A high performance approach with MATSim for traffic road simulation

Sara Moukir, Nahid Emad, Stéphane Baudelocq

A high performance approach with MATSim for traffic road simulation

Sara Moukir
sara.moukir@uvsq.fr
sara.moukir@eiffage.com

Nahid Emad
nahid.emad@uvsq.fr

Stéphane Baudelocq
stephane.baudelocq@eiffage.com

***Abstract*—Road traffic simulation becomes more and more accurate over time. From macroscopic simulations based on fluid equations, for example, to microscopic simulations based on the multi-agent paradigm, innovations have continued to emerge in recent years. Multi-agent models such as MATSim [1], POLARIS [2] or SimMobility [3] have been highlighted lately in response to complex and microscopic simulation requirements. However, this microscopic and large-scale approach requires much higher computing power than that delivered by a home computer. High-performance computing approach can be a relevant response to this kind of problem. MATsim due to its module structure was selected as the basis of our study. We introduce a new concept for the design of parallel algorithm of MATSim. The parallel architectures used as support for the experiments presented are provided by GRID'5000 (or G5K) [4].**

***Index Terms*—road traffic simulation, big data analysis, high performance computing, complex and heterogeneous dynamic system**

I. Introduction

The study of road traffic is similar to the study of a complex and heterogeneous dynamic system. Actually, the individual will of each agent combined with the diversity of the means of transport make the simulation difficult. The microscopic and erratic nature of this phenomenon requires a more complex paradigm than that of a simple mathematical model. Multi-agent simulation turns out to be a relevant paradigm to express this phenomenon effectively. Indeed, a multi-agent simulation offers a dynamic environment and a behavior specific to each agent framed by a certain number of rules. It makes it possible to obtain a co-evolutionary simulation that best imitates the real phenomenon of road traffic. Agent-based road traffic simulators such as MATSim, SimMobility or POLARIS have been developed and used over the past few years to simulate the transport systems of several different cities. They have the ability to accurately replicate movement patterns. Given their disaggregated representation of travel demand and supply as well as rich spatial and temporal detail, these agent-based transport models have the potential to enable unprecedented detailed analysis.

MATSim provides a framework to implement large-scale agent-based transport simulations [5]. The framework consists of several modules which can be combined or used stand-alone. Its modularity, extensibility and performance caught our attention. These modules support are as follows : demand modeling, agent-based mobility simulation, replanning, and output analysis methods [5]. However, the computational requirements are increasingly important. For example, from our experiments MATSim takes about 9 minutes on average to run an iteration for moving about 187,000 agents in its Los Angeles scenario (MATSim Los Angeles 1%) [6]. As a result, high performance computing approach turns out to be relevant to have reasonable execution times. Multi-threading is already used in some sections of MATSim [7], but the use of high performance computing field is still exploitable [8]. In this paper, we propose a parallel algorithm design approach for MATSim as well as its implementation. The parallel architectures used as support for the experiments presented are provided by GRID'5000 (or G5K). Grid'5000 is a French national and international cluster constituted by multiple large homogeneous sub-clusters situated through France mainly.

The rest of the article is organized as follows: section 2 describes the challenges of the parallelization carried out, section 3 describes in a more optimal way the operation of the traffic simulator MATsim, and section 4 describes our algorithm proposal.

II. Related work

Several works have been carried out to date to try to effectively improve the execution time of multi-agent road traffic simulators thanks to parallel / distributed computing [8].

Different strategies have been deployed. One of these strategies consists in dividing the road network into several sub-spaces so as to assign each calculation unit a sub-network [9]. This strategy raises other issues, the main one being that of inter-process communication, which dominates in terms of execution time. Among the microscopic road traffic simulators using these strategies, TRANSIMS [10], AIMSUN [10] and Paramics [10] are the most popular parallel implementations, although they are not based on the multi-agent paradigm.

Regarding agent-based simulators, the distribution of agents between the different computing units is part of the strategies used to speed up the execution [11]. The same problem of inter-process communication remains since for each time step, all the servers must know the state of the whole network. Indeed, the advancement of agents on the network depends on the movements of other agents, managed by other hosts; hence the continuous communication between servers to inform of the local dynamics of each of them.

Other strategies have been applied to MATSim in particular, such as the implementation of a specific library for paralleliza-

978-1-6654-9755-8/22/$31.00 ©2022 IEEE
DOI 10.1109/IIAIAAI55812.2022.00140

tion [12] or the modification of the operation of certain modules. The replanning module, for example, has been revised so that this step is executed according to other parameters, so as to have faster convergence [13]. The simulation module has also been modified, so that there is an "event-driven" approach instead of the "time-step based" approach (QSIM vs. JDEQSIM and HERMES)[14, 15]. Also, we present a new algorithmic concept for the replanning module for the design of new plans from an initial plan; based on the use of parallelism. Our goal is to obtain a faster convergence and therefore a reduced execution time.

III. MATSim Simulator

Figure 1 [14] illustrates how MATSim works, which basically consists of three sections: execution of the mobsim, scoring, and replanning [14]. MATSim aims to optimize each agent's initial daily plan by iteratively running the three sections in sequence. The main outputs of the iterative simulation includes the optimal daily plans (the "best tested plans"), traffic flow and infrastructure occupancy. In summary, here is how the three modules fit together:

- **Execution module (mobsim) :** executes the daily plans of each agent in the population. This module simulates the way agents carry out their daily activities and move from one place of activity to another ;
- **Scoring module :** uses a utility function (taking into account both activity and movement) to evaluate the daily plans based on the performance of each agent in the execution module ;
- **Replanning module :** adjusts the plan elements (e.g. departure time or mode of transportation) based on plan scores, to adapt plans to traffic flow.

As iterations progress, MATSim duplicates the initial plan and tries to improve it (with different strategies such as changing transport mode or changing departure time) [16], and keeping the different nb_plans duplicated plans (nb_plans being a parameterizable variable). Once the number of iteration $iter\_max$ is reached, the plan with the best score will be kept for each agent. Roughly, these stages and particularly that of the replanning are renewed at each iteration sequentially so as to optimize the plans, until obtaining a convergence of the average of the scores of the best plans of all the agents.

Figure 2 shows the convergence of the average optimal score of all agents over the iterations.

Algorithm 1 MATSim algorithm

1: **Start.** Choose $iter\_max$, nb_plans...
2: **Iterate.** For i =1, ... , $iter\_max$
Run one iteration of MATSim : mobsim, scoring and replanning for each agent. The replanning module duplicates one plan for each agent independantly (if $i < nb\_plans$) or modifies the existing ones. At each iteration, all the agents are processed (their daily plan over 24 hours is executed).

IV. Algorithm proposal

We propose an approach which aims to reduce the convergence time so as to have a shorter execution time. For this, we exploit the potential offered by parallel computing. The idea is to start by running an instance of MATSim sequentially, with a number of iterations that corresponds to the number of plans desired for each agent. There will therefore be nb_plans iterations for the first sequential execution of MATSim. We consider this first phase as an initialization. Subsequently, after obtaining nb_plans per agent, nb_plans processes will each execute an instance of MATSim in parallel, taking as input one of the nb_plans generated during the first sequential execution thanks to a mapping.

From there, we initialize the value of nb_plans to 1, so that each agent cannot generate other plans, and continues to improve the only one he has, the one he has inherited from the initialization phase.

We set an iteration step $iter\_step$. At each of these iteration steps, the nb_plans processes send the score and the associated unique plan of each agent to the other processes, and will also receive those generated by the other processes. It will only save them if their own score is lower than the plans received. All of this is illustrated in pseudocode in Algorithm 2.

Algorithm 2 Proposed parallel algorithm for MATSim

1: **Start.** Choose $iter\_max$, nb_plans, $iter\_step$ ...
2: **Initialization.** For i =1, ... , nb_plans
Run one iteration of MATSim (mobsim, scoring and replanning) to produce one plan per agent.
Mapping. Map each one of the nb_plans to a process dedicated to run a new instance of the nb_plans of MATSim. From there, nb_plans = 1 for each one of the process.
3: **Iterate.** For i = 1, ..., $iter\_max$, **do in parallel**
Run one iteration of MATSim and modify the plan.
if i mod $iter\_step$ == 0 **then**
Send all the other processes the best-scoring plans of all the agents
Receive the best-scoring plans from other processes for all agents and saves only the best for each agent if it is higher than its own.

V. Experiment and performance

The experiments are in progress. We will present the complete experiments in the last version of this paper.

VI. Conclusion

So far and with our analyses, we see in MATsim a still highly exploitable potential of the high performance computing approach. We are also working on the possibility of adding

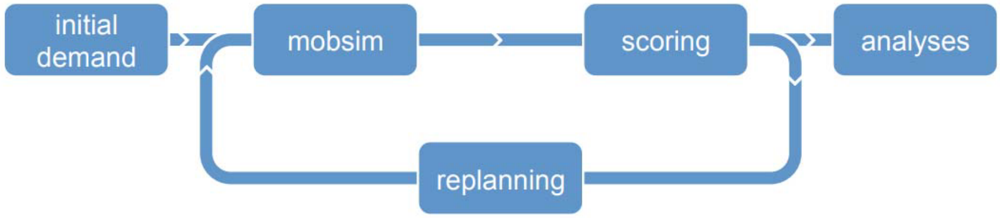

Fig. 1. Iterative MATSim loop [14]

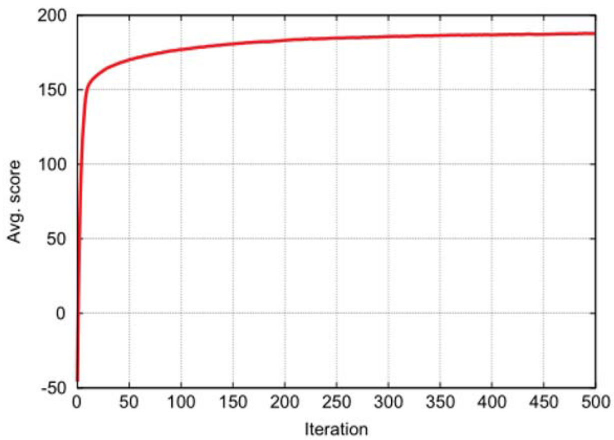

Fig. 2. Typical score progress [14]

an empirical convergence criterion for MATSim based on our experimental results.

REFERENCES

[1] Horni, A., K. Nagel and K. W. Axhausen (2016) The multi-agent transport simulation MATSim, Ubiquity Press, London.

[2] Auld, J., M. Hope, H. Ley, V. Sokolov, B. Xu and K. Zhang (2016) POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations, Transportation Research Part C: Emerging Technologies, 64, 101-116. 101–116.

[3] Adnan, M., F. C. Pereira, C. M. L. Azevedo, K. Basak, M. Lovric, S. Raveau, Y. Zhu, J. Ferreira, C. Zegras and M. Ben-Akiva (2016) SimMobility: A multi-scale integrated agent-based simulation platform, paper presented at the 95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record, Washington, D.C., January 2016.

[4] Grid'5000: A large scale and highly reconfigurable experimental grid testbed, http://www.grid5000.fr

[5] Rieser, M, Horni, A and Nagel, K. 2016. Let's Get Started. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) The Multi-Agent Transport Simulation MATSim, Pp. 9–22. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.2. License: CC-BY 4.

[6] Rieser, M, Horni, A and Nagel, K. 2016. Scenarios Overview. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) The Multi-Agent Transport Simulation MATSim, Pp. 367–368. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.52. License: CC-BY 4.0

[7] Horni, A and Nagel, K. 2016. More About Conguring MATSim. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) The Multi-Agent Transport Simulation MATSim, Pp. 35–44. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.4. License: CC-BY 4.0

[8] Nagel, K. 2016. Other Experiences with Computational Performance Improvements. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) The Multi-Agent Transport Simulation MATSim, Pp. 267–268. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.40. License: CC-BY 4

[9] Tomas Potuzak, "Reduction of Inter-process communication in Distributed Simulation of Road Traffic", 2020

[10] Johannes Nguyen, Simon T. Powers, Neil Urquhart, Thomas Farrenkopf, Michael Guckert "An overview of agent-based traffic simulators", 2021

[11] Matthieu Mastio, Mahdi Zargayouna, Omer Rana, Gérard Scemama "Méthodes de distribution pour les simulations de mobilité des voyageurs", 2015

[12] Zhiyuan Ma, Munehiro Fukuda, "A multiagent spatial simulation library for parallelizing transport simulation", 2015

[13] Chengxiang Zhuge, Mike Bithell, Chunfu Shao, Xia Li, Jian Go "An improvement in MATSim computing time for large-scale travel behaviour microsimulation", 2019

[14] Horni, A, Nagel, K and Axhausen, K W. 2016. Introducing MATSim. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) The Multi-Agent Transport Simulation MATSim, Pp. 3–8. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.1. License: CC-BY 4.

[15] Dan Graur, Rodrigo Bruno, Joschka Bischoff, Marcel Rieser, Wolfgang Scherr, Torsten Hoefler, Gustavo Alonso "Hermes: Enabling efficient large-scale simulation in MATSim", 2021

[16] Horni, A and Nagel, K. 2016. More About Conguring MATSim. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) The Multi-Agent Transport Simulation MATSim, Pp. 35–44. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.4. License: CC-BY 4.0