Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

MODELING THE COST TO EXECUTE A DEVS MODEL

Guillermo G. Trabes
Gabriel A. Wainer
Department of Systems and Computer
Engineering, Carleton University
1125 Colonel By Dr.
Ottawa, ON K1S 5B6, CANADA

Veronica Gil Costa

Universidad Nacional de San Luis
and CCT CONICET San Luis
Ejercito de Los Andes 950
San Luis, D5700, ARGENTINA

ABSTRACT

The Discrete Event System Specification (DEVS) is a very extended formalism to precisely define discrete event systems and simulate them. As more complex systems are developed, the performance of DEVS models is becoming a crucial subject. To execute a DEVS model, protocols were designed for the classical DEVS model and for the Parallel DEVS model (PDEVS) but no work have formalized the cost to execute a model with these protocols. In this work we propose a definition of the cost and a new metric to analyze model's executions.

1 INTRODUCTION

Discrete Event System Specification (DEVS) is a mathematical formalism for modeling and simulating discrete-event dynamic systems. DEVS decomposes the system into basic models called atomic models and composite models called coupled models. One big advantage of DEVS is its ability to separate the models from the engine that execute them. Nevertheless, to obtain high performance models it is important to count with tools that allow to analyze how the models execute. Well defined protocols exist to execute a DEVS model, the classical DEVS protocol and the parallel DEVS protocol (PDEVS) (Chow 1994) are the most important ones. We propose a definition for the cost to execute a model on both protocols and a new metric to calculate the expected cost for the coupled components.

2 BACKGROUND

The DEVS simulation protocol is composed by a Coordinator, and one or more Simulators, each with an associated model. During simulation, the interaction/communication between different model components is achieved through event messages exchanged between the Simulators and Coordinators, each representing an event to be processed.

The execution of the model is achieved through *global transitions* on which the coordinator receives by all the coupled models, the time of next event t_N and calculates the minimum. Imminent components, whose t_N equals the minimum, compute their outputs and send them to receivers. The active components, imminents and their receivers, then compute their state transition functions (internal, external, or confluent, depending on whether they're imminent or have inputs). In classic DEVS, whenever two coupled models are scheduled for state transitions at the same time, an execution order is chosen according to a *select* function provided in the coupled model specification. Classical DEVS was extended to resolve serialization constraints and allow simultaneous events, defining what is called Parallel DEVS (PDEVS).

3 DEVS EXECUTION MODELLING

On the classical and parallel DEVS protocols, a model execution will be completed after a finite sequence of global transitions, M. A DEVS model is composed by a set *Coupled* $=\{C_1,..,C_N\}$ of coupled models, on every global transition *k*, a subset $Active_k=\{I_1,..,I_S\}\subseteq$ *Coupled* will execute, the *Active* subset involve *imminents* and *receivers* coupled models. The complexity cost to execute a model on the classical and the parallel DEVS protocols is given by the maximum of the cost of the coupled components, but to define the cost to execute any given algorithm a more detailed analysis is required. The total cost to execute a model with the Classical and Parallel DEVS protocol algorithms is the sum of the execution cost of every global transition and we define it in the following way:

- On the classical DEVS protocol the execution is made by a single processor. The cost of executing a model is the following:

  $$\sum_{k=0}^{M-1} \sum_{i \in Active_k} cost_i \tag{1}$$

  The cost for every global transition, is the sum of the cost for the active components.
- On the PDEVS protocol, the *N* coupled models are divided among the *p*≥1 processors. For every global transition *k*, each processor j will have an associated set $Active_{j,k}=\{I_1,..,I_S\} \subseteq$ *Coupled* models to execute.

  $$\sum_{k=0}^{M-1} max(\sum_{i \in Active_{j,k}} cost_i) \tag{2}$$

  As the processors execute concurrently, the computational cost for each global transition will be the maximum computational cost of the processors.

Two factors play a key role on the cost of execution in both protocols. In first place, how often each coupled model becomes *active*, this can be model as proposed in (Ziegler 2017), where a probability 0≤*prob≤1* is assigned to each coupled model. In second place, the execution cost of every coupled model is a factor to consider, costly components (even if they have low probability to become active) will have a great impact on the model's execution. With these two factors the following metric for every C ϵ *Coupled*:

$$ExpectedComputationCost\ = (prob_C * cost_C) \tag{3}$$

where $prob_C$ is the probability of C becoming active on a global transition and $cost_C$ is the cost in number of operation that C needs to perform. This metric can be calculated by doing a profile in a model's execution. With formula (3) the component with the maximum expected computing cost can be identify in the following way:

$$max_{i \in Coupled}(prob_i * cost_i) \tag{4}$$

In addition, for the PDEVS protocol the model's division also plays a key role on the performance. Formula (3) can also be used to provide a division aiming to reduce the expected execution cost on every processor. As a first approach, we propose the following algorithm for components division:

1. Compute Formula (3) for every coupled model.
2. Sort the results obtained in (1) in descendent order.
3. Divide the components between the processors in a round-robin way.

This algorithm aims to balance the sum of the *ExpectedComputationCost* of the models assigned to each processor.

4 CONCLUSIONS AND FUTURE WORK

In this work we propose a definition for the cost to execute a DEVS model and a new metric to analyze the model's performance. Understand and analyze models in terms of its execution allow to design more efficient ones and to compare them.

On PDEVS implementations, a suited work division can be made to minimize the model's cost. One interesting topic not discussing on this work is the overhead on the execution caused by the communication and synchronization between the processors. To include these factors, we propose to model further the execution cost with parallel computing models as the bulk-synchronous parallel (BSP) (Valiant 1990). This will allow to predict the execution time for a model, including the overhead of communication and synchronization between processors.

Also, as future work we aim to provide empirical results for the metric defined in this work.

REFERENCES

Chow A. C. and Zeigler B. P. 1994. "Revised DEVS: a Parallel, Hierarchical, Modular Modeling Formalism". In: *Proceedings of the 1994 Winter Simulation Conference (WSC),* edited by J. D. Tew, M. Manivannan, D. A. Sadowski, and F. Seila, 716-722. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Valiant, L. G. 1990. "A Bridging Model for Parallel Computation". *Communications of ACM* 33(8):103-111.

Ziegler, B. P. 2017. "Using the Parallel DEVS Protocol for General Robust Simulation with Near Optimal Performance". *Computing in Science and Engineering* 19(3):68-77.