Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

Towards Large-Scale What-If Traffic Simulation with Exact-Differential Simulation

Masatoshi Hanai

Tokyo Institute of Technology
2-12-1 Oo-okayama
Meguro Tokyo, JAPAN

Toyotaro Suzumura

IBM T.J. Watson Research Center
/ University College Dublin / JST
1101 Route 134 Kitchawan Road
Yorktown Heights, NY 10598, U.S.A.

Georgios Theodoropoulos

Durham University
South Road
Durham, DH1 3LE, U.K.

Kalyan S. Perumalla

Oak Ridge National Laboratory
One Bethel Valley Road
Oak Ridge, TN 37831-6085, U.S.A.

ABSTRACT

To analyze and predict a behavior of large-scale traffics with what-if simulation, it needs to repeat many times with various patterns of what-if scenarios. In this paper, we propose new techniques to efficiently repeat what-if simulation tasks with exact-differential simulation. The paper consists of two main efforts: what-if scenario filtering and exact-differential cloning. The what-if scenario filtering enables to pick up meaningful what-if scenarios and reduces the number of what-if scenarios, which directly decreases total execution time of repeating. The exact-differential cloning enables to execute exact-differential simulation tasks in parallel to improve its total execution time. In our preliminary evaluation in Tokyo bay area's traffic simulation, we show potential of our proposals by estimating how the what-if scenarios filtering reduces the number of meaningless scenarios and also by estimating a performance improvement from our previous works with the exact-differential cloning.

1 Introduction

Large-scale microscopic traffic simulation has been a beneficial way to research on areas such as prediction of traffic congestion, planning of urban developments, citizen behavior in emergencies. Unlike other statistical and mathematical ways, such approaches can give detail analysis results of the individual vehicles and other entities like junctions and roads because it actually emulates the vehicles' behavior and interactions with each other.

To analyze by such microscopic simulation, it needs to repeat execution many times with different scenarios and parameters, called what-if simulation. For example in simulation of Tokyo traffic (Osogami, Imamichi, Mizuta, Morimura, Raymond, Suzumura, Takahashi, and Ide 2012, Osogami, Imamichi, Mizuta, Suzumura, and Ide 2013, Suzumura and Kanezashi 2013, Hanai, Suzumura, Ventresque, and Shudo 2014), which has 770K junctions, we need to repeat simulation tasks 770K times when we pick up one junction from the 770K junctions and simulate what happens if the junction is blocked. When we simulate multiple blocks of the junctions, we need to execute 2^{770K} times (the sum of combination from 770K junctions choosing 0 to 770K). Also, it often needs to execute a lot of times for parameter tuning (e.g. road speed limit, a time interval of signals) to imitate a realistic situation. In our previous works, *exact-differential simulation* (Hanai, Suzumura, Theodoropoulos, and Perumalla 2015), we achieved to improve the execution time of such repeating simulation by reducing redundancy of events processing.

However, there is still a big problem in the number of repeating times. The number of repeating times for what-if simulation increases as the simulation scale becomes large. For example, when we simulate bigger city than Tokyo (the number of junctions is 770K as mentioned above), the number of repeating times becomes over 770K when we simulate what happens if one of the junctions is blocked.

In this paper, we will propose new techniques to efficiently repeat such a large number of simulation tasks for reducing total analyzing time. There are two main ideas in our proposal: what-if scenario filtering and exact-differential cloning.

- *What-if Scenario Filtering*: This is a technique to reduce the number of scenarios itself by filtering meaningless scenarios, which does not impact results of simulation.
- *Exact-Differential Cloning*: This is a technique to run exact-differential simulating tasks in parallel. We show our new exact-differential simulator which stores events and states logs in a distributed file system instead of a local storage and which enables to execute exact-differential simulating tasks in parallel.

The rest of paper is organized as follows. In Section 2, we describe the context of our research. In Section 3, we give an overview of our proposal. In Section 4, we discuss the way to filter what-if scenarios. Also we show its preliminary and estimated evaluation. In Section 5, we discuss the implementation of our system extended by previous simulator and a way to run simulation tasks in parallel with some preliminary and estimated evaluation. Finally we conclude this paper in Section 6.

2 Background: Exact-Differential Traffic Simulation

In this section, we describe the exact-differential simulation, which improves execution time of repeating simulation by reducing redundancy of event processing, and the traffic simulation modeling in our simulator.

The exact-differential simulation flow basically consists of two parts: an initial whole execution and repeating executions. In the initial whole execution (Figure 1), the simulator stores all processed events and intermediate state. After the initial whole execution, the simulator starts repeating executions (Figure 2) from a changing point and reprocesses only affected events using the stored events and states (Figure 3).

In the initial whole execution (Figure 1), the events are processed in the almost same way as the optimistic PDES, where unlike the normal optimistic PDES, processed events, cancel events and stored states are never released by the global synchronization (or *GVT, global virtual time*). Instead, such events, cancel events and states are stored to storage to reuse in repeating execution.

In repeating exact-differential execution (Figure 2), the system first inputs a what-if query including a changing LP and time before it distributes to the changing LP. The query is received in the LP before accessing a local storage to load changing events and all events affected by the changing events. These events (the changing events and the affected events) are inserted to an event queue to process again. After that, the LP passes the earliest unprocessed event to application before it gets a new generated event and a new changed state from the application. The new generated event is sent to a destination LP and the new state is stored to a state queue. After receiving new events, the new events' received time is checked and affected events are loaded from a local storage.

The model of the traffic simulation is based on IBM Megaffic (Osogami, Imamichi, Mizuta, Morimura, Raymond, Suzumura, Takahashi, and Ide 2012, Osogami, Imamichi, Mizuta, Suzumura, and Ide 2013, Suzumura, Kato, Imamichi, Takeuchi, Kanezashi, Ide, and Onodera 2012) and SCATTER (Perumalla 2006, Yoginath and Perumalla 2008), where traffic systems are represented as discrete event systems. The individual vehicles' behavior is based on Megaffic, where a vehicle's track of junctions from origin to destination is all fixed before execution by estimation with some defined behavior model, for example, shortest path or minimum hops of junctions. After that, in the execution, the vehicle's speed, traveling time to next junction and selection of a lane are calculated based on some defined behavior models. The

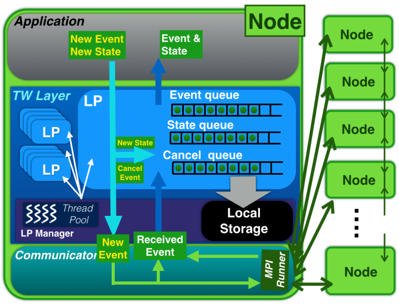

Figure 1: Initial Execution.

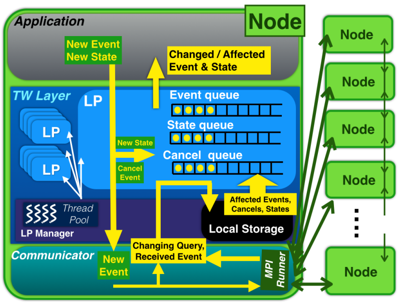

Figure 2: Repeating Execution.

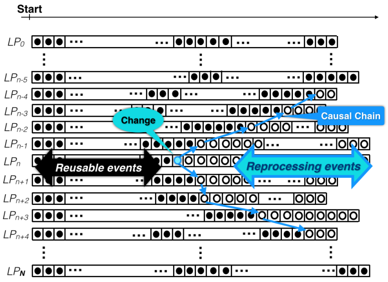

Figure 3: Reprocessing Events.

global interaction of vehicles around the road map is based on SCATTER, where synchronization of LPs is controlled by optimistic PDES.

3 System Overview

In this section, we describe an overview of our proposal, showed in Figure 4. The key ideas of our proposal are to reduce as much what-if scenarios as possible in advance and then to execute what-if simulation tasks in parallel. There are 3 steps to efficiently repeat what-if simulation: what-if scenarios filtering, exact-differential cloning and actual simulation executing with exact-differential simulation. In the what-if scenarios filtering, the system pre-analyzes what-if scenarios and picks up some ”meaningful” what-if scenarios filtered by some defined condition (e.g. the number of affected events, departing time of the vehicles) with some mechanisms (e.g. machine learning, statistical way) after initial execution, where all processed event and state logs are stored in a database. We discuss more details on the filtering in Section 4. In the exact-differential cloning, a cloning controller schedules what-if simulation tasks based on the already filtered what-if scenarios and on the number of available computer resources. After that, each task clones events and states in parallel from a distributed file system which can be accessed by every machines in clusters, in the same way as loading events and states from a local storage in the previous work’s implementation discussed in Section 2. We discuss more detail in Section 5. After that, the exact-differential simulation tasks start to execute.

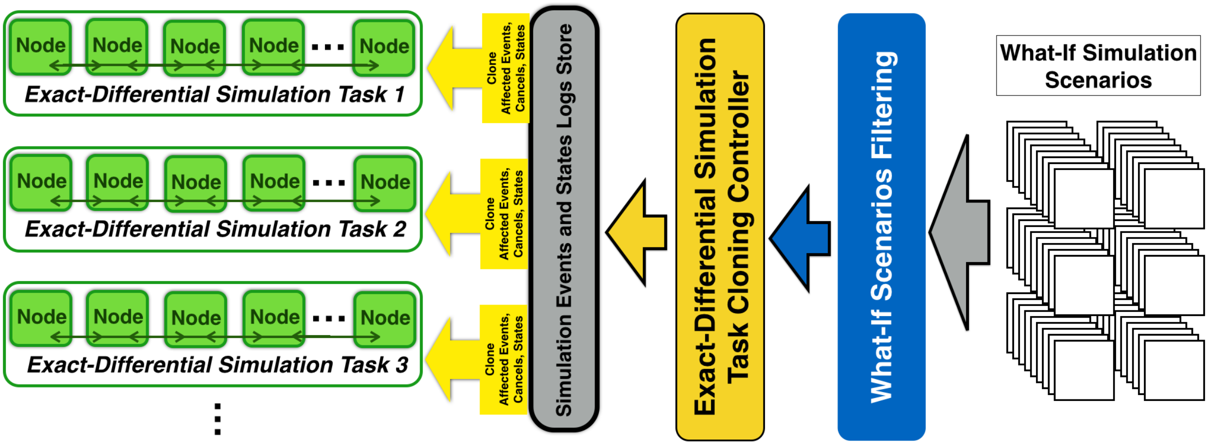
Node
Node
Node
Node
Node
Exact-Differential Simulation Task 1
Exact-Differential Simulation Task 2
Exact-Differential Simulation Task 3
Clone Affected Events, Cancels, States
Clone Affected Events, Cancels, States
Clone Affected Events, Cancels, States
Simulation Events and States Logs Store
Exact-Differential Simulation Task Cloning Controller
What-If Scenarios Filtering
What-If Simulation Scenarios

Figure 4: Overview of Large-Scale What-If Simulation with Exact-Differential Simulation.

4 What-If Scenarios Filtering

In this section, we clarify objectives and potential on a what-if scenarios filtering and show its overview. What-if scenario filtering is a technique to reduce the number of what-if scenarios in advance of what-if simulation.

4.1 Objectives

Our aim of the what-if scenarios filtering is to pick up only ”meaningful” scenarios for what-if simulation. For example in changing roads’ speed limit, if the changing is very little and this changing hardly influences the simulation result, then this what-if scenario is ”meaningless”. On the other hand, if the changing is very large and this changing strongly influences the simulation result, then this scenario is ”meaningful”. As we discussed, the number of what-if patterns becomes large as the size of simulation. For example in Tokyo traffic simulation, there are 770K junctions and when we simulate what happens if a accident happens in each road, we need to execute simulation 770K times. Thus, it is necessary to reduce the number of scenarios as much as possible, which directly reduces total execution time of repeating simulation tasks.

Figure 5 shows a flow of what-if scenarios filtering. With input data (vehicles’ origin/destination data and road map data) and historical simulation results, the system filters meaningless scenarios and picks up the meaningful scenario by some filtering technique (e.g. machine learning, statistical way). Such mechanism is highly depends on simulation models and is generally difficult to achieve but it should be feasible in traffic simulation because traffic input data (vehicle data and road map data) has relatively rich information in advance, including departing time and a track of all paths (origin/destination and all junctions to reach) and it enables to pre-analyze what-if scenarios closely. Also, we can use the simulation result in initial execution for filtering.

Our main goal is to develop the traffic specific filtering, which has never been proposed in large-scale traffic simulation domain. Actually, such filtering mechanisms have been already proposed and achieved good results in other domain of simulation researches (Brueckner and Van Dyke Parunak 2003, Cabrera, Luque, Taboada, Epelde, and Ma Iglesias 2012, Calvez and Hutzler 2005). For example in agent-based simulation of emergency departments in hospitals (Cabrera, Luque, Taboada, Epelde, and Ma Iglesias 2012), the all patterns of scenarios are filtered by random sampling and clustering algorithm before executing what-if simulation tasks for getting the optimum scenario. In (Calvez and Hutzler 2005), they use a machine learning method based on GA (genetic algorithm) to pick up the appropriate scenarios to be simulated in general agent-based simulation.

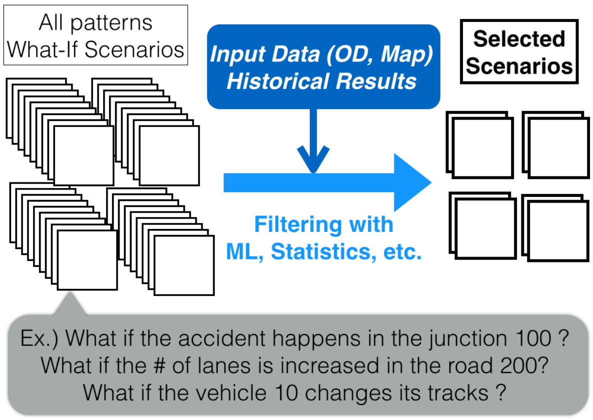
All patterns
What-If Scenarios
Input Data (OD, Map)
Historical Results
Selected
Scenarios
Filtering with
ML, Statistics, etc.
Ex.) What if the accident happens in the junction 100 ?
What if the # of lanes is increased in the road 200?
What if the vehicle 10 changes its tracks ?

Figure 5: What-If Scenarios Filtering.

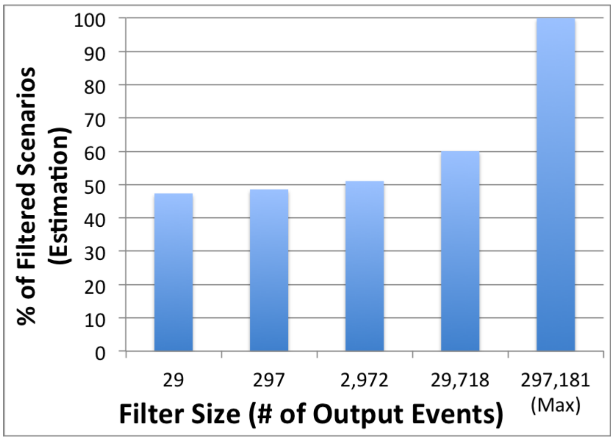

Figure 6: Estimated % of Filtered Scenarios.

4.2 Preliminary Evaluation

In this part, we describe potential of the filtering by estimating evaluation with our previous work's result (Hanai, Suzumura, Theodoropoulos, and Perumalla 2015). We evaluate how many what-if scenarios are estimated to be filtered in the previous work's situation showed in Figure 7, which represents relation between LP ID (junction ID) and the number of output events by changing the LP's parameter (junction and road parameter).

Figure 6 shows the relation between the percentage of filtered scenarios and a filter size, which implies how strict the filtering condition is and is defined as follows.

> What-if scenarios predicted to generate smaller number of output events than *the filter size* are filtered and ignored to be processed.

In the evaluation, we assume that we can filter what-if scenarios by the filter size with some prediction technique. For example, we should be able to roughly predict the number of events affected by the what-if scenarios with pre-analyzing the result of initial execution, which includes logs of all vehicles tracks, and we can get causal chains of all events (sequences of affected events).

As you can see in Figure 6, the percentage of filtered scenarios keeps high (over 45%) even though the filter size is very small (1/10000 of the maximum). The reason why such filtering is highly effective in the traffic simulation, especially in the scenarios (picking up one junction and changing its parameter), is that in most junctions their changes hardly affect the results because very few vehicles go through the junction. Such what-if scenarios (picking up such uncrowded junction) is filtered even though the filtering size is very small. On the other hand, there are very few "hub" junctions, which a lot of vehicles go through and whose changing strongly affects the simulation result.

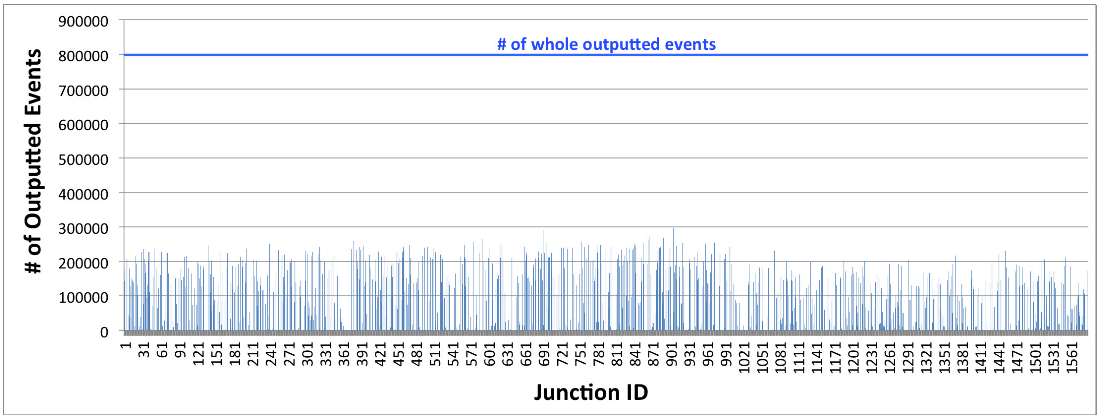

Figure 7: # of Differential Outputted Events with Junction&Road Change in Our Previous Work (Hanai, Suzumura, Theodoropoulos, and Perumalla 2015).

5 Exact-Differential Cloning for Parallel Task Execution

In this section, we describe a exact-differential cloning technique to execute exact-differential simulation tasks in parallel. In the exact-differential simulation, repeating execution is independent of each other and it is basically possible to run these tasks in parallel. Thus in this section, we describe especially on how we will implement a system extended from our previous exact-differential simulator.

5.1 Implementation

We will implement the system by reconstruct the local storage in Figure 1 and 2 to be able to access from other machines. To meet this requirement, we will use a distributed file system (e.g. Lustre, GPFS), which can be access from every machine in a cluster. Figure 8 and Figure 9 show a mechanism of the exact-differential cloning. In initial execution, the simulator stores all historical events and states logs to the database on a distributed file system instead of local storage, which can be globally accessed from every machine in a cluster (Figure 8). In repeating execution, the simulator clones affected events and states in parallel from the database on a distributed file system (Figure 9).

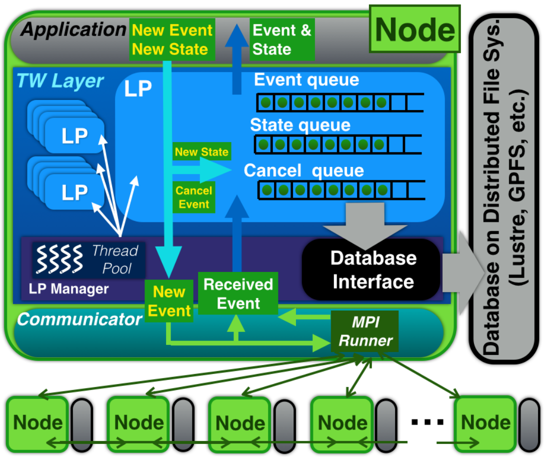

Figure 8: Store logs in Initial Execution.

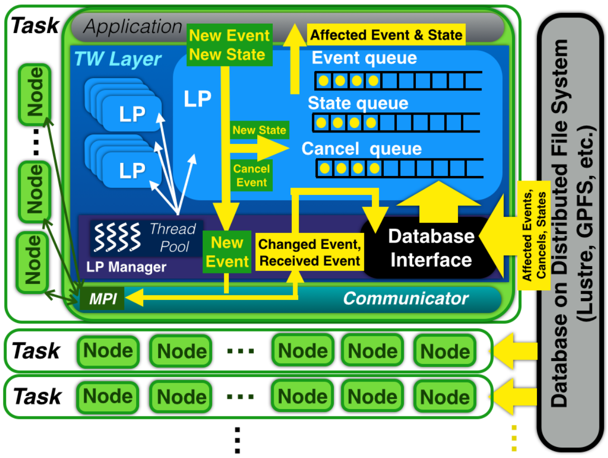

Figure 9: Clone Affected Events and States in Repeating Execution.

5.2 Preliminary Evaluation

In this part, we show preliminary and estimated evaluation. According to our previous work's results (Hanai, Suzumura, Theodoropoulos, and Perumalla 2015), we estimate how the task parallel execution with the exact-differential cloning can improve the performance. First, we briefly show our previous work's result (Figure 10). Second, we state assumptions for estimation and finally, we illustrate estimated elapsed time of 161K times repeating execution, comparing 3 cases (whole simulation, previous sequential repeating, our proposing parallel repeating) in Figure 11.

Figure 10 shows strong scaling of Tokyo bay area's traffic simulation (161,364 junctions and 203,363 roads for 3 hours) with what-if scenarios of junctions' condition changing, comparing whole simulation (798K events), exact-differential simulation in the worst case (297K events), and exact-differential simulation in the average case (61K). We evaluated our simulator with 192 cores (12 cores × 16 nodes) in parallel on TSUBAME 2.5 supercomputer in Tokyo Institute of Technology.

There are 3 assumptions for this estimation to simplify the following discussion.

- Event processing time is enough bigger than event cloning time from distributed file systems and an overhead to clone event can be ignored even though there is approximately over 10 times overhead compared to in-memory reading.
- We can use the same number of cores and machines (192 cores, 16 nodes) in the 3 cases.
- Elapsed time of each exact-differential execution is always same as the average case (thus the total elapsed time is equal to actual total time).

Figure 11 shows the estimated elapsed time of 161,364 times repeating execution, where the 161,364 means the number of junctions in the Tokyo road map. Thus, we consider what-if scenarios where all junctions' conditions are changed respectively. We evaluate in 3 ways to run the 161K times simulation tasks.

First in whole simulation, which is the naivest way, we estimate the elapsed time to simply repeat whole simulation 161K times with 192 cores. It takes 6788 hours (= 151 seconds × 161,364 times / 3600) according to the Figure 10. Seconds in sequential repeating with exact-differential simulation, which is our previous work, we estimate the elapsed time by calculating,

$$(Elapsed\ time\ in\ average\ case\ with\ 192\ cores) \times 161,364\ times,$$

on the assumption that elapsed time of one time repeating is always same as the average case. It takes 2117 hours in this case according to our previous works. Finally in task parallel repeating with exact-differential cloning, which is our proposal, we estimate the elapsed time by calculating

$$(Elapsed\ time\ of\ average\ with\ \text{''}most\ efficient\text{''}\ \#\ of\ cores) \times 161,364\ times \div \frac{192\ cores}{\text{''}most\ efficient\ \#\text{''}},$$

where we fix the "most efficient" number of cores as "48" based on the strong scaling, because the performance on average case is saturated at 48 cores and it does not improve from 96 cores as Figure 10. It takes 675 hours in our proposed task parallel way, which is 31.9 % of the previous sequential way and is only 9.94 % of the naive repeating way.

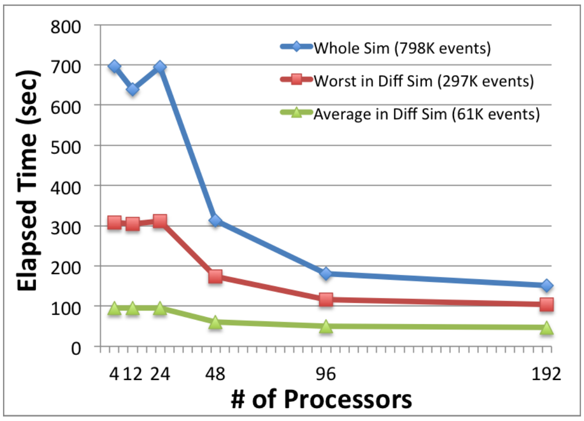

Figure 10: Strong Scale in Our Previous Work (Hanai, et al. 2015).

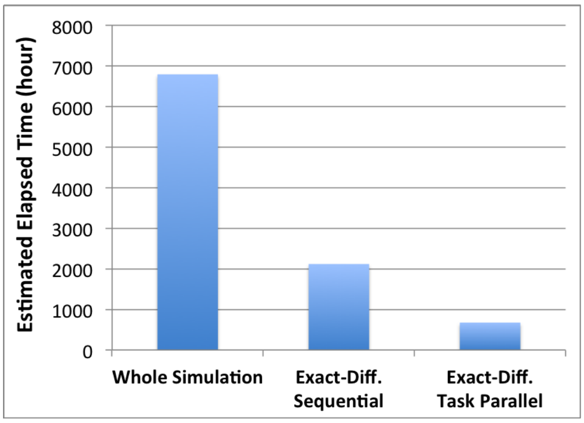

Figure 11: Estimated Elapsed Time of 161K Times Repeating Execution with 192 cores.

The main reason of such improvement is that in the exact-differential simulation, the number of processing events is sometimes (e.g. average case) too small to be scaled well. This task parallel technique solves such imbalance between data size and the number of cores and it achieves efficient usage of the computer resources.

6 Conclusion

In this paper, we proposed the technique for large-scale what-if simulation of traffic systems with the exact-differential simulation, including the what-if scenarios filtering and the exact-differential cloning. In what-if scenarios filtering, we clarified the objectives to reduce the number of what-if scenarios and its preliminary estimated evaluation showed such filtering technique should have big impacts in large-scale traffic simulation. For example of the what-if simulation changing LP (junctions & roads) in the Tokyo traffic simulation, the evaluation showed that the filtering reduces over 45 % even though the filtering condition is very week (1/10000 of maximum). In exact-differential cloning, we showed the way to extend our simulator and its preliminary estimated evaluation showed the elapsed time with our proposed task parallel way is 31.9 % of the previous sequential way and is only 9.94 % of the naive whole repeating way.

For future work, we should implement the actual system and evaluate it. Also we should propose the actual algorithms or techniques to filter the what-if scenarios and evaluate it.

REFERENCES

Brueckner, S., and H. Van Dyke Parunak. 2003. “Resource-aware exploration of the emergent dynamics of simulated systems”. In *Proceedings of the 2nd international joint conference on Autonomous agents and multiagent systems (AAMAS’03)*, 781–788.

Cabrera, E., E. Luque, M. Taboada, F. Epelde, and L. Ma Iglesias. 2012. “ABMS optimization for emergency departments”. In *Proceedings of the 2012 Winter Simulation Conference (WSC’12)*, 1–12.

Calvez, B., and G. Hutzler. 2005. “Automatic tuning of agent-based models using genetic algorithms”. In *Proceedings of the 6th international conference on Multi-Agent-Based Simulation(MABS’05)*, 41–57.

Hanai, M., T. Suzumura, G. Theodoropoulos, and K. Perumalla. 2015. “Exact-Differential Large-Scale Traffic Simulation”. In *Proceedings of the 2015 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS’15)*.

Hanai, M., T. Suzumura, A. Ventresque, and K. Shudo. 2014. “An Adaptive VM Provisioning Method for Large-Scale Agent-Based Traffic Simulations on the Cloud”. In *Proceedings of IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom’14)*, 130–137.

Osogami, T., T. Imamichi, H. Mizuta, T. Morimura, R. Raymond, T. Suzumura, R. Takahashi, and T. Ide. 2012. “IBM Mega Traffic Simulator”. Technical report, Technical Report RT0896, IBM Research–Tokyo.

Osogami, T., T. Imamichi, H. Mizuta, T. Suzumura, and T. Ide. 2013. “Toward simulating entire cities with behavioral models of traffic”. *IBM Journal of Research and Development* 57 (5): 6:1–6:10.

Perumalla, K. 2006. “A Systems Approach to Scalable Transportation Network Modeling”. In *Proceedings of the 2006 Winter Simulation Conference (WSC’06)*, 1500–1507.

Suzumura, T., and H. Kanezashi. 2013. “Accelerating Large-Scale Distributed Traffic Simulation with Adaptive Synchronization Method”. In *Proceedings of the 20th ITS World Congress*.

Suzumura, T., S. Kato, T. Imamichi, M. Takeuchi, H. Kanezashi, T. Ide, and T. Onodera. 2012. “X10-based massive parallel large-scale traffic flow simulation”. In *Proceedings of the 2012 ACM SIGPLAN X10 Workshop (X10’12)*, 3:1–3:4.

Yoginath, S., and K. Perumalla. 2008. “Parallel Vehicular Traffic Simulation using Reverse Computation-based Optimistic Execution”. In *Proceedings of the 22nd ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation (PADS’08)*, 33–42.

AUTHOR BIOGRAPHIES

MASATOSHI HANAI is a Ph.D. candidate at Tokyo Institute of Technology. He holds M.S. degree in computer science also from Tokyo Institute of Technology. His research interest is parallel computing and distributed systems, especially parallel discrete event simulation (PDES) and large-scale traffic simulation. His email address is mhanai@acm.org.

TOYOTARO SUZUMURA, Ph.D., is a research staff member at IBM T.J. Watson Research Center in the U.S. as well as a visiting associate professor of University College Dublin in Ireland. He received his Ph.D. in Computer Science from Tokyo Institute of Technology in 2004. He joined IBM Research - Tokyo in 2004 and had involved with several projects such as high performance XML processing, the PHP scripting language, large-scale stream computing, the X10 programming language and so forth. Since 2010, he has started to develop an X10-based agent simulation platform and its application to large-scale traffic simulation. He had served as a visiting associate professor at the Graduate School of Information Science and Engineering of Tokyo Institute of Technology since April 2009 until October, 2013 and explores research on high performance computing and large-scale graph analytics. In October 2013, he joined the Smarter Cities Technology Center of IBM Research located in Dublin, Ireland and in April 2015 he has joined IBM T.J. Watson Research Center in the U.S as a research staff member. His e-mail address is suzumura@acm.org.

GEORGIOS THEODOROPOULOS, Ph.D., is a Professor and the Executive Director of the Institute of Advanced Research Computing at Durham University in the UK. His research interests are in the areas of Modeling and Distributed Simulation, Complex and Multi-agent systems; Data intensive computing and Info-Symbiotic Systems; High Performance, Parallel and Distributed Computing, and computational infrastructures for data driven analytics of of complex socio-technical systems. His e-mail address is georgios.theodoropoulos@durham.ac.uk.

KALYAN S. PERUMALLA, Ph.D., is a Distinguished Research Staff Member and Manager in Oak Ridge National Laboratory, USA where he is also the Group Leader of the Discrete Computing Systems Group in the Computational Sciences and Engineering Division. He serves as an Adjunct Professor in the School of Computational Sciences and Engineering at the Georgia Institute of Technology. His e-mail address is perumallaks@ornl.gov.