

**EGI-Engage**

Scipion cloud deployment for MoBrain

D6.14

|  |  |
| --- | --- |
| **Date** | 21 November 2016 |
| **Activity** | WP6 |
| **Lead Partner** | CNB - CSIC |
| **Document Status** | DRAFT |
| **Document Link** | <https://documents.egi.eu/document/2982> |

Abstract

Scipion is a powerful and versatile framework that allows researchers to obtain 3D maps of macromolecular complexes combining the best from the most popular 3DEM software package. We have deployed the Scipion framework into the EGI Federated Cloud and have tested the deployments with real Electron Microscopy user cases.

Now end users can try the features of Scipion without installing it, thanks to the Scipion Web Tools virtual machine available in the EGI Federated Cloud.

They can also access the full power of Scipion, leveraging the EGI Federated Cloud computing resources with the easy-to-deploy ScipionCloud appliance.

**COPYRIGHT NOTICE**



This work by Parties of the EGI-Engage Consortium is licensed under a Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/). The EGI-Engage project is co-funded by the European Union Horizon 2020 programme under grant number 654142.

**DELIVERY SLIP**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Name*** | ***Partner/Activity*** | ***Date*** |
| **From:** | Laura del Cano | CNB – CSIC / WP6 | 30/11/2016 |
| **Moderated by:** | Malgorzata Krakowian |  |  |
| **Reviewed by** | Gergely Sipos | EGI.eu / WP6 | 18/11/2016 |
| **Approved by:** |  |  |  |

**DOCUMENT LOG**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Issue*** | ***Date*** | ***Comment*** | ***Author/Partner*** |
| **v0.9** | 10/11/2016 | First full draft | Laura del Cano / CNB-CSIC |
| **V1** | 14/11/2016 | Draft ready for external review | Jesús Cuenca-Alba / CNB-CSIC |
| **V2** | 17/11/2016 | Completed document metadata | Malgorzata Krakowian / EGI.eu |
| **V3** | 21/11/2016 | Modified upon reviewers comments | Jesús Cuenca-Alba / CNB-CSIC |
| **Final** |  | Final version for approval |  |

**TERMINOLOGY**

A complete project glossary and acronyms are provided at the following pages:

* <https://wiki.egi.eu/wiki/Glossary>
* <https://wiki.egi.eu/wiki/Acronyms>

Additional acronyms, specific to this document:

|  |  |
| --- | --- |
| ***Acronym*** | ***Description*** |
| **3DEM** | 3D Electron Microscopy |
| **AWS** | Amazon Web Services |
| **EM** | Electron Microscopy |
| **FedCloud** | EGI Federated Cloud |
| **GPU** | Graphical Processor Unit |
| **HPC** | High Performance Computing |
| **IaaS** | Infrastructure as a Service |
| **SPA** | Single Particle Analysis |
| **SWT** | Scipion Web Tools |

**Contents**

1 Introduction 6

2 Service architecture 9

2.1 High-Level Service architecture 9

1.1.1 Scipion 9

3.1.1 ScipionCloud 11

2.2 Integration and dependencies 12

3.1.2 Mobrain and the EGI Federated Cloud 12

3.1.3 Scipion deployments on CESNET-MetaCloud 13

2.3 Technical challenges 16

3.1.4 Interactive performance 16

3.1.5 Resources management 17

3 Feedback on satisfaction 19

4 Plan for Exploitation and Dissemination 20

5 Future plans 22

5.1 ScipionCloud improvement 22

5.1.1 GPU acceleration 22

5.1.2 Automatic cluster deployment 22

5.1.3 Security 23

5.2 Plans for exploitation 23

3.1 Plans for dissemination 23

6 Conclusions 24

7 References 25

**Executive summary**

Scipion[2] is a powerful and versatile framework that allows researchers to obtain 3D maps of macromolecular complexes combining the best from the most popular 3D Electron Microscopy (EM) software packages, satisfying the complex demands of contemporary EM studies.

We have deployed Scipion in EGI Federated Cloud both in single node and cluster architectures. We tested it with real user cases. We also deployed one running Scipion Web Tools appliance that serves as testbed for novice users who want to try the features of Scipion without installing it, and with no direct access to cloud resources.

In this context of installing Scipion on cloud instances, we faced challenges regarding interactive performance, scalability and resource management. We created ScipionCloud[[1]](#footnote-1), a gateway that solves those challenges, hence allowing easily deploying and accessing the full power of Scipion on commercial and academic clouds.

# Introduction

3D Electron Microscopy under cryogenic conditions (“cryoEM”) was named “Method of the Year” in 2015, for good reason [1]. The introduction of new types of direct electron detectors (that have several times higher detective quantum efficiency -DQE- than previously available ones), together with continuous improvements in computing hardware and in software tools, foresee a promising future for this technique [5].

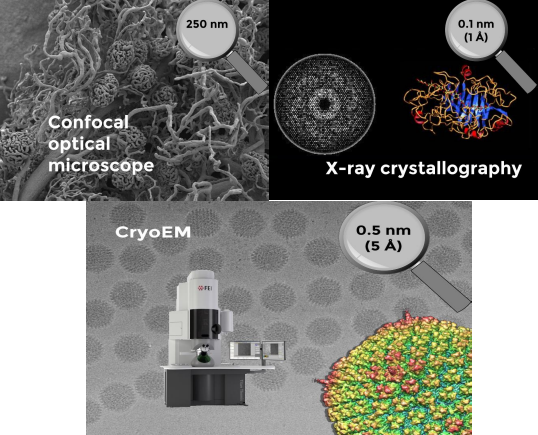


Figure 1 CryoEM allows the high-resolution (5A) reconstruction of a large and flexible macromolecular complexes, without 3D crystals, and from small amount of diluted samples.

One key factor regarding hardware is the high performance of Graphical Processor Units, which is motivating more and more scientific programmers to optimize their software packages for these accelerators.

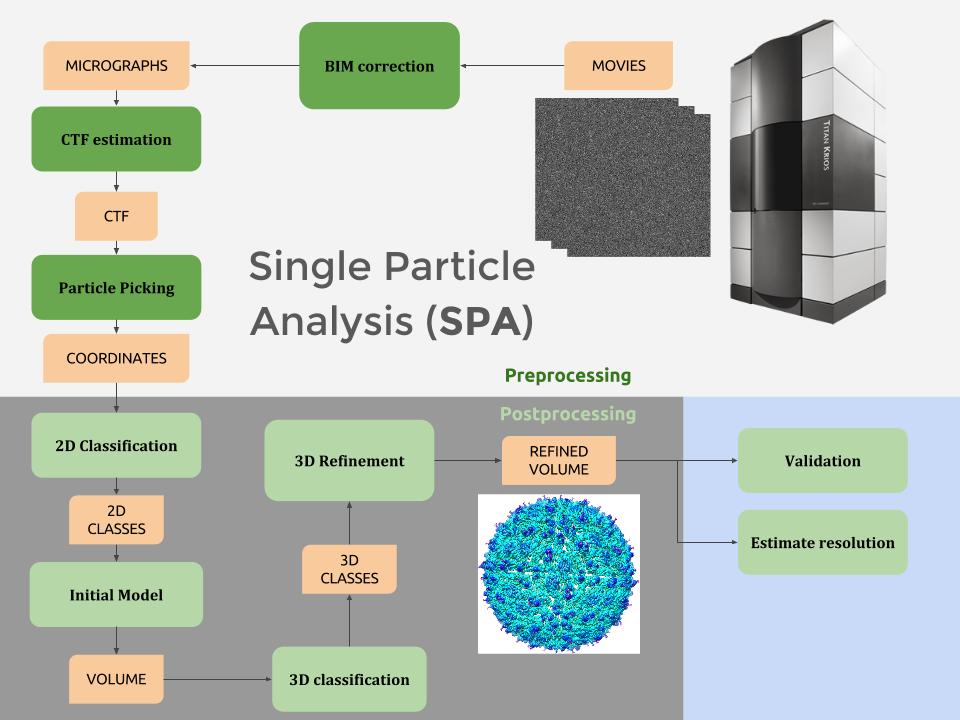
Another key hardware factor is the availability and reduction in net cost of Cloud Computing. Even though the required resources may be substantial, now it is feasible to process whole 3DEM workflows in the cloud.

Figure CryoEM workflows start with a set of movies produced by the Electron Microscope. After 2D and 3D analysis steps, a 3D volume of the specimen is obtained.

Raw computing power and new algorithms, per se, are not enough to satisfy the growing and diverse needs of cryoEM scientists. Their studies demand versatile project management tools that provide adequate support to the increasingly larger and more complex nature of the involved projects.

Scipion is an image processing framework that allows researchers to obtain 3D maps of macromolecular complexes combining the best from the most popular 3DEM software packages. All the low-level details (such as taking care of formats and conversions, or tracking the parameters of each step) are handled by Scipion. This careful management of the details makes easy to share and to repeat workflows, encouraging reproducibility in science.

On this report we present the work that has been done to deploy and test Scipion on the EGI Federated Cloud, and the real EM processing achieved on such deployments. The EGI Federated Cloud [3] is a new type of research e-infrastructure, based on the mature federated operations services that make EGI a reliable resource for science. As an IaaS-type cloud, it is built upon academic private clouds, virtualised resources and open standards.

Based on the experience deploying Scipion on cloud, we created ScipionCloud, a gateway that allows the cryoEM users community to easily deploy Scipion on their cloud of choice. ScipionCloud is available as an EGI Virtual Appliance and as an Amazon Machine Image.

|  |  |
| --- | --- |
| **Tool name** | ScipionCloud |
| **Tool url** | <https://appdb.egi.eu/store/vappliance/scipion.v1.0> |
| **Tool wiki page** | <https://github.com/I2PC/scipion/wiki/ScipionCloud> |
| **Description** | ScipionCloud is a virtual appliance that can be used to instantiate a Virtual Machine on the EGI Federated Cloud with a full installation of Scipion 1.0.1. The Virtual Machine includes a graphical environment that can be efficiently accessed with a web browser. |
| **Value proposition** | ScipionCloud will allow users to have a ready-to-use machine with a preconfigured Scipion installation and a powerful remote desktop solution to ensure the interaction happens as if deployed on the user’s computer. |
| **Customer of the tool** | INSTRUCT (ESFRI); West-Life (backend infrastructure); MoBrain CC; Individual researchers and research teams. |
| **User of the service** | Individual researchers and research teams. |
| **User Documentation** | <https://github.com/I2PC/scipion/wiki/User-Documentation> |
| **Technical Documentation** | <https://github.com/I2PC/scipion/wiki/Developers-Page> |
| **Product team** | Centro Nacional de Biotecnologia (CNB-CSIC) |
| **License** | GPL v2 |
| **Source code** | <https://github.com/I2PC/scipion> |

# Service architecture



## High-Level Service architecture

### Scipion



Scipion is an image processing framework aimed at obtaining 3D maps of macromolecular complexes using cryo Electron Microscopy. It has emerged as a solution offered by the Instruct Image Processing Center (I2PC), allowing cryoEM end users to work with several software packages on the same reconstruction workflow without having to care about formats and conversions.

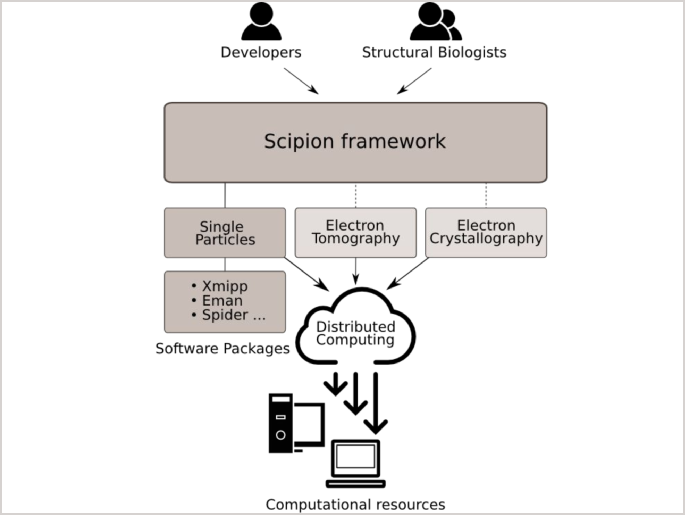


Figure Scipion serves as an interface between users, software packages and computing resources.

End users operate through a workflow editor that exposes a list of protocols, conveniently grouped by categories. When these protocols are executed, all the steps of the workflow are tracked so they can be reproduced later on. Users have the possibility to add or modify steps or parameters. Workflows can even be exported and imported across different projects. Furthermore, the Graphical User Interface (GUI) of Scipion provides a homogeneous access to all the underlying packages, enhanced with rich data viewers, wizards and other practical tools.

Scipion offers two User Interfaces: a desktop interface and a web interface.

The desktop interface supports the whole set of programs and protocols, while the web interface, Scipion Web Tools[[2]](#footnote-2), offers a subset of the protocols that are useful for a first try without any local installation.

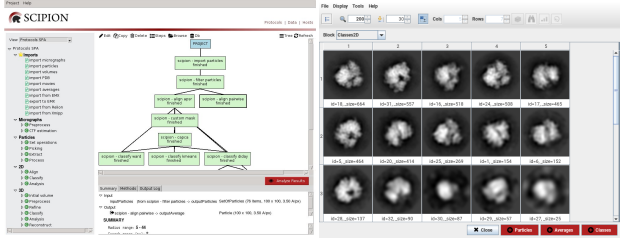


Figure Scipion desktop interface.

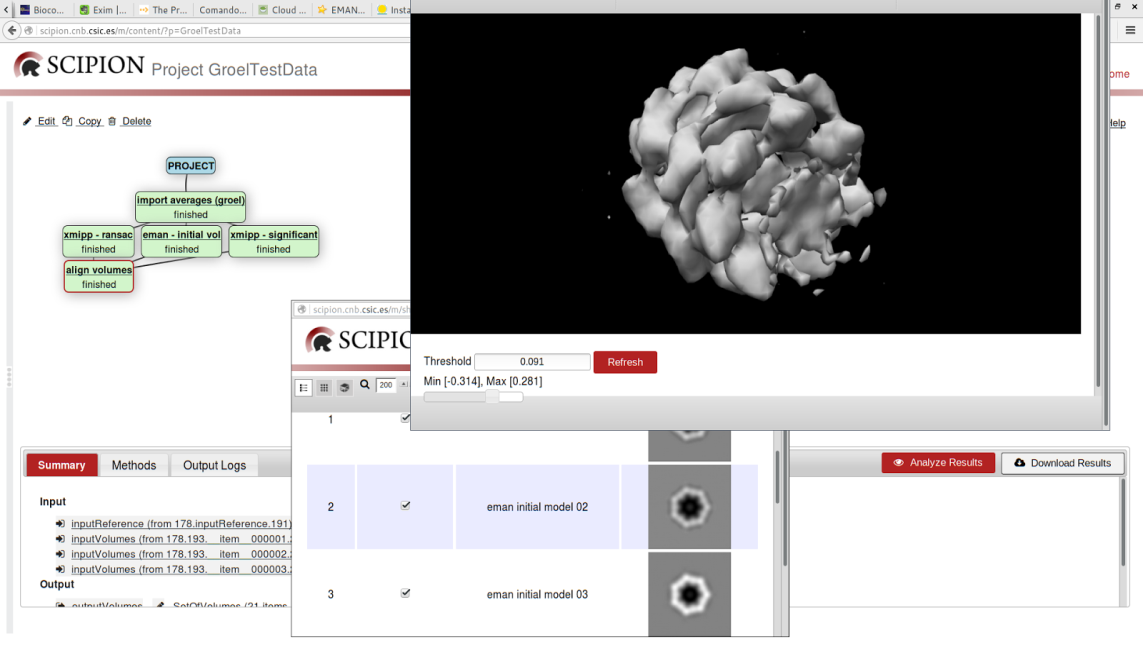


Figure Scipion Web Tools interface.

Current web tools version includes: initial volume estimation, movie alignment, local resolution (ResMap), reliability tools and interaction explorer.

### ScipionCloud

Scipion can be installed from precompiled binaries or built from source code. It offers an automatic mechanism to install the EM packages that are integrated in the framework. Until now the approach described above has been the usual way to install Scipion, both in personal computers and in large clusters, but the arrival of cloud computing has offered new possibilities for software distribution, and it is in this context that ScipionCloud has been developed.

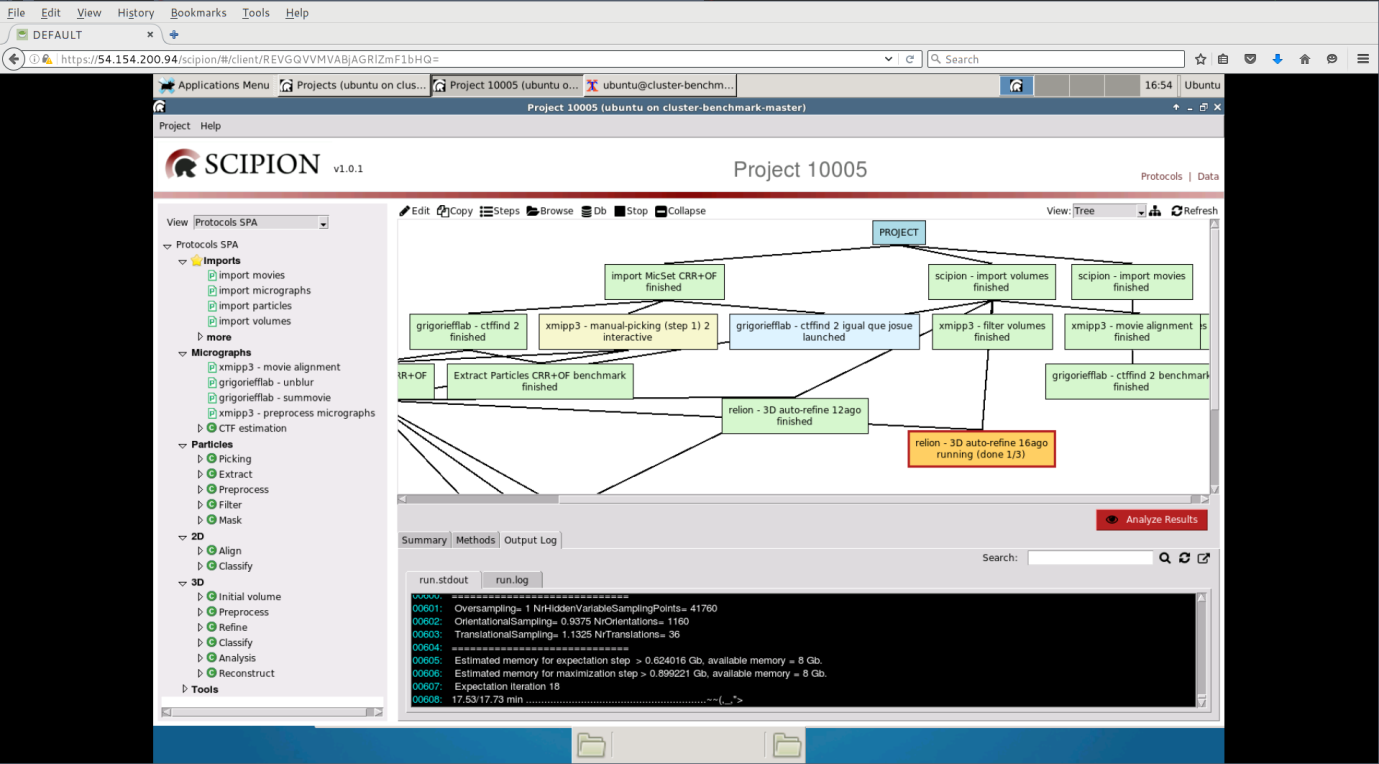


Figure Example of a ScipionCloud session.

ScipionCloud can be easily deployed following the documentation at the Scipion project wiki[[3]](#footnote-3) but essentially, once a user has available a set of cloud resources, the procedure to use ScipionCloud in FedCloud is:

1. Launch a ScipionCloud instance, using the OCCI client[[4]](#footnote-4), either as a standalone program, or from an OCCI virtual appliance. We advise to attach external storage, since cloud images normally have small root disks and Scipion projects usually require quite a lot of storage.
2. Once the instance is running...
   1. Connect to the instance with ssh to grab the randomly-generated remote display password and
   2. Use this password to authenticate in the browser.
3. Upload your data through ssh or rsync. Or download it from the remote desktop.
4. Start Scipion.

Table List of packages and libraries included in the first version of ScipionCloud.

|  |  |  |
| --- | --- | --- |
| ***Category*** | ***Package*** | ***Version*** |
| **EM** | Scipion | 1.0.1 |
| Xmipp | 3.1 |
| Relion | 1.4 |
| Ctffind | 4.0.15 |
| EMAN | 2.12 |
| Motioncorr | 2.1 |
| Spider | 21.13 |
| Bsoft | 1.9.0 |
| Chimera | 1.10 |
| Frealign | 9.07 |
| Resmap | 1.1.5-s2 |
| Summovie | 1.0.2 |
| **System** | Guacamole | 0.9.9 |
| Apache HTTP server | 2.4.18 |
| Apache Tomcat | 7.0.68 |
| XRDP | 0.6.1 |
| OpenMPI | 1.6.5.8 |
| NFS | 4 |
| Mesa | 11.0.7 |
| Starcluster (available only for AWS) | 0.95.6 |
| Nvidia CUDA (currently only on AWS) | 5.5 |
| Glu (currently only on AWS) | 9.0.0 |

## Integration and dependencies

### Mobrain and the EGI Federated Cloud

The EGI Federated Cloud is an IaaS-type cloud, made of academic private clouds and virtualised resources and built around open standards. It offers a wide set of services that can be accessed through different access policies. In the case of the MoBrain Competence Center, an agreement (SLA) has been signed that allows MoBrain to access several computing and storage resources which can be used to further exploit EGI services and kick-start the development of a larger and integrated virtual research environment for life and brain scientists worldwide.

MoBrain users can access the EGI Federated Cloud resources through the Virtual Organization enmr.eu, which is currently the second largest VO in the area of life sciences.

One of the centres involved in the agreement is the CESNET-MetaCloud site in Czech Republic that has agreed to provide a small number (4-6) of cloud (2x 6-8 cores Intel E5) and GPU (nVidia Tesla M2090) nodes to the project for proof-of-concept use. This pool could be extended considerably (up to dozens of such nodes) temporarily for e.g. scalability experiments.

The CESNET-MetaCloud site has been chosen by our team to deploy Scipion due to the high availability of their resources and the quality of their support.

Different scenarios for this deployment are explained in the following section.

### Scipion deployments on CESNET-MetaCloud

#### Scipion single node deployment

We first started to deploy Scipion desktop on a single node to test basic user interaction and performance on the Cloud.

These initial tests were performed on a relatively small machine (*mem\_large*: 4 cores, 16 GB RAM), and evolved to a final installation on a “fat node” (*universe*: 40 cores and 232 GB RAM) with a 1 TB Block Storage attached, where real cryoEM processing could be run.

Besides Scipion v.1.0, remote desktop software, called Guacamole, was installed. A more detailed report on interactive performance problems and remote desktop solutions is presented on section Interactive performance.

This single node virtual machine was used to perform a benchmark of a typical SPA reconstruction workflow compared to local fat node and different setups on Amazon's Elastic Cloud Computing platform.

Currently it is in use by cryoEM scientists in our center to reconstruct a protein fago.

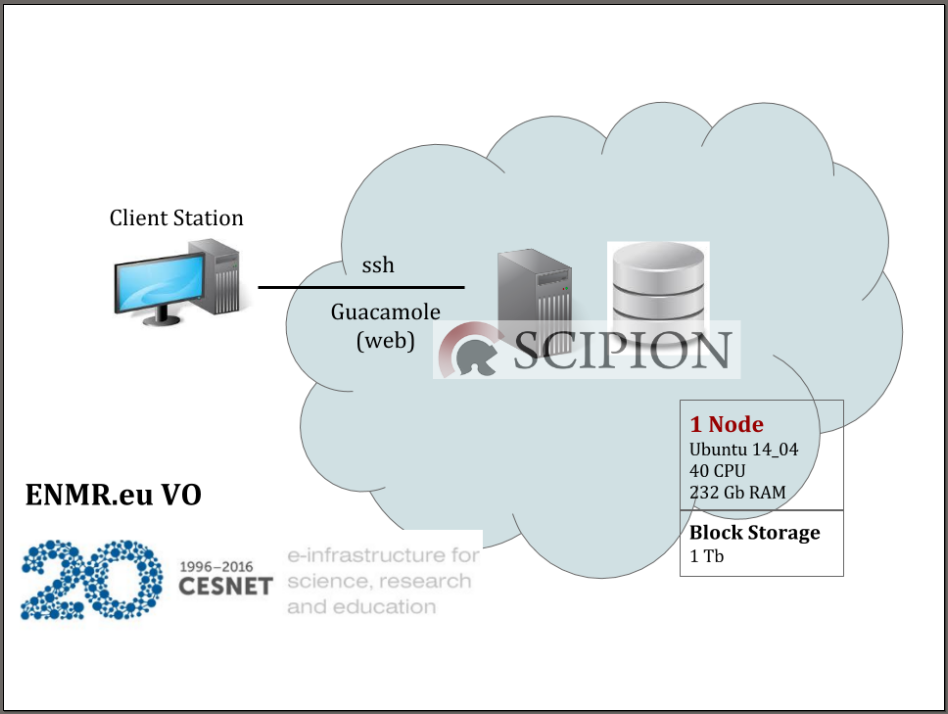


Figure 6 Scipion single node deployment on FedCloud.

#### Scipion HPC cluster deployment

Next step was to deploy Scipion on a cluster of nodes to test the feasibility and performance of this setup.

Figure 7 shows the final deployment on a cluster of 4 nodes of type *mem\_extra\_large* (8 cores and 32 GB RAM) and a 2 TB Block Storage attached.

In this case we tested a different remote desktop solution called X2go. The performance of x2go was similar to guacamole, but x2go had the counterpart of needing software installation in the client computer (while guacamole can be accessed with a standard web browser).

#### Scipion Web Tools deployment

Figure Scipion HPC cluster deployment on FedCloud.

Finally we have deployed the latest version of Scipion Web Tools on a single node of type *mem\_extra\_large* (8 cores and 32 GB RAM), since Web Tools predefined workflows do not require as much computational resources as a typical SPA workflow.

This virtual machine is currently being used as the development environment for the Scipion Web Tools integration on Westlife Virtual Research Environment[[5]](#footnote-5).

## Technical challenges

Figure Scipion Web Tools deployment on FedCloud

In this section we review the different technical challenges faced when installing and testing Scipion, as well as the solutions adopted, which have been integrated on the ScipionCloud image.

### Interactive performance

Commonly, applications developed for web (such as Scipion Web Tools) have an optimized front-end that works well with very low network resources. In the case of Scipion and the EM packages it incorporates, their interfaces are based on X11, which was developed for local area network operation. Therefore, under the cloud environment restrictions (lower bandwidth, higher latency) those interfaces feel too slow for interactive use. This is not a problem specific to EM software, so different generic solutions have been developed to address this issue. In our case we have tested VNC, NX, x2go[[6]](#footnote-6) and Guacamole[[7]](#footnote-7). Although NX, x2go and Guacamole provide similar levels of performance, we have chosen Guacamole since it does not require any software installation on the client side.

Additionally, some applications display 3D graphics based on OpenGL. When such graphics are rendered locally, GPUs can be used to provide real-time performance even with huge and complex volumes. Indeed, that was the initial purpose of the GPU in graphic cards. In cloud context, the rendering is done in the cloud instance where usually there is not a GPU plus a middleware to handle the particularities of the scenario. We are working on integrating such a middle-ware in ScipionCloud but in the meantime, we provide non-accelerated OpenGL support that does not require a GPU.

### Resources management

#### Memory

We found memory to be the current limiting factor in some EM packages, like Relion [6], that sporadically requires large amounts of memory per node (up to 20 GB/MPI process for Expectation steps, and 36 GB/MPI process in Maximization steps, for a typical reconstruction workflow). Therefore, we could not run such programs on our cluster setup; we could only run them on a fat node with enough memory for those most demanding steps.

#### Disk

A typical EM reconstruction workflow starts by processing the movies acquired on the microscope that can easily reach the TeraBytes order. Transferring and storing movies on the virtual machine translates into long times and big storage. One practical solution is to process movies locally (reasonable when using a GPU) and then transfer the resulting micrographs, which are an order of magnitude lower (GigaBytes), to the cloud machine. In any case, to reduce transfer times we recommend using specialized transfer protocols, such as Aspera FASP or bcp, over generic ones, like rsync or scp.

#### Scalability

The demands of EM workflows are heterogeneous. We find memory to be the current limiting factor when deciding the parallelization parameters. Still, some packages greatly benefit from a huge number of CPUs and less RAM, a setup that is achieved more efficiently with a set of instances working cooperatively (cluster architecture).

When using a single instance, the data is already available to the node on a local filesystem (typically, ext4 or xfs), accessed with POSIX standard operations. In a cluster, some mechanism is needed to simulate that the same files are locally available to all nodes. When a dedicated storage subsystem is available (typically, a Storage Area Network, or SAN), popular solutions like IBM Spectrum Scale (commercial license) and Lustre (open source) are implemented. This is not the case in cloud environments, so for this release of ScipionCloud the mechanism we integrated is NFS, an open-source file system that offers a good compromise of simplicity and performance.

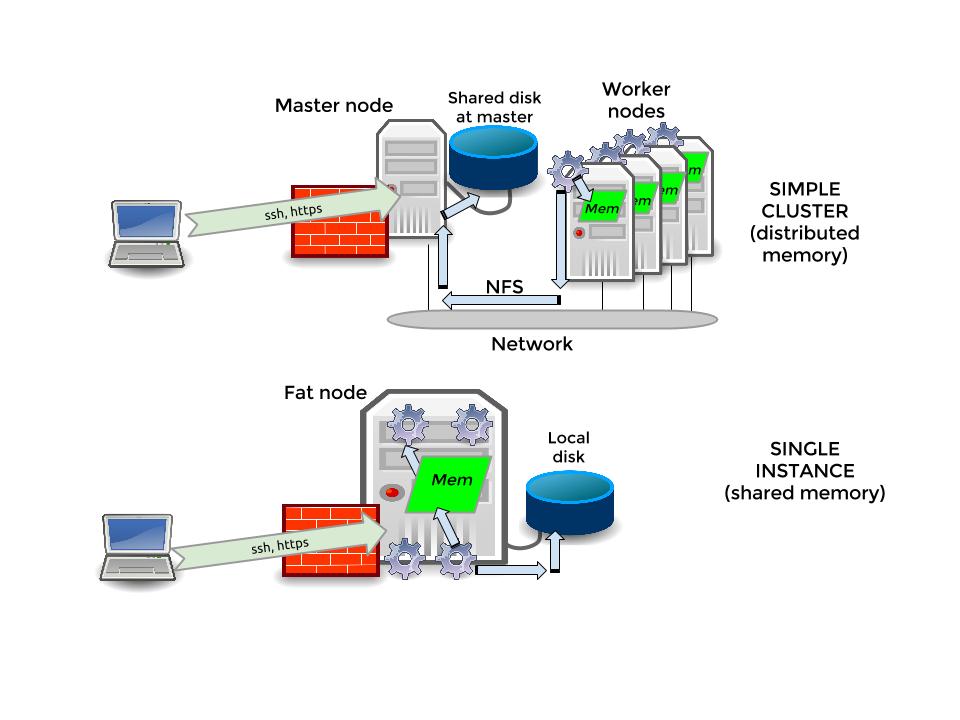
To coordinate the parallel processing across multiple nodes, a middleware is required. In EM the de-facto standard is MPI. MPI has a long tradition, so plenty of choices are available. In ScipionCloud we opted for OpenMPI.

Figure 10 Single node versus cluster architecture in ScipionCloud.

# Feedback on satisfaction

CryoEM scientist from CNB-CSIC were involved in testing the deployments with some of the cryoEM studies they were working on.

They liked the ease of use and the real-time performance of the interactions with the graphical user interfaces. They were also satisfied with the technical support offered.

While the computing performance was good enough, some workflows took more time to complete than in alternative platforms (commercial cloud, local HPC facilities). We attribute this difference to the combination of NUMA (Non-Uniform Memory Access) and KVM (Kernel-based Virtual Machine) technologies used on that Federated Cloud particular instance, which reduce the performance when using big memory areas, like it is the case in Relion.

# Plan for Exploitation and Dissemination

|  |  |
| --- | --- |
| *Name of the result* | *ScipionCloud Virtual Appliance.* |
| *DEFINITION* | |
| *Category of result* | *Software & service innovation.* |
| *Description of the result* | *The image provides cloud-based Scipion v1.0, with high-performance remote graphic user interface support (through a standard Web Browser).* |
| *EXPLOITATION* | |
| *Target group(s)* | *User communities linked to Westlife, related ESFRI projects (e.g. INSTRUCT) and the long-tail of science.* |
| *Needs* | *Ready-to-use Scipion software on EGI Federated Cloud.* |
| *How the target groups will use the result?* | *ScipionCloud can be used as a VirtualBox appliance or instantiate as a Virtual Machine on a Federated Cloud site.* |
| *Benefits* | *Benefit from EGI cloud resources with no investment in installation and configuration of the Scipion software.* |
| *How will you protect the results?* | *Scipion is distributed with a GPL license. ScipionCloud is distributed via the EGI AppDB and protected as the other appliances.* |
| *Actions for exploitation* | *Register as a user to EGI Applications Database.*  *Become part of a Virtual Organization that endorsed the appliance.*  *ScipionCloud can also be used as a VirtualBox appliance.* |
| *URL to project result* | *https://appdb.egi.eu/store/vappliance/scipion.v1.0* |
| *Success criteria* | *Number of users to instantiate or download it.*  *Number of Virtual Organizations to endorse it.* |
| *DISSEMINATION* | |
| *Key messages* | *Scipion software now available ready-to-use in your cloud of choice.* |
| *Channels* | *EGI Newsletter. Submitted paper or technical note in relevant scientific journals.*  *Current channels of communication with Scipion users: wiki, newsletter, courses and presentations.* |
| *Actions for dissemination* | *Submitted paper to relevant scientific journals.*  *Online tutorial.*  *Article on EGI Newsletter.*  *Presentations on relevant workshops or seminars.* |
| *Cost* |  |
| *Evaluation* | *Paper accepted.*  *Number of visits of the online tutorial.*  *Users requests for support.* |

# Future plans



## ScipionCloud improvement

The first version of ScipionCloud is already available and functional but we have identified some areas of improvement.



### GPU acceleration

GPU vector architecture allows for very high parallel performance, whenever the programs are carefully optimized for such architecture. The optimization step may have significant costs in developer man-hours but, nevertheless, the processing time savings have led to an increase in the number of GPU-optimized versions of popular EM programs, like RELION-GPU[4], GCTF and Ge-FREALIGN. Therefore, it has become critical to support such GPU-optimized programs.

ScipionCloud includes GPU support for AWS. The EGI Federated Cloud is working on standardizing GPU-enabled instances, and in particular the CESNET-MetaCloud site has some powered GPU instances that we could use to test GPU-optimized applications. We expect to have GPU-powered ScipionCloud on EGI appliances in future releases.

### Automatic cluster deployment

ScipionCloud has been automated in AWS with StarCluster, a utility for creating and managing distributed computing clusters hosted on Amazon's Elastic Cloud Computing. Currently the version of ScipionCloud available on the EGI AppDB is not prepared to automatically deploy a cluster, but we are studying different solutions such us EC3[[8]](#footnote-8) and Occopus[[9]](#footnote-9) to improve the image with this functionality.

While FedCloud infrastructure is quite robust, in cloud environments one cannot expect instances to live forever. Hence, we plan to include mechanisms to improve the fault tolerance of Scipioncloud, both at the data level (fault-tolerant distributed filesystem), job level (support for down nodes in cluster) and software level (improve checkpointing of EM programs).

### Security

In the current release of ScipionCloud, we use password authentication for the web-based remote desktop Guacamole. The web connection is secured with HTTPS, but we find that getting the password via SSH is not convenient enough for the user. We plan to integrate other authentication approaches that are more convenient for the user of the Virtual Machine.

Besides, in the context of the future support of elastic clusters, a mechanism would dynamically create new virtual machines to accommodate the workload. This mechanism will require some credentials, proxy certificate or similar.

We will start by looking at the EGI Checkin service who would enable federated login for users, so they could come in with their existing accounts (e.g. institutional accounts).

## Plans for exploitation

We foresee different ways of helping users to benefit from ScipionCloud :

1. Registered users of the EGI Applications Database can download the appliance. Besides, if they belong to a Virtual Organization that has endorsed the appliance they will be able to instantiate virtual machines using ScipionCloud appliance.
2. We could also prepare a working environment for certain users, setting up the needed virtual machines upon request.
3. We are also considering offering an online service where users can register and process their data online using ScipionCloud. A similar service is on the scope of the Westlife project but referring only to Scipion Web Tools and will be made available through the Virtual Research Environment. In order to have a full Scipion installation on EGI resources we will have to first integrate a login mechanism such as the EGI checkin service and a mechanism to provide elastic allocation of resources to fulfil users needs.

## Plans for dissemination

We have the following ideas to help disseminate our work on ScipionCloud:

1. Write and submit a paper to one of the most important journals on structural biology.
2. Use the existing channels of communication with Scipion users to advertise ScipionCloud and the new possibilities it offers. These include Scipion wiki, mailing lists, newsletter and scheduled courses and talks.
3. Write an article on EGI newsletter.
4. Create an online tutorial that users can follow.
5. Publish a list of successful deployments.

# Conclusions

We implemented ScipionCloud, a solution that allows to easily deploy the Scipion framework on the EGI Federated Cloud. We developed a Virtual appliance of Scipion abd published it in AppDB, and deployed Scipion Web Tools portal on one of CESNET machines (this including only a subset of what is possible in Scipion). The operation of the online portal is on of the aims of the Westlife project. We could also have a full online service with Scipion thorugh the remote desktop, but this would require login mechanisms, and dynamic resources allocation (cluster) mode in place.

With ScipionCloud, researchers no longer need to invest up front in expensive facilities, nor face the complexities of productively deploying EM software: all Scipion advantages (ease of use, reproducibility, distributed computing) are now available through a standard web browser.

Regarding data transfer rates, typical academic network setups of 1 Gb/s bandwidth are perfectly adequate to handle the transfer of micrographs for several projects overnight. At 1 Gb/s, the transfer of movies is much slower, suggesting either higher speed networks (10Gb/s) or processing for BIM correction locally.

The degree of interactivity in ScipionCloud Graphical User Interfaces has been brought to a level in which it is basically the same as in a local computer.

We have three deployments in production that serve as real testbeds for cryoEM real use cases.

# References

1. “3DEM Method of the year 2015”, Nature Methods 13, 1 (2016).
2. de la Rosa-Trevín, J. e. a. (2016) ‘Scipion: A software framework toward integration, reproducibility and validation in 3d electron microscopy.’, J Struct 530 Biol 195(1), 93–99.
3. Fernández-del Castillo, E., Scardaci, D. and Lopez García, A. (2015) ‘The egi federated cloud e-infrastructure.’, Procedia Computer Science 68, 196–205.
4. Kimanius, D., Forsberg, B. O., Scheres, S. and Lindahl, E. (2016), ‘Accelerated cryo-em structure determination with parallelisation using gpus in relion-2’, bioRxiv .
5. Vinothkumar, K. and Henderson, R. (2016), ‘Single particle electron microscopy: trends, issue and future perspective.’, Quaterly Review of Biophysics 49.
6. Scheres, S. (2012), ‘Relion: implementation of a bayesian approach to cryo-em structure determination.’, J Struct Biol 180(30), 519–530.

1. <https://github.com/I2PC/scipion/wiki/ScipionCloud> [↑](#footnote-ref-1)
2. http://scipion.cnb.csic.es/m/services*/* [↑](#footnote-ref-2)
3. <https://github.com/I2PC/scipion/wiki/ScipionCloud#how-to-use-scipioncloud> [↑](#footnote-ref-3)
4. https://wiki.egi.eu/wiki/HOWTO11 [↑](#footnote-ref-4)
5. [http://west-life.eu](http://west-life.eu/) [↑](#footnote-ref-5)
6. http://wiki.x2go.org/doku.php [↑](#footnote-ref-6)
7. http://guacamole.incubator.apache.org/ [↑](#footnote-ref-7)
8. www.grycap.upv.es/ec3/ [↑](#footnote-ref-8)
9. <http://occopus.lpds.sztaki.hu/> [↑](#footnote-ref-9)