Having it both ways: Bring Data to Computation & Computation to Data with iRODS

Nirav Merchant
The University of Arizona
nirav@email.arizona.edu
http://www.cyverse.org Twitter: @CyVerseOrg
Topic Coverage:

• Motivation/Use case
• Constraints, challenges
• Technology options
• Our solution, early results
• Next steps
CyVerse: Platform Philosophy

• Strive to provide the CI Lego blocks
• Danish 'leg godt' - 'play well’
• Also translates as 'I put together' in Latin
• If desired functionality is not available, the community can craft their own by using and extending CyVerse CI components (like lego blocks)
• Through these extensible and customized platforms create a ecosystem of interoperable tools that benefit the broad community (and not few lab groups)
• Provide the tools to allow community to manage their digital assets (cloud, HPC etc.)
• Improve Computational Productivity
The CyVerse Technology Stack
A Blueprint for Cyberinfrastructure Design

- ** PRODUCTS **
  - Atmosphere
  - Discovery Environment
  - DNA Subway
  - Bisque
  - Third Party Consumers
  - Data Commons

- ** FOUNDATIONAL SERVICES **
  - Data Store
  - Science APIs
  - Auth
  - ElasticSearch

- ** LOW-LEVEL SERVICES, SECURITY, ASSETS, etc **
  - iRODS (federated storage)
  - CAS/Shibboleth/OAuth2 (single sign-on)
  - OpenStack (virtualization)
  - Condor (job scheduling)
  - XSEDE (national CI)

- ** HARDWARE RESOURCES **
  - Cloud Systems
  - High-performance Computers
  - Databases
  - Storage

- ** http://www.cyverse.org **

- ** Ease of Use **
- ** Flexibility **

- ** Ready to use Platforms **
- ** Extensible Services **
- ** Established CI Components **
- ** Foundational Capabilities **

NSF
How is it being used?

• User **build** their own systems (powered by CyVerse components) but managed by them
• **Share** analysis methods, algorithms, data (reproducibility)
• **Consume** specific components (a la carte, Data Store, Atmosphere)
• Directly **use** applications (DE)
• Custom design **appliances** (Atmosphere)
• **Publish** their findings (PNAS, Nature)
• **Advocate** use and build “**your**” community
• Create new **learning material** and courses, special topics workshops
Cohesive Platform for Data lifecycle
The eternal question.....

Data to Compute or Compute to Data
Toolchest

- iRODS
- Condor
- Docker
- Rethinking the role of a “resource server”
Motivation: Data to Compute

• Most of our use cases operated on ~100-200 GB data at a time
• Many of the analysis steps were few cores (~12) and reasonable RAM (~128 GB)
• Tasks were “naturally data parallel”
• Easier to provision, share, scale and maintain “shared nothing” (or not much) computing infrastructure
Our Solution: Data to Compute

- Discovery Env.
- Condor Master (Docker)
- iRODS
- Other Compute infrastructure (HPC, Cloud)
- Condor Worker

Note: Conceptual View
Motivation: Compute to Data

• Moving data to compute not feasible in many cases (100 TB+, large repositories)
• Availability of “fat nodes” (or choice for resource servers)
• Availability of specialized compute with storage systems (Wrangler)
Our Solution: Compute to Data

Note: Conceptual View

Discovery Env.

Condor Master (Docker)

Condor Worker

Condor Worker

Condor Worker

Condor Worker

iRODS

Other Compute infrastructure (HPC, Cloud)
Steps

• Bring in data (choose your method)
• Register the data with iRODS
• Apply the metadata (ipc_data_set=IPCC-WG2)
• Let condor announce it (class ads), also configure limits (num of concurrent jobs, core, ram, space to write output etc.)
• Submit job with class add and let condor scheduler match and manage it
• If you need more, create more copies (replica) and profit
• If you need to send it else where (HPC etc) use glidein and bosco
ireme

IREME is a command-line utility which allows registering dataset(s) with irods, and assigning metadata to those datasets, which are then used with condor’s classads mechanism to match jobs with machines. Ireme is also responsible for orchestrating the process of advertising metadata and datasets present on the condor worker / resource worker, in the form of machine classads.

Usage
-p --path: Physical path of the resource to be registered with irods
-c --coll: Collection name within the irods database where files are registered
-m --meta: Comma-separated meta data tags (key:value pairs) associated with the collection

Example Syntax
ireme -p /home/user/sample_folder -c /tempZone/home/user/sample_coll -m key1:value1,key2:value2
iRODS ClassAds

IRODS_RESOURCE is the classad custom variable which advertises iRODS resource required by the job in the form of metadata tags or dataset name (collection name). The Condor Negotiator matches job classad requirement (metadata or dataset) with classads advertised by the Condor Worker.

Sample Job ClassAd w/ iRODS requirement

Executable=test2
Log=test.log
Output=test.out
error=test.error
log=test.log
+IRODS_RESOURCE="key=value"
Requirements=TARGET.meta_available==true
Queue

Sample Machine ClassAd w/ iRODS Ads

meta_available = isMetaAvailable(TARGET.IRODS_RESOURCE)
STARTD_EXPRS=meta_available, $(STARTD_EXPRS)
Data Flow
Data Flow after Classads Matching

- User Job
- iRODS Meta Data ClassAd
- iRODS Dataset ClassAd
- Condor Master / Negotiator
- Condor Worker / Resource Server
- Condor Worker / Resource Server w/ required iRODS resource
Syndicate: Using CDN & beyond (Edge Computing)