ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

UNCERTAINTY QUANTIFICATION USING CLOUD COMPUTING FOR MONTE CARLO PARALLELIZATION

Americo Cunha Jr
Rubens Sampaio
Department of Mechanical Engineering
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225, Gávea, Rio de Janeiro - RJ, Brasil - 22453-900
americo.cunhajr@gmail.com — rsampaio@puc-rio.br

Rafael Nasser
Hélio Lopes
Karin Breitman
Department of Informatics
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225, Gávea, Rio de Janeiro - RJ, Brasil - 22453-900
rafael.nasser@globo.com — lopes@inf.puc-rio.br — karin@inf.puc-rio.br

Abstract. *Due to its simplicity and good statistical results, the Monte Carlo (MC) method is the most commonly technique used for uncertainty quantification. However, its computational cost is significant, and, in many cases, prohibitive. Fortunately, the MC algorithm can be can be parallelized, which may allows its use in complex simulations. In this sprit, this work presents a methodology for the parallelization of MC method in a cloud computing setting. The methodology is illustrated on a stochastic problem of structural dynamics, and the simulation results show good accuracy for low-order statistics. Also, this methodology shows a good performance in terms of processing-time and storage space usage.*

Keywords: *uncertainty quantification, Monte Carlo method, parallel algorithm, cloud computing, MapReduce*

1. INTRODUCTION

Nowadays, most of the predictions that are necessary for decision making in engineering, economics, actuarial sciences, etc., are made based on computer models, which are subjected to uncertainties due to wrong assumptions made on their conception and/or variabilities on model parameters (Soize, 2013). The most successful approach to take into account these uncertainties is one that uses the probability theory to describe the uncertain parameters as random variables, random process, and/or random fields. This approach allows one to obtain a model where it is possible to quantify the variability on the response. For instance, the reader can see Ritto *et al.* (2013), which apply techniques of stochastic modeling to describe the dynamics of a drillstring. It is also worth of mentioning the contributions of Zio and Rochinha (2012), in the context of hydraulic fracturing, Lopes *et al.* (2012), for estimation of financial reserves, Clément *et al.* (2013), for the analysis of structures built by heterogeneous hyperelastic materials.

To compute the propagation of uncertainties of the random parameters through the model, the most used technique in the literature is the Monte Carlo (MC) method (Robert and Casella, 2010). Among the good features of the MC method we can highlight the guarantee of convergence and its non-intrusive nature, which allow us to perform the simulation of the stochastic model using only a deterministic code, which is executed several times. Unfortunately, the MC is a very time-consuming method.

This work presents a methodology for implementing the MC method, in a cloud computing setting, which is inspired in the MapReduce paradigm (Dean and Ghemawat, 2004). This approach consists in splitting, among several instances of the cloud environment, the MC calculation, process each one of these tasks in parallel, and finally merge the results into a single instance to compute the statistics. As an example, the methodology is applied to a simple problem of stochastic structural dynamics.

This paper is organized as follows. The section 2. presents a parallelization strategy for MC method in the context of cloud computing. The section 3. describes the case of study in which the proposed methodology is exemplified. The section 4. presents and discusses the statistics done with the data and the convergence of the results. Finally, the section 5. presents the conclusions and highlight the main contribution of this work.

ISSN 2176-5480

2. PARALLELIZATION OF MONTE CARLO METHOD

We propose a MapReduce (Dean and Ghemawat, 2004) strategy for parallel execution of MC method in the cloud that is composed by three steps: *split*, *process* (Map), and *merge* (Reduce). This strategy of parallelization was implemented in a cloud computing setting called McCloud (Nasser, 2012; Nasser *et al.*, 2013), which runs in Microsoft Windows Azure platform (http://www.windowsazure.com). An general overview of the strategy can be seen in Figure 1, and a detailed description of each step is made below.

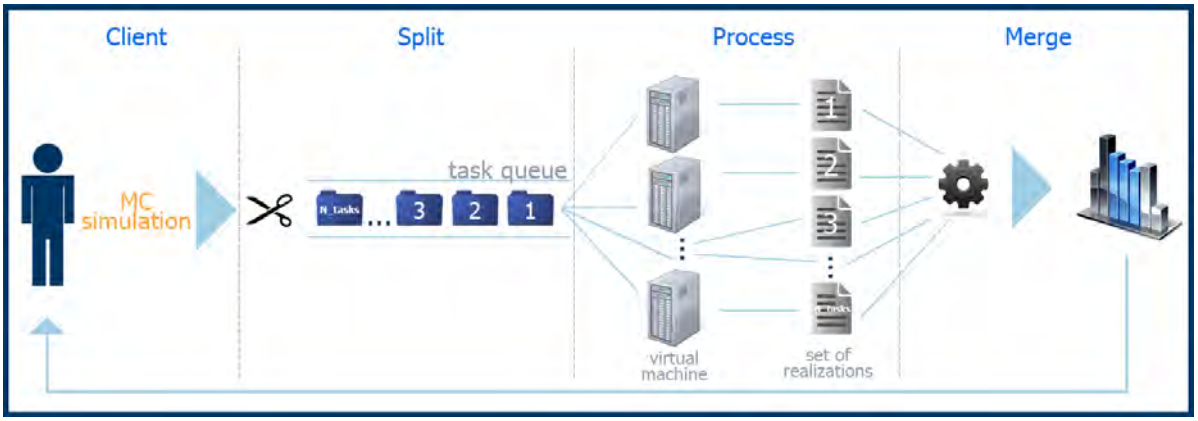


Figure 1. General overview of Monte Carlo parallelization strategy in the cloud.

2.1 Split

First of all, the split step choose the number of cloud virtual machines to be used and turn then on. Then, it divides a MC simulation with N_{MC} realizations into tasks and put then into a queue. Each one of these tasks is composed by an ensemble of N_{serial} realizations to be simulated. Thus, it is necessary to process a number of tasks equal to $N_{tasks} = N_{MC}/N_{serial}$. These tasks are distributed in a manner (approximately) uniformly among the virtual machines.

In the simulations above, the realizations of the random parameters are obtained by the use of a pseudorandom numbers generator. This generator construct deterministically, and indexed by the value of a statistical seed, a sequence of numbers that emulates a set of random numbers. To avoid the possibility of repeated seeds, and hence, redundancy, it is adopted a strategy of seed distribution among the virtual machines which guarantees that the sequence of random numbers generated in each one these machines is different from the sequences generated on the other machines.

2.2 Process

The process step uses the available virtual machines to pick a process, asynchronously, task by task in the queue. The output data generated by each executed task is saved into the hard disk for subsequent post-processing. So, the total amount of storage space used by a MC simulation is proportional to the number of realizations. Therefore, this step also requires attention in terms of storage space usage, because the demand of hard disk space may become unfeasible for a simulation with a large number of realizations.

To reduce the storage space usage in the example of section 4. we chose to calculate the mean and standard deviation using the strategy of pairwise parallel and incremental updates of the statistical moments described in Bennett *et al.* (2009), which uses the Welford-Knuth algorithm. Thus, for each executed task, instead of saving all the simulation data for subsequently calculation of the statistical moments, we save only the mean and the centered sum of squares. For the calculation of the histograms, the random variables of interest were identified before the processing step, and their realizations are saved for being used in the histogram construction, during the post-processing step.

2.3 Merge

The merge step, starts when a the last task in a virtual machine finishes, and this can occur in any virtual machine. This step read, from the hard disk, all the information contained in the output data from the simulations, and combines then through statistics to obtain relevant information about the problem under analysis. At the end of this stage, the saved data is discarded and only the merged result is stored in order to reduce future costs of data storage.

ISSN 2176-5480

3. AN EXAMPLE IN STOCHASTIC STRUCTURAL DYNAMICS

3.1 Physical System

The system of interest in this study case is the elastic bar fixed at a rigid wall, on the left side, and attached to a lumped mass and two springs (one linear and one nonlinear), on the right side, such as illustrated in Figure 2. The stochastic nonlinear dynamics of this system was investigated by Cunha Jr and Sampaio (2012, 2013a,b), where the reader can see more details about the modeling procedure presented below. For simplicity, from now on, this system will be called the fixed-mass-spring bar or simply the bar.

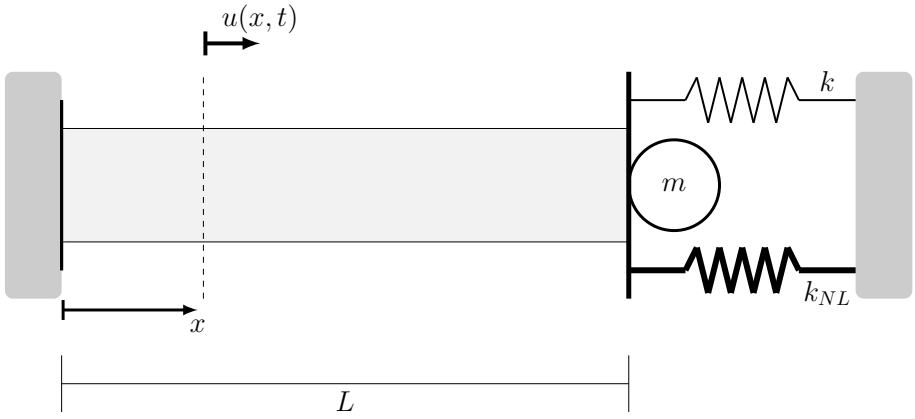


Figure 2. Sketch of a bar fixed at one and attached to two springs and a mass on the other extreme.

3.2 Strong Formulation

The physical quantity of interest is the bar is its displacement field u, which depends on the position x and the time t, and evolves, for all $(x,t) \in (0,L) \times (0,T)$, according to the following partial differential equation

$$\rho A \frac{\partial^2 u}{\partial t^2} + c \frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(EA \frac{\partial u}{\partial x}\right) + \left(ku + k_{NL}u^3 + m \frac{\partial^2 u}{\partial t^2}\right)\delta(x-L) = f(x,t), \tag{1}$$

where ρ is mass density, A is the cross section area, c is the damping coefficient, E is the elastic modulus, k is the stiffness of the linear spring, k_{NL} is the stiffness of the nonlinear spring, m is the lumped mass, and f is a distributed external force, which depends on x and t. The symbol $\delta(x-L)$ denotes Dirac's delta distribution at $x = L$.

For this problem the boundary conditions are given by

$$u(0,t) = 0, \quad \text{and} \quad EA \frac{\partial u}{\partial x}(L,t) = 0, \tag{2}$$

while the initial position and the initial velocity are respectively given by

$$u(x,0) = u_0(x), \quad \text{and} \quad \frac{\partial u}{\partial t}(x,0) = v_0(x), \tag{3}$$

being u_0 and v_0 known functions of x, defined for $0 \leq x \leq L$.

3.3 Weak Formulation

Let $\mathcal{U}_t$ be the space of (time-dependent) basis functions and $\mathcal{W}$ be the space of weight functions, both assumed to be sets of square integrable functions that satisfy the essential boundary condition given by Eq.(2). The weak formulation, to the problem defined by Eqs.(1) to (3), says to find a function $u \in \mathcal{U}_t$ such that, for all $w \in \mathcal{W}$, satisfy

$$\mathcal{M}(\ddot{u}, w) + \mathcal{C}(\dot{u}, w) + \mathcal{K}(u, w) = \mathcal{F}(w) + \mathcal{F}_{NL}(u, w), \tag{4}$$

ISSN 2176-5480

$$\widetilde{\mathcal{M}}(u(\cdot,0),w) = \widetilde{\mathcal{M}}(u_0,w), \quad \text{and} \quad \widetilde{\mathcal{M}}(\dot{u}(\cdot,0),w) = \widetilde{\mathcal{M}}(v_0,w), \tag{5}$$

where $\mathcal{M}$ is the mass operator, $\mathcal{C}$ is the damping operator, $\mathcal{K}$ is the stiffness operator, $\mathcal{F}$ is the external force operator, $\mathcal{F}_{NL}$ is the nonlinear force operator, is the $\widetilde{\mathcal{M}}$ is the associated mass operator, and the upper dot is an abbreviation for the time derivative.

3.4 Galerkin Formulation

To approximate the solution of the variational problem given by Eqs.(4) to (5), we employ the Galerkin method (Hughes, 2000). This results in the following system of ordinary differential equations

$$[M]\,\ddot{\mathbf{u}}(t) + [C]\,\dot{\mathbf{u}}(t) + [K]\,\mathbf{u}(t) = \mathbf{f}(t) + \mathbf{f}_{NL}\left(\dot{\mathbf{u}}(t)\right), \tag{6}$$

supplemented by the following pair of initial conditions

$$\mathbf{u}(0) = \mathbf{u}_0 \qquad \text{and} \qquad \dot{\mathbf{u}}(0) = \mathbf{v}_0, \tag{7}$$

where $\mathbf{u}(t)$ is the vector of unknowns in $\mathbb{R}^N$, $[M]$ is the mass matrix, $[C]$ is the damping matrix, $[K]$ is the stiffness matrix. Also, $\mathbf{f}(t)$, $\mathbf{f}_{NL}\left(\mathbf{u}(t)\right)$, $\mathbf{u}_0$, and $\mathbf{v}_0$ are vectors of $\mathbb{R}^N$, which respectively represent the external force, the nonlinear force, the initial position, and the initial velocity. The initial value problem of Eqs.(6) and (7) has its solution approximated by Newmark method (Hughes, 2000).

3.5 Stochastic Model

In order to introduce randomness in the bar model, we consider a probabilistic space $(\Theta, \mathbb{A}, \mathbb{P})$, where Θ is sample space, $\mathbb{A}$ is a σ-field over Θ, and $\mathbb{P} : \mathbb{A} \to [0,1]$ is a probability measure. In this probabilistic space, the external force f is assumed to be the random field $F : [0,L] \times [0,T] \times \Theta \to \mathbb{R}$, which is proportional to a normalized Gaussian white noise. Also, the elastic modulus is assumed to be a random variable $E : \Theta \to \mathbb{R}$.

The probability distribution of E is characterized by its probability density function (PDF) $p_E : (0,\infty) \to \mathbb{R}$, which is gamma distributed. To specify the distribution of E we have used the maximum entropy principle (Soize, 2013), based only on the known information about this parameter (the mean, the second statistical moment, and the mean of the parameter logarithm).

Due to the randomness of F and E, the bar displacement becomes a random field $U : [0,L] \times [0,T] \times \Theta \to \mathbb{R}$, which satisfies a initial/boundary value problem similar to the one defined by Eqs.(1) to (3), changing u by U only.

4. NUMERICAL EXPERIMENTS

We employ the MC method in the cloud to approximate the solution of the stochastic initial/boundary value problem defined in the section 3.5. This procedure uses a sampling strategy with the number of realizations always being equal to a power of four. In this procedure, each realization of the random parameters defines a new variational problem given by Eqs.(4) to (5). Each one of these variational problems is solved deterministically as described in section 3.4 Then, these results are combined through the statistics that are shown in what follows.

4.1 Probability Density Function

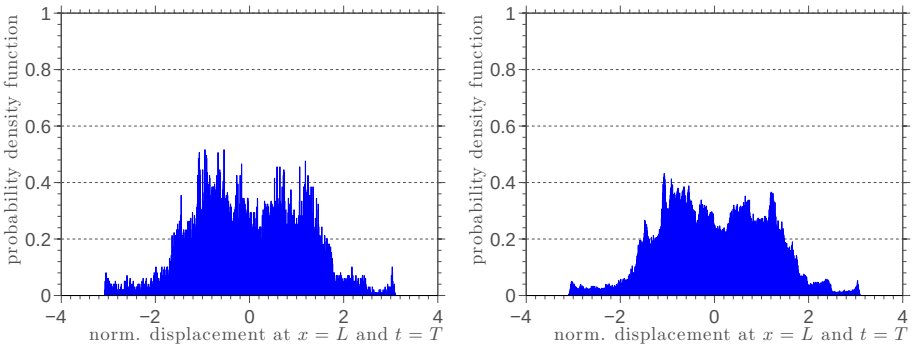
The estimations for the PDF of the (normalized[1]) bar right extreme displacement, for a fixed instant of time, is shown in Figure 3, for different values of the total number of realizations in MC simulations. We can note that, as the number of samples in MC simulation increases, small differences may be noted on the peaks of successive estimations of the PDF.

We use a convergence criterion based on a residue of the random variable $U(L,T,\theta)$, defined as the absolute value of the difference between two successive approximations of $p_{U(L,T,\cdot)}$, i.e.,

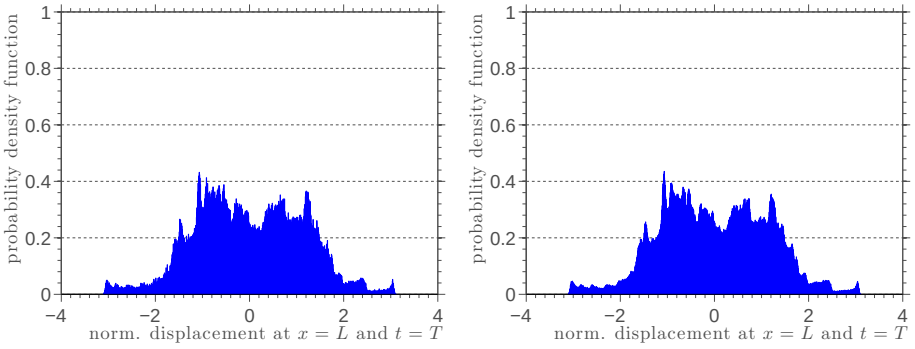
$$R_{U(L,T,\cdot)} = \left| p^{4n}_{U(L,T,\cdot)} - p^{n}_{U(L,T,\cdot)} \right|, \tag{8}$$

[1]By normalized we want to say a random variable with zero mean and unit standard deviation

ISSN 2176-5480



(a) 16 384 realizations (b) 65 536 realizations



(c) 262 144 realizations (d) 1 048 576 realizations

Figure 3. This figure illustrates estimations for the PDF of the (normalized) random variable $U(L, T, \cdot)$, for several values of MC realizations.

where the superscript n indicates the number of realizations in the MC simulation. In this case, we say that the MC simulation reached a satisfactory result if this residue is less than a prescribed tolerance ϵ, i.e., $R_{U(L,T,\cdot)} < \epsilon$ for all $\theta \in \Theta$. For instance, $\epsilon = 0.05$.

The reader can observe the distribution of the residue of $U(L, T, \theta)$, for several values of MC realizations, in the Figure 4. Note that although the residue decrease with the increase of the MC realizations, only one simulation with 1 048 576 samples was able to fulfill the convergence criterion. Note that, despite the number of samples used in MC simulation be very high and the problem be relatively simple, the tolerance achieved was relatively low. It is common to observe in the literature works that analyze problems much more complex, for instance Spanos and Kontsos (2008), Liang and Mahadevan (2011) and Murugan *et al.* (2012), among many others, that use some hundreds or a few thousands of samples. In this context, the use of MC in a cloud computing setting appears to be a viable solution, able to make the work feasible at a low-cost.

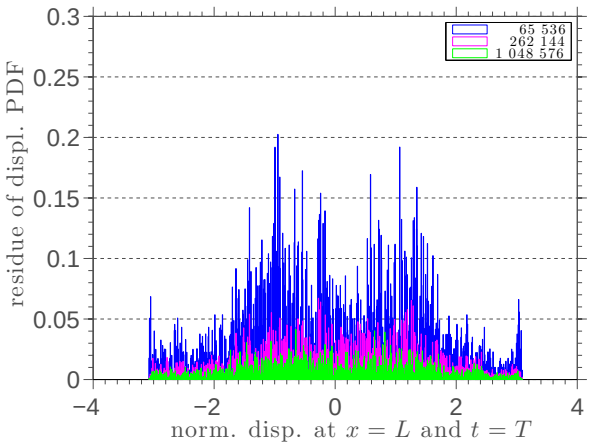
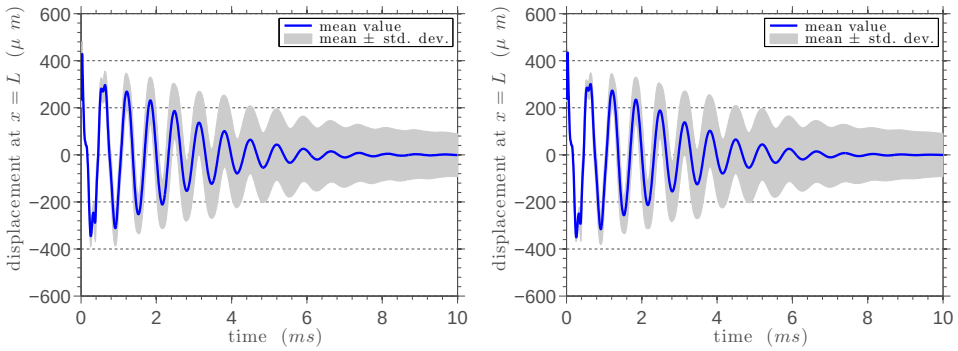


Figure 4. This figure illustrates the residue of the $U(L, T, \cdot)$ PDF.

ISSN 2176-5480

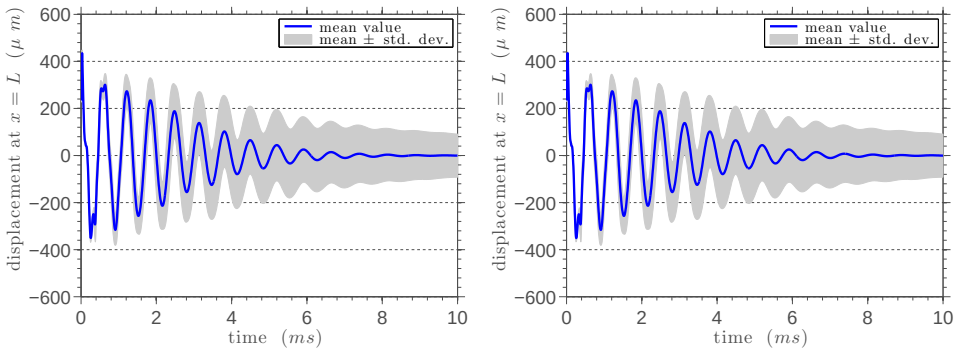
4.2 Mean and Standard Deviation

The Figure 5 shows the evolution of the bar right extreme displacement mean (blue line) and an envelope of reliability (grey shadow) around it, obtained by adding and subtracting one standard deviation around the mean. This figure shows these graphs for different values of the total number of realizations in MC simulations. We can conclude that the low-order statistics, from the qualitative point of view, do not undergo major changes when the total number of realizations is higher than 16 384.



(a) 16 384 realizations

(b) 65 536 realizations



(c) 262 144 realizations

(d) 1 048 576 realizations

Figure 5. This figure illustrates the mean value (blue line) and a confidence interval (grey shadow) with one standard deviation, of the random process $U(L, \cdot, \cdot)$, for several values of MC realizations.

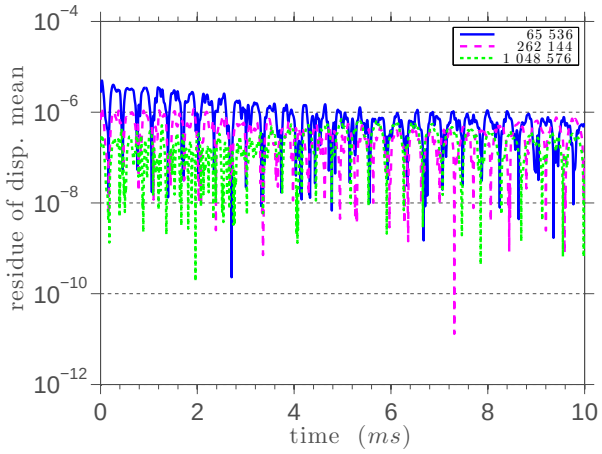


Figure 6. This figure illustrates the evolution of residue of the $U(L, \cdot, \cdot)$ mean.

ISSN 2176-5480

The Figure 6 illustrates the evolution of several residues[2] of $U(L,\cdot,\cdot)$ mean, and the Figure 7 illustrates the evolution of several residues of $U(L,\cdot,\cdot)$ standard deviation. We note that, the logarithm of the mean value residues are almost always less than $\mathcal{O}(10^{-6})$, for the case of statistics with larger samples, and presents an alternate behavior between big drops and climbs, as can be seen in Figure 6. On the other hand, the logarithm of the standard deviation residue is greater than $\mathcal{O}(10^{-6})$ in the initial instants. This behavior is not maintained after 2 ms, when the residue curves keeps their alternate behavior, but almost always below $\mathcal{O}(10^{-6})$, as shown in Figure 7. These results show that statistics of first and second order may be obtained with great accuracy using the methodology presented in this work.

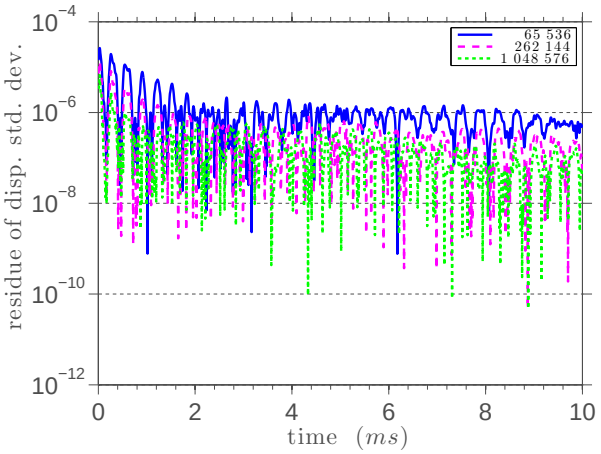


Figure 7. This figure illustrates the evolution of residue of the $U(L,\cdot,\cdot)$ standard deviation.

4.3 Costs Analysis

The Table 1 allows us to make a comparison between the computational time spent by each one of MC simulations in the cloud, and the corresponding speed-up factors compared to a serial[3] simulation. Note that, in all of the experiments the strategy of parallelism in the cloud provided performance gains, and such gains are greater as the number of realizations.

Table 1. Comparison between the computational time spent by each one of MC simulations in the cloud, and the corresponding speed-up factors compared to a serial simulation.

N_{MC}	tasks	VMs	split (ms)	process (ms)	merge (ms)	total (min)	serial (min)	speed-up
256	1	1	—	111 998	2 250	1,9	1,9	1,0
16 384	64	19	1 391	2 188 705	2 094	36,5	121,6	3,3
65 536	256	19	2 516	4 475 018	29 281	75,1	486,4	6,5
262 144	1 024	19	8 109	7 905 771	90 673	133,3	1 945,6	14,6
1 048 576	4 096	19	33 734	28 804 980	355 342	486,0	7 782,4	16,0

A comparison between the storage space used and the financial cost associated to each simulation can be seen in the Table 2, in which the column *space* represents the total (temporarily) storage space, in MB, and the column *cost* represents the cost of this experiment in US dollar. We can see that our parallelization strategy used fairly little storage space, even for a large number of MC realizations. Also, the financial cost, even for the most complex simulation, is very small, a major advantage when compared to the costs of acquiring and maintaining a traditional cluster.

[2]We have defined the residue of $U(L,\cdot,\cdot)$ mean/standard deviation similarly to the one defined to the Eq.(8), by changing the PDF for the mean or standard deviation of $U(L,\cdot,\cdot)$ only.

[3]This time was obtained by extrapolation of the processing time spent to run a single task, which uses $N_{serial} = 256$.

ISSN 2176-5480

Table 2. Comparison between the storage space used by each one of MC simulations in the cloud, and the financial cost associated to each simulation.

N_{MC}	space (MB)	cost ($)
16 384	12,4	5,39
65 536	49,0	7,67
262 144	195,2	9,95
1 048 576	780,8	23,63

5. FINAL REMARKS

We present a strategy for parallelizing the Monte Carlo method in the context of cloud computing, using the fundamental idea of the MapReduce paradigm. This strategy is described and illustrated in of simple stochastic problem of structural dynamics. The results show good accuracy for mean and variance, low storage space usage, and gains of performance that increase with the number of Monte Carlo realizations. Due to its high scalability capacity and low-cost, the cloud computing strategy shows to be very attractive for MC simulations which demands a large number of realizations for convergence.

6. ACKNOWLEDGMENTS

The authors acknowledge the Brazilian agencies CNPq, CAPES, and FAPERJ for the financial support.

7. REFERENCES

Bennett, J., Grout, R., Pebay, P., Roe, D. and Thompson, D., 2009. "Numerically stable, single-pass, parallel statistics algorithms". In *IEEE International Conference on Cluster Computing and Workshops 2009*. doi:10.1109/CLUSTR.2009.5289161.

Clément, A., Soize, C. and Yvonnet, J., 2013. "Uncertainty quantification in computational stochastic multiscale analysis of nonlinear elastic materials". *Computer Methods in Applied Mechanics and Engineering*, Vol. 254, pp. 61–82. doi:10.1016/j.cma.2012.10.016.

Cunha Jr, A. and Sampaio, R., 2012. "Effect of an attached end mass in the dynamics of uncertainty nonlinear continuous random system". *Mecánica Computacional*, Vol. 31, pp. 2673–2683.

Cunha Jr, A. and Sampaio, R., 2013a. "Analysis of the nonlinear stochastic dynamics of an elastic bar with an attached end mass". In *Proceedings of the III South-East Conference on Computational Mechanics*.

Cunha Jr, A. and Sampaio, R., 2013b. "Uncertainty propagation in the dynamics of a nonlinear random bar". In *Proceedings of the XV International Symposium on Dynamic Problems of Mechanics*.

Dean, J. and Ghemawat, S., 2004. "MapReduce: Simplified data processing on large clusters". In *OSDI*.

Hughes, T.J.R., 2000. *The Finite Element Method*. Dover Publications, New York.

Liang, B. and Mahadevan, S., 2011. "Error and uncertainty quantification and sensitivity analysis in mechanics computational models". *International Journal for Uncertainty Quantification*, Vol. 1, pp. 147–161. doi:10.1615/IntJUncertaintyQuantification.v1.i2.30.

Lopes, H., Barcellos, J., Kubrusly, J. and Fernandes, C., 2012. "A non-parametric method for incurred but not reported claim reserve estimation". *International Journal for Uncertainty Quantification*, Vol. 2, pp. 39–51. doi:10.1615/Int.J.UncertaintyQuantification.v2.i1.40.

Murugan, S., Chowdhury, R., Adhikari, S. and Friswell, M., 2012. "Helicopter aeroelastic analysis with spatially uncertain rotor blade properties". *Aerospace Science and Technology*, Vol. 16, pp. 29 –39. doi:10.1016/j.ast.2011.02.004.

Nasser, R., 2012. *McCloud Service Framework: Development Services of Monte Carlo Simulation in the Cloud*. M.Sc. Dissertation, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro. (in portuguese).

Nasser, R., Cunha Jr, A., Lopes, H., Breitman, K. and Sampaio, R., 2013. *McCloud: Easy and quick way to run Monte Carlo simulations written in MATLAB, C or R in the cloud*. Pre-Print, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

Ritto, T.G., Escalante, M.R., Sampaio, R. and Rosales, M.B., 2013. "Drill-string horizontal dynamics with uncertainty on the frictional force". *Journal of Sound and Vibration*, Vol. 332, pp. 145–153. doi:10.1016/j.jsv.2012.08.007.

ISSN 2176-5480

Robert, C.P. and Casella, G., 2010. *Monte Carlo Statistical Methods*. Springer, New York.

Soize, C., 2013. "Stochastic modeling of uncertainties in computational structural dynamics - recent theoretical advances". *Journal of Sound and Vibration*, Vol. 332, pp. 2379—2395. doi:10.1016/j.jsv.2011.10.010.

Spanos, P. and Kontsos, A., 2008. "A multiscale Monte Carlo finite element method for determining mechanical properties of polymer nanocomposites". *Probabilistic Engineering Mechanics*, Vol. 23, pp. 456–470. doi:10.1016/j.probengmech.2007.09.002.

Zio, S. and Rochinha, F., 2012. "A stochastic collocation approach for uncertainty quantification in hydraulic fracture numerical simulation". *International Journal for Uncertainty Quantification*, Vol. 2, pp. 145–160. doi:10.1615/Int.J.UncertaintyQuantification.v2.i2.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.