HPC Performance and Energy Efficiency

Overview and Trends

June 9th, 2015
SMAI 2015 Congress
Les Karellis (Savoie)

Dr. Sébastien Varrette
Parallel Computing and Optimization Group (PCOG)

http://hpc.uni.lu
Outline

- Introduction & Context
- HPC Data-Center Trends: Time for DLC
- HPC [Co-]Processor Trends: Go Mobile
- Middleware Trends: Virtualization, RJMS
- Software Trends: Rethinking Parallel Computing
- Conclusion
Introduction and Context
HPC at the Heart of our Daily Life

■ Today... R&D, Academia, Industry, Local Collectivities

■ ... Tomorrow: digital health, nano/bio techno...
Performance Evaluation of HPC Systems

- Commonly used metrics
  - ✓ FLOPs: raw compute capability
  - ✓ GUPS: memory performance
  - ✓ IOPS: storage performance
  - ✓ bandwidth & latency: memory operations or network transfer

- Energy Efficiency
  - ✓ Power Usage Effectiveness (PUE) in HPC data-centers
    - Total Facility Energy / Total IT Energy
  - ✓ Average system power consumption during execution (W)
  - ✓ Performance-per-Watt (PpW)
Ex (in Academia): The UL HPC Platform

http://hpc.uni.lu

- 2 geographical sites, 3 server rooms
- 4 clusters, ~281 users
  - 404 nodes, 4316 cores (49.92 TFlops)
  - Cumul. *shared* raw storage: 3,13 PB
  - Around 197 kW
- > 6,21 M€ HW investment so far
- Mainly Intel-based architecture
- Mainly Open-Source software stack
  - Debian, SSH, OpenLDAP, Puppet, FAI...
Ex (in Academia): The UL HPC Platform

http://hpc.uni.lu
General HPC Trends

- **Top500**: world’s 500 most powerful computers (since 1993)
  - Based on High-Performance LINPACK (HPL) benchmark
  - Last list [Nov. 2014]
    - #1: Tianhe-2 (China): 3,120,000 cores
      - 33.863 PFlops... and 17.8 MW
    - Total combined performance:
      - 309 PFlops
      - 215.744 MW over 258 systems
        (which provided power information)

- **Green500**: Derive PpW metric from Top500 (MFlops/W)
  - #1: L-CSC GPU Cluster (#168): 5.27 GFlops/W

- **Other Benchmarks**: HPC{C,G}, Graph500...
Computing Needs Evolution

1 ZFlops
100 EFlops
10 EFlops
1 EFlop
100 PFlops
10 PFlops
1 PFlop
100 TFlops
10 TFlops
1 TFlop
100 GFlops
10 GFlops
1 GFlop


Multi-Scale Weather prediction

Human Brain Project

Genomics

Computational Chemistry Molecular Dynamics

Manufacturing
Computing Power Needs Evolution

- **1 ZFlops**
- **100 EFlops**
- **10 EFlops**
- **1 EFlop**
- **100 PFlops**
- **10 PFlops**
- **1 PFlop**
- **10 TFlops**
- **1 TFlop**
- **100 GFlops**
- **10 GFlops**
- **1 GFlop**

### Applications
- Manufacturing
- Computational Chemistry
- Molecular Dynamics
- Genomics
- Human Brain Project
- Multi-Scale Weather prediction

### Timeline
- **1993**
- **1999**
- **2005**
- **2011**
- **2017**
- **2023**
- **2029**

### Energy Consumption
- **100 kW**
- **1 MW**
- **10 MW**
- **100 MW**
- **1 GW**
Computing Less Power Needs Evolution

1 ZFlops
100 EFlops
10 EFlops
1 EFlop
100 PFlops
10 PFlops
1 PFlop
100 TFlops
10 TFlops
1 TFlop
100 GFlops
10 GFlops
1 GFlop


Human Brain Project
Computational Chemistry
Molecular Dynamics
Genomics
Multi-Scale Weather prediction

Manufacturing

< 20 MW
10 MW
10 MW
1 MW
100 kW

Facility of Sciences, Technology and Communication
The Budgetary Wall

- Manufacturing
- Computational Chemistry
- Molecular Dynamics
- Genomics
- Human Brain Project
- Multi-Scale
- Weather prediction


- < 20 MW
- 10 MW
- 10 MW
- 1 MW
- 100 kW

- 1 ZFlops
- 100 EFlops
- 10 EFlops
- 1 EFlops
- 100 PFlops
- 10 PFlops
- 1 PFlops
- 10 TFlops
- 10 TFlops
- 1 TFlops
- 100 GFlops
- 10 GFlops
- 1 GFlops

- 100 kW
- 1 MW
- 10 MW
- 10 MW
- < 1 M€ / MW / Year
- 1,5 M€ / MW / Year
- > 3 M€ / MW / Year

- FACULTY OF SCIENCES, TECHNOLOGY AND COMMUNICATION
Energy Optimization paths toward Exascale

- H2020 Exascale Challenge: 1 EFlops in 20 MW
  ✓ Using today’s most energy efficient TOP500 system: 189MW

Reduced Power Consumption

- New [co-]processors, interconnect...
- Virtualization, RJMS...
- New programming/execution models

Data-center

Hardware

Middleware

Software

PUE optim.
DLC...

FACULTY OF SCIENCES, TECHNOLOGY AND COMMUNICATION
HPC Data-Center Trends: Time for DLC

- Reduced Power Consumption
  - Hardware
  - Middleware
  - Software

Data-center ➔ Reduced Power Consumption
Cooling and PUE

- **Air-cooled**
  - 10(-20) kW/rack
  - Room 20°C
  - A/C water 7-12°C
  - PUE ≥ 1.7

- **Water-cooled doors**
  - 40 kW/rack
  - Room 20°C
  - Water 7-12°C
  - PUE ~ 1.4

- **Direct-Liquid-cooling**
  - 80 kW/rack
  - Room up to 27°C
  - WaterUp to 30°C
  - PUE < 1.1

Co-generation

Courtesy of Bull SA
Cooling and PUE

- Direct immersion: the CarnotJet example (PUE: 1.05)
HPC [Co-]Processor Trends: Go Mobile
Microprocessors ~10x slower than one vector CPU
✓ ... thus not faster... But cheaper!
Microprocessors ~10x slower than one vector CPU
✓ ... thus not faster... But cheaper!
How about now?

- Mobile SoCs ~10x slower than one microprocessor
  ✓ ... thus not faster... But cheaper!

✓ the “already seen” pattern?

- Mont-Blanc project: build an HPC system from embedded and mobile devices
Mont-Blanc (Phase 1) project outcomes

- (2013) Tiribado: the first ARM HPC multicore system

Q7 Tegra 2
2 x Cortex-A9 @ 1GHz
2 GFLOPS
5 Watts (?)
0.4 GFLOPS / W

Q7 carrier board
2 x Cortex-A9
2 GFLOPS
1 GbE + 100 MbE
7 Watts
0.3 GFLOPS / W

1U Rackable blade
8 nodes
16 GFLOPS
65 Watts
0.25 GFLOPS / W

2 Racks
32 blade containers
256 nodes
512 cores
10x 48-port 1GbE switch
8x 48-port 100 MbE switch
512 GFLOPS
3.4 Kwatt
0.15 GFLOPS / W

0.15 GFlops/W

Courtesy of BCS
The UL HPC viridis cluster (2013)

- 2 encl. (96 nodes, 4U), 12 calxeda boards per enclosure

- 4x ARM Cortex A9 @ 1.1 GHz [4C] per Calxeda board
  - 2x300W, “10” GbE inter-connect

**Performance Evaluation and Energy Efficiency of High-Density HPC Platforms Based on Intel, AMD and ARM Processors**


0.513 GFlops/W
Commodity vs. GPGPUs: L-CSC (2014)

- The German L-CSC cluster (Frankfurt) (2014)
- Nov 2014: 56 (out of 160) nodes, on each:
  - 4 GPUs, 2 CPUs, 256 GB RAM
  - #168 on Top 500 (1.7 PFlops)
  - #1 on Green 500

5.27 GFlops/W
Mobile SoCs and GPGPUs in HPC

- Very fast development for Mobile SoCs and GPGPUs
- Convergence between both is foreseen
  - CPUs inherits from GPUs multi-core with vector inst.
  - GPUs inherits from CPUs cache-hierarchy
- In parallel: large innovation in other embedded devices
  - Intel Xeon Phi co-processor
  - FPGAs etc.

Objective: 50 GFlops/W
Middleware Trends: Virtualization, RJMS

- Reduced Power Consumption

[Diagram showing relationships between hardware, middleware, data-center, software, and reduced power consumption.]
**Virtualization in an HPC Environment**

- **Hypervisor:** Core virtualization engine / environment
  - ✓ Type 1 adapted to HPC workload
  - ✓ Performance Loss: > 20%

  - Xen, VMWare (ESXi), KVM, Virtualbox
Virtualization in an HPC Environment

- **Hypervisor**: Core virtualization engine / environment
  - ✓ Type 1 adapted to HPC workload
  - ✓ **Performance Loss**: > 20%

---

Cloud Computing vs. HPC

- World-wide advertised as THE solution to all problems
- Classical taxonomy:
  - ✓ {Infrastructure,Platform,Software}-as-a-Service
  - ✓ Grid’5000: Hardware-as-a-Service
Cloud Computing vs. HPC

- World-wide advertised as THE solution to all problems
- Classical taxonomy:
  ✓ `{Infrastructure, Platform, Software}-as-a-Service`
  ✓ Grid’5000: Hardware-as-a-Service
Cloud Middleware for HPC Workload

<table>
<thead>
<tr>
<th>Middleware:</th>
<th>vCloud</th>
<th>Eucalyptus</th>
<th>OpenNebula</th>
<th>OpenStack</th>
<th>Nimbus</th>
</tr>
</thead>
<tbody>
<tr>
<td>License</td>
<td>Proprietary</td>
<td>BSD License</td>
<td>Apache 2.0</td>
<td>Apache 2.0</td>
<td>Apache 2.0</td>
</tr>
<tr>
<td>Supported Hypervisor</td>
<td>VMWare/ESX</td>
<td>Xen, KVM, VMWare</td>
<td>Xen, KVM, VMWare</td>
<td>Xen, KVM, Linux Containers, VMWare/ESX, Hyper-V, QEMU, UML 8 (Havana)</td>
<td>Xen, KVM</td>
</tr>
<tr>
<td>Last Version</td>
<td>5.5.0</td>
<td>3.4</td>
<td>4.4</td>
<td>Python</td>
<td>2.10.1</td>
</tr>
<tr>
<td>Programming Language</td>
<td>n/a</td>
<td>Java / C</td>
<td>Ruby</td>
<td>Ubuntu, ESX, Debian, RHEL, SUSE, Fedora</td>
<td>Ubuntu, Debian, RHEL, SUSE, Fedora</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>VMX server</td>
<td>RHEL 5, ESX, Debian, Fedora, CentOS 5, openSUSE-11</td>
<td>RHEL 5, Debian, Fedora, CentOS 5, openSUSE-11</td>
<td>Ubuntu, ESX, Debian, RHEL, SUSE, Fedora</td>
<td>Ubuntu, Debian, RHEL, SUSE, Fedora</td>
</tr>
<tr>
<td>Contributors</td>
<td>VMWare</td>
<td>Eucalyptus systems, Community</td>
<td>C12G Labs, Community</td>
<td>Rackspace, IBM, HP, Red Hat, SUSE, Intel, AT&amp;T, Canonical, Nebula, others</td>
<td>Community</td>
</tr>
</tbody>
</table>
Cloud Middleware for HPC Workload

<table>
<thead>
<tr>
<th>Middleware</th>
<th>vCloud</th>
<th>Eucalyptus</th>
<th>OpenNebula</th>
<th>OpenStack</th>
<th>Nimbus</th>
</tr>
</thead>
<tbody>
<tr>
<td>License</td>
<td>Proprietary</td>
<td>BSD License</td>
<td>Apache 2.0</td>
<td>Apache 2.0</td>
<td>Apache 2.0</td>
</tr>
<tr>
<td>Supported Hypervisor</td>
<td>VMWare/ESX</td>
<td>Xen, KVM, VMWare</td>
<td>Xen, KVM, VMWare</td>
<td>Xen, KVM</td>
<td>Xen, KVM</td>
</tr>
<tr>
<td>Last Version</td>
<td>5.5.0</td>
<td>3.4</td>
<td>4.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programming Language</td>
<td>n/a</td>
<td>Java / C</td>
<td>Ruby</td>
<td>Java / Python</td>
<td></td>
</tr>
<tr>
<td>Host OS</td>
<td>VMX server</td>
<td>RHEL 5, ESX, Debian, Fedora, CentOS 5, openSUSE-11</td>
<td>RHEL 5, Debian, Fedora, CentOS 5, openSUSE-11</td>
<td>Ubuntu, ESX, Debian, RHEL, SUSE, Fedora</td>
<td>Ubuntu, Debian, RHEL, SUSE, Fedora</td>
</tr>
<tr>
<td>Guest OS</td>
<td>Windows (S2008,7), openSUSE, Debian, Solaris</td>
<td>Windows (S2008,7), openSUSE, Debian, Solaris</td>
<td>Windows (S2008,7), openSUSE, Debian, Solaris</td>
<td>Rackspace, IBM, HP, Red Hat, SUSE, Intel, AT&amp;T, Canonical, Nebula, others</td>
<td>Community</td>
</tr>
<tr>
<td>Contributors</td>
<td>VMware</td>
<td>Eucalyptus systems, Community</td>
<td>C12G Labs, Community</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Avg. Performance drop

<table>
<thead>
<tr>
<th>HPL</th>
<th>STREAM</th>
<th>RandomAccess</th>
<th>Graph500</th>
<th>Green500</th>
<th>GreenGraph500</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenStack+Xen</td>
<td>41.5%</td>
<td>19%</td>
<td>89.7%</td>
<td>21.6%</td>
<td>56.5%</td>
</tr>
<tr>
<td>OpenStack+KVM</td>
<td>58.6%</td>
<td>7.2%</td>
<td>67.5%</td>
<td>23.7%</td>
<td>38.5%</td>
</tr>
</tbody>
</table>

### Avg. Energy-efficiency drop

<table>
<thead>
<tr>
<th>GreenGraph500 PpW</th>
<th>Intel (Lyon)</th>
</tr>
</thead>
</table>

Cloud IaaS (OpenStack) on Mobile SoCs

Cloud IaaS (OpenStack) on Mobile SoCs

Cloud IaaS (OpenStack) on Mobile SoCs

Virtualization not suitable for pure HPC performance
✓ YET not all workloads running on HPC are pure-parallel

Virtualization, RJMS and HPC

On-demand optimization of computing platforms based on:
- Workload analysis
- User/Job characterization
- Performance Evaluation

Evalix
- monitoring
- energy-saving configuration
- Virtual resources configuration

Local Computing Resources
- Sleeping (powered off)
- Ready / Busy (power on)

RJMS
- (OAR, PBS etc.)
- scheduling

Remote Cloud Resources
- Virtualized / on the Cloud (running VMs instance)

Virtualization not suitable for pure HPC performance
✓ YET not all workloads running on HPC are pure-parallel

Other Middleware approaches

- Multi-Agent System (MAS) for energy aware executions

Software Trends: Rethinking Parallel Computing

- Reduced Power Consumption
- Hardware
- Middleware
- Data-center
- Software
Why is Exascale different for Software?

- Extreme power constraints, leading to:
  - ✓ clock rate similar to today’s systems
  - ✓ heterogeneous computing elements. Ex: IBM Power Cell
  - ✓ Memory per {core | Flops} will be **smaller**
  - ✓ Moving data will be expansive (time and power)

- HW→SW Fault detection/correction
  - ✓ becomes programmer’s job

- Extreme Scalability
  - ✓ $10^8 - 10^9$ concurrent threads
  - ✓ Performance is likely to be variable
    - static decomposition will not scale

![Graph showing failing probability F(t) over number of processors with execution time as a parameter]
# HPC Applications Compatibility Roadmap

<table>
<thead>
<tr>
<th>Application</th>
<th>Traditional (x86_64)</th>
<th>Traditional +GPU</th>
<th>Energy efficient ARMv7</th>
<th>CC</th>
<th>(C)ompute/(D)data intensive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Synthetic benchmarks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPCC</td>
<td>✓</td>
<td>TBI</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td>HPCG</td>
<td>✓</td>
<td>TBI</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td>Graph500</td>
<td>✓</td>
<td>TBI</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td><strong>Finite Element Analysis, Computational Fluid Dynamics software</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LS-DYNA</td>
<td>✓</td>
<td>TBI</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td>OpenFOAM</td>
<td>✓</td>
<td>TBI</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td><strong>Molecular dynamics applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMBER</td>
<td>✓</td>
<td>✓</td>
<td>TBI</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td>NAMD</td>
<td>✓</td>
<td>✓</td>
<td>TBI</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td><strong>Bio-informatics applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROMACS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td>ABBySS</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td>mpiBLAST</td>
<td>✓</td>
<td>× alt.: GPU-BLAST</td>
<td>✓</td>
<td>✓</td>
<td>D</td>
</tr>
<tr>
<td>MrBayes</td>
<td>✓</td>
<td>× alt.: GPU MrBayes</td>
<td>✓</td>
<td>✓</td>
<td>C</td>
</tr>
<tr>
<td><strong>Materials science software</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABINIT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td>QuantumESPRESSO</td>
<td>✓</td>
<td>✓ QE-GPU</td>
<td>✓</td>
<td>✓</td>
<td>C+D</td>
</tr>
<tr>
<td><strong>Data analytics and machine learning benchmarks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HiBench/Hadoop</td>
<td>✓</td>
<td>TBI</td>
<td>✓</td>
<td>✓</td>
<td>D</td>
</tr>
</tbody>
</table>
Rethinking Parallel Computing

- Today’s execution model might be obsolete
  - Von Neumann machine
    - Program Counter, Arithmetic Logic Unit (ALU), addressable memory
  - Classic vector machine, GPUs w. collec. of threads (Warps)

- Plan change in the execution model:
  - no assumption on performance regularity
    - not unpredictable but imprecise
  - synchronization is costly: don’t make it desirable
  - Memory operation are costly: move operations to data?
  - Represent key HW operations, beyond simple ALU
    - Remote update (RDMA), Remote atomic op. (compare & swap)
    - Execute short code sequence (active messages, parcels...
Challenges for Programming Models

- Probably successful: MPI, Map-Reduce
- Still pending challenges for exascale:
  - ✓ provide a way to coordinate resource allocation
  - ✓ clean way to share data with consistent memory models
  - ✓ Mathematical Model Guidance
    - ‣ continuous representation, possibly adaptative
    - ‣ lossy (within accuracy limits) yet preserving essential properties
  - ✓ Manage code by Abstract Data Structure Language (ADSL)
  - ✓ Adaptative with a multi-level approach
    - ‣ lightweight, locally optimized vs. intra node vs. regional
    - ‣ may rely on different programming models
Still a long way to go ;)

- New [co-]processors, interconnect...
- Virtualization, RJMS...

Data-center — Hardware — Middleware — Software

- Reduced Power Consumption
- PUE optim.
- DLC...
- New programming/execution models

Questions?