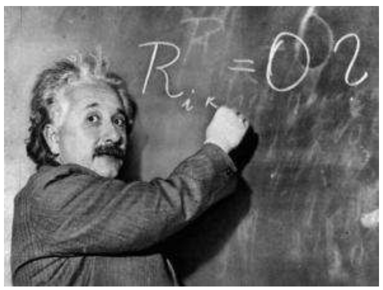
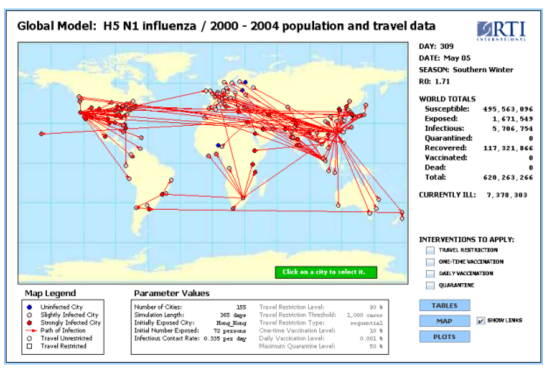
The value of HPC in Scientific Projects

Diglio A. Simoni
dsimoni@rti.org

RTI International is a trade name of Research Triangle Institute

A New Science Paradigm





- A thousand years ago
  - Experimental Science
    - Description of Natural Phenomena
- Last few hundred years
  - Theoretical Science
    - Newton, Maxwell, Einstein, String Theory…
- Last few decades
  - Computational Science
    - Simulation of complex phenomena
- Today
  - e-Science or Data-centric Science
    - Unify theory, experiment and simulation
    - Using data exploration and data mining

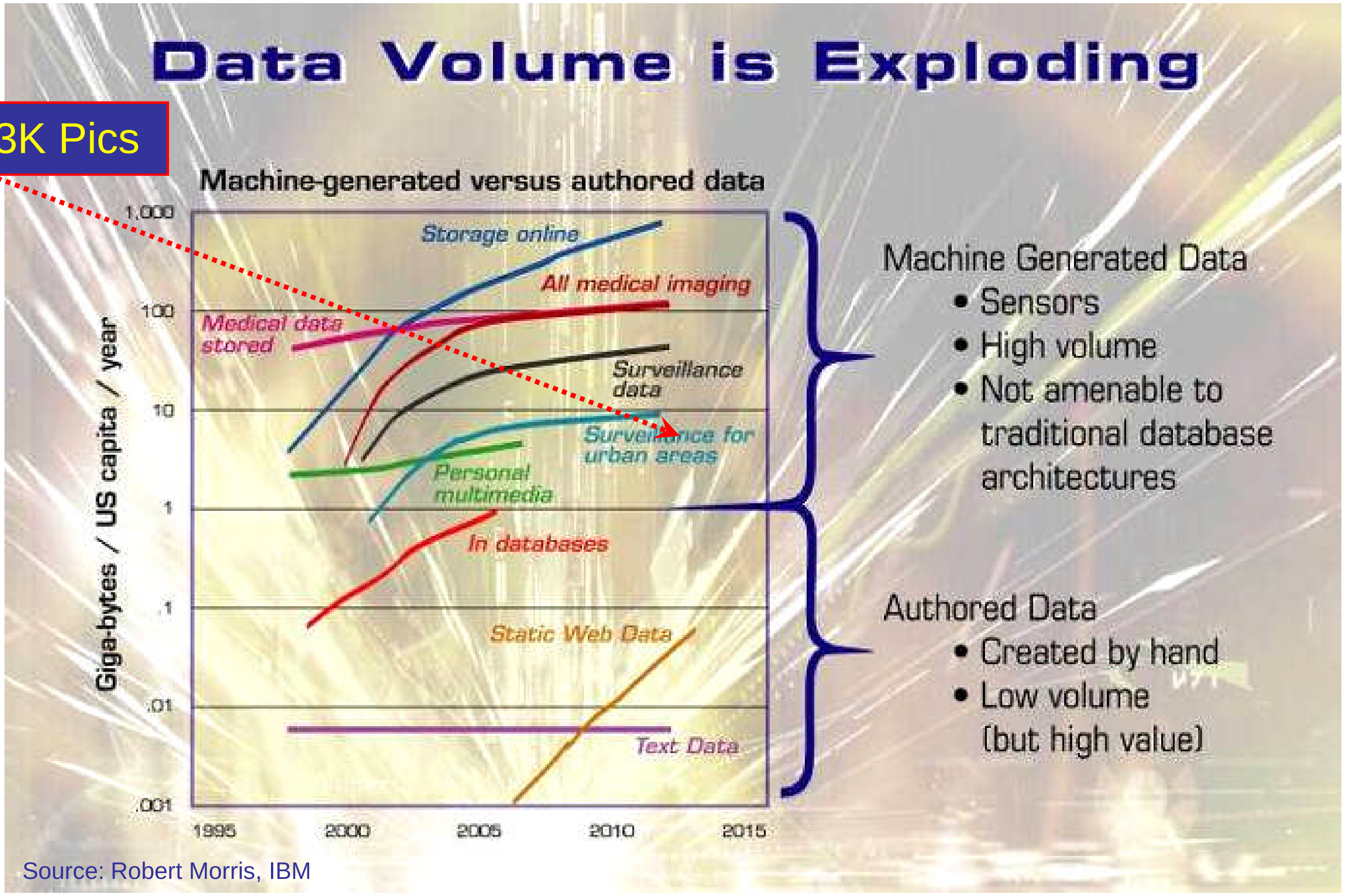
e-Science

- e-Science is about data-driven, multidisciplinary science and the technologies to support such distributed, collaborative scientific research
- Science is being overwhelmed by a ‘data deluge’ from
  - Data generated by sensor networks
  - Data captured by instruments
  - Data generated by simulations
  - Data derived from scientific analyses

High Performance Computing is a key technology to support the e-Science revolution

Data Explosion

7 GB = ~3K Pics

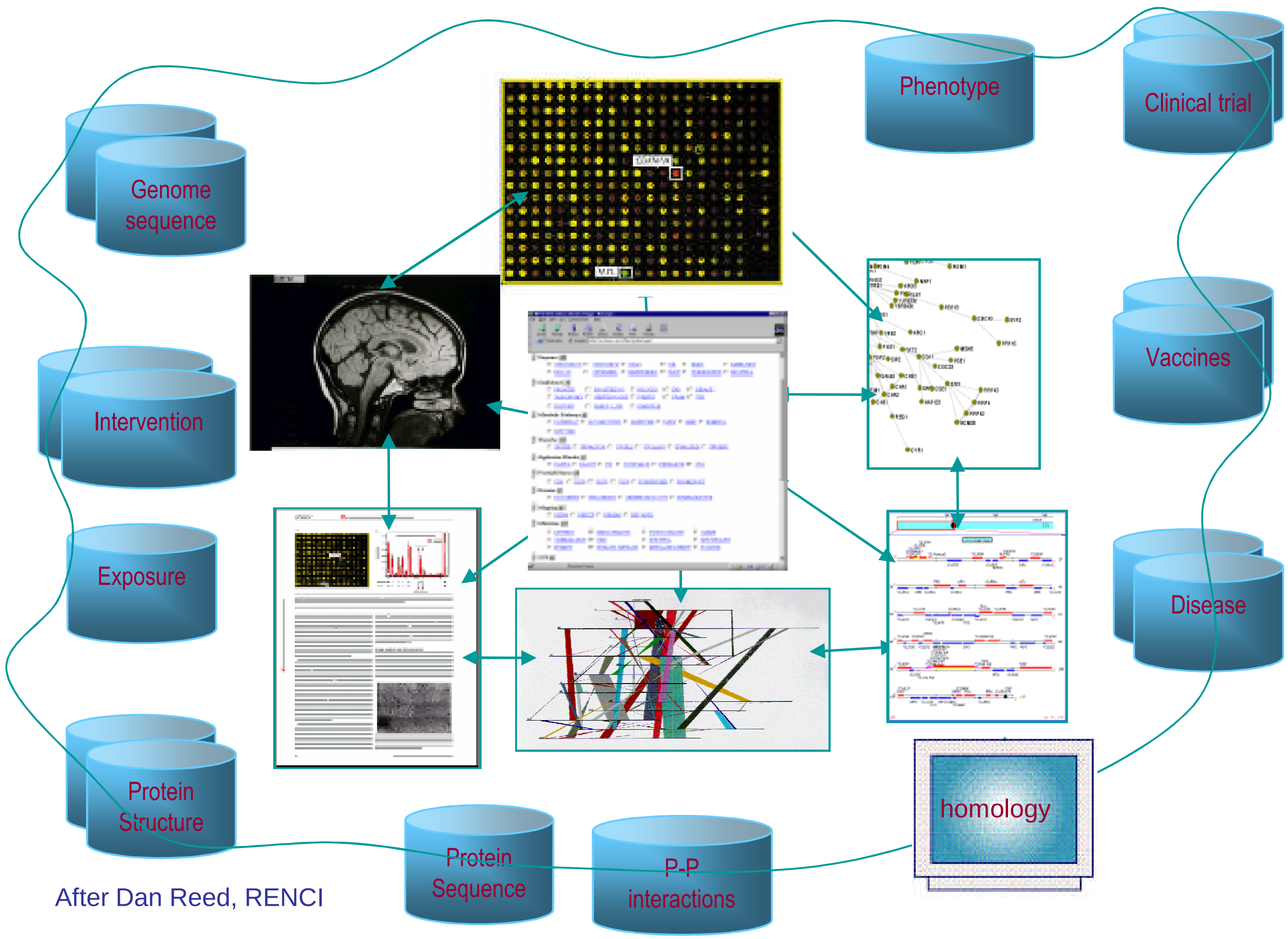


Growth of the Digital Universe

Kilobyte : ~1,000 = 1/10 e-mail
Megabyte: ~1,000,000 = 100 e-mails
Gigabyte: ~1,000,000,000 = 100,000 e-mails
Terabyte: ~1,000,000,000,000 = 100,000,000 e-mails
Petabyte: ~1,000,000,000,000,000 = 100,000,000,000 e-mails
Exabyte: ~1,000,000,000,000,000,000 = 100,000,000,000 e-mails

988 EB = ~210B 4.7GB DVDs. Circle the earth 63 times.

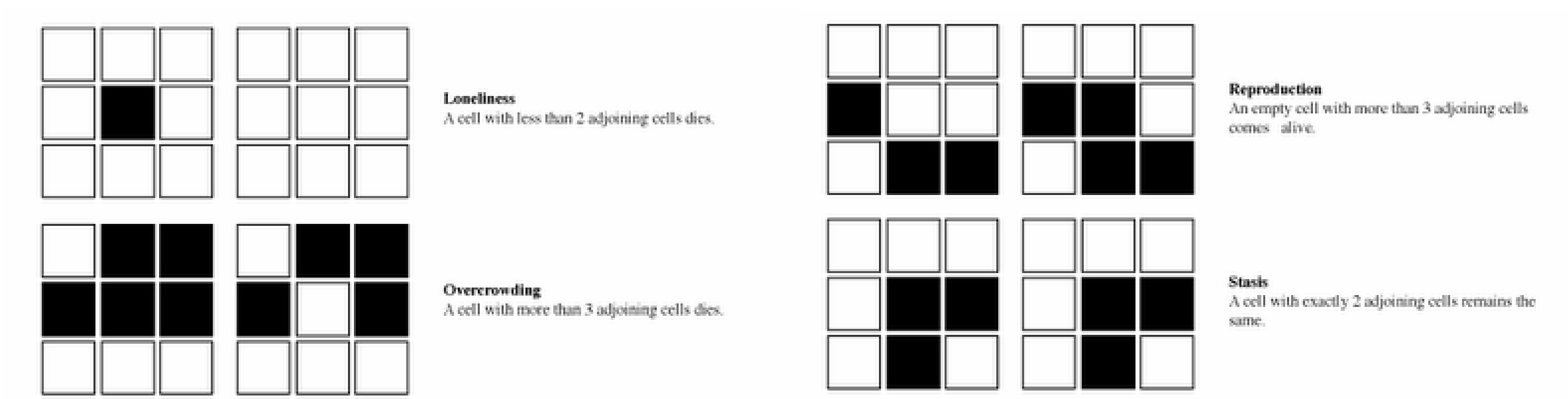
Data Heterogeneity

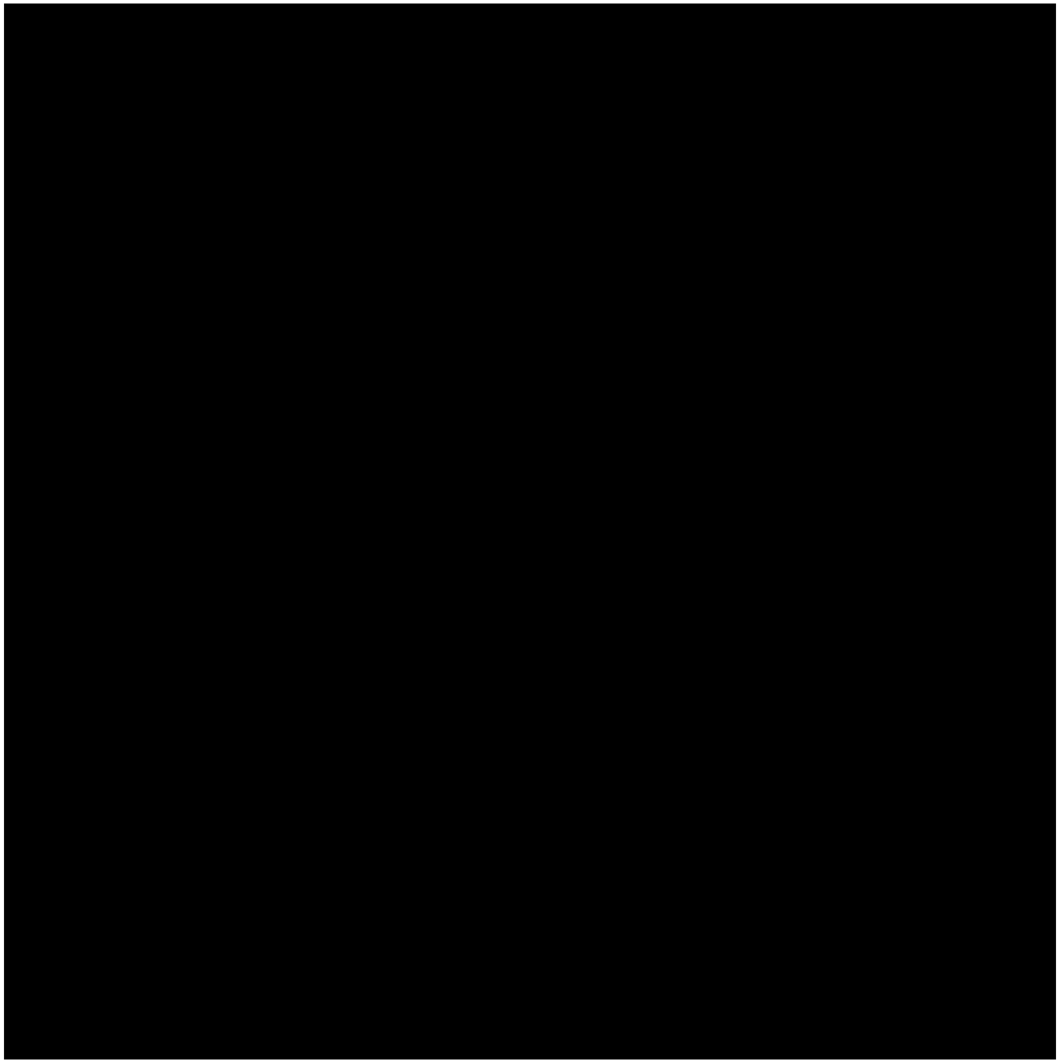


After Dan Reed, RENCI

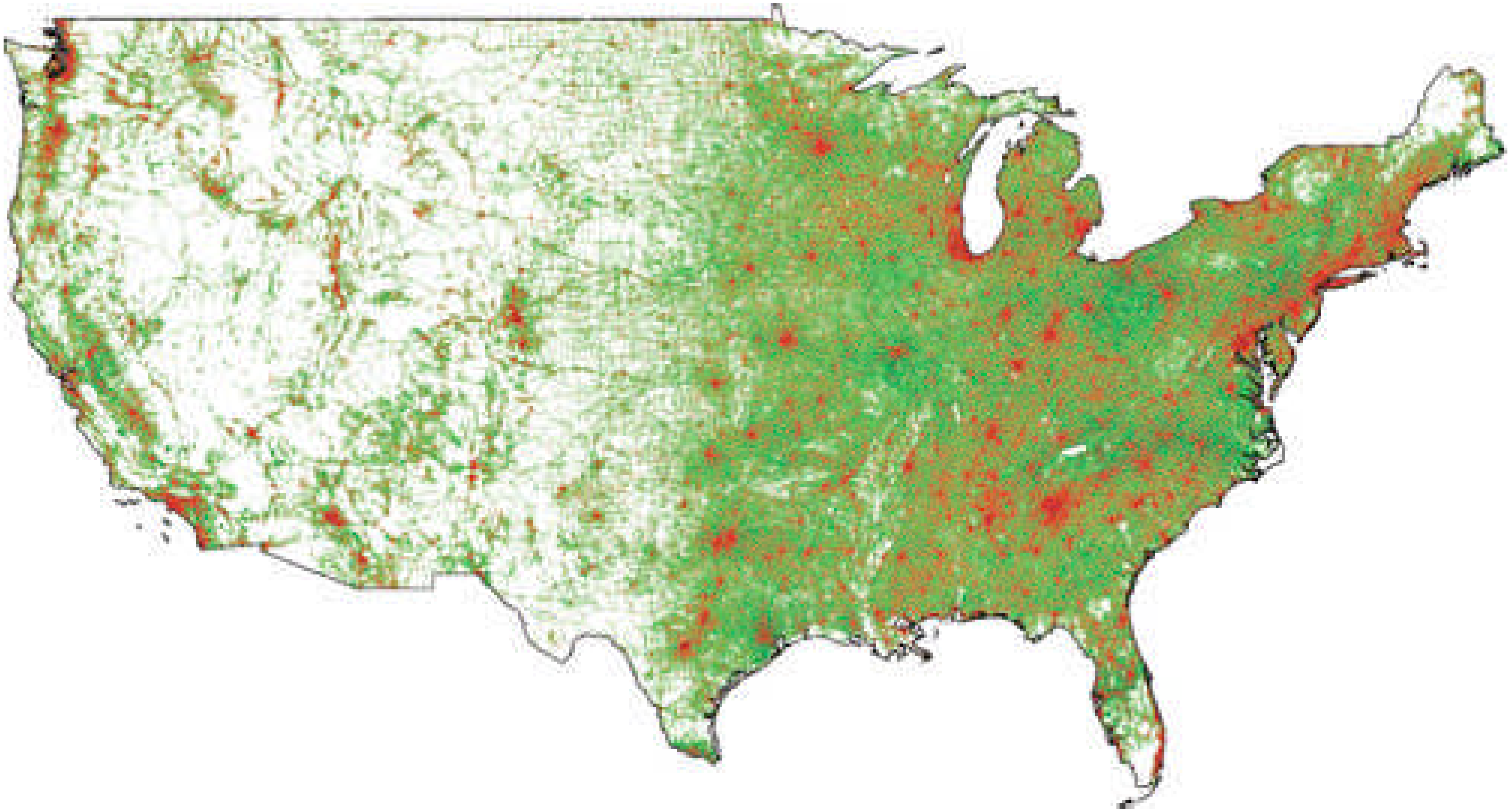
Agent Based Models

An Agent Based Model (ABM) is a specific individual based computational model for computer simulation extensively related to themes in complex systems, emergence, Monte Carlo methods, computational sociology, multi agent systems, and evolutionary programming.





8



Distributed ABM Challenges

- Heterogeneity and Temporal Variation
  - Load balancing
- Scale and Scope of Simulations
  - I/O constraints
- Computing Speed and Memory Requirements
  - Global data structures?
- Efficiency and Sustained Performance
  - Fault tolerance
- **We’re just starting to scratch the surface**

Position: We need better tools

Ideal characteristics:

- Easy to learn, and provide good documentation
- Easy to do very rapid prototyping
- Provide a good library of functions
- Have excellent display capabilities
- Widely used in research and industry

15:45	Invited Speaker: Simulating Costs and Benefits of SBI in an EAP (Diglio Simoni - 25 min)
16:10	SIMUL Panel: Challenges in simulations: large scale, education, performance (Diglio Simoni - 2 min)
16:12	SIMUL Panelist #1 (Roy Crosbie - **California State University, Chico, USA** - 7 min)
16:19	SIMUL Panelist #2 (Diglio Simoni - **RTI International, RTP, USA** - 7 min)
16:26	SIMUL Panelist #3 (Gregor Papa - **Jozef Stefan Institute, Slovenia** - 7 min)
16:33	SIMUL Panelist #4 (Yiping Yao - **University of Defense Technology, P. R. China** - 7 min)
16:40	SIMUL Panel Discussion (All - 20 min)
17:00	Open Discussion: Special Topics on Simulation (All - 15 min)
17:15	Open Discussion: Online Journals (All - 15 min)
17:30	End Afternoon Session

Panel Questions

- What in your experience has been most successful educational activities for future HPC simulation researchers to develop their abilities and intuition across multiple disciplines?
- How do we bridge the gap between computing at the desktop and computing on HPC systems, especially given that either newcomers want the "cadillac" and don’t want to start with a small simulation, or viceversa?
- It appears that industry trends over the past 15-20 years have made things more challenging by going from shared memory to distributed memory, from proprietary systems to Linux, from single-node to multicore. On that last note, what is the level of preparedness of the scientific community for multicore computing?

Panel Questions

- What we can do to help the next generation of scientists accept and use HPC?
- What about HPC clouds? As we look at HPC as a service, are we ready for it?
- We're starting to see HPC make tremendous inroads in "nontraditional" areas, as a result of the so-called data deluge that appears to be pervasive. How can we help non specialists learn quickly about the benefits of HPC?
- What simulation tools are best suited for HPC (e.g. MATLAB PCT, gridMathematica, parallel R, etc.)

Using FPGAs to Support Low-Cost HPC

Roy Crosbie

California State University, Chico, USA

SIMUL 2010
Nice, France
22-27 August 2010

Attached Processors-FPGAs

- Field-Programmable Gate Arrays (FPGAs) are now being used in high-speed real-time simulations
- Conventional Real-Time Operating Systems (RTOS) can't provide <10-μS frame times necessary for some high-speed real-time applications.
- Attached processors based on FPGAs can deliver this capability
- FPGA-based attached processors have potential beyond real-time simulation applications

Attached Processors - History

- **Minicomputers**
  - Floating-point Processors (e.g. for PDP 8, PDP 11)
  - Attached Array Processors (e.g. AP 120B)
- **Microprocessors**
  - Floating-point Processors (e.g. 8087, 80387)
  - Graphics Processing Units (e.g. GTX400)

FPGA Features

- **Highly parallel with thousands of functional units**
- **Memory is highly distributed among functional units**
- **Reconfigurable architecture to match application**
- **Fixed-point arithmetic operations**
- **Various programming methods**
  - Hardware Description Languages (VHDL, Verilog etc.)
  - Simulink Blockset
  - Matlab M-code
  - Variants of C
- **Can incorporate PowerPC or Intel style processors**
- **Flexible data transfer options**

Challenges for simulations in industrial daily optimization processes

Gregor Papa

Jožef Stefan Institute

:: CSD.IJS.SI

Jožef Stefan Institute

Computer Systems Department

Simulation for optimization

- Simulation as an evaluation tool of optimization process
  - During optimization (numeric or combinatorial) each potential solution has to be evaluated
  - In many systems there is no exact (mathematical) solution known
    - The system has to be simulated

Simulation time complexity

- Depending on the complexity of the system the simulation can be computationaly very consuming
  - It might take a lot of time
- Two solutions
  - Simplification of simulation
  - Parallelization of simulation

:: CSD.IJS.SI ::

Simulation customization

- Commercial simulation systems
  - Simplification is not always possible
  - The need of parallelization
    - Cluster multicore execution

- Homemade simulation sistems
  - Can use any level of simplification
  - Can fully use multicore systems

:: [CSD.IJS.SI

In our practice

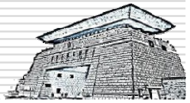
- Most of industrial optimization projects require simulation
  - When too complicated to made homemade simulator or when very high precision is needed we use commercial simulators running on our multicore cluster
    - This is limited to offline simulations with our computational resources
    - Or high performance computing system is needed on client's site (not always realizable)

Characteristics that PDES Engine should have

Yiping Yao

National University of Defense Technology

Changsha, P.R. CHINA.



Characteristics that PDES Engine should have

- “In science of the 21st century, simulation and high-end computation are equal partners with theory and experiment.”
- “SIMULATION AS A PEER METHODOLOGY TO EXPERIMENT AND THEORY ”。
- scientific questions in all fields are growing more complex and interconnected
- The simulation systems also are becoming more and more large and complex
- to solve the performance bottleneck, **using HPC Resources** is an inevitable trend for **Large Simulations**
- To bridge the gap between computing at the desktop and computing on HPC systems, HPC-based Simulation Engine should have the following Characteristics:

Characteristics that PDES Engine should have

- Object Oriented
  - Modelling in accord with the real word
  - Plane, ladar, Missile, command post ……
- Transparency on Parallelization
  - Parallel program is hard to develop and error prone
  - Simulation Engine should adapt the application to different communication infrastructure (shared memory, MPI and TCP/IP) automatically
  - User needn't consider programming with OpenMP、MPI、TCP/IP
- Transparency on event scheduling
  - Hide the implementation details of event scheduling between simulation objects
  - User needn't know where the object located and how the event queue implemented

Characteristics that PDES Engine should have

- Transparency on time management
  - Hide the implementation details of time management
  - User needn’t consider time management strategy(conservative or optimistic) during programming
  - time management strategy can be set as a parameter during runtime
- Transparency on Rollback
  - For optimistic strategy, user needn’t to take care of rollback
  - Simulation Engine can do rollback automatically
- Simulation objects can be dispatched to different nodes flexibly
  - Minimum the communication overhead
  - Balance the computation in different nodes
- Can run on both Windows and Linux system
  - Switch easily from Windows to Linux

Thanks!

Questions?