Editor:
Glenn Wasson, UVA

http://forge.gridforum.org/projects/ogsa-hpcp-wg

12/12/2007

Interoperability Experiences with the High Performance Computing Basic Profile (HPCBP), Version 0.3

Status of this Memo

This memo provides information to the Grid community regarding the experiences of the authors in implementing the HPC Basic Profile. Distribution is unlimited.

Copyright Notice

Abstract

This document describes the experience of interoperability testing of independent implementations of the High Performance Computing Basic Profile (HPCBP) and the specification which it profiles, the Basic Execution Service (BES) and the Job Submission Description Language (JSDL).

Contents

1 Introduction

This document describes the experiences of multiple groups performing interoperability testing on their implementations of the HPC Basic Profile [HPCP10], including the HPCProfileApplication [HPCP-A], BES [BES10] and JSDL [JSDL10]. Much of this testing was organized around a SuperComputing 2007 interoperability demo. The organization for this document is as follows. Section 2 briefly describes the groups involved in the interop testing and the software systems employed. Section 3 describes the results of this testing and section 4 describes note-worthy experiences and discoveries made while performing the tests.

2 Implementations

The name and affliation of each implementation are listed here along with a brief description of their software stack and related tools.

Group	Software stack/Tools	Project URL
University of Virginia e-Science Group	.NET 2.0, WSE 3.0, CCS	http://www.cs.virginia.edu/~humphrey
University of Virginia e-Science Group	Linux, gSoap, C, PBS	http://www.cs.virginia.edu/~humphrey
Microsoft	WCF, .NET 3.0, CCSv1, HPCS 2008 beta 1	http://www.microsoft.com
Platform Computing	gSoap, Linux, C, LSF	http://www.platform.com
OMII-UK GridSAM	Java, OMII-UK Container(Axis, Tomcat, Linux)	http://gridsam.sourceforge.net/2.0.1/index.html
EGEE 2/OMII Europe CREAM-BES	Java, Linux, Tomcat/Axis1.4, LSF, PBS	http://grid.pd.infn.it/cream/
UNICORE	Java, XFire, XMLBeans, Linux/Windows, PBS, SGE, CCS (Paederborn), Loadleveller	http://www.unicore.eu
NorduGrid/KnowARC A-REX	Linux, C++, libxml2	http://www.nordugrid.org/documents/a-rex.pdf

3 HPC Base Profile Interoperability Tests

This section describes the tests performed using clients and services implementing the HPC Base Profile as well as BES and JSDL as they relate to HPCBP. As part of a demonstration for SC07, the groups shown in section 2 each implemented a service and client complying with HPCBP. Each group then performed a series of tests using their client against each other group's service.

These tests involved testing the five BES-defined methods, CreateActivity, GetActivityStatuses, TerminateActivities, GetActivitiesDocuments, and GetFactoryAttributesDocument. Since the "back-end" compute resources associated with each service differ, a standard JSDL document to define jobs was not provided. Instead, each service implementation provided a JSDL document that all clients could use. Most groups' services supported client authentication using either username/password or X.509 certificates (some allowed only X.509) and many groups tested their clients using both mechanisms.

The UVA eScience group's client is a web form based tester that is meant to be long-lived and provide automated interoperability testing for both the SC07 demonstration and for future service implementations. This client, called the HPC Basic Profile Interoperability Tester, acts as an HPBCP-compliant client, sending messages to a service designated by the user, and verifying both the schema and values of the responses to see if they are consistent with the specifications. This site allows its user to test the five BES methods supported by HPCBP (and used in the SC07 demo), CreateActivity, GetActivityStatuses, TerminateActivities, GetActivitiesDocuments, and GetFactoryAttributesDocument. In addition, the Interoperability Tester can generate "erroneous" messages designed to test services' responses to standard error conditions including the UnsupportedFeatureFault, InvalidRequestMessageFault and the UnknownActivityIdentifierFault. The Interoperability Tester can authenticate itself using either an X.509 certificate (via a mutually-authenticated SSL connection to the service) or using a username/password.

Since the Interoperability Tester is meant to test many services with different back-ends, it allows users to enter job information which it transforms into an HPCProfileApplication element [HPCPA10] for inclusion in a JSDL document. The JSDL elements profiled by HPCPA which can be set by the user are:

Job name	Job project	Executable
Input	Output	Error
Working directory	Arguments	Environment
Candidate hosts	Exclusive execution	Operating system type
Operating system version	CPU architecture	Total CPU count

The user can also provide the service's URL, select which client authentication mechanism to use, and select which tests to perform. A typical test sequence will involve creating a job (calling CreateActivity using a JSDL document formed from the values in the above table), using the resulting activity identifier to poll the job's status (calling GetActivityStatuses), retrieving the job's JSDL document (calling GetActivitiesDocuments using that activity id) and then terminating the job (calling TerminateActivities for that id). The results of each test appear on the web page after the user presses the "Begin" button.

The error tests involve the generation of messages which contain erroneous values. The UnsupportedFeatureFault test involves creating a JSDL document using the values entered for the above elements, but then adding the element <ThisIsABogusElement> to the JSDL document as a child of the <JobDescription> element. This child element is undefined by the specifications (and likely all services), so its inclusion in the JSDL should result in an UnsupportedFeatureFault being thrown. The InvalidRequestMessage test involves creating a JSDL document in which the of the <TotalCPUCount> element is set to 1.5. The BES specification defines all numeric values as floating point values, but notes that all values will not make sense in all cases. Since 1.5 CPUs is non-sensical, an InvalidRequestMessageFault should be thrown by the receiving service. Finally, the UnknownActivityIdentifier test is performed by creating a bogus activity identifier (i.e. one which is not returned from a CreateActivity call on the service being tested) and sending a GetActivityStatuses message referencing that identifier. The activity identifier used is:

```
<bes:ActivityIdentifier><wsa:Address>https://thereisnothinghere.com/HPCPService
</wsa:Address></bes:ActivityIdentifier>
```

3.1 Test Results

The results of invoking the 5 BES methods can be seen in the following tables. The first table shows clients that authenticated with username/password while the second table shows client that authenticated with X.509 certificates.

Client/service	UVA.NET	Microsoft	CREAM	Unicore	GridSAM	NorduGrid	Platform
UVA .NET	---		✓	✓	✓	✓	✓
Microsoft	✓	---			✓		✓
CREAM			---	✓			✓
Unicore	✓		✓	---	✓	✓	✓
GridSAM	✓		✓	✓	---	✓	✓
NorduGrid	✓		✓	✓	✓	---	✓
Platform	✓	✓		✓	✓		---

Table 1. Interop Matrix for Clients Authenticating with Username/Password

Client/service	UVA.NET	Microsoft	CREAM	Unicore	GridSAM	NorduGrid	Platform
UVA .NET	---		✓	✓	✓	✓	✓
Microsoft	✓	---			✓		✓
CREAM			---	✓			✓
Unicore	✓		✓	---	✓	✓	✓
GridSAM	✓		✓	✓	---	✓	✓
NorduGrid	✓		✓	✓	✓	---	✓
Platform	✓	✓		✓	✓		---

Table 2. Interop Matrix for Clients Authenticating with X.509 Certificates

3.2 Security Interoperability

This section describes the experience using the HPC Profile defined security measures [HPCP10]. Namely, the HPC Profile requires services to support SSL v3.0/TLS v1.0 and therefore services are identified by X.509 certificates. Clients may be identified either with X.509 certificates or username/password.

The HPCBP Interoperability Tester can identify itself using either mechanism. Currently, all implementations have successfully interoperated using username/password, except NorduGrid which supports only X.509. The UVA e-Science Group implementation, NorduGrid, GridSAM and CREAM-BES have all successfully performed client authentication using an X.509 certificate.

4 Issues Encountered

While most service implementations have successfully interoperated with the HPCBP Interoperability Tester, there were several issues encountered while performing these tests. These issues are described here to potentially assist other implementers / Interoperability testers, who may encounter similar issues in the future, or people authoring future profiles.

Issue: The most common issue (or class of issues) that implementers faced during the interop testing surrounded establishment of SSL connections. While this is outside the bounds of the HPCBP, it illustrates that there are many issues that must be faced in order to achieve operational interoperability – not just spec compliance.

Resolution: There was no single cause of SSL issues and therefore pairwise resolutions were used. A common resolution involved correctly configuring trust chains.

Issue: The Microsoft implementation required both Operating System Type and Operating System Version information in order to create an activity. Initially, the Interoperability Tester only allowed specification of OS Type. While OS version is not required by JSDL, it is permissible for an implementation to require it.

Resolution: The Interoperability Tester was changed to allow this field to be specified.

Issue: The UVA e-Science Group’s Linux implementation requires that ExclusiveExecution be set to true. The HPCBP does not specify a value that must be used for this element and the service does not specify that this value was required anywhere.

Resolution: The Interoperability Tester must be set to specify the correct value when communicating with this service.

Issue: Some implementations had issues generating the UnsupportedFeatureFault. The BES specification says that this fault should be thrown for unsupported non-JSDL elements. “Out of the box”, however, some tooling parses input based only on a provided schema (JSDL in this case). This means it ignores unsupported elements instead of generating faults.

Resolution: Code must be explicitly added to check for these “unsupported” elements.

Issue: The UVA e-Science Group’s Linux/PBS implementation had difficulty returning semantically correct values for the ActivityStatus. While syntactically correct (i.e. schema compliant) values could always be returned, the service’s PBS queue was configured such that finished jobs are removed immediately. In non-web services environments, this is not an issue because the job owner is sent an email informing them that their job is done. However, this means that an HPCBP service cannot simply rely on PBS’s queue status to determine the status of any activity for which it has given out an ActivityIdentifier.

Resolution: The service must provide another mechanism for saving the state of jobs which are not currently queued or running.

Issue: Some tooling used incompatible timestamp formats in the SOAP security headers. While this is outside of the HPCBP, it can be an issue when using username/password for client authentication. Since the username token is framed according to the WS-Security UsernameToken Profile, it is placed in a message’s SOAP headers. Some tools, by default, place a timestamp in any SOAP Security header.

Resolution: This auto-timestamping behavior either must be disabled or some mutually compatible timestamp format must be found.

Issue: The similarity between the internal GridSAM Core Engine interface and the HPCBP interface allowed for an easy mapping of their respective methods with the exception of the GetFactoryAttributesDocument method.

Resolution: For prototype demonstration purposes partial support for this method was included for the Linux local (process-forking) configuration mode.

HPCBP Client -> GridSAM HPCBP Service Issues:

Issue: The service was detecting XML syntactical errors in the SOAPBody element of Platform's client’s request. The service could not parse the SOAP message due to missing/truncated closing XML tags.

Resolution: Turning on/off a configuration parameter in the gSOAP server controlling the length of the in-memory data buffer holding serialized request messages.

Issue: The service was detecting an XML syntactical error in the JSDL document sent by Platform's client. The service could not parse the document due to an invalid namespace prefix declarations.

Resolution: Fix in Platform's gSOAP client code.

Issue: Platform's and Microsoft's clients were failing to establish a SSL connection with the GridSAM HPCBP Service due to an invalid host certificate. The clients were detecting a discrepancy between the Common Name(CN) of the service certificate's Distinguished Name(DN) and the host name part in the service URL.

Resolution: This checking of CN and host name in URL was a configurable option for the clients. Alternatively, a host certificate with matching CN and URL host name was uploaded in the tested service installations.

Issue: The University of Virginia and Microsoft implementations were failing to establish a connection with GridSAM service due to unsupported SOAP security headers. Both implementations were including a timestamp.

Resolution: Including a timestamp in the SOAP security headers was a configurable option. The default OMII configuration includes timestamps in all message headers, however by replacing the default OMII security libraries removed that functionality from the tested GridSAM HPCBP installations.

GridSAM Client -> HPCBP Service Issues:

Issue: The University of Virginia and Microsoft service implementations, both based on .NET, were detecting an invalid value in the HTTP headers section of the client request. These two services were rejecting the request due to inconsistency between the SOAPAction header value in the request and the SOAPAction value in the service's WSDLs.

Resolution: The client code was modified to include the correct SOAPAction value.

Issue: Nordugrid's A-REX service does not support the WS-Security UsernameToken Profile

Resolution: This feature will be implemented in the next software release.

Issue: The UVA e-Science Group’s Linux implementation needed the SOAPAction attribute in the HTTP header therefore the Nordugrid's client cannot sent job to it.

Resolution: This attribute added to the client side module.

Issue: There were problem with multiple same namespaces at the Nordugrid's client therefore the client cannot submit job to the CREAM-BES implementation.

Resolution: The A-REX's client must understand the multiple same namespaces.

Issue: CREAM-BES does not accept self-signed X.509 certificates; clients using these certificates cannot create new activities (submit jobs).

Resolution: Clients must provide X.509 certificates which are signed by same Certification Authority (even a “fake” one will suffice, provided that CREAM-BES is instructed to trust the CA).

5 Author Information

Glenn Wasson
University of Virginia

6 Contributors

We gratefully acknowledge the contributions made to this specification by Marty Humphrey, Steven Newhouse, Chris Smith, Bill Nitzberg, Vesso Novov, Moreno Marzolla, Gabor Roczei, and Mohammad Shahbaz Memon.

7 Acknowledgements

We are grateful to numerous colleagues for discussions on the topics covered in this document, in particular (in alphabetical order, with apologies to anybody we've missed) [insert names].

We would like to thank the people who took the time to read and comment on earlier drafts. Their comments were valuable in helping us improve the readability and accuracy of this document.

8 Full Copyright Notice

9 Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director (see contact information at GGF website).

10 Normative References

[JSDL10] Anjomshoaa, A., Brisard, F., Drescher, M., Fellows, D., Ly, A., McGough, S., Pulsipher, D., and Savva, A. Job Submission Description Language (JSDL), v. 1.0. GFD-R.056. Available at http://www.ggf.org/documents/GFD.56.pdf.

[BES10] Foster, I., Grimshaw, A., Lane, P., Lee, W., Morgan, M., Newhouse, S., Pickles, S., Pulsipher, D., Smith, C., Theimer, M. OGSA Basic Execution Service, v. 1.0. GFD-R-P.108. Available at http://www.ogf.org/documents/GFD.108.pdf.

[HPCP10] Dillaway, B., Humphrey, M., Smith, C., Theimer, M. and Wasson, G. HPC Basic Profile, v. 1.0. GFD-R-P.114. Available at http://www.ogf.org/documents/GFD.114.pdf.

[HPCPA10] Humphrey, M., Smith, C., Theimer, M., and Wasson, G. HPC Profile Application Extension v1.0. GFD-RP.111. Available at http://www.ogf.org/documents/GFD.111.pdf.