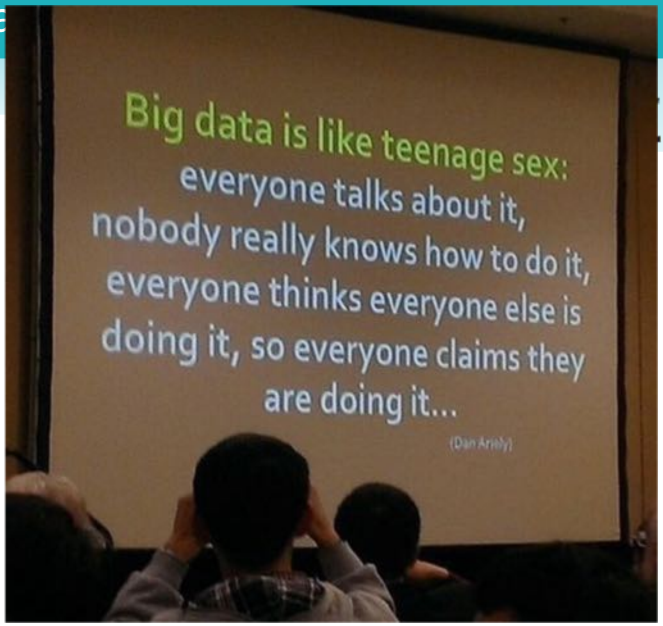
imati

eScience: new information technologies for research

Daniele D’Agostino
dagostino@ge.imati.cnr.it

Basic concepts



Big Science: large-scale scientific research consisting of projects funded usually by a national government or group of governments

- Big machines/Hardware
- Big staffs
- Big data
- Big budgets

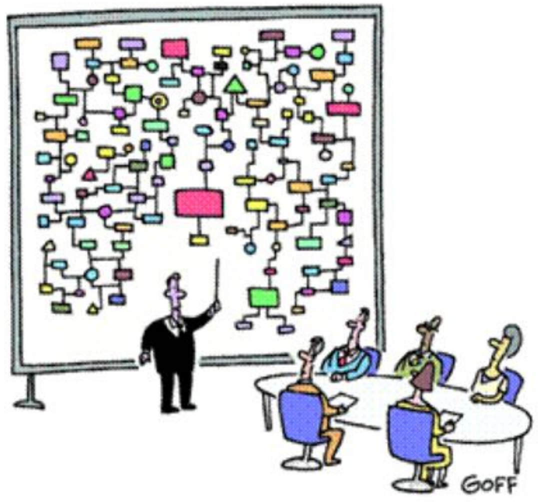
- Military science and military technology
  - Manhattan Project [1]
- Physics
  - Shiva laser → Nova laser → National Ignition Facility
  - Extreme Light Infrastructure
- Space exploration
- Life sciences
  - Human Genome Project
  - Human Brain Project
- Particle Physics (ex: Large Hadron Collider)
- Astronomy (ex: Hubble Space Telescope, other Great Observatories)
  - Gravitational-wave astronomy

Big science – Big IT Infrastructure

- The Worldwide LHC Computing **Grid** (WLCG) project is a global collaboration of more than 170 computing centres in 42 countries, linking up national and international grid infrastructures.
- The mission of the WLCG project is to provide global computing resources to store, distribute and analyse the ~50-70 Petabytes of data expected every year.
- Numbers
  - Over 2 million tasks run every day
  - 1 million computer cores
  - 1 exabyte of storage

http://wlcg-public.web.cern.ch

Other basic concepts

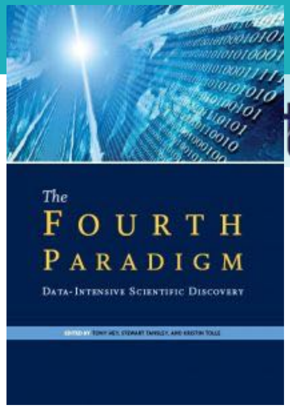


"And that's why we need a computer."

- Big Science and Big Data require Big IT Infrastructure

- E-Science:
  - (long) the application of computer technology to the undertaking of modern scientific investigation, including the preparation, experimentation, data collection, results dissemination, and long-term storage and accessibility of all materials generated through the scientific process. These may include data modeling and analysis, electronic/digitized laboratory notebooks, raw and fitted data sets, manuscript production and draft versions, pre-prints, and print and/or electronic publications
  - (short) eScience promotes innovation in collaborative, computationally- or data-intensive research across all disciplines, throughout the research lifecycle
  - e- = enhanced? enabled? ..

e-Science



- Turing award winner Jim Gray imagined "data-intensive science" or "e-science" as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge.
- e-Science is a research field
- e-Science is done on e-infrastructures
  - Large, distributed, advanced, shared set of resources
  - Data - https://nxsa.esac.esa.int/nxsa-web/#home
  - Computing - https://www.egi.eu
  - Networks - https://www.geant.org
  - Visualization - https://vis.tacc.utexas.edu
  - Software - https://catalog.sciencegateways.org/#/home

A.M. TURING AWARD

Digital Science Since 2005
eScience SERIES
IEEE INTERNATIONAL CONFERENCE

e-infrastructure / cyberinfrastructure

- enable researchers in different locations across the world to **collaborate** in the context of their home institutions or in national or multinational scientific initiatives.
- Scientists can **work together** by having shared access to unique or distributed scientific facilities (including data, instruments, computing and communications).
- e-infrastructures have not only become necessary to deal with increased complexity in tackling scientific challenges, but also as a strategic tool for fostering **collaborative** innovation globally
- How: an example is represented by science gateways

e-Science and Astronomy

The Association of Universities for Research in Astronomy (AURA) is organizing a series of workshops on "Petabytes to Science", supported by The Kavli Foundation. These workshops aim to identify the highest priority technologies and capabilities required for a broad-based user community to take full scientific advantage of the opportunities in data-intensive astronomy being created by the Large Synoptic Survey Telescope (LSST), the Wide Field Infrared Space Telescope (WFIRST), and other massive survey facilities. We also looked at the implications for the workforce, education and public outreach.

https://petabytestoscience.github.io

Table 1. Astronomer recommendations.

Recommendation	Area	Term
REC-1 Adopt common data models throughout the astronomical community	Data Management	Short
REC-3 Proprietary data time scales should be limited, and all datasets should be eventually made publicly available	Data Management	Short
REC-8 Improve long-term software and service support	Technology	Short
REC-11 Funding for sustaining core astronomical "community infrastructure" projects	Software	Medium
REC-12 Cultivating a sustainable research software ecosystem	Software	Short
REC-13 Create funding models and programs to support the development of advanced algorithms and statistical methods specifically targeted to the astronomy domain	Analysis	Medium
REC-14 Build automated discovery engines	Analysis	Long
REC-15 Promote interdisciplinary collaboration between institutions, fields, and industry	Analysis	Long
REC-16 Develop an open educational curriculum and principles for workforce training in both algorithms and statistics	Analysis	Medium
REC-17 Encourage, support, and require open publication and distribution of algorithms	Analysis	Short
REC-22 Software training as part of science curriculum	Workforce	Medium

E-Science categories

- Software
- Technology and Infrastructures
  - Computing infrastructures
  - Science gateways
- Data Management

Part I: Software

Software

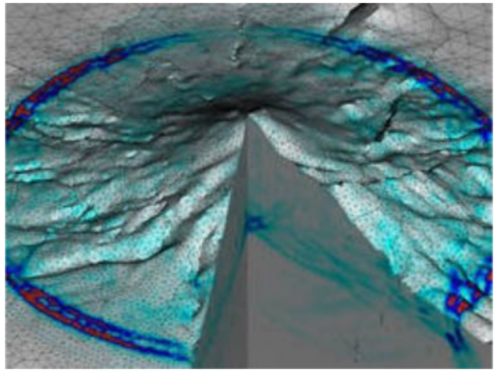
- None wants to pay for the software

BUT

- **The fundamental purpose of astronomical software is to *allow* science to be done**.
- To repeat, the purpose of software is to allow us, the users, to do our jobs, which is to discover new and interesting things about the universe. **Thus, software is necessary (but not sufficient) to accomplish this goal**.
- We must remember that **better software can be equivalent to bigger telescopes and new high tech instruments**. One gets more out of the data and more of the data can be useful.
- ***I believe that if software were easier to use and more robust we would get a lot more science out of our present instruments***.

- R. Mushotzky, in Astronomical Data Analysis Software and Systems XX. 2011, ASP Conf. Proc., Vol. 442, p.235

Why it is stupid – part 1



The extensive optimization and the complete parallelization of the 70,000 lines of SeisSol code results in a peak performance of up to 1.42 petaflops.

This corresponds to 44.5 percent of Super MUC's theoretically available capacity

https://link.springer.com/chapter/10.1007/978-3-319-07518-1_1

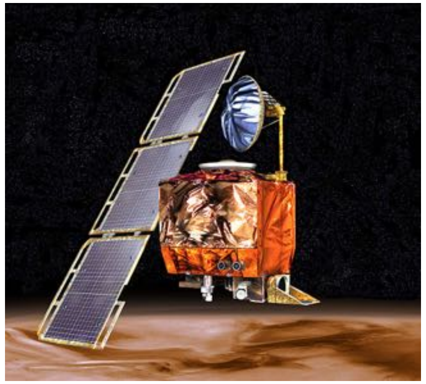
TOP 10 Sites for June 2012

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	16,324.8	20,132.7	7,890
2	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
3	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,162.4	10,066.3	3,945
4	Leibniz Rechenzentrum Germany	**SuperMUC** - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband FDR IBM	147,456	2,897.0	3,185.1	3,423

Name	MPI	# cores	Description	TFlop/s/island	TFlop/s max
Linpack	IBM	128000	TOP500	161	2560
Vertex	IBM	128000	Plasma Physics	15	245
GROMACS	IBM, Intel	64000	Molecular Modelling	40	110
Seissol	IBM	64000	Geophysics	31	95
waLBerla	IBM	128000	Lattice Boltzmann	5.6	90
LAMMPS	IBM	128000	Molecular Modelling	5.6	90
APES	IBM	64000	CFD	6	47
BQCD	Intel	128000	Quantum Physics	10	27

Why it is stupid – part 2

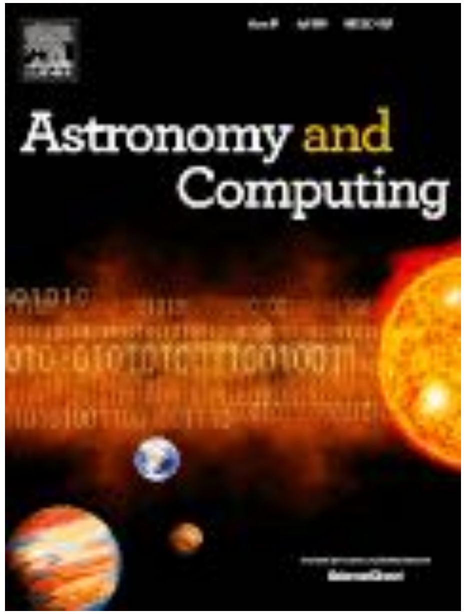
- It is not only a waste of time, computing power and power consumption…
- Much of the code is looked after as a pet: it has a given name, is unique, not shared with others or documented.
- These shortcomings can have real-world consequences, as illustrated by the failed Mars Climate Orbiter mission
  - software calculations were carried out assuming metric units, but navigation software was programmed assuming imperial units, leading to a premature and fiery end to the mission in the Martian atmosphere.
- And what about Mariner 1 in 1962?



```
DO 15 I = 1.100
```
when what should have been written was:
```
DO 15 I = 1,100
```
but somehow a dot had replaced the comma. Because Fortran ignores spaces, this was seen by the compiler as:
```
DO15I = 1.100
```
which is a perfectly valid assignment to a variable called DO15I and not at all what was intended.

Therefore

- Software training as part of science curricula
- Interdisciplinary collaborations
- Software-related publications



ELSEVIER

SoftwareX

Volume 9, January–June 2019, Pages 28-34

Some definitions

- High Performance Computing (HPC)
  - speed up programs as much possible so that results are achieved more quickly
- High Throughput Computing (HTC)
  - The focus is not to speed up individual jobs
  - But to execute many copies of the *same program* at the *same time*
  - Many copies of the same program to run *in parallel* or *concurrently*
  - Maximize the throughput
- HTC infrastructures tend to deliver large amounts of computational power over a long period of time
  - In contrast, HPC environments deliver a tremendous amount of compute power over a short period of time
- HTC is more interested in how many jobs they can complete over a long period of time instead of how fast an individual job can complete.
- HPC is more interested in squeezing all the possible GFLOPS from a machine
  GFLOPS: Billions of Floating Point Operations per second, calculated as :

$$\text{GFLOPS} = \text{sockets} \times \frac{\text{cores}}{\text{socket}} \times \text{clock} \times \frac{\text{FLOPs}}{\text{cycle}} \qquad \text{(Clock in GHz)}$$

High Performance Computing and the Software

HPC =

Software development has not evolved as fast as hardware capability and network capacity. **Nominal and sustained performance of computing systems is further diverging**, unless they are manually optimised which limits portability to other systems.

Development and maintenance of software for advanced computing systems is becoming increasingly effort-intensive requiring dual expertise, both on the application side and on the system side.

...

In order to program the next generation of computing systems, everyone must become a parallel programmer!

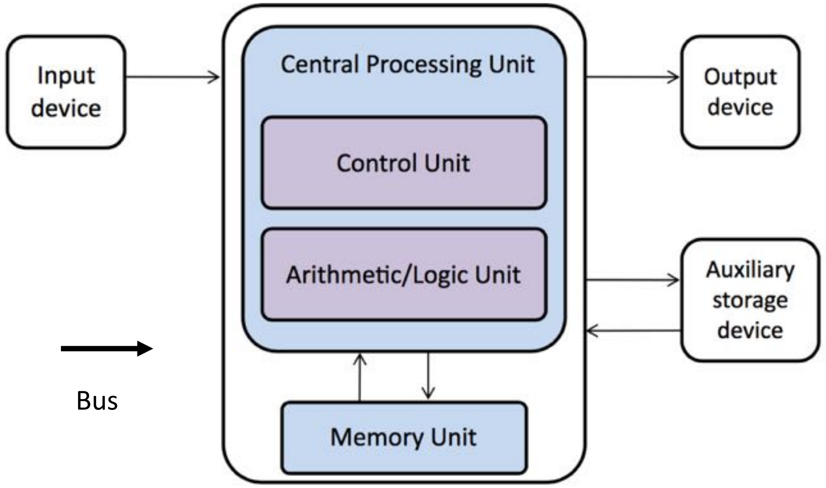
European Commission

Towards a Breakthrough in Software for Advanced Computing Systems

Report from a Workshop organised by the European Commission in preparation for HORIZON 2020

held in July 2012 in Brussels, Belgium

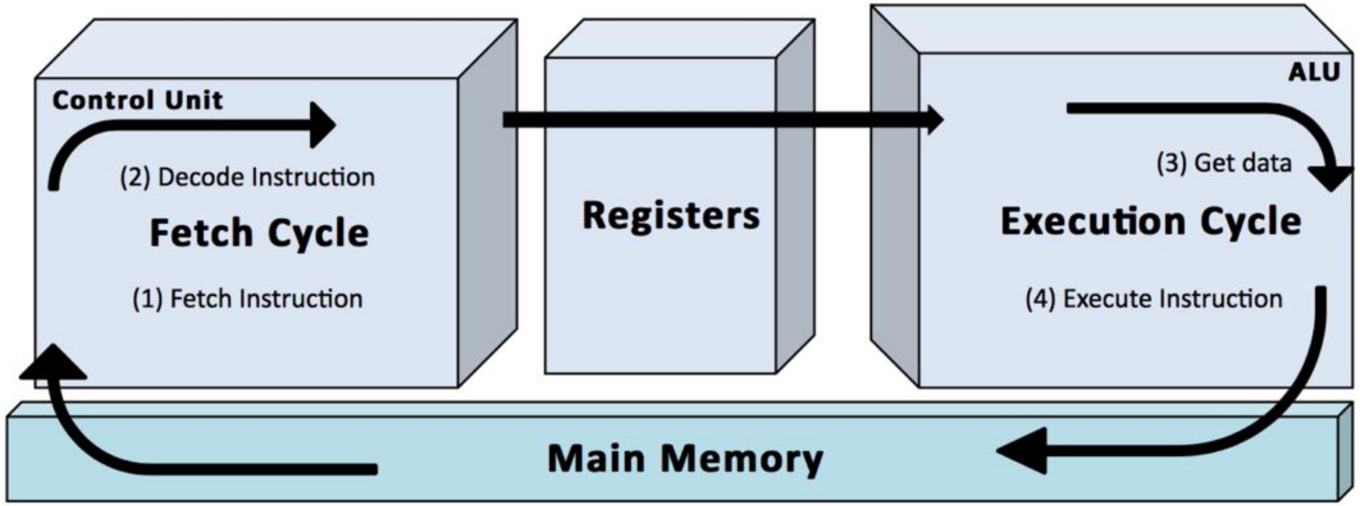
Von Neumann Architecture



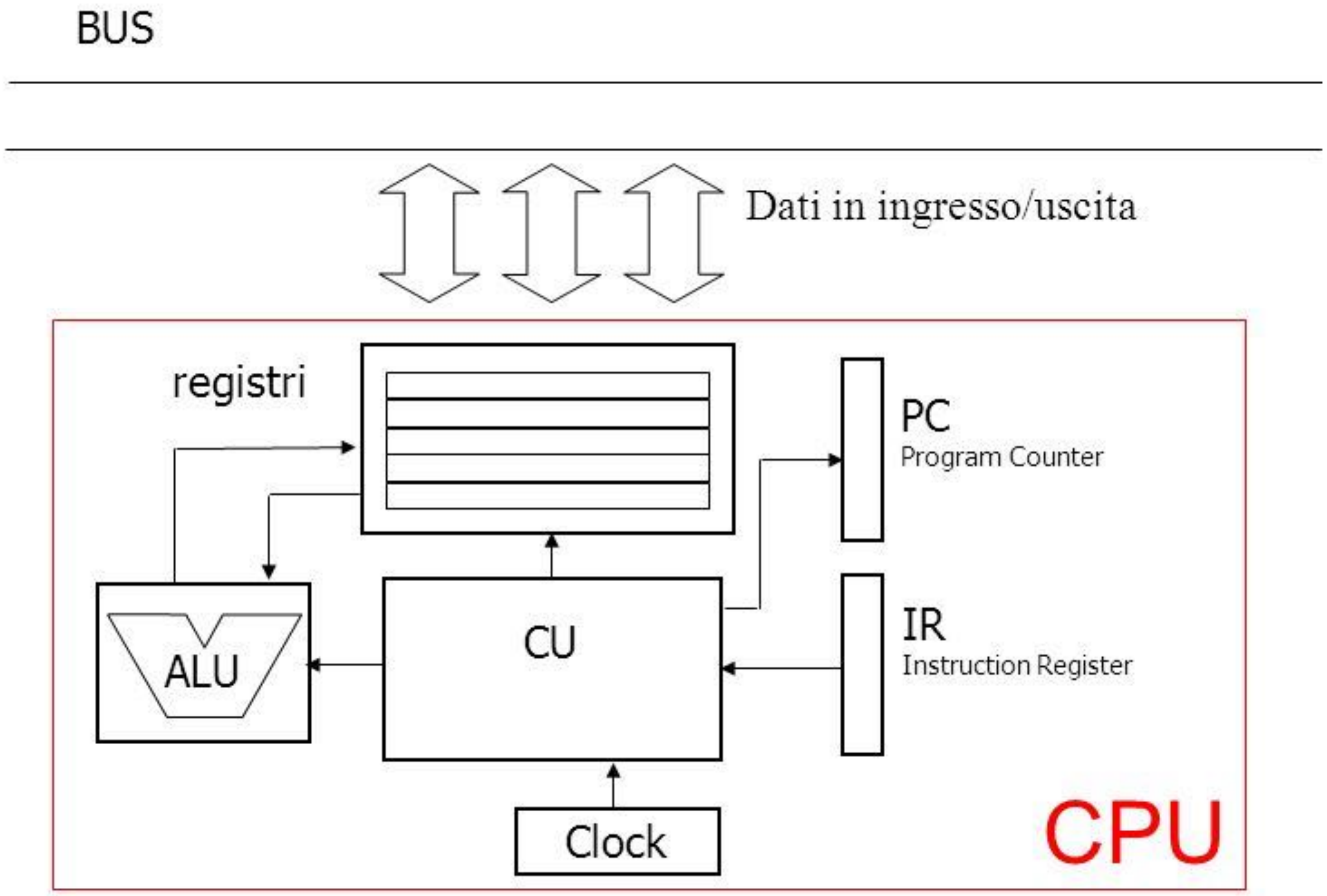
Four main components

- CPU, Memory, I/O devices, BUS

Also the processing cycle has four main steps

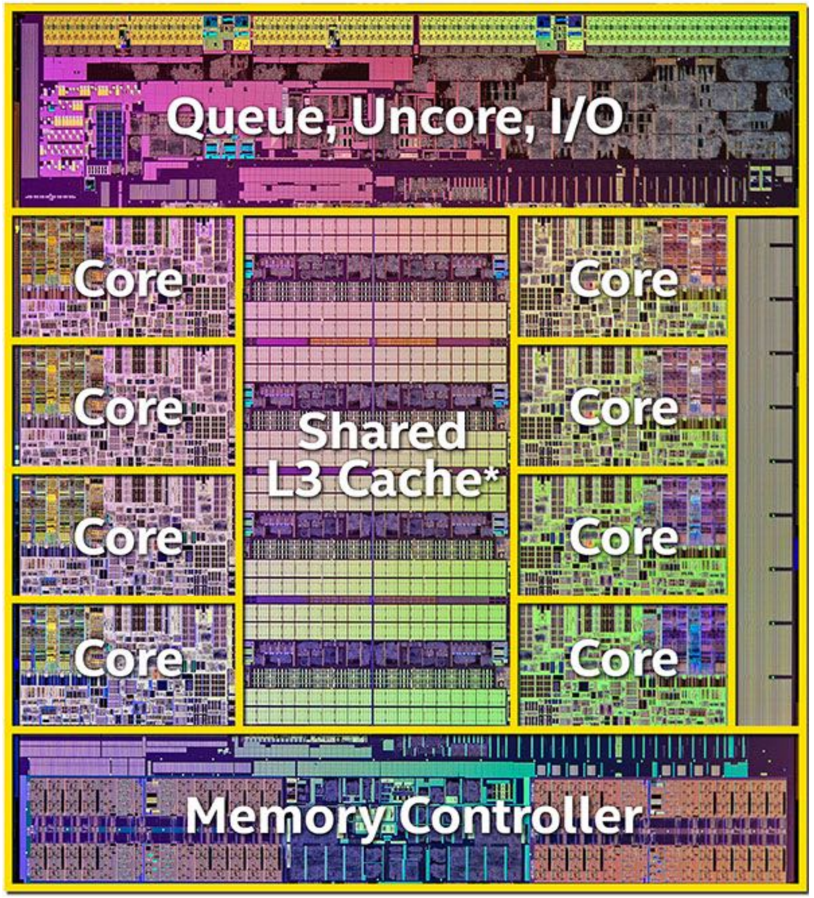


Inside the CPU



CPU programming

- CPU are designed for sequential code, but… 1 core x 3Ghz = 3GFlops at best
- Can we process 1 instruction per clock ?
- Intel/AMD claim to produce >100GFlops CPUs
- To reach such performances we have to exploit:
  - Instruction level (e.g. fma = fused multiply add)
  - Vector processing (e.g. data parallelism)
  - Hyperthreading (more hardware threads/core)
  - Multi-Cores per processor
- We can even go faster with
  - Multi-Processors (or sockets) per node
  - Processors + accelerators (e.g. CPU+GPU)
  - Multi-node



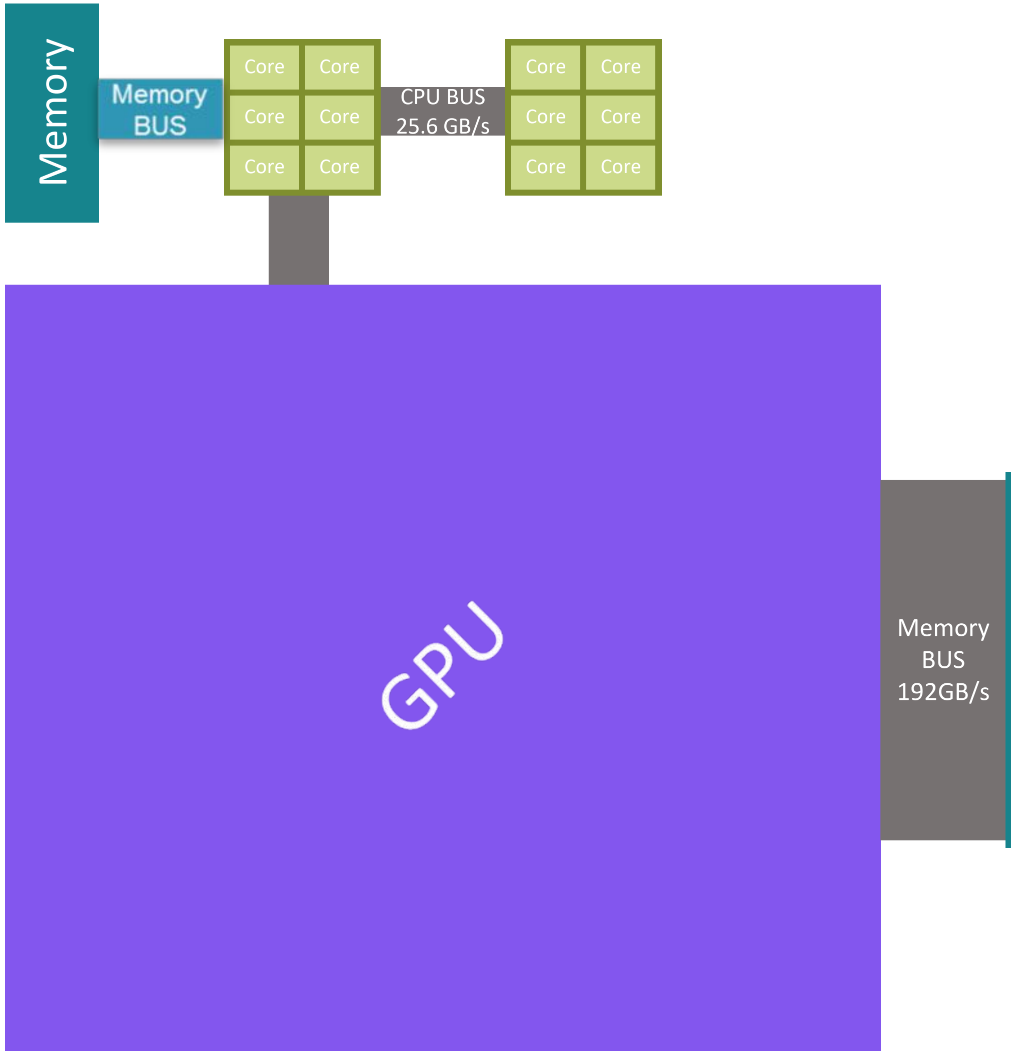
Raw numbers or real performances ? Architecture of test workstation (fp32)

2 Intel Xeon E5645
(115 Gflops, 9.6 per core)

64 GB main memory

nVidia GTX 580
(1581 Gflops)

1.5 GB GPU memory



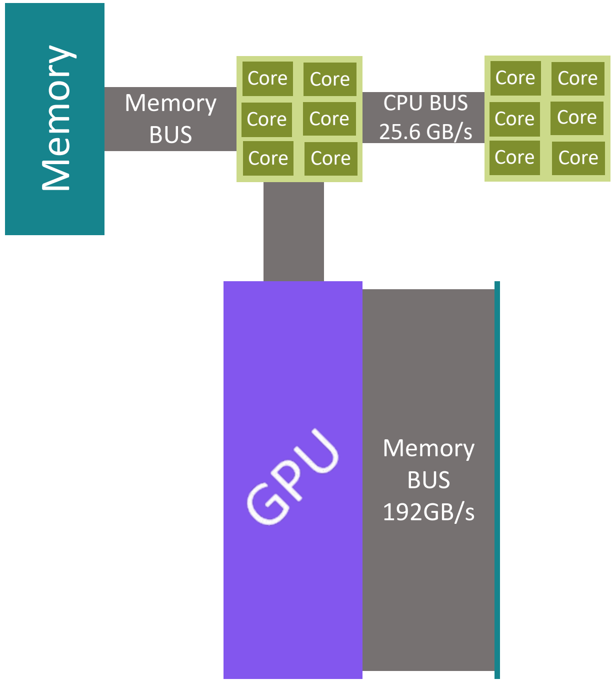
Architecture of test workstation (fp64)

2 Intel Xeon E5645 (58 Gflops)

64 GB main memory

nVidia GTX 580
(198 Gflops)

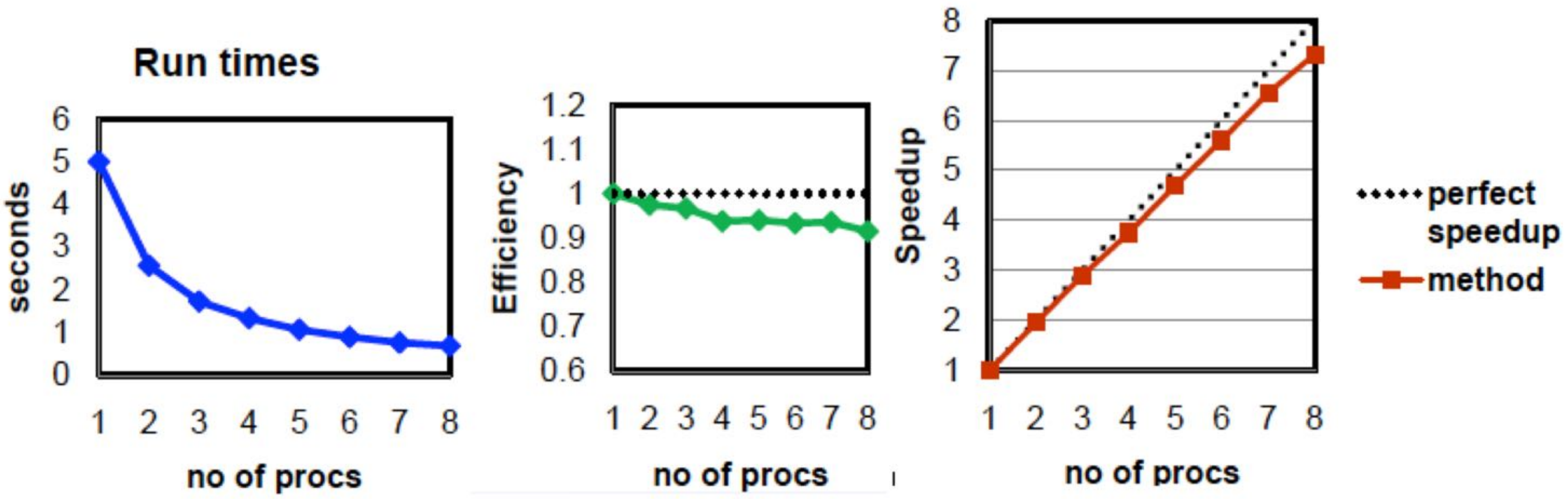
1.5 GB GPU memory



Parallel Computing Performance Metrics

Let T(n,p) be the time to solve a problem of size n using p processors

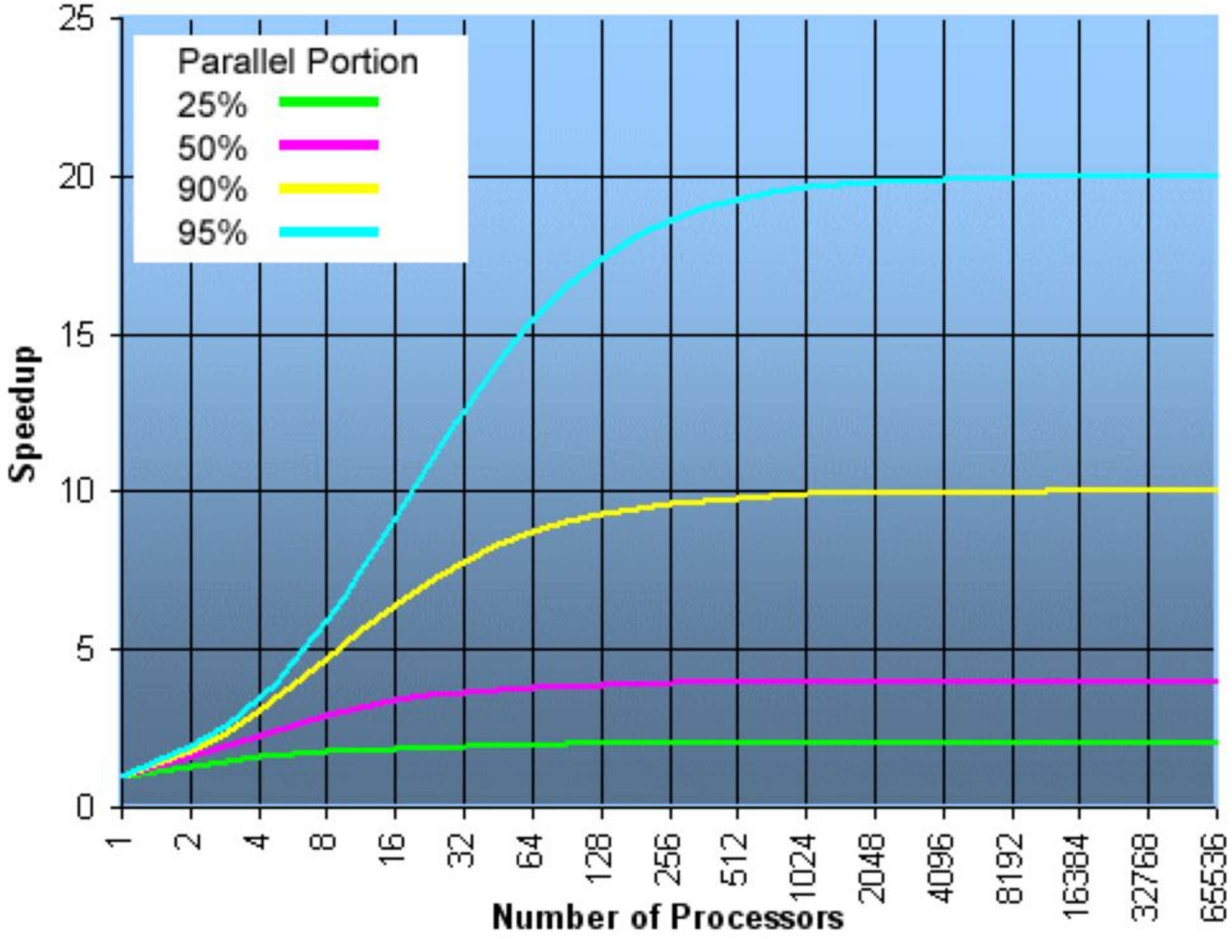
- Speedup: S(n,p) = T(n,1)/T(n,p)
- Efficiency: E(n,p) = S(n,p)/p



Amdahl's Law

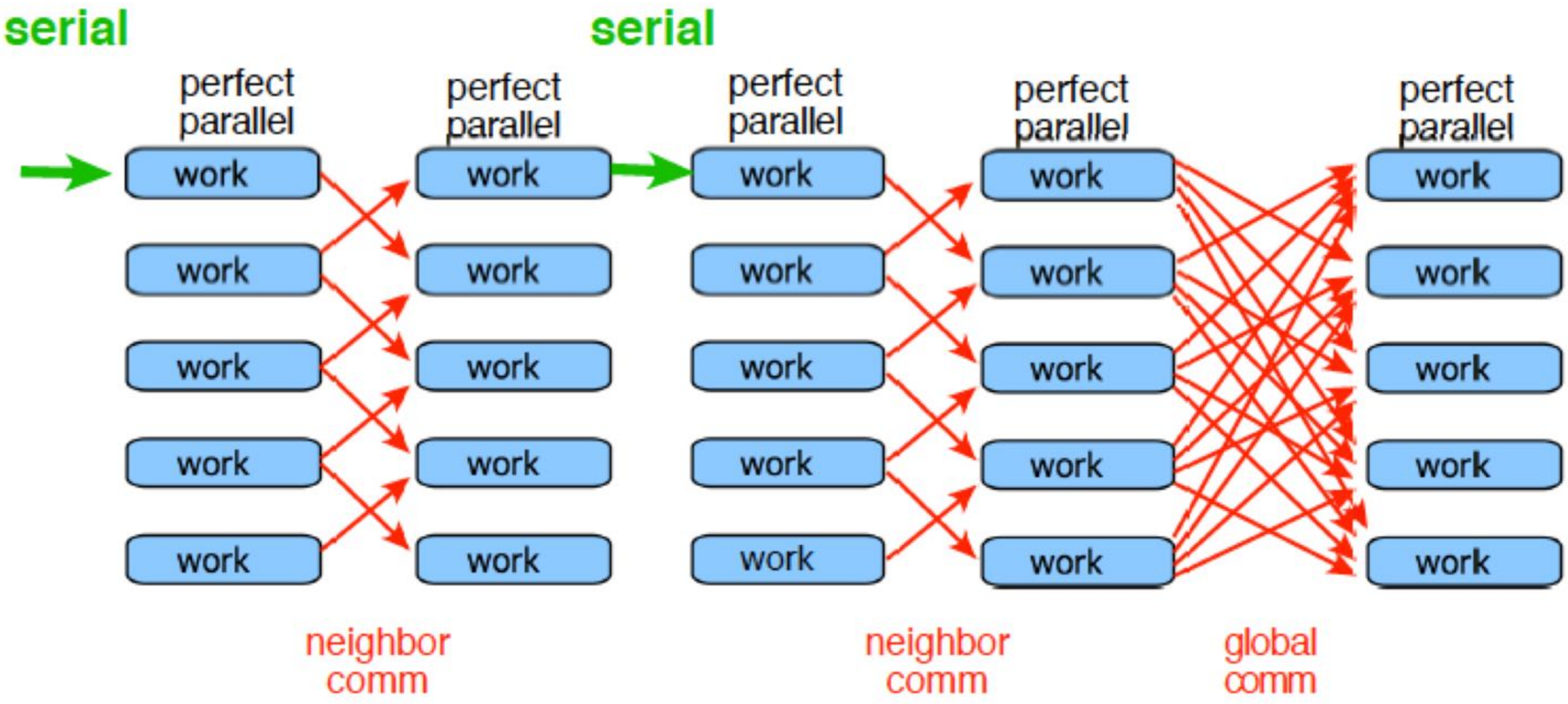
$$Speedup \leq \frac{1}{(1 - pctPar) + \frac{pctPar}{p}}.$$

P no. of processors, (1-pctPar) serial portion of code



Amdahl Was an Optimist

Parallelization usually adds communications/overheads



Amdahl was a Pessimist

Superlinear speedup is very rare.
Some reasons for speedup > p (efficiency > 1)

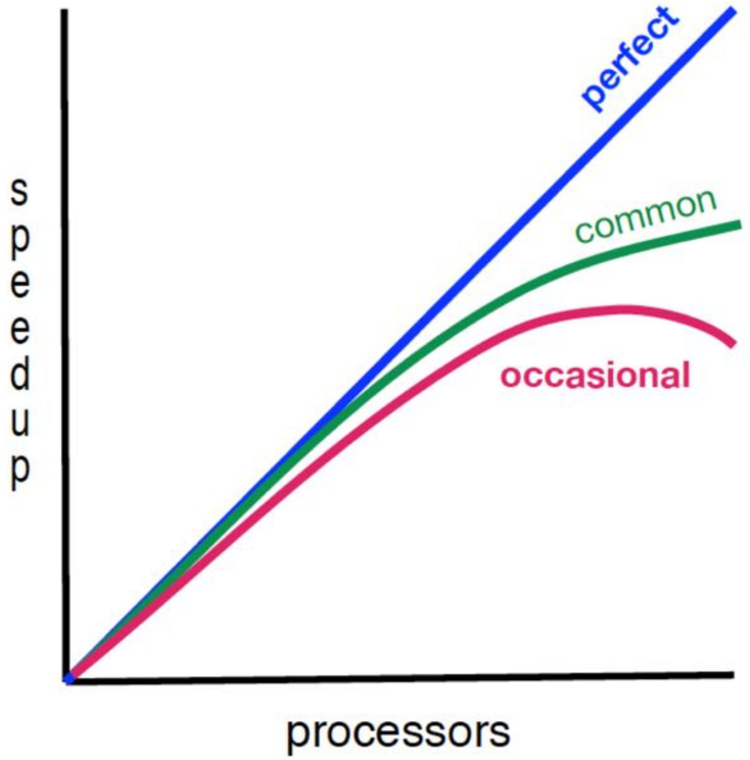
- Parallel computer has p times as much RAM so higher fraction of program memory in RAM instead of disk.

An important reason for using parallel computers

- In developing parallel program a better algorithm was discovered, older serial algorithm was not best possible.

A useful side-effect of parallelization

- In general, the time spent in serial portion of code is a decreasing fraction of the total time as problem size increases.



The lesson is

- Linear speedup is rare, due to communication overhead, load imbalance, algorithm/architecture mismatch, etc.
- Further, essentially nothing scales to arbitrarily many processors.
- However, for most users, the important question is:

Have I achieved acceptable performance on my software for a suitable range of data and the resources I'm using?

The lesson is

- It's easier to optimize a slow correct program than to debug a fast incorrect one
  - *Nobody cares how fast you can compute a wrong answer...*
- Programs typically spend 80% of their time in 20% of the code
  - Programmers typically spend 20% of their effort to get 80% of the total speedup possible for the application
  - *Know when to stop!*
- Don't optimize what does not matter
  - *Make the common case fast!*

The N-Body simulation

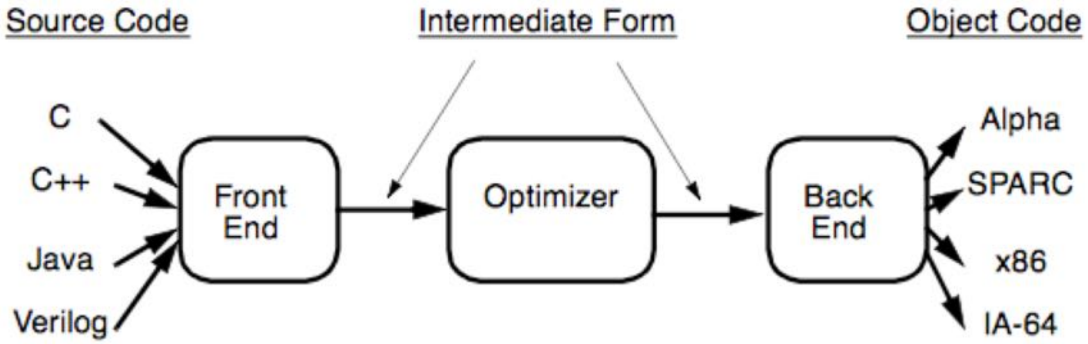
```
for (k=0; k<timesteps; k++) {
  swap(oldbodies,newbodies);
  for (i=0; i<N; i++) {
    tot_force_i[X] = tot_force_i[Y] = tot_force_i[Z] = 0.0;
    for (j=0; j<N; j++) {
      if (j==i) continue;
      //20 floating point operations
      r[X] = oldbodies[j].pos[X] - oldbodies[i].pos[X];
      // analogous for r[Y] and r[Z]
      distSqr = r[X]*r[X] + r[Y]*r[Y] + r[Z]*r[Z] + EPSILON2;
      distSixth = distSqr * distSqr * distSqr;
      invDistCube = 1.0/sqrtf(distSixth);
      s = oldbodies[j].mass * invDistCube;
      tot_force_i[X] += s * r[X];
      // analogous for Y and Z
    }
    //24 flops
    dv[X] = dt * tot_force_i[X] / oldbodies[i].mass;
    newbodies[i].pos[X] += dt * ( oldbodies[i].vel[X] + dv[X]/2 );
    newbodies[i].vel[X] = oldbodies[i].vel[X] +dt * dv[X];
    // analogous for Y and Z
  }
}
```

Compilers and Optimizations

- Compilers translate one language into another
- Optimize the code: transform a computation to an equivalent but “better” form

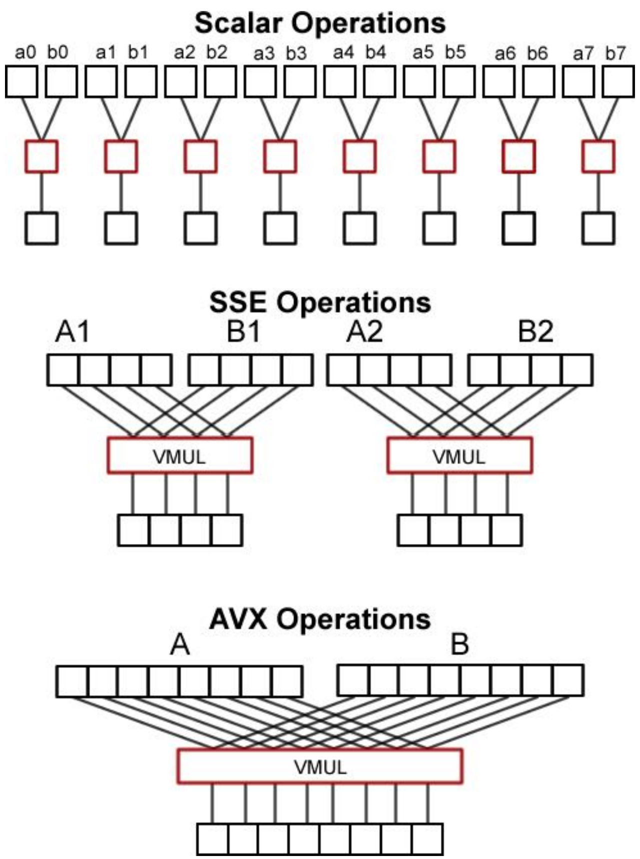
```
Executiontime=Operationcount*Machinecyclesperoperation
```

- Minimize the number of operations
- Replace expensive operations with simpler ones
- Possibily minimize object code size



N-Body – sequential algorithm

- Maximum theoretical performances:
  9.6 GFLOPs (single core – **SIMD** instructions – 551$ CPU, 700$ icc)
- All-pairs algorithm O(N^2) x timesteps
- We consider timestep=100 and N=1K, 10K




```
void multiply(float *A, float *B, float * C, int size) {
        for (i = 0; i < size; i++) {
                C[i] = A[i] * B[i];
        }
}
```

Compiler	GFLOPs		Efficiency		MFLOPs/US$	
	1K	10K	1K	10K	1K	10K
gcc	1.59	1.67	16.6%	17.4%	3	2
Intel	2.70	5.71	28.1%	59.5%	2	5

N-Body – OpenMP and MPI

```
#pragma omp parallel for private(j, r, distSqr,
        distSixth, invDistCube, s, tot_force_i, dv)
```

```
for (i=0; i<N; i++) {
   tot_force_i[X] = tot_force_i[Y] = tot_force_i[Z] = 0.0;
   for (j=0; j<N; j++) {
      if (j==i) continue;
      //20 floating point operations
      r[X] = oldbodies[j].pos[X] - oldbodies[i].pos[X];
      // analogous for r[Y] and r[Z]
      distSqr = r[X]*r[X] + r[Y]*r[Y] + r[Z]*r[Z] + EPSILON2;
      distSixth = distSqr * distSqr * distSqr;
      invDistCube = 1.0/sqrtf(distSixth);
      s = oldbodies[j].mass * invDistCube;
      tot_fo
      // ana
   }
   //24 flop
   dv[X] = d
   newbodies
   newbodies
   // analog
 }
}
```

```
bnum = N % numprocs;
bstart=(N/numprocs*id)      + ((id>=bnum) ? bnum   : id);
bstop =(N/numprocs*(id+1)) + ((id>=bnum) ? bnum-1 : id);
bnum = bstop-bstart+1;

MPI_Barrier(MPI_COMM_WORLD); //for timing purpose
for (k=0; k<num_steps; k++) {
  swap(oldbodies,newbodies);
  for (i=bstart; i<=bstop; i++) { ... }
  MPI_Allgatherv(MPI_IN_PLACE, 0, MPI_DATATYPE_NULL, newbodies,...);
}
```

N-Body – OpenMP and MPI

- Maximum theoretical performances:
  57.6 GFLOPs (1 CPU – 6 cores)
  115 GFLOPs (2 CPUs – 12 cores)
- Intel compiler only (cost disregarded)
- OpenMP is easier
- Think parallel

Compiler	GFLOPs		Efficiency		MFLOPs/US$	
	1K	10K	1K	10K	1K	10K
OpenMP - 1 cpu	25.49	33.1	44.3%	57.5%	23	30
OpenMP - 2 cpus	35.50	63.24	30.8%	54.9%	32	57
MPI - 1 cpu	27.21	29.12	47.2%	50.6%	25	26
MPI - 2 cpus	50.27	59.93	43.6%	52.0%	46	54

N-Body – CUDA and OpenACC

```
for (k=0; k<timesteps; k++) {
  swap(oldbodies,newbodies);
  for (i=0; i<N; i++) {
     tot_force_i[X] = tot_force_i[Y] = tot_force_i[Z] = 0.0;
     for (j=0; j<N; j++) {
        if (j==i) continue;
        //20 floating point operations
```

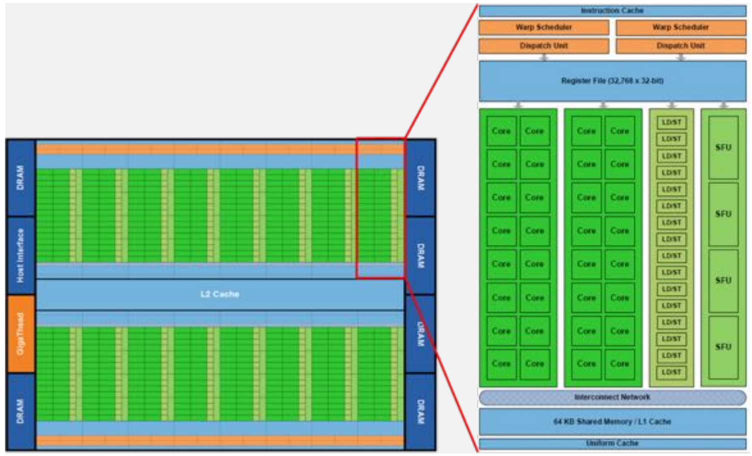
```
distSixth = rsqrtf(distSqr);
invDistCube = distSixth * distSixth * distSixth;
```

```
        distSixth = distSqr * distSqr * distSqr;
        invDistCube = 1.0/sqrtf(distSixth);
        s = oldbodies[j].mass * invDistCube;
        tot_force_i[X] += s * r[X];
        // analogous for Y and Z
     }
     //24 .
     dv[X]
     newbo
     newbo
     // an
  }
}
```

```
#pragma acc data copy(bodies1[0:N], bodies2[0:N])
                 copyin(bodiesvel[0:N], bodiesm[0:N], dt)
for (k=0; k<num_steps; k++) {
  #pragma acc kernels loop independent
  for (i=0; i<num_bodies; i++) {
    tot_force_x = 0.0; ...
    #pragma acc loop independent
                   reduction(+:tot_force_x,tot_force_y,tot_force_z)
```

N-Body – CUDA and OpenACC

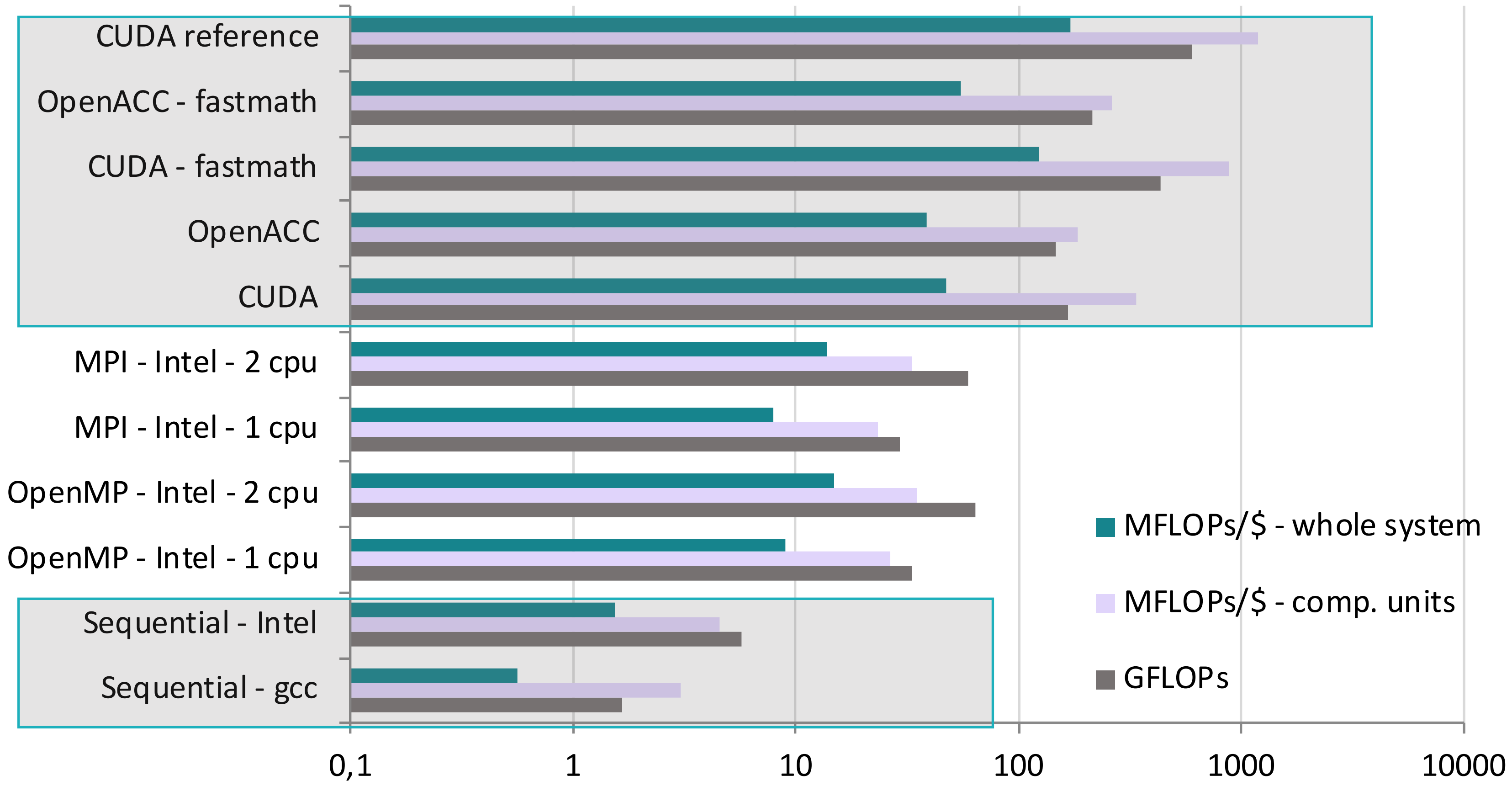
- Maximum theoretical performances:
  1581 GFLOPs (1 GPU – 512 cores – 499 US$)
- OpenACC is much much easier
- Performances with and w/o *fastmath* (rsqrtf function), N=10K



Algorithm	GFLOPs	Efficiency	MFLOPs/US$
CUDA	167.71	10.6%	336
OpenACC	147.57	9.3%	185
CUDA - fastmath	434.46	27.5%	871
OpenACC - fastmath	211.80	13.4%	265
CUDA reference	597.11	37.8%	1197

Performance/Price (10K)

Programmers and tools make the difference!



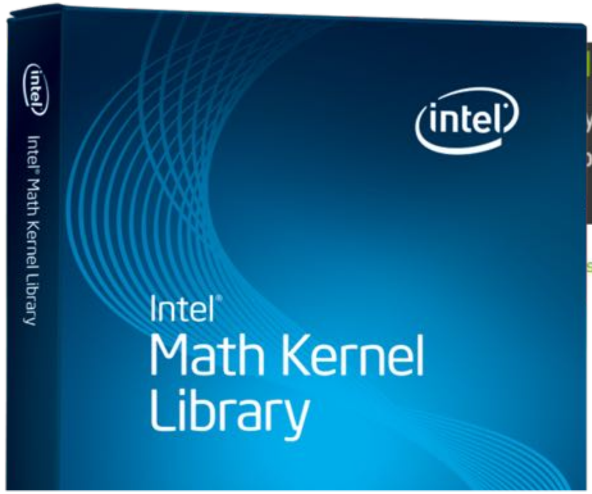
Conclusions

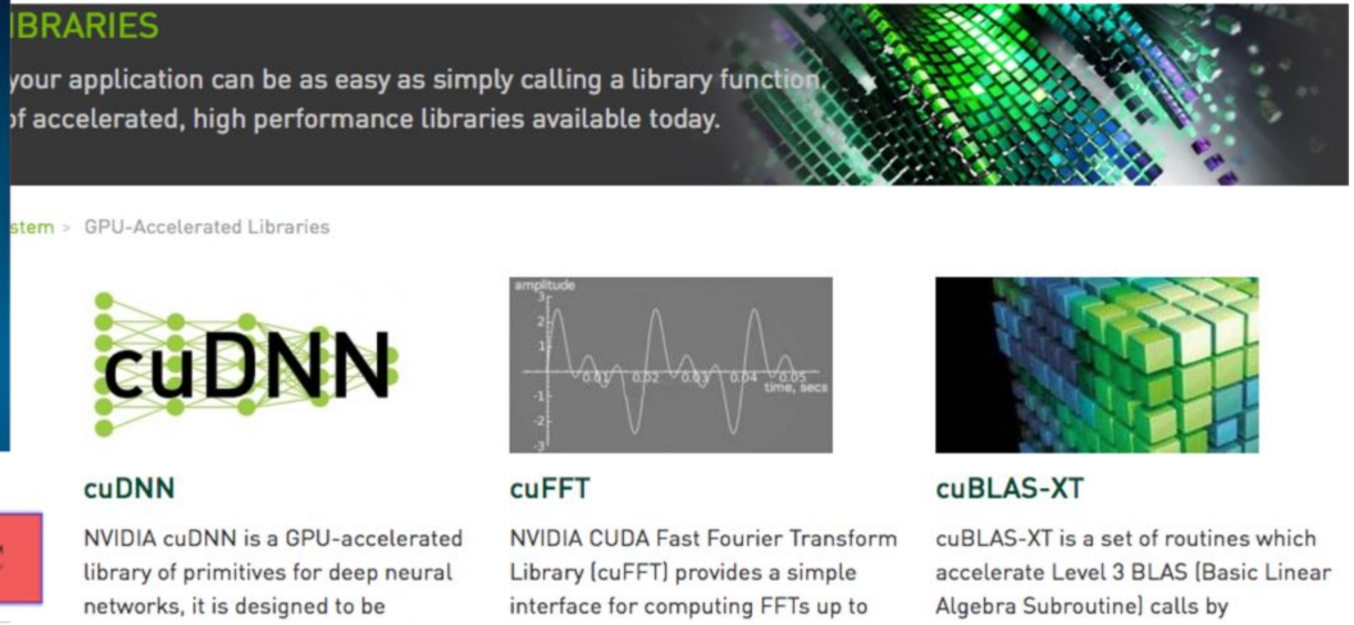
The efficient exploitation of current heterogeneous HPC solutions require good understanding of HW and SW features (architectures, instructions sets, sdk, …)

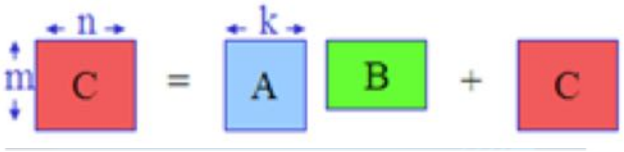
- Not only HW
- Skilled developers
- State-of-the-art software libraries and programming tools.

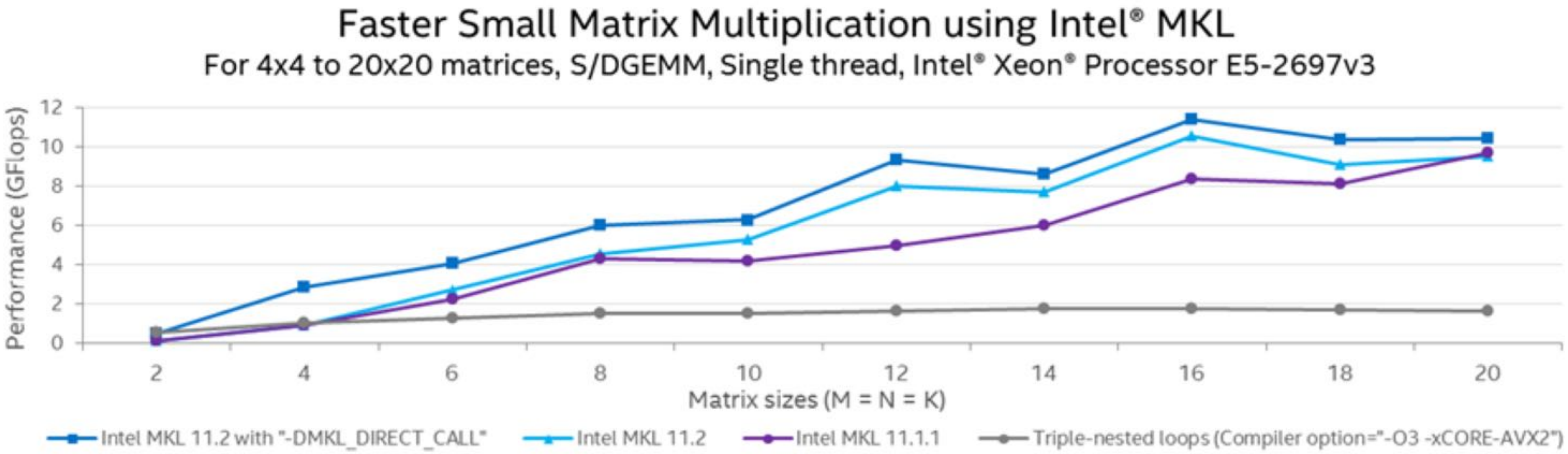
Good tools and developers are worth the money

For example use available, optimized libraries







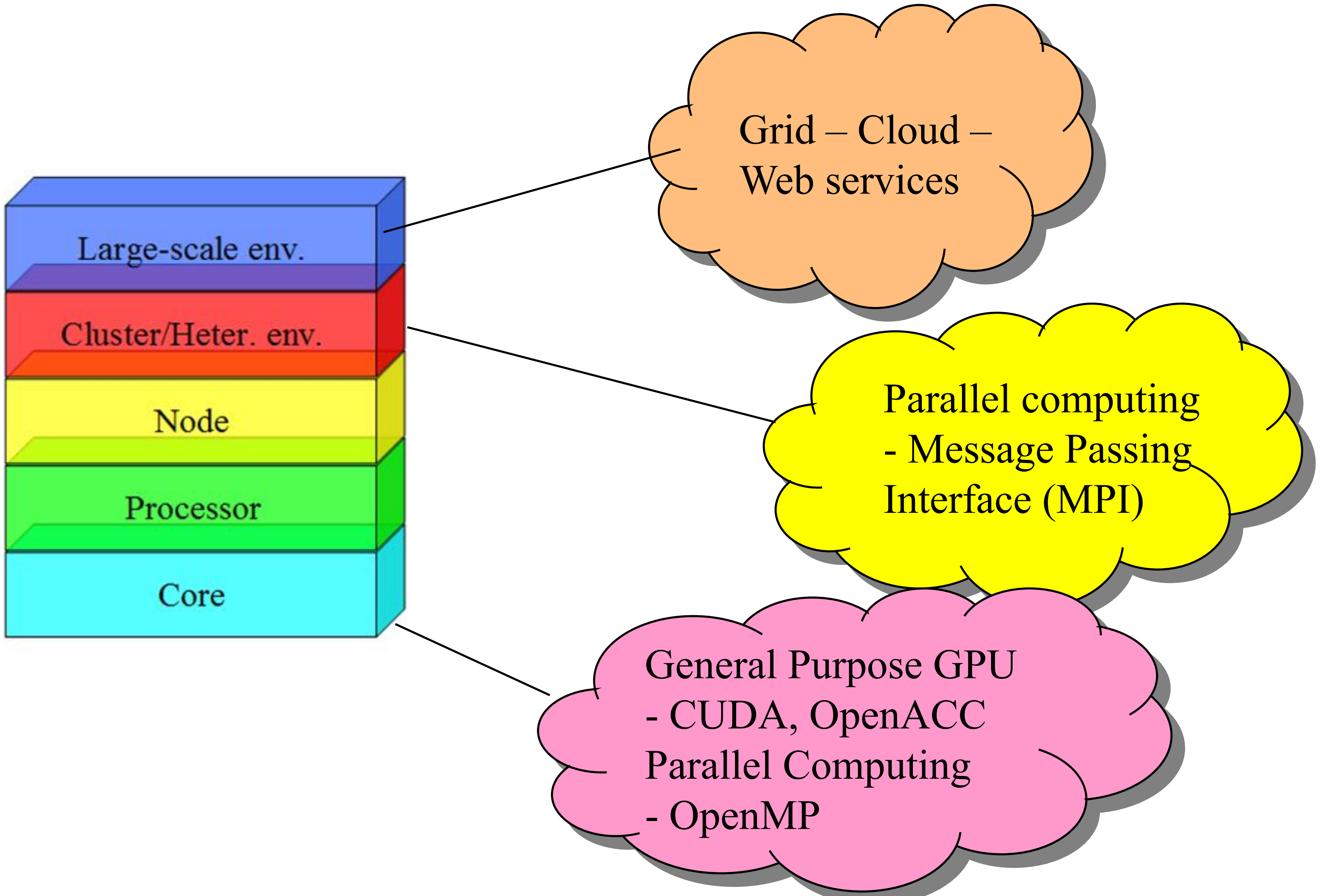


Part II - Computing Infrastructures

At first

- **Before selecting the infrastructure and the computing model chose *carefully* the algorithm/software used to solve your research challenge**
- **Cloud, Grid, Supercomputers, Artificial Intelligent will NOT save you if you are using the wrong tool/approach/strategy**
  - Study literature
    - Rule of thumb: if you created it from scratch probably is a non-optimal/costly solution
  - If possible use open source
  - If possible avoid vendor lock-in
  - Use standard or de-facto standard
  - Think in advance to the evolution of the application when complexity of the problem increase
    - Try to find solutions to scalability issues in advance

Computing Infrastructures



Clusters and Supercomputers

- The best solution for medium-size, limited time projects

SCAI SuperComputing Applications and Innovation

CINECA, through the **Italian SuperComputing Resource Allocation - ISCRA**, releases Call for Proposals.

CINECA, the Italian most powerful HPC center, twice a year **will directly award in excess of 300 millions core processor hours**, to ensure an adequate supply to scientists and engineers for HPC-related research.

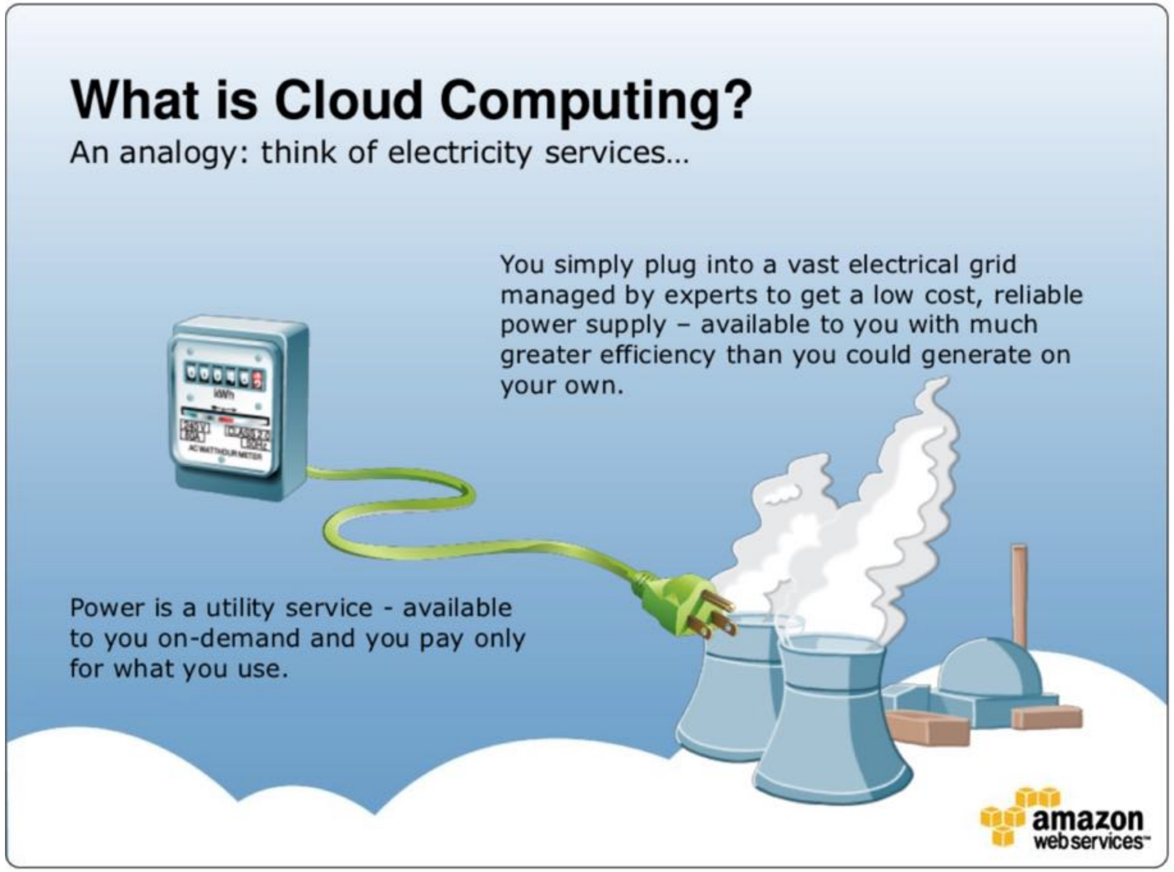
CINECA infrastructure offers different HPC resources to its users. The available resources are divided in three categories:

- The TIER-0, top level computing resources, which is the **MARCONI A2 (KNL)** machine and can be accessed trough class B and C projects: Class B between 200 Thousand and **10 Million** and Class C up to 400 Thousand total CPU hours.
- The TIER-1 level, which is **GALILEO2 (BROADWELL)** and can be accessed trough class B and C projects: Class B up to 2 Million and Class C up to 200 Thousand total CPU hours. For the class C project is temporarely unavailable.
- The Big Data resource (**PICO**), which is available for data analysis, visualization, post-processing, bio-informatics application and can be accessed trough class C projects: up to 50 Thousand core hours

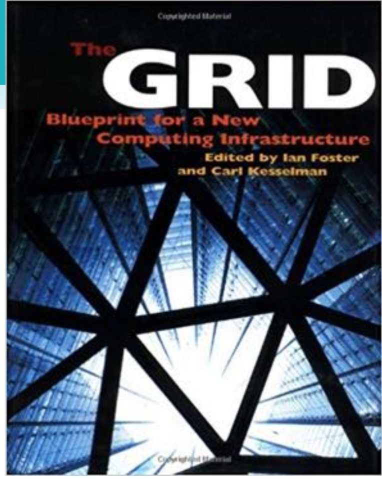
Class B projects are received twice a year. They go under peer-review evaulation and a 3 month delay is expected before your project gets access to HPC resources. For each user it is allowed to have only one class B project each 6 months as Project Investigator.

Class C projects are received through continuous submission and reviewed once per month. An average period of about 15 days is required for activating the project. For each user it is allowed to have only one class C project each 6 months as Project Investigator.

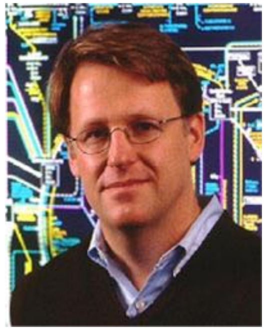
Otherwise: Cloud computing



https://www.slideshare.net/AmazonWebServices/introduction-to-amazon-web-services-7708257



Once upon a time: the GRID


Ian Foster

I.Foster, C.Kesselman: The Grid: Blueprint for a New Computing Infrastructure", 1998


Carl Kesselman

A computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities. In particular it:

- Coordinates resources that are not subject to centralized control
- Uses standard, open, general-purpose protocols and interfaces
- Delivers nontrivial qualities of service

The single batch Grid Job

JOB Type	JobType = “Normal";
Prologue	Prologue = "prologue.sh";
Input SandBox	InputSandbox = {"test.sh", "fileA"};
Requirements	Requirements = false;
Executable	Executable = "test.sh";
Std Output/Error	StdOutput = "std.out"; StdError = "std.err";
Output SandBox	OutputSandbox={"std.out", "std.err"};
Epilogue	Epilogue = "compress.sh";
Error Recovery	RetryCount = 1; ShallowRetryCount = 2;

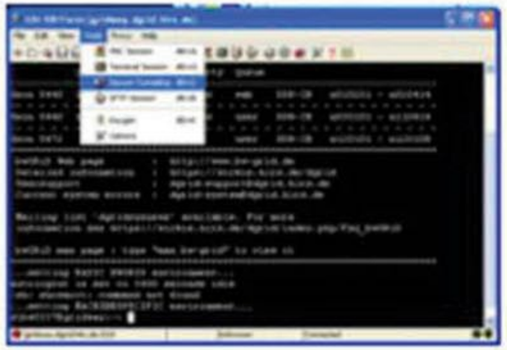
Grid Users

- **Virtual Organizations: a set of individuals and/or institutions that share resources under certain rules**
  - *Sharing is highly controlled,* resource providers and consumers define clearly and carefully what is shared, who is allowed to share, and the conditions under which sharing occurs
- I submit my jobs to "the GRID" and they get processed: somehow, somewhere, after some time if nothing occurs (e.g. failure due to misconfigured nodes).

```
====================== glite-wms-job-submit Success ======================

The job has been successfully submitted to the WMProxy

Your job identifier is:

https://lb-server-03.cnaf.infn.it:9000/C-Et5jbMMBjjUHkT1X6wVg

==========================================================================
```

Grid computing

Command Line Interface



Graphical User Interface

Grid Portal

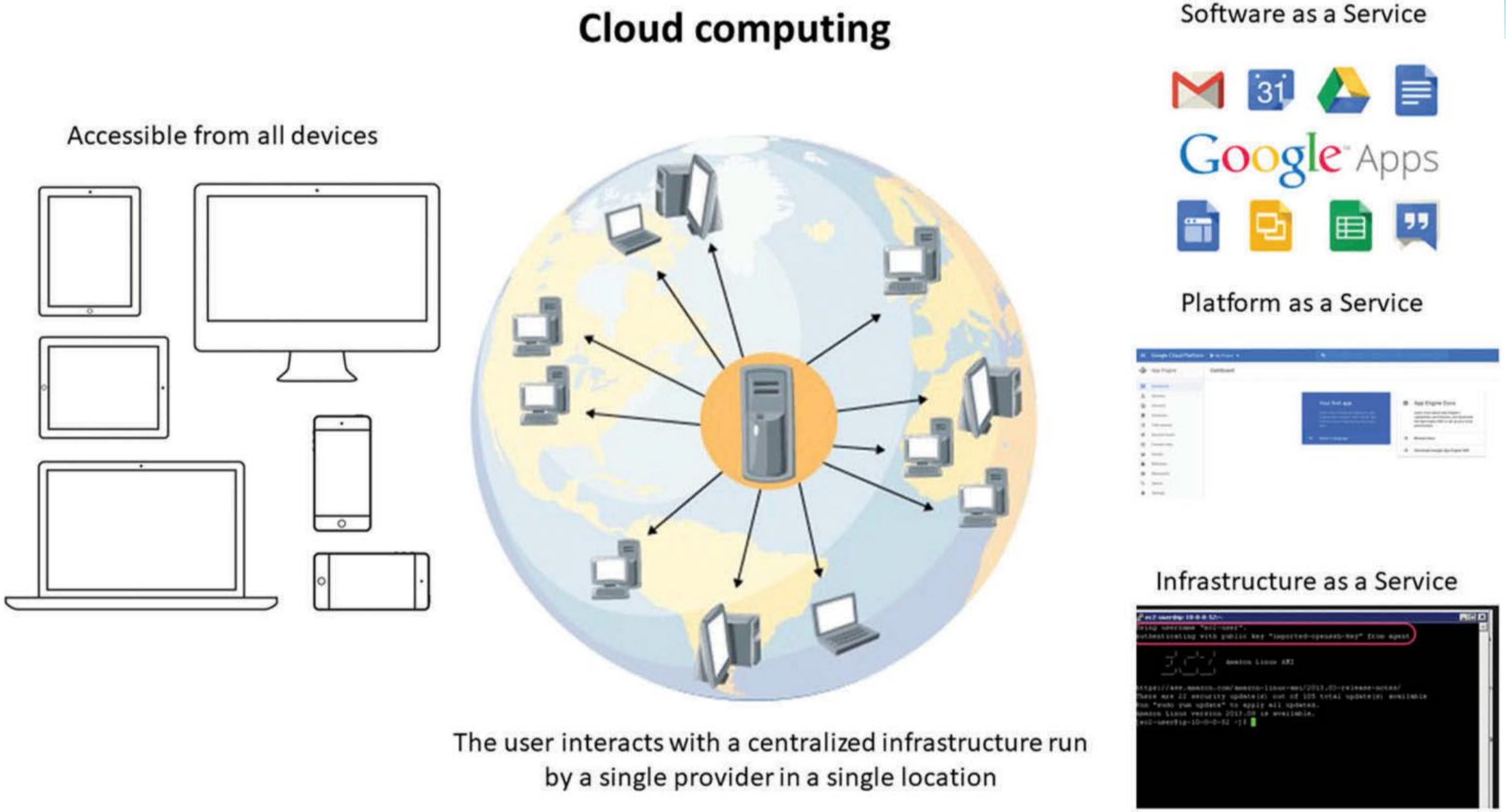
The user connects with a federated infrastructure to use geographically distributed resources for running jobs

Grid Services

Grid Platform

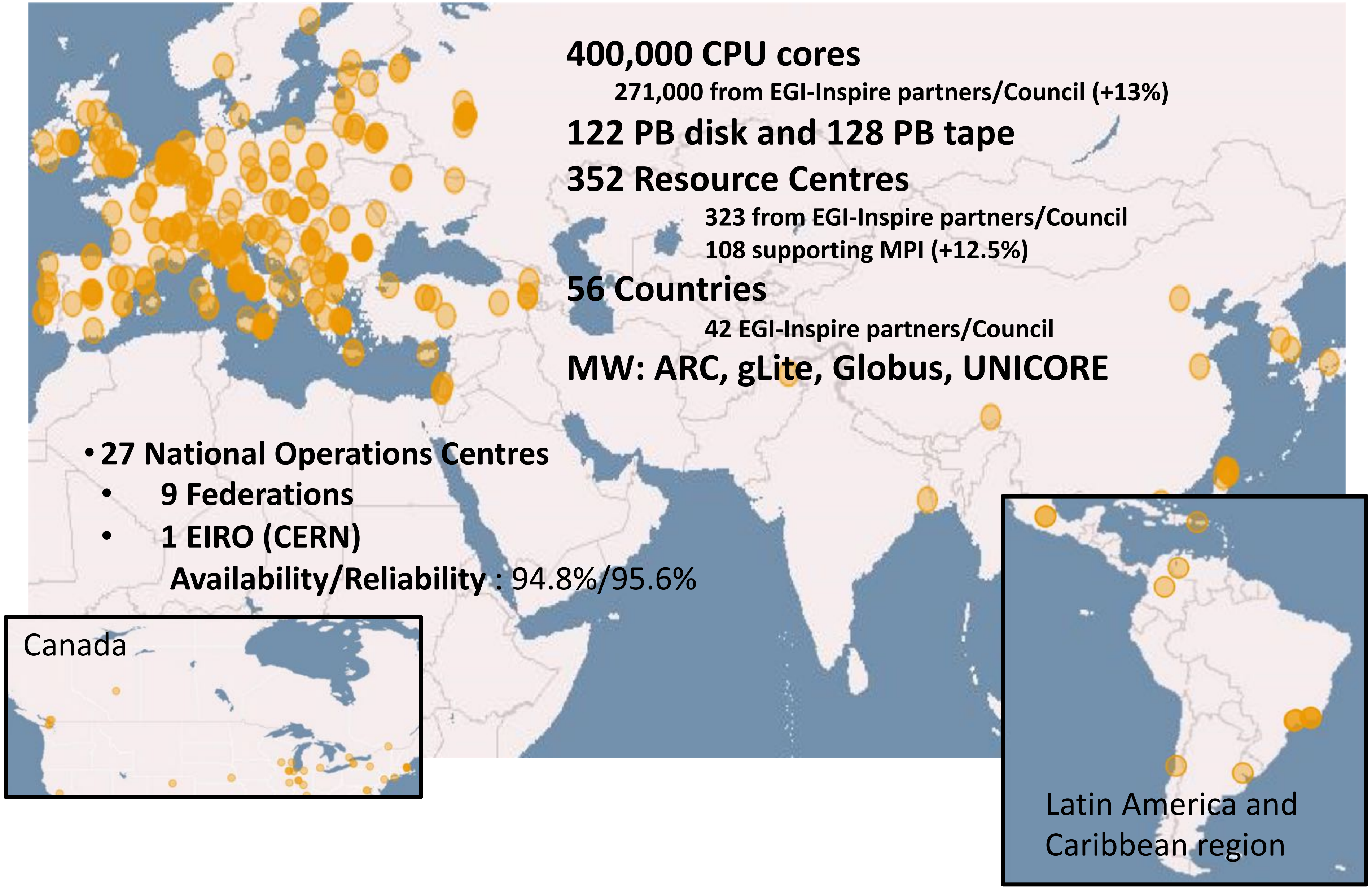
EGI Advanced Computing for Research

Volunteer computing



- Cloud computing is a distributed technology **more flexible and usable** than other distributed systems (e.g. Grid computing)
- In every case there is no Cloud, it's just someone else's computer

EGI – formerly The European Grid Initiative



400,000 CPU cores

271,000 from EGI-Inspire partners/Council (+13%)

122 PB disk and 128 PB tape

352 Resource Centres

323 from EGI-Inspire partners/Council

108 supporting MPI (+12.5%)

56 Countries

42 EGI-Inspire partners/Council

MW: ARC, gLite, Globus, UNICORE

- **27 National Operations Centres**
  - **9 Federations**
  - **1 EIRO (CERN)**

Availability/Reliability : 94.8%/95.6%

Science gateways

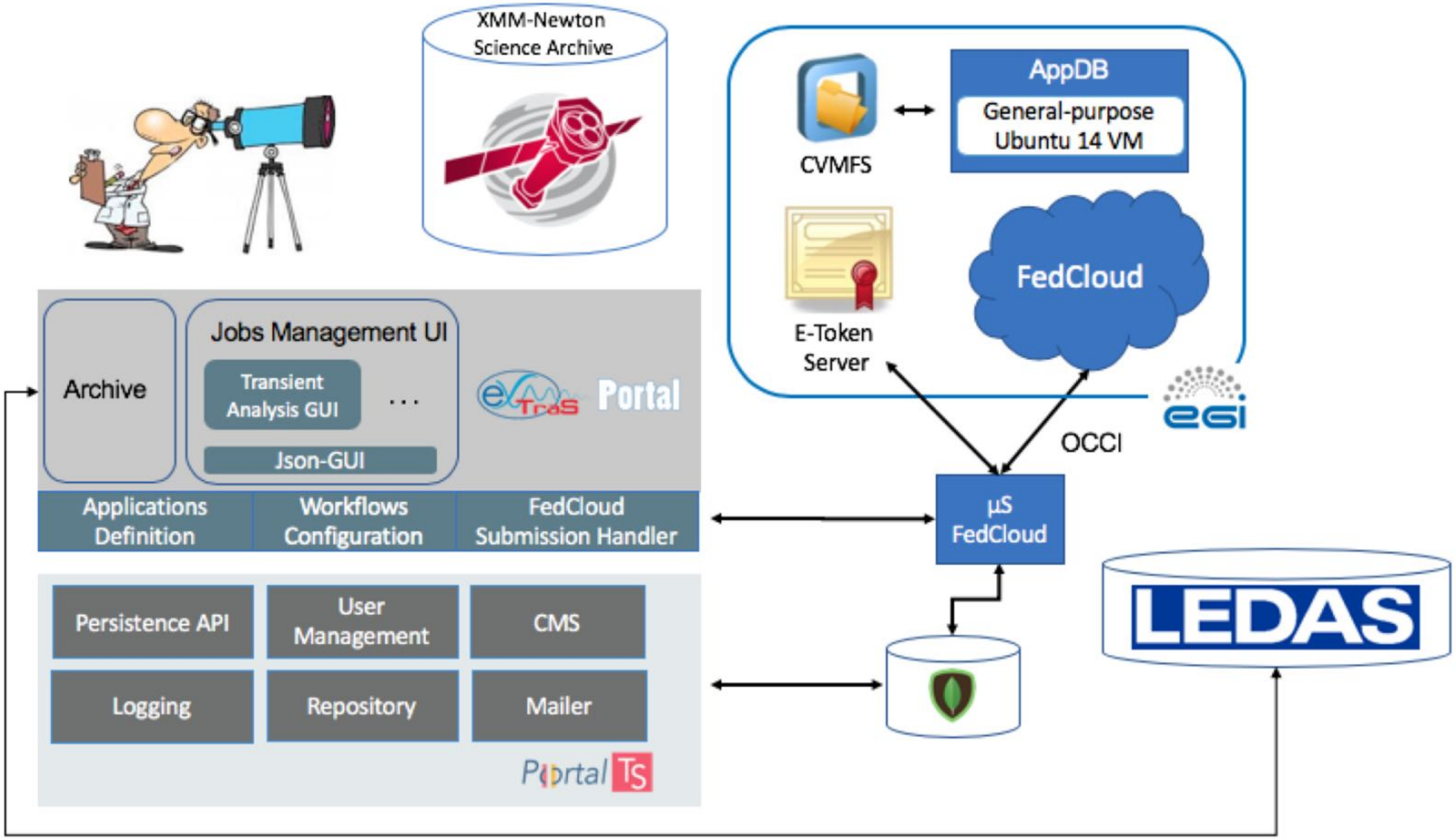
A science gateway

- is a community-specific set of tools, applications, and data collections
- that are integrated together via a web portal or a desktop application,
- providing access to resources and services from any provider, e.g. EGI or XSEDE.
- These gateways can support a variety of community-specific capabilities including workflows, virtualisation of software and hardware, visualization, resource discovery, job execution, access to data collections, repositories, applications and tools.
- A science gateway enables community members to define and perform custom research scenarios. Science gateways can hide the complexities of distributed infrastructures from researchers, therefore several communities build or would like to build science gateways for their members.

Science gateways

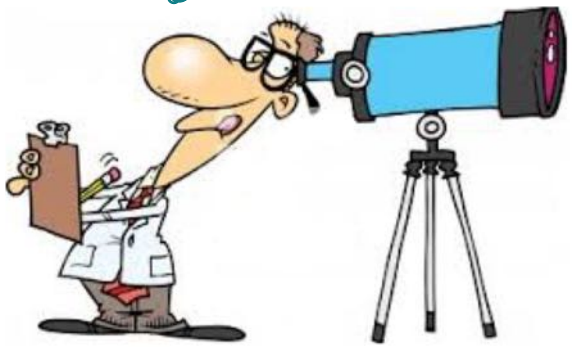
- Very important in USA
  - https://www.xsede.org/ecosystem/science-gateways
  - https://www.xsede.org/web/site/ecosystem/science-gateways/gateways-listing
  - https://sciencegateways.org
- But considered also in EU
  - https://www.egi.eu/use-cases/research-infrastructures/

The EXTraS Science Gateway



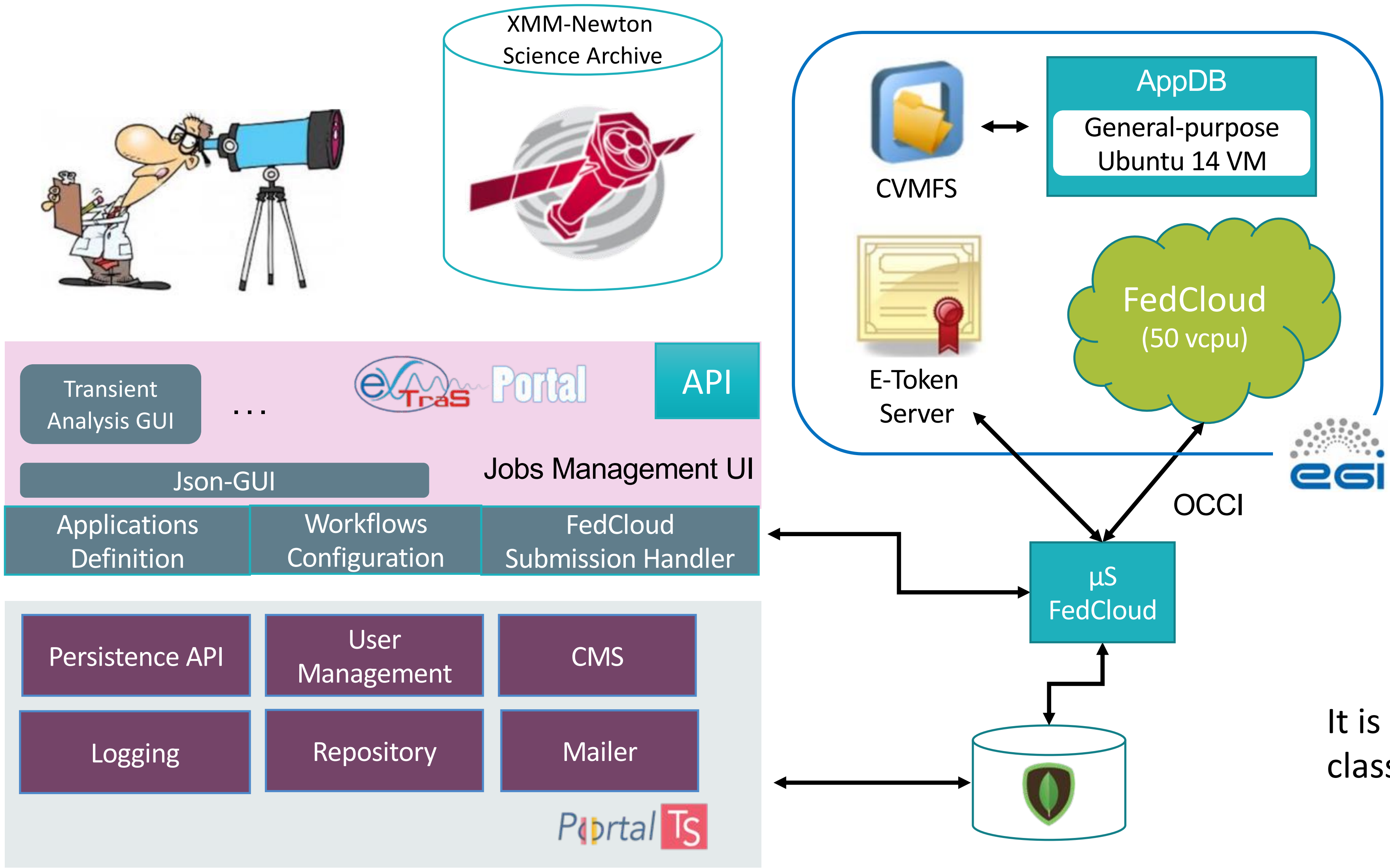
The EXTraS Portal

Nice. And now I want to analyse the last observation….

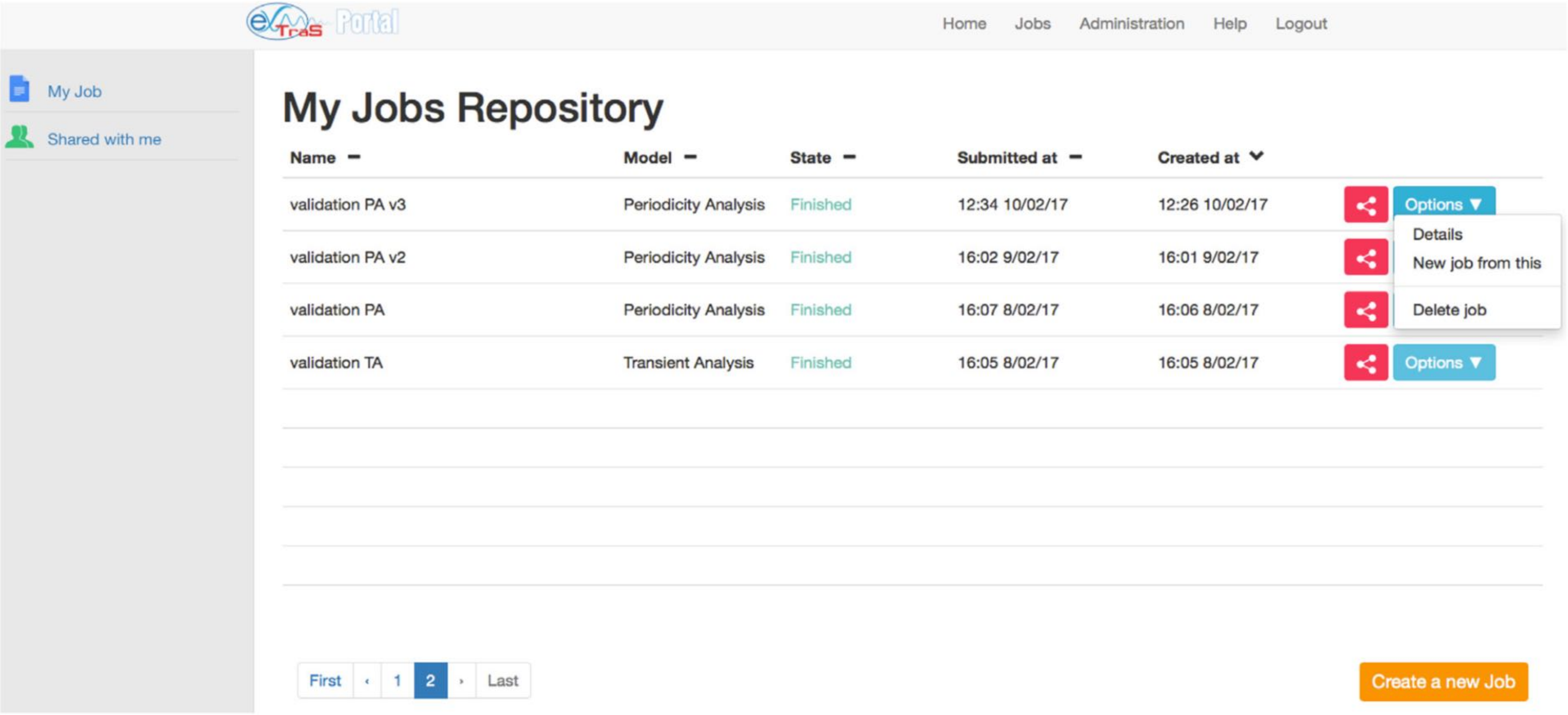


Select the observation
Configure the analysis
Submit and monitor it
Download the results

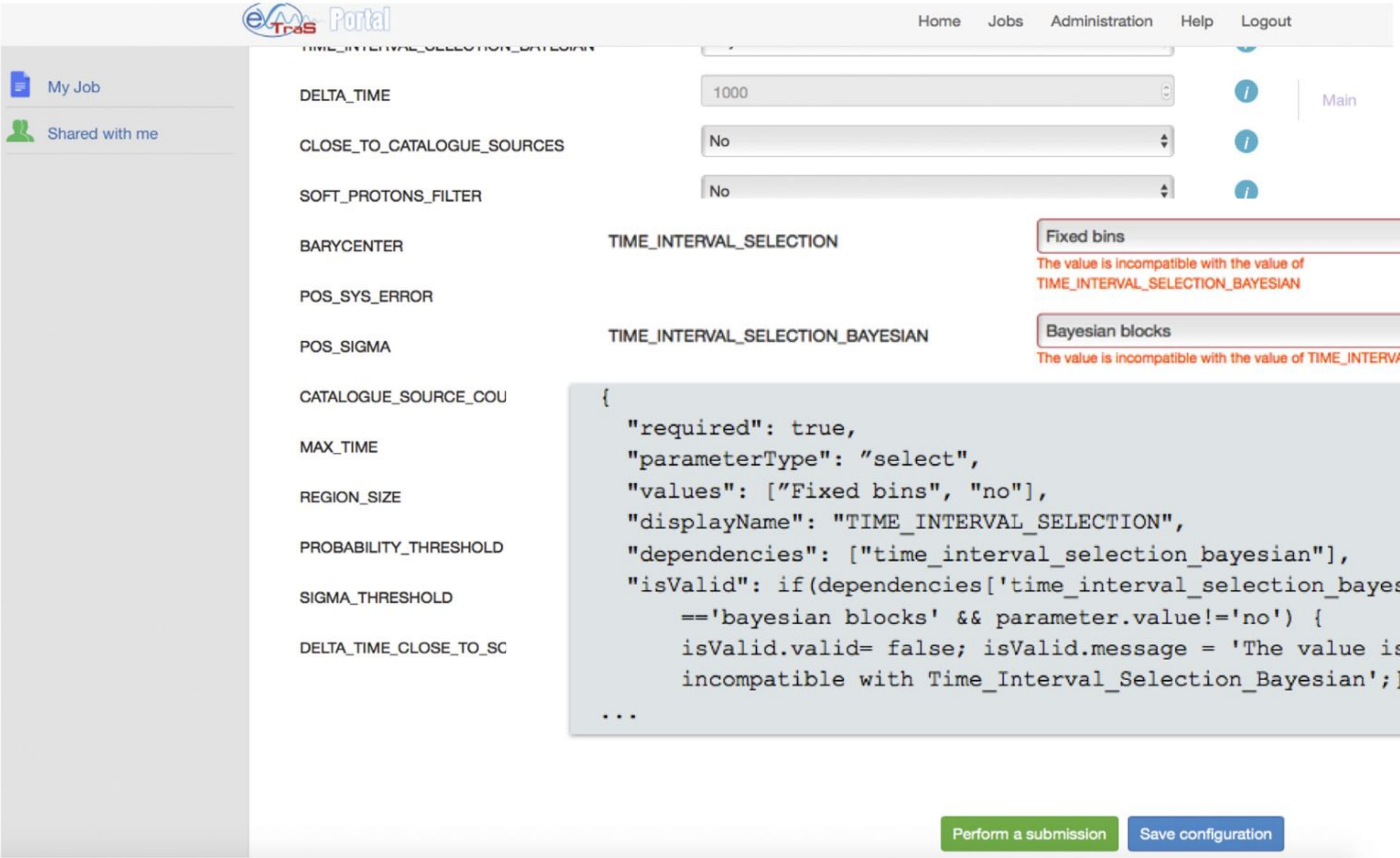
The architecture



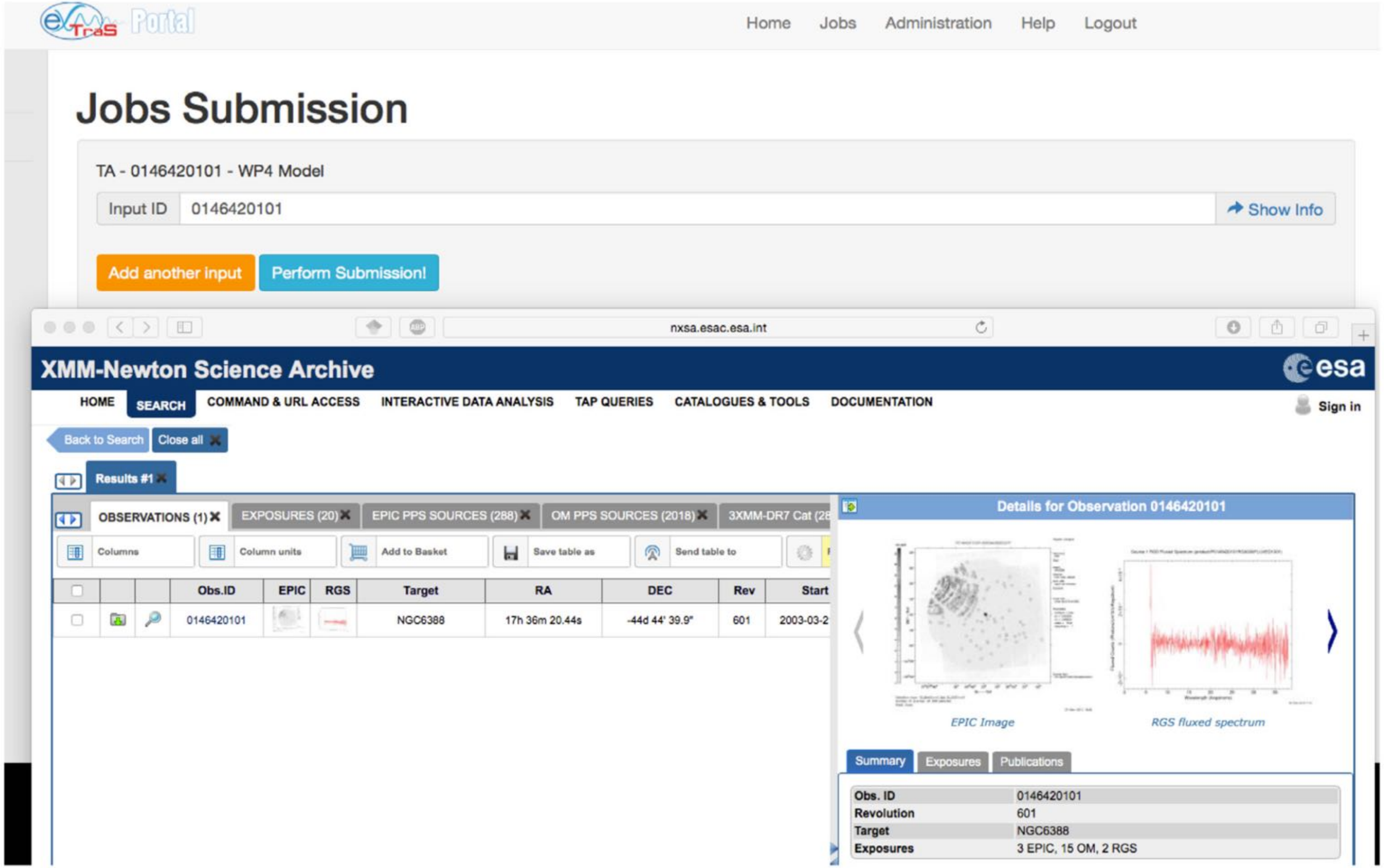
It is possible to classify it as SaaS

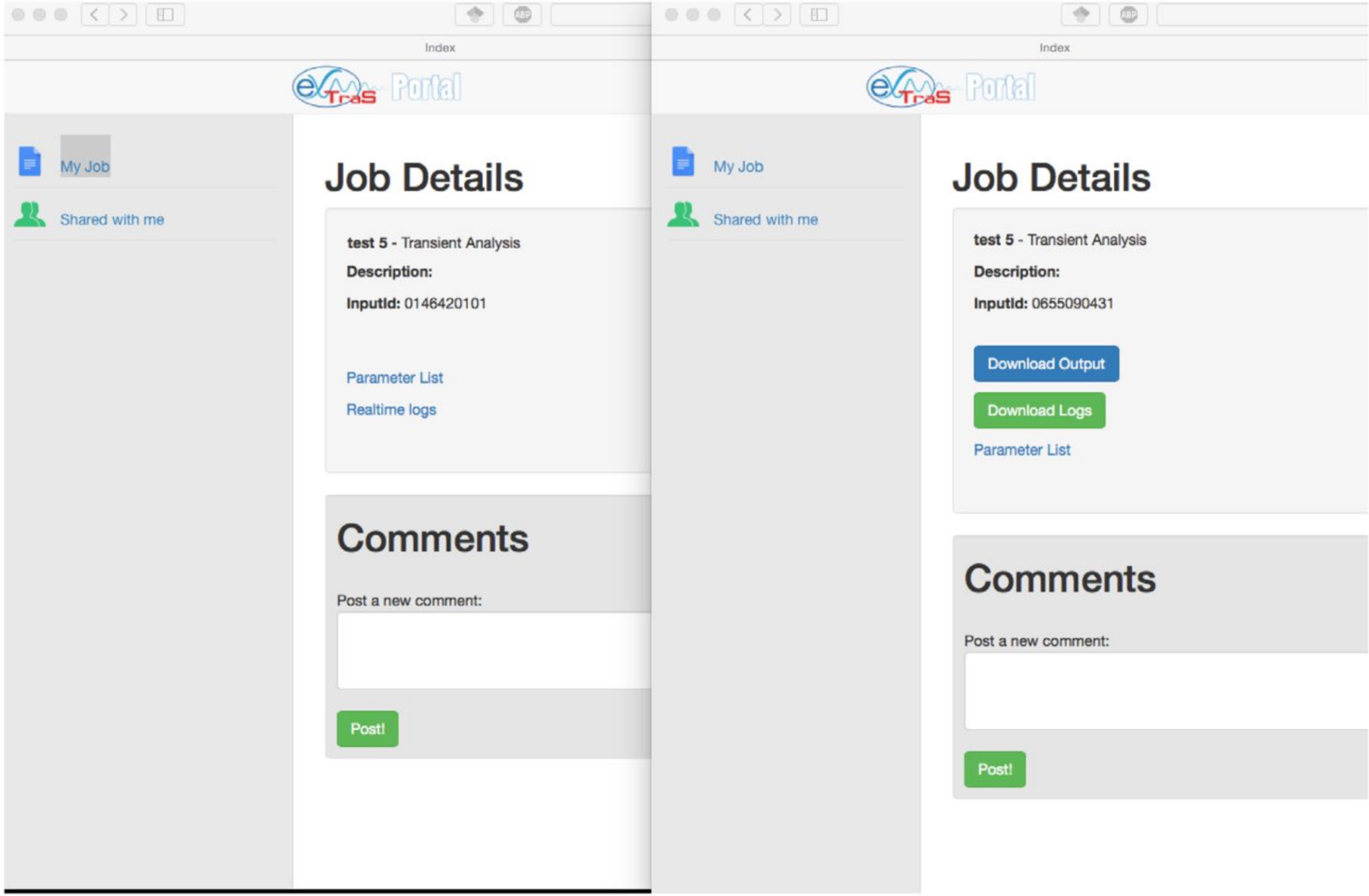


eXTraS Portal
Home
Jobs
Administration
Help
Logout
My Job
Shared with me
My Jobs Repository
Name
Model
State
Submitted at
Created at
validation PA v3
Periodicity Analysis
Finished
12:34 10/02/17
12:26 10/02/17
Options
Details
New job from this
Delete job
validation PA v2
Periodicity Analysis
Finished
16:02 9/02/17
16:01 9/02/17
validation PA
Periodicity Analysis
Finished
16:07 8/02/17
16:06 8/02/17
validation TA
Transient Analysis
Finished
16:05 8/02/17
16:05 8/02/17
Options
First
1
2
Last
Create a new Job




```
{
  "required": true,
  "parameterType": "select",
  "values": ["Fixed bins", "no"],
  "displayName": "TIME_INTERVAL_SELECTION",
  "dependencies": ["time_interval_selection_bayesian"],
  "isValid": if(dependencies['time_interval_selection_bayesian'].value
      =='bayesian blocks' && parameter.value!='no') {
      isValid.valid= false; isValid.message = 'The value is
      incompatible with Time_Interval_Selection_Bayesian';}"}
...
```


Portal
Home
Jobs
Administration
Help
Logout
Jobs Submission
TA - 0146420101 - WP4 Model
Input ID
0146420101
Show Info
Add another input
Perform Submission!
nxsa.esac.esa.int
XMM-Newton Science Archive
esa
HOME
SEARCH
COMMAND & URL ACCESS
INTERACTIVE DATA ANALYSIS
TAP QUERIES
CATALOGUES & TOOLS
DOCUMENTATION
Sign in
Back to Search
Close all
Results #1
OBSERVATIONS (1)
EXPOSURES (20)
EPIC PPS SOURCES (288)
OM PPS SOURCES (2018)
Columns
Column units
Add to Basket
Save table as
Send table to
Obs.ID
EPIC
RGS
Target
RA
DEC
Rev
Start
0146420101
NGC6388
17h 36m 20.44s
-44d 44' 39.9"
601
Details for Observation 0146420101
EPIC Image
RGS fluxed spectrum
Summary
Exposures
Publications
Obs. ID
0146420101
Revolution
601
Target
NGC6388
Exposures
3 EPIC, 15 OM, 2 RGS


Index
Portal
My Job
Shared with me
Job Details
test 5 - Transient Analysis
Description:
InputId: 0146420101
Parameter List
Realtime logs
Comments
Post a new comment:
Post!
Index
Portal
My Job
Shared with me
Job Details
test 5 - Transient Analysis
Description:
InputId: 0655090431
Download Output
Download Logs
Parameter List
Comments
Post a new comment:
Post!

Science gateways

- ‘Citizen science’ (CS) = online collaboration between scientists and members of the public who volunteer to take part in research; the ‘crowdsourcing’ of scientific research
- www.zooniverse.org : an umbrella site containing over 90 projects in different scientific domains

The major challenge of 21st century research is dealing with the flood of information we can now collect about the world around us. Computers can help, but in many fields the human ability for pattern recognition — and our ability to be surprised — makes us superior. With the help of Zooniverse volunteers, researchers can analyze their information more quickly and accurately than would otherwise be possible, saving time and resources, advancing the ability of computers to do the same tasks, and leading to faster progress and understanding of the world, getting to exciting results more quickly.

- A science gateway represents also a state-of-the-art accessible online tools to engage fellow citizens in the era of petabyte-scale astronomy.

Part III - Data Management

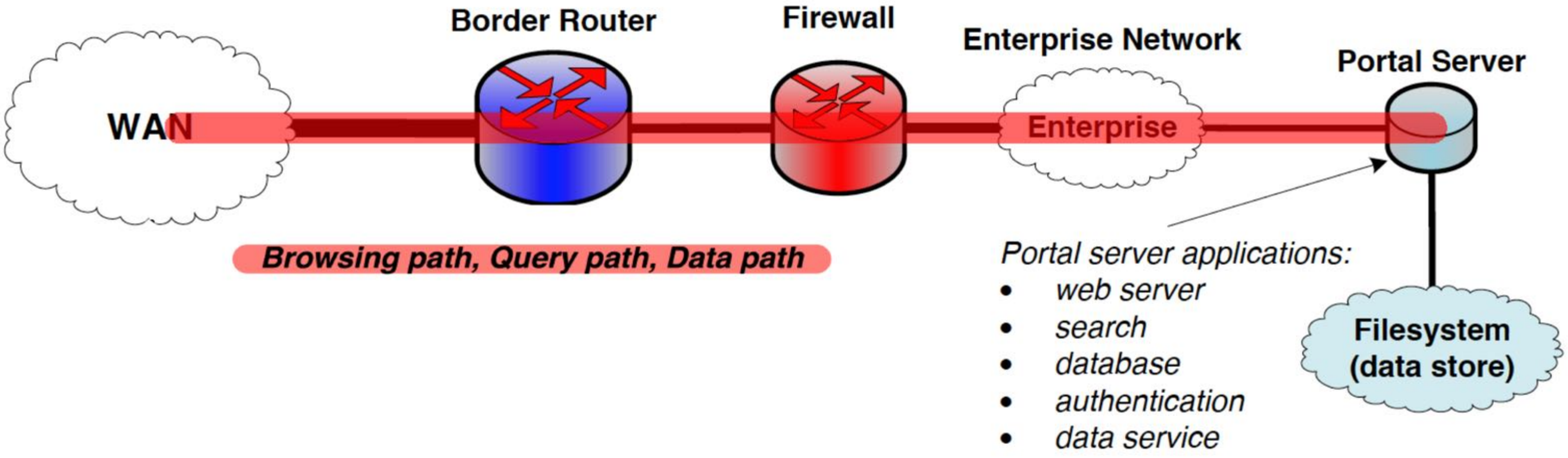
Data Storage

Data Centers should adopt industry standards for data storage when possible, see also Section 5.2. Perhaps the most obvious challenge is simply storing the data. The large volumes mean efficiency in storage representation is important.

We recommend that data centers leverage 'off-the-shelf', open data management services, tools, and technologies that have been developed by industry. Moving to industry standards for things like images allow us to leverage new technologies such as the ability to stream and operate remotely on *objects* using standard tools. File systems as we know them will not be the most appropriate storage model at petascale levels. Alternatives include the use of cloud object stores, cloud compute, 'big data native' formats such as Apache parquet and OpenEXR, and cloud-optimized FITS (see cloud optimized GeoTIFF as an example https://www.cogeo.org). Traditional astronomy file formats (e.g. FITS) should be used as they were originally intended, for transport only. That being said, one big advantage of FITS files is their ability to co-package meta-data, while e.g. for Parquet there are only limited options to have meta-data included directly with the data. Data and meta-data should be managed together to not lose efficiency in analysis performance.

The legacy research data portal architecture

Most systems used to exchange research data today are not so different from the first web servers. In particular, a single server handles request processing, data access, authentication, and other functions. It is the simple and monolithic architecture.
They works but performance and management issues…



Modern research data portals – An example

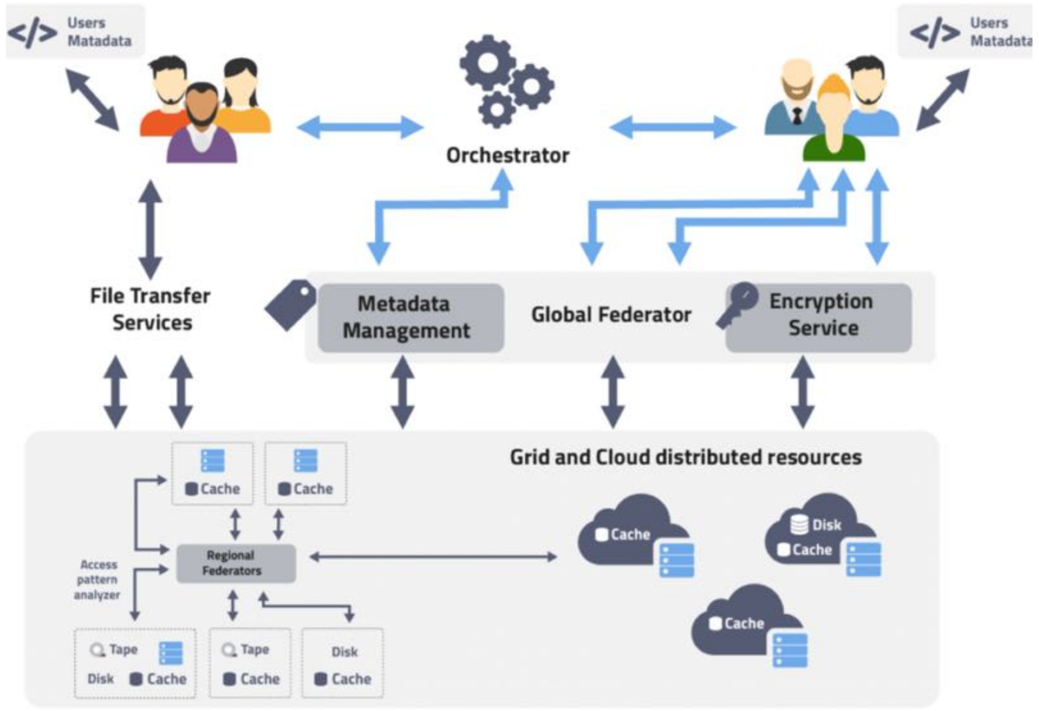
The Horizon2020 eXtreme DataCloud – XDC project aims at developing scalable technologies for federating storage resources and managing data in highly distributed computing environments, as required by the most demanding, data intensive research experiments in Europe and worldwide.

Very high-energy electromagnetic radiation reaches Earth from a large part of the Cosmos, carrying crucial and unique information about the most energetic phenomena in the Universe.

CTA (Cherenkov Telescope Array) will answer many of the persisting questions by enabling the detection of more than thousands sources over the whole sky. CTA builds on the proven technique of detecting gamma-ray induced particle cascades in the atmosphere through their Cherenkov radiation, simultaneously imaging each cascade stereoscopically with multiple telescope and reconstructing the properties of the primary gamma ray from those images.

Through deployment of about 50–100 telescopes per site at two sites in the southern (Chile) and the northern hemispheres (Canary Islands), CTA will achieve full-sky coverage.

CTA is interested in the following functionalities implemented by XDC:



The FAIR Guiding Principles for scientific data management and stewardship

To be Findable:
F1. (meta)data are assigned a globally unique and persistent identifier
F2. data are described with rich metadata (defined by R1 below)
F3. metadata clearly and explicitly include the identifier of the data it describes
F4. (meta)data are registered or indexed in a searchable resource

To be Accessible:
A1. (meta)data are retrievable by their identifier using a standardized communications protocol
A1.1 the protocol is open, free, and universally implementable
A1.2 the protocol allows for an authentication and authorization procedure, where necessary
A2. metadata are accessible, even when the data are no longer available

To be Interoperable:
I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (meta)data use vocabularies that follow FAIR principles
I3. (meta)data include qualified references to other (meta)data

To be Reusable:
R1. meta(data) are richly described with a plurality of accurate and relevant attributes
R1.1. (meta)data are released with a clear and accessible data usage license
R1.2. (meta)data are associated with detailed provenance → Reproducibility
R1.3. (meta)data meet domain-relevant community standards

Conclusions

Most of the presented concepts have been exploited in the EXTraS SG

- EXTraS portal
  - ✓ User-friendly interface
  - ✓ Extensible to other Computing Infrastructures
  - ✓ Can include other analysis modules
  - ✓ Supports the results sharing (with an ID for results)
  - ✓ Supports a basic collaboration among the users
  - ✓ Useful as prototype for Citizen Science
  - ✗ Effective remote visualization service
  - ✗ Full Citizen Science support
- Data archive
  - ✓ Allows to share the products
  - ✗ Not FAIR
  - ✗ Extensibility…