Interactive Simulation and Visualization in Medicine

Chris Johnson
Scientific Computing and Imaging Institute
School of Computing
University of Utah
Computational Science Pipeline

Construct a model of the physical domain *(Mesh Generation, CAD)*

Apply boundary conditions

Numerically approximate governing equations *(FE, FD, BE)*

Compute *(Preconditioners, Solvers)*

Visualize *(Isosurfaces, Vector Fields, Volume Rendering)*
Computational Science - Today

Modeling

Simulation

Visualization
Computational Science - Today
“Scientists not only want to analyze data that results from super-computations; they also want to interpret what is happening to the data during super-computations. Researchers want to steer calculations in close-to-real-time; they want to be able to change parameters, resolution or representation, and see the effects. They want to drive the scientific discovery process; they want to interact with their data....”
Computational Science - Tomorrow?

Modeling

Simulation

Visualization

user guides
Computational Science - Tomorrow?
If this is so great, why is it just starting to “catch on”?  

• Scientists greedy for CPU cycles  
• Faster machine - Larger problems  
• Different sets of expertise  
• It’s hard to make it all work well!
What if questions (a “computational workbench”)
Iterative design (medical device design)
Time-critical (diagnosis, surgery)
“Minor” Challenges

Accommodating parallelism
Large data sets
Complex physics/physiology
Existing code
3D user interaction
Efficiency
Fault tolerance
Etc., etc., etc.
Device Design: Defibrillation
Time-critical: Neurosurgery

Harvard & Brigham Women’s Hospital
Interactive Large-Scale Visualization

SCI Utah

Medical

Scientific Computing

GeoScience
Visualization at All Levels

Application level
- Streamlines, cutting planes, isosurfaces, surface maps, etc.

System level
- Module profiling
- Memory allocator visualization
- More in progress...
### Numerical Feedback

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration</td>
<td>237</td>
</tr>
<tr>
<td>Original Error</td>
<td>1</td>
</tr>
<tr>
<td>Current Error</td>
<td>9.43408e-05</td>
</tr>
<tr>
<td>Flop Count</td>
<td>8.97562e+07</td>
</tr>
<tr>
<td>MFlops</td>
<td>167.099</td>
</tr>
<tr>
<td>Memory bytes accessed</td>
<td>1.45805e+10</td>
</tr>
<tr>
<td>Memory bandwidth (MB/sec)</td>
<td>2942.4</td>
</tr>
</tbody>
</table>
Real-Time Ray Tracer
Maximum Intensity Projection
35 million spheres
Adaptive Finite Elements
Time-dependent Adaptation

SCI Utah
C-SAFE Uintah Network
ASCI
Blue Mountain
Los Alamos National Lab
Goals

Help foster interest/research in PSEs/Components
  • Computational Workbench

Help realize a common API for PSEs/Components
  • Common Component Architecture (CCA) Forum
    - www.acl.lanl.gov/cca-forum
More Information

crj@cs.utah.edu

www.cs.utah.edu/~sci
The frustration of using bad software...
SCIRun Ports

Requirements:
- OpenGL
- Tcl
- p-threads

Unix
- single and multiprocessors

PC - NT and Linux
SCIRun Availability

Not generally available yet

Approx. 10 beta users now

Research version available as soon as we finish documentation

Commercial license available from Visual Influence:

www.visualinfluence.com
Conclusions

Computational steering (interactive computing) can be a more efficient paradigm for iterative design problems and time-critical computational problems.
Future Work

Detachable User Interfaces
Distributed Memory Implementation
More Modules
New Applications
Finish Documentation!!
Acknowledgements

DOE ASCI
NSF PACI and PFF
SGI Visual Supercomputing Center
Utah Centers of Excellence
Visual Influence
Applications 1

Mark Ellisman – UCSD, NPACI, NCRR

• Linking expensive data acquisition devices – high resolution microscopes
• Compare data from the microscopes with data from simulation and databases
• Data size – $2K^3$ (will be $4K^3$ within a couple of years) – lots of computing – currently using distributed workstations using Globus
Mark Ellisman - cont

- Time critical because of mass loss
- Data -> Modeling -> Analysis -> Visualization -> Database -> Feedback (and feedforward) throughout

Could enable further science/applications with protein structures (and others)

Useful for extending (via simulation and/or experimentation) functional information within multilayer databases
Joel Saltz

U. Of Maryland, Johns Hopkins
Alpha Project (NPACI) with Mary Wheeler (UT Austin) on reservoir simulation

Tighten the loops between production information, sensor data and simulation data.

Satellite data, classification, visualization, large-scale data query and processing.
Patient specific diagnosis and treatment – need to access and register distributed data, integrate radiology, microscopy, pathology data.

Applications in drug delivery, interventional radiology, etc.
John Miller

Center for Computational Biology – Montana State

Figure out how the brain works

How information is encoded

Sensors – receive data, coupled with information from a large database, then via a combination of experimental and simulation data, control parameters to manipulate the system (the visual system, for example)
Massive data streams – analyze on the fly – use this data to interact with a model drawing parameters from databases – and do it VERY fast.
Long running simulations for stochastic differential equations – doesn’t need to be interactive.

Spinal cords – chips that stimulate spinal cords in adaptive way such that it can take sensory feedback and maintain a particular motor pattern.

Use an adaptive analog system

Understanding algorithms for input/output of systems
Carlos Felippa

Aerospace engineering
Multiphysics, embedded systems with real-time control
Reconfigurable systems
Heirarchical systems
Model systems and control
Robust against uncertainty
Figure out commonalities
Michael Creutz

Particle physicist
Long running computational jobs
Visualization not useful (yet)
Charbel Farhat
Univ. of Colorado – Aerospace engineering
Data driven embedded systems
Feedback control of embedded systems – need automatic system for a control
Autocallibration between experimental apparatus and simulation
Currently there are long (weeks) delays in market feedback/analysis

Situational awareness sensory data of Army troops – what to do with all the data, how to use the data for real-time response – how do you manage such situations
Abhi Deshmukh


Distribution systems of transportation networks, getting feedback from people on the road and planning shortest path

Using algorithms based upon how ants find food
Jacobo Bielak

CMU – seismology
Need a hierarchy of problems and methods/techniques
Some are real-time, some aren’t
What if questions
Design questions
Real-time (related to prediction and control)
Earthquake ground motion

Is there a design criteria based upon ground motion (not real-time at this time)

What if – how does the underlying structure relate to ground motion – does it depend upon the local of the source.

Why now? – more/better/cheaper sensors and integration with simulation

Distributed collaboration, steering, and visualization
U. Of Oklahoma
Center for Analysis and Prediction of Storms

Numerical weather prediction systems – large-scale (1000 CPUs)
Run at higher resolution (1K on a side)

Registration of multiple modalities of input data (radar, doppler, etc)
Actually works
Feedback – cycling using updated data

This is the right time to take the next step (entire US)
Robert Lodder
Univ. of Kentucky – cardiac catherization
Common Themes

Hardware Needs:
• Need more cycles
• Need more bandwidth

Software Needs:
Decision process

• Objective functions
• Value of information
• Treatment of uncertainty
• Perception
• Stopping rules
New Applications

Radio astronomy – 90 radio telescopes – full sky survey – pulsars, comets, variable stars – are time-dependent – need a tight connection between data collection, then move the specific type of telescope towards the location – must have dynamic link or experiment doesn’t work.
What Will DDDAS Enable?

Better weather prediction because of data feedback

Enable new level of physiological experiments because of the tight coupling between analysis and experiment – this would alter the way some experiments are done

Next level of embedded systems – ability to react to uncertain or unpredictable input
Why Now?

Leverage existing NSF programs
Think tactfully about implementation of new programs
Networking/interconnectivity, cycles, disks, and new algorithms are enabling new applications
New sensors/data is available
Dynamic Data-Driven Application Systems

Applications Group I

Chris Johnson  Greg McRae
Jacobo Bielak  Sandy Boyson
Janice Coen    Abhi Deskhmukh
Mark Ellisman  Robert Lodder
John Miller    Joel Saltz
Klaus Schulten Carlos Felippa
Avis Cohen     Charbel Farhat
Michael Creutz Daniel Weber
Bad Day

Anonymous ftp to:
sci2.cs.utah.edu
cd to /pub
Download badday.mpg
DDDAS Motivation

Reduce time to adapt to new conditions and to decide how to allocate resources to respond to the change

• Experiments on short-lived processes (e.g. physiology)
• Capture sporadic astronomic events
• Active control of structures during an earthquake
• Disturbances in a chemical plant
• Early warning systems (fire, tornado, earthquakes, hurricanes, pollution, floods)
• Financial and management systems (supply-chain coordination)
• Crisis management (terrorist attacks, epidemics)
• Adaptive structures (car suspension, buildings, space structures)
• Autonomous systems (decision processes)
• Interactive system analysis and control of experiments
• Predict extreme geospace conditions (space weather)
Real-time
Feedback and Control (closing the loop, robust)
How uncertainty controls the output and parameter selection (sensitivity analysis)
Model reduction
Relationships to sensors
Data Driven System Characteristics

Predictive modeling (combinations of hardware and software)

Better techniques to solve large-scale inverse problems (inverse correlation)

Relationships between space/time scales and measurements

Computational Workbenches
DDDAS Adaptive Observation

Infusing data into the simulation and improving the model for the next simulation

• Understand where errors are and understand where more data is needed
• Understand where to get the initial conditions
Why Now?

There is a convergence of computing, networking, algorithmic, sensor, software, and application technologies. Integration of these technologies affords taking “the next step” in many application areas.

- Can’t do the kinds of experiments unless one can interact with large systems (for example – neuroscience)
- Use simulation more than a posteri way – DDDAS can move us beyond that
- Now we have computational resources (hardware and software) to approach realistic problems
But...

Artificial department boundaries are an impediment to creating needed expertise

Computational Science at NSF is not well defined

Sociologically issues with regard to the interaction of theory, experiment, and computation

Education/training a large issue for computational scientists
Enabling Technologies

Model Building
Algorithms
Sensors
Computational systems
Visualization and analysis
Database management systems
Communications
Integration software
Algorithms

Mathematical development

- Improved Bayesian methods for model-based experimental design, parameter estimation, state estimation, sensor placement
- Inverse methods for large-scale integro-partial differential equation
- Identification of time-varying systems
- Uncertainty propagation
- Time-series analysis
- Solution of large-scale nonlinear programming problems
Sensors/Actuators

Can dramatically change the way one looks at a problem, but requires interaction across many disciplines to build and use them, e.g.

- Chemical lab on a chip
- Molecular markers
- Noninvasive (and very invasive) physiological monitoring
- Microelectronics (smart materials)
- Remote sensing
- Adaptive optics (multiple mirror telescopes)
- Particle tracking
- Damage detection

All have high data rates
Interactive visualization techniques for large data

- Graphical user interface design
- Haptics, visual and other feedback mechanisms
- Scientific and higher dimensional data streams
- Distributed collaborative visualization (workstation and VR)
- Remote visualization (compression, view dependent, perception-based)
Need to interact and manage large data

- Visual databases
- Distributed databases
- Interaction
- Legacy (heritage) databases
- Develop of tools for supporting interactive dataset manipulation
- Tools to couple simulations to databases
- Merging different measurements of the same process (e.g. registration)
Communication (between humans and machines) infrastructure to facilitate interaction (both locally and remotely) and to expand the potential for collaboration (between humans)

- Bandwidth (more) management (connect adaptively to systems)
- Compression technologies (feature detection, multiresolution, etc.)
- Fast wireless and distributed sensors
- Sensors that send out upon need and/or demand
- Smart sensors that compute locally and send updated/changed information
Integration Software

Encourage open source
Common API (to software and to sensors)
Common component software architecture
Dealing with heritage codes
Role of filters and wrappers (scripting languages, etc.)
Related Reports and Initiatives

1998 NSF Workshop on PSEs (Abdali)
1998 DOE Report on Large Data Visualization
1999 NIH Report on Biomedical Computing

Model Based Simulation – caswww.colorado.edu/MBS.Workshop.d/index.html

DOE ASCI Program
PITAC Report
Industry Relations

Students (although we need cooperative programs to allow students to finish degrees)

Spawn new industries and multi-industry collaborations

Tighter connections between industry data output and use in academic models/simulations (airlines, weather, FAA example)

Pricing models based upon need/consumption
Implementation

DDDAS is cross/multi-disciplinary in nature!
Don’t implement in ITR
Cross directorate reviewing required
Need to figure out computational science within NSF
Need all directorates on board
Need LOTS of $$
Some projects beyond the current 3-5 year limits
Balance the risk portfolio to include more speculative endeavors
Dynamic Data-Driven Application Systems

Applications Group I

Chris Johnson  Greg McRae
 Jacobo Bielak  Sandy Boyson
 Janice Coen  Abhi Deskmukh
 Mark Ellisman  Robert Lodder
 John Miller  Joel Saltz
 Klaus Schulten  Carlos Felippa
 Avis Cohen  Charbel Farhat
 Michael Creutz  Daniel Weber