High Performance System Modeling and Performance Evaluation for Grid Computing

Jong Sik Lee
School of Computer Science and Engineering
Inha University
jslee@inha.ac.kr

Abstract

This paper introduces a Quantization-based System Modeling (QSM) which performs reduction of data traffic requirements in grid computing. The QSM is based on a quantization of system state and provides a high performance grid computing. For realization of QSM in grid computing, this paper introduces how to perform data transmission reduction using quantization approach and suggests the more efficient concept and realization, e.g. quantization prediction. For a realization of the approach, we develop a state update transmission reduction component, e.g. quantization prediction component, which provides behavior and characteristic of a quantization prediction concept. A kinetics of spaceship is taken for a case study to evaluate performance. Empirical results, performing data traffic reduction and system execution time saving, apparently show system performance improvement through the quantization-based approach in grid computing.

Keywords:
Quantization-based System Modeling, High Performance Grid Computing, Quantization Prediction

1. Introduction

As large-scale system behavior and complexity analyzed by computer has been increasing, grid computing is demanded to deal with behavior and complexity of a large-scale system with reasonable computation and communication resources. For effective execution of grid computing, high-resolution and large-scale representations of system is needed to handle behavior of large-scale modern system. This paper presents a Quantization-based System Modeling (QSM) [1, 2, 3, 4, 5] of a complex and large-scale system to support high-resolution and large-scale representations. The QSM is based on a quantization of state of system and provides a high performance grid computing. For realization of QSM in grid computing, this paper introduces how to perform data transmission reduction using quantization approach and suggests the more efficient concept and realization which is called quantization prediction. To evaluate system performance of the quantization-based system modeling, a kinetics of a spaceship is taken an application and simulated. The kinetics maintains an accounting of where ships are and predicting their future destinations. In addition, we develop a workable system of the DTSS (Discrete Time System Specification) formalism in DEVS (Discrete Event System Specification) [6, 7, 8]. For performance evaluation, we model the kinetics on both of DTSS and DEVS formalisms and environments and compare system performance for each other. Section 2 introduces the QSM and shows how to reduce data transmission with it. Section 3 presents a quantization-based system component to realize a quantization prediction approach. Section 4 presents a QSM of a kinetics of spaceship as a case study. Section 5 discusses experiment and performance evaluation. Section 6 is conclusion.

2.Quantization and Data Traffic Reduction

The QSM is created by combining quantization theory [1, 2] and Discrete Event System (DES) modeling. This section specifies characteristics of quantization theory and DES and focuses on data traffic reduction to improve system performance of grid computing system. When we develop a system with a digital computer, a requirement issue is to approximate a continuous trajectory with a finite number of values in a finite time interval. One way to provide a finite time interval is to discretize a time base to obtain a Discrete Time System (DTS) approximation. The advanced way is to define a time base when any event is occurred and to obtain a time interval with a DES approximation. We introduce a mathematical formalism, which is the DEVS to represent a DES approximation in this paper.

Alternatively, rather than discretization of the time base, this paper partitions the trajectory into a finite number of segments, each of which has a finite computation associated with it. One way to do this is to quantize the value space. Quantization is a very general concept applicable to any value set, *X*. We can assign a partition *x* on *X* with a finite number of partition blocks (equivalence classes). Here, we notice a relationship between data transmission requirement among distributed components and quantization. Figure 1 illustrates a basic idea to reduce data transmission requirement by applying a quantization approach. Basic quantization separates a state variation with a range (e.g. quantum D) and specifies a finite number of states as Figure 1 (a) illustrates. As time proceeds, the value of state is changing. The approach applies when a sender component is updating a receiver component on a numerical, real-valued, state variable, which is a dynamically changing attribute. A state quantizer is applied to the sender's output, which checks for a threshold crossing whenever a change in the variable occurs. Only when such a crossing occurs, a new value of the variable is sent across the network to the receiver. The approach reduces the data sent and incurs some local computation at the sender. In addition, the reduction of data communication causes to decrease local computation at the receiver.

As Figure 1 (b) illustrates, a more efficient approach of quantization is quantization prediction, where the sender employs a model to predict the next boundary crossing and the time this crossing will occur. Since the next boundary crossing is either one above or one below the last recorded boundary, the sender does not need to send the full floating point (double word) value to the receiver, so that it sends a one-bit message at crossings. The one-bit message represents whether the next higher or next lower boundary has been reached. In the quantization prediction approach, the main advantage over basic quantization is that both the number of messages and their size can be reduced. A second advantage is that discrete event prediction can also greatly reduce the sender's state transition computation execution time and frequency if simple prediction models are used.

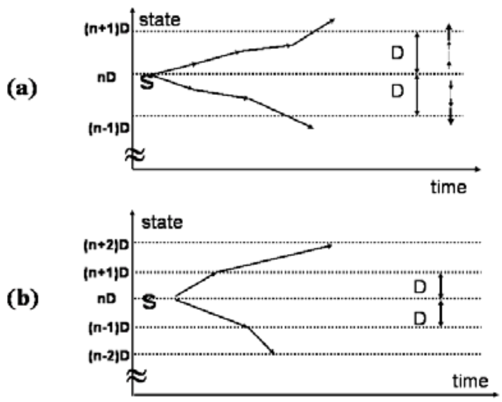

Figure 1 Data transmission reduction with quantization ((a) Basic Quantization, (b) Quantization Prediction)

3. Quantization Prediction Component

This paper suggests a quantization prediction integrator component which realizes a quantization prediction concept. We represent four functions of the DEVS formalism to construct an integrator with the quantization prediction. If there is an external event after an elapsed time, e, the external transition function will immediately update its state using its current input, and store a new input from the input quantizer.

$$\delta_{ext}((q,x,n),e,x') = (q+x*e,x', n)$$

Unless there is an external input, the internal transition function will update its state after a next output.

$$\delta_{int}(q, x, n) = (n + D*\mathrm{sign}(x), x, n+ \mathrm{sign}(x))$$

This function occurs when input and output come to the system at the same time. In first, the external transition function operates and the internal transition function provides the second operation.

$$\delta_{con}((q,x,n), x') = \delta_{int}((q,x,n),\mathrm{ta}(q,x,n),x')$$

The output at the next internal event is the quantized output of state at that time.

$$\lambda(q,x) = n + D*\mathrm{sign}(x)$$

The time advance function determines whether the state may be inside a block (hence not a multiple of *D*) or not. The external transition handles in case the state would be inside a block (hence not a multiple of *D*).

$$ta(q,x,n) = |((n+1)D - q)/x| \text{ if } x > 0 \text{ and } (n+1)D - q > 0$$
$$= |(q - nD)/x| \text{ if } x < 0 \text{ and if } |q-nD| > 0$$
$$= \infty \text{ otherwise (i.e., } x = 0)$$

While the internal transition handles in case the state would be on a boundary.

$$ta(q,x,n) = |D/x| \quad \text{if } x \,!\!= 0$$

When a new input is received, we update the state using the old input and the elapsed time in the external transition function. From this new state, q, the new time to reach either the upper or lower boundary is computed. Even if we start on a boundary, the state may eventually be inside a block (hence not a multiple of quantum size *D*) as a result of an external transition. If a state is on a boundary, the time advance computation merely divides D by the current input x which is a derivative in the internal transition function. If we reach the upper boundary (n+1) or lower boundary (n–1) we output and update the state accordingly. While the input remains with the same value, the time to cross a successive boundary will be the same.

Figure 2 compares a time trajectory of the DTSS integrator and the quantization prediction integrator and shows the difference between two integrators that are mentioned with representation of each formalism previously. The integrator of DTSS Simulation, after every step time, puts an output event and gets an input event basically. While the quantization prediction integrator puts an output event at the time point from the time advance function, ta(). Time point from the time advance function is when a state of system crosses a boundary of quantum based on a partition block of the state. That means that the crossing of the partition block boundary is implemented as state events. Time point from the time advance function depends on a quantum, a current input, and a current state. And this quantization prediction integrator gets an input event when an input event occurs in this integrator.

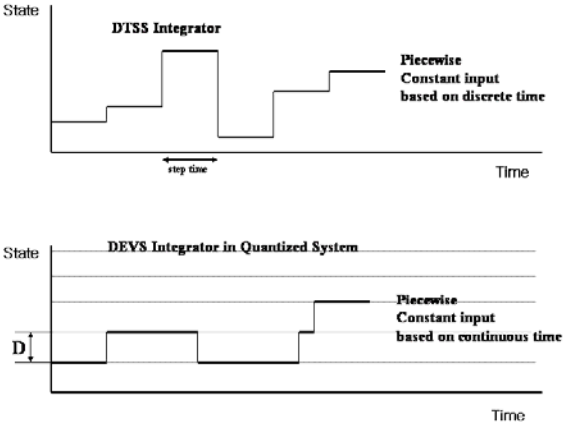

Figure 2 Time trajectory (DTSS integrator vs. Quantization Prediction integrator)

4. Case Study: Kinetics of spaceship

A kinetics of spaceship, especially circulation of spaceship with an ideal circle on the earth, is taken as a case study to evaluate performance of a quantization-based system in a grid computing. For spaceship to maintain an ideal circular orbit with radius D and speed v around a massive body, it is also required that a centripetal force, mv^2/d, which equals to the force of gravity. The force of gravity pulls along the line joining the two centers and has magnitude $F = GMm/d^2$, where G is the gravitational constant, and M and m are the masses. The distance of a ship with center at (x,y) to the center of gravity of a massive body (x_0,y_0) is $d = ((x - x_0)^2 + (y - y_0)^2)^{1/2}$. The force is projected in the x and y directions in proportions, $p_x = x/d$ and $p_y = y/d$, respectively. In an ideal orbit with d = D (constant), the coordinate dynamics separate into two independent 2nd order linear oscillators. We develop a spaceship model as a quantization-based system. From this model, we construct an abstraction that is to maintain the accounting of where ships are and to predict their future destinations. Thus, our overall modeling objectives are to construct a space travel scheduling and test it. The modeling is based on the differential equations which are based on Newtonian mechanics [11, 12]. Figure 3 gives us information about the movement of spaceship and about how to know where ships are and predict their future destinations with several parameters.

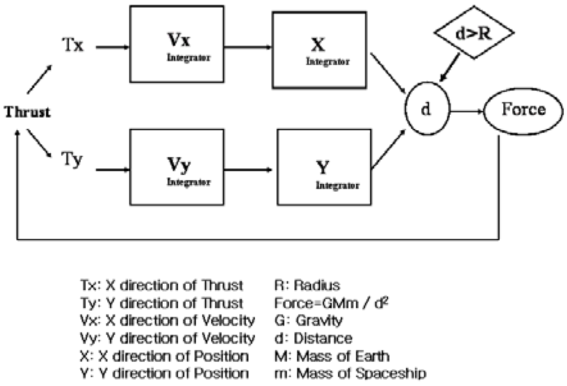

Figure 3 Modeling for kinetics of spaceship

5.Experiment and Performance Evaluation

A spaceship circulates around the earth and sends its position update information to a ground system on the earth whenever it moves to new space position. Actually, the space is represented with three dimensions. This paper abstracts the three dimensional space to two dimensional space to evaluate the quantization prediction approach. We develop a kinetics model of spaceship as shown in Figure 3. The kinetics model has a total of four integrators and is developed on the DEVSJAVA modeling and simulation environment [9, 10]. We develop three different systems: Non-Quantization, Basic Quantization and Quantization Prediction. Basic quantization system includes quantization DTSS integrators and quantization prediction system includes quantization prediction integrators. Figure 4 and Figure 5 compares system performance and scalability of the three systems with two performance measures: data transmission bits and system execution time. Both of basic quantization and quantization prediction reduces apparently data transmission bits and system execution time as the number of spaceships increases. The quantization prediction shows better system performance than that of basic quantization since the quantization prediction predicts a new quantized value, sends only one bit data, and reduces transmission data bits, effectively. Evaluation through two performance measures shows system performance improvement through execution cost reduction, and thus we can say the quantization approach can save communication and computation resources in grid computing.

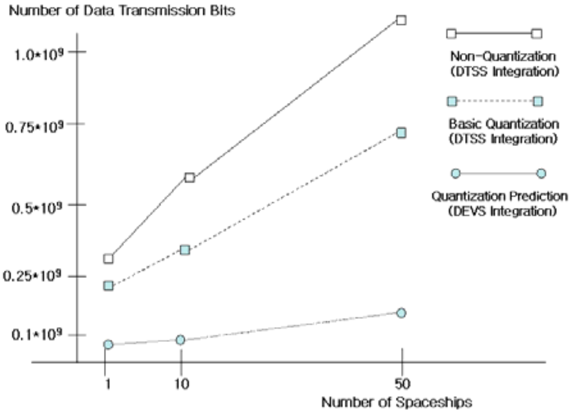

Figure 4 Number of Data Transmission Bits (D = 0.1) (Non-Quantization vs. Basic Quantization vs. Quantization Prediction)

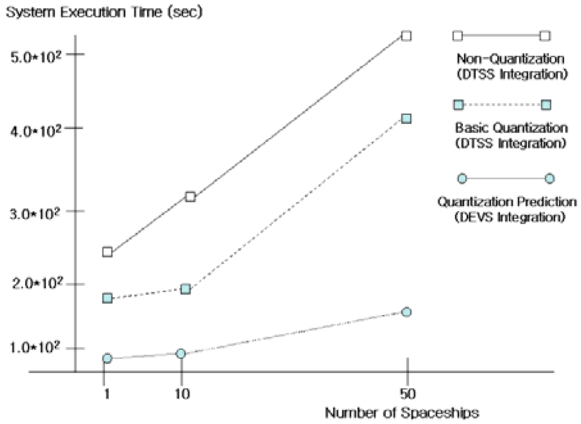

Figure 5 System Execution Time (D = 0.1) (Non-Quantization vs. Basic Quantization vs. Quantization Prediction)

The average of error measures system accuracy in Figure 6. The average error is increasing as quantum size, D, is increasing. Also, as D is increasing, transmission data bits and system execution time are decreasing in both of basic quantization and quantization prediction. There exists a tradeoff between execution cost reduction and error increment. We should control the quantum size, D, and reduce execution cost within a tolerable error. In comparison between basic quantization and quantization prediction, the quantization prediction apparently reduces data transmission bits and system execution time in the same error of the basic quantization.

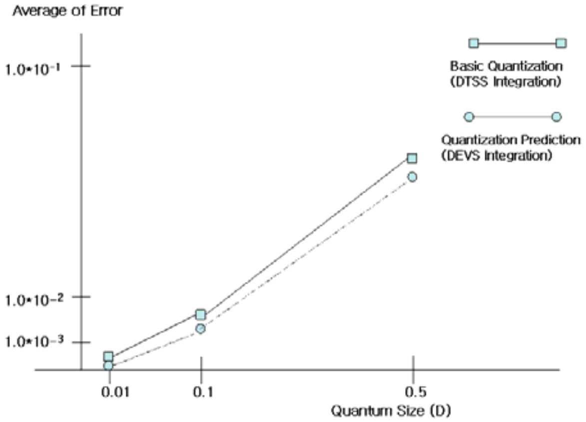

Figure 6 Average of Error
(Number of Spaceships = 10)
(Basic Quantization vs. Quantization Prediction)

6. Conclusion

This paper presents a quantization approach with the DTSS and DEVS representations. The quantization approach reduces an amount of transmission data bits between senders and receivers, thus reduces the total execution time in complex and large-scale grid systems. Especially, the quantization prediction apparently reduces data communication among components due to sending only one bit to represent increasing or decreasing of quantization. The quantization approach is able to provide high performance computing in grid computing systems by reducing data transmission requirement among distributed components. The execution of large-scale grid system is achieved with high performance computing in limited communication and computing resources. This paper suggests a quatization-based integrator to realize the quantization approach. We develop two types of integrators: quantization DTSS and quantization prediction. The quantization DTSS system is developed with the DTSS formalism by using the quantization DTSS integrators. The quantization prediction system is developed with the DEVS formalism by using the quantization prediction integrators. The empirical results from the quantization DTSS and quantization prediction systems showed performance improvement with a tradeoff from system accuracy. With this limitation, the quantization approach should be applied within a tolerable error. In further research, the quantization approach will be used to develop a real-time grid computing system and its execution infrastructure. A real-time grid computing system with real-time constraint requirements should require improvement of system performance and accuracy through the quantization approach.

Reference

[1] Zeigler, B.P. and J.S. Lee., Theory of Quantized Systems: Formal Basis for DEVS/HLA Distributed Simulation Environment. in Enabling Technology for Simulation Science(II), SPIE AeoroSense 98. 1998. Orlando, FL

[2] Zeigler, B.P., DEVS Theory of Quantization, . 1998, DARPA Contract N6133997K-0007: ECE Dept., UA, Tucson, AZ.

[3] Ernesto Kofman, Sergio Junco, Quantized-State Systems, a DEVS Approach for Continuous System Simulation, Transactions of SCS, 2001

[4] Bernard. P. Zeigler, H. Sarjoughian, and H. Praehofer, Theory of Quantized Systems: DEVS Simulation of Perceiving Agents, J. Sys. & Cyber, Vol. 16, No. 1, 2000.

[5] G. Wainer, and B.P. Zeigle,. Experimental Results of Timed Cell-DEVS Quantization, AI and Simulation, AIS 2000, Tucson, AZ.

[6] Zeigler, B.P., T.G. Kim, and H. Praehofer, Theory of Modeling and Simulation. 2 ed. 1998, New York, NY: Academic Press

[7] Zeigler, B.P., et al., The DEVS Environment for High-Performance Modeling and Simulation. IEEE C S & E, 1997. 4(3): p. 61-71.

[8] Zeigler, B.P., et al., DEVS Framework for Modelling, Simulation, Analysis, and Design of Hybrid Systems, in Hybrid II, Lecture Notes in CS, P. Antsaklis and A. Nerode, Editors. 1996, Springer-Verlag: Berlin. p. 529-551.

[9] Bernard. P. Zeigler and D. Kim. "Design of High Level Modelling / High Performance Simulation Environments," in 10th Workshop on Parallel and Distributed Simulation. 1996. Philadelphia.

[10] Zeigler, B.P., et al., "The DEVS Environment for High-Performance Modeling and Simulation," IEEE C S & E, 1997. 4(3): p. 61-71.

[11] Roger R. Bate, Donald D. Mueller, Jerry E. White, Fundamentals of Astrodynamics, Dover Publications, New York, 1971

[12] Erwin Kreyszig, Advanced Engineering Mathematics: Seventh Edition, John Wiley& Sons Inc, New York, 1993.