A short overview of High Performance Computing

Stéphane Vialle

Stephane.Vialle@centralesupelec.fr
http://www.metz.supelec.fr/~vialle

What is « HPC » ?

High Performance Hardware

Numerical algorithms
Serial optimization – tiling
Vectorization
Multithreading
Message passing
Parallel programming
Intensive computing & large scale applications

High Performance Computing

« HPDA »

Big Data

TeraFlops
PetaFlops
ExaFlops

TeraBytes
PetaBytes
ExaBytes

High Performance Hardware

Inside …

… high performance hardware
High Performance Hardware

From core to SuperComputer

Computing cores with vector units
Multi-core processor
Multi-core PC/node
Multi-core PC cluster

Super-Computer
High Performance Hardware

From core to SuperComputer

- Computing cores with vector units
- Multi-core processor
- Multi-core PC/node
- Multi-core PC cluster
- Super-Computer + hardware accelerators

Top500 site: HPC roadmap

Required computing power

- Current machinery
- Expected machine
- Present day
- Machine installed
- New machine in production

- Regular improvement allows to plan the future

- Modeling and Numerical investigations
- Implementation, debug and test (on small pb)

Slow down?
Architecture issues

Why multi-core processors?  
Shared or distributed memory?

Re-interpreted Moore’s law

Processor performance increase due to parallelism since 2004:

CPU performance increase is due to (different) parallel mechanisms since several years… and they require explicit parallel programming

Auteur:
Jack Dongara

Impossible to dissipate so much electrical power in the silicium
It became impossible to dissipate the energy consumed by the semiconductor when the frequency increased!

Architecture issues
Re-interpreted Moore’s law

* Power = Voltage^2 x Frequency (V^2F)
* Frequency = Voltage
* Power = Frequency^2

To bound frequency and to increase the nb of cores is energy efficient

Auteur: Jack Dongara

New computer science interpretation:
each 24 months → x2 number of cores

Leads to a massive parallelism challenge:
to split many codes in 100, 1000, ….. 10^6 threads ... 10^7 threads!!

Initial (electronic) Moore’s law:
each 18 months → x2 number of transistors per µm^2

Previous computer science interpretation:
each 18 months → x2 processor speed

New computer science interpretation:
each 24 months → x2 number of cores
3 classic parallel architectures

Shared-memory machines (Symetric MultiProcessor):

One principle:
- several implementations,
- different costs,
- different speeds.

Distributed-memory machines (clusters):

Cluster basic principles, but cost and speed depend on the interconnection network!

Distributed Shared Memory machines (DSM):

Hardware implementation: fast & expensive…
Software implementation: slow & cheap!

Up to 1024 nodes
Support global multithreading

Cache coherence Non Uniform Memory Architecture (ccNUMA)
Extends the cache mechanism

Hardware implementation: fast & expensive…
Software implementation: slow & cheap!
Architecture issues

3 classic parallel architectures

- Simple and efficient up to ... 16 processors. Limited solution
- Unlimited scalability. But efficiency and price depend on the interconnect.
- Comfortable and efficient solution. Efficient hardware implementation up to 1000 processors.

2016: almost all supercomputers have a cluster-like architecture

Evolution of parallel architectures

% of computing power in Top500

2016: almost all supercomputers have a cluster-like architecture

Modern parallel architecture

Cluster architecture:

One PC cluster

One NUMA node
Modern parallel architecture

Cluster of NUMA nodes

Network

Time to access a data in the RAM of the processor
< Time to access a data in a RAM attached to another processor
(s several access times exist inside one NUMA node)
< Time to send/recv a data to/from another node

» Hierarchical architecture »:
→ [data – thread] location become critical

Multi-paradigms programming

Cluster of multi-processor NUMA nodes with hardware vector accelerators

Hierarchical and hybrid architectures:
→ multi-paradigms programming
(or new high level paradigm…)

AVX vectorization:
#pragma simd
+ CPU multithreading:
#pragma omp parallel for
+ Message passing:
MPI_Send(..., Me-1,...);
MPI_Recv(..., Me+1,...);
+ GPU vectorization:
myKernel<<<grid,bloc>>>(...);
+ checkpointing

Deployment problem

Distributed Software Architecture

• Achieving efficient mapping
→ How to map software and hardware resource?
• Achieving fault tolerance
→ Which strategy and mechanisms?
• Achieving scalability
→ Are my mapping and fault tolerance solutions adapted to larger systems?
What is a good interconnection network for a parallel computer?

**Main features**

- **Bandwidth**
- **Latency**
- **Contention & saturation resilience**
  - Many algorithms are synchronous ones: all nodes compute, and then enter a communication step at the same time.
- **Performances of Point-to-Point Communications**
- **Performances of Collective Communications**
  - Broadcast, scatter, gather, reduce, all_to_all, ...
- **Maximum latency and bandwidth variation between 2 nodes**
- **Extension capability (to increase machine size)**
  - Ex: hypercubic topology is hard/expensive to extend

**Interconnect in Top500 (Nov. 2016)**

<table>
<thead>
<tr>
<th>Interconnect</th>
<th>Function of machine peak performances</th>
<th>Function of machine core numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omnipath</td>
<td>Eth 100G</td>
<td>Eth 10G</td>
</tr>
<tr>
<td>Infiniband</td>
<td>Eth 10G</td>
<td>Eth 10G</td>
</tr>
<tr>
<td>Eth 1G</td>
<td>Infiband</td>
<td>Tofu 2</td>
</tr>
<tr>
<td>Custom</td>
<td>Eth 1G</td>
<td>Fujitsu</td>
</tr>
<tr>
<td>Intel</td>
<td>Eth 10G</td>
<td>Infiniband</td>
</tr>
<tr>
<td>Custom</td>
<td>Eth 10G</td>
<td>Eth 10G</td>
</tr>
<tr>
<td>Intel</td>
<td>Eth 100G</td>
<td>Eth 10G</td>
</tr>
</tbody>
</table>
Interconnect

Interconnect in Top500 (nov. 2017)

Function of machine max performances
Function of machine core numbers

- Eth-1G/10G → Eth 10G/25G
- Custom interconnects decrease
- Infiniband interconnects increase

- Custom vs Infiniband: changes regularly

10-Gigabit Ethernet vs Infiniband

10/25-Gigabit Ethernet:
- Used in many machines in Top500…
- …but not in the most powerful
- High latency
- Cheap interconnect!
- Well known technology (not only in HPC)
  → knowledge already exist in any company/institution

Infiniband:
- Used in many machines in Top500…
- …more powerful machines than Eth-10G machines
- Latency is lower than Eth-10/25G
- More expensive than Eth-10G (25G ?)
- Used only in HPC → special knowledge is required

Watch out: different versions of Infiniband exist, with different perf!

Proprietary custom networks

Proprietary custom networks

- Majors build SuperComputers with their proprietary Interconnect…
  …or customise high quality Infiniband (?)
- CRAY/IBM/Fujitsu/Chinese Supercomputers
- Different networks and network topologies in one machine
  pt-to-pt comm. network, collective comm. network, ctrl network
- They are the key component of a SuperComputer

Ex: Cray T3D has been the first SuperComputer to have an
interconnect fast enough for its processor computing power
Fault Tolerance in HPC

Can we run a very large and long parallel computation, and succeed?
Can a one-million core parallel program run during one week?

Fault tolerance
Mean Time Between Failures

MTBF definition:

\[ \text{MTBF} = \frac{\sum (\text{start of downtime} - \text{start of uptime})}{\text{number of failures}} \]

The Cray-1 required extensive maintenance. Initially, \textbf{MTBF was on the order of 50 hours}. MTBF is Mean Time Between Failures, and in this case, it was the average time the Cray-1 worked without any failures. Two hours of everyday was typically set aside for preventive maintenance.…… (Cray-1 : 1976)

System Resilience at Extreme Scale
White Paper
Prepared for Dr. William Harrod, Defense Advanced Research Project Agency (DARPA)
Today, 20% or more of the computing capacity in a large high-performance computing system is wasted due to failures and recoveries. \textbf{Typical MTBF is from 8 hours to 15 days.} As systems increase in size to field petascale computing capability and beyond, the MTBF will go lower and more capacity will be lost.

Addressing Failures in Exascale Computing
report produced by a workshop on "Addressing Failures in Exascale Computing" 2012-2013
Processor frequency is limited and number of cores increases
\rightarrow we use more and more cores

We do not attempt to speedup our applications
\rightarrow we process larger problems in constant time! (Gustafson’s law)

We use more and more cores during the same time
\rightarrow probability of failure increases!

We (really) need for fault tolerance
or large parallel applications will never end!

**Fault tolerance strategies**

**High Performance Computing:** big computations (batch mode)
\rightarrow Checkpoint/restart is the usual solution
\rightarrow Complexify src code, time consuming, disk consuming!

**High Throughput Computing:** flow of small and time constrained tasks
\rightarrow Small and independent tasks
\rightarrow A task is re-run (entirely) when failure happens

Fault tolerance in HPC remains a « hot topic »

**Big Data:**
\rightarrow Data storage redundancy
\rightarrow Computation on (frequently) incomplete data sets …

**Who need for fault tolerance?**

In a HPC cluster: computing resources are checked regularly
\rightarrow Wrong resources are identified and not allocated
\rightarrow Many users do not face frequent failures (good!)
(parallel computers are not so bad!)

Which users/applications need for fault tolerance?
\rightarrow When running applications on large numbers of resources during long times
\rightarrow Need to restart from a recent checkpoint

Remark:
- Critical parallel applications (with strong deadlines)
- Need for redundant resources and runs
- Impossible on very large parallel runs

Fault tolerance

High availability
Energy Consumption

1 PetaFlops: 2.3 MW !
⇒ 1 ExaFlops: 2.3 GW !! 350 MW ! ….. 20 MW ?

Perhaps we will be able to build the machines, but not to pay for the energy consumption !!

Energy consumption

From Petaflops to Exaflops

1.00 Exaflops : 2018-2020
2020-2022

×1000 perf
× 100 cores/node
× 10 nodes
× 50 IO
× 10 energy (only × 10)

122 Petaflops : juin 2018
Summit – IBM, Oak Ridge - USA
IBM POWER9 22C 3.07GHz
NVIDIA Volta GV100
2 282 544 « cores »
8.8 M Watt

1.03 Petaflops : June 2008
RoadRunner (IBM)
Opteron + PowerXCell
122440 « cores »
500 Gb/s (IO)
2.35 M watt !!!!!!!

• How to program these machines ?
• How to train large developer teams ?

Energy consumption

Sunway TaihuLight - China: N°1 2016 - 2017
93.0 P f l o p s
• 41 000 processeurs Sunway SW26010 260C 1.45GHz
⇒ 10 649 600 « cores »
• Sunway interconnect:
5-level integrated hierarchy
(Infiniband like ?)

15.4 M Watt
122.3 PFlops (×1.31)
• 9 216 processeurs IBM POWER9 22C 3.07GHz
• 27 648 GPU Volta GV100
  ⇒ 2 282 544 « cores »
• interconnect: Dual-rail Mellanox EDR InfiniBand

8.8 MWatt (×0.57)
Flops/Watt : ×2.3

Cooling

Cooling is close to 30% of the energy consumption
Optimization is mandatory!

Cooling is strategic !

Des processeurs moins gourmands en énergie :
• on essaie de limiter la consommation de chaque processeur
• les processeurs passent en mode économique s’ils sont inutilisés
• on améliore le rendement flops/watt

Mais une densité de processeurs en hausse :
• une tendance à la limitation de la taille totale des machines (en m² au sol)

⇒ Besoin de refroidissement efficace et bon marché (!)
Souvent estimé à 30% de la dépense énergétique!
**Liquid and immersive cooling**

Refroidissement par immersion des cartes dans un liquide électriquement neutre, et refroidi.

Cray 2 (1985)

Refroidissement liquide par immersion sur le CRAY-2 en 1985

Refroidissement liquide par immersion testé par SGI & Novec en 2014

**Cold doors (air+water cooling)**

On refroidit par eau une « porte/grille » dans laquelle circule un flux d'air, qui vient de refroidir la machine.

Le refroidissement se concentre sur l'armoire.

**Direct liquid cooling**

On amène de l'eau froide directement sur le point chaud, mais l'eau reste isolée de l'électronique.

- Expérimental en 2009
- Adopté depuis (IBM, BULL, …)

Carte expérimentale IBM en 2009
(projet Blue Water)

Lame de calcul IBM en 2012
(Commercialisée)
Refroidissement avec de l’air à température ambiante :
- circulant à grande vitesse
- circulant à gros volume

⇒ Les CPUs fonctionnent proche de leur température max supportable (ex : 35°C sur une carte mère sans pb)

⇒ Il n’y a pas de refroidissement du flux d’air.

Economique !
Mais arrêt de la machine quand l’air ambiant est trop chaud (l’été) !

Installation Ilium à CentraleSupélec à Metz (blockchain - 2018)
HPC in the cloud

Microsoft solution: AZUR BigCompute
- High performance nodes
- High performance interconnect (Infiniband)
  - Customers can allocate a partition of HPC clusters

AWS allows to allocate a huge number of nodes for short time
  - Comfortable for scaling benchmarks

But running HPC codes in clouds requires some adaptations:
- **Rental price** is divided by 5 if nodes can be pre-empted!
  - need to run kinds of fault-tolerant codes
  - need to run codes able to continue on a smaller or greater number of resources
- Expensive features of the cloud evolve and codes need to adapt
  Ex: disk Bw becomes expensive → improve the disk IO strategy

Architecture des machines parallèles
Histoire des ordinateurs CRAY

- Cray-1, 1976: 133Mflops
- Cray-2, 1985: 1.9 gigaflops
- Cray-YMP, 1988
- Cray-C90, 1991: 16 gigaflops
- Cray-190, 60 gigaflops

Interesting history of CRAY company

If you were plowing a field, which would you rather use? Two strong oxen or 1024 chickens?

- SuperNat. Cray
Cray est démembre et semble avoir disparu. Puis en 2002 un événement survient ….

Architecture des machines parallèles
Histoire des ordinateurs CRAY

Apparition du Earth Simulator : gros cluster vectoriel NEC :
• 640-nœuds de 8 processeurs : 5120 processeurs
• 40 Tflops crête, a atteint les 35 Tflops en juin 2002

"Vector MPP"

Le vectoriel revient à la 1ère place du Top500 (en 2002) !

Architecture des machines parallèles
Histoire des ordinateurs CRAY

Japan's Impressive Earth Simulator Is As Fast As the Top 20 U.S. Supercomputers Combined

Forte inquiétude des USA !
CRAY était de nouveau là, avec de grandes ambitions:

**Architecture des machines parallèles**

**Histoire des ordinateurs CRAY**

- **1900-2010: Supercomputer speed increases will outpace Moore’s Law progress by 100x**

<table>
<thead>
<tr>
<th>Year</th>
<th>Supercomputer speed</th>
<th>Moore’s Law progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>1900</td>
<td>1000 flops</td>
<td>1000 flops</td>
</tr>
<tr>
<td>2000</td>
<td>10,000,000,000 flops</td>
<td>100,000,000,000 flops</td>
</tr>
<tr>
<td>2010</td>
<td>10,000,000,000,000 flops</td>
<td>1,000,000,000,000,000 flops</td>
</tr>
</tbody>
</table>

- **Cray-T90**, 60 gigaflops
- **Cray-SV11 teraflop**
- **Cray-SV2**
- **Cray-SV3**
- **Cray-SX-6**
  - 52.4 Tflops
- **Vector MPP**
- **Machine annoncée**
- **Cray-X1** – 52.4 Tflops
- **Cray-X1 – 52.4 Tflops**
- **Cray-XT3**
- **Cray-XT4**
- **Cray-XT5**
  - cluster de CPU multicœurs, Linux
- **Cray-XT6**
  - Opteron 6-cores
  - Tore 2D
  - Réseau Cray
- **Cray-XT5h (hybrid)**
  - cluster de nœuds
  - CPU/Vectoriels/FPGA
  - Unicos (Cray Unix)
Cray XT6 :
1er au top500 en novembre 2009 : 1.7 Pflops avec 6.9 Mwatt
Architecture : réseau d’interconnexion propriétaire + Opteron 6-cœurs
→ Architectures traditionnelles et très consommatrices d’énergie mais très efficace et sous Linux (logiciels disponibles)
Machine dénommée « Jaguar »

Cray XK7 : entrée des accélérateurs dans la gamme CRAY
1er au top500 en novembre 2012 : 17.6 Pflops avec 8.2 Mwatt
Architecture :
• réseau d’interconnexion propriétaire
• chaque nœud : Opteron 16-cœurs + GPU NVIDIA Tesla K20
• 18688 nœuds → 299008 CPU cores + 18688 GPU K20

Cray à la 1ère place en nov 2012 avec Opteron + GPU

Cray à la 3ème place en juin 2016 avec Opteron + GPU
A short overview of
High Performance Computing

Questions?