

Accelerated computing platforms for EGI

Marco Verlato
INFN-Padova

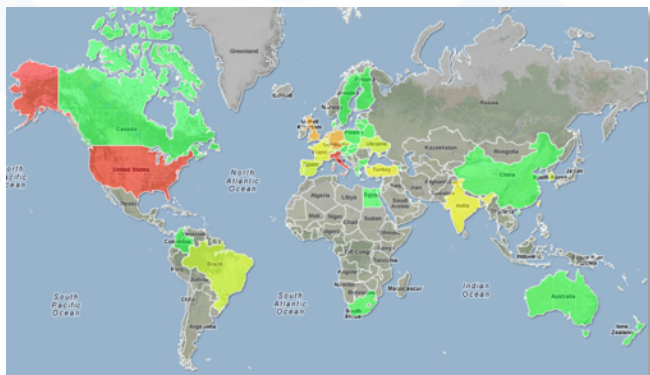
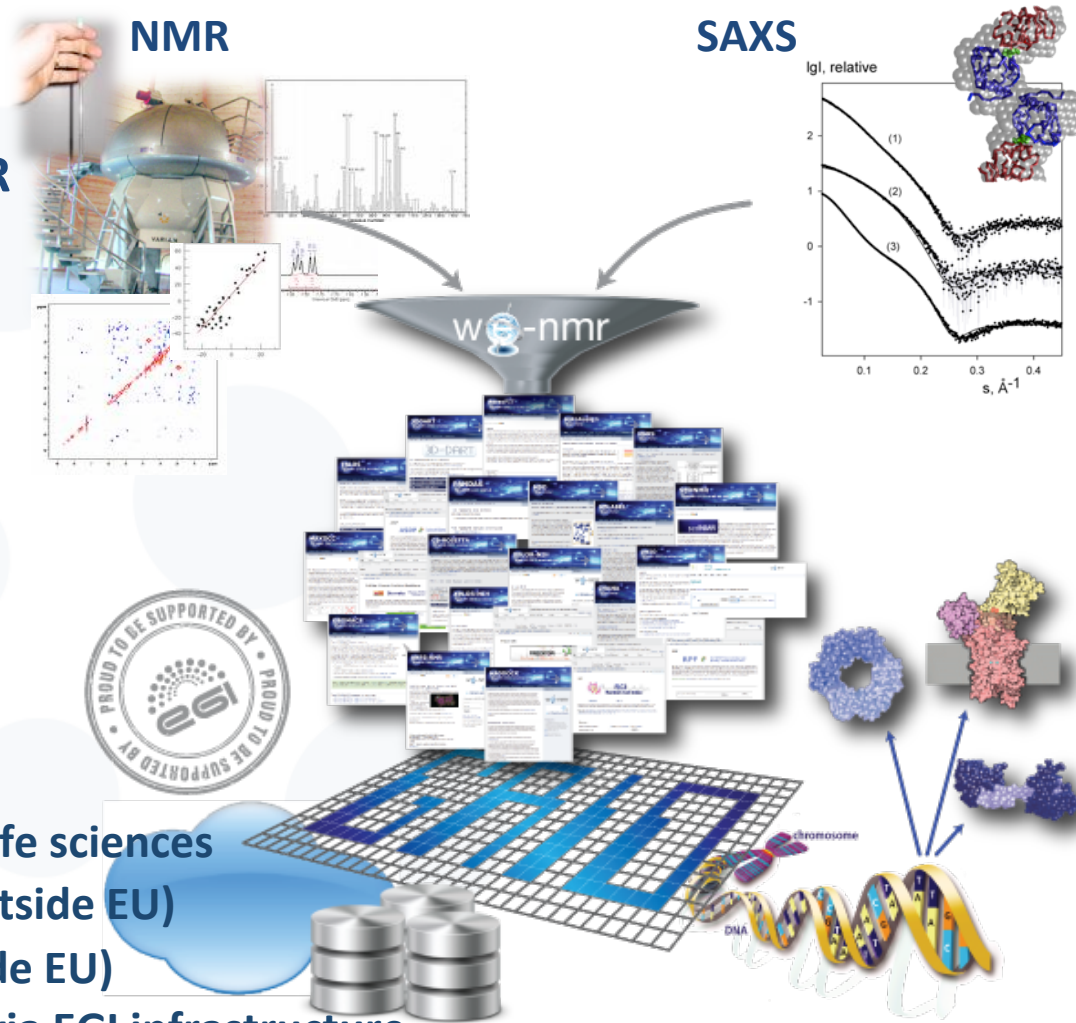
Workshop della
Commissione Calcolo e Reti dell'INFN
19 Maggio 2016, La Biodola – Isola d'Elba



- Task part of EGI-Engage JRA2: Platforms for the Data Commons
 - Expand the EGI Federated Cloud platform with new IaaS capabilities
 - Prototype an open data platform
 - **Provide a new accelerated computing platform**
 - Integrate existing commercial and public IaaS Cloud deployments and e-Infrastructures with the current EGI production infrastructure
- Duration: 1st March 2015 – 31st May 2016 (15 Months)
- Partners:
 - **INFN**: CREAM developers at Padua and Milan divisions
 - **IISAS**: Institute of Informatics, Slovak Academy of Sciences
 - **CIRMMP**: Scientific partner of MoBrain CC (and WeNMR/West-Life, INDIGO-DataCloud H2020 projects)

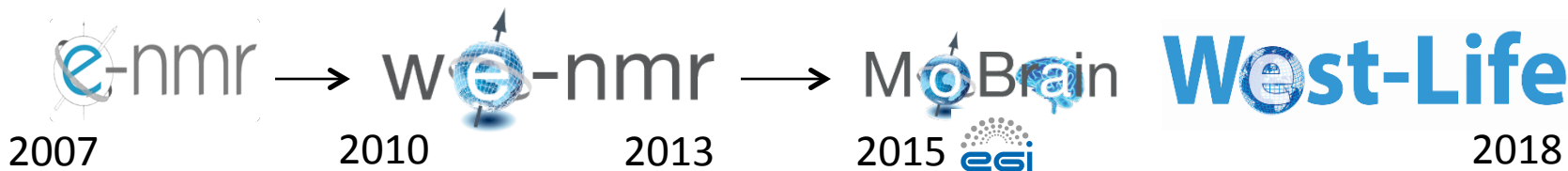
- Goal:
 - **implement the support in the information system**, to expose the correct information about the accelerated computing technologies available – both software and hardware – at site level, developing a common extension of the information system structure, based on OGF GLUE standard...
 - **extend the HTC and cloud middleware support for co-processors**, where needed, in order to provide a transparent and uniform way to allocate these resources together with CPU cores efficiently to the users
- Ideally divided in two subtasks:
 - Accelerated computing in Grid (= HTC Platform)
 - Accelerated computing in Cloud
- Requirements from user communities collected at EGI Conference in May 2015: see <http://bit.ly/Lisbon-GPU-Session>

A worldwide e-Infrastructure for NMR and structural biology

WeNMR VRC (December 2015)

- One of the largest (#users) VO in life sciences
- > 720 VO registered users (36% outside EU)
- > 2250 VRC members (>60% outside EU)
- ~ 41 sites for >142 000 CPU cores via EGI infrastructure
- User-friendly access to Grid via web portals



- A Competence Center to Serve Translational Research from Molecule to Brain (core partners also involved in WeNMR/West-Life and INDIGO)
- Dedicated task for “GPU portals for biomolecular simulations”
 - To deploy the AMBER and/or GROMACS packages on GPGPU test beds, develop standardized protocols optimized for GPGPUs, and build web portals for their use
 - Develop GPGPU-enabled web portals for exhaustive search in Cryo-EM density. This result links to the Cryo-EM task 2 of MoBrain
- Requirements
 - GPGPU resources for development and testing (CIRMMP, CESNET, BCBR)
 - Discoverable GPGPU resources for Grid (or Cloud) submission (for putting the portals into production)
 - GPGPU Cloud solution, if used, should allow for transparent and automated submission
 - Software and compiler support on sites providing GPGPU resources (CUDA, openCL)

Accelerated Computing in Grid

Paolo Andreetto (INFN-PD)

David Rebatto (INFN-MI)

Marco Verlato (INFN-PD)

Lisa Zangrando (INFN-PD)



Accelerated computing in Grid

- The problem:
 - CREAM-CE is the most popular grid interface (Computing Element) to a number of LRMSes (Torque, LSF, Slurm, SGE, HTCondor) since many years in EGI
 - Most recent versions of these LRMSes do support natively GPGPUs (or MIC cards), i.e. servers hosting these cards can be selected by specifying LRMS directives
 - CREAM must be enabled to publish this information and support these directives
- Work plan
 - Identifying the relevant GPGPU-related parameters supported by the different LRMS, and abstract them to significant JDL attributes
 - Implementing the needed changes in CREAM-core and BLAH components
 - Writing the info-providers according to GLUE 2.1
 - Testing and certification of the prototype
 - Releasing a CREAM update with full GPGPU support

- Started from previous analysis and work of EGI Virtual Team (2012) and GPGPU Working Group (2013-2014)
- CIRMMP/MoBrain use-case and testbed
 - **AMBER**, **Powerfit** and **DisVis** applications with CUDA 5.5
 - 3 nodes (2x Intel Xeon E5-2620v2) with 2 NVIDIA Tesla K20m GPUs per node
 - Torque 4.2.10 (source compiled with NVML libs) + Maui 3.3.1
 - EMI3 GPU-enabled CREAM-CE prototype
- Tested local job submission with the different GPGPU supported options, e.g. with Torque/pbs_sched:

```
$ qsub -l nodes=1:gpus=2:default job.sh
```

```
$ qsub -l nodes=1:gpus=2:exclusive_process job.sh
```

- ...and with Torque/Maui:

```
$ qsub -l nodes=1 -W x='GRES:gpu@2' job.sh
```



- NVIDIA Compute Mode can have the following values:
 - 0/default – accept simultaneous CUDA contexts.
 - 1/exclusive_thread - Single context allowed, from a single thread.
 - 2/prohibited - No CUDA contexts allowed.
 - 3/exclusive_process - Single context allowed, multiple threads OK. Most common setting.
- However the Compute Mode is supported only with pbs_sched, so we only defined the new JDL attribute "GPUNumber" and implemented it in BLAH and CREAM parser
- Example of JDL for DisVis (see <https://wiki.egi.eu/wiki/CC-MoBrain>):

```
$ glite-ce-job-submit -o jobid.txt -d -a -r cegpu.cerm.unifi.it:8443/cream-pbs-batch disvis.jdl
$ cat disvis.jdl
[
executable = "disvis.sh";
inputSandbox = { "disvis.sh" , "O14250.pdb" , "Q9UT97.pdb" , "restraints.dat" };
stdoutput = "out.out";
outputsandboxbasedesturi = "gsiftp://localhost";
stderr = "err.err";
outputsandbox = { "out.out" , "err.err" , "res-gpu.tgz" };
GPUNumber=2;
]
```

- CIRMMP testbed was also used to test new JDL attributes propagation from CREAM-core to different batch systems
- A new testbed with GPU and MIC cards managed by HTCCondor was made available at GRIF/LLR in March 2016
- A CREAM/HTCCondor prototype supporting both GPU and MIC cards was successfully implemented and tested (thanks to A. Sartirana)
- Two additional JDL attributes were defined, other than GPUNumber
 - GPUModel: for selecting the servers with a given model of GPU card
 - e.g. GPUModel="Tesla K20m"
 - MICNumber: for selecting the servers with the given number of MIC cards

- The new attributes are supported also by LSF and Slurm batch systems
- CREAM/Slurm prototype supporting GPUs was successfully implemented and tested at ARNES data centre in April 2016 (thanks to B. Krasovec)
 - 3 GPU nodes with 2 Tesla GPUs each (K40c, K20c and K10 models)
 - Support to GPUModel selection not available yet (needs upgrade to Slurm version 15)
 - This site is in production, CREAM is maintained for supporting Belle VO, other EGI VOs could be enabled with lower priority
- Basic tests successfully carried out at Queen Mary data centre, a SGE based cluster with OpenCL compatible AMD GPU (thanks to D. Traynor)

- GLUE2.1 draft analysed as a base for writing GPU-aware infoprovider
- Accelerators:
 - GPGPU (General-Purpose computing on Graphical Processing Units)
 - NVIDIA GPU/Tesla/GRID, AMD Radeon/FirePro, Intel HD Graphics,...
 - Intel Many Integrated Core Architecture
 - Xeon Phi Coprocessor
 - Specialized PCIe cards with accelerators
 - DSP (Digital Signal Processors)
 - FPGA (Field Programmable Gate Array)
- **ExecutionEnvironment** class: represents set of homogeneous WNs
 - Is usually defined statically during the deployment of the service
 - These WNs however can host different types/models of accelerators
- New class proposal: **AcceleratorEnvironment**
 - Describes an homogeneous set of accelerator devices
 - Can be associated to one or more Execution Environments

- Static info-provider based on GLUE2.1 AcceleratorEnvironment class

- Info can be obtained in Torque e.g. from pbsnodes:

```
gpus = 2
gpu_status = gpu[1]=gpu_id=0000:42:00.0;gpu_pci_device_id=271061214;gpu_pci_location_id=0000:42:00.0;gpu_product_name=Tesla
K20m;gpu_display=Enabled;gpu_memory_total=5119 MB;gpu_memory_used=11
MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_utilization=0%;gpu_ecc_mode=Disabled;gpu_temperatur
e=18 C,gpu[0]=gpu_id=0000:04:00.0;gpu_pci_device_id=271061214;gpu_pci_location_id=0000:04:00.0;gpu_product_name=Tesla
K20m;gpu_display=Enabled;gpu_memory_total=5119 MB;gpu_memory_used=13
MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_utilization=0%;gpu_ecc_mode=Disabled;gpu_temperatur
e=16 C,driver_ver=319.37,timestamp=Thu Sep 17 10:18:07 2015
```

- Dynamic info-providers need new attributes in GLUE2.1 draft:

- ComputingManager class (the LRMS)

- TotalPhysicalAccelerators, TotalAcceleratorSlots, UsedAcceleratorSlots

- ComputingShare class (the batch queue)

- MaxAcceleratorSlotsPerJob, FreeAcceleratorSlots, UsedAcceleratorSlots

- CREAM Accounting sensors, mainly relying on LRMS logs, were in the past developed by the APEL team
- APEL team has been involved in the GPGPU accounting discussion
- Batch systems should report GPU usage attributable to the job in the batch logs. APEL would then parse the logs files to retrieve the data.
- Unfortunately job accounting records of Torque and LSF8, LSF9 do not contain GPGPU usage info ☹️
- NVML allows to enable per-process accounting of GPGPU usage using Linux PID, but not LRMS integration yet, e.g.:

```
$ nvidia-smi --query-accounted-apps=pid,gpu_serial,gpu_name,gpu_utilization,time --format=csv  
pid, gpu_serial, gpu_name, gpu_utilization [%], time [ms]  
44984, 0324713033232, Tesla K20m, 96 %, 43562 ms  
44983, 0324713033232, Tesla K20m, 96 %, 43591 ms  
44984, 0324713033096, Tesla K20m, 10 %, 43493 ms  
44983, 0324713033096, Tesla K20m, 10 %, 43519 ms
```

- Discussion ongoing with APEL team

- CREAM GPU-enabled prototype available and tested at:
 - CIRMMP (local Torque based GPU cluster)
 - GRIF/LLR (Production HTCondor based GPU & MIC cluster)
 - ARNES (Production Slurm based GPU cluster)
 - Queen Mary (local SGE based cluster with AMD GPUs)
- Plans to test the prototype at INFN-CNAF (LSF9 based GPU cluster)
- The goal is to have a major release of CREAM-CE on CentOS7, in order to be included in UMD4, with GPU/MIC support for Torque, HTCondor, Slurm, SGE, LSF
- Still missing:
 - GLUE2.1 draft approval
 - Writing GLUE2.1 compliant info-providers
 - GPU accounting

Accelerated Computing in Cloud

Viet Tran (IISAS)

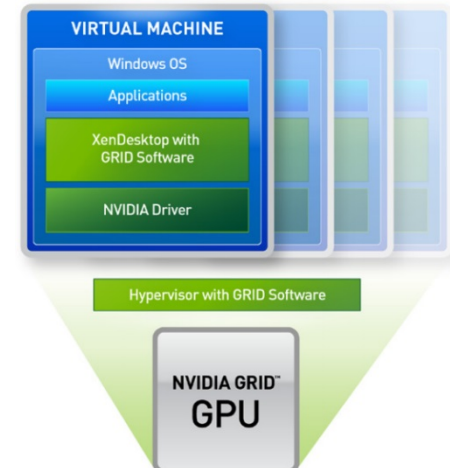
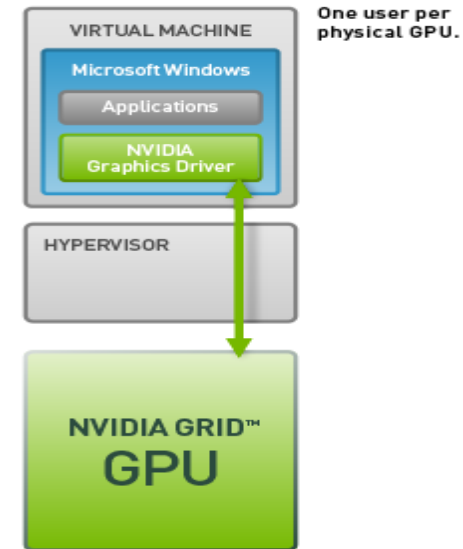
Jan Astalos (IISAS)

Miroslav Dobrucky (IISAS)



Accelerated computing in Clouds

- Virtualization technologies
 - KVM with passthrough is rather mature
 - But maximum 1 VM attached to 1 physical card
 - Virtualized GPU is in early stage:
 - NVIDIA GRID vGPU available sofar for XenServer and VMWare hypervisors only
- Cloud framework support
 - Openstack support for PCI passthrough
 - OpenNebula support for PCI passthrough from v4.14
- FedCloud services support
 - Information system
 - Accounting



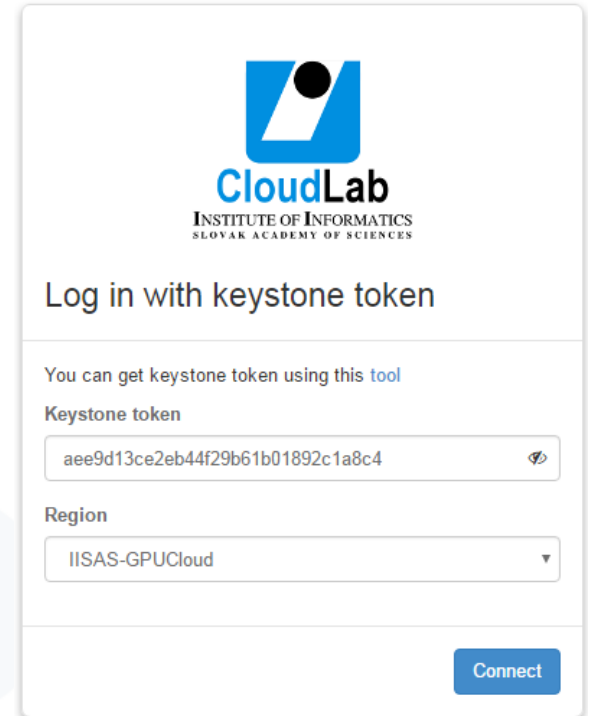
- First test-bed set up at IISAS
- Hardware:
 - 2 IBM dx360 M4 servers with 2x Intel Xeon E5-2650v2
 - 16 CPU cores, 64GB RAM, 1 TB storage on each WN
 - 2x NVIDIA Tesla K20m on each WN
- Software
 - Base OS: Ubuntu 14.04 LTS
 - KVM hypervisor with PCI passthrough virtualisation of GPU cards
 - OpenStack Kilo middleware
 - Newest FedCloud tools

Testing, optimization, troubleshooting

- Default setting is not suitable for production
 - Low performance
 - Random crashing
- Extensive testing, optimization and troubleshooting has been carried out behind the scenes:
 - Tuning BIOS setting (hardware dependent):
VM can interact directly with hardware, e.g. sending NMI (Non-maskable interrupt) to BIOS caused system crashing. Setting BIOS to tolerate/immune to events from devices. Typical case: loading nouveau in VM cause system reboot
 - Disabling CPU hyperthreading
 - Setting correct CPU type in nova.conf:
most safely `cpu_mode = host-passthrough`

- Results
 - Fully working cloud site with GPGPU
 - VMs reach native performance (around 2% differences)
 - Exact, repeatable crashing scenarios and workarounds
- OpenStack/Kilo site fully certified and integrated with EGI FedCloud in October 2015:
 - Pre-defined images with NVIDIA drivers and CUDA toolkit installed
 - GPU-enabled flavors: *gpu1cpu6* (1GPU + 6 CPU cores), *gpu2cpu12* (2GPU +12 CPU cores)
 - Supported VOs: fedcloud.egi.eu, [ops](http://ops.egi.eu), [dteam](http://dteam.egi.eu), [moldyngrid](http://moldyngrid.egi.eu), [enmr.eu](http://enmr.egi.eu), vo.lifewatch.eu

- User/site admins support:
 - How to use GPGPU on IISAS-GPUCloud
 - Access via rOCCI client
 - Access via OpenStack dashboard with token
 - How to create your own GPGPU server in cloud
 - Automation via scripts:
 - NVIDIA + CUDA installer
 - Keystone-VOMS client for getting token
 - How to enable GPGPU passthrough in OpenStack
- All this in a wiki:
<https://wiki.egi.eu/wiki/GPGPU-FedCloud>



The screenshot shows the CloudLab login page. At the top is the CloudLab logo, which includes a blue square with a white circle and the text 'CloudLab INSTITUTE OF INFORMATICS SLOVAK ACADEMY OF SCIENCES'. Below the logo is the heading 'Log in with keystone token'. A sub-heading reads 'You can get keystone token using this tool'. There are two input fields: 'Keystone token' with the value 'aee9d13ce2eb44f29b61b01892c1a8c4' and a copy icon, and 'Region' with a dropdown menu showing 'IISAS-GPUCloud'. A blue 'Connect' button is located at the bottom right of the form.

- Conceptual Model of the Cloud Computing Service is being defined in GLUE2.1 draft
 - The **CloudComputingInstanceType** class describes the hardware environment of the VM (i.e. the flavour)
 - New **CloudComputingVirtualAccelerator** entity defined to describe a set of homogeneous virtual accelerator devices, who can be associated to one or more CloudComputingInstanceTypes
- GPU accounting easier in cloud environment (1 VM \leftrightarrow 1 GPU)
 - Cloud systems currently return wallclock time only
 - If the wallclock for how long a GPU was attached to a VM is available then the GPU reporting would be in line with cloud CPU time, i.e. wallclock only
 - APEL team involved to define an extended usage record and new views to display GPU usage in the Accounting Portal

New sites soon available in EGI FedCloud/1

- **At INCD/LIP**
 - 2 compute nodes with NVIDIA GPUs - Tesla K40
 - OpenStack Kilo, PCI passthrough
 - Being integrated to EGI FedCloud: OCCl installed, waiting for keystone-voms for keystone v3
 - Support for docker images
- **At Seville (CSIC-EBD-LW)**
 - Supporting LifeWatch community
 - ~500 cores and 1 PB of storage, with Tesla K20m GPUs
- **IISAS will provide supports/helps for installation and configuration**

- CESNET-MetaCloud OpenNebula site
 - A multi-purpose PCI pass-through capabilities were introduced in OpenNebula version 4.14
 - First release with this functionality → still a bit rough around the edges
 - The site upgraded to 4.14 in April 2016, 4 Tesla M2090 available
 - It is considered still experimental and UNSTABLE (i.e., just for testing purposes)
 - Plans to provide templates & user guides for GPU-enabled virtual machines (as done for IISAS-GPGPUCloud)
 - The long-term goal is to provide OCCl extensions to select these "additional" capabilities for virtual machines on a case-by-case basis (not just by using a pre-defined template)

Applications

Antonio Rosato (CIRMMP)

Andra Giachetti (CIRMMP)

Alexandre Bonvin (Univ. of Utrecht)

Zeynep Kurkcuoglu (Univ. of Utrecht)

Ales Krenek (CESNET)

Mario David (LIP)

Jesus Marco (IFCA-CSIC)

Fernando Aguilar (IFCA-CSIC)

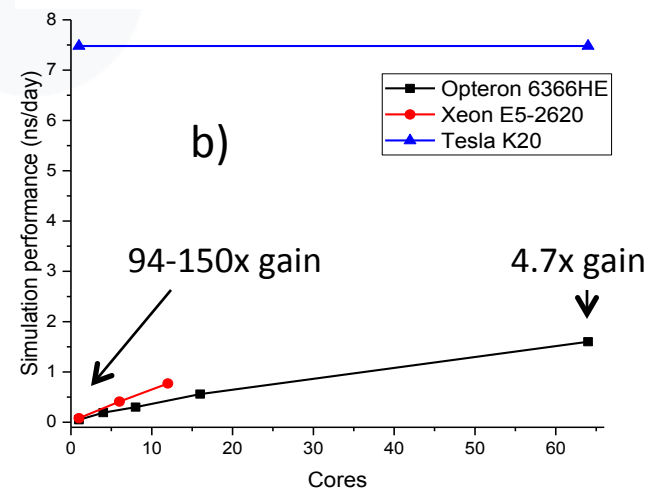
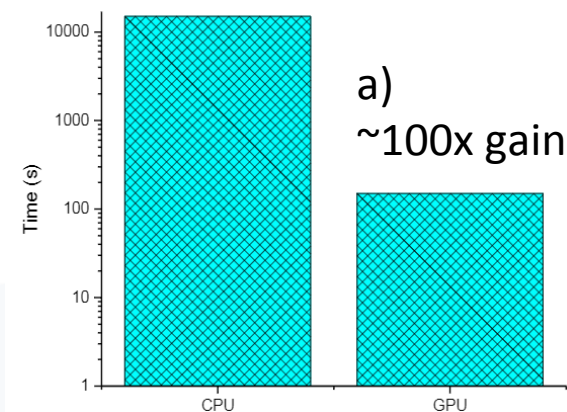
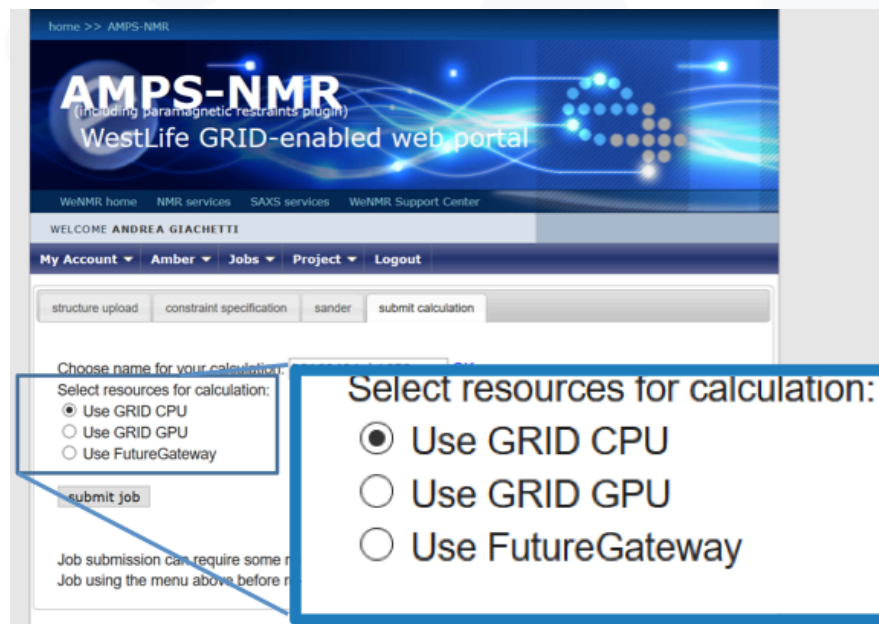
Slides from <http://bit.ly/Amsterdam-GPU-Session>



- MD simulations with **AMBER**
- MD simulations with **GROMACS**
- **DisVis**: visualisation and quantification of the accessible interaction space of distance restrained binary biomolecular complexes
- **PowerFit**: automatic rigid body fitting of biomolecular structures in Cryo-Electron Microscopy densities
- Full report in D6.7 EGI-Engage deliverable:
 - <http://bit.ly/EGI-D67>

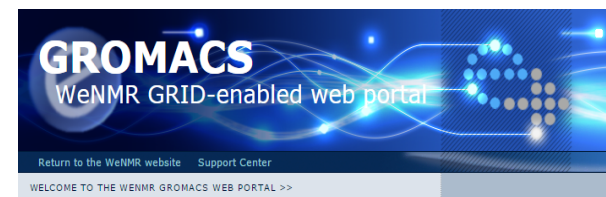
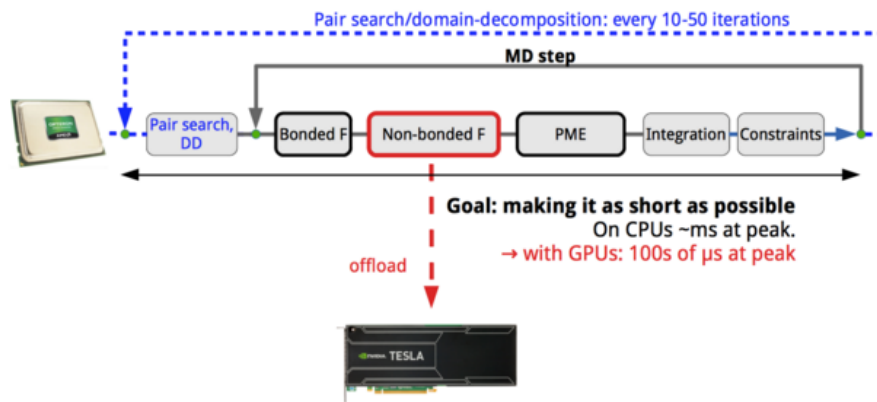
a) Restrained (rMD) Energy Minimization on NMR Structures

b) Free MD simulations of ferritin



WeNMR/AMBER grid portal can now exploit GPU resources

- GPU acceleration introduced in version v4.5
 - Grid portal runs it in multi-threading mode
 - No significant cloud overhead measured for GPU speedups

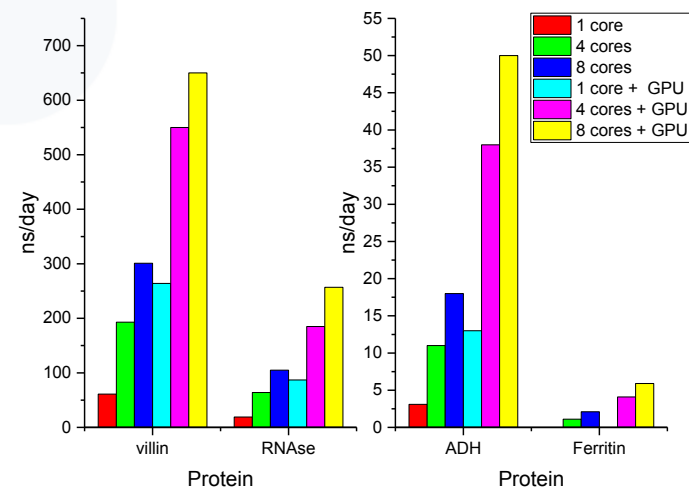


THE GROMACS WEB SERVER

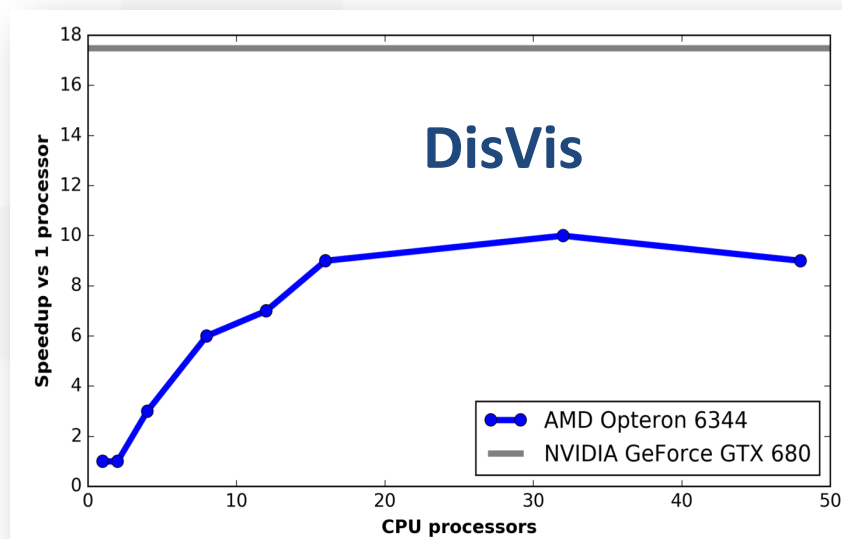
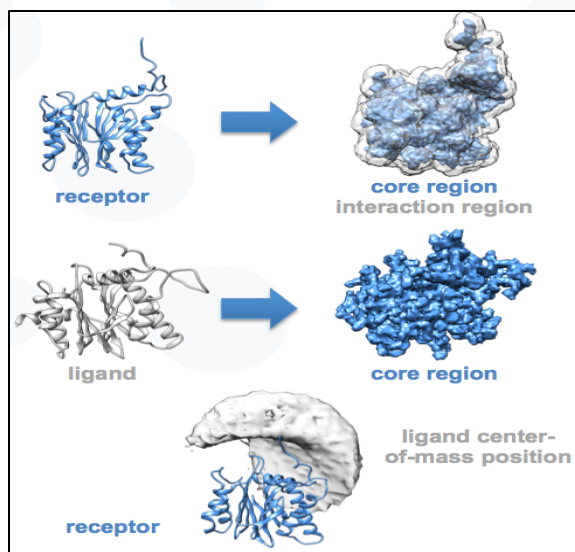
Welcome to the GROMACS web server your entry point for molecular dynamics on the GRID. GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. GROMACS is able to work with many biochemical molecules like proteins, lipids and nucleic acids. The WeNMR GROMACS web portal combines the versatility of this molecular dynamics package with the calculation power of the eNMR grid. This will enable you to perform many simulations from the comfort of your internet browser anywhere in the world. The server is furthermore aimed to provide a user friendly and efficient MD experience by performing many preparation and optimization steps automatically.



Dataset	Protein size (aa)	Simulation performance in ns/day						GPU Acceleration		
		1 core	4 cores	8 cores	1 core + GPU	4 cores + GPU	8 cores + GPU	1 core	4 cores	8 cores
Villin	35	61	193	301	264	550	650	4.3x	2.8x	2.2x
RNAse	126	19	64	105	87	185	257	4.6x	2.9x	2.4x
ADH	1,408	3.1	11	18	13	38	50	4.2x	3.5x	2.8x
Ferritin	4,200	-	1.1	2.1	-	4.1	5.9	-	3.7x	2.8x

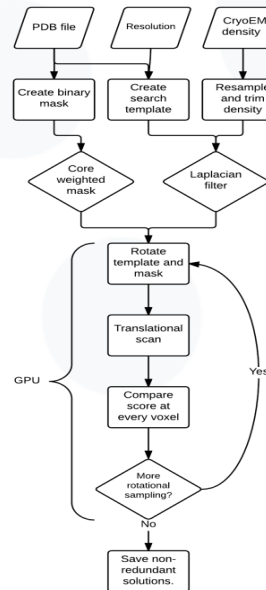
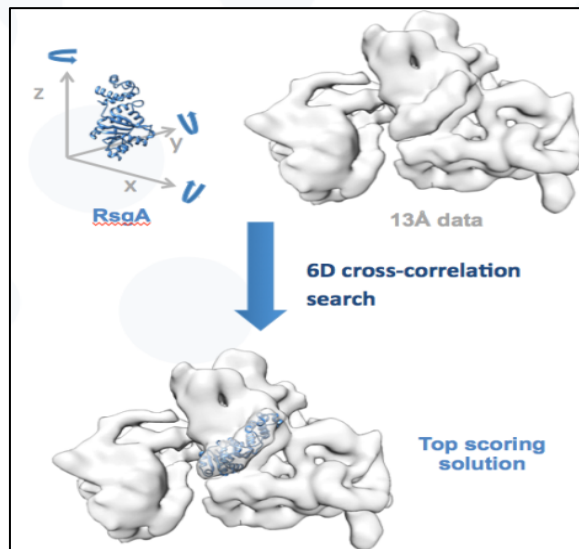


- Visualisation and quantification of the accessible interaction space of distance restrained binary biomolecular complexes
- Bare metal benchmarks with UU lab hardware



System	Number of complexes sampled	CPU time (m)	GPU time (m)	Acceleration
RNA polymerase II	$19 \cdot 10^9$	1,184	56	21x
PRE5-PUP2	$7 \cdot 10^9$	432	15	29x

- Automatic rigid body fitting of biomolecular structures in Cryo-Electron Microscopy densities
- Bare metal benchmarks with UU lab hardware



$$CW-LCC = \frac{1}{N} \frac{\sum_i^N \rho_c^n \cdot w_i \rho_o}{\sqrt{(\overline{\rho_o^w})^2 - (\rho_o^w)^2}}$$

$$CW-GCC = \mathcal{F}^{-1} [\mathcal{F} (w \rho_c^n)^* \times \mathcal{F} (\rho_o)]$$

$$(\overline{\rho_o^w})^2 = \mathcal{F}^{-1} [\mathcal{F} (w)^* \times \mathcal{F} (\rho_o)]^2$$

$$(\rho_o^w)^2 = \mathcal{F}^{-1} [\mathcal{F} (w^2)^* \times \mathcal{F} (\rho_o^2)]$$

**Fast Fourier Transform
for fast translational
scans**

System	Map size (voxels)	Rotations sampled	CPU time (s)	GPU time (s)	Acceleration
GroEL-GroES	90 x 72 x 72	70,728	5,340	249	21x
RsgA into ribosome	72 x 80 x 72	70,728	4,560	242	19x

DisVis and PowerFit on EGI platforms

Requirements:

- **Basic:**
 - Python2.7
 - NumPy 1.8+
 - SciPy
 - GCC (or another C-compiler)
- **Optional for faster CPU version:**
 - FFTW3
 - pyFFTW
- **Optional for GPU version:**
 - OpenCL1.1+
 - pyopencl
 - cIFFT
 - gpyfft

Solution for grid and cloud computing:

**Docker containers
built with proper libraries and
opencl support:**

Base dockerfile with opencl:

<https://github.com/indigo-dc/docker-opencl>

Dockerfile for PowerFit:

<https://github.com/indigo-dc/docker-powerfit>

Dockerfile for DisVis:

<https://github.com/indigo-dc/docker-disvis>

Baremetal vs grid vs cloud

ID	Type	GPU	#Cores CPU type	Mem (GB)
B-K20	Baremetal	Tesla K20	24 HT (12 real) Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz	32
B-K40	Baremetal	Tesla K40	48 HT (24 real) Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz	512
D-K20	Docker on K20	Tesla K20	24 Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz	32
K-K40	KVM on K40	Tesla K40	24 Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz	32

Courtesy of Mario David
INDIGO

Case	Machine	TimeGPU (sec)	TimeCPU 1 core	CPU1/GPU
B-K40	Baremetal	674	7928	11.8
K-K40	KVM	671	7996	11.9
B-K20	Baremetal	830	11839	14.3
D- K20	Docker	837	11926	14.3

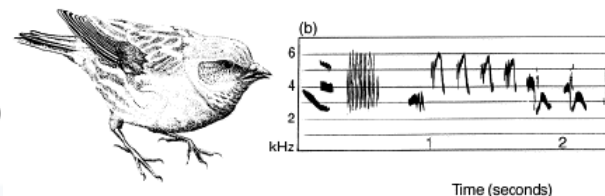
<- Cloud

<- Grid

No loss of performance



- LifeWatch is the European e-Science infrastructure for Biodiversity and Ecosystem Research (ESFRI)
- ANN & Pattern Recognition Tools can be applied in many cases:
 - Bird recognition (by sound)
 - Satellites data (land type, land use, water...)
 - Species classification



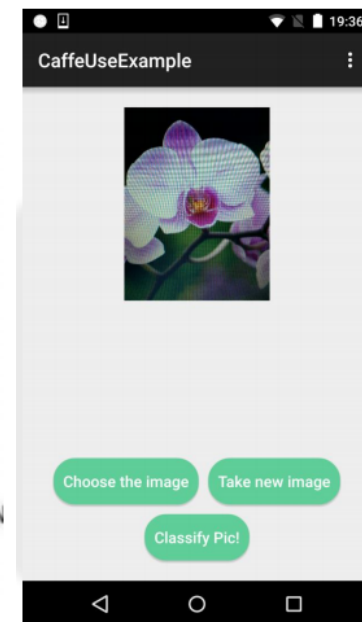
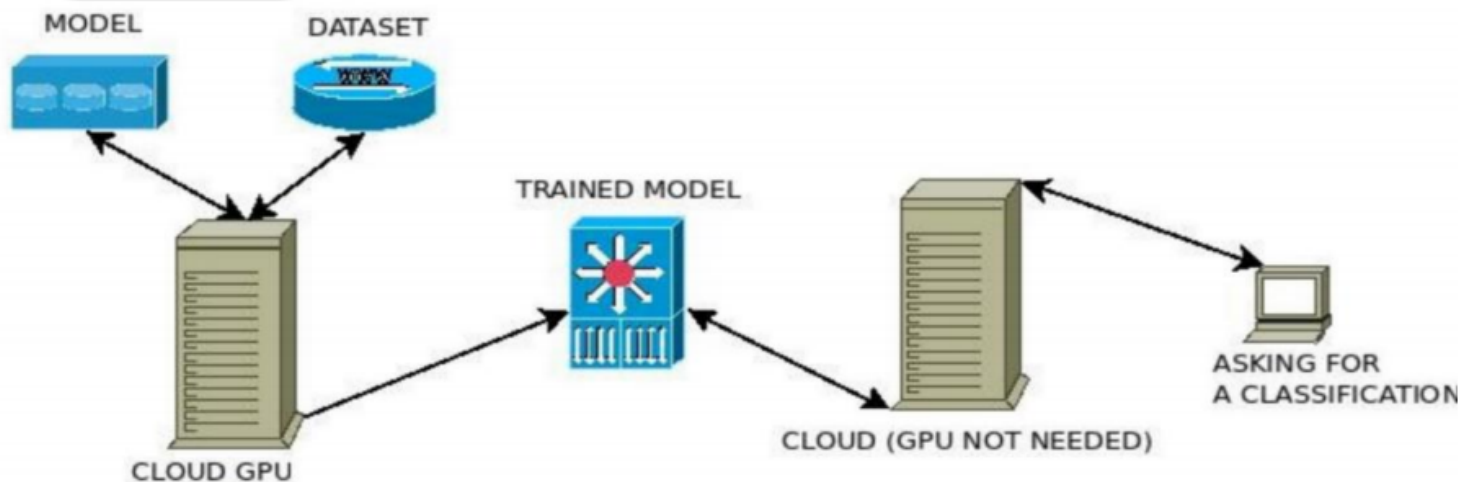
- Due to different features, like memory bandwidth or architecture, GPUs get much better performance in training ANN than CPUs
- They adopt Caffe: one of the most popular deep learning frameworks, implemented in pure C++/CUDA

<http://caffe.berkeleyvision.org>



Figure 1: cuDNN performance comparison with CAFFE, using several well known networks. CPU is 16-core Intel Haswell E5-2698 2.3 GHz with 3.6 GHz Turbo. GPU is NVIDIA GeForce GTX TITAN X.

- ANN on image recognition for photos taken with mobiles (see <http://bit.ly/Bari-Lifewatch>)
- Prototype based on Caffe framework trained with some flora images
- Deployed at IISAS and at Seville cloud site with Tesla K20m GPUs to be integrated in EGI FedCloud



Thank you for your attention.

Questions?



www.egi.eu

EGI-Engage is co-funded by the Horizon 2020 Framework Programme
of the European Union under grant number 654142

