Future Generation Computer Systems 56 (2016) 571–583

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Globus Nexus: A Platform-as-a-Service provider of research identity, profile, and group management

Kyle Chard[a,*], Mattias Lidman[a], Brendan McCollam[a], Josh Bryan[a], Rachana Ananthakrishnan[a], Steven Tuecke[a], Ian Foster[a,b,c]

[a] Computation Institute, University of Chicago and Argonne National Laboratory, Chicago, IL, USA
[b] Department of Computer Science, University of Chicago, Chicago, IL, USA
[c] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA

HIGHLIGHTS

- Globus Nexus provides hosted identity, profile and group management capabilities.
- Globus Nexus is used by thousands of researchers across the world.
- We describe seven applications that leverage Globus Nexus as a platform.
- Our cloud-based deployment model provides high reliability and availability.
- We show that Globus Nexus can scale well beyond our current requirements.

ARTICLE INFO

Article history:
Received 2 March 2015
Received in revised form
1 August 2015
Accepted 2 September 2015
Available online 24 September 2015

Keywords:
Globus Nexus
Platform-as-a-Service
Identity management
Identity federation
Group management

ABSTRACT

Globus Nexus is a professionally hosted Platform-as-a-Service that provides identity, profile and group management functionality for the research community. Many collaborative e-Science applications need to manage large numbers of user identities, profiles, and groups. However, developing and maintaining such capabilities is often challenging given the complexity of modern security protocols and requirements for scalable, robust, and highly available implementations. By outsourcing this functionality to Globus Nexus, developers can leverage best-practice implementations without incurring development and operations overhead. Users benefit from enhanced capabilities such as identity federation, flexible profile management, and user-oriented group management. In this paper we present Globus Nexus, describe its capabilities and architecture, summarize how several e-Science applications leverage these capabilities, and present results that characterize its scalability, reliability, and availability.

1. Introduction

Collaboration has long been a cornerstone of scientific discovery. However, as researchers increasingly move towards multi-institutional collaborations, the task of managing user identities and groups often becomes a significant burden for application developers and system administrators. Typically, collaborations resort to developing custom software to provide these capabilities to their community, in some cases building upon and re-purposing existing collaboration hubs [1] or science gateways [2]. While these efforts are often considered successes, they require significant investments, both financially and in terms of developer effort, to develop and support. Moreover, these ad hoc implementations often do not follow best-practices approaches to security (e.g., they store cleartext passwords) and create identity "silos" used only for the duration of a project.

While collaboration systems are used in many settings, scientific collaborations introduce unique concerns. They frequently span institutional boundaries and use (computational) resources at multiple institutions. For example, the apparently simple task of analyzing a dataset can involve accessing data in multiple storage systems, submitting jobs to a compute resource for analysis, and publishing results to a shared repository. When thus crossing institutional boundaries and therefore security domains, collaboration members typically require separate identities for each domain in

* Corresponding author.
E-mail addresses: chard@uchicago.edu (K. Chard), mattias@uchicago.edu (M. Lidman), bmccollam@uchicago.edu (B. McCollam), josh.bryan@gmail.com (J. Bryan), ranantha@uchicago.edu (R. Ananthakrishnan), tuecke@uchicago.edu (S. Tuecke), foster@anl.gov (I. Foster).

http://dx.doi.org/10.1016/j.future.2015.09.006
0167-739X/© 2015 Elsevier B.V. All rights reserved.

which they act. To reduce the barriers inherent in having to manage many identities, researchers require a single identity to which they link their various external identities. Similarly, developers require interfaces through which they can outsource identity management and enable Single-Sign-On (SSO) across their resources, while ensuring current best practices are followed.

Scientific collaborations typically manage resource access for their users. At small scales, managing resources on a per-user basis is not difficult. However, as the numbers of users and resources increase, management becomes time consuming and error prone, particularly given the frequency with which resource allocations change. Administrators need to be able to manage groups of users collectively and to use groups to control access to resources. In such environments, administrators and users need to be able to find and identify other users based on descriptive profiles—in order, for example, to share data with colleagues.

In this paper, which extends our previous work [3], we present Globus Nexus, a Platform-as-a-Service (PaaS) approach to providing identity, profile, and group management for collaborative research. Globus Nexus is a professionally hosted service to which researchers and developers can outsource mundane yet crucial identity, profile, and group management tasks. Powerful and intuitive web-based user interfaces and flexible application programming interfaces (APIs) make it easy for both end users and developers to leverage its capacities. Originally developed as the identity management service for Globus [4], Globus Nexus has proven equally valuable to external e-Science applications as it removes the need to develop and support user and group management infrastructure. Globus Nexus is offered as a hosted service operated by the University of Chicago for the research community. In the four years and three months since deployment it has been adopted by several large research projects and currently manages more than 24,000 registered users with more than 8600 external identities, and over 1300 unique groups with more than 6200 combined active memberships.

Here we extend upon our previous work by describing enhancements made to Globus Nexus in the previous year. In particular, we focus on a new relational database-based architecture and compare how these, and other enhancements, have improved performance, availability and reliability of the service. Our new architecture leverages a cloud-based database service hosted by Amazon Web Services (AWS) and therefore removes the need for deployment and operations of data storage, thus significantly reducing internal overhead. We show, via evaluation, that the enhancements made over the past year have also improved performance for complex identity, group and membership scenarios when compared with our previous work.

The structure of the paper is as follows: Section 2 presents an overview of the capabilities provided by Globus Nexus. Section 3 describes how seven e-Science applications leverage Globus Nexus. Section 4 describes the architecture of the system and the scalable cloud-based hosting model. Section 5 evaluates the scalability and performance of Globus Nexus under a variety of realistic usage scenarios. Section 6 presents related work in commercial and academic settings. Finally Section 7 summarizes the paper.

2. Globus Nexus capabilities

Globus Nexus acts as an identity provider and identity hub; provides identity, group, profile, and credential management for users; and supports the creation of customized "branded" sites designed to offer seamless and integrated access to Globus Nexus and other Globus services.

2.0.1. Globus identities

Globus Nexus provides convenient methods for creating and managing user identities. Creating a Globus identity requires only minimal information about the user (username, full name, email address, and password). Globus Nexus applies current best-practices approaches such as password strength guidelines (e.g., minimum length and prohibiting dictionary based words) and username uniqueness. It also supports common workflows such as validating email addresses through an email-based workflow. Having created an identity, users can update and add information to their profile; however they cannot change their username, as it is uniquely identifiable.

In some situations the core set of profile information may not be sufficient to disambiguate users. For example, a username and full name may not be sufficient for an administrator to ensure that they are inviting the correct user to join a group. Globus Nexus implements an extensible attribute model that allows users to associate custom attributes, such as institution or principal investigator, with their profile. These attributes can then be shared with group owners and other Globus users to permit more accurate determination of identity. The Globus Nexus policy model allows users to define who can view identities, profile attributes, and group memberships.

2.1. Identity hub and credential management

Researchers often have multiple different identities that they use regularly to access different resources. For example, a researcher may have accounts at their home institution, their collaborators' institutions, and on resources they use, and also commercial accounts for tools and services such as Google Docs. Globus Nexus acts as a hub for associating these different external identities with a Globus identity, thereby allowing researchers to authenticate using different identities and authentication protocols. The Globus identity becomes an anchor that ties these diverse external identities to a single identity. This functionality enables users of Globus Nexus to authenticate using their campus identity via CILogon/InCommon [5,6] using SAML, their Google account using OpenID [7], or their Extreme Science and Engineering Discovery Environment (XSEDE) account using MyProxy OAuth [8]. Other associated identities include SSH public keys and X.509 certificates, both of which can be used with private keys held by the user to authenticate with Globus Nexus (e.g., via SSH). Importantly, while Globus Nexus stores external identities, it does not store or even see any external passwords or other private data (e.g., SSH private key); rather, it knows only the external identity name so it can redirect the user to authenticate with the appropriate identity provider.

Globus Nexus may cache short term, delegated, or limited proxy credentials created by third-party services. A common example is the limited proxy credential created by a MyProxy server and consumed by a GridFTP server. By caching these ("activated") credentials users may continue to access consuming services (e.g., GridFTP) without having to re-authenticate with the identity provider on each use. Globus Nexus exposes a secure interface for pre-approved Globus services that enable them to retrieve these activated credentials, for example, to conduct a transfer.

2.2. Globus Nexus as an identity provider

As an identity provider, Globus Nexus supports several mechanisms for third-party services to authenticate users via either their Globus identity or any one of their linked identities. The Globus website, itself a consumer of Globus Nexus authentication, and other Globus services rely on username/password authentication and OAuth-based web session cookies. Globus Nexus provides LDAP and OAuth 2 interfaces designed specifically for external third-party services to delegate authentication to Globus Nexus.

The LDAP interface enables third-party integration using standard client tools. It facilitates authentication using a Globus identity and read-only access to profile and group information. While the LDAP interface makes it trivial for third-party services to integrate Globus Nexus authentication, it has several limitations: first, it requires that users trust the third-party service with their Globus password, as that must be relayed via the LDAP interface; second, external identities supported by Globus Nexus cannot be used for authentication; and third, as it is read-only, identity creation and profile updates are not supported.

The second, and preferred, authentication approach uses the OAuth 2 protocol [9]. This protocol is an authorization standard employed by many large web companies (e.g., Google, Facebook, Twitter) to enable clients to access resources on behalf of resource owners. Third-party services can be granted a delegated token that enables access to a user's resources (e.g., profiles or groups) without revealing their password. The protocol is based on a redirection workflow. Using a pre-registered Globus Nexus client account, a third party application can compose a request URL which redirects users to authenticate with Globus Nexus. The request URL includes a redirect URL back to the third party application, after a user has authenticated with Globus Nexus the user is redirected back to the third party application with an OAuth 2 access token included. This access token can then be used by the third party client to access resources on the users behalf. Unlike the LDAP interface, external Globus Nexus identities are also supported via the OAuth 2

Various third-party systems leverage Globus Nexus authentication via both the LDAP and OAuth 2 interfaces, several of which are described in Section 3. Several other third-party applications and services have also been configured to use these authentication mechanisms, including Atlassian Confluence and Jira, Drupal, Wordpress and Zendesk.

2.3. Group management

Resource owners and administrators often require a coarse-grained authorization model through which groups of users can be managed collectively. To support this need, Globus Nexus provides flexible user-driven group management capabilities that allow any authenticated user to create and administer groups. These groups can then be used to control access to data through Globus transfer and sharing capabilities, or can be used by external applications for other purposes, for example to control access to resources or to assemble mailing lists. For example, Globus Nexus groups have been used to implement access control to consortium wikis.

Globus Nexus employs a hierarchical group model. Subgroups can be created within parent groups and group-based policies can be inherited from parent groups. For example, a child group may only be visible to members of their parent group. Importantly, Globus Nexus groups are not identified by name and namespaces. Instead, they are assigned a unique ID and search mechanisms are used to discover groups; thus, groups created by different users may have the same name and are only differentiated via their unique ID or other group properties (e.g., owner or description). This feature means that groups and even subgroups can be completely hidden from other users. Group creators and administrators can control the visibility of their groups and members. For example, groups and members may be visible to the public, to all Globus users, or only to members of the group or parent group. Administrators may also choose who can create subgroups, who can add users to these groups, and who can perform certain roles within the group.

Globus Nexus group management capabilities extend to the workflows and task queues regarding invitation, requests, acceptance, and suspension. Group administrators can configure membership policies, such as who can request membership and how members are approved. They can also select and configure the email templates used in membership workflows. For example, administrators may choose to allow any user to request membership to a group but require that administrators approve individual requests. In this case, any user who discovers the group (or is informed of its existence) can apply for membership; the administrators will be sent an email with a unique acceptance link to approve the request. A task will also be added to the administrator's queue. Alternatively, an administrator can invite existing Globus users to join a group, or invite researchers who do not yet have a Globus identity using only their email address. In the latter case the invitation workflow will send an email to the user enabling them to create a new Globus identity and join the group.

Fig. 1 shows the group management interface for an authenticated user. On the left is a list of all the groups for which the user is a member or administrator; on the right is the member view of a specific group. The members view highlights the different roles in the group, including administrators and users. It also shows different membership workflow states: for example, some users are awaiting acceptance of their invitations, while others have not yet been approved access to the group.

2.4. Branded site deployment

Globus supports the creation of branded sites. A branded site is a complete Globus deployment that is tailored to a particular project's branding. While the underlying Globus Nexus system is shared amongst these sites, branded sites enable different communities to have their own interface to the Globus ecosystem. Globus currently hosts branded sites for research projects such as the Department of Energy's Systems Biology Knowledgebase (Kbase); campus data management systems such as at Exeter University and Indiana University; large compute providers like the Open Science Grid (OSG) Connect and the National Center for Supercomputing Applications' (NCSA) Blue Waters supercomputer; and campus research computing providers such as the University of Chicago's Research Computing Center (RCC). It also underpins Globus and the separate Globus Europe initiative. Fig. 2 shows several of these branded sites.

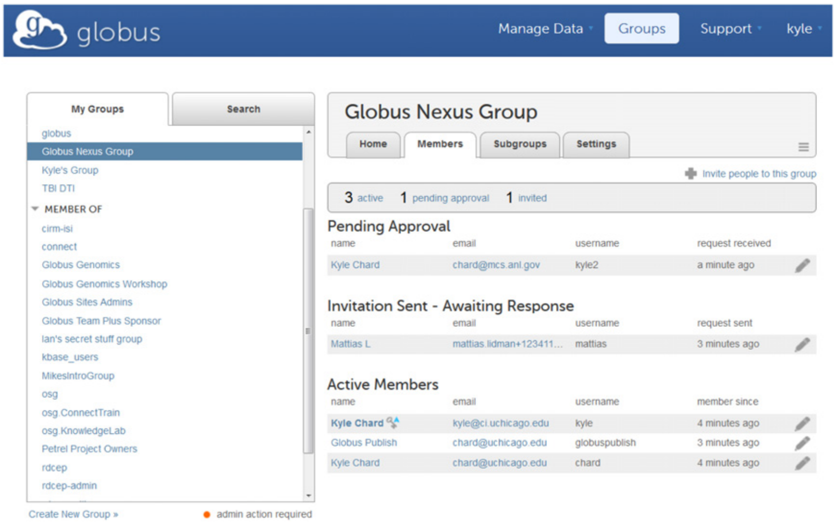
Each community can use their own URL and branding on their Globus instance while still accessing Globus Nexus identity, profile, and group management capabilities. The advantage for developers is that they can preselect valid identity providers for their site, manage policies around access to their site, and also provide integrated access to other Globus services. Each branded site has its own root group that is preselected under that branded view. This feature allows users of the branded site to have a community specific view of the groups and members in Globus Nexus.

3. Scenarios

We present seven scenarios to illustrate some of the ways in which Globus Nexus functionality is used, by ourselves and others.

3.1. Systems Biology Knowledgebase (KBase)

The Department of Energy's KBase [10] is an integrated software and data environment for executing sophisticated large-scale systems biology analyses. It integrates diverse data and tools to


globus

Manage Data · Groups · Support · kyle

My Groups | Search

globus
Globus Nexus Group
Kyle's Group
TBI DTI
MEMBER OF
cirm-isi
connect
Globus Genomics
Globus Genomics Workshop
Globus Sites Admins
Globus Team Plus Sponsor
Ian's secret stuff group
kbase_users
MikesIntroGroup
osg
osg.ConnectTrain
osg.KnowledgeLab
Petrel Project Owners
rdcep
rdcep-admin

Create New Group » ● admin action required

Globus Nexus Group

Home | Members | Subgroups | Settings

Invite people to this group

3 active 1 pending approval 1 invited

Pending Approval

name	email	username	request received
Kyle Chard	chard@mcs.anl.gov	kyle2	a minute ago

Invitation Sent - Awaiting Response

name	email	username	request sent
Mattias L	mattias.lidman+123411...	mattias	3 minutes ago

Active Members

name	email	username	member since
Kyle Chard	kyle@ci.uchicago.edu	kyle	4 minutes ago
Globus Publish	chard@uchicago.edu	globuspublish	3 minutes ago
Kyle Chard	chard@uchicago.edu	chard	4 minutes ago

Fig. 1. The group management interface.

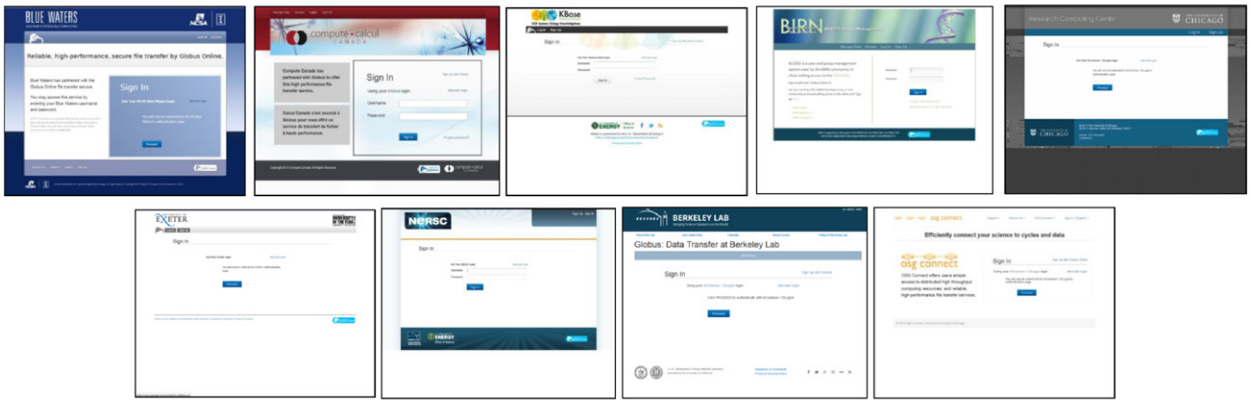


Fig. 2. Some branded Globus sites: Blue Waters, Compute Canada, KBase, BIRN, UChicago RCC, University of Exeter, NERSC, Lawrence Berkeley National Laboratory, OSG Connect.

provide a single system that allows researchers to upload and analyze data, build new models, and share workflows and conclusions. It enables collaborative generation, testing, and sharing of hypotheses about gene and protein functions. KBase also provides large-scale computing infrastructure to enable analysis and modeling of interactions between microbes, plants and their communities.

KBase uses Globus Nexus to outsource identity and group management functionality. Thus, each KBase user has a Globus identity and profile that are used throughout KBase. A branded Globus site allows users to create and manage identities, authenticate using any of their linked identities, and create and manage groups. Users can self-register for KBase using a customized sign-up workflow. Upon registration, users are added to Globus Nexus groups with KBase-specific membership acceptance policies. KBase uses Globus Nexus REST APIs to retrieve user identities and groups, these are then used to configure and provision policies within local compute and storage resources. This model allows KBase to define user- and group-specific admission policies to its resources, data and tools using Globus identities.

3.2. Biomedical Research Informatics Network (BIRN)

BIRN [11] was a national infrastructure designed to enable data sharing and to support multi-institution collaboration for biomedical researchers. BIRN provided a framework that allowed for large scale data to be shared efficiently and securely across distributed computing infrastructure irrespective of security domain. It provided a set of services that enabled distributed heterogeneous database queries as well as construction and execution of analysis pipelines via a graphical interface.

BIRN used Globus Nexus to manage identities and groups. As BIRN was designed to support multi-institution collaboration, it required support for federating identities. Globus Nexus provides this capability via a centralized identity model through which participating researchers can self-register – via a branded Globus site – and subsequently link, and then use, their local campus identity for authentication. Using a customized sign-up workflow, registered BIRN users were placed in a special group, thus allowing administrators to approve membership requests before users were allowed to access BIRN resources. BIRN leveraged Globus identities in its data sharing and transfer services, enabling use of user-defined groups to control sharing and access to collaborative resources and data. BIRN used Globus Nexus REST APIs and LDAP interface to enable user identities and groups to be provisioned across its management and collaboration services; for example, Globus identities and groups were used in the consortium-wide Confluence Wiki.

3.3. CI connect

CI Connect is a web-based platform that provides services for accessing campus, cloud, and national cyberinfrastructure. CI Connect enables virtual extensions of campus clusters via a unique "overflow" model in which additional computing capacity is used when required. It also provides methods for bridging cyberinfrastructure across institutions. CI Connect provides a number of services to users such as Globus for data transfer, Stash high-volume file storage for supporting workflows that span local and distributed computing resources, Open Science Grid (OSG) high throughput computing services for executing large-scale computations over shared and federated computing resources, and an integrated portal for accessing these capabilities. Currently there are six instances of CI Connect supporting national cyberinfrastructure (OSG), large-scale experiments (ATLAS, CMS) and three research institutions (UChicago, Duke, Michigan).

CI Connect leverages Globus Nexus to manage identities and groups, and to provide an authentication infrastructure that integrates with campus identities and that can be used to control access to cyberinfrastructure. Using Globus Nexus APIs, CI Connect provisions Unix accounts for each identity registered with a CI Connect instance (based on Globus Nexus group membership). Each Unix account is provisioned with the user's SSH public keys obtained from linked SSH identities managed by Globus Nexus. This approach allows users to SSH to CI Connect resources using their linked SSH identities. Unix groups are also provisioned for each Globus Nexus subgroup associated with a particular CI Connect instance, these groups are used to provide group-based access control to data and resources. CI Connect uses Globus Nexus to manage over 500 user identities across 140 groups with over 2000 combined memberships.

3.4. FaceBase 2 consortium

FaceBase [12] is a national data sharing network that supports craniofacial development and morphology research. The FaceBase data portal contains several terabytes of contributed data, ranging from genomic to imaging data and including both protected and non-protected biomedical data. FaceBase also provides web-based tools for visualization and analyses and a collection of community curated information. FaceBase currently serves a community of more than 2000 registered users.

FaceBase uses Globus Nexus to manage identities and groups. Through a branded Globus site users are able to self-register to gain access to FaceBase. Using a customized sign-up workflow, every FaceBase registration results in a membership request to join the FaceBase group. FaceBase administrators can view these requests, interrogate profile attributes (e.g., institution and project), and then approve or reject memberships. Users must have approved memberships before they are able to access FaceBase data. When using the FaceBase data portal, users must authenticate via Globus Nexus, using any of their linked identities, before they are able to download non-protected data. Protected data access requires an out-of-band approval process not managed by Globus.

3.5. Research data archive

The National Center for Atmospheric Research (NCAR) Research Data Archive (RDA) [13] is a large collection of atmospheric and geoscience data. It includes meteorological and oceanographic observations, model outputs, and remote sensing data. RDA is designed to support long-term preservation of data and aims to provide data to a broad community of users.

RDA uses Globus Nexus to provide SSO across their services and to manage access to data. Using a branded Globus site users are able to create a Globus identity and link it with their RDA account. The branded site, and Globus Nexus, supports RDA as an identity provider and therefore allows users to authenticate using their RDA email and password. Thus, the use of Globus identities is (semi) transparent to users. However, when accessing services, such as data transfer, and associating sharing permissions with data the user's Globus identity is used for authentication and authorization. A common workflow supported by RDA manages the sharing of data. At the conclusion of an analysis, a directory is created on a storage server and an ACL is set using the email address of the user. When the user follows the email link and authenticates using Globus Nexus (via their linked RDA identity), the ACL is updated to be associated with their Globus identity for subsequent access.

3.6. Computation institute research sites

The Computation Institute (CI) at the University of Chicago hosts and operates research sites for groups and centers within the institute. These Drupal-based sites allow CI researchers to create, customize, and operate their own web presence using a toolkit designed for rapid development.

Four of these research sites, spanning climate change, theoretical chemistry, and traumatic brain injury research, use Globus Nexus to manage identities and groups. All users accessing these sites follow an OAuth 2-based workflow to authenticate with Globus Nexus before being able to access administrator functionality, restricted content, or author content on the site. Globus Nexus groups are used to restrict access to individual research sites, so that only site members can log in. Groups are also mapped to roles (e.g., contributor, editor, and administrator) in Drupal to further restrict user capabilities. The CI research sites rely on the Drupal OAuth 2 module to integrate Globus Nexus authentication and to map Globus groups to Drupal roles.

3.7. Globus Galaxies platform

The Globus Galaxies platform [14] is an integrated suite of services that provide scalable analysis pipeline creation and execution on the cloud. Across its 28 active deployments, including climate change, cosmology, materials science, and genomics, the platform serves more than 300 researchers across 30 institutions. The Globus Galaxies platform addresses the challenges associated with running analyses at scale by providing state of the art analysis algorithms, sophisticated data management tools, a graphical workflow environment, and an elastic cloud-based infrastructure for scaling analyses.

The Globus Galaxies platform leverages Globus Nexus for identity management and authentication. This integration means that users can log in with any of their supported linked identities. Globus Nexus groups are used to control access to which Globus Galaxies instance a user may access and for supporting collaborative group-based access to data, resources and workflows. Globus identities are also used to (transparently) authenticate with Globus transfer when moving data to and from Globus Galaxies deployments.

4. Architecture and implementation

Conceptually Globus Nexus maintains a graph in which user identities, linked identities, groups, and profiles are nodes and is-owner-of, is-member-of, is-administrator-of, contains and other relationships form the edges. Fig. 3 shows an example of the Globus Nexus graph, which here includes identities, external linked identities, policies, groups and group memberships. In the figure, the unshaded boxes represent *identity* nodes, each of which has attributes describing the identity profile, plus various relationships to other nodes: for example, to other *linked identity* nodes, one per unique linked identity, and to *policy* nodes, used to describe visibility. *Group* nodes have typed relationships to identities that define the type of membership (admin, member, owner, etc.)

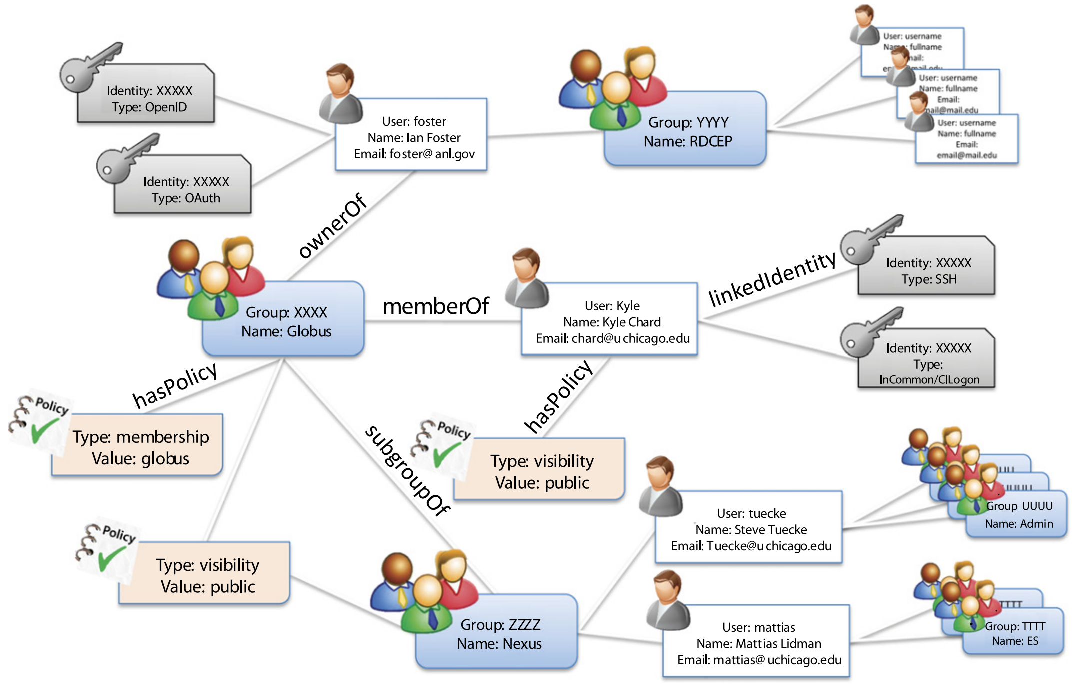


Fig. 3. An example Globus Nexus graph.

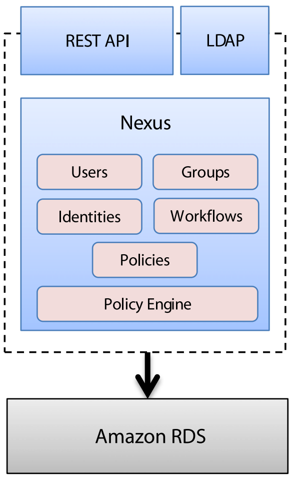


Fig. 4. Globus Nexus architecture.

and membership status (active, admin, suspended, pending, etc.). Group nodes, like identity nodes, also include associated attributes and relationships to policy nodes, which in their case define group workflows, visibility, and attribute requirements.

Fig. 4 shows the high-level architecture of the Globus Nexus service. The architecture relies on a stateless Python Nexus service that exposes various interfaces including REST APIs and LDAP. The stateless service is designed such that it can be scaled horizontally to meet increasing demand. All state is stored in a database hosted on AWS Relational Database Service (RDS). As Globus Nexus is relied upon by thousands of researchers it is important that the service be highly available, scalable and secure.

The Globus Nexus architecture addresses several important technical challenges. Most importantly, identity management, authentication, and authorization are crucial components of any collaborative application and as such Globus Nexus must be online for other components to function, it is therefore crucial that Globus Nexus is highly available and highly reliable. Of equal importance are the need to securely and reliably store data, following best practices approaches to avoid potential compromise and implementing fail-over techniques in case of unforeseen events. Given our focus on research communities, Globus Nexus must address the needs of researchers; examples of these needs include: flexible user profiles supporting research attributes (e.g., institution, PI, etc.), support for common research identity providers (e.g., XSEDE, NERSC, InCommon/CILogon), and flexible security models. These requirements pose unique challenges, such as requiring integration with research identity providers and their associated security protocols (e.g., MyProxy). From a security perspective, Globus Nexus must support fine grain security policies (e.g., profiles, groups attributes, group memberships) that can be changed and reflected immediately. These policies may be expressed in different manners, for example, users may keep their profiles private, make them public, or make them accessible to others with shared group memberships. Researchers have expressed requirements for opaque naming models (e.g., group UUIDs) to hide the existence of names from users whom do not have permission to see them. Such requirements necessitate access controlled search capabilities to discover groups and users.

In the remainder of this section we describe the architecture and implementation of the Globus Nexus data model, data storage, service model, interfaces, and deployment model. Throughout this section we refer to differences between our relational database-based architecture and our previous NoSQL-based architecture [3].

4.1. *Data storage*

We investigated a range of implementation approaches to storing the Globus Nexus graph, including various relational databases, NoSQL databases and the Neo4J graph database [15]. We used a NoSQL-based approach for the first four years of operation but have recently migrated to a relational database model.

In our previous architecture we selected a NoSQL-based model due to its support for schemaless data storage and focus on availability and reliability. We explored the use of Neo4J but choose not to use it as it did not provide a Python client library at the time. We used Apache Cassandra [16], as it provided high availability, tunable eventual consistency, fast write performance, and high throughput when compared with other NoSQL and SQL databases [17]. As Cassandra lacks support for rich search, we also augmented the data storage model to leverage Elastic Search [18], a distributed document-oriented data store. Due to the lack of support for managed databases at the time we instead developed and maintained a distributed cluster of Cassandra and Elastic Search instances—in total five running instances (three Cassandra and two ElasticSearch nodes) supported Globus Nexus.

After Amazon released support for PostgreSQL via their RDS service in 2013, we investigated its capabilities and suitability for Globus Nexus, with a particular focus on the availability and reliability benefits that might be obtained from a managed database service. We concluded that RDS was not only viable but in fact would provide enhancements in terms of ACID properties, support for extensible JSON-based columns, and integrated search capabilities. In addition, a hosted relational database provides features that significantly reduce development and operations burden, such as elastic scalability, automated backup, and automated failover when deployed in a multi-availability zone configuration.

Since this time we have undertaken a re-write of the Globus Nexus implementation to support a relational database model. Globus Nexus now operates on a multi-availability zone RDS deployment running PostgreSQL 9.3. The entire Globus Nexus graph is stored in this database. Schemaless attributes associated with nodes (such as user profiles) are stored in JSON columns which are parsed by the Globus Nexus application when accessed. We leverage PostgreSQL's full text search support on user and group attributes to enable access controlled rich text and faceted search capabilities.

Our new relational database model benefits from the ability to use transactions and ACID properties rather than relying on Cassandra's model of eventually consistency. In addition, we have been able to significantly simplify our query model using optimized SQL queries to improve the performance of various operations (Section 5). We are also able to enforce data constraints at the database level, rather than relying solely on application logic. Most importantly, we have reduced our database operations workload significantly via the use of a managed service which removes the need to deploy and manage five instances from the Globus Nexus deployment model.

4.2. *Data model*

The Globus Nexus data model maps the graph to a series of relational tables that model typed nodes (users, groups, credentials, policies) as well as typed relationships (member, owner, etc.). Tables are associated with one another using foreign key relationships. We leverage an Object-relational mapping (ORM) model to create, access and manipulate data. We use a number of indexes specified on attributes associated with nodes and relationships to enable efficient retrieval of both. For example, indexes are used to retrieve Globus identities based on an external linked identity.

Our previous architecture built on Agamemnon [19], a custom Python-based graph library for Cassandra. Agamemnon provides an API to support general graph operations such as creating typed nodes and relationships. It also supports the creation of arbitrary, schemaless attributes to be associated with nodes and relationships. The major reason for using Agamemnon was to provide a high level graph abstraction for adding and querying data rather than needing to interact with low-level NoSQL APIs. While Agamemnon provides these advantages it also has its limitations. In particular, it is not optimized for data retrieval and was found to be inefficient when retrieving nodes due to its aggressive proactive caching of relationships. Our move to a standard ORM model and SQL-based queries has further reduced the technical knowledge required when building and maintaining Globus Nexus.

4.3. *Globus Nexus service*

Globus Nexus is implemented as a Python application based on the Pyramid web framework. It exposes a REST interface and resource model through which all clients, including the Globus web site, interact. The Globus Nexus service is stateless, that is, all data is stored in an RDS database rather than in the application itself. A load balancer is used to distribute load across a pool of Globus Nexus services. This pool can be scaled elastically without disrupting service, by adding and removing nodes from the load balancer.

The Globus Nexus service implements a RESTful resource model for each major component, including identities, groups, memberships, policies, profiles, and linked identities. Each resource is mapped to its own unique model and control logic is used to implement resource-specific workflows, such as email address validation or group membership actions.

An overarching authorization model governs access to each resource. As the result of authentication, users are given a token that must be presented for subsequent requests. All requests are validated and the user's identity is associated with the request. Each request is then mapped to a resource based on the request URL. Permissions are checked to determine if the user is able to perform the requested action (get, create, update, delete) on the resource. If the user does not have the permissions required to perform the requested action on the given resource, the request is forbidden. Otherwise the request is passed on to the specific resource to perform the action and a response is returned to the client.

For many Globus Nexus access decisions, the computation of permissions is simple—as, for example, when only an owner is able to access their external identities. However, Globus Nexus also supports the definition of dynamic policies related to actions on identities, groups, and memberships. For example, users can set account privacy settings on their profiles that allow other users to view certain pieces of information based upon group membership or upon roles within groups; group owners can set required profile attributes to join a group; and groups may inherit policies from parent groups. In addition, users may delegate access to their resources using the OAuth protocol. This feature allows external services to access user resources (profiles, groups, etc.) that are managed by Globus Nexus.

To support these complex delegated, inherited, and user-defined policies, Globus Nexus implements a policy engine that dynamically evaluates a series of policies to determine if a user has permission to perform a particular action. Policies that forbid access are given priority over those that allow access. While these access control decisions concern the Globus Nexus service itself, external services may access and use Globus Nexus identities, profiles, and groups to make their own authorization decisions.

The Globus Nexus application is also responsible for encrypting information stored in the graph. While some information is not encrypted (e.g. profile attributes used for search), all information related to external identities and activated credentials is encrypted. Following an extensive external security review [20] Globus Nexus has been extended to apply the latest best-practices security implementations across the service and deployment environment.

4.4. *Globus Nexus interfaces*

The primary interface to Globus Nexus is the Globus website, located at www.globus.org. This interface is served using a customized web framework based on RequireJS and Backbone.js to construct dynamic web pages. Every page is composed of several independent "widgets" that provide dynamic features on the page. The widget-based architecture supports reuse across

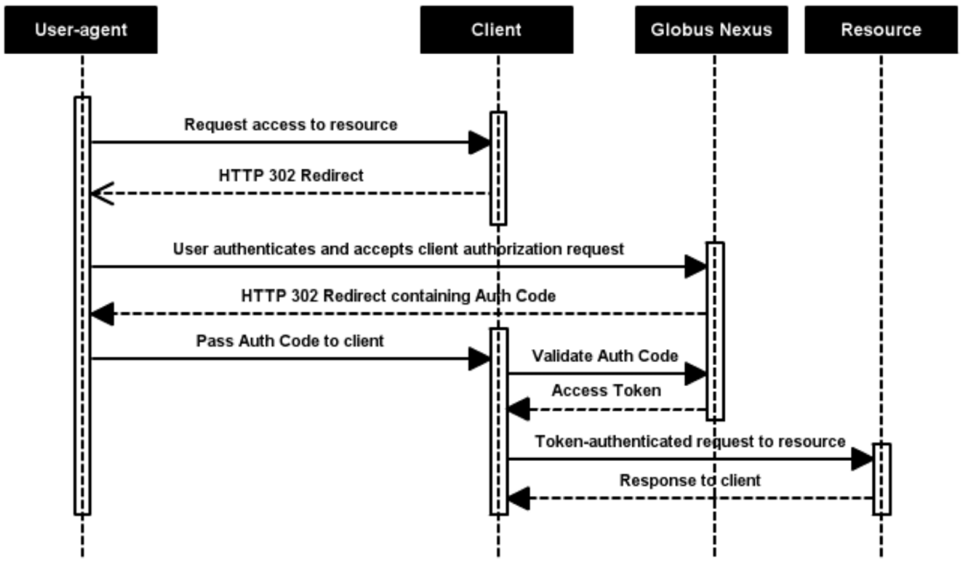

User-agent
Client
Globus Nexus
Resource
Request access to resource
HTTP 302 Redirect
User authenticates and accepts client authorization request
HTTP 302 Redirect containing Auth Code
Pass Auth Code to client
Validate Auth Code
Access Token
Token-authenticated request to resource
Response to client

Fig. 5. Sequence diagram of Globus Nexus implementation of the OAuth 2 protocol.

pages. Widgets are built using JavaScript and jQuery and have their own HTML and CSS to control style and layout. All communication with Globus Nexus uses AJAX requests passed directly to the REST API with OAuth 2 access tokens stored in stateful session cookies.

The REST API provides a programmatic interface to Globus Nexus, enabling users to interact with features of the service. Java and Python programming language clients are also provided to simplify access to the REST API. Globus Nexus implements a subscription interface that enables third party services to subscribe to particular events, such as identity creation and modification. This interface is only accessible to pre-approved services, such as the Globus transfer service [21], which uses this information to create user accounts mapped to Globus identities.

The LDAP interface is developed as a standalone server that runs on Twisted, an event-driven networking engine written in Python. The LDAP interface allows users to bind using their Globus identity username and password. After successful binding the directory can be explored using common LDAP search commands. The LDAP server translates directory requests to the Globus Nexus REST API on behalf of the user that is bound. Users and groups are mapped to their own Distinguished Names (DNs) and memberships are translated to reflect the LDAP DNs.

Third-party clients may utilize Globus Nexus via its public REST APIs and using the OAuth 2 protocol for delegated authentication and authorization. Fig. 5 shows a sequence diagram of the Globus Nexus OAuth 2 implementation. Before using this flow, a third-party application must register their client id (a Globus Nexus identity) and a valid redirect URL. The client and URL are white-listed in Globus Nexus to avoid malicious client spoofing. The OAuth 2 flow is initiated by the User-agent (typically a user's web browser) making a request to a Client. The Client responds with an HTTP 302 redirect to the Globus Nexus authorization URL. The User-agent following the redirect will need to authenticate with Globus Nexus and grant the Client permission to act on the user's behalf. Globus Nexus responds with another HTTP redirect containing an *Auth Code* and redirects the User-agent back to the Client server. The Client makes a request directly to Globus Nexus, exchanging the short-lived Auth Code for a persistent *Access Token*. The Client may store this Access Token and use it to make authenticated requests to access resources or take actions on behalf of the user. Access tokens are created with a fixed duration (currently one year) after which they cannot be used to access Globus Nexus. Users may choose to invalidate a token through the Globus Nexus APIs at any time.

4.5. *Cloud-based deployment*

Critical science services such as gateways require infrastructure that can easily scale with increased demand and that is also resilient to failures of individual infrastructure components. Thus, we have designed Globus Nexus to provide a high degree of availability, a goal that we have so far met, as indicated by our achieved 99.94% availability in 2014 and 99.96% in 2013 (including planned and unplanned downtime).

We achieve this high degree of availability by leveraging experiences learned by commercial platform and service providers. Thus, Globus Nexus is deployed on a commercial Infrastructure-as-a-Service provider, Amazon Web Services (AWS). Its implementation leverages numerous AWS services, including Elastic Compute Cloud (EC2), Simple Storage Service (S3), Elastic Load Balancing (ELB), Relational Database Service (RDS), Simple Notification Service (SMS), and Simple Email Service (SES).

Fig. 6 shows the deployment topology that we use to run Globus Nexus on AWS. We deploy three instances of the Globus Nexus services in a redundant manner, all hosted on EC2 instances in the US East region. We use an ELB to balance web and REST requests across registered instances. The ELB enables a high degree of fault tolerance as it seamlessly removes unhealthy registered instances from the pool. If load increases, additional Globus Nexus instances can be deployed behind the ELB to meet demand. This approach also allows us to update running Globus Nexus services without interrupting operations.

We use a db.m3.large multi-availability zone instance to host the RDS database. This instance contains 2vCPUs, 7.5 GB RAM and provisioned IOPS. AWS maintains for the multi-availability zone instance a standby instance in a different availability zone for automatic failover in the event of an outage. In addition to the automated database backups provided by Amazon – which provide point-in-time recovery via database and transaction logs – we store daily (encrypted) backups of the entire Globus Nexus graph in Amazon S3.

We perform monitoring, logging, and intrusion detection across all Globus services. Specifically, we use Nagios to monitor each deployed service instance and to send alerts to the Globus operations team. We use an external logging service to ensure that all logs are stored reliably from each of the running instances; thus, if an instance fails, developers can still investigate the logs generated immediately before failure. Finally, we operate

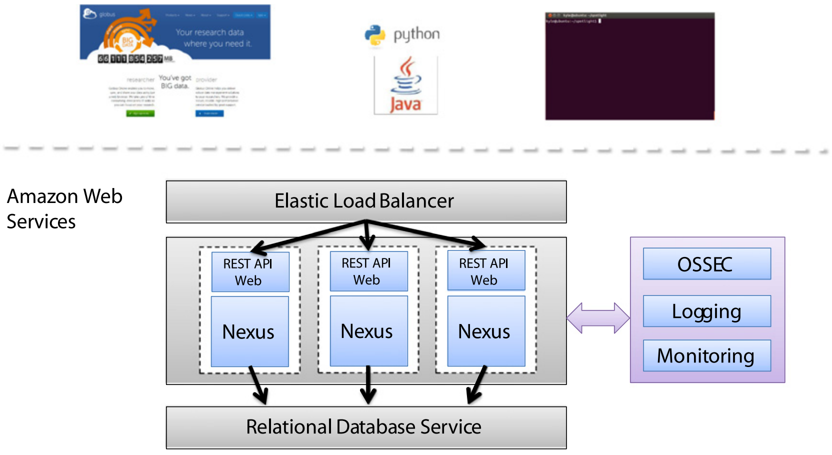


Fig. 6. Globus Nexus cloud deployment.

an Open Source SECurity (OSSEC) service to perform intrusion detection across the deployment infrastructure. As an added layer of security, we use Amazon security groups to restrict communication between instances and the general Internet.

5. Evaluation

To evaluate scalability and performance we emulate an increasing number of user identities and groups. The following experiments explore the performance of our architecture under increasing load. Specific experiments are based on common user operations and queries as determined from analysis of past user interactions with Globus Nexus.

Our tests use a test client that remotely invokes a deployed Globus Nexus instance using the REST API. We hosted the test client and Globus Nexus service on separate small AWS EC2 instances (1 CPU, 1.7 GB Memory, 160 GB Disk). We use a db.3.large RDS instance with 2vCPUs, 7.5 GB RAM, "moderate" network performance, 100 GB General Purpose SSD and no provisioned IOPS running PostgreSQL 9.3.5. We compare the results presented here with our previous results which use the same deployment topology with the Cassandra database hosted on a small AWS EC2 instance [3]. We executed all client requests sequentially over a period of several weeks. Reported results are measured at the client application and include the overhead of REST calls and network latency.

5.1. *Performance of user identity management*

Fig. 7 shows the time taken to create a new identity and to retrieve an existing identity as the size of the graph grows from 0 to 250,000 identities. At each step, a single Globus identity is created and then a random identity from the entire user base is retrieved. We see that creating new identities takes on average 0.48 s (Min: 0.26, Max: 9.77) while retrieving an identity takes on average 0.05 s (Min: 0.04, Max: 1.78). In comparison, our previous Cassandra-based architecture took on average 0.29 s to create new identities (Min: 0.13, Max: 10.81) and 0.05 s to retrieve identities (Min: 0.03, Max: 12.33). Retrieval times are thus roughly equivalent for the two architectures, but creation is 58% slower due to the transactional overhead of SQL databases vs. the optimized write performance of Cassandra. While this change may seem large, the impact on users is negligible as the service still meets response time of less than 0.5 s.

We are also pleased to see that creation and retrieval times do not change discernibly with the number of identities, suggesting that Globus Nexus can scale well beyond the 250,000 unique identities considered in our experiments.

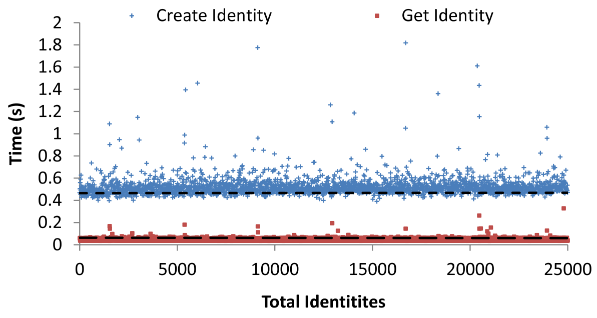


Fig. 7. Identity creation and retrieval time.

5.2. *Performance of group management*

To evaluate the scalability of the Globus Nexus group model we measured the time taken for group management operations (group creation, policy addition, membership addition, group listing) as the number of groups and memberships is varied. In each experiment we use a base configuration with 10,000 registered identities and no existing groups.

5.2.1. *Single group membership creation and retrieval*

We measured the time taken to add memberships (Globus identities) to a single group and to retrieve all memberships of that group, as the number of memberships increases from 0 to 5000. Fig. 8 shows our results. We see that the time required to add a membership increases linearly with the number of members. This increase is due to the growing number of memberships in which a single node has an increasingly large number of relationships. In this case, due to policies on the group – requiring administrator approval for membership creation – each membership created results in a query over all group memberships to find group administrators. In addition to changing our data storage architecture we have also optimized membership creation to avoid unnecessary operations, the 5000th group membership is now created within two seconds whereas previously it took approximately four seconds.

The time required to list all group memberships also increases linearly with group size. This is due to the complex policies available for users and groups. For example, when listing memberships, Globus Nexus must ensure that each membership is visible to the requesting user. We are not currently concerned about this linear scaling behavior, as we expect that users will page results or use search functionality to retrieve specific memberships. Even for large paged results, the time to retrieve 250 memberships is approximately two seconds. This represents a small improvement over our previous implementation that took over three seconds to retrieve 250 memberships.

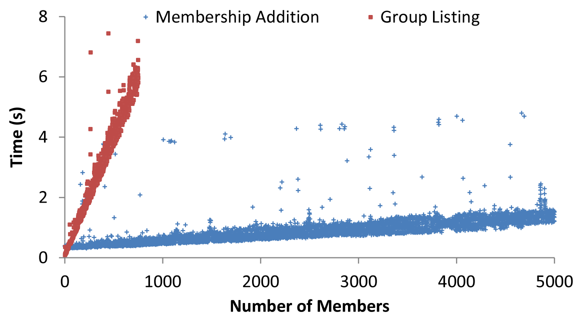


Fig. 8. Membership creation and listing time for a single owner and group.

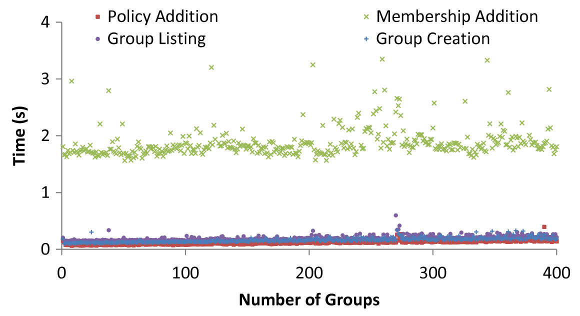


Fig. 9. Group creation, membership creation, policy addition and membership listing time for a single owner.

5.2.2. *Single owner with multiple groups*

We next measured the time taken for a single owner to perform various group operations as the number of groups increases. Specifically, we measured the time required for a user to create a new group, add 10 memberships to that group (the average number of group memberships in Globus Nexus is currently four), associate five policies with that group (approval, join, invites, visibility and member visibility) and retrieve a list of all members of this group. Fig. 9 shows our results.

The results highlight the significant performance improvements from our previous implementation. Here the time taken for each operation remains constant as the number of groups grows. Whereas previously the time taken increased with the number of groups. This behavior was due to unoptimized queries which would retrieve all memberships for the owner before conducting each group operation (even if they were not necessary). The new results show that group creation, policy addition, and group membership listing are completed in less than 1 s. Previously these operations took approximately 5 s for membership listing and 3 s for group creation for the 200th group. While the time taken to create a group does increase with each group created the increase is small (approximately half a second difference between the 1st and 400th groups).

Addition of 10 memberships takes on average 1.9 s for 400 groups and 1.8 s for the 200th group, significantly lower than the more than 8 s in our previous architecture for the 200th group. The majority of this cost associated with membership addition is due to the database accesses required to retrieve and alter the graph. In the case of adding a membership, individual policies regarding user, group and membership permissions (e.g., visibility, joining, etc.) must be checked—and due to individual user policies adding these 10 memberships cannot be batched.

5.2.3. *Multi-group membership query*

We next measured the time required to retrieve an increasing number of memberships for an individual user. In this experiment we created 400 groups with an increasing number of members per group, such that user *i* is a member of *i* groups. A total of 80,200

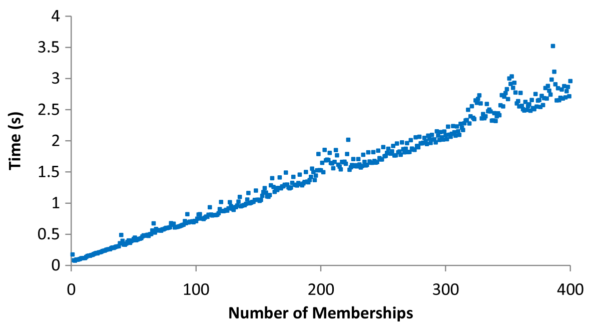


Fig. 10. Time to retrieve and increasing number of memberships.

memberships. We record the time to retrieve all groups for which user *i* is a member, for *i* from 1 to 400. Fig. 10 shows our results. Note, in comparison with our previous results we do not include here a request to check subgroup creation policies, this is an enhancement to the API used by the web interface and is not typically required by API users. Here, we see a linear increase in retrieval time as the number of memberships increases. An individual user can retrieve over 400 memberships in less than 3 s. This is a significant improvement over our previous architecture which took approximately 10 s to retrieve 400 memberships. Performance is again dominated by the time to retrieve memberships (which may be cached) as well as the explicit action to check membership, user and group visibility against the requesting user. Paging and search approaches can be used to improve performance.

5.2.4. *Multi-owner group operations*

Finally, we measured the performance of group operations (group creation, policy addition, membership addition, and membership retrieval) under more realistic group ownership scenarios. In each scenario we vary the number of groups per owner and the number of memberships per group. We start with 10,000 user identities and in each experiment create 10,000 groups across a varied number of owners. Fig. 11 shows our results, using the notation (*groups per owner*, *memberships per group*) so that, for example, the tuple (1,10) denotes one group per owner (and thus 10,000 different owners) and 10 memberships per group. When assigning members to groups, we select them at random from the 10,000 registered identities.

In all four figures there is no obvious performance degradation as the number of groups increases. Group creation, policy addition and group membership listing can all be performed within 0.2 s per operation for each configuration. Adding 10 memberships to a group is performed within 2.5 s. This is a significant improvement over our previous architecture in which these operations, under the same experimental conditions, took 1.2 s for group creation, 0.6 s for policy addition, 0.6 s for membership retrieval, and 3.5 s for membership addition.

Fig. 11(a) shows the time taken to create groups. The results show that the configurations are ordered (roughly) by the number of groups per owner. This indicates that there is some, albeit small, performance impact when increasing the number of groups owned. However, the average difference between all configurations is only 0.028 s. This represents a significant improvement over our previous results where the average difference was 0.42 s.

Fig. 11(b) shows the time taken to add policies to a group. Again, the results are ordered based on the number of groups owned. The average difference between configurations is 0.019 s. The results seem to show a small upward trend, increasing policy addition time with the total number of groups. However, comparison between the average policy addition time for the first 1000 and last 1000 groups across all configurations shows an increase of only

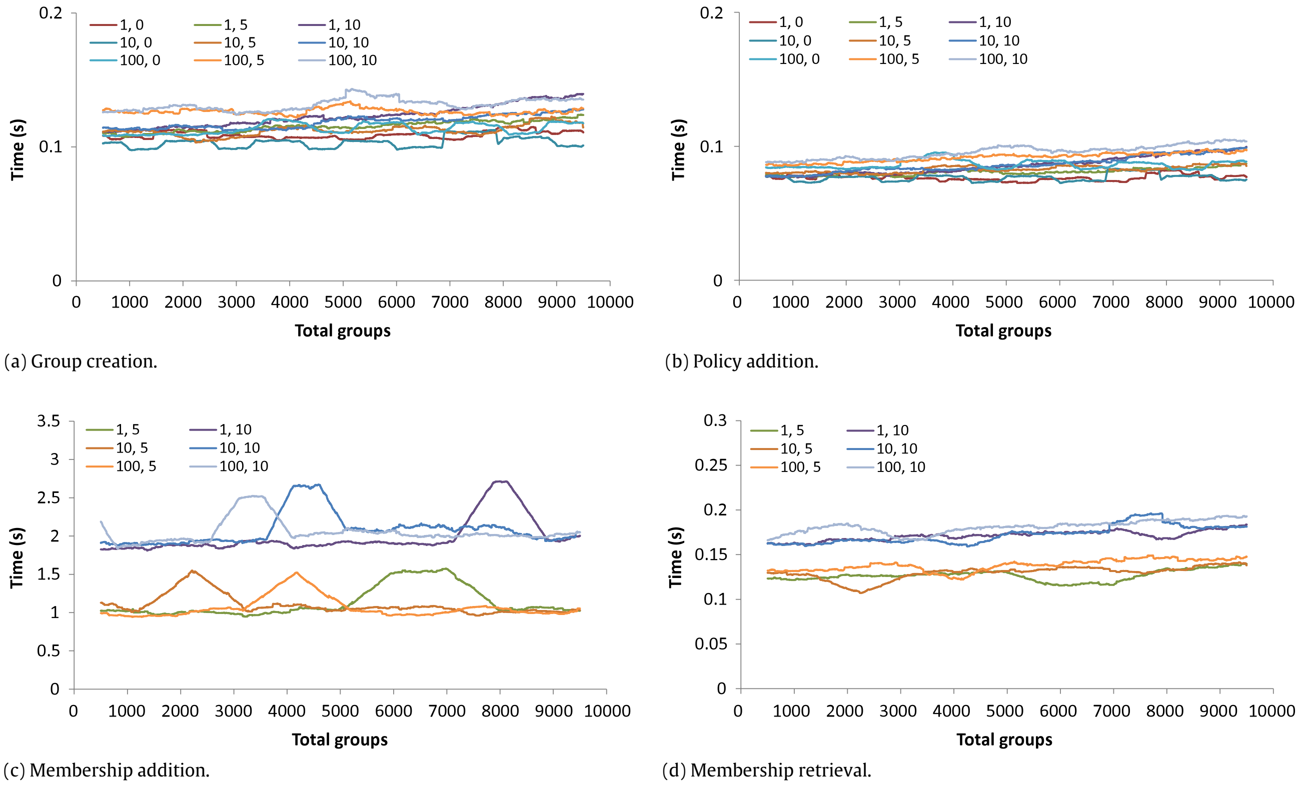


Fig. 11. Per-user group operations with varied number of groups per owner and members per group. Data points are shown using a 1000 point moving average. Series are labeled as (*groups per owner*, *memberships per group*).

0.007 s. Thus, we do not expect there to be a significant increase in the time to add policies as the number of groups becomes large.

Fig. 11(c) shows the time taken to add memberships to a group to be between 1 and 2.5 s. This is much longer than the other operations due to the requirement to check group and member policies for each membership. As expected the operations are grouped based on the number of group memberships added. Adding five memberships takes, on average, 1.09 s. Adding 10 memberships takes, on average, 2.04 s. As expected, the time taken to add 10 memberships is approximately double that required to add five memberships for each configuration. The time to add memberships is approximately 15% faster than our previous results (1.22 s for 5 memberships and 2.41 s for 10 memberships). Importantly, unlike our previous architecture, membership addition time is no longer dependent on the number of groups owned. Interestingly, the results show prolonged periods of time where the membership addition time increases by approximately half a second, for every configuration. Based on the experiment execution time, we estimate these periods to be approximately 30–60 min. We suspect these periods are caused by the RDS configuration used for testing which was not optimized for IOPS or configured for background operations. We are actively investigating these aspects.

Fig. 11(d) shows the time to retrieve group membership. As expected, this figure shows the time to be influenced by the number of members of a group rather than the number of groups owned. The average time to retrieve 10 memberships and 5 memberships is 0.17 and 0.13 s, respectively. This represents almost a 50% improvement over our previous architecture which took 0.31 and 0.24 s to retrieve 10 and 5 memberships, respectively.

5.3. Production deployment

The number of identities registered with Globus has increased steadily over the four years and three months since its initial launch (November 2010–February 2015). Fig. 12 shows the number of registered users over time. As of February 18, 2015, Globus Nexus has 23,911 registered identities, of which 8775 were registered in 2014, compared with 6264 in 2013, 4063 in 2012 and 3159 in 2011. On average, 31 new users registered every weekday in 2014, while only 22 users registered each weekday in 2013. While we do not record how often users use their Nexus identity, we know that between 200 and 300 unique users use it to transfer files using Globus every day.

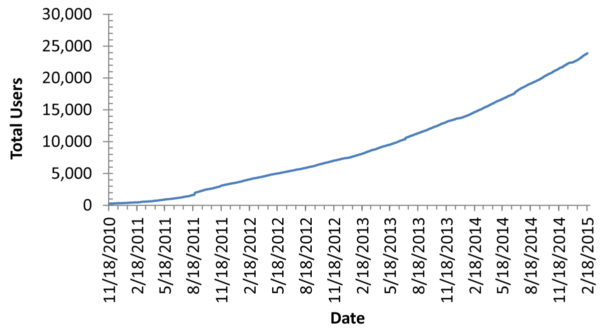


Fig. 12. Total registered identities over time.

6. Related work

Globus Nexus is the first user management service designed and operated for the research community that offers an identity hub model and rich identity and group management capabilities.

Developers of collaborative scientific applications often leverage content management systems such as Wordpress, Drupal, and Joomla, or web frameworks such as Django. Most often, they then use built-in identity and group management capabilities provided by the selected framework. However, the result is then an identity and group management solution that operates in a silo for each implementation and that must be customized for each deployment. It is often technically challenging to integrate external identity providers or to provide SSO for other applications. Challenges include implementing and supporting different security protocols, modifying existing implementations, hosting and operating the service for high availability, and auditing implementations for adherence to best practices. Thus, there can be value in integrating with Globus Nexus, as was done with Drupal for the CI Research Sites described in Section 3.6.

Independent identity management services are often created as a byproduct of authorization services. Many commercial service providers – particularly social networking providers – now offer identity management and authentication as services. For example, Google and Facebook identities are commonly used for authentication in other third-party services, including scientific projects such as Polar Grid [22]. In research domains, the CILogon [5] authentication framework is often used to permit users to authenticate using a campus identity. However, it does not support identity creation or linking.

Commercial identity management systems such as Amazon Identity and Access Management (IAM) [23] and Atlassian Crowd [24] enable identity management across services. IAM is used to manage access to Amazon services and resources via permissions associated with users and groups. IAM supports federation of public identities such as Facebook and Google, as well as linking corporate identity providers such as Microsoft Active Directory, which can be integrated into third party services. IAM's federation model supports any identity provider that implements OpenID Connect. Atlassian Crowd [24] is a service-based identity management service for web applications. It enables user identities to be sourced from external directories and exposes different authentication interfaces that can be embedded in external applications (e.g., OpenID). Neither IAM nor Crowd support common research identity providers.

Many group management and authorization services are also available. For example, Google Groups supports user-defined groups to be used for authorization to Google services. Google Apps for Education provides a number of collaboration tools (e.g., Docs, Groups, Hangouts) that can be integrated with external applications via OAuth and published REST APIs. Amazon IAM and Atlassian Crowd both support the creation and management of user-defined groups which may be subsequently used for authorization decisions. The Virtual Organization Management Service (VOMS) [25] provides group-based authorization capabilities using short-lived proxy credentials. Atlassian Crowd also provides user-defined group support that can be incorporated in external applications. Grouper [26] is perhaps most similar to Globus Nexus in its group management capabilities. Globus Nexus is distinguished by its focus on user-driven group management.

Increasing awareness of the high costs involved in operating reliable research infrastructure has spurred a number of efforts to develop platforms for supporting scientific services. Two prominent examples are Agave [27] and Apache Airavata [28]. The Agave platform provides user management, authentication and authorization, job submission, and data management capabilities, all accessible via REST APIs. Its user management capabilities support the creation and management of users and associated profiles. It supports OAuth 2-based authentication workflows that allow third party applications to leverage its capabilities. However, unlike Globus Nexus, Agave does not support an identity hub or provide for groups. Apache Airavata provides services for supporting science gateways, such as application, workflow, and resource management. These services are accessed via APIs and enable developers to outsource functionality to a scalable and elastic platform. Apache Airavata does not support identity and group management, but instead relies on gateways to provide their own implementations. Thus, an Apache Airavata gateway could use Globus Nexus for identity, group, and profile management while still using Apache Airavata for other tasks.

7. Conclusion

Scientific collaborations invariably require management infrastructure to assign identities to users, manage profiles, and organize groups. While these activities are commonplace, developing and maintaining robust and scalable solutions represents a considerable challenge to researchers and developers alike. In the research community these activities consume valuable time that can be better spent elsewhere.

Globus Nexus provides a Platform-as-a-Service on which developers of scientific services and collaborative web applications can outsource identity, profile, and group management functions. Globus Nexus is professionally operated by the University of Chicago and is architected to provide highly available, reliable, and scalable collaboration management. It features global identity provisioning, an identity hub to link different user identities to a single Globus identity, rich profile management, and user-defined group management. These features are all exposed via powerful management interfaces that support both user and programmatic interactions. Globus Nexus underpins the Globus family of services and is also increasingly used for identity and group management by various scientific projects and research institutions.

Our Globus Nexus implementation leverages a suite of cloud computing services to provide reliability, scalability, and availability. The enhanced relational database-based architecture that we describe in this paper significantly improves the performance of the identity and group operations evaluated; however, we expect that its greatest impact will be the improvements to the overall availability and reliability of the service that it delivers over the coming years. We have shown that Globus Nexus can scale to hundreds of thousands of registered identities and many thousands of groups with thousands of identities and hundreds of thousands of total memberships. These results show that Globus Nexus can satisfy the requirements of complex identity, group, and membership scenarios that well exceed the requirements of the service at present.

Acknowledgments

We thank the Globus team for implementing and operating Globus services. This work was supported in part by the NIH through NIGMS grant U24 GM104203, the Bio-Informatics Research Network Coordinating Center (BIRN-CC), and by the DOE through grant DE-AC02-06CH11357.

References

[1] M. McLennan, R. Kennell, HUBzero: A platform for dissemination and collaboration in computational science and engineering, Comput. Sci. Eng. 12 (2) (2010) 48–53. http://dx.doi.org/10.1109/MCSE.2010.41.

[2] N. Wilkins-Diehr, Science gateways: Common community interfaces to grid resources, Concurr. Comput.: Pract. Exper. 19 (6) (2007) 743–749. http://dx.doi.org/10.1002/cpe.1098.

[3] K. Chard, M. Lidman, J. Bryan, T. Howe, B. McCollam, R. Ananthakrishnan, S. Tuecke, I. Foster, Globus Nexus: Research identity, profile, and group management as a service, in: e-Science (e-Science), 2014 IEEE 10th International Conference on, Vol. 1, 2014, pp. 31–38. http://dx.doi.org/10.1109/eScience.2014.25.

[4] I. Foster, Globus Online: Accelerating and democratizing science through cloud-based services, IEEE Internet Comput. 15 (3) (2011) 70–73. http://dx.doi.org/10.1109/MIC.2011.64.

[5] J. Basney, T. Fleury, J. Gaynor, CILogon: A federated X.509 Certification Authority for cyberinfrastructure logon, in: Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery, XSEDE '13, 2013, pp. 53:1–53:7. http://dx.doi.org/10.1145/2484762.2484791.
[6] V. Welch, A. Walsh, W. Barnett, C.A. Stewart, A roadmap for using NSF cyberinfrastructure with InCommon, in: Proceedings of TeraGrid Conference: Extreme Digital Discovery, TG '11, 2011, pp. 28:1–28:2. http://dx.doi.org/10.1145/2016741.2016771.
[7] B. Ferg, B. Fitzpatrick, C. Howells, D. Recordon, D. Hardt, D. Reed, H. Granqvist, J. Ernst, J. Bufu, J. Hoyt, K. Turner, M. Scurtescu, M. Atkins, M. Glover, OpenID authentication 2.0 specification, 2007. http://openid.net/specs/openid-authentication-2_0.html [accessed Feb 2015].
[8] J. Basney, J. Gaynor, OAuth for MyProxy Protocol Specification v1.0, 2012. https://goo.gl/apSiUB [accessed Feb 2015].
[9] D. Hardt, OAuth 2.0 authorization framework specification, 2012. http://tools.ietf.org/html/rfc6749 [accessed Feb 2015].
[10] The DOE Systems Biology Knowledge Base (KBase). http://kbase.us [accessed Feb 2015].
[11] K. Helmer, J. Ambite, J. Ames, R. Ananthakrishnan, G. Burns, A. Chervenak, I. Foster, L. Liming, D. Keator, F. Macciardi, R. Madduri, J. Navarro, S. Potkin, B. Rosen, S. Ruffins, R. Schuler, J. Turner, A. Toga, C. Williams, C. Kesselman, Enabling collaborative research using the Biomedical Informatics Research Network (BIRN), J. Am. Med. Inform. Assoc. 18 (2011) 4. Biomedical Informatics Research Network.
[12] Facebase. https://www.facebase.org/ [accessed Feb 2015].
[13] Research Data Archive. http://rda.ucar.edu/ [accessed Feb 2015].
[14] R. Madduri, K. Chard, R. Chard, L. Lacinski, A. Rodriguez, D. Sulakhe, D. Kelly, U. Dave, I. Foster, The globus galaxies platform: delivering science gateways as a service, Concurr. Comput.: Pract. Exper. (2015) http://dx.doi.org/10.1002/cpe.3486.
[15] Neo4j. http://neo4j.org/ [accessed Feb 2015].
[16] A. Lakshman, P. Malik, Cassandra: A decentralized structured storage system, SIGOPS Oper. Syst. Rev. 44 (2) (2010) 35–40. http://dx.doi.org/10.1145/1773912.1773922.
[17] T. Rabl, S. Gómez-Villamor, M. Sadoghi, V. Muntés-Mulero, H.-A. Jacobsen, S. Mankovskii, Solving big data challenges for enterprise application performance management, Proc. VLDB Endow. 5 (12) (2012) 1724–1735.
[18] ElasticSearch. http://elasticsearch.org [accessed Feb 2015].
[19] T. Howe, https://pypi.python.org/pypi/agamemnon/0.4.0 [accessed Feb 2015].
[20] V. Welch, Globus Online security review, Scholarworks, Indiana University (2012). http://dx.doi.org/http://hdl.handle.net/2022/14147.
[21] B. Allen, J. Bresnahan, L. Childers, I. Foster, G. Kandaswamy, R. Kettimuthu, J. Kordas, M. Link, S. Martin, K. Pickett, S. Tuecke, Software as a service for data scientists, Commun. ACM 55 (2) (2012) 81–88. http://dx.doi.org/10.1145/2076450.2076468.
[22] Z. Guo, R. Singh, M. Pierce, Building the PolarGrid portal using Web 2.0 and OpenSocial, in: Proceedings of the 5th Grid Computing Environments Workshop, GCE'09, 2009, pp. 5:1–5:8. http://dx.doi.org/10.1145/1658260.1658267.
[23] Amazon identity and access management (IAM). http://aws.amazon.com/iam [accessed Feb 2015].
[24] Atlassian Crowd. http://atlassian.com/softwar/crowd/overview [accessed Feb 2015].
[25] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, K. Lőrentey, F. Spataro, From gridmap-file to VOMS: Managing authorization in a grid environment, Future Gener. Comput. Syst. 21 (4) (2005) 549–558. http://dx.doi.org/10.1016/j.future.2004.10.006.
[26] Grouper groups management toolkit. http://internet2.edu/grouper [accessed Feb 2015].
[27] R. Dooley, M. Vaughn, D. Stanzione, E. Skidmore, Software-as-a-service: The iPlant Foundation API, in: 5th IEEE Workshop on Many-Task Computing on Grids and Supercomputers, MTAGS, 2012.
[28] M. Pierce, S. Marru, L. Gunathilake, T. Kanewala, R. Singh, S. Wijeratne, C. Wimalasena, C. Herath, E. Chinthaka, C. Mattmann, A. Slominski, P. Tangchaisin, Apache Airavata: Design and directions of a science gateway framework, in: 6th International Workshop on Science Gateways, IWSG, 2014, pp. 48–54. http://dx.doi.org/10.1109/IWSG.2014.15.

Kyle Chard is a Senior Researcher and Fellow in the Computation Institute at the University of Chicago and Argonne National Laboratory. He received his Ph.D. in Computer Science from Victoria University of Wellington. His research interests include distributed meta-scheduling, Grid and Cloud computing, economic resource allocation, social computing, and services computing.

Mattias Lidman is a senior software engineer at the University of Chicago Computation Institute. He holds a M.Sc. in Computing Science from Umeå University. He was awarded the Lilla Polhemspriset in 2012 for his thesis on predictive social media analytics.

Brendan McCollam is a software engineer at the University of Chicago Computation Institute working on the Globus project. He has a B.A. from Pomona College and was the recipient of a Thomas J. Watson Fellowship.

Josh Bryan is Director of Engineering at GT Nexus, a cloud-based company specializing in global trade and logistics. Before joining GT Nexus, Josh was a programmer at the Computation Institute.

Rachana Ananthakrishnan is a Product Manager at the Computation Institute in the University of Chicago, and has a Staff Appointment at Argonne National Laboratory. Rachana works for the Globus project, designing and building secure data management solutions for researchers across domains, and help shape the services and offerings of the team. She has worked on security and data management solutions on various projects including Earth System Grid, Biomedical Informatics Research Network (BIRN) and XSEDE.

Prior to this, she worked on the Globus Toolkit engineering team, leading the efforts in web services and security technologies. Rachana received her M.S. in Computer Science at Indiana University, Bloomington.

Steven Tuecke is Deputy Director of the Computation Institute (CI) at The University of Chicago and Argonne National Laboratory, and co-leads the Globus project (www.globus.org) with Dr. Ian Foster. His focus is on the development of sustainable, cloud-based, software-as-a-service data management solutions to accelerate research. Prior to CI, Steven was co-founder, CEO and CTO of Univa Corporation from 2004 to 2008, providing open source and proprietary software for the high-performance computing and cloud computing markets. Before that, he spent 14 years at Argonne as research staff. Tuecke graduated summa cum laude with a B.A. in mathematics and computer science from St. Olaf College.

Ian Foster is Director of the Computation Institute, a joint institute of the University of Chicago and Argonne National Laboratory. He is also an Argonne Senior Scientist and Distinguished Fellow and the Arthur Holly Compton Distinguished Service Professor of Computer Science. His research deals with distributed, parallel, and data-intensive computing technologies, and innovative applications of those technologies to scientific problems in such domains as climate change and biomedicine.