Project Zeta: an integrated simulation and analysis platform for earth system science

Dr. Richard Loft
Director, Technology Development
Computational and Information Systems Laboratory
National Center for Atmospheric Research

ZETA = ZEro-copy Trans-petascale Architecture
Application developer’s view of exascale technology
New technologies, faster science?

Stacked memory: Fast, hot & small

Memory-class storage

Storage-class memory
New technologies, faster science?

Stacked memory: Fast, hot & small

Memory-class storage

Storage-class memory

Cloud-base object store (public or private)
Preformance Portability?

- CPUs With HBM
- FPGA
- Earth System Models
- Neuromorphic
- Intel
- GPU Accelerators
- Xeon Micro-processors
- EPYC
- CAVIUM
- THUNDERx2

“Falcon Mesa (HE)” FPGA • SoC • SiP
Project Zeta Goals

• **Focus on a design in Zeta that:**
  – Enhances the end-to-end rate of science throughput
  – Reduces costs and/or enhance reliability

• **Harness emerging technologies for Zeta like:**
  – Accelerators (GPUs)
  – New memory technologies (stacked, NV memory)
  – Machine learning techniques (DL)

• **Prepare application/workflow codes for Zeta:**
  – scalability and performance
  – Performance-portability
Existing Architecture

- Xeon Super-computer: $O(10^5)$ cores, $O(0.3 \text{ PB DRAM})$
- Web servers
- Small Analysis Cluster
  - Analysis Nodes: $O(10)$
  - Hot Cache (Disk): $\sim O(200) \times \text{DRAM}$
- Warm Cache (Tape): $\sim O(500) \times \text{DRAM}$
What’s wrong with our performance?

Yellowstone: Sustained fraction of FP peak was 1.57%

Yellowstone Floating Point Efficiency

Y(\cdot) = Y(\cdot) + a*x(\cdot) – a.k.a. DAXPY
Knowing your limits: the roofline diagram

Source: Barba and Yokota, SIAM News, Volume 46, Number 6, July/August 2013

MOM6 barotropic stencil
0.125 flop/byte (DP)

RBF-FD SWE Model
0.5 flop/byte (DP)
HOMME (NE=8, PLEV=70, qsize=135)

75% reduction in cost!

Xeon Phi is 1.9x faster than Xeon!
Simulation rate for HOMME on Xeon and KNL

Superlinear speedup due to L3 cache on Xeon

100 km

Good simulation rates!

30% cost of 20 SYPD

30% cost of 5 SYPD

30% cost of 1 SYPD

Marginal for climate!

Too slow for climate simulations!

12 km

25 km
Optimizing Stencils for different architectures

Benchmark Problem

- Shallow Water Equations (SWE)
  - A set of non-linear partial differential equations (PDE)
  - Capture features of atmospheric flow around the Earth
- Radial basis function-generated finite difference (RBF-FD) methods

RBF-FD solution to SWE test case “Flow over an isolated mountain” using 655,532 points [1]

An example of 75-point stencil on a sphere [1]
CISL experiences with directive-based portability:
RBF-FD shallow water equations:
2D unstructured stencil

- CI roofline model generally predicts performance well, even for more complicated algorithms.
- Xeon performance crashes to DRAM BW limit when cache size is exceeded, with some state reuse.
- Xeon Phi (KNL) HBM memory is less sensitive to problem size than Xeon, saturates with CI figure.
- NVIDIA Pascal P100 performance fits CI model GPU’s require higher levels of parallelism to reach saturation.
**MPAS 5 Performance**

Execution time for single timestep (in seconds)

<table>
<thead>
<tr>
<th>Kernels</th>
<th>Broadwell Node</th>
<th>Pascal P100</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>120 Km</td>
<td>60 Km</td>
<td>120 Km</td>
</tr>
<tr>
<td>Integration Setup</td>
<td>1.21E-02</td>
<td>5.31E-02</td>
<td>1.86E-03</td>
</tr>
<tr>
<td>Moist coefficients</td>
<td>2.08E-03</td>
<td>9.28E-03</td>
<td>1.49E-03</td>
</tr>
<tr>
<td>imp_coef</td>
<td>4.66E-03</td>
<td>1.28E-02</td>
<td>3.20E-03</td>
</tr>
<tr>
<td>dyn_tend</td>
<td>3.91E-03</td>
<td>1.41E-01</td>
<td>1.41E-02</td>
</tr>
<tr>
<td>small_step</td>
<td>3.20E-02</td>
<td>1.44E-02</td>
<td>1.08E-03</td>
</tr>
<tr>
<td>acoustic_step</td>
<td>3.70E-03</td>
<td>3.78E-02</td>
<td>4.70E-03</td>
</tr>
<tr>
<td>large_step</td>
<td>1.03E-02</td>
<td>5.09E-02</td>
<td>2.78E-03</td>
</tr>
<tr>
<td>diagnostics</td>
<td>1.63E-02</td>
<td>8.22E-02</td>
<td>4.53E-03</td>
</tr>
<tr>
<td>Time step Loop</td>
<td>0.92</td>
<td>3.49</td>
<td>0.37</td>
</tr>
</tbody>
</table>

Code currently being upgraded to MPAS 5.2
NCAR performance portability experiences...

- Refactoring code for vectorization can yield \(~2.5-4x\) performance improvements for x86 multi-/many-cores. We’ve been co-designing a vectorizing ifort....

- Directive-based parallelism provides portability across Xeon, Xeon-Phi and GPU. Maintaining single source feasible for many cases (RBFs & MPAS).

- OpenACC is in a sense a “domain specific language”. We’ve been co-designing OpenACC with PGI...

- Would be nice if a std emerge (e.g. OpenMP)

- Portability across 3 architectures is all great but...
CESM/CMIP6 Workflow

- **Model Run**
  - CESM Model Run

- **Post-Processing**
  - Time Series Conversion (PyReshaper)
  - Data Compliance Tool (PyConform)
  - Re-Designed Diagnostics (PyAverager)

- **Publication**
  - Push to ESGF (Improved process)

Automated Workflow Management
NCAR Analytics Accomplishments: The Low Hanging Fruit

- Parallel tools: **PyReshaper, PyAverager, PyConform**
- Parallelizing **PyReshaper** yielded ~6.5x on Edison
- **NAND-based** tests
  - Py{*} analytics 2.5-6x
  - subsetting (RDA) 20x
- Automating workflows (Cycl) saved O(3x)
- 5x storage volume savings through lossy data compression (discussed yesterday).
Unsupervised Learning: Generative Adversarial Networks

Unsupervised method of learning complex feature representations from data
Requires 2 deep neural networks

**Discriminator**: determines which samples are from the training set and which are not

**Generator**: Creates synthetic examples similar to training data to fool discriminator

Both networks have a “battle of wits” either to the death or until the discriminator is fooled often enough

**Advantages**

- Unsupervised pre-training: learn features without needing a large labeled dataset
- Dimensionality reduction: reduce image to smaller vector
- Learns sharper, more detailed features than auto-encoder models
- Do not need to specify a complex loss function

Credit: Princess Bride
Pros and cons of building DL emulators

• **Pros**
  – Drafts behind DL-driven technology
  – May be less (80x?) computationally intensive
  – Deep Learning leverages frameworks.
  – Less code to develop (code is in the weights and the network design)

• **Cons**
  – Potential loss of understanding of the physical basis of results.
  – Over-fitting, curse of dimensionality, etc. Kind of an art.
  – Not clear how conservation laws/constraints are preserved in DL systems.
Existing Architecture

Xeon Super-computer

O(10^5) cores
O(0.3 PB DRAM)

Small Analysis Cluster

O(10) Analysis Nodes

Web servers

Disk

Hot Cache (Disk):
~O(200)x DRAM

tape

~Warm Cache (Tape):
~O(500) x DRAM
Zeta Architecture

Simulation & Data Assimilation

O(1M cores)
O(1 PB DRAM)

Deductive

Viz/FPGA nodes

Parallel Analytics & Machine Learning

Data movers

HBM devices

Super-cache
O(5x) DRAM memory

NVRAM

O(TBs/s)

Inductive

Warm Cache (Disk):
~O(40x) DRAM

Disk

~DR/Collections (Tape):
~O(100) x DRAM

Cloud

More Analysis Nodes

More Analysis Nodes
Thanks!
Current supercomputers struggle on HPCG relative to HP Linpack:

<table>
<thead>
<tr>
<th>Site</th>
<th>Computer</th>
<th>Cores</th>
<th>HPL Rmax (Pflops)</th>
<th>HPL Rank</th>
<th>HPCG (Pflops)</th>
<th>HPCG/HPL</th>
<th>% of Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSCC / Guangzhou</td>
<td>Tianhe-2 NUDT, Xeon 12C 2.2GHz + Intel Xeon Phi 57C + Custom</td>
<td>3,120,000</td>
<td>33.9</td>
<td>1</td>
<td>0.632</td>
<td>1.8%</td>
<td>1.1%</td>
</tr>
<tr>
<td>RIKEN Advanced Inst for Comp Sci</td>
<td>K computer Fujitsu SPARC64 VIIIfx 8C + Custom</td>
<td>705,024</td>
<td>10.5</td>
<td>4</td>
<td>0.461</td>
<td>4.4%</td>
<td>4.1%</td>
</tr>
<tr>
<td>DOE/OS Oak Ridge Nat Lab</td>
<td>Titan, Cray XK7 AMD 16C + Nvidia Kepler GPU 14C + Custom</td>
<td>560,640</td>
<td>17.6</td>
<td>2</td>
<td>0.322</td>
<td>1.8%</td>
<td>1.2%</td>
</tr>
<tr>
<td>DOE/OS Argonne Nat Lab</td>
<td>Mira BlueGene/Q, Power BQC 16C 1.6GHz + Custom</td>
<td>786,432</td>
<td>8.59</td>
<td>5</td>
<td>0.167</td>
<td>1.9%</td>
<td>1.7%</td>
</tr>
<tr>
<td>Swiss CSCS</td>
<td>Piz Daint, Cray XC30, Xeon 8C + Nvidia Kepler 14C + Custom</td>
<td>115,984</td>
<td>6.27</td>
<td>6</td>
<td>0.105</td>
<td>1.7%</td>
<td>1.3%</td>
</tr>
<tr>
<td>Leibniz Rechenzentrum</td>
<td>SuperMUC, Intel 8C + IB</td>
<td>147,456</td>
<td>2.90</td>
<td>14</td>
<td>0.0833</td>
<td>2.9%</td>
<td>2.6%</td>
</tr>
<tr>
<td>DOE/OS LBNL</td>
<td>Edison, Cray XC30, Xeon, 12c, 2.4GHz + Custom</td>
<td>133,824</td>
<td>1.65</td>
<td>24</td>
<td>0.0786</td>
<td>4.8%</td>
<td>3.1%</td>
</tr>
<tr>
<td>GSIC Center TiTech</td>
<td>Tsubame 2.5 Xeon 6C, 2.93GHz + Nvidia K20x + IB</td>
<td>76,032</td>
<td>2.78</td>
<td>15</td>
<td>0.073</td>
<td>2.6%</td>
<td>1.3%</td>
</tr>
<tr>
<td>Max-Planck</td>
<td>iDataPlex Xeon 10C, 2.8GHz + IB</td>
<td>65,320</td>
<td>1.28</td>
<td>34</td>
<td>0.061</td>
<td>4.8%</td>
<td>4.2%</td>
</tr>
<tr>
<td>CEA/TGCC-GENCI</td>
<td>Curie fine nodes Bullx B510 Intel Xeon 8C 2.7 GHz + IB</td>
<td>77,184</td>
<td>1.36</td>
<td>33</td>
<td>0.051</td>
<td>3.8%</td>
<td>3.1%</td>
</tr>
<tr>
<td>Exploration and Production Eni S.p.A.</td>
<td>HPC2, Intel Xeon 10C 2.8 GHz + Nvidia Kepler 14C + IB</td>
<td>62,640</td>
<td>3.00</td>
<td>12</td>
<td>0.0489</td>
<td>1.6%</td>
<td>1.2%</td>
</tr>
</tbody>
</table>
Processor flops/byte: trending upwards
Energy usage for HOMME on Xeon and Xeon Phi @ 100 km