Evolution of the HPC Landscape in the Coming Future

Jordi Blasco (HPCNow!)
1. **New Hardware Technologies**
   - Architectures
   - New Fabric Technologies
   - IO Intensive workloads

2. **New Software Technologies**
   - Linux Containers in HPC

3. **Cloud Convergence**
   - New Hypervisor technologies

---

Jordi Blasco (HPCNow!)

Evolution of the HPC Landscape in the Coming Future
During the next couple of years, new exciting hardware and software technologies will define a completely new HPC landscape.

Jordi Blasco (HPCNow!)
Experience with early prototypes is key to making the most of new technologies before they are available in the regular market.
New Architectures

- Intel
- NVIDIA
- AMD
- ARM
- OpenPOWER

Jordi Blasco (HPCNow!) Evolution of the HPC Landscape in the Coming Future
Bigger systems

Future architectures will provide several cores per socket and several sockets per node.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Cores/Socket</th>
<th>Sockets/Node</th>
<th>GFLOPS/node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Skylake</td>
<td>up to 28</td>
<td>2,4,8</td>
<td>TBD</td>
</tr>
<tr>
<td>Intel KNL</td>
<td>72 (288)</td>
<td>1</td>
<td>+3TF</td>
</tr>
<tr>
<td>ARM ThunderX</td>
<td>24-48</td>
<td>2</td>
<td>TBD</td>
</tr>
<tr>
<td>POWER9</td>
<td>24</td>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>

**New Architectures**

**New and Improved Instruction Sets**

Future architectures will include more floating point operations per cycle.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Registers Size</th>
<th>Instruction Set</th>
<th>FLOPS/cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Westmere</td>
<td>128-bit</td>
<td>SSE2</td>
<td>4</td>
</tr>
<tr>
<td>Intel Sandy Bridge</td>
<td>256-bit</td>
<td>AVX</td>
<td>8</td>
</tr>
<tr>
<td>Intel Haswell</td>
<td>265-bit</td>
<td>AVX2</td>
<td>16</td>
</tr>
<tr>
<td>Intel Skylake</td>
<td>512-bit</td>
<td>AVX512</td>
<td>32</td>
</tr>
<tr>
<td>Intel KNL</td>
<td>512-bit</td>
<td>AVX512</td>
<td>32</td>
</tr>
</tbody>
</table>
New architectures will dramatically change the way we use HPC resources as well as the way we write code.
**New Challenges: NUMA Topologies**

*Figure:* From left to right: examples of two, four and eight sockets per node. First example uses two channels, second and third examples are using three channels.
Fine-grained NUMA topology allocation should be considered and scheduling mechanisms should be refined to provide less job scattering.
### Read After Write
- aka “flow” dependency
- Written first, then read
- Not vectorizable

```c
for (i=1; i<N; i++)
a[i] = a[i-1] + b[i];
```

### Write after Read
- aka “anti” dependency
- Read first, then written
- Vectorizable

```c
for (i=1; i<N-1; i++)
a[i] = a[i+1] + b[i];
```
New Fabric Technologies

New fabric technologies will enable new ways to communicate between nodes and will also redefine the concept of scalability.

Jordi Blasco (HPCNow!)
New Fabric Technologies

Intel Omni-Path
- Latency: 100-110ns
- Bandwidth: 100Gbps
- Ports/switch: 48
- Adaptive routing OPA2

Mellanox
- Latency: TBD
- Bandwidth: 100-200Gbps
- Ports/switch: 36
- Switch-IB 2 & SHArP
Network Topologies

Cost Effective Oriented

- Fat tree or ring topology (3-nodes) can reduce the cost significantly.
- Fat node cluster could reduce the cost in low latency network infrastructure.
- Topology aware scheduling can reduce network contention and enable multiple layers for MPI job allocation.
- New mechanisms to avoid or reduce scattering will be developed.
Cost Effective Low Latency Topologies

Three Nodes Ring Infiniband Topology with 8:1 Blocking

Jordi Blasco (HPCNow!) Evolution of the HPC Landscape in the Coming Future
Cost Effective Low Latency Topologies

Three Nodes Ring Omni-Path Topology with 11:1 Blocking

Jordi Blasco (HPCNow!) Evolution of the HPC Landscape in the Coming Future
Cost Effective Low Latency Topologies

Estimate Your Supercomputer’s Rank in the Top500

Figure: Simulated projection of HPL in the current Top500 list, based on Intel Skylake system with four sockets per node, 28 cores (112 cores / node) @ 2.6GHz (Rpeak ~ 9,318.4 GFLOPS/node) and assuming 60% efficiency.

Jordi Blasco (HPCNow!)
Network Topologies

Strong Scalability Oriented

- Suitable for applications and problems able to **efficiently** scale over 4,928 cores.
- Fat tree or Dragonfly topology.
- Technology based on Omni-Path, Omni-Path 2, Infiniband.
IO Intensive Requirements

Technologies and Solutions for Temporary IO

Usually, in order to get very good performance, several disks are required, but at the same time, temporary IO doesn’t require a lot of capacity.

- BeeGFS on demand (BeeOND)
- CRAY DataWarp
- SGI Zero-Copy Architecture
Linux containers will play a key role in application portability without impacting the level of performance of required in HPC workloads.
Linux Containers

Figure: Virtual Machines (left) require much more resources and higher operational cost than Linux containers (right). Linux containers only include the application and its dependencies.

Jordi Blasco (HPCNow!)
Docker Containers

Allows us to wrap up a piece of software in a complete filesystem that contains everything it needs to run (code, runtime, system tools, system libraries).
Binaries built in older architectures will not take advantage of new hardware instruction sets. Scientific code (as always) will be required to be compiled in the platform to be run.
Application building with Easybuild

EasyBuild is a software building and installation framework that allows management of (scientific) software on HPC systems.

Developed by the HPC team at Ghent University together with the members of the EasyBuild community. It is made available under GNU GPLv2. [http://hpcugent.github.io/easybuild/](http://hpcugent.github.io/easybuild/)
Current Challenges

Support for MPI jobs

The current implementations (Docker / Shifter) don’t allow more than one MPI job tenant per node.
At the same time new hypervisor technologies will open new opportunities to accommodate security and privacy requirements and also new features never explored before in HPC environments.
Linux Containers as a stand alone service can be very insecure compared to virtual machines. **LXD** opens new opportunities for privacy in standard HPC without reducing performance.

The LXD project was founded and is currently led by Canonical Ltd and Ubuntu with contributions from a range of other contributors. [https://linuxcontainers.org/lxd/](https://linuxcontainers.org/lxd/)
LXD

Features

- Secure by design
- Scalable
- Intuitive
- Architecture agnostic
- Image based
- Live migration
LXD will play a key role in the HPC and cloud convergency. OpenStack already supports LXD through compute-nova-lxd module.
Slurm federated cluster support will open new ways to collaborate between HPC centers and it will also reduce the complexity of cloud bursting.
Cloud and HPC convergency

Federated Cluster Support in Slurm

- Job migration
- Fault tolerance
- Cross-cluster job dependencies.
- Unified views.

As a result, less MPI job scattering and more efficiency.
Get ready for the coming challenges, get involved with the HPC community!
info@hpcnow.com
www.hpcnow.com
Almogàvers, 165 - 08018 Barcelona (Spain)
1A Taranto Place, 1071 Auckland (New Zealand)