of dependability bounds [14] is a method to calculate dependability metrics (e.g., reliability) when the RBD model is too large. In this case, such a technique can provide approximations for the exact metric faster than solving all closed-form equations precisely. As the bounds are calculated iteratively, Mercury environment shows the estimated bounds according to the user parameters and, also, all intermediate computed values.

### EFM Editor.

The Energy Flow Model (EFM) is proposed to evaluate the cost, sustainability and availability of datacenter (or clouds) power and cooling infrastructures, respecting the power constraints of each device.

Figure 4 depicts the EFM editor and evaluator. The highlighted rectangle shows the set of evaluations available in the Mercury tool, where:

- **Cost Evaluation** allows designers to evaluate acquisition, operational and total costs;
- **Exergy Evaluation** represents the sustainability evaluation through the exergy metric. Exergy enables one to estimate energetic efficiency of a system;
- **Energy Flow Evaluation** Computes the energy that flows through each device respecting the power constraints of each device;
- **Combined Evaluation** provides an integrated quantification of sustainability impact, cost and dependability of data center (or cloud) power and cooling infrastructures [4];
- **Flow Optimization (GRASP, PLDA, PLDAD)** is able to evaluate RBD, SPN, CTMC and EFM models by calling the Mercury engine. A GRASP [15] based algorithm, a power load distribution algorithm (PLDA) [16] and a power load distribution algorithm depth (PLDAD) represent three optimization techniques that were implemented in that module.

Figure 4 shows an example of the EFM. The rectangles equates to the type of equipment, and the labels represent each device name. The edges have weights that are used to direct the energy flow between the components. The EFM computes

the overall energy required to provide the necessary energy at the target point.

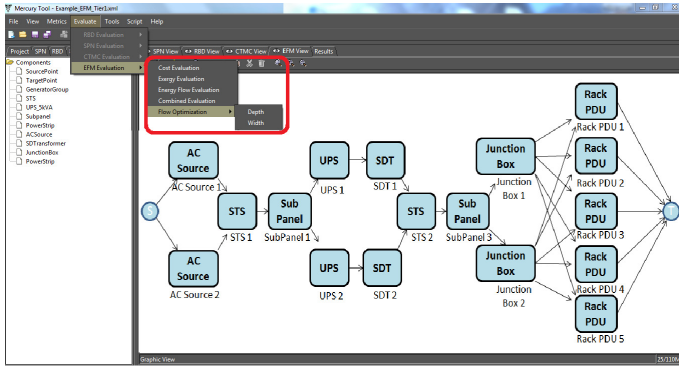


Fig. 4. EFM Editor and Evaluator

### Auxiliary Tools.

Mercury environment has an additional tool responsible for generating random numbers, namely, random variate generator, which contemplates several probability distributions. Statistical summaries are also provided for the generated numbers (e.g., standard deviation) and the data can be exported to files in order to be utilized for external applications. Moreover, the generator is utilized by the SPN simulation, in which the user can choose different probability distributions associated to transitions.

Another important auxiliary tool presented in Mercury environment is the moment matching [17] wizard. By adopting this tool, users can estimate what exponential-based probability distribution best fits the mean (first moment) and standard deviation (second moment) of user data for a given model parameter.

## III. MERCURY LANGUAGE

One feature introduced on version 4.4 of Mercury tool is a scripting language for creating and evaluating models. The scripts can be executed by command line interface (CLI) or via graphical interface. The main objective of the scripting language is to allow the use of Mercury evaluation engine with greater flexibility than the GUI provides. The scripting language enables, for example, using shell scripts to automate an evaluation workflow. The scripting language offers an additional advantage that is the increased support to hierarchical evaluation [18] [19]. Input parameters of any model can be defined as function of an output metric defined by another model, independent of the modeling formalism. The Mercury scripting language currently supports SPN, RBD, and CTMC models.

In the Listing 1, we show an example of script in Mercury language. A script has the following structure: first, we define a set of models, each one with its parameters and metrics, and after the models we declare a main section. In the main section we can set the input parameter values, evaluate the models and print the results. The example of listing 1 shows a CTMC definition and evaluation. This model was originally presented in [18], and represents a cluster controller for a private cloud platform with active/passive redundancy scheme. Inside the

model we declare a set of states, with the reserved word *state* followed by the state name, plus an optional *up* keyword. The *up* keyword denotes a state of the system when it is functioning normally, for availability evaluation. Next, we declare the set of transitions between the states, with the *transition* keyword. For the transition rate, we can set a fixed value, use an algebraic expression or use variables that can be assigned when the model is evaluated. Next, we define the metrics of the model, that are the results that we want to compute. The scripting language allows computing the following metrics for CTMCs: stationary availability, stationary probability for a state, and steady-state reward rate, through explicit assignment of rewards to specific states<sup>1</sup>. Finally, we define the main section, where we set the input parameters values, evaluate the model, and write the results on screen.

Listing 1. Mercury script for a CTMC model

```
markov RedundantGC{
  state fu up;
  state fw;
  state ff;
  state uf up;
  state uw up;

  transition fw -> fu(rate = sa_s2);
  transition fu -> ff(rate = lambda_s2);
  transition ff -> uf(rate = mu_s1);
  transition uf -> uw(rate = mu_s2);
  transition uw -> fw(rate = lambda_s1);

  transition fw -> uw(rate = mu_s1);
  transition uw -> uf(rate = lambda_s2);
  transition uf -> ff(rate = lambda_s1);

  transition fw -> ff(rate = lambda_s2);
  transition fu -> uw(rate = mu_s1);

  metric aval = availability;
  metric aval2 = reward( fu = 1, uf = 1, uw = 1);
  metric p1 = stationaryProbability( st = fw );
  metric p2 = stationaryProbability( st = ff );
}

main{
  lambda_s1 = 1/180.72;
  mu_s1 = 1/0.966902;
  mu_s2 = 1/0.966902;
  lambda_s2 = 1/216.865;
  lambda_s2 = 1/180.721;
  sa_s2 = 1/0.005555555;

  a1 = solve( model = RedundantGC, metric = aval );
  println(a1);

  a2 = solve( model = RedundantGC, metric = aval2 );
  println(a2);
}
```

## IV. CASE STUDY

This section presents a case study to demonstrate the feasibility of adopting the Mercury tool to create and evaluate models for some proposed architectures. We evaluate a simple cloud computing architecture originally presented in [18]. The architecture implements a redundant mechanism in the main component following a warm-standby strategy.

The hierarchical heterogeneous model [18] of this study comprises RBDs and a CTMC. The RBD describes the high-level components, whereas the CTMC represents the components involved in the redundancy mechanism. This case study

<sup>1</sup>It is important to stress that other metrics (e.g., state probabilities at a given point in time) can be computed via GUI

considers the example of an architecture with five nodes, where at least one node must be available for the cloud to work properly. Therefore the Nodes subsystem is represented by a set of five node blocks in parallel.

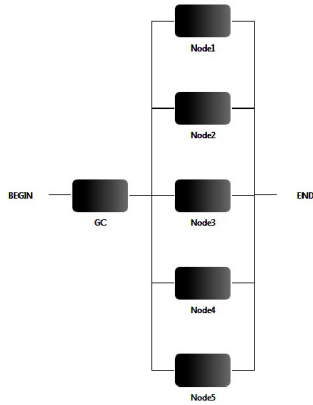


Fig. 5. RBD model for the non-redundant cloud system [18]

Figure 5 shows the complete RBD model with one (non-redundant) General Controller and its five nodes. It is important to stress that a warm-standby replication strategy cannot be properly represented by RBD models, due to dependency between states of components, therefore another model is used to represent a redundant architecture, with one General Controller active and one replicated GC host in warm-standby. So the redundant General Controller subsystem is represented by a CTMC, shown in Figure 6.

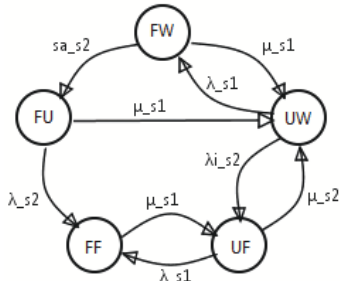


Fig. 6. CTMC Model to redundant system with two hosts [18]

Mercury tool was used to create the models and compute the availability measures of cloud system with and without redundant general controller. Table I presents the measures obtained with Mercury.

TABLE I. AVAILABILITY MEASURES OF THE SYSTEM WITH AND WITHOUT REDUNDANCY

Measure	GC without redundancy	GC with redundancy
Steady-state availability	0.9946782	0.9999179
Number of 9's	2.27394	4.08581
Annual downtime	46.66 h	0.72 h

## V. FINAL REMARKS

This paper presented the main features of Mercury tool and demonstrated the feasibility of using it for analyzing a cloud computing system. Mercury tool has been used in several research projects which had the results published in

peer-reviewed journals and conferences. The MODCS research group is constantly improving the tool, by adding new features and updating existing functions. New versions are released every six months approximately, but this is a flexible policy adjusted according to some variables. This modeling tool is helpful for academy and industry efforts on capacity planning of many kinds of systems through performance evaluation, dependability prediction through reliability and availability evaluation, and data-center energy flow planning.

## REFERENCES

- [1] B. Silva, "Mercury tool," [http://www.modcs.org/?page\\_id=1397](http://www.modcs.org/?page_id=1397), accessed: 2015-01-05.
- [2] K. Trivedi, *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*, 2nd ed. Wiley Interscience Publication, 2002.
- [3] P. Maciel, K. S. Trivedi, R. Matias, and D. S. Kim, *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*, ser. Premier Reference Source. Igi Global, 2011, ch. Dependability Modeling.
- [4] G. Callou, P. Maciel, D. Tutsch, J. Ferreira, J. Araújo, and R. Souza, "Estimating sustainability impact of high dependable data centers: A comparative study between brazilian and us energy mixes," *Computing*, vol. 95, no. 12, pp. 1137–1170, 2013.
- [5] M. Molloy, "Performance Evaluation Using Stochastic Petri Nets," *IEEE Trans. Comput.*, vol. 31, no. 9, pp. 913–17, 1982.
- [6] B. Silva, P. Maciel, J. Brilhante, and A. Zimmermann, "Geoclouds modes: A performability evaluation tool for disaster tolerant iaas clouds," in *Systems Conference (SysCon), 2014 8th Annual IEEE*. IEEE, 2014, pp. 116–122.
- [7] "Blocksim tool," <http://www.reliasoft.com/BlockSim>.
- [8] "Relex tool," <http://www.relex.com>.
- [9] A. R. Sahner and S. K. Trivedi, "Reliability modeling using sharpe," Durham, NC, USA, Tech. Rep., 1986.
- [10] G. Balbo and G. Chiola, "Stochastic petri net simulation," in *WSC '89: Proceedings of the 21st conference on Winter simulation*. New York, NY, USA: ACM, 1989, pp. 266–276.
- [11] G. Balbo, "Introduction to stochastic petri nets," *Lectures on Formal Methods and Performance Analysis: First EEF/Euro Summer School on Trends in Computer Science, Berg en Dal, The Netherlands, July 3-7, 2000: Revised Lectures*, 2001.
- [12] M. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, "Modelling with Generalized Stochastic Petri Nets," *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, no. 2, 1998.
- [13] W. K. Grassmann, M. I. Taksar, and D. P. Heyman, "Regenerative analysis and steady state distributions for markov chains," *Operations Research*, vol. 33, no. 5, pp. 1107 – 1116, 1985.
- [14] W. Kuo and M. J. Zuo, *Optimal Reliability Modeling - Principles and Applications*. Wiley, 2003.
- [15] G. Callou, J. Ferreira, P. Maciel, D. Tutsch, and R. Souza, "An integrated modeling approach to evaluate and optimize data center sustainability, dependability and cost," *Energies*, vol. 7, no. 1, pp. 238–277, 2014.
- [16] J. Ferreira, G. Callou, and P. Maciel, "A power load distribution algorithm to optimize data center electrical flow," *Energies*, vol. 6, no. 7, pp. 3422–3443, 2013.
- [17] A. Desrochers and R. Al-Jaar, *Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis*. IEEE Press, 1995.
- [18] J. Dantas, R. Matos, J. Araujo, and P. Maciel, "Eucalyptus-based private clouds: availability modeling and comparison to the cost of a public cloud," *Computing*, pp. 1–20, 2015.
- [19] R. Matos, J. Araujo, D. Oliveira, P. Maciel, and K. Trivedi, "Sensitivity analysis of a hierarchical model of mobile cloud computing," *Simulation Modelling Practice and Theory*, vol. 50, no. 0, pp. 151 – 164, 2015, special Issue on Resource Management in Mobile Clouds.