HPSS Treefrog
Introduction
Disclaimer

Forward looking information including schedules and future software reflect current planning that may change and should not be taken as commitments by IBM or the other members of the HPSS collaboration.
HPSS Treefrog Goals

Manage and share data across the life of your mission’s projects, procurements, infrastructure, deployment, user access, and staffing cycles.

Store, protect, and error correct project data across a wide variety of local and remote classic and cloud storage products and services.

Effectively exploit and scale tape and other high latency storage by using data containers to group and store files and data objects!
A Single User Namespace

Managed across industry storage devices and solutions called storage endpoints:
- Cloud
- HSMs including HPSS
- Optical
- Tape
- File system
- Disk
- SSD

Managed across data repositories
- Storage endpoints provide real storage for data repositories.
- Repositories are wholly contained inside a storage endpoint.
Manage Data by Project

- Projects provide the nexus between data management and data organization.
- Administrators manage project policies including:
  - Storage quotas
  - Storage access
  - Service limits
  - Access authorization
- Users store data within the projects and group data within data containers (called managed data sets):
  - Data are shared amongst project members (allowed users)
- Project members will have different roles:
  - Owner, reader, writer, modify, delete
- Data will be owned by the project:
  - Insures data will always have an owner.
  - Allows for easy on and off boarding of users.
Policy Defined Storage Management

- Policies determine how and where data are stored.
- Make multiple copies of data:
  - At ingest from the golden copy
  - After a delay from a managed copy
- Control data recall:
  - Assign primary recall copy
  - Assign failover copies
  - Block recall of copies from storage endpoint requiring administrator authorization
Smart Data Storage

- Manage data containers not individual data objects and files.
- Grouped data will be stored as an immutable collection of files or objects called a managed data set.
- As a bonus, grouping data benefits high latency storage.
  - Decreases the number of tape syncs.
  - Allows for all data to be recalled with a single IO.
- Data will be grouped into date sets using a data retention format.
- The Treefrog interface will make grouping data simple.
Parallel Data Transfer

- **Managed Data Sets** may be broken into smaller fragments.
  - Based on storage policy settings.
  - Fragments are contiguous sections of Treefrog managed data set that are distributed across repositories.

- Maximum degree of parallelism will be based on configuration.
Data Redundancy via Erasure Coding

- Parity fragments will be generated based on storage policy settings.
- The number of fragments that may be recovered will be based on the number of parity fragments created.
More About Storage Policies

- A copy of a data set may be:
  - Stored to a single repository
  - Fragmented to a single repository
  - Fragmented across multiple repositories

- Changing storage policies only moves data when required.
Simple Insertion of New Storage Endpoints

- Copy agent based on Apache Jclouds Blobstore.
  - Copy agent interface will be extensible.
  - AWS, Google Cloud Storage, Azure, and Rackspace already supported.
  - HPSS interface is planned.
- Adding a storage endpoint will be as simple as adding a new Jclouds interface.
Data and Metadata Verification

- Each fragment will be stored with a checksum.
- Treefrog can verify both the metadata and data of managed data sets.
  - Administrators use storage policies to control the verification settings.
- **Metadata Verification** will verify the location, checksum, and size of each fragment in the repository match the value Treefrog has stored.
  - Metadata Verification will not access the data.
- **Data Verification** will verify the checksum of each fragment.
  - Data Verification may access the data.
  - Treefrog will use the built-in verification on storage systems that have it.
  - Treefrog will stage fragments to verify checksum.
All of that in an Extreme Scale Architecture

- Scale-out design allows incremental horizontal growth by adding new servers and devices.
- Load Balancing using HAProxy.
- Agents may run at the client to take advantage of available processing power and reduce store and forwards.
But wait there’s more!!!

In addition HPSS Treefrog will:
- Decrease software development delivery time.
- Decrease software deployment time.
- Enable user installation.
- Increase timely access to trending technology.
- Increase use of trending programming language skills and open software.
- Avoid impact to on-going HPSS core services development.
Treefrog will be an HPSS Interface

HPSS Treefrog interface & services

Cloud, Object & File Storage and Services including LTFS

Massively scalable global HPSS namespace enabled by DB2

RHEL Core Server & Mover computers
Intel Power

Extreme-scale high-performance automated HSM
Disk Tape

Block or Filesystem Disk Tiers

Hardware Vendor Neutral

Enterprise • LTO Tape
IBM • Oracle • Spectra Logic

IBM http://www.hpss-collaboration.org
Treefrog will use Existing Technologies

Existing Products
- Only configuration changes are required

Extendable Functionality
- Open Source code or library

Treefrog Specific Code
- Code specific to the Treefrog application
- Requires from-scratch development
Treefrog will use Existing Technologies

Legend

- **Existing product**
- **Extendable Functionality**
- **TreeFrog specific code**

Endpoints
Questions?