CernVM
File system

Open Science Grid

UC San Diego

PACIFIC RESEARCH
PLATFORM

kubernetes

Presented by Igor Sfiligoi – UCSD

CernVM Workshop 2019

Using CVMFS on a distributed Kubernetes cluster

The PRP experience

Outline

- What is PRP?
- How does Kubernetes fit in?
- Why do we need CVMFS?
- What was done?
- Operational experience
- Wishlist

- Integration with OSG StashCache

The Pacific Research Platform

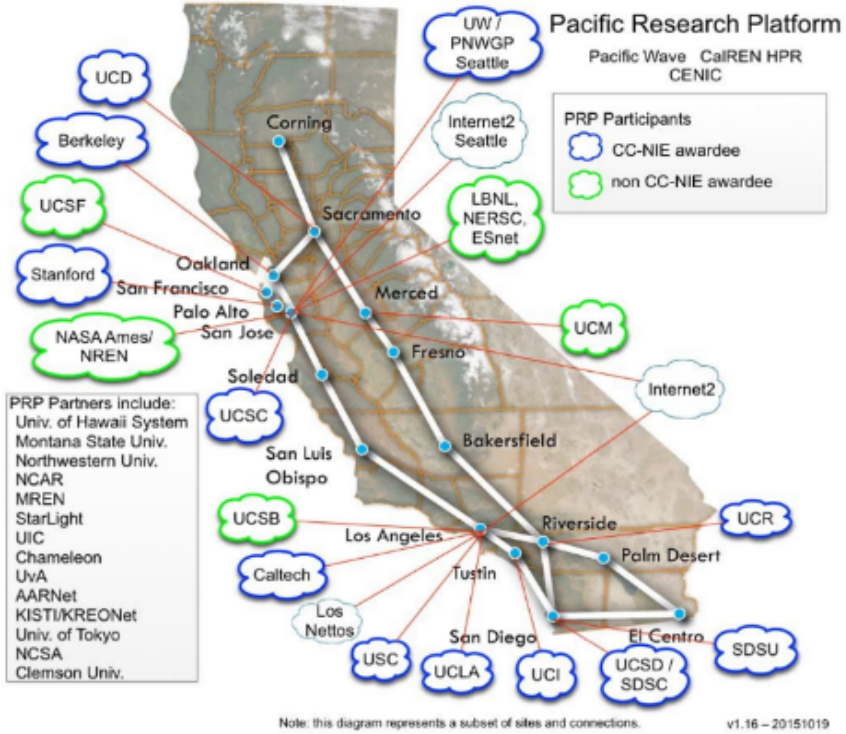
The PRP originally created as a regional networking project

- Establishing end-to-end links between 10Gbps and 100Gbps

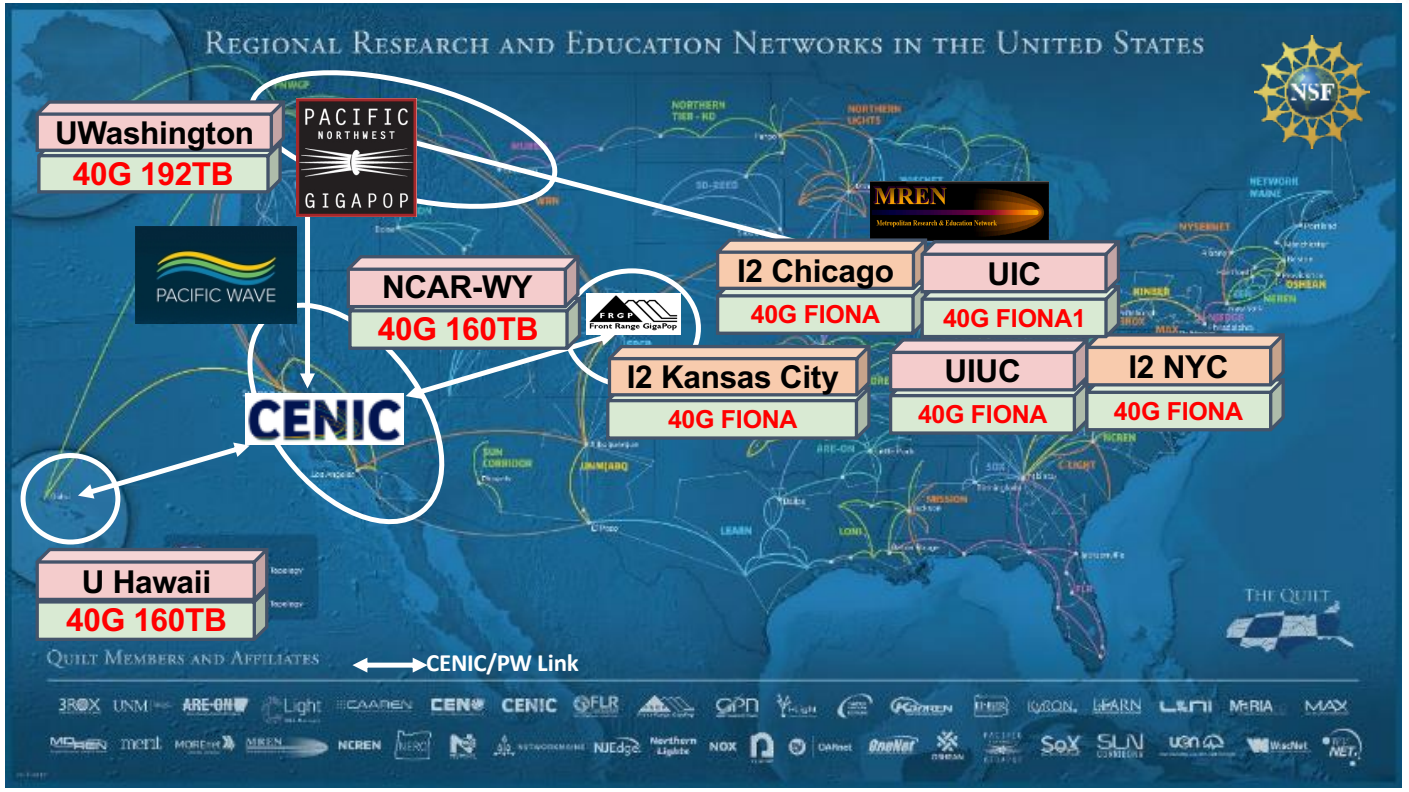
PACIFIC RESEARCH PLATFORM

http://pacificresearchplatform.org

(GDC)



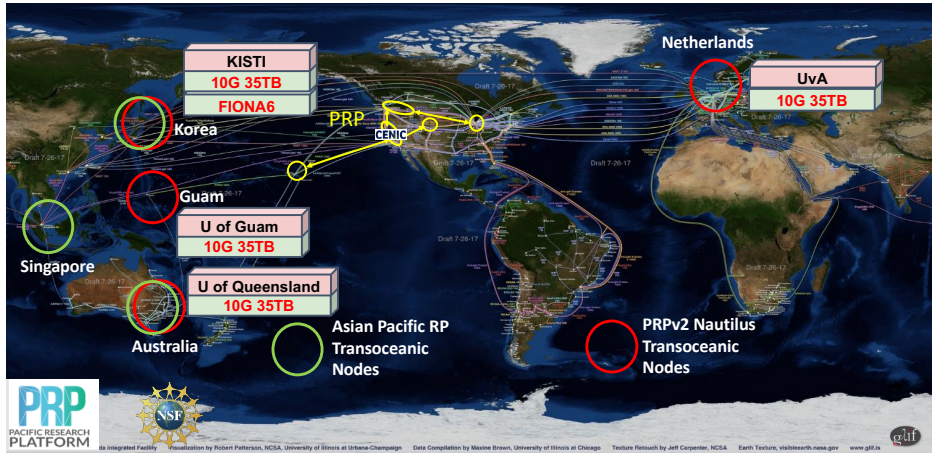
Toward the National Research Platform



ESnet
CENIC
SDSC
CITRIS BANATAO INSTITUTE
it²

PRP morphing into TNRP

- We expanded nationally
- And beyond, too



Getting into the Compute Business

Recently we evolved into a major resource provider, too

- Because scientists really need more than bandwidth tests
- They need to share their data at high speed and compute on it, too

Extensive compute power

- About 330 GPUs and 3.5k CPU cores

A large distributed storage area

- About 2 PBytes

Kubernetes as a Resource Manager

PRP decided to use Kubernetes as the Resource Manager	• Industry standard • Large and active development and support community
Containers provide major benefits	• Great for network heavy workloads • Very convenient for users
Kubernetes has flexible scheduler	• Great for mixing service and user Pods
Great plugin infrastructure	• Both for networking • and storage

Docker based

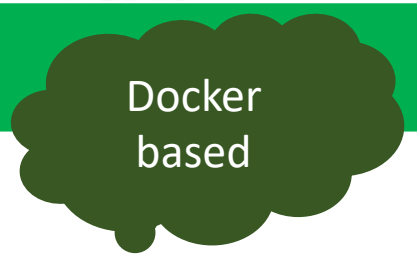
Serving OSG users

Earlier this year we wanted to add support for OSG users

And OSG has long been an active user of CVMFS

CVMFS and Unprivileged Containers

OSG Pods run only unprivileged containers



- Like all other users in the k8s cluster
- To minimize risk

CVMFS cannot be mounted from inside an unprivileged container

- Using FUSE is a privileged operation
- Unless (maybe) using properly configured latest kernel, which we cannot assume

Installing CVMFS bare-metal not an option

- We want to have only k8s at bare-metal level

Kubernetes CSI to the Rescue

Kubernetes Container Storage Interface (CSI)

- Provides a standard way to add custom filesystems

Driver deployed by cluster admin

- Privileged operation
- But admin controls and can inspect the container images

User Pods see it as an additional mount option

- No privileges needed

https://kubernetes.io/blog/2019/01/15/container-storage-interface-ga/

API Version Blues

CernVM File system

CSI

- The CVMFS team developed a Kubernetes CSI driver https://github.com/cernops/cvmfs-csi
- But they developed against the beta version
  - Which has since been deprecated!

Kubernetes	CSI Spec Compatibility	Status
v1.9	v0.1.0	Alpha
v1.10	v0.2.0	Beta
v1.11	v0.3.0	Beta
v1.13	v0.3.0, v1.0.0	GA

https://kubernetes-csi.github.io/docs/introduction.html

UCSD Team Does the Refactoring

The changes in the API were not huge

- But big enough to make the CERN-provided version unusable out-of-the-box

Dima Mishin from our UCSD team fixed it

- Now fully 1.0 compliant
- Also did minor polishing so it can co-exist with other CSI plugins

Have been running it since February 2019 on our cluster

Contributed back as a Pull Request

- Still not merged, though

https://github.com/cernops/cvmfs-csi/pull/1

CERN vs OSG Setup

CERN provided version was optimized for CERN use

- Assuming only CERN-based repositories would ever be used
- Would not allow for mounting of OSG repositories

The restriction is not really necessary

- Removed it from the code
- Contributed back as a PR (also still pending) https://github.com/cernops/cvmfs-csi/pull/2

Also switched to OSG packaged CVMFS

- Gives me all the needed config out-of-the-box
  https://github.com/sfiligoi/cvmfs-csi/blob/prp-osg/deploy/docker/Dockerfile
  https://cloud.docker.com/u/sfiligoi/repository/docker/sfiligoi/csi-cvmfsplugin

A little more details, now

Deploying CVMFS

CVMFS Driver Pods deployed as a DaemonSet

- One (privileged) Pod starts on each node
- Plus a couple Service Pods
  https://github.com/sfiligoi/prp-osg-cvmfs/tree/master/k8s/cvmfs/csi-processes

CVMFS config needs a Squid

- We deploy one as a Kubernetes Service
- Using the OSG maintained Frontier Squid Image
  https://github.com/sfiligoi/prp-osg-cvmfs/tree/master/k8s/frontier

```
$ kubectl get services -n cvmfs
NAME                          TYPE        CLUSTER-IP      PORT(S)
csi-cvmfsplugin-attacher      ClusterIP   10.100.161.182  12345/TCP
csi-cvmfsplugin-provisioner   ClusterIP   10.103.119.130  12345/TCP
frontier-squid                ClusterIP   10.97.246.52     3128/TCP
```

```
$ kubectl get pods -n cvmfs
NAME                                READY   STATUS
csi-cvmfsplugin-2mlqw               2/2     Running
csi-cvmfsplugin-2zx76               2/2     Running
…
csi-cvmfsplugin-8qfdf               2/2     Running
csi-cvmfsplugin-provisioner-0       1/1     Running
frontier-squid-77bb5546bd-swwdh     1/1     Running
```

Defining Mountpoints

AutoFS not an option

- Seems a limit of the Linux kernel
- We have to explicitly list all the supported mountpoints

Using one StorageClass x mountpoint

- The desired mountpoint is specified as a parameter

https://github.com/sfiligoi/prp-osg-cvmfs/tree/master/k8s/cvmfs/storageclasses

```
$ cat storageclass-oasis.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-cvmfs-oasis
provisioner: csi-cvmfsplugin
parameters:
  repository: oasis.opensciencegrid.org
```

```
$ cat storageclass-stash.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-cvmfs-stash
provisioner: csi-cvmfsplugin
parameters:
  repository: stash.osgstorage.org
```

Using CVMFS

PersistentVolumeClaims can now be used by the users

- Note: Need to be defined in each and all namespaces that use them
  https://github.com/sfiligoi/prp-osg-cvmfs/tree/master/k8s/cvmfs/pvcs

No privileges needed by users to use them in Pods

- Just regular “external volumes”
- One per CVMFS mountpoint

```
$ cat pvc-oasis.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: csi-cvmfs-pvc-oasis
  namspace: osggpus
spec:
  accessModes:
  - ReadOnlyMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-cvmfs-oasis
```

Using CVMFS

PersistentVolumeClaims car

- Note: Need to be defined in each
  https://github.com/sfiligoi/prp-osg-cvmfs/t

No privileges needed by use

- Just regular “external volumes”
- One per CVMFS mountpoint

```
kind: Deployment
metadata:
  name: osg-wn-gpu
  namespace: osggpus
spec:
  template:
    spec:
      containers:
      - name: osg-wn-gpu
        …
        volumeMounts:
        - name: cvmfs-connect
          mountPath: /cvmfs/connect.opensciencegrid.org
          readOnly: true
        - name: cvmfs-stash
          mountPath: /cvmfs/stash.osgstorage.org
          readOnly: true
      volumes:
      - name: cvmfs-oasis
        persistentVolumeClaim:
          claimName: csi-cvmfs-pvc-oasis
          readOnly: true
      - name: cvmfs-stash
        persistentVolumeClaim:
          claimName: csi-cvmfs-pvc-stash
          readOnly: true
```

Operational experience

Mostly a smooth ride

No major problems so far

- CVMFS CSI just works

Driver Pod restarts can be annoying

- The user pods using CVMFS on that node will hang
- Makes CVMFS maintenance non-trivial

Current CSI Driver leaving Zombie processes behind

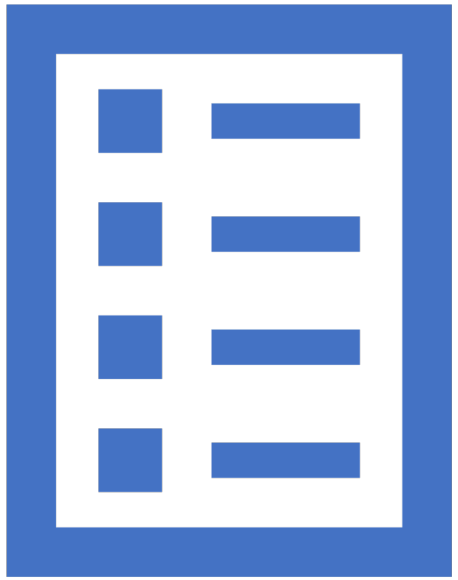
- Not critical, but still annoying

```
# ps -ef |awk '{print $1 " " $8 " " $9}' | \
          sort |uniq -c
     73 cvmfs [cvmfs2] <defunct>
     75 cvmfs [sh] <defunct>
      1 root awk {print
      1 root /bin/bash
      1 root /csi-cvmfsplugin --nodeid=…
      1 root ps -ef
    195 root [sh] <defunct>
      1 root sort
      1 root uniq -c
      1 UID CMD
```

Outstanding problems

Cannot mount the came CVMFS mountpoint from two namespaces

- CVMFS CSI Driver fails on an internal check
- Recently discovered, did not have time to properly investigate or fix it



Wishlist

Can we get our PRs accepted?

- Would rather use a CERN-maintained version
- Please let us know if there is anything else we can do to help there

AutoFS capabilities

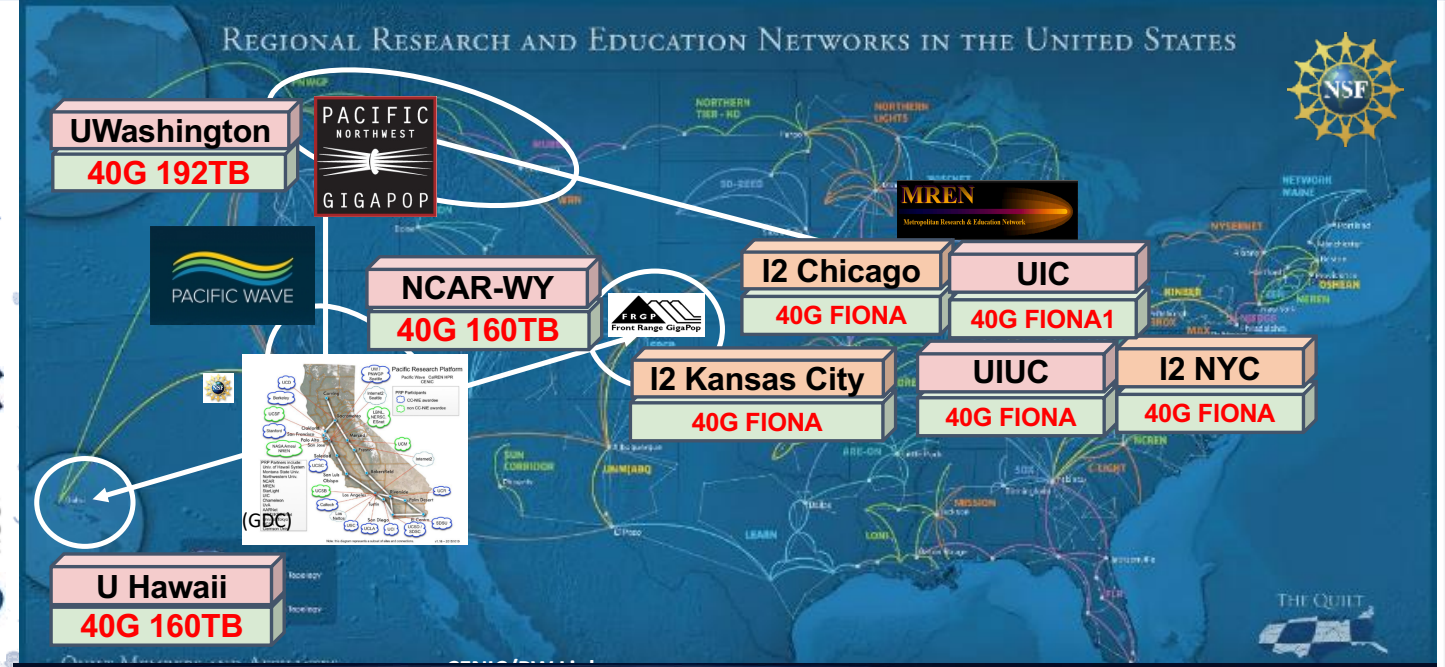
- If at all possible
- Any other ideas to avoid explicit listing of all possible repositories?

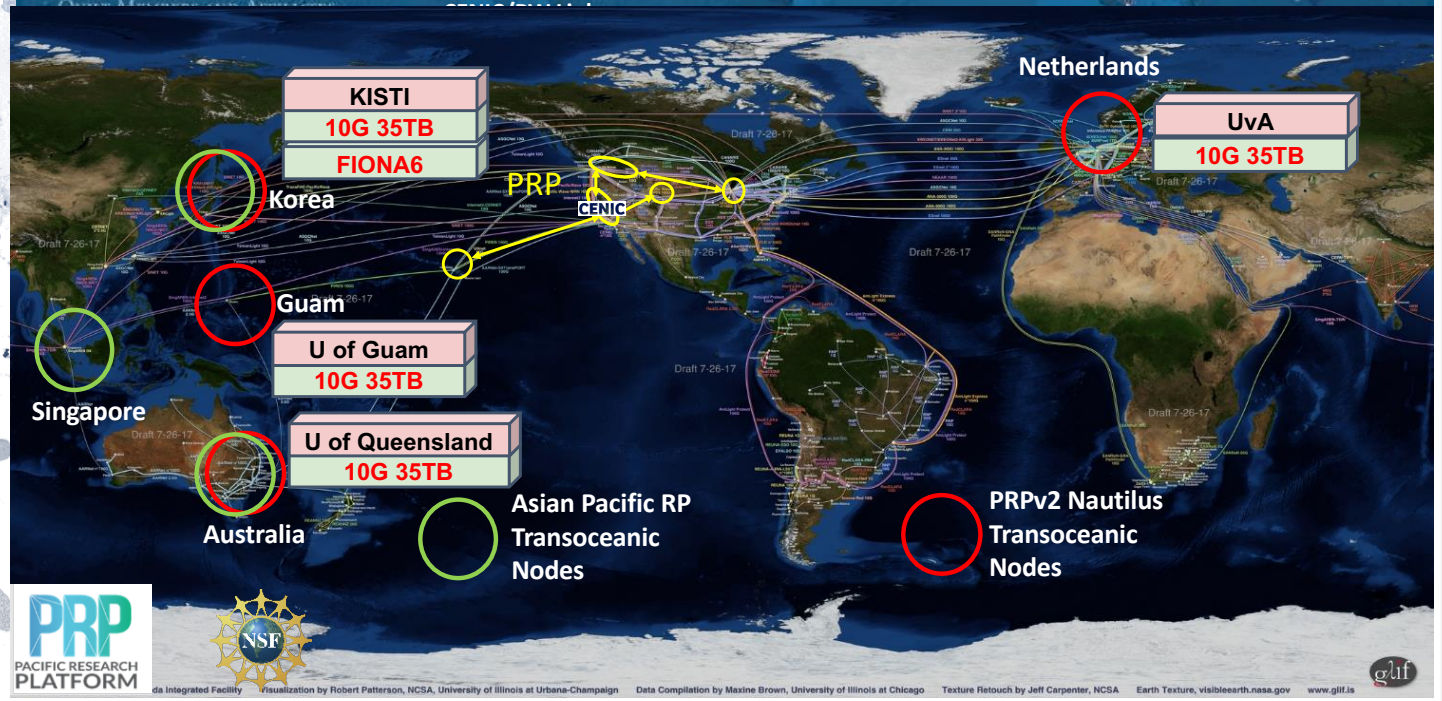
CVMFS and OSG StashCache

PACIFIC RESEARCH PLATFORM

ESnet
CENIC
SDSC
CITRIS BANATAO INSTITUTE
it²

PRP/TNRP a Distributed Setup

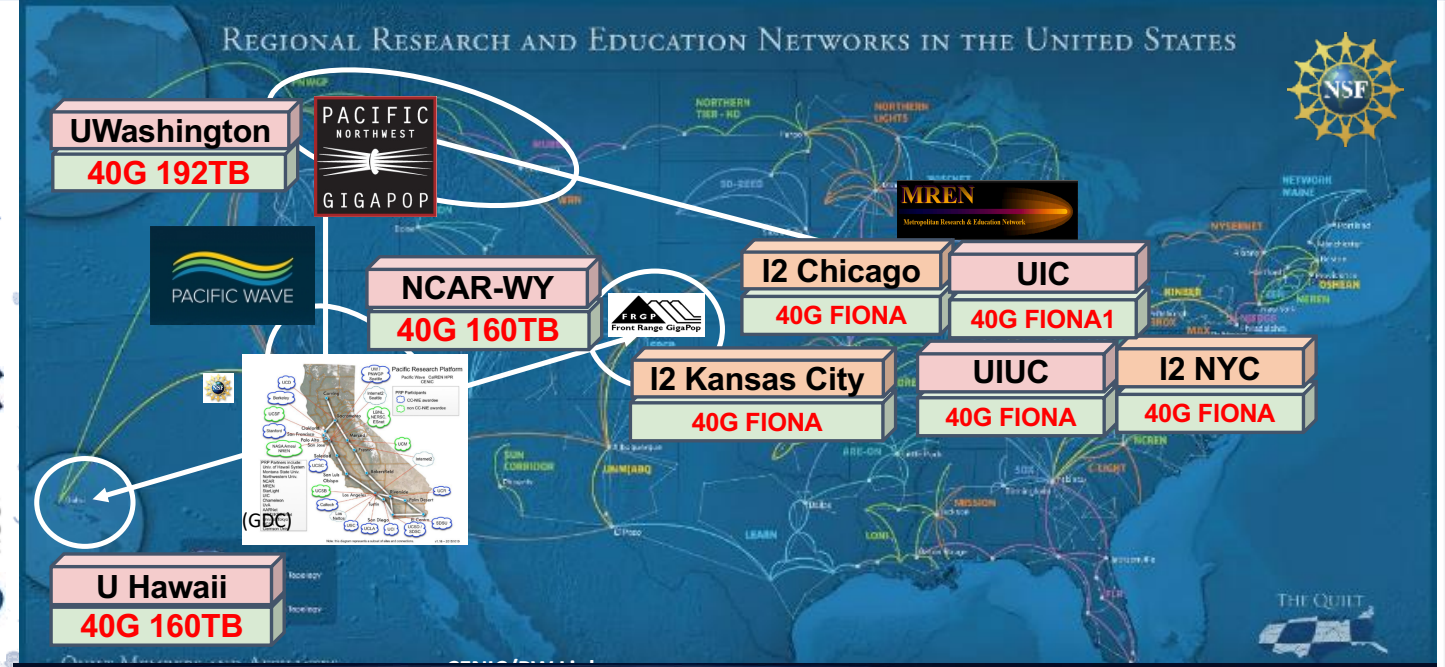


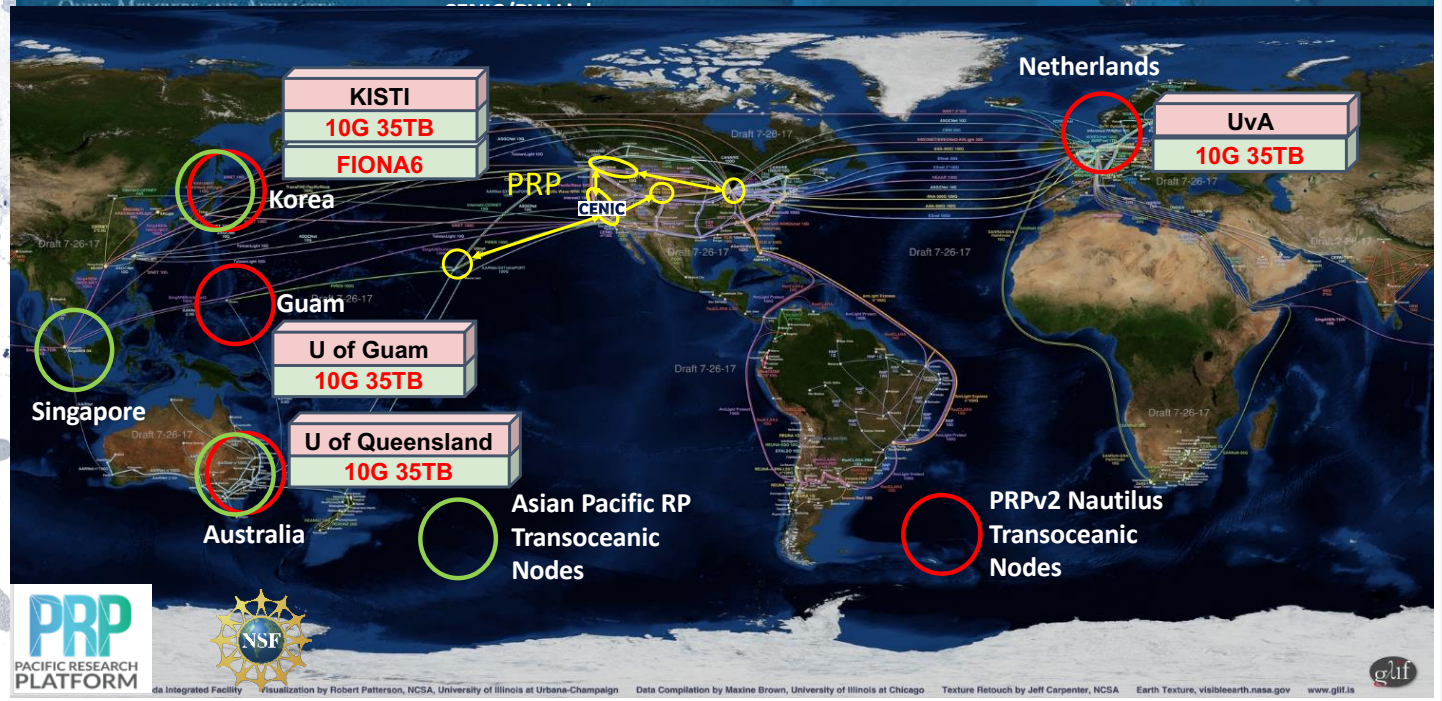


PACIFIC RESEARCH PLATFORM

ESnet
CENIC
SDSC
CITRIS BANATAO INSTITUTE
it²

Single Squid makes no sense





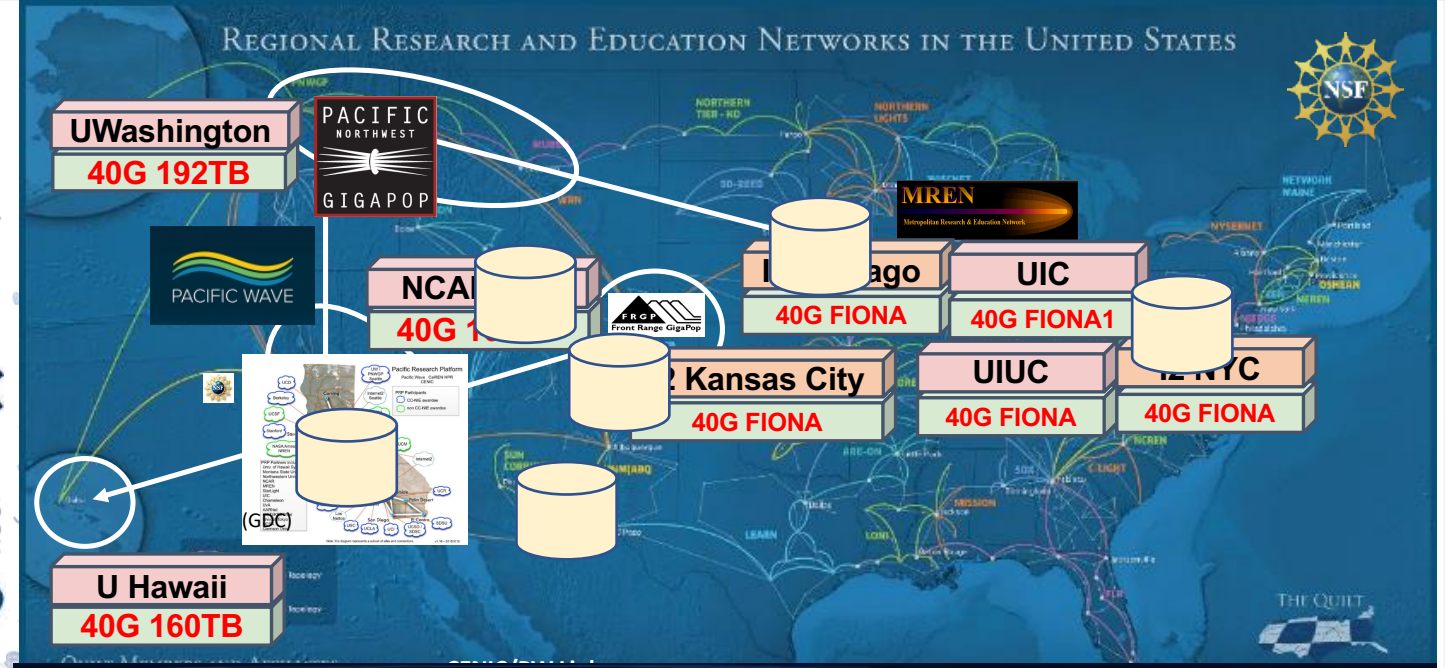
PRP PACIFIC RESEARCH PLATFORM

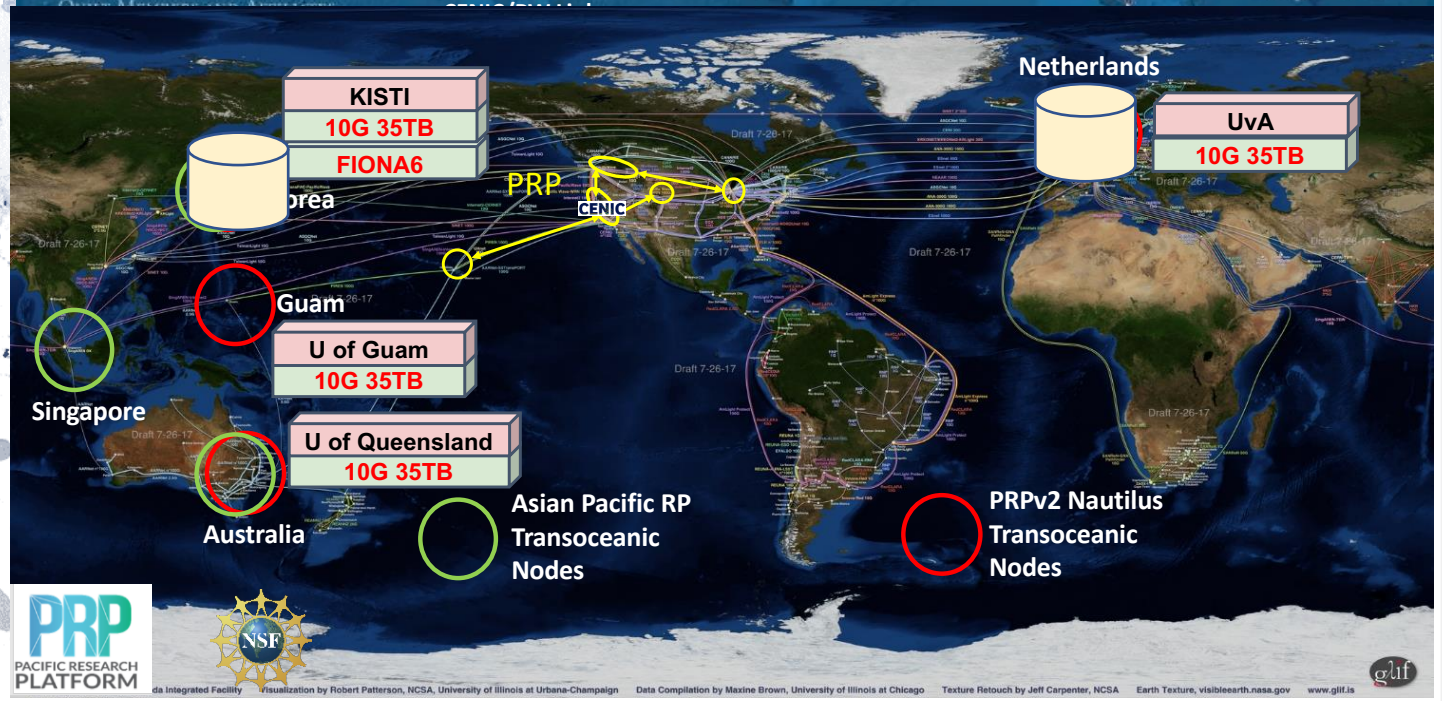
Open Science Grid

ESnet CENIC SDSC CITRIS BANATAO INSTITUTE it²

OSG Operates a set of StashCache nodes

Using the same PRP Kubernetes cluster





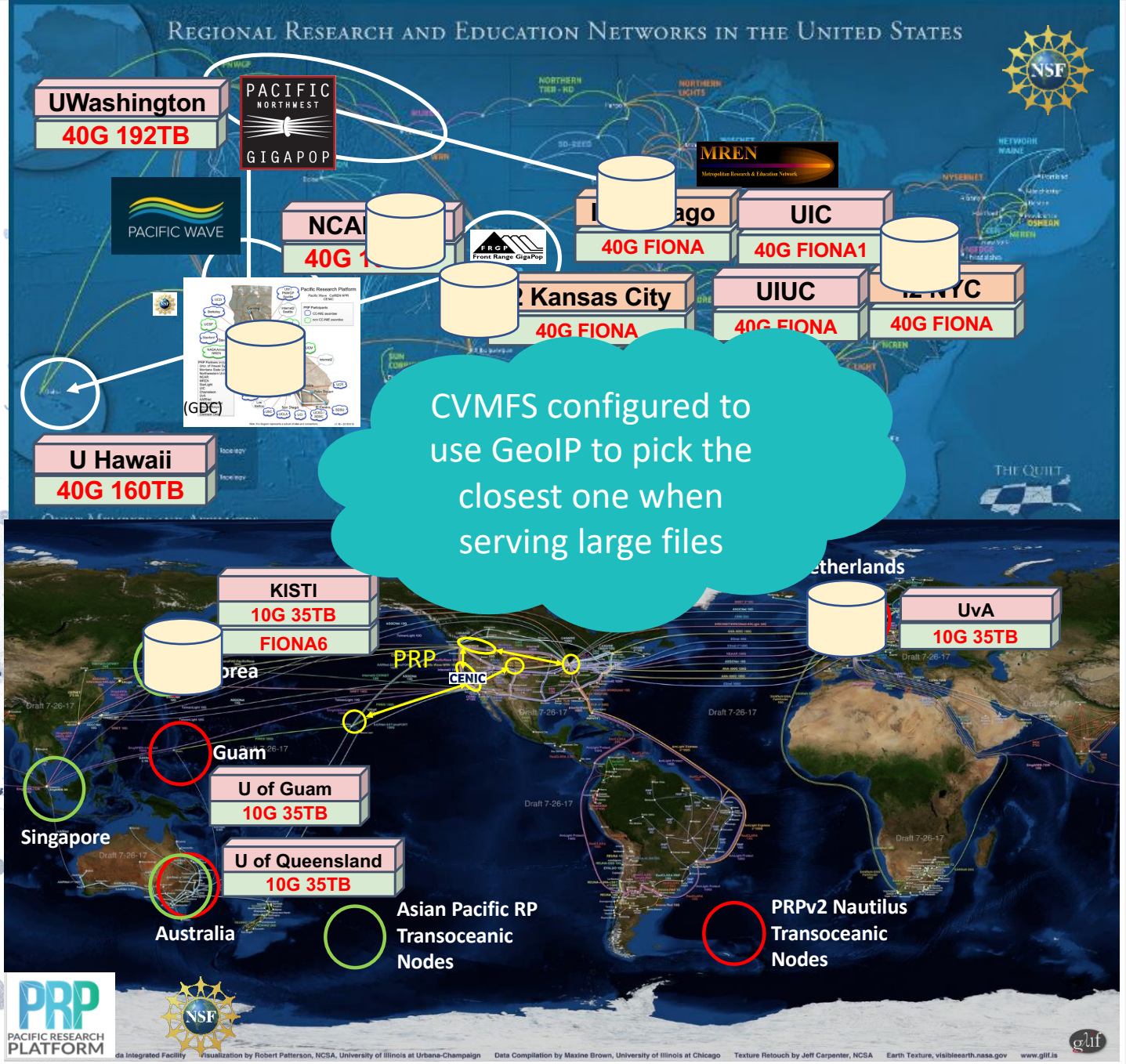
PACIFIC RESEARCH PLATFORM

Open Science Grid

ESnet CENIC SDSC CITRIS BANATAO INSTITUTE it²

OSG Operates a set of StashCache nodes

Essentially XRootD servers



Summary

- PRP is using Kubernetes as a resource manager
- OSG users need CVMFS
- CVMFS cannot be mounted from unprivileged containers
- Using Kubernetes CSI to mount CVMFS in PRP
- Had to fix CERN-provided version
- No major operational issues found
- Using Squid for small files and OSG StashCache for large files

Acknowledgents

This work was partially funded by
US National Science Foundation (NSF) awards
CNS-1456638, CNS-1730158,
ACI-1540112, ACI-1541349,
OAC-1826967, OAC 1450871,
OAC-1659169 and OAC-1841530.