Automated Reliability Analysis in Fog Computing Environment

Marco Mialaret
matmj@cin.ufpe.br
Faculdade Senac
Centro de Informática, UFPE
Recife, PE, Brazil

Thiago Pinheiro
tfs3@cin.ufpe.br
Centro de Informática, UFPE
Recife, PE, Brazil

Antonio Neto
antonio.neto@paulista.ifpe.edu.br
Instituto Federal de Pernambuco,
Campus Paulista
Paulista, PE, Brazil

Pablo Philipe Pessoa
ppp2@cin.ufpe.br
Centro de Informática, UFPE
Recife, PE, Brazil

Paulo Pereira
paulo.pereira@ifpb.edu.br
Instituto Federal da Paraíba
João Pessoa, PB, Brazil

Jamilson Ramalho Dantas
jrd@cin.ufpe.br
Centro de Informática, UFPE
Recife, PE, Brazil

Paulo Romero Martins Maciel
prmm@cin.ufpe.br
Centro de Informática, UFPE
Recife, PE, Brazil

ABSTRACT

This paper introduces the EMA (Expectation Maximization Algorithm) tool, an automated solution designed to fit Hyper-Erlang distributions for reliability analysis, particularly in fog computing environments. Traditional methods for evaluating Time Failure (TTF) distributions are time consuming and do not scale well in complex systems. The EMA tool addresses this limitation by automating the fitting process, significantly reducing computational time. Applying a resampling method across 100 scenarios, the EMA tool captures system variability and efficiently generates key metrics, such as Probability Density Functions (PDFs) and Cumulative Distribution Functions (CDFs). The tool estimates Hyper-Erlang distribution parameters for each scenario and derives critical reliability metrics, including Mean Time to Absorption (MTTA). The fitting process for the 100 scenarios was completed in just 8 minutes and 22 seconds, and subsequent reliability analysis took 24 and 35 seconds. The EMA tool provides essential information on system performance under various conditions, improving the scalability and efficiency of reliability analysis in distributed environments.

1 INTRODUCTION

Fog computing, as an extension of the Internet of Things (IoT) architecture, enhances data processing by bringing computation closer to the edge of the network, reducing latency between devices and the cloud. In such distributed systems, reliability is critical due to the unpredictable nature of failures, dynamic workloads, and the need for continuous availability. Traditional reliability assessment methods, such as direct measurements or simulations, are often laborious and do not scale effectively for large or complex systems. This creates a pressing need for automated tools to streamline the analysis process while providing comprehensive system insights.

Several existing tools attempt to address the challenge of automating reliability analysis. Tools such as HyperStar provide a framework for fitting phase-type (PH) distributions to system failure data, generating reliability metrics based on Markov chains [17] [9] [10]. However, limitations such as proprietary code and restricted flexibility hinder wider adoption and refinement by the research community. BuTools [5], another notable tool, supports traffic modeling and queue analysis for PH distributions but focuses primarily on Markovian processes, lacking flexibility in handling complex failure scenarios.

To overcome these limitations, this paper introduces the EMA tool, which automates the fitting of Hyper-Erlang distributions for system reliability analysis. Hyper-Erlang distributions are particularly effective in capturing the TTF behavior of systems with multiple failure phases. Unlike HyperStar and BuTools, which focus on specific aspects of reliability modeling, the EMA tool offers a fully automated and scalable solution that can efficiently fit distributions to observed data, even when the data are incomplete or latent. This automation allows for rapid, large-scale system reliability analysis, crucial for modern, complex, distributed systems.

The EMA tool meets the growing demand for scalable reliability analysis, handling various failure patterns and system behaviors. This paper presents a case study that applies the EMA tool in a fog computing environment, demonstrating its ability to generate essential reliability metrics such as PDFs, CDFs, and MTTA. By automating these tasks, the tool significantly reduces computational burden, enabling the analysis of large data sets of system failures in a fraction of the time typically required.

This paper details the use of the EMA tool in analyzing 100 resampled failure scenarios from a fog computing infrastructure. The tool's ability to automatically fit Hyper-Erlang distributions and generate reliability functions for each scenario provides critical insights into the system's performance across various conditions. The EMA tool improves system understanding by identifying the best- and worst-case reliability scenarios, providing data-driven insights for system optimization and decision making.

The paper is organized as follows: Section 2 provides an overview of PH distributions and the Expectation-Maximization (EM) algorithm, which forms the foundation of the EMA tool. Section 3 reviews related work in the field of reliability analysis, with a focus on tools and methodologies for automating the fitting process. Section 4 describes the technical details of the EMA tool and its integration into the Mercury modeling suite. Section 5.1 presents the experimental setup, where the EMA tool was applied to analyze failure data.

Section 5.2 discusses the results of the 100 resampled scenarios, highlighting the efficiency and insights offered by the EMA tool. Finally, Section 6 offers concluding remarks and explores future directions for developing automated reliability analysis tools like EMA.

2 HYPER-ERLANG DISTRIBUTION AND FITTING PROCESS

The probability density function of the Erlang distribution, which forms the basis of the Hyper-Erlang distribution, is given by:

$$f(t;k,\lambda) = \lambda^k \frac{t^{k-1}e^{-\lambda t}}{(k-1)!}, \tag{1}$$

where λ is the rate parameter and k is the number of phases.
The probability density function for the mixed distribution is given by:

$$f_T(t) = \sum_{i=1}^{n} \alpha_i f_{T_i}, \tag{2}$$

where α_i represents the weights of each component in the mixture. To ensure a balance between model accuracy and complexity, the Bayesian Information Criterion (BIC) is employed during the fitting process. The BIC penalizes models with too many parameters, reducing the risk of overfitting while maintaining a good fit for the data. This makes BIC particularly useful for selecting appropriate models in large-scale reliability analysis.
The BIC is defined as:

$$\text{BIC} = -2\ln(L) + k\ln(n), \tag{3}$$

where L is the likelihood of the model given the data, k is the number of parameters in the model, and n is the number of data points. Using the BIC ensures that the selected model strikes a balance between fitting the data well and maintaining a simple structure. This approach was employed in [11], demonstrating how the BIC can automatically determine the optimal number of branches and phases in Hyper-Erlang distributions.
The fitting process begins by assigning initial values for branches and phases, and the EM algorithm refines these values based on likelihood maximization. At each iteration, new parameter sets are evaluated until the BIC shows no further significant improvements. Convergence of the BIC indicates that the obtained parameters minimize the cost function, providing an appropriate balance between model accuracy and simplicity.

3 RELATED WORKS

PH distributions emerged as a flexible approach to model complex time-dependent events in various fields, addressing limitations in conventional distributions. Foundational algorithms like EMpht [3] and EM [1] laid the groundwork, establishing methodologies that paved the way for modern tools in phase-type distribution analysis. Building on these contributions, recent tools like HyperStar [17, 18] and BuTools [5, 6] expanded the accessibility and capability of PH distributions. HyperStar, for instance, introduced a user-friendly interface and broad export compatibility, facilitating its application in diverse modeling contexts. Its design allows PH distributions to approximate almost any non-negative distribution, offering critical benefits such as closed-form metrics and Markov chain representations for simplified analytical evaluations.
BuTools extended PH distribution application with tools for traffic modeling, queue analysis, and empirical trace analysis. Its functionality encompasses density calculations, moment matching, and support for Markovian Arrival Processes (MAPs), with adaptability across Matlab, Mathematica, and Python/NumPy. However, BuTools has faced processing time challenges and has not received updates since 2015, indicating potential areas for advancement.
In response to these evolving needs, we developed the EMA tool, presented in [11]. EMA simplifies PH distribution fitting by automating parameter selection, integrating user-friendly features, and improving modeling accuracy. Leveraging the Bayesian Information Criterion (BIC) with a Bayesian optimization framework and Expectation-Maximization (EM) algorithm, EMA enables precise PDF derivation for system performance analysis, offering a scalable solution for complex reliability assessments.
Various methods have been developed to enhance the estimation of PDFs and CDFs for assessing system performance and reliability. Thümmler et al. [21] demonstrated that mixtures of Erlang distributions could accurately represent complex time patterns in real-world scenarios, providing a flexible model for failure or service times with significant variability. In 2009, Horvath et al. [4] advanced these approaches by introducing a canonical representation for third- and fourth-order PH distributions, emphasizing parameter minimization to improve computational efficiency in reliability modeling. Building on this, Okamura et al. [12] provided a comprehensive guide to PH distributions in 2016, detailing parameter estimation with the EM algorithm and applications in reliability analysis while noting limitations like extended execution times. Together, these foundational ideas contributed to developing the EMA tool by inspiring techniques to streamline parameter estimation and improve modeling accuracy in complex reliability scenarios.
Studies highlight practical modeling challenges that EMA seeks to address. In 2017, Prados et al. [16] proposed a model for open queue systems to estimate response times without assumptions about arrival distributions. Pereira et al. [14] in 2020, applied closed-form equations for web server performance in fog computing environments, relying on Markov chains but requiring considerable manual configuration. EMA addresses these limitations by automating the fitting process and reducing manual intervention, enhancing the usability and scalability of PH distributions in complex modeling tasks.
Recent work by Li et al. [7] underscores ongoing efforts to address predictive maintenance challenges in the Industrial Internet of Things (IIoT). They employ a hyper-Erlang distribution within a hidden semi-Markov model (HSMM) to enhance the accuracy of remaining useful life (RUL) predictions. Their approach models system degradation through a three-state HSMM, using the EM algorithm to estimate parameters and Bayesian updating for real-time warning state probabilities. Tested on gear shaft data, this model outperforms traditional methods, highlighting the hyper-Erlang distribution's potential for precise RUL predictions. This study reflects the continuous need for advanced solutions, as integrated into the EMA tool, to overcome existing limitations in predictive maintenance and system reliability modeling.

4 EMA FITTING TOOL

The EMA fitting tool offers both ease of use and flexibility, catering to users seeking an automated fitting process and experts who prefer fine-tuned control. This balance ensures that the tool is accessible to a wide range of users, from those looking for a straightforward fitting experience to those requiring more advanced customization. The EMA algorithm has been integrated into the Mercury tool (version 5.2 onwards) [8, 19, 20], which supports a variety of models used in reliability and performance evaluation.

Mercury is a comprehensive software suite that facilitates modeling performance, dependability, and energy flow in complex systems. It offers a graphical user interface (GUI) for constructing and evaluating models, such as stochastic Petri nets (SPNs), continuous-time Markov chains (CTMCs), discrete-time Markov chains (DTMCs), reliability block diagrams (RBDs), fault trees (FTs), and energy flow models (EFMs). Developed by the Modeling of Distributed and Concurrent Systems (MoDCS) research group at the Informatics Center (CIn) of the Federal University of Pernambuco (UFPE) since 2009, Mercury has evolved to meet the demands of various users in system modeling and analysis.

The integration of the EMA fitting tool enhances Mercury's capabilities by offering both automated and manual fitting options for Hyper-Erlang distributions. This feature simplifies the process for users who need quick, reliable results, allowing experts to refine the fitting process as needed.

EMA prioritizes user-friendliness, offering both manual and automated fitting options. Users can manually set parameters, but the key feature is its automated adjustment through Random Search. When this method is selected, the user specifies the minimum and maximum numbers of phases and branches (the number of Erlang-n distributions to be combined), along with the maximum number of iterations and repetitions for achieving the lowest BIC. Screens from the tool are shown in 1.

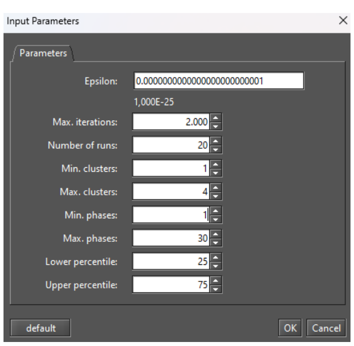

Figure 1: A configuration screen for the automated search algorithm allows users to set the number of phases and branches based on the BIC.

After the algorithm completes, the adjusted parameters are displayed. Users also have the option to generate functions for the PDF, CDF, CCDF, MTTF, Hazard Function, and Mean Residual Life, along with the fitting curve, as shown in 2.

Figure 2: Result screen, generated functions, and the fitting.

5 EXPERIMENTS

This section presents two experiments designed to evaluate the application of the EMA tool for reliability analysis. The first experiment investigates the adoption of the EMA tool for studying TTF distributions in a fog computing environment. The second experiment applies the EMA tool to analyze synthetic failure scenarios generated through resampling, allowing for exploring system behavior under varying conditions.

5.1 Experiment 1: Adoption of EMA for Studying TTF

This experiment aims to characterize the TTF distributions within a fog computing environment by accelerating failure observations using a fault injection technique, as detailed in [2, 13, 15]. The case study evaluates whether the availability models accurately reflect real-world conditions by assessing a baseline infrastructure consisting of two physical nodes—one at the edge and one in the fog—each running a single application. Failure and repair rates were adopted from previous studies, assuming exponential distributions for both time to failure and time to repair, with rates being the inverses of the Mean Time To Failure (MTTF). Table 1 shows the adopted MTTF values.

Table 1: Mean Time To Failure (MTTF) for Components

Category	Component	MTTF (h)
Edge	Raspberry Pi	4767.8
	OS	2880
	Python App	217.8
	Nginx App	217.8
Fog	Hardware	8760
	OS	2880
	Python App	217.8
	Nginx App	217.8

Fault injection is critical for accelerating failure observations to provide controlled insights into how system availability is affected under fault conditions. The edge and fog nodes were configured with minimal components necessary to provide the required service. The edge node included a Raspberry Pi, an operating system, a Python-based face recognition application, and an Nginx RTMP server to handle drone video streaming. The fog node had similar components, with the primary difference being using a personal computer for the fog node's hardware.

Faults were injected into the system infrastructure, and the availability was monitored over a specific observation period. Afterward, TTF data were collected, and the maximum likelihood estimation algorithm was used to fit an Erlang-r distribution. This approach ensured that the failure data accurately reflected the behavior of the system components over time, providing a robust basis for validating the availability models.

The fitted distributions' PDF and CDF were computed, enabling a system reliability study. The relation $F(t) + R(t) = 1$ was used to calculate the reliability, while the mean time to absorption (MTTA) was computed as:

$$MTTA = \int_0^\infty R(t)\, dt. \tag{4}$$

Using the automatic fitting method, the EMA tool generated results that fit the data, as shown in Figure 3. Parameters $(0.00338, 0.012433, 0.046129)$ were identified for the Hyper-Erlang distribution, with the initial probability vector $(0.39, 0.45, 0.16)$. The resulting reliability metrics provided insights into the system's behavior under varying conditions, although further validation of underlying assumptions was necessary.

5.2 Experiment 2: Fitting for Resampled Scenarios

In this experiment, a resampling method was implemented to evaluate the effectiveness of the EMA tool's auto-fitting process. Synthetic failure data were generated based on Weibull distributions for various system components. Weibull distributions are commonly used in reliability engineering due to their flexibility in representing different failure rate behaviors.

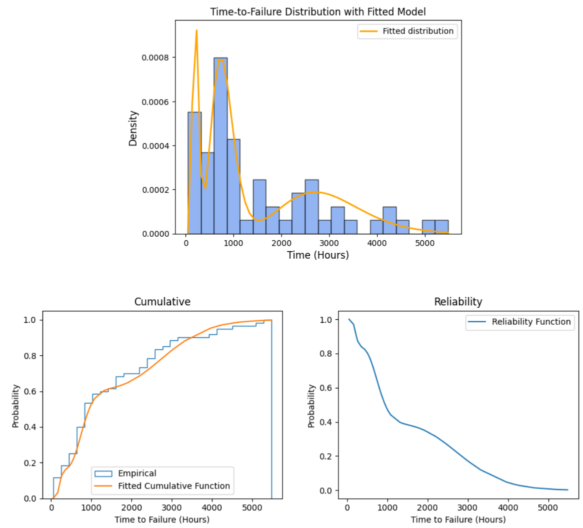

Figure 3: Fitted distribution, PDF, and CDF.

The resampling process began by defining Weibull distribution parameters (shape and scale) for each system component. Random failure times were then generated, with the shortest time identifying the first failed component. This failure time was logged in a structured format, and the remaining failure times were adjusted to account for elapsed time. The process was repeated until a full dataset of failure events was constructed.

This iterative approach ensured that the simulated failure data mirrored the temporal behavior of real system components. Each failure event was logged systematically, allowing for the validation and testing of the EMA tool's fitting capabilities. By generating multiple failure scenarios, the performance of the automated fitting algorithm could be thoroughly evaluated.

The resampling method was applied 100 times, generating scenarios that captured variability in system behavior. For each scenario, the parameters of the Hyper-Erlang distribution were estimated using the Expectation-Maximization (EM) algorithm. The PDF and CDF were computed, and key reliability metrics, such as the mean time to absorption, were calculated. Figure 4 illustrates the generated scenarios and their corresponding adjusted PDFs.

The automated fitting process for the 100 scenarios was completed in 8 minutes and 22 seconds. The PDFs, CDFs, and reliability functions were determined, allowing for detailed analysis of each scenario. Figure 5 shows the reliability functions for all the resampled scenarios, highlighting the highest and lowest reliability cases.

This experiment demonstrated that resampling, combined with automated fitting, allows for efficient analysis of best- and worst-case system behavior, providing key insights into system performance and reliability. The approach can be applied to systems with multiple devices connected to the same network, enhancing the comparison and analysis of system reliability in various scenarios.

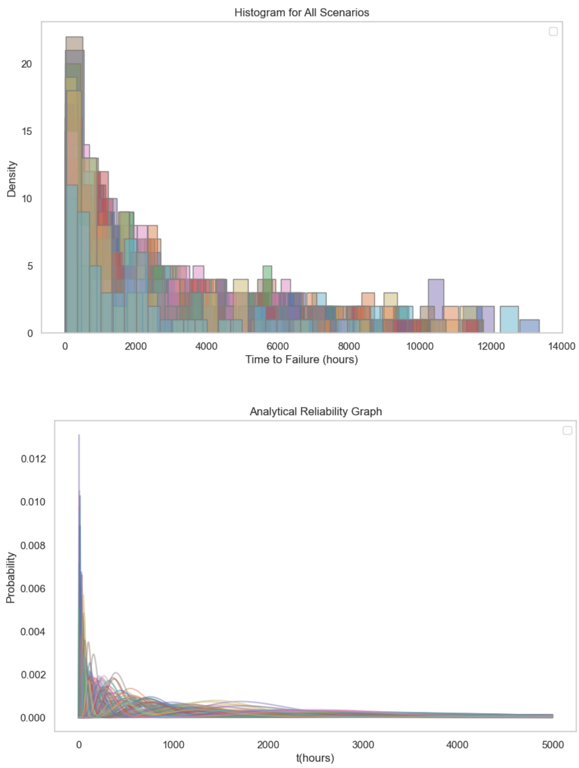

Figure 4: 100 resampling scenarios and respective PDF fittings.

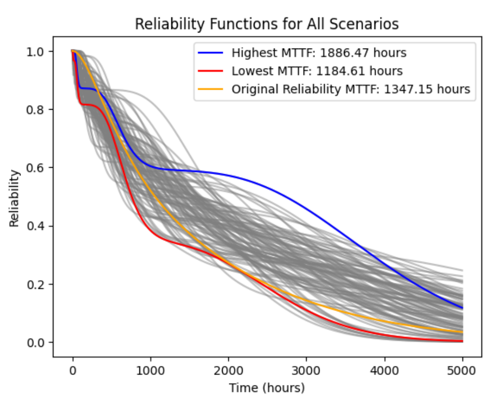

Figure 5: Reliability functions for 100 resampling scenarios, highlighting the highest and lowest reliability cases.

6 CONCLUSION

This study applied Hyper-Erlang distributions for reliability analysis in fog computing environments, leveraging the EM algorithm to generate PDFs and CDFs for characterizing TTF distributions. Using a resampling method, we captured the variability in system behavior, enabling a detailed analysis of system reliability. The automated fitting process, supported by the BIC, accurately determined distribution parameters and facilitated the calculation of key reliability metrics, such as MTTA. The entire process for 100 scenarios was completed in 8 minutes and 22 seconds, highlighting the efficiency of the approach.

The reliability functions derived from the fitted distributions provided insights into the best and worst-case scenarios, enhancing the ability to compare system reliability under different conditions. This methodology can be extended to more complex systems and scenarios, allowing for further exploration of fault injection techniques and their effects on reliability analysis. Based on this approach, future work could further integrate real-time monitoring and predictive maintenance to improve system reliability and performance.

ACKNOWLEDGMENT

We would like to thank the Coordination of Improvement of Higher Education Personnel – CAPES, the National Council for Scientific and Technological Development – CNPq, the Foundation for the Support of Science and Technology of Pernambuco – FACEPE, and the MoDCS Research Group for their support.

REFERENCES

[1] Asmussen, S., Nerman, O., and Olsson, M. Fitting phase-type distributions via the em algorithm. *Scandinavian Journal of Statistics* (1996), 419–441.

[2] Fernandes, S., Tavares, E., Santos, M., Lira, V., and Maciel, P. Dependability assessment of virtualized networks. In *2012 IEEE International Conference on Communications (ICC)* (2012), pp. 2711–2716.

[3] Häggström, O., Nerman, O., and Asmussen, S. *EMPHT: A Program for Fitting Phase-type Distributions*. Chalmers tekniska högskola, 1992.

[4] Horváth, G., and Telek, M. On the canonical representation of phase type distributions. *Performance Evaluation 66*, 8 (2009), 396–409.

[5] Horváth, G., and Telek, M. Butools 2: a rich toolbox for markovian performance evaluation. In *VALUETOOLS* (2016), pp. 135–141.

[6] Horváth, G., and Telek, M. Markovian performance evaluation with butools. In *Systems Modeling: Methodologies and Tools*. Springer, 2019, pp. 253–268.

[7] Li, X., Duan, C., Cai, J., Zuo, H., Liu, Z., and Liu, Y. Remaining useful life prediction of iiot equipment using hidden semi-markov model with hyper-erlang sojourn time. *IEEE Internet of Things Journal* (2024).

[8] Maciel, P., Matos, R., Silva, B., Figueiredo, J., Oliveira, D., Fé, I., Maciel, R., and Dantas, J. Mercury: Performance and dependability evaluation of systems with exponential, expolynomial, and general distributions. In *2017 IEEE 22nd Pacific Rim international symposium on dependable computing (PRDC)* (2017), IEEE, pp. 50–57.

[9] Maciel, P. R. M. *Performance, Reliability, and Availability Evaluation of Computational Systems, Volume I: Performance and Background*. Chapman and Hall/CRC, 2023.

[10] Maciel, P. R. M. *Performance, Reliability, and Availability Evaluation of Computational Systems, Volume II: Performance and Background*. Chapman and Hall/CRC, 2023.

[11] Mialaret, M., Pereira, P., Sá Barreto, A., Pinheiro, T., and Maciel, P. Automated phase-type distribution fitting via expectation maximization. *Journal of Reliable Intelligent Environments 10* (2024), 339–355.

[12] Okamura, H., and Dohi, T. Ph fitting algorithm and its application to reliability engineering. *Journal of the Operations Research Society of Japan 59*, 1 (2016), 72–109.

[13] Pereira, P., Araujo, J., Melo, C., Santos, V., and Maciel, P. Analytical models for availability evaluation of edge and fog computing nodes. *The Journal of Supercomputing 77*, 9 (2021), 9905–9933.

[14] Pereira, P., Araujo, J., Torquato, M., Dantas, J., Melo, C., and Maciel, P. Stochastic performance model for web server capacity planning in fog computing. *The Journal of Supercomputing* (2020), 1–25.

[15] Pereira, P., Melo, C., Araujo, J., Dantas, J., Santos, V., and Maciel, P. Availability model for edge-fog-cloud continuum: an evaluation of an end-to-end infrastructure of intelligent traffic management service. *The Journal of Supercomputing 78*, 3 (2022), 4421–4448.

[16] Prados-Garzon, J., Ameigeiras, P., Ramos-Munoz, J. J., Andres-Maldonado, P., and Lopez-Soler, J. M. Analytical modeling for virtualized network functions. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)* (2017), IEEE, pp. 979–985.

[17] Reinecke, P., Krauss, T., and Wolter, K. Hyperstar: Phase-type fitting made easy. In *2012 Ninth International Conference on Quantitative Evaluation of Systems* (2012), IEEE, pp. 201–202.

[18] Reinecke, P., Krauss, T., and Wolter, K. Phase-type fitting using hyperstar. In *European Workshop on Performance Engineering* (2013), Springer, pp. 164–175.

[19] Silva, B., Matos, R., Callou, G., Figueiredo, J., Oliveira, D., Ferreira, J., Dantas, J., Lobo, A., Alves, V., and Maciel, P. Mercury: An integrated environment for performance and dependability evaluation of general systems. In *Proceedings of Industrial Track at 45th Dependable Systems and Networks Conference, DSN* (2015).

[20] Silva Pinheiro, T. F., Oliveira, D., Matos, R., Silva, B., Pereira, P., Melo, C., Oliveira, F., Tavares, E., Dantas, J., and Maciel, P. *The Mercury Environment: A Modeling Tool for Performance and Dependability Evaluation.* 06 2021.

[21] Thummler, A., Buchholz, P., and Telek, M. A novel approach for fitting probability distributions to real trace data with the em algorithm. In *2005 International Conference on Dependable Systems and Networks (DSN'05)* (2005), IEEE, pp. 712–721.