Introduction:
Robust and High Performance Tools for Scientific Computing

Tony Drummond
Lawrence Berkeley National Laboratory
ACTS Collection Workshop
August 5-8, 2003
Motivation

Grand Challenges are fundamental problems in science and engineering, with potentially broad social, political, and scientific impact, that could be advanced by applying high performance computer resources.

Office of Science and Technology

- Some grand challenges: electronic structure of materials, turbulence, genome sequencing and structural biology, global climate modeling, speech and language studies, pharmaceutical design, pollution, etc.
Motivation

With the development of new kinds of equipment of greater capacity, and particularly of greater speed, it is almost certain that new methods will have to be developed in order to make the fullest use of this equipment. It is necessary not only to design machines for the mathematics, but also to develop a new mathematics for the machines - 1952, Hartree

• **Metropolis** grew out of physical chemistry in 1950's through attempts to calculate statistical properties of chemical reactions. Some areas of application: astrophysics, many areas engineering, and chemistry

• **Fast Fourier Transform (FFT)**: implementation of Fourier Analysis. Some areas of application: image and signal processing, seismology, physics, radiology, acoustics and engineering

• **Multigrids**: method for solving a wide variety of PDE. Some areas of application: physics, biophysics and engineering
Motivation

**Computational science:** can be characterized by the needs to gain understanding through the analysis of mathematical models using high performing computers.

**Community:**
- Scientists
- Engineers
- Mathematicians
- Economists, artists

**Multidisciplinary!**

**Computer Science**
Provides services ranging from networking and visualization tools to algorithms

**Mathematics:**
Credibility of algorithms (error analysis, exact solutions, expansions, uniqueness proofs and theorems)
Some lessons learned from Earth System Modeling
Motivation - Example I

SPECTRUM OF ATMOSPHERIC PHENOMENA

PLANETARY SCALE
SYNOPTIC SCALE
MESO-SCALE
DEEP CONVECTION
SHALLOW CONVECTION

TURBULANCE
LARGE INERTIAL EDDIES
VISCOUS SUBRANGE

10^6 km  10^3 km  10^2 km  10 km  1 km  10^2 m  10 m  1 m  1 dm  1 cm  1 mm

GCM
Motivation - Example I

- **CCM3** - spectral truncations of T170 and T239
- 50 Km spatial resolution is 32 times more grid cells and takes roughly **200 times longer** to run
Motivation - Example II

- Non-linear demand for resources (CPU - Memory)
- Multi-disciplinary application is more demanding
Using today's hardware to tackle today's Grand Challenges

---

Q. Why is it still difficult to obtain High Performance?
Some common and *interesting* answers

- Technology
- Memory latency
- Algorithms
- Programming Practices

...
### Some options for New Architectures

<table>
<thead>
<tr>
<th>OPTION</th>
<th>SOFTWARE IMPACT</th>
<th>COST</th>
<th>TIMELINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modification of commodity processors</td>
<td>Minimal</td>
<td>2 or 3 times commodity?</td>
<td>Can be achieved in a few years</td>
</tr>
<tr>
<td>U.S.-made vector architecture</td>
<td>Moderate</td>
<td>2 or 3 times commodity at present</td>
<td>Available now</td>
</tr>
<tr>
<td>Processor-in-memory (Blue Gene/L)</td>
<td>Extensive</td>
<td>Unknown, 2 to 5 times commodity?</td>
<td>Only prototypes available now</td>
</tr>
<tr>
<td>Japanese-made vector architecture</td>
<td>Moderate</td>
<td>2.5 to 3 times commodity at present</td>
<td>Available now</td>
</tr>
<tr>
<td>Research Architectures (Streams, VIRAM..)</td>
<td>Extensive or unknown</td>
<td>Unknown</td>
<td>Academic research prototypes only available now</td>
</tr>
</tbody>
</table>
Memory Hierarchy

Where is the data? Why is data locality important?

- CPU
  - Registers
  - On-Chip Cache
- SRAM
- Main Memory
  - DRAM
  - Secondary Storage (Disk),
  - Distributed Memory
- Tertiary Storage (Disk or Tape),
  - Remote Cluster Memory

<table>
<thead>
<tr>
<th>Speed</th>
<th>1's ns</th>
<th>10's ns</th>
<th>100's ns</th>
<th>1's -10's ms</th>
<th>10's s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>100's bytes</td>
<td>Kbytes</td>
<td>Mbytes</td>
<td>Gbytes</td>
<td>Tbytes</td>
</tr>
</tbody>
</table>

8/4/03
Memory Latency

Hybrid-Model

Shared Memory

Distributed Memory

Different interconnection mechanisms
The GRID

• A large pool of resources
  • Computers
  • Networks
  • Software
  • Databases
  • Instruments
  • people

Requirements from GRID implementation:
• Ubiquitous: ability to interface to the grid at any point and leverage whatever is available
• Resource Aware: manage heterogeneity of resources
• Adaptive: tailored to obtain maximum performance from resources
• Since 1980’s, CPU performance has increased at a rate of almost 60%/year

Grows 50%/year

• Since 1980’s, DRAM (latency) has improved at a rate of almost 9%/year
### Parallel Programming Paradigms

#### Shared Memory

<table>
<thead>
<tr>
<th>Memory</th>
<th>P1</th>
<th>P2</th>
<th>...</th>
<th>Pn</th>
</tr>
</thead>
</table>

- **Data parallelism**
  - easier to implement
  - shared memory space
  - mutual exclusion, contention

- **Message Passing**
  - shared area is used for sending and receiving data

#### Distributed Memory

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>...</th>
<th>Mn</th>
</tr>
</thead>
</table>

| P1 | P2 | ... | Pn |

- **virtual shared memory**
  - data is implicitly available to all

- **Implicit mutual exclusion**
- **Only explicit synch**
- Depends on Memory Hierarchy and Network
Large Scientific Codes: A Common Programming Practice

Algorithmic Implementations

Application Data Layout

Control

I/O

Tuned and machine Dependent modules
Shortcomings

New Architecture:
- May or may not need re-rewriting

New Developments:
- Difficult to compare

New Architecture:
- Extensive re-rewriting

New or extended Physics:
- Extensive re-rewriting or increase overhead

New Architecture or S/W:
- Extensive tuning
- May require new programming paradigms
- Difficult to maintained!
"We need to move away from a coding style suited for serial machines, where every macrostep of an algorithm needs to be thought about and explicitly coded, to a higher-level style, where the compiler and library tools take care of the details. And the remarkable thing is, if we adopt this higher-level approach right now, even on today's machines, we will see immediate benefits in our productivity."

*Numerical Recipes: Does This Paradigm Have a future?*
Alternative Programming Approach

USER's APPLICATION CODE
(Main Control)

AVAILABLE
Application
Data Layout
LIBRARIES & PACKAGES

AVAILABLE
Algorithmic
Implementations
LIBRARIES & PACKAGES

AVAILABLE
I/O
LIBRARIES

Tuned and machine
Dependent modules
Software Development
Levels of abstraction

- Scientific or engineering context
  - Domain expertise
    - Simulation codes
    - Data Analysis codes
- Templates
- Scientific Computing Tools
- General Purpose Libraries
- Data Structures
- Algorithms
- Code Optimization
- Programming Languages
- O/S - Compilers

Hardware - Middleware - Firmware
<table>
<thead>
<tr>
<th>Method/Refinement</th>
<th>Climate Change</th>
<th>Material Science</th>
<th>High Energy Physics</th>
<th>Astrophysics Cosmology</th>
<th>Biology</th>
<th>Chemistry</th>
<th>Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monte Carlo (Quantum and Classical)</td>
<td>PCM CCSM POP</td>
<td>Quantum MC Classical KMC</td>
<td>FASTER SYNPOL</td>
<td>FASTER SYNPOL</td>
<td>NWChem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fast Fourier Transform</td>
<td></td>
<td>VASP Paratec Petot Escan</td>
<td>IMPACT LANGEVIN3D MAD9P ccSHT</td>
<td>SPIDER NAMD</td>
<td>NAMD</td>
<td>WARP GTC</td>
<td></td>
</tr>
<tr>
<td>Fast Multipole &amp; Variants</td>
<td>Classical MD</td>
<td>IMPACT LANGEVIN3D QuickPIC</td>
<td>Classical MD</td>
<td>NWChem Classical MD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sparse Linear systems</td>
<td>PCM CCSM POP</td>
<td>O(N) Methods</td>
<td>OMEGA3P</td>
<td>SPIDER</td>
<td>pVarDen</td>
<td>NIMROD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DFT FLAPW PW codes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DFT</td>
<td></td>
<td>NWChem Gaussian QChem</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dense Linear Solvers</td>
<td>LSMS FLAPW</td>
<td></td>
<td>MADCAP</td>
<td>NWChem Gaussian</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptive Mesh Refinement</td>
<td>BoxLib Paramesh</td>
<td>BoxLib Paramesh</td>
<td>FLASH Paramesh</td>
<td>pVarDen BoxLib</td>
<td>WARP BOX Chombo</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8/4/03
## Partial Matrix of Methods and Disciplines

<table>
<thead>
<tr>
<th></th>
<th>Climate</th>
<th>Material</th>
<th>High Enregy</th>
<th>Astrophysics</th>
<th>NMROD</th>
<th>Chemistry</th>
<th>Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monte Carlo</strong></td>
<td>POP</td>
<td>VASP</td>
<td>IMPACT</td>
<td>SPIDER</td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td>(Quantum &amp; Classical)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td><strong>Fast Fourier Transform</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td><strong>Fast Multipole &amp; Variants</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td><strong>Sparse Linear Systems</strong></td>
<td>PCM</td>
<td>O(N)</td>
<td>OMEGA3P</td>
<td>SPIDER</td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCSM POP</td>
<td>Methods</td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td><strong>Eigenvalue Solvers</strong></td>
<td>PCM</td>
<td>OMEGA3P</td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POP</td>
<td>OMEGA3P</td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td><strong>Dense Linear Solvers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td><strong>Adaptive Mesh Refinement</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NWChem</td>
<td></td>
</tr>
</tbody>
</table>

**NWChem**: Uses Global Arrays for implementing distributing computing environments

**MADCAP**: Uses ScaLAPACK for the solution of large and dense linear systems of equations

**FLAPW**: uses ScaLAPACK for the solutions Dense Eigenvalue Problems
What is the DOE ACTS Collection?

http://acts.nersc.gov

• Advanced CompuTational Software
• Tools for developing parallel applications
  • Developed (primarily) at DOE Labs
  • Separate projects originally
  • ~ 20 tools
• ACTS is an “umbrella” project
  • Leverage numerous independently funded projects
  • Collect tools in a toolkit
ACTS: Project Goals

- Extended support for experimental software
- Make ACTS tools available on DOE computers
- Provide technical support (acts-support@nersc.gov)
- Maintain ACTS information center (http://acts.nersc.gov)
- Coordinate efforts with other supercomputing centers
- Enable large scale scientific applications
- Educate and train
Related Activities

Software Repositories:
  - **Netlib**: [http://www.netlib.org](http://www.netlib.org)
  - **HPC-Netlib**: [http://www.nhse.org/hpc-netlib](http://www.nhse.org/hpc-netlib)
  - **National HPCC Software Exchange NHSE**: [http://www.nhse.org](http://www.nhse.org)
  - **MGNet**: [http://www.mgnet.org](http://www.mgnet.org)
  - **OO Numerics**: [http://oonumerics.org/oon](http://oonumerics.org/oon)

Portable timing routines, tools for debugging, compiler technologies:
  - **Ptools**: [http://www.ptools.org](http://www.ptools.org)
  - **Center for Programming Models for Scalable Parallel Computing**: [http://www.pmodels.org](http://www.pmodels.org)

Education:
  - **Computational Science Educational Project**: [http://csep1.phy.ornl.gov](http://csep1.phy.ornl.gov)
  - **U C B’s Applications of Parallel Computers**: [http://www.cs.berkeley.edu/~demmel/cs267_Spr99](http://www.cs.berkeley.edu/~demmel/cs267_Spr99)
  - Dictionary of algorithms, data structures and related definitions: [http://www.nist.gov/dads](http://www.nist.gov/dads)
Why is ACTS unique?

- Extended support for tools
- Accumulates the expertise and user feedback on the use of the software tools and scientific applications that used them:
  - independent software evaluations
  - participation in the developer user groups e-mail list
  - presentation of a gallery of applications
  - leverage between tool developers and tool users
  - workshops and tutorials
  - tool classification
  - support

8/4/03
The DOE ACTS Collection

The DOE ACTS (Advanced Computational Software) Collection is a set of DOE-developed software tools that make it easier for programmers to write high performance scientific applications for parallel computers. This site is the central information center for the ACTS Collection and is brought to you by NERSC and the Mathematics, Information, and Computational Sciences (MICS) Division of DOE. Correspondence regarding the collection (including requests for support) should be directed to acts-support@nersc.gov.

http://acts.nersc.gov
How much effort is involved in using these tools?
When a tool is not available at your site. . .

• Download the tools (Freeware!)

• Most of the tools support many of the available computational platforms (even Windows!)

• Follow installation instructions (some tools provide “configuration scripts”)

Using the ACTS Collection
• Most of the tools provide interfaces (calling functions and subroutines) from Fortran and C (some even C++)

CALL BLACS_GET( -1, 0, ICTXT )
CALL BLACS_GRIDINIT( ICTXT, 'Row-major', NPROW, NPCOL )
: CALL BLACS_GRIDINFO( ICTXT, NPROW, NPCOL, MYROW, MYCOL )
: :
: CALL PDGESV( N, NRHS, A, IA, JA, DESCA, IPIV, B, IB, JB, DESCB, $ INFO )
Using the ACTS Collection

- \texttt{-ksp\_type} [\texttt{cg},\texttt{gmres},\texttt{bcgs},\texttt{tfqmr},...]
- \texttt{-pc\_type} [\texttt{lu},\texttt{ilu},\texttt{jacobi},\texttt{sor},\texttt{asm},...]

More advanced:

- \texttt{-ksp\_max\_it} <\texttt{max\_iters}>
- \texttt{-ksp\_gmres\_restart} <\texttt{restart}>
- \texttt{-pc\_asm\_overlap} <\texttt{overlap}>
- \texttt{-pc\_asm\_type} [\texttt{basic},\texttt{restrict},\texttt{interpolate},\texttt{none}]
- Many more (see manual)
Using the ACTS Collection

Linear System Interfaces

GMG, ... FAC, ... Hybrid, ... AMGe, ... ILU, ...

Data Layout

structured composite block-struc unstruc CSR
• Best approach is to start with examples for beginners!

• Several efforts are targeting Tool Interoperability!
What needs to be computed?

- ScaLAPACK
- Aztec/Trilinos
- SuperLU
- PETSc
- Hypre
- TAO
- OPT++
- SUNDIALS

**Ax = b**

**Az = \[\cdot\]**

\[
\min \left\{ \frac{1}{2} \| r(x) \|^2 : x_l \leq x \leq x_u \right\}
\]

\(A = U\square V^T\)

- PDEs
- ODEs
What codes are being developed?

Global Arrays
- Parallel programs that use large distributed arrays
- Support for Grids and meshes
- Language Interoperability
- Infrastructure for distributed computing
- Performance analysis and monitoring
- Coupling distributed applications
- On-line visualization and computational steering

Overture
- Chasm
- PAWS
- CUMULVS
- Globus
- TAU

8/4/03
Tool Interoperability
Tool-to-Tool

PETSc

TAU

Ex 1

Ex 2
Component Technology!

Tool A

Tool B

Tool C

Tool D

CCA
% run 4

User: nkang
Repo: mpccc
Job Name: <none specified>
Group: mpccc
Class of Service: interactive
Job Class: interactive
Job Accepted: Wed Jul 30 09:42:16 2003

submit: Processed command file through Submit Filter: "/usr/common/nsg/etc/sub filter".

>>> import sys
>>> sys.path.append("/u6/nkang/kn/pyacts_1/build/lib.aix-5.1-mpi-2.2")

>>> import scalapack

scalapack.ex2("ex2_mat","ex2_rhs","sol",6,1,2,2,2,1)

Scalapack Example Program #2 (C-version) -- 07/24/2003
Solving AX=B
where A is a 6 by 6 matrix,
B is a 6 by 1 matrix,
with a block size of 2
Running on 4 processes, where the process grid is 2 by 2
INFO code returned by PDGESV = 0

According to the normalized residual the solution is correct.

||AX-B|| / ||A||*||X||*eps*N) = 1.25878215e-01

The solution is written to file sol

End of test.
This weeks agenda!
<table>
<thead>
<tr>
<th>Tuesday  Aug 5</th>
<th>Wednesday Aug 6</th>
<th>Thursday Aug 7</th>
<th>Friday Aug 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Computational Environments</td>
<td>Invited Talk</td>
<td>Invited Talk</td>
<td>Support for Computational Environments</td>
</tr>
<tr>
<td></td>
<td>Solution of Linear Systems (direct) and Eigenvalue problems</td>
<td>Numerical Optimization</td>
<td></td>
</tr>
<tr>
<td>Support for PDEs</td>
<td>Support for PDEs</td>
<td>Numerical Grid/Mesh Manipulation</td>
<td></td>
</tr>
<tr>
<td>Numerical Optimization</td>
<td>Grid Mesh/Mesh Manipulation</td>
<td>Remote Steering and Visualization</td>
<td>CCA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Performance And Tuning</td>
</tr>
</tbody>
</table>
acts-support@nersc.gov
http://acts.nersc.gov