PanSim: A Performance-Portable Agent Based Model

ISTVÁN Z. REGULY, BENCE KEÖMLEY-HORVÁTH, GÁBOR SZEDERKÉNYI, and ATTILA CSIKÁSZ-NAGY*, National Laboratory for Health Security„ Pázmány Péter Catholic University, Faculty of Information Technology and Bionics, Hungary

Additional Key Words and Phrases: PanSim, Agent-based model, portability, CPU, GPU

ACM Reference Format:
István Z. Reguly, Bence Keömley-Horváth, Gábor Szederkényi, and Attila Csikász-Nagy. 2018. PanSim: A Performance-Portable Agent Based Model. In *SC23: Supercomputing 23, November 12–17, 2023, Denver, CO.* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 EXTENDED ABSTRACT

The COVID-19 pandemic presented a clear and immediate threat to global health, instigating a surge of research into various modeling and forecasting techniques to understand and combat the virus. Among the plethora of modeling approaches, the agent-based model (ABM) emerged as a significant tool. Within an ABM, virtual individuals are simulated moving through virtual environments, modeling the transmission and spread of disease. Though ABMs have a rich historical background, the unprecedented demands from modeling the complexities of the pandemic truly stretched the boundaries of data assimilation, model scaling, and performance: there was an acute need to provide accurate forecasts within highly constrained time frames. Our group has evaluated existing ABM tools such as Repast [1] and MASON [2], but found their performance to be an inhibiting factor for simulations of this scale, therefore decided to develop a new GPU-enabled model, PanSim.

PanSim is a specialized ABM constructed at Pazmany Peter Catholic University University, designed to support Hungary's data-driven response to the COVID-19 pandemic. Spanning from the fall of 2020 until early 2022, the development and use of PanSim involved a close collaboration with policy-makers, with frequent weekly reports and rapid iterations based on feedback. The model was configured to analyze a broad array of interventions, encompassing lockdowns, quarantine protocols, testing strategies, vaccination guidelines, school closures, and more. The myriad combinations, often with subtle and non-trivial effects, had to be explored on a regular basis. PanSim's design was specifically tailored to capture the precise effects of these varied interventions [3, 4].

To ensure optimal performance and portability across diverse computing platforms, PanSim was implemented using C++ with the thrust template library. The focus of this work is on the critical data structure and algorithm that define the overall performance of the model: given a list of agents, their respective locations, and other relevant properties, along with a corresponding list of locations and their attributes, one has to calculate the cumulative values of various quantities (such as infectiousness) of agents at each location. Two distinct computational formulations are presented: (1) an agent-centric loop, in which the desired quantity is computed individually for each agent and then added to an

Manuscript submitted to ACM

accumulator specific to their location - a process that can be performed either sequentially or in parallel using atomic instructions (not universally supported across hardware); (2) a location-centric loop, where, given a sorted list of agents per location, the quantities are summed in parallel across different locations using the *reduce by key* parallel primitive, albeit at the expense of additional sort operation.

The simulation can be broken into distinct phases: a one-off initialization step, then every 10-minute timesteps consist of (1) movement, where agents decide which location to move to (2), update of data structures after movement, consisting of a reduce by location step and optionally sorting, and (3) and infection step, which contains a count agent sub-step and a reduce by location sub-step as well.

This study offers a comparative analysis of the time required to execute both formulations on CPUs and GPUs, presenting the distribution of total simulation time across these steps. Subsequently, comprehensive benchmarks are conducted across a selection of cutting-edge CPU and GPU architectures, such as NVIDIA V100 and A100 GPUs, AMD MI100 GPUs, Intel Xeon (Cascade Lake) processors, AMD EPYC (Genoa-X) processors, and Ampere Altra processors.

A key concern, particularly when employing CPUs, is the parallel efficiency of these algorithms. Consequently, an evaluation of thread scaling ranging from 1 thread (utilizing formulation (1)) to the full size of the NUMA domain (employing formulation (2)) is undertaken, revealing a tendency towards suboptimal efficiency, primarily attributed to the lack of scaling of the sorting step.

While in certain workloads, executing a single simulation as quickly as possible is paramount (such as parameter fitting), for more relevant workloads such as parameter exploration or intervention design, there's a need to run hundreds of simulations to assess various scenarios, with each scenario requiring numerous runs for averaging. In these contexts, overall throughput takes precedence. This study assesses the time consumed to execute a total of 704 simulations on each platform, contrasting CPUs (by running one simulation per core) and GPUs (by leveraging Multi-Process Service or Multi-Instance GPUs). The findings highlight the superior performance of the A100 (2.85 hours) and the Genoa-X (1.45 hours) platforms.

Yet, the time it takes to do these runs is merely one side of the coin; the financial cost is the other. In a bid to quantify this, the on-demand and spot pricing of these platforms within Azure Cloud were analyzed to determine the monetary expense (in USD) associated with executing the 704 simulations. The results present a contrasting landscape from the throughput analysis, with the A100 and Genoa-X costs being relatively comparable for both on-demand (9.7 and 10.4 USD) and spot (3.6 and 4.2 USD) scenarios. Interestingly, the substantial discounts offered on Ampere Altra spot instances render it the most cost-effective platform, with rates of 15.5 USD on-demand and a mere 1.5 USD for spot.

ACKNOWLEDGMENTS

REFERENCES

[1] Nicholson Collier and Michael North. 2013. Parallel agent-based simulation with Repast for High Performance Computing. *SIMULATION* 89, 10 (2013), 1215–1235. https://doi.org/10.1177/0037549712462620 arXiv:https://doi.org/10.1177/0037549712462620

[2] Jill Bigley Dunham. 2005. An agent-based spatially explicit epidemiological model in MASON. *Journal of Artificial Societies and Social Simulation* 9, 1 (2005).

[3] Bence Keömley-Horváth, Gergely Horváth, Péter Polcz, Bálint Siklósi, Kálmán Tornai, János Juhász, Gábor Szederkényi, György Cserey, Attila Csikász-Nagy, and István Z Reguly. 2022. The design and utilisation of PanSim, a portable pandemic simulator. In *2022 First Combined International Workshop on Interactive Urgent Supercomputing (CIW-IUS)*. IEEE, 1–9.

[4] István Z. Reguly, Dávid Csercsik, János Juhász, Kálmán Tornai, Zsófia Bujtár, Gergely Horváth, Bence Keömley-Horváth, Tamás Kós, György Cserey, Kristóf Iván, Sándor Pongor, Gábor Szederkényi, Gergely Röst, and Attila Csikász-Nagy. 2022. Microsimulation based quantitative analysis of COVID-19 management strategies. *PLOS Computational Biology* 18, 1 (01 2022), 1–14. https://doi.org/10.1371/journal.pcbi.1009693