Parallelization of Entity-Based Models in Computational Social Science: A Hardware Perspective

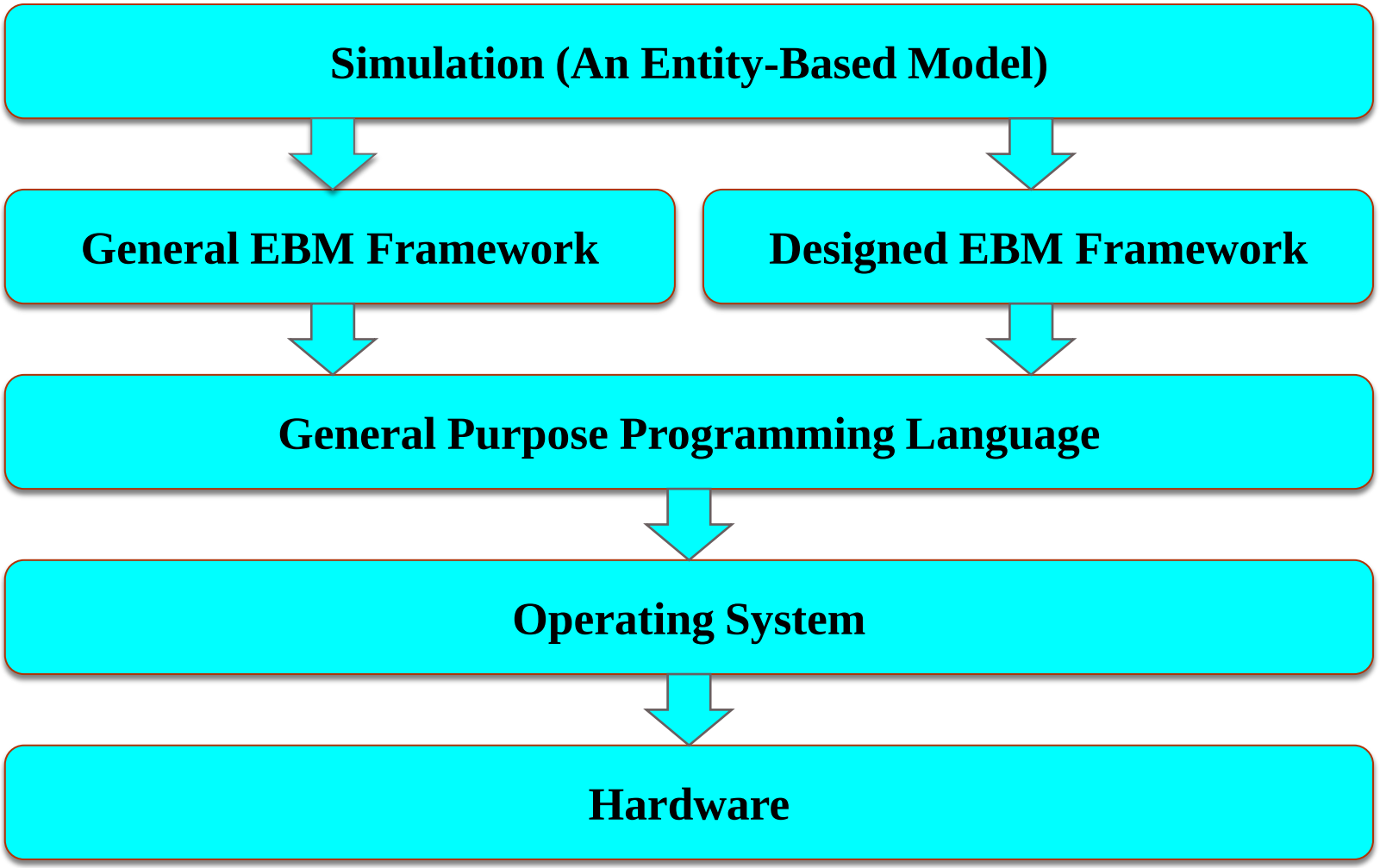
Dale K. Brearcliffe

27 October 2017

Research Questions

1) Does the underlying hardware play a role in the social scientist's capability to create large-scale models?
2) If so, does the hardware change the approach and skills needed for modeling?
3) Is it worth the effort?

An Abstraction of an Entity-Based Model's Relationship to Hardware



Programming Language Considerations

- **EBM** frameworks were rejected because there is no assurance that the framework developers had taken full advantage of the underlying hardware
- Programming languages are closer to the hardware and makes it easier to see how the hardware created or solved **EBM** challenges
- The programming language was selected based on:
  - General availability
  - The **EBM** that was to be implemented
  - The hardware architecture
- This process aided in answering research question three, "Is it worth it?"

Hardware Categories

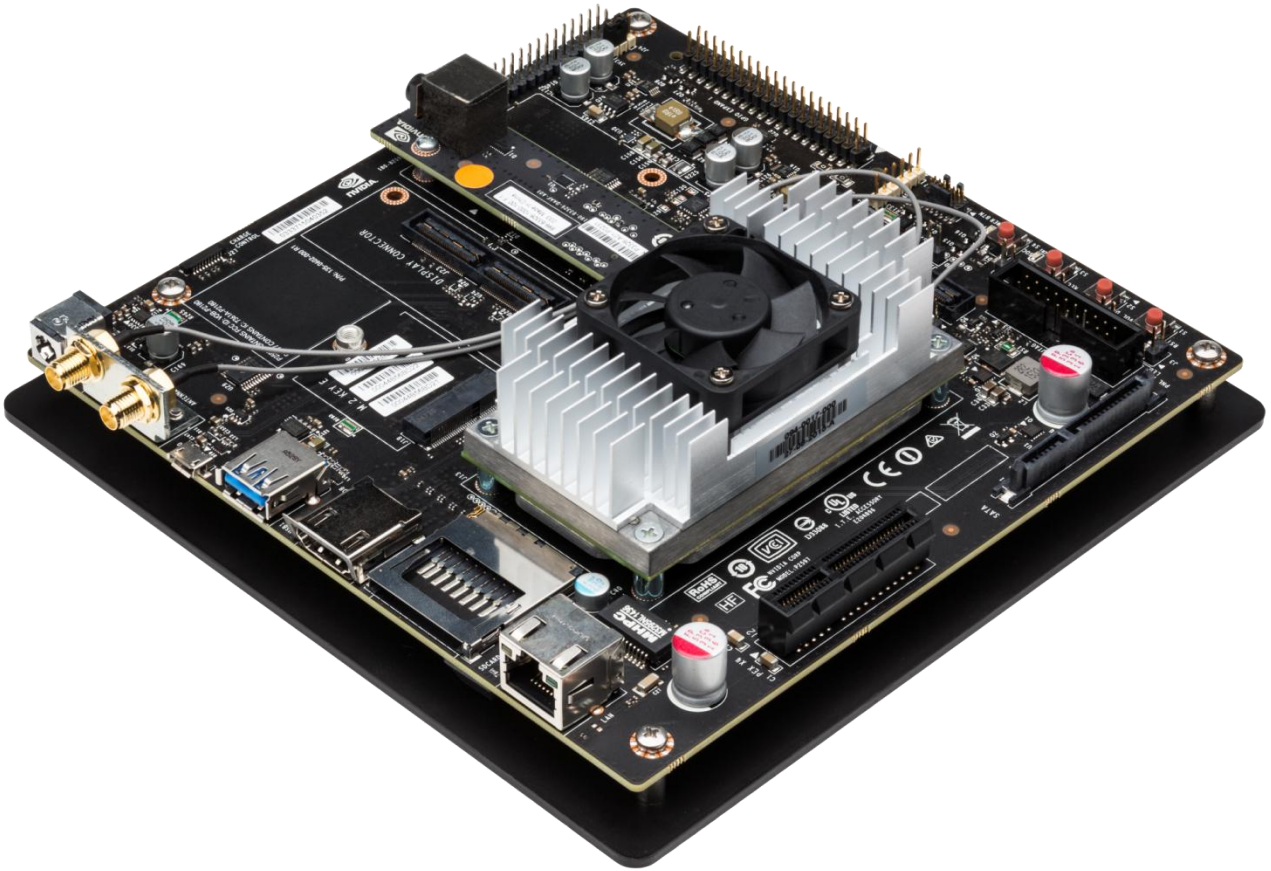
Hardware Category	Description
Multi-Core Central Processing Unit	A **CPU** with more than one **logical processing element**.
Graphic Processing Unit	Composed of thousands of **logical processing elements**. These **LPE**s do not communicate with each other, are simpler, and less capable individually than those found in **CPU**s.
Application Specific Integrated Circuit	Contains multiple highly specialized **RISC cores** that operate independently or collectively on internal **NoC**s.
Homogeneous Computing Nodes	Nodes of similarly designed and configured multi-core computers physically located together and optimized for high-speed communication.
Heterogeneous Computing Nodes	A distributed network of nodes different types of computer hardware and operating systems. They can be geographically distributed, communicating across the Internet.

Pseudo Code Parallelization of the ZIT Model

```
INSTANTIATE and INITIALIZE BUYER, SELLER, DATA and THREAD objects;
Assign sub-populations of BUYERS and SELLERS to THREADS;
FORK all THREADS;
FOR each THREAD, REPEAT:
 - Randomly activate 1 BUYER agent + 1 SELLER agent:
 - - BUYER proposes a BID price;
 - - SELLER proposes an ASK price;
 - - IF (BID > ASK) THEN
 - - - Pick EXECUTION price between BID and ASK;
 - - - INCREMENT BUYER holdings;
 - - - DECREMENT SELLER holdings;
 - - - Collect DATA on the trade;
 - INCREMENT the attempted number of trades;
 - END when maximum trade attempts exceeded;
JOIN all THREADS;
Collect final DATA;
```

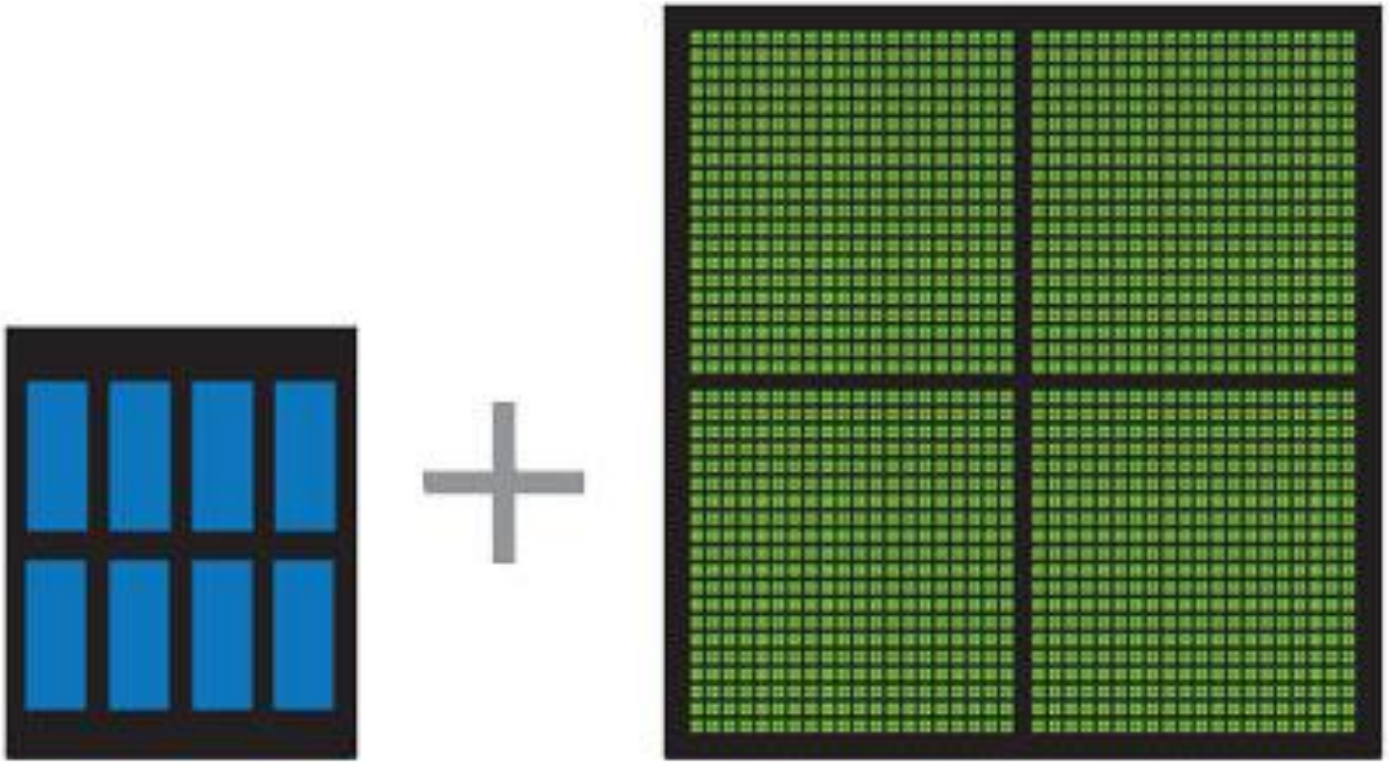
(Source: McCabe, et al., 2016)

General Purpose Graphic Processing Unit (GPGPU)



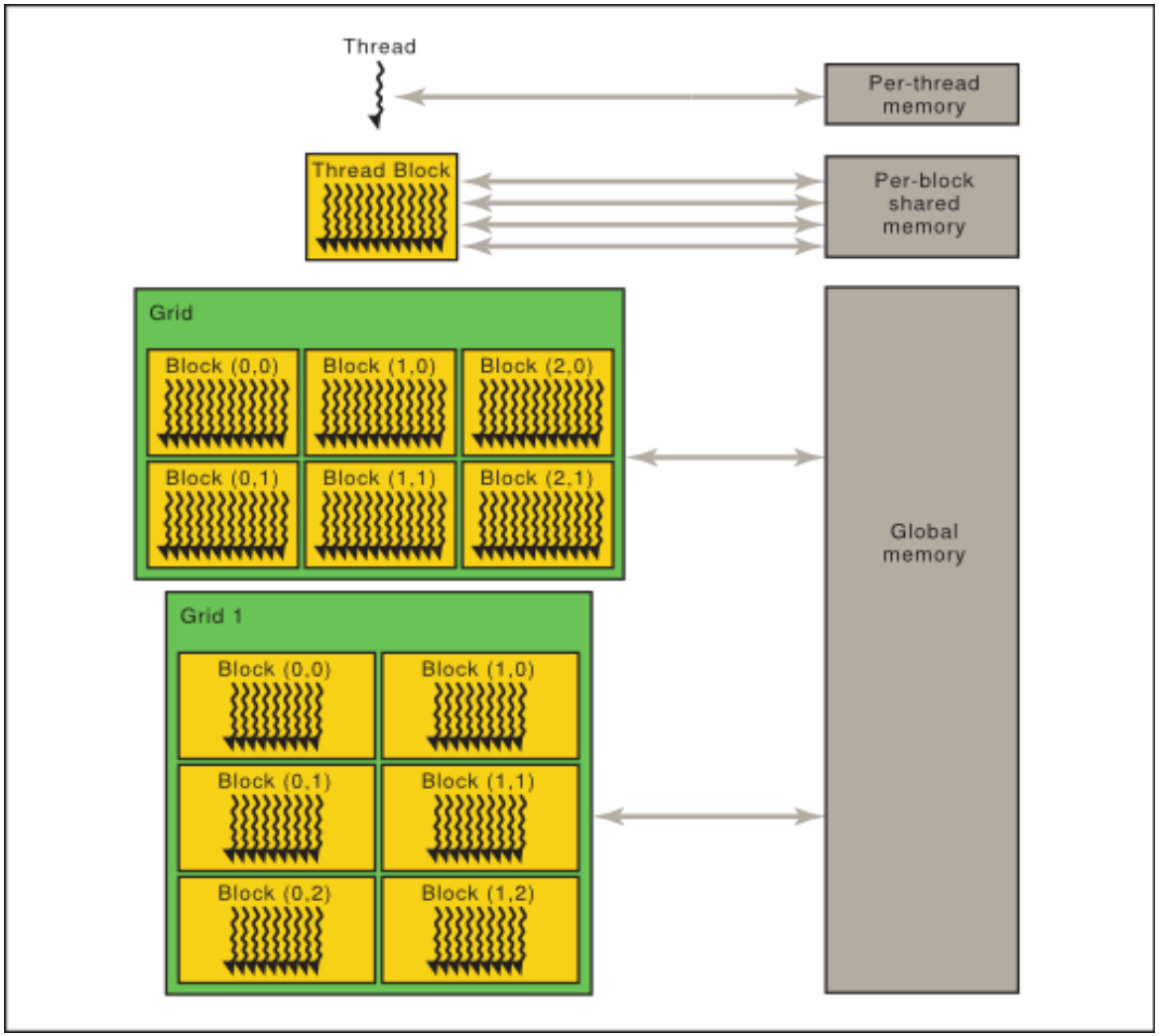
NVIDIA Jetson TX1 Development Kit
https://devblogs.nvidia.com/parallelforall/nvidia-jetson-tx1-supercomputer-on-module-drives-next-wave-of-autonomous-machines/

Comparison of CPU to GPGPU Cores



(Source: NVIDIA Corporation, 2017d)

GPGPU Threads, Blocks, Grids, and Memory

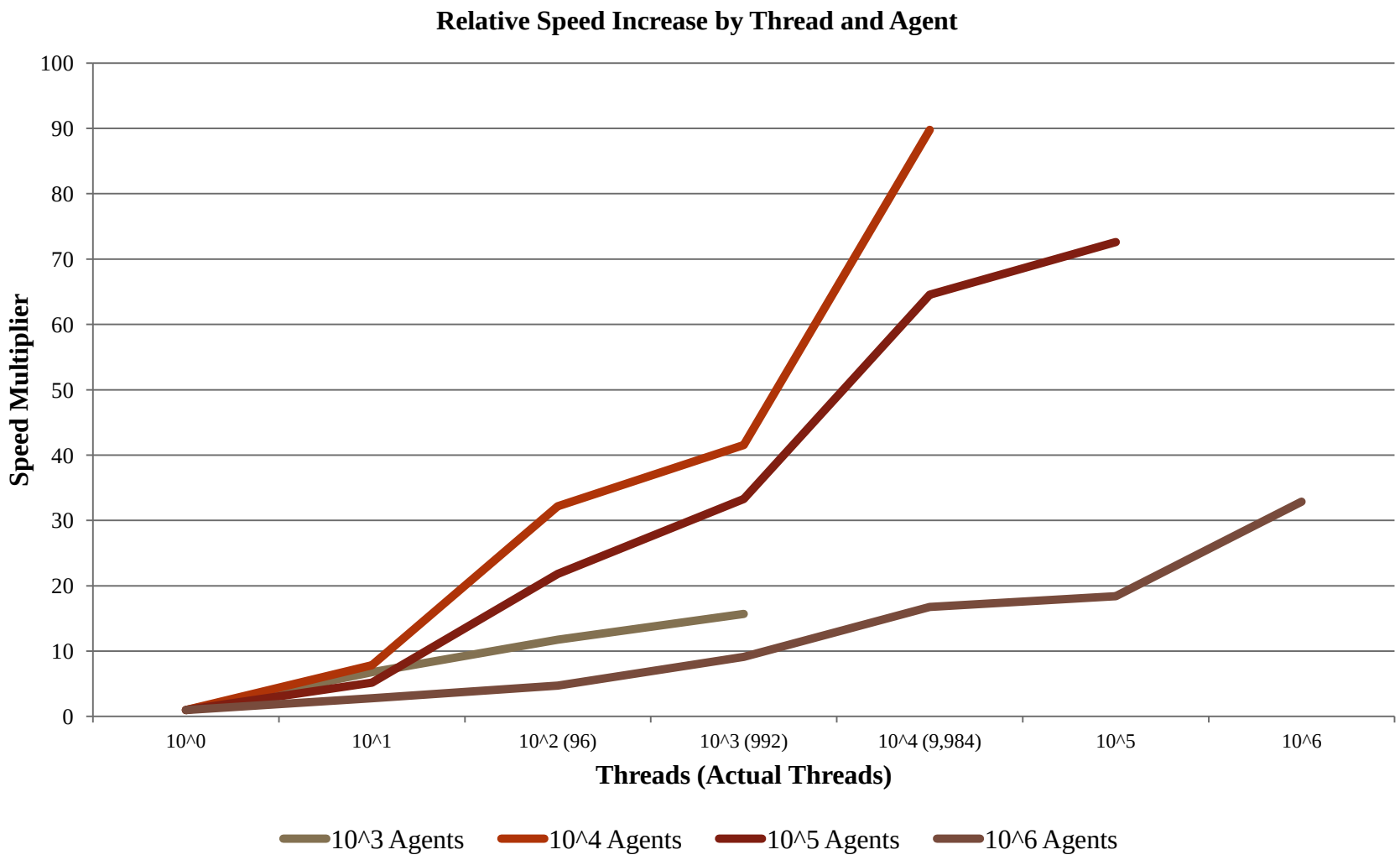


(Source: NVIDIA Corporation, 2017e)

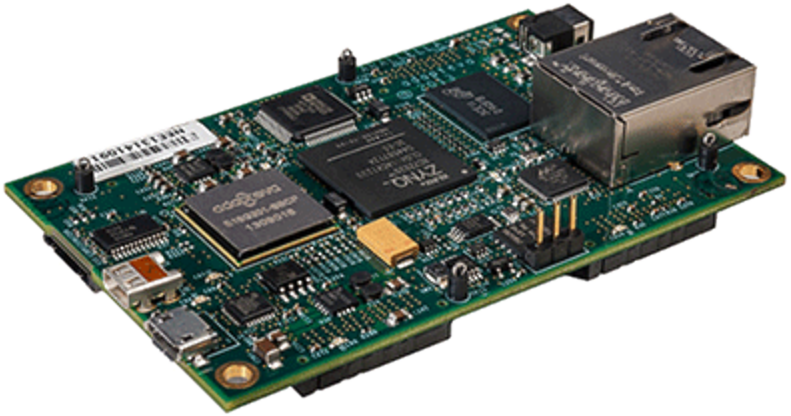
GPGPU ZIT Results

	Mean in Milliseconds				Standard Deviation in Milliseconds			
Threads	**10^6 Agents**	**10^5 Agents**	**10^4 Agents**	**10^3 Agents**	**10^6 Agents**	**10^5 Agents**	**10^4 Agents**	**10^3 Agents**
10^0	158,001	50,929	5,916	306	21,678	13,643	2,432	53
10^1	56,655	9,851	756	45	10,106	2,598	169	19
10^2	33,566	2,333	184	26	21,664	374	26	12
10^3	17,353	1,530	142	20	5,235	129	24	8
10^4	9,430	789	66	---	3,483	35	14	---
10^5	8,588	702	---	---	3,378	2	---	---
10^6	4,806	---	---	---	2,461	---	---	---

Speedup Results Using a GPGPU

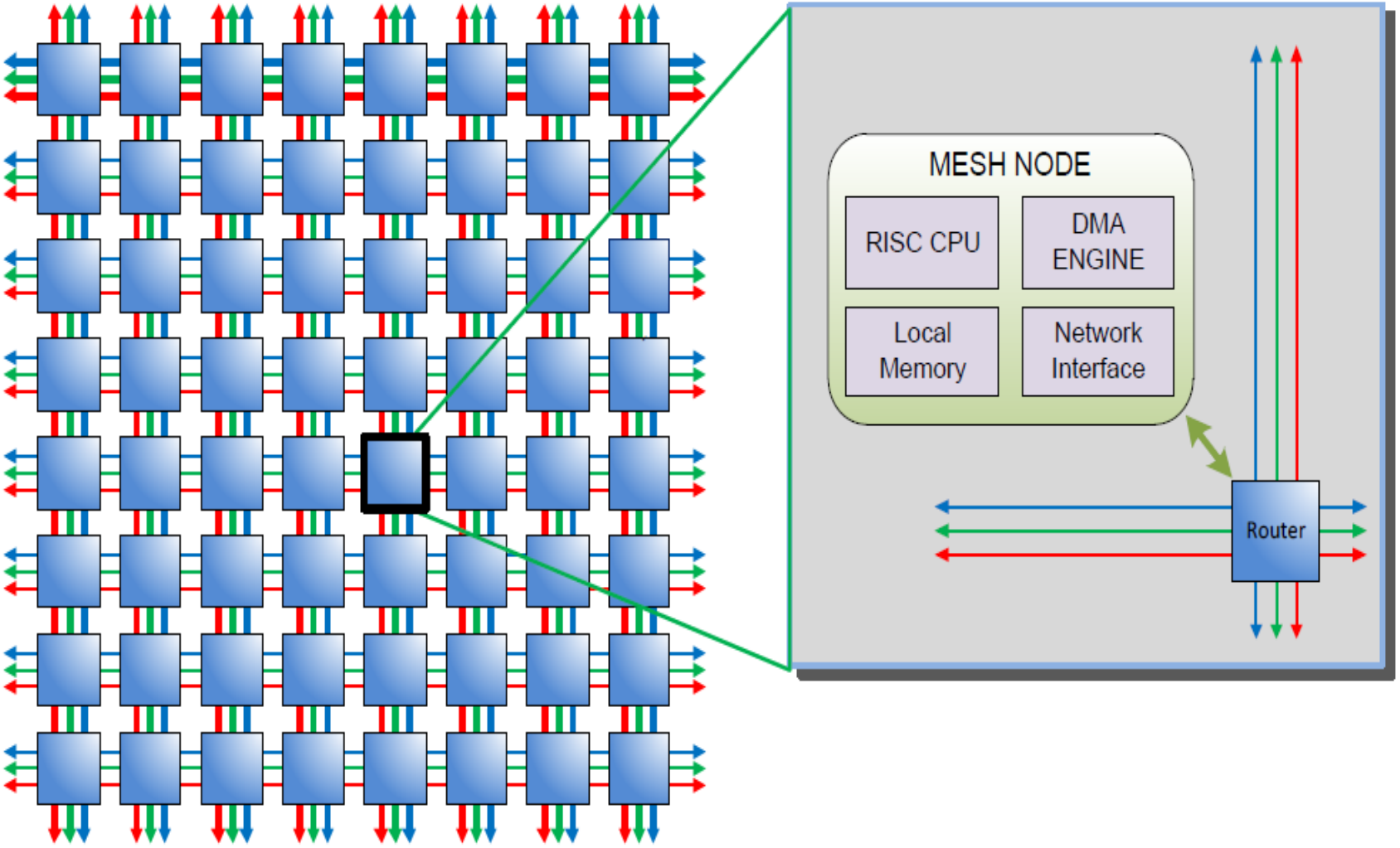


Application Specific Integrated Circuit (ASIC)



Adapteva Parallella with Epiphany III ASIC
http://www.adapteva.com/parallella/

The Epiphany Architecture (64 cores shown)

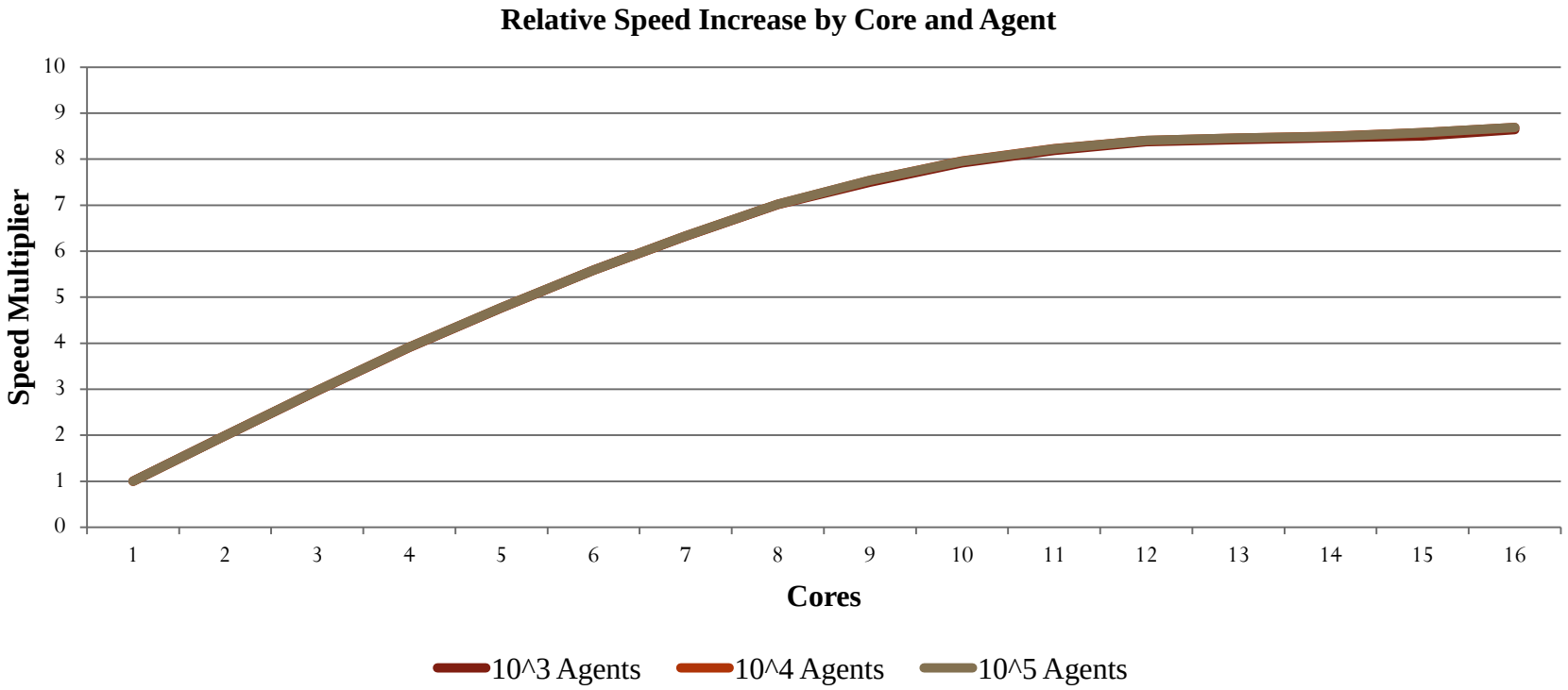


(Source: Adapteva, 2013)

ASIC ZIT Results

	Mean in Milliseconds			Standard Deviation in Milliseconds		
Cores	**10^5 Agents**	**10^4 Agents**	**10^3 Agents**	**10^5 Agents**	**10^4 Agents**	**10^3 Agents**
1	986,110	98,693	9,874	437	81	7
2	494,684	49,554	4,962	390	36	5
3	332,037	33,231	3,328	265	28	3
4	251,713	25,184	2,525	188	20	2
5	206,530	20,660	2,070	140	15	3
6	176,471	17,658	1,767	105	13	2
7	155,705	15,580	1,561	87	7	2
8	140,431	14,056	1,408	89	14	1
9	130,789	13,091	1,316	84	5	1
10	123,955	12,404	1,246	157	12	1
11	119,898	11,995	1,206	117	18	3
12	117,324	11,741	1,178	75	11	2
13	116,556	11,668	1,172	107	13	1
14	116,121	11,618	1,166	71	9	1
15	115,007	11,509	1,161	46	5	15
16	113,511	11,359	1,142	115	17	5

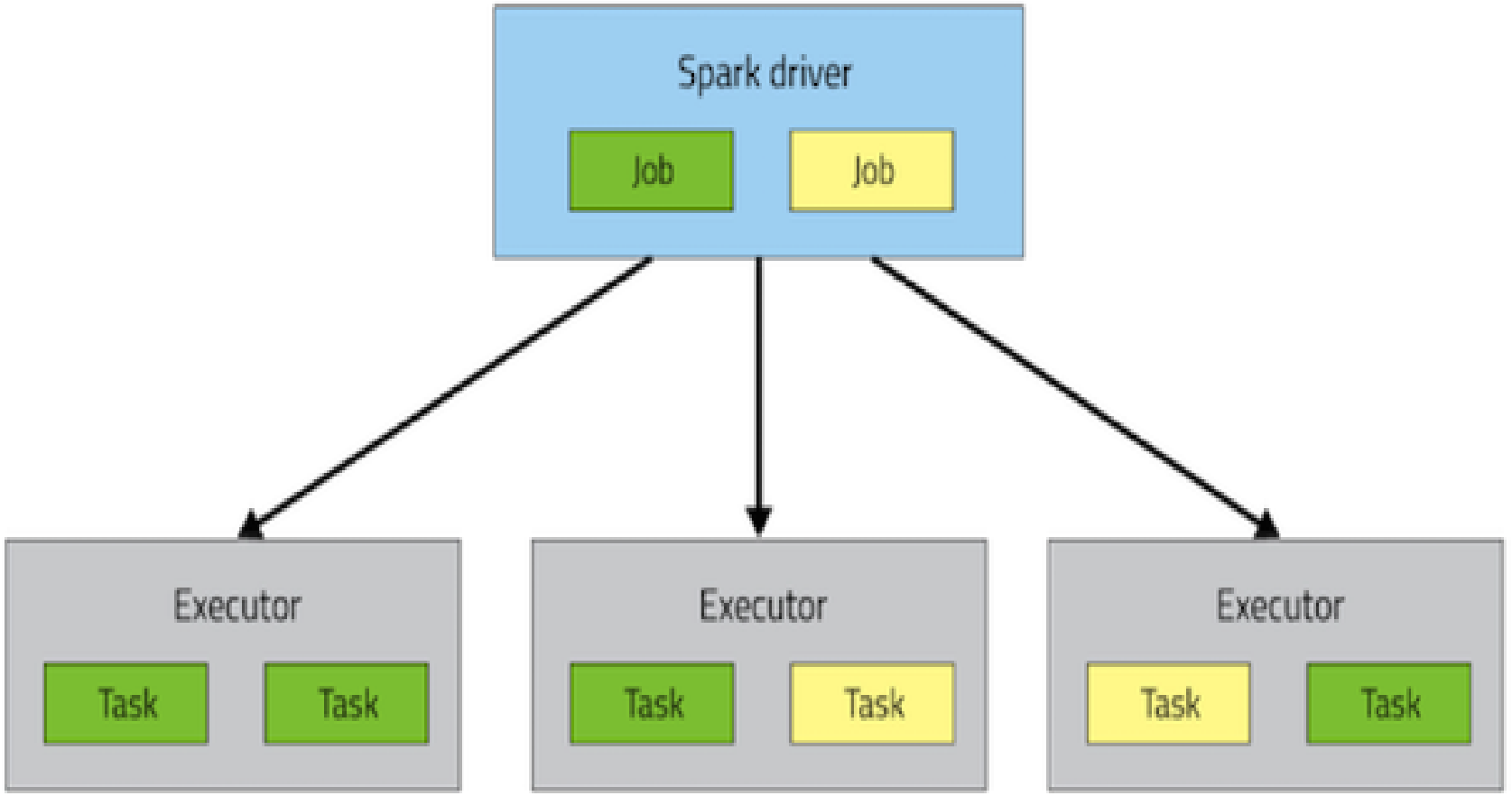
Speedup Results Using an ASIC



High Performance Computing (HPC) Cluster

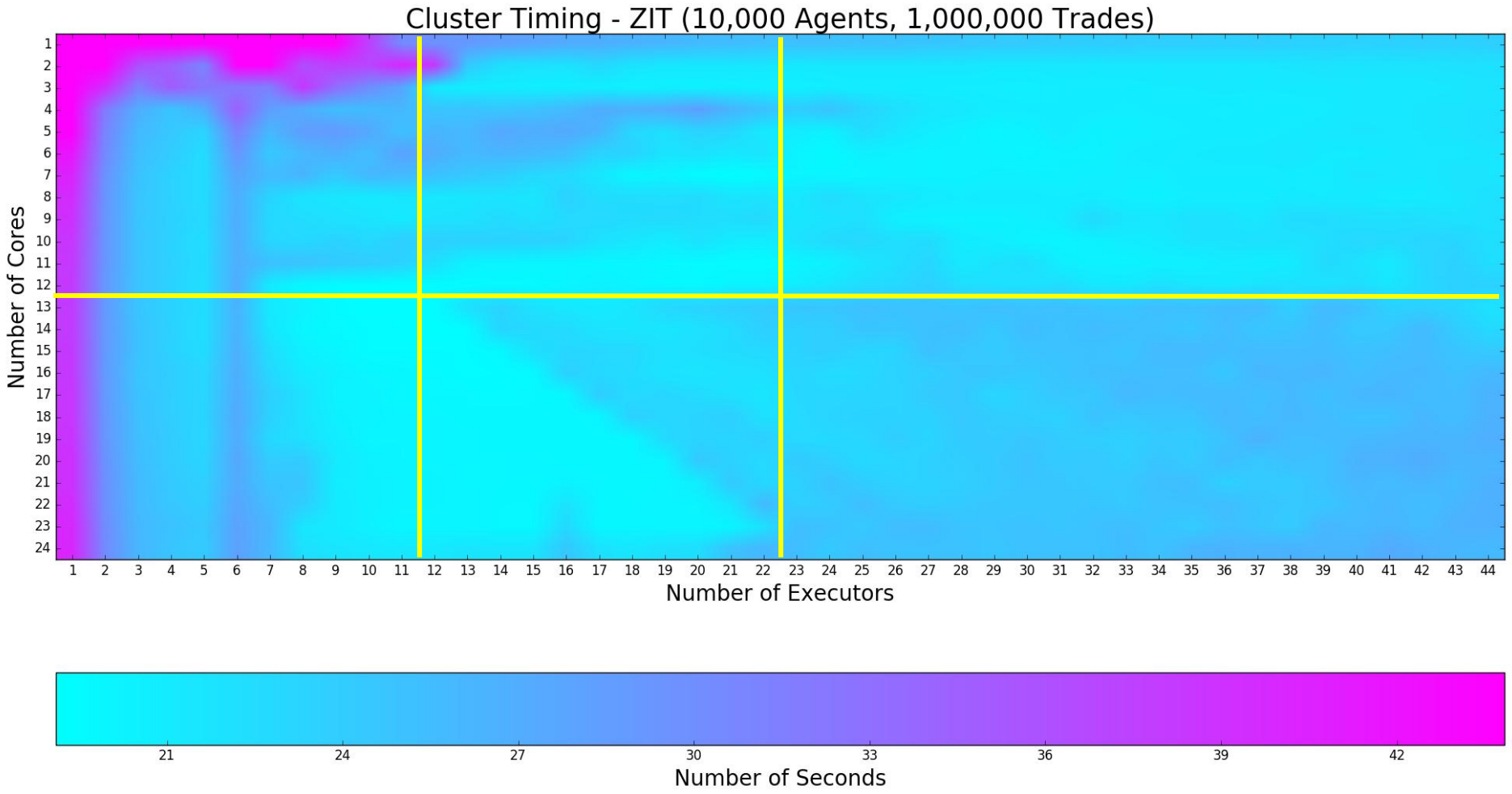
Blue Gene/L
https://asc.llnl.gov/computing_resources/bluegenel/

How Spark Manages Work

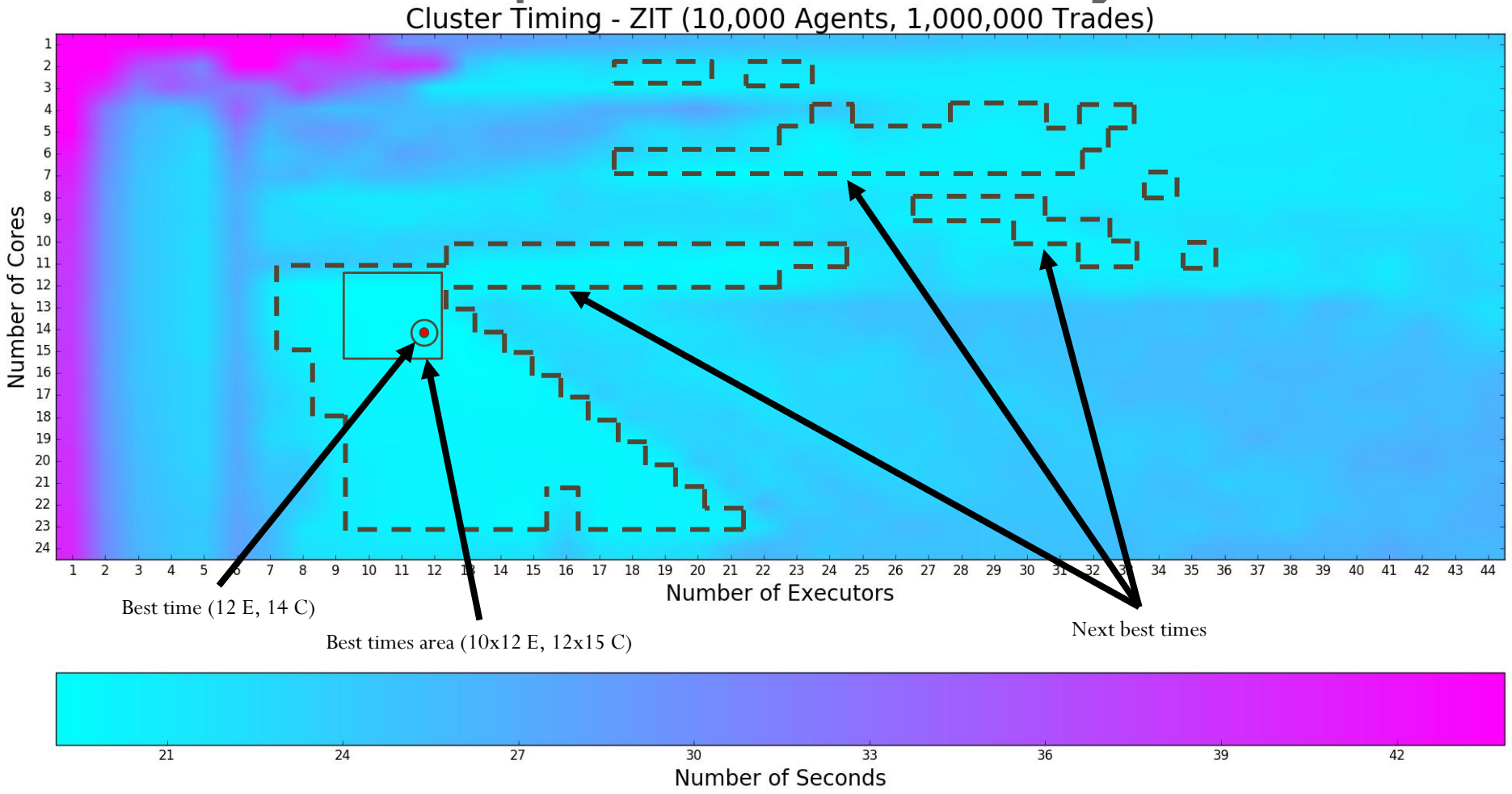


(Source: Cloudera, 2015a)

ZIT HPC Optimization Parameter Sweep Results



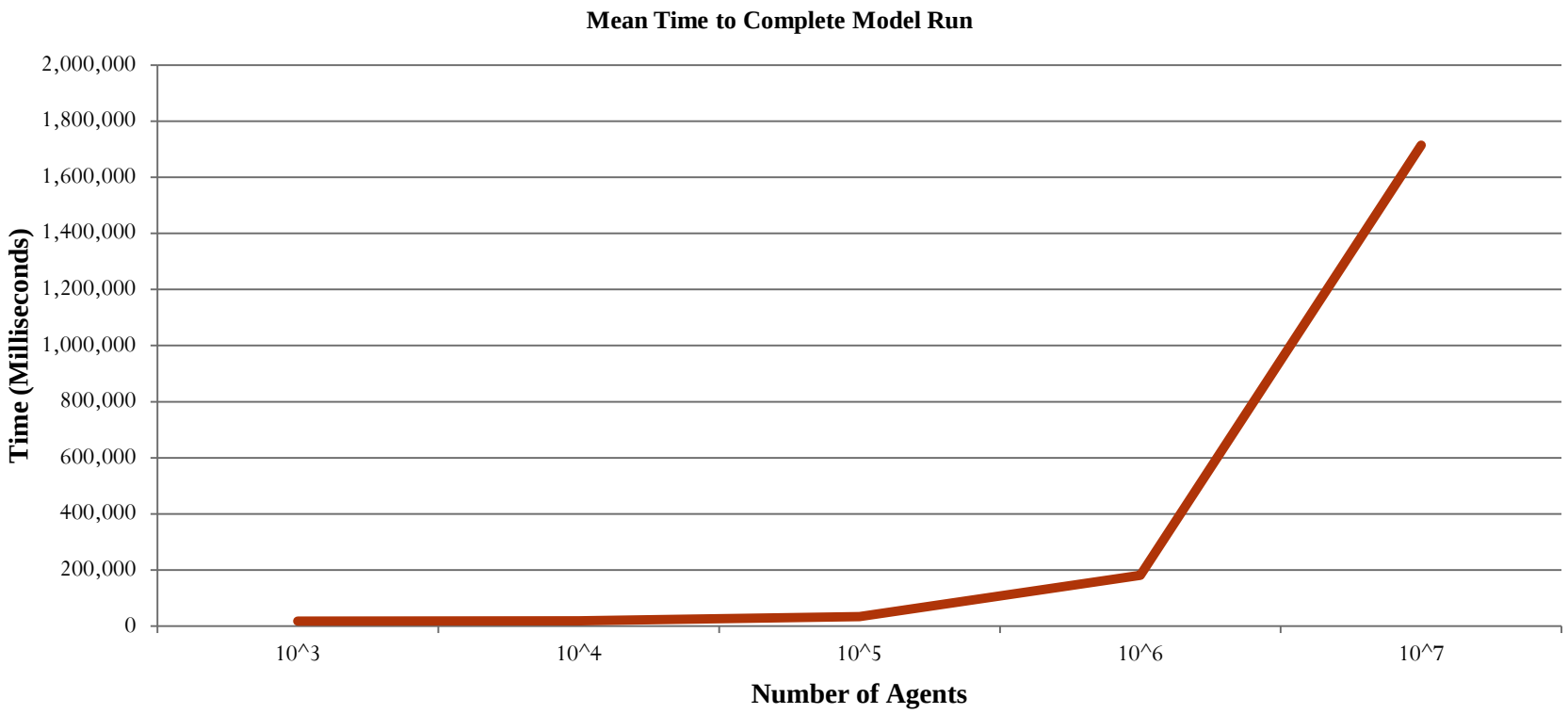
ZIT HPC Optimization Analysis



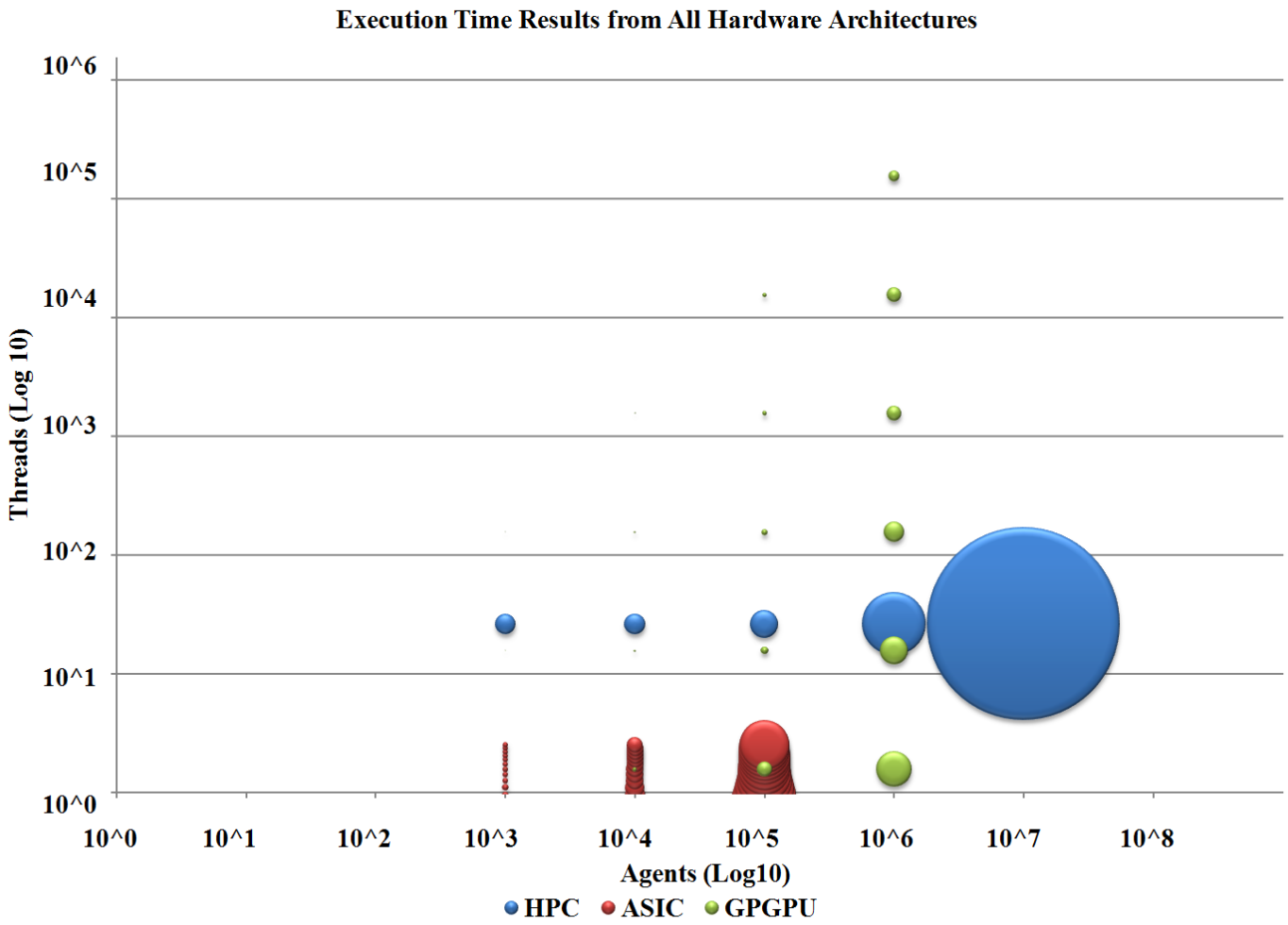
ZIT Time to Complete on HPC Cluster

Agents	Mean in Milliseconds	Standard Deviation in Milliseconds
10^3	17,688	813
10^4	19,068	765
10^5	34,427	734
10^6	180,999	3,971
10^7	1,714,556	17,418

HPC ZIT Speedup Results



Execution Times



Larger circles are longer execution times

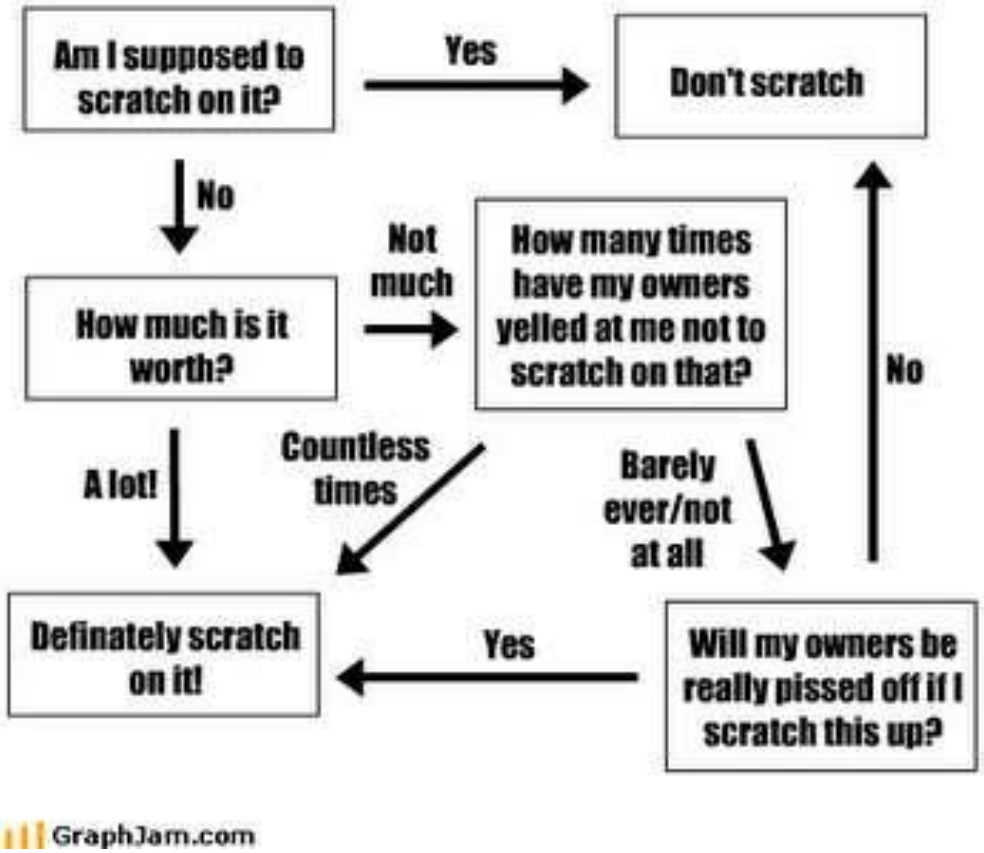
Research Questions

1) Does the underlying hardware play a role in the social scientist's capability to create large-scale models?
   - **The correct hardware can increase the social scientist's capability to create large-scale models by allowing for the possibility of parallelism**
2) If so, does the hardware change the approach and skills needed for modeling?
   - **The hardware does change the approach and skills needed for modeling. The amount of change and the level of programming skills will vary between hardware types. A SPMD co-processer architecture may require the least additional skills**
3) Is it worth the effort?
   - **Is it worth the effort only when it is the only available solution**

Future Work

- Hardware specific research should include:
  - Increasing the number of data nodes in **HPC** clusters
  - Using multiple **GPGPU** boards on a single host
  - **HPC** clusters whose commodity computers include **GPGPU** boards
  - Chaining **ASIC** boards to increase the number of **LPE**s
  - Exploring **HPC** clusters with installed **ASIC** coprocessors
- Using more complex **EBM**s that requires agent communication during and not just at the conclusion of the model run
- Tying agent and environment locality to an **LPE** minimizing data transfer across computational boundaries

How a cat decides whether to scratch on something



GraphJam.com

List of Abbreviations

Agent-Based Model	ABM
Application Programming Interface	API
Application Specific Integrated Circuit	ASIC
Cellular Automata	CA
Central Processing Unit	CPU
Computational Social Science	CSS
Compute Unified Device Architecture	CUDA
Entity-Based Model	EBM
General Purpose Graphics Processing Unit	GPGPU
GNU Compiler Collection	GCC
GNU's Not Unix (A recursive acronym)	GNU
Graphics Processing Unit	GPU
High Performance Computing	HPC
Individual-Based Model	IBM

List of Abbreviations

Integrated Circuit	IC
Logical Processing Element	LPE
Multiple Instruction Multiple Data	MIMD
Multiple Instruction Streams Single Data Stream	MISD
Network on Chip	NoC
NVIDIA CUDA Compiler	NVCC
Reduced Instruction Set Computing	RISC
Single Instruction Multiple Data	SIMD
Single Instruction Stream Single Data Stream	SISD
Single Program Multiple Data	SPMD
System on Chip	SOC
Yet Another Resource Negotiator	YARN
Zero-Intelligence Traders	ZIT

Lexicon

- **Application Programming Interface (API)**: A means for a software component to communicate with another software component. A component may provide the **API** as a standardized means by which another component may request access to data or resources.
- **ASIC (Application Specific Integrated Circuit)**: An integrated circuit (**chip**) designed for some specific use. An example for parallel processing is a set of homogeneous **RISC** processors in a mesh grid on a single **chip**.
- **Cellular Automata (CA)**: A limited type of **EBM** that is of the simplest types of social simulation models. There are no agents; instead, each region in some environment reacts to its surroundings. Over time, these reactions can create the illusion of movement.
- **Chip**: See **Integrated Circuit**.
- **Cluster**: See **Computer Cluster**.
- **Computer Cluster**: A collection of computers of which some or all can be focused on the same task. Resource allocation within the cluster is left to cluster management software, thus the cluster can be viewed as a single system.
- **Concurrent Computing**: The ability to simultaneously execute several tasks that may be unrelated. Similar, but distinct from **parallel computing**.
- **Core**: See **Logical Processing Element**.
- **CPU (Central Processing Unit)**: An electronic circuit, or **chip**, which performs low-level program instructions. Multiple CPUs on a single **chip** are referred to as **cores**. A single chip with multiple cores can be called a **socket**.
- **EBM Framework**: A software application purposely designed to facilitate the execution of an Entity-Based Model.

Lexicon

- **GPGPU (General purpose computing on graphics processing units**: The use of a **GPU** to perform non-graphic computational tasks normally performed by a **CPU**. **EBM frameworks** that use a **GPU** for computing fall under this definition.
- **GPU (Graphics Processing Unit)**: A system of parallel configurations that can efficiently process blocks of data simultaneously. Originally designed to speed up visual displays, they have been adapted to scientific computing. (See **SIMD**.)
- **Hadoop**: Apache Hadoop is open source software that provides an environment for distributing large data sets across a **computer cluster**. Also, see **YARN**.
- **High performance computing (HPC)**: An increase in computation power achieved by aggregating multiple computing resources and focusing them on a single task.
- **Integrated Circuit (IC)**: A collection of electronic components and wires on a single flat piece of semiconductor material. An essential component of most current computing systems.
- **Logical Processing Element (LPE)**: The use of hyper-threading technology can create the illusion of multiple processors where there is but one physically. Most often, there appears to be two logical processing elements per physical element. When determining the number of available cores, software applications count these logical processing elements.
- **MIMD (Multiple instruction streams, multiple data streams)**: Multiple cores work on multiple blocks of data simultaneously.
- **MISD (Multiple instruction streams, single data stream)**: Multiple cores work on the same block of data. Useful in situations where a **core** could fail to produce correct results.

Lexicon

- **Network on Chip (NoC)**: One or more communication networks between **cores** on a single **integrated circuit**. The network can operate either synchronously or asynchronously.
- **Parallel Computing**: The ability to simultaneously execute a group of tasks created by dividing an original task into many smaller identical tasks. Similar, but distinct from **concurrent computing**.
- **RISC (Reduced instruction set computing) coprocessor**: A special purpose processor using a limited but fast set of instructions that is tightly coupled to the **CPU**.
- **SOC (System on a Chip)**: A single **integrated circuit** that encompasses all necessary components for a functioning computer.
- **Socket**: See **CPU**.
- **SIMD (Single instruction stream, multiple data streams)**: A single set of instructions is simultaneously applied to multiple blocks of data. (See **GPU**.)
- **SISD (Single instruction stream, single data stream)**: A single set of instructions is applied to a single block of data. This is a sequential (non-parallel) process.
- **Speed Multiplier**: A relative comparison of execution time where the base is set to the time to complete using a single **LPE** or **core**. The **Speed Multiplier** is calculated by dividing the base execution time by the execution time using multiple **LPEs.**
- **Speedup**: See **Speed Multiplier**.
- **YARN (Yet Another Resource Negotiator)**: An operating system for large-scale distributed data processing. Used by **Hadoop**.