Exascale Agent-Based Modelling for Policy Evaluation in Real-Time (ExAMPLER)

Alison Heppenstall[1]
School of Social and Political Sciences, University of Glasgow, UK

J. Gary Polhill
The James Hutton Institute, Aberdeen, UK

Mike Batty
Bartlett Centre for Advanced Spatial Analysis, University College London, UK

Matt Hare
The James Hutton Institute, Aberdeen, UK

Doug Salt
The James Hutton Institute, Aberdeen, UK

Richard Milton
Bartlett Centre for Advanced Spatial Analysis, University College London, UK

Abstract

Exascale computing can potentially revolutionise the way in which we design and build agent-based models (ABM) through, for example, enabling scaling up, as well as robust calibration and validation. At present, there is no exascale computing operating with ABM (that we are aware of), but pockets of work using High Performance Computing (HPC). While exascale computing is expected to become more widely available towards the latter half of this decade, the ABM community is largely unaware of the requirements for exascale computing for agent-based modelling to support policy evaluation. This project will engage with the ABM community to understand what computing resources are currently used, what we need (both in terms of hardware and software) and to set out a roadmap by which to make it happen.

2012 ACM Subject Classification Computing methodologies → Modeling and simulation

Keywords and phrases Exascale computing, Agent-Based Modelling, Policy evaluation

Digital Object Identifier 10.4230/LIPIcs.GIScience.2023.38

Category Short Paper

Funding Supported by the Engineering and Physical Sciences Research Council(project reference EP/Y008839/1) and the Scottish Government Rural and Environment Science and Analytical Services Division (project reference JHI-C5-1). AH was supported by grants from UKPRP (MR/S037578/2), Medical Research Council (MC_UU_00022/5) and Scottish Government Chief Scientist Office (SPHSU20).

1 The transformative potential of exascale computing for agent-based modelling

Exascale computing is defined as computing capable of 10^{18} floating-point operations per second (FLOPS). Though instructions executed per second is of greater relevance to agent-based modelling than purely floating-point operations, the two are approximately the same. A recent experiment with an agent-based model conducted by the authors used 76 CPU

[1] Corresponding author

12th International Conference on Geographic Information Science (GIScience 2023).
Editors: Roger Beecham, Jed A. Long, Dianna Smith, Qunshan Zhao, and Sarah Wise; Article No. 38; pp. 38:1–38:5
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

days of computing time on a cluster with mean 4390 "bogomips" ("bogus" (i.e. approximate) million instructions per second) CPUs. This is approximately 3×10^{16} CPU instructions, which an exascale computer could theoretically complete in three hundredths of a second, in comparison with roughly a working day (8 hours or so) on a 200 CPU high-performance computing cluster. This is a six orders of magnitude improvement in computing time.

The potential benefits to agent-based modellers of access to exascale computing are immediate, even based on existing practice. These would be manifested most trivially in being able to sample models' high-dimensional parameter spaces more densely during calibration. Since exascale computers are massively parallel architectures, there is also immediate potential in appropriately parallelised larger-scale simulations enabling us to model megacities and countries with millions or billions of agents; though here there are challenges in taking full advantage of exascale computing because of thread-blocking and shared memory issues.

However, this huge gain in computing power has rather more revolutionary potential for agent-based modelling than merely doing what we already do, but bigger. There are three main activities occupying a significant time in empirical agent-based modelling. First is assembling and preparing data; second is designing and building the agent-based model itself and any supporting software; third is running the simulation experiments for calibration, validation and scenario analysis and processing and visualizing the outputs. The third of these – assuming the tools already exist – is most trivially addressed by exascale computing: a process that takes days can be completed in a few seconds. The first two, which are less embarrassing in duration when experimentation takes so long, then start to look rather more embarrassing.

Squazzoni et al.'s [6] call for improvements in data sharing and modelling practice in the early stages of the COVID crisis are no less relevant now than they were then. Though COVID brought some of the benefits of agent-based modelling into sharp focus as authors such as Thompson et al. [7] and Badham et al. [3] worked with policymakers to evaluate scenarios for managing the crisis. With an *existing* agent-based model, data, and analysis and visualisation tools, exascale computing could already support a creative, transdisciplinary discussion about how to handle a developing emergency. Such a discussion would be greatly enhanced, however, if the model could be adapted and new data brought in, as people became aware of potential cascading consequences of their interventions. With appropriate software and institutional support, enhancements like these could be realised, significantly improving the attractiveness of bringing agent-based modelling in to such conversations.

2 Challenges

Existing work with high-performance computing (HPC) infrastructure in agent-based modelling makes it clear that realising the potential of exascale computing in the area will not be without its challenges. These all largely pertain to accessibility. There are three main areas to consider: technical, institutional, and cultural. Much of the following is anecdotal, but the points will be familiar to those who have tried to access HPC to run their agent-based models.

From a technical perspective, Alessa et al. [1] put out their "All Hands" call to create a community of practice around social simulation and cyberinfrastructure in 2006, referring explicitly to the fact that developing agent-based models, even on platforms such as NetLogo, entails a learning curve that is a significant barrier to adoption of agent-based modelling in the social sciences. Introducing a special issue of *JASSS* on "grand challenges" more than

Please see notes in the Service Specification document regarding the maximum amounts of time that can be applied for and technical specifications.

	Largest Job	Typical Job	Smallest Job
Number of nodes	[Please Complete Table]		
Number of cores/GPUs used per node			
Wallclock time for each job*			
Number of jobs of this type			
Memory per node required.			

*The maximum permitted wallclock time per job is a function of local Service centre policy.

Figure 1 A screenshot from one of the EPSRC's "Technical Assessment" forms to access national computing infrastructure.

ten years later, An et al. [2] are still in a position to refer to the "steep learning curves" (para. 3.2 – note the plural) faced by modelling novices. Accessing HPC currently involves command-line interfaces, shell scripts, and SSH (Secure Shell Protocol) arcanery that social scientists are not desperately keen to learn. This does not mean that they do not want to use the technology. Their primary interest in doing so, however, is in the practical benefits to them in the insights gained for their case study. Social scientists don't necessarily get their intellectual "kicks" from playing with advanced technology. This means HPC needs to be easy to use – ideally (though impossibly) to such an extent that there is a button on the interface of their modelling tool that says "Run this experiment on HPC".

Institutionally, accessing HPC infrastructure is surrounded by gatekeepers, who need forms to be filled, stipulating information that social scientists may not be in a position to provide. For example, in the UK, national high-end computing infrastructure access is managed by the Engineering and Physical Sciences Research Council. This is managed by calls with deadlines[2], which require applicants to complete "Technical Assessment" forms stipulating information such as that in figure 1. From the perspective of managing the HPC, this kind of information makes sense in that it helps with planning usage of the machines to ensure that everyone's needs can be met. Further, individuals running the facilities can be extremely helpful to the first-time user in advising on how to complete these forms. However, Polhill [5] has pointed out that agent-based modellers may not be able to provide accurate estimates of run-time or memory demand from running the models, for sound theoretical computer science reasons *that anyone with a degree in computer science should know.*

The cultural side is somewhat harder to articulate, and could be ironically phrased as the question of whether your code is "worthy" of the very expensive computing equipment on which you hope to run it. The social sciences suffer perennially from physics envy, but physicists regularly use HPC as part of work on particle collision and cosmology, some even claiming to be in search of the "mind of God" [4, p. 175]. How can the social sciences compete with such majesty? Rather more prosaically, computer scientists are not especially excited about "embarrassingly parallel" problems such as running models repeatedly to explore parameter space. Massive distributed models with difficult multi-threading memory and

[2] e.g. https://www.ukri.org/councils/epsrc/facilities-and-resources/using-epsrc-facilities-and-resources/apply-for-access-to-high-performance-computing-facilities/

CPU co-ordination problems that don't break the benefits of parallelism are more interesting to them. This is strange because they have built machines that are brilliant and hugely efficient for the former kind of problem, but aren't really designed to do the latter nearly so well. Perhaps most problematically (and mundanely), however understandable it may be, the worthiness of your code is also reflected in its efficiency – have you used the right data structures and algorithms to reach the results of running the code with as few instructions executed as possible? Some might simply be pleased that they've got a model that runs without crashing...

3 Benefits beyond exascale

In thinking about the software, institutional and data support that empirical agent-based modellers need to take full advantage of the potential of exascale computing, there are opportunities to think about wider benefits to the community. By being involved in the conversation about HPC access, and making the case for our requirements, we may be able to break down some of the barriers described above, reaching a point whereby HPC use is more routine in agent-based modelling work. The software developed to support exascale agent-based modelling could also be useful for agent-based modelling on a laptop – especially if we are somehow able provide tools that enable agent-based models to be built rapidly.[3] If there are ways of easing access to data suitable for using in empirical agent-based models, and learning from and building on others' experiences with doing so, then this will advance empirical applications of agent-based models by reducing the time investment this modelling step currently requires.

References

1 Lilian Na Alessa, Melinda Laituri, and Michael Barton. An "all hands" call to the social science community: Establishing a community framework for complexity modeling using agent based models and cyberinfrastructure. *Journal of Artificial Societies and Social Simulation*, 9(4):6, 2006. URL: `https://www.jasss.org/9/4/6.html`.

2 Li An, Volker Grimm, and Billie L. Turner II. Editorial: Meeting grand challenges in agent-based models. *Journal of Artificial Societies and Social Simulation*, 23(1):13, 2020. `doi:10.18564/jasss.4012`.

3 Jennifer Badham, Pete Barbrook-Johnson, Camila Caiado, and Brian Castellani. Justified stories with agent-based modelling for local COVID-19 planning. *Journal of Artificial Societies and Social Simulation*, 24(1):8, 2021. `doi:10.18564/jasss.4532`.

4 Stephen W. Hawking. *A Brief History of Time: From the Big Bang to Black Holes*. Bantam Press, London, UK, 1988.

5 Gary Polhill. Antisocial simulation: using shared high-performance computing clusters to run agent-based models. *Review of Artificial Societies and Social Simulation*, 14 December 2022, 2022. URL: `https://rofasss.org/2022/12/14/antisoc-sim/`.

6 Flaminio Squazzoni, Gary Polhill, Bruce Edmonds, Petra Ahrweiler, Patrycja Antosz, Geeske Scholz, Émile Chappin, Melania Borit, Harko Verhagen, Francesca Giardini, and Nigel Gilbert. Computational models that matter during a global pandemic outbreak: A call to action. *Journal of Artificial Societies and Social Simulation*, 23(2):10, 2020. `doi:10.18564/jasss.4298`.

[3] The "reusable building blocks" work, such as that being run by CoMSES.Net is a case in point. See `https://github.com/comses-model-building-blocks`

7 Jason Thompson, Rod McClure, Tony Blakely, Nick Wilson, Michael G. Baker, Jasper S. Wijnands, Thiago Herick De Sa, Kerry Nice, Camilo Cruz, and Mark Stevenson. Modelling SARS-CoV-2 disease progression in Australia and New Zealand: an account of an agent-based approach to support public health decision-making. *Australian and New Zealand Journal of Public Health*, 46(3):292–303, 2022. doi:10.1111/1753-6405.13221.