EGI-InSPIRE

MPI WITHIN EGI

VIRTUAL TEAM PROJECT REPORT

https://wiki.egi.eu/wiki/VT_MPI_within_EGI

Date:	**27/07/2012**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	**https://documents.egi.eu/document/1260**

Abstract

This report is the final report of the 'MPI within EGI' Virtual Team project. The project ran between November 2011 – May 2012 by the European Grid Infrastructure (EGI) collaboration to collect and address those issues that block the uptake of EGI by communities who want to run MPI applications. The report describes the work that was carried out by the project covered, reports about achievements of the activities and captures the issues and actions that have been identified by the project but will be dealt with by EGI members outside of the Virtual Team project.



I. COPYRIGHT NOTICE

II. DOCUMENT LOG

Version	Date	Comment	Author/Partner
0.1	01/06/2012	First draft with content outline.	Gergely Sipos/EGI.eu
0.2	29/06/2012	Section 2.6: Gather info from MPI sites.	Zdenek Sustr/CESNET
0.3	29/06/2012	Sections 2.1 , 2.2: Doc, Nagios probes.	Enol Fernandez/IFCA
0.4	29/06/2012	Sections 2.2 , 2.3: Nagios probes, Inf sys and Appendix.	Gonçalo Borges/LIP
0.5	02/07/2012	Section 2.5: Batch systems support.	Roberto Rosende/CESGA
0.6	02/07/2012	Section 2.4: Accounting. SGE appendix	Ivan Díaz/CESGA
0.7	02/07/2012	Document review and section changes.	Álvaro Simón/CESGA
0.8	06/07/2012	Document review and comments	Gergely Sipos/EGI.eu
0.9	06/07/2012	Sections 2.2, 2.3 review	Gonçalo Borges/LIP
1.0	06/07/2012	Section 2.4 review	John Gordon/STFC
1.1	09/07/2012	PBS appendix	Enol Fernandez/IFCA
1.2	09/07/2012	Section 2.6: NGIs report summaries	Alvaro Simon/CESGA
1.3	10/07/2012	Write abstract, introduction, conclusions	Gergely Sipos/EGI.eu
2	12/07/2012	Finalise based on input from area leaders	Gergely Sipos/EGI.eu
3	27/07/2012	Complete area leaders' lists; Update Section 2.5: Batch systems support	Karolis Eigelis/EGI.eu Gergely Sipos/EGI.eu

III. APPLICATION AREA

This document is a public report produced by the members of the 'MPI in EGI' EGI Virtual Team project, run under the EGI-InSPIRE NA2 virtual team framework. Further information is available at https://wiki.egi.eu/wiki/Virtual_team.

IV. TERMINOLOGY

A complete project glossary is provided at the following page: http://www.egi.eu/about/glossary/.

TABLE OF CONTENTS

1 INTRODUCTION

Despite the dedicated support that a number of NGIs provides to the EGI community through the EGI-InSPIRE 'Heavy User Communities' activity, until now there was still significant issues in uptake and satisfaction of MPI services amongst various user communities. The EGI community therefore setup and run a six month long 'Virtual Team project'[1] to collect and address these issues and make EGI a more attractive platform for MPI jobs. The project started in November 2011 and finished in May 2012. During its six month long lifetime the project has collaborated with different user communities, NGIs user support teams, middleware technology providers and resource providers to identify issues and to establish services by which MPI applications can work successfully in the European Grid Infrastructure. The work spanned across a number of technical areas and these will be all covered in the demonstration:

- Documentation: Improved documentation has been prepared in the EGI wiki for site administrators and for application developers. These provide guidance as to how to configure and to use MPI resources correctly.
- Nagios probes: New monitoring probes for the EGI Service Availability Monitor (SAM) has been defined. These will be implemented and put into production by the Heavy User Community and Operations teams.
- Information system: The typical problems with the registration of MPI resources have been collected and reported to Operations. The Nagios probes have been designed to be able to detect these problems.
- Accounting: Issues with collecting accounting information about parallel applications have been collected and reported to responsible technology developers and providers with request for addressing.
- Batch system integration: Issues with interfacing MPI applications and some of the local batch job schedulers of EGI have been collected and addressed.
- MPI VO: A new VO which includes only correctly configured MPI sites have been setup on the production infrastructure. The VO can be used to port MPI applications to EGI. During the demo MPI members will show how many MPI resources are available in EGI and how to use them. Real MPI applications will be sent to show the capabilities of the VO.

This report is the final output of this project. The document describes the technical areas that the work covered, reports about achievements and captures the issues and actions that have been identified by the project but need to be dealt with by EGI members outside of the Virtual Team project. A summary of these actions is given in the Conclusions section.

[1] MPI within EGI Virtual Team project: https://wiki.egi.eu/wiki/VT_MPI_within_EGI

2 AREAS OF WORK

This section of the document reports about the technical areas of the work that was carried out by the Virtual Team project. Each section was written by the corresponding area leader and consists of three parts:

1. Introduction to the area and the identified issues.
2. Description of achievements: What was achieved within the Virtual Team project in that area?
3. Description of open actions: What are the actions that have been identified but could not be finished during the lifetime of the Virtual Team project? Who will complete these actions?

2.1 Documentation

Area leader: Enol Fernandez and Paschalis Korosoglou

The execution of MPI jobs in the Infrastructure requires documentation that properly describes the details about configuration and operations of the sites, and helps users in finding those sites and using them in an efficient and effective way. Documentation on MPI was scattered in several places and out dated in most of the cases. The MPI-VT established a specific area of work to improve the MPI documentation.

Achievements:

All the MPI documentation was reviewed and classified in two types: user oriented and admin oriented. A single entry point for each type was created in the wiki. The user documentation[2] was updated to cover the latest releases of the middleware included in UMD1 and the next release UMD2, including resources based in ARC and UNICORE stacks that were not considered previously. Links to tutorial material was also added to the MPI User Guide[1] in the wiki.

Documentation for site administrators[3] was also reviewed and updated to reflect the updates of the middleware and to cover the configuration of the batch system queues. This documentation was tested with the deployment of the MPI-kickstart.egi.eu VO[4] and with the verification and staged-rollout of the UMD1 release.

Open Actions:

Documentation needs to be constantly updated to take into consideration the new updates of the software and the improvements introduced in the Infrastructure. The SA3.2 team, that handles the MPI support within EGI-InSPIRE, will perform this task, taking special care of the changes that the different open actions of the MPI VT will bring when they are closed.

2.2 SAM MPI probes

Area leader: Gonçalo Borges and John Walsh

[2] MPI User Guide: https://wiki.egi.eu/wiki/MPI_User_Guide
[3] MPI Manual for site administrators: https://wiki.egi.eu/wiki/MAN03
[4] VO mpi-kickstart.egi.eu: https://www.metacentrum.cz/en/VO/MPI/index.html

Since the beginning of the EGI project that EGI Operations focused considerable effort on the infrastructure monitoring, through the Service Availability Monitoring (SAM[5]), which is now recognized as a robust production service. Such effort helped to increase the confidence of generic users regarding the underlying infrastructure supporting their applications, but the same level of trust is not present in user communities who need environments for running parallel jobs. Part of the problem is coming from the fact that the current SAM tests are not sufficient to properly assess the status of sites supporting MPI. Therefore, the MPI VT felt the need to further enhance the present set of SAM probes by specifying new tests focused on the monitoring of sites supporting MPI.

Achievements:

The current SAM MPI testing framework is completely dependent on the information published by individual sites of the infrastructure. If a site publishes the MPI-START tag, the resource is tested by SAM using the MPI probes, otherwise it is not. This information system dependency does not allow for test sites which are offering MPI functionality but are not broadcasting it, or sites which are broadcasting the MPI/Parallel support in an incorrect way. To break this dependency we requested[6] the definition of a new service type in GOCDB (MPI or Parallel). Taking as basis the current Nagios probes, the VT has created the specification of a new set of probes that try to perform a comprehensive testing of the MPI support in the infrastructure. These probes consist of three different tests:

- MPI Sanity Check (***eu.egi.eu.mpi.EnvSanityCheck***): this probe checks that the values published by the information system has complete information and are within reasonable limits (CPU and Wall-clock values should be coherent with the execution of the MPI applications).
- Simple Job (***eu.egi.mpi.SimpleJob):*** tests that the CE is able to execute parallel jobs with MPI-Start using at least two different WN. The probe detects typical errors such as misconfiguration of the batch system, incorrect installation of the MPI implementation or errors in the distribution of files across nodes.
- Complex Job (***eu.egi.mpi.ComplexJob***): This probe checks the execution of larger jobs and tests a broader functionality of the MPI standard. The test checks if large jobs can be executed at the sites without issues. The MPI application should request 4 slots with 2 instances running in different dedicated machines (JobType="Normal"; CpuNumber = 4; NodeNumber=2; SMPGranularity=2; WholeNodes=True).

The complete documentation of the probes specification is available at the wiki[7] and in the Appendix of this document. These specifications have been communicated to EGI-InSPIRE SA1 Operations, and to EGI-InSPIRE SA3 who will deliver and deploy the tests on the production infrastructure.

Open Actions:

The new set of probes is being developed now by the SA3 team using the resources of the verification testbed at CESGA. Once the development is completely finished, which is expected to end during July 2012, the tests will proposed to the OMB for inclusion in the production SAM release. Inserting a new probe into EGI SAM is typically a lengthy process and for the MPI probes it is expected to finish by the end of the year. The whole process will be monitored by the SA3 team to track any deviations. The following timeline of actions is expected:

- MPI Nagios development by SA3.

5 https://tomtools.cern.ch/confluence/display/SAMDOC/SAM+Intro
6 https://rt.egi.eu/rt/Ticket/Display.html?id=3396
7 https://wiki.egi.eu/wiki/VT_MPI_within_EGI:Nagios

- Testing of SAM nagios probe by EGI-InSPIRE SA2/JRA1
- Delivery of SAM nagios probes to EGI-InSPIRE SA1 Operations. SA1 Operations should trigger the proper procedure for their integration in the production monitoring infrastructure[4].
- In order to improve and check availability and reliability statics for MPI sites a new 'service type' need to be added to GOCDB. (Requirement ticket reference[8])

2.3 Information system

Area leader: Gonçalo Borges, John Walsh and Enol Fernandez

This task aimed to assess the status of the information published by sites supporting MPI while new SAM MPI probes are not around. The goal was to identify the most common issues with the registration of sites that support MPI, communicate these issues to EGI Operations so that they could be handled in the proper forums with the resource providers. Knowing the typical problems also helped members of the VT define SAM probes that are capable of detecting these problems automatically (See section 2.2).

Achievements:
The assessment of the information system was done developing a simple perl script based on the **eu.egi.mpi.EnvSanityCheck** Nagios probe specification. These scripts can also be used as a good starting base for the development of the **eu.egi.mpi.EnvSanityCheck** Nagios probe. The implemented algorithm is the following:

1. Get list of certified sites from GOCDB.
2. Get list of *GlueClusterUniqueIDs* for the different sites.
3. Check which *GlueClusterUniqueIDs* support MPI and inspect the RunTimeEnviroment in the ones that do support MPI.
4. Check which CEs are under a given *GlueClusterUniqueID* supporting MPI and inspect the relevant GlueCE information in the CEs that do support MPI.

The script produces two output files:

- *info.txt*: a container file for the relevant MPI information per *GlueClusterUniqueID* / site, and per *GlueCEInfoHostName / GlueClusterUniqueID*.
- *warn.txt*: a container file with the issues found per *GlueClusterUniqueID* / site, and per *GlueCEInfoHostName / GlueClusterUniqueID*. A warning entry is added to this container following the directives specified for the **eu.egi.mpi.EnvSanityCheck** Nagios probe

Both the scripts and the reports have been distributed through the MPI VT mailing list. The analysis of the reports led to the following conclusions:

- There are 84 *GlueClusterUniqueID* with information issues, either:
  - publishing MPI flavours but not the MPI-START tag;
  - not publishing any MPI flavour;
  - publishing MPI flavours using an incorrect format.

[8] https://rt.egi.eu/rt/Ticket/Display.html?id=3396

- There are 147 GlueCEInfoHostName with information issues, either publishing an incorrect value for GlueCEPolicyMaxSlotsPerJob, or with incorrect Wall-clock time execution limits. Current Glue Schema assumes that sites may have one or more Clusters (ClusterUniqueID) and one or more CEs within each Cluster. The information regarding MPI support and MPI flavours are published for each ClusterUniqueID not for each CE (MaxCPUTime and WallClockTime values are published within CE bdii). Given this scenario first It is gathered MPI support from ClusterUniqueID and then CE information system.

Most of the sites seem to be using a default 999999 value for *GlueCEPolicyMaxSlotsPerJob*. This is because the batch system information providers are not currently prepared to collect that information automatically. A request has been raised to EMI to change the current behaviour (see open actions section). Also, the MPI Wiki page was updated with recommendations on what should be published for *GlueCEPolicyMaxSlotsPerJob* (https://wiki.egi.eu/wiki/MAN03_MPI-Start_Installation_and_Configuration#Job_Policies), and for *GlueCEPolicyMaxCPUTime* and *GlueCEPolicyMaxWallClockTime* (https://wiki.egi.eu/wiki/MAN03_MPI-Start_Installation_and_Configuration#Job_Limits).

Open actions:
The scripts and the reports have been delivered to EGI-InSPIRE SA1 Operations which should follow up the incidents in the right forums with the resource providers. On the other hand, it is expected that EMI follow and enhance the batch system information providers to properly collect the GlueCEPolicyMaxSlotsPerJob value.

- Opened a GGUS ticket to track this issue: https://ggus.eu/ws/ticket_info.php?ticket=82902. linked with Savannah tickets for each batch system:
  - LSF: https://savannah.cern.ch/bugs/index.php?95182
  - SGE: https://savannah.cern.ch/bugs/index.php?95183
  - Torque: https://savannah.cern.ch/bugs/index.php?95184

EMI's progress with these tickets will be monitored by the SA3 MPI team and in case of any deviation they will alert the EGI Technology Coordination Board.

2.4 Accounting system

Area leader: John Gordon and Iván Díaz

At the time of writing accounting for parallel jobs (i.e. MPI jobs) can happen in the EGI accounting system only through CPU job efficiency values (CPU time/wall time). For parallel jobs the efficiency is above 100%, for sequential jobs it is below 100%. The VT aimed to establish a more reliable method for accounting MPI jobs. Project members studied batch system records to identify log characteristics by which MPI jobs, MPI flavours and number of cores used by parallel jobs could be collected. Based on this information the APEL team could deploy accounting plug-ins that are capable of collecting details of parallel job executions (see Appendix 2.3).

Achievements:

- The current usage record structure employed on APEL was found to be detailed enough to include additional details about MPI jobs. MPI flavour; number of cores used by the job)

Open actions:

- The number of nodes and cores used by a job is one of the new metrics to gather about MPI jobs. Meanwhile this metric is already part of the CAR definition[9], it still has to be implemented and supported by the EMI and UMD middleware services. This is expected to happen in EMI-3 in the middle of 2013. After the EMI release the functionality can be integrated into UMD.
- The summaries of the EGI Accounting Portal need to be grouped on site, VO, FQAN or UserDN and include 'Number of cores' as a new variable to be displayed.
- The following requirement has been opened to EMI in the EGI RT in February: 'Accounting system should keep track of the type of the job: MPI or serial. This should be recorded in the Usage Record in order to be easily queried in the accounting repository.' (Ticket reference[10]) EMI requested technical details from EGI for the requirement. EGI Operations through the Operations Management Board need to collect and supply this information for EMI in order to proceed with addressing the requirement.

2.5 Batch system integration

Area leader: Roberto Rosende and Enol Fernandez

Parallel jobs are supported by different batch system. This task was created to track any issue related with executing MPI jobs through the various local batch schedulers that are used on EGI sites: MAUI, Torque, LSF and SGE.

Achievements:

- Issue was identified about MAUI & Torque. MAUI was not able to schedule a new job that requires more than a single CPU. The issues were submitted to EMI:
  - https://ggus.eu/ws/ticket_info.php?ticket=57828
  - https://ggus.eu/ws/ticket_info.php?ticket=67870

  EMI recently provided the EMI-2 release that includes these fixes. Recently released UMD-2 does not include MAUI and TORQUE packages from EMI-2 due to globus libraries dependency issues. These packages are planned to be included in UMD 2.0.1 release in early August, 2012. There are no other open items for LSF and SGE batch systems either.

Open actions:

- Release the MAUI & TORQUE packages of EMI-2 in UMD-2.0.1.

2.6 Gather information from MPI sites – VO setup and testing

Area leader: Zdenek Sustr

[9] CAR is an EMI proposed update to the OGF Usage Record.
[10] Accounting of parallel jobs requirement: https://rt.egi.eu/guest/Ticket/Display.html?id=3328

The VT decided to establish a new VO in the production infrastructure to bring together MPI experts, site administrators and keen users. The purpose of the VO was to facilitate the learning and improvement process of MPI support services and to provide a small testbed where new MPI configurations can be tested. The VO was not intended to serve as the only VO that provides MPI capabilities in the European Grid Infrastructure, it meant to be a VO that provides known well configured sites for application prototyping and testing.

Achievements:

- A new VO, called mpi-kickstart.egi.eu[11] has been configured on the infrastructure. At the end of the MPI VT activities the VO has 22 members from eight institutes. Members' mailing list is available at mpi-kickstart@metacentrum.cz. The VO is fully registered with EGI. It is a Multidisciplinary VO with Grid ID 260. The full VO ID Card is available at https://operations-portal.egi.eu/vo/update/voserial/260.
- Resources for the VO have been provided by two partners:
  - CESNET – no immediate plan to pull out as of mid-2012
  - CESGA – planning to stay at least until the end of 2012

  Although other partners were also considering support at the time the VT was being established, none of them decided to provide resources in the end.
- Documentation adjustment was done in several rounds:
  - Initial review of both the site admin and user documentation (performed by Enol Fernández del Castillo)
  - Clean deployment on the CESNET site in strict adherence to the documentation (performed by Tomáš Kouba), with regular feedback. Several issues were identified and the documentation was rectified accordingly. For instance ambiguous configuration file locations for the local batch system and the submit filter, explanation of additional environmental variables through examples and correct publishing of MPI implementation through the information system. A new documentation section of smoke tests has also been requested and delivered.
  - User-side testing, performed primarily by Viera Šipková (member of the MPI VT), Institute of Informatics, Slovak Academy of Sciences, and by Pavel Fibich, Department of Botany, Faculty of Biological Sciences University of South Bohemia. Additional issues were discovered in user-side testing and addressed jointly with site administrators at the affected sites and MPI experts with the MPI VT.
- Feedback from the NGIs about MPI services has been gathered. Several NGIs have submitted experience reports about current EGI MPI status, quirks and features request:
  - Summary of NGI_IT MPI report[12]:
    - The aim of this survey was to identify the current status of the NGI_IT MPI computational resources.
    - NGI_IT sites are using different MPI flavours: MPICH1, MPICH2, OPENMPI.

[11] VO mpi-kickstart.egi.eu: https://www.metacentrum.cz/en/VO/MPI/index.html
[12] NGI_IT MPI report: https://indico.egi.eu/indico/materialDisplay.py?contribId=1&materialId=slides&confId=828

- 19% of the sites supporting MPI declare to hide the correct MPI TAGs due to Nagios failures and MPI configuration problems (now tracked by MPI VT).
- MPI documentation should be improved (already addressed by the MPI VT).

- Summary of NGI_SK MPI report[13]:
  - NGI_SK has tested different MPI flavours using CREAM client commands and Job Description Language (JDL) attributes.
  - MPI-start framework is working properly for MPI, OpenMP and MPI+OpenMP jobs.
  - User hooks "pre-run" and "post-run" are also working properly.
- Summary of NGI_BG MPI report[14]:
  - MPI and Infiniband clusters were tested: NGI_Bulgaria has two Infiniband and one Myrinet cluster in production.
  - It was developed a torque submitter filter script which requires full worker nodes for MPI jobs.
  - Infiniband capabilities should be published by EGI information system (The updated MPI admins guide[3] already address this and describes how to publish Infiniband capabilities).
  - It was tested also GPU support with torque batch system. Torque batch system includes GPU support (users can request GPUs apart from CPUs). This capability should be exposed through the middleware, however, this is not yet supported in Maui. GPU resources are now published through grid information system. This experience is fed into and will be used by GPGPU Virtual Team[15] which is tracking these issues.

Open actions:

- Bringing in more sites to the MPI-Kickstart VO would be beneficial. The demo to be held by the VT at the EGI Technical Forum in Prague will provide an opportunity for this. Besides the demo SA1 mechanisms (OMB, Site broadcasts), UCST mechanisms (UCB, VO Managers' broadcast), Dissemination mechanisms (EGI Technical Forum, Blog, newsletter) will be also used. This will be planned and executed by the MPI team in SA3.

[13] NGI_SK MPI report: https://indico.egi.eu/indico/materialDisplay.py?contribId=4&materialId=0&confId=828
[14] NGI_BG MPI report: https://indico.egi.eu/indico/materialDisplay.py?materialId=1&confId=1075
[15] EGI Virtual Team on GPGPU requirements: https://wiki.egi.eu/wiki/VT_GPGPU

3 CONCLUSION

The Virtual Team project focused on different aspects of what's needed to successfully run MPI jobs on the EGI production grid. Most of the issues have been solved, but there are still open actions that need to be followed up outside of the Virtual Team. These actions have been described in the report and are collected in the table below. The overall responsibility of monitoring progress with open actions, providing and if needed further improving EGI MPI services is on the SA3.2 task of EGI-InSPIRE.

Within the EGI Helpdesk the SA3.2 team operates a support unit (called 'MPI User Support'[16]) that serves as the primary contact point for MPI application developers and MPI site administrators to feedback experiences, to report issues and requirements about EGI MPI services. (Another MPI-related support unit, called MPI, also exists in the EGI Helpdesk, but that one provides middleware development support through EMI.) On top of this the SA3.2 team will setup a dedicated page in the EGI Wiki where up to date information about the MPI services and support channels will be provided. The promotion of these services and support mechanisms will need to happen within the EGI community. The promotion activity will start at the EGI Technical Forum in September 2012, where the members of the Virtual Team will deliver a demonstration of the services and of a few applications that already benefit from these. After the Technical Forum additional promotion can take place through EGI the dissemination, operation and user support channels.

Summary table of open actions. This table is available online at https://wiki.egi.eu/wiki/MPI_VT_Open_Actions, where progress with the actions will be recorded.

ID	Action Description	Responsible	Ticket (where applicable)
Documentation			
1.	Update the user and site administrator MPI documentations when any of the MPI services (e.g. scripts, SAM probes), or relevant parts of the infrastructure are updated.	EGI-InSPIRE SA3.2	
SAM MPI probes			
2.	Implement the MPI SAM Nagios probes according to the specification prepared by the MPI VT.	EGI-InSPIRE SA3.2	
3.	Test the MPI SAM Nagios probes, then deliver to EGI Operations for integration.	EGI-InSPIRE SA2 and JRA1	
4.	Integration of the MPI SAM Nagios probes into the production monitoring infrastructure.	EGI-InSPIRE SA1 and JRA1	
5.	In order to improve and check availability and reliability statics for MPI sites a new 'service type' need	EGI-InSPIRE JRA1	https://rt.egi.eu/guest/Ticket/Display.html?id=3396

[16] MPI User Support in EGI Helpdesk: https://wiki.egi.eu/wiki/GGUS:MPI_User_Support_FAQ

	to be added to GOCDB (serial or parallel).		
Information System			
6.	Correct misconfigured registration of MPI sites in the Information System.	EGI-InSPIRE SA1 with resource providers	
7.	Enhance the batch system information providers to properly collect the GlueCEPolicyMaxSlotsPerJob value.	EMI, monitored by EGI-InSPIRE SA3.2	https://ggus.eu/ws/ticket_info.php?ticket=82902 Savannah tickets for each batch system: LSF: https://savannah.cern.ch/bugs/index.php?95182 SGE: https://savannah.cern.ch/bugs/index.php?95183 Torque: https://savannah.cern.ch/bugs/index.php?95184
Accounting			
8.	Provide technical details for a requirement ticket for EMI.	EGI OMB	https://rt.egi.eu/guest/Ticket/Display.html?id=3328
9.	The EMI middleware services need to implement and support the CAR definition that already includes 'Number of nodes and cores used by a job' as an accounting metric. This is expected to happen in EMI-3.	EMI, monitored by EGI-InSPIRE SA3.2	
10.	The summaries of the EGI Accounting Portal need to be grouped on site, VO, FQAN or UserDN and include 'Number of cores' as a new variable to be displayed.	EGI-InSPIRE JRA1	https://rt.egi.eu/guest/Ticket/Display.html?id=4071
Batch system integration			
11.	Release the MAUI & TORQUE packages from EMI-2 in UMD-2.0.1.	EGI-InSPIRE SA2	
VO Setup and user support			
12.	Bringing in more sites to the MPI-Kickstart VO would be beneficial.	VT members (Demo at TF2012) EGI-InSPIRE SA3.2	https://rt.egi.eu/rt/Ticket/Display.html?id=3396
13.	Setup a page in the EGI Wiki that collects and provides information about the MPI services and support	EGI-InSPIRE SA3.2	

	mechanisms for application developers and resource providers.		

4 APPENDIX

4.1 SAM MPI probes specifications

4.1.1 *eu.egi.mpi.EnvSanityCheck description*

- **Name:** eu.egi.mpi.EnvSanityCheck
- **Requirements:** The service should be registered in GOCDB as a MPI (or Parallel) Service Type
- **Purpose:** Test the information published by the (MPI or Parallel) service
- **Description:** The probe should test:

  1. if the service publishes the **MPI-START** tag under GlueHostApplicationSoftwareRunTimeEnvironment*;*
  2. if the service publishes the **MPI flavour** tag under GlueHostApplicationSoftwareRunTimeEnvironment according to one of the following formats: **<MPI flavour>, <MPI flavour>-<MPI version or <MPI flavour>-<MPI version>-<Compiler>***;*
  3. if the **GlueCEPolicyMaxSlotsPerJob** variable published under GlueCE has a reasonable value (not 0 nor 1 nor 999999999) for the queue where the MPI job will execute;
  4. if the **GlueCEPolicyMaxWallClockTime** variable published under GlueCE has a resonable value (not 0 nor 999999999) for the queue where the MPI job will execute;
  5. if the **GlueCEPolicyMaxCPUTime** allows to execute a parallel application requesting, at least, 4 slots where each task will spend GlueCEPolicyMaxWallClockTime minutes of WallClockTime.

- **Dependencies:** None
- **Frequency:** Each hour
- **Timeout: 120 s**

4.1.1.1 eu.egi.mpi.EnvSanityCheck expected behaviour

#	Use Case	Probe Result
1	**MPI-START** tag is not present under ***GlueHostApplicationSoftwareRunTimeEnvironment***	CRITICAL
2	No **MPI flavour** tag (following any of the proposed formats) is present under	CRITICAL

	GlueHostApplicationSoftwareRunTimeEnvironment	
	The probe reaches a timeout and the probe execution is cancelled	UNKNOWN
3	***GlueCEPolicyMaxSlotsPerJob*** is equal to 0 or 1 or 999999999	WARNING
4	(***GlueCEPolicyMaxWallClockTime*** is equal to 0 or to 999999999) OR (***GlueCEPolicyMaxCPUTime*** / ***GlueCEPolicyMaxWallClockTime < 4***)	WARNING
5	If (**MPI-START** tag is present in ***GlueHostApplicationSoftwareRunTimeEnvironment***) **AND (MPI-FLAVOUR** tag is present in ***GlueHostApplicationSoftwareRunTimeEnvironment***) AND (***GlueCEPolicyMaxSlotsPerJob*** variable is not 0 or 1 or 999999999) **AND** (***GlueCEPolicyMaxCPUTime*** / ***GlueCEPolicyMaxWallClockTime >=4***)	OK

4.1.2 *eu.egi.mpi.SimpleJob description*

- **Name:** org.sam.mpi.SimpleJob
- **Requirements:** The service should be registered in GOCDB as a MPI (or Parallel) Service Type; Job submission requesting two slots in different machines (JobType="Normal"; CpuNumber = 2; NodeNumber=2)
- **Purpose:** Test the MPI functionality with a minimum set of resources.
- **Description:** The probe should check if:
  1. **MPI-START** is able to find the type of scheduler
  2. **MPI- START** is able to determine if the environment for the MPI flavour under test is correctly set
  3. The application correctly compiles
  4. **MPI-START** is able to distribute the application binaries
  5. The application executes with the number of requested slots and finishes correctly.
  6. **MPI-START** is able to collect the application results in the master node.
- **Dependencies:** Executed after eu.egi.mpi.EnvSanityCheck if that probe exits with WARNING or OK status.
- **Frequency:** Same frequency as a regular job submission test.
- **Timeout**: Same timeouts as a regular job submission test.

4.1.2.1 eu.egi.mpi.SimpleJob expected behaviour

#	Use Case	Probe Result
1	**MPI-START** is not able to determine which kind of scheduler is used at the site	WARNING
2	**MPI-START** is not able to determine if the environment for the MPI flavour under test is correctly set	WARNING
3	The compilation of the parallel application failed	CRITICAL
4	**MPI-START** failed to distribute the application binaries	CRITICAL
5	The MPI application execution failed	CRITICAL
6	**MPI-START** failed to collect the application results in the master node	CRITICAL
7	The application executed successfully with less slots than the requested ones	CRITICAL
8	The probe reached a timeout and the probe execution is cancelled	WARNING
9	The probe reaches a timeout (in two successive attempts) and the probe execution is cancelled	CRITICAL
10	The application executed successfully with the requested slots **AND MPI-START** was able to collect the application results in the master node	OK

4.1.3 *eu.egi.mpi.ComplexJob description*

- **Name:** org.sam.mpi.ComplexJob
- **Requirements:** The service should be registered in GOCDB as a MPI (or Parallel) Service Type; Job submission requesting 4 slots with 2 instances running in different dedicated machines (JobType="Normal"; CpuNumber = 4; NodeNumber=2; SMPGranularity=2; WholeNodes=True)
- **Purpose:** Test the MPI functionality and check the recommendation from the EGEE MPI WG are being implemented.
- **Description:** The probe should check if:

  1. MPI-START is able to find the type of scheduler
  2. MPI- START is able to determine if the environment for the MPI flavour under test is correctly set
  3. The application correctly compiles

4. MPI-START is able to distribute the application binaries
5. The application executes with the number and characteristics of requested slots and finishes correctly.
6. MPI-START is able to collect the application results in the master node.

- **Dependencies:** Executed after org.sam.mpi.envsanitycheck if that probe exits with WARNING or OK status.
- **Frequency:** Once per day.
- **Timeout:** just until the next probe is to be submitted.

4.1.3.1 eu.egi.mpi.ComplexJob expected behaviour

#	Use Case	Probe Result
1	**MPI-START** is not able to determine which kind of scheduler is used at the site	WARNING
2	**MPI-START** is not able to determine if the environment for the MPI flavour under test is correctly set	WARNING
3	The compilation of the parallel application failed	CRITICAL
4	**MPI-START** failed to distribute the application binaries	CRITICAL
5	The MPI application execution failed	CRITICAL
6	**MPI-START** failed to collect the application results in the master node	CRITICAL
7	The application did not executed as requested (on less slots than the requested ones or on a single machine)	CRITICAL
8	The probe reached a timeout and the probe execution is cancelled	WARNING
9	The application executed successfully with the requested slots **AND** *MPI-START* was able to collect the application results in the master node	OK

4.2 MPI (S)GE accounting usage records

Work Type	Mpich2 (SL5)* 4 cores	Openmpi (SL5) 4 cores		Not MPI
qname	dteam	dteam	dteam	dteam
hostname	sa3-wn002.egee.cesga.es	sa3-wn002.egee.cesga.es	sa3-wn001.egee.cesga.es	sa3-wn002.egee.cesga.es
group	dteam	dteam	dteam	dteam
owner	dteam047	dteam047	dteam047	dteam047
job_name	cream_998098136	cream_021370514	cream_021370514	cream_416876310
job_number	350	351	351	352
account	sge	sge	sge	sge
priority	19	19	19	19
submission_time	1327492453	0	1327492703	1327494326
start_time	1327492467	1327492726	1327492707	1327494342
end_time	1327492494	1327492745	1327492748	1327494358
failed	12	0	12	0
exit_status	0	0	0	0
ru_wallclock	27	19	41	16
project	NONE	NONE	NONE	NONE
department	defaultdepartment	defaultdepartment	defaultdepartment	defaultdepartment
granted_pe	mpi	mpi	mpi	NONE
slots	4	4	4	1
task_number	0	0	0	0
cpu_time	1.000000	18.000000	19.000000	0.110000
mem	0.007557	2.688694	2.692575	0.004864
io	0.000000	0.000000	0.000000	0.000000
category	-U dteam -q dteam -pe * 4	-U dteam -q dteam -pe * 4	-U dteam -q dteam -pe * 4	-U dteam -q dteam
iow	0.000000	0.000000	0.000000	0.000000
pe_taskid	NONE	1.sa3-wn002	NONE	NONE
maxvmem	623706112.000000	434458624.000000	878596096.000000	489734144.000000

(SL5)*: Mpich2 for SL5 does not show the number of WN used by the parallel job. This issue was fixed in SL6 (UMD2).

4.3 MPI PBS accounting usage records

Work Type	Mpich2 (4 cores)	OpenMPI (4 cores)	Not MPI
queue	dteam_q	dteam_q	dteam_q
user	dteam001	dteam001	dteam001
group	dteam	dteam	dteam
owner	dteam001@wn1.novalocal	dteam001@wn1.novalocal	dteam001@wn1.novalocal
jobname	mpich2-new.sub	ompi.sub	test.sub
ctime	1341339227	1341337828	1341337907
qtime	1341339227	1341337828	1341337907
etime	1341339227	1341337828	1341337907
start	1341339228	1341337828	1341337908
exec_host	wn2.novalocal/1+wn2.novalocal/0+wn1.novalocal/1+wn1.novalocal/0	wn2.novalocal/1+wn2.novalocal/0+wn1.novalocal/1+wn1.novalocal/0	wn2.novalocal/0
Exit_status	0	0	0
resources_used.cput	00:00:32	00:00:32	00:00:06
resources_used.mem	5568kb	796kb	3964kb
resources_used.vmem	164804kb	13368kb	149868kb
resources_used.walltime	00:00:09	00:00:10	00:00:07
Resource_List.nednodes	2:ppn=2	2:ppn=2	
Resource_List.nodect	2	2	
Resource_List.nodes	2:ppn=2	2:ppn=2	

Ppn: processors per node

2:ppn=2: 2 Nodes and 2 processors per node