# ARIS High Performance Computing Infrastructure, access policy, tools and usage.

Dr. Dimitris Dellis

GRNET

Thessaloniki, 11 Dec. 2017

# GRNET in 1’

**Internet Provider for Greek Universities and Research Centers**

- 87 POPs
- Connection to GEANT
- GR-IX (Greek Internet Exchange)
- Computation
  - Grid (HellasGrid)
  - Cloud (Okeanos)
  - HPC

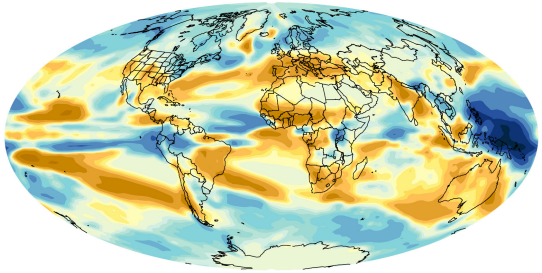
# High Performance Computing

- HPC means use of a high scalability system to solve cpu demanding problems
- Implies parallel Processing
- Computation : The $3^{rd}$ pillar of science, together with theory and experiment.
- Safety, Flexibility, Accuracy, Economy, Development time.

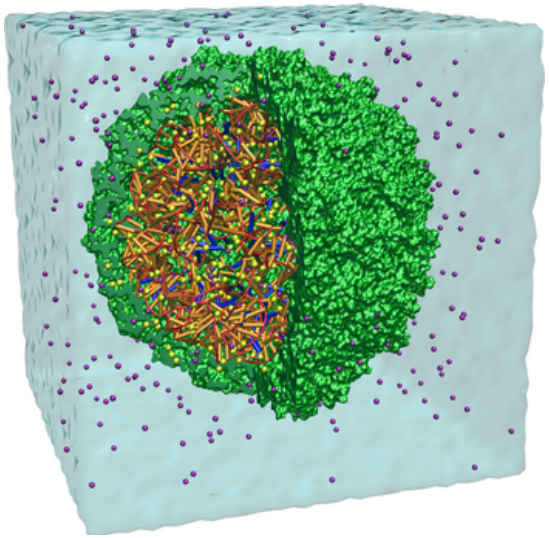
Today to out-compete is to out-compute

# Scientific Fields in HPC

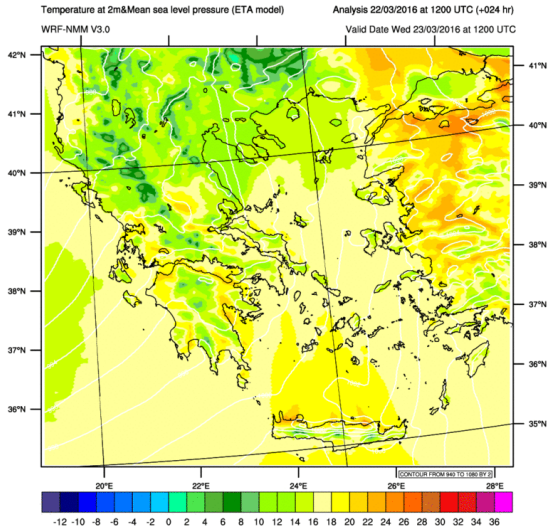
## Climatology



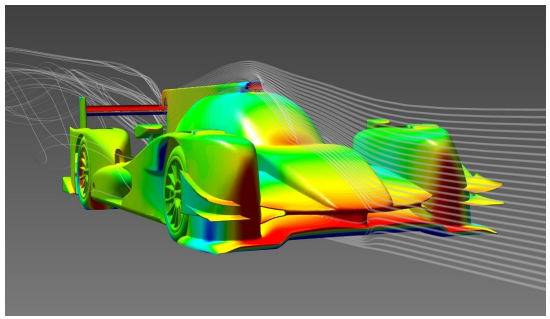
## Life Sciences



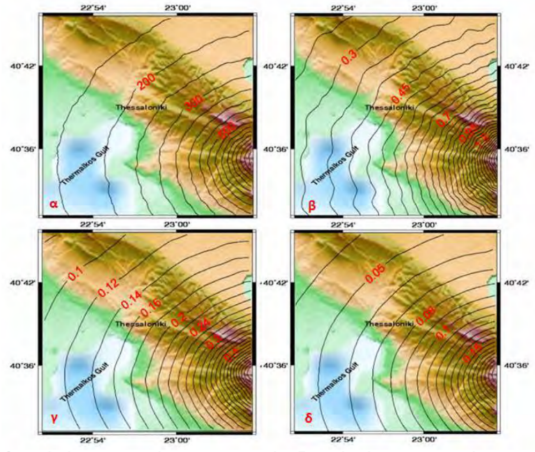
## Meteorology



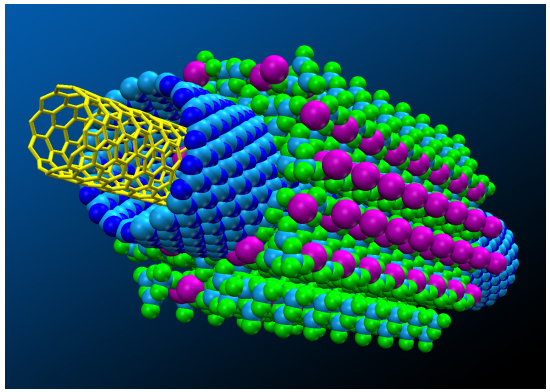

## Engineering/Fluids



## Seismology



## Materials



and much more.

# Funding : Phase I

- PRACE-GR : ”Ανάπτυξη Εθνικής Υπερυπολογιστικής Υποδομής και Παροχή Συναφών Υπηρεσιών στην Ελληνική Ερευνητική και Ακαδημαϊκή Κοινότητα - MIS 379417”
- ΠΕΠ «Αττική», ΑΞΟΝΑΣ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ 3: «Ενίσχυση της ανταγωνιστικότητας της καινοτομίας και της ψηφιακής σύγκλισης»
- Στόχοι
  - Ανάπτυξη υπερυπολογιστικής υποδομής στην Ελλάδα για την πραγματοποίηση Έρευνας υψηλού επιπέδου και Ισχυροποίηση του ρόλου της Ελλάδας στον τομέα των Υπερυπολογιστών σε Πανευρωπαϊκό επίπεδο.
  - Εκμετάλλευση από μεγάλο εύρος επιστημονικών πεδίων.
  - Έμφαση στις εφαρμογές υψηλής κλιμάκωσης (μεγάλη παραλληλία). Χρήση μοντέλων προγραμματισμού MPI και OpenMP
  - Επεκτασιμότητα
  -

ΕΣΠΑ 2007-2013 πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΠΕΠ ΠΕΡΙΦΕΡΕΙΑΚΟ ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΑΤΤΙΚΗΣ πραγματική ανάπτυξη
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ Υπουργείο Παιδείας & Θρησκευμάτων

# Funding : Phase II

- «Παροχή ψηφιακών υπηρεσιών μέσω της δημιουργίας ενεργειακά αποδοτικού κέντρου δεδομένων» - MIS 311568 ΕΠ «Ψηφιακή Σύγκλιση», ΑΞΟΝΑΣ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ 1: «Βελτίωση της παραγωγικότητας με αξιοποίηση των ΤΠΕ»
- Επέκταση Κέντρου Δεδομένων ΕΔΕΤ στο Κτίριο του Υπουργείου Παιδείας στο Μαρούσι
- Δημιουργία Πράσινου Κέντρου Δεδομένων στο Λούρο
- Προμήθεια υπολογιστικού εξοπλισμού για παροχή υπηρεσιών υπολογιστικού νέφους
- Προμήθεια υπολογιστικού εξοπλισμού για εξειδικευμένες επιστημονικές εφαρμογές.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης

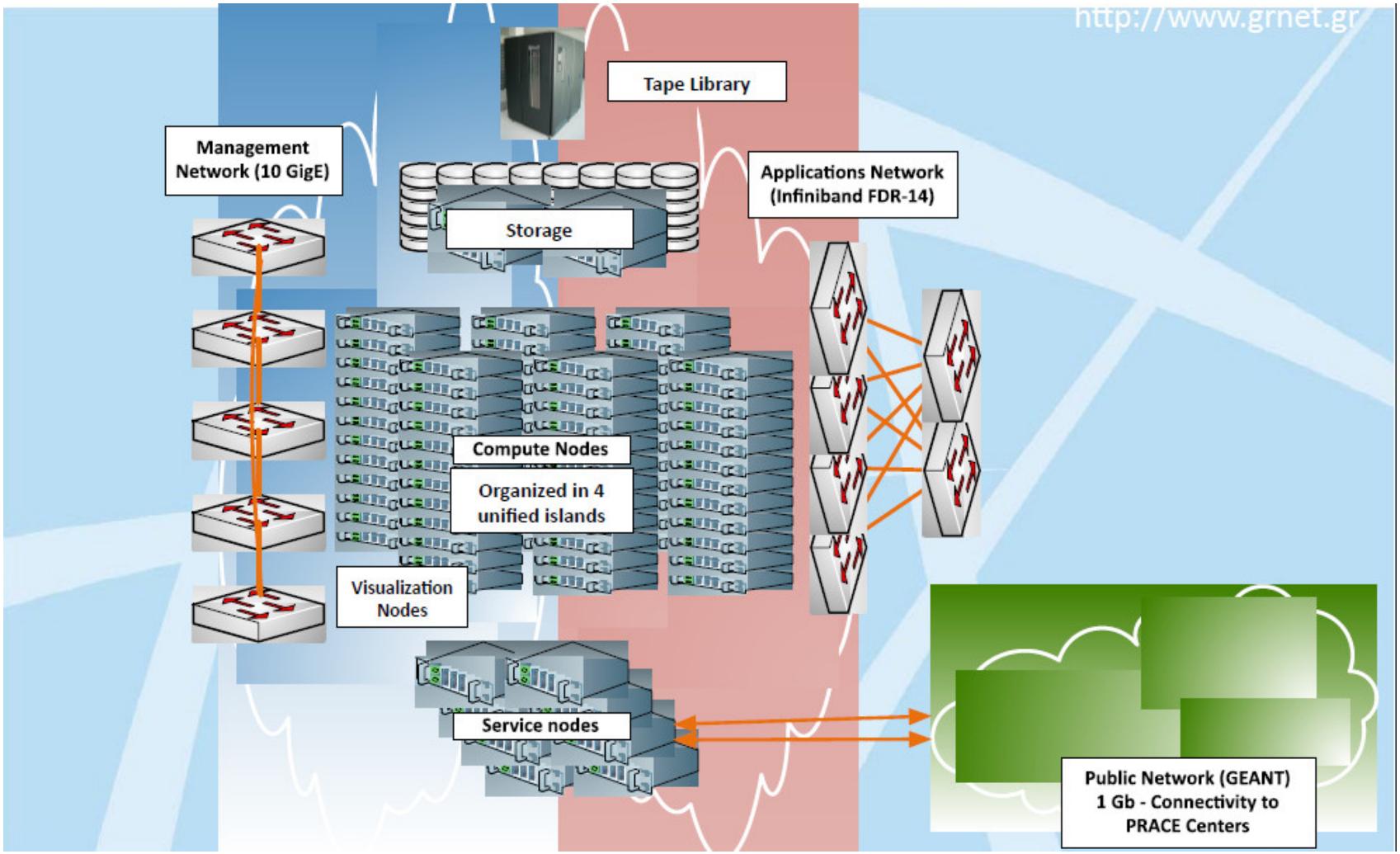
ψηφιακήελλάδα
Όλα είναι δυνατά
Επιχειρησιακό Πρόγραμμα "Ψηφιακή Σύγκλιση"

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
Ποιότητα ζωής για όλους

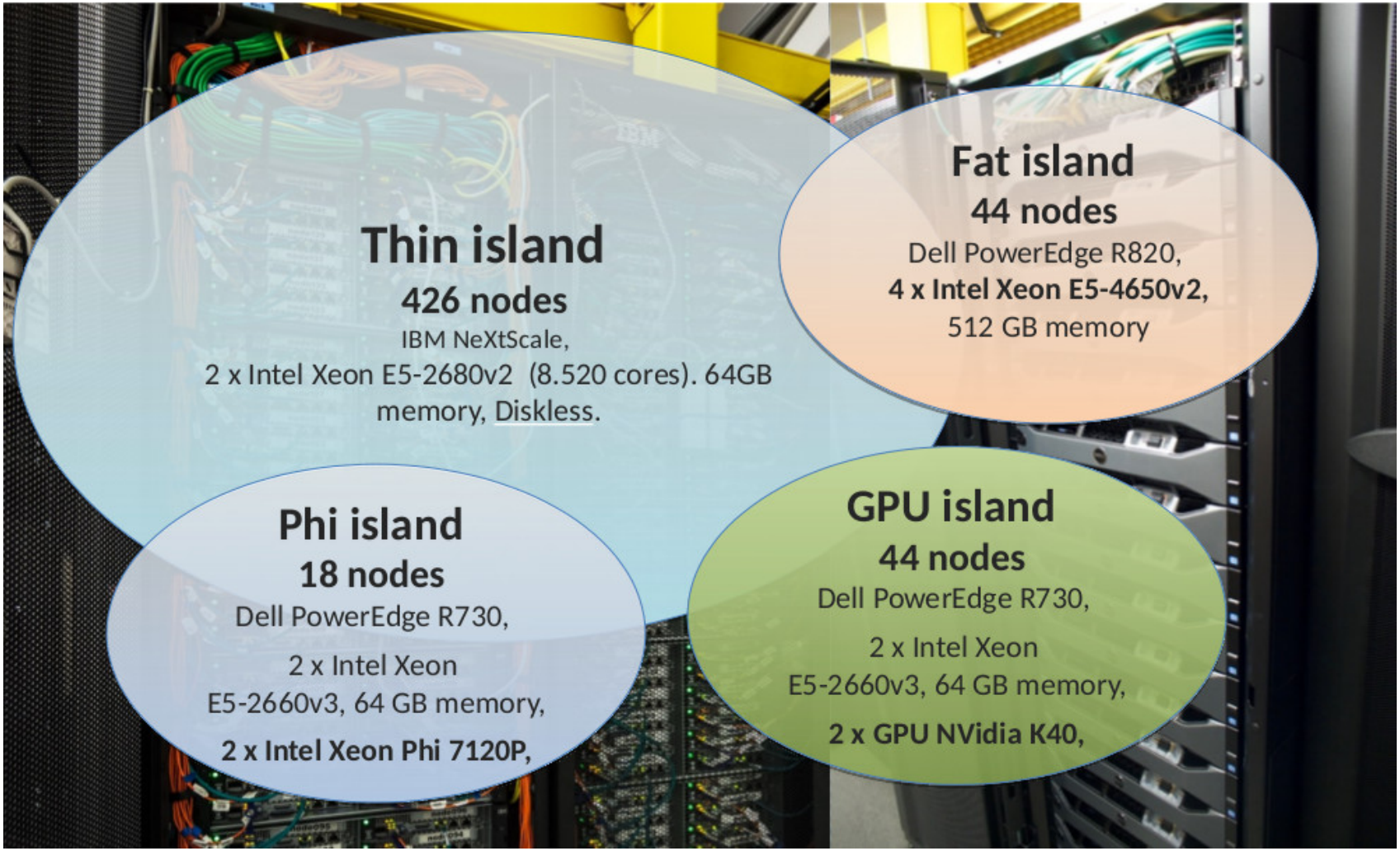
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

- και χτίστηκε το ARIS, σε 2 φάσεις.

# ARIS : System Organization



# Compute Nodes



# Service Nodes

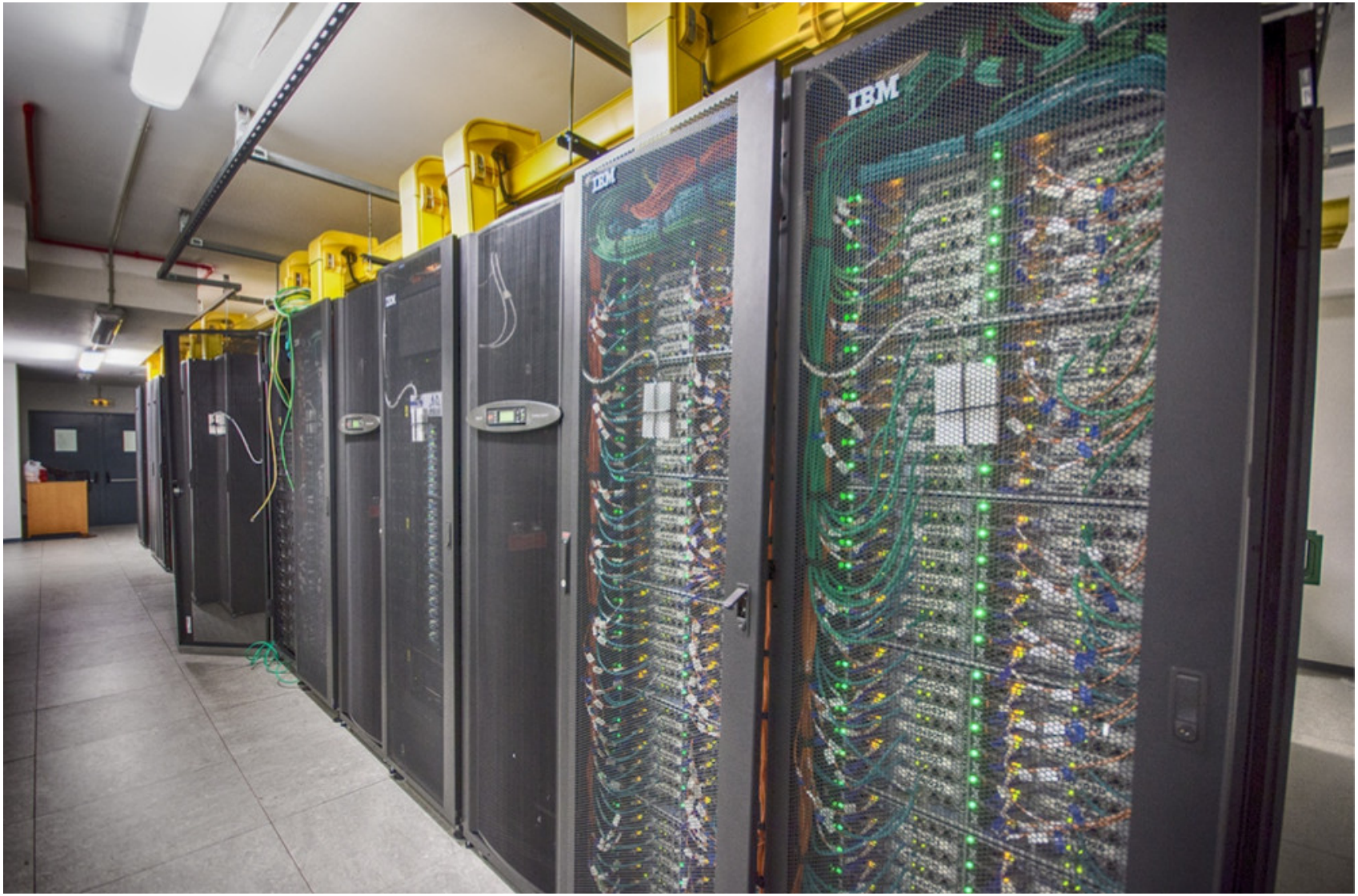
# Visualization Nodes

**2 x Dell Precision 7910**

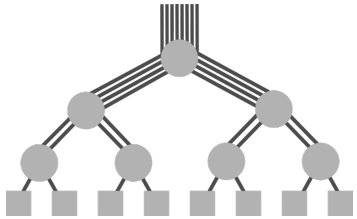
- 2 x Intel Xeon Processor E5-2630 v4 (10C, 2.2GHz)
- 256GB 2400MHz DDR4
- **2 x NVIDIA Quadro M5000 8GB**


- **Image/Video rendering of data**
- **Interactive Connection VNC/VirtualGL**
- **Access using batch system. login nodes as gateway**

# Compute Nodes



# Infiniband



- FDR14 : Full non blocking Fat Tree, 56 Gbits all to all

# Storage

- 2 Racks, raw capacity 2 PB, usable $\sim 1.5$ PB, GPFS.

# Cooling

# UPS

# Software

- RedHat Enterprise/CentOS x86-64 6.9
- Slurm 16.05.11
- Libraries/Applications Software organized with Environment Modules


CentOS

# Administration and Support Team

- Infrastructure Operation/Administration
- User support
- Application Support : Porting, Optimization, Profiling
- System Documentation
- Training

# System Performance

- Phase I : 179.83 TFlops, No 468 in 06/2015 Top500 List
  169.73 TFlops reported

TOP 500 The List.

CERTIFICATE

ARIS - IBM NeXtScale nx360M5, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR14

Greek Research and Technology Network, Greece

is ranked

No. 468

among the World's TOP500 Supercomputers

with 169.73 Tflop/s Linpack Performance

in the 45th TOP500 List published at ISC15 in Frankfurt, Germany, July 13th, 2015.

Congratulations from the TOP500 Editors

Erich Strohmaier
NERSC/Berkeley Lab

Jack Dongarra
University of Tennessee

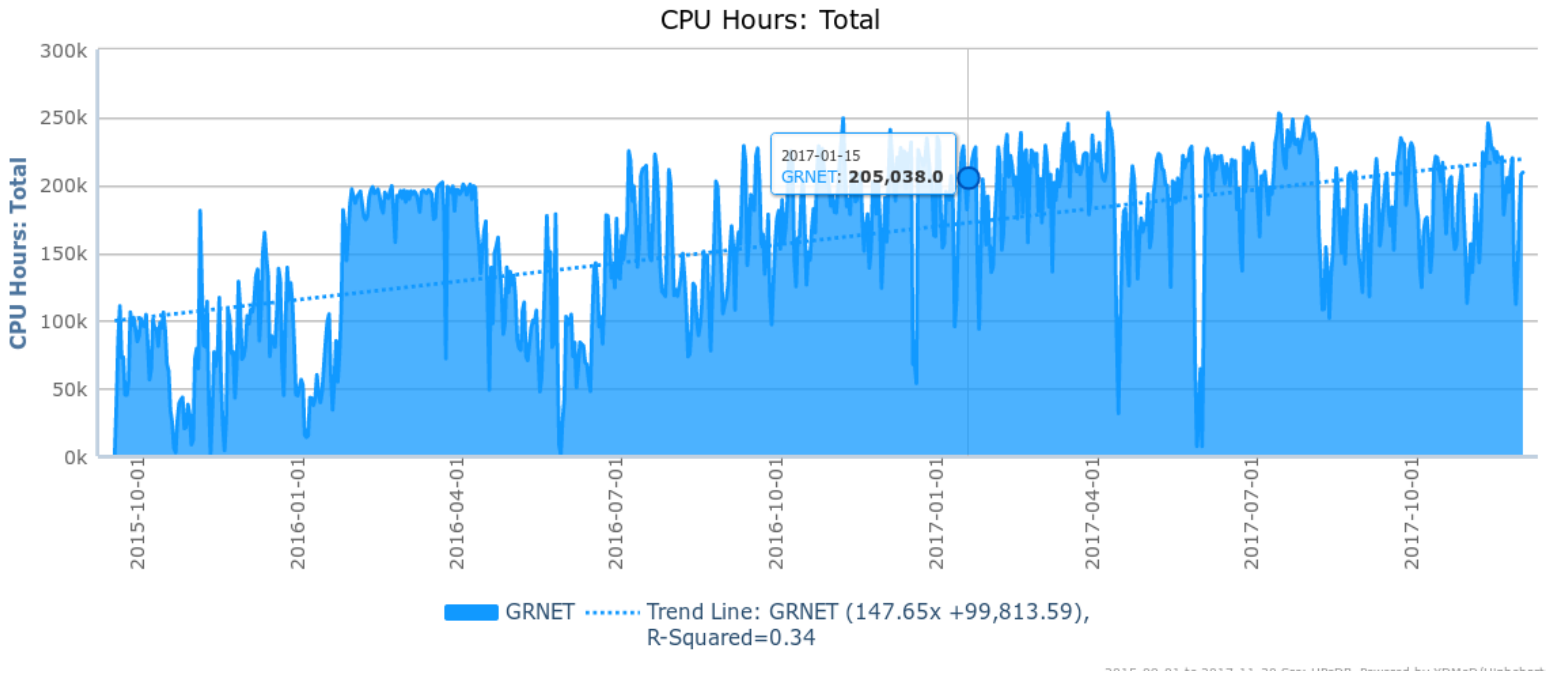
Horst Simon
NERSC/Berkeley Lab

Martin Meuer
Prometeus

- Phase II : Theoretical : 444 TFlops

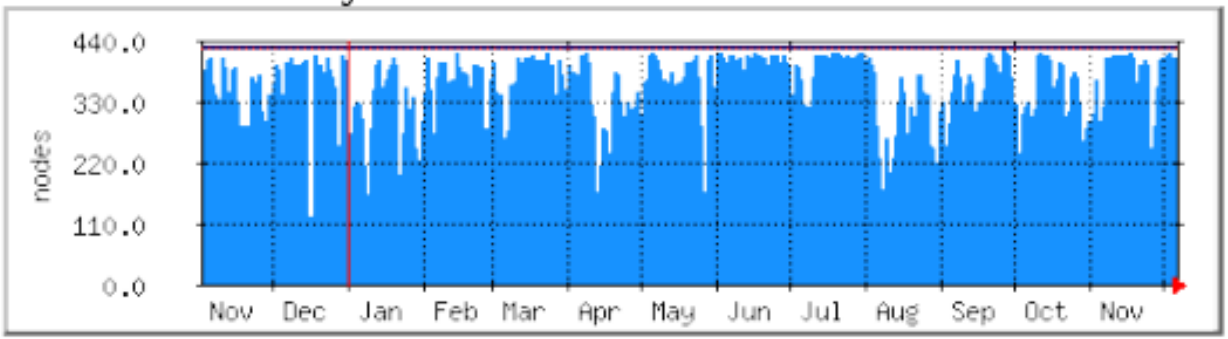
# System Usage

- System Capability :
  8520 / 11520 core Years / year before/afrer Aug 2016.
- Allocated up to now : $\sim$ 26,000 core Years
- In 4 Production calls, always open preparatory call, contribution in DECI, VI-SEEM, SoHPC, etc.
- $3^{rd}$ and $4^{th}$ production calls in progress, $5^{th}$ just closed.
- Consumed up to now : $\sim$ 15,000 core Years



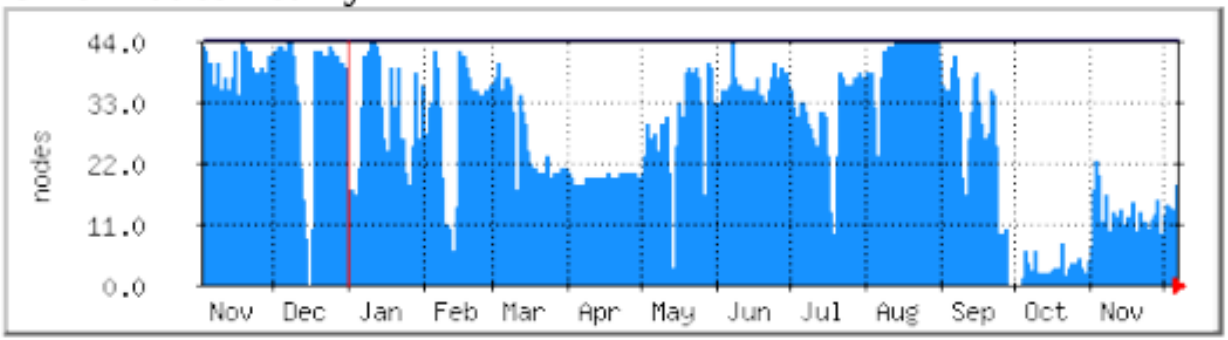
# System Usage

## Thin Nodes Yearly




| | Max | Average | Current |
|---|---|---|---|
| **Nodes Allocated** | 419 Nodes (98.4%) | 344 Nodes (80.8%) | 401 Nodes (94.1%) |
| **Total Nodes** | 426 Nodes (100.0%) | 426 Nodes (100.0%) | 426 Nodes (100.0%) |

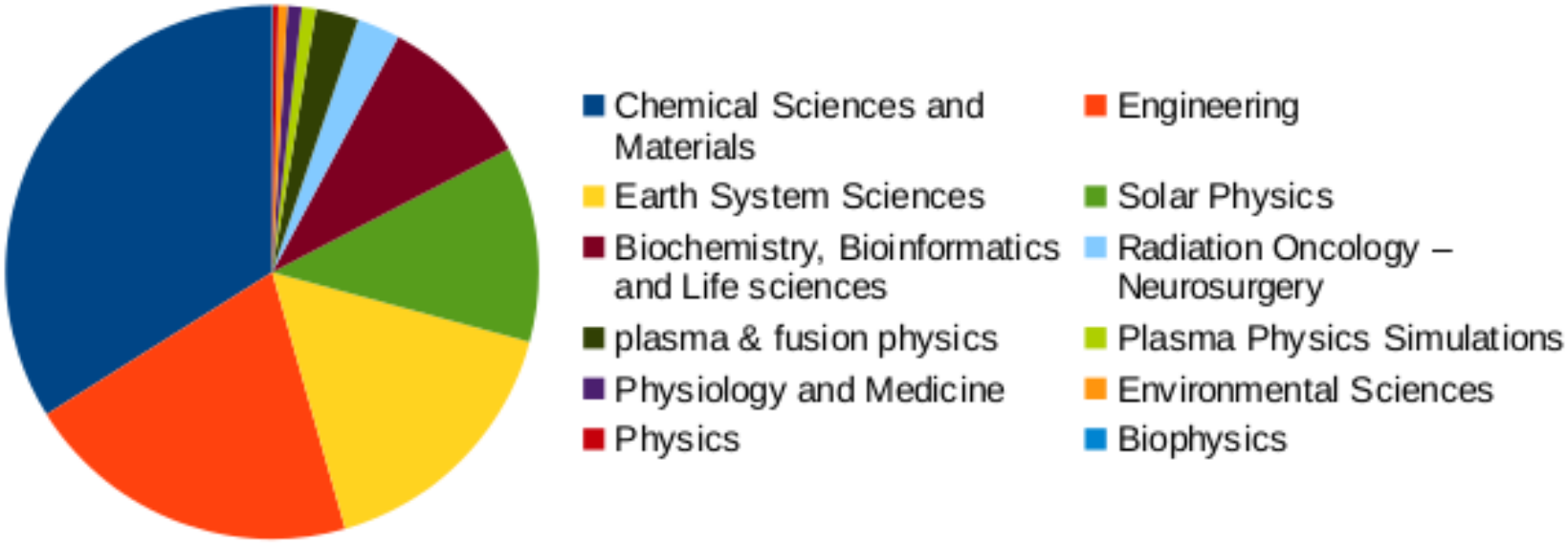
## GPU Nodes Yearly




| | Max | Average | Current |
|---|---|---|---|
| **Nodes Allocated** | 44 Nodes (100.0%) | 27 Nodes (61.4%) | 18 Nodes (40.9%) |
| **Total Nodes** | 44 Nodes (100.0%) | 44 Nodes (100.0%) | 44 Nodes (100.0%) |

# System Usage

$3^{rd}$ production call Scientific Fields allocations:

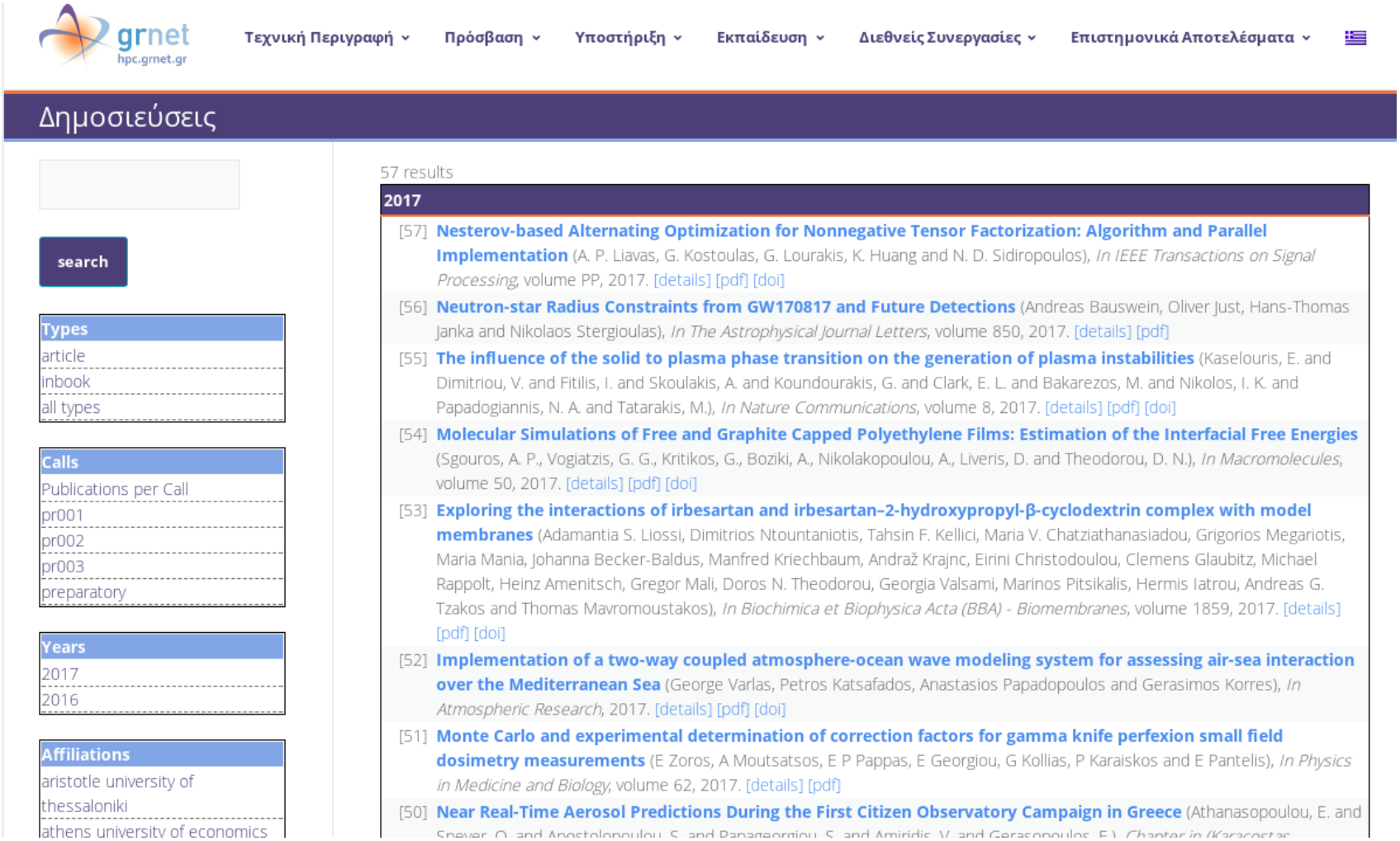


# System Usage

$3^{rd}$ production call Institutes allocation:


- Picture is usually different between calls : 12 months allocation, calls every 6 months.

# Publications



grnet hpc.grnet.gr

Τεχνική Περιγραφή | Πρόσβαση | Υποστήριξη | Εκπαίδευση | Διεθνείς Συνεργασίες | Επιστημονικά Αποτελέσματα

Δημοσιεύσεις

search

**Types**
article
inbook
all types

**Calls**
Publications per Call
pr001
pr002
pr003
preparatory

**Years**
2017
2016

**Affiliations**
aristotle university of thessaloniki
athens university of economics

57 results

**2017**

[57] **Nesterov-based Alternating Optimization for Nonnegative Tensor Factorization: Algorithm and Parallel Implementation** (A. P. Liavas, G. Kostoulas, G. Lourakis, K. Huang and N. D. Sidiropoulos), *In IEEE Transactions on Signal Processing*, volume PP, 2017. [details] [pdf] [doi]

[56] **Neutron-star Radius Constraints from GW170817 and Future Detections** (Andreas Bauswein, Oliver Just, Hans-Thomas Janka and Nikolaos Stergioulas), *In The Astrophysical Journal Letters*, volume 850, 2017. [details] [pdf]

[55] **The influence of the solid to plasma phase transition on the generation of plasma instabilities** (Kaselouris, E. and Dimitriou, V. and Fitilis, I. and Skoulakis, A. and Koundourakis, G. and Clark, E. L. and Bakarezos, M. and Nikolos, I. K. and Papadogiannis, N. A. and Tatarakis, M.), *In Nature Communications*, volume 8, 2017. [details] [pdf] [doi]

[54] **Molecular Simulations of Free and Graphite Capped Polyethylene Films: Estimation of the Interfacial Free Energies** (Sgouros, A. P., Vogiatzis, G. G., Kritikos, G., Boziki, A., Nikolakopoulou, A., Liveris, D. and Theodorou, D. N.), *In Macromolecules*, volume 50, 2017. [details] [pdf] [doi]

[53] **Exploring the interactions of irbesartan and irbesartan–2-hydroxypropyl-β-cyclodextrin complex with model membranes** (Adamantia S. Liossi, Dimitrios Ntountaniotis, Tahsin F. Kellici, Maria V. Chatziathanasiadou, Grigorios Megariotis, Maria Mania, Johanna Becker-Baldus, Manfred Kriechbaum, Andraž Krajnc, Eirini Christodoulou, Clemens Glaubitz, Michael Rappolt, Heinz Amenitsch, Gregor Mali, Doros N. Theodorou, Georgia Valsami, Marinos Pitsikalis, Hermis Iatrou, Andreas G. Tzakos and Thomas Mavromoustakos), *In Biochimica et Biophysica Acta (BBA) - Biomembranes*, volume 1859, 2017. [details] [pdf] [doi]

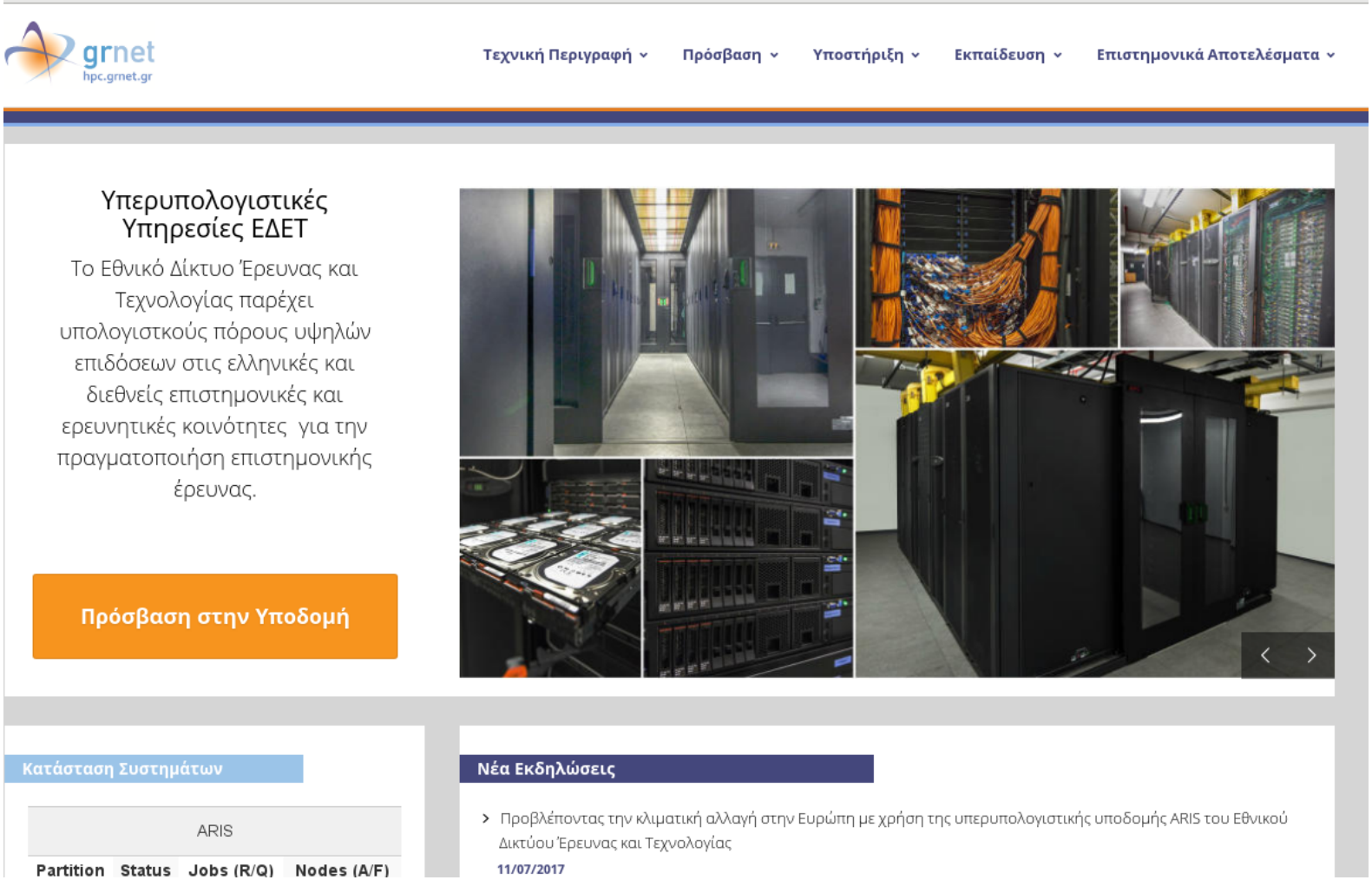
[52] **Implementation of a two-way coupled atmosphere-ocean wave modeling system for assessing air-sea interaction over the Mediterranean Sea** (George Varlas, Petros Katsafados, Anastasios Papadopoulos and Gerasimos Korres), *In Atmospheric Research*, 2017. [details] [pdf] [doi]

[51] **Monte Carlo and experimental determination of correction factors for gamma knife perfexion small field dosimetry measurements** (E Zoros, A Moutsatsos, E P Pappas, E Georgiou, G Kollias, P Karaiskos and E Pantelis), *In Physics in Medicine and Biology*, volume 62, 2017. [details] [pdf]

[50] **Near Real-Time Aerosol Predictions During the First Citizen Observatory Campaign in Greece** (Athanasopoulou, E. and Spyrou, C. and Apostolopoulou, S. and Papageorgiou, S. and Amiridis, V. and Gerasopoulos, E.), *Chapter in (Karacostas*

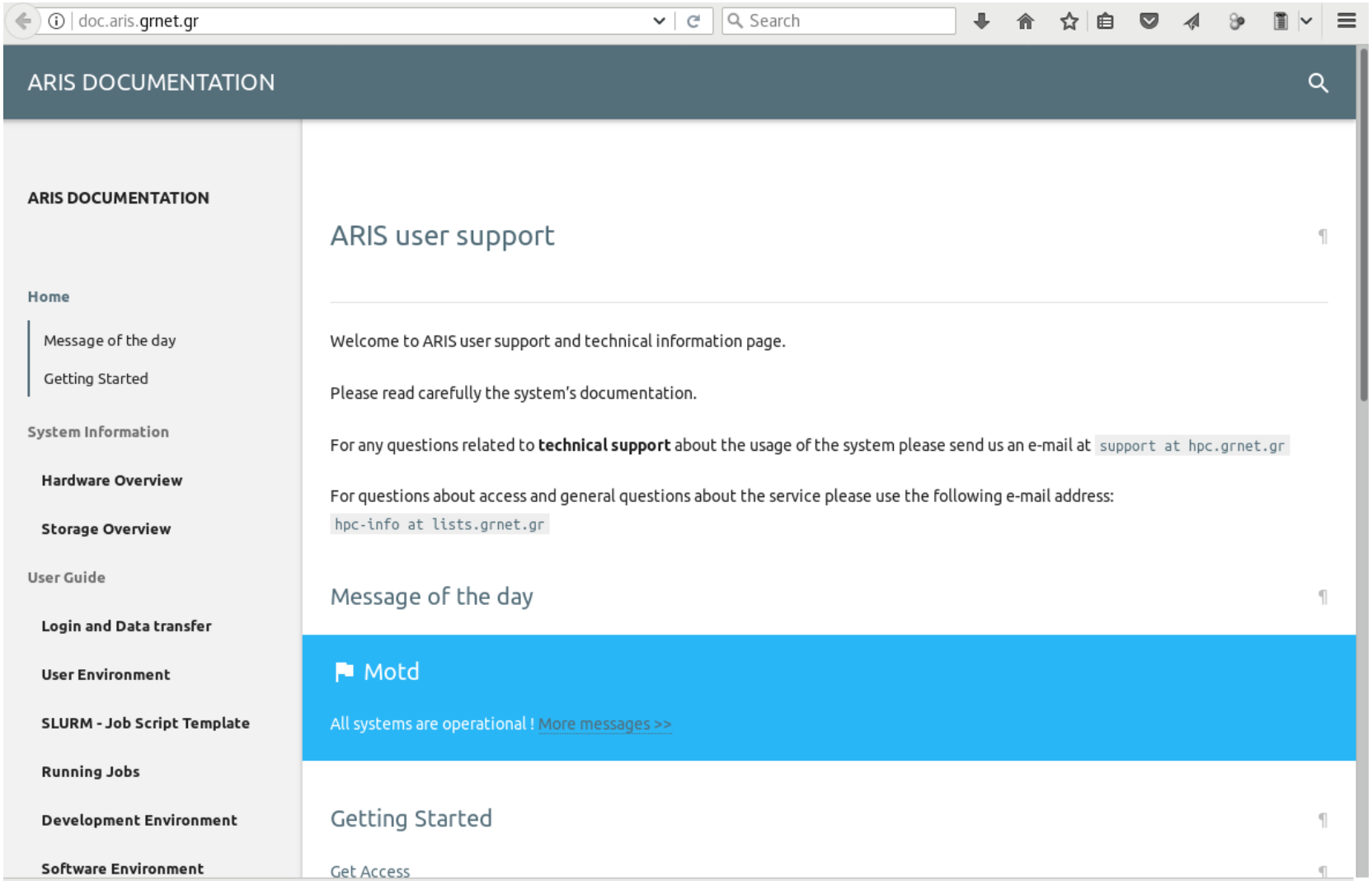
# GRNET HPC related information

- Information for Access, News etc. (mainly in Greek)
  https://hpc.grnet.gr/



# GRNET ARIS documentation

System Documentation (Only in English) :
http://doc.aris.grnet.gr/



# Contact

| | |
|---|---|
| hpc-info@lists.grnet.gr | General Information |
| hpc-access@lists.grnet.gr | Access Information, reports etc. |
| support@hpc.grnet.gr | User support |
| events.hpc.grnet.gr | Events announcement, registration etc. |

# Access Policy to ARIS (and other European Systems)

# Access Policy, Project Types

- Basic Targets
  - Efficient use of System, maximize scientific production given the resources.
  - Maximize the impact of research projects.

- Production : Periodic call for Production projects (every 6 months, for 1 year). Need to pass both technical and scientific review.
- Preparatory : Open call for projects in order to verify scaling, fit on HPC system etc. Duration 2 months. Only techical review and very basic scientific review.
- Development : Development / modification of Parallel applications. Basic technical and scientific review. Duration 4 months.

# Access Policy, Review process

- Call Announchement.
- Call open for $\sim$ 1 month. Applications
- Technical Review
- Reviewers assignment, Scientific Review.
- Summarize technical and scientific reviews, accept or reject.
- Allocation of resources (may be different than what requested, usually less core hours but not only)
- Results announcement, sign AUP, start of project.
- Periodic check of activity.
- Final Report, Results dissemination.
- Follow up : Inform for any publication with results from project.

# Notes on applications I

- Read carefully the goals of call announcement and preqrequisities.
- Technical description has the same weight as scientific description.
- Carefully calculate the requested resources.
- Describe the social, scientific etc. impact of your research.
- Describe your team's background in scientific field but also in the use this type of systems.
- Describe and give reference to the software you plan to use. In case of multi-method packages describe which methods etc. of package you'll use.
- Describe the problem size of your research.
- Carefully describe why you need an HPC system.

# Notes on applications II

- Describe the scaling of your application as function of data size/methods etc.
- The fact that an application is highly scalable does not imply that the same happens with your data.
- Describe how the code is parallelized (MPI/OpenMP/Hybrid/Other)
- Detailed description of application performance with your data on other machines (MachineName, CPU type, Memory etc.)

# Application Example

| System NAME timing | | |
|---|---|---|
| cores | Code(or case) A | Code (or case) B |
| 10 | 40 h | 80 h |
| 20 | 20 h | 40 h |
| 40 | 10 h | 20 h |
| 60 | 8 h | 10 h |
| 80 | 7 h | 15 h |

You have limited scaling data ?
Apply for a preparatory project to obtain.

# Application Example

| Run Type | no.Runs | Steps/Run | Time/step | no.cores | Total Core Hours |
|---|---|---|---|---|---|
| 1 | 20 | 1000 | 1s | 100 | $= 20 \times 1000 \times 1 \times 100/3600$=555.5 |
| 2 | 10 | 1000000 | 0.001 s | 1000 | $= 10 \times 1000000 \times$ $0.001 \times 1000/36000 = 2777$ |
| .. | | | | | 3332.5 |

You have limited scaling data ?
Apply for a preparatory project to obtain.
Some reasons that may result in reject

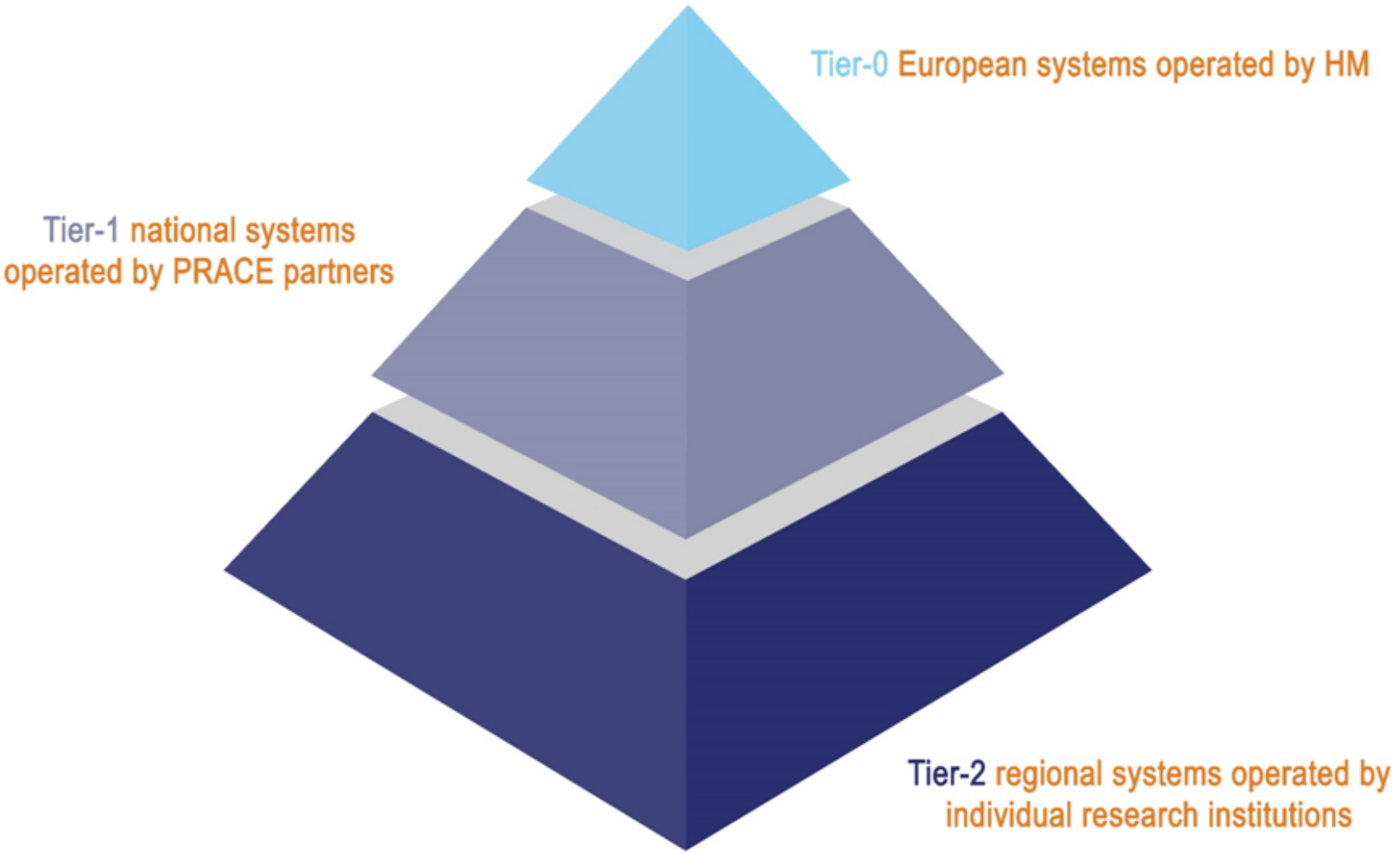
- Request memory per core more than the node memory
- Request cores per node more than the maximum available
- You ask for commercial software requiring license that either you don't have or it is locked to certain machines.

# European HPC Ecosystem - About PRACE

Partnership for Advanced Computing in Europe

- EU Organization
- Coordinates the development of Computational Infrastractures in Europe
- Offers access to Petaflop level machines (Tier-0)
- Much more.
- Greece is Founder member of organization - non hosting member since 2007
- Since 2015 is hosting Tier-1 system.

# PRACE Systems Hierarchy



# PRACE Tier-0 Systems

MareNostrum: **Lenovo**
BSC, Barcelona, Spain

#16 on Top500

**Piz Daint**: Cray XC 50
CSCS
Lugano, Switzerland

#3 on Top500 Nov17

#22 on Top500

JUQUEEN: IBM
BlueGene/Q
GAUSS/FZJ
Jülich, Germany

SuperMUC: IBM
GAUSS/LRZ Garching, Germany
#44 & #45 on Top500

Hazel Hen: Cray
GAUSS/HLRS,
Stuttgart, Germany

#19 on Top500

#93 on Top500

CURIE: Bull Bullx
GENCI/CEA
Bruyères-le-Châtel, France

#14 on Top500

MARCONI: Lenovo
CINECA
Bologna, Italy

# European HPC Tier-1 Ecosystem - PRACE

- DECI : Resources exchange program : Each Tier-1 hosting country contributes a part of compute capacity, and researchers from this country can get access to other Tier-1 systems.
- Main reasons
  - Trigger International Scientific Cooperations
  - Possibility to use resources of different type that are not available. For example, Bigger than available systems, BlueGene, Cray, KNL, etc.
  - Intermediate stage before Tier-0 access.
  - Evaluation of projects in home country.
- Countries in DECI : Cyprus, Czech Republic, Finland, Greece, Hungary, Ireland, Italy, Norway, Poland, Spain, Sweden, Netherlands, UK.
- Calls for DECI Projects every 6 months. Announced in prace (and hpc.grnet.gr) web site.

# ARIS : Efficient use, Tools, Best Practices

Once access is granted :

- Remember : Starting point the system documentation http://doc.aris.grnet.gr/
- It is also mentioned in login screen

```
================================================================================
PLEASE REMEMBER :
       Each compute/gpu/phi node has 20 cores and  56G of memory
       Each fat             node has 40 cores and 490G of memory
================================================================================

System Documentation available at : http://doc.aris.grnet.gr/
================================================================================
```

- To start with,

# ARIS : Connect to

- ONLY ssh connections are allowed
- Policy is : Deny all except.
- SSH ONLY from certain IPs/Networks.
- Use your organization VPN service if you need to connect from other places.
- SSH ONLY with keys
- Shell could be obtained ONLY on login nodes. Compute nodes are unreachable (from login nodes too).
- Exception : Visualization nodes. See the corresponding section in documentation.
- ONLY login nodes have partial internet access. SSH from login nodes to everywhere is also denied.
- Need help with SSH ? : doc.aris.grnet.gr

# Working with installed Software

- Software is organized with Environment Modules
- Environment modules dynamically alter `PATH`, `LD_LIBRARY_PATH` and other variables.
- Currently 5 sections
  - Compilers : various versions of : gnu, intel, pgi, cuda, sun, clang, java, binutils, scala, etc.
  - Parallel : various versions of : IntelMpi, OpenMPI, mvapich2, mpich, and few parallel profile tools, scalasca, mpiP etc.
  - Libraries : Linear Algebra, Fourier Transofroms, I/O (hdf5, netcdf) and much more, optimized on system architecture(s).
  - Applications : All the applications that users asked for (opensource), some licensed applications i.e. available to users who own the license.
  - Commonly used tools : Like recent versions of make, cmake, git etc.

# Working with installed Software I

- List available modules

```
module avail
```

- List active modules

```
module list
```

- Deactivate all active modules

```
module purge
```

- Deactivation of a certain module

```
module unload MODULENAME
```

- Switch module version

```
module switch MODULENAME/VER1 MODULENAME/VER2
```

- To make users life easier, the gnu/4.9.2, intel/15.0.3, intelmpi/5.0.3 modules are preloaded upon login.

# Working with installed Software

## Example : module avail

```
...
binutils/2.25       gnu/5.2.0              intel/17.0.5
binutils/2.26       gnu/5.3.0              intel/18.0.0
...
intelmpi/2017.0     mvapich2/gnu/2.2.2a    openmpi/1.8.8
intelmpi/2017.1     mvapich2/intel/2.2.2a  openmpi/2.0.0/gnu
...
gsl/2.2.1/intel     parmetis/4.0.3/intel
hdf5/1.8.12/gnu     petsc/3.6.2(default)
...
lammps/7Dec15       visit/2.10.2
lsdalton/1.2        visit/2.11.0
...
```

- If you have your own code, you should compile it.
- Suggested Compilers, MPI and Flags : Intel, Intelmpi, mpiicc -O3 -xCORE-AVX-I and other typical for your source.

# Running Applications

- Running on login nodes is not allowed, although someone can run a few minutes check with small number of cores.
- You should use Resource Manager/Batch system to submit a job to compute nodes.
- Batch system on ARIS is SLURM.
- How to use it ? doc.aris.grnet.gr
- There is a script generator validator that is a good starting point to create a SLURM script.
- What is the content of this script ?
- You define the resources you need for your job and how to run.

# Workload manager/Batch system : SLURM

```
#!/bin/bash
#SBATCH --job-name="testSlurm" # JobName
#SBATCH --error=job.err.%j     # Filename : stderr
#SBATCH --output=job.out.%j    # Filename : stdout
                               # %j value of JobID
#SBATCH --nodes=2              # Number of nodes
#SBATCH --ntasks=4             # Number of (usually MPI) Tasks
#SBATCH --ntasks-per-node=2    # Number of Tasks / node
#SBATCH --cpus-per-task=10     # Number of Threads / MPI Task
#SBATCH --mem=56G              # Memory per node  #  One of these 2 specs
#SBATCH --mem-per-cpu=2800M    # Memory per core  #
#SBATCH -A ptc                 # Accounting tag # ptc for training
#SBATCH -t 1-01:00:00          # Requsted  DD-HH:MM:SS
#SBATCH -p compute             # partition, one of compute, gpu, phi, fat, taskp, short
#SBATCH --gres=gpu:2           # Accelerated partitions. gres=gpu or mic
module purge
module load gnu/4.9.2
module load intel/15.0.3
module load intelmpi/5.0.3
if [ x$SLURM_CPUS_PER_TASK == x ]; then            #
  export OMP_NUM_THREADS=1                         #
else                                               # Never delete these lines
  export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK      # unless you exactly know what you do
fi                                                 #
srun EXECUTABLE ARGUMENTS   # Executable and possible arguments
```

# Working with SLURM

- Job Submission

```
sbatch SLURM_JobScript.sh
Submitted batch job 123456
```

- Job List

```
squeue
```

- Cancel a Job

```
scancel JobID
```

- Send KILL signal (instead of the default TERM) to a job

```
scancel -s KILL JobID
```

- Estimation of job start time that is queued due to not available resources

```
squeue --start
```

- Information for the resources status.

```
sinfo
```

# Working with SLURM I

- DO NOT use the typical mpirun/mpiexec(.hydra). Use srun for SLURM.
- You may omit some of requirements if the rest can define the required resources
- Examples : You may omit ntasks, requires nnodes, ntasks-per-node, cpus-per-task to be defined. System can calculate how many tasks to use
- Especially for hybrid MPI/OpenMP applications DO NOT delete the piece of code that checks if you set correctly threads/tasks : A common mistake in production runs.
- Required time is mandatory. If you omit it, either job will never run (default for ARIS) or will use the default maximum wall time (2 days for aris)

# Working with SLURM II

## SLURM User/Group resource limits

- Each account has certain resource limits.
  - Maximum number of running Jobs, Jobs in queue.
  - Maximum number of concurrently used cores and/or nodes
  - Maximum Wall time duration of a job
  - Maximum consumable Core Hours for project duration (=Budget).

# Applications Profiling

- Profiler is software that gets metrics on source execution, without addition of timers in source code.
- Serial Profilers
  - One can find detailed time spent in code procedures, i.e. How many times a procedure was called, average time per call, total time spent in procedure, from which point in source was called etc.
  - Standard Unix profiler gprof and its variants, for example sprof.
  - Compiler specific profilers, like vtune for Intel compilers or pgprof for PGI.

# Applications Profiling I

- MPI
  - MPI implementations profilers, for example OpeMPI VampirTrace.
  - mpiP : Traces MPI calls and gives performance indicators, possible bottlenecks etc. OpenSource, Works with any compiler and MPI implementation.
  - The simplest to use, with great reporting, may be used by just adding some libraries at link stage.

```
module load mpiP
mpif90 $OBS -g -L$MPIPROOT/lib -lmpiP -lbfd -lunwind -o myexe.x
```

  - Run as usual and look the report in file myexe.x.NPROCS.PID.mpiP. Probably you'll find bottlnecks in your code (or data driven bottlenecks) with just one run

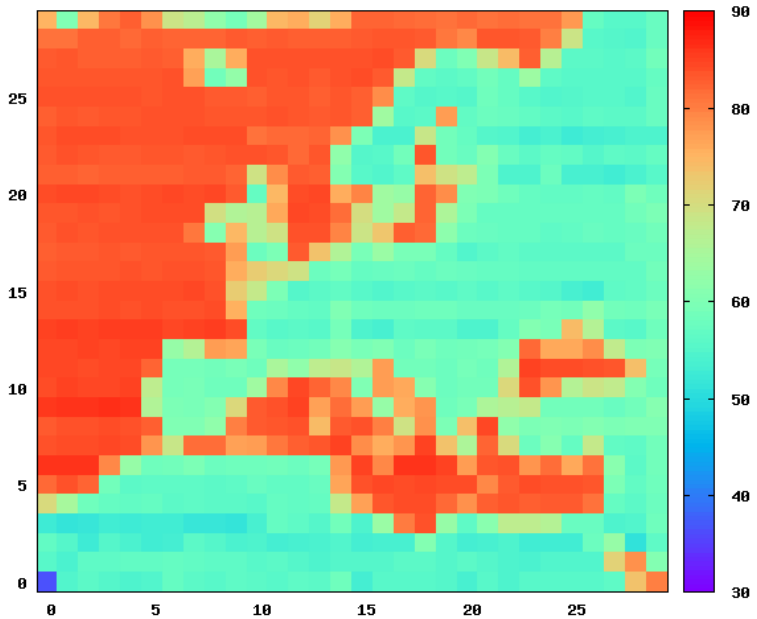
# Applications Profiling II

```
@ mpiP
@ Command : ./06.x
@ Version                  : 3.4.1
@ MPIP Build date          : Sep  7 2015, 16:33:51
@ Start time               : 2017 11 29 21:45:28
@ Stop time                : 2017 11 29 21:45:31
@ Report generation        : Collective
@ MPI Task Assignment      : 0 login01
......
@--- MPI Time (seconds) ---------------------------------------------------
--------------------------------------------------------------------------
Task    AppTime    MPITime     MPI%
   0       2.72        0.7    25.69
   1       2.72       1.16    42.52
   2       2.72       1.07    39.11
.....
 31       2.72       1.13    41.51
  *       87.1       34.9    40.05
......
@--- Callsites: 11 --------------------------------------------------------
--------------------------------------------------------------------------
 ID Lev File/Address                        Line Parent_Funct             MPI_Call
  1   0 06_md_inhomegeneous_reduce.f         115 md                       Bcast
  2   0 06_md_inhomegeneous_reduce.f         137 md                       Bcast
  3   0 06_md_inhomegeneous_reduce.f         202 md                       Reduce
.....
@--- Aggregate Time (top twenty, descending, milliseconds) ----------------
--------------------------------------------------------------------------
Call                 Site       Time    App%    MPI%      COV
```

# Applications Profiling III

```
Reduce                    4   1.27e+04   14.57   36.37    0.94
Barrier                   8   1.03e+04   11.78   29.41    1.09
Bcast                     6    8.8e+03   10.10   25.22    0.18
......
@--- Aggregate Sent Message Size (top twenty, descending, bytes) ----------
-----------------------------------------------------------------------------
Call                   Site       Count       Total       Avrg  Sent%
Reduce                    3          32    3.2e+07      1e+06  11.11
Bcast                    11          32    3.2e+07      1e+06  11.11
......
```

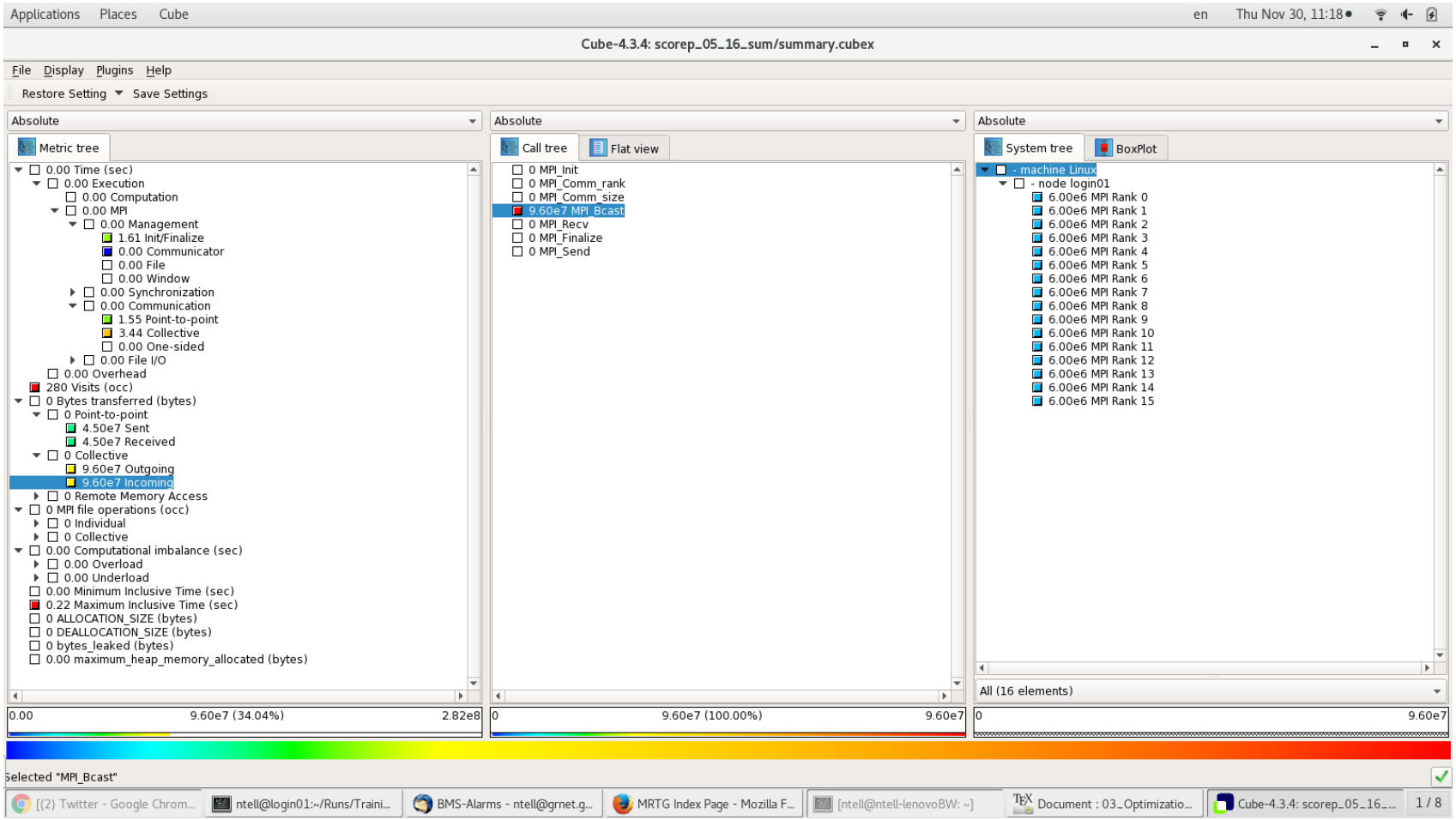
- Example : WRF load imbalance.



# Applications Profiling I

- Hybrid MPI/OpenMP/Threads Profilers
  - scalasca : Traces MPI calls, as well as OpenMP calls, provides detailed information timing information per thread, task, node, code line. Graphical Interface to explore profile information.
  - It is necessary to compile the code with scalasca wrapper :
    scalasca -instrument mpicc FLAGS .....
    scalasca -analyze srun exe .....
    scalasca -examine Report Directory .....

# Applications Profiling



# Efficient use

- ARIS compute nodes have 20 or 40 cores. Use if possible full nodes, i.e. 20/40 cores/node.
- If it is not the case, limit the required nodes.

| cores | Nodes | tasks/node | Unused cores |
|---|---|---|---|
| 64 | 4 | 20 | 16 on 1 node |
| 128 | 7 | 20 | 12 on 1 node |
| 256 | 13 | 20 | 4 on 1 node |
| 512 | 26 | 20 | 8 on 1 node |

- Common mistake

| cores | Nodes | tasks/node | Unused cores |
|---|---|---|---|
| 64 | 8 | 8 | 12 cores/node on 8 nodes=96 |
| 64 | 4 | 16 | 4 cores/node on 4 nodes = 16 |
| 90 | 6 | 15 | 5 cores/node on 6 nodes = 30 |
| 128 | 8 | 16 | 4 cores/node on 8 nodes = 32 |
| 480 | 40 | 12 | 8 cores/node on 40 nodes = 320 |
| 512 | 32 | 16 | 4 cores/node on 32 nodes = 128 |

# Efficient use

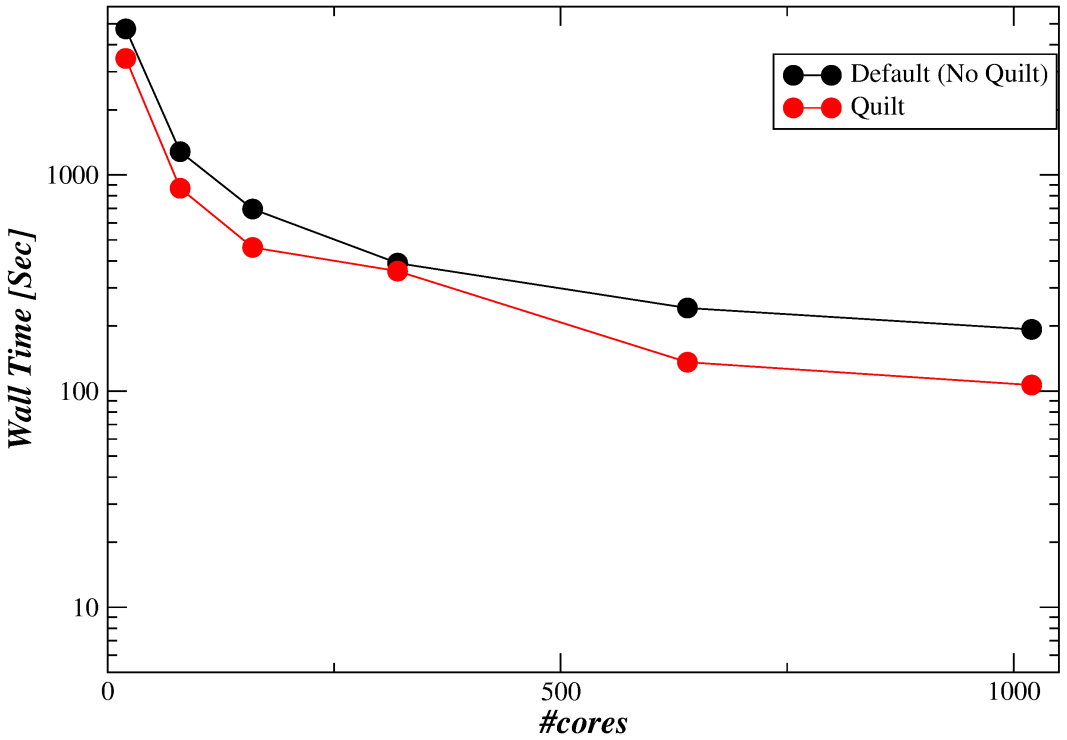
- Do not use mpirun/mpiexec nor typical desktop arguments like -np. It happens to forget to change the really needed resources, for example :

```
#SBATCH --nodes=10
#SBATCH --ntasks=200
mpirun -np 8
or
srun -n 8
```

  You allocate (and charged for) 200 cores while you use only 8.
- Try to use the correct combination of tasks and threads with Hybrid applications. Check that the `OMP_NUM_THREADS` is set. In SLURM script template there is code that checks for this.
- Surprisingly, this piece of code is frequently removed.

# Efficient use

- Explore the capabilities of your application. With some options in input file(s) you may see much better performance.
- Example : WRF quilting

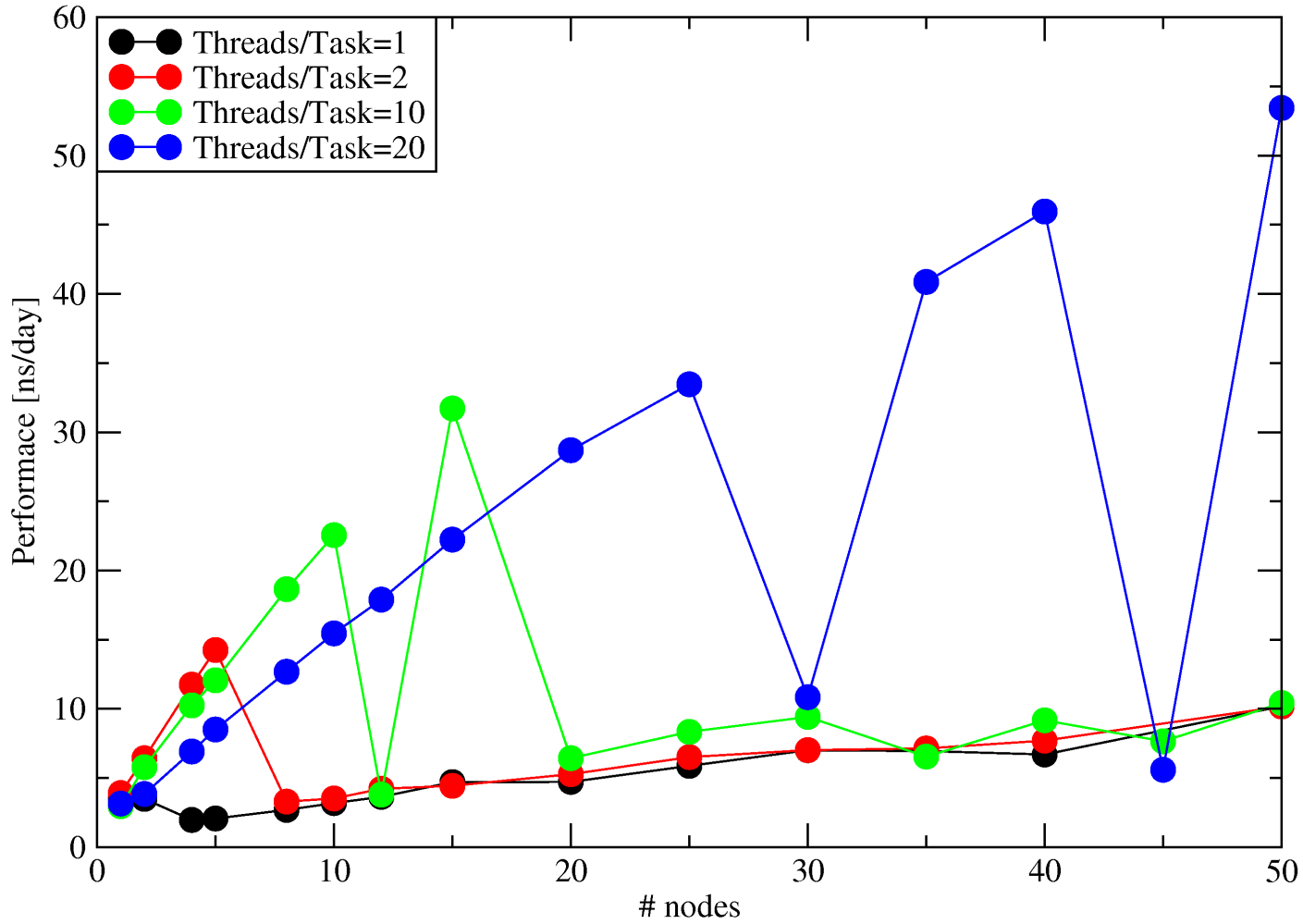


# Efficient use

- Usually applications have a recipe for number of tasks to use as function of data details, For example WRF, cores as f(domain dimensions), MD cores as f(atoms), ab-initio cores as f(wfns, atoms, etc.)
- Direct/Semidirect/Scratch methods/variables in ab-initio codes.
- A highly scalable application may be very inefficient with your data. For example, namd is highly scalable on many nodes and many gpus. This does not apply if your system is small. If your system contains less than 100k atoms, you should use half node and one (of two) gpus to obtain efficiency of $\sim$ 80%.
- With hybrid applications, check before production runs the performance with various combinations tasks/threads.

# Efficient use

- Example : MD of an inhomogeneous system



# Efficient use

- If you can use save/restart and need very long time, use it. Instead of a job of 10 days, use 10 jobs of 1 day (propability of a HW failure in 10 days much higher - especially with multinode runs).
- Request from the Resource Manager wall time slightly higher than the expected. NOT the typical 2 days.
- Example : Submit 100 jobs requesting 2 days each. Scheduler will arrange to run them in $\sim$ 1 week. If each run takes 5 minutes, requesting 6 minutes, all runs will finish in $\sim$ 1 hour instead of $\sim$ 1 week.
- Even better, submit few jobs with multiple srun, for example 10 jobs with 10 srun.
- Stats : Sept. 2017
  65% of jobs took up to 5% of requested time
  9% between 5 and 10%.
  11% more than 50%