A short overview of High Performance Computing

Stéphane Vialle

What is « HPC » ?

High performance hardware
Numerical algorithms
   - Serial optimization – tilling
   - Vectorization
   - Multithreading
   - Message passing
Parallel programming
Intensive computing & large scale applications
Big Data

High Performance Computing

« HPDA »

TeraFlops
PetaFlops
ExaFlops

TeraBytes
PetaBytes
ExaBytes
High Performance Hardware

Inside ….

… high performance hardware

From core to SuperComputer

Computing cores
High Performance Hardware

From core to SuperComputer

Computing cores with vector units

High Performance Hardware

From core to SuperComputer

Computing cores with vector units

Multi-core processor
High Performance Hardware

From core to SuperComputer

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

High Performance Hardware

From core to SuperComputer

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

Fat Tree interconnect

+ hardware accelerators
High Performance Hardware

Top500 site: HPC roadmap

Required computing power

Current machine

Regular improvement

Expected machine

Present day

Machine installed

New machine in production

Modeling and Numerical investigations

Implementation, debug and test (on small pb)

⇒ Regular improvement allows to plan the future
HPC in the cloud?

Microsoft Azure
- “AZUR BigCompute”
  - High performance nodes
  - High performance interconnect (Infiniband)
  - Customers can allocate a part of a HPC cluster

AWS
- Allows to allocate a huge number of nodes for a short time
- No high performance interconnection network
  - Comfortable for Big Data scaling benchmarks

Some HPC or Large Scale PC-clusters exist in some Clouds
But no SuperComputer available in a cloud

Architecture issues

Why multi-core processors?  
Shared or distributed memory?
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

**Re-interpreted Moore’s law**

**Architecture issues**

Impossible to dissipate so much electrical power in the silicium

Auteur : Jack Dongara
It became impossible to dissipate the energy consumed by the semiconductor when the frequency increased!

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

\[ 2 \times 0.75^3 = 0.8 \]

Auteur : Jack Dongara
Re-interpreted Moore’s law

Initial (electronic) Moore’s law:
each 18 months $\rightarrow$ x2 number of transistors per $\mu$m²

Previous computer science interpretation:
each 18 months $\rightarrow$ x2 processor speed

New computer science interpretation:
each 24 months $\rightarrow$ x2 number of cores

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

3 classic parallel architectures

Shared-memory machines (*Symetric MultiProcessor)*:

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

Overview of Recent
Supercomputers
Aad J. van der Steen
Jack J. Dongarra²⁰
3 classic parallel architectures

Distributed-memory machines (*clusters*):

- Hypercubes
- Fat trees
- Gigabit Ethernet

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

Distributed Shared Memory machines (*DSM*):

- cache coherence Non Uniform Memory Architecture (ccNUMA)
- Extends the cache mechanism
- Up to 1024 nodes
- Support global multithreading

Hardware implementation: fast & expensive…
Software implementation: slow & cheap!
3 classic parallel architectures

- **Shared memory « SMP »**
  - Simple and efficient up to … 16 processors. Limited solution.

- **Distributed memory « Cluster »**
  - Unlimited scalability. But efficiency and price depend on the interconnect.

- **Distributed shared memory « DSM »**
  - Comfortable and efficient solution. Efficient hardware implementation up to 1000 processors.

2016: almost all supercomputers have a **cluster-like architecture**.

---

Architecture issues

**Evolution of parallel architectures**

- **% of computing power in Top500**
  - 1993: Pure SIMD, MPP, Single processor, ShM vector machine, Cluster
  - 2017: Distribution changing

- **% of 500 systems in Top500**
  - 1993: Pure SIMD, MPP, Single processor, ShM vector machine, Cluster
  - 2017: Distribution changing

2016: almost all supercomputers have a **cluster-like architecture**.

**BUT …**
Architecture issues

Modern parallel architecture

Cluster architecture:
One PC cluster

Cluster of NUMA nodes

One NUMA node

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

⇒ « Hierarchical architecture »:
⇒ {data – thread} location become critic
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

Hierarchical and hybrid architectures:

EU/France China USA

• Achieving efficient mapping
  How to map software and hardware resource?

• Achieving fault tolerance
  Which strategy and mechanisms?

• Achieving scalability
  Are my algorithm, mapping and fault tolerance strategy adapted to larger systems?
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:

\[ \text{Mean time between failures} = \text{MTBF} = \frac{\sum (\text{start of downtime} - \text{start of uptime})}{\text{number of failures}} \]
Fault tolerance

Mean Time Between Failures

Experiments:

The Cray-1 required extensive maintenance. Initially, 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. 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

Fault tolerance

Why do we need fault tolerance?

Processor frequency is limited and number of cores increases → we use more and more cores

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

We use more and more cores during the same time → 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)
- **Checkpoint/restart** is the usual solution
- **Complexify src code**, time consuming, disk consuming!

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

Fault tolerance in HPC remains a « hot topic »

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

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

How much electrical power for an Exaflops?

1.0 Exaflops should be reached close to 2020:

- 2.0 GWatts with the flop/watt ratio of 2008 Top500 1st machine
- 1.2 GWatts with the flop/watt ratio of 2011 Top500 1st machine
- 350 MWatts if the flop/watt ratio increases regularly
- 20 MWatts if we succeed to improve the architecture? …
  … « the maximum energy cost we can support ! » (2010)
- 2 MWatts …
  … « the maximum cost for a large set of customers » (2014)

Energy consumption

From Petaflops to Exaflops

1.00 Exaflops: 2018–2020

2020–2022

25 Tb/s (IO)
20/35 MWatt max….

122 Petaflops: juin 2018

Summit – IBM, Oak Ridge - USA
IBM POWER9 22C 3.07GHz
NVIDIA Volta GV100
2 282 544 « cores »
8.8 MWatt

1.03 Petaflops: June 2008

RoadRunner (IBM)
Opteron + PowerXCell
122440 « cores »
500 Gb/s (IO)
2.35 MWatt !!!!!!

- How to program these machines?
- How to train large programmer teams?
Sunway TaihuLight - China: N°1 2016 - 2017

93.0 Pflops
- 41 000 processors Sunway SW26010 260C 1.45GHz → 10 649 600 « cores »
- Sunway interconnect:
  5-level integrated hierarchy
  (Infiniband like ?)

15.4 MWatt

Energy consumption

Summit - USA: N°1 June 2018

122.3 Pflops (∗1.31)
- 9 216 processors 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
Energy consumption

**Summit - USA: N°1 November 2018**

143.5 Piflops ($\times 1.54$)
- 9 216 processors IBM POWER9 22C 3.07GHz
- 27 648 GPU Volta GV100
  $\Rightarrow$ 2 282 544 « cores »
- interconnect: Dual-rail Mellanox EDR Infiniband

9.8 MWatt ($\times 0.64$)

Flops/Watt : $\times 2.4$

**What is the sustainable architecture ?**

**Différentes stratégies s’affrontent dans le Top500 :**

- La performance à tous prix avec de gros CPUs très gourmands
  
  Cray XT6 : 1.7 Pflops, 6.9 Mwatts
  
  K-Computer : 10.5 Pflops, 12.6 MWatts

- Beaucoup de processeurs moyennement puissants et peu gourmands
  
  IBM Blue Gene (gamme terminée)

- Utilisation d’accélérateurs matériels : GPU, Xeon-phi, …
  
  machines hybrides : CPU + accélérateurs
  
  difficiles à programmer et pas adaptées à tous les problèmes

**Quel est le(s) bon(s) choix pour atteindre l’Exaflops ?**

**Quel est le choix pertinent pour de « plus petits » clusters ?**
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!
Cooling

Optimized air flow

Optimisation des flux d’air : en entrée et en sortie des armoires

• Architecture Blue Gene : haute densité de processeurs
• Objectif d’encombrement minimal (au sol) et de consommation énergétique minimale
• Formes triangulaires ajoutées pour optimiser le flux d’air

IBM Blue Gene

---

Cooling

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, …)

Liquid and immersive cooling

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

- Cray 2 (1985)
  - 4 processeurs
  - 1.9 Gflops
  - Fluorocarbon

Refroidissement liquide par immersion sur le CRAY-2 en 1985

Refroidissement liquide par immersion testé par SGI & Novec en 2014
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) !
Interesting history of CRAY company

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

A short overview of High Performance Computing

Questions?