Large scale parallel execution of urban scale traffic simulation and its performance on K computer*

Daigo Umemoto[1] and Nobuyasu Ito[1,2]

[1]RIKEN Center for Computational Sciences
7-1-26 Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan
Tel.: +81-78-940-5868
daigo.umemoto@riken.jp
[2]Department of Applied Physics, Graduate School of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Abstract. We attempt to perform many case simulations of urban scale traffic by performing massive parallel computing using K computer and CARAVAN job manager. We obtain 1025 variations of simulation results with the same condition and different random seeds within 13 hours. Each of simulation runs took about six hours, which is twice the longer than the case of using conventional workstations or clusters, while our approach allows further massive parallel computation. The performance and limitations when using K computer are discussed.

Keywords: Traffic flow · High performance computing · Multi agent social simulation

1 Introduction

In this new era of so-called big-data and machine-learning, a hope is emerging that we can establish a unified and quantitative sociology in these decades. However, as results of big-data analyses in the past decade, we are already beginning to realize that the big-data is not *sufficiently big* since their size is too limited to exhaustively cover the huge parameter space that social phenomena should have, and particularly, often their coverage is biased and makes it difficult to analyze in our interests.

Simulation approach shall be valid to overcome this defect. As pointed out in [1], Multiagent Social Simulation (MASS) is now attracting attention as a powerful methodology of research on social science. Together with the high performance computing, exhaustive data construction is possible: we can virtually conduct many-case social experiment and deal with them on statistical basis, which has been difficult to conduct in the real world.

* Supported by MEXT as "Exploratory Challenges on Post-K computer(Studies of multi-level spatiotemporal simulation of socioeconomic phenomena).".

For quantative and rigorous simulation research, the elementary behavior of each agent obey is desired to be rigidly and quantitatively established. Here, the traffic model on straight single lane is well understood in the context of MASS. With the expectation that the results of traffic simulation as a MASS is going to contain some features that closely resembles the actual traffic flow as pointed out in [2], we attempt to construct a simulation system that provides many-case results using K computer.

2 Our traffic simulation system using K computer

To construct urban scale traffic simulation system, we use agent based and open source simulator SUMO (Simulation of Urban MObitlity, [3]). We also introduce available digitized map of Kobe city in Japan into the system, which was edited as mentioned in [4] from a digital road map produced by ZENRIN CO., LTD.. More detailed simulation conditions are mentioned in [5]. Each simulation with this setup takes about two to three hours of real time, using Intel(R) Xeon(R) CPU E5-1620 @ 3.60GHz. The size of the result files are about 2 GB which includes all the intermediate file, to enable trial and error offline analysis after the simulations.

To make use of high performance computing hence execute exhaustive many-case simulations explained above in parallel, an efficient job manager is required. We use open source job manager framework CARAVAN [7] created by our team in purpose of large scale parameter-space exploration which requires using high performance computer including K computer. It allows to simultaneously execute the simulation run instances with different parameters and random seeds each other.

In our setup, Each instances takes single process and are isolated hence there is no inter-process communication which sometimes becomes a bottle neck of computation speed. CARAVAN also allows to sequentially execute one simulation run after another within a given wall time, while we do not use this feature at this stage and finish the entire calculation when the single simulation run instance with longest computation time is completed.

We implemented exactly the same simulation conditions explained above and re-compiled SUMO source code so that it can be entirely executed on K computer. The schematic diagram of entire system and data flow is drawn in Figure . It requires stage-in procedure in the beginning of the calculation, and stage-out procedure in the end of the calculation to fetch the result, because the global file system of computation node of K computer is provided only for computation, not for data storage purpose. All files existing in the computation node during the computation will be entirely deleted when the submitted job finishes. Log-in node is provided for storage purpose instead.

In our setup, input file and executives have the size of about 250 MB in total and result files have the size of about 2 GB on each instance. Soon after the stage-in the computation starts, if there is enough space in computational resource is available in K computer. Otherwize, we have to wait until assigned

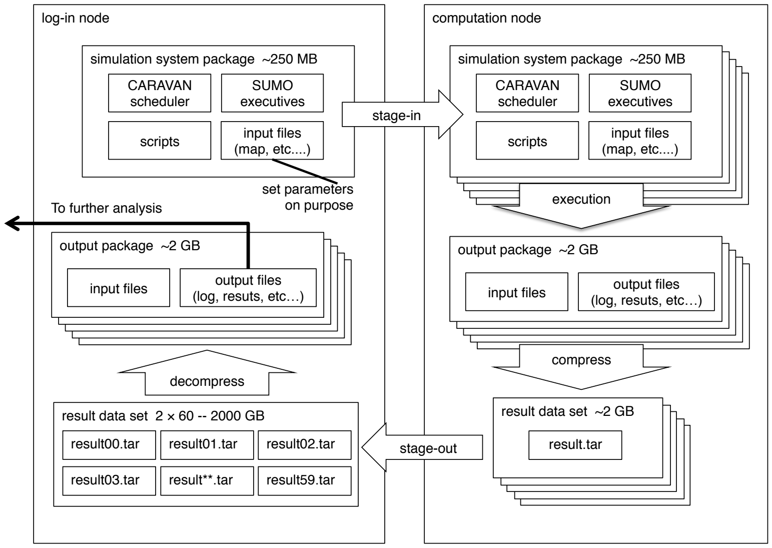

Fig. 1. Entire system of exhaustive simulation system using CARAVAN and K computer

calcutation start time. This happens when we demand large number of node usage, therefore it is convenient to demand smallest node number available many times and gather up them later on. The smallest node number is 12.

We assign single CPU on each simulation run instance. Each node has eight CPUs, and they share the same file system hence the disk capacity is also shared. The maximum can be set fo 29 GB by adding following line in the submitted job script:

```
#PJM --rsc-list "node-quota=29 GB"
```

otherwize the default maximum limitation 14 GB is used. When calculation finishes, each process owns files with size of about 4.5 GB in total finally, consisting of 250 MB of compressed simulation system package, 250 MB of decompressed package, 2 GB of raw result, and 2 GB of compressed result to be staged-out in the following process. We chose to use five CPUs in each node to keep total file size under the 29 GB of limitation, as well as based on the test result to attempt using six CPUs which caused disk full failure. The number of CPUs shall be increased by suppressing the intermediate files, when our analysis becomes sophisticated.

3 Performance result of large-scale simulation using K computer

Job execution time result provided by K computer is shown in Table 1. There is a tendency that larger demand of number of runs require longer waiting time till the beginning of the execution. When we attempted to submit jobs containing more than 1200 runs, the waiting time exceeded more than two days. However, smaller jobs shown in 1 have affordable waiting time. K computer allows users to submit 15 jobs at a time. We made a strategy to use this feature to submit smallest jobs whose run number is 58 as much as possible, and within 13 hours, obtained 1052 simulation results with different random seeds and the same simulation parameter. With different parameters, the execution time may vary. Mixing up simulations with different parameter may cause the white region to be larger, therefore we suggest to submit simulations with the same parameter and different seeds at once.

As shown in the table, entire computation time which includes stage-in and out takes about 6 hours. This is not affected by number of simulation run instances or even shorter with larger number of instances because all processes are independently and parallelly executed. Stage-in procedure takes about 30 seconds and Stage-out procedure takes about seven minutes which is negligible to the entire processing time. Therefore, the longest computation time of single simulation run instance is about 6 hours, which is almost twice the longer than using the conventional CPU. This is because K computer consists of Fujitsu SPARC64 VIIIfx CPU architecture which have almost half clock frequency compared to the conventional CPU architecture, which ensures the stable operation. However, the number of parallel instances are massively larger than conventional

Table 1. Execution time

job ID	number of runs	submission time	wait	execution	finish time
7497111	118	2018/09/02 02:38:57	1:58:06	6:00:31	10:37:34
7497114	238	2018/09/02 02:41:53	8:10:25	5:53:12	16:45:30
7497122	58	2018/09/02 02:52:57	0:02:36	6:05:26	09:00:59
7497123	58	2018/09/02 02:53:48	0:03:08	6:11:06	09:08:02
7497146	58	2018/09/02 05:19:48	0:02:51	5:58:01	11:20:40
7497147	58	2018/09/02 05:19:51	0:02:48	5:57:56	11:20:35
7497148	58	2018/09/02 05:19:53	0:02:46	6:00:37	11:23:16
7497149	58	2018/09/02 05:24:09	0:26:21	6:05:08	11:55:38
7497150	58	2018/09/02 05:24:11	0:04:55	6:01:25	11:30:31
7497151	58	2018/09/02 05:24:12	3:07:13	5:49:59	14:21:24
7497152	58	2018/09/02 05:24:13	3:07:12	6:02:00	14:33:25
7497153	58	2018/09/02 05:24:14	3:07:12	5:52:41	14:24:07
7497154	58	2018/09/02 05:24:22	1:51:52	5:53:15	13:09:29
7497156	58	2018/09/02 05:24:24	3:53:05	5:59:08	15:16:37

workstations or clusters and it is affordable to attempt further larger parallel execution. For example, a cluster with 100 cores requires about 600 hours (about ten days) of computation to finish the same amount of task.

Job management result of job ID 7497156 provided by CARAVAN is shown in Figure 2. The vertical axis is time in seconds since the beginning of each simulation run and the horizontal axis is process ID. There is an fluctuation in execution times which is within 20,000 seconds, which is about 10% of the entire time of runs. The white region represents the time space that CPUs are allocated but unused, which is undesireble. We will making use of these to optimize the node usage in some way in future research.

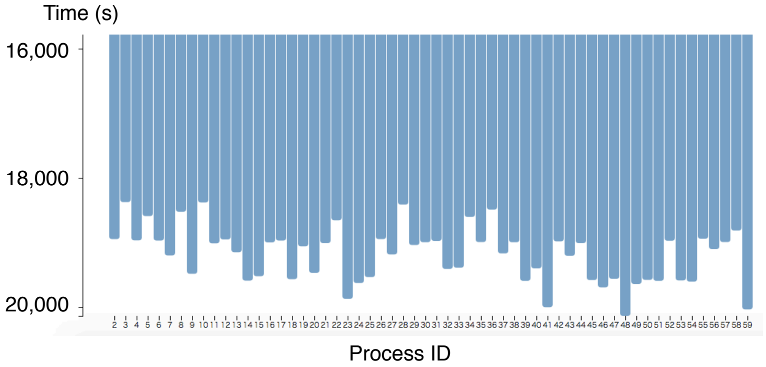

Fig. 2. Job management result

References

1. Noda, I., Ito, N., Izumi, K., Mizuta, H., Kamada, T., & Hattori, H. (2018). Roadmap and research issues of multiagent social simulation using high-performance computing. *Journal of Computational Social Science*, 1(1), 155-166.
2. Uchitane, T., & Ito, N. (2016). Applying Factor Analysis to Describe Urban Scale Vehicle Traffic Simulation Results. *Transactions of the Society of Instrument and Control Engineers*, 52, 545-554.
3. Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent development and applications of SUMO-Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4).
4. Asano, Y., Ito, N., Inaoka, H., Imai, T., & Uchitane, T. (2015). Traffic Simulation of Kobe-City. *In Proceedings of the International Conference on Social Modeling and Simulation, plus Econophysics Colloquium 2014* (pp. 255-264). Springer, Cham.
5. Umemoto, D., & Ito, N. (2018). Power-law distribution in an urban traffic flow simulation. *Journal of Computational Social Science*, 1(2), 493-500.
6. Murase, Y., Uchitane, T., & Ito, N. (2014). A tool for parameter-space explorations. *Physics Procedia*, 57, 73-76.
7. Murase, Y. https://github.com/crest-cassia/caravan
8. Murase, Y. https://github.com/crest-cassia/caravan-viz