Current Trends in High-Performance Computing

Celso L. Mendes

INPE

São José dos Campos – SP , Brazil

Email: celso.mendes@inpe.br

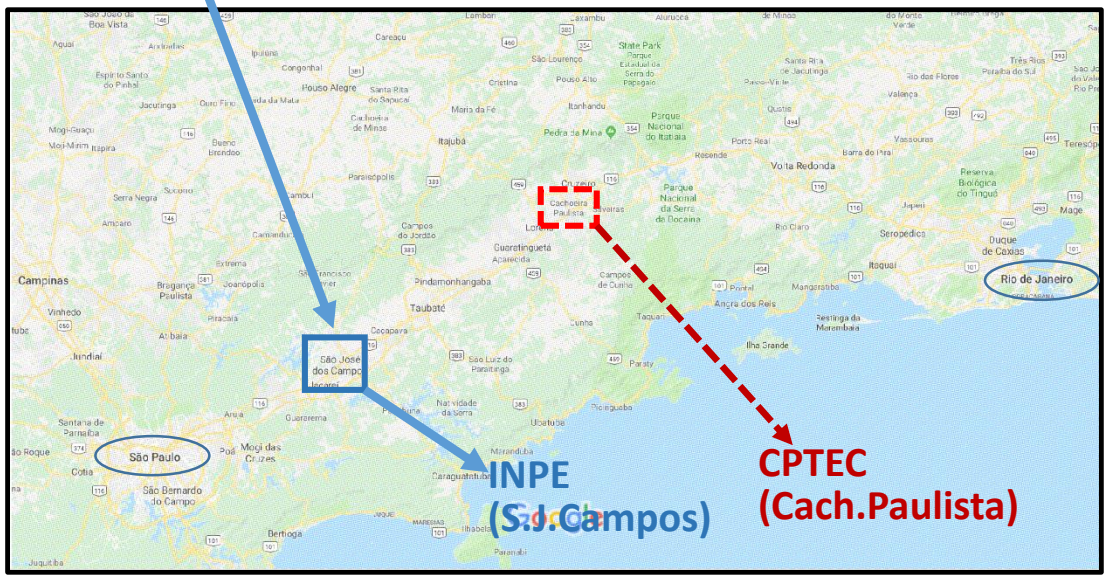
October 11, 2018

About Me

celso.mendes
@
inpe.br

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.)



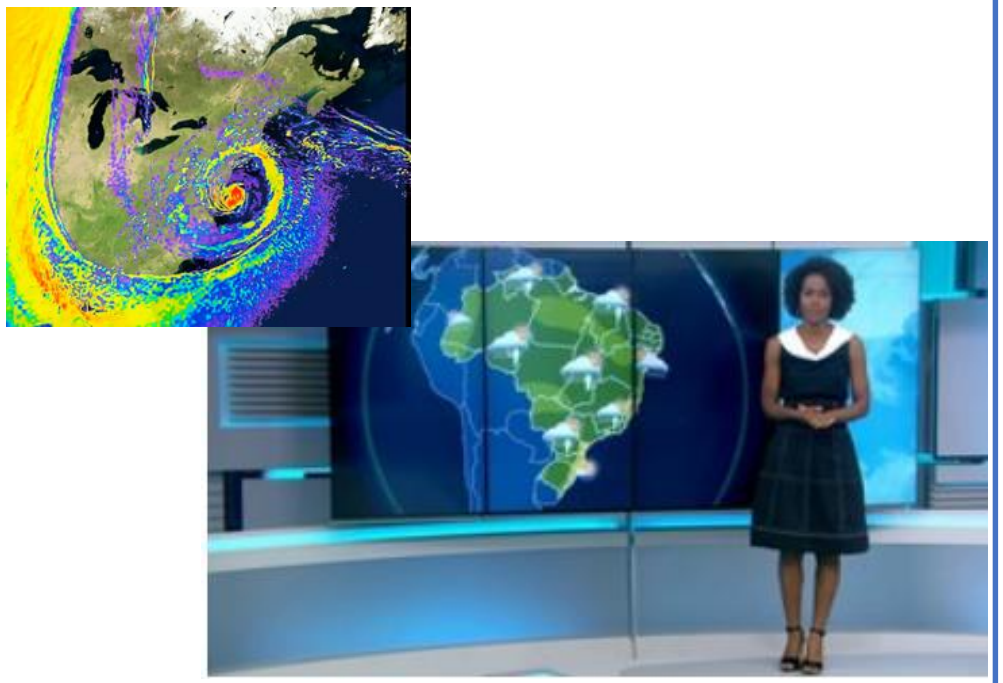
Topics

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

Motivations for HPC

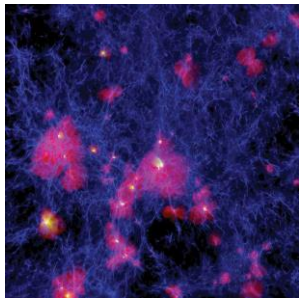
Economical Motivation:

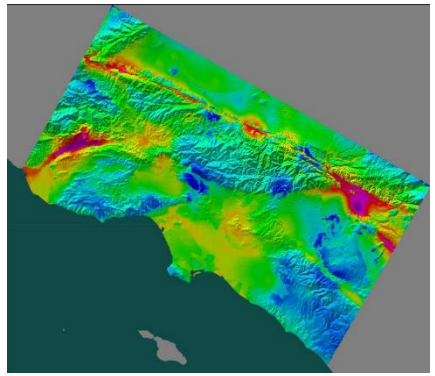
- Impact on everyone's life



Sientific Motivation:

- Acceleration of discoveries





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

Evolution & Moore's Law

Gordon Moore (Fairchild,Intel), 1965:

- Prediction about future increase in density of chips

The experts look ahead

Cramming more components onto integrated circuits

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

By Gordon E. Moore

Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

The future of integrated electronics is the future of electronics itself. The advantages of integration will bring about a proliferation of electronics, pushing this science into many new areas.

Integrated circuits will lead to such wonders as home computers—or at least terminals connected to a central computer—automatic controls for automobiles, and personal portable communications equipment. The electronic wristwatch needs only a display to be feasible today.

But the biggest potential lies in the production of large systems. In telephone communications, integrated circuits in digital filters will separate channels on multiplex equipment. Integrated circuits will also switch telephone circuits and perform data processing.

Computers will be more powerful, and will be organized in completely different ways. For example, memories built of integrated electronics may be distributed throughout the machine instead of being concentrated in a central unit. In addition, the improved reliability made possible by integrated circuits will allow the construction of larger processing units. Machines similar to those in existence today will be built at lower costs and with faster turn-around.

Present and future

By integrated electronics, I mean all the various technologies which are referred to as microelectronics today as well as any additional ones that result in electronics functions supplied to the user as irreducible units. Those technologies were first investigated in the late 1950's. The object was to miniaturize electronics equipment to include increasingly complex electronic functions in limited space with minimum weight. Several approaches evolved, including microassembly techniques for individual components, thin-film structures and semiconductor integrated circuits.

Each approach evolved rapidly and converged so that each borrowed techniques from another. Many researchers believe the way of the future to be a combination of the various approaches.

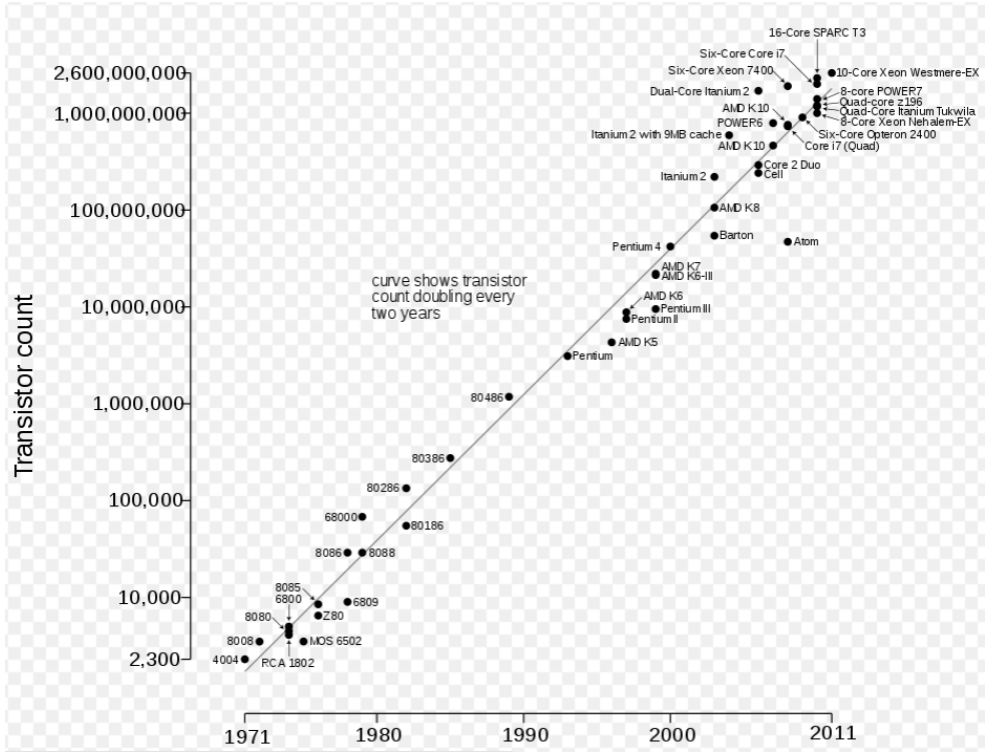
The advocates of semiconductor integrated circuitry are already using the improved characteristics of thin-film structures by applying such films directly to an active semiconductor substrate. Those advocating a technology based upon films are developing sophisticated techniques for the attachment of active semiconductor devices to the passive film arrays.

Both approaches have worked well and are being used in equipment today.

The author

Dr. Gordon E. Moore is one of the new breed of electronic engineers, schooled in the physical sciences rather than in electronics. He earned a B.S. degree in chemistry from the University of California and a Ph.D. degree in physical chemistry from the California Institute of Technology. He was one of the founders of Fairchild Semiconductor and has been director of the research and development laboratories since 1959.

Electronics, Volume 38, Number 8, April 19, 1965



Rate of increase constant across time!

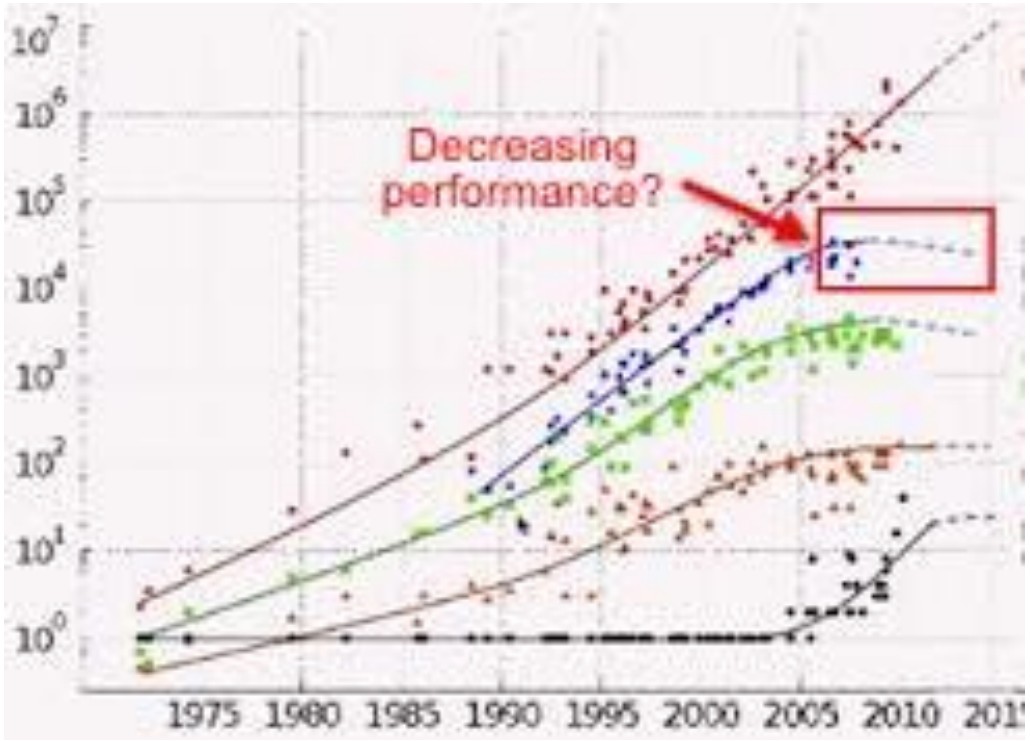
Doubles @ 2 years

Source: Wikipedia

Moore's Law (cont.)

Moore's Law in practice:

- Beginning of *Multi-Core* era in 2005



Number of Transistors (Moore's Law)

Performance per Core

Frequency

Power Consumption

Number of Cores

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

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

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

Abbrev.	Unity	Flops/s
MF	Megaflops	10^6
GF	Gigaflops	10^9
TF	Teraflops	10^{12}
PF	Petaflops	10^{15}
EF	Exaflops	10^{18}

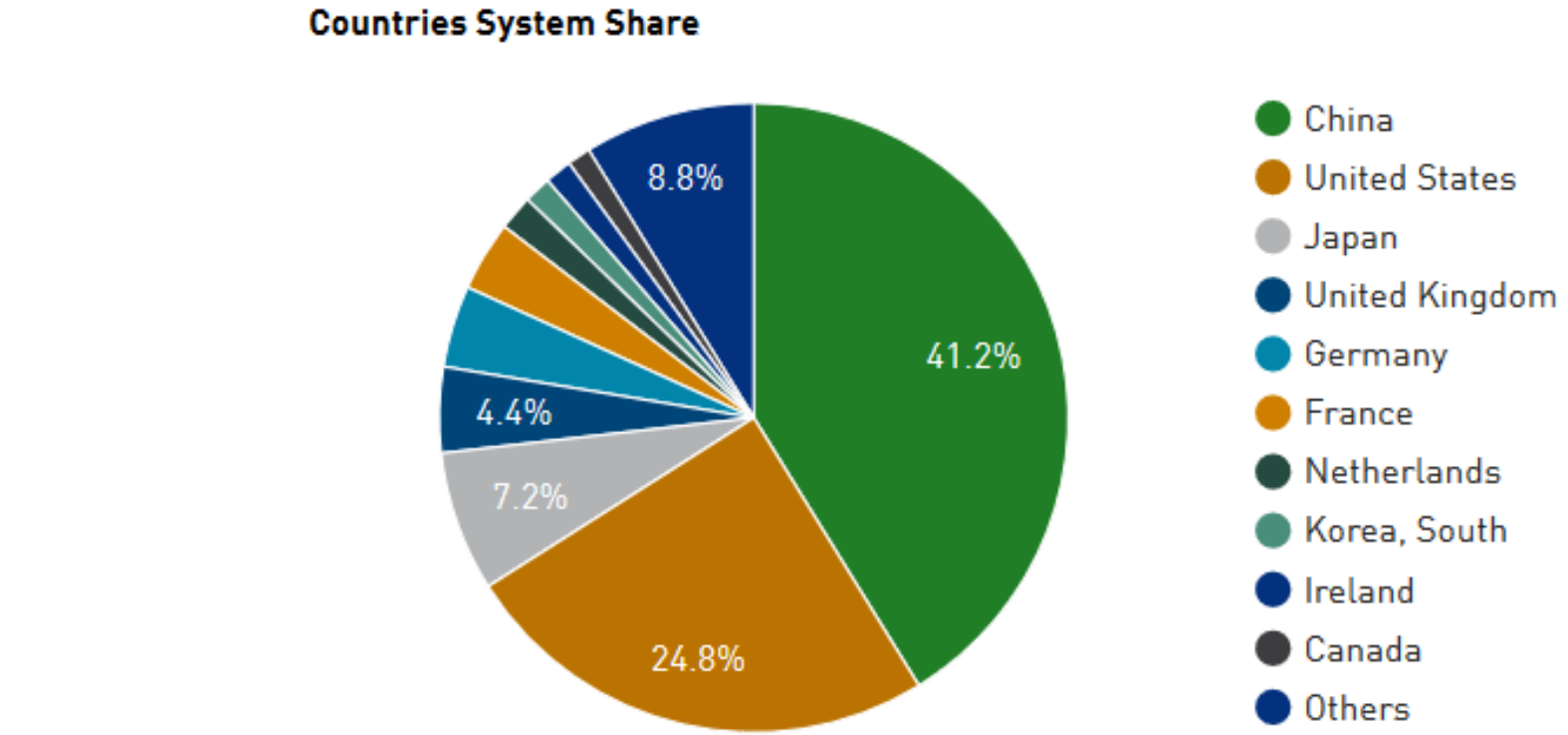
Current Top500 Edition

- 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

Position	System	Country	R_{max} (PFlops)	R_{peak} (PFlops)
1	Summit (IBM)	USA	122.30	187.66
2	Sunway TaihuLight (NRCPC)	China	93.01	125.44
3	Sierra (IBM)	USA	71.61	119.19
4	Tianhe-2A (NUDT)	China	61.44	100.68
5	ABCI (Fujitsu)	Japan	19.88	32.58

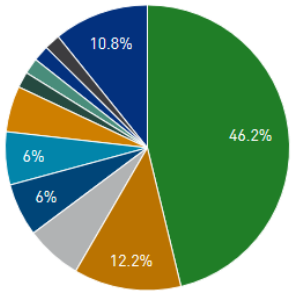
Top500 List - Geography

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

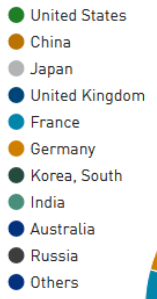


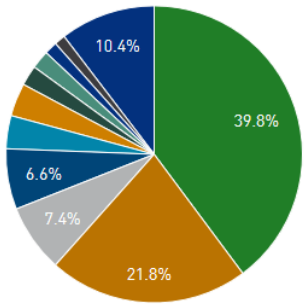
Top500 List – Geography / Past

- Participation by country: evolution (number of systems)

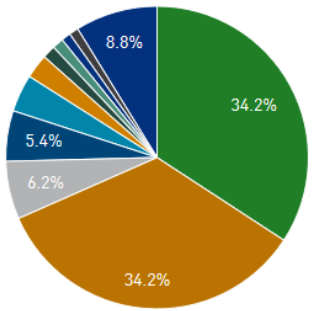


Nov. 2014

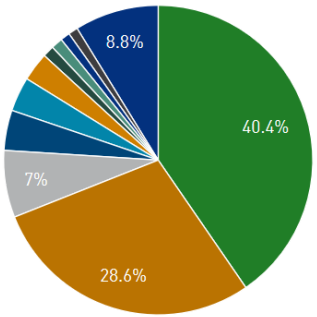




Nov. 2015



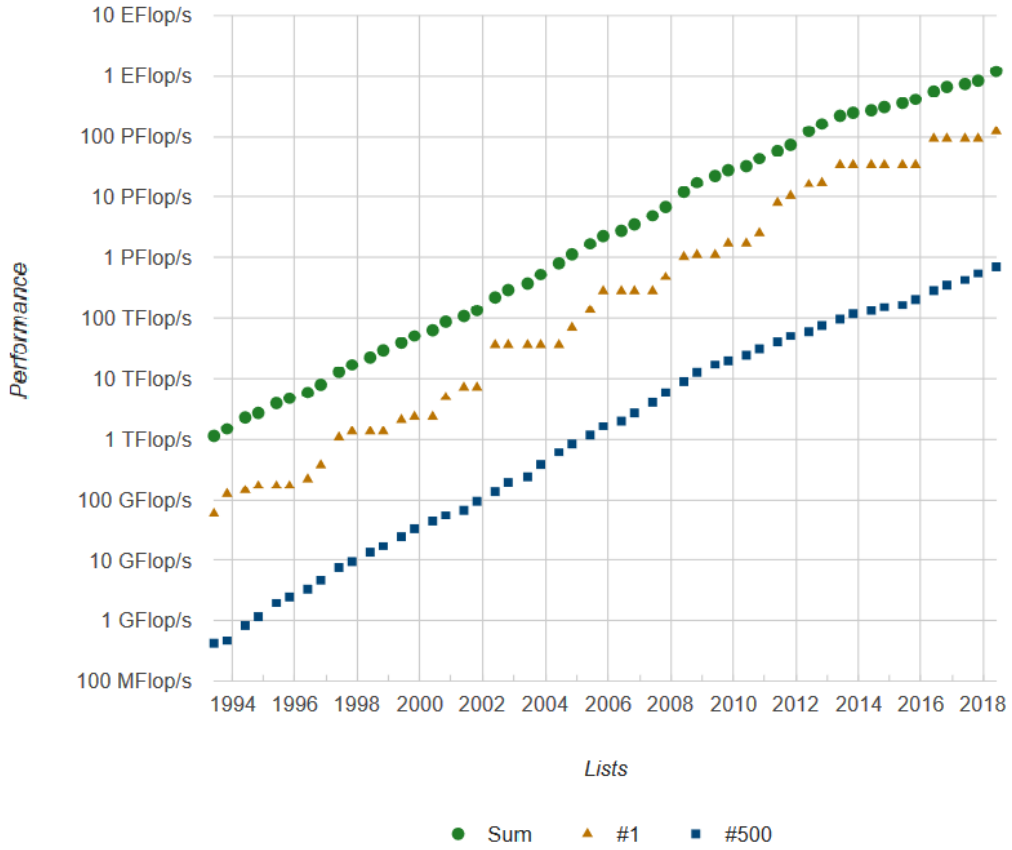
Nov. 2016



Nov. 2017

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)
  - 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)
  - 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

Alternatives to Top500 (cont.)

- **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

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

Top500 #1,#3: Summit, Sierra

- CORAL Program – U.S. Dep.Energy, 2015 https://asc.llnl.gov/coral-info
  - 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
- CORAL-2 Program – 2018 https://procurement.ornl.gov/rfp/CORAL2/
  - Goal: Systems with performance beyond 1 Exaflop in 2021~2023
  - Budget of US$ 1.8 B
  - Selections expected in Oct~Nov/2018

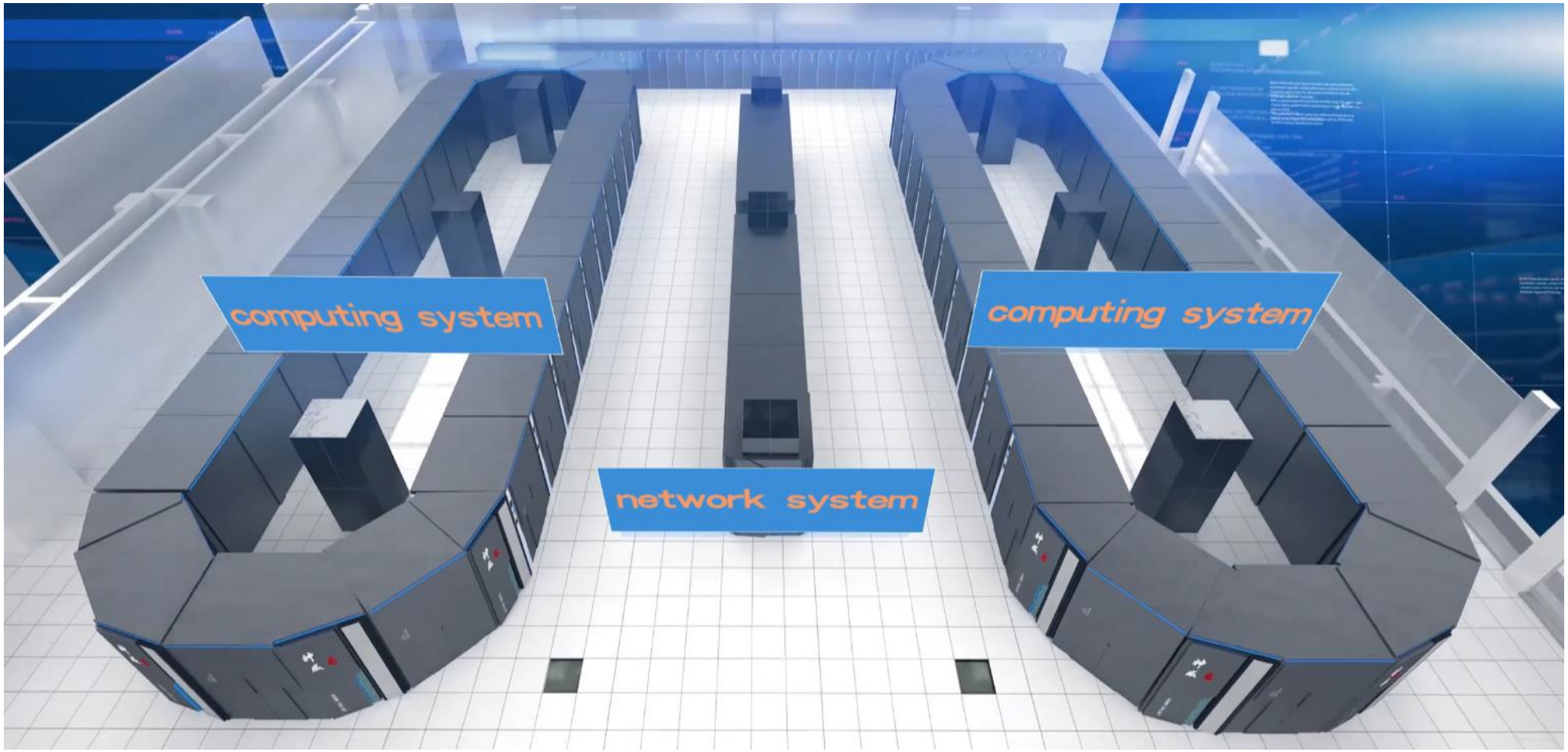
Top500 #1: Summit

- Location: Oak Ridge National Lab.
  - Vendors: IBM + Nvidia
  - 4,608 compute nodes
    - 2 Power9 processors @ 3.1 GHz per node
      - 2 × 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 × 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: ~12 MW
  - Details: https://hpc.llnl.gov/hardware/platforms/sierra

Top500 #2: Sunway TaihuLight



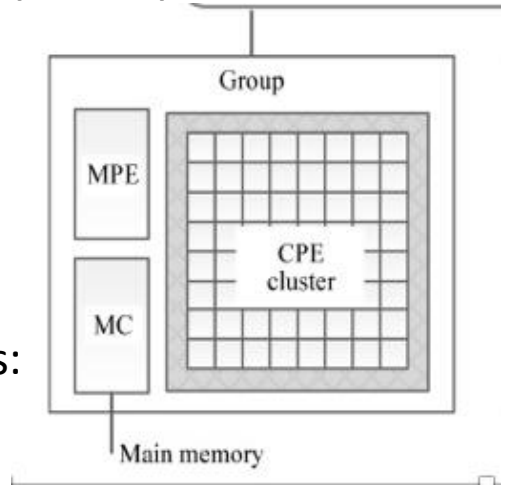
Top500 #2: Sunway TaihuLight

- Location: Wuxi, China
  - Installled at the supercomputing center of Wuxi, 2016
  - URL: http://www.netlib.org/utk/people/JackDongarra/PAPERS/sunway-report-2016.pdf
  - 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

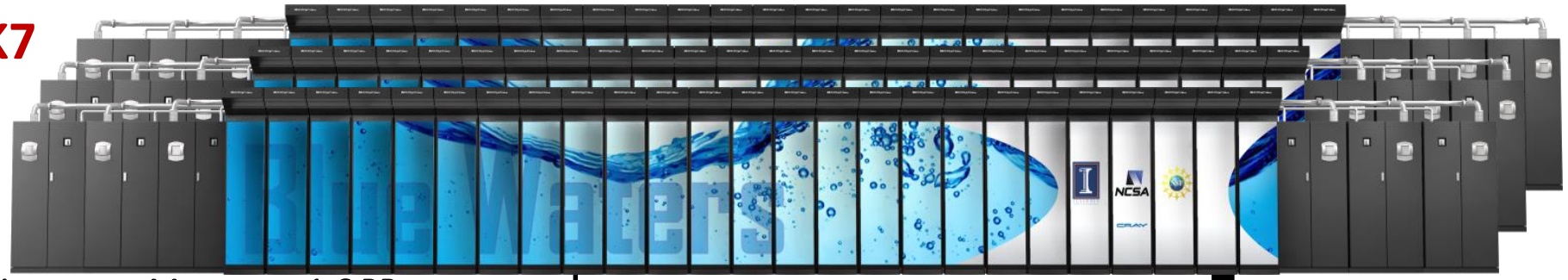
- Processor: SW26010 (Shanghai, China)
  - Architecture: many-core, RISC
  - Peak performance of 11.6 Gflops/s for each core
  - 4 groups of cores
    - each group: 64 cores (CPE), plus a control unit (MPE)
  - 260 cores per chip: > 3 TFlops/s per chip
  - 32 GB external memory per node, 1.2 PB total

Grupo of cores:



Blue Waters System – Univ.Illinois

Cray XE6/XK7



Aggregate Memory – 1.6 PB

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

10/40/100 Gb Ethernet Switch

External Servers

IB Switch

>1 TB/sec

120+ Gb/sec

100 GB/sec

Spectra Logic: 300 usable PB

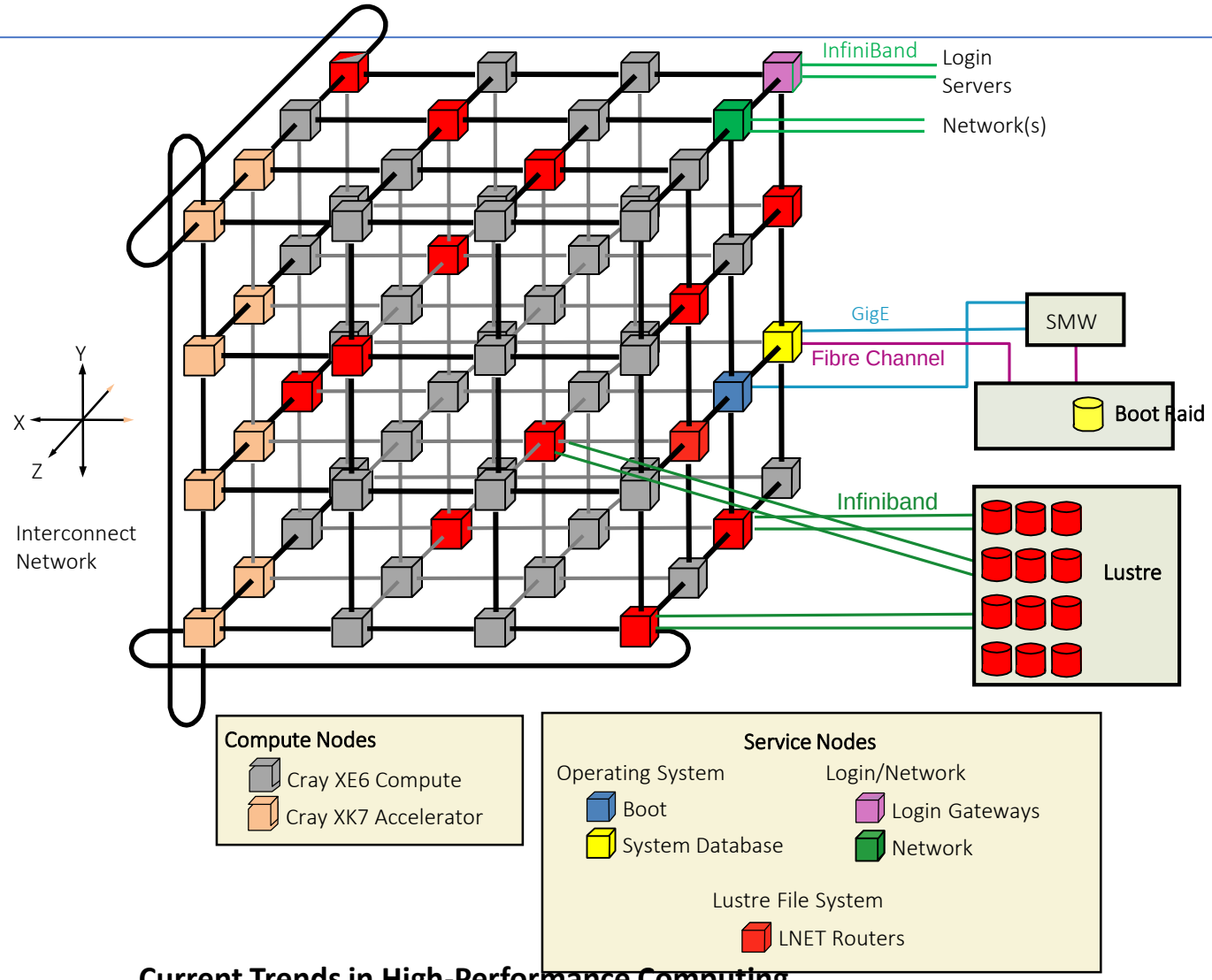
Sonexion: 26 usable PB

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



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

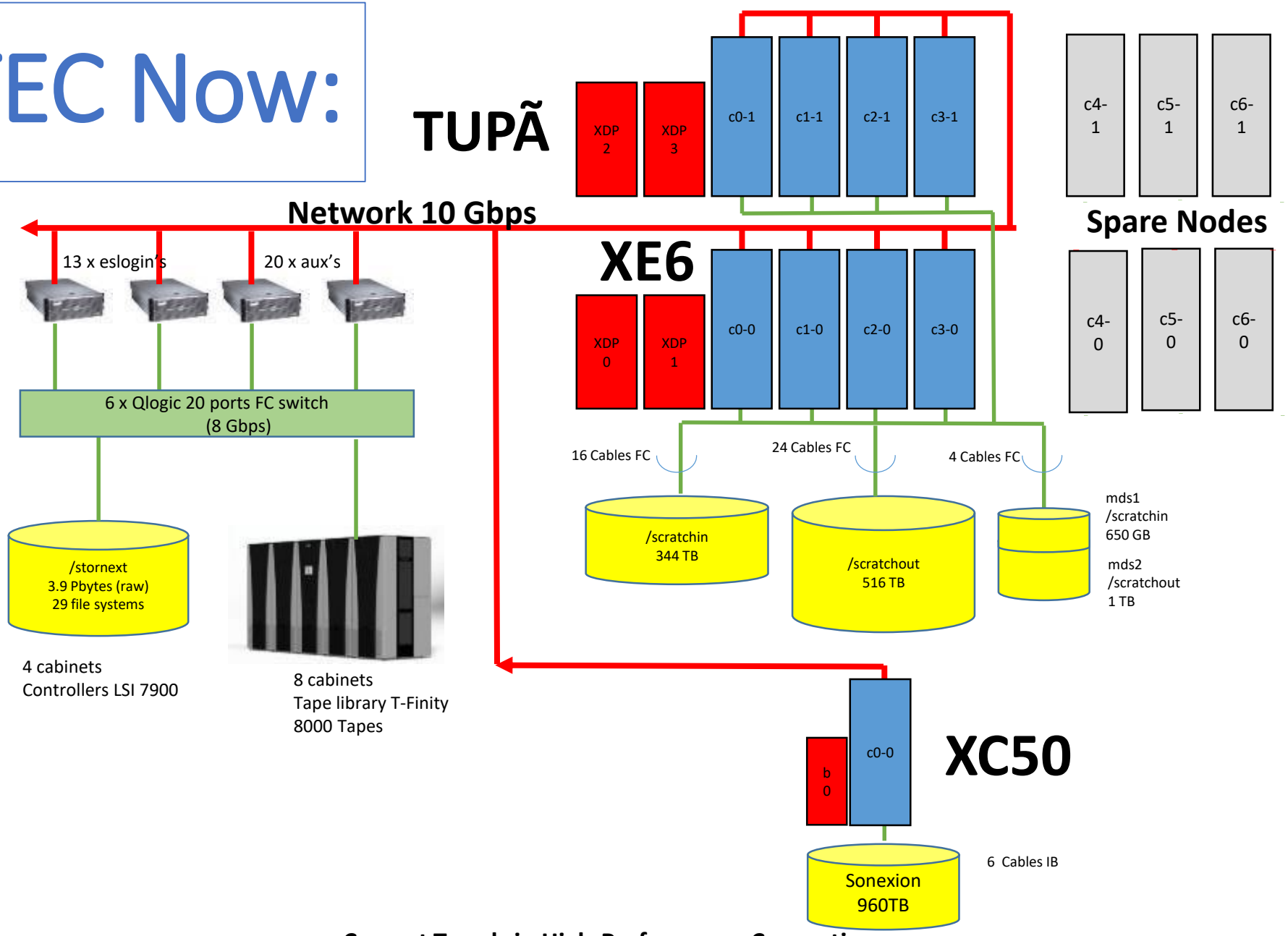
HPC Systems at INPE

- 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

Tupã System – Upgrade

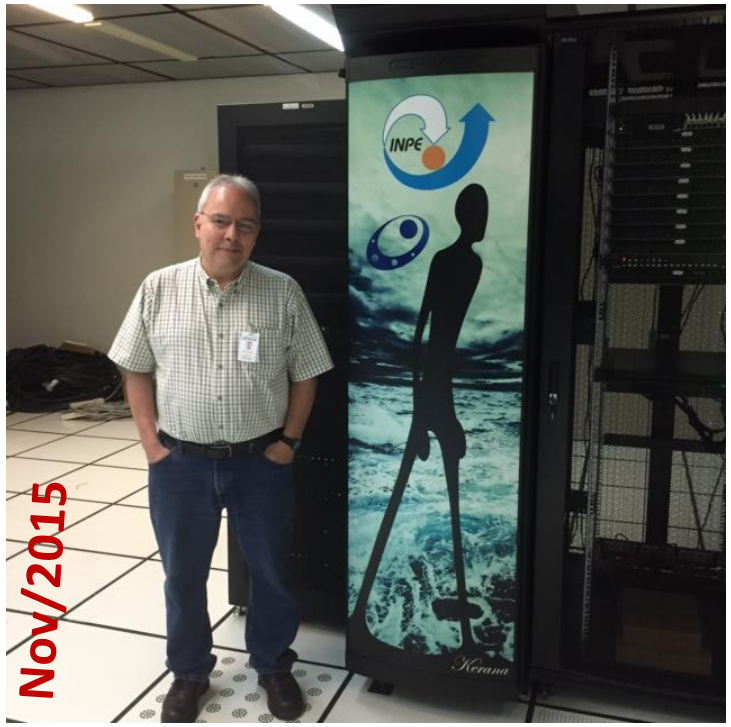
- New Partition: XC50 (2018)
  - 102 nodes, 2 Intel Skylakes
    - Cores: 20/chip, 40/node
    - Total: 4,080 Intel cores
    - Peak perform.: 313 Tflops/s
    - New network: Aries
  - Original Tupã reconfigured:
    - 8 cabinets still working
    - 6 cabinets retired (spare)
    - Total: ~17,800 AMD cores
    - Peak performance: ~~258~~ → 147 Tflops/s

CPTEC Now:

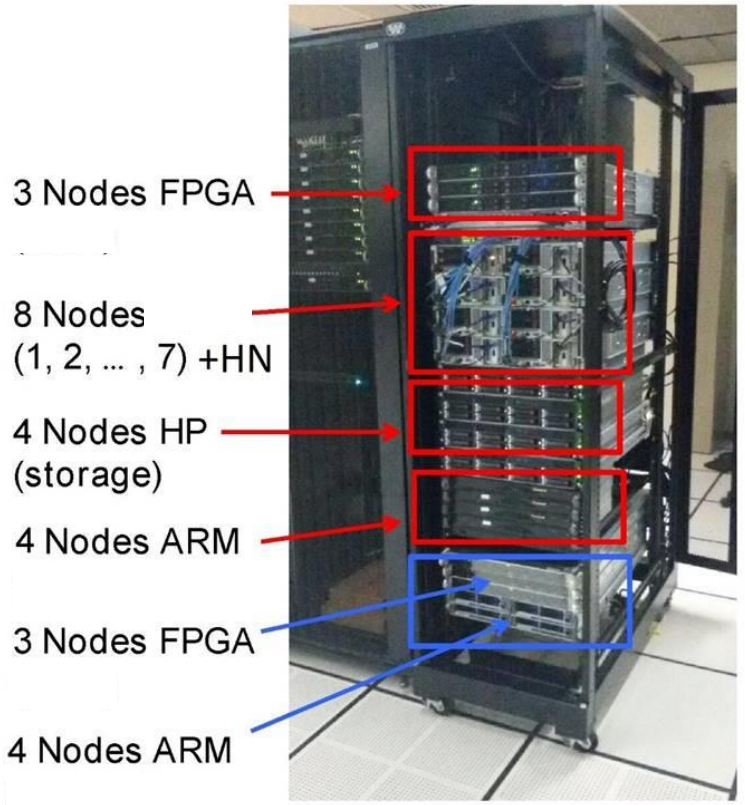

TUPÃ
XDP 2
XDP 3
c0-1
c1-1
c2-1
c3-1
c4-1
c5-1
c6-1
Spare Nodes
Network 10 Gbps
XE6
13 x eslogin's
20 x aux's
XDP 0
XDP 1
c0-0
c1-0
c2-0
c3-0
c4-0
c5-0
c6-0
6 x Qlogic 20 ports FC switch (8 Gbps)
16 Cables FC
24 Cables FC
4 Cables FC
/scratchin 344 TB
/scratchout 516 TB
mds1 /scratchin 650 GB
mds2 /scratchout 1 TB
/stornext 3.9 Pbytes (raw) 29 file systems
4 cabinets Controllers LSI 7900
8 cabinets Tape library T-Finity 8000 Tapes
b 0
c0-0
XC50
Sonexion 960TB
6 Cables IB

Other HPC Systems at INPE

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



- LAChibrido cluster (research)
  P.I.: Prof. Haroldo F. C. Velho, LAC/INPE



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

Programming for Parallelism

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

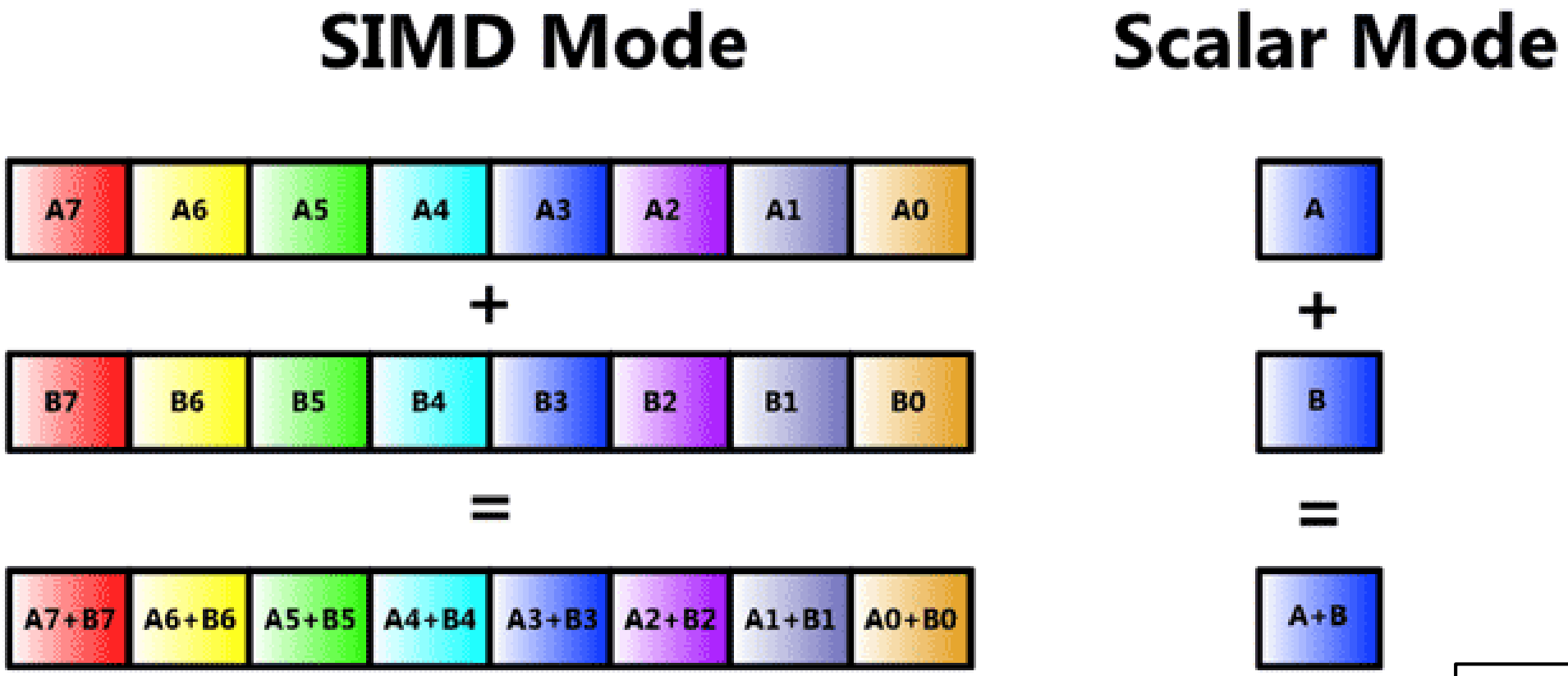
- Between 1990~2000: x86 Processors
  - Vectorization enabled by SSE extensions
    - SSE: *Streaming SIMD Extension*
    - Support in hardware, **vector registers** with L=4 or more
  - Similar concept to old vector machines:

    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
    - AVX (2011), AVX-2 (2015)

Vector Operations / SIMD

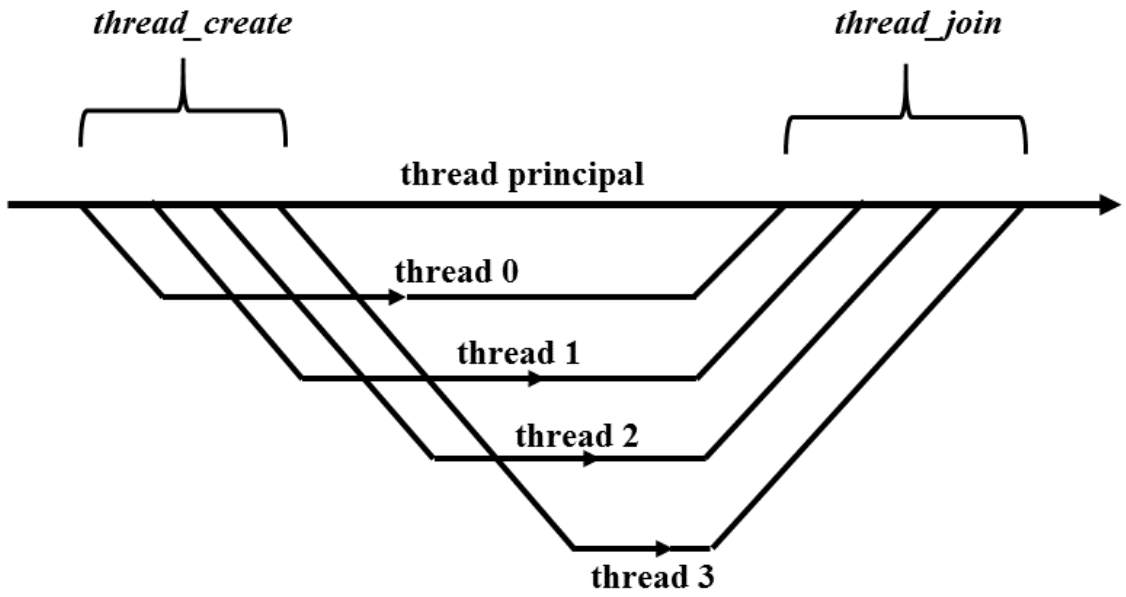


source: Intel

Shared-Memory Parallelism

Main Technique: use of *threads*

- Execution Model: *fork/join*



Shared-Memory Parallelism

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

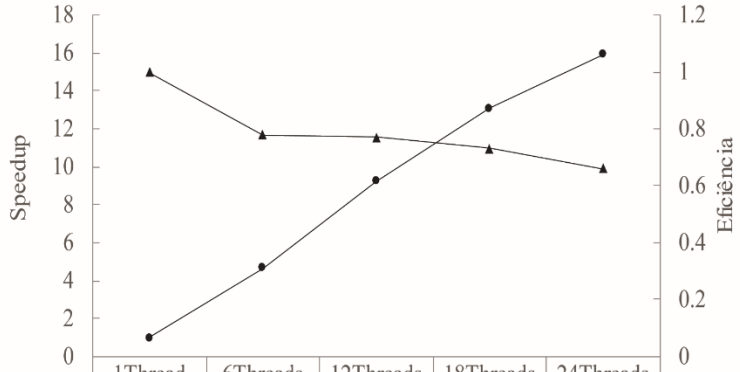
b) Managed by the compiler: OpenMP standard

- Programmer inserts directives in serial code
- Compiler processes the directives, creates threads
- Allows “gradual” parallelization of existing codes

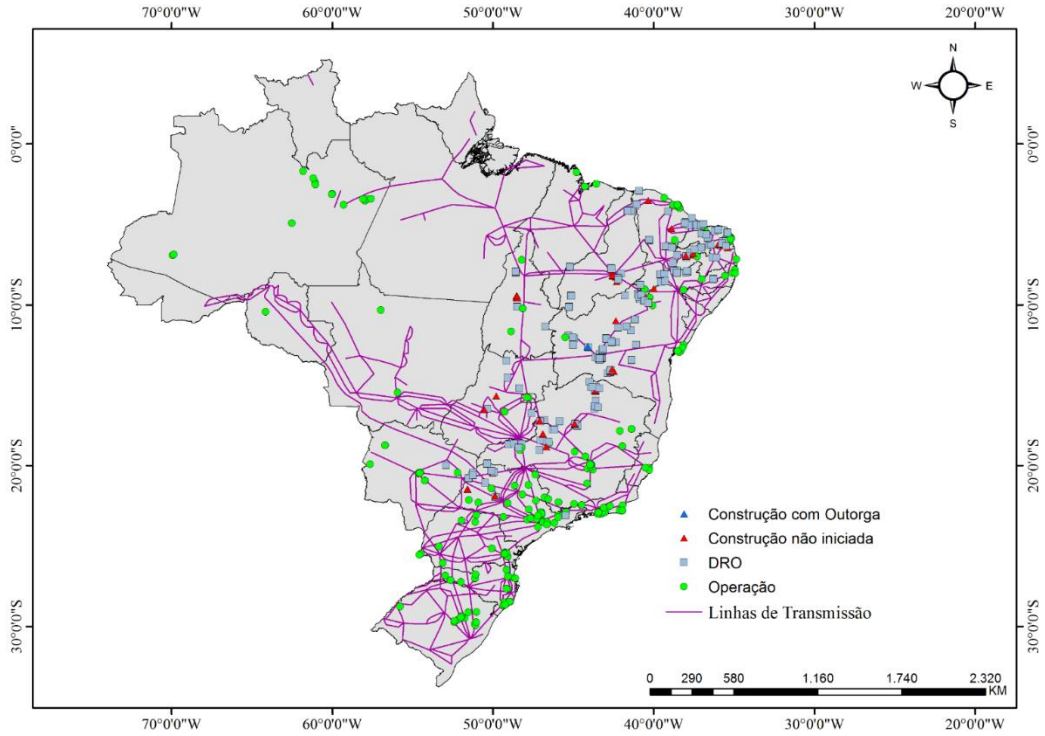
OpenMP Parallelization @ INPE (1)

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:



	1Thread	6Threads	12Threads	18Threads	24Threads
Speedup	1	4.66	9.25	13.08	15.9
Eficiência	1	0.78	0.77	0.73	0.66
Tempo (s)	90700	19469	9808	6935	5703

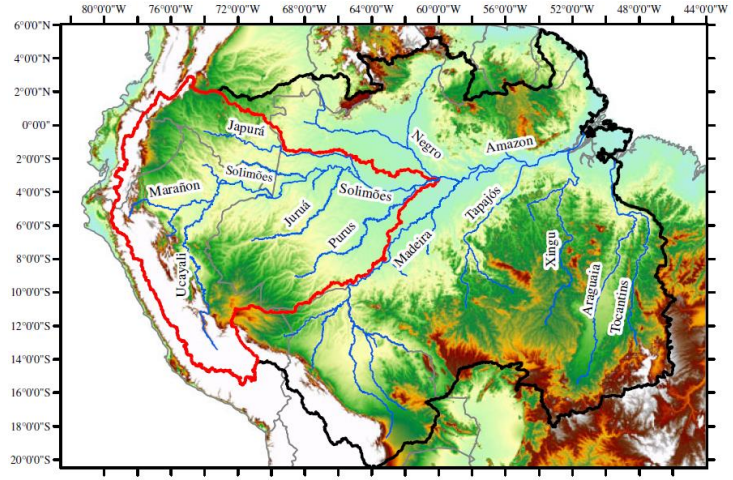


Ref: Jefferson G. Souza, **C.L.Mendes** & R.S.Costa, "*Otimização Computacional do Modelo BRASIL-SR*", Anais do WSCAD'2018, S.Paulo, Out. 2018.

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



Ref: Henrique R.A. Freitas & **C.L.Mendes**, "*Parallelization of a Large-Scale Watersheds Hydrological Model using CPU and GPU*", Anais do ERAD-SP'2018, S.J.Campos, Abr. 2018.

Distributed-Memory Parallelism

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

Distributed-Memory Parallelism (2)

Emerging Technique: PGAS Languages

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

Goals:

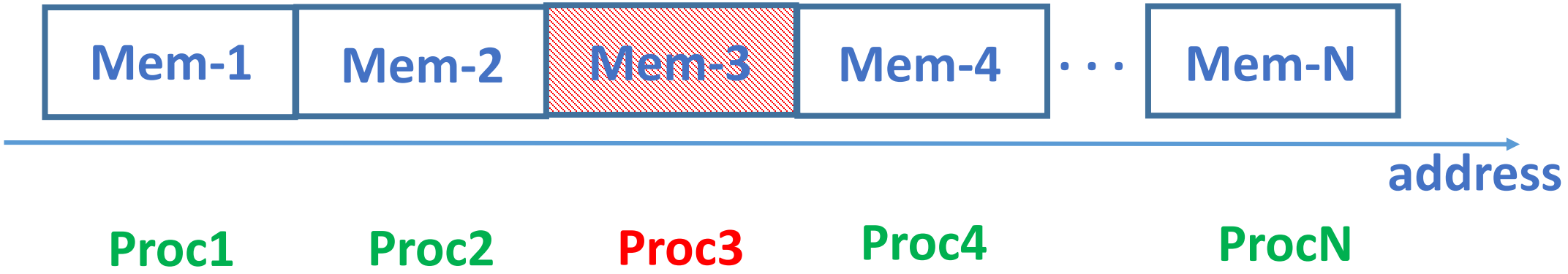
- Raise the level of abstraction for programming distributed-memory systems
- Avoid the need to deal with message passing in the source program

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

```
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
```

Fortran2008 Program

co-array

initializations

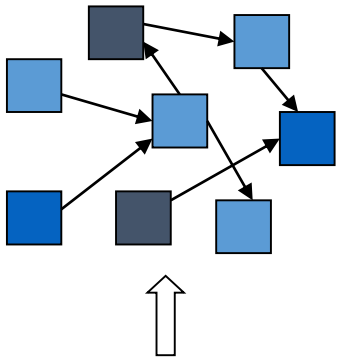
barrier (sincronizes all elements)

Proc 1 does all the work

Charm++ (Univ. Illinois)

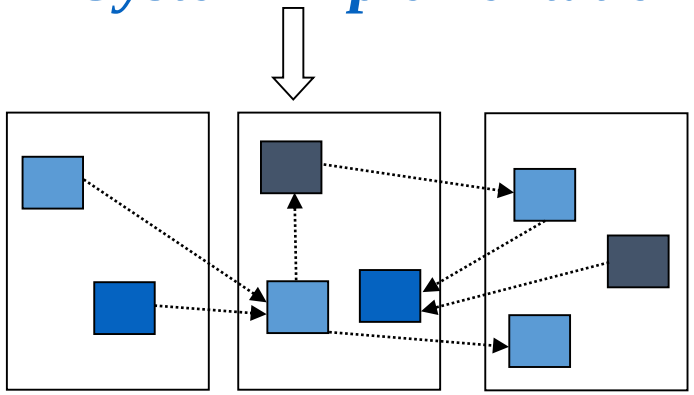
Object-oriented parallel programming paradigm (**charm.cs.uiuc.edu**)

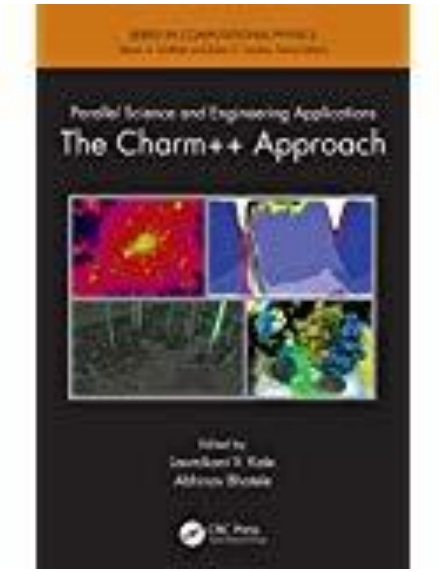
- User (programmer) defines objects and interaction between objects
- Charm++ runtime system maps objects onto processors
- Typically, #objects > #processors
- Objects can migrate across processors dynamically
- Implemented as a C++ library, optimized for many systems



User View

System implementation

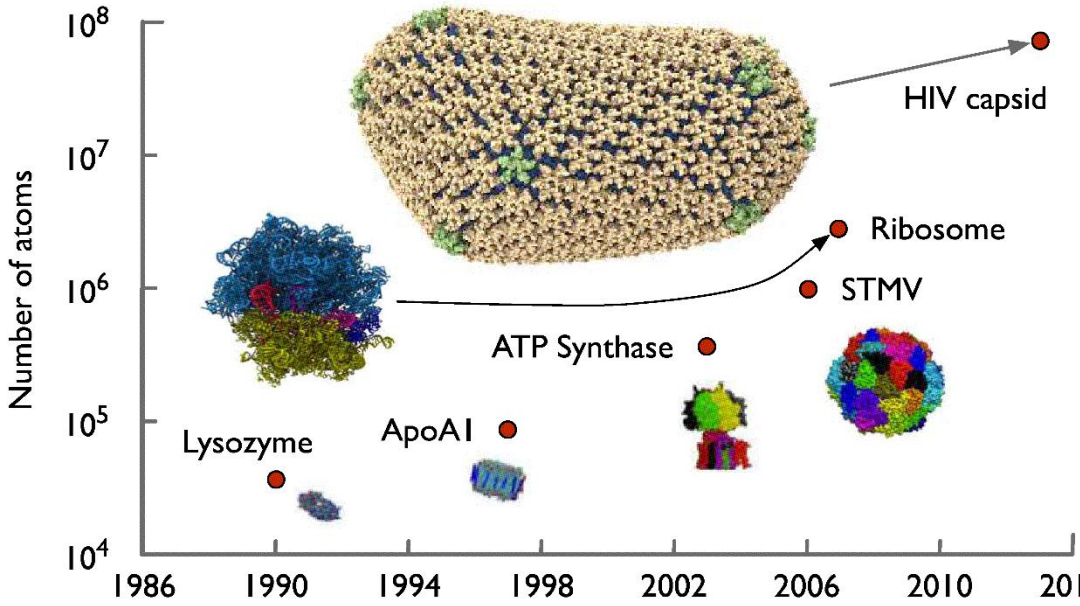




Charm++ Application 1: NAMD

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



https://www.ks.uiuc.edu/Research/namd/

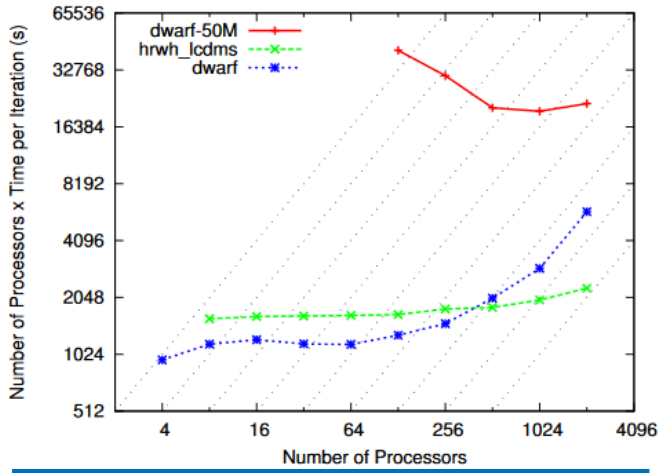
Fig.source: Jim Phillips, UIUC

Charm++ Application 2: ChaNGa

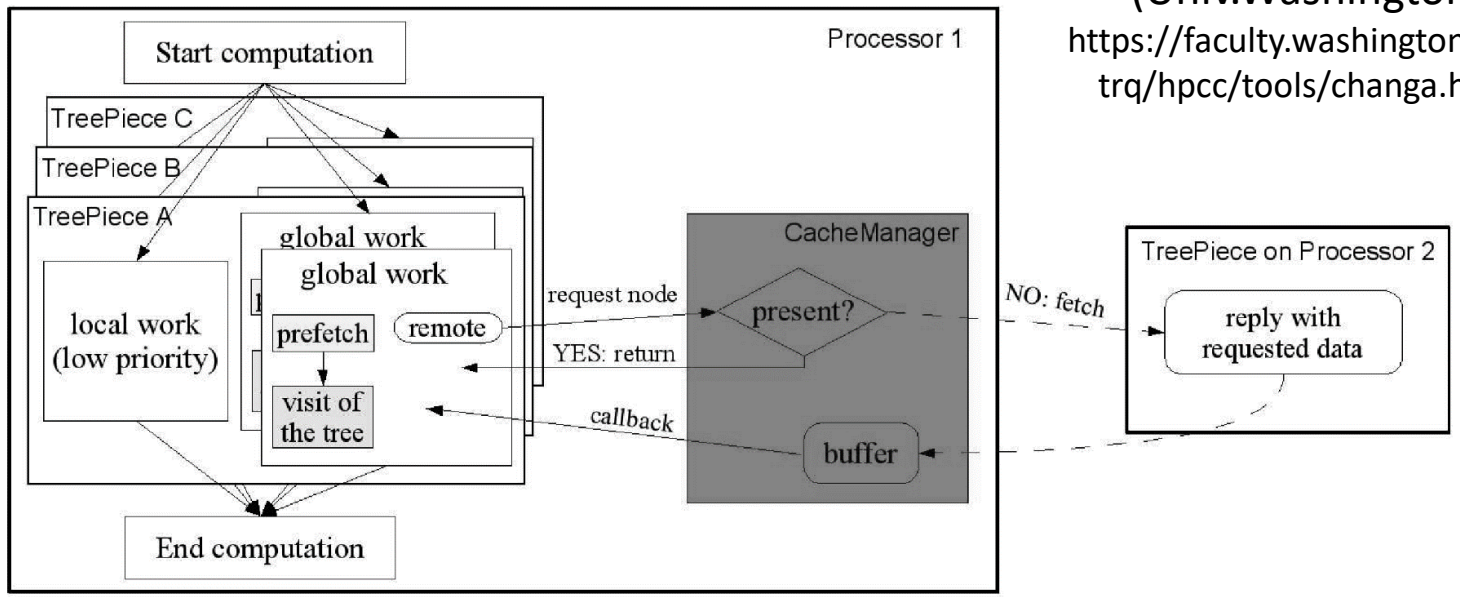
ChaNGa: Charm N-Body GrAvity solver

- 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

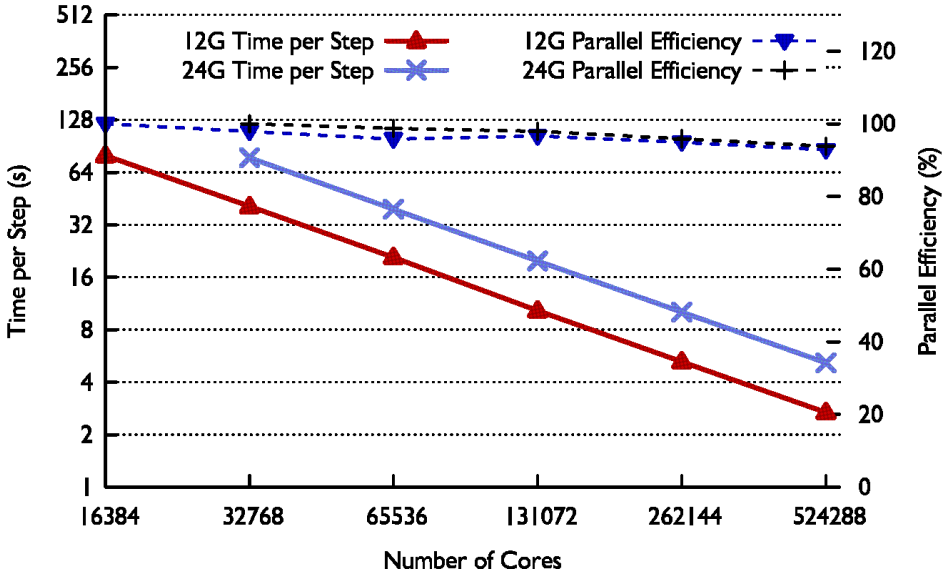


Ref.: P.Jetley, F.Gioachin, **C.Mendes**, L.V.Kalé & T.Quinn, "*Massively Parallel Cosmological Simulations with ChaNGa*", Proceedings of IPDPS, 2008.

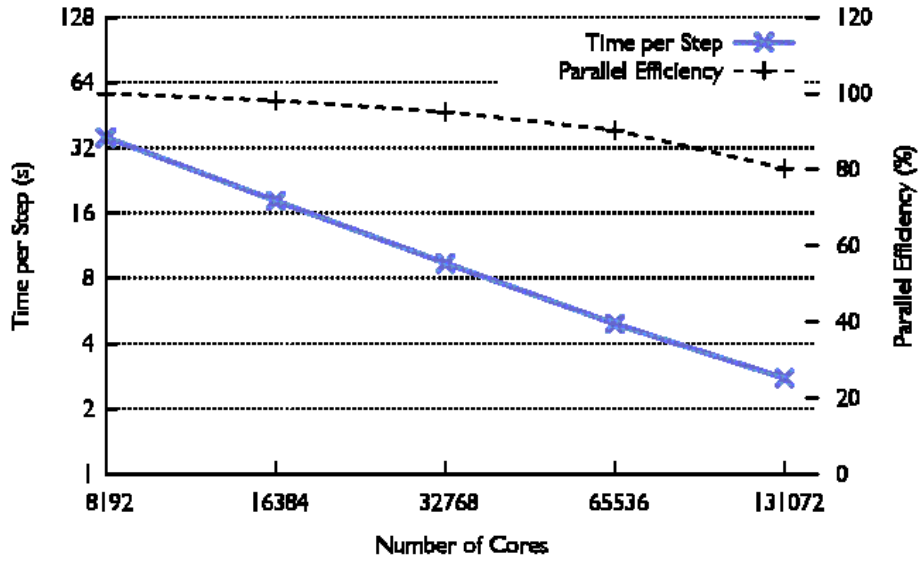


ChaNGa Scalability on Blue Waters

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



b) dataset: *cosmo25*, z≈0
very clustered distribution
2 billion particles

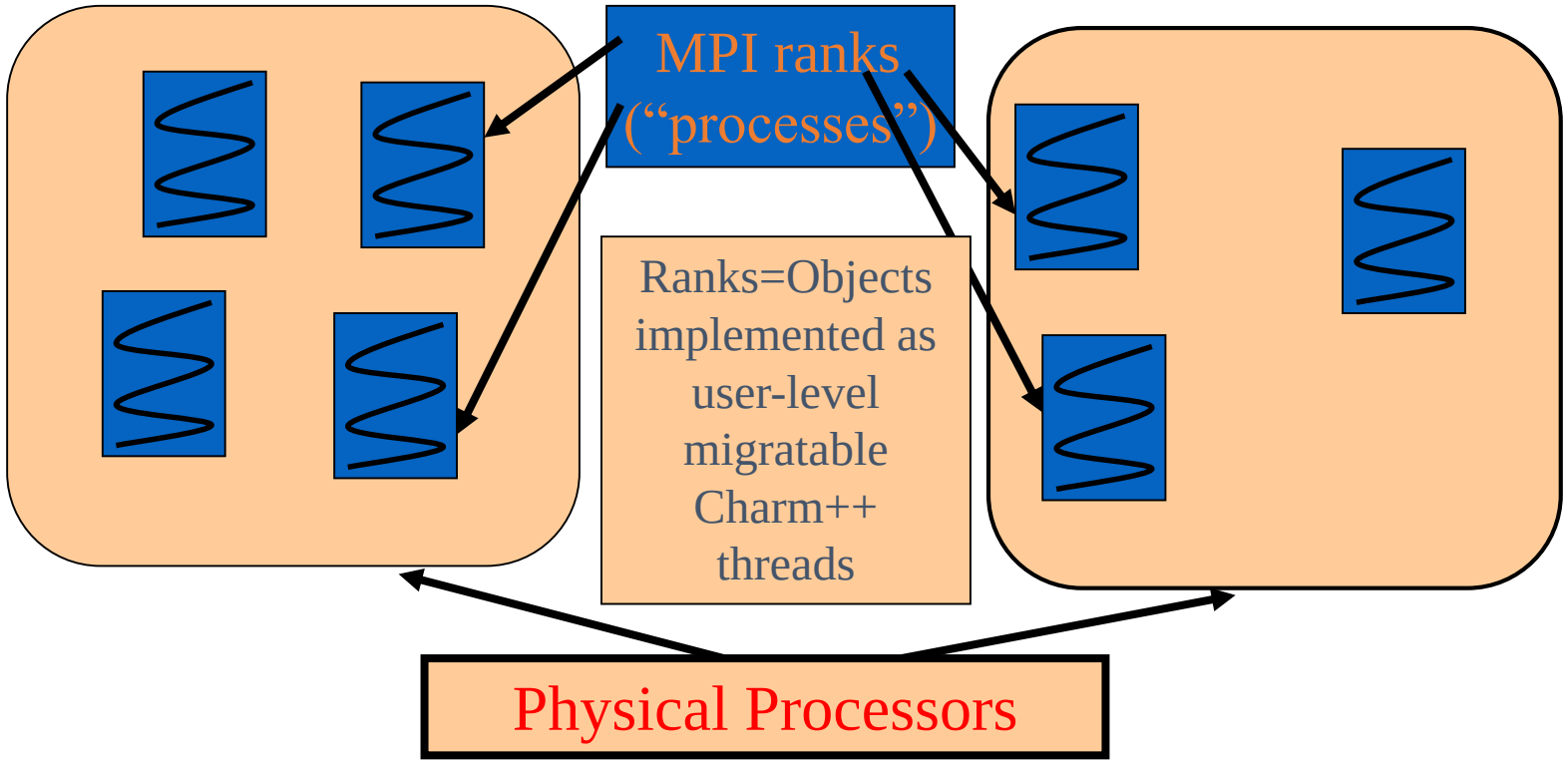


Source: H.Menon et al, "*Adaptive techniques for clustered N-body cosmological simulations*", Computational Astrophysics and Cosmology, v.2(1), pp.1-16, 2015.

AMPI: Adaptive MPI

AMPI: MPI implemented with Charm++ virtualization

- Aimed at legacy MPI applications



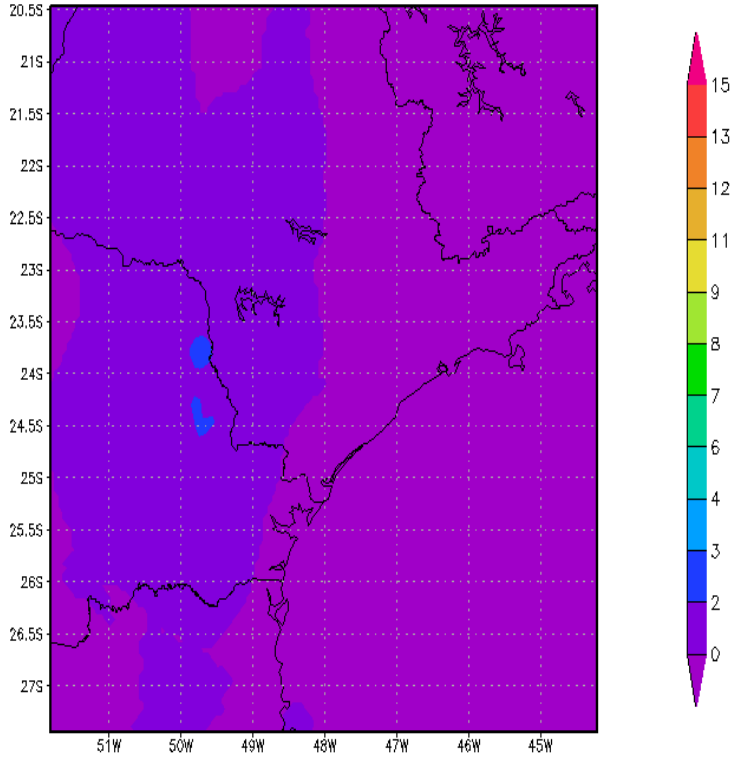
Application to Weather Forecasting

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

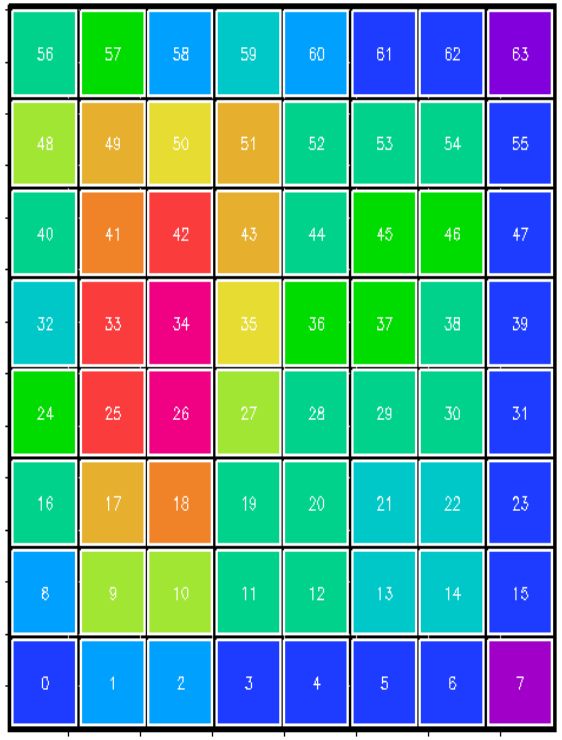
brams.cptec.inpe.br

→ Processor load follows the major weather activity in BRAMS execution!

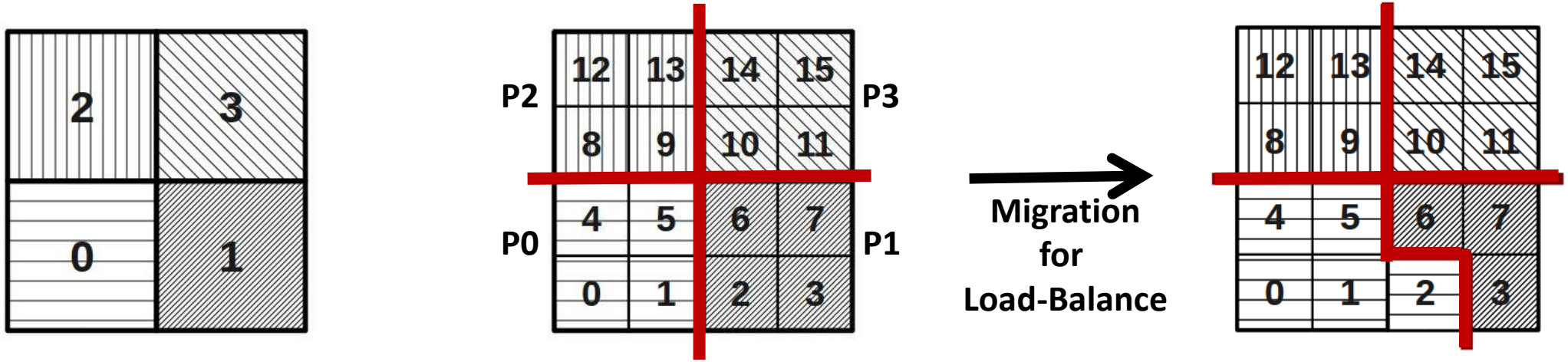
Weather Forecast, Feb.2010



Processor Load, P=64 (8x8)



AMPI: Virtualization and Migration



MPI
4 ranks

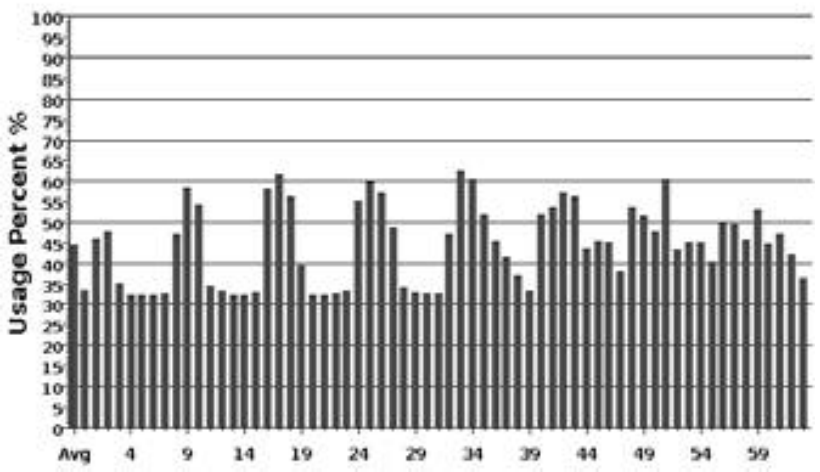
AMPI
(Virtualization)
16 ranks

AMPI
(Post-Migration)

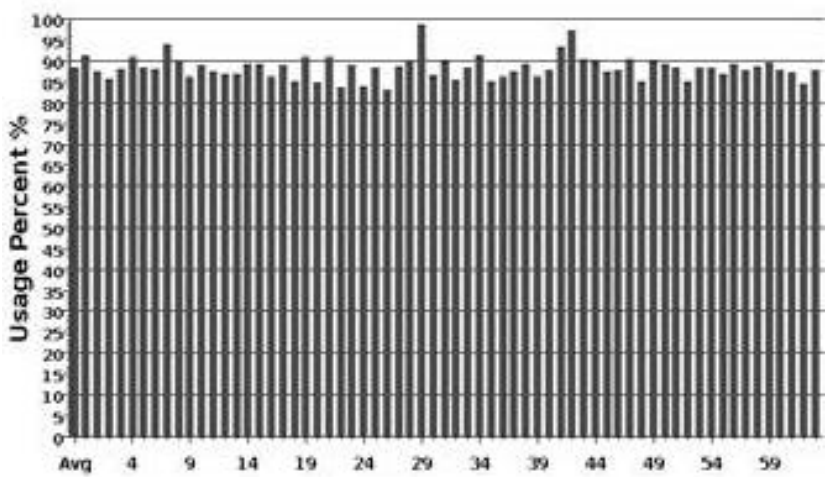
- **AMPI:**
  - Virtualization ≈ Overdecomposition
  - Migration: move ranks to other processors, to balance load (can be automated by the Charm++ runtime system)

Load-Balance Effect on BRAMS

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



Original processor utilization



Proc. utilization with Load-Balance

Ref: E.Rodrigues, P.Navaux, J.Panetta, **C.Mendes** & L.V.Kalé, "*Optimizing and MPI Weather Forecasting Model via Processor Virtualization*", Proceedings of HiPC, India, 2010.

Ref2: **C.L.Mendes**, L.V.Kalé, E.R.Rodrigues & J.Panetta, "*Adaptive and Dynamic Load Balancing for Weather Forecasting Models*", Proceedings of Cray User Group Meeting, Stuttgart, 2012.

Paralellism with Accelerators

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

Language Adoption

Example: PRAC applications on Blue Waters system (USA) in 2014

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

Language Adoption (2)

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

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 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

Current Challenges (cont.)

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**

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

CAP-372: http://www.lac.inpe.br/~stephan/CAP-372.htm

CAP-396: http://www.lac.inpe.br/~celso/cap396-2018/

Unifesp: Prof. Álvaro Fazenda

ITA: Prof. Jairo Panetta

Thank You!

Questions?