rt.iu.edu

RESEARCH TECHNOLOGIES
UNIVERSITY INFORMATION TECHNOLOGY SERVICES

RESEARCH TECHNOLOGIES
UNIVERSITY INFORMATION TECHNOLOGY SERVICES

Jetstream Overview: A national research and education cloud

ECSS Webinar
Dec 1, 2020 – Bloomington, IN.

Jeremy Fischer – Jeremy@iu.edu - Indiana University

Manager, Jetstream Cloud, UITS Research Technologies

Fischer, J. (2020). Jetstream Overview: A national research and education cloud. Bloomington, IN. Retrieved from https://jetstream-cloud.org/research/publications.php

Code of Conduct

XSEDE has an external code of conduct which represents our commitment to providing an inclusive and harassment-free environment in all interactions regardless of race, age, ethnicity, national origin, language, gender, gender identity, sexual orientation, disability, physical appearance, political views, military service, health status, or religion. The code of conduct extends to all XSEDE-sponsored events, services, and interactions.

Code of Conduct: https://www.xsede.org/codeofconduct

Contact:

- Event organizer: **Jay Alameda (alameda@illinois.edu)**
- XSEDE ombudspersons:
  - Linda Akli, Southeastern Universities Research Association (akli@sura.org)
  - Lizanne Destefano, Georgia Tech (lizanne.destefano@ceismc.gatech.edu)
  - Ken Hackworth, Pittsburgh Supercomputing Center (hackworth@psc.edu)
  - Bryan Snead, Texas Advanced Computing Center (jbsnead@tacc.utexas.edu)
- Anonymous reporting form available at https://www.xsede.org/codeofconduct.

Terminology Statement

In line with XSEDE's Code of Conduct, XSEDE is committed to providing training events that foster inclusion and show respect for all. This commitment applies not only to how we interact during the event; it also applies to the training materials and presentation. It is not XSEDE's position to use, condone, or promote offensive terminology.

XSEDE instructors strive to keep inclusive language at the forefront. In the event that we have included inappropriate materials, verbal or written, please let us know at terminology@xsede.org

While XSEDE has no control over external third-party documentation, we are taking steps to effect change by contacting the relevant organizations; we hope this will be addressed by all third parties soon.

If you see any terminology concerns in the following presentation or slides, we want to know!
Please contact the Terminology Task Force: terminology@xsede.org

What is Jetstream and why does it exist?

- *NSF's first production cloud facility*
- *Focus on ease-of-use, broad accessibility*
- *User-selectable library of preconfigured virtual machines*
- *Provides on-demand interactive computing and analysis or persistent services such as gateways*
- *Enables configurable environments; programmable cyberinfrastructure*

Who uses Jetstream?

- *The researcher needing a handful of cores (1 to 44/vCPU)*
- *Software creators and researchers needing to create their own VMs and workflows*
- *Science gateway creators using Jetstream as either the frontend or processor for scientific jobs*
- *STEM Educators teaching on a variety of subjects*

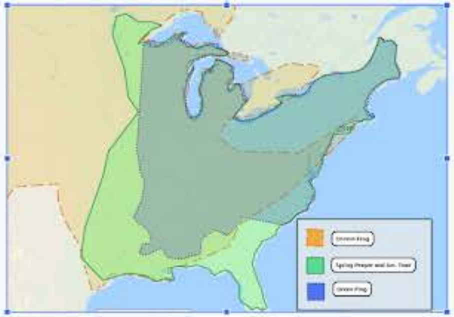
M87 black hole: how cloud computing supports astronomy

Event Horizon Telescope

M87 black hole image generated by EHT

- *Event Horizon Telescope (a telescope array consisting of a global network of radio telescopes), a large number of scientists, NASA spacecraft, and a variety of computing resources enabled the first image of a black hole.*
- *For the M87 back hole image, two critical steps were done in the cloud and piloted on Jetstream*
  - *correcting for anomalies, so that further image processing could occur, and*
  - *large survey study of how image reconstruction algorithms affect the final images.*
- *The team is also developing new methods to correlate data from multiple telescopes (to reduce data from petascale to terascale) in the cloud*

AI for Everyone – Recognizing Frog Calls

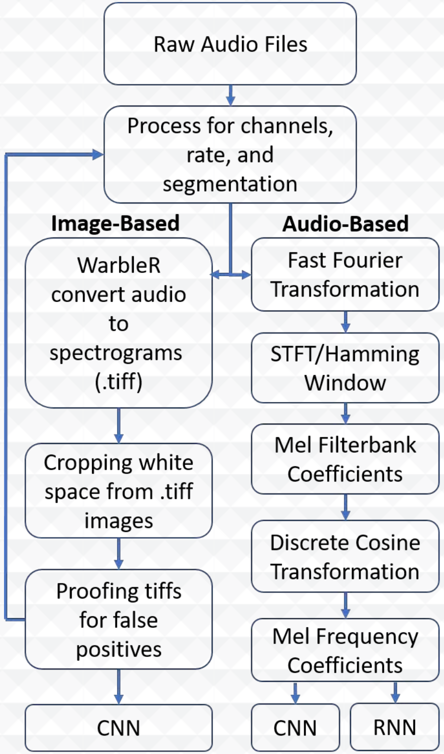


American Toad

Spring Peeper

Chorus Frog

Green Frog



Raw Audio Files

Process for channels, rate, and segmentation

Image-Based

WarbleR convert audio to spectrograms (.tiff)

Cropping white space from .tiff images

Proofing tiffs for false positives

CNN

Audio-Based

Fast Fourier Transformation

STFT/Hamming Window

Mel Filterbank Coefficients

Discrete Cosine Transformation

Mel Frequency Coefficients

CNN

RNN

2019 Jetstream REU participants examined best-practices for supporting AI projects for field biologists

RESEARCH TECHNOLOGIES
UNIVERSITY INFORMATION TECHNOLOGY SERVICES

Brainlife.io

Brainlife.io is a science gateway for neuroscience analysis.

Allows creation of custom workflows that can be saved and shared

Began using only Jetstream and other XSEDE resources and has grown to use

Expanded to use Microsoft's Azure cloud via the Midwest Big Data Hub

Backed with multiple virtual Slurm clusters on IU and TACC clouds

Jetstream for education – in action at AMS2020

- Unidata-led workshop at American Meteorological Society (AMS) 2020 conference
- 127 users actively participating
- Participants used a JupyterHub running on Jetstream (40 node Kubernetes cluster of 6 core m1.medium VMs) for a 90 minute Unidata PyAOS (Python for the Atmospheric and Oceanic Sciences) workshop
- The students were successfully able to run their interactive Python code notebooks as the instructors presented their material

What Jetstream isn’t…

- *It’s not traditional HPC*
- *There’s no shared filesystem (think cloudy!)*
- *There’s no high-end interconnect fabric (keep thinking cloudy!)*
- *There aren’t GPUs widely available*
- *It isn’t Amazon, Azure, or GCE (similar, but…)*

HPC vs Cloud

Adapting to a different environment:

- No reservations, no queueing – more interactive usage
- Being your own admin – hey, we have root!**
- You really can have almost any (linux) software you want**

** Here there be dragons…

Jetstream and way of the cloud…

- **Cloudy Technologies**: clouds are more than just virtual machines (VM)
  - **Old way**: robust (expensive) infrastructure, weak (cheap) software
    - You expect the hardware to not fail
    - State in maintained in volatile data structures
  - **Cloudy way**: commodity infrastructure, robust software
    - Expect & plan for infrastructure to fail
    - Put intelligence into the software to handle infrastructure failure
  - **And my favorite…**

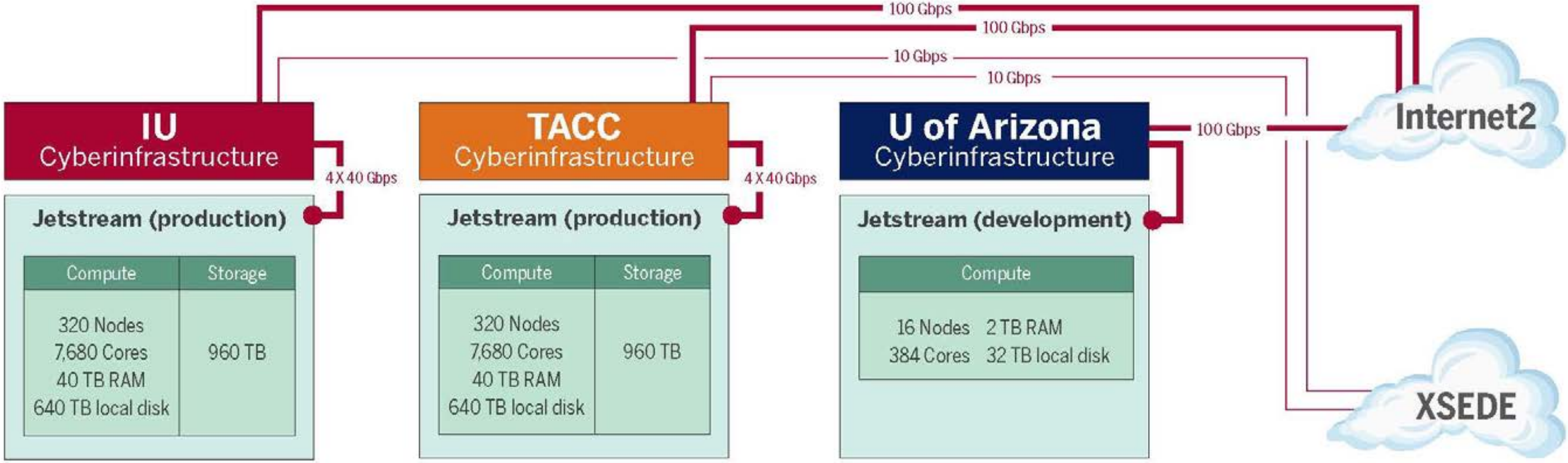
Thinking about VMs…

Cows, not pets: pets take great amount of care, feeding, and you name them; cows you intend to have high turnover and you give them numbers.

-- Mike Lowe (Jetstream architect)

***some caveats for gateways…*

Jetstream1 System Overview



IU
Cyberinfrastructure

Jetstream (production)

Compute	Storage
320 Nodes 7,680 Cores 40 TB RAM 640 TB local disk	960 TB

TACC
Cyberinfrastructure

Jetstream (production)

Compute	Storage
320 Nodes 7,680 Cores 40 TB RAM 640 TB local disk	960 TB

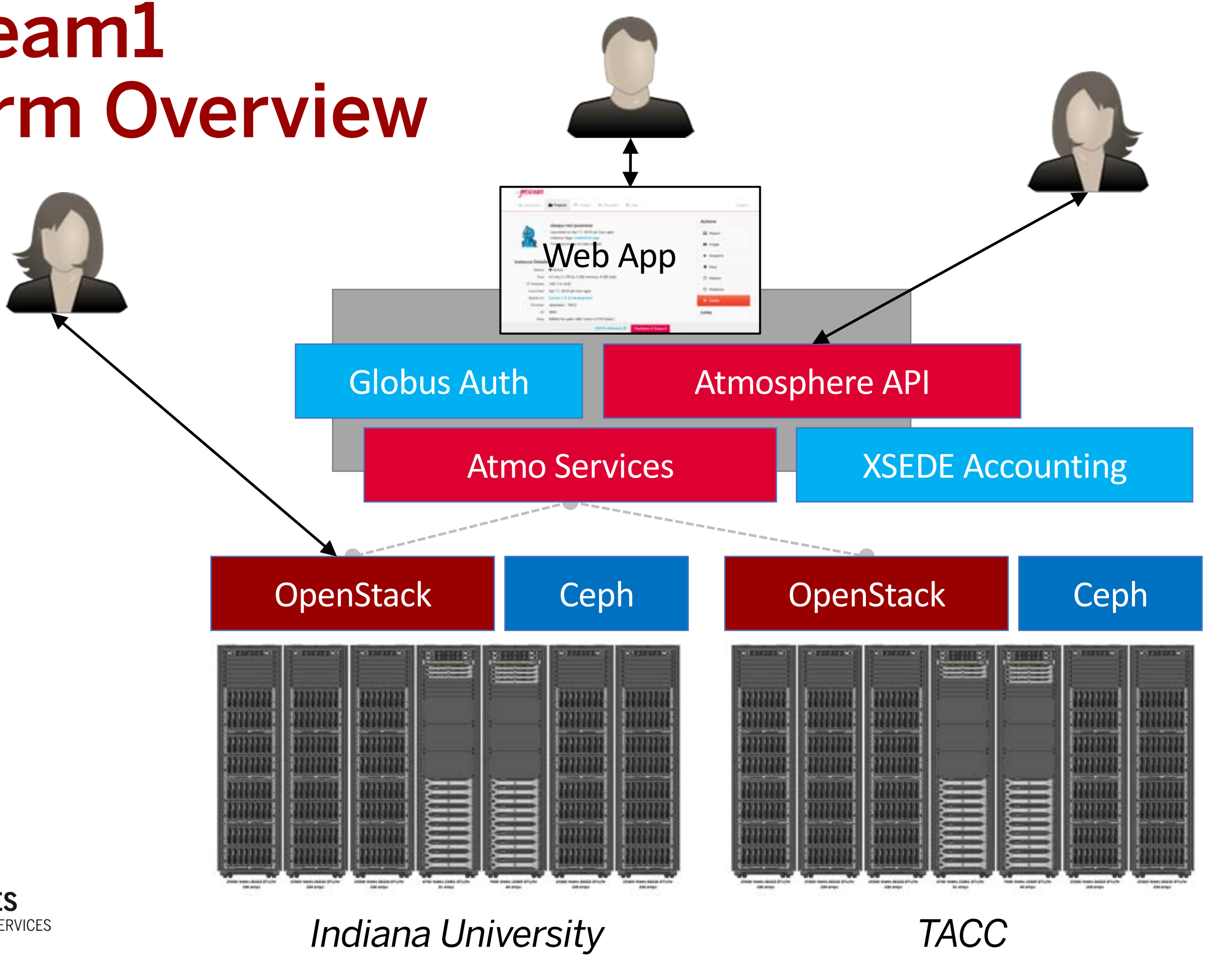
U of Arizona
Cyberinfrastructure

Jetstream (development)

Compute	
16 Nodes 384 Cores	2 TB RAM 32 TB local disk

http://wiki.jetstream-cloud.org/Network+configuration+and+policies

Jetstream1
Platform Overview



What worked?

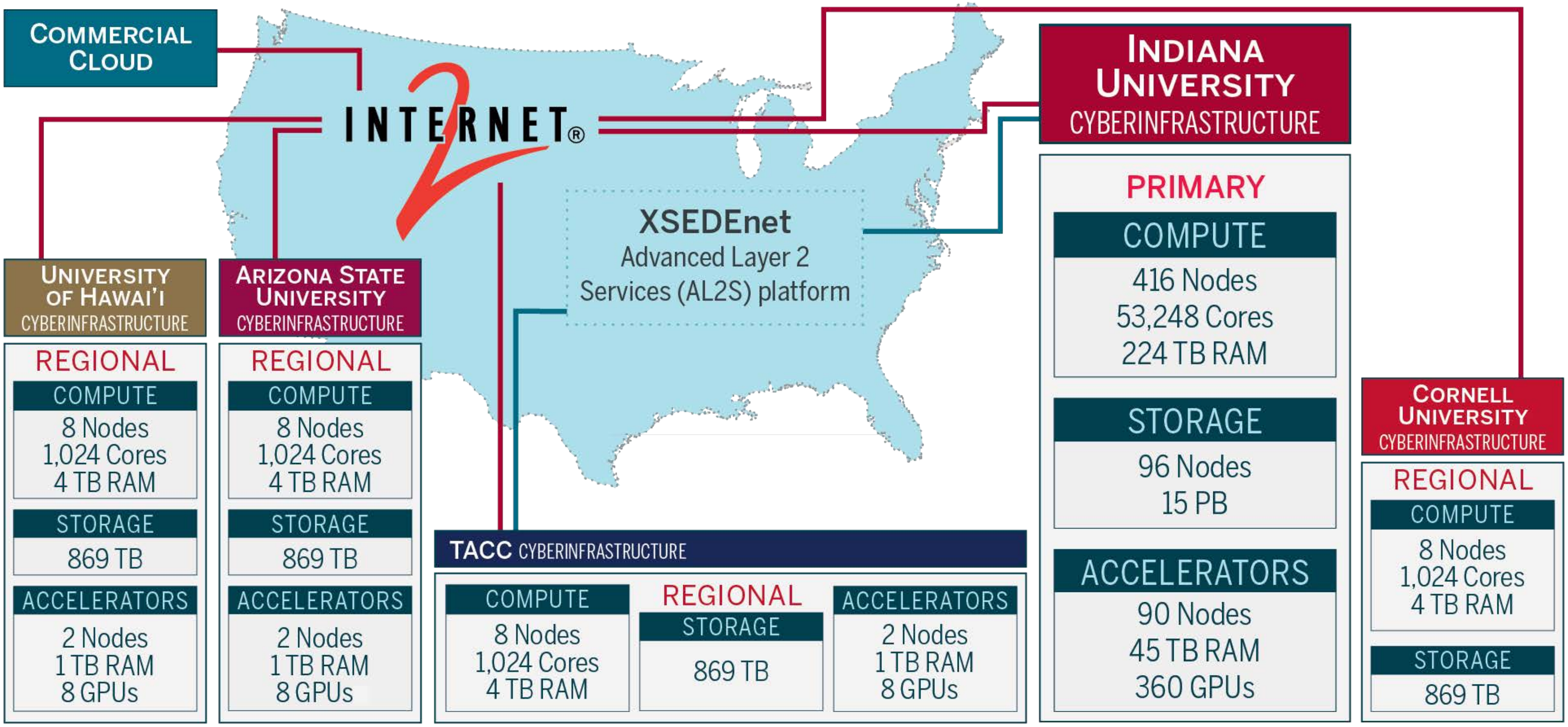
- Allowing API access and full control (root privileges)
- Allowing allocations to run continuously – as long as the PI renewed – allowing workflows to run indefinitely
- Development of trial allocations

What didn’t work?

- Forcing small allocations into the research allocation process
- Lack of multi-year allocations
- Lack of shared data set storage

Jetstream usage highlights – 1 Nov 2020

- *410 XSEDE projects covering 75 fields of science and almost 2400* ***active users*** *representing* ***407 institutions***
- *80% of Jetstream users have* ***not used any other XSEDE system***
- *>362M CPU hours allocated to XSEDE projects since June 2016*
- *48 active science gateways*
- *49 education/teaching allocations serving almost 900 students*
- *1189 mean active VMs in previous qtr, 1632 peak active VM count*
- *Highest user satisfaction in most recent XSEDE survey*



COMMERCIAL CLOUD

INTERNET2®

XSEDEnet
Advanced Layer 2
Services (AL2S) platform

UNIVERSITY OF HAWAI'I CYBERINFRASTRUCTURE

REGIONAL

COMPUTE
8 Nodes 1,024 Cores 4 TB RAM

STORAGE
869 TB

ACCELERATORS
2 Nodes 1 TB RAM 8 GPUs

ARIZONA STATE UNIVERSITY CYBERINFRASTRUCTURE

REGIONAL

COMPUTE
8 Nodes 1,024 Cores 4 TB RAM

STORAGE
869 TB

ACCELERATORS
2 Nodes 1 TB RAM 8 GPUs

TACC CYBERINFRASTRUCTURE

REGIONAL

COMPUTE	STORAGE	ACCELERATORS
8 Nodes 1,024 Cores 4 TB RAM	869 TB	2 Nodes 1 TB RAM 8 GPUs

INDIANA UNIVERSITY CYBERINFRASTRUCTURE

PRIMARY

COMPUTE
416 Nodes 53,248 Cores 224 TB RAM

STORAGE
96 Nodes 15 PB

ACCELERATORS
90 Nodes 45 TB RAM 360 GPUs

CORNELL UNIVERSITY CYBERINFRASTRUCTURE

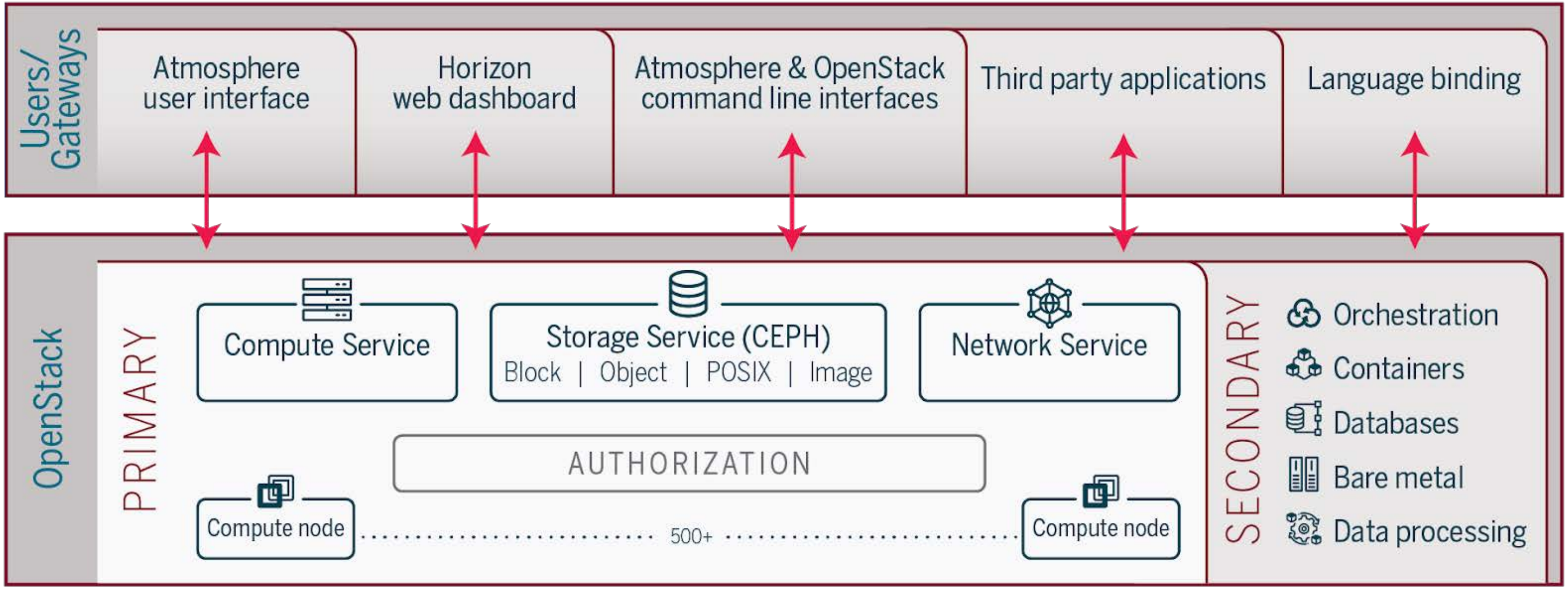
REGIONAL

COMPUTE
8 Nodes 1,024 Cores 4 TB RAM

STORAGE
869 TB

Jetstream2 Proposed Architecture

Conceptual Jetstream2 Architecture



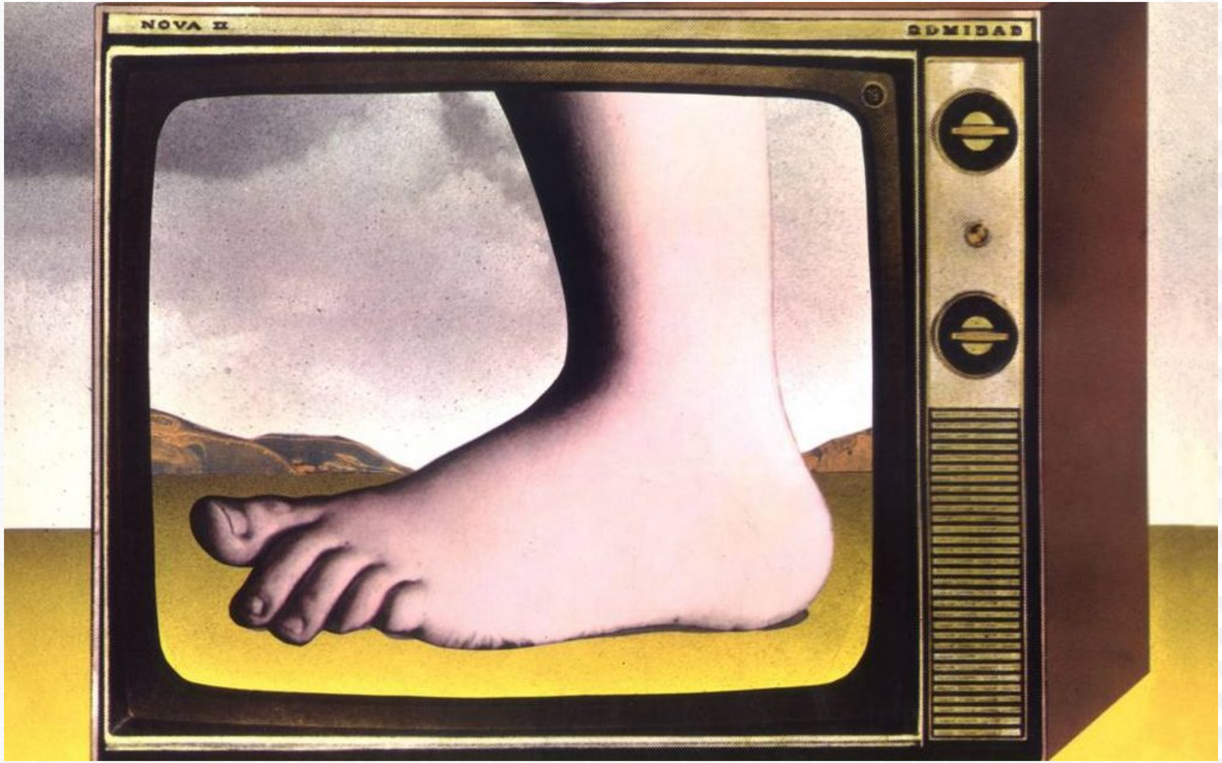
What improvements are planned?

- Improving access to higher level orchestration
- *Improving documentation and training for orchestration*
- *Implementing "push button" virtual clusters*
- *Federating JupyterHubs and making the implementation of JupyterHubs a simple process*
- *Creating a shared application service for VMs to make common scientific software more accessible*
- *Improved storage access, including object storage and storage that is sharable between VMs in the same allocation*

Future Plans with Jetstream2

- Focusing on ***programmable cyberinfrastructure*** using technologies like Terraform to make creating infrastructure easy on Jetstream2, commercial clouds, or other private clouds
- Making enhanced container support for interoperability a priority
- Planned collaborations with commercial clouds:
- AWS to provide workshops on cloud interoperability
- Bursting to Azure via on-premises data gateway
- Implementation of Google's Cloud Service Platform (allowing management of hybrid cloud environments via gcloud CLI or Google GUI.
- Interactive GPU access and the ability to have long-running training for AI workloads

…and now for something completely different…



Cloud Computing Terms…simplified

Image: a file on a disk. It will be booted to create an…

Instance: a running virtual server; i.e. something you can log into.

State: something worth remembering; i.e. the state of the system

Cloud Computing Terms…simplified

Flavor: the size of a running instance; i.e. #core, RAM, disk

Hypervisor: the thing the instance runs on; something akin to a software defined hardware compute server.

Snapshot: the process of taking an instance and turning it to an image.

Cloud Computing Terms…simplified

States:

- **Running:** the instance is up & running
- **Suspended:** the instance is memory resident on the hypervisor but not running
- **Stopped:** the instance is shutdown akin to powering down
- **Shelved:** the instance is shutdown, written to disk, and stored

---

XSEDE/Jetstream specific information:

http://wiki.jetstream-cloud.org/XSEDE+Service+Units+and+Jetstream

Cloud Computing Terms…simplified (Cont.)

Object store: a blob of bits; it has a starting address & a size. There may be metadata associated with the object. The data is consumed in a streaming manner.

Block store: a software defined entity akin to an unformatted hardware disk drive, data is stored in blocks and has no metadata associated at the lowest levels

Filesystem: hierarchical in nature, directories & files, ability to open, seek, read, write.

Persistent storage: If you pull the plug, it will still exist when power is restored. Safe to store data or state here.

Ephemeral storage: If you pull the plug, it no longer exists. (Don’t leave your data here!!!)

What is Jetstream – a closer look

- **Software layers**
  - **Atmosphere** web interface
    - library of images, generic, domain specific
    - simplify VM administration
  - **OpenStack:** software tools for building and managing cloud computing platforms for public and private clouds.
  - **KVM** hypervisor: what the VMs run on
  - **Ceph**: storage platform that stores data on a single distributed computer cluster, and provides interfaces for **object**-, **block**- and *file-level* storage.
  - **Operating systems**: CentOS, Ubuntu, Windows(?)
  - **Applications**; e.g. software developed by the domain specialist, gateways, etc.

API Access to Jetstream

- What was **unexpected**
  - **Demand** for **programmable cyberinfrastructure**
  - Great platform for learning **system administration skills**
  - Great platform for **teaching & learning cloudy technologies**

- **Command line clients**
- **Horizon dashboard** very popular; but, incomplete
- **Programmatic control**; python is popular (https://docs.openstack.org/openstacksdk/latest/)
- **Slack channel** for collaboration API users of Jetstream

Using the OpenStack CLI on Jetstream

What an openrc file looks like:

```
export OS_AUTH_URL=https://iu.jetstream-cloud.org:35357/v3
export OS_PROJECT_NAME="TG-ABC190028"
export OS_USER_DOMAIN_NAME="tacc"
export OS_USERNAME=”taccusername”
export OS_IDENTITY_API_VERSION=3
# export OS_PASSWORD=’string’
read -sr OS_PASSWORD_INPUT
export OS_PASSWORD=$OS_PASSWORD_INPUT
```

- Please do not publish the AUTH URLs anywhere
- CLI is python based – reads this information from the environment.
- Horizon can generate an openrc file for you (see the Wiki docs)
- Common pitfall – make sure you specify the correct Project (allocation) if you have more than one!

Installing the client

- Simple on most Mac OS X and Linux hosts (a single pip command)
- Less simple, but still do-able on Windows
  - Once you have a python installed, becomes a simple pip install
- Latest python-openstackclient (> 4.0.0) works with Python 3
- Best practice – use a virtual environment like virtenv for your install
- Docs on the wiki for this!
- Other CLI clients are available – e.g. python-swiftclient (Swift and S3), python-heatclient (Heat templates), etc
  - These are optional and not necessary for basic operations!

Horizon GUI interface

- Allows most things you can do from the CLI
- Nice for some tasks
  - Network visualizer is something we tend to use as a troubleshooting tool
  - Easier to look at security groups on Horizon (IMHO)
- Downsides:
  - considerably slower than using CLI
  - not all features are present that are in CLI
  - can't do things programmatically

Log In

Domain: TACC

User Name:

Password:

Getting started with the API

Things you'll set up once (hopefully):

- SSH keys
- Security groups (though you'll build on the basics as you do more advanced things)
- Create a network
- Create a subnet
- Create a router

Things you'll likely do many times:

- Create and launch instances
- Screw up and delete instances
- Launch more instances
- Expand security groups

API CLI Tutorial walkthrough: https://github.com/jlf599/JetstreamAPITutorial (We'll be using this today!)
API Horizon walkthrough: http://wiki.jetstream-cloud.org/Using+the+OpenStack+Horizon+GUI+Interface

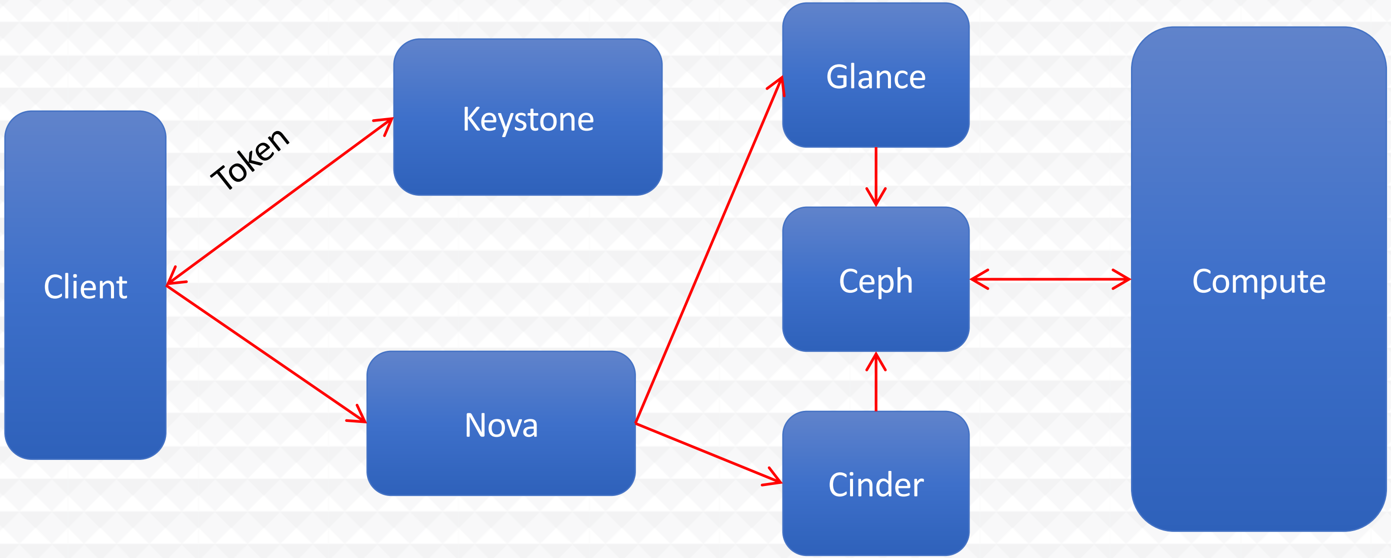
API General Best Practices

- Jetstream-specific – don’t use Atmosphere images on the API side (start with JS-API-Featured-* images)
- Think about your security groups and only open what you REALLY need to open.
- Give objects unique and descriptive names
- When in doubt, use the universally unique identifier (UUID)
- When deleting items, use the universally unique identifier (UUID)
- Before deleting anything, though, “measure twice, cut once”
- Understand that an allocation/tenant lets you see everyone else’s things. Be aware and be ware of deleting things – do unto others…
- Put your toys away if you’re done with them

Security Best Practices

- Think about your security groups and only open what you REALLY need to open. (yes, it's in the slides twice…on purpose…)
- In a production system, you'd likely want to also run a host-based firewall in addition to security groups (defense in depth!)
- Update often! Unattended security upgrades should be turned on in JS-API-Featured-* images…but still…
- Turn off any services/listeners you do not need
- For any service you run on a host, limit the access as much as possible – if it's world accessible, make sure permissions and privileges are as limited as possible
- Limit the number of people that interactively login – and create accounts for them instead of using shared accounts (e.g. centos or ubuntu account)
- Monitor the logs – lots of tools out there to help with this!

OpenStack Overview



Where can I get help?

- *Wiki / Documentation: http://wiki.jetstream-cloud.org*
- *API CLI Tutorial: https://github.com/jlf599/JetstreamAPITutorial*
- *User guides: https://portal.xsede.org/user-guides*
- *XSEDE KB: https://portal.xsede.org/knowledge-base*
- *Email: help@xsede.org*

Acknowledgements

NSF Awards 1053575 & 1548562 (XSEDE), 1445604 (Jetstream) and 2005506 (Jetstream2)

This document was developed with support from the National Science Foundation. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

Jetstream Partners

INDIANA UNIVERSITY
PERVASIVE TECHNOLOGY INSTITUTE

JOHNS HOPKINS
UNIVERSITY

THE UNIVERSITY OF ARIZONA®

TACC

THE UNIVERSITY OF CHICAGO

CORNELL UNIVERSITY FOUNDED A.D. 1865

DELL

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
UTSA

UNIVERSITY OF HAWAI'I
MĀLAMALAMA 1907
UA MAU KE EA O KA ʻĀINA I KA PONO

NSIDC
National Snow and Ice Data Center

JSU 1877 JACKSON STATE UNIVERSITY®

UNC
THE ODUM INSTITUTE

Jetstream2 partners

ASU Arizona State University
THE UNIVERSITY OF ARIZONA
CORNELL UNIVERSITY FOUNDED A.D. 1865
UNIVERSITY OF HAWAI'I
MĀLAMALAMA
1907
UA MAU KE EA O KA 'ĀINA I KA PONO
JOHNS HOPKINS UNIVERSITY
TACC
UCAR

NSF
http://jetstream-cloud.org/
National Science Foundation
Award #ACI-2005506

Getting into the hands on part -

Open https://tinyurl.com/jetstreamAPI in your browser

You'll be connecting to a prepped host…account list is here:

https://tinyurl.com/xsede-ecss

```
ssh trainXX@tutorial.jetstream-cloud.org
```

or

```
ssh -o PubkeyAuthentication=no your_training_user@tutorial.jetstream-cloud.org
```