The Matrix: An Agent-Based Modeling Framework for Data Intensive Simulations

P. Bhattacharya[1], S. Ekanayake[3], C. J. Kuhlman[1], C. Lebiere[2], D. Morrison[2], S. Swarup[1], M. L. Wilson[1], and M. G. Orr[1]

[1]Network Systems Science and Advanced Computing Division, Biocomplexity Institute and Initiative, University of Virginia
[2]Human-Computer Interaction Institute, School of Computer Science, Carnegie Mellon University
[3]Lawrence Berkeley National Laboratory

1. Overview

- The Matrix is an agent-based modeling (ABM) framework
- The Matrix is free and open source software
  - `github.com/NSSAC/socioneticus-matrix`
- Can be used to model 'hybrid' time simulations
  - A combination of discrete time and discrete event
- Specialized for 'compute and data intensive' simulations
- Successfully used to model 3M individual cognitive agents
  - A three order of magnitude increase over previous studies

2. Capabilities of the Matrix

- Allows writing agents in popular programming languages
  - Python, R, C, C++, Java, Lisp, …
- Supports use of GPU units, and popular neural network libraries
  - TensorFlow, PyTorch, Keras, Lens, …
- Enables use of cognitive system libraries like ACT-R
- Supports running simulations on cloud platforms
  - Amazon EC2, Google Compute Cloud, and Microsoft Azure
- Gracefully handles large (≈ hundreds of gigabytes) system state

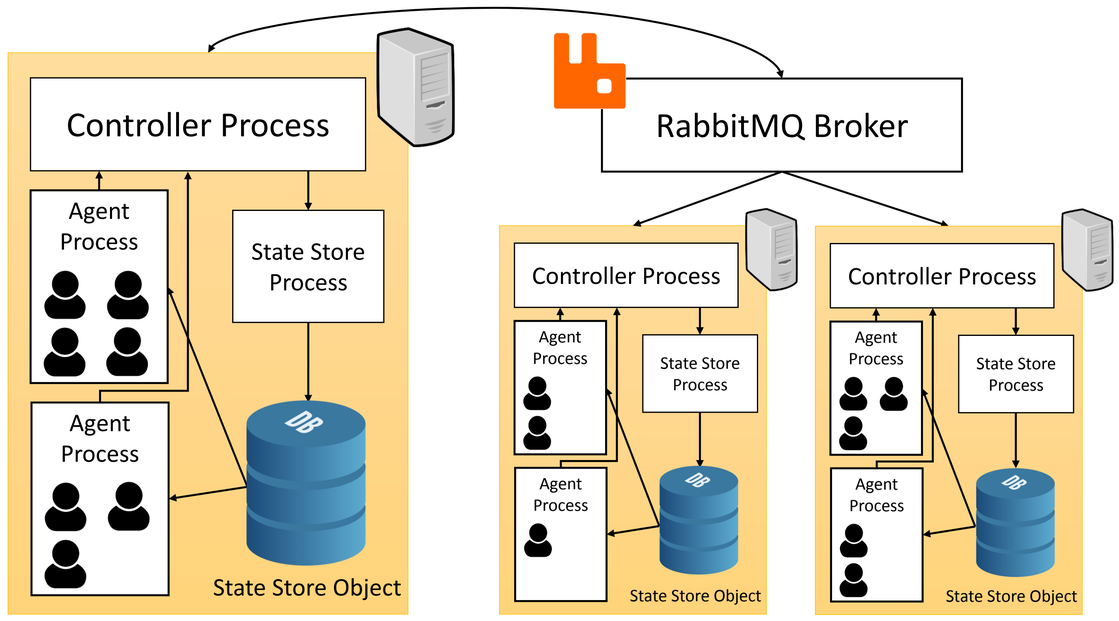
3. Models Implemented with the Matrix

Name	Model Type	Prog. Lang.
Freq-Stat	Frequentist statistical model	Python
Soc-Th	Social structure theory model	Python
CM-ANN	Artificial neural network model	C++
CM-Bayes	Bayesian cognitive theory model	R
CM-ACTR	ACT-R cognitive theory model	Common Lisp

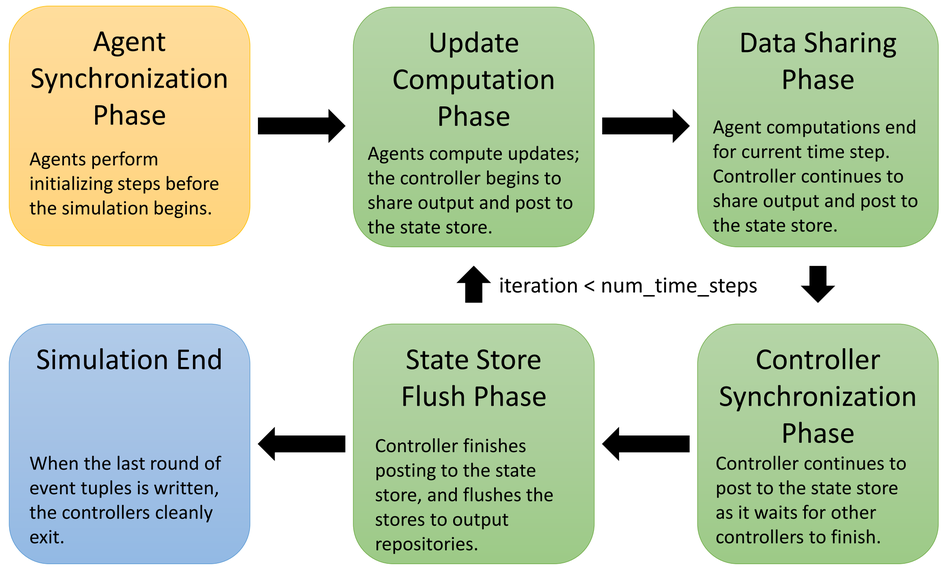
4. Formal Model of a Matrix Simulation

- System state: $x(0) \to x(1) \to \cdots \to x(n)$
- Transition function: $x(t+1) = g_{sim}(x(t))$
- Compute updates and apply: $g_{sim}(x(t)) = g_{red}(x(t), g_{act}(x(t)))$
- Compute updates for v_j: $g_{act}(x(t)) = \bigcup_{v_j \in V} g_{act}^{v_j}(x(t))$

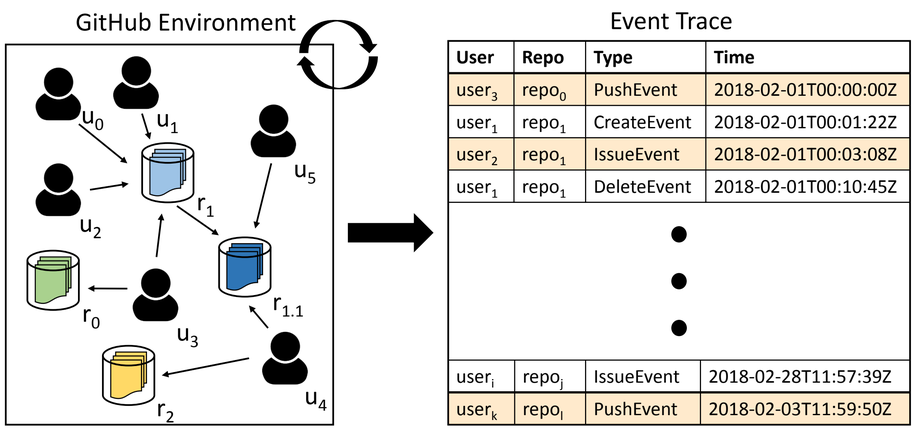
5. The Matrix Runtime System



6. States of a Controller Process



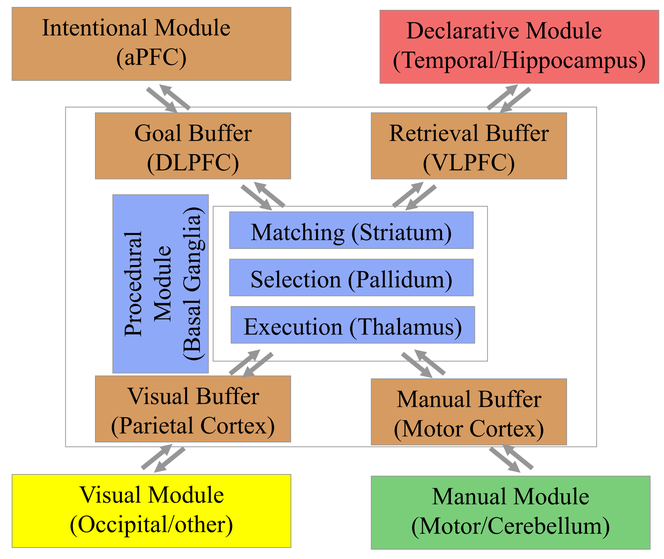
5. Simulating GitHub



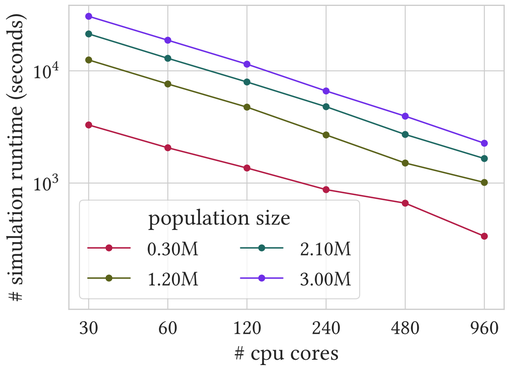
- Learning: Learn parameters from training event trace
- Simulation: Generate simulated event trace

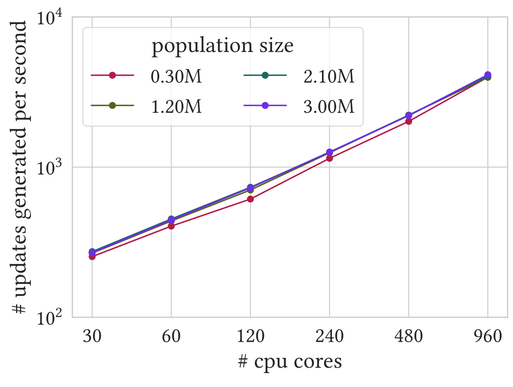
6. An ACT-R Model of GitHub

- CM-ACTR used the ACT-R library and was written in Common Lisp
- CM-ACTR used only declarative memory and procedural modules
- Previously seen events stored in memory
- New event computed one element of the event tuple at a time
- Chosen elements used as retrieval context for next elements



7. Scaling up CM-ACTR





8. Conclusion

- The Matrix facilitates rapid prototyping of 'compute and data intensive' agent models
- The Matrix allows flexibility in use of programming languages and libraries
- Matrix simulations can run on commodity clusters and cloud computing platforms

This work is partially supported by DARPA (Grant No. FA8650-18-C-7826) and DTRA (Grant No. HDTRA1-17-0118)