Current Trends in High-Performance Computing

Celso L. Mendes
INPE
São José dos Campos – SP, Brazil
Email: celso.mendes@inpe.br

October 11, 2018
Career Path:

• **INPE**, starting in 1978 as an intern; some periods at **Univ.Illinois**, USA
• Currently a professor at INPE’s graduate program – **CAP** (applied comput.)
1. Introduction
2. Moore’s Law, Processor Evolution
3. Top500 and Alternatives
4. Notable HPC Machines (Brazil and Abroad)
5. Main Programming Paradigms
6. Current Trends and Challenges
7. Conclusion
Motivations for HPC

Economical Motivation:
• Impact on everyone’s life

Scientific Motivation:
• Acceleration of discoveries
1. Introduction
2. Moore’s Law, Processor Evolution
3. Top500 and Alternatives
4. Notable HPC Machines (Brazil and Abroad)
5. Main Programming Paradigms
6. Current Trends and Challenges
7. Conclusion
Gordon Moore (Fairchild, Intel), 1965:

- Prediction about future increase in **density** of chips

Rate of increase constant across time!
Doubles @ 2 years

Moore’s Law in practice:

• Beginning of *Multi-Core* era in 2005
Current Multi-Core Era

- **x86 Processors:**
  - Intel: Xeon Skylake – 28 cores
  - AMD: Rome – 48 or 64 cores

- **Other Processors:**
  - IBM: Power9 – 24 cores
  - ARM: several, up to 48 cores

- **Accelerators:** (aka Many-Core)
  - GPUs – e.g.: Nvidia: Kepler, Pascal, Volta, ???
  - Intel Xeon-Phi – KNC, KNL
1. Introduction
2. Moore’s Law, Processor Evolution
3. Top500 and Alternatives
4. Notable HPC Machines (Brazil and Abroad)
5. Main Programming Paradigms
6. Current Trends and Challenges
7. Conclusion
Top500 List

• **Goal**
  • List the 500 “fastest” supercomputers in the world
  • Focus on numerical capability – Linpack execution

• **Periodicity**
  • 2 editions per year: June (ISC-Europe) & November (SC-USA)
  • Started in 1993

• **Participation Process**
  • Execution of unchanged Linpack code
  • Report values of $R_{max}$ in flops/s
  • Report also $R_{peak}$: theoretical maximum

<table>
<thead>
<tr>
<th>Abbrev.</th>
<th>Unity</th>
<th>Flops/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF</td>
<td>Megaflops</td>
<td>$10^6$</td>
</tr>
<tr>
<td>GF</td>
<td>Gigaflops</td>
<td>$10^9$</td>
</tr>
<tr>
<td>TF</td>
<td>Teraflops</td>
<td>$10^{12}$</td>
</tr>
<tr>
<td>PF</td>
<td>Petaflops</td>
<td>$10^{15}$</td>
</tr>
<tr>
<td>EF</td>
<td>Exaflops</td>
<td>$10^{18}$</td>
</tr>
</tbody>
</table>
Current Top500 Edition

• Latest edition: June/2018 (www.top500.org)
  • System #1: IBM Power9 + GPU Nvidia Volta
  • System #2: Chinese processor, many-core, RISC
  • System #3: IBM Power9 + GPU Nvidia Volta
  • System #4: Intel Xeon + coprocessor Matrix-2000
  • System #5: Intel Xeon + GPU Nvidia Volta

<table>
<thead>
<tr>
<th>Position</th>
<th>System</th>
<th>Country</th>
<th>$R_{\text{max}}$ (PFlops)</th>
<th>$R_{\text{peak}}$ (PFlops)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Summit (IBM)</td>
<td>USA</td>
<td>122.30</td>
<td>187.66</td>
</tr>
<tr>
<td>2</td>
<td>Sunway TaihuLight (NRCPC)</td>
<td>China</td>
<td>93.01</td>
<td>125.44</td>
</tr>
<tr>
<td>3</td>
<td>Sierra (IBM)</td>
<td>USA</td>
<td>71.61</td>
<td>119.19</td>
</tr>
<tr>
<td>4</td>
<td>Tianhe-2A (NUDT)</td>
<td>China</td>
<td>61.44</td>
<td>100.68</td>
</tr>
<tr>
<td>5</td>
<td>ABCI (Fujitsu)</td>
<td>Japan</td>
<td>19.88</td>
<td>32.58</td>
</tr>
</tbody>
</table>

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Top500 List - Geography

• Participation by country: June/2018 (number of systems)
  • China >> EUA!

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Top500 List – Geography / Past

• Participation by country: evolution (number of systems)
Top500 List - Evolution

• Evolution across the years:

- All 500 (aggregate performance)
- System #1
- System #500

Improvement greater than Moore’s Law!
(technological and architectural factors)
Alternatives to Top500

• **HPCG List** ([www.hpcg-benchmark.org](http://www.hpcg-benchmark.org))
  - **Goal**: Complement Linpack with more than floating-point evaluation (e.g. memory, network, ...)
  - **Metric**: Flops/s
  - **Code**: C++, MPI+OpenMP
  - **June/2018 Edition**: #1 IBM Summit – 2.9 Pflops/s (1.5% of peak!)

• **Green500 List** ([www.green500.org](http://www.green500.org))
  - **Goal**: Evaluate energetic efficiency in numerical processing
  - **Metric**: Flop/Watt
  - **May/2016 and beyond**: unified submission with Top500
  - **Nov./2018 edition**: #1: Shoubu-B (Japan), 18.4 GF/W
SPP Benchmarks (bluewaters.ncsa.illinois.edu/spp-benchmarks)
- **Goal:** Evaluate sustained performance of real scientific applications
- **Metric:** SPP Index (Sustained Petascale Performance)
- **Codes:** Fortran/C/C++, MPI & OpenMP, CPU and GPU
- **Execution:** includes I/O and checkpointing (like production)

Graph500 List (www.graph500.org)
- **Goal:** Evaluation of non-numerical processing (e.g. graphs)
- **Metric:** Speed of graph search – *teps: traversed edges per second*
- **June/2018 edition:** #1 = K Computer (Fujitsu, Japan), 38.6K gteps
1. Introduction
2. Moore’s Law, Processor Evolution
3. Top500 and Alternatives
4. Notable HPC Machines (Brazil and Abroad)
5. Main Programming Paradigms
6. Current Trends and Challenges
7. Conclusion
Top500 #1,#3: Summit, Sierra

  - CORAL: Collaboration of Oak Ridge, Argonne, Livermore
  - Budget of US$ 520 M
  - HPC support for 3 national labs in the USA
    - Oak Ridge: Summit (#1) – 188 PF
    - Livermore: Sierra (#3) – 119 PF
    - Argonne: Aurora, planned for 2020/2021 – 1 Exaflop
      - Current System: Theta – Cray XC40, 11.69 PF, Intel Xeon-Phi
      - Delay in Aurora due to closing of Xeon-Phi line by Intel
  - Goal: Systems with performance beyond 1 Exaflop in 2021~2023
  - Budget of US$ 1.8 B
  - Selections expected in Oct~Nov/2018
Top500 #1: Summit

• Location: Oak Ridge National Lab.
  • Vendors: IBM + Nvidia
  • 4,608 compute nodes
    • 2 Power9 processors @ 3.1 GHz per node
      • $2 \times 22 = 44$ cores in each node
    • 6 GPUs Nvidia Volta per node
  • Total system: 9,216 Power9s, 27,648 GPUs
  • Interconnection network: Infiniband
  • Energy consumption: 8.8 MW on Linpack
  • Details: https://www.olcf.ornl.gov/summit/
Top500 #3: Sierra

- **Location:** Lawrence Livermore National Lab.
  - Vendors: IBM + Nvidia
  - 4,320 compute nodes
    - 2 Power9 processors @3.1 GHz per node
      - \(2 \times 22 = 44\) cores in each node
    - 4 GPUs Nvidia Volta per node
  - Total system: 8,640 Power9s, 17,280 GPUs
- Interconnection network: Infiniband
- Maximum energy consumption: \(\sim 12\) MW
- Details: [https://hpc.llnl.gov/hardware/platforms/sierra](https://hpc.llnl.gov/hardware/platforms/sierra)
Top500 #2: Sunway TaihuLight

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Top500 #2: Sunway TaihuLight

- **Location:** Wuxi, China
  - Installed at the supercomputing center of Wuxi, 2016
- Total of 10,649,600 cores, in 40,960 nodes (1 chip per node)
  - Total peak performance: 125 Pflops/s, in 40 racks
  - Linpack performance: 93 Pflops/s (~ 74.4% of peak)
    - Linpack: numerically intensive code
  - HPCG performance: 0.3% of the peak!
    - HPCG: numerical code + memory access + communication
- Conclusion: extremely unbalanced system
Top500 #2: Sunway TaihuLight

- Processor: SW26010 (Shanghai, China)
  - Architecture: many-core, RISC
  - Peak performance of 11.6 Gflops/s for each core
  - 4 groups of cores
    - each group: 64 cores (CPE), plus a control unit (MPE)
  - 260 cores per chip: > 3 TFlops/s per chip
  - 32 GB external memory per node, 1.2 PB total
Blue Waters System – Univ.Illinois

Cray XE6/XK7

Aggregate Memory – 1.6 PB

10/40/100 Gb Ethernet Switch

External Servers

IB Switch

>1 TB/sec

120+ Gb/sec

100 GB/sec

Spectra Logic: 300 usable PB

Sonexion: 26 usable PB

Deployment: 2012
NCSA, Urbana-IL
Funding: U.S. NSF

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Blue Waters by Numbers

- Total number of cabinets: 288
  - XE cabinets: 243
  - XK cabinets: 45
- Peak performance: 13.3 PF
  - x86 CPUs: 7.1 PF
  - GPUs: 6.2 PF
- Memory: 1.6 PB
- Disk: 26 PB for users (8+2 ECC)
- Tape archival: ~300 PB
- Power consumption: ~10 MW
Blue Waters Interconnection

Interconnection: Gemini network
Torus 3D
24 x 24 x 24

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Sample of Blue Waters Users (2014)

PRAC:
Petascale Allocations (80% of cycles)
Selected yearly by NSF
Brazil: Santos Dumont - LNCC

• Major System Features:
  • Aggregate peak performance: over 1 Pflops/s
  • Vendor: Bull (France)
  • Physical installation: containers
Santos Dumont System (cont.)

• Architecture:
  • Diversity of processors: CPU, GPU, Xeon-Phi
  • 3 sub-systems listed on Top-500 of Nov/2015, none today
  • Four types of compute nodes:
    a) 504 nodes with 2 Intel Xeon CPUs – 24 cores/node
    b) 198 nodes with 2 Intel Xeon CPUs and 2 GPUs Nvidia K40
    c) 54 nodes with 2 Intel Xeon CPUs and 2 Intel Xeon-Phi (KNC)
    d) 1 node with 16 Intel Xeon CPUs (240 nodes), 6 TB RAM
  • Interconnection network: Infiniband
  • Access: allocations available to the community
  • Details: https://sdumont.lncc.br
HPC Systems at INPE

• **Tupã System:** Cray XE6, deployed in 2010
  • @CPTEC: Centro de Previsão do Tempo e Estudos Climáticos
  • Peak performance ≈ 258 Tflops/s, ~30,000 cores AMD Opteron
  • #29 on Top500 list of November/2010! (out of current list)
  • Approaching end of life
    • 8 years of good services!
    • Cray-XE out of production
    • 6 (of 14) cabinets turned off

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
• New Partition: XC50 (2018)
  • 102 nodes, 2 Intel Skylakes
    • Cores: 20/chip, 40/node
    • Total: 4,080 Intel cores
    • Peak perform.: 313 Tflops/s
    • New network: Aries
• Original Tupã reconfigured:
  • 8 cabinets still working
  • 6 cabinets retired (spare)
  • Total: ~17,800 AMD cores
  • Peak performance: 258 → 147 Tflops/s
CPTEC Now:

TUPÃ

Network 10 Gbps

/stonext
3.9 Pbytes (raw)
29 file systems

13 x eslogin
20 x aux’s

6 x Qlogic 20 ports FC switch (8 Gbps)

4 cabinets
Controllers LSI 7900

8 cabinets
Tape library T-Finity
8000 Tapes

XE6

XDP 2
XDP 3
C0-1
C1-1
C2-1
C3-1

XDP 0
XDP 1
C0-0
C1-0
C2-0
C3-0

16 Cables FC
24 Cables FC
4 Cables FC

/mds1
/scratchin
344 TB

/scratchout
516 TB

Spare Nodes

c4-1
c5-1
c6-1

c4-0
c5-0
c6-0

XC50

Sonexion
960TB

6 Cables IB

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Other HPC Systems at INPE

- Kerana (Cray XE6: 1 rack)
  - Climate studies

- LACHibrido cluster (research)
  P.I.: Prof. Haroldo F. C. Velho, LAC/INPE
1. Introduction
2. Moore’s Law, Processor Evolution
3. Top500 and Alternatives
4. Notable HPC Machines (Brazil and Abroad)
5. Main Programming Paradigms
6. Current Trends and Challenges
7. Conclusion
Scaling Challenge:
• Ex: System #2 on Top500: 10 million cores!

In principle, there are 4 levels of parallelism to explore:
i. Inside a CPU: vectorization
ii. Between CPUs sharing memory (intra-node)
iii. Between CPUs with distinct memories (inter-node)
iv. Between CPU and accelerator(s)
Vectorization - History

• Between 1990~2000: x86 Processors
  • Vectorization enabled by SSE extensions
    • SSE: *Streaming SIMD Extension*
    • Support in hardware, **vector registers** with L=4 or more
  • Similar concept to old vector machines:
    **A single instruction triggers several operations in concert**
  • Some compilers have extensive vectorization support
  • In some cases, one may need to *help* the compiler
    • Code restructuring, elimination of dependencies
    • Assertions, by the programmer, of no dependencies in a loop
“Modern” Vectorization

• **AVX: Advanced Vector Extensions**
  • Several levels so far: 128, 256, 512 bits
    • 2, 4, 8 values in double precision (4, 8, 16 in single precision)
  • Present in several Intel processors:
    • Ivy Bridge processor (Santos Dumont): AVX (128 bits)
    • Haswell and Broadwell processors: AVX-2 (256 bits)
    • KNL processor (Xeon-Phi): AVX-512
  • Present in some AMD processors
Vector Operations / SIMD

SIMD Mode

A7 A6 A5 A4 A3 A2 A1 A0

+  

B7 B6 B5 B4 B3 B2 B1 B0

=  

A7+B7 A6+B6 A5+B5 A4+B4 A3+B3 A2+B2 A1+B1 A0+B0

Scalar Mode

A

+  

B

=  

A+B

source: Intel
Shared-Memory Parallelism

Main Technique: use of threads

- Execution Model: fork/join
Two typical ways of implementing threads:

a) Managed by the programmer: e.g. *pthreads*
   - Creation/removal of threads is explicit in the code
   - Allows maximal flexibility to the programmer

b) Managed by the compiler: OpenMP standard
   - Programmer inserts directives in serial code
   - Compiler processes the directives, creates threads
   - Allows “gradual” parallelization of existing codes
Solar Radiation Model:
- *Brasil-SR* model, adapted to Brazil
- Used to assess solar potential
- 8,300 lines of Fortran code
- More than 25 hours in serial mode
- Multi-core performance:

OpenMP Parallelization @ INPE (2)

Hidrological Model:

- MGB model (*Modelo de Grandes Bacias*)
- Developed originally at IPH-UFRGS
- Hydrological processes in large-scale watersheds
  - Simulates 1D propagation of water flows
- 53 Fortran90 files
- Two test cases:
  - Purus river (Amazon), Niger river (Africa)
- Parallelization:
  - OpenMP (CPU) or OpenACC (GPU)
  - Ongoing: hybrid execution

Traditional Technique: message passing

• Typical implementation: use of **MPI**
  - MPI - mature standard: v.3.1 now, 4.0 in discussion
• Programmer must insert **sends/receives** in the code
• Common execution model: SPMD
• MPI libraries widely available, both from vendors as in public domain
• Model **MPI+X** is viewed by some as popular in the future
  - X = ?
  - So far, X = OpenMP
Emerging Technique: PGAS Languages

- **PGAS**: Partitioned Global Address Space
- Provide to each processor the *illusion* of global memory

Goals:

- Raise the level of abstraction for programming distributed-memory systems
- Avoid the need to deal with message passing in the source program

Implementations:

- CoArray Fortran (Fortran2008), Unified Parallel C (UPC), Chapel (Cray), X10 (IBM), etc.
PGAS Languages

• Basics:
  • All memories are part of the “global” address space
  • For each processor, one range of addresses is local (fast access), the other ranges are remote (slow access)
  • ProcK: Mem-K=local (fast), Mem-j(j≠K)=remote (slow)
CoArray Fortran Example

Fortran2008 Program

```fortran
program reduce
    integer :: my_rank, num_procs, i, result
    integer :: coarray[*]

    my_rank = this_image()
    num_procs = num_images()
    if ( (my_rank/2)*2 .EQ. my_rank ) then
        coarray[my_rank] = my_rank;
    else
        coarray[my_rank] = my_rank * (-1);
    end if
    result = 0
    sync all
    if (my_rank .EQ. 1) then
        do i=1, num_procs
            result = result + coarray[i]
        end do
    end if
    sync all

    if (my_rank .EQ. 1) print *,"CAF_size: ",num_procs," result: ",result
    sync all
end program reduce
```

**co-array initializations**

**coarray**

**barrier (synchronizes all elements)**

**Proc 1 does all the work**

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Object-oriented parallel programming paradigm (charm.cs.uiuc.edu)

- User (programmer) defines objects and interaction between objects
- Charm++ runtime system maps objects onto processors
- Typically, #objects > #processors
- Objects can migrate across processors dynamically
- Implemented as a C++ library, optimized for many systems

System implementation

User View
NAMD: Parallel molecular dynamics code based on Charm++

- Designed for simulation of large biomolecular systems
- Public distribution, more than 80,000 registered users
- Widest used application on several NSF supercomputers

Fig. source: Jim Phillips, UIUC

https://www.ks.uiuc.edu/Research/namd/
Charm++ Application 2: ChaNGa

ChaNGa: Charm N-Body GrAvity solver

- Collisionless N-body cosmological simulations, based on PKDGRAV code
- $O(n \log n)$ algorithm, implemented with Charm++

Collab.: Tom Quinn (Univ. Washington)
https://faculty.washington.edu/trq/hpcc/tools/changa.html

ChaNGa Scalability on Blue Waters

a) dataset: artificial uniform distribution
   12 and 14 billion particles

b) dataset: \textit{cosmo25}, \(z \approx 0\)
   very clustered distribution
   2 billion particles

\begin{figure}
\centering
\includegraphics[width=\textwidth]{chart.png}
\caption{Chart showing scalability of ChaNGa on Blue Waters}
\end{figure}

AMPI: MPI implemented with Charm++ virtualization

- Aimed at legacy MPI applications

Physical Processors

MPI ranks (“processes”)

Ranks=Objects implemented as user-level migratable Charm++ threads

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Application to Weather Forecasting

**BRAMS** Model:
Regional weather forecast
Written in Fortran+MPI
2D domain decomposition

`brams.cptec.inpe.br`

→ Processor load follows the major weather activity in BRAMS execution!
• **AMPI:**
  - Virtualization ≈ Overdecomposition
  - Migration: move ranks to other processors, to balance load (can be automated by the Charm++ runtime system)
Load-Balance Effect on BRAMS

BRAMS Execution with AMPI: P=64, #ranks=256

Original processor utilization

Proc. utilization with Load-Balance


Techniques vary according to the accelerator

- **GPUs:**
  - CUDA (proprietary)
  - OpenACC standard – based on directives
  - OpenMP standard (recent versions: 4.0, 4.5)

- **FPGAs:**
  - VHDL, Verilog – close to hardware
  - OpenCL – low support?

**Shared Problems:**
- How to ensure code portability with performance
- How to debug the running code
Language Challenges

Programming Languages

• Hundreds (thousands ?) of academic proposals of new languages for expressing parallelism

Important requirements for wide adoption:
• Relatively easy to learn
• Good performance of produced code
• Availability on various platforms
### Example: PRAC applications on Blue Waters system (USA) in 2014

<table>
<thead>
<tr>
<th>Discipline</th>
<th>Code</th>
<th>Language</th>
<th>Parallel/Paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice QCD</td>
<td>MILC, Chroma, CPS</td>
<td>C/C++</td>
<td>MPI/threads, GPU</td>
</tr>
<tr>
<td>Biomolecular dynamics</td>
<td>NAMD</td>
<td>C++</td>
<td>Charm++, GPU</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>ENZO/Cello</td>
<td>C++/Fortran</td>
<td>MPI+OpenMP</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>Pgadget/ENZO</td>
<td>C++/Fortran</td>
<td>MPI</td>
</tr>
<tr>
<td>Chemistry</td>
<td>ACESSIII</td>
<td>C, C++, Fortran</td>
<td>MPI</td>
</tr>
<tr>
<td>Social networks</td>
<td>Episimdemics</td>
<td>C++</td>
<td>Charm++</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>Cactus, LazEv, Harm3D, Whisky</td>
<td>C, C++, Fortran</td>
<td>Cactus, MPI, OpenMP</td>
</tr>
<tr>
<td>Fluid dynamics</td>
<td>PSDNS</td>
<td>Fortran</td>
<td>MPI</td>
</tr>
<tr>
<td>Climate/weather</td>
<td>GCRM, CCSM-IE, SP-CCSM (POP, CICE-4, CAM3.5)</td>
<td>C and F77/F90</td>
<td>MPI/OpenMP, MCT</td>
</tr>
<tr>
<td>Geophysics</td>
<td>AWP-ODC, Hercules</td>
<td>F77, F90</td>
<td>MPI, OpenMP, GPU</td>
</tr>
<tr>
<td>Material science</td>
<td>QMCPACK/ AFQMC</td>
<td>C++/Fortran</td>
<td>MPI/OpenMP, GPU</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>Cactus, CCATIE, McLachlan, Whisky, DiFranco</td>
<td>C, C++, Fortran</td>
<td>Cactus, MPI, OpenMP</td>
</tr>
<tr>
<td>Chemistry</td>
<td>GAMESS/ NWCHEM</td>
<td>C and Fortran</td>
<td>DDI/ GA/ARMCI, OpenACC</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>PPM</td>
<td>Fortran</td>
<td>MPI/OpenMP, GPU</td>
</tr>
<tr>
<td>Biology</td>
<td>ecology simulator</td>
<td>C++</td>
<td>MPI</td>
</tr>
<tr>
<td>Climate/weather</td>
<td>CM1</td>
<td>Fortran/F90</td>
<td>MPI/OpenMP</td>
</tr>
</tbody>
</table>
# Language Adoption (2)

<table>
<thead>
<tr>
<th>Turbulence in Fluids</th>
<th>DISTUF</th>
<th>F95</th>
<th>MPI PETSc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Physics</td>
<td>MS-FLUKSS with Chombo</td>
<td>Fortran/C++ python</td>
<td>MPI+OpenMP</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>ChaNGa</td>
<td>C++</td>
<td>Charm++</td>
</tr>
<tr>
<td>Space-based Earth Science</td>
<td>GRIPS SHARK Revisit</td>
<td>C/F90</td>
<td>MPI</td>
</tr>
<tr>
<td>Materials Science</td>
<td>OMEN NEMO5</td>
<td>C++</td>
<td>MPI PETSc</td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td>C/C++/F90</td>
<td>MPI+OpenMP, GPU</td>
</tr>
<tr>
<td>Biophysics</td>
<td>Gromacs</td>
<td>C</td>
<td>MPI+OpenMP</td>
</tr>
<tr>
<td>Climate/weather</td>
<td>CCSM</td>
<td>C/Fortran</td>
<td>MPI+OpenMP</td>
</tr>
<tr>
<td>Space Physics</td>
<td>Magtail OSIRIS</td>
<td>F90</td>
<td>MPI</td>
</tr>
<tr>
<td>Biophysics</td>
<td>AMBER</td>
<td>F90/C</td>
<td>MPI, GPU</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>Maestro Castro</td>
<td>F90/C++/F90</td>
<td>MPI+OpenMP</td>
</tr>
<tr>
<td>Biophysics</td>
<td>LAMMPS</td>
<td>C++/F90</td>
<td>MPI+OpenMP</td>
</tr>
<tr>
<td>Heliophysics</td>
<td>H3D/VPIC</td>
<td>F90/C</td>
<td>MPI+OpenMP</td>
</tr>
<tr>
<td>Heliophysics</td>
<td>BIFROST STAGGER RADMHD</td>
<td>C/C++</td>
<td>MPI, OpenMP, GPU</td>
</tr>
<tr>
<td>Materials Science</td>
<td>RMG, LAMMPS</td>
<td>C/C++</td>
<td>MPI, GPU</td>
</tr>
<tr>
<td>Astrophysics</td>
<td>PGADGET</td>
<td>C/C++</td>
<td>MPI+threads</td>
</tr>
<tr>
<td>Weather</td>
<td>WRF</td>
<td>F90/C</td>
<td>MPI+OpenMP</td>
</tr>
<tr>
<td>Earth Sciences</td>
<td>SPECFEM3D</td>
<td>F90</td>
<td>MPI</td>
</tr>
</tbody>
</table>
Language Adoption (3)

Summary:
Total: 34 samples (in 2014)
Fortran: 23 (68%)
C: 16 (47%)
C++: 19 (56%)
MPI: 30 (88%)
OpenMP: 16 (47%)
GPU: 9 (26%)

Trends after 2014:

• Some more ports to GPU
• Slow but steady Python adoption
• PGAS: only CoArray Fortran
Topics

1. Introduction
2. Moore’s Law, Processor Evolution
3. Top500 and Alternatives
4. Notable HPC Machines (Brazil and Abroad)
5. Main Programming Paradigms
6. Current Trends and Challenges
7. Conclusion
Current HPC Trends

a) World race towards 1 Exaflop!
   • Ongoing Exaflop programs: China, Japan, Europe, USA
   • Question-1: will it happen before 2020?
   • Question-2: what is a flop? Single or double precision?
   • Question-3: what is the acceptable power budget?

b) Growing adoption of “faster” memory
   • High-bandwidth memory (HBM)
   • Cache use vastly adopted (L1, L2, L3, ...)

Current Trends in High-Performance Computing
Celso L. Mendes (INPE)
Current HPC Trends (cont.)

(c) Wide use of hardware accelerators
   • GPUs tremendously popular, despite proprietary CUDA
   • Divergence of accelerator types (FPGA, etc)
   • Programming remains a bottleneck

d) Use of A.I. techniques ("fashion of the moment")
   • Goal: machines that can be trained and "learn"
   • Heavy use of neural networks
   • Desire of emulating human brain
     • If achieved, what will the power cost be?
Current Challenges

i. Fault Tolerance (Resiliency)

• Main technique currently: *checkpoint/restart*
  • Save periodically entire program state
  • In case of failure, return to latest checkpoint
  • Ideal checkpoint period: function of MTBF, number of processors, time to produce checkpoint, etc. (J. Daly, 2003)

With an increase in the number of processors, the probability of failure increases - *checkpoint/restart* may become prohibitive!
→ New resiliency techniques may be needed
ii. **End of Moore’s Law?**

- Current processor generation: ~14 nm
  - AMD and Intel promising to get down to 5~6 nm
  - Some more years available (a few?)

When Moore’s Law ends, what technology will come?

→ **Quantum computing**: concrete candidate, for some domains; a “cool” technology!
  - Many *big dogs* working on it: Google, Microsoft, IBM, etc.
  - Perhaps the quantum device could be an *accelerator* to conventional/traditional processors
Topics

1. Introduction
2. Moore’s Law, Processor Evolution
3. Top500 and Alternatives
4. Notable HPC Machines (Brazil and Abroad)
5. Main Programming Paradigms
6. Current Trends and Challenges
7. Conclusion
• **HPC Motivation**
  • Economic and scientific incentives
  • Propelled by Moore’s Law, Multi-core chips

• **Current HPC Systems**
  • Post-petaflop era, race to exaflop
  • Many concrete scientific discoveries enabled
  • Programming (for performance) remains a hard task
    • MPI, OpenMP still popular

• **The Future**
  • What’s next after Moore’s Law? Quantum computing?
  • “It's tough to make predictions, especially about the future!”
    (Yogi Berra)
To see more details: disciplines of High-Performance Computing in São José dos Campos at INPE, Unifesp, ITA

**INPE:** Drs. Celso Mendes, Stephan Stephany

**Unifesp:** Prof. Álvaro Fazenda

**ITA:** Prof. Jairo Panetta
Thank You!

Questions?