Vishnu: implementation and tests of a Beowulf cluster at the OV/UFRJ

Carlos R. Rabaça / OV-UFRJ

François Chr. Cuisinier / OV-UFRJ

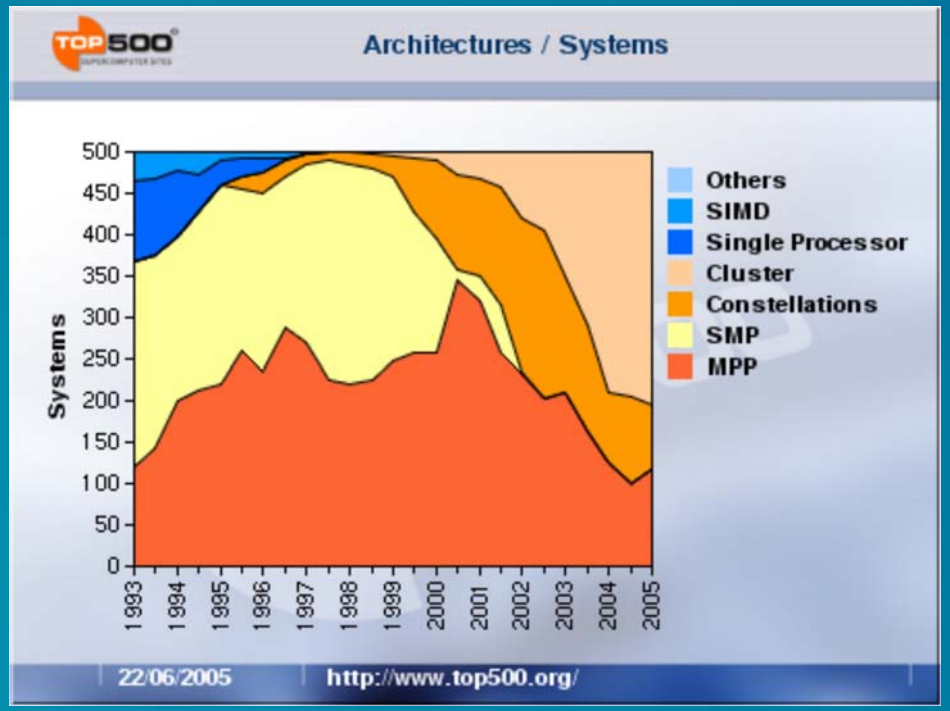
Proposal

- So is it a V.O. that we want?
  We need more than data...
  We want enough compute power to support instant on line data queries, correlations, and computations.
  Obviously we fall short of those ideals!
- How can we get more compute power, for instance, to add two vectors?

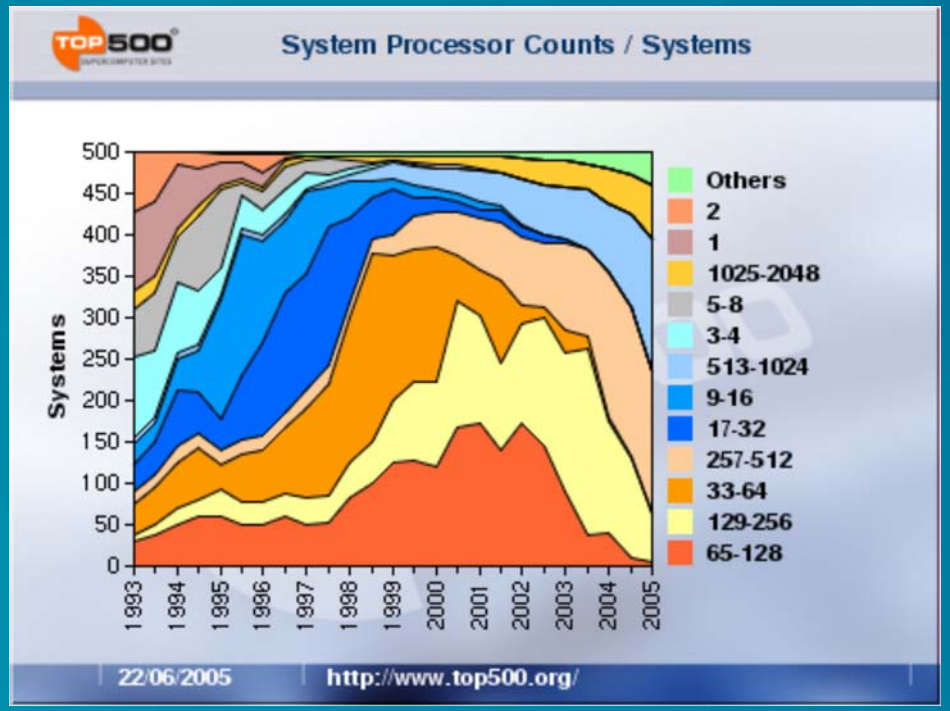
$$(x1, y1, z1) + (x2,y2,z2) = (x1+x2,y1+y2,z1+z2)$$

  - Make a big computer that makes additions really fast
    dedicated high speed computer (70´s, 80´s)
  - Parallelize the computation
    computer cluster (mid 90´s, 00´s)

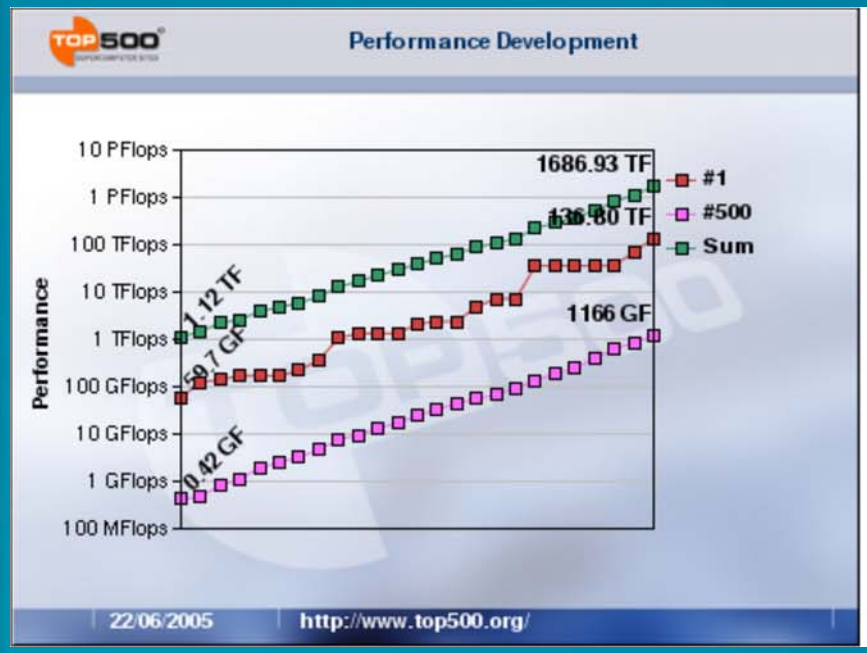
Top500.org

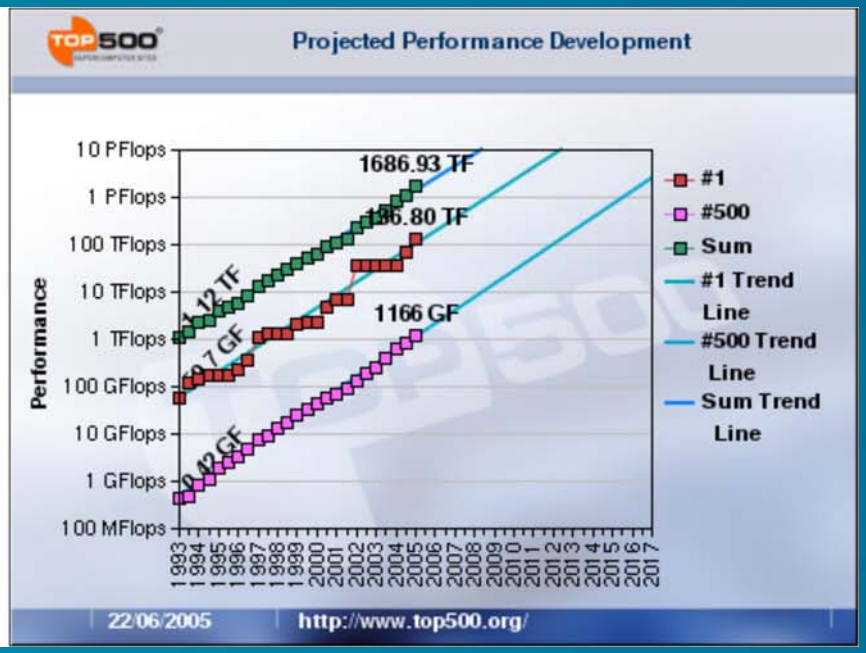

TOP500
Architectures / Systems
Systems
500
450
400
350
300
250
200
150
100
50
0
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
Others
SIMD
Single Processor
Cluster
Constellations
SMP
MPP
22/06/2005
http://www.top500.org/

Top500.org


TOP500
System Processor Counts / Systems
Systems
500
450
400
350
300
250
200
150
100
50
0
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
Others
2
1
1025-2048
5-8
3-4
513-1024
9-16
17-32
257-512
33-64
129-256
65-128
22/06/2005
http://www.top500.org/

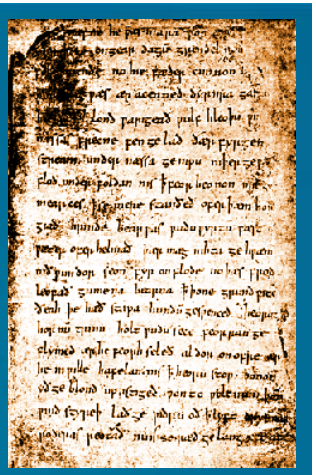
Top500.org





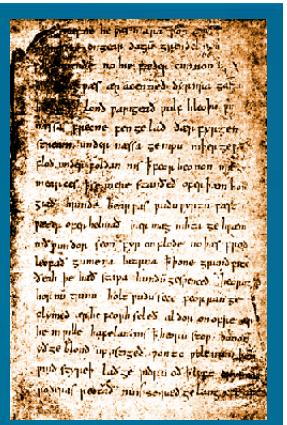
Flops = FLoating-point Operations Per Second

Beowulf.org

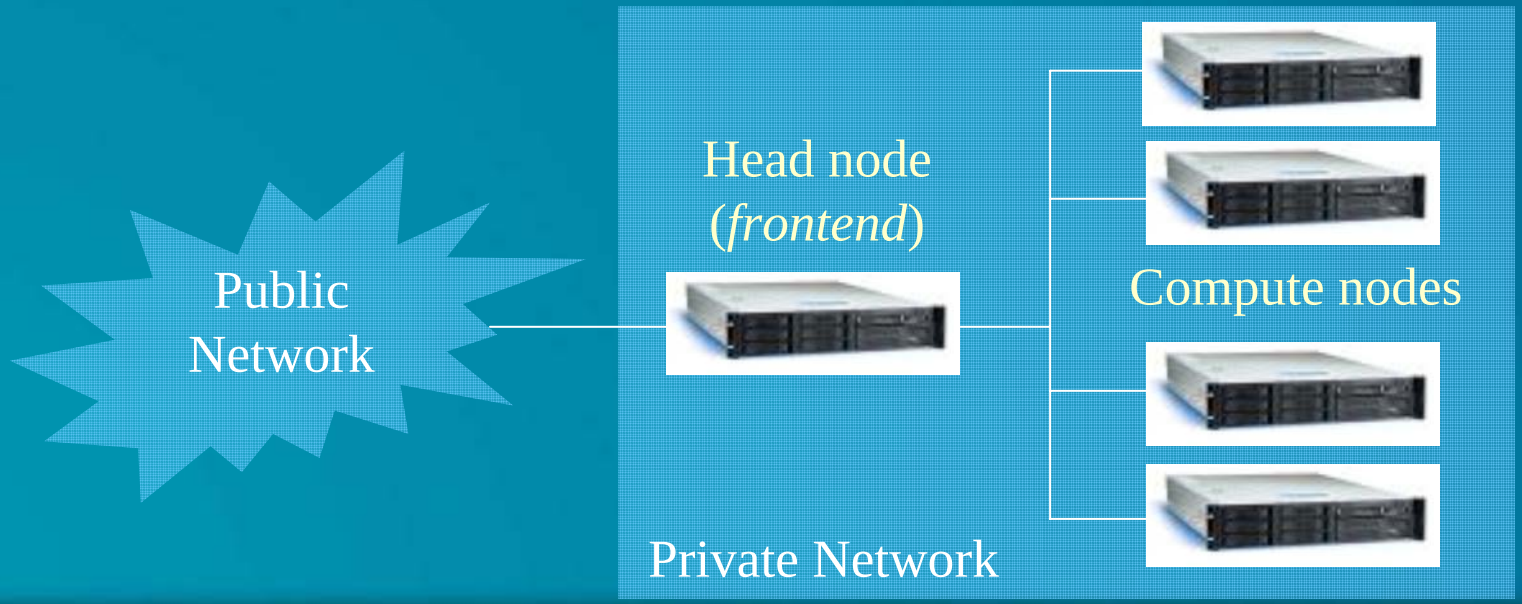


- Mass market networking can be very cheap and have reasonably high performance these days.
- Noticing that, in 1994, Don Becker e Thomas Sterling, from NASA, came up with the idea of using commercial components to build a high performance computer (HPC) from commodity parts
  16 Intel 486 computers running Linux, connected with 10 Mbps Ethernet – “BEOWULF” Project
- “Beowulf” denotes an approach – commodity hardware and networking – rather than a piece of software or an operating system.

Beowulf.org



- Definition of a Beowulf cluster:
  - Independent machines, connected in a private network, running open software (read GNU Linux) and aiming scalable performance computing.



Vishnu: a concept cluster

- Implementation at the Department of Astronomy, OV/UFRJ.
- 4 Dell computers (1 *frontend* e 3 nodes):
  Intel (x86) Celeron 2.4 GHz processor (32-bit ☹)
  512 MB RAM
  80 GB HD
  200 Mbps Ethernet network
  1 KVM
- Peak theoretical performance:
  Peak GFlops = [CPUs] * [CPU clock rate (GHz)] * [CPU floating point issue rate] = 4 * 2.4 * 2 = 19.2

Vishnu

Vishnu

- Basic software infrastructure:

RocksClusters (www.rocksclusters.org)

and

CentOS (www.centos.org)

  - Both are free
  - Out-of-the-box solution (or the closest you can get to it...)
  - Low labor requirements, easy setup
  - Scalability

Vishnu

- Many rolls:
  - Area51: security
  - SGE: Sun Grid Engine (scheduler)
  - Ganglia: status
  - Viz: visualization no

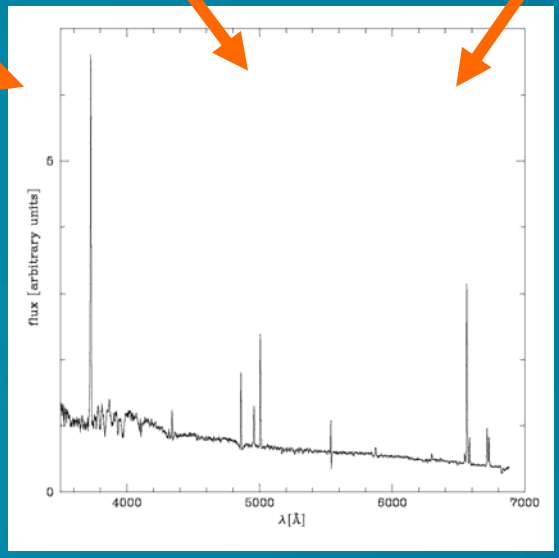
    Chromium, VTK, Nvidia tools
  - Grid: Globus toolkit
  - Condor: Grid scheduler no
  - Intel Fortran: F77 / F90 / F95 programming
  - Scalable Informatics - status

Lines of Research to be Developed

- Observational Research
  - Individual objects
    3-Dimensional spectroscopy of circunstellar nebula and galaxies
  - Large samples
    Studies of galactic evolution from large surveys, e.g., SDSS

Spectral modeling of photoionized regions / galaxies

?

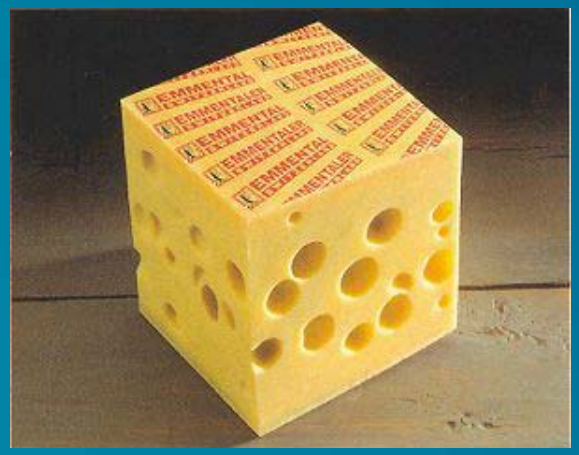


In general, a 1D structure is assumed (i.e., Cloudy)...

Spectral modeling of photoionized regions / galaxies

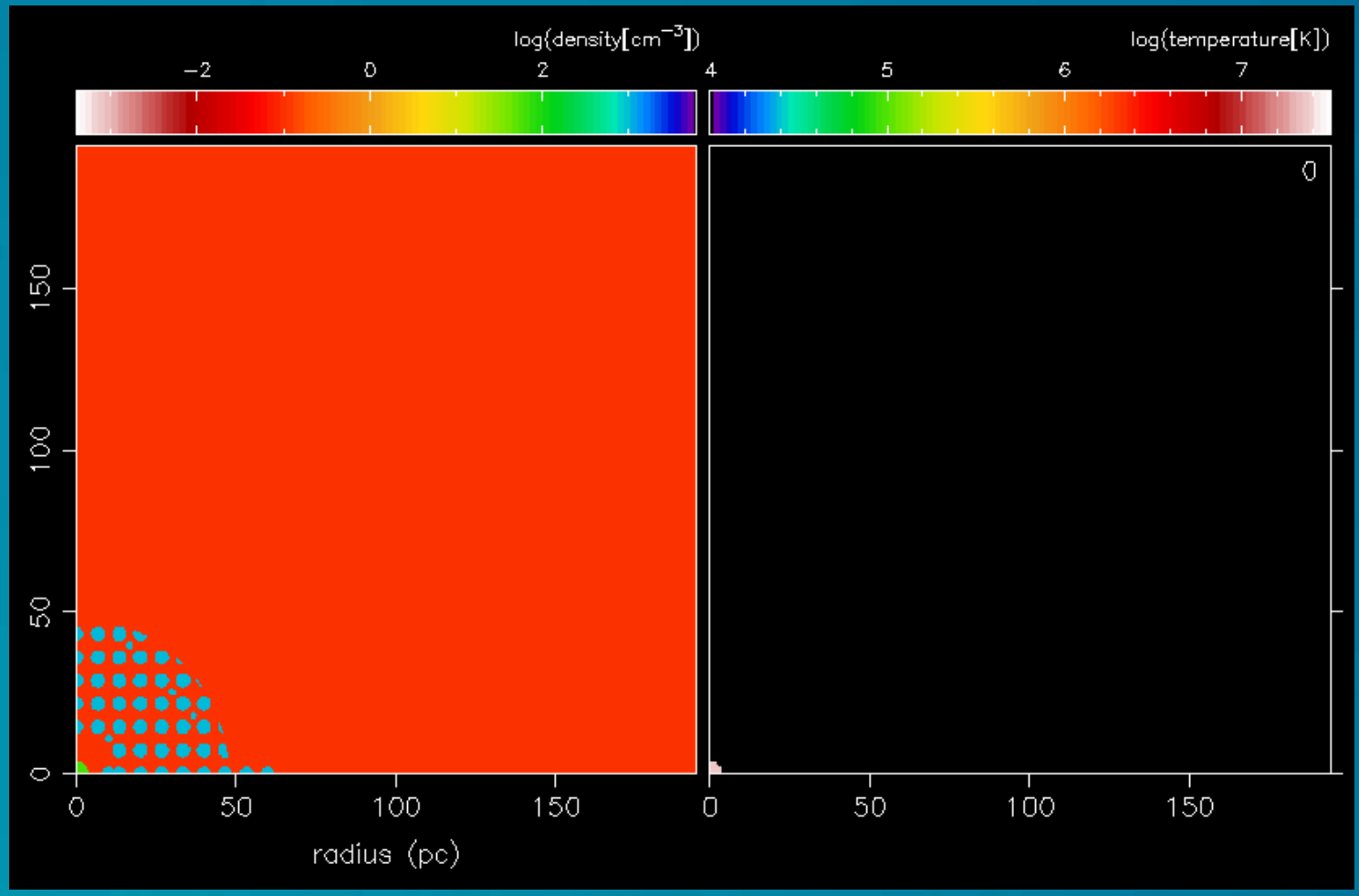
- Mocassin 3D (Ercolano et al. 2003)
- Modeling is physically consistent with any 3D structure type

Basis for a Theoretical VO



- Problem:
  - Heavy compute (~ 1h to run a simple benchmark in one node) ⇒ cluster!

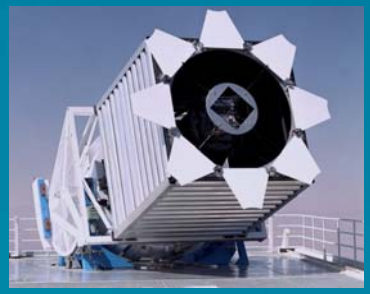
We can model more realistic structures as well...

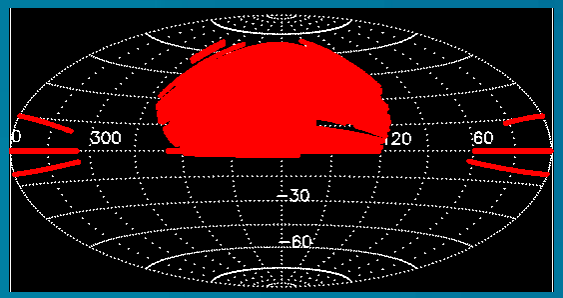


Tenorio-Tagle et al. 2006

The Sloan Digital Sky Survey

2.5-m dedicated telescope

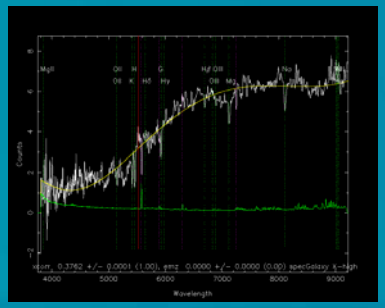


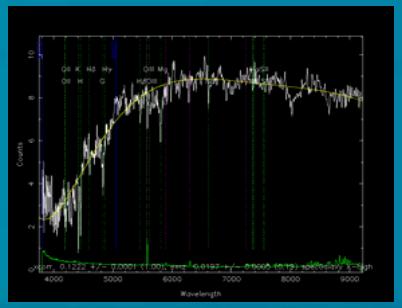


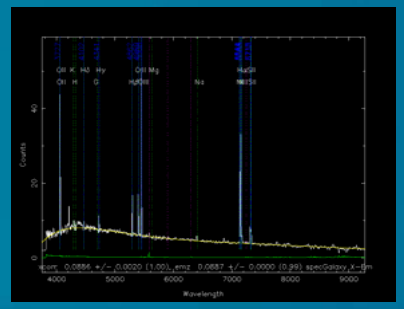
8.000 degree^2

5-band (ugriz) images for 200×10^6 objects (15 TB)

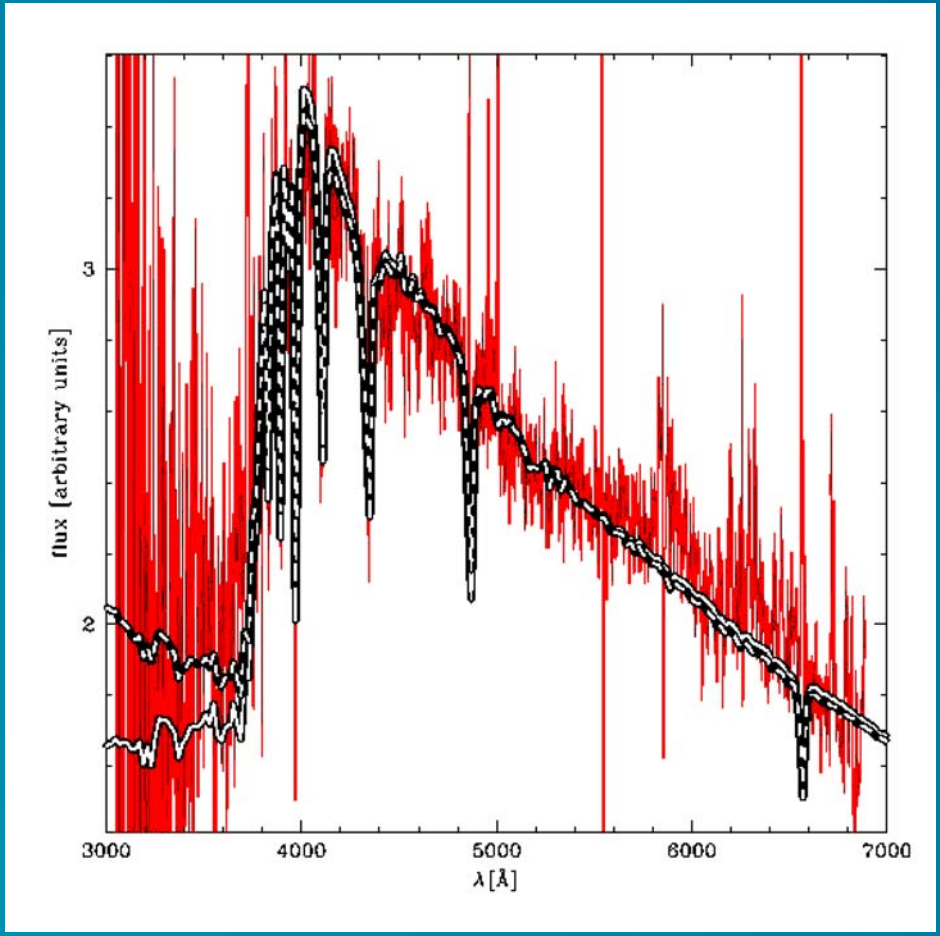
Spectrum for 10^6 objects (250 GB)





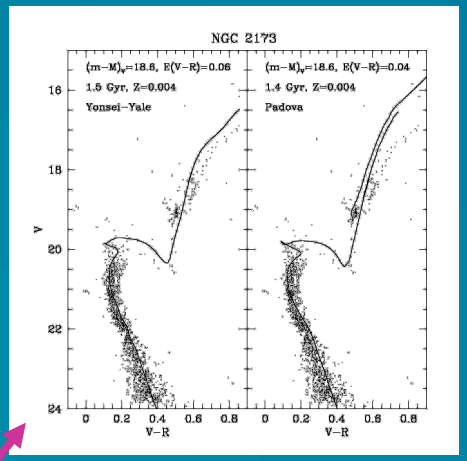


Population Synthesis

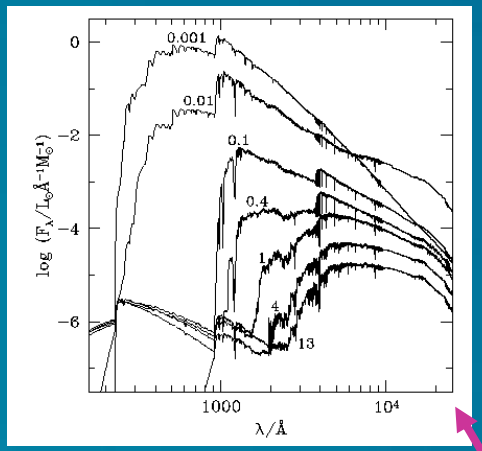


Westera, Cuisinier, Telles, Kehrig 2004

Population Synthesis - how?



Bertelli et al. 2003



Bruzual & Charlot 2003

- Stellar evolution theoretical tracks

  ⇒ Isochronal ages τ from the T_{eff} x L plane (or T_{eff} x log g (equiv.))

- Stellar Spectrum (theoretical or observational)

  ⇒ (**[M/H]** , log g, T_{eff})

⇒ Simple Stellar Population SSP (**τ, [M/H]**)

Population Synthesis & SDSS

- Population Synthesis is easy...
  e.g., we can limit ourselves to 2 parameters (τ,[M/H]) to be determined for a galaxy...
  But can easily give wrong results!
- We need a set of parameters (τ_i, [M/H]$_i$)...
  in order to get a more realistic representation of the stellar formation history in galaxies
- Large compute power
  $\Rightarrow$ cluster!

Benchmarks
- So how fast is it?

- Meaningless, unless they relate to actual application! But we don't buy computers to run benchmarks...
- High Performance Linpack (HPL)
  ⇒ used in the Top500 rankings
  Solves a system of linear equations using MPI, so that the nodes are communicating with each other during the calculation
  - Tends to stress memory, floating point, and I/O
  - Be careful – HPL falls off a cliff if you have a bigger problem than can fit in physical memory – then things start to page in VM, and it becomes a measure of disk performance, which will be lousy compared to memory. Also highly sensitive to input parameters

Benchmarks
- So how fast is it?

- HPCChallenge ⇒ designed to augment Top500 list
  Bound to performance of many applications as a function of memory access characteristics
  - Composed of several well known computational kernels
    - HPL Toward Peak Performance – FPRE for solving a linear system of equations
    - DGEMM – FPRE for matrix-matrix multiply
    - PTRANS – parallel matrix transpose
    - RandomAccess – rate of integer random updates of memory (GUPs)
    - STREAM – sustainable memory bandwidth in GB/s
    - FFT – FPRE of double precision complex one-dimensional DFT
    - Bandwidth/latency tests of a number of simultaneous communication patterns
  - Provides framework for including additional benchmarks

HPL and HPCC

- *machines* file - list of computer nodes; need to enter a host twice if it is dual processor
- *HPL.dat* or *hpccinf.txt* files - P X Q should equal the number of processors and be approximately equal. Ns controls problem size; if this gets too big you will run out of memory and the benchmark will appear to not terminate.

  – HPL:
  > /path/to/mpirun -nolocal -n 4 –machinefile machines /path/to/xhpl

  – HPCC:
  > lamboot –d machines
  > /path/to/mpirun –n 4 /path/to/hpcc

HPCC results

1 node:

HPL_Tflops=0.0020719
DGEMM_N=1280
StarDGEMM_Gflops=2.87347
SingleDGEMM_Gflops=2.86551
StarRandomAccess_GUPs=0.00664904
SingleRandomAccess_GUPs=0.00537335
StarSTREAM_Copy=1.63437
StarSTREAM_Scale=1.69384
StarSTREAM_Add=1.91745
StarSTREAM_Triad=1.92202
SingleSTREAM_Copy=1.63575
SingleSTREAM_Scale=1.69525
SingleSTREAM_Add=1.91766
SingleSTREAM_Triad=1.92202
StarFFT_Gflops=0.366248
SingleFFT_Gflops=0.36765
MPIFFT_Gflops=0.198742
MPIFFT_maxErr=1.24127e-15
MaxPingPongLatency_usec=0

2 nodes:

HPL_Tflops=0.00263904
DGEMM_N=1244
StarDGEMM_Gflops=3.01215
SingleDGEMM_Gflops=3.08495
StarRandomAccess_GUPs=0.00690526
SingleRandomAccess_GUPs=0.00709883
StarSTREAM_Copy=1.80577
StarSTREAM_Scale=1.85835
StarSTREAM_Add=2.17733
StarSTREAM_Triad=2.17439
SingleSTREAM_Copy=1.97602
SingleSTREAM_Scale=2.02844
SingleSTREAM_Add=2.45488
SingleSTREAM_Triad=2.44579
StarFFT_Gflops=0.36858
SingleFFT_Gflops=0.367954
MPIFFT_Gflops=0.0733421
MPIFFT_maxErr=1.27782e-15
MaxPingPongLatency_usec=80.9955

3 nodes:

HPL_Tflops=0.00317976
DGEMM_N=1293
StarDGEMM_Gflops=2.86758
SingleDGEMM_Gflops=2.91166
StarRandomAccess_GUPs=0.00692094
SingleRandomAccess_GUPs=0.00713531
StarSTREAM_Copy=1.86469
StarSTREAM_Scale=1.91441
StarSTREAM_Add=2.25698
StarSTREAM_Triad=2.25406
SingleSTREAM_Copy=1.97985
SingleSTREAM_Scale=2.03193
SingleSTREAM_Add=2.44947
SingleSTREAM_Triad=2.44377
StarFFT_Gflops=0.367554
SingleFFT_Gflops=0.364723
MPIFFT_Gflops=0.0735241
MPIFFT_maxErr=1.27782e-15
MaxPingPongLatency_usec=79.5698

4 nodes:

HPL_Tflops=0.00425618
DGEMM_N=250
StarDGEMM_Gflops=2.29872
SingleDGEMM_Gflops=2.41835
StarRandomAccess_GUPs=0.00969447
SingleRandomAccess_GUPs=0.0100519
StarSTREAM_Copy=1.82099
StarSTREAM_Scale=1.77897
StarSTREAM_Add=1.97522
StarSTREAM_Triad=1.99173
SingleSTREAM_Copy=1.84182
SingleSTREAM_Scale=1.83924
SingleSTREAM_Add=2.01424
SingleSTREAM_Triad=2.03642
StarFFT_Gflops=0.386494
SingleFFT_Gflops=0.384536
MPIFFT_Gflops=0.0102013
MPIFFT_maxErr=8.67112e-16
MaxPingPongLatency_usec=81.0016

Vishnu...!