romeoLAB, le portail web HPC : cas d'utilisation pour la pédagogie et les logiciels à la demande

Dr Arnaud RENARD - HPC CTO & Research Engineer
Dr Jean-Matthieu ETANCELIN - HPC Research Engineer
Pr Michael KRAJECKI - HPC Center Director

romeolab@univ-reims.fr
@HPCromeo
#romeoHPC
University of Reims
Université de Reims Champagne-Ardenne (URCA)

Multidisciplinary university
- about 27,000 students
- 5 campuses: Reims, Troyes, Charleville-Mézières, Chaumont et Châlons-en-Champagne
- a wide initial undergraduate studies program
- graduate studies and PhD program linked with research labs
Its **mission** is to deliver, for both **industrial and academic** researchers:

- **high performance** computing **resources**,  
- **secured storage** spaces,  
- specific & **scientific softwares**,  
- **advanced user support** in exploiting these resources,  
- **in-depth expertise** in different engineering fields: HPC, applied mathematics, physics, biophysics and chemistry, ...

**CLASSICAL MISSIONS**

- promote and diffuse HPC and **simulation** to companies / **SMB**  
- identify, experiment and master **breakthrough technologies**  
  - which give new **opportunities** for our user  
  - from **technology-watching** to **production**  
  - for all **research domains**

**SPECIFIC MISSIONS**

- **Teaching High Performance Computing to Researchers and Students**
### Principales machines de production

<table>
<thead>
<tr>
<th>Année / Constructeur</th>
<th>2013 / Bull</th>
<th>2018 / Bull-ATOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investissement</td>
<td>2,5 M€</td>
<td>6,5 M€</td>
</tr>
<tr>
<td>CPU</td>
<td>2 080 cœurs 700 000 cœurs GPU (260 K20X)</td>
<td>3 220 cœurs (skylake @2,6 GHz) 1 000 000 cœurs GPU (280 P100)</td>
</tr>
<tr>
<td>Puissance</td>
<td>254,0 Tflops</td>
<td>1,022 Pflops x 4</td>
</tr>
<tr>
<td>Energie</td>
<td>70 KW</td>
<td>120 KW</td>
</tr>
<tr>
<td>Mémoire</td>
<td>4 To</td>
<td>15 To</td>
</tr>
<tr>
<td>Stockage</td>
<td>200 To (//)</td>
<td>600 To (//)</td>
</tr>
<tr>
<td>Serveurs</td>
<td>130</td>
<td>115</td>
</tr>
<tr>
<td>Interconnexion</td>
<td>40 Gb/s IB</td>
<td>100 – 200 Gb/s BXI</td>
</tr>
</tbody>
</table>

- GPU
- romeoLAB
- TOP500 #151
- GREEN500 #5
- GRAPH500 #105

- DLI & DL
- TOP500 #254
- GREEN500 #20
- HPCG #63
- GRAPH500 #??
Necessity of computer science courses dedicated to parallel programming on heterogeneous architectures. (Students, Reserchers, European & French projects)

• HPC is more and more complex (different processors, memory, network, ...)
• Hardware specific phenomenon (bandwidth bottleneck, cache page issues, ...)

➢ We want to execute code on a real HPC facility.

Romeo works perfectly, got software already installed and supported.

As an example, we organized a 5 days event dedicated to GPU technologies in 2016

But using HPC supercomputer is difficult:
- Ssh, Sftp (+installing clients),
- Password, login, account opening account process,
- Load module environment,
- Deal with Slurm,
- Xserver or VNC client for Graphical interface (profiler, debugger, viewers, ...),

Using those tools:
- is time-consuming;
- is not pedagogic objectives of courses;
- are obstacles to the pedagogical process efficiency;

Motivation: romeoLAB must be powerful
A web-based solution is the most easy-to-use solution.
- Users already have a browser
- Lot of references / other experiences
- HTML5 allow almost everything
- Multi-device
- Poor internet connections

User management
- Simple registration process
- Disposable accounts
- Access to session with access code
- Start and Stop labs thru SLURM jobs / reservation
Fonctionnalités :

- romeoLAB
  - Orienté HPC
  - Execution sur notre cluster de prod (2500 coeurs / 260 GPU), slurm (reservation)
  - Cours (session, via token, persistance) = n * exercice (Lab) = n * notebook
  - Contenus dans GIT
  - Interface Enseignant
In romeoLAB, (like modern MOOC - Massive Open Online Course), development environments is completely integrated into pedagogical content.

- We use Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.
Motivation: romeoLAB must be Pedagogic

On the same web-page:

- development environments
  - File browser & Editors (native & NodeMirror)
  - Compilation, command lines & modules loading (native)
  - Execution live (native)
  - Execution in batch for large runs (in-house Ipython-batch-execution-magics):
    24 students can each run a 32-nodes MPI runs with 32 nodes in total
- Desktop access to run Graphical software
  - Profilers, Debuggers, Graphical interfaces like Paraview
    (in-house integration with VNC + x11vnc + VirtualGL)
- Pedagogical content
  - Pdf, Video, content, images, iframed web pages ... (native or in-house)


Romeo: THE MOST AMAZING GPU CLUSTER IN EUROPE
Pr Krajecki - GTC2016

Hosted at University of Reims Champagne-Ardenne.
Designed by Atos-Bull.
Powered by 260 GPU Tesla Cards
Ranked 151@Top500 in Nov. 2013

Romeo is dedicated to research, professional services and education, in the GrandEst Region, France

2,5 M€, funded by

- GPU Accelerated Librarais (cuBLAS, cuFFT, cuRand, ...)
- Multi-GPU Programming with OpenACC
- Advanced Multi-GPU: rCUDA and GPU Direct RDMA
- OpenFoam Tutorial
- OpenMP for HPC, Python programming
- Caffe for DeepLearning
The Jupyter NoteBook is running on the compute node (protected mode)
All the management of *romeoLAB* is made on the web server (FrontEnd)
We need to develop our own server because existing one (like jupyterHub) do not correspond to our needs of customization, integration, evolutivity, dedicated to teaching ... Written in PHP + MVC + RedBean + MariaDb + ...

**Users** can reach and leave sessions, start and stop labs, ...
**Teachers** can create sessions or labs, update labs contents, and manage students.
User view of accessing an interactive content in romeoLAB:
1. The user creates an account, and log in to the platform
2. The user reach an active Session (=classroom) with the access code given by teacher (or provided by an activation link, provided by email, on a webpage or Paypal)
3. The user can list available labs and their description to finally start one lab and reach his IPython Notebook running on one compute node. He can watch videos and documents, fill table with performance results, edit, compile and also profile code via a remote desktop.
User view of accessing an interactive content in romeoLAB:

1. The user creates an account and logs in to the platform.
2. The user reaches an active session (=classroom) with the access code given by the teacher (or provided by an activation link, provided by email, on a webpage or Paypal).
3. The user can list available labs and their description to finally start one lab and reach his IPython Notebook running on one compute node. He can watch videos and documents, fill a table with performance results, edit, compile and also profile code via a remote desktop.
**Internal behavior** when a user starts a lab:

4. Server assign a **temporary cluster-user** to the user and dynamically load initial lab content (from the lab repository with GIT protocol).

5. It launches a job through the **workload scheduler** (and possibly via reserved dedicated resources).

6. This job setting up all resources parameters (available ports), starts all services (notebook, editors, VNC server, ...)

7. romeoLAB is waiting / probing the start of those services

8. **Proxy routes** are setup (Main proxy) to provide direct access to these services (we use “Configurable HTTP Proxy”, which is a NodeJs tool)
Demo

TODO:
une video de 1 minute?
### Previous events and Sessions

<table>
<thead>
<tr>
<th>Name</th>
<th>Attendees</th>
<th>Duration</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2017:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JDEV2017</td>
<td>24</td>
<td>4 days</td>
<td>GPU programming</td>
</tr>
<tr>
<td>Groupe Calcul</td>
<td>38</td>
<td>3 days</td>
<td>Advanced Python for HPC</td>
</tr>
<tr>
<td>Profiler Days</td>
<td>17</td>
<td>3 days</td>
<td>Profiling tools for parallel codes</td>
</tr>
<tr>
<td><strong>2016:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Master Courses</td>
<td>40</td>
<td>4 months</td>
<td>GPU programming and HPC</td>
</tr>
<tr>
<td>OpenFOAM School</td>
<td>20</td>
<td>3 days</td>
<td>OpenFOAM software</td>
</tr>
<tr>
<td>10th LoOPS day</td>
<td>60</td>
<td>2 days</td>
<td>C++ (HPX) vs Python (DSLs)</td>
</tr>
<tr>
<td>GTCEU2016</td>
<td>55</td>
<td>90 minutes</td>
<td>MPI and OpenACC</td>
</tr>
<tr>
<td>GPU Spring School</td>
<td>36</td>
<td>1 week</td>
<td>GPU programming</td>
</tr>
<tr>
<td>GTC2016</td>
<td>60</td>
<td>90 minutes</td>
<td>Advanced tools for hybrid cluster</td>
</tr>
</tbody>
</table>

**2017/2018:**
- linux / Gaussian / Molden / OpenFOAM / industriels / VI-HPS / Abaqus
**usage:**

Available content

**romeoLAB** is addressing a wide range of technologies and audience levels. As we encourage mutualization, this list is growing:

**Beginner :**
- Introduction to: Python, OpenMP, MPI, CUDA, OpenACC, ...
- GPU accelerated applications: CUDA, OpenACC, Python, ...
- GPU accelerated libraries: cuBLAS, cuRand, cuFFT, ...

**Intermediate**
- OpenFOAM
- Gaussian / Molden
- OpenCL, CUDA Asynchronism

**Advanced**
- Profiling: TAU, MAQAO
- Advanced Python: Cython, Numba, Pythran
- CUDA Optimizations
- Multi-GPU with CUDA
- Multi-GPU with OpenACC and MPI
- Multi-GPU with rCUDA & GPU-Direct RDMA
Powerfull: we want to execute code on a real HPC facility, because ours works perfectly and we’ve got all the software already installed and supported. Where’re using a proxy for external users to access to compute nodes.

Easy: romeoLAB is a modern MOOC platform making it possible to run HPC in a simple web browser. Ssh, ftp and job managers (Slurm) are not part of courses educational objectives nor prerequisites.

Pedagogic: on the same web page, student must find lessons (video, pdf, images, ...) and the edition / compilation / execution interfaces. Jupyter Notebook is our solution for strong interactivity. Teacher can create his courses on the same platform and manage attendees to his courses,

Multi-Application: Compiling and executing code is not enough. We must run profilers, GUIs, and other scientific software.
Applications IGMPlot


Réaction chimique SN₂

Interaction ligand-protéine
IGMPlot On-demand

Your identity

Email: eric.henon@univ-reims.fr
Name: Henon
Running ID: lab00093

Join a new Session

Session code to join: CODE

Centre de Calcul
ROMEO
Maison de la SIMULATION
de Champagne-Ardenne

IGMPlot on demand

IGMPlot is a free open-source program developed to identify molecular interactions and prepare data to build 2D and 3D representations of them. "IGMPlot on demand" allows you to perform an IGM (and NCI) calculations online and to download the results. Please refer to the igmplot details. The IGMPlot tool was published here (PCCP) and here (Chem. Phys. Chem.).

Preliminary step

Prepare the needed input files before going further. Please refer to the documentation made available at http://igmplot.univ-reims.fr to prepare the calculations.

Perform a new calculation

Click on the "Perform a new calculation" button

Get the results

Click on the "Reload status" link to refresh the current job status.

When finished click on the corresponding link to access the details and results of your calculation; click ONCE!

There is no guarantee that IGMPlot on-demand will work. No computing or storage services are guaranteed. The authors are not liable for any damages suffered as a result of using this service.

New on-demand calculation

Perform a new calculation
IGMPlot On-demand

DEMO
romeoLAB : actualités

15 – 19 octobre

2 – 4 février

19 – 20 septembre

8 novembre
romeoLAB : access single nodes (DGX - QLM - ...)
Perspectives romeoLAB

- Maintenir
- Version distribuable
  - nouveau supercalculateur
  - partenaires / opensource
  - rename
- Sécurité
- OpenStack & Single machie
romeolab@univ-reims.fr
Quantum computing

- Solve difficult problems
  - Classical bit VS Qubits

- QLM Group (partners on right)
  - Quantum simulator platform
  - Develop new algorithms (BD, AI, SC, Cyber security)
  - Designing computing architectures
  - Quantum safe cryptography algorithms

Quantum simulator platform