Parallel Discrete Event Simulation: Past, Present, and Future

Richard Fujimoto

School of Computational Science & Engineering

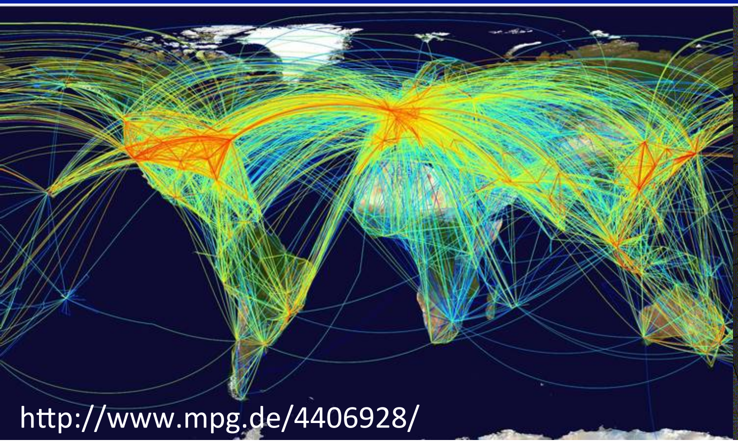
Georgia Institute of Technology

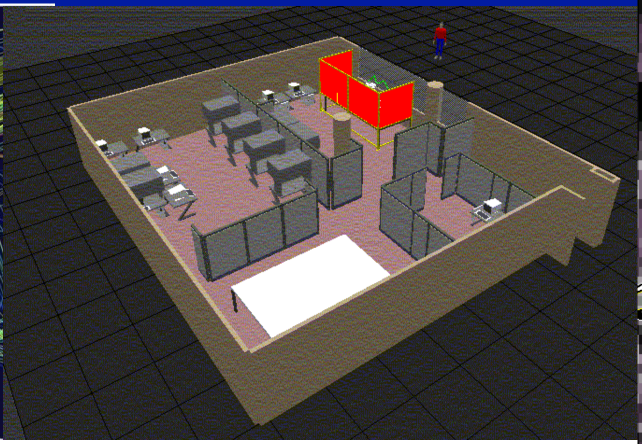
SIMULTECH 2015
July 22, 2015

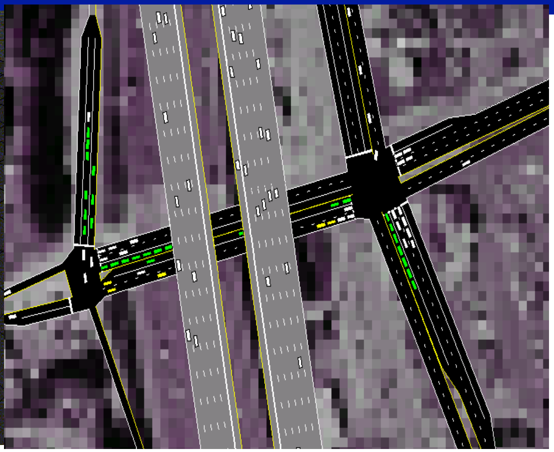
Outline

- The Early Years
- The Field Matures
- Parallel Discrete Event Simulation (PDES) Today
- Toward an M&S Research Agenda

Discrete Event Simulation Applications



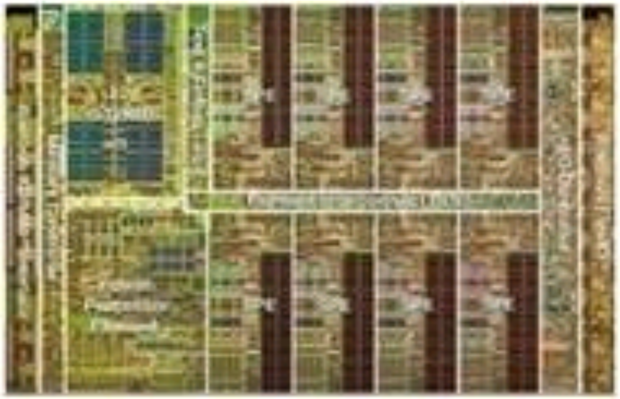




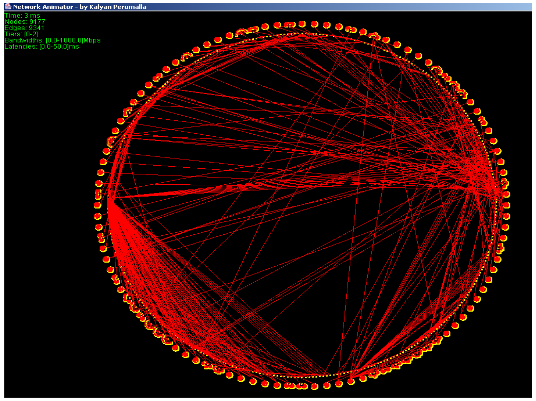
Epidemics

Manufacturing and supply chains

Transportation

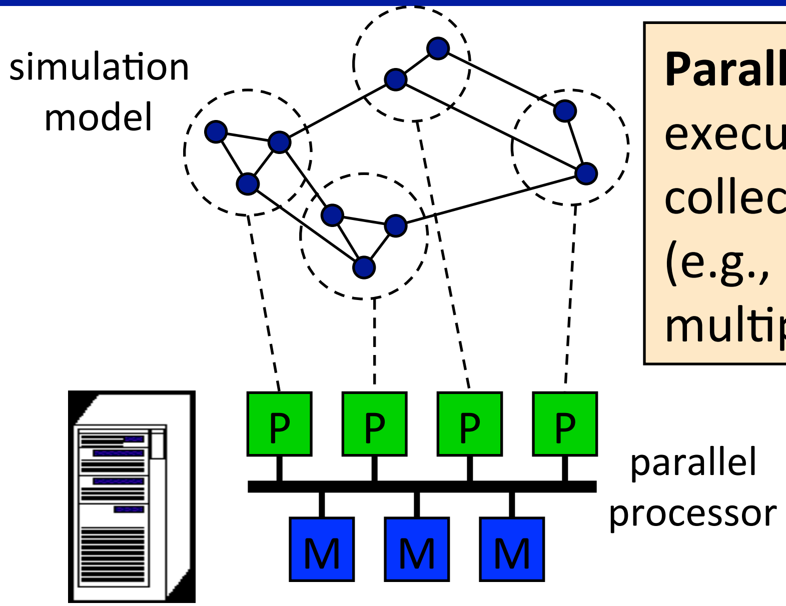


High Performance computer architectures

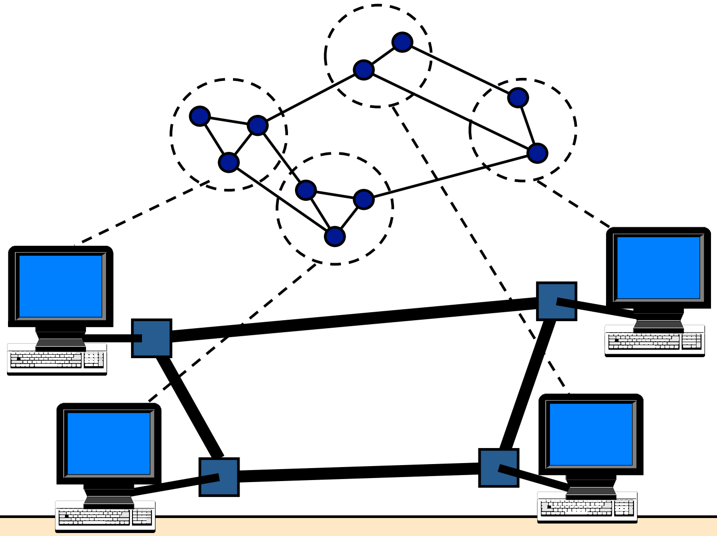


Communication networks, Cybersecurity

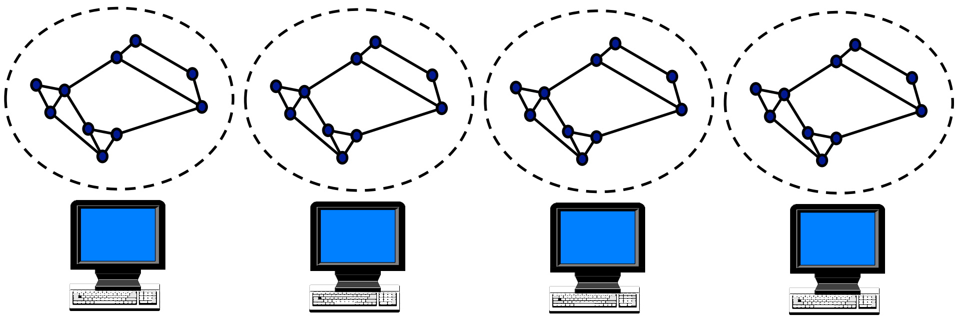
Parallel and Distributed Simulation



Parallel simulation involves the execution of a *single* simulation on a collection of *tightly* coupled processors (e.g., a shared memory multiprocessor).



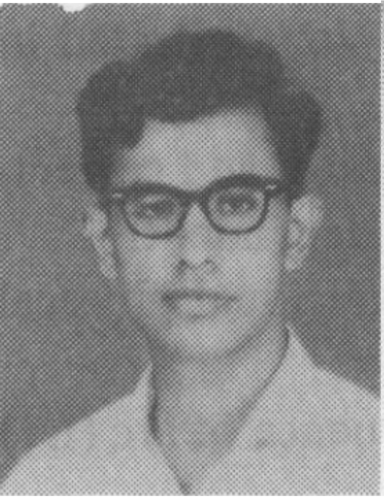
Distributed simulation involves the execution of a *single* simulation on a collection of *loosely* coupled processors (e.g., PCs interconnected by a LAN or WAN).



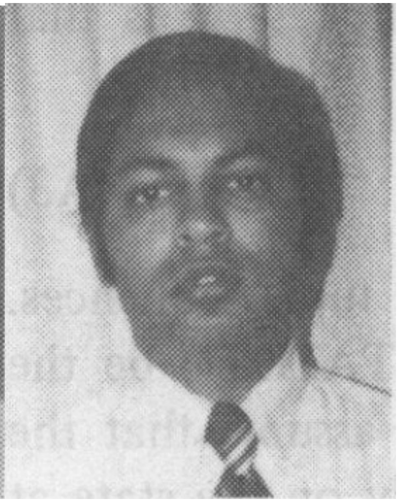
Replicated trials involves the execution of *several*, independent simulation runs concurrently on different processors

Beginnings: Chandy, Misra and Bryant (1977)

How can one execute a discrete event simulation in parallel, but obtain the same result as a sequential execution?



K. Mani Chandy
(circa 1979)



Jayadev Misra
(circa 1979)

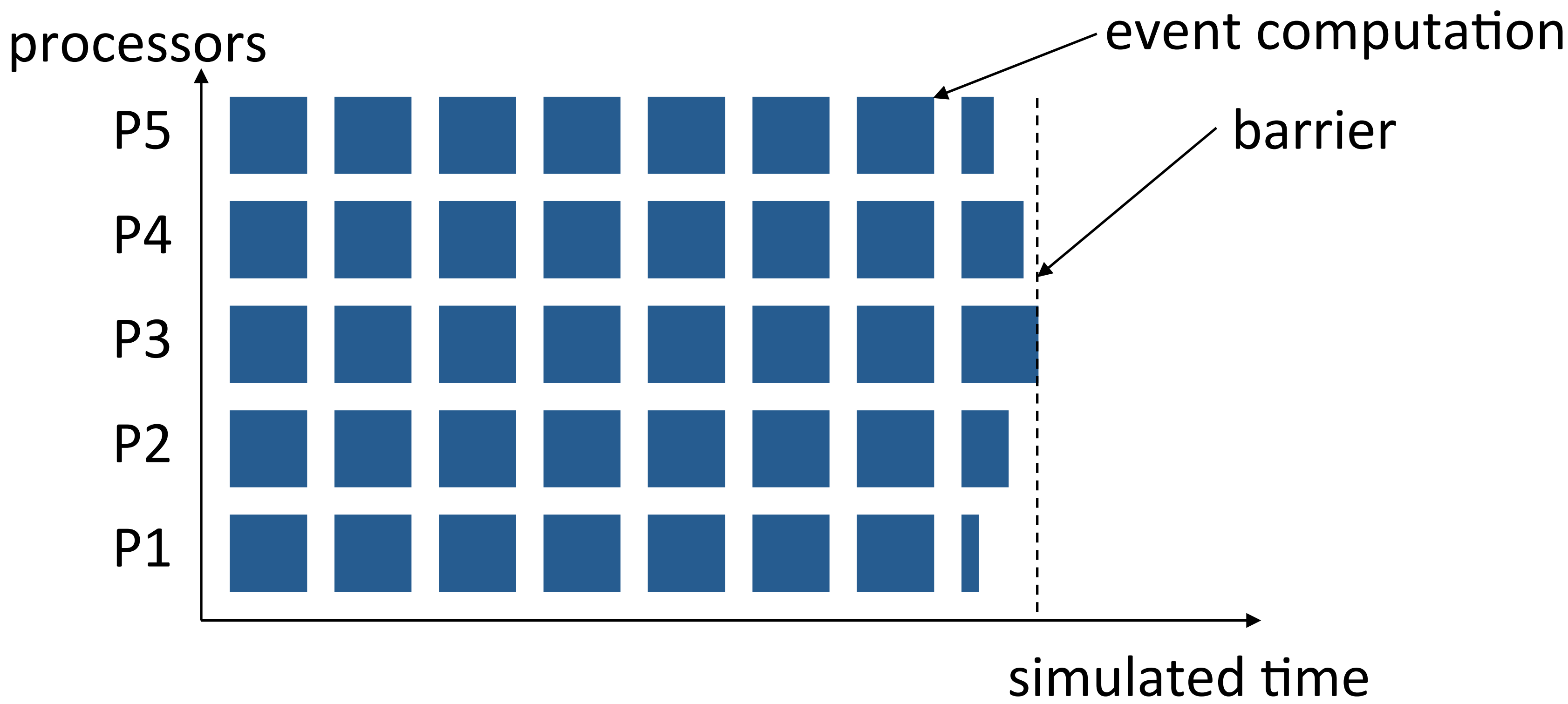
Chandy, K. M. and J. Misra (1979). "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs," IEEE Transactions on Software Engineering **SE-5**(5): 440-452.

Randal E. Bryant, “Simulation of Packet Communication Architecture Computer Systems,” M.S. Thesis, MIT, 1977.

Randy Bryant
(circa 1980)

Parallel Time Stepped Simulation

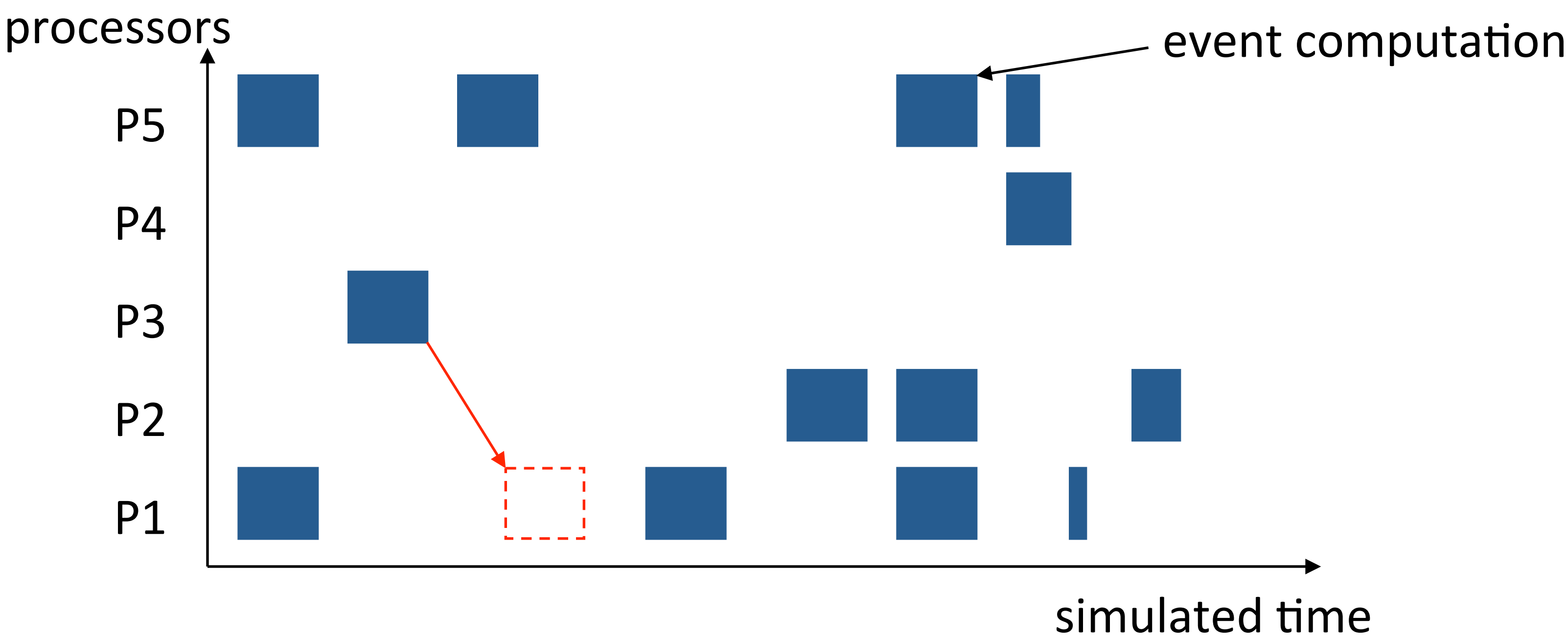
The simulator computes the state of the system across simulated time



Parallel time-stepped simulation: all processors complete the simulation for the current time step before advancing to the next time step.

PDES: The Synchronization Problem

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



- Allow processes to advance ahead of others
- A simulation on one processor might affect the past of another
- Chandy/Misra/Bryant solved this problem by blocking processes until one could guarantee it would not receive a message in its past.

Time Warp (1982)

An entirely different approach:

Synchronize the computation using a rollback mechanism

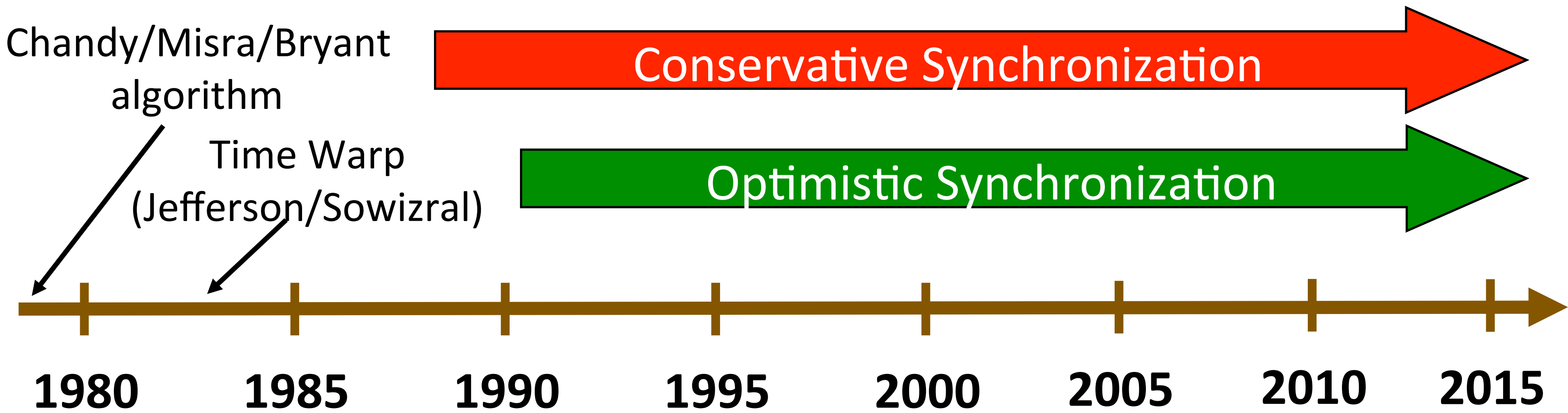
David Jefferson

David Jefferson and Henry Sowizral

"Fast Concurrent Simulation Using the Time Warp Mechanism" RAND Note N-1906-AF, December 1982.

Henry Sowizral

Synchronization Algorithms



- Conservative synchronization: block processes to ensure events are always processed in timestamp order
- Optimistic synchronization: allow out-of-order event processing, but detect them at runtime and recover using a rollback mechanism

PDES in the 1980's and early 1990's

- Dominant Question: Which was better – conservative vs. optimistic?
  - No definitive answer
  - The real answer is it depends…
- Nevertheless, this competition helped drive many new developments in the field

Meanwhile, in the Defense Community …

Conservative

Optimistic

Distributed Simulation

HLA (IEEE 1516)

DIS (IEEE 1278)

TENA

DARPA

SIMulator NETworking (SIMNET)

Distributed Interactive Simulation (DIS)

HLA

High Level Architecture (HLA)

1980 1985 1990 1995 2000 2005 2010 2015

Challenge: Simulation Interoperability

Parallel Discrete Event Simulation (circa 1993)

R. M. Fujimoto, "Parallel Discrete Event Simulation: Will the Field Survive?," *ORSA Journal on Computing,* vol. 5, No. 3, pp. 213-230, 1993. Commentaries by Marc Abrams, Rajive Bagrodia, Yi-Bing Lin, Paul Reynolds Jr., Brian Unger & John Cleary

PDES not widely adopted by the mainstream M&S community

Proposed approaches to achieve broader adoption included:

- Application specific PDES libraries
- New PDES programming languages
- Support for shared state
- Automated parallelization of sequential simulations

Parallel Discrete Event Simulation (circa 1993)

R. M. Fujimoto, "Parallel Discrete Event Simulation: Will the Field Survive?," *ORSA Journal on Computing,* vol. 5, No. 3, pp. 213-230, 1993. Commentaries by Marc Abrams, Rajive Bagrodia, Yi-Bing Lin, Paul Reynolds Jr., Brian Unger & John Cleary

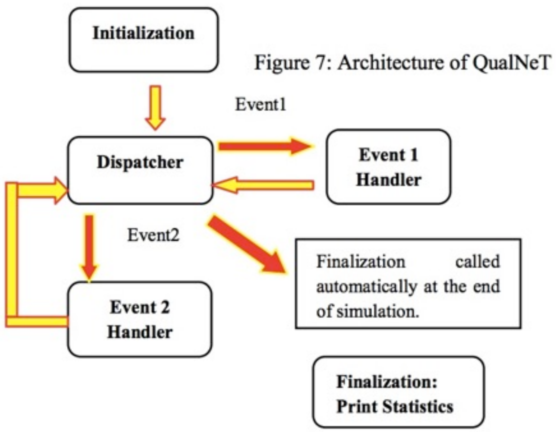
PDES not widely adopted by the mainstream M&S community

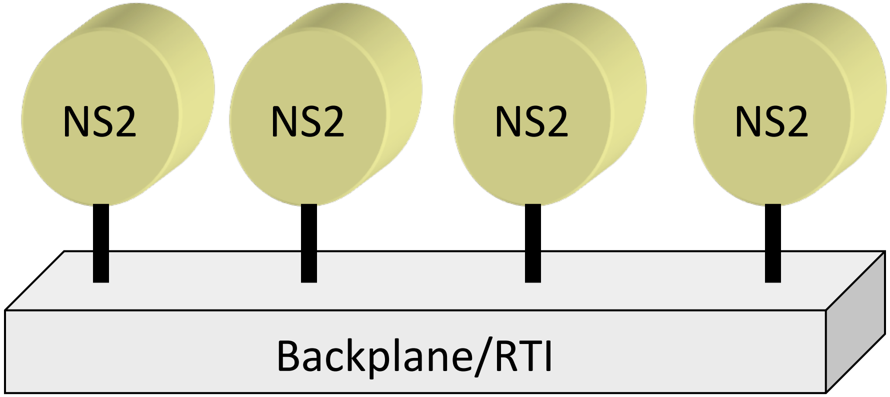
Proposed approaches to achieve broader adoption included:

- Application specific PDES libraries
- New PDES programming languages
- Support for shared state
- Automated parallelization of sequential simulations

Approaches to Parallel Discrete Event Simulation

Siraj, Gupta, Rinku-Badgujar, "Network Simulation Tools Survey," Intl. Journal of Adv. Research in Computer and Comm. Eng., Vol 1, No 4, June 2012





PDES library approach:

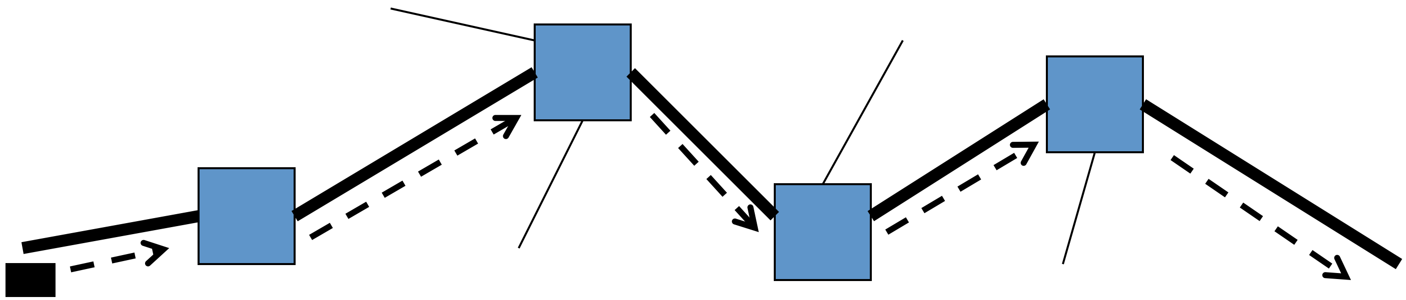
- New models built "from scratch"
- Must develop, validate models
- Optimized for parallel execution
- Users must learn a new simulator
- SSFNet, TeD, Qualnet, ROSS, Javasim, Warped, TeleSim…

Federated simulation approach:

- Simulators integrated via a software backplane/RTI
- Exploit existing software & validated model & user base
- Heterogenous simulations
- UPS (queueing nets), PDNS, GTNets, Genesis

Case Study: Communication Networks

Packet-Level Simulation of Large-Scale Wired Networks

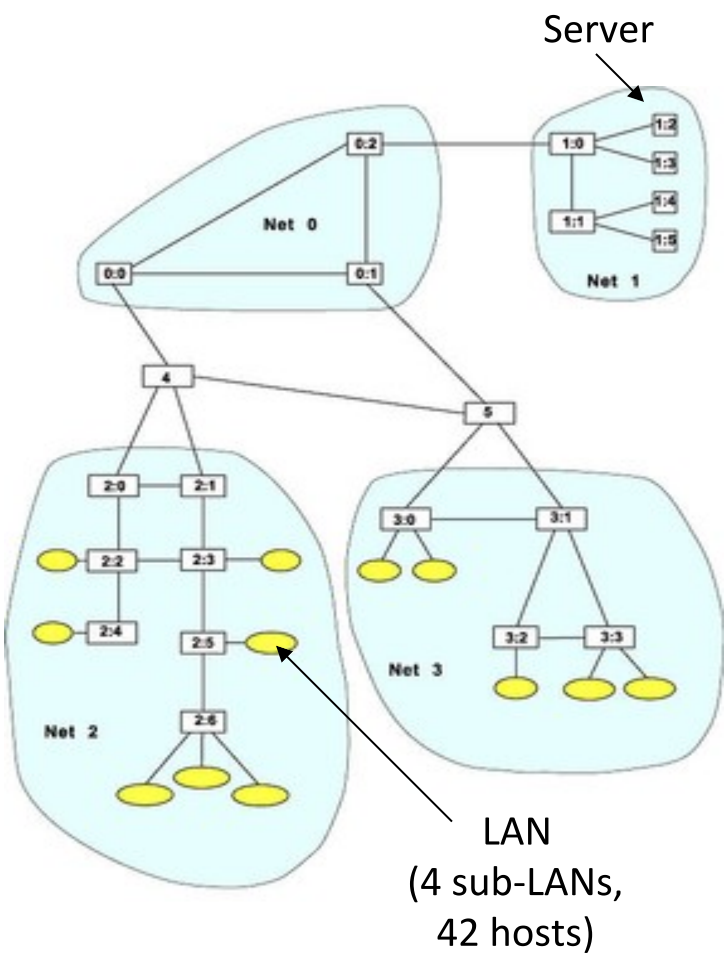


- One can characterize a simulation workload by the number of packet transmissions that must be simulated
  - Bulk of the computation involves simulating packets moving hop by hop through the network (queueing, transmitting over link, etc.)
  - Typically, two simulator events per "packet hop"
  - Define a packet transmission as sending one packet over a single communication link
- Simulator performance: the number of simulated packet transmissions per second (PTS) of wallclock time

Fujimoto, Perumalla, Park, Wu, Ammar, Riley, "Large-Scale Network Simulation: How Big? How Fast?" *Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 116-123, October 2003.

Benchmark: Campus Network "Building Block"

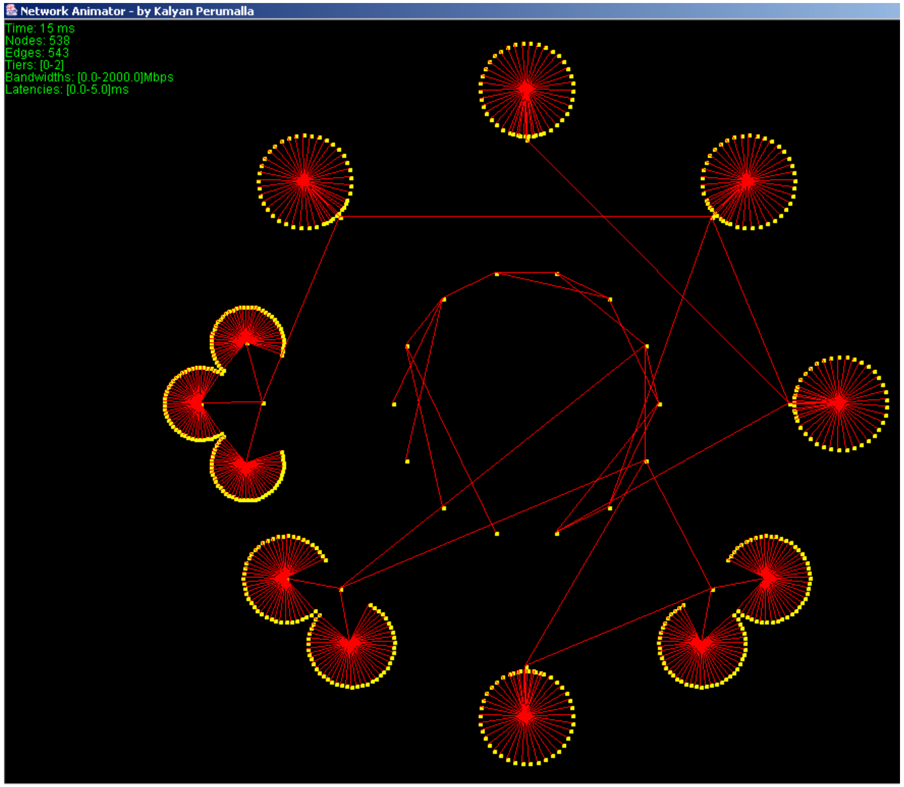
- Benchmark developed at Dartmouth (Nicol, et al.)
- Building Block: Campus Network Model
  - 538 nodes
  - 504 clients
- Multiple Campus Networks (CNs) connected to form a ring
  - Up to 10,000 campus networks (~5 Million nodes)
  - Links up to 2Gb/s
  - Link delay ranging from 1ms to 200ms
- Additional chord links

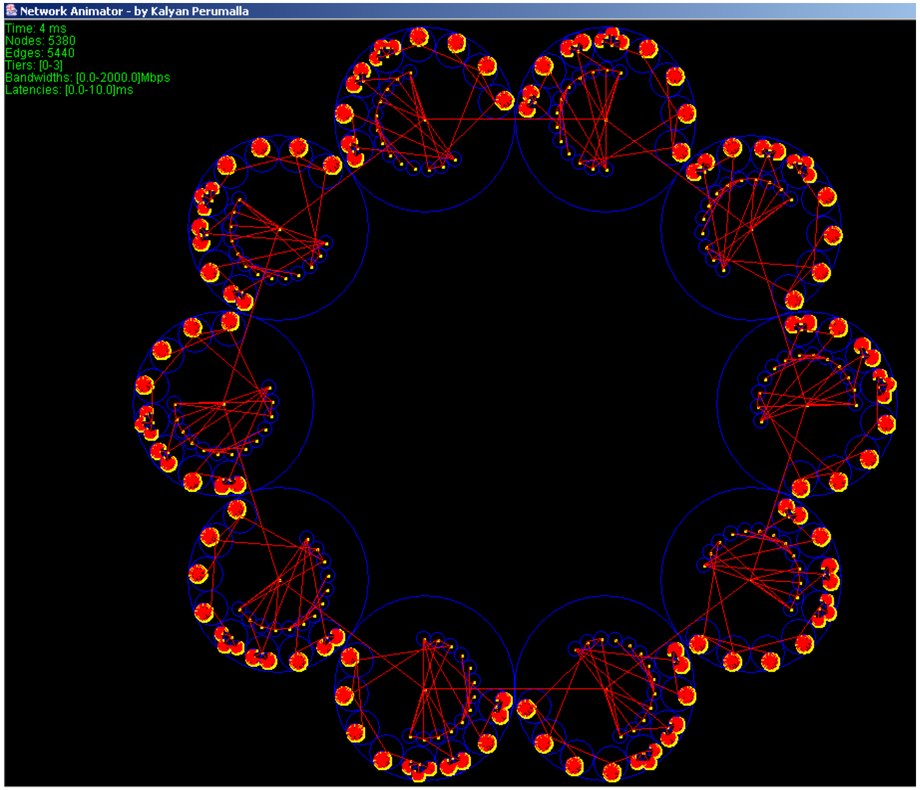


Single Campus Network

Figure courtesy of David Nicol

Network Topologies Based on CampusNet (Dartmouth)



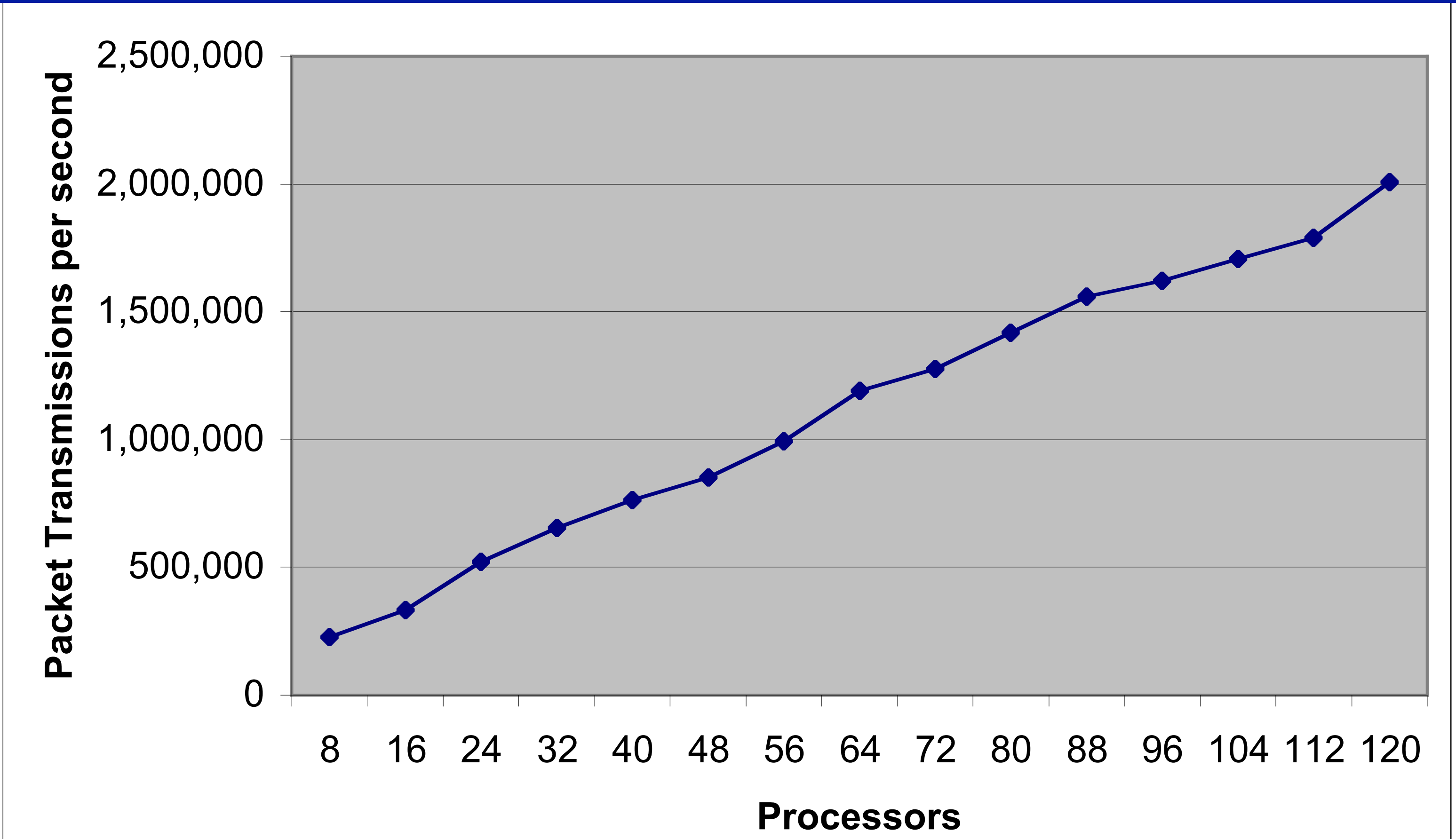


Single Campus Network

- 538 nodes
- 543 links

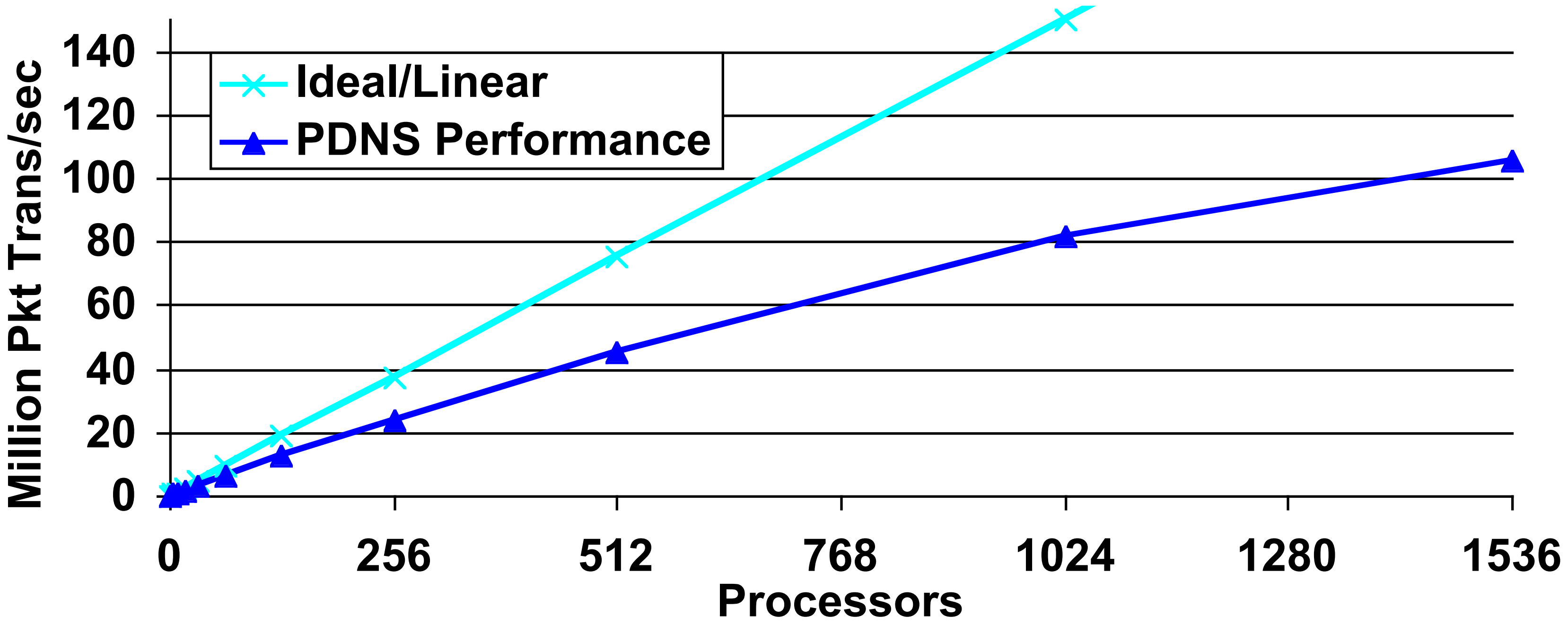
10 campus networks connected in ring

Parallel Simulation Performance (2003)



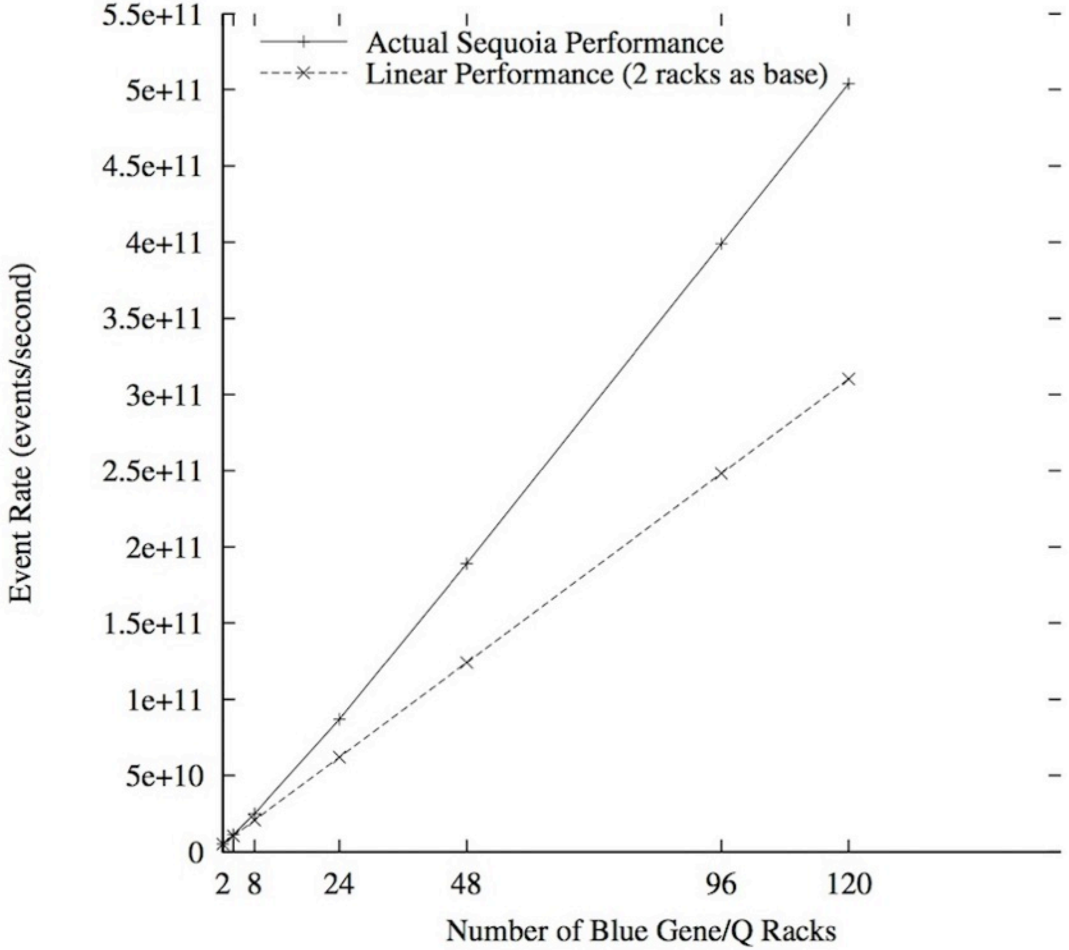
- Cluster of 550 MHz Pentium III Xeon (SMP) machines; gigabit Ethernet
- PDNS: Parallel version of NS2; conservative synchronization
- Each processor simulates 10 subnetworks (weak scaling)
- Up to 120 processors simulating 645,600 network nodes
- Approx. 33K ev/sec/CPU; later optimizations improved to 81K ev/sec/cpu

PDNS on a Supercomputer (Lemieux, PSC, 2003)



- HP-Alpha ES45 servers, 1 GHz CPUs, Quadrics switch
- 147K PTS on one CPU (approximately 300K ev/sec)
- *Campus network* topology, FTP traffic (500 packets/flow, TCP)
- Weak scaling (up to ~4 million network nodes)
- Performance up to 106 Million PTS (138K ev/sec/cpu)

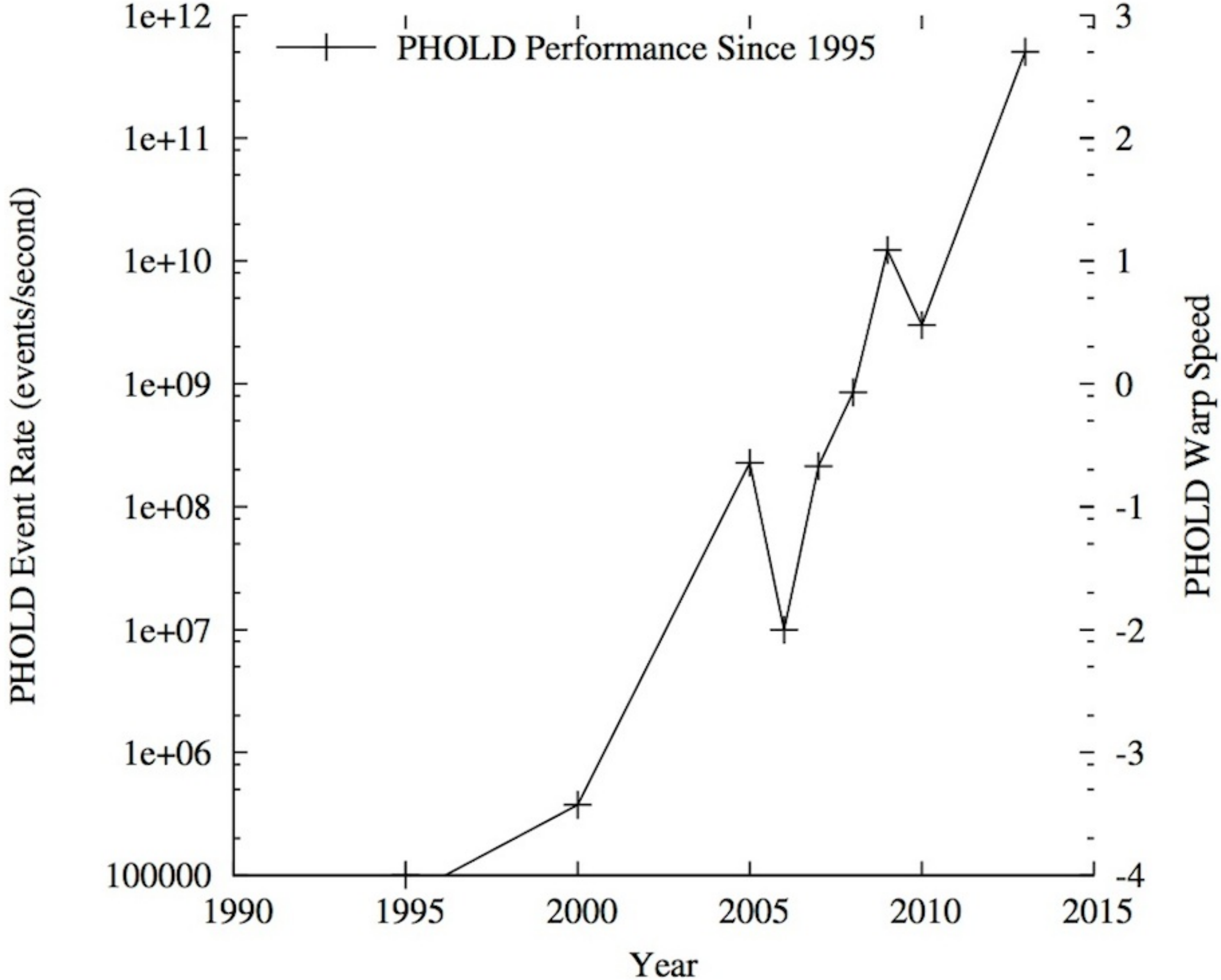
Recent Performance Data (2013)



- Sequoia LLNL Bluegene/Q Supercomputer; almost 2 million cores
- Synthetics benchmark: Parallel Hold (PHOLD)
- Up to 504 billion events/second (256K events/sec per core)

Barnes, et al., "Warp Speed: Executing Time Warp on 1,966,080 Cores," PADS Conference, 2013.

Historical Performance (PHOLD)



Barnes, et al., "Warp Speed: Executing Time Warp on 1,966,080 Cores," PADS Conference, 2013.

Performance Per Core is Stagnant

Three orders of magnitude performance improvement from 2003 to 2013, but

- 2003: 138K events/second/core

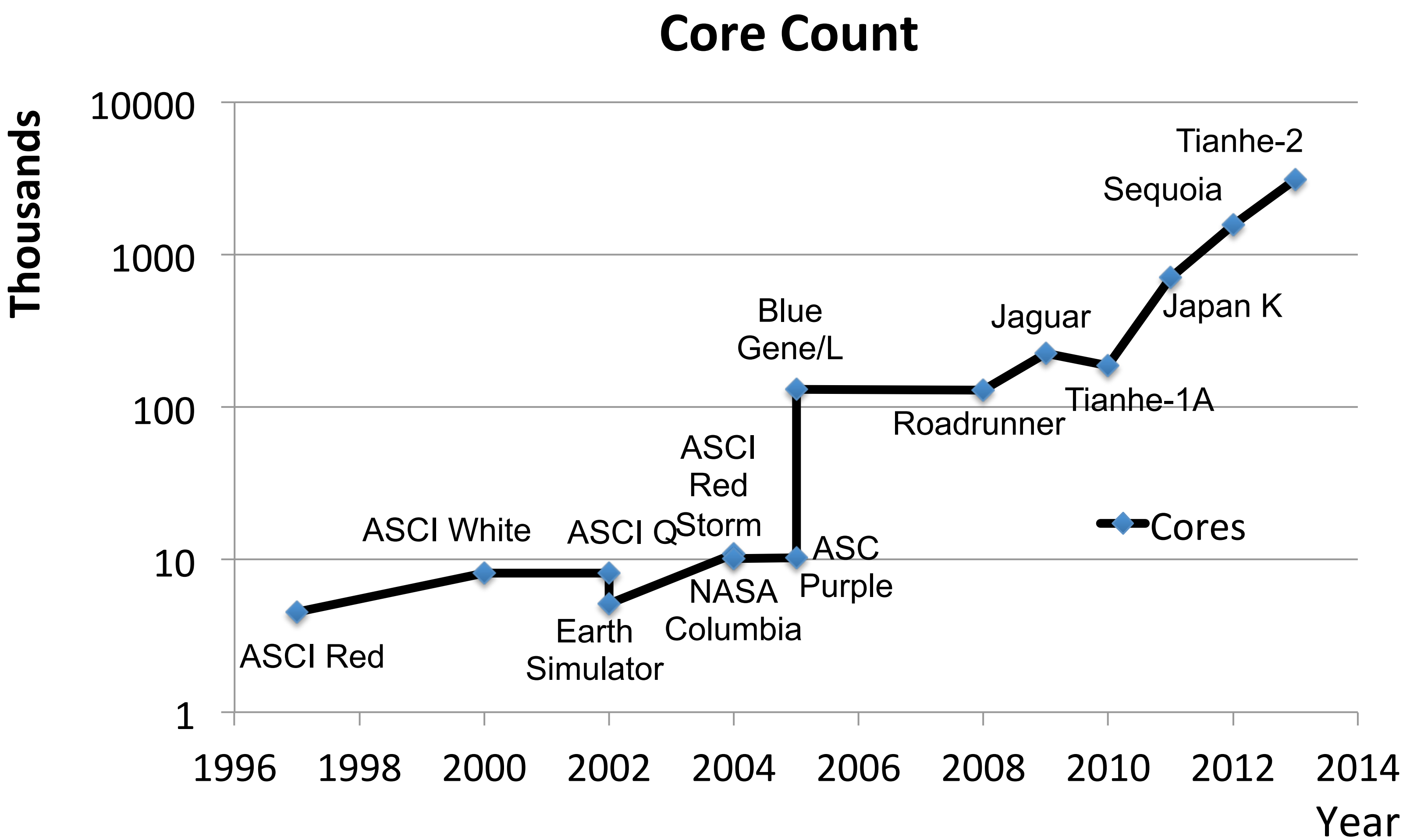
  (Network simulation; Lemieux, 1536 cores)
- 2007: 32K events/second/core

  (PHOLD, Blue Gene/L, 16,384 cores)
- 2013: 256K events/second/core

  (PHOLD, Blue Gene/Q, 1,966,880 cores)

Performance increases coming almost exclusively from increased parallelism

Supercomputer Core Count



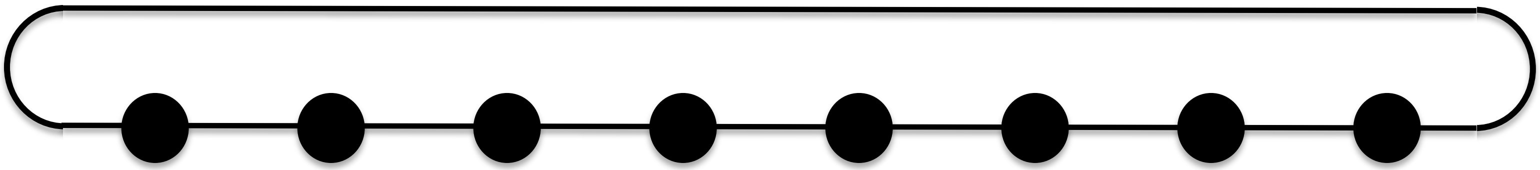
Pre-2005: Increased clock speed and modest core count increases

Post-2005: Core count increases

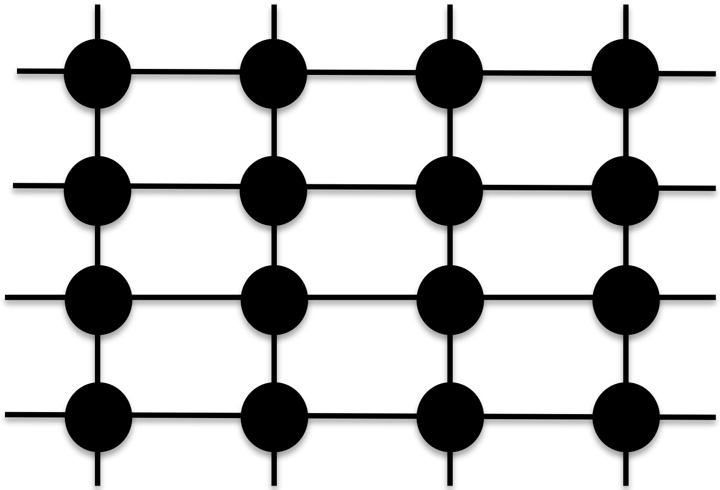
PDES Today

- Several technology demonstrations have yielded impressive results, highlighting the ability of PDES technology to greatly accelerate large-scale discrete event simulations
- Performance improvements are now coming almost exclusively from increased parallelism
- Nevertheless, limited penetration into the commercial M&S marketplace

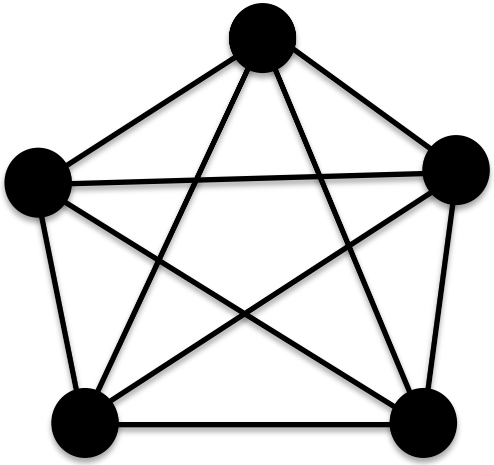
A Caveat: Network Topology



ring



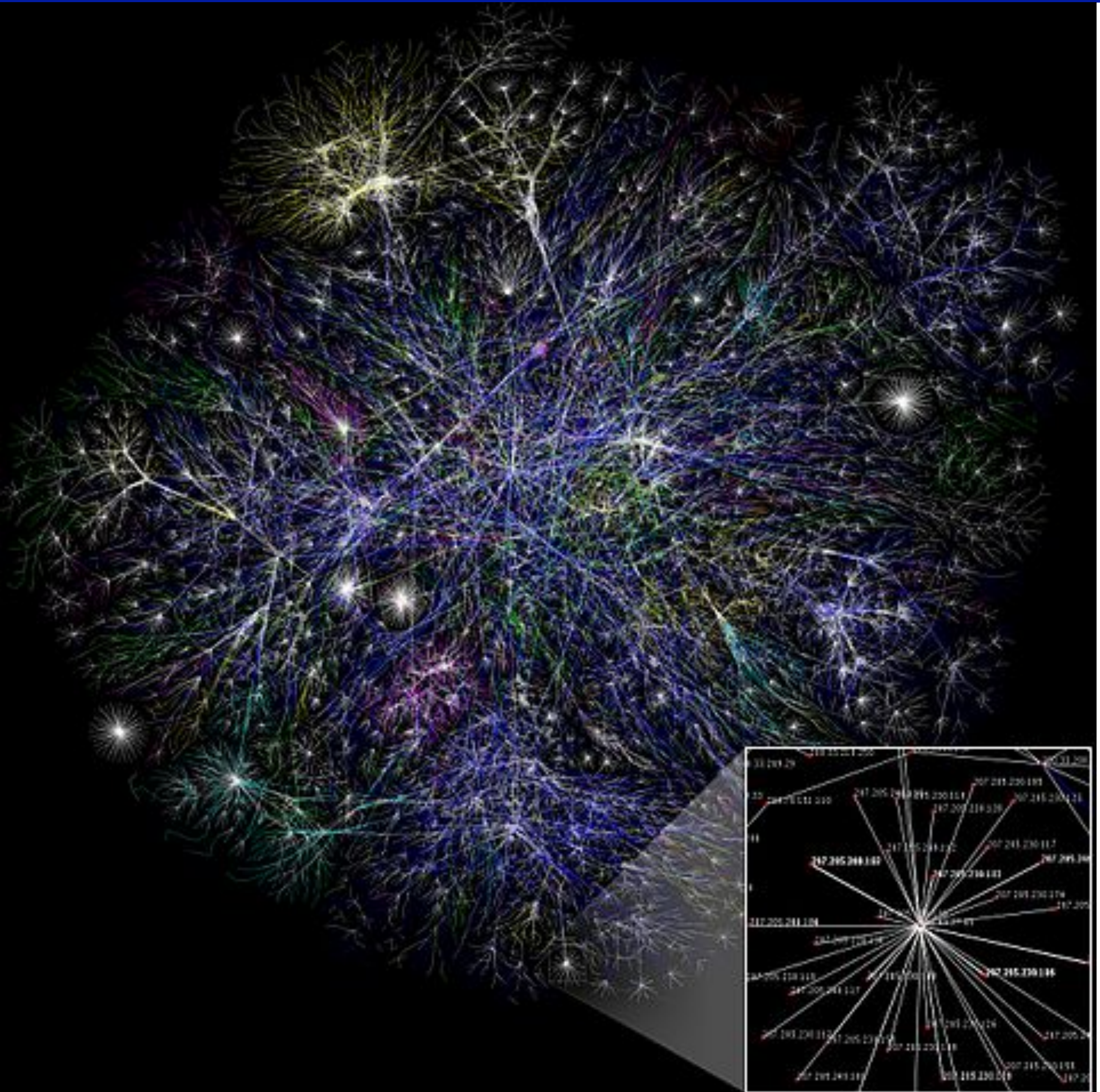
grid/toroid



fully connected

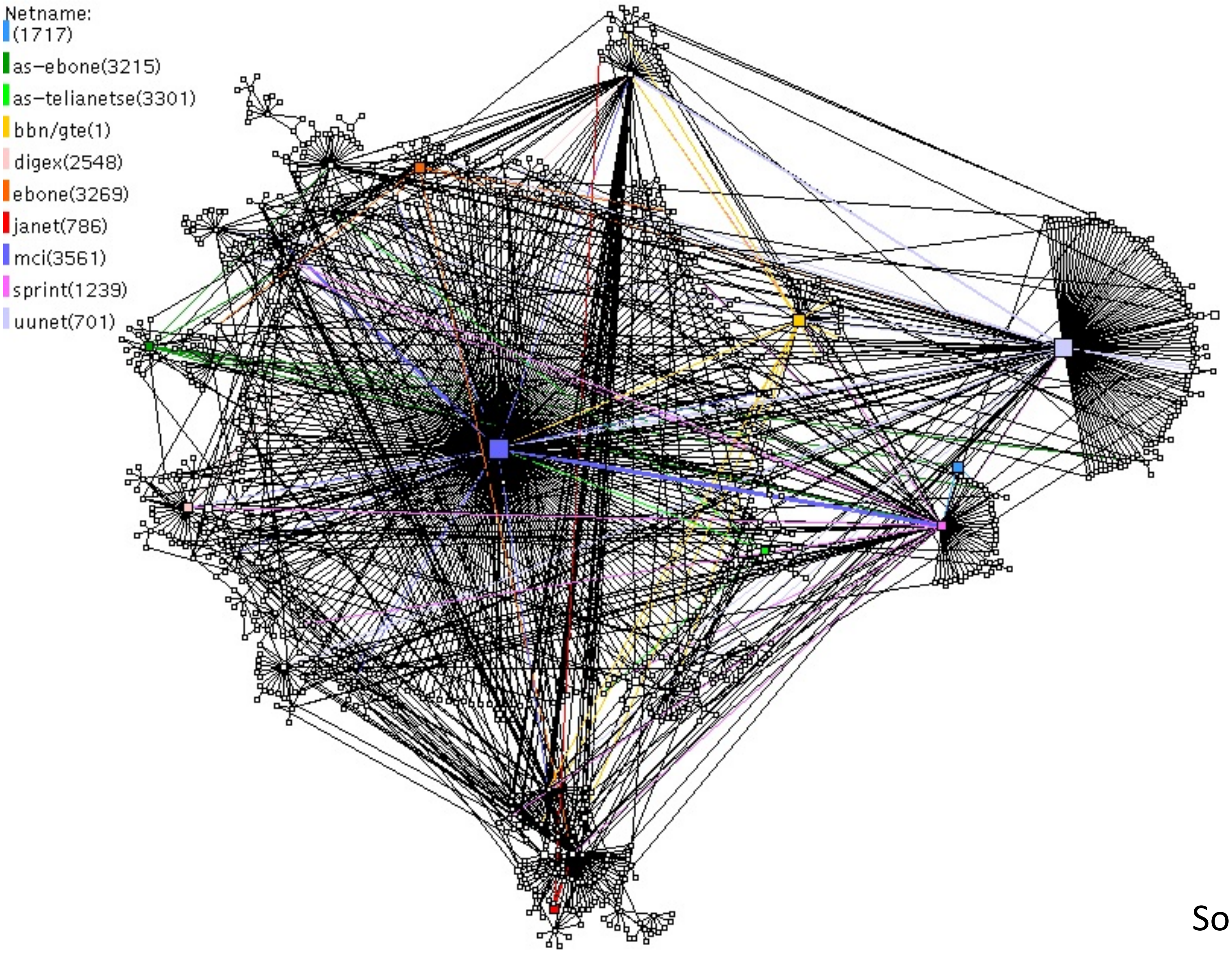
- Most large-scale PDES studies to date focus on highly regular, symmetric network topologies
- Real-world networks are not so well structured!

Internet Topology



Source: wikipedia

Internet: Autonomous System Level

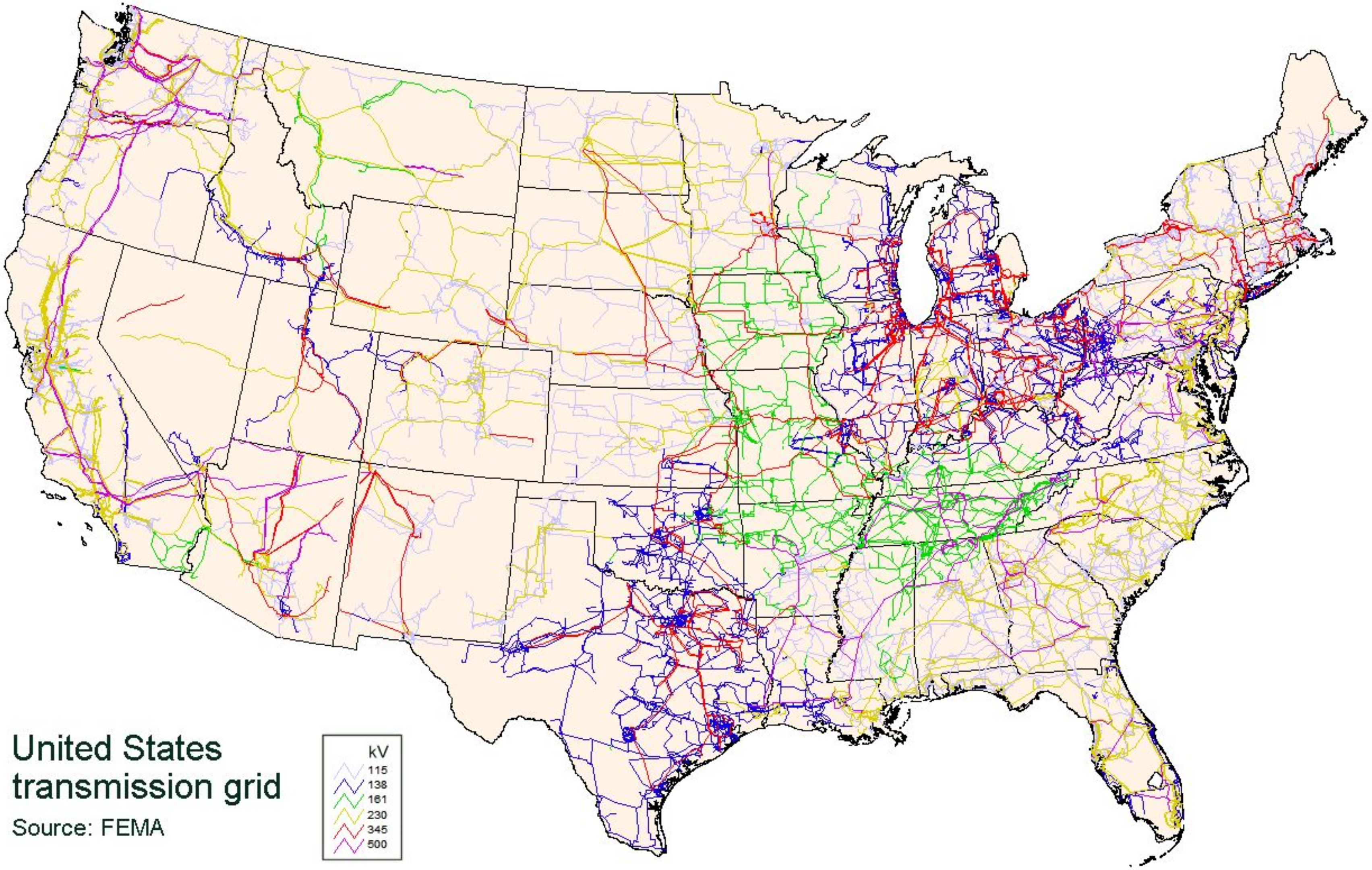


Source: CAIDA

Social Networks

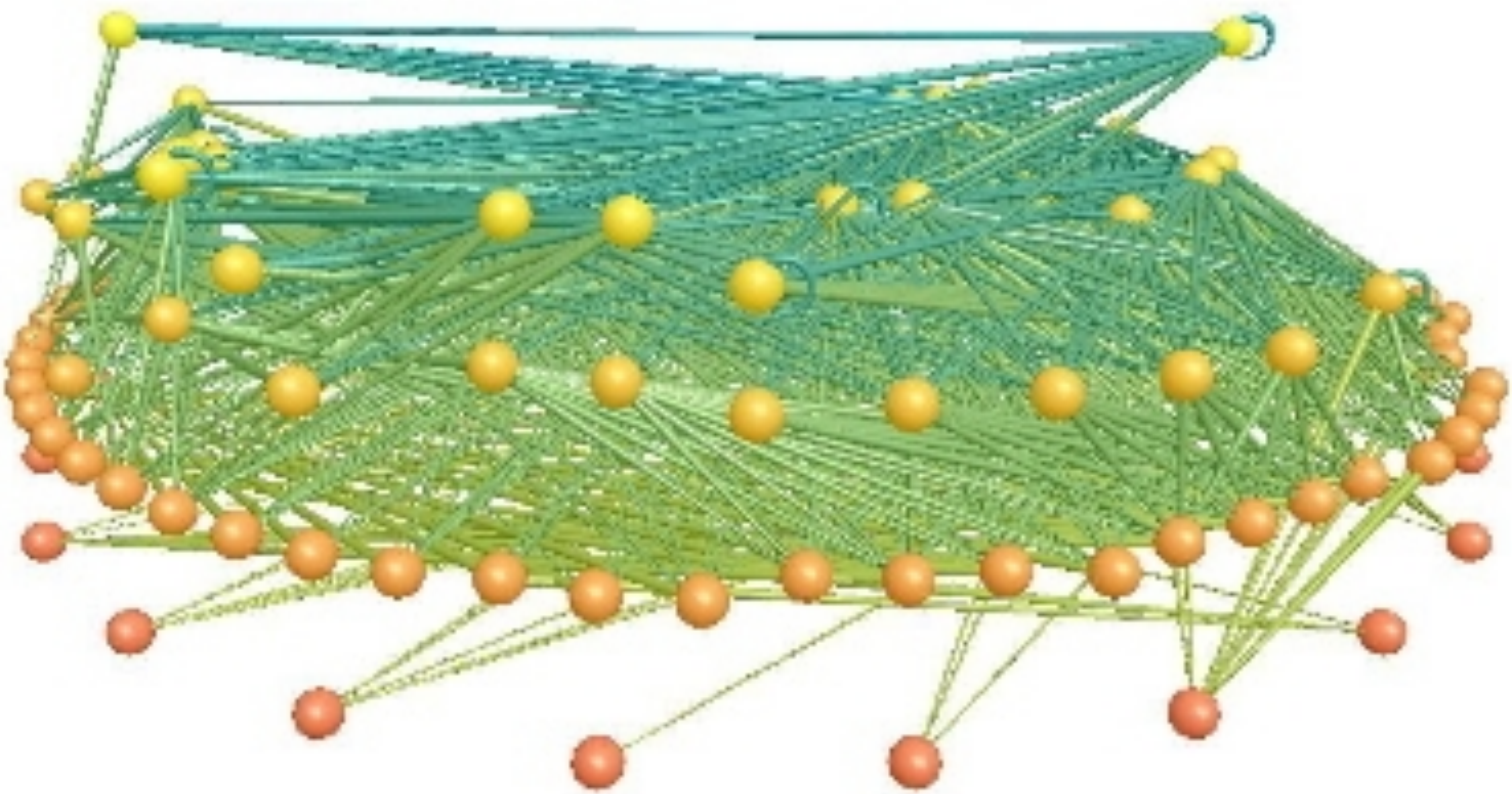
http://blog.revolutionanalytics.com/2010/12/facebooks-social-network-graph.html

Electric Power Grid



http://www.geni.org/globalenergy/library/national_energy_grid/united-states-of-america/americannationalelectricitygrid.shtml

Food Web



http://www.virtualtravelog.net/2003/04/the-emerging-science-of-food-webs/

From Technology Demonstrations to Practice

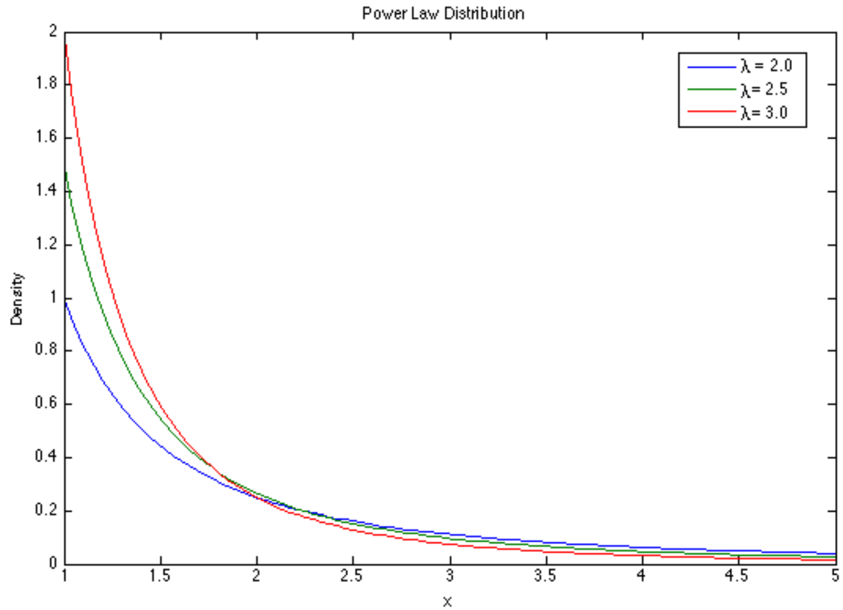
Work to date largely technology demonstrations illustrating PDES can accelerate large-scale simulations

Many challenges for real-world problems

- Irregular topologies
- Non-homogeneous behavior
- Workload imbalances

What performance is achieved in networks of practical interest?

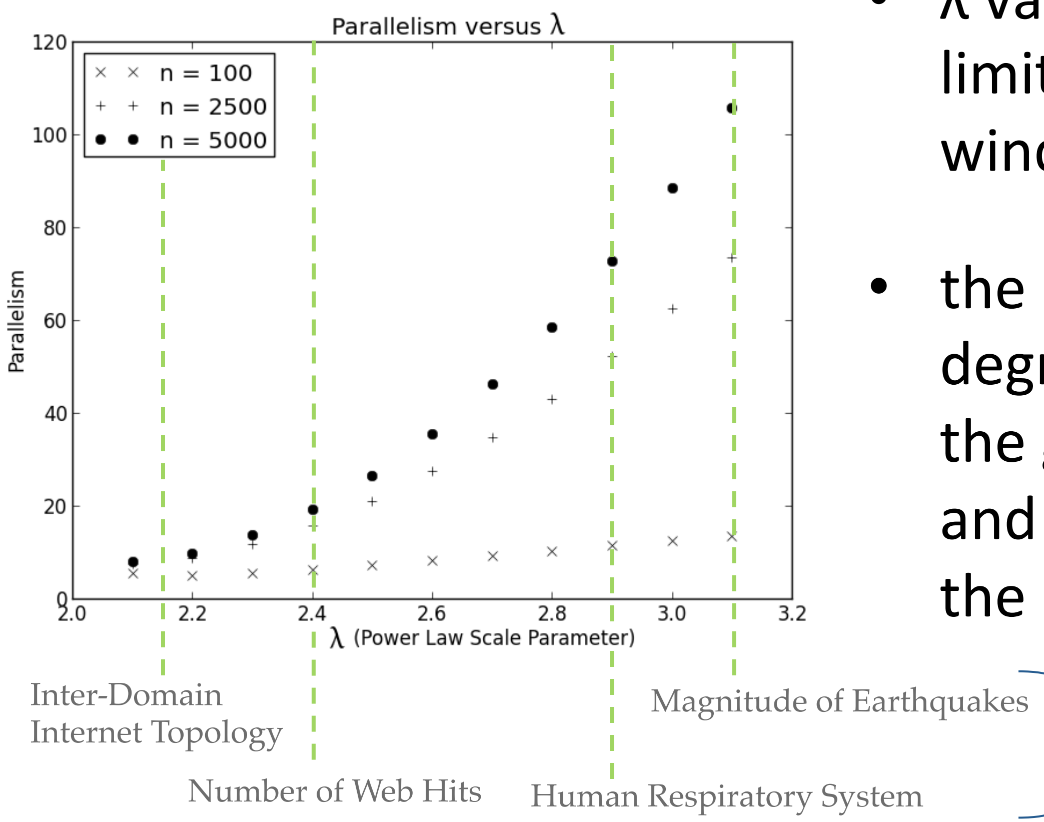
Scale Free Networks



$$P(k) \sim ck^{-\lambda}$$

- **Scale free networks** are those where the node degree distribution follows a power law; "hub" nodes with relatively high degree and "leaf" nodes
- Scale free networks have received much attention because many real-world systems exhibit this property: Social networks, World-wide web [Barbasi, Reza 99], Internet (autonomous system level) [Faloutsos et al. 99, Zhang et al. 11], Mobile communication networks [Onnela 07], Biological systems [Solé 96]

Analytical Results



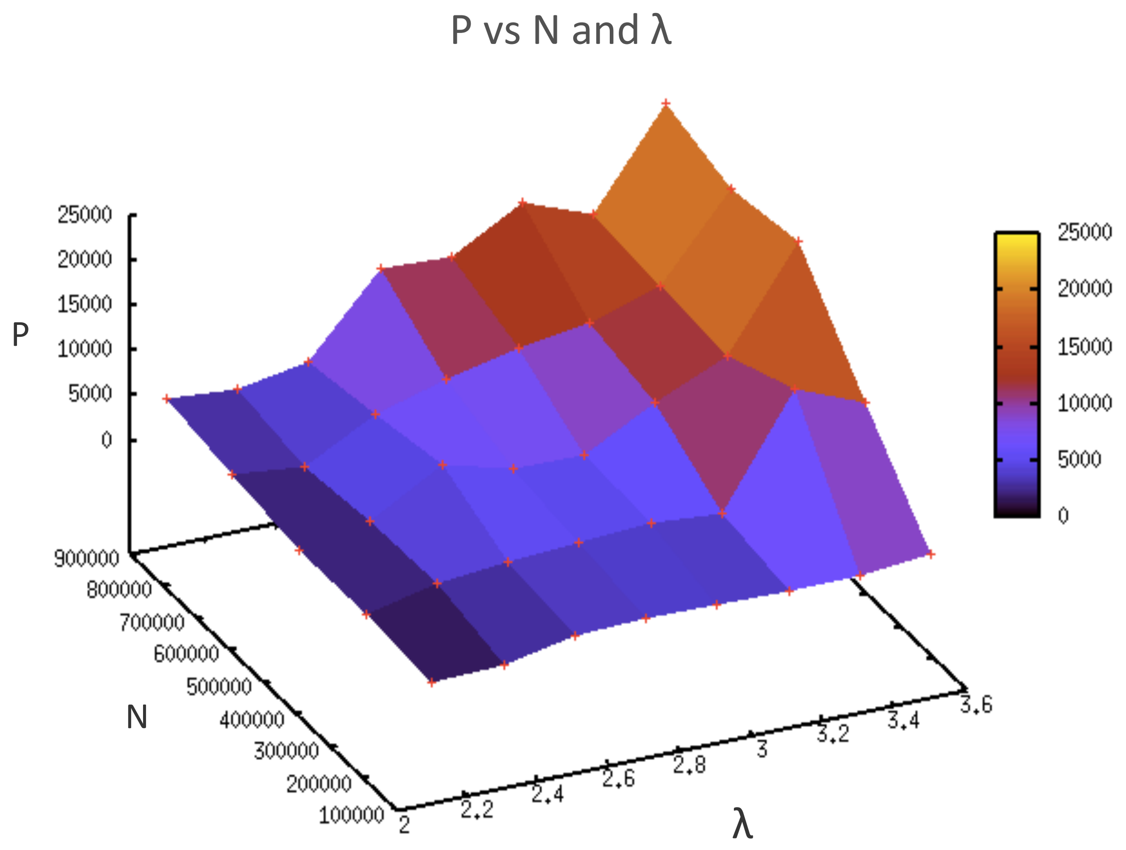
- λ values around 2 significantly limit the available parallelism of windowed approaches
- the heavier the tail of the degree distribution (lower λ), the greater the connectivity and subsequent workload of the largest hub

λ' s from the literature

Conservative parallel simulation using a synchronous "window" synchronization algorithm

Parallelism Results

- Parallelism plotted against the number of vertices in the graph and varied λ
- Lower λ's yield both less parallelism and slower increases with the addition of vertices
- Limited parallelism, even for large networks



R. Pienta and R. M. Fujimoto, "On the Parallel Simulation of Scale-Free Networks," *PADS Conference,* May 2013.

PDES for Scale-Free Networks

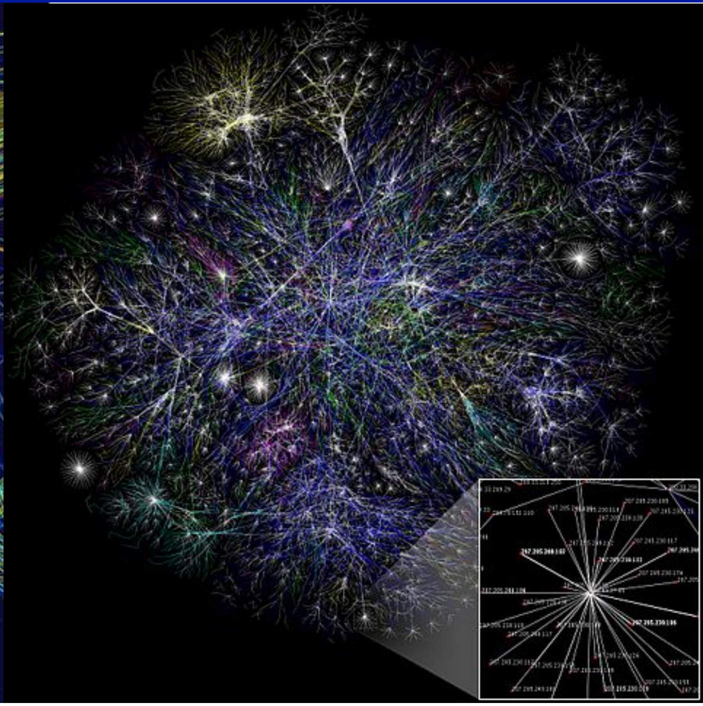
- Large scale-free networks may exhibit limited parallelism: orders of magnitude less than the number of nodes
- Our analytical model shows that: Lower λ's create larger deg.-dist. tails that cause richly connected hubs.
  - These large degree hubs receive a disproportionate amount of simulation-time messaging and overhead
  - This results in decreased potential parallelism
- Problematic for both synchronous and asynchronous conservative synchronization algorithms for networks with low λ
- Modern HPC machines require large massive parallelism to achieve high performance

TOWARD A COMMON M&S RESEARCH AGENDA

Common M&S Research Agenda

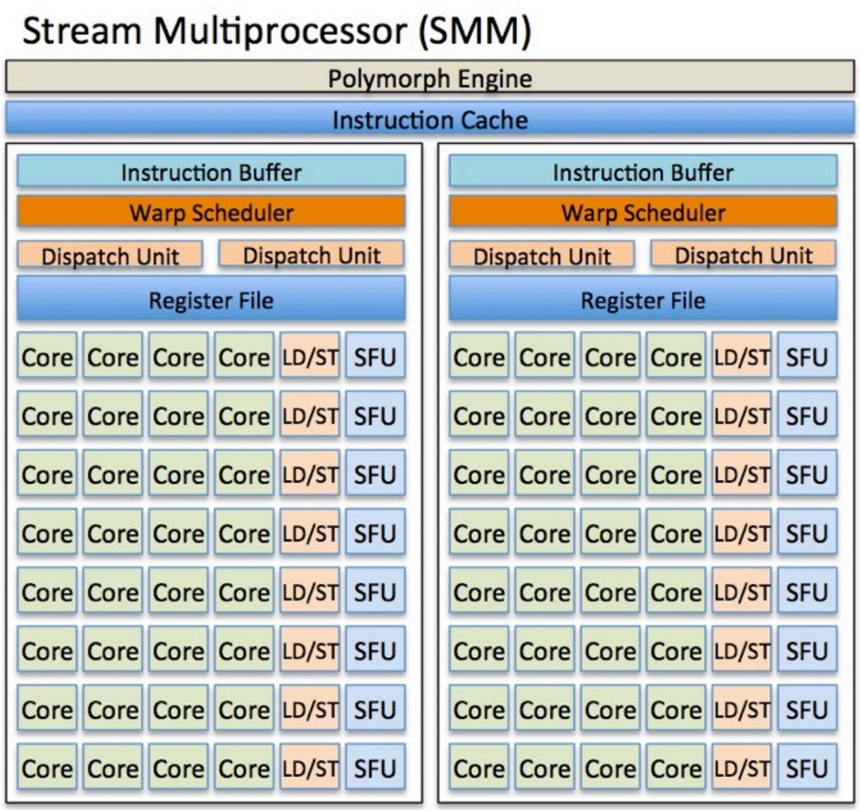
- Should the modeling and simulation community identify, build consensus for, and promote a common research agenda that includes major problems being studied today?
- Should the community define a (small) set of grand challenge problems that if solved, will have large impacts?
- If the answer to these questions is yes, how should the community organize itself to address these issues?

Challenge 1: Scalable Simulations of Irregular Networks



Challenge: Create scalable, realistic large-scale simulations of irregular networks including those with skewed node degree distributions and demonstrate their ability to create new insights into real world systems.

Challenge 2: Exploitation of Heterogenous Supercomputers (Graphical Processing Units)

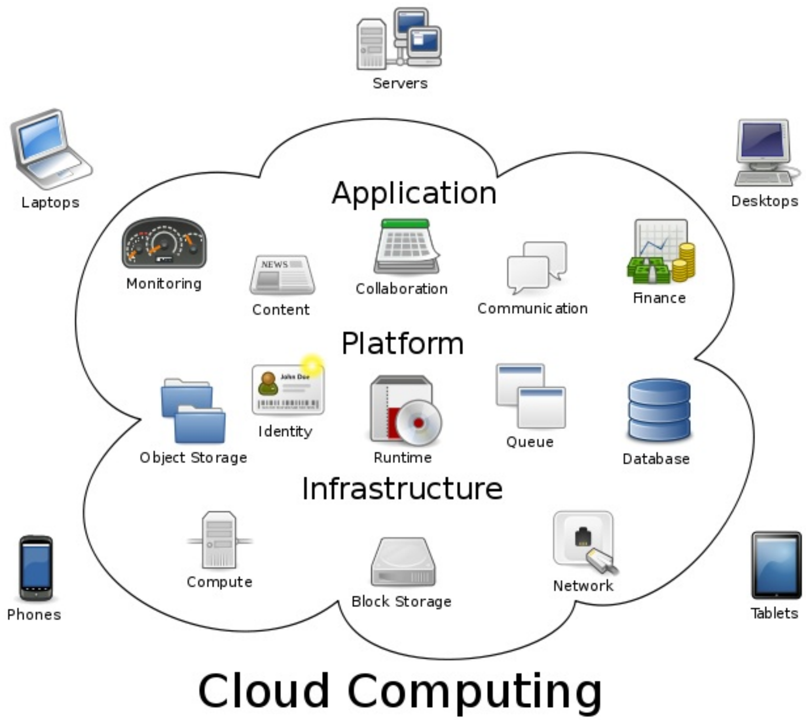


Challenges

- Single Program Multiple Data execution
- Memory architecture: Irregular data structures
- Limited communications bandwidth
- Difficult to Program – architecture specific

Challenge: Develop practical, scalable techniques to exploit graphical processing unit (GPU) accelerators in modern high performance computing and mobile computing platforms to significantly accelerate real-world simulation applications.

Challenge 3: Exploitation of the Cloud



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

Opportunity

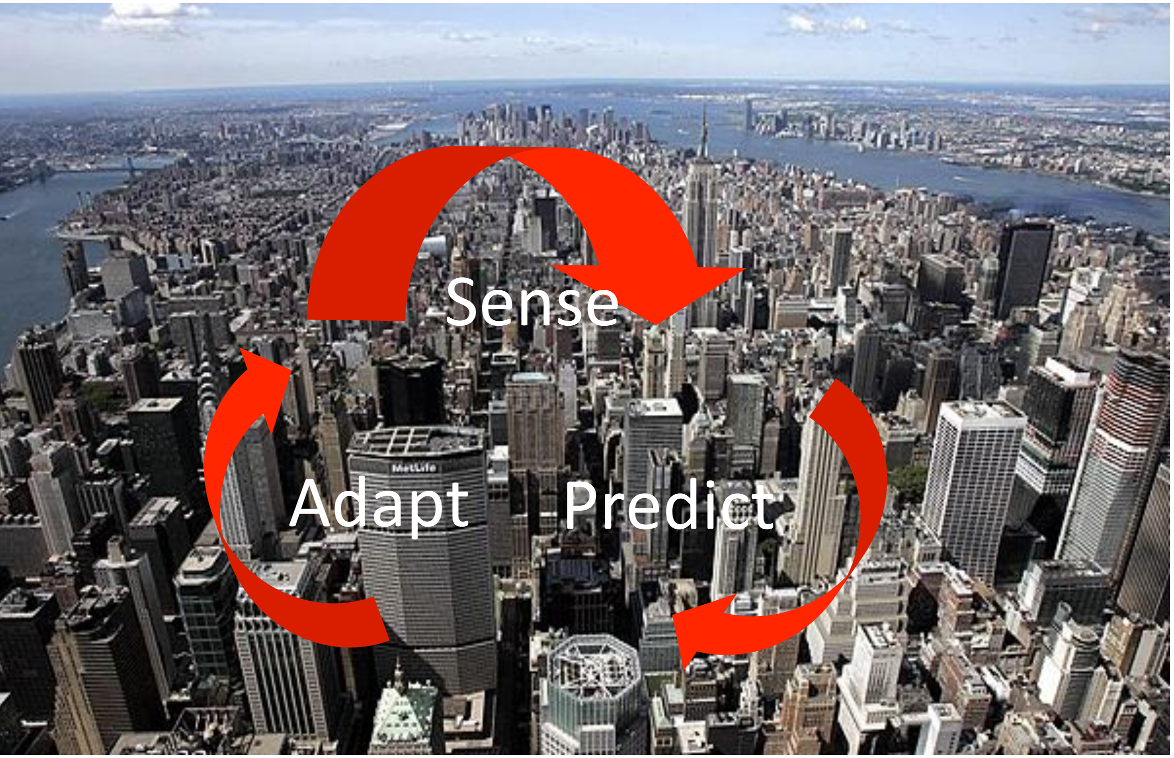
- Users need not own a supercomputer!
- Potential to make parallel simulation widely accessible

Challenges

- Communications can have high jitter
- Shared computing platform

Challenge: Enable parallel and distributed simulation technologies to become broadly accessible to the general modeling and simulation community by simplifying the development of simulation code and exploitation of cloud computing.

Challenge 4: Real-Time Management of Large-Scale Systems



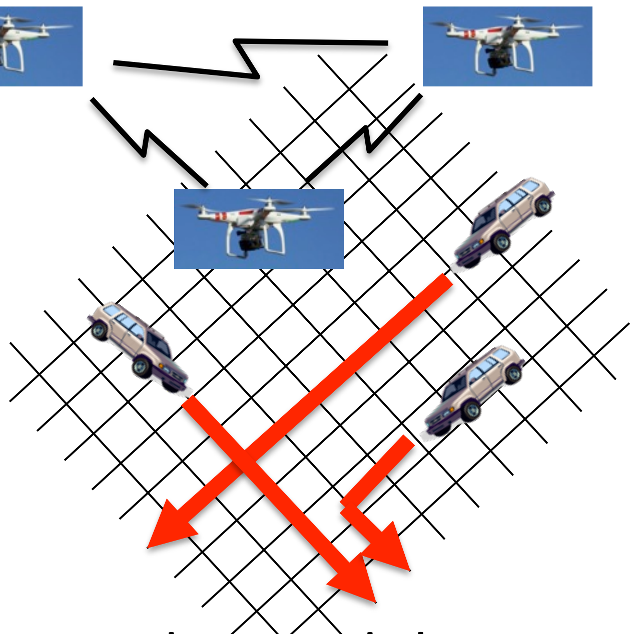
Automate M&S life cycle

- 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

Challenge: Create on-line parallel and distributed simulation technologies that are suitable for real-time decision-making utilizing live data feeds from sensors and other sources.

Challenge 5: Understand and Reduce Energy Consumed by Parallel/Distributed Simulations

Autonomous team of mobile sensors monitoring an evolving system (e.g., forest fire, cloud plume, traffic)



Processing Loop

1. Sense: construct current system state
2. Predict: distributed simulation to estimate future state
3. Adapt: relocate sensors

Challenge: Understand the power and energy consumed by parallel and distributed simulations and developed means to adapt the simulation to optimally manage its use of energy subject to constraints imposed by the environment in which it executes.

Challenge 6: Easily Composed Simulations

Levels of integration

Composeability

Interconnection of **models**

- Consistent model objectives
- Alignment of model assumptions
- Common abstractions

- Domain specific
- Multi-scale spatial issues
- Multi-scale temporal issues

Interoperability

Interconnection of **simulation software**

- Consistent data types
- Common software APIs
- Little endian/big endian

- High Level Architecture (IEEE 1516)
- Distributed Interactive Simulation (IEEE 1278)
- CORBA, MS COM, Enterprise JaveBeans

Integrateability

Interconnection of **computing systems**

- Hardware
- Firmware
- Networks

- Virtual machines
- Protocols (e.g., TCP/IP)
- Hardware interconnection standards

Challenge: Enable rapid composition of separately developed simulation models for execution in parallel or distributed simulation environments.

National Modeling & Simulation Coalition

Mission: … to create a unified national community of individuals and organizations around the M&S discipline and professional practice and to be the principal advocate for M&S.

- Initiatives brought forward by members (industry, education, research & development)
- M&S Common Research Agenda
- MOUs with M&S organizations (NTSA, SCS, ACM-SIGSIM)
- NAICS code for modeling and simulation
- Interface to U.S. M&S Congressional Caucus

Concluding Remarks

- Parallel discrete event simulation offers tremendous potential to accelerate large-scale simulations
  - PDES is increasing in importance because single processor performance is not improving
- PDES has attracted much research over the years
  - Synchronization a well-studied problem, but not completely solved
  - High levels of performance demonstrated on supercomputers
- Emerging new technologies (big data, cloud, IoT, GPGPU computing) create new opportunities and challenges for parallel and distributed simulation
  - Addressing properties of real-world applications
  - Technology needs to be more accessible to non-PDES-experts

Questions?