Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

Multitenancy at SDCC

Chris Hollowell - Scientific Data and Computing Center (SDCC)

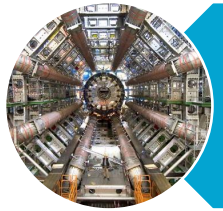
Middleware and Grid Interagency Coordination (MAGIC) Meeting - 2/1/2023

@BrookhavenLab

BNL: A Multi-Disciplinary Laboratory

RHIC
Relativistic Heavy Ion Collider

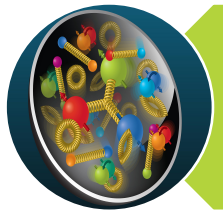
NSLS II
National Synchrotron Light Source II



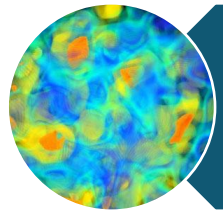
ATLAS
Large Hadron Collider

ARM
Atmospheric Radiation Measurement

Brookhaven
National Laboratory



EIC
Electron-Ion Collider



LQCD
Lattice Quantum Chromo-Dynamics

CFN
Center for Functional Nanomaterials

Belle II
Experiment in Japan

NVBBCC
National Virtual Biosecurity for Bioenergy Crops Center

SDCC Overview

- Supporting over 20 projects with ~2,500 users

- Providing our users with:
  - ~100k CPU cores
    - 2 large High Throughput Compute (HTC) farms
    - 5 High Performance Compute (HPC) Clusters
    - ~4.5 PFLOPS aggregate compute capacity
  - ~200 PB tape storage
  - ~100 PB disk storage
  - 2x100 Gbps external connectivity

- *Facility-level multitenancy allows SDCC to leverage synergies and share infrastructure and staff between groups*
  - Reduces cost to programs
    - Economies of scale and critical mass of expanding knowledge
  - Allows for the use of additional resources and staff during peak demand

Storage Multitenancy at SDCC

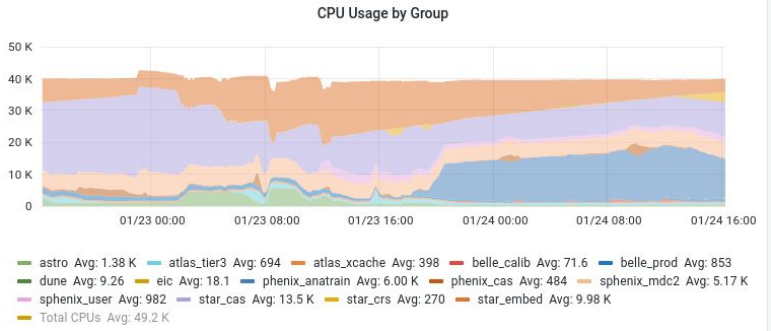
- GPFS and Netapp AFF NFS appliance network filesystems (home directories) at our facility are generally shared by experiments/tenants
  - Space restrictions implemented via quota
  - Standard UNIX file permissions/ACLs for access control sufficient in our environment
  - We have found the performance/scaling of these systems to be sufficient that the I/O operations of one experiment do not typically impact others

- Have adopted Lustre for new storage
  - Primarily on separate filesystems/hardware for tenants as of now
    - Additional testing of multi-tenant performance/availability impacts needed

- HPSS tape system multitenancy
  - We have a Single HPSS core supporting multiple experiments/tenants
  - Dedicated mover hardware for each large tenant
    - Dedicated TS4500/SL8500 libraries in some cases
    - Unique class of service (COS) setting for each tenant, and a dedicated ERADAT tape batch system instance

Compute Multitenancy at SDCC

- The majority of our HPC and HTC resources are shared among multiple experiments/tenants
  - Allows tenants to make opportunistic use of additional compute resources when others are not using their allocations



- SLURM is configured for whole-node scheduling on all but one of our HPC clusters
  - Primarily for performance reasons, but also simplifies the accounting/charge model

- On our HTC farms, HTCondor is configured to allow multiple jobs slots per node
  - Multiple jobs/users per batch node
  - Partitioned dynamically via job CPU core and memory allocation requests
    - HTC applications are typically embarrassingly parallel, and not as sensitive to absolute individual process performance as HPC applications
      - More of a focus on how many jobs you can run simultaneously over individual job performance

Brookhaven National Laboratory

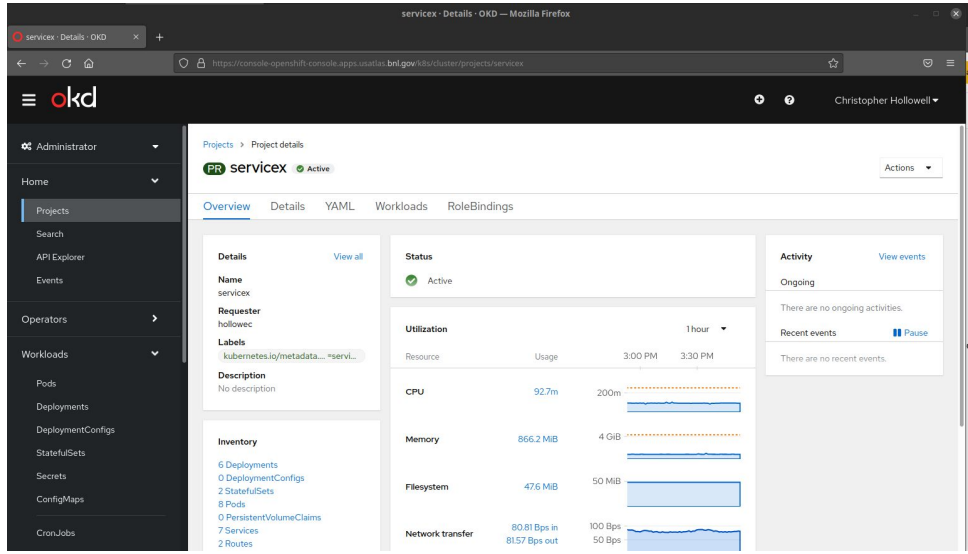
Compute Multitenancy at SDCC (Cont.)

- On our muti-user/job batch hosts, Linux Cgroups is utilized to reduce potential cross-job interference/contention
  - Supported by both HTCondor and SLURM
    - In our configuration:
      - HTCondor - cpu.shares, memory.soft_limit_in_bytes
        - Utilizing memory soft limit can still result in excessive swapping, which can impact other jobs
      - SLURM - using the task/cgroup plugin to pin jobs to CPU/NUMA-node sets, GPUs, and set memory limits
  - Other cgroups controllers available to limit IOPs (blkio) and network bandwidth (net_cls), but not used in our environment

- Our multi-tenant environment does not permit users to utilize Docker on our systems
  - Docker users are root by default in containers in the default installation/configuration, and can bindmount in arbitrary host paths
  - *We provide Singularity/Apptainer as a secure alternative*
  - Podman also available for some users/systems
    - Both Apptainer and Podman utilize user namespaces for rootless execution by default

Multitenancy and Container Orchestration

- Increasing requests from our user community to internally support user-provisioned services
  - Databases, analysis platforms, web services, etc.
  - Essentially, interested in a private cloud
- Kubernetes (k8s) is the logical tool to provide this capability, but its default configuration creates a number of security issues in a multi-tenant environment
  - *Users can start containers with root privileges, mount arbitrary system paths into containers, etc.*
  - Possible to work around issues through the setup of admissions controllers, RBAC, etc.
    - Not trivial and easily opens the door for administrator error
- *Therefore, decided to adopt OKD for our k8s needs*
  - The community release of Red Hat's Openshift k8s platform
  - Secure out the box - suitable for multi-tenant use
    - Users are never root in containers by default
    - OKD/Openshift adopted at a number of other US national labs including FNAL and ORNL



Brookhaven National Laboratory

Conclusions

- The Scientific Data and Computing Center (SDCC) at BNL is a large multitenant computing facility
  - Supporting over 20 programs/experiments with ~2,500 users
  - Facility-level multitenancy allows for the leveraging of synergies
    - Share staff and infrastructure - reduces overall cost to experiments/programs
    - Allows opportunistic use of resources by any other tenants not fully utilizing their share

- Most of our storage and compute resources are shared by our tenants
  - Our HTC compute nodes run multiple jobs from different tenants/users simultaneously
    - Cgroups used to enforce resource limits and reduce potential interference between jobs
  - HPC clusters primarily utilizing whole-node scheduling

- Utilizing Singularity/Apptainer, Podman, and OKD to provide secure multi-tenant container and container orchestration support for our users

"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 Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development (NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: https://www.nitrd.gov