Improving Effectiveness of Simulation Performance Prediction in Clouds

László Muka and István Derka
Széchenyi István University, Department of Telecommunications, Győr, Hungary
e-mails: muka@sze.hu, steve@sze.hu

***Abstract*—The cloud computing is becoming increasingly important execution environment for discrete event simulation models too. The cloud services are expected to provide an automatic and effective parallel and distributed execution but the occurring communication delay in the network may decrease the cloud performance. The paper describes closed queuing network model of on-demand resource use in cloud execution. The paper introduces a method based on coupling factor method and on rough set train-and-test approach of effective simulation performance prediction.**

***Keywords*—parallel and distributed simulation, prediction of simulation performance, coupling factor method, cloud execution environment, rough sets, prediction effectiveness**

I. INTRODUCTION AND MOTIVATION

Over the last few years, the need for the discrete event simulation (DES) of large-scale complex networks and network services has been constantly growing because DES turned to be an efficient tool for the analysis of these systems. The computing capacity requirements large-scale and complex networks can be fulfilled by parallel and/or distributed discrete event simulation approach [2, 12,13].

According to a common and simple definition [2], Parallel and Distributed Simulation (PADS) is defined as any simulation in which more than one processor is used. PADS is the execution of a single discrete event simulation model on a high performance computing platform: on clusters of homogeneous and heterogeneous computers, on WEB, grid and *cloud* execution environment.

The typical situations of PADS approach applications from the point of view of the runtime performance requirements: time consuming applications can be for example the simulation of large and/or complex systems and networks.

The development and use of systems based on the PADS methods are resource consuming, not easy task even today. The *simulation performance prediction* method is appropriate for support to reach good PADS performance [11].

The question of support may be formulated in the following way: How to build a model with a good parallelization potential and how to execute it with a good runtime performance involving the necessary (available) resources of a parallel and/or distributed environment and how to do it in an effective way?

The main *contributions* of the paper can be summarized as follows:

- Closed Queuing Network (CQN) Virtual Machine (VM) model of cloud work including modelling of latencies among cloud instances has been defined.
- Closed Queuing Network (CQN) Virtual Machine (VM) model is enhanced to take into account the use of cloud resources (VMs) on on-demand base.
- Rough model of prediction is introduced to support modelling and the performance prediction in an effective way.

The paper is organized as follows. First, the issues of simulation execution in cloud are examined. Then questions of simulation performance are described including the enhanced CQB and VM model of cloud execution. Forth section presents the rough set modelling of performance prediction in an example. The fifth section concludes the work.

II. ISSUES OF SIMULATION EXECUTION IN CLOUD

A. Cloud Services

It has four deployment models private cloud, community cloud, public cloud and hybrid cloud.

Cloud services can be provided according to three service models, e.g., Software as a Service (S-a-a-S), Platform as a Service (P-a-a-S) and Infrastructure as a Service (I-a-a-S) [1].

B. Parallel and Distributed Simulation in the Cloud

Cloud can be the execution environment for PADS with huge capacity requirement since it has made accessible high performance computing platforms (HPC) to end users of the cloud. For example, in the Amazon's very successful Elastic Compute Cloud (EC2), the Message Passing Interface (MPI) – the standard for parallel programming message communications protocol – is supported by the cloud.

Cloud environments are often better at providing high bandwidth communications among applications than in providing *low latency* [15].

PADS applications *typically* work with significant *communication among segments* with sending a lot of short messages between the processes. Thus, for PADS good performance *quick transport (= low latency)* is more important than high bandwidth alone

Problems related with functioning of the cloud, for the PADS applications using *optimistic synchronization* protocol, may lead to performance degradations [16].

There has been made researches in cloud computing and but less attention have been paid to parallel and distributed simulation and even less to *conservative synchronization method.*

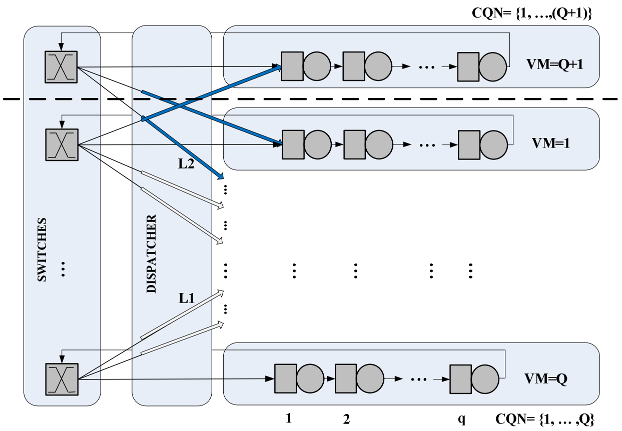

Figure 1. CQN switching and latency modell of cloud execution environment (CPU=VM, LP=LP)

III. SIMULATION PERFORMANCE IN CLOUD

A. General Model of Cloud Performance Simulation

Buyya define cloud computing in the following way [21]: “A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and *virtualized* computers that are *dynamically* provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers.”

The principle of locality of execution allows to model the PADS cloud executions as an addition of a stable set and a changing set of processes (Figure 1).

The CQN model with tandems (1, … ,Q) of simple queues (q) for the stable set of processes and ((Q+1),(Q+2), …) for changing set of processes. Logical process of a tandem is assigned to a virtual machine (VM) Modelling of delay in switching between tandems can be used for modelling the PADS performance (represented by L1 and L2 of dispatching in Figure 1). (The CQN model itself is appropriate for modelling of PADS execution [3,4,9,10,14].)

B. Latency and Jitter in Cloud

In this point, the network performance of EC2 will be discussed, focusing on the measurements of packet delay measurement in spatial experiment published in [5].

In the experiments in [5], the packet round trip delay (RTT) has been measured in EC2, for 750 small instance pairs and 150 medium instance pairs using 5000 ping probes.

For the examined instance pairs, the measured hop count values were within 4 hops.

Diagram in Figure 2 (that has been built on data measured in [5]), shows the inverse cumulative distribution function (ICDF) of RTTs for small and medium instance pairs.

The diagram shows, that the RTT values among the examined instances are not stable.

Remarks for the min and max RTT values for the small and medium instance pairs:

- the max RTT values for the 95% of small instances are higher than for the medium instances. Range for small instances is are between 4 and 60 msec
- the min RTT values are higher for medium instances (average difference 0,055 msec)

C. The Coupling Factor Method

The principle of the *Coupling Factor Method (CFM)* of performance prediction [7, 9,17] may be formulated as an inequity:

$$L * E \gg \tau * P$$

where L is the lookahead value characterizing the model (simsec), E is the event density generated by the model (ev/simsec), $\tau$is the latency of messages between logical process (LPs) of the model (sec), and P is the event processing computation hardware performance (ev/sec).

According to the method, the coupling factor λ is calculated according to the formulas

$$\lambda = \frac{L * E}{\tau * P}$$

The high value of the coupling factor λ shows the good potential of the simulation model for parallelization. The formula involves only four parameters for the calculation which can be measured in simple sequential simulation runs.

For a separate process, the λ_N parallelization potential of a process is only a part of the whole potential:

$$\lambda_N = \frac{L * E}{\tau * P} * \frac{1}{N_{LP}}$$

where N_{LP} the number of the LPs [8].

D. Setting up the Scale Prediction for a Homogeneous Cluster

For the presented method, the results described in [8] will be used in the cloud performance analysis.

The hardware environment is a *homogeneous cluster* of 12 PCs. For the experiments in the case, the starting number of jobs is 2 jobs/simple queue with exponential inter-arrival time and with exponential service time distributions (the expected value for arrival and service time is 10sec). The delay on links between simple queues is 1sec. The number of tandem queues is 24 (Q), the number of simple queues/tandem queue is 50 (q). The switching in CQN model is performed by uniform distribution to switch to the next tandem queue, and the delay of switching models lookahead.

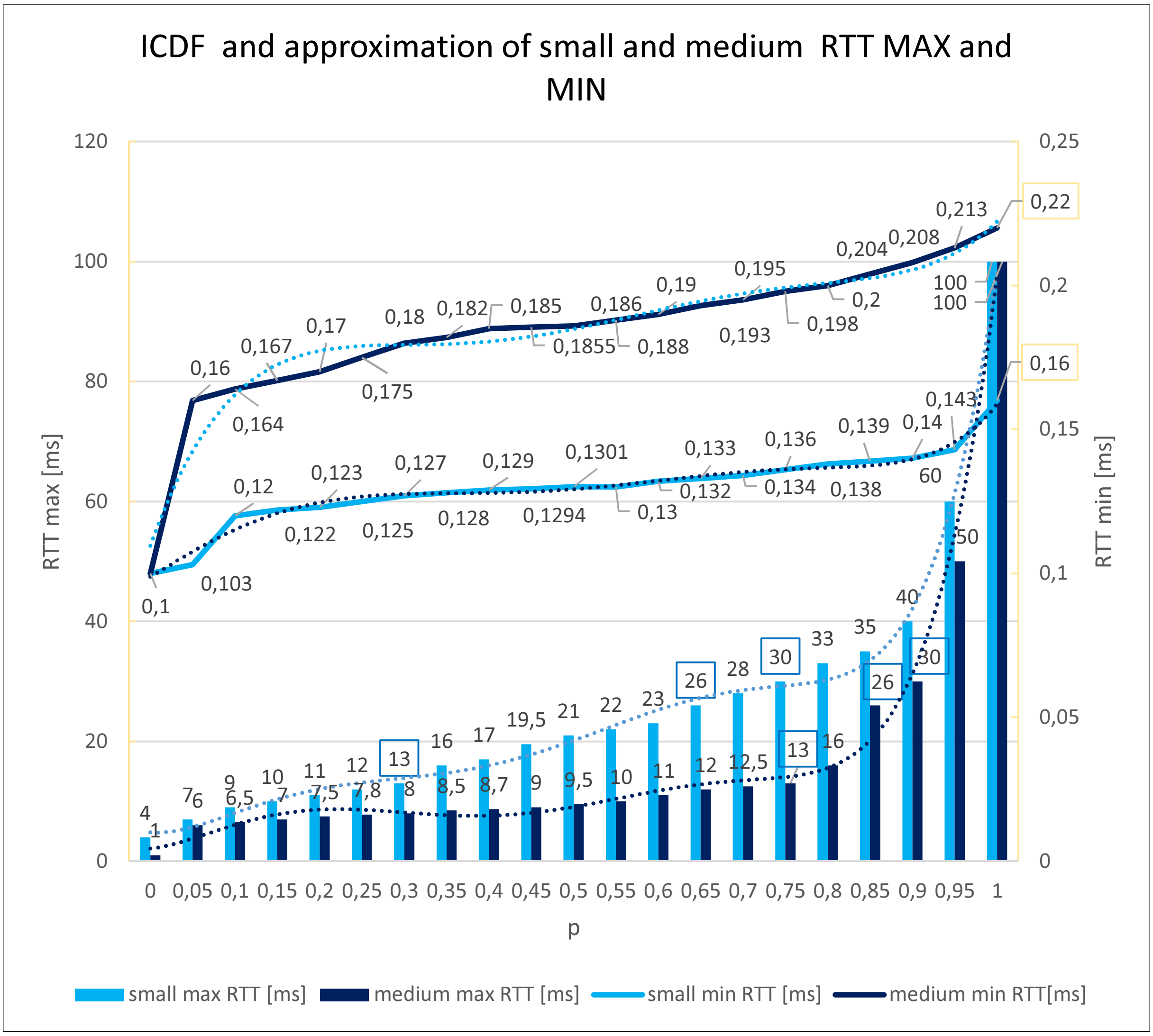

Figure 1. ICDF of max and min RTTs for small and medium sized instances in EC

2

The task assignment principles in the execution are the partitioning into LPs and Load Balancing Criterion.

TABLE I.
COUPLING FACTOR MEASUREMENT AND CALCULATION

1.	L	[simsec]	0.1	1	10	100	1000
2.	Number of events	[ev]	138122606	138091806	137816386	134885378	102957082
3.	WCT (N=1)	[sec]	524	521	523	516	416
4.	Simulated virtual time	[simsec]	864000	864000	864000	864000	864000
5.	P	[ev/sec]	263502	264868	263465	261132	247653
6.	E	[ev/simsec]	159	159	159	156	119
7.	τ	[sec]	0.000025	0.000025	0.000025	0.000025	0000025
8.	R=P/E	[simsec/sec]	1648	1657	1651	1672	2078
9.	L/ τ	[simsec/sec]	4000	40000	400000	4000000	40000000
10.	λ measured		2.43	24.1	242	2391	19246

The measured value of τ is 0,025ms for the system in the case. The value of variables that have been measured in sequential simulation runs are summarized in Table I.

TABLE II.
COUPLING FACTOR MEASUREMENT AND CALCULATION

1.		RTT(ms)	latency(ms)	τ latency	proportion	λ shift (log10)
2.	CQN model τ		0,025			
3.	EC2 average	0,2	0,1	4		0,60
4.	EC2 average	3	0,15	4,2		0,62
5.	EC2 small instances	10	5	200	55%	
6.	EC2 medium instances	20	10	400	20%	
7.	EC2 small instances weighted τ latency		2,84	113,6		2,06
8.	EC2 medium instances weighted τ latency		2,16	86,4		1,94

E. Analysis

Table II summarizes the EC2 latency measurements [5] and the calculated λ shifts in predicted relative speedup [19].

IV. EXAMPLE OF ROUGH SET MODEL OF CLOUD EXECUTION

The *RST model of simulation performance prediction* can be set up in the form of *decision information system (DIS)* and decision tables:

$I = (U, C \cup D, f_{V'}, f, V', V)$, (before *discretization and coding*)

$I = (U, A = C \cup D, f, V)$,

$I = (U, A = C \cup \{d\}, f, V), d \in D$.

The *simulation experiments* (sequential and PADS runs) will be the *objects* of the universe

$U = \{x_1, x_2, x_3, \ldots, x_{|U|}\}$

where x_i denotes the i-th simulation experiment of the universe.

The set of *condition* attributes may be defined as follows

$$C_{Cloud\ CFM\ model\ hardware-software\ features} = \{P, E, L1, L2, \lambda, \lambda_N, N_{LP}, \tau(RTT/2), N_{VMs}$$

$$simulated\ time, runtime\ (wall-clock\ time)\} \cup$$

$$C_{hardware-software\ environment}\{OMNet++, MPI, Linux\ Debian\} \cup$$

$$C_{Cloud\ CQN\ models'\ features}$$

$$\left\{\begin{matrix} conservative\ synchronisation\ protocol \\ with\ null\ message\ algorithm \end{matrix}\right\} \cup$$

$$\{Q, q, propagation\ delay\ between\ simple\ queues,$$

$$switching\ time\ between\ tandems,$$

$$starting\ number\ of\ jobs\ of\ simple\ queues\ in\ cloud\ CQN\ models,$$

$$inter\ arrival\ time\ of\ jobs,$$

$$service\ time, service\ discipline\ (FCFS)\}$$

In set C, attributes, relating to the coupling factor method, are in accordance with the *homogeneous execution environment* and with the equal (fair *load balancing* criterion too (implicit) condition attributes)

The set of *decision* attributes under consideration can be set as

$$D = \left\{\begin{matrix} speedup, R-speedup, \\ number\ of vacationing\ jobs\ in\ L1\ and\ L2 \end{matrix}\right\}.$$

The RST exploring of the coupling factor method is performed using the following decision table: *discretization and coding*

$I_t = (U, A = C \cup \{d_t\}, f, V), d_t \in D$,

The examination is executed in a form of a *train-and-test analysis*:

$$U \rightarrow (U_{training(i)}, U_{test(i)})(i = 1,2,3, \ldots, |steps|),$$

$$f_{RNDsplit(U,i)}: U \rightarrow (U_{training}, U_{test}) \text{ with}$$

$$\forall(i)(i = 1,2,\ldots)\left(U_{training(i)} \cup U_{test(i)} = U\right) \wedge$$

$$\left(|U_{training(i)}|/|U_{test(i)}| = \frac{x}{y}\right) \wedge (RND_{seed} = i).$$

For the *train-and-test* examination, the ROSETTA system [18] can be used with some selected rule generation algorithm G (for example, Johnson's RSES (Rough Set Exploration System)) and with the subsequent classification of objects.

For evaluation purposes, *re-classification rules* $S_{self-training(i)}$ can be generated for every $U_{test(i)}$ using the same rule generation and classification algorithm.

For the *efficacy, efficiency and effectiveness analysis*, [20] the simulation *cost* $K[sec]$ expressed in *computing time "consumption"* should also be calculated, including cost of *attributes*, *rules*, *simulation experiments* and *predictions,* for the series of predictions ($K(S_{training(i)}$, $i = 1,2,3, \ldots, |steps = number\ of\ predictions|$)

This analysis provides a feedback to the simulation model and to the the simulation performance model too, supporting model features identification and refinement.

V. CONCLUSION

In the paper, a model of cloud execution has been described which models the network latencies

This is a Closed Queuing Network (CQN) and Virtual Machine (VM) model of cloud work modelling network latencies among cloud instances

In the paper, the CQN and VM model has been extended in to take into account the on-demand base use of cloud resources

In an example, a rough model of prediction is introduced to support effective modelling and performance prediction

REFERENCES

[1] P. Mell, and T. Grance. The NIST definition of Cloud Computing, National Institute of Standards and Technology (NIST), Special Publication Draft-800-145, 2011. p. 2.

[2] R. M. Fujimoto, "Parallel Discrete Event Simulation", *Communications of the ACM*, vol. 33, (1990.) no 10, pp. 31-53

[3] Lencse, G., I. Derka and L. Muka. 2013. "Towards the efficient simulation of telecommunication systems in heterogeneous execution environments". submitted to the *36th International Conference on Telecommunications and Signal Processing (TSP), July 2-4, 2013, Rome, Italy.*

[4] R. L. Bagrodia, M. Takai, (2000). "Performance evaluation of conservative algorithms in parallel simulation languages", *Parallel and Distributed Systems, IEEE Transactions on*, *11*(4), 395-411.

[5] Wang, G., and Ng, T. E. (2010, March). The impact of virtualization on network performance of amazon ec2 data center, In INFOCOM, 2010 Proceedings IEEE, pp. 1-9.

[6] Xu, Y., Musgrave, Z., Noble, B., Bailey, M. (2013, April). Bobtail: Avoiding Long Tails in the Cloud. In NSDI pp. 329-341.

[7] A. Varga, Y. A. Sekercioglu and G. K. Egan. "A practical efficiency criterion for the null message algorithm", *Proceedings of the European Simulation Symposium (ESS 2003)*, (Oct. 26-29, 2003, Delft, The Netherlands) SCS International, 81-92.

[8] G. Lencse and A. Varga, "Performance Prediction of Conservative Parallel Discrete Event Simulation", *Proceedings of the 2010 Industrial Simulation Conference (ISC'2010)* (Budapest, Hungary, 2010. June 7-9.) EUROSIS-ETI, 214-219.

[9] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment", Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, (Marseille, France, March 3-7, 2008) 1-10.

[10] A. Varga and A. Y. Sekercioglu, "Parallel Simulation Made Easy with OMNeT++", *Proceedings of the 15th European Simulation*

Symposium (ESS 2003), (Oct. 26-29, 2003, Delft, The Netherlands) SCS International, 493-499

[11] Z. Juhasz, S. Turner, K. Kuntner, M. Gerzson, "A Performance Analyser and Prediction Tool for Parallel Discrete Event Simulation",*International Journal of Simulation*, vol. 4, no. 1, May 2003. pp. 7-22.

[12] Kunz, G., Tenbusch, S., Gross, J., Wehrle, K. (2011, July). "Predicting Runtime Performance Bounds of Expanded Parallel Discrete Event Simulations. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on* (pp. 359-368). IEEE.

[13] R. Ewald, A. M. Uhrmacher, "Automating the runtime performance evaluation of simulation algorithms", In: *Simulation Conference (WSC), Proceedings of the 2009 Winter*. IEEE, 2009. pp. 1079-1091.

[14] Lencse, G. and I. Derka 2013. "Testing the Speed-up of Parallel Discrete Event Simulation in Heterogeneous Execution Environments". submitted to the *2010 Industrial Simulation Conference (ISC'2010)* (Ghent, Belgium, May 22-24, 2013.)

[15] Malik, A. W., Park, A. J., Fujimoto, R. M. (2010). An optimistic parallel simulation protocol for cloud computing environments", *SCS Modeling and Simulation Magazine*, 1(4).

[16] Fujimoto, R. M., Malik, A. W., Park, A. (2010). "Parallel and distributed simulation in the cloud*" SCS M&S Magazine,* 3, pp. 1-10.

[17] Varga, A.; Hornig, R., 2008. "An overview of the OMNeT++ simulation environment", In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering),

[18] Komorowski, J.; Øhrn, A.; Skowron, A., 2002. "The ROSETTA Rough Set Software System", In Handbook of Data Mining and Knowledge Discovery, W. Klösgen and J. Zytkow (eds.), Oxford University Press, ch. 2-3, ISBN 0-19-511831-6.

[19] Muka,L.; I Derka, I.; 2014. "Simulation Performance Prediction in Clouds", In: Orosz Gábor Tamás (szerk.), 9th International Symposium on Applied Informatics and Related Areas – AIS2014., Konferencia helye, ideje: Székesfehérvár, Magyarország, 2014.11.12 Székesfehérvár, Óbudai Egyetem, pp. 142-147., ISBN:978-615-5460-21-0

[20] Muka,L.; I Derka, I.; 2015. "Rough Set Based Efficiency Improvement of Simulation Performance Prediction", In: Orosz Gábor Tamás (szerk.), 10th International Symposium on Applied Informatics and Related Areas – AIS2015, Székesfehérvár, Magyarország, Székesfehérvár, Óbudai Egyetem (accepted for publication)

[21] Buyya, R., Yeo, C. S.; Venugopal, S., Broberg, J., Brandic, I.; 2009. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Generation computer systems*, 25(6), pp. 599-616.

BIOGRAPHIES

LÁSZLÓ MUKA graduated in electrical engineering at the Technical University of Lvov in 1976. He got his special engineering degree in digital electronics at the Technical University of Budapest in 1981, and became a university doctor in architectures of CAD systems in 1987. Mr. Muka finished an MBA at Brunel University of London in 1996. Since 1996 he has been working in the area of modeling and simulation of infocommunications systems, including human subsystems. Mr. Muka got his PhD at the Budapest University of Technology and Economics in 2011. Dr. Muka has been working as an Associate Professor for the Széchenyi István University in Győr since the February of 2012. He teaches system simulation and network security

ISTVÁN DERKA received his Msc at Faculty of Electrical Engineering and Informatics at the Technical University of Budapest in 1995. He worked for the Department of Informatics from 1999 to 2003 and since then has been working for Department of Telecommunications, Széchenyi István University in Győr. He teaches Programming of communication systems and Interactive TV systems. He is an Assistant Professor. The area of his research includes multicast routing protocols and IPTV services in large scale networks.