Containerization of HSI/HTAR Clients at NERSC

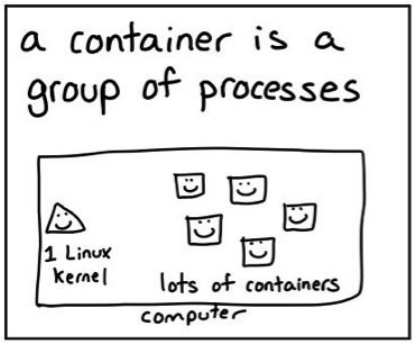
Melinda Jacobsen
Rosario Martinez
Storage Systems Group
October 13, 2021

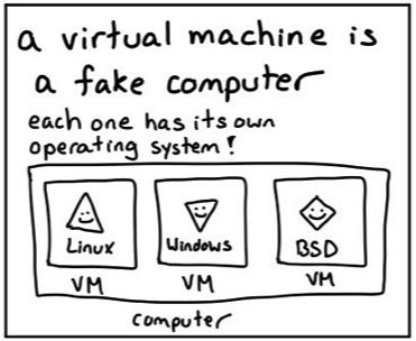
Introduction

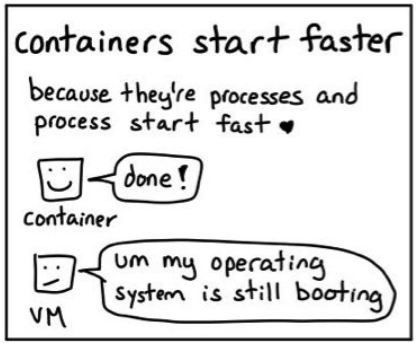
- Containers are increasing in use in modern system architectures
- Perlmutter is the new NERSC computational system
  - Will feature containerized user environments with Kubernetes orchestration
  - Container instances will manage the data transfer process
  - We will look at the design for HPSS access on the system
- Some users require special distribution of HPSS client software
  - Distributing a client image alleviates support and maintenance
  - We will cover results in this area

Container concepts

- Container
  - Lightweight package of an application and its dependencies
  - Makes code portable
  - Can run many - use less RAM than a VM
  - Can use host network or its own network namespace







Reference: Julia Evans, “containers vs. VMs”, https://twitter.com/b0rk/status/1237744128450072578

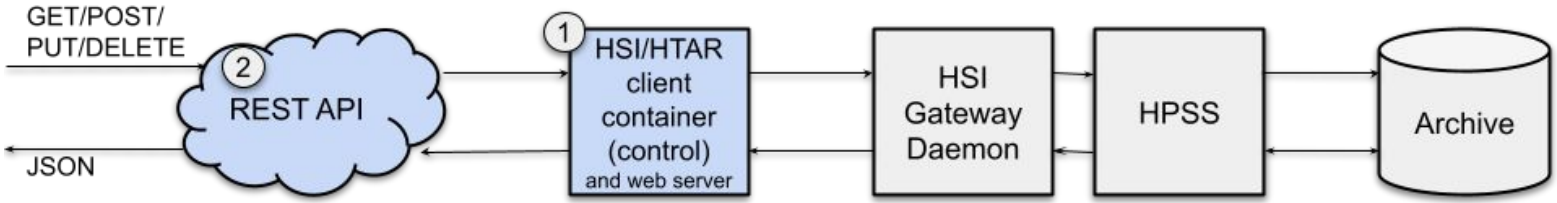
Container concepts

- Container image
  - Executable that creates a container
  - Supports a reproducible container environment
  - Can be stored in a container registry for download

- Container orchestration
  - Microservice: Service that runs in its own container
  - Scale, schedule, and monitor a large number of containers
  - Decides where to run the containers

HSI/HTAR as a microservice

- The new Perlmutter computational system will run a Cray OS featuring containerized instances for managing the data transfer process over a high speed network
- Preparation for integration on Perlmutter:
  1. Demonstrate HSI/HTAR control interface can run in a **container** and move data
  2. Provide a programmatic interface via **REST API** for sending commands to HSI/HTAR in the container



HSI/HTAR in a container

Create an HSI/HTAR client docker image

Base operating system image

HSI/HTAR dependencies

HSI/HTAR software

```
FROM centos:centos7.3.1611

RUN yum -y install \
        openssl-libs \
        glibc \
        glibc.i686 \
        keyutils-libs \
        krb5-libs \
        libcom_err \
        libedit \
        libselinux \
        ncurses-libs \
        pcre \
        zlib \
        openssh-clients

USER root
WORKDIR /root

COPY hpss-hsi-clnt-frontend-nersc-5.0.2.p12-617.el7.noarch.rpm \
     /root/hpss-hsi-clnt-frontend-nersc-5.0.2.p12-617.el7.noarch.rpm
COPY hpss-hsi-clnt-nersc-5.0.2.p12-617.el7.x86_64.rpm \
     /root/hpss-hsi-clnt-nersc-5.0.2.p12-617.el7.x86_64.rpm

RUN rpm -i hpss-hsi-clnt-frontend-nersc-5.0.2.p12-617.el7.noarch.rpm \
           hpss-hsi-clnt-nersc-5.0.2.p12-617.el7.x86_64.rpm
```

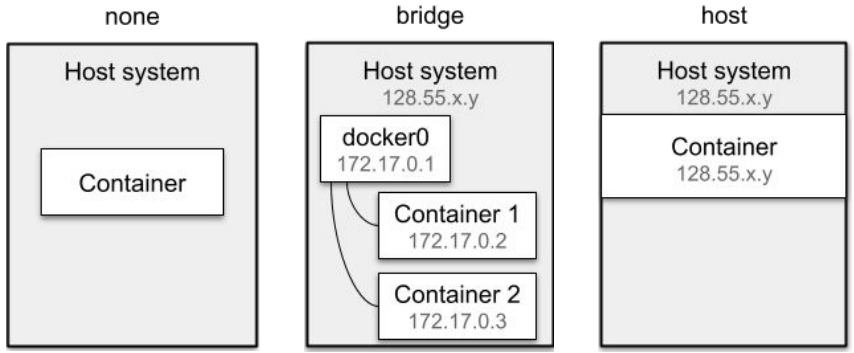
yum is configured to perform a “minimal” installation for a docker container. Files in the NERSC HSI RPM defined with `config(noreplace)` will not be installed, so rpm itself is used instead to achieve a complete installation.
https://serverfault.com/questions/998497/yum-claims-package-is-installed-but-files-not-there-in-docker

Data movement

Container networking options

- Research how containerized HSI/HTAR control communicates with HSI GWD
- Looked at host and bridge networking

	HSI parallel mode	HSI firewall mode	HTAR
bridge	incomplete	Y	N
host	Y	unknown	Y



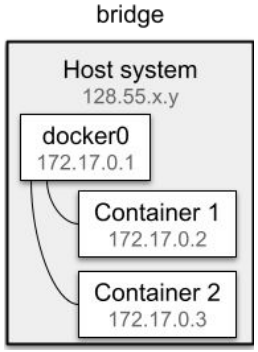
- bridge networking
  - Containers are assigned a private IP
  - Host and containers use mapped ports for communication
- host networking
  - Removes any network isolation between host and container

Data movement

HSI in parallel mode with bridge networking

- Neither HSI nor HTAR worked as-is

	HSI parallel mode	HSI firewall mode	HTAR
bridge	incomplete	Y	N
host	Y	unknown	Y



- Looked at this early on when Perlmutter networking was unknown since it had the best chance of working in most environments
  - Development on HSI was prioritized but not completed
  - No further work was done with HTAR

- The next slides cover what we learned for this networking option

Data movement

HSI in parallel mode with bridge networking: Port forwarding

- Docker can forward a range of ports on the host to an HSI container
  - HSI containers can be created from the same image
  - A container is assigned a unique range of ports on the host when it is started

```
docker run                                                  \
        -p $CONTAINER_PORT_RANGE:$HOST_PORT_RANGE \
        -e HPSS_PFTC_PORT_RANGE=$HOST_PORT_RANGE
```

  - HPSS will connect back to the client using the host port(s) that will forward to the nominal range in the container. Simple example:

Container instance	HOST_PORT_RANGE	CONTAINER_PORT_RANGE
1	7000 - 7009	7000 - 7009
2	7010 - 7019	7000 - 7009
10	7090 - 7099	7000 - 7009

Data movement

HSI in parallel mode with bridge networking: Client control hostname

- An HPSS mover needs to connect back to the host machine of the container
  - The HSI client must send the host IP in the IOD message, so the movers know how to connect back
  - Environment settings can be passed to the container to set this

```
docker run                                  \
        -e HPSS_HOSTNAME=$HOSTNAME          \
        -e HPSS_CTL_HOSTNAME=$HOSTNAME
```

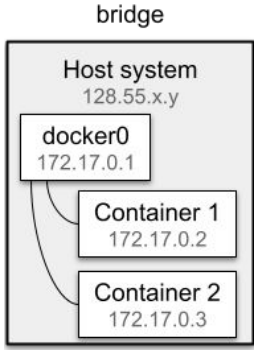
- The client contains logic that forces the host IP back to the container IP
  - The IOD message contains the container IP
  - Calls to `getsockname` or `hpss_net_getsockname` were undoing the host IP override. This logic helps secure a transaction but is not able to support this use case.
  - This work was not completed because new directions were identified

Data movement

HSI in firewall mode with bridge networking

- HSI worked as-is

	HSI parallel mode	HSI firewall mode	HTAR
bridge	incomplete	Y	N
host	Y	unknown	Y



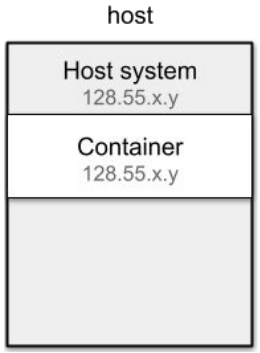
- Does not help with integration onto Perlmutter but helpful to know for other possible uses

Data movement

HSI in parallel mode with host networking

- HSI and HTAR worked as-is

	HSI parallel mode	HSI firewall mode	HTAR
bridge	incomplete	Y	N
host	Y	unknown	Y



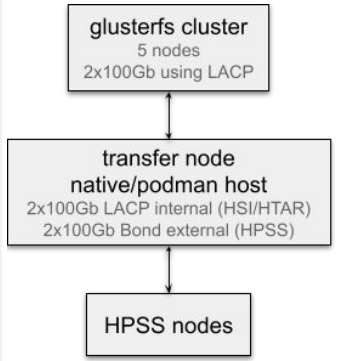
- Perlmutter will support networking similar to host: macvlan
  - A container instance will be assigned a unique IP address on the high speed network

User application with host networking

- NERSC groups using HSI/HTAR are no longer tied to RHEL 7
- With host networking enabled and no firewall, performance is close to native

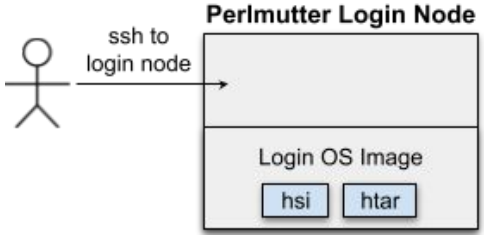
native

podman



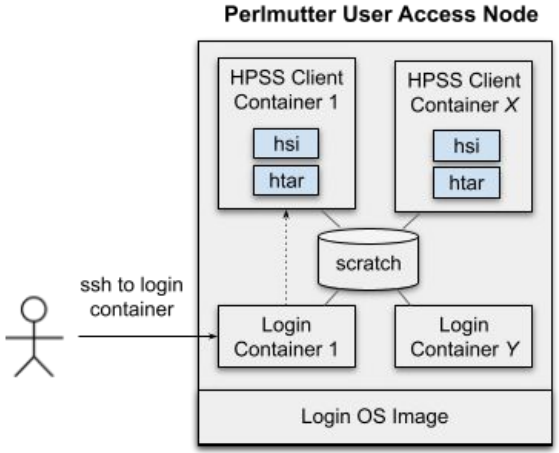
Work performed by Thomas Davis, Operations Technology Group, NERSC

HSI/HTAR client containers on Perlmutter



CURRENT
"Bare Metal" installation

- User can ssh to Login Node
- User runs HSI and HTAR



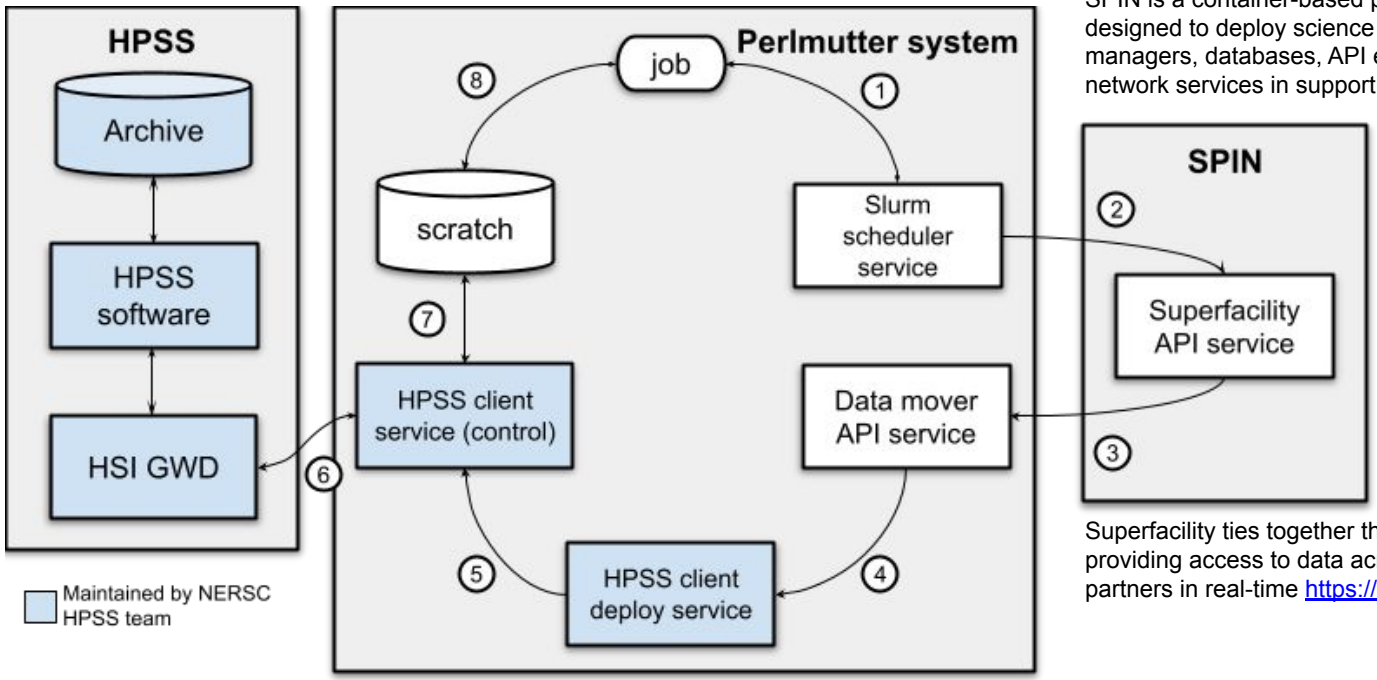
A key benefit of separating login instances from HSI/HTAR instances:

Enables access methods other than simply ssh
(e.g. automated workflow, jupyter notebook)

PLANNED FOR SPRING 2022
Containerized Clients

- User can ssh to a Login Container
- Login Container may start one or more HPSS client containers downstream depending on need

Data staging for computational workflows



SPIN is a container-based platform at NERSC designed to deploy science gateways, workflow managers, databases, API endpoints, and other network services in support of scientific projects.

Superfacility ties together the user community, providing access to data across science facility partners in real-time https://api.nersc.gov/api/v1.2/

Design input from Lisa Gerhardt, Data & Analytics Services Group, NERSC

HPSS client REST API

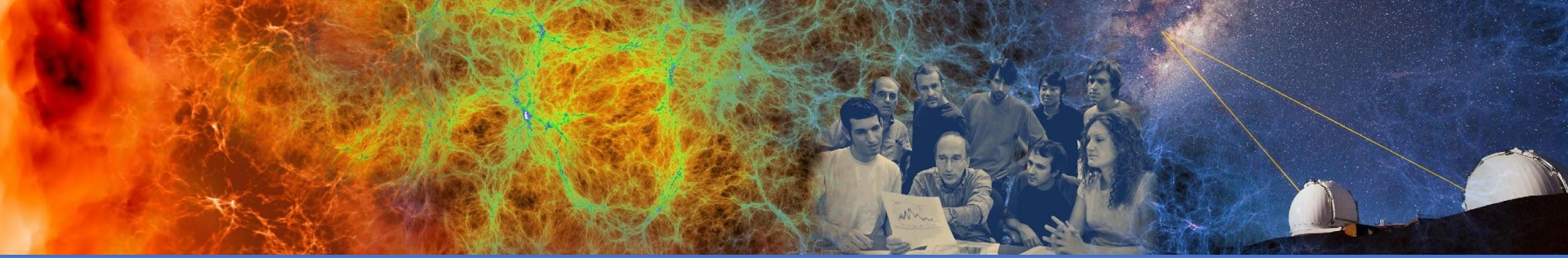
- A REST API will allow users or system to automatically stage or store data
  - Frontend equivalent to the CLI
  - Data movement will be performed by the client and not use http

- As a starting point, five operations will be handled:
  - **Get** one or more files
  - **Put** one or more files
  - **Delete** one or more files
  - **List** a directory or files
  - **Status** of the operation, e.g., complete, error, in progress, etc.

- Will look at S3 interface, HSB, and Globus REST APIs for potential reuse

Future path

- Spring 2022 timeframe
  - Ensure clients can run in Perlmutter's Kubernetes environment
  - Develop initial REST API frontend for HSI/HTAR
  - Integration and performance testing
- Longer term
  - Incorporate HPSS-aware features into workflow
    - Which media files are on, ordering/sorting, etc.
  - Evaluate other HPSS clients and REST APIs
    - S3/MinIO + HPSSFS (CR 563 in progress)
    - HSB
    - Globus
    - Quaid
    - Others
  - Key consideration: Much of NERSC data is stored using HTAR



Next: Update from IBM