

An Overview of Computational Science

(Based on CSEP)

Craig C. Douglas

January 15, 2002

CS 521, Spring 2002

What Is Computational Science?

- There is **no accepted definition!!!** (yet)
- **Ken Wilson's definition, circa 1986:** A common characteristic of the field is that problems...
 - Have a precise mathematical model.
 - Are intractable by traditional methods.
 - Are highly visible.
 - Require in-depth knowledge of some field in science, engineering, or the arts.
- Computational science is neither computer science, mathematics, some traditional field of science, engineering, a social science, nor a humanities' field. **It is a blend.**

Ken Wilson's Four Questions

- Is there a computational science community?
 - Clearly yes (or why would you be taking this course?)
- What role do grand challenge problems play in defining the field?
 - Initially the grand challenge problems were the entire field.
 - Now they are minor issues for bragging purposes.
 - However, if you solved one of the early ones, you became famous.
- How significant are algorithm and computer improvements?
 - What is the symbiotic relationship between the two?
 - Do you need one more than the other?
- What languages do practitioners speak to their computers in?
 - Fortran (77, 90, 95, or 2000), C, C++, Ada, or Java

Is There a Computational Science Community?

- Computational science projects are always **multidisciplinary**.
 - Applied math, computer science, and...
 - One or more science or engineering fields are involved.
- Computer science's role tends to be
 - A means of getting the low level work done efficiently.
 - Similar to mathematics in solving problems in engineering.
 - **Oh, yuck...** a service role if the computer science contributors are not careful.
 - **Provides tools** for **data manipulation, visualization, and networking**.
- Mathematics' role is in providing analysis of (new?) numerical algorithms to solve the problems, even if it is done by computer scientists.

New Field's Responsibilities

- Computational science is still an evolving field
 - There is a common methodology that is used in many disparate problems.
 - Common tools will be useful to all of these related problems if the common denominator can be found.
- The field will really be unique once it solves some small collection of problem for which there is clearly no other solution methodology.
- The community is still trying to define the age old question, “What defines a high quality result?” This is slowly being answered.
- An education program must be devised. This, too, is being worked on.
- Appropriate journals and conferences already exist and are being used to guarantee that the field evolves.
- Various government programs throughout the world are pushing the field.

Grand Challenge Problems

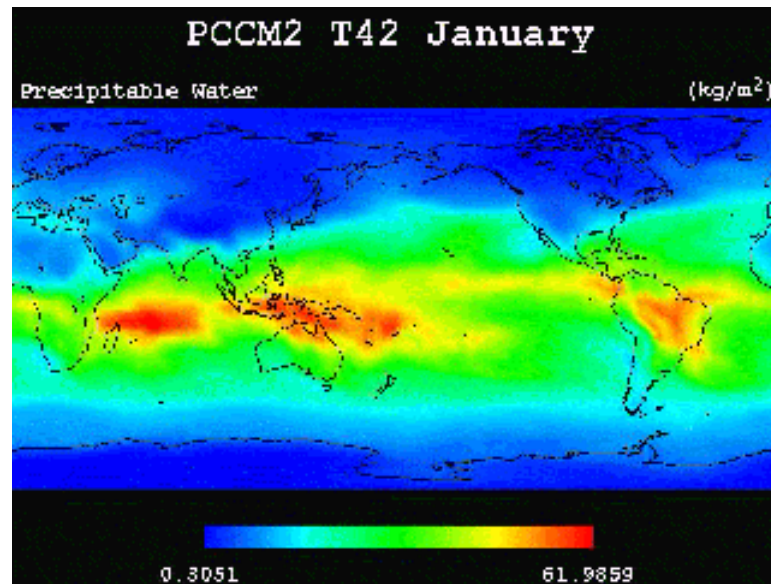
- New fields historically come from breakthroughs in other fields that resist change.
- **Definition:** Grand Challenges are fundamental problems in science and engineering with potentially broad social, political, economic, and scientific impact that can be advanced by applying high performance computing resources.
- Grand Challenges are dynamic, not static.
- **Grand Challenge problems** are defining the field. There is great **resistance** in mathematics and computer science to these problems. Typically, the problems are defined by **infidels from applied science and engineering fields** who do not provide sufficient applause to the efforts of mathematicians and computer scientists. The **infidels** just want to solve (ill posed) problems and move on.

Some Grand Challenge Areas

- Combustion
- Electronic structure of materials
- Turbulence
- Genome sequencing and structural biology
- Climate modeling
 - Ocean modeling
 - Atmospheric modeling
 - Coupling the two
- Astrophysics
- Speech and language recognition
- Pharmaceutical designs
- Pollution tracking
- Oil and gas reservoir modeling

A Grand Challenge Example

- CHAMMP: <http://www.epm.ornl.gov/champp/champpions.html>
 - Oak Ridge and Argonne National Labs and NCAR collaborated to improve NCAR's Community Climate Model (CCM2).
 - A sample visualization of a computer run:



How Significant are Algorithm and Computer Improvements?

- There is a race to see if computers can be speeded up through new technologies faster than new algorithms can be developed.
- Computers have doubled in speed every 18 months over many decades. The **ASCI** program is trying to drastically reduce the doubling time period.
- Some algorithms cause quantum leaps in productivity:
 - **FFT** reduced solve time from $O(N^2)$ to $O(N\log N)$.
 - **Multigrid** reduced solve times from $O(N^{3/2})$ to $O(N)$, which is **optimal**.
 - Monte Carlo is used when no known reasonable algorithm is available.
- Most parallel algorithms do not linearly reduce the amount of work.
- A common method of speeding up a code is to wait three years and buy a new computer that is four times faster and no more expensive than the current one.

Three Basic Science Areas

- Theory
 - Mathematical modeling.
 - Physics, chemistry, engineering principals incorporated.
- Computation
 - Provide input to what experiments to try.
 - Provide feedback to theoreticians.
 - Two way street with the other two areas.
- Experimentation
 - Verify theory.
 - Verify computations. Once verified, computations need not be verified again in similar cases!

Why Computation?

- Numerical simulation fills a gap between physical experiments and theoretical approaches.
- Many phenomena are too complex to be studied exhaustively by either theory or experiments. Besides complexity, many are too expensive to study experimentally, either from a hard currency or time point of view.
- Computational approaches allow many outstanding issues to be addressed that cannot be considered by the traditional approaches of theory and experimentation alone.
- Problems that computation is driving as the state of the art will eventually lead to computational science being an accepted, new field.

What computer languages?

- Fortran
 - 77, 90, and 95 are all common with 2000 on its way.
 - Fortran 9x compilers tend to produce much slower code than Fortran 77 compilers do. There are tolerable free Fortran 77 compilers whereas all Fortran 9x compilers are somewhat costly.
 - Fortran 90 is the de facto standard language in western Europe.
- C
 - Starting to become the language of choice.
- C++
 - US government labs pushing C++.
- Ada
 - US department of defense has pushed this language for a number of years.
 - C++ is replacing it slowly in new projects.

Parallel Languages

- While there are not too many differences between most Fortran and C programs doing the same thing, this is not always true in parallel Fortran variants and parallel C variants.
- High Performance Fortran (HPF), a variant of Fortran 90, allows for parallelization of many dense matrix operations trivially and quite efficiently. Unfortunately, most problems do not result in dense matrices, making HPF an orphan.
- Many parallel C's can make good use of C's superior data structure abilities. Similar comments can be said about parallel C++'s.
- MPI and OpenMP work with Fortran, C, and C++ to provide portable parallel codes for distributed memory (MPI) or shared memory (OpenMP) architectures, though MPI works well on shared memory machines, too. Both require the user to do communications in an assembly language manner.

Three Styles of Parallel Programming

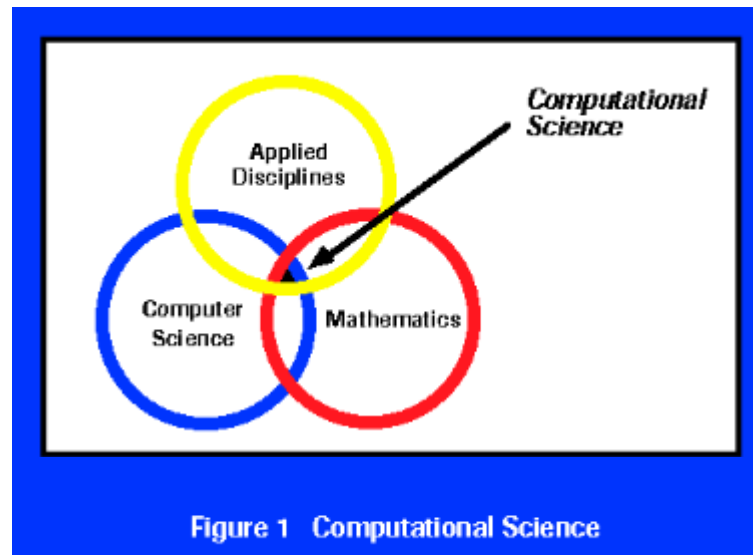
- **Data parallelism**
 - Simple extensions to serial languages to add parallelism.
 - These are the easiest to learn and debug.
 - HPF, C*, MPL, pc++, OpenMP, ...
- **Parallel libraries**
 - PVM, MPI, P4, Linda, ...
- **High level languages with implicit parallelism**
 - Functional and logic programming languages.
 - This requires the programmer to learn a new paradigm of programming, not just a new language syntax.
 - Adherents claim that this is worth the extra effort, but others cite examples where it is a clear loser.
- Computational science is splintered over a programming approach and language of choice.

Computational Science Applications

- Established
 - CFD
 - Atmospheric science
 - Ocean modeling
 - Seismology
 - Magnetohydrodynamics
 - Chemistry
 - Astrophysics
 - Reservoir & pollutant tracking
 - Nuclear engineering
 - Materials research
 - Medical imaging
- Emerging
 - Biology/Bioinformatics
 - Economics
 - Animal Science
 - Digital libraries
 - Medical imaging

Computational Scientist Requirements

- Command of an applied discipline.
- Familiarity of leading edge computer architectures and data structures appropriate to those architectures.
- Good understanding of analysis and implementation of numerical algorithms, including how they map onto the data structures needed on the architectures.
- Familiarity with visualization methods and options.



Current Trends in Architectures

- Parallel supercomputers

- Multiple processors per node with shared memory on the node (a node is a motherboard with memory and processors on it).
- Very fast electrical network between nodes with direct memory access and communications processors just for moving data.
- SGI Origin 3000, IBM SP4, SUN Enterprise, HP Superdome.
- Cluster of PC's
- Take many of your favorite computers and connect them with a fast ethernet running 100-1000 Mbs.
- Usually runs Linux, IRIX, True64 UNIX, HP-UX, AIX-L, Solaris, or Windows NT with MPI and/or PVM.
- Intel, Alpha, or SPARC processors. Intel is most common in clusters of cheap micros.

Peak Speeds of Selected Computers

Machine	Mflops (per processor)
CDC 6600	3
Cray 1	160
Cray C90	1000
Compaq Alpha 21264	1000
Cray T90	2200
Intel Pentium 4 (2200MHz)	2200
AMD Athlon XP 1900+	3200
HP PA 8700	3200
Intel Itanium	3400
Sony PlayStation 2	6300

Network Speeds

		Transmission Time (seconds)		
Name	Speed (bps)	24 bit screen	Bible	Encyclopedia Britannica
T3	45M	0.5	1.2	60
Cable modem	30M/10M/2M	3	6	270
T1	1.544M	15	36	30 min
56kbs	53,000	7 min	1 hour	2 days

NSF Supercomputing Program (PACI/Terascale)

- NCSA
 - 1536 SGI Origin 2000 processors (NCSA).
 - 1500 IA32 processor cluster (NCSA).
 - 128-256 processor PC clusters (NCSA, UNM, UW).
 - 192 SGI Origin 2000 processors (BU).
 - 224 HP Superdome processors (UK).
- SDSC (U. San Diego)
 - IBM SP3 (1.5 Tflops).
 - Cray T90, 16 processors.
 - Sun Fire Server.
- PSC (Pittsburgh)
 - 3000 Compaq AlphaServer processors: 6 Tflops.
- DTF (NCSA, SDSC, Caltech, Argonne National Lab)
 - 6800 Intel McKinley processors (IBM): 13 Tflops.

ASCI Program (DOE)

- Sandia National Lab (Sandia)
 - 9000+ Pentium Pro processors. Most successful ASCI machine to date!!!
 - 100 Tflop machine, joint with Celera Genomics (Compaq).
- Los Alamos National Lab (LANL)
 - 6188 SGI Origin 2000 processors.
 - Buying a new machine from Compaq.
- Livermore National Lab (LLNL)
 - 5088 IBM SP PowerPC 604e processors.
 - 8192 IBM SP3.
- Lawrence Berkeley National Lab (LBL)
 - 644 Cray T3E-900 processors.
 - 2048 IBM SP3 processors.