

**EGI-Engage**

Scipion cloud deployment for MoBrain

D6.14

|  |  |
| --- | --- |
| **Date** | 17 November 2016 |
| **Activity** | WP6 |
| **Lead Partner** | CNB - CSIC |
| **Document Status** | DRAFT |
| **Document Link** | <https://documents.egi.eu/document/2982> |

Abstract

We have deployed Scipion, 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, in the EGI Federated Cloud. We have tested the deployments with real EM 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 | 30/11/2016 |
| **Moderated by:** | Malgorzata Krakowian |  |  |
| **Reviewed by** |  |  |  |
| **Approved by:** |  |  |  |

**DOCUMENT LOG**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Issue*** | ***Date*** | ***Comment*** | ***Author/Partner*** |
| **v0.8** | 10/11/2016 | First Draft | Laura del Cano (CNB-CSIC) |
| **V0.9** | 14/11/2016 | Second draft | Jesús Cuenca-Alba (CNB-CSIC) |
| **V1.0** |  | Modified upon reviewers comments |  |
| **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>

|  |  |
| --- | --- |
| ***Acronym*** | ***Description*** |
| **3DEM** | 3D Electron Microscopy |
| **AWS** | Amazon Web Services |
| **EC2** | Elastic Cloud Computing |
| **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 5](#_Toc467154819)

[2 Service architecture 8](#_Toc467154820)

[2.1 High-Level Service architecture 8](#_Toc467154823)

[2.2 Integration and dependencies 10](#_Toc467154827)

[2.2.1 Mobrain and the EGI Federated Cloud 10](#_Toc467154829)

[2.2.2 Scipion deployments on CESNET-MetaCloud 10](#_Toc467154830)

[3 ScipionCloud 14](#_Toc467154831)

[4 Technical challenges 16](#_Toc467154832)

[4.1 Interactive performance 16](#_Toc467154835)

[4.2 Resources management 16](#_Toc467154836)

[4.2.1 Memory 16](#_Toc467154841)

[4.2.2 Disk 16](#_Toc467154842)

[4.2.3 Scalability 17](#_Toc467154843)

[5 Feedback on satisfaction 19](#_Toc467154844)

[6 Plan for Exploitation and Dissemination 20](#_Toc467154845)

[7 Future plans 23](#_Toc467154846)

[7.1 GPU acceleration 23](#_Toc467154850)

[7.2 Automatic cluster deployment 23](#_Toc467154851)

[7.3 High availability 23](#_Toc467154852)

[7.4 Security 24](#_Toc467154853)

[8 Conclusions 25](#_Toc467154854)

[9 References 26](#_Toc467154855)

**Executive summary**

Scipion[[1]](#footnote-1) 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 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, 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” 2015 for good reason. Due to 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, the possibilities are endless.

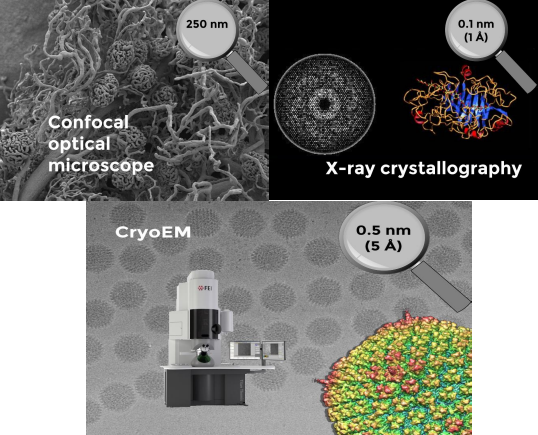


Figure : CryoEM workflows start with a set of movies produced by the Electron Microscope. After 2D and 3D analysis steps, a 3D volume of the specimen is obtained.

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

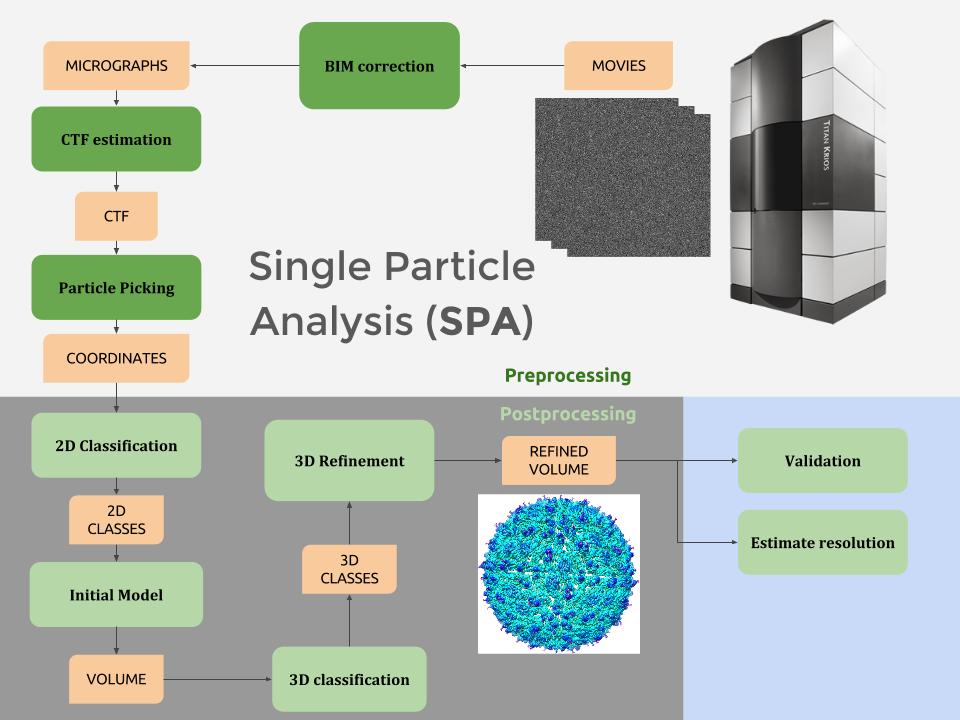


Figure CryoEM workflows start with a set of movies produced by the Electron Microscope. After 2D and 3D analysis steps, a 3D volume of the specimen is obtained.

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.

Obviously, 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. Enter Scipion.

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 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 CCIndividual 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 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.

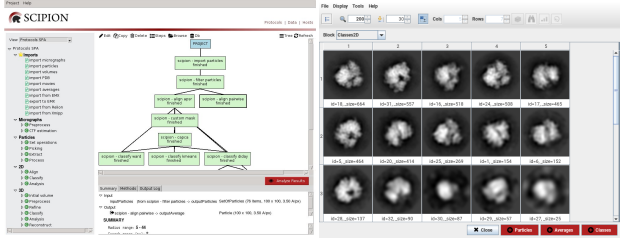


Figure 4 Scipion desktop interface.

The desktop interface supports the whole set of programs and protocols, while the web interface, Scipion Web Tools, offers a subset of the protocols that are useful for a first try without any local installation.

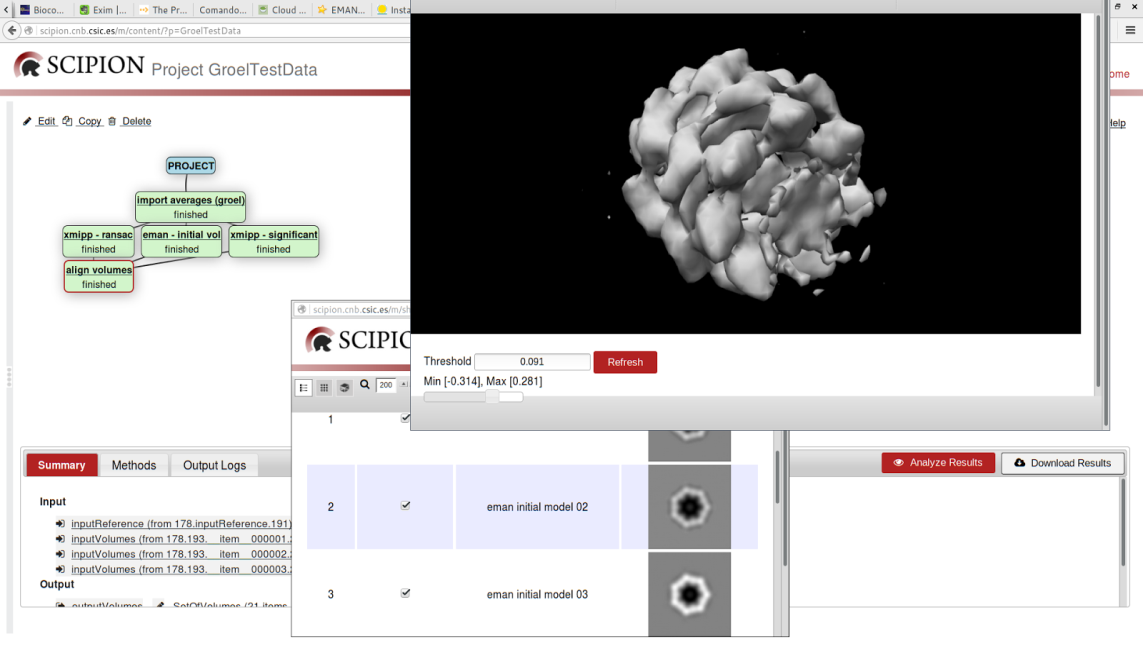


Figure Scipion Web Tools interface.

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

## 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 EC2 platform.

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

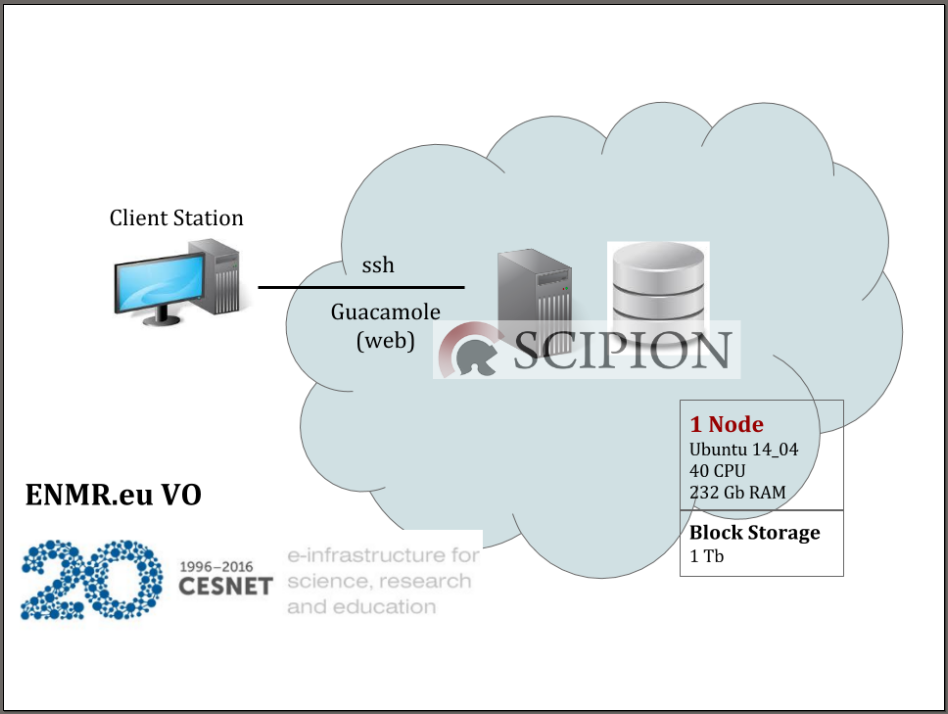


Figure 6 Scipion single node deployment on FedCloud.

#### 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).

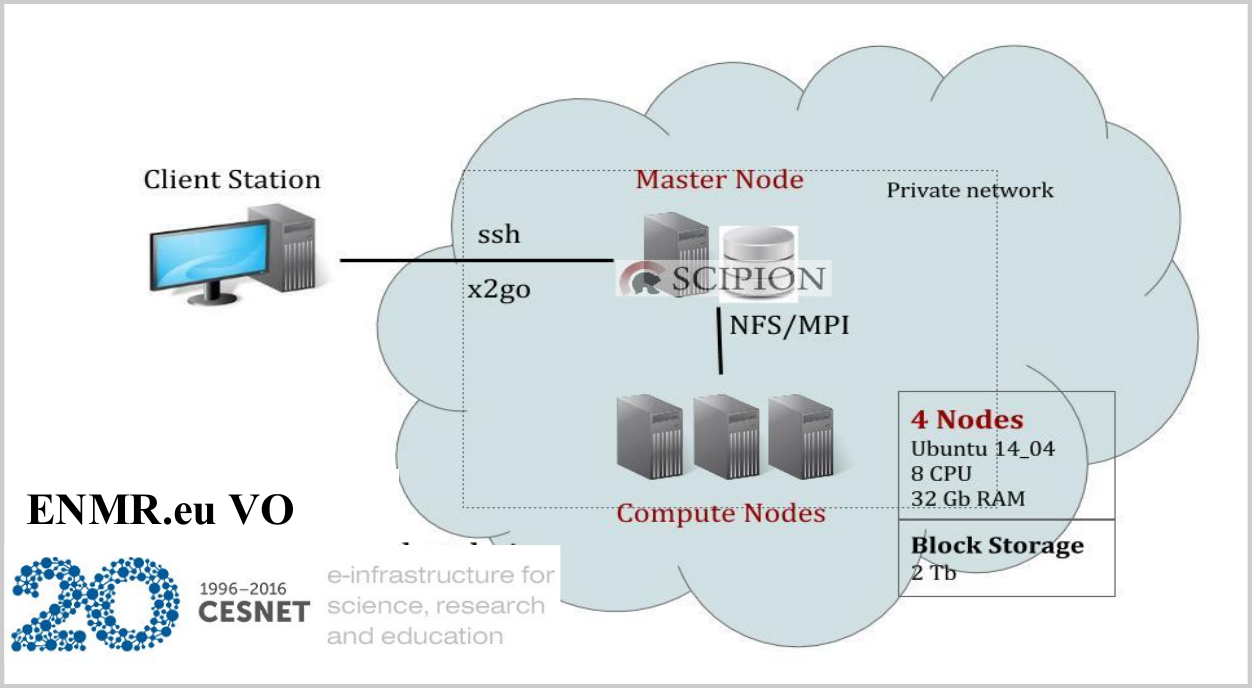


Figure Scipion HPC cluster deployment on FedCloud.

#### Scipion Web Tools deployment

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[[2]](#footnote-2).

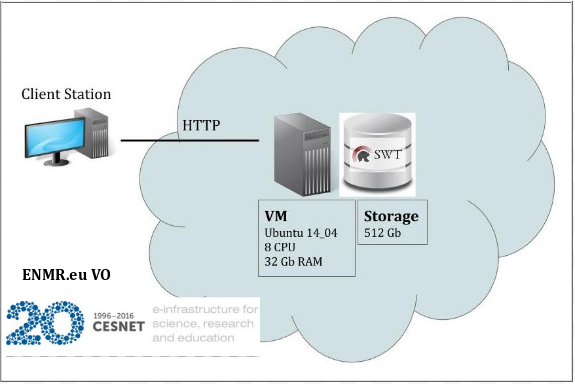


Figure Scipion Web Tools deployment on FedCloud

# 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.

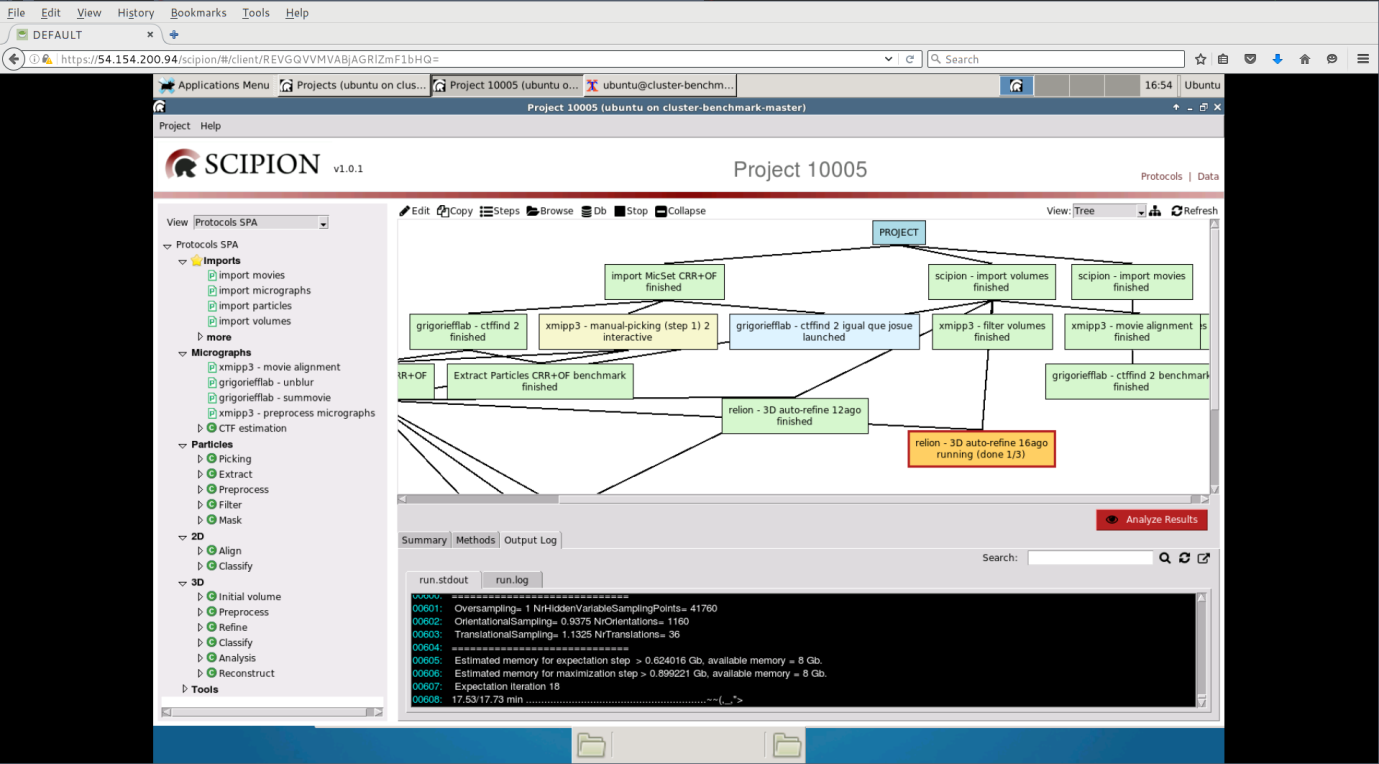


Figure Example of a ScipionCloud session.

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 |

# Technical challenges

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 and Guacamole. 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, 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).

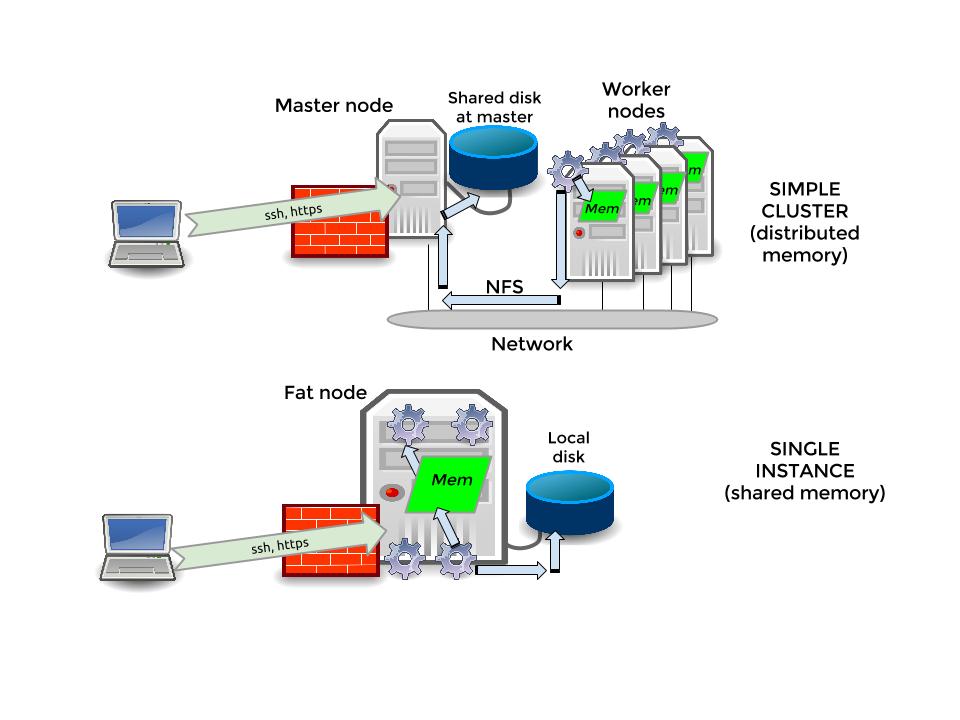


Figure 10 Single node versus cluster architecture in ScipionCloud

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.

# Feedback on satisfaction

*Who was involved in testing and what the outcome of the review was*

# Plan for Exploitation and Dissemination

*This section should provide a plan for exploitation and dissemination (PEDR) of the project results documented in this deliverable. If a plan was already provided in an earlier deliverable, then this plan should provide an update. The content will be used to update the catalogue of project results (*[*http://go.egi.eu/egi-engage-results*](http://go.egi.eu/egi-engage-results)*) and to develop an overall PEDR for the whole project.* ***You can create as many tables as the number of results being described.***

|  |  |
| --- | --- |
| *Name of the result* | *Short name for the result (results generated under the project could be any tangible or intangible output, more particularly data, knowledge or information whatever its form or nature, whether it can be protected or not.)* |
| *DEFINITION* | |
| *Category of result* | * *Technical input to standards: Technical specifications or extensions to standards adopted within the project* * *Policy & Procedure developments: Technical procedures directed at users, service and infrastructure providers (for example to govern access and allocation to resources), policy reports and recommendations, and strategic analysis* * *Software & service innovation: Software developments: (e.g.: workflows, Virtual Machines, applications), new software services deployed for the direct benefit of researchers (e.g.: web portals, gateways), e-Infrastructure Commons such as accounting, AAI, and the Federated Cloud platform and the Open Data platform, demonstrators and prototypes.* * *Business model innovation: Business and sustainability-related outputs (the EGI Service Marketplace concept, the contribution to the Innovation space for the big data value chain, sustainability plans, pay-for-use models)* * *Know-how: Includes all results from fact-finding activities (e.g. surveys, requirement gathering), but also the results from internal exercises (e.g. security challenges) and outputs that can be used for knowledge transfer as training materials.* |
| *Description of the result* | *Description of the result* |
| *EXPLOITATION* | |
| *Target group(s)* | *Describe who will use those results. Es: RIs, international research collaborations and the long-tail of science, industry/SMEs, service providers, Funding agencies and decision/policy makers, Standardisation bodies"* |
| *Needs* | *What are the needs of the target groups that the results aims to fulfil?* |
| *How the target groups will use the result?* | *How the project result will be used? How are you going to achieve the best benefits from the project outcomes? How can you make sure the results they owned are used:*   * *in further research activities other than those covered by the project concerned* * *in developing, creating and marketing a product or process* * *in creating and providing a service* * *in standardisation activities*   *Note: The exploitation does not need necessarily to be done by participants, who may prefer to ensure its use by another entity. Such indirect exploitation can be performed by licensing the results or assigning them to third parties, in accordance with the requirements established in the grant agreement "* |
| *Benefits* | *What are the expected benefits of the result when this will be used by the target groups?* |
| *How will you protect the results?* | *Protection of results is indeed essential in Horizon 2020, since an effective exploitation depends on it. Thus, participants must assess the possibility of protecting their results once these are generated. Please, describe what IP protection approach will you put in place for this result. This can range from simple attribution via open source license to full copyright for commercially exploitable results. (For more information you can read “How to manage IP in Horizon 2020: project implementation and conclusion”* [*https://www.iprhelpdesk.eu/sites/default/files/newsdocuments/FS\_IP\_Management\_h2020\_implementation\_0.pdf*](https://www.iprhelpdesk.eu/sites/default/files/newsdocuments/FS_IP_Management_h2020_implementation_0.pdf) |
| *Actions for exploitation* | *Please, describe the concrete actions that need to be executed to make the result reusable by the target group (e.g., for a software, this can include software packaging for distribution, documentation for the installation, etc). Once executed, the target groups should be able to use the results without barriers.* |
| *URL to project result* | *Link where the result will be made available* |
| *Success criteria* | *What are the success criteria in terms of adoption by the end of the project?* |
| *DISSEMINATION* | |
| *Key messages* | *What messages will you tell to the target groups when informing about the results?* |
| *Channels* | *What channels will you use to deliver the messages to the target? (e.g. Scientific publications, EGI web site, EGI newsletter, participation in conferences or trade fairs)* |
| *Actions for dissemination* | *Describe the concrete set of actions that will be put in place to disseminate this project output. When this result is ready, how will you reach to target group to ensure uptake of the result? (You can list the preliminary list of events where you plan to promote the results or material that will be produced or any other concrete actions that will be put in place during the project)* |
| *Cost* | *What is the expected cost of dissemination actions?* |
| *Evaluation* | *How will you evaluate the impact of the dissemination actions?* |

# Future plans

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, 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, an utility for creating and managing distributed computing clusters hosted on Amazon's EC2.

Currently the version of ScipionCloud available on the EGI AppDB is not prepared to automatically deploy a cluster, but we are studying different solutions to improve the image with this functionality.

## High availability

While FedCloud infrastructure is quite robust, in cloud environments one can not 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. 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 end user. One possibility could be reusing the grid certificates, since they are already needed to instantiate the Fedcloud virtual machines.

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.

# Conclusions

We implemented ScipionCloud, a solution that allows to easily deploy Scipion framework on the EGI Federated Cloud.

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. Brilot, A. F., Chen, J. Z., Cheng, A., Pan, J., Harrison, S. C., Potter, C. S., Carragher, B., Henderson, R. and Grigorieff, N. (2012), ‘Beam-induced motion of vitrified specimen on holey carbon film’, J. Struct. Biol. 177(3), 630–637. 525
3. Cianfrocco, M. and Leschziner, A. (2015) ‘Low cost, high performance processing of single particle cryo-electron microscopy data in the cloud.’, Elife 4.
4. 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.
5. de la Rosa-Trevin, J. M., Oton, J., Marabini, R., Zaldivar, A., Vargas, J., Carazo, J. M. and Sorzano, C. O. (2013) ‘Xmipp 3.0: an improved software suite for image processing in electron microscopy’, J. Struct. Biol. 184(2), 321–328. 535
6. Fernández-del Castillo, E., Scardaci, D. and Lopez García, A. (2015) ‘The egi federated cloud e-infrastructure.’, Procedia Computer Science 68, 196–205.
7. Kimanius, D., Forsberg, B. O., Scheres, S. and Lindahl, E. (2016), ‘Accelerated cryo-em structure determination with parallelisation using gpus in relion-2’, bioRxiv .
8. Kucukelbir, A., Sigworth, F. and H.D., T. (2014), ‘Quantifying the local resolution of cryo-em density maps.’, Nature Methods 11, 63–65
9. Li, X., Mooney, P., Zheng, S., Booth, C., Braunfeld, M., Gubbens, S., Agard, D. and Cheng, Y. (2013), ‘Electron counting and beam-induced motion correction enable near-atomic-resolution single-particle cryo-em.’, Nat Methods 10(6), 584–590.
10. Liao, M., Cao, E., Julius, D. and Cheng, Y. (2013), ‘Single particle electron microscopy: trends, issue and future perspective.’, Nature 504, 107–112.
11. Vinothkumar, K. and Henderson, R. (2016), ‘Single particle electron microscopy: trends, issue and future perspective.’, Quaterly Review of Biophysics 49.

1. [http://scipion.cnb.csic.es](http://scipion.cnb.csic.es/) [↑](#footnote-ref-1)
2. [http://west-life.eu](http://west-life.eu/) [↑](#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)