Distributed Modeling System

Exploring ideas to make model usage and data management easier

Shawn Freeman, Northrop Grumman
What is the Distributed Modeling System (DMS)?

- Provides data management over a distributed network
- Provides virtual machines for working with models
- Provides workflow tools for efficient management of model runs and results
- Provides tools for distributed post-processing/visualizing output
Distributed Modeling System

Remote Users

Local and Remote users can access the data system.

Internet

Data system software is responsible for load balancing and distributing resources across all the nodes.

Remote Nodes

Local Nodes

FUSE
Users view the distributed data system as one large disk. Nodes can be added and removed dynamically from the system.

Data system nodes store all or part of data items depending on access frequency and load balancing. Data items can include VMs, data sets, and source code.
Distributed Data Management

- Integrated Rule-Oriented Data System (iRODS)
  - Middleware for managing data and users across a distributed data network
  - Allows custom rules and policies for working with and accessing different data
  - Provides search capabilities for locating resources
Data Management System

- Hadoop distributed data system
  - A fast, efficient distributed data system
  - Provides redundancy and failover
  - Provides map-reduce capability for large data-processing jobs
  - Can be used as a “back-end” for iRODS
Data Management System

- FUSE
  - FUSE provides a native level of access to distributed data systems like Hadoop
  - Mounts like any other Linux mount point on the system
  - Allows users to use standard Linux operations and directory paths to reference data on the network
Virtualization

- Compiling and running models can be problematic
  - System specific settings
  - Scattered input data sets
  - Library dependencies, compiler dependencies, etc.
  - Lack of documentation
Virtualization

- Virtual machines
  - New work enables RDMA/Infiniband in virtualized environments
  - Entire environment for a model can be created and stored
    - All necessary libraries, tools, source, compilers, and settings are pre-installed
  - Allows users to be “root”
    - The user can experiment and customize inside of a sandbox
Virtualization

- Other VM advantages
  - Reproducibility
  - Backups of entire system is just saving a file
  - Sharing/distributing
Combining Ideas

- Data management system hosts pre-created virtual machines as well as data
  - Access controlled by policy/rules
  - Allows the creation of a model “marketplace”
    - “Official” VMs maintained by model group or designated authority
    - User VMs can be submitted for others to use
    - Searchable and retrievable by various methods
Combining Ideas

Possible workflow use case

- User configures workflow
- Workflow pulls down virtual machine
- Workflow places all necessary files (data, configuration, etc.) in a “shared” folder
- Workflow submits run, which uses the VM
- The VM writes any necessary status, logs, and output files to shared folder
Other Ideas

- Common visualization interface
  - Various interfaces to tools for data visualization
  - Integrated with data system for searching and using data
  - Shared library of “scripts” for performing visualizations
Issues

- Security, security, security
  - Security environments can often make collaborative tools difficult or impossible to use

- Sharing Resources
  - Distributed system means distributed resources
Issues

- Who is the maintainer?
  - Who is responsible for the overall system?
  - Who is responsible for determining rules and policies?
  - Who is responsible for maintaining “master” data sets, VMs, etc.?
  - Who is Batman?
Issues

- Metadata
  - DMS needs metadata to help identifying objects stored within (models, VMs, input data, output data, etc.)
  - Individual groups can have their own metadata, but it must map to global metadata standard
  - What’s the standard?