Design and Validation of an Analytical Model to Evaluate Monitoring Frameworks Limits

Abdelkader Lahmadi, Laurent Andrey, Olivier Festor

Design and Validation of an Analytical Model to Evaluate Monitoring Frameworks Limits

Abdelkader Lahmadi, Laurent Andrey and Olivier Festor
INRIA Nancy - Grand Est
615 Rue du Jardin Botanique
F-54602 Villers-Lès-Nancy, France
Email: {Abdelkader.Lahmadi,Laurent.Andrey.Olivier.Festor}@loria.fr

***Abstract*—It is essential that a monitoring system is being designed with performance and scalability in mind. But due to the diversity and complexity of both the monitoring and the monitored systems, it is currently difficult to reason on both performance and scalability using ad hoc techniques. Thus, both simulation is required and analytical models based on well established techniques such as queueing theory have to be developed.**

In this paper we provide an analytical modeling of the behaviour of the commonly used manager-agent monitoring frameworks within two scenarios: single manager-single agent and single manager-multiple agents. The two designed models enable the automation of the estimation of the scalability limit of the two types of management monitoring schemes regarding a performance metric like the monitoring delay. We validate our developed models through simulation based on parameters values obtained from the performance measurement of a life running JMX-based monitoring applications with the two scenarios.

***Index Terms*—Monitoring, Scalability, Modeling techniques**

I. Introduction

Despite the increasing use of on site measurement techniques, analytical models still play an important and complementary role in the performance evaluation of communication networks and protocols, as they are often more rigorously defined and provide an elegant method to obtain performance evaluation and results. The increase in both size and requirements on management systems which have to deal with more managed systems and management data while at the same time being able to maintain an accurate and near real time vision of the managed systems puts high pressure on the monitoring part and algorithms of the management system. In this context, modeling of monitoring frameworks needs more attention and the development of queueing based model becomes crucial to both assess their performance and verify results obtained from measurement. Such models are also very important to address the following problems:

(i) managed system capacity provisioning, which enables a managed system to determine how much capacity to allocate to monitoring activities in order to still be managed;
(ii) management cost identification, which enables the resource consumption to be determined for a given management configuration;
(iii) monitoring system tuning, which enables management system bottlenecks to be identified for the purpose of tuning.

However, modeling management frameworks is not trivial for the following reasons. First, various management frameworks have vastly different organisational, communication and information models, underlying protocols and technologies with different performance characteristics. Furthermore, in a management system :

(i) there may be resources constraints at the agent side, usually integrated in different ways on the managed system;
(ii) management data caching, filtering and aggregation [1] may be performed directly at by agent which complicates the performance modeling;
(iii) real management workload has different time scales, where some requests need small time scales, and others are less frequent with larger time scales.

Finally, the management community has a lack of input on workloads that management systems experience in real life. Management of networks and services is usually achieved through the use of a management framework consisting of a management algorithm running on different entities and using an underlying management protocol. Each entity provides a certain management facility to its preceding entity and uses the facility provided by its successor to carry out its part of the overall management task. For instance, a typical management framework, consists of three entities- a manager that is responsible for triggering management requests initiated by human beings (administrator) operations or a hard and software modules running a management algorithm; an agent that implements managed objects with their derived attributes, operations and notifications. The agent is also responsible to fetch management data from the managed system, respond to the manager requests and issues notifications when a threshold is crossed. The third entity is the real managed system that provides the primary functions to overcome service's users requirements [2]. In such a management framework, incoming management operations trigger requests, processed by the agent, and fetch management data from the managed system. In this paper, we developed models based on closed network of queues for a manager-agent pattern, where each queue represents one monitoring entity. Although, management frameworks may be different, they are constructed using this pattern

as a subsystem. Thus, a key contribution of this work is that a complex model of a management framework is reduced to model management requests handling at the individual entities and across them. We validate our developed mathematical formula using queueing simulation where parameters values are obtained from performance measurement of a Java Management eXtension (JMX) [3] based management platform to monitor a Java based web server.

The remainder of this paper is structured as follows. Section II provides an overview of related works, mainly those where some analytical models are proposed to assess the performance of monitoring systems. Section III depicts some properties of monitoring frameworks that impact their scalability. In section IV, we develop our analytical models for the main scenarios : single manager-single agent and single manager-multiple agents. We also provide an analysis of these models. In section V, we provide simulation results that verify the developed mathematical formula. Section VII presents our conclusion.

II. Related work

The main distinction of our case study is the performance and scalability evaluation of monitoring systems. The techniques used in this study have benefited from the literature on systems performance evaluation (e.g., Jain [4] and Gunter et al. [5]).

In the network and services management community rigorous analytical models are rare. To our knowledge, only the following works have provided queueing models dedicated to management frameworks so far. In [6], the authors propose a queueing model of a manager within an Enterprise management system. In [7], the authors use some queueing modeling features to simulate the behaviour of an SNMPv1 agent. The thesis of Subramanyan [8] contains a model of the behaviour of a hierarchical SNMP based management platform using queueing networks. In [9], authors provide some formulas to assess the performance of different management approaches (centralized, distributed and mobile agents). These mathematical results are empirical rather than developed from well defined modeling approaches.

Performance modeling of Internet services and applications is well studied in other disciplines for other puroposes [10], [11] but these models usually omit the monitoring part from their models despite the fact that the quality of a service derived from performance quantities is driven by the monitoring part and its impact is not negligible. Thus, coupling services and monitoring performance models is a point of interest to provide service quality requirements. The very recent work of Beitgang [12] does such a coupling. There, authors assess the impact of monitoring with a load sharing strategy on servers ressources. Our work follows a similar path to provide well defined monitoring performance models that will be usefull to reliably optimize monitoring systems to fit their monitored environments mainly large scales ones.

III. Monitoring frameworks models

In our work, we focus only on the manager-agent pattern since it is the most used and any traditional monitoring framework can be seen as a composition of this pattern. Recent management frameworks such as autonomic management still relies on this pattern where an autonomic entity is a local loop of a manager and an agent. The question that arises when using a monitoring framework with this pattern is related to it's scalability limit while increasing a scale factor. We want to be able to predict this limit with regard to monitoring delays. The factors that affect this delay are mainly the number of agents, managed devices, monitoring variables and their monitoring rates. This limit defines the range of values of a scale factor that guarantees a suitable delay with respect to a tolerance value. We study limit prediction models for two monitoring scenarios which are the single manager-single agent and single manager-multiple agents. Figure 1 depicts their monitoring models, where management variables are collected by a built-in algorithm on the manager that connects to a single or to many agents to retrieve variables values.

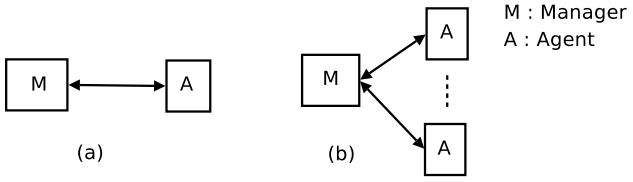


Fig. 1. Fundamental topologies of traditional monitoring frameworks. (a) single manager-single agent.(b) single manager-many agents

We consider a periodic polling workload issued by an implicit monitoring algorithm [2] on the manager. This type of workload is interactive since each polled variable issues a monitoring operation when its monitoring interval arises.

The performance of monitoring has to be characterised by some suitable metrics. As stated in [13], the delay that monitoring calls experience is a good metric for a monitoring algorithm performance. The attribute delay is defined as the time experienced by a monitoring attribute to retrieve its value from an agent to a manager.

A. Monitoring workload model

Monitoring traffic generated either with polling or notification activity can be modeled as an ON/OFF process where a monitoring variable alternates between ON (busy) and OFF (idle) states. This is depicted in Figure 2.

A monitoring variable enters the ON state, when it attempts to retrieve its value by calling a monitoring operation. This operation needs one or more monitoring protocol messages interlevead by the ITM (Inter-Message) times. Each message may carries one or more values of a monitoring variable. Therefore, the duration of the ON period depends on the variable granularity (simple or tabular), the network conditions and monitoring protocol operations (logical or single). After all values of the monitoring variable are fetched, the variable goes into the OFF state. The duration of ON state and OFF

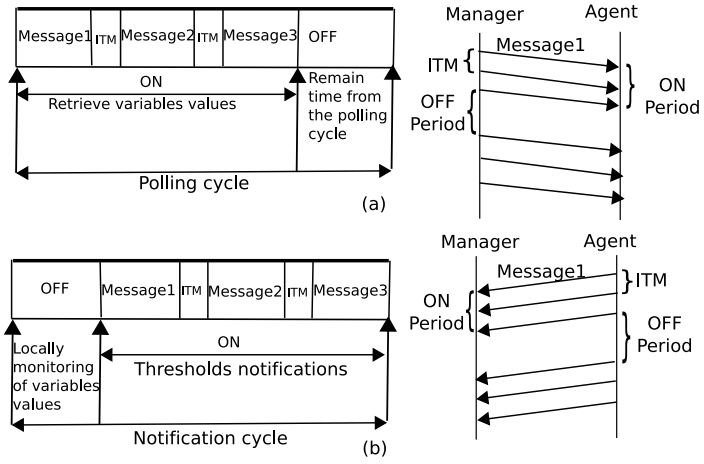


Fig. 2. ON/OFF model for a monitoring variable workload within (a) polling activity and (b) notification activity.

state correspond to monitoring delay and monitoring interval, respectively. OFF time denotes any time that the variable is inactive. ON time is the time taken to fetch all values of a monitoring variable in a monitoring operation. The ON state can be split into successive monitoring messages used to deliver variable's values.

IV. Analytical Models

We have developed a simple queueing network model that applies to a manager agent pattern shared by an interactive set of monitoring variables with the ON/OFF activity model described above. We consider, in this work, closed queueing networks where the number of monitoring variables in the system is always constant. Each queue within the model represents a monitoring entity, i.e. the monitoring algorithm, the manager and the agent. Here, we assume an implicit monitoring algorithm running on the manager using polling operations. Each monitoring variable incurs a single monitoring protocol message to carry its value from the agent.

Here, we consider a finite population of M interactive monitoring variables with ON/OFF traffic, which are sharing the manager queue to request their values from one or many agents. We assume that the M variables OFF times distribution fits an exponential distribution whose mean is Δ. Herein, we assume a naive monitoring algorithm where monitoring calls of different variables occurs continuously and independently at a constant rate. We model the monitoring algorithm, that fetches variables values, as an infinite server (IS). The respective service times of a monitoring message (which is a component of the ON times) have a general distribution whose mean is S. Figure 3(a) depicts the single manager-single agent scenario modeled with a closed queueing network. This monitoring pattern fits perfectly to the well established machine repair model [5]. The figure 3(b) depicts the single manager-multiple agents scenario modeled with a closed queueing network model. The latter maps to to the well known central server model [14]. We assume processor-sharing queues, and our results hold for any general service

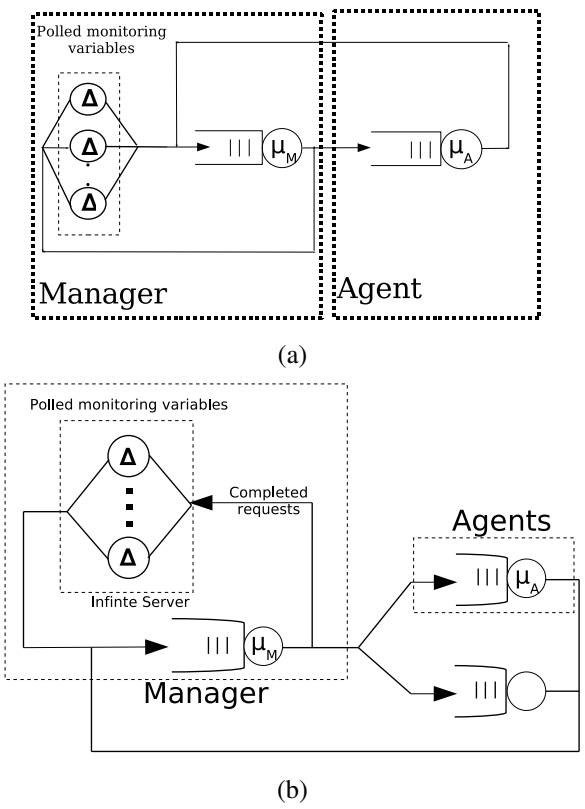


Fig. 3. Closed queueing models for two monitoring scenarios: (a) single manager-single agent, (b) single manager-many agents.

time distribution as long as the monitoring intervals (think times) have an exponential distribution. We have a closed-form solution for the two models and we can describe the delays with simple parameters. In this work, our model analysis is based on the use of Little's Law [5]. This law states that the response time of a queue is equal to the mean number of jobs within the queue to its throughput. If one denotes by K the queue length, γ the throughput and T is the response time, then Little's law is given by :

$$T = \frac{K}{\gamma} \tag{1}$$

The performance model parameters are computed using the Mean Value Analysis (MVA) technique [5]. This method is based on two fundamental equations and it allows us to compute the mean values of measures of interest such as mean waiting times, throughput, and the mean number of jobs at each node. It is based on Little's law and the theorem of the distribution at the arrival time (in short, arrival theorem) for closed product-form networks. This technique allows us to identify scalability limits for each scenario with respect to a scale factor variation. It provides mean values of metrics rather than a full distribution.

Below is a list of parameters used for expected response times using the selected queueing models:

- M : the number of monitoring attributes polled by an algorithm residing on the manager.

- N : the number of agents attached to the single manager.
- Δ : the mean monitoring interval of monitoring attributes.
- R : the response time from a queueing theory view that represents monitoring delays.
- γ : the throughput in terms of completed jobs. In our context it represents the number of polled attributes.
- μ_i : service rate at node i (manager or agent)
- S_i : service time at node i.
- e_i : the visit ratio of a node i.
- D_i : service demand $D_i = e_i \times S_i$ at node i.

We use closed queueing networks with an interactive traffic (ON/OFF model). The response time T in Little's law, expressed by equation 1, represents the total cycle time. We thus obtain: $T = R + \Delta$. The mean number K of customers within the closed queueing network is constant, hence we obtain $K = M$. Little's law can thus be transformed as follows:

$$R = \frac{M}{\gamma} - \Delta \tag{2}$$

We use the above formula to estimate the monitoring delay R with the two monitoring scenarios. Following, we denote by the symbol SS the single manager-single agent model and by the symbol SM the single manager-multiple agents model.

A. Single manager-single agent model analysis

We begin by considering the monitoring delay of a single agent as depicted in Figure 3(a). The monitoring delay R_{SS} computed from 2 is given by:

$$R_{SS}(M) = \frac{M}{\gamma_{SS}(M)} - \Delta_{SS} \tag{3}$$

We observe that the scalability limit regarding delays is the number of monitoring attributes M and their monitoring times Δ. Thus when M increases or Δ decreases, the response times increase. A monitoring system needs to provide a suitable delay to guarantee a good timeliness of polled attributes. Consequently, we need to bound monitoring delays to some threshold value. In this case, we take the monitoring interval Δ as this threshold.

We will now assess the bound on the number M of variable that provides bounded delays to the monitoring interval Δ. Let respectively D_{max}, D_{sum} and D_{avg} be the maximum, respectively the sum and the average of service demand by queue of all non IS-node. We have :

$$D_{max} = max(D_i), D_{sum} = \sum D_i \tag{4}$$

where $i \in Manager, agent$ and $D_{avg} = \frac{D_{sum}}{M}$

From the balanced job bound analysis [5] we have the upper bound of the throughput denoted by

$$\gamma_{max} = min(\frac{1}{D_{max}}, \frac{M}{\Delta + R_{min}}) \tag{5}$$

where

$$R_{min} = max(M \times D_{max} - \Delta, D_{sum} + ((M-1) \times D_{avg} \times \frac{D_{sum}}{D_{sum}+\Delta})) \tag{6}$$

is the lower bound of response time. The highest possible overall throughout is restricted by each node in the network especially by the bottleneck nodes. Thus, we get $\gamma_{SS}(M) \leq \gamma_{max}$. From the model in Figure 3(a), the visit ratios are given by : $e_A = 1$ and $e_M = 2$. At first we determine that : $\gamma_{max} = \frac{1}{D_{max}}$, where $D_{max} = max\{e_M \times S_M, e_A \times S_A\} = 2 \times S_M$. herein, we observe that the bottleneck node is the manager. Using the upper bound Δ on the response time $R_{SS}(M)$, we get as a bound on the number M of monitoring attributes :

$$M^* \leq \frac{\Delta}{S_M} \tag{7}$$

B. Single manager-many agents model analysis

We use the same methodology as above to determine the monitor in delays (i.e. those for which values where received in time) within the model depicted in Figure 3(b). We assume that each Monitoring attribute generates N jobs to retrieve its value from the N agents. We assume also that jobs are asynchronous and do not account for any synchronisation between them. Thus, the monitoring delay R_{SM} computed from 2 is given by :

$$R_{SM}(M, N) = \frac{M \times N}{\gamma_{SM}(M, N)} - \Delta_{SM} \tag{8}$$

We observe that the scalability limits are the number of monitoring attributes, the number of agents and the monitoring times Δ_{SM}. As the first scenario, we would like to predict the number of agents that provide a monitoring delay less or equal to the monitoring time. We determine that the throughput $\gamma_{SM} \leq \frac{1}{(N+1) \times S_M}$ since the visit ratio of the Manager is equal to $N+1$. Given equation 8, the bound on the number of agents N^* is given by :

$$N^* \leq \frac{2 \times \Delta \times \gamma_{max}}{M} \tag{9}$$

V. Simulation

We have simulated the queueing models that we propose in section IV for the polling based manager-agent model with the single manager-single agent scenario and the single manager-multiple agents one. We developed a Matlab script that implements the Mean Value Analysis (MVA) algorithm to compute our performance metrics mainly the monitoring delays while varying a scalability factor.

The simulation input parameters, mainly the service rates for the manager and agents are obtained from the measurement of a running JMX based manager-agent pattern application monitoring a web server. Table V summarizes the input parameters for the queueing model that we consider.

We note that all measurements of service times are obtained from low monitoring rates (1 request/second) to avoid measuring the queueing latency in addition to the service times. This

Queue	Type	Description	Service time strategy	Mean service time (seconds)
Monitoring Attributes	-/M/∞-IS	Polled attributes	load independent	0.994
Manager	-/M/1-FCFS	Manager subsystem	Load dependent	0.00114
Agent	-/M/1-FCFS	Agent subsystem	Load dependent	0.0016

TABLE I
MODEL INPUT PARAMETERS FOR SIMULATION RESULTS.

method has been successfully employed in similar problems [10].

A. Results

First we provide results of the single manager-single agent scenario where we have varied the number of monitoring variables M from 1 to 1000 with a mean monitoring interval $\Delta = 1$ second. Using the parameters in table V and the equation 7, we obtain as theoretical bound of attributes M^* close to 877. In this case, delays are less or equal to 1 second (monitoring interval). As depicted in Figure 4, we find that simulation results give the same value as the mathematical formula 7.

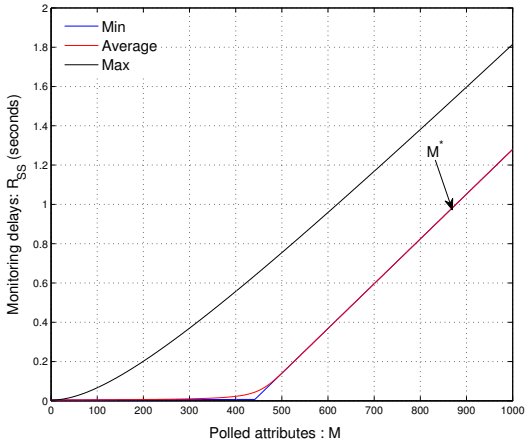
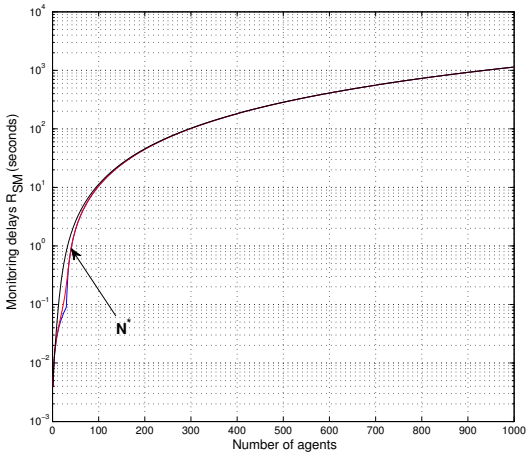


Fig. 4. Monitoring delays of the single manager-single agent scenario as a function of the number of polled attributes

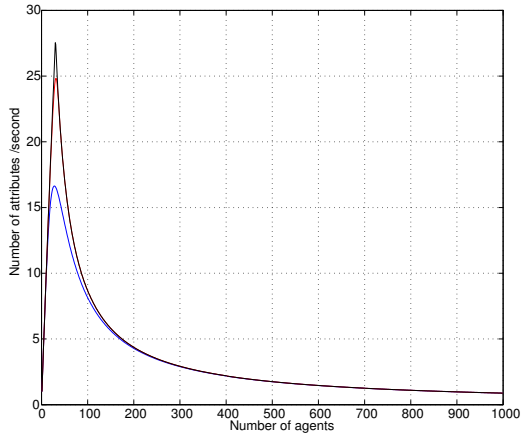
Secondly, we want to predict the maximal number of agents N^* within the single manager-multiple agents scenario and compare it with simulation results. Using values of table V, we observe that $\gamma_{max} = \frac{1}{(N+1)\times S_M}$. We transform the equation 9 and we obtain:

$$N^* \leq \frac{1}{2}\sqrt{\frac{8\times\Delta}{M\times S_M}+1}-1 \qquad (10)$$

In the simulation of a single manager-multiple agents scenario, we fixed $M = 1$ and $\Delta_{SM} = 1$ and we varied the number of agents N. Using equation 10 and values from the table V, we obtain the optimal number of agents that guarantee a monitoring delay less or equal to 1 second close to $N^* = 41$ agents. As depicted in Figure 5(a), we observe that the theoretical result corresponds to the value obtained using simulation.



(a)



(b)

Fig. 5. Performance of the single manager-multiple agents scenario as a function of the number of agents. (a) monitoring delays. (b) total throughput in terms of attributes/s.

VI. DISCUSSION

Our analysis confirms an overall intiuition that the single manager-single agent has a better scalability limit regarding the number of attributes than the single manager-multiple agents regarding an increase in the number of agents. According to our mathematical formulation, designing a monitoring system with the single manager-single agent pattern as a management entity like an autonomic entity is more efficient than traditional central manager with many agents. Despite that this result seems to be trivial from an empirical manner, in this work we validate it in a more rigorous analytical manner.

Our models are simple and usefull. Their underlying assumptions appear however to be heavy and don't reflect all real world deployed monitoring applications. First, we consider in this work a concurrency level of the monitoring algorithm equal to 1, however main monitoring algorithms have a concurrency level greater than one. Usually they rely on a multi-threading approach to fetch variables values. Second, we considered in the modeling phase that monitoring events follow a Poisson process with an exponential distributed monitoring intervals. This assumption needs to be validated from real work monitoring traffic traces. Recent work in [15] has started traffic analysis of real world monitoring applications. Their data sets will be very useful to identify statistical assumptions of queueing models of monitoring applications.

VII. Conclusion

In this paper, we respond to the following question: "how to predict the scalability of a monitoring application following the manager-agent pattern?" By using simple analytical models based on the well developped queueing theory. To this end we have developped two analytical models for two major scenarios: the single manager-single agent and the single manager-multiple agents. The performance metric of interest is the delay that experiences a manager to retrieve the values of monitoring variables from a set of agents. This delay is important to collect accurate data that reflects a real time view of the monitored system or when monitored attributes are short live like SIP[1] transactions for example. Our aim is to predict the scalability limits of each scenarios with regard to delays when the number of agents and attributes increases. We have shown that our analytical models are in line with simulations. We also assessed that the single manager-single agent scales better than the second with respect to their scale factors.

While the model is performant, it needs to be enhanced to support other monitoring features among which the concurrency level is the most important one, especially in the single manager-multiple agents scenario where attributes are in some case retrieved in parallel from agents. This will be the main objective of our future work in this modeling activity. A second activity will be to apply the model to SNMP-based systems. This mainly consists in measuring the parameters needed to feed the model from real world systems. This latter activity is pursued in the context of the FP6-IST EMANICS Network of Excellence.

References

[1] F. Wuhib, M. Dam, R. Stadler, and A. Clemm, "Control considerations for scalable event processing," in *DSOM 2005. Ambient Networks*, J. Schöwälder and J. Serrat, Eds., vol. 3775. Springer, 2005, pp. 233–244.

[2] A. Pras, "Network management architecture," Ph.D. dissertation, University of Twente, Nettherlands, 1995, iSBN: 90-365-0728-6.

[3] SUN, "Java^{TM} management extensions, instrumentation and agent specification, v1.2," http://jcp.org/en/jsr/detail?id=3, october 2002, maintenance Release 2.

[4] R. Jain, *The art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc, 1991, iSBN : 0-471-50336-3.

[5] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications, 2nd Edition*. New York, NY, USA: Wiley-Interscience, 2006.

[6] F. Stamatelopoulos and B. Maglaris, "Performance and efficiency in distributed enterprise management," *Journal of Network and Systems Management*, vol. 7, no. 1, pp. 47–71, March 1999.

[7] C. Pattinson, "A study of the behaviour of the simple network management protocol," in *International Workshop on Distributed Systems: Operations & Management (DSOM)*, Nancy, France, 15-17 October 2001.

[8] R. Subramanyan, "Scalable snmp-based monitoring systems for network computing," Ph.D. dissertation, Purdue University, 2002.

[9] T. M. Chen and L. S. Liu, "A model and evaluation of distributed network managemetn approaches," *IEE Journal on selected areas in communications*, vol. 20, no. 2, pp. 850–857, may 2002.

[10] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An analytical model for multi-tier internet services and its applications," in *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM Press, 2005, pp. 291–302.

[11] Y. Diao, "Stochastic modeling of lotus notes with a queueing model." in *24th International Computer Measurement Group Conference, December 6-11, 1998, Anaheim, California, USA, Proceedings. Computer Measurement Group 1998*, 2001, pp. 229–238.

[12] D. Breitgand, R. Cohen, A. Nahir, and D. Raz, "Using the right amount of monitoring in adaptive load sharing," in *ICAC '07: Proceedings of the Fourth International Conference on Autonomic Computing*. Washington, DC, USA: IEEE Computer Society, 2007, p. 7.

[13] A. Lahmadi, L. Andrey, and O. Festor, "On delays in management frameworks: Metrics, models and analysis," in *17th IFIP/IEEE Distributed Systems: Operations and Management, DSOM 2006, Dublin, Ireland*, D. O. Radu State, Sven van der Meer, Ed., vol. LNCS. Springer-Verlag's Lecture Notes in Computer Science (LNCS), 24-26 October 2006.

[14] J. Buzen, "Queueing network models of multiprogramming," Ph.D. dissertation, Division of Engineering and Applied Science, Harvard University, Cambridge Mass, 1971.

[15] J. Schönwälder, A. Pras, H. Matus, J. Schippers, and R. van de Meent, "SNMP traffic analysis: Approaches, tools, and first results," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Munich, Germany, May 2007, pp. 323–332.

[1] Session Initiation Protocol.