Characterization Analysis of Resource Utilization Distribution

Robert Birke and Lydia Y. Chen
IBM Research Zurich Lab
Rüschlikon, Switzerland
(bir,yic)@zurich.ibm.com

Marco Gribaudo and Pietro Piazzolla
Politecnico Milano
Milan, Italy
(gribaudo,piazzolla)@elet.polimi.it

***Abstract*—To efficiently manage resources and provide guaranteed services, today's computing systems monitor and collect a large number of resource usages, for example the average and time series of CPU utilization. However, little is known about the analytical distribution of resource usages, which are the crucial parameters to infer performance metrics defined in service level agreements (SLAs), such as response times and throughputs. In this paper, we aim to characterize the entire distribution of CPU utilization via stochastic reward models. In particular, we first study and derive the probability density function of the utilization of widely known and applied queuing systems, namely Poisson processes, Markov modulated Poisson processes and time-varying Poisson processes. Secondly, we apply our proposed analysis on characterizing the CPU usage of live production systems, and simulated queuing systems. Evaluation results show that analytical characterization of the selected queueing models can capture the utilization distribution of a wide range of real-life systems well, and we argue the robustness of our methodology to further infer system performance metrics.**

***Keywords*-distribution of utilization; reward model; MMPP/M/1/k; time varying M/M/1/k**

I. Introduction

CPU utilization, memory space utilization, and disk utilization, are the most prevailing performance metrics collected on today's computing systems. Using those resource usage statistics one can glimpse into how individual system components are utilized, and further develop resource management and capacity planning policies. However, those utilization values reflect little on user perceived performance, such as response time and throughput per transaction, which are commonly defined in service level agreements (SLA). Therefore, the challenge faced in today's systems is how to leverage the large number of resource utilization values monitored and stored by standard performance tools to further estimate the SLA metrics. To such an end, it is important to accurately characterize resource usages and then apply the analysis on available performance inference methodologies, like simulation and queueing models.

Typically, characterization studies [1]–[3] collect time series resource usages[1] from on-production systems or the execution of benchmarks which emulate the systems. Their methodologies are drawn from statistical methods, i.e., regression method, time series modeling, and machine learning. Their objectives are often to capture/predict the trend of coarse-grained average resource utilization, e.g., time series of hourly CPU utilization, so that the system dynamics can be further reproduced by means of simulation and modeling to design various resource management policies. Indeed, a large number of system studies [4], [5] adopt resource usage as the workload input, and require not only coarse-grained average queueing utilization values but also finer grained information, such as utilization distribution. Such information is commonly resorted to by either fine-grained trace-driven simulations, which have a very high storage overhead, or the introduction of workload assumptions about their higher statistic moments, which are yet to be characterized. Consequently, it is not clear how effectively and accurate user perceived performance metrics can be induced from the state-of-art characterization analysis, due to the lack of light-weight methodologies to obtain higher order information about utilization traces.

The state-of-art queueing models [6] applied on computing systems require workload parameters, specified in terms of arrival and system service rates, and then generate performance metrics, such as response time and resource utilization. Depending on the distribution of workload parameters, the distribution of response times and queueing lengths can be characterized analytically. Surprisingly, little is said about the distribution of the resource utilization, even for known queueing models, which actually resemble many of today's computing and networking systems. The advantage of having such analytical characterization of resource utilization distribution is that one can straightforwardly compare it with the empirical utilization distributions collected from real systems. As such, the underlying queueing models and corresponding parameters can be identified, and hence the performance measured.

Motivated by the importance and the lack of analysis on utilization distributions, this paper focuses on the CPU utilization and develops an analytical characterization methodology suitable for a wide range of real and simulated systems. To such an end, our efforts are twofold: first, we analytically derive the distribution of utilization for well-known queueing models, i.e., simple Poisson processes,

[1]In this paper, we interchangeably use "time series of resource usage" and "utilization trace"

Markov modulated Poisson processes (*MMPP*), and time-varying Poisson processes, using stochastic reward models. Secondly, we characterize utilization traces from real production systems, and simulated systems, by mapping their utilization distribution with the ones derived from the aforementioned queueing models. Results obtained show that the proposed methodology can efficiently capture the utilization distribution and further preserve the system dynamics with negligible storage overhead, compared to existing trace driven simulations.

This paper is organized as follows: the motivating example from real systems is explained in Sec. II. The analytical derivation of utilization distribution of $MMPP/M/1/K$, and time-varying $M/M/1/K$ is described in Sec. III. Sec. IV contains the evaluation results. Sec. V concludes this paper.

II. Motivating Examples

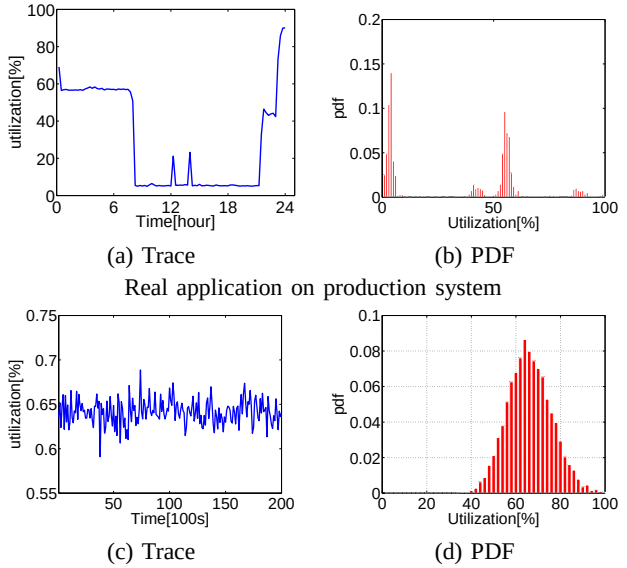


Figure 1. Time series and probability density function (PDF) of CPU utilization usage.

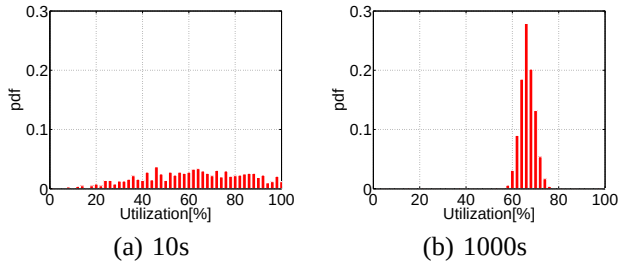


Figure 2. PDF of CPU utilization of simulated $M/M/1/200$ with different sampling window sizes.

Today's on-production computing systems are equipped with easily-accessed low-level performance monitoring tools, such as `vmstat` and `iostat`, to collect and store resource utilization values. Due to the limited storage space, the utilization sample points, collected for example each minute, are often either aggregated into average values over larger time windows, such as 15 minutes or one hour, or fused into a probability density function (PDF). Hence, time series of utilization values are collected in a coarse granularity and, even if the pdf is collected in a finer granularity, it is typically aggregated over a discrete range. We collected 15-minute aggregated utilization values and the minute pdf from an on-production server on Jan 12, 2012. We depict the 15-minute CPU utilization time-series and the minute-based pdf in Fig. 1 (a) and (b), respectively. From Fig. 1 (a), one can observe that the workload fluctuates in a time-varying fashion, which is commonly observed in many computer and network systems. Correspondingly, one can find several peaks in its pdf (Fig.1 (b)). Such a real life system is well attributed to the time-varying queueing models.

In addition to the time-variability, many systems, especially web systems, also show a strong variability in their stationary workload, i.e., even if the workload parameters, such as average inter-arrival and service times remain constant, their distribution around the mean value is quite wide. To study the utilization, we use a $MMPP/M/1/K$ simulator which records the percentage of time the server is busy in every sampling window. The details of $MMPP/M/1/K$ are described in Sec. III. In Fig. 1 (c) and (d), we present the time series and the empirical pdf of $MMPP_2/M/1/200$ with workload intensity[2] 0.65, using 100s as the sampling window size. One can see that the time series of utilization fluctuates up and down around 65%. This is also where the peak in the utilization pdf is located. Furthermore, the pdf appears to be a normal distribution with an average around 65%. In contrast, existing queueing analysis can only provide the closed form formula of the average utilization, given the average arrival rate and service rate, but sheds no light on the utilization distribution.

Then we use an $M/M/1/K$ simulator with workload intensity 0.65, and we apply twodifferent window sizes, i.e., 10s and 1000s on this system. The resulting empirical pdfs are shown in Fig. 2 (a) and (b) respectively. One can see that the shape of the empirical distribution varies depending on the sampling window size. When the sampling window size is bigger, the pdf is more concentrated around the average values, whereas when the sampling window size is smaller the empirical pdf is more spread out and deviates from the average values. In summary, the sampled values are closer to the average workload intensity when applying a bigger

[2]Workload intensity is defined as the average arrival rate divided by the average service rate. One can also refer to such a value as the average utilization for a stationary system.

sampling window, because the system is moving toward the steady state from the transient states. To characterize the empirical utilization distribution, the effect of sampling window size should be well factored in.

III. Analytical Characterization of Utilization Distribution

In this section, we derive analytical characterization of utilization distribution for two different queueing systems: $MMPP/M/1/K$ and time-varying $M/M/1/K$ systems, which can be used to model a wide range of computing systems. All of these queueing models have one queue and a single server, whose utilization is defined by the ratio between the *busy* time and the *total* time. In particular, when a system is observed as not being idle for B time units over the total time T, the utilization can be computed as $U = B/T$. As T is a constant value, the busy-time B can capture the dynamics of U. Consequently, to analytically derive utilization distribution, we first derive the distribution of the busy-time for the aforementioned queueing systems, i.e., MMPP/M/1/K and time varying M/M/1/k, using reward models [7].

A. Reward Models

Reward models are an extension of ordinary continuous time Markov chains (CTMC), which in turn are widely used to solve queueing models, such as $M/M/1/K$ and $MMPP/M/1/K$ systems. States in a CTMC represent different occupation levels of the queue, i.e., number of jobs, or account for different "phases" of service when a complex service distributions are used. Reward models introduce continuous variables (*reward variables*) whose values continuously change over time at constant speed. The rate, at which the reward variables evolve, depends on the state of the CTMC. In this work we use the CTMC part of the reward models to characterize the queueing component of the system we are interested in studying. We use a single reward variable b to account for the time a server has been busy. In particular, we set the reward rate to 1 for all the states that represent a busy system, and we set it to 0 for all the idle states. In this way the value of the reward variable represents the total time B the system has not been idle up to the current time T.

B. $MMPP/M/1/K$ Queueing Systems

A *MMPP* (see [8]) is used to capture high variations in the workload. $MMPP/M/1/K$ queues can thus better characterize web based systems, whose arrival process is highly varying. Here, inter-arrival times between jobs can fluctuate a lot and each job still requests a service time following an exponential distribution. Particularly, in an *MMPP*, the arrival rates are governed by another CTMC (with infinitesimal generator C) called the *modulator process*. The states of the modulator process are used to characterize different time

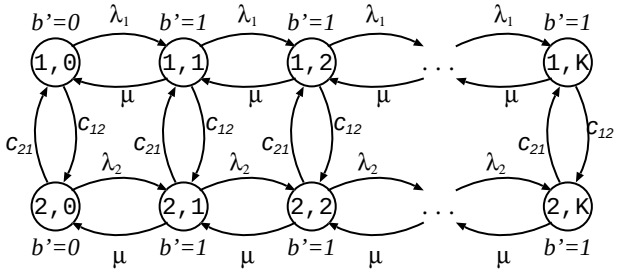


Figure 3. A reward model to compute the distribution of the utilization of an $MMPP/M/1/K$ queue.

epochs, where requests arrive at a different Poisson rate. We call λ_i the arrival rate of requests in epoch i. The elements of the infinitesimal generator, C, define the evolution of the modulating process. An element c_{ij} with $i \neq j$, represents the transition rate from epoch i to epoch j. Since C is an infinitesimal generator, we have $c_{ii} = \sum_{j \neq i} c_{ij}$. Depending on the number n of epochs in the modulating CTMC, we refer to the corresponding arrival process as $MMPP_n$, and the corresponding queueing system as $MMPP_n/M/1/K$. The queueing model can be analyzed by building a CTMC with $n \cdot (K+1)$ states. Each state represents the combination of a given number of jobs j in a given epoch l. The infinitesimal generator of the modulating process C governs the transition among the epochs while maintaining the same length of the queue. The queue length increases at different rates λ_l, depending on the considered epoch l. Contrarily, the queue length decrease at rate μ since the exponentially distributed service time does not depend on the modulating process. Fig. 3 shows the complete CTMC of a $MMPP_2/M/1/K$ queue, where the modulating process is defined as:

$$C = \begin{vmatrix} -c_{12} & c_{12} \\ c_{21} & -c_{21} \end{vmatrix}, \tag{1}$$

and the jobs, in epochs 1 and 2 arrive according to Poisson processes with rates λ_1 and λ_2, respectively. The distribution of the utilization for the $MMPP/M/1/K$ queue can be computed by adding a reward variable b that accounts for the accumulated busy time. In this case, the rate of the continuous variable is $b' = 0$ for all epochs when the system is idle, and it is $b = 1$ for all other epochs and queue lengths where the system has at least one job in service. If we call $\pi_{l,j}(b,t)$ the probability density of being in epoch l, with j jobs in the queue, and the system having been busy for b time units at time t, the equations of the corresponding

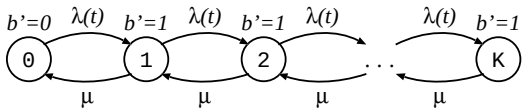


Figure 4. A reward model to compute the distribution of the utilization of an $M/M/1/K$ queue with time-varying arrivals.

reward model are the following:

$$\begin{aligned}
&\frac{\partial \pi_{l,0}(b,t)}{\partial t} = -\lambda \pi_{l,0}(b,t) + \mu \pi_{l,1}(b,t) + && \text{for } j=0,\\
&\qquad + \sum_h c_{hl} \pi_{h,0}(b,t)\\
&\frac{\partial \pi_{l,j}(b,t)}{\partial t} + \frac{\partial \pi_{l,j}(b,t)}{\partial b} = -(\lambda+\mu)\pi_{l,j}(b,t) + && \text{for } 0<j<K,\\
&\qquad + \mu \pi_{l,j+1}(b,t) + \lambda_l \pi_{l,j-1}(b,t) +\\
&\qquad + \sum_h c_{hl} \pi_{h,j}(b,t)\\
&\frac{\partial \pi_{l,K}(b,t)}{\partial t} + \frac{\partial \pi_{l,K}(b,t)}{\partial b} = -\mu \pi_{l,K}(b,t) + && \text{for } j=K.\\
&\qquad + \lambda_l \pi_{l,K-1}(b,t) +\\
&\qquad + \sum_h c_{hl} \pi_{h,K}(b,t)
\end{aligned} \tag{2}$$

Boundary and initial conditions can be easily derived, but are skipped for space constraints. The probability density of the utilization u over a time window w can then be derived as:

$$p(u,w) = w \sum_{j=0}^{K} \sum_{l=1}^{n} \pi_{l,j}(u \cdot w, w), \qquad \text{with } 0 \le u \le 1. \tag{3}$$

C. Time-varying $M/M/1/K$ Queuing Systems

While $MMPP/M/1/K$ systems well capture the workload fluctuation caused by the modulating process with different arrival poisson processes, it is not suitable to study the workload variability caused by the effect of time, e.g., peak v.s. off-peak working hours, shown in Fig. 1 (a). In such a system, the arrival rate changes as a function of time and an nonhomogeneous Markov process is more effective. We thus introduce the *time-varying $M/M/1/K$ Queueing System*. In this case, arrivals are governed by a time-varying poisson process of rate $\lambda(t)$. The corresponding CTMC can be described by the process depicted in Fig. 4.. The corresponding equations are the following:

$$\begin{aligned}
&\frac{\partial \pi_0(b,t)}{\partial t} = -\lambda(t)\pi_0(b,t) + \mu \pi_1(b,t) && \text{for } j=0.\\
&\frac{\partial \pi_j(b,t)}{\partial t} + \frac{\partial \pi_j(b,t)}{\partial b} = -(\lambda(t)+\mu)\pi_j(b,t) + && \text{for } 0<j<K.\\
&\qquad + \mu \pi_{j+1}(b,t) + \lambda(t)\pi_{j-1}(b,t)\\
&\frac{\partial \pi_K(b,t)}{\partial t} + \frac{\partial \pi_K(b,t)}{\partial b} = -(\mu)\pi_K(b,t) + && \text{for } j=K.\\
&\qquad + \lambda(t)\pi_{K-1}(b,t)
\end{aligned} \tag{4}$$

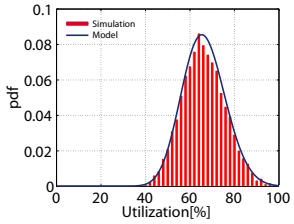


Figure 5. Comparison of analytical results with statistics on the simulated traces of MMPP/M/1/200.

Also in this case the boundary, initial conditions, and the distribution of the utilization $p(u,w)$ can be easily derived from the precious expressions.

IV. Analysis and Evaluation

In this section we use the analytical results defined in Sec. III to study the behavior of the utilization distribution. For what concerns the computation of the solution, several different techniques are available depending on the particular structure of the model, and on the measures in which we are interested. For time homogeneous models (e.g., 2), moments can be efficiently derived using techniques based on uniformization such as the one presented in [9]. For the computation of the full distribution, under non-homogeneous assumptions (e.g. Eq 4), the full equation must be solved: in this work we have used the numerical method presented in [10][3].

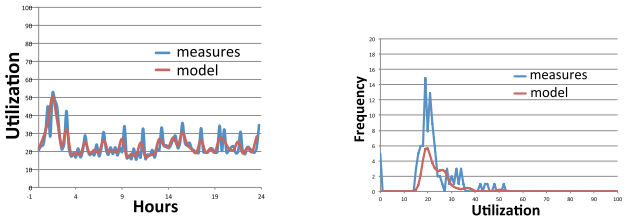
A. Model Validation on MMPP/M/1/k

To validate the model, we compare the analytical results obtained from the reward model with simulation traces from the discrete event simulator. In particular, we have developed a simulator with Omnet [11] of a simple transactional system, where jobs arrive according to a *MMPP*, with two states having arrival rates $\lambda_1 = 0.6 job/s$ and $\lambda_2 = 0.7 job/s$, respectively, and phase-change rates $c_{12} = c_{21} = 1.0$.We set $K = 200$ and $w = 100s$, with the queue being initially empty, and the *MMPP* modulating process starting in epoch 1. The comparison between utilization traces of MMPP/M/1/200 and analytical results is shown in Fig. 5. The results obtained with the model strongly agree with the ones computed from the simulator traces.

B. Evaluation of Real Systems Using the Time-Varying $M/M/1/K$ Model

Finally, we compare the results of the time-varying $M/M/1/K$ model described in Sec. III-C with the data acquired from a real trace. We consider the trace of a SAP [12] application server, which can be considered as a system running primarily CPU bound workloads, but in this case it

[3]The technique mentioned here is defined for fluid models. However, since reward models are special cases of fluid models, it can also be applied in the scenario considered in this paper.



(a)Time series (b) PDF

Figure 6. Comparison of CPU utilization collected from a real system against model results.

is little utilized. We fix the service rate of the application to $\mu = 0.2 job/s$, and we set $\lambda(t)$ as:

$$\lambda(t) = \mu\tilde{u}(t), \tag{5}$$

where $\tilde{u}(t)$ is obtained from the utilization trace. In particular, we have applied a moving average filter of width ω to the traces. If we call $u(t)$ the utilization measured from the traces, then $\tilde{u}(t)$ is computed as:

$$\tilde{u}(t) = \frac{1}{\omega}\int_{-\omega/2}^{\omega/2} u(t+x)dx. \tag{6}$$

In our experiment, we use $\omega = 1800s$ (0.5 hours). Also, the length of the sampling window is set to $w = 1800s$, the size of the queue to $K = 100$ and the system starts initially empty. Fig. 6 (a) compares the time evolution of the mean utilization (computed from the model) with the original trace. As we can see, there is a close match between the real data and the one obtained by the time-varying $M/M/1/K$ model. The model however tends to smooth-out the curve: this is a typical side effect due to the filtering applied. Fig. 6 (b) shows the distribution of the utilization, both of the real trace and of the analytical model. For the latter, the utilization is computed by averaging the output of the model over the total time. Here, peaks tend to be smoothed out a lot, and there is a slight mis-alignment between the analytical solution and the statistics from the real trace. This could be partially due to the different discretization schemes used by the analytical solution and the statistics from the real trace.

V. Conclusions

In this paper we develop an analytical characterization of the utilization distribution, computed on known queueing models using finite sampling window sizes. The proposed analysis is able to match the data that are obtained from traces and logs commonly stored on real systems. In particular, the first two moments of the utilization can be used to infer both the arrival and the service rate of a server at various simulated queueing systems and real systems.

Future works will go in two different directions. From an analytical point of view, we will try to capture a closed form relation between the arrival rate, the service rate, the moments of the utilization and the sampling window sizes. We will also study the effect of more complex arrival processes and non-exponential service times on the utilization distribution. From a practical point of view, we will try to apply the proposed results to infer the operating parameters of the servers from their logs and traces, and use the models to infer possible erroneous or non-normal behaviors of such servers.

References

[1] Q. Zhang, J. Hellerstein, and R. Boutaba, "Characterizing task usage shapes in google compute clusters," in *Proceedings the 5th International Workshop on Large Scale Distributed Systems and Middleware (LADIS)*, 2011.

[2] M. Chen, X. Wang, and B. Taylor, "Integrated control of matching delay and cpu utilization in information dissemination systems," in *IWQoS*, 2009, pp. 1–9.

[3] Q. Diao and J. J. Song, "Prediction of cpu idle-busy activity pattern," in *HPCA*, 2008, pp. 27–36.

[4] L. Y. Chen, A. Das, A. Sivasubramaniam, Q. Wang, R. Harper, and M. Bland, "Consolidating clients on back-end servers with co-location and frequency control," in *SIGMETRICS/Performance*, 2006, pp. 383–384.

[5] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, "Efficient Resource Provisioning in Compute Clouds via VM Multiplexing," in *Proceedings of International Conference on Autonomic Computing*, ser. ICAC, 2010, pp. 11–20.

[6] L. Kleinrock, *Queueing Systems*. Wiley Interscience, 1975.

[7] R. A. Marie, A. L. Reibman, and K. S. Trivedi, "Transient analysis of acyclic markov chains," *Perform. Eval.*, vol. 7, no. 3, pp. 175–194, 1987.

[8] W. Fischer and K. Meier-Hellstern, "The markov-modulated poisson process (mmpp) cookbook," *Performance Evaluation*, vol. 18, no. 2, pp. 149 – 171, 1993.

[9] F. Castella, G. Dujardin, and B. Sericola, "Moments analysis in homogeneous markov reward models," *Methodology and Computing in Applied Probability*, vol. 11, no. 4, pp. 583–601, 2009. [Online]. Available: http://dx.doi.org/10.1007/s11009-008-9075-5

[10] M. Gribaudo and A. Horváth, "Fluid stochastic petri nets augmented with flush-out arcs: A transient analysis technique," *IEEE Trans. Software Eng.*, vol. 28, no. 10, pp. 944–955, 2002.

[11] "Omnet++," http://www.omnetpp.org/.

[12] "Sap business management software solutions, applications and services," http://www.sap.com/.