Performance Evaluation of Programming Models for the Multicore Era

Nicholas J. Wright
NERSC/LBNL
njwright@lbl.gov

Programming weather, climate, and earth-system models on heterogeneous multi-core platforms
NCAR Sept 2011
Background and Biases

NERSC Staff: represent science
• Over 1500 publications per year
• Over 3000 users, 400 projects, 500 code instances: hard to move!

Background:
• PhD in Computational Chemistry
• Started in User Services at SDSC – moved to PMaC Lab (Snavely). Now Advanced Technologies Group at NERSC
• Tools Developer – IPM – www.ipm2.org
• Benchmarking and Performance Analysis, Procurements.
The path to Exascale: How and When to Move Users?

How do we ensure that Users Performance follows this trend and their Productivity is unaffected? How do we avoid them re-writing their code more than once?
Numerical Methods at NERSC
(Caveat: survey data from ERCAP requests)
Challenges to Exascale

1. System power is the primary constraint
2. Concurrency (1000x today)
3. Memory bandwidth and capacity are not keeping pace
4. Processor architecture is open, but likely heterogeneous
5. Programming model heroic compilers will not hide this
6. Algorithms need to minimize data movement, not flops
7. I/O bandwidth unlikely to keep pace with machine speed
8. Reliability and resiliency will be critical at this scale
9. Bisection bandwidth limited by cost and energy

Unlike the last 20 years most of these (1-7) are equally important across scales e.g., 1000 1-PF machines
Horst Simon’s Distractions

The Road to Exaflop/s – four distractions and a road block

- clouds
- power
- GPUs
- China
- Japan

Exaflop/s

our community
Power: It's all about moving data

- Power is about moving data
- PicoJoules
- 2018
- Intranode/SMP Communication
- Intranode/MPI Communication
- Data from John Shalf
Case for Lightweight Cores and Heterogeneity

<table>
<thead>
<tr>
<th></th>
<th>Intel QC Nehalem</th>
<th>Tensilica</th>
<th>Overall Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power (W)</td>
<td>100</td>
<td>.1</td>
<td>$10^3$</td>
</tr>
<tr>
<td>Area (mm$^2$)</td>
<td>240</td>
<td>2</td>
<td>$10^2$</td>
</tr>
<tr>
<td>DP flops</td>
<td>50</td>
<td>4</td>
<td>.1</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
<td>$10^4$</td>
</tr>
</tbody>
</table>

Lightweight (thin) cores improve energy efficiency

- Number of Cores = 1 (Fat) + 256 (Thin) − R (Thin)
- R is the amount of area used for the Fat core in Thin core units
- Assumes speedup for Fat / Thin = Sqrt of Area advantage

Heterogeneity Analysis by: Mark Hill, U. Wis
What Will an Exascale ‘Processor’ look like?

- Lots of small power efficient cores ‘GPU’
- A few ‘fat’ cores optimised for single thread performance ‘CPU’

How are we going to program this thing?!?!
What’s Wrong with MPI Everywhere

• We can run 1 MPI process per core
  – This works now (for CMPs) and will work for a while

• How long will it continue working?
  – Depends on performance expectations

• What is the problem?
  – Latency: some copying required by semantics
  – Memory utilization: partitioning data for separate address space requires some replication
    • How big is your per core subgrid? At 10x10x10, over 1/2 of the points are surface points, probably replicated
  – Memory bandwidth: extra state means extra bandwidth
  – Weak scaling: success model for the “cluster era;” will not be for the many core era -- not enough memory per core
  – Heterogeneity: MPI per SIMD element or CUDA thread-block?
What are the Basic Differences Between MPI and OpenMP?

• Program is a collection of processes.
  - Usually fixed at startup time
  - Single thread of control plus private address space -- NO shared data.
• Processes communicate by explicit send/receive pairs
  - Coordination is implicit in every communication event.
• MPI is most important example.

• Program is a collection of threads.
  - Can be created dynamically.
• Threads have private variables and shared variables
• Threads communicate implicitly by writing and reading shared variables.
  - Threads coordinate by synchronizing on shared variables
• OpenMP is an example

K. Yelick, CS267 UCB
MPI + OpenMP Two Questions:

1. Given a fixed number of cores can I solve the same problem with the same performance using hybrid programming and save memory?
   - GTC, fvCAM, PARATEC, VASP, QE

2. Using a fixed number of MPI tasks can I run on more cores using OpenMP to increase my performance?
   - GTC, fvCAM, PARATEC, VASP, QE

Performance vs. “Processors”

```
Performance

Nodes

MPI
MPI+OpenMP
Ideal

1
10
100
1000

1
10
100
1000

2.6X
```
Hybrid MPI-OpenMP Programming

Benefits

+ Less Memory usage
+ Focus on # nodes *(which is not increasing as fast)* instead of # cores
+ Larger messages, less tasks in collectives, less time in MPI

+ Attack different levels of parallelism than possible with MPI

Potential Pitfalls

- NUMA / Locality effects
- Synchronization overhead
- Inability to saturate network adaptor

Mitigations

- User training
- Code examples using *real* applications
- Hopper system configuration changes
- Feedback to Cray on compiler & system software development
One-Sided vs Two-Sided Communication

- A one-sided put/get message can be handled directly by a network interface with RDMA support
  - Avoid interrupting the CPU or storing data from CPU (preposts)
- A two-sided messages needs to be matched with a receive to identify memory address to put data
  - Offloaded to Network Interface in networks like Quadrics
  - Need to download match tables to interface (from host)
  - Ordering requirements on messages can also hinder bandwidth
Partitioned Global Address Space (PGAS) Languages

- Unified Parallel C
- Co-Array Fortran
- Global Arrays
- Titanium
- Chapel
- X10
- ......
Defining PGAS principles:
- The Global Address Space memory model allows any thread to read or write memory anywhere in the system.
- It is Partitioned to indicate that some data is local, whereas other data is further away (slower to access).
PGAS: Messaging Rate Bandwidths

**Aggregate BW (GB/s)**

- MPI 4 Pairs
- CAF 4 Pairs
- MPI 24 Pairs
- CAF 24 Pairs

**Message Size (Bytes)**

- 8, 16, 32, 64, 128, 256, 512, 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K, 256K, 512K, 1M, 2M

**Platforms**

- Hopper
- Franklin

**Legend**

- CAF 24 Pairs
- MPI 24 Pairs
- MPI 4 Pairs
- CAF 4 Pairs
PGAS: NAS-FT Class B

Performance:
@ 16K CAF 2.8x MPI
• For medium-sized messages single sided protocol provides large potential performance win
• This does not mean simply swapping MPI calls for PGAS equivalents will work
• However if you switch to more fine grained messaging.. (Priessel et al SC11)
Dirac GPU Testbed Configuration

- **Hardware**
  - 44 nodes w/ 1 GPU per node
    - integrated into carver cluster
  - **Host Node**
    - dual-socket Xeon E5530 (Nehalem)
    - 76.5GF Peak DP (153 GF SP)
    - QDR Infiniband
  - 24GB GB DDR-1066 memory
  - 51 GB/s peak mem BW
  - **GPU**
    - Nvidia Tesla C2050 (Fermi)
    - 515GF peak DP (1030GF SP): 6x more than host
    - 3 GB memory
    - 144 GB/s peak mem BW (3x)

- **Software**
  - CUDA 3.2
  - PGI Compilers
  - GPU Direct
    - OpenMPI
    - MVAPICH
  - Matlab Parallel Computing Toolbox coming soon
Roofline Performance Model

Sam Williams

https://ftg.lbl.gov/assets/staff/swwilliams/talks/parlab08roofline.pdf
The Roofline Performance Model

- The flat room is determined by arithmetic peak and instruction mix.
- The sloped part of the roof is determined by peak DRAM bandwidth (STREAM).
- X-axis is the computational intensity of your computation.
Relative Performance Expectations

**Fermi & Nehalem Roofline**

- 6.7x peak
- 2.2x Measured BW
- 1.7x 7-point Stencil

Arithmetic Intensity

Attainable Gflop/s

- 1/32
- 1/16
- 1/8
- 1/4
- 1/2
- 1
- 2
- 4
- 8
- 16
- 32

Double-precision peak

- Device bandwidth
- STREAM bandwidth
- 7pt Stencil
Relative Performance Across Kernels

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Speedup on GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGEMM</td>
<td>3.8</td>
</tr>
<tr>
<td>Reverse Time Migration - 12th order Stencil</td>
<td>3.9</td>
</tr>
<tr>
<td>27pt Stencil</td>
<td>2.2</td>
</tr>
<tr>
<td>7pt Stencil</td>
<td>2.3</td>
</tr>
<tr>
<td>GTC/pushi</td>
<td>1.3</td>
</tr>
<tr>
<td>GTC/chargei</td>
<td>0.5</td>
</tr>
<tr>
<td>spMV min</td>
<td>1.8</td>
</tr>
<tr>
<td>spMV median</td>
<td>1.0</td>
</tr>
<tr>
<td>spMV max</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Xeon X5550 (Nehalem)

NVIDIA C2050 (Fermi)
## Performance Summary: Full Applications

<table>
<thead>
<tr>
<th>Domain</th>
<th>Algorithm</th>
<th>Performance Summary cf. 8 core Nehalem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Molecular Dynamics (HOOMD)</td>
<td>N-body</td>
<td>~6-7x</td>
</tr>
<tr>
<td>Lattice Boltzmann CFD</td>
<td>Lattice Boltzmann</td>
<td>~1.17x</td>
</tr>
<tr>
<td>Geophysical Modeling</td>
<td>quasi-minimum-residual (QMR) solver</td>
<td>~3.33x</td>
</tr>
<tr>
<td>QCD</td>
<td>Krylov space solvers to compute intensive matrix inversion</td>
<td>~3.0x (matrix multiply) ~2.5 multi-shifted bi-conjugate gradient algorithm</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>AMR</td>
<td>~5x</td>
</tr>
</tbody>
</table>
ISAAC is a three-year (2010-2013) NSF funded project to focus on research and development of infrastructure for accelerating physics and astronomy applications using and multicore architectures.

Goal is to successfully harness the power of the parallel architectures for compute-intensive scientific problems and open doors for new discovery and revolutionize the growth of science via, Simulations, Instrumentations and Data processing /analysis.

Visit us – http://iccs.lbl.gov
GPU User Community and Seminars

- GP-GPU programming mailing list
  - gpgpu-discuss@nersc.gov
- Also bi-weekly online GP-GPU seminars
  - Contact Hemant Shukla (HShukla@lbl.gov)
Going forward....

What about GPU’s?
What about X?
• IBM
  – BG/Q

IBM’s new transactional memory: make-or-break time for multithreaded revolution

By Peter Bright | Published 7 days ago
<table>
<thead>
<tr>
<th>Green500 Rank</th>
<th>MFLOPS/W</th>
<th>Site*</th>
<th>Computer*</th>
<th>Total Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2097.19</td>
<td>IBM Thomas J. Watson Research Center</td>
<td>NNSA/SC Blue Gene/Q Prototype 2</td>
<td>40.95</td>
</tr>
<tr>
<td>2</td>
<td>1684.20</td>
<td>IBM Thomas J. Watson Research Center</td>
<td>NNSA/SC Blue Gene/Q Prototype 1</td>
<td>38.80</td>
</tr>
<tr>
<td>3</td>
<td>1375.88</td>
<td>Nagasaki University</td>
<td>DEGIMA Cluster, Intel i5, ATI Radeon GPU, Infiniband QDR</td>
<td>34.24</td>
</tr>
<tr>
<td>4</td>
<td>958.35</td>
<td>GSIC Center, Tokyo Institute of Technology</td>
<td>HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows</td>
<td>1243.80</td>
</tr>
<tr>
<td>5</td>
<td>891.88</td>
<td>CINECA / SCS - SuperComputing Solution</td>
<td>iDataPlex DX360M3, Xeon 2.4, nVidia GPU, Infiniband</td>
<td>160.00</td>
</tr>
<tr>
<td>6</td>
<td>824.56</td>
<td>RIKEN Advanced Institute for Computational Science (AICS)</td>
<td>K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect</td>
<td>9898.56</td>
</tr>
<tr>
<td>7</td>
<td>773.38</td>
<td>Forschungszentrum Juelich (FZJ)</td>
<td>QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus</td>
<td>57.54</td>
</tr>
<tr>
<td>8</td>
<td>773.38</td>
<td>Universitaet Regensburg</td>
<td>QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus</td>
<td>57.54</td>
</tr>
<tr>
<td>9</td>
<td>773.38</td>
<td>Universitaet Wuppertal</td>
<td>QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus</td>
<td>57.54</td>
</tr>
<tr>
<td>10</td>
<td>718.13</td>
<td>Universitaet Frankfurt</td>
<td>Supermicro Cluster, QC Opteron 2.1 GHz, ATI Radeon GPU, Infiniband</td>
<td>416.78</td>
</tr>
</tbody>
</table>
July 28-29, 2011 Meeting

This memorandum is made publicly available for any interested parties to review. A more detailed summary of the meeting will be forthcoming and posted on the National Science Board (Board, NSB) public Web site (http://www.nsf.gov/nsb/). The minutes of the Plenary Open Session for the July 2011 meeting will also be posted on the Board’s public Web site following Board approval at the September 2011 meeting.

Major actions and approvals at the 420th meeting of the Board included the following (not in priority order):

1. The Board authorized the NSF Director, at his discretion, to make an award to the University of Texas at Austin for support of Enabling, Enhancing, and Extending Petascale Computing for Science and Engineering (NSB-11-48).

---

**TACC Signs Up for Manycore Development Using Intel's MIC Processor**

**AUSTIN, Texas, April 21 --** The Texas Advanced Computing Center (TACC) at The University of Texas at Austin today announced that it will collaborate with Intel Corp. to help prepare the national open science research community to take full advantage of the new capabilities of Intel's forthcoming "many integrated core" (MIC) processor line.
Evolution of Abstract Machine Model
(underpinning of programming model)

SMP

MPI Distributed Memory

PGAS

HPGAS????
• Parallelism
  – !dir$ ‘run this loop on the lightweight cores’
  – !dir$ ‘here are some hints about parallelisation’
• Locality?
  – !dir$ ‘move these arrays’
  – !dir$ ‘these arrays are already on the lightweight cores’
Summary

• Disruptive technology changes are coming
• From a NERSC perspective
  – We want our users to remain productive - Ideally we want them to only re-write their code once
• Only solve the problems that need to be solved
  – Locality (how many levels are necessary?)
  – Heterogeneity
  – Vertical communication management
    • Horizontal is solved by MPI (or PGAS)
  – Fault resilience, maybe
    • Look at the 800-cabinet K machine
• Dynamic resource management
  • Definitely for irregular problems
  • Maybe for regular ones on “irregular” machines
• Resource management for dynamic distributed runtimes