Beyond Data Management with Globus

Vas Vasiliadis
University of Chicago – Globus
Adjunct Associate Professor, Masters Program in Computer Science

vas@uchicago.edu, vasv@anl.gov
Automating Research with Globus Flows

• A platform for orchestrating distributed research tasks
• Flows comprise **Actions**
• **Action Providers**: Called by Flows to perform tasks
• **Triggers**: Start flows based on events
• Extensible via **Action Provider API**
A simple, and very common, use case

1. Transfer data from HPC cluster to...

2. Share data with partners and collaborators

Actions
cryoEM automation

Globus Flows

Transfer
- Transfer raw files

Compute
- Launch analysis job

Carbon!
- Correct, classify, …

Share
- Set access controls

Transfer
- Move final files to repo

Describe
- Search index ingest
How to best enable distributed, remote compute?

Borrow page from data management playbook

- "Fire-and-forget" computation
- Uniform access interface
- Federated access control
- Move closer to researchers’ environments
  - Researchers primarily work in high level languages
  - Functions are a natural unit of computation
Globus Compute

Managed, federated Function-as-a-Service for reliably, scaleably and securely executing functions on remote endpoints from laptops to supercomputers.
Globus Compute transforms any computing resource into a function serving endpoint

- Python `pip` installable agent
- Elastic resource provisioning from local, cluster, or cloud system (via Parsl)
- Parallel execution using local fork or via common schedulers – Slurm, PBS, LSF, Cobalt, K8s
Executing functions with Globus Compute

- Users invoke functions as tasks
  - Register Python function
  - Pass input arguments
  - Select endpoint(s)
- Service stores tasks in the cloud
- Endpoints fetch waiting tasks (when online), run tasks, and return results
- Results stored in the cloud and on Globus storage endpoints
- Users retrieve results asynchronously
User interaction with Globus Compute

1. You request a function be executed on endpoints A and B

2. Globus Compute manages the reliable and secure execution on these endpoints

3. Globus Compute returns results or stores them until requested
Use Case

Executing a bag of tasks, e.g., running simulations with different parameters, executing ML inferences, on multiple remote computers directly from your environment, e.g., Jupyter notebook
Use Case

Constructing and running automated analysis pipelines with data processing steps that need to be executed in different locations.
Use Case

Building new applications and services that seamlessly execute application components or user workloads on remote resources
Scalable data discovery with Globus Search

- Scalable metadata store
- Fine-grained visibility controls
- Schema agnostic → dynamic schemas
- Federated auth integration
- Robust query API
  - GET with URL parameters
  - POST with facets

[Diagram]

User publishes metadata into search index

Globally accessible multi-tenant service

Globus manages metadata access

Users query and discover data of interest

docs.globus.org/api/search
CR3 Portal (simulated data)

- Federated logon using Globus Auth with 1,800+ identity providers
- Google-like text search with facets for filtering
- Variable facets based on source registry index
- Dynamically updating charts as facets change
- Developed using a framework based on the Globus Modern Research Data Portal* design pattern (docs.globus.org/mrdp)

* PeerJ Articles:cs-144 https://peerj.com/articles/cs-144/
Repository data distribution

- Faceted search via data portal
- Enforces fine-grained authZ
- HTTPS download for “small” data
- Managed file transfer for larger data sets

Example: acdc.alcf.anl.gov
Resources

• Web app access: app.globus.org
• Documentation: docs.globus.org
• Helpdesk: support@globus.org
• Mailing Lists: globus.org/mailing-lists