Federated Accelerated Computing Platforms for EGI

Marco Verlato

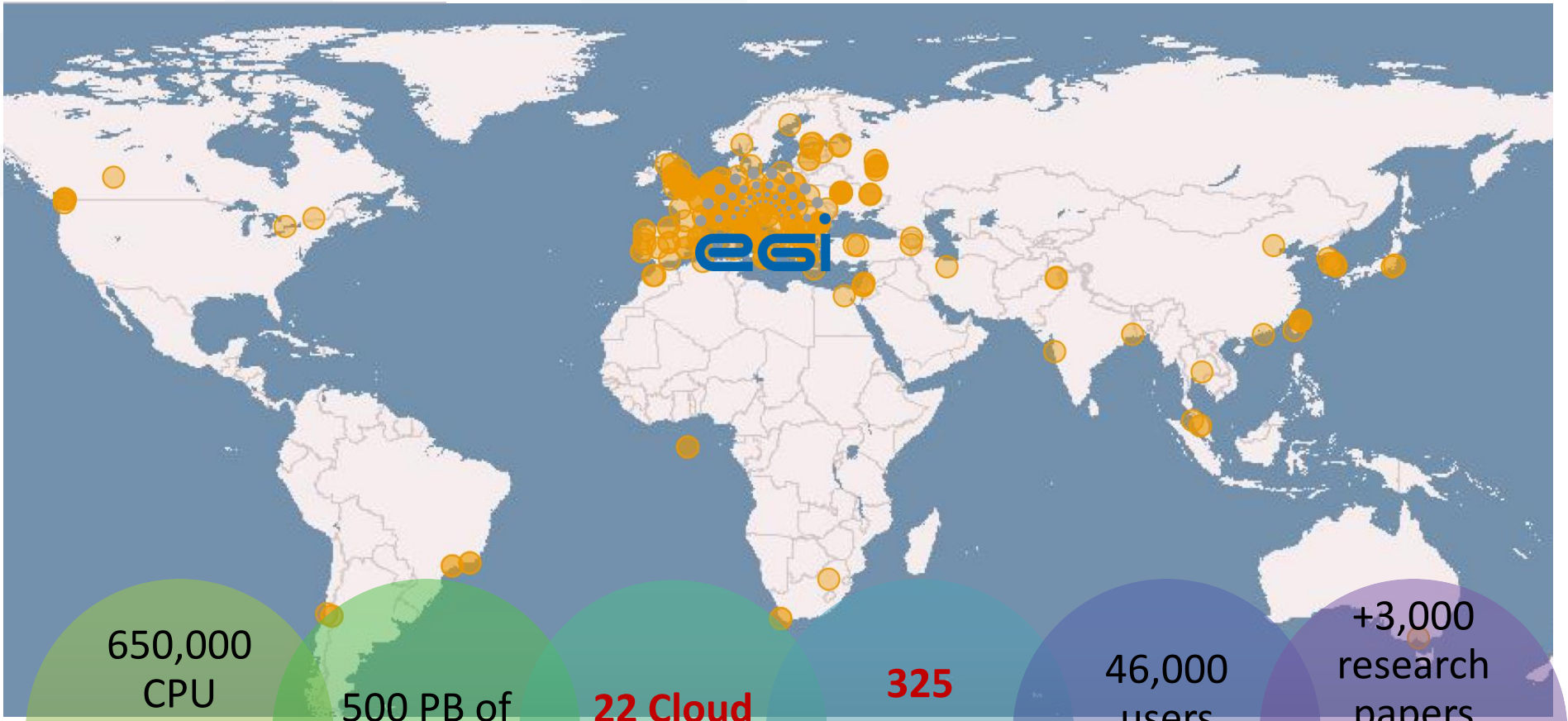
INFN-Padova

Digital Infrastructures for Research 2016

28-30 September 2016, Krakov, Poland

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

EGI federated infrastructure, QR1 2016



650,000 CPU Cores
+23.6%

500 PB of storage

22 Cloud providers

325 resource providers

46,000 users
+21%

+3,000 research papers
> 780 in 2016

Introduction/1

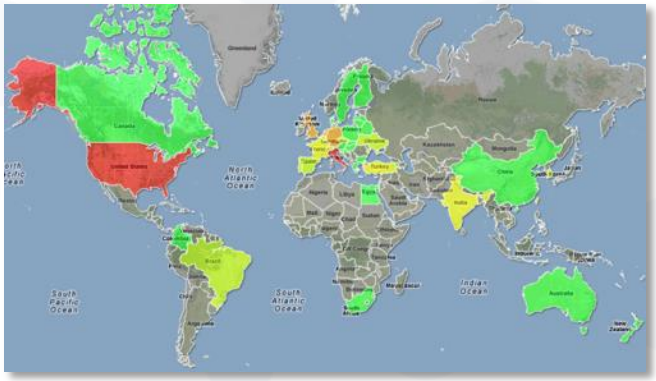
- EGI-Engage is an H2020 project supporting the EGI infrastructure
  - Has a task for "**Providing a new accelerated computing platform**"
- Accelerators:
  - **GPGPU** (General-Purpose computing on Graphical Processing Units)
    - NVIDIA GPU/Tesla/GRID, AMD Radeon/FirePro, Intel HD Graphics,...
  - Intel Many Integrated Core (MIC) Architecture
    - Xeon Phi Coprocessor
  - Specialized PCIe cards with accelerators
    - DSP (Digital Signal Processors)
    - FPGA (Field Programmable Gate Array)
- Partners involved in the task:
  - **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)

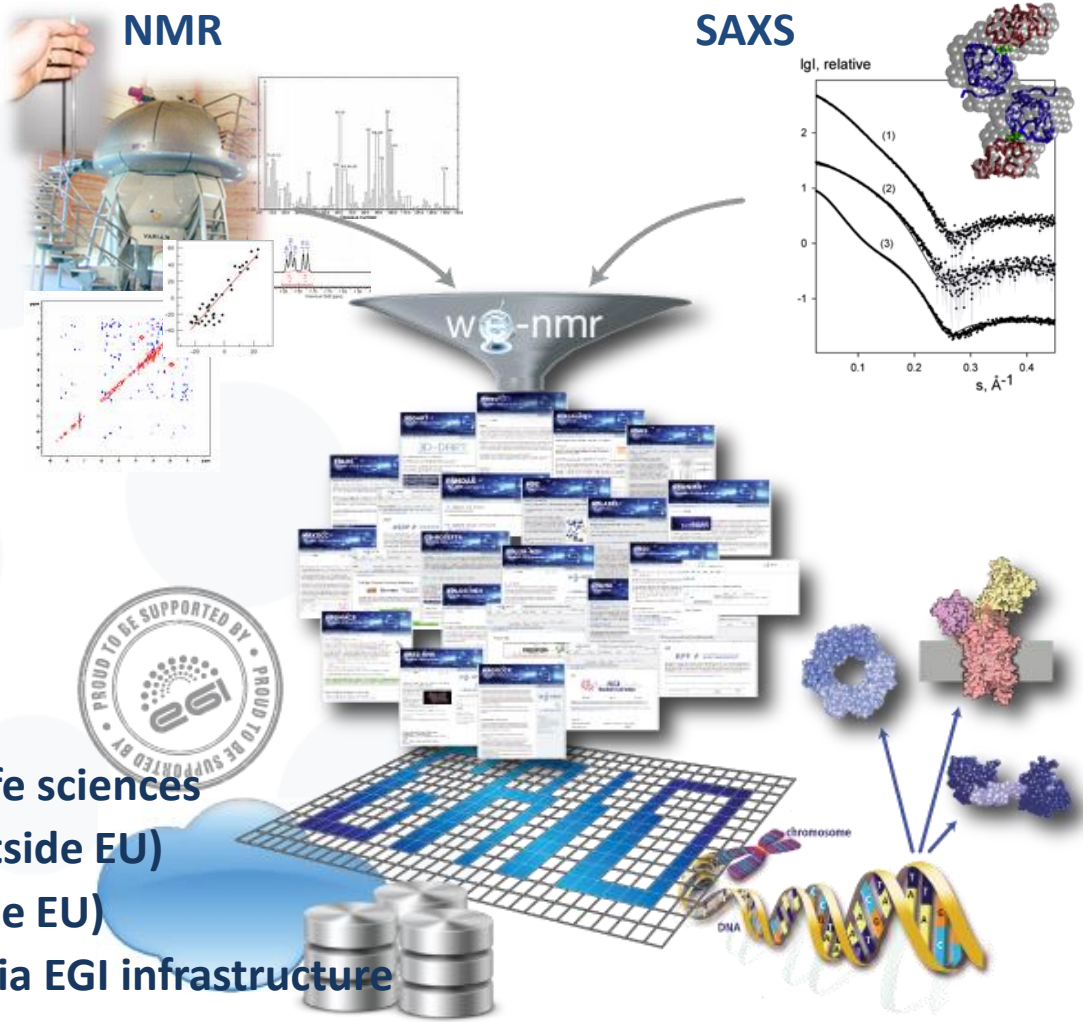
Introduction/2

- Goal of the task:
  - **To 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**…
  - **To 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 were collected at EGI Conference in May 2015: see http://bit.ly/Lisbon-GPU-Session

EGI-ENGAGE

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
- > 3M jobs/year, > 30M HS06.hrs/year
- User-friendly access to Grid via web portals

e-nmr → we-nmr → MoBrain West-Life

2007 2010 2013 2015 2018

Driven by the user communities

- EGI-Engage has a Competence Center to Serve Translational Research from Molecule to Brain: MoBrain (core partners also involved in WeNMR/West-Life and INDIGO-DataCloud projects)
  - Dedicated task for "GPU portals for biomolecular simulations"
  - To deploy the **AMBER** and **GROMACS** MD packages on GPGPU test beds and build web portals for their use
  - To develop GPGPU-enabled web portals for:
    - exhaustive search in Cryo-EM density (→ **PowerFit**)
    - quantifying the information content of inter-subunit distance restraints, determined for example by using CXMS technique (→ **DisVis**)
- Requirements
  - GPGPU resources for development and testing (at 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)

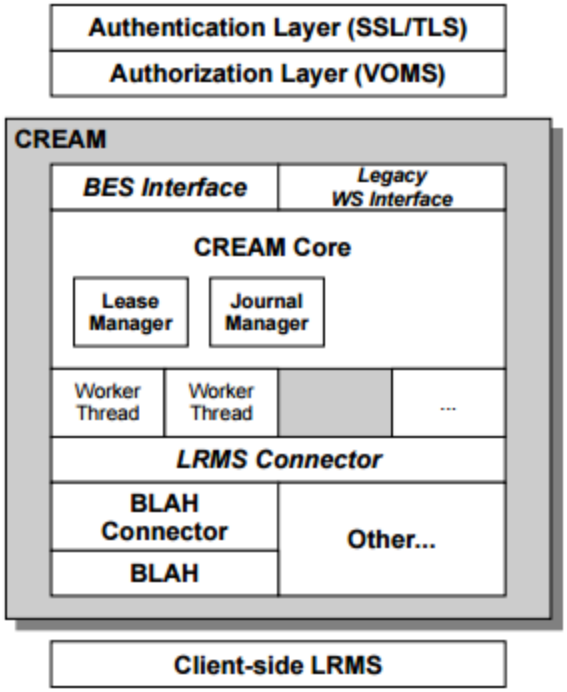
Accelerated Computing in Grid

Paolo Andreetto (INFN-PD)
David Rebatto (INFN-MI)
Marco Verlato (INFN-PD)
Lisa Zangrando (INFN-PD)
Antonio Rosato (CIRMMP)
Andra Giachetti (CIRMMP)

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

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 (and 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
  - Indentifying the relevant GPGPU-related parameters supported by the different LRMSes, and abstract ther 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



Job submission/1

- Started in 2015 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 and OpenCL
  - 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
  - GPU-enabled CREAM-CE prototype

- **First step**: testing local job submission with the different GPGPU supported options, e.g. with Torque/pbs_sched:

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

- ...and with Torque/Maui:

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

Job submission/2

- **Second step**: defining the new JDL attribute "**GPUNumber**" and implement it in CREAM Core and BLAH components
- The first GPGPU-enabled CREAM prototype working with the CIRMMP Torque/Maui cluster was implemented the 1st December 2015
- Example of JDL for running the DisVis application (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";
stderror = "err.err";
outputsandbox = { "out.out" , "err.err" , "res-gpu.tgz"};
GPUNumber=1;
]
```

Job submission/3

- **Third step**: investigating GPGPU and MIC supported options for the other LRMSes (HTCondor, LSF, Slurm, SGE)
- 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 by HTCondor, LSF and Slurm LRMSes
- New CREAM prototypes supporting when possible both GPU and MIC cards was successfully implemented and tested in 2016

Job submission/4

- A **CREAM/HTCondor** prototype supporting both GPUs and MIC cards was successfully implemented and tested at **GRIF/LLR** data centre in March 2016 (thanks to A. Sartirana)
- A **CREAM/SGE** prototype supporting GPUs was successfully implemented and tested at **Queen Mary** data centre in April 2016 (thanks to D. Traynor)
- A **CREAM/Slurm** prototype supporting GPUs was successfully implemented and tested at **ARNES** data centre in April 2016 (thanks to B. Krasovec)
- A **CREAM/LSF** prototype supporting GPUs was successfully implemented and tested at **INFN-CNAF** data centre in July 2016 (thanks to S. Dal Pra)
- A **CREAM/Slurm** prototype supporting GPUs was successfully implemented and tested at **Queen Mary** data centre in August 2016 (thanks again to D. Traynor)
  - With Slurm Version 16.05 which supports the GPUModel specification

Infosys/1

- **GLUE2.1 draft** as starting point for Accelerator-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 (MIC) 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

Infosys/2

- 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_temperature
=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_temperature
=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

Accounting

- 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, LSF and other LRMSes 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
```

EGI-ENGAGE

Summary

- CREAM GPU-enabled prototype available and tested at 5 sites who hosted 5 different LRMSes
  - 3 new JDL attributes defined: GPUNumber, GPUModel, MICNumber
- New classes and attributed describing accelerators proposed and included in GLUE2.1 draft after discussion with the OGF WG
- A **major release of CREAM** is scheduled by the end of 2016
  - with GPU/MIC support for Torque, LSF, HTCondor, Slurm, and SGE LRMSes
  - with the GLUE2.1 draft prototype as information system
    - future official approval of GLUE 2.1 would occur after the specification is revised based on prototype lessons learned
  - on CentOS7, in order to be included in UMD-4 release

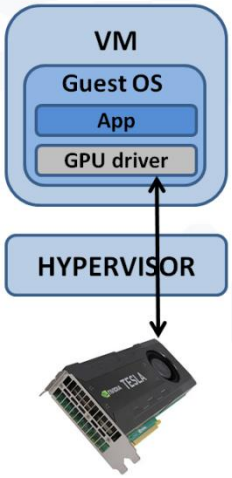
Accelerated Computing in Cloud

Viet Tran (IISAS)
Jan Astalos (IISAS)
Miroslav Dobrucky (IISAS)

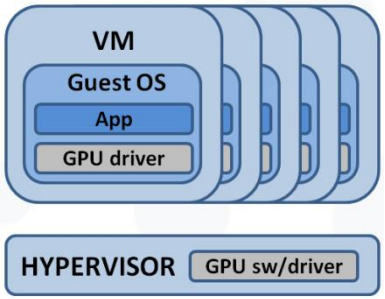
EGI-Engage is co-funded by the Horizon 2020 Framework Programme
of the European Union under grant number 654142

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 a early stage:
    - NVIDIA GRID vGPU (XenServer, VMWare hyperv. only)
    - SR-IOV based AMD MxGPU (VMWare hyperv. only)



- Cloud framework support
  - Openstack support for PCI passthrough
  - OpenNebula support for PCI passthrough from v4.14
- EGI Federated Cloud services support
  - Information system
  - Accounting

IISAS-GPUCloud site

- 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 Federated Cloud 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

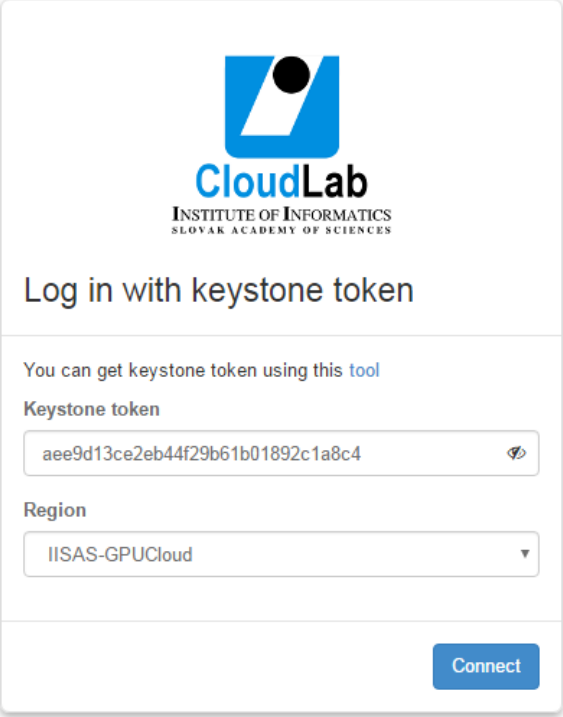
EGI Federated Cloud integration/1

- Results
  - Fully working OpenStack based cloud site with GPGPUs
  - VMs reach native performance (around 2% differences)
  - Exact, repeatable crashing scenarios and workarounds
- **OpenStack/Kilo** site fully certified and integrated with EGI Federated Cloud 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, dteam, moldyngrid, enmr.eu, vo.lifewatch.eu

EGI Federated Cloud integration/2

- 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



EGI Federated Cloud integration/3

- GPGPU support in **OpenNebula** based cloud
  - A multi-purpose PCI passthrough capabilities were introduced in OpenNebula version 4.14
  - CESNET-MetaCloud site upgraded in **May 2016** with 4 NVIDIA Tesla M2090 cards available
  - It is considered still experimental and UNSTABLE (i.e., just for testing purposes)
  - Plans to provide templates and user guides for GPU-enabled virtual machines (as done for IISAS-GPGPUCloud)
  - The long-term goal is to provide OCCI extensions to select these "additional" capabilities for virtual machines on a case-by-case basis (not just by using a pre-defined template)

Information System and Accounting

- 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 homogeneus virtual accelerator devices, who can be associated to one or more CloudComputingInstanceTypes
- GPU accounting easier in cloud environment (1 VM ←→ 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

Work in progress

- Experimental cloud site set up at IISAS to enable GPGPU support with LXC/LXD hypervisor with OpenStack
  - LXC/LXD is a full container solution supported by Linux
  - Expected to provide better performance and stability than KVM (must faster startup time, better integration with OS), especially in terms of GPGPU support (simpler site setup, more stable than KVM PCI passthrough)
- New cloud sites joining soon the EGI Federated Cloud
  - At INCD/LIP OpenStack PCI passthrough with NVIDIA Tesla K40 GPUs
  - At Seville (CSIC-EBD-LW), supporting LifeWatch community with ~500 cores and 1 PB of storage, and NVIDIA Tesla K20m GPUs
  - And a new OpenNebula based cloud site at IISAS
- IISAS will provide supports/helps for installation and configuration

Applications

Antonio Rosato (CIRMMP)
Andra Giachetti (CIRMMP)
Alexandre Bonvin (Univ. of Utrecht)
Zeynep Kurkcuoglu (Univ. of Utrecht)
Jörg Schaarschmidt (Univ. of Utrecht)
Mikael Trellet (Univ. of Utrecht)
Ales Krenek (CESNET)
Mario David (LIP)
Jesus Marco (IFCA-CSIC)
Fernando Aguilar (IFCA-CSIC)

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

Molecular Dynamics

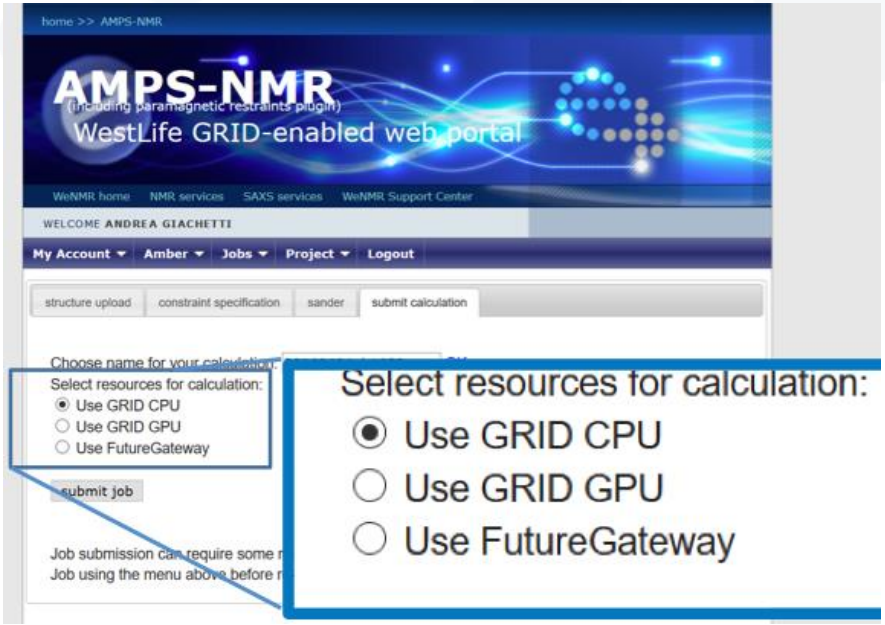
- MD simulations with **AMBER**
- MD simulations with **GROMACS**
- **DisVis**: visualisation and quantification of the accessible interaction space of distance restrained binary biomolecular complexes
- **PowerFit**: automatic rigid body fitting of biomolecular structures in Cryo-Electron Microscopy densities

- Full report in D6.7 EGI-Engage deliverable:
  - http://bit.ly/EGI-D67

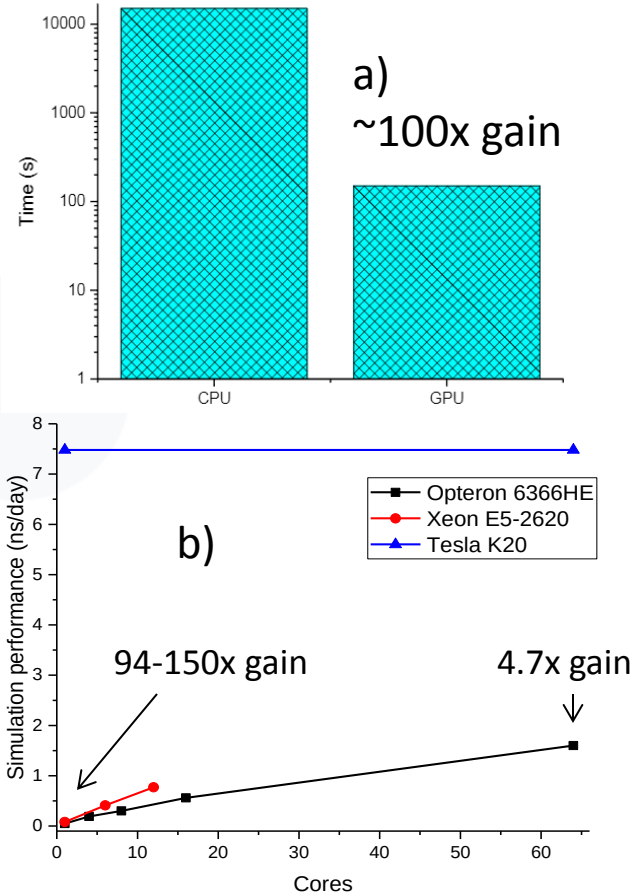
AMBER

a) Restrained (rMD) Energy Minimization on NMR Structures

b) Free MD simulations of ferritin

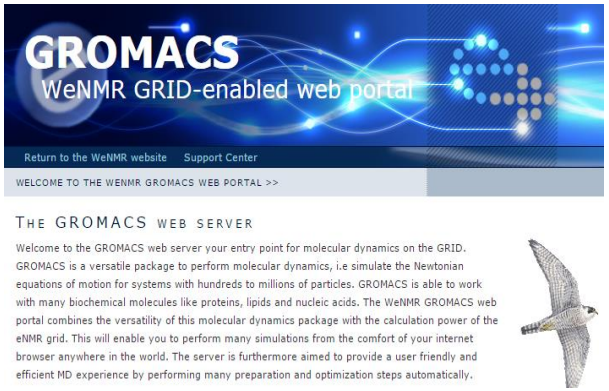


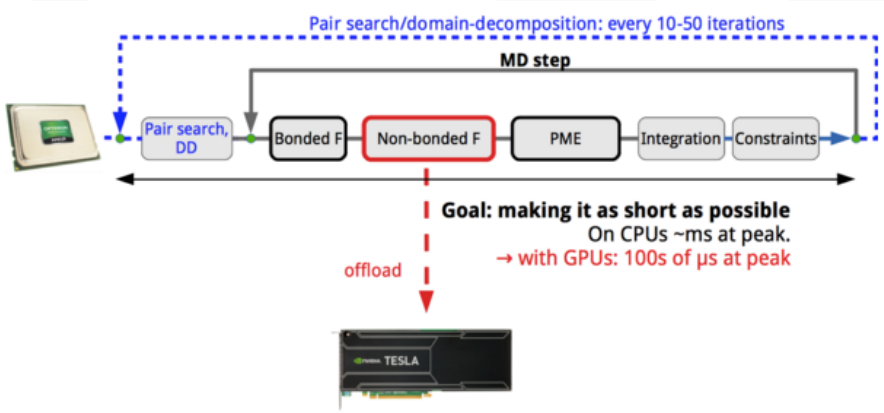
WeNMR/AMBER grid portal can now exploit GPU resources



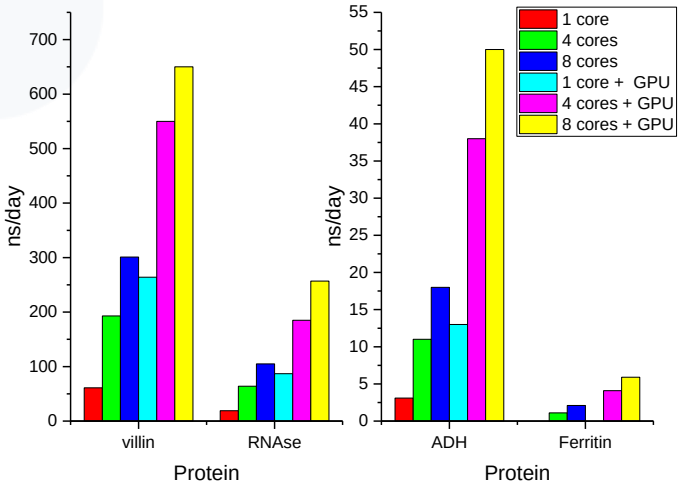
GROMACS

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





		Simulation performance in ns/day						GPU Acceleration		
Dataset	Protein size (aa)	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



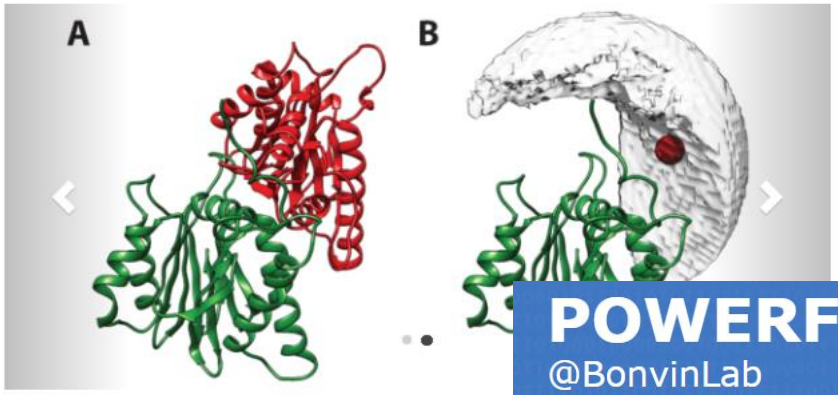
DISVIS
@BonvinLab

HADDOCK CPORT **DISVIS** POWERFIT PRODIGY 3D-DART BONVIN

About Submit Register Example Help/Manual Support

WELCOME TO THE DISVIS WEBSERVER! >>

DisVis visualizes the accessible interaction space!



DisVis allows you to visualize and quantify the information content of dista complexes.
It performs a full and systematic 6 dimensional search of the three translational and number of complexes consistent with the restraints. In addition, it outputs the perce represents the center-of-mass position of the scanning chain corresponding to the hi in space.

DisVis webserver

REGISTRATION: To use the DisVis server you must have registered for an account. **here**

Submit your job to:
- DISVIS GPU accelarated Grid server
- DISVIS server

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

POWERFIT
@BonvinLab

HADDOCK CPORT DISVIS **POWERFIT** PRODIGY 3D-DART BONVIN LAB

About Submit Register Example Help/Manual Support

WELCOME TO THE POWERFIT WEBSERVER! >>

PowerFit fits your 3D structures in any map!

PowerFit automatically fits high-resolution atomic structures into cryo-EM densities.
To this end it performs a full-exhaustive 6-dimensional cross-correlation search between the atomic structure and the density. It takes as input an atomic structure in PDB-format and a cryo-EM density with its resolution; and outputs positions and rotations of the atomic structure corresponding to high correlation values. PowerFit uses the local cross-correlation function as its base score. The score can optionally be enhanced by a Laplace pre-filter and/or a core-weighted version to minimize overlapping densities from neighboring subunits.

PowerFit webserver

REGISTRATION: To use the PowerFit server you must have registered for an account. If you do not have an account yet you can **register here**

Submit your job to:
- POWERFIT server
- POWERFIT GPU accelarated Grid server

Reference for use of the server

Powered by

West-Life

bioexcel

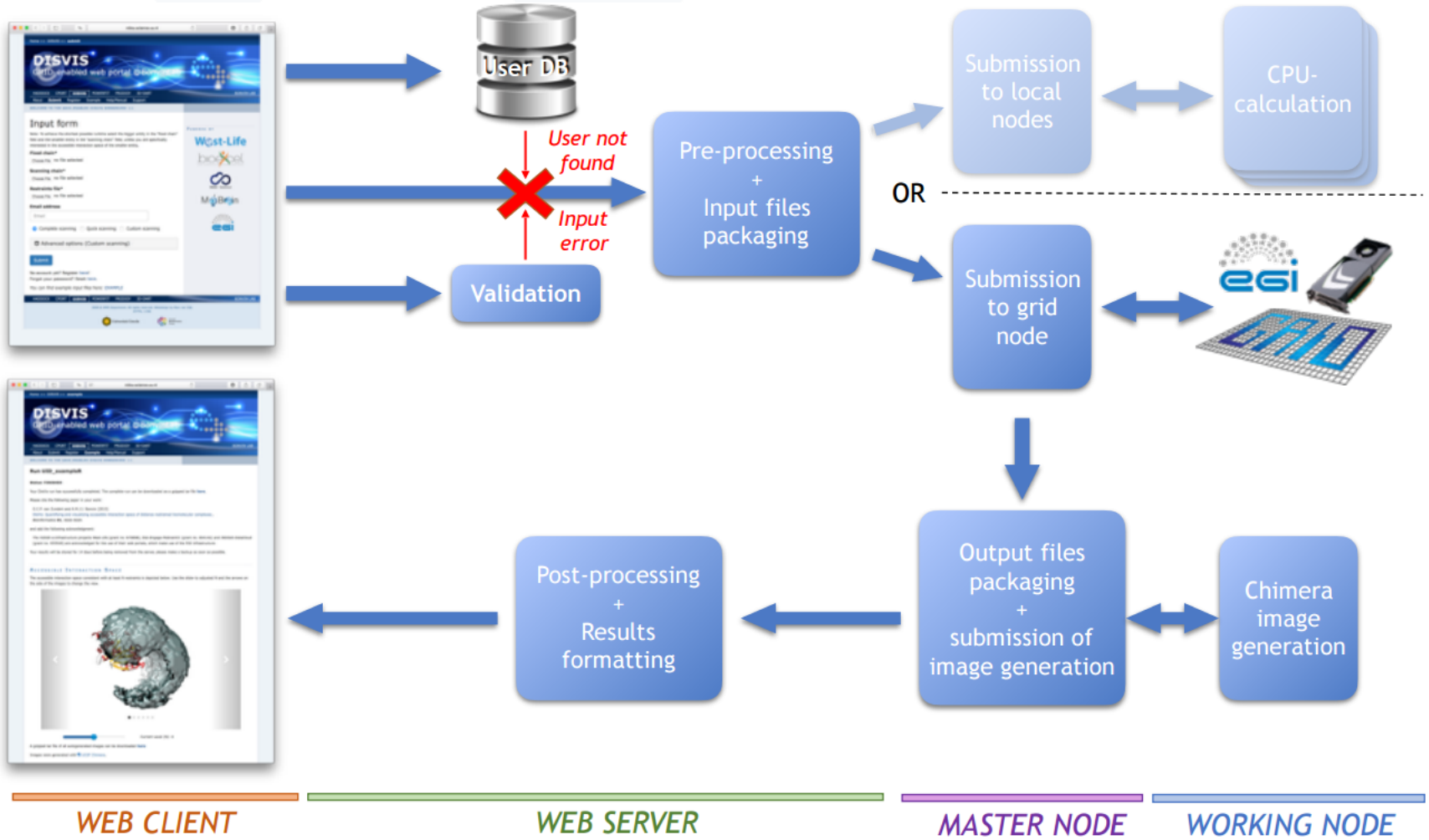
INDIGO - DataCloud

MoBrain

egi

Open to users in August 2016

Grid-enabled web portal architecture



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:

INDIGO - DataCloud

Docker containers built with proper libraries and OpenCL support:

Base dockerfile with NVIDIA driver:
https://github.com/LIP-Computing/ansible-role-nvidia

Dockerfile for DisVis and PowerFit:
https://github.com/indigo-dc/ansible-role-disvis-powerfit

Docker engine not required on grid WNs: use udocker tool to run docker containers in user space (https://github.com/indigo-dc/udocker)

GPGPU-enabled CREAM-CE at CIRMMP

disvis.sh job example

```
#!/bin/sh
driver=$(nvidia-smi | awk '/Driver Version/ {print $6}')
export WDIR=`pwd`
git clone https://github.com/indigo-dc/udocker
cd udocker
./udocker.py pull indigodatacloudapps/disvis:nvdrv_$driver
rnd=$RANDOM
./udocker.py create --name=disvis-$rnd indigodatacloudapps/disvis:nvdrv_$driver
mkdir $WDIR/out
./udocker.py run -v /dev --volume=$WDIR:/home disvis-$rnd "disvis \
/home/O14250.pdb /home/Q9UT97.pdb /home/restraints.dat -g -a 5.27 -vs 1 \
-d /home/out"
./udocker.py rm disvis-$rnd
./udocker.py rmi indigodatacloudapps/disvis:nvdrv_$driver
cd $WDIR
tar zcvf res-gpu.tgz out/
```

Driver identification

Install udocker tool

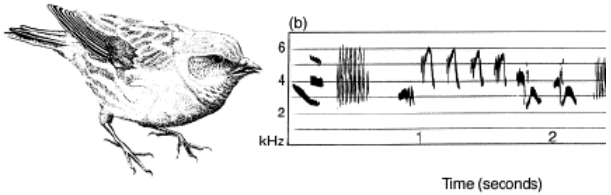
Pull DisVis image

Create the container

Run the container executing DisVis

Biodiversity

- 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

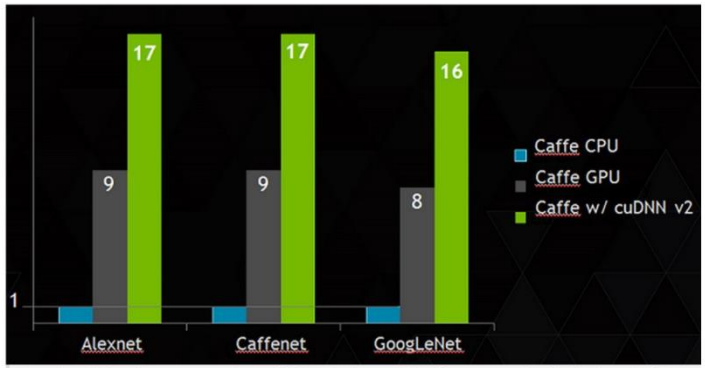
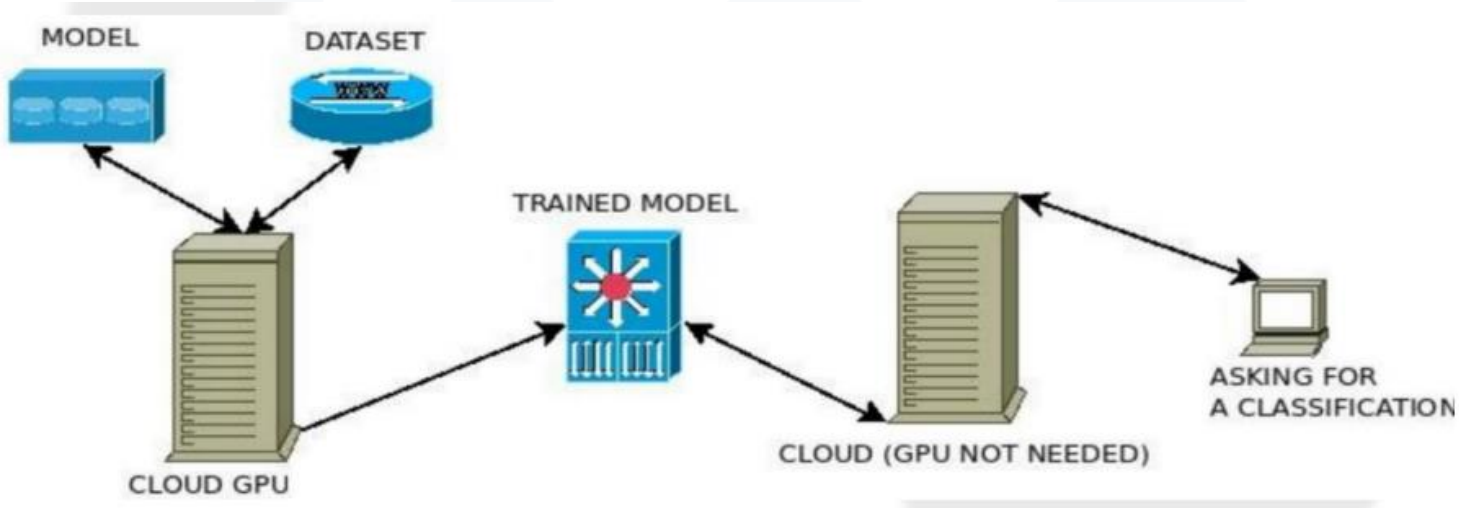
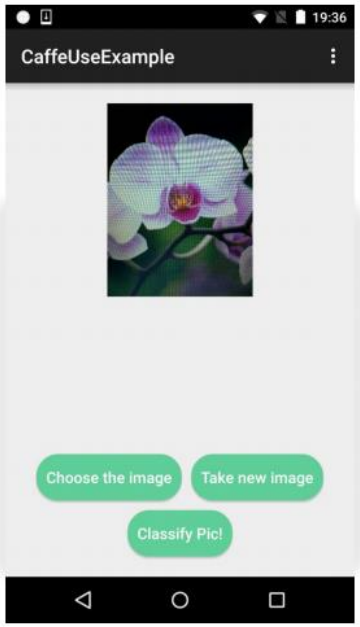


Figure 1: cuDNN performance comparison with CAFFE, using several well known networks. CPU is 16-core Intel Haswell E5-2698 2.3 GHz with 3.6 GHz Turbo. GPU is NVIDIA GeForce GTX TITAN X.

Flower recognition

- 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?

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