M1.1: Input from SAGA to the regular revisions of the UMD Roadmap

Introduction

The Simple API for Grid Applications (SAGA) aims at providing a simple, stable and standardized client API to support the development of distributed applications and application frameworks. As such, SAGA is contributing to the *Client Capabilities* component of the UMD Roadmap. The SAGA Project in particular intents to provide the following components as input to the next version of the UMD:

1. ***saga-core:*** provides the actual SAGA client API libraries and headers in C++, along with API documentation and a wide variety of associated tools for supporting the application build process, for supporting the development of middleware adaptors, and to interact with distributed infrastructures via the command line.

2. ***saga-bindings-python:*** provides the python language bindings on top of the SAGA C++ API.

3. ***saga-adaptor-globus*** provides bindings of the SAGA API to globus based infrastructures. The provided functionality covers job submission and management, file management, and replica management.

4. ***saga-adaptors-bes*** provides access to a wide variety of middlewares which expose the Basic Execution Service (BES) interface. In particular, that adaptor can interface to ARC, Unicore, Genesis, SMOA, and others.

5. ***saga-adaptors-ssh*** provides ssh based job and file management capabilities to the SAGA API. Although ssh based infrastructures are not easy to scale up, ssh still represents a very pervasive access technology.

6. ***saga-glite*** provides adaptors to the gLite service registry to the SAGA service discovery API, and also an adaptor for gLite job submission.

We follow with great interest the recently increasing interest in adding virtualization technologies to EGI infrastructures. Towards that goal, we plan to increase both API level and adaptor level support for cloud based infrastructures for the following release of the UMD. In particular will we adapt the SAGA resource management API to cater to the EGI virtualization use cases, with the ability to map that API to, for example, OCCI based infrastructures.

It must be noted that the SAGA release cycle, and in particular for releases planned for EGI, depend heavily on the integration with the underlying middleware infrastructures. In the scope of this document, this is specifically globus (via IGE) and gLite/ARC/Unicore (via EMI). The actual release dates for the

respective software packages are as much defined by the IGE and EMI release and integration work cycles as by our own development cycles.

Below we include an updated version of Section 11 of the original UMD Roadmap document, which reflects the current state of the UMD Client Capabilities from our perspective.

11 Client Capabilities

11.1 Client API Capability

Instead of addressing interface heterogeneity on the service level, an alternative approach proposes the abstraction of distributed services on the client side, providing a common interface to client application developers. Adopting a client API may benefit domain specific application developers from evolving middleware while it may all the way easier to maintain a client side API for the most common Grid Use Cases that keeping track of and synchronizing middleware interfaces.

11.1.1 Supported Interfaces

OGF provides the SAGA API as an approach to a common, lightweight and simple API for client-side abstraction of distributed computing resource. The SAGA API itself maps semantically very well onto interfaces which are standardized on a lower level of the middleware stack, such as GLUE, BES, DRMAA, GridFTP, and several others. In general, SAGA can be implemented on any DCI which provides (a subset of) SAGA semantic capabilities.

11.1.2 Implementation Roadmap

This crosscutting capability needs close interaction and synchronization with all distributed service types it interfaces with. Particularly, the implementation of middleware bindings (i.e. adaptors to SAGA) should be under control of the middleware service implementers, a model that has already been adopted for the development of Nagios plugins for the current EGI monitoring infrastructure. The main SAGA implementation in C++ and Python is maintained at the Louisiana State University Center for Computation and Technology (LSU/CCT). It currently support, amongst others, GT5, gLite, Condor and ssh. In addition, the LSU implementation continues to support, to an increasing degree, the OGSA-BES and HPC Basic Profile standards in addition to JSDL. This aligns with IGE's plans to support these standards in GT5.2 and GridWay in 2012.