Distributed Simulation Challenges in Sensor Networks and the Cloud

Richard Fujimoto

School of Computational Science and Engineering
Georgia Institute of Technology

Acknowledgements

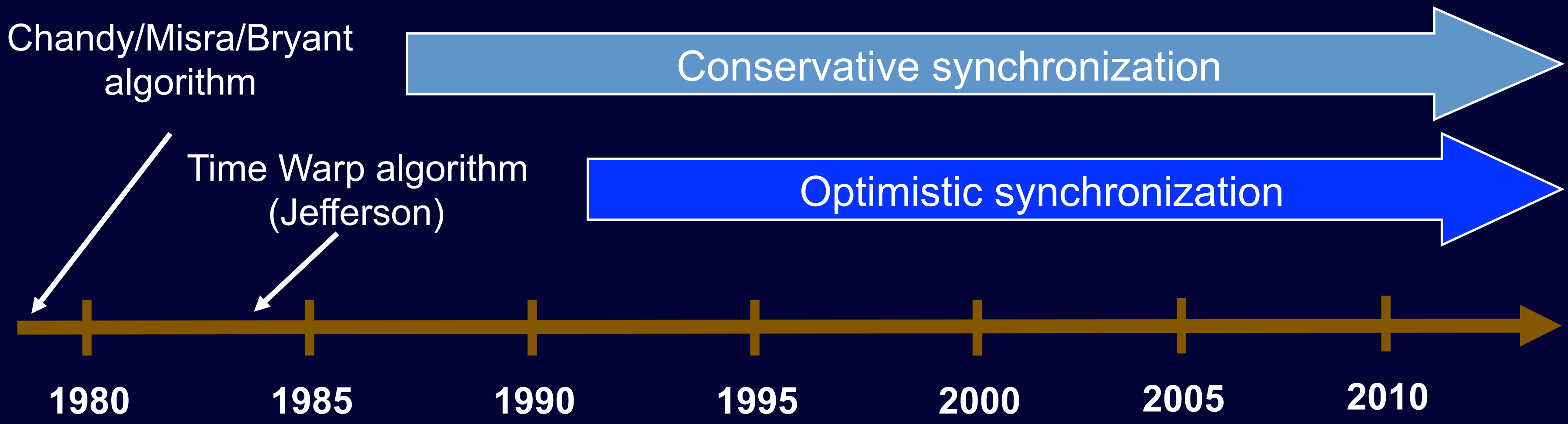
Christos Alexopoulos
Brandon Baker
Randall Guensler
Edward Hagler
Toyan Harvey
Dwayne Henclewood
Ya-Lin Huang
Michael Hunter
Kendra Jones
Hoe-Young Kim
Ying Li
Asad Malic
Alfred Park
Michael Pieper
Benjamin Stensland
Wonho Suh
Prashant Thiruvengadachari
Eric Thompson
Jewel Watts
Javier Young

Sponsors:
National Science Foundation
Georgia Department of Transportation

Outline

- Historical Perspective
- Cloud Computing (the current “big thing”)
- Embedded Distributed Simulation (the next “big thing”)

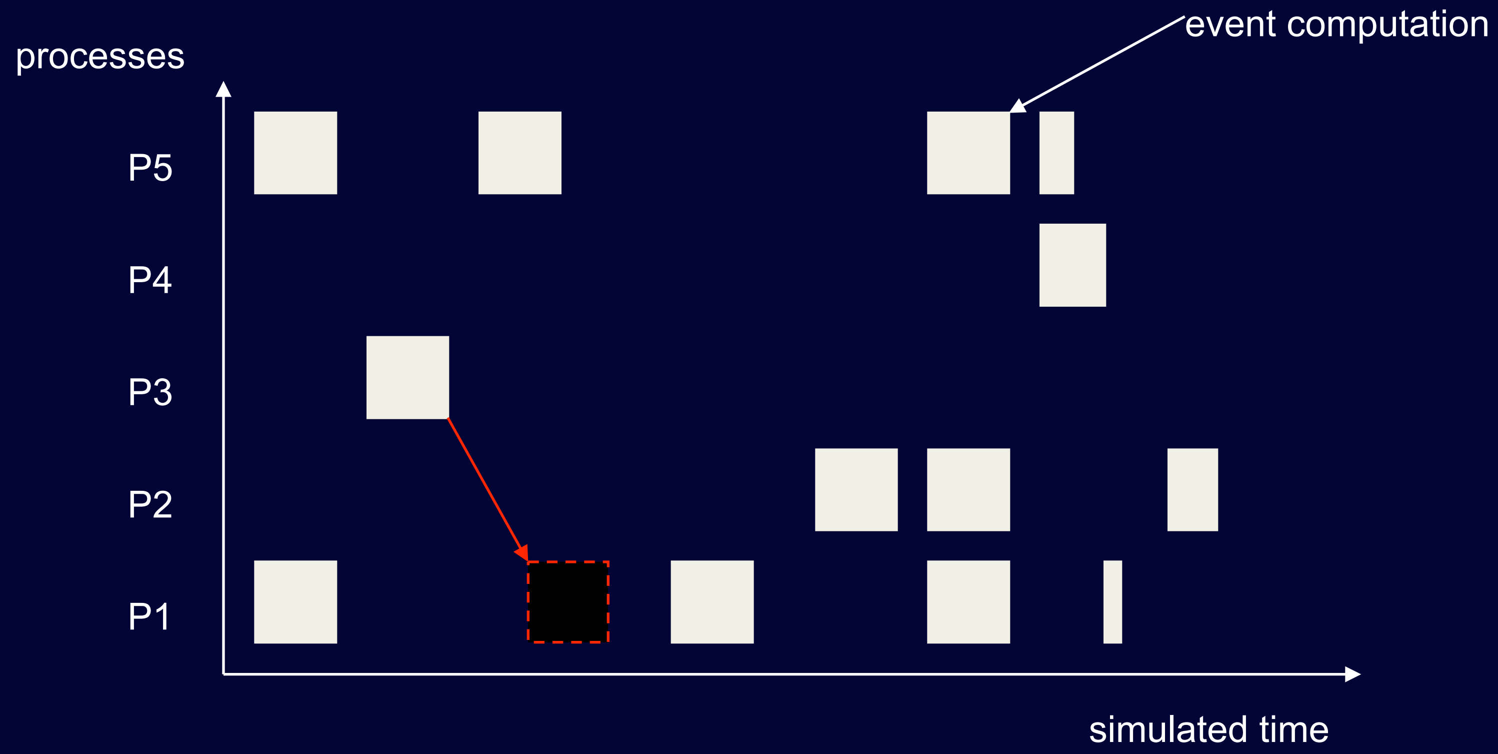
Parallel Discrete Event Simulation



- *Goal:* Speed up / scale the execution of large discrete event simulations via parallel computing
- *The synchronization problem*: How can one distribute the execution of a discrete event simulation across multiple processors and get the same results as a sequential execution?

The Synchronization Problem

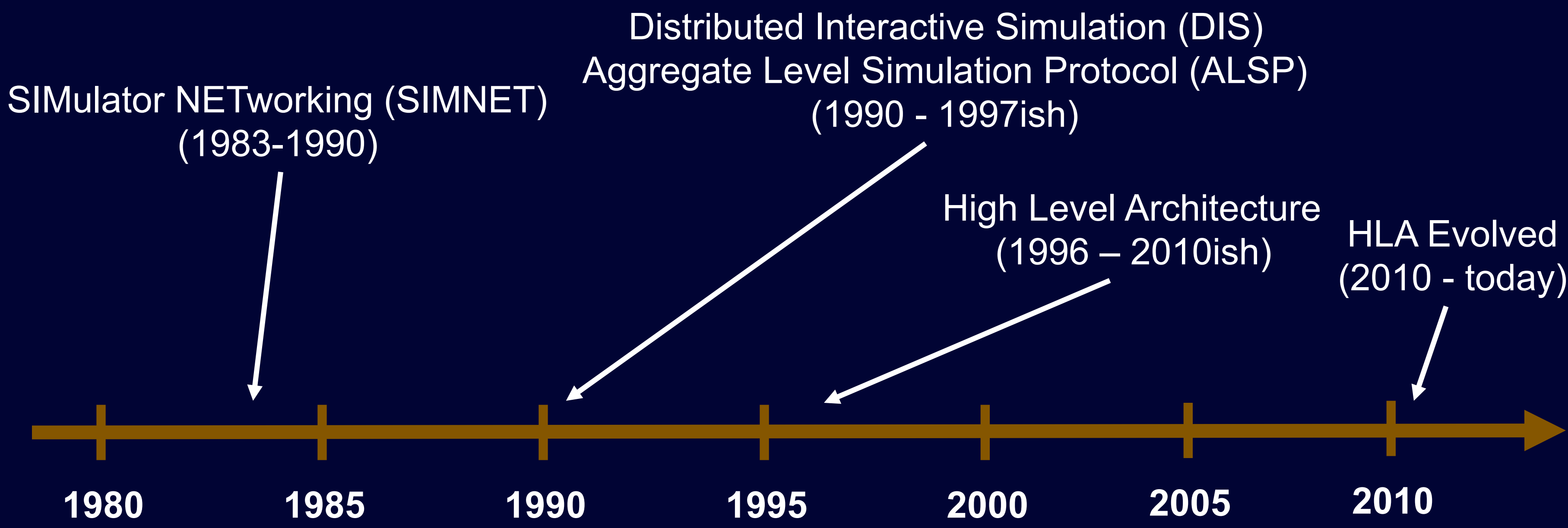
Many discrete event simulations contain too little computation in each time step to achieve much concurrent execution



Processes must be allowed to advance ahead of others in simulated time.
The synchronization problem: a simulation running on one processor might affect the past of another simulator

- Conservative: Block processes to ensure no events are received in the past
- Optimistic: Allow synchronization errors, but recover using a rollback procedure

Distributed Simulation



- *Goal*: Enable reuse of simulations for new purposes in training, analysis, engineering applications
- *The interoperability problem*: What architectures, processes, technology, standards and tools are needed to support interoperability among new and existing simulations

Retrospectives

- ORSA Journal on Computing special issue (summer 1993)
  - “Parallel Discrete Event Simulation: Will the Field Survive?” (Fujimoto)
  - Commentaries by Marc Abrams, Rajive Bagrodia, Yi-Bing Lin, Paul Reynolds, Jr.,Brian Unger, John Cleary
- The technology has not been widely adopted by the mainstream simulation community
  - Need to exploit parallel computing not compelling; one can simply wait for faster machines (Moore’s law)
  - Exploiting PDES technology is too hard to gain widespread acceptance

Retrospectives (cont.)

- Grand challenges in modeling and simulation workshop (Dagstuhl 2002)
  - Term “symbiotic simulation” coined [Turner]
  - Methodology and application challenges
  - Several subsequent “grand challenges in M&S” conferences
- Peer study: distributed simulation and virtual environments in industry (Strassburger, 2005)
  - High practical relevance and potential economic impact
  - Limited adoption in industry to date (outside defense)

The Next Big Thing?

To have impact, focus on emerging platforms;

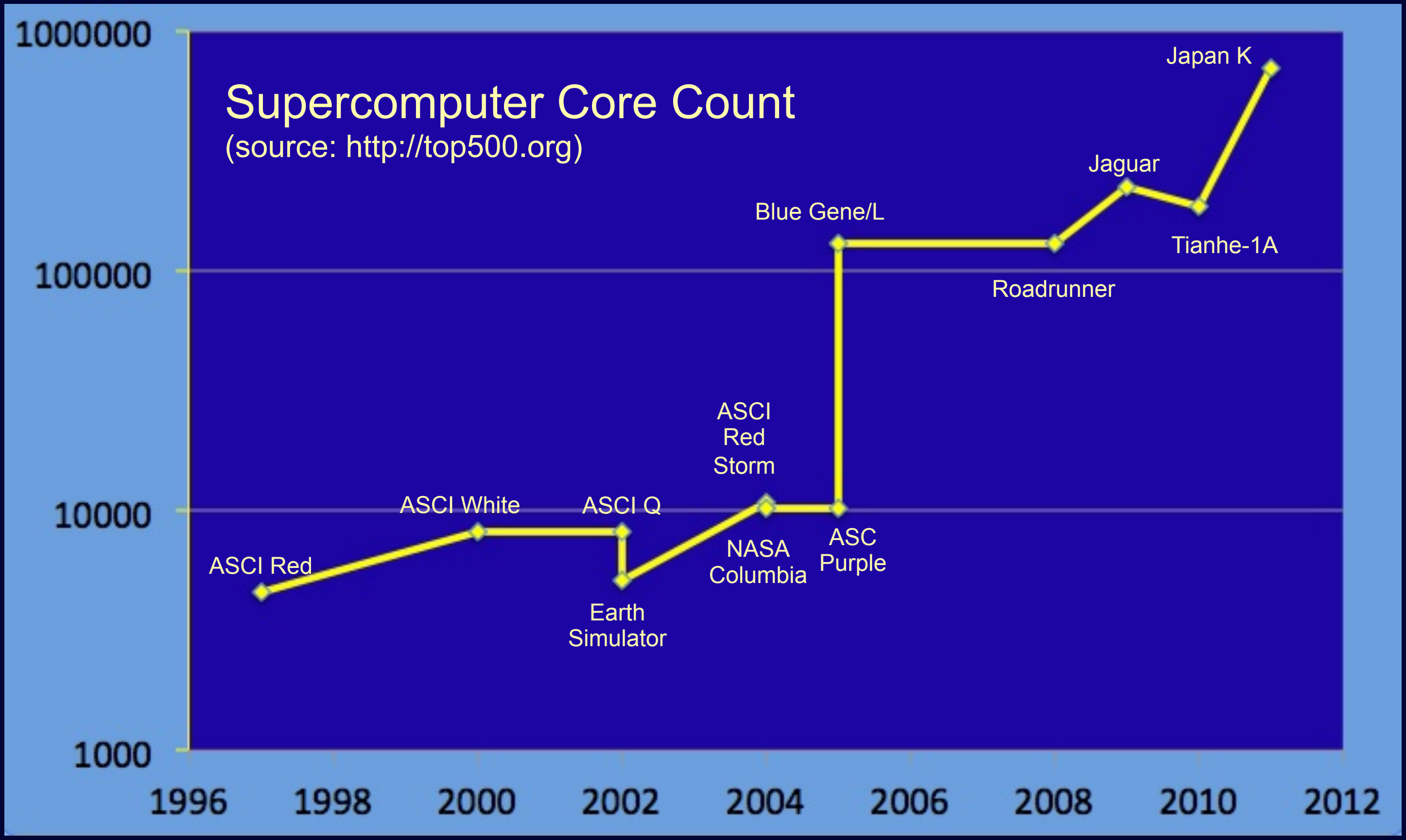
Parallel & distributed simulation strongly influenced by emerging technologies in the commercial sector

- Parallel discrete event simulation precipitated by increasing availability of commercial multiprocessors in late 1970's
- Distributed simulation field precipitated by commercial availability of local area networks (LANs) and inexpensive graphics hardware in 1980's

Emerging platforms:

- Already here: massive parallelism, GPGPU computing
- Now emerging: Cloud computing
- Coming: Ubiquitous computing, sensor networks

Emergence of Massive Parallelism



- Single processor clock speed improvements essentially stopped ~2005
- Massively parallel computers, GPGPU computing emerge

Not Just Supercomputers

- Parallel / distributed execution needed to exploit hardware performance improvements in *all* platforms
  - Supercomputers with millions of cores
  - Clusters and cloud computing
  - Multicore desktop and laptops
  - Handhelds and cell phones
- Challenge: will the mainstream M&S community be able to exploit these new, emerging architectures?

Cautionary Words from the Wise

Parallel computing is a problem, not a solution.

- Bill Joy

Parallel computing is the way of the future…

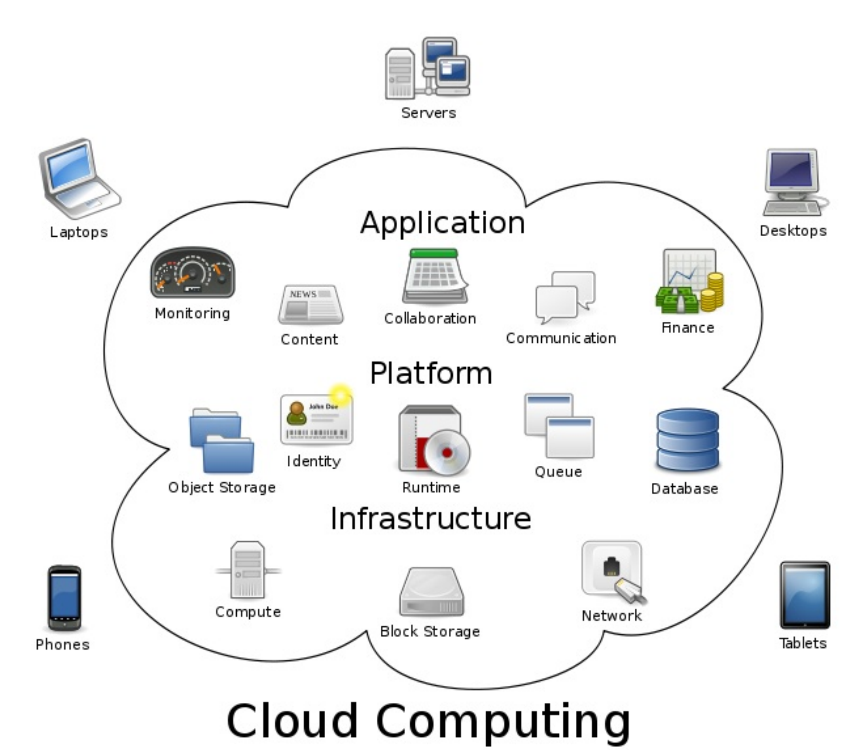
… and always will be!

- Cleve Moler

Parallel Discrete Event Simulation (PDES) in the Cloud

Cloud Computing

"… delivery of computing and storage capacity as a service to a heterogeneous community of end-recipients." [wikipedia]



http://en.wikipedia.org/wiki/Cloud_computing

- Infrastructure as a service (IaaS)
  - Virtualized processors, memory, storage, network
  - User provides OS, applications
  - Example: Amazon EC2
- Platform as a service (PaaS)
  - IaaS + OS, prog. lang., database
  - Example: Microsoft Azure
- Software as a service (SaaS)
  - PaaS + software applications
  - Example: Google Apps

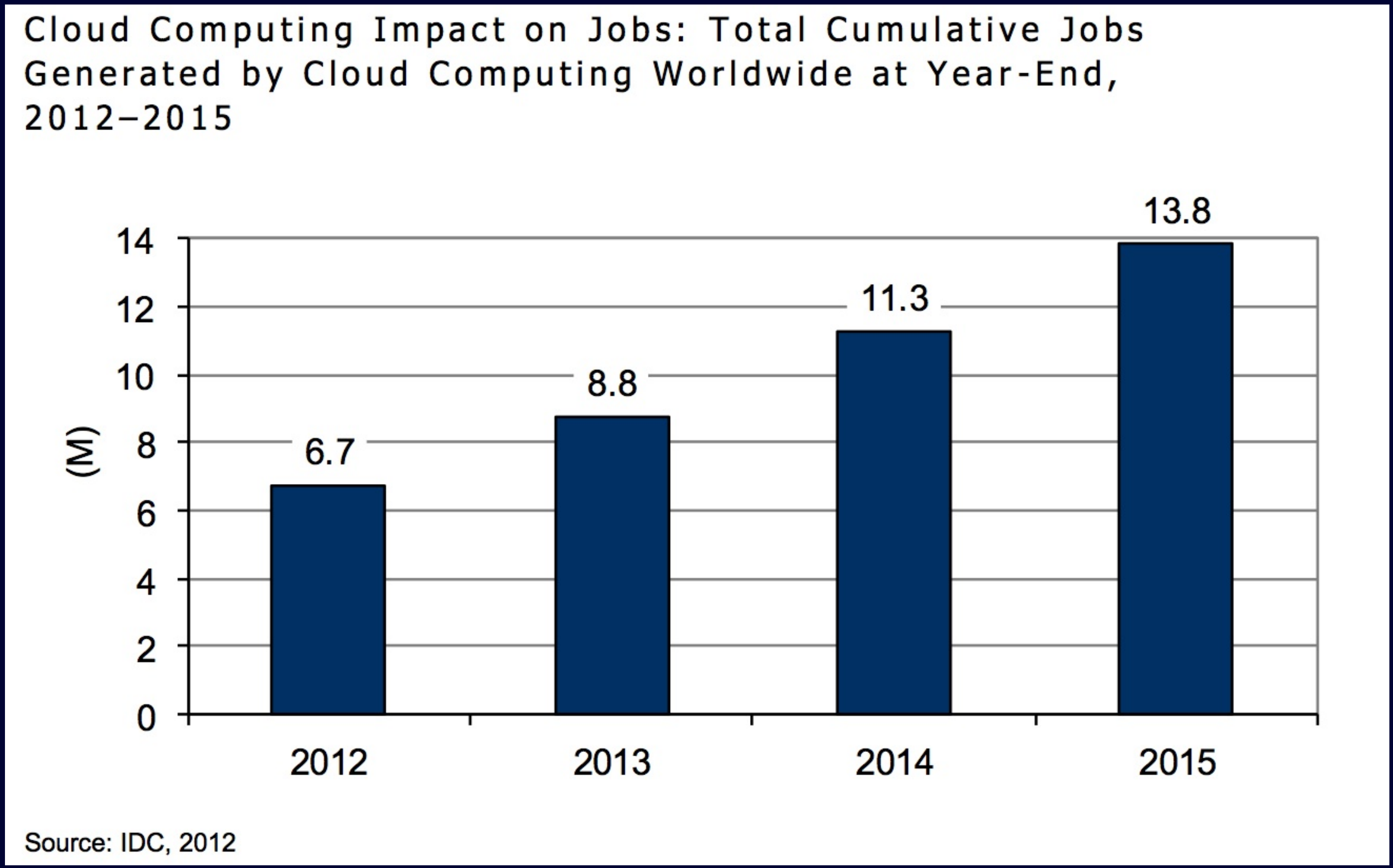
Cloud Computing Properties

- Cloud computing fundamentally a business model
- ***Out-source***: IT infrastructure management by a third party
- ***Remote access***: services accessed over a network (e.g., Internet) via web browser or light weight client
- ***Computing a utility***: "pay-as-you-go" cost model; advantageous if computing needs are bursty
- ***Elastic***: Computing resources can be rapidly expanded or contracted based on the user's needs
- ***Aggregation***: computing infrastructure shared by many users; yields economy of scale and (hopefully) lower operating cost
- ***Virtualized:*** uses virtualization to isolate users, allow customization of software environment

What’s New?

- Many of the ideas have been around for some time
  - Timeshare mainframe computers
  - Metacomputing
  - Grid computing
  - Autonomic computing
  - Utility computing …
- New elements
  - Widespread deployment in commercial sector
  - Ubiquitous access via the Internet
  - Opportunity for large impact

Cloud Computing is Important



- Important for economic growth
- Wherefore art the modeling & simulation industry?

PDES in the Cloud: Benefits

Cloud computing offers the opportunity to help make PDES broadly accessible to the M&S community

- "Pay as you go" model lowers the bar to gaining access to HPC platforms
- Continued, cost effective access to HPC resources especially for organizations with bursty needs for HPC
- Ability to customize software stack improves portability
- PDES as a Service: challenge lies in hiding the complexity of PDES codes and systems

Traditional PDES vs. Cloud

	PDES on HPC Platforms	PDES in Cloud Environment
Scalability	Important	Important
Resource sharing	Dedicated, static	Shared, dynamic
Dynamic load distribution	Sometime implemented	Mandatory
Fault-tolerance	Rarely implemented	Mandatory
Communication	Low delay, jitter	Large delay, jitter
Synchronization	Tightly coupled	Loosely coupled

Cloud designed for enterprise computing (not HPC); must rethink implementation approach for cloud environment

- In general, does not guarantee dedicated nodes
- Communications characteristics not favorable to HPC

Cloud Latency and Jitter [Zou et al. 2011]

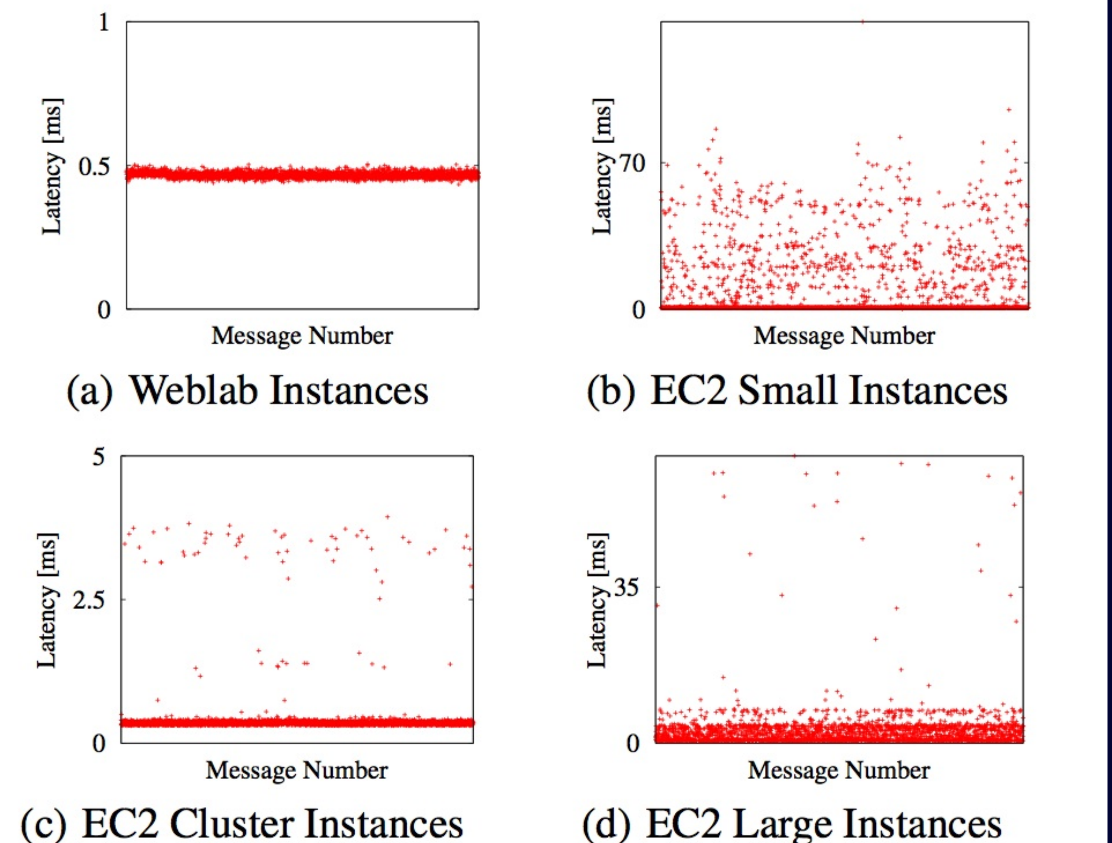


Figure 1: Latency in Our Weblab Cluster and in EC2

- Commodity hardware
- Nodes may be allocated in different subnetworks
- Interference among users

Cloud vs. dedicated cluster

- Significantly higher latency
- Much higher jitter (latency variance)

Zou, et al., “Making Time-Stepped Applications Tick in the Cloud,” ACM Symposium on Cloud Computing 2011.

Cloud Performance [Jackson et al., 2010]

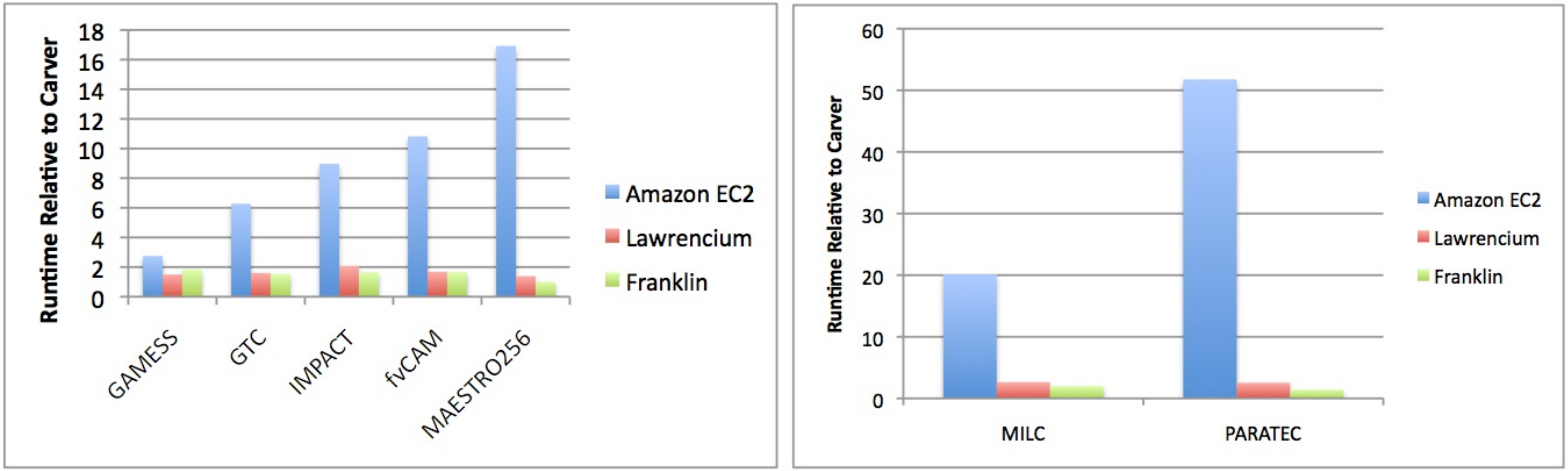


Figure 9.1: Runtime of each application on EC2, Lawrencium and Franklin relative to Carver.

Machine	Latency (us)	Bandwidth (GB/s)
Carver (IBM iDataPlex cluster)	2.1	3.4
Franklin (Cray XT4)	7.8	1.6
Lawrencium (Linux cluster)	4.1	1.2
Amazon EC2	145	0.06

Jackson, et al. Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. In 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010.

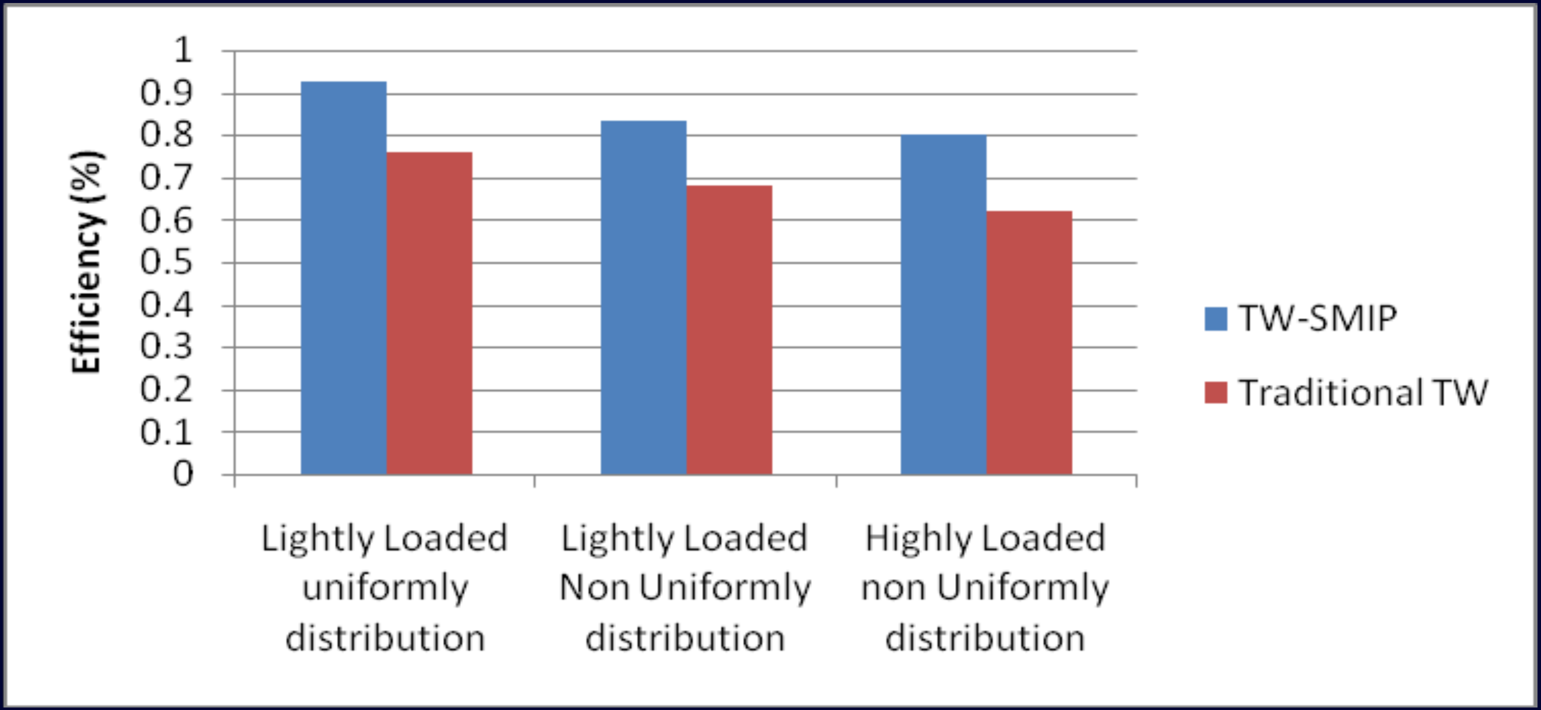
PDES Synchronization

- Conservative: Synchronous protocols
  - Examples: Bounded lag, YAWNS
  - Jitter a challenge
  - Problematic on non-dedicated processors
- Conservative: Asynchronous
  - Examples: Chandy/Misra/Bryant
  - Communication overheads a challenge
- Optimistic
  - Examples: Time Warp and variants on TW
  - With appropriate scheduling, tends to push communication costs (for synchronization) off critical path of computation
  - Metered use means a cost for rolled back computation

Time Warp in the Cloud

- Time Warp overheads arise largely from state saving and rollbacks
- Background computations can lead to an excessive amount of rollback
  - Synchronization algorithm must adapt as the underlying execution environment changes
- Unpredictable, and sometimes long communication delays can increase number of straggler messages
- Solutions must preserve inherent advantages of optimistic execution

TW-Straggler Message Identification Protocol



- Add heart beat (HB) messages to Time Warp
- Sent at fixed time intervals to check for straggler messages
- Prevent LPs from progressing too far ahead in simulation time relative to other LPs

Malik, Park, Fujimoto, "Optimistic Synchronization of Parallel Simulations in Cloud Computing Environments," *IEEE International Conference on Cloud Computing*, September 2009.

The Aurora System

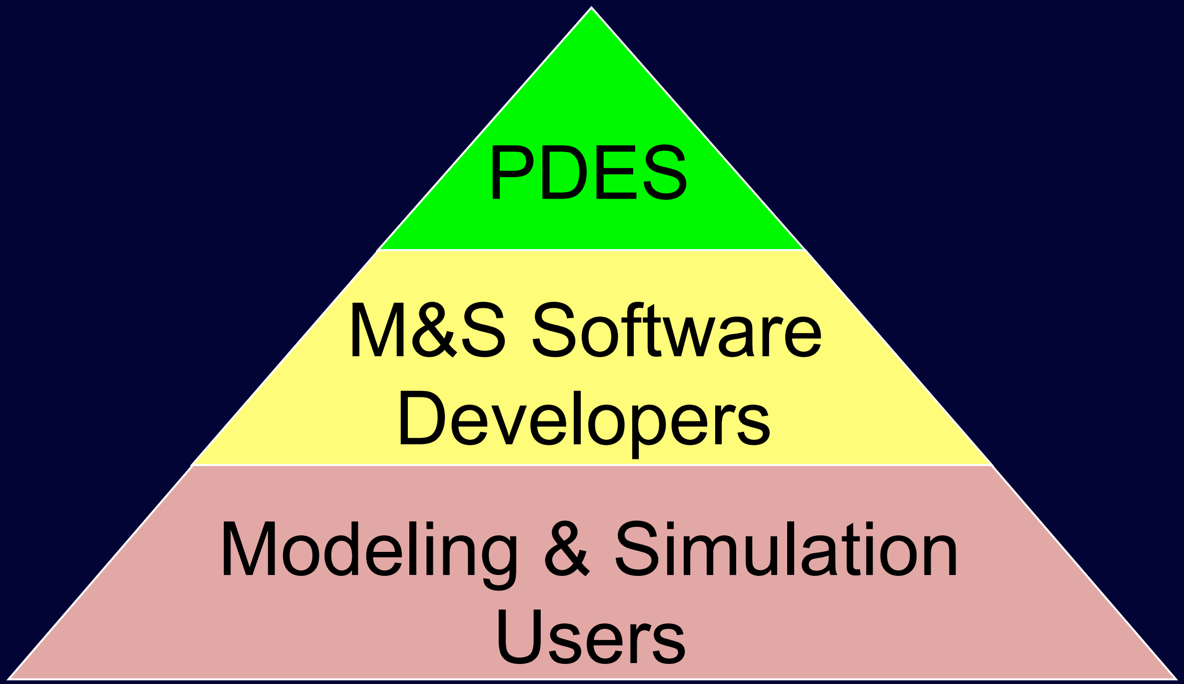
- Execution on shared computing resources using a master/worker paradigm
  - Platforms ranging from supercomputers, to clusters, to desktop machines connected to the Internet
- Scalable parallel and distributed execution
  - Large-scale PDES codes
  - Many replicated runs
  - Conservative and optimistic synchronization
- Fault-tolerant execution
  - Client/Server approach and redundant execution
  - Transparent server and client failure recovery
- Download and run anywhere
  - Open, industry standards
  - Web Services

PDES in the Cloud: Research Issues

Cloud can help make PDES broadly accessible, however many challenges remain

- Overcoming high communication latency, jitter
- Automated model partitioning
- Automatic mapping of logical processes to virtualized processors
- Automatic dynamic load balancing
- Transparent fault tolerance
- Efficient execution over shared, dynamic computing platforms
- Scalable parallel and distributed execution
  - Large-scale PDES codes as well as replicated runs

About PDES Education



PDES can have broad impact through education

- Bulk of M&S community users; need not have PDES expertise
- Modelers can be expected to be sophisticated developers
- PDES can achieve broad impact by educating developers, providing tools to assist creation of PDES codes

Cloud Research Other than Parallel & Distributed Simulation

- Cloud platform well suited for replicated simulation runs, parametric studies
  - Embarrassingly parallel
  - Communication requirements less stringent
- Time parallel simulation
  - Segment simulation time access; separate process works on a different time segment
  - Mechanism weave together different time segments
  - Works for a few specific application domains
- Both areas well suited for parallel computation in general

Embedded Distributed Simulation

Embedded Computing

- Embedded systems market today is more than $1 trillion; to double from 2011 to 2015*
- New systems projected to utilize 14.5 billion microprocessor cores by 2015
- Today, shipments exceed that of PCs, servers, and mobile phones combined
- Cloud enables new opportunities for embedded systems

Embedded computing a huge potential growth area for simulation (and distributed simulation)

* IDC, Worldwide Intelligent Systems 2011-2015 Forecast: The Next Big Opportunity, September 2011

On-Line Distributed Simulation

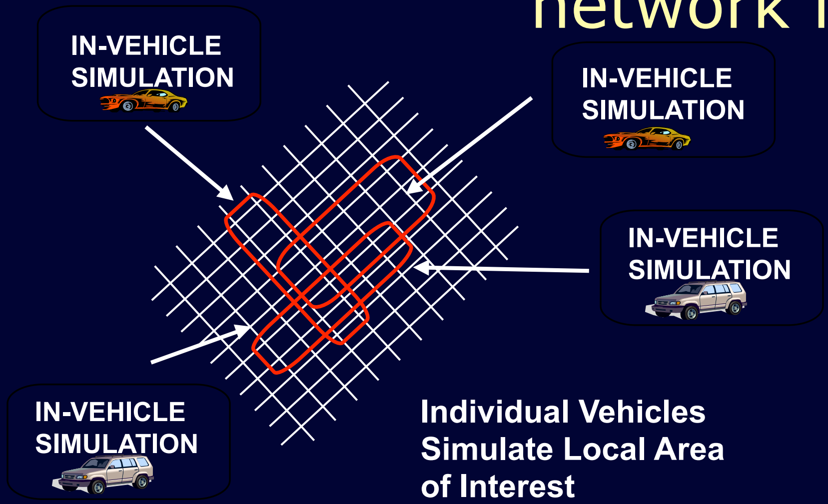
- On-line simulation (Symbiotic Simulation, Dynamic Data-Driven Application Systems, Cyber-physical systems)
  - Collect sensor data from environment
  - Construct current system state from sensed data
  - Compute future states via simulation
  - Optimize system to steer toward desired states
- Example applications
  - Manufacturing, Business Processes (NTU)
  - Telecommunications (UCLA, GT, UCB)
  - Preparation for Inclement Weather (Univ. of Oklahoma, Indiana, …)
  - Crisis Management (Purdue, …)
  - Defense (SAIC, …)

What’s New?

- On-line simulation has been used in a variety of applications
- New elements
  - Sensor networks, especially large-scale sensor networks
  - Ubiquitous computing and communications now widely deployed; can one effectively move simulations into these devices?
  - Cloud computing
- On-line simulation in these new context, to manage large-scale real world systems has not been well studied

Ad Hoc Distributed Simulations

Can we embed a distributed simulation into the sensor network itself?



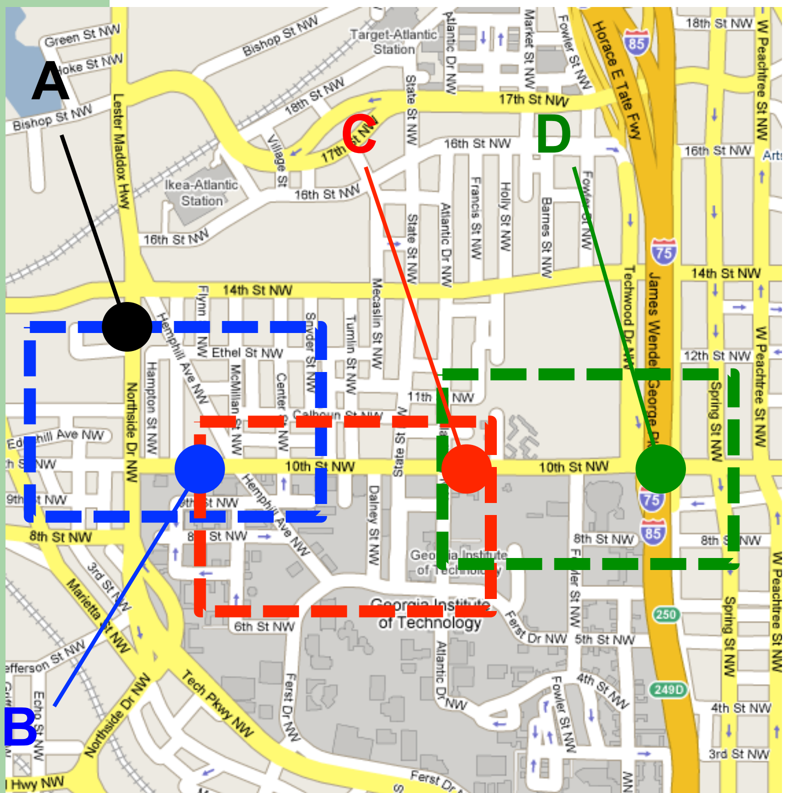
State prediction of each simulator must adapt to

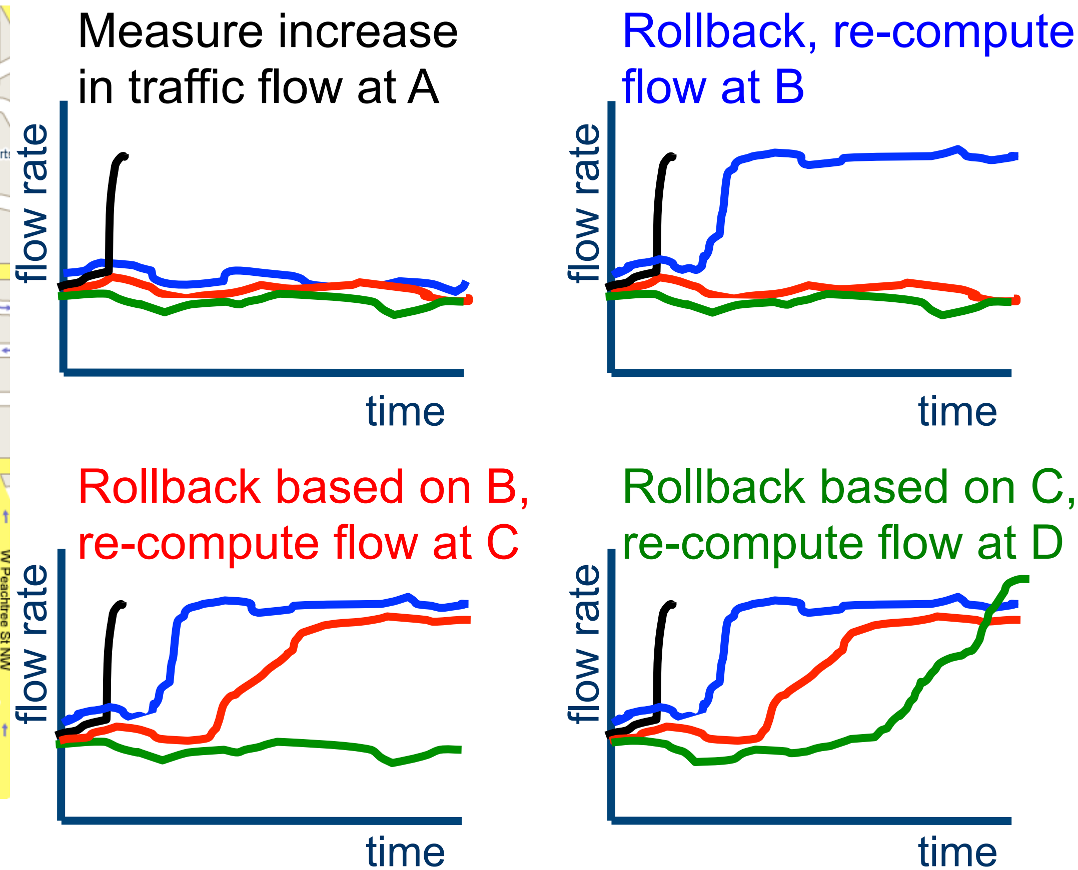
- Live sensor data indicating current system state
- Predicted state provided by other simulators

- An ad hoc distributed simulation is a collection of autonomous, self-configuring, online simulators connected by a network, where each simulator models a portion of the physical system and the union of the simulators predicts future states of the overall system
  - Area modeled by each simulator may vary over time
  - Areas modeled by different simulators may overlap; some areas may not be modeled at all
- Constructed “bottom-up,” in contrast to conventional distributed simulations constructed “top-down” by partitioning overall system
- Hybrid of replication and conventional distributed simulations
- Rollback mechanism to adapt to new events in the physical system

Georgia Institute of Technology

Automated Update via Optimistic Synchronization





Roll back simulator when

- Prediction and measurement disagree
- Predictions from other simulators change

Relationship to Other Simulation Approaches

- Conventional distributed simulations
  - In ad hoc simulations, multiple simulators compute the value of state variables
  - Synchronization algorithm needed to coordinate simulators, but based on aggregated state estimates rather than “correct” values of system state
- Replicated Trials
  - Replications model subsets of the entire physical system
  - Replicated simulations interact via the synchronization algorithm
  - Outlier simulations are not rolled back


7
9
5
1
10
6
3
2
8
4

Field Camera View

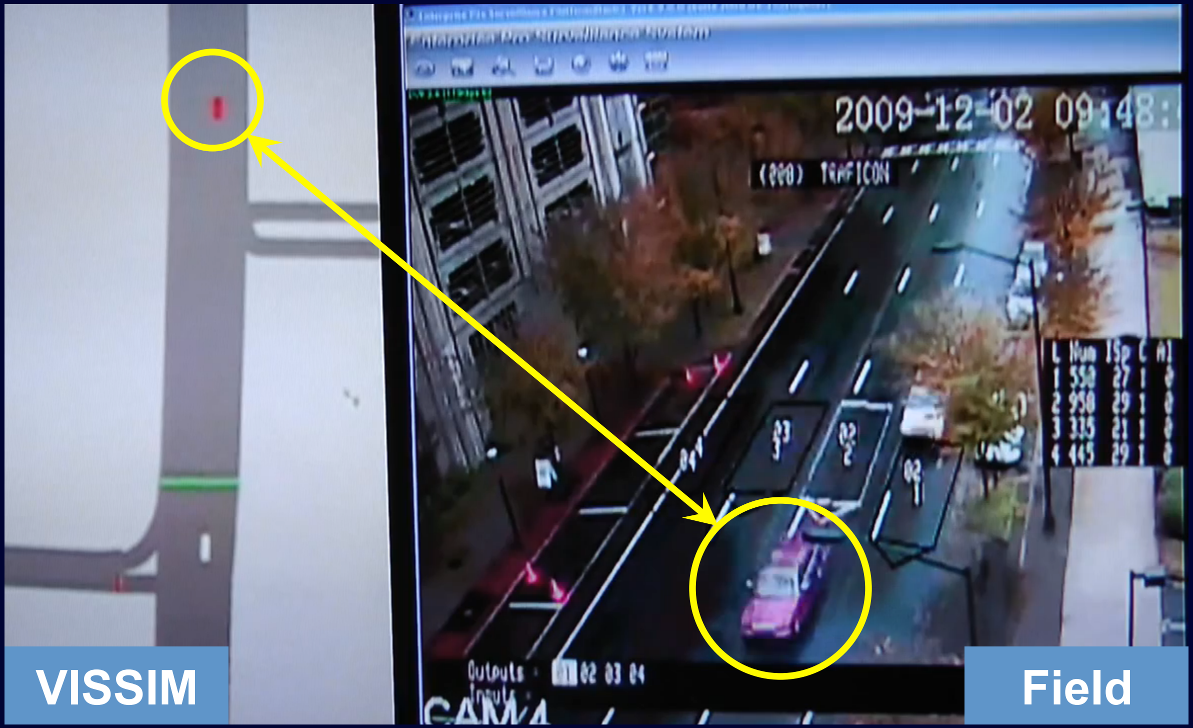
W. Peachtree St. and 5th Street

Regular View

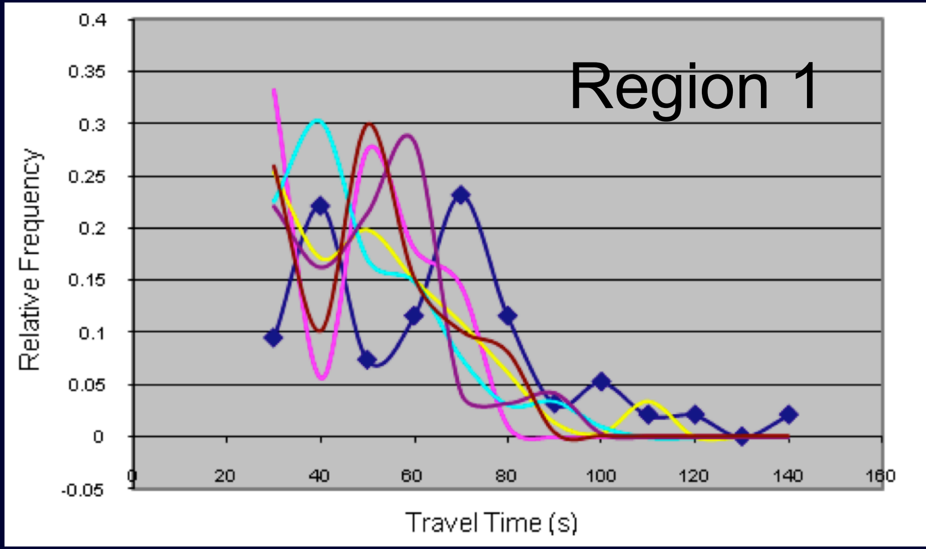
Data Acquisition View

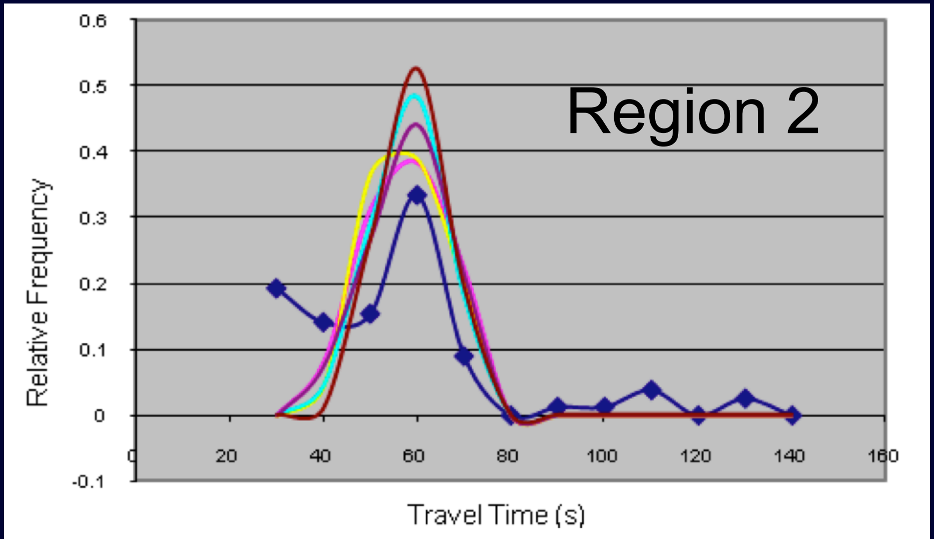
Data-Driven Simulation

Using Video Stream to Drive Simulation (VISSIM)



Initial Travel Time Predications

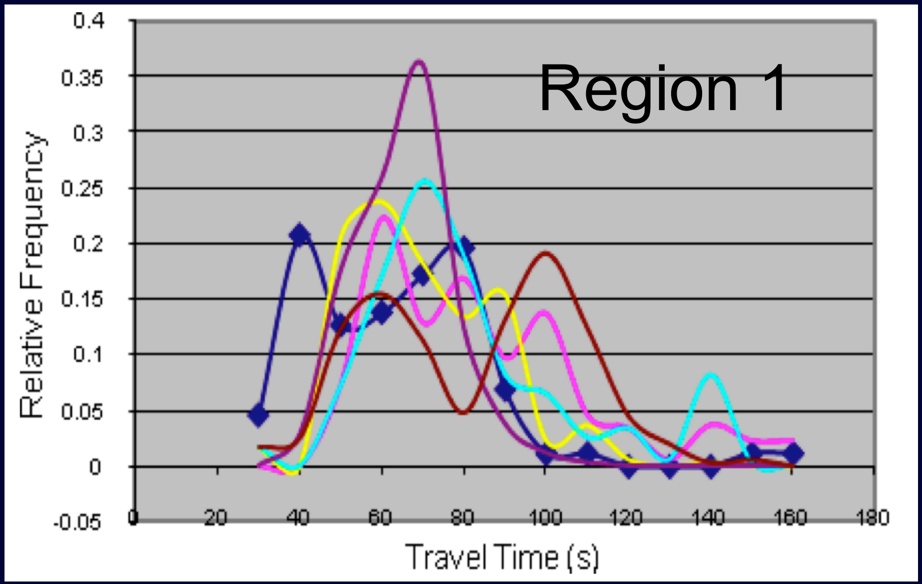


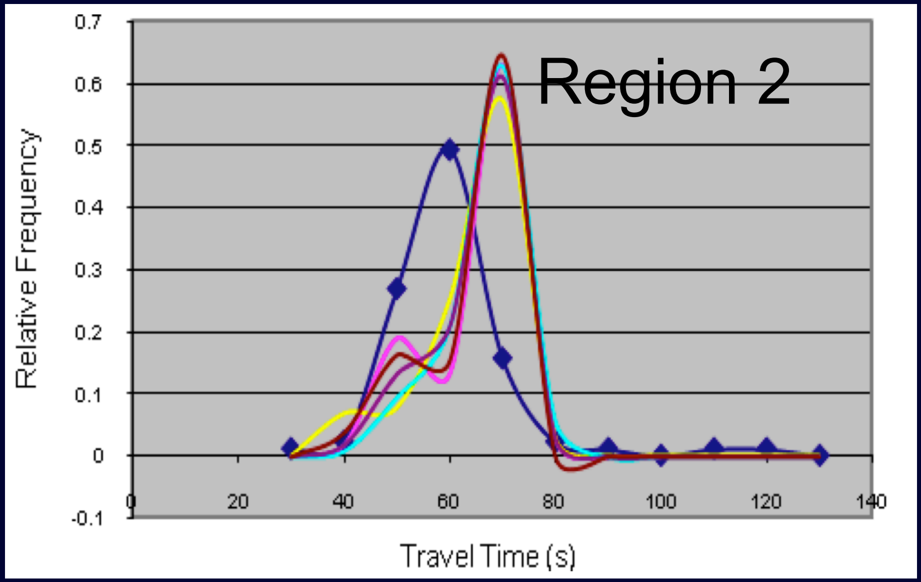


- 6:30-7:30 PM
- Five client simulations
- Region 1
  - Relative good prediction of mean
  - Varying success in capturing bimodal distribution
  - Lack of synchronized signals in model
  - Lack of buses in model
- Region 2
  - Good agreement in absence of signal synchronization, buses

Baker, et al., “An Online Transportation System Simulation Testbed,” *Winter Simulation Conference*, Dec 2009.

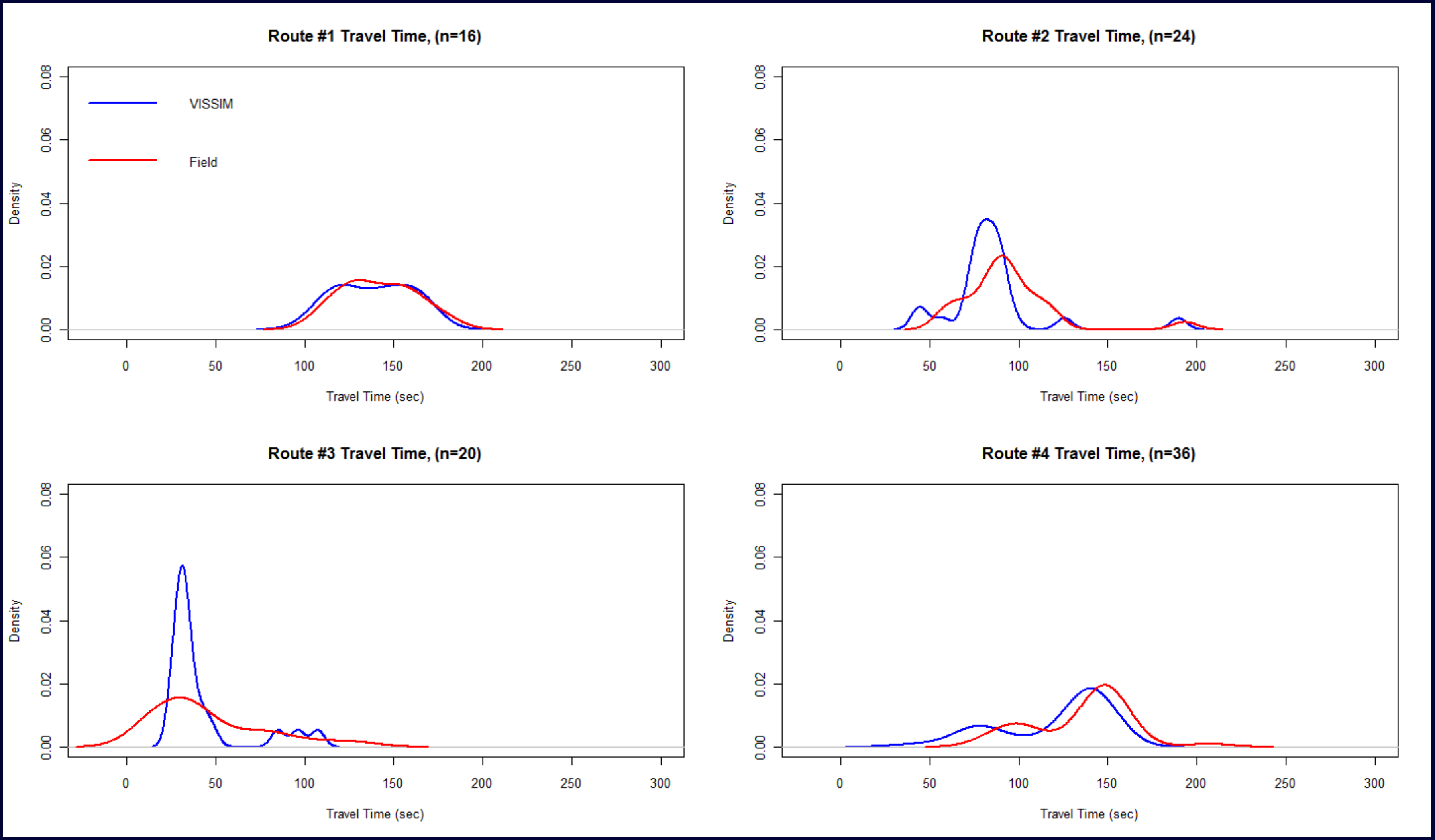
Travel Time Predictions





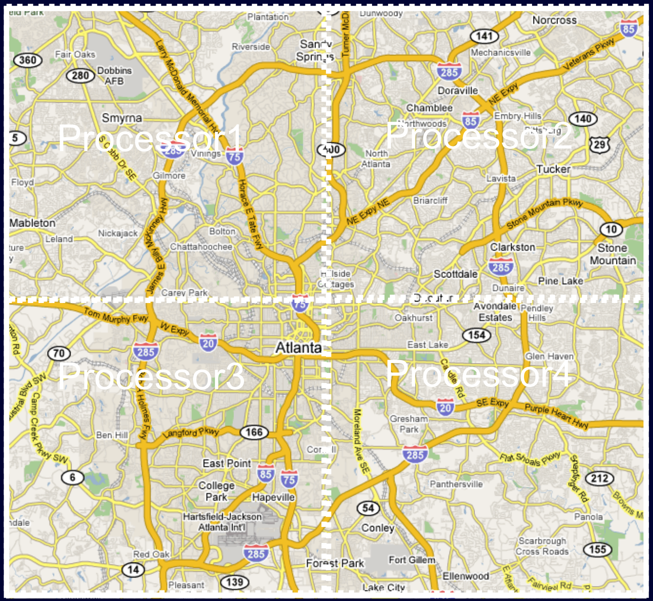
- 4:30-5:30 PM
- Heavier traffic than 6:30-7:30 case
- Simulations tend to overestimate travel time

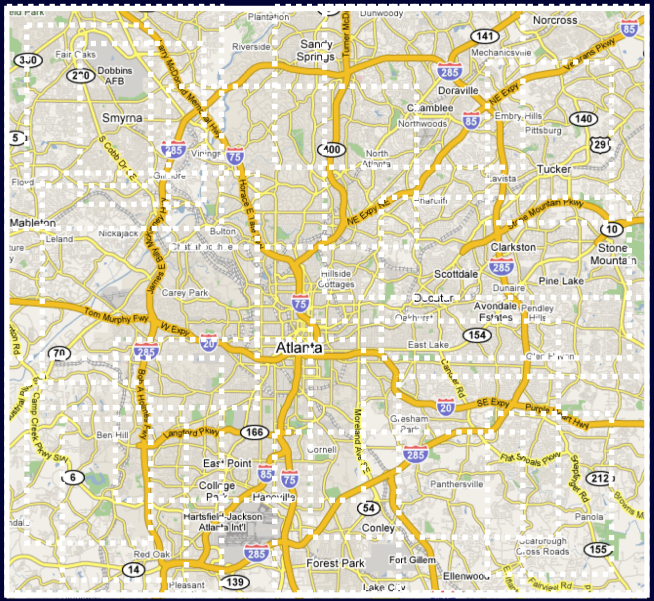
Travel Time: On-Line Simulation vs. Field Experiments



Henclewood, Real-Time Estimation of Arterial Performance Measures Using a Data-Driven Microscopic Traffic Simulation Technique, PhD Dissertation, 2012.

Conventional vs. Ad Hoc Distributed Simulation





Conventional

- Top-Down construction
- Clean partition of state space; static partition
- Produce same results as a single run

Ad Hoc

- Bottom-Up construction
- Ad Hoc partition of state space; dynamic partition
- Produce same statistical results as replicated runs

State Prediction Questions

State prediction problems:

- Can a collection of autonomous, localized simulations provide accurate predictions of the overall system state?
- Static prediction: Given a current snapshot of the state of the system, what is the predicted, future state?
- Dynamic predication: Given a *new* snapshot of the state of the system, what is the (revised) prediction of future system state?

Research Directions

automate

“Just one word … ~~plastics~~”

Automating the M&S Life Cycle
(Human-out-of-the-loop Simulation)

- Data collection, input analysis
- Model configuration, instantiation
- Optimization strategy
- Experiment design
- Model execution and management of runs
- On-line model calibration/validation
- Output analysis
- Derive and implement recommendations

Research Challenges (cont.)

- Large-scale distributed simulation on mobile computing platforms
  - Mobile computing platforms present new challenges for distributed simulation
  - Unreliable communications, high latency
- Interoperable simulations and tools
  - Simulations from different manufacturers must be composeable
  - Continuously operating distributed simulations call for integrating new simulations into running systems

Power Aware Distributed Simulation

- Power has become a first class citizen as a metric in computing
  - Obviously important in embedded environments
  - Important for all simulations, especially high end simulations
- Research Questions
  - What distributed simulation algorithms and techniques minimize power consumption (e.g., reduced communication)?
  - Can lower precision arithmetic be exploited to reduce power consumption?

Interoperability
An Analogy with Personal Computers

PC

Mac

Open architecture, standards

- Flexible, options to users
- Complex
- Operationally difficult
- M&S: HLA, other standards

Tightly integrated design

- Limited choice
- Simple
- Easier to manage
- M&S: single company models

Is there a middle ground?

Concluding Remarks

- These are exciting times to work in parallel and distributed simulation
  - There is need for the technology
  - There are opportunities for impact
- Cloud computing and on-line simulation represent areas of growing importance and opportunity, but many challenging research problems remain

The success (or failure) of the M&S community to embrace parallel and distributed computing will impact the ability of the field to exploit emerging architectures and platforms

Questions?