High-Performance Data-Intensive Computing Systems Laboratory

National Research Platform Nautilus Research Cluster

MORENet Technical Summit

20 Feb 2023

University of Missouri

NSF NRP Nautilus HyperCluster

- The NSF Nautilus HyperCluster is a Kubernetes cluster with vast resources that can be utilized for various research purposes:
  - Prototyping research code
  - S3 cloud storage for data and models
  - Accelerated small-scale research compute
  - Scaling research compute for large scale experimentation
- Resources Available:
  - CPU Cores: 14,462
  - RAM: 69 TB
  - NVIDIA GPUs: 1150

Nautilus Logo: https://portal.nrp-nautilus.io/
Docker Logo: https://developers.redhat.com/blog/2014/05/15/practical-introduction-to-docker-containers
Kubernetes Logo: https://commons.wikimedia.org/wiki/File:Kubernetes_logo_without_workmark.svg

Nautilus ~1,100 GPUs Distributed over US Networks—Fall 2022

U S. Dakota + SD State
9 GPUs over GPN

U. Nebraska-L
160 GPUs over GPN

MGHPCC
144 GPUs over NEREN

UCI + UCR + UCM + UCSC + UCSB
99 GPUs over CENIC

U Oklahoma
8 GPUs over GPN

UIC
21 GPUs over MREN

NYU
12 GPUs over NYSERNet

U Delaware
12 GPUs over NYSERNet

U Hawaii
1 GPU over CENIC/PW

CSUSB + SDSU
48 GPUs over CENIC

CWRU
2 GPUs over OARnet

Clemson U
21 GPUs over SCLR

UCSD
612 GPUs over CENIC

U New Mexico
4 GPUs via Albuquerque GigaPoP

FAMU
8 GPUs over FLR (pending)

U Guam
1 GPU over CENIC/PW

Minority Serving Institutions

EPSCoR Institutions

Non-MSI Institutions

City Pair

State-to-State Pair

Inter-Connect Point

QUILT MEMBERS & AFFILIATES

5 PB Nautilus Ceph Storage Over Networks—2022

UCR + UCSC + UCSB
940 TB over CENIC

UN-L
400 TB over GPN

U Kansas
200 TB over GPN

UIC
175 TB over MREN

NYU + NYSERNet
400 TB over NYSERNet

USC + Stanford + Caltech
737 TB over CENIC

SDSU
109 TB over CENIC

OneNet
200 TB over GPN

U Delaware
200 TB over NYSERNet

U Hawaii
266 TB over CENIC/PW

UCSD + UCLA
901 TB over CENIC

U Arkansas
200 TB over GPN

FAMU + FIU
318 TB over FLR

U Guam
118 TB over CENIC/PW

Minority Serving Institutions

EPSCoR Institutions

Non-MSI Institutions

City Pair

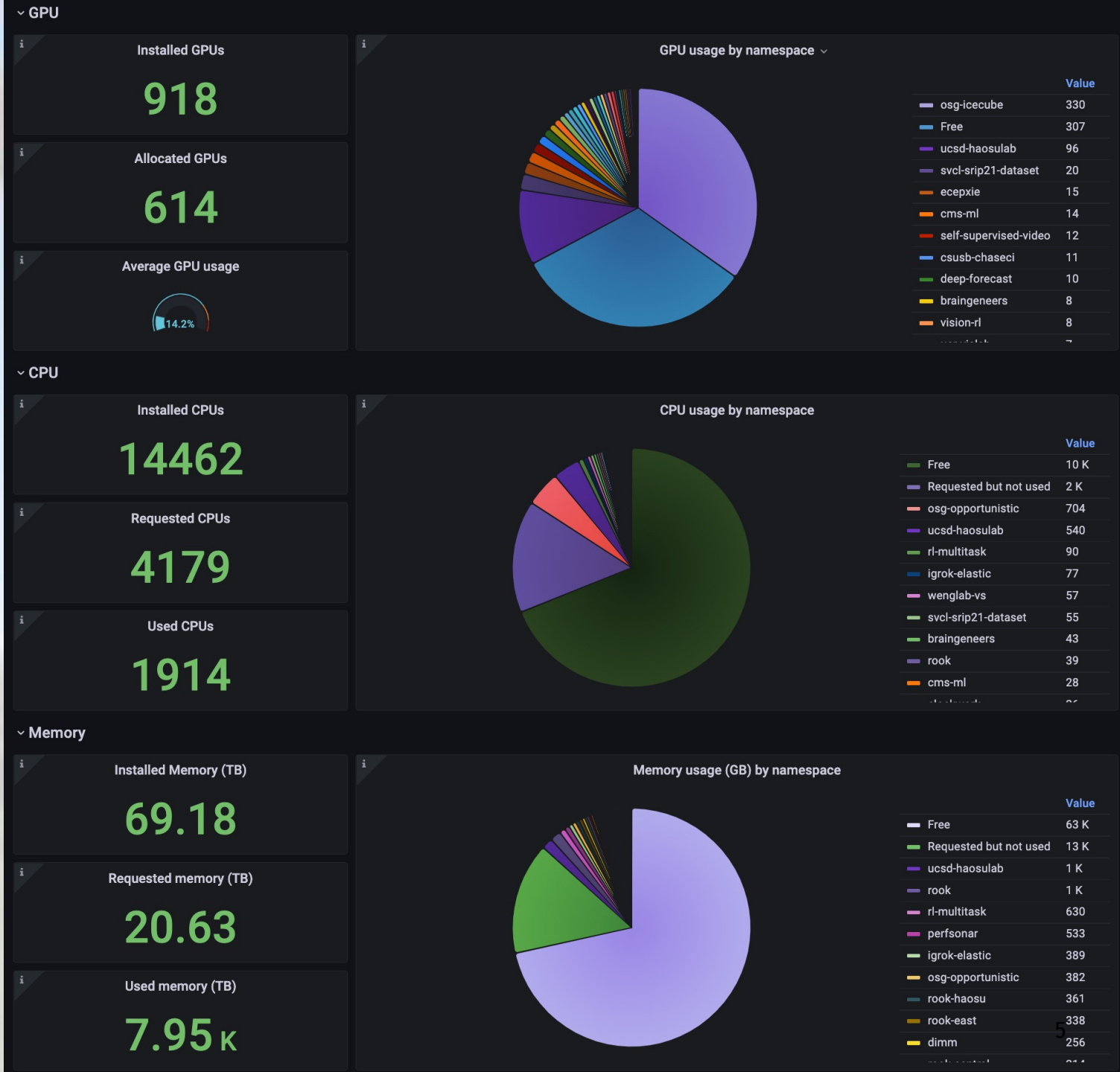
State-to-State Pair

Inter-Connect Point

THE QUILT

QUILT MEMBERS & AFFILIATES

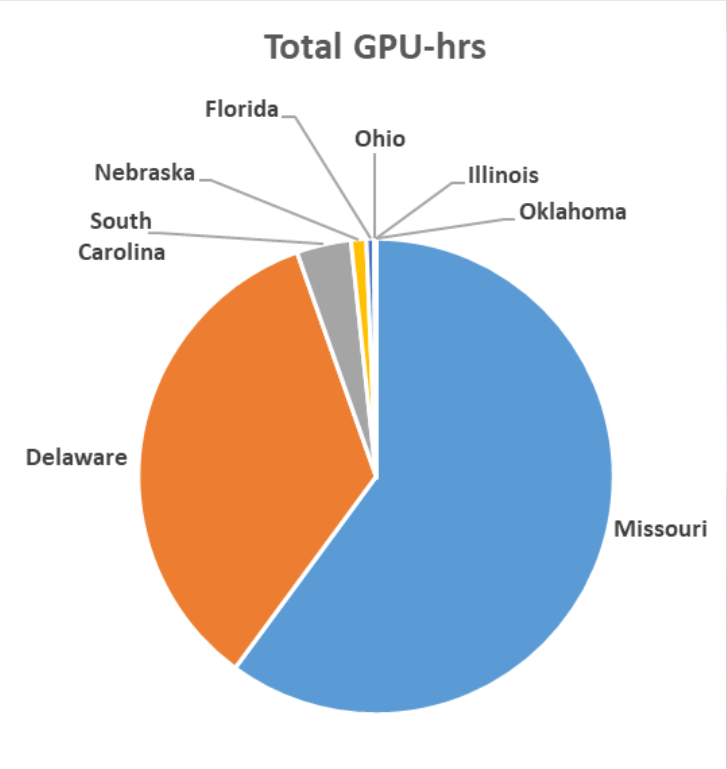
Resource Dashboard



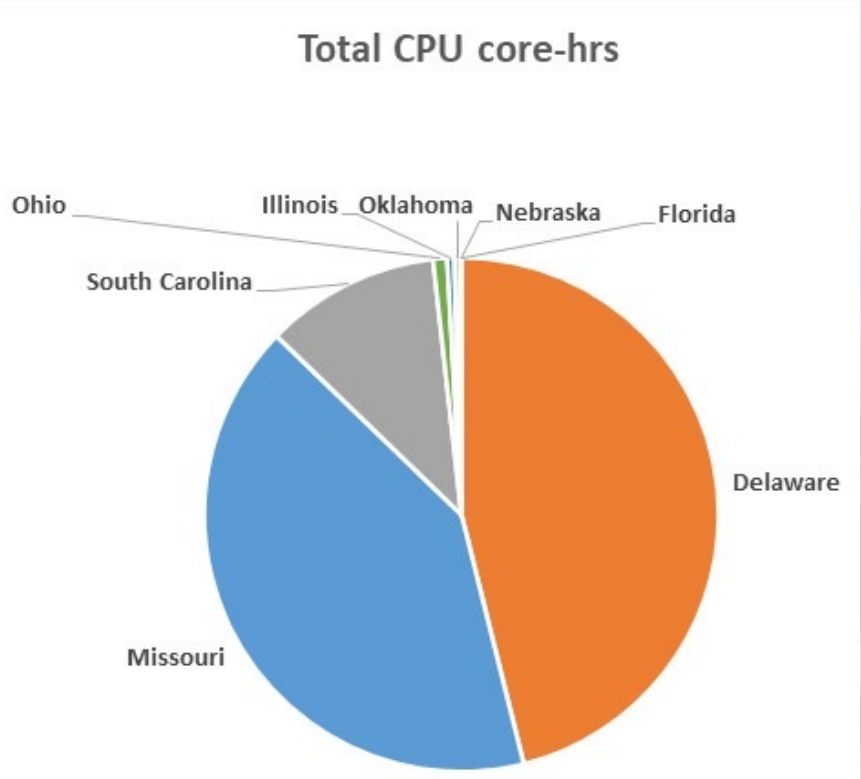
University of Missouri

Non-California Nautilus PI Namespace 2021 Usage by State: "Big MO!"

Data/Plots provided by Larry Smarr (PI, National Research Platform & father of US Super Computing Centers)



17,217 GPU-hrs



28,088 CPU core-hrs

University of Missouri - Columbia:
42,000 GPU-hrs in 2022!

University of Missouri

Grant Scott, UMC
Helped Organize the UMC PRP Usage

How MU is using Nautilus: Teaching & Research

- JupyterHub integration enables creation of an interactive learning environment for STEM Education:
  - Data Science
  - Mathematics
  - High Performance Computing
- Access to vast amount of RAM, CPU Cores, and NVIDIA GPUs can accelerate research in various fields:
  - Bioinformatics
  - Remote Sensing
  - Materials Science
  - Computer Vision
  - Machine Learning

HPC Interactive Learning Environment

High-Performance Data-Intensive Computing Systems Laboratory

https://scottgs.mufaculty.umsystem.edu/

File Edit View Run Kernel Tabs Settings Help

fork_example.cpp

```cpp
/* Fork'n example

*/

#include <sys/types.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <unistd.h>
#include <cstdlib>

#include <iostream>


int main(int argc, char * argv[])
{
    pid_t pid; // process identifier, pid_t is a process id type defined in sys/types

    pid = fork(); // fork, which replicates the process

    if ( pid < 0 )
    {
        std::cerr << "Could not fork!!! ("<< pid <<")" << std::endl;
        exit(1);
    }

    std::cout << "I just forked without error, I see ("<< pid <<")" << std::endl;

    if ( pid == 0 ) // Child process
    {
        std::cout << "In the child (if): " << std::endl;

        //int myChildExitCode = 0;  // this could be set to something meaningfull or 0;
        int myChildExitCode = 8;  // this could be set to something meaningfull or 0;
        _exit(myChildExitCode);
    }
    else // Parent Process
    {
        std::cout << "In the parent (if-else): " << std::endl;
    }


    std::cout << "In the parent: " << std::endl;

    int status;// catch the status of the child

    do  // in reality, mulptiple signals or exit status could come from the child
    {

```

Terminal 1

```
jovyan@jupyter-jhurt-40missouri-2eedu:~/fork$ make -f fork.make
g++ -o child fork_example.cpp
g++ -o bad_child_seg_fault fork_example_segfault.cpp
g++ -o bad_child_free fork_example_bad_free.cpp
jovyan@jupyter-jhurt-40missouri-2eedu:~/fork$ ./child
I just forked without error, I see (99)
In the parent (if-else):
In the parent:
I just forked without error, I see (0)
In the child (if):
Child process (99) exited with non-zero status of 8
jovyan@jupyter-jhurt-40missouri-2eedu:~/fork$
```

Simple 1 0 Terminal 1

University of Missouri