OSG Blueprint/Architecture Meeting Minutes/Analysis
November 9-10, 2010
Fermilab, Chicago, IL

Attendees (overall)
John Hover (Blueprint Coordinator, ATLAS)
Ruth Pordes (OSG Executive Director)
Miron Livny (OSG Technical Director)
Chander Sehgal (OSG Program Director)
Rob Quick, Scott Tiege, Parag Mhashilkar (OSG GOC)
Alain Roy (OSG Software Tools)
Mine Althunay (OSG Software Tools, OSG Security)
Gabrielle Garzolio, Tanya Levshina (Fermilab)
Ian Fisk (CMS)
Burt Holzman (Interoperability, GIP, CMS)
Brian Bockelman (CMS, U. Nebraska)
Igor Sfigoli (CMS, UCSD)

Top-Level OSG-based Global BDII Instance Proposal

Problem:

CERN's top-level BDII suffers from relatively poor reliability and interoperability problems (relative to OSG grid-level BDII). WLCG proposed establishing a collection of official top-level BDIIs, with several on each continent. For North America the suggestion was one at BNL and one at Fermilab. Michael Ernst suggested that since OSG already provides a grid-level BDII service they should also provide a global BDII, at least for the US.

Considerations and Questions:

Is there really a need that can be concretely met by OSG providing this service? The service will only be useful if various other services and tools can in fact be configured to use it.
What exactly is the update frequency for the VO services that only do periodic queries? E.g. Panda schedconfig and FTS endpoint update.

Discussion:

Scott T. presented a concrete GOC plan to roll out a set of top-level BDII instances.

Observations:

- CERN's BDIIs are presented via DNS round-robin. OSG GOC is considering using LVS (Linux Virtual Server) to allow dynamic maintenance.
- The BDII model is not particularly scalable. As used, it only provides two levels of hierarchy, with many top-level BDIIs all querying the same sources.

Agreed:

Before proceeding, OSG should have clear commitments from VOs to point current services and tools at the OSG BDII instance. (Miron) Versioning and interoperability between CERN/WLCG BDII and OSG BDII will continue to be handled via the Interoperability activity. (Burt)

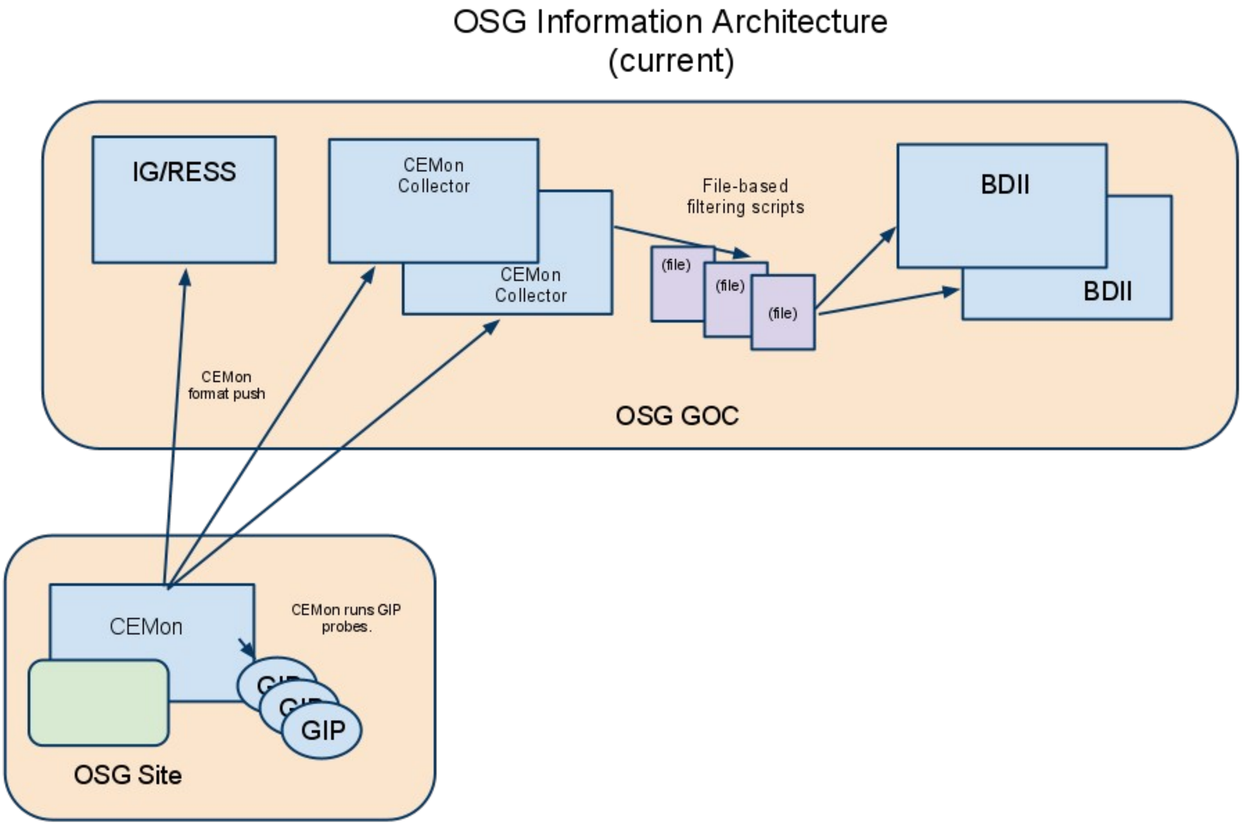
To Do:

- Get assurances mentioned above.
- Assemble more comprehensive list of BDII-dependent services and tools, along with their update frequency and expected load. For each of these determine if they can be configured to use the OSG global BDII either as primary or fail-over.
- Investigate proper way(s) to reduce size of query output (e.g. make GIP probes less verbose).

Future of the OSG Information System

Problem:

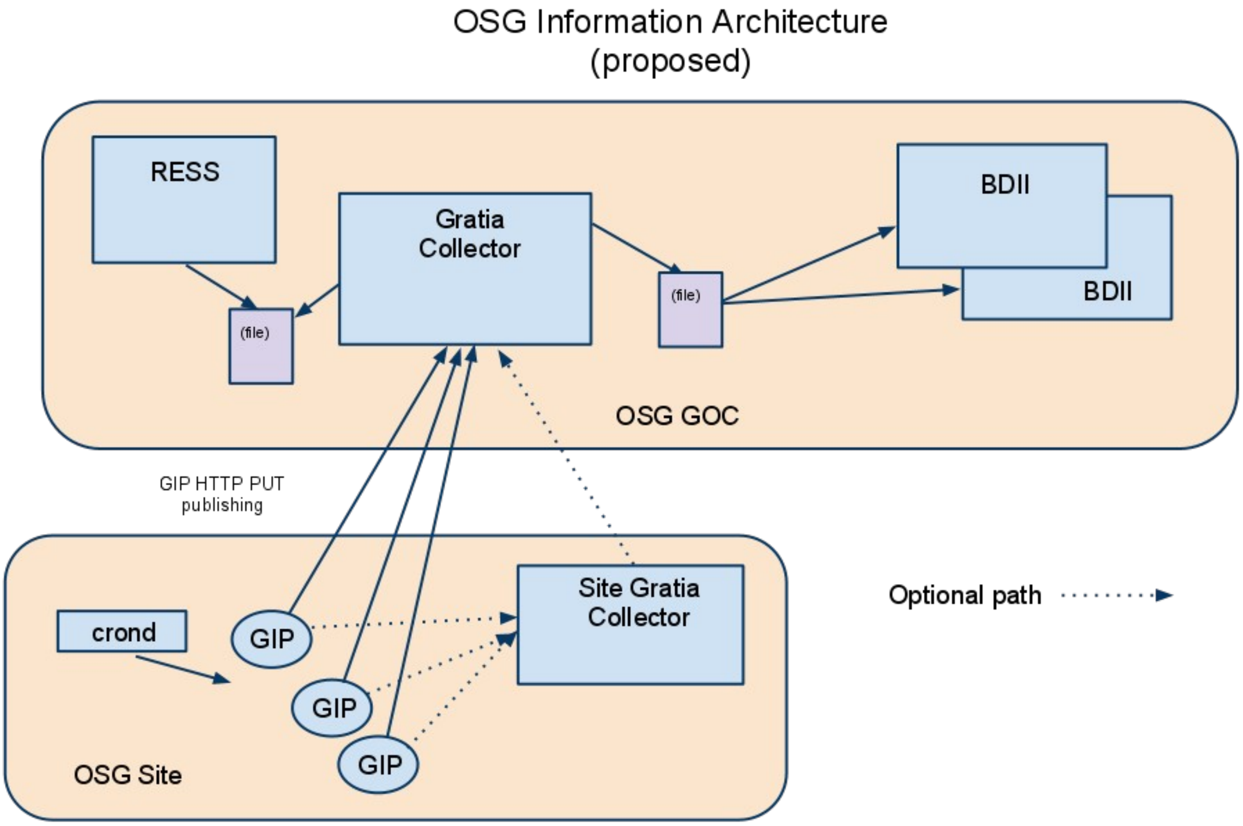
OSG has seen problems with the interaction between CEMon clients (running on gatekeepers) and the CEMon collector/server. There is also an issue (real or percieved) of running such a heavyweight client (a Tomcat/J2EE Java application). This client has had memory usage problems (at BNL), even with 2GB assigned to Tomcat.



https://docs.google.com/drawings/edit?id=15k0rQ11I8RxisUB1IDC3PUnHu7FJ6AcNPq9_eJPewwo&hl=en&authkey=CKq1groD

Proposed:

Brian Bockelman has proposed an information system based on the use of the current GIP and Gratia infrastructure. This new approach consolidates all site-to-central information flow (RSV, Accounting, and BDII info)--now it uses Gratia as the transport for everything. This would leverage the existing Gratia buffer-and-forward logic for records, and reduce by one the software components used by OSG.



https://docs.google.com/drawings/edit?id=1iCkoE9Huzq1lBSxUVRW04nUMuuFlSbJkj8sh-vJcfWQ&hl=en&authkey=COy07pAM

Discussion:

CEMon on the client side simply runs information providers (GIP probes) and publishes the results to the CEMon collector. Does this function really require a J2EE container?

Considerations:

Although the design plans for Gratia included aggregation functionality, at the moment Gratia is limited to filtering. Additional work would need to be done to enable aggregation.

Gratia would also need ot add in whatever record types and store-and-forward modes would be suitable for dynamic state information (as opposed to permanent, unique record style needed for accounting).

A benefit of this approach is that BDII information state could be archived, allowing one to answer the question “What was the BDII output of site X two weeks ago?”

This approach would, however, put Gratia on a critical service path (unlike accounting and RSV) and may involve

To Do:

- Aggregation logic, scalability (if needed), record types, and forwarding logic within Gratia.
- Confirm that GIP info probes currently used by CEMon can publish to Gratia.

Generic Data Management in OSG

Observations:

At the moment, large OSG VOs have written their own, custom data management systems. DQ2 (ATLAS) and PHedex (CMS) handle the management side, with FTS performing reliable transfer and LFC (LHC File Catalog) handling cataloging. LIGO uses its own LDR (Ligo Data Replicator) toolkit, which is built on some off-the-shelf tools (e.g. Globus RLS).

Currently, storage is private to, and managed entirely by the VO, often with semi-custom systems built on top of off-the-shelf technology (dCache, xrootd, etc.).

Problem:

Smaller VOs have struggled to handle input stage-in in a scalable way. Output stage-out is generally simpler since everything typically is transferred back to the VOs central storage.

A related issue is that of opportunistic storage, defined as the ability of a VO to utilize significant storage at non-owned sites. The ability to do this effectively would rely on space reservation and dynamic space allocation at sites. In general this is just too unwieldy with current tools.

So, is there an area where OSG can provide tools or a ready-made system for non-HEP VOs to handle data management and/or opportunistic storage?

Considerations and Discussion:

It may be informative to develop a value metric for data. I.e. the value of stored data is some combination of the invested cost of the CPU and bandwidth to copy it and/or the cost of transferring it again when needed. (Miron)

What about the scaling issue of many small files written into SRM. What about HTTP/S 1.1 with multiple posts? Gratia does this now (Brian). But the standard transfer tools do not.

The major VOs would find detailed storage usage accounting (opens, reads, copies) very useful. (Ian)

Site-level dynamic caching (e.g. via Squid) is one way to get large input to many worker nodes. What about WN-based cluster storage (e.g. hadoop or xrootd) via **glide-ins**? (Miron) This in turn brings up the question of authentication/authorization read/write into this dynamically created storage. The authentication question also applies to Squid for reading (if input data is supposed to be private). Xrootd and hadoop do have auth.

Proposed:

Simple, low-hanging fruit may be making it easy to use Squid-based site cacheing by adding a variable for third-party input transfer for Condor/Condor-G (rather than via sandbox).

Slightly more complex would be xrootd with local cache and external fetching.

POSIX file access is still a concern. Parrot and chirp discussed but no consensus.

To Do:

- Data usage accounting, i.e. number of open()s and read()s, in addition to currently collected data transfer accounting. (?)
- Clean up Squid deployment for stand-alone usage. (VDT)
- Advertise existence, endpoint, and size of Squid cache(s) to BDII along with documentation for this feature. (GIP?)
- Investigate bulk copy operations via SRM client tools?
- Investigate third-party input tranfer via URL for Condor/Condor-G. What about xrootd or hadoop? (Condor team?)
- Continue actively observing the adoption of new storage and caching approaches by large experiments, with an eye toward what might be usable by smaller VOs.

VO-specific testing, Worker node-based tests, and external testing via RSV

Problem:

This topic involves three interrelated issues. First, how does OSG recommend VOs perform VO-specific tests? Second, can/should OSG help VOs perform RSV testing external to their sites? Third, how should we test features/functions that can only be tested directly on worker nodes?

Observations:

VO Testing/External Testing

While SAM used to run via custom scripts, it now is Nagios based. At one time, SAM provided a way for VOs to inject their own tests centrally into the SAM system--test which would then typically be sent as jobs to sites.

The EGEE and OSG testing systems use different technologies, but are functionally similar. EGEE/EGI runs probes with Nagios, and results are fed to ActiveMQ messaging system. OSG runs probes via

Condor-cron, and the probe results are published to Gratia.

New RSV version (3.3) makes probes easier, and is more easily installed stand-alone. This should make it easier for the large VOs to establish VO-specific testing from their T1s or central sites, making OSG-provided central RSV unnecessary. Probe results would be fed to GOC for forwarding to EGEE's ActiveMQ, allowing VO-specific probes to report to SAM similarly to standard probes.

WN Testing

In EGEE, standard convention is that a highest-priority local batch queue is established, and any credentials with Role=lcgadmin gets mapped to a user with access to this queue.

At least for testing basic local batch functionality, an RSV probe already exists which runs on the gatekeeper, submits a job, confirms that it is present in the local batch system (albeit not running), and then removes it. This probe can run rapidly because it doesn't require a WN.

Two aspects to probes which must run on WN:

First, getting probe jobs to run. Options for this are:

- Establish convention for creating low-latency, high-priority batch queues and credentials for accessing them. Or..
- Create mechanisms/infrastructure for cleanly handling probe output with very high latency.
- For pilot-based VOs, they could run WN probes directly from their WMSs, but this would require defining and managing VO jobs which run RSV probes.

Second, handling the output of WN-based probes and defining standards for success/warning/failure.

We need a taxonomy of what we want:

- Probe needs to run on a specific WN (targetted mode, perhaps for special function).
- Probe needs to run on all WNs (i.e. in aggregate, over some period of time. E.g. kernel version or package version test?).
- Probe needs to run on any one random WN (e.g. OSG wn-client test?).
- Probe needs to run (on any node) in as little time as possible (e.g. proxy renewal test job?).

We need a taxonomy of probe output handling.

- Each probe reports centrally to OSG Gratia
- Each probe reports to site Gratia and aggregate results get sent to OSG.
- Each probe reports only to site Gratia (for site admin reports, but not OSG).

Do we need a probe framework on the Gatekeeper (which knows about the population of WNs) to submit these different WN probe types correctly?

Proposed:

For the problem of permitting sites to run tests from outside (in order to account for firewalls and other unpredicatable network phenomena) it was proposed that the GOC provision VMs with pre-installed stand-alone RSV. Site admins would manage and configure the systems, while the GOC would provide the VM and snapshotting for backup.

To Do:

- Investigate the feasibility of VM services at the GOC. (Rob)
- Determine status and feasibility of Gratia aggregation functionality. (Brian?)
- Determine if Gratia probes can easily be defined which run against a Collector, and either publish back to that Collector or publish to external Collector.

Cloud Computing and Virtualization

Problem:

Where and how should OSG support Cloud computing interfaces in its architecture? Where and

Observations:

What do we mean by “Cloud”. At first glance, clouds have several properties:

- Nearly instantaneous, at least when small quantities of VMs are needed.
- Nearly infinite, provided your credit card has a sufficient limit.
- Based on pre-defined virtual machines (VMs) either provided by client or modifed from cloud provider. This establishes a homogeneous, well-defined system platform.
- Often leverages map-reduce to do work.
- Accessed via a clearly-defined API (VM loading, starting, stopping, monitoring, security/credential exchange, node discovery, etc.: all the means clients need to interact with their nodes.)

There are several places and ways Cloud computing might be used in the context of OSG:

- Dynamic OSG sites (1 day, 1 week, 1 month): VO sets up an OSG site on a commercial cloud provider, registers the site in OIM, and runs.
- Adding cloud-provisioned resources to existing OSG grid site. This is handled via whatever batch system is in use, which merges the site-located dedicated cluster and cloud-based cluster. If a grid gatekeeper is added to handle submission, this case becomes the previous one.
- Allowing submission of work, in the form of VMs, to grid sites using Cloud APIs.
- User support for usage of Clouds and Grids intermixed, or with priority (i.e. submit to sites X,Y, and Z, but if they are full submit to EC2). This capability already exists in Condor-G.

Setting aside cloud computing, there is a significant interest from VOs in running jobs supplied as VMs. In the classic grid context this is driven by the difficulty some VOs have in writing their analysis frameworks to run on arbitrary systems (kernel version, GCC version, libc version, etc.)

Various Cloud APIs exist

- Amazon EC2: 2 APIs (EC2 WSDL and EC2 Query)
- Eucalpytus is an open-source (and commercial) EC2 API-compatible cloud service implementation (VMs: Xen, KVM, VMWare)

- OpenNebula is an open source cloud infrastructure that implements some of EC2 APIs along with others. (VMs: Xen, KVM, and VMware)
- Nimbus is another open source project that also supports (VMs: Xen and KVM)

Discussion:

OSG could probably accomodate dynamic sites now. Dynamically adding resources (CE,SE) to existing sites is already trivial, so expanding sites via back-end cloud mechanisms is already feasible. All that is necessary is to think about implementing automatic OIM registration of CE, SE, BDII,etc.

The large VOs with pilot systems can already (probably) extend their approach to clouds, by creating worker-node VMs with their WMS clients already installed. (e.g. ATLAS with the CERNVM project).

To what degree can Condor glideIns provide the kind of capabilities that cloud computing is intended to enable?

Constraints:

There has not yet been sufficient convergence on Cloud APIs in the commercial realm, nor maturation of open source implementation of Cloud APIs, for OSG to pick an example and push for support. Likewise, in the underlying virtualization platforms, there is no clear winner (with Xen, KVM, VMWare, and others providing non-interoperable bases for running VMs).

What security issues/concerns are introduced if VMs enter common use on OSG?

Information Needed:
Proposed:

In the face of competing VM/Cloud technologies, perhaps OSG could research and/or provide tools to allow VOs to convert VMs between the various formats?

Should OSG provide EC2-compatible pre-defined OSG WN-client VM images, *a la* CERNVM?

To Do:

- Continue looking for convergence in cloud APIs and open source implementations.
- Investigate whether there is a particular VM technology that is ready to be adopted as an official OSG site raw VM platform.

Directions in Identity Management

Problem:

Identity management and user credential management remain a source of frustrations both for VOs, site administrators, and end users. The classic (e.g. DOEgrids CA plus OSG/VO RA) grid credential process is too slow and difficult to make well-secured.

Observations:

CILogon is a developing technology for generating user certificates based on institutional identites (typically LDAP). It allows the user to take their institutional username/password account and create a valid grid-usable certificate.

Interestingly, all LHC collaborators can already get certs from the CERN Shib provider. (Brian)

From a purely security-minded standpoint, it may actually be *more* secure that the existing, more ad-hoc RA system, because institutions usually have well-developed face-to-face identity confirmation.

Considerations:

What if a site doesn't like the new CILogon-style RA and wants to refuse to trust certificates issued via that path?

Currently the certificate signing policies are not first-class attributes of certificates. Should they be made so?

Can CILogon be implemented to generate both .pem files and the combined .p12 file.

Agreed:

OSG should be mindful of changes in trust models/basic security (such as that presented by CILogon) and provide ways for sites to make choices about access. GUMS could be used to do this if the user DN can be made to reflect the RA distinctions.

OSG should look into setting up a CILogon-compatible LDAP for generic users (e.g. under osg or engage) to generate certificates. (Miron)

To Do:

- CILogon-compatible LDAP for OSG.
- ?