A SciDAC Performer’s View of the Ever-Growing Challenge of Building and Maintaining Scientific and Engineering Software

Bob Lucas
USC – Lockheed Martin Quantum Computing Center
Information Sciences Institute
University of Southern California

Support for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research
Increasing H/W Complexity in Pursuit of Higher Performance

- **In the beginning, single system images, von Neumann architectures**
  - Languages reflect underlying machine (Fortran is the IBM 604, C the DEC PDP-11)
  - Intel 386 had a flat memory, no cache

- **Vector mainframes**
  - 90/10 rule bounded changes to codes like NASTRAN
  - Others like LLNL’s DYNA had to be reengineered from scratch
  - Still others, like SPICE, didn’t make it, even if they could have

- **Shared Memory Multiprocessors**
  - IBM mainframes, then X-M/P
  - Relative simple task-based concurrency, prior art for OpenMP

- **Distributed memory**
  - Hypercubes, initially “science projects”
  - Scalable distributed memory capability systems (e.g., ASCI Red) and MPI
  - Beowulf, the ubiquitous “lowest common denominator”

- **Today’s hybrid systems, with accelerators**
  - Tens of thousands of multicore nodes, in distributed memory systems, with accelerators
  - Diminishing number of credible applications
Increasing S/W Complexity in Pursuit of Science

- **There is no requirements limit for computing in science, engineering, or defense**
  - Any such suggestion merely gauges the imagination of the commentator
- **Multiple physical phenomenon**
  - Combustion combines structures, fluids, chemistry, and heat transport
- **Increased scale of models**
  - Bigger objects and finer meshes
  - Solvers often scale super linearly
- **Multiple scales in geometry and time**
  - Cracking of materials ranges form the macroscopic to the atomic level.
- **Established codes need to evolve**
  - Their users trust them, so they can’t be discarded lightly
  - New features must be added to stay relevant
  - Performance porting to new architectures
- **New codes need to be written**
  - Respond to new scientific discoveries or challenges
  - Sometimes the established ones can’t make the next leap
False Economy

• In 1988, a Cray 2 cost ~$20M
  – It shipped with three people
  – People were cheap

• Moore’s Law drove down the cost of hardware
  – Hardware cost relative to capability that is

• No corresponding decrease in the cost of software
  – Programmer productivity only marginally improved
  – The cost of labor has increased

• More complicated algorithms
  – Requires more, highly specialized labor
  – We don’t have really have control of this, we want better algorithms

• More complicated execution models
  – Requires more, increasingly specialized labor
  – We cede control of the execution model to our detriment
Massive Scaling (Tianhe-2 has 3M cores)
Divergent “Swim Lanes”

New Phenomenon
Adiabatic Quantum Annealing

Application Specific Systems
D.E. Shaw Research Anton
• SciDAC has always had a performance oriented project
  – This is because performance has always been a well established problem

• SciDAC-1 Performance Evaluation Research Center (PERC)
  – Benchmarking
  – Performance analysis
  – Performance modeling

• SciDAC-2 Performance Engineering Research Institute (PERI)
  – Performance engineering
  – Automatic performance tuning

• SciDAC-3 Institute for Sustained Performance, Energy, and Resilience (SUPER)
  – Performance measurement and analysis
  – Performance tuning, including further automation
  – Energy consumption
  – Resilience
Many Forms of SciDAC Application Engagement

• SciDAC explicitly encourages engagement of scientists and performance experts
  – SUPER can contribute expertise the design of new codes
  – Performance optimization and portability for existing codes

• Science Application Partnerships
  – Math and computer scientists incorporated into teams
  – Similar to small-scale ASC code teams

• SciDAC Institutes
  – Explicitly working with other Institutes, SAPs, and DOE science teams
  – Science teams provide context for our research
  – Science teams benefit from improvements we demonstrate
  – Not just SUPER, but true of all four SciDAC-3 institutes

• SUPER Liaisons
  – Long-term, one-on-one relationships with application scientists
  – Often built on prior relationships and proximity

• PERI Tiger Teams
  – Intensive, ephemeral interactions where many PERI investigators simultaneously focused on the same DOE science code’s performance issues
65x and 1.6x Speedup for CCSD and Fock matrix for NWChem

PI: Chris Cramer BES/ASCR Collaboration

**Objectives**

**Impact**
- Faster time-to-solution enabling larger and more accurate excited-state simulations with NWChem.

**Progress & Accomplishments**

SUPER Institute collaboration implemented OpenMP parallelism for two NWChem modules
- Native mode optimization to prepare for next-generation NERSC8 Cori
- Threading is essential to exploit full capability of MIC architecture

Performance of triples part of CCSD(T) improved 65x over original flat MPI implementation
- Flat MPI constrained to single process because of memory limitation

Performance of Fock matrix construction improved 1.64x over original flat MPI
- Flat MPI constrained to 60 MPI processes

**Table:**

<table>
<thead>
<tr>
<th>Total No. of Thread Contexts</th>
<th>No. of MPI Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>1 2 3 4 6 8 12</td>
</tr>
<tr>
<td>120</td>
<td>1.00 1.03 1.03 1.03 1.01 0.91 0.83</td>
</tr>
<tr>
<td>180</td>
<td>1.32 1.42 1.42 1.48 1.42 1.32 1.25</td>
</tr>
<tr>
<td>240</td>
<td>1.35 1.48 1.54 1.54 1.54 1.48 1.45</td>
</tr>
</tbody>
</table>

**Graph:**

Speedups of MPI+OpenMP Hybrid Code

**Image:**

Performance improvement of hybrid MPI+OpenMP over flat MPI code for Fock matrix construction on Intel MIC architecture. The hybrid code exploits all 240 hardware thread contexts on card.
Up to 40% Performance Improvement from New Load Balancing Scheme  
*PI: C.S. Chang, Fusion SciDAC Center for Edge Physics Simulation (EPSi)*

**Objectives**
- Address performance degradation due to load imbalance in nonlinear collision calculation for XGC1 on DOE Leadership Computing Systems.

**Impact**
- Low overhead automatic adjustment of parallel decomposition improves computational performance robustly and with minimal user input.

**Accomplishment highlight**

![Diagram](attachment:image.png)

Example load imbalance in collision operator cost, comparing load balancing only particle distribution with also load balancing collision cost. Cost is summed over rows of virtual 2D processor grid. Full model performance improvement is 30% for this example.

**Progress and Accomplishments:**

**Challenge**
- Existing particle load balancing algorithm does not also balance distribution of collision cost in parallel decomposition.
- Both particle count and collision cost per grid cell distributions evolve with the simulation.

**Solution**
- Two level optimization strategy: (a) balance collision cost subject to constraint on particle load imbalance, (b) optimize XGC1 performance by varying constraint periodically, converging to the optimum if distributions are static and adapting to the changing distributions otherwise.

**Result**
- 10%-40% improvement for production runs.
SUPER Brings Lots of Tools and Expertise to the Party

Edges/arrows show SUPER tools that have been integrated.

Commercial tools are conservative, while SUPER can afford to be aggressive (autotune etc)
Linking Performance MPAS-O Data into Scientific Visualization Tools
SDAV/SUPER Collaboration (BER/ASCR Multiscale PI: Bill Collins)

Objectives

- Map TAU performance measurements to the MPAS-Ocean spatial domain to assist in optimization of partition strategies

Impact

- Integrated TAU performance measurement data with application scientific data in VisIt
- Reduced execution time up to 15% for 60km model on 256 processes

Progress & Accomplishments

- Demonstrated that the load imbalance problem is correlated with variability among partition block size due to relatively large halo regions
- Visualizations also show that vertical depth, coastlines and number of neighbors likely affect computation, communication times
- Hindsight partition refinement using block+halo weights reduced mean MPI_Wait times by 40%, and overall execution time up to 15%
- Workshop publication: Huck et al., “Linking Performance Data into Scientific Visualization Tools”, Visual Performance Analytics at SC’14

Results collected on Edison@NERSC using 60km data and 240 processes
Performance Optimizations for MPAS-Ocean

Objectives:
- Accelerate MPAS-Ocean code performance on state-of-the-art supercomputers
- Prepare MPAS-Ocean for transition to next generation highly parallel architectures
- Work collaboratively with SUPER Institute to leverage performance optimization tools and expertise

Impact:
- Demonstrated that space-filling curve based ordering essential for intra-node communication reduction
- Developed partitioning optimization approaches broadly applicable to numerous of unstructured-mesh based computations
- Allow higher Simulated-Years-Per-Day throughput for ocean modeling simulations with MPAS-Ocean.

Progress and Accomplishments:
- SFC-based mesh data reordering: average 1.25x performance gain
- New halo-aware hypergraph partitioning algorithm improves scalability at high concurrency by over 2x.
- Combined with SUPER optimizations, including pointer reduction at 12K cores: 3-4x MPAS overall speedup
- A. Sarje et al. MSES/M/CCS 2015 nominated best paper
Objectives:

- Performance modeling is critical for identifying and ameliorating bottlenecks on emerging HPC systems
- Roofline performance model is a highly recognized approach for quantifying system and program behavior, but requires expert knowledge
- Goal: produce set of software tools allowing non-experts to automatically leverage Roofline modeling capabilities

Impact:

- Automated roofline code could be used to diagnose performance problems for DOE & SciDAC codes
- The community can focus on addressing appropriate performance impediments via optimization, algorithm design, or hardware selection
- Parameterized roofline models can be used to predict behavior of future systems, to help drive forward-looking algorithms and architectures

Progress and Accomplishments:

- Detailed roofline analysis of 4 leading HPCs: Edison, Mira, Babbage, and Titan
- Quantified benefits of emerging GPU software managed cache technologies
- Insights resulted in 03/2015 public release V1.0: Empirical Roofline Tool (LBNL) & Roofline Visualizer (U Oregon)
- Roofline Toolkit is a community tool for automatic hardware introspection & analysis
When is this Most Effective?

• **Long-term partnerships**
  – It takes time to build mutual respect and learn how to collaborate
  – Pat Worley well integrated into the climate and fusion communities
  – Rob Fowler is a member of USQCD

• **Common objectives**
  – All participants have to be focused on the same high-level goals
  – Individual work should be complimentary

• **Rewarding for all involved**
  – Novel solutions to performance problems are publishable in CS journals and conferences

• **Common funding**
  – SAP model establishes long-term relationships, but also fixes them for five years

• **Bring-your-own funding**
  – SciDAC Institutes allow DOE flexible redistribution of expertise as needed

• **DOE needs a balance of both looking forward**
  – A subtle SciDAC-4 suggestion 😊
  – Applies generally to all ASCR research topics
A Software Crisis Can Take Several Forms

• **Frog in the Pot metaphor**
  – The temperature has been slowly rising for three decades
  – It looks like it’ll get a lot warmer 😞

• **Labor affordability crisis**
  – What if we can’t afford to keep pace?
  – Programmer productivity does not keep pace with growth in complexity
  – Compounded when existing codes need replacement or major reengineering
  – We’ve done this before in the switch to vectors and then distributed memory
  – Could be an ongoing process if there is end-of-Moore’s Law market turbulence
    • How man “swim lanes” can DOE support if AMD and ARM get in the game?

• **Scientific capability crisis**
  – What if applications can’t be ported or recreated, regardless of effort?
  – Future “general purpose” machines only run a small subset of our problems
  – Remaining fields of science and engineering hit a capability plateau
    • To quote Bill Dally, “No amount of clever programming overcomes lousy hardware”
  – Specialized machines aren’t an alternative when the math and scientific algorithms are themselves objects of research
How to Mitigate the Looming Software Crisis

• Rethink the investment strategy in scientific computing
• How did we get here?
  – Reacting to the dynamics of commodity markets
  – Maximizing Flop/s per dollar
  – Let be honest, inter-Lab and international competition
• Looking forward
  – Retake control of our destiny
  – Require features in machines that broaden their utility
    • E.g., E-registers for covering latency and enabling a shared address space
  – Encourage adoption of new technology that enhances productivity
    • I’m a big believer in shared memory programming abstractions, such as UPC
    • Adoption is hindered when user’s think their codes have a future, and are naturally risk averse
  – Invest in research towards future systems and tools
    • The above examples are two decades old
• Plan for punctuated evolution
  – We need to evolve to maximize return on our investments (H/W and S/W)
  – We also need to adapt to new technology opportunities as they arise
• The end of Moore’s Law is creating a richer space of processors to choose from
  – Intel has multiple paths forward
  – IBM is in the game, teamed with Nvidia
  – AMD and ARM are also viable
• Consolidation in the memory market makes innovation feasible there too
  – Micron’s HMC is the tip of the iceberg
  – Fast Forward 2 includes a New Memory Interface
• DOE could engineer its own swim lanes
  – Real topical systems, somewhere between Anton and NERSC in terms of generality
  – Perhaps with domain-specific programming languages
  – Greg Astfalk’s “Bespoke SOC” vision is an example of this from an HP perspective
• Taking control of our own destiny requires research investments now
  – What should such systems look like? Green Flash? QCDOC?
  – How can we know that they will be more productive?
  – How would we even define productivity?