ACCESS Advancing Innovation

XDMoD METRICS ON DEMAND

University at Buffalo Center for Computational Research

iACS INSTITUTE FOR ADVANCED COMPUTATIONAL SCIENCE Stony Brook University

Utilization and Performance Monitoring of Ookami, an ARM Fujitsu A64FX Testbed Cluster with XDMoD

Nikolay A. Simakov[1], *Joseph P. White*[1], *and Matthew D. Jones*[1], *Eva Siegmann*[2], *David Carlson*[2], *and Robert J. Harrison*[2]

[1]*Center for Computational Research*

SUNY University at Buffalo

[2]*Institute for Advanced Computational Science*

Stony Brook University

OOKAMI

Cray User Group Meeting, Minutes of New York City, NY, May 2025

NSF Award: OAC 1927880 and 2137603

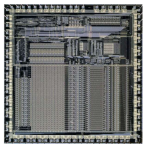
The Ookami Apollo80 system – an ARM Fujitsu A64FX Machine

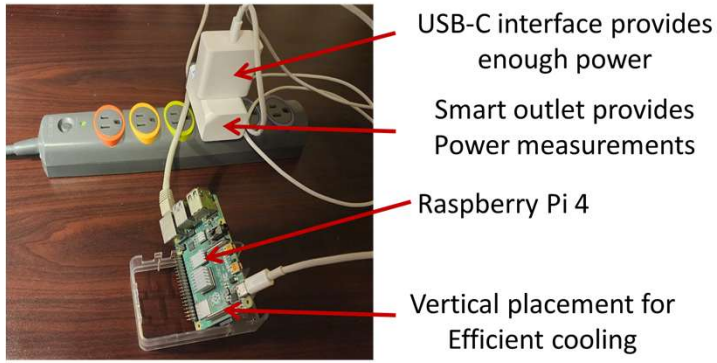
Node	
Processor	A64FX
#Cores	48
Peak DP	2.76 TOP/s
Peak INT8	22.08 TOP/s
Memory	32GB@**1TB/s**

System	
#Nodes	176
Peak DP	486 TOP/s
Peak INT8	3886 TOP/s
Memory	5.6 TB
Disk	0.8 PB Lustre
Comms	IB HDR-100

- **ARM** Fujitsu A64FX – specifically designed for HPC
- First implementation of **SVE** with a 512-bit wide instruction set
- Smaller sibling of the Fugaku supercomputer: homogeneous CPU resource, #1 in Top500 until 06/2022
- NSF testbed system
- Project started in 2020 with first users getting on the system in 2021
- Currently near completion

ARM in HPC





Early Days: ARM's Origin and Mobile Focus (1980s–2000s)

- 1978 -ARM1 by Acorn Computers “aimed to put a computer in every classroom in the UK”
- 1990 - Advanced RISC Machines Ltd was founded
- 1993 - Arm goes into mobile, with the Arm7 processor (1993-2001) becoming the flagship mobile design for Arm

Shift Toward HPC Interest (2010s)

- 2011 - ARMv8-A architecture introduced 64-bit support
- 2012 - Raspberry Pi Model B (ARM11)
- 2013–2015: Early prototypes and clusters using ARM Cortex-A processors (e.g. Cortex-A15)
- 2014 - Cavium ThunderX

Breakthroughs and Ecosystem Growth (2017–2020)

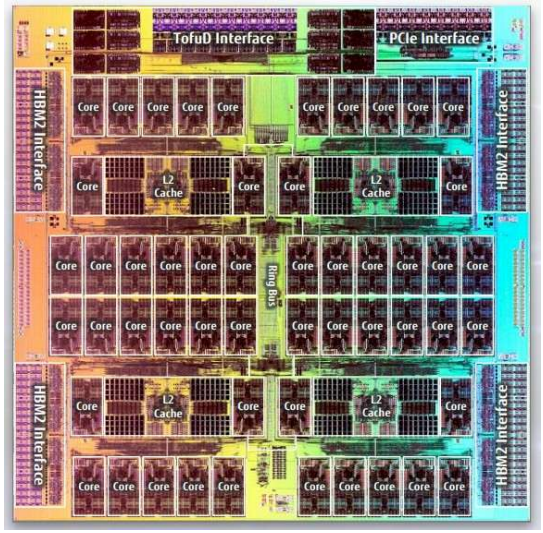
- 2017- Fujitsu, in partnership with RIKEN, began developing the A64FX
- 2018/2019 - AWS Graviton/Graviton 2
- 2020- Fugaku, powered by over 150,000 A64FX processors, became the #1 supercomputer in the world

Recent Developments (2021–2025)

- 2021 - AWS Graviton3
- 2023 - NVIDIA released Grace superchip
- 2023 - AWS Graviton4
- Ampere, European Processor Initiative and more

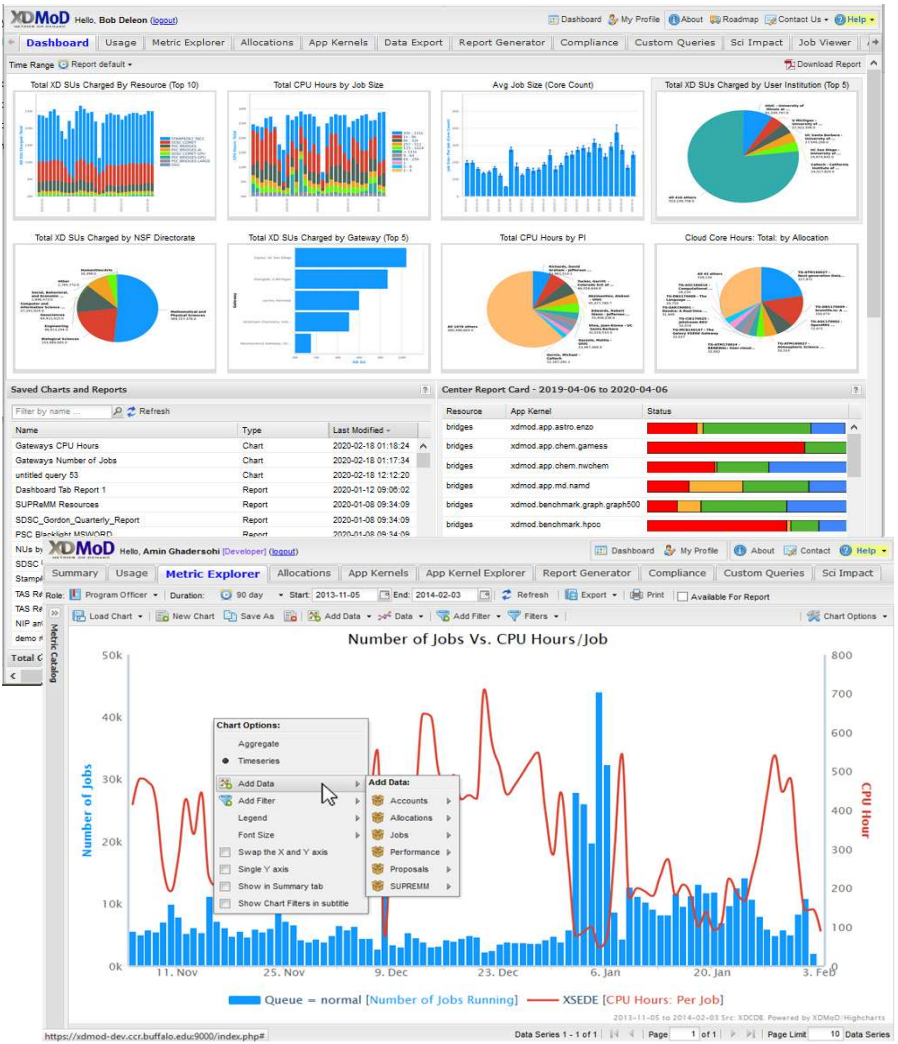
Started with ChatGPT

Why to Monitor Utilization and Performance



- Ensure that users are on the system
  - It is a testbed that poses different types of challenges
  - Higher level of engagement with users
- Inspect what users are doing in the system
  - Applications, job sizes, walltime,...
- Report the usage and analyse whether it matches the desired state of the project
- Continuous performance monitoring to ensure optimal software and hardware state
  - Monitore the performance improvement with higher adoption of technologies
  - Benchmarking and comparison with other platforms

XDMoD: A Comprehensive Tool for HPC System Management



Goal: Optimize Resource Utilization and Performance

- Provide detailed information on utilization
- Continuous performance monitoring
- Enable data-driven upgrades and procurements
- Measure and improve job and system-level performance

NSF ACCESS Measurement and Metrics Service (MMS)

- Develop & deploy the **XDMoD** for monitoring ACCESS-CI

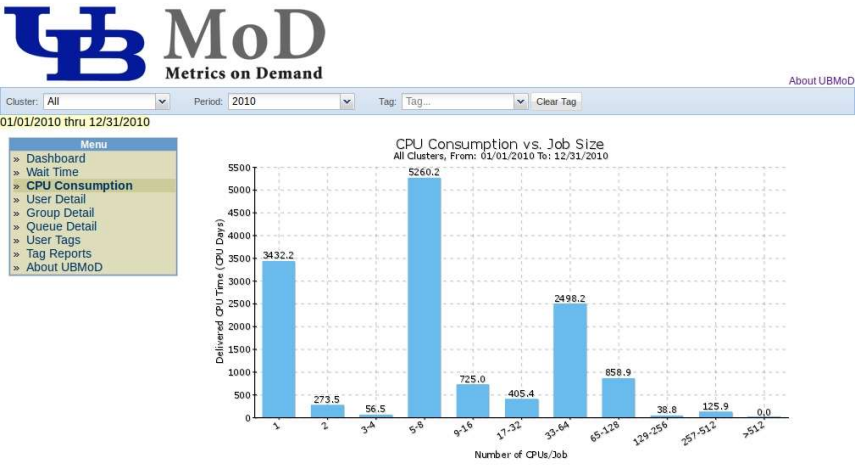
Open XDMoD: Open Source version for Data Centers

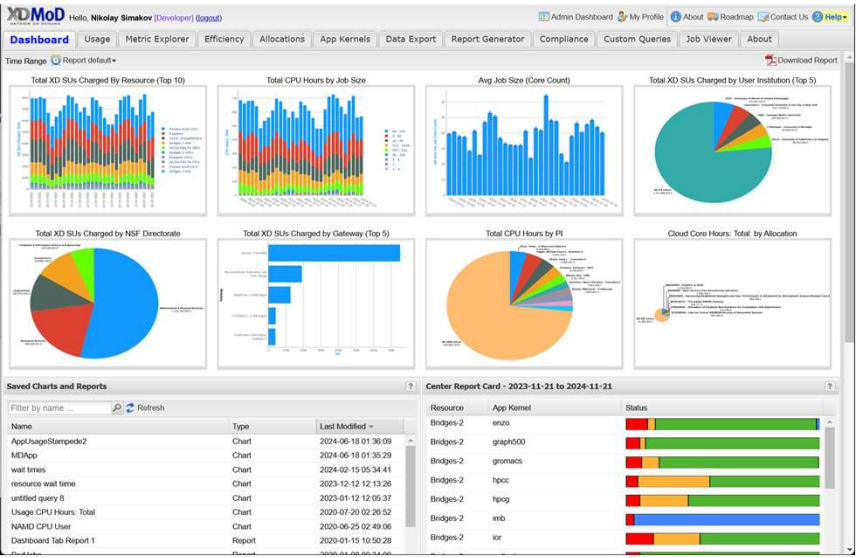
- Used to measure and optimize the performance of HPC centers
- 300+ academic & industrial installations worldwide

XDMoD METRICS ON DEMAND

University at Buffalo
Center for Computational Research

XDMoD History

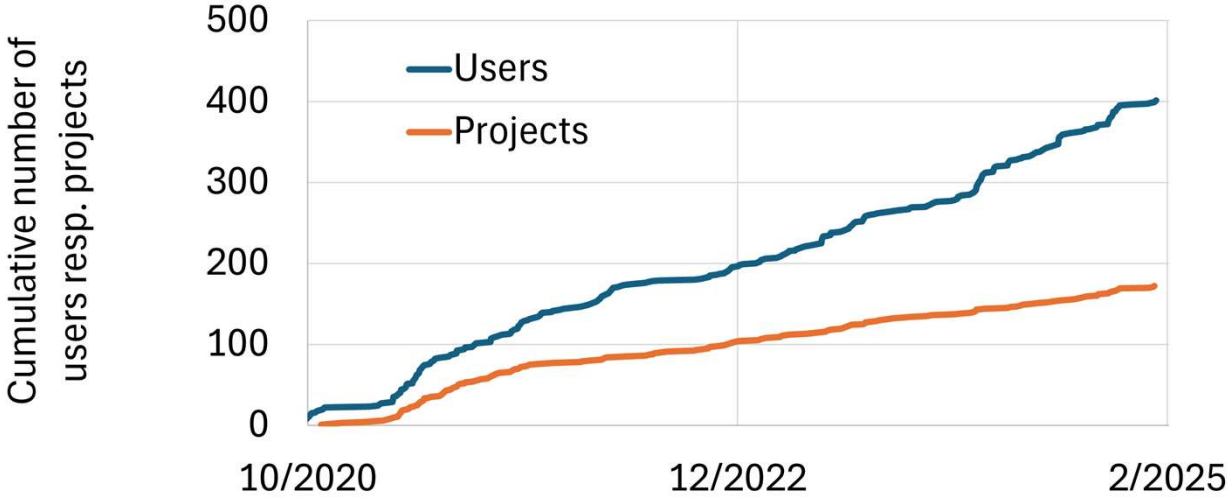


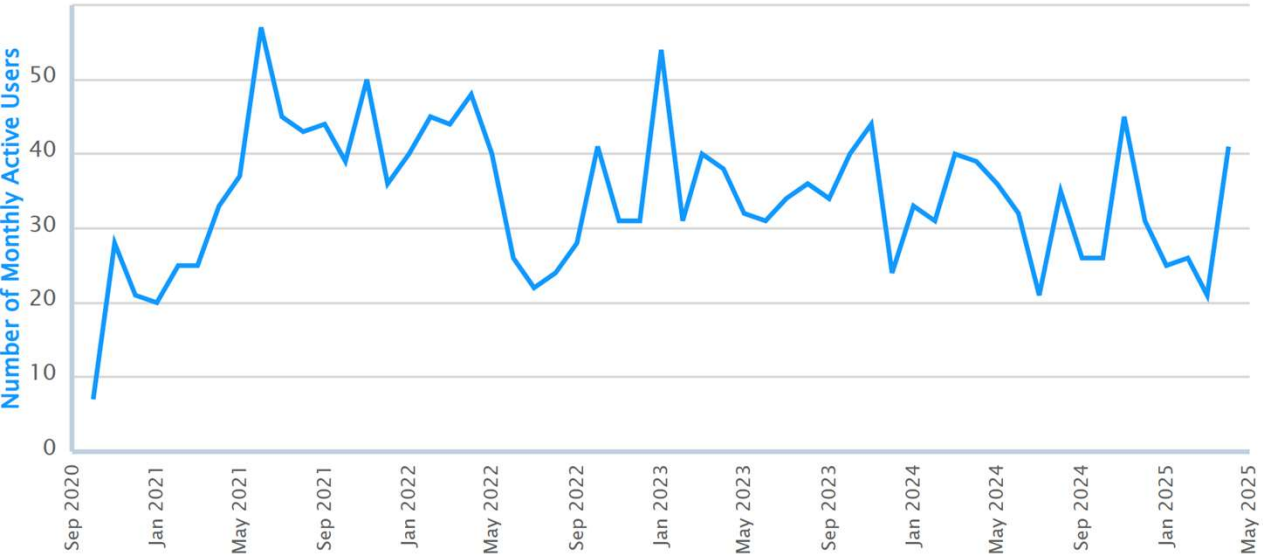


- UBMoD (UB Metrics on Demand) - tool for basic usage and accounting information visualization. 2007
- XDMoD – NSF-funded tool for monitoring XSEDE/ ACCESS cyberinfrastructure, a collection of NSF-funded HPC resources. From 2010. Open XDMoD – open source version of XDMoD for HPC centers. First public release 2014
- Continuous Performance Monitoring was part of XDMoD from inception
  - Referred to as “App Kernels”
  - Initially tried to leverage Inca (automated user-level testing of the software and services) for job execution automation
  - Developed Application Kernel Remote Runner for automated jobs execution on HPC resources
  - **Monitoring XSEDE/ACCESS resources from 2011**
  - First public release 2014

Ookami Utilization

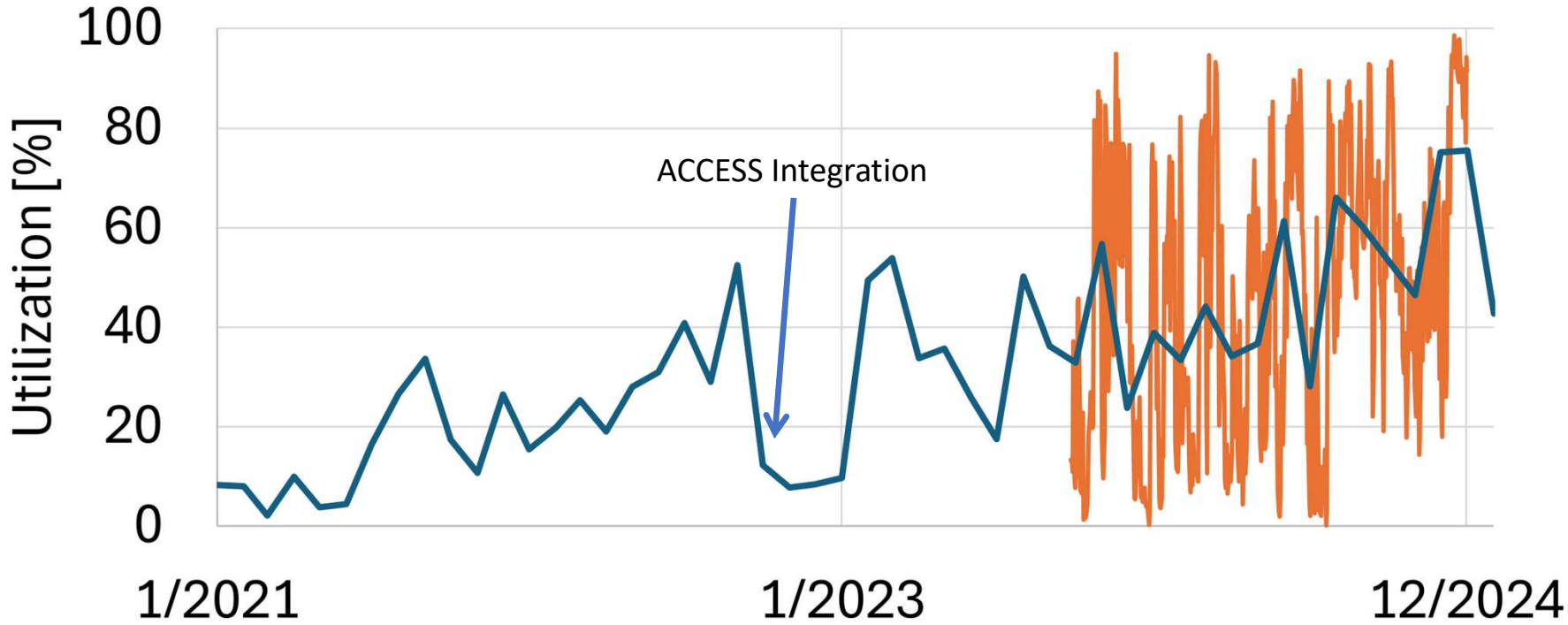
Number of Users





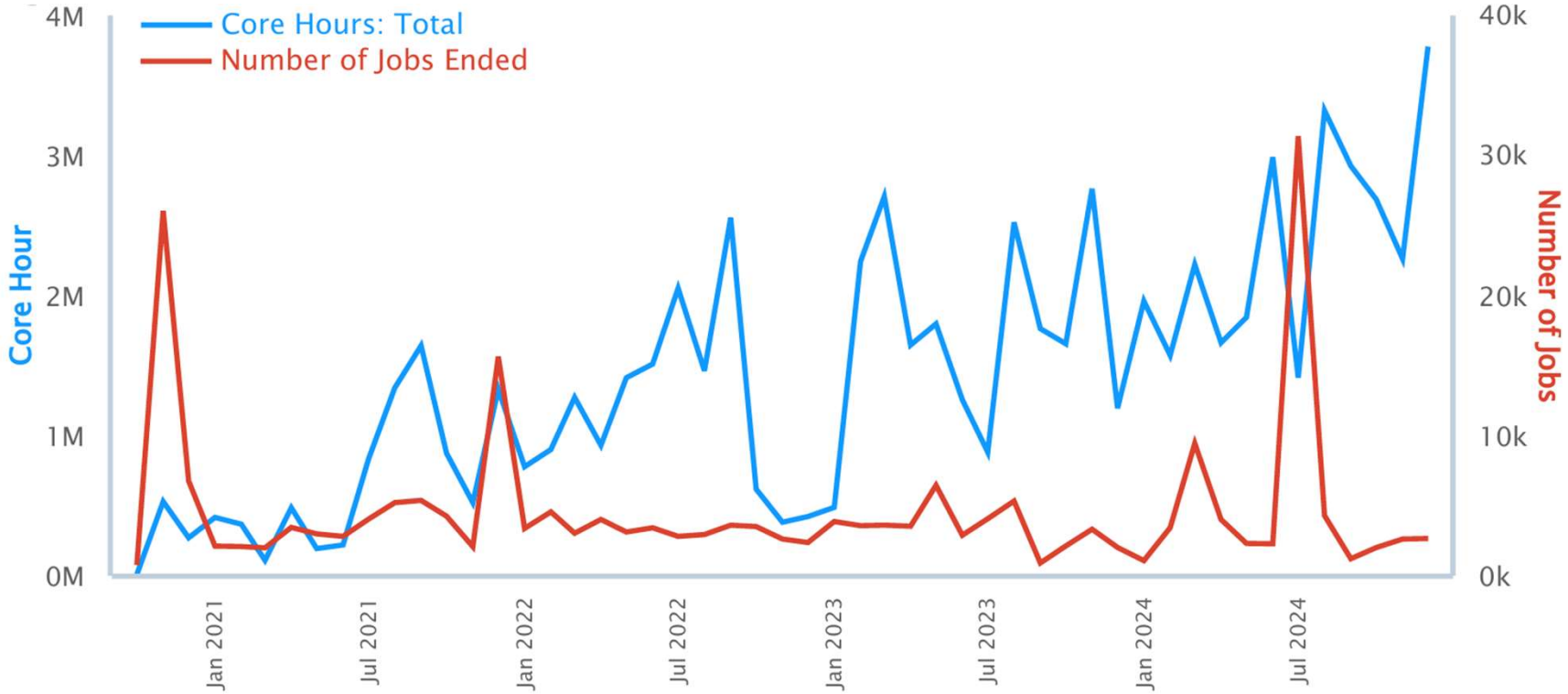
- The cumulative number of users increases with a relatively stable number of active users

Overall Utilization



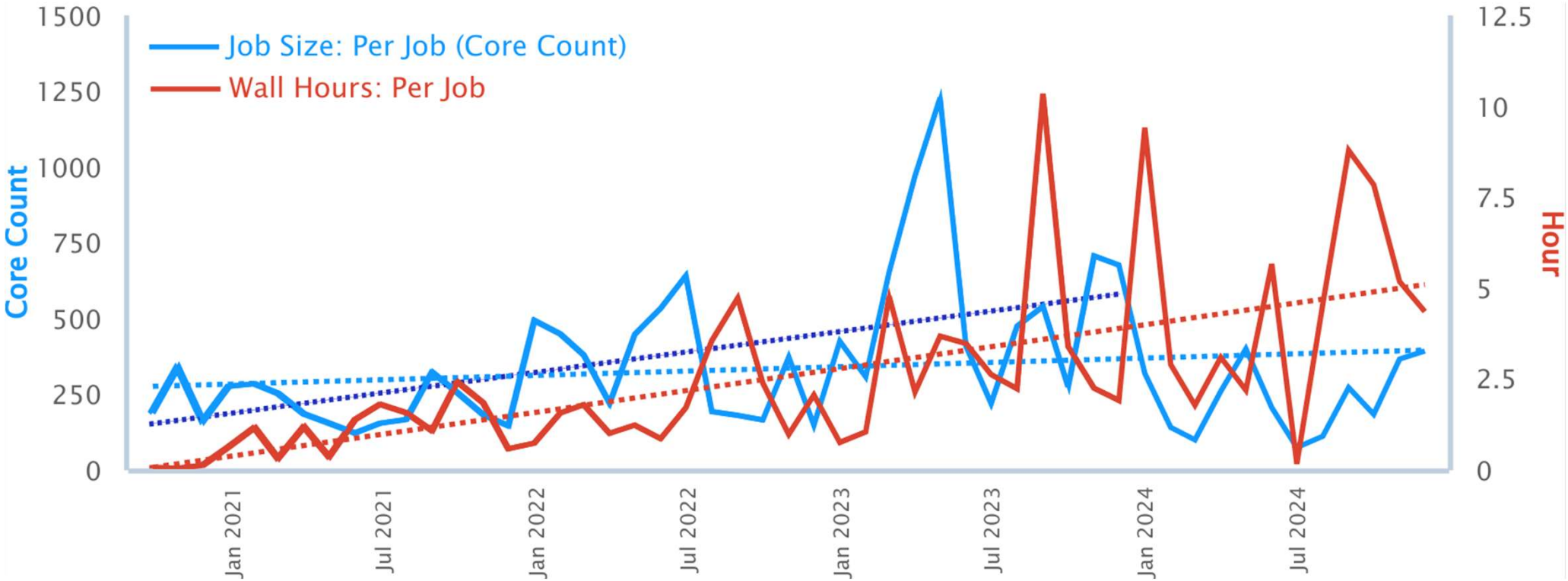
- Low to moderate utilization
  - Good for development
- Monthly utilization increases over time
- On some days, utilization is 80+%

Overall utilization 2



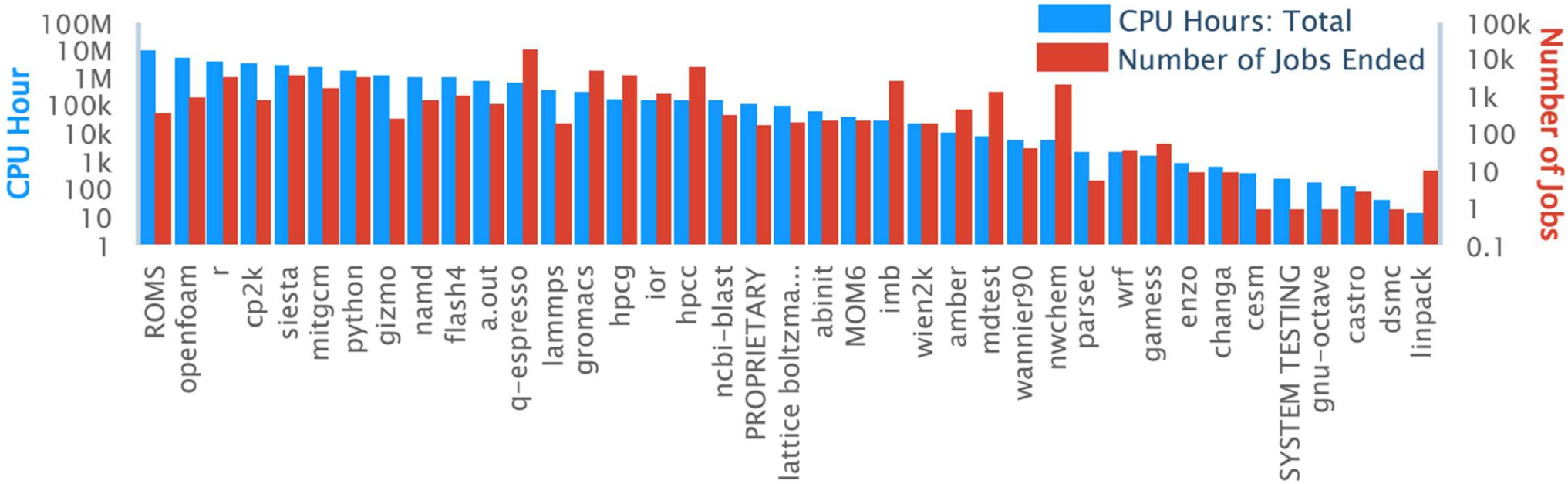
- Total core hours increase while number of jobs is relatively flat

Overall utilization 3



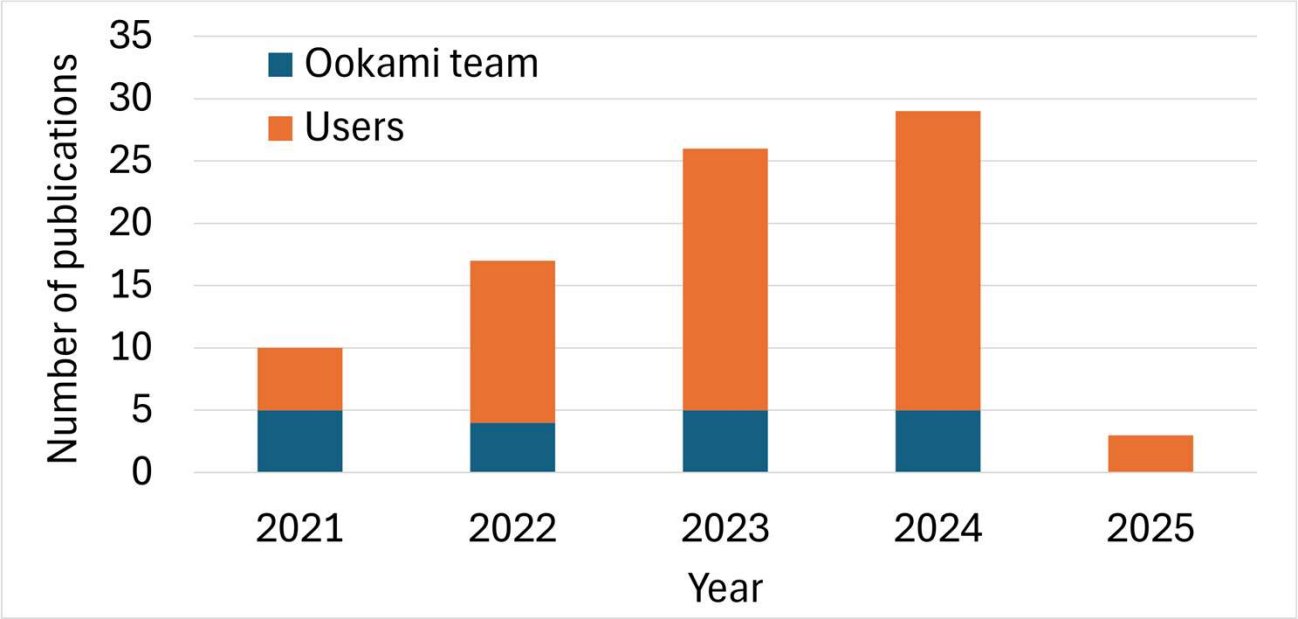
- Both core count and walltime increase

Applications



- Wide range of applications were used

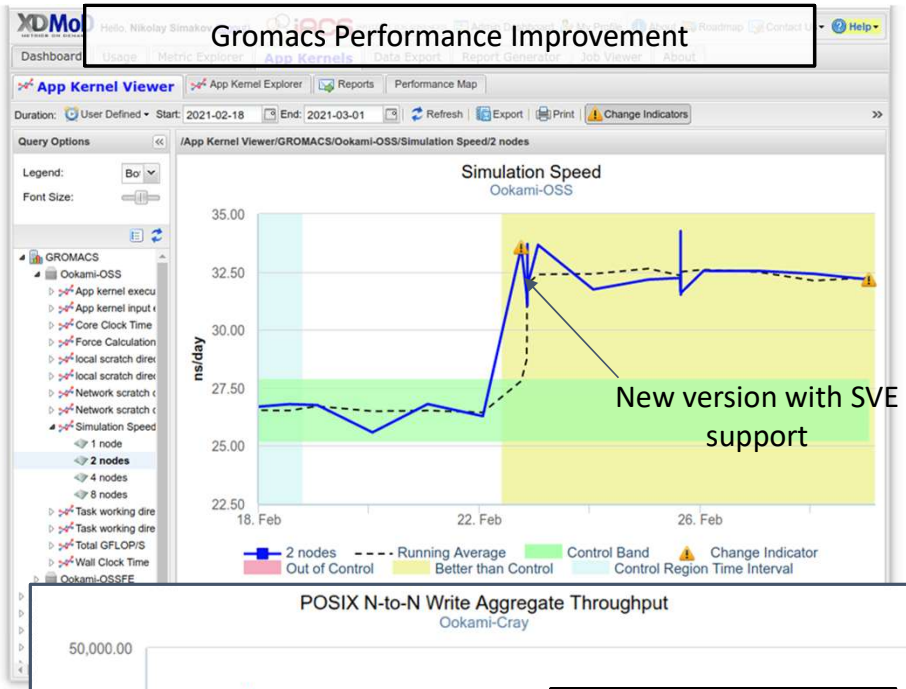
Publications

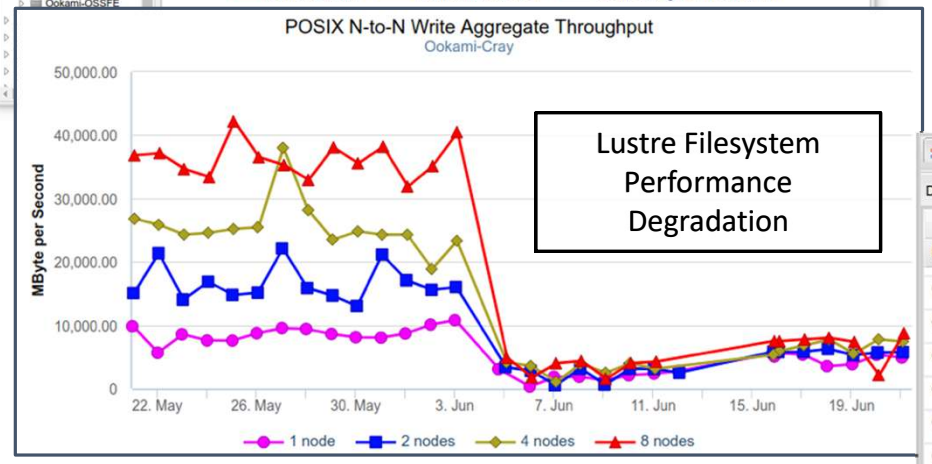


- A large number of publications by Ookami team and users

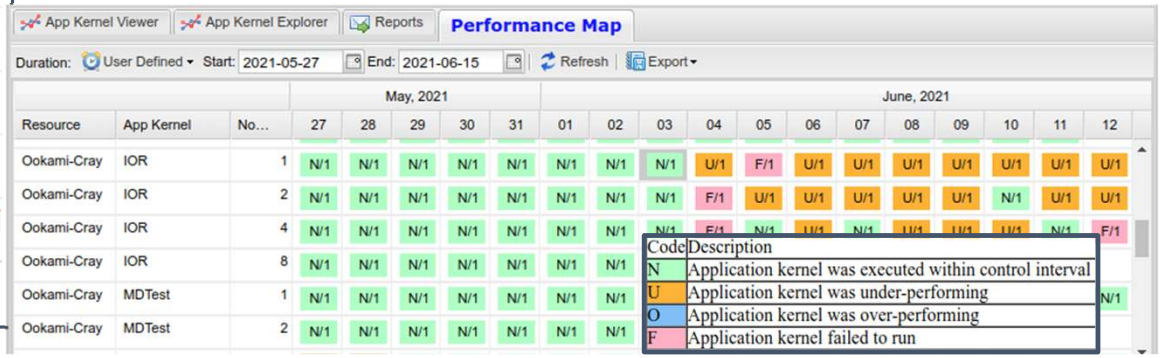
Continuous Performance Monitoring with XDMoD

Continuous Performance Monitoring with XDMoD–



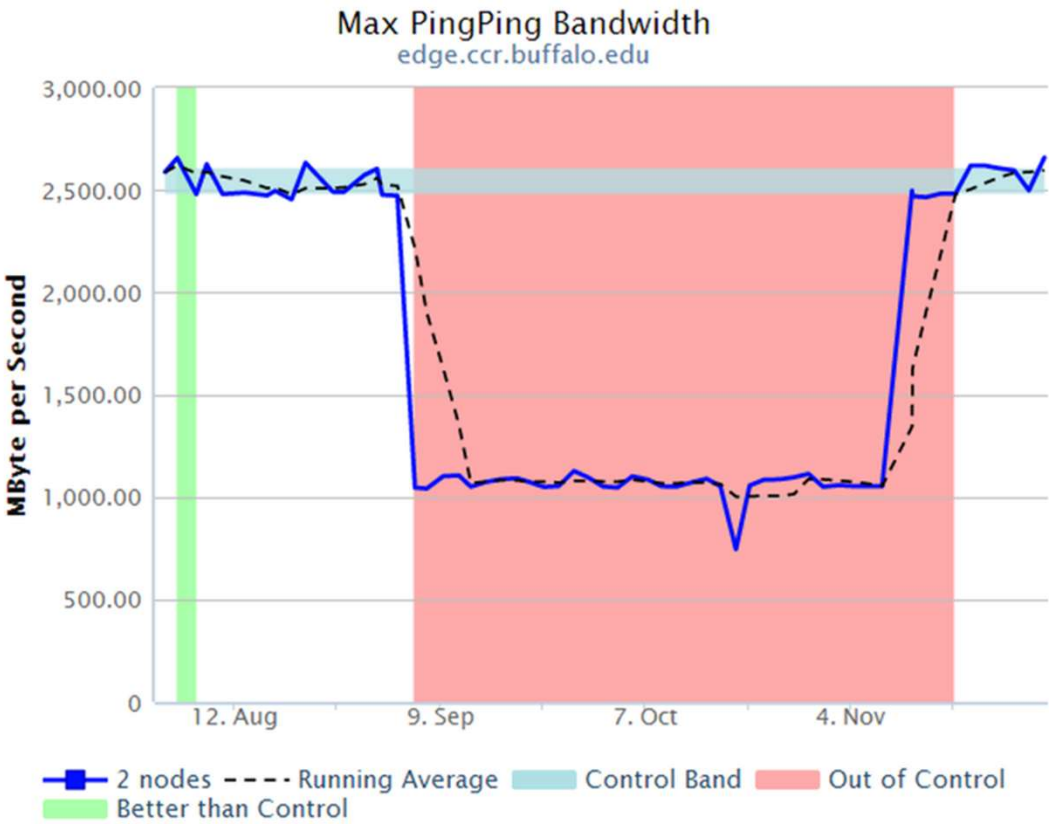


- Application Kernels = Application + Input + Periodic execution
- Proactively identify underperforming hardware and software by regular (daily)
  - Measure Quality of Service (QoS)
- Computationally intensive but short
  - Run periodically or on demand to actively measure performance
- Measure system performance from User's view
  - Local scratch, global filesystem performance, local processor-memory bandwidth, allocatable shared memory, processing speed, network latency and bandwidth



Performance Changepoint Detection

Faulty Core Switch



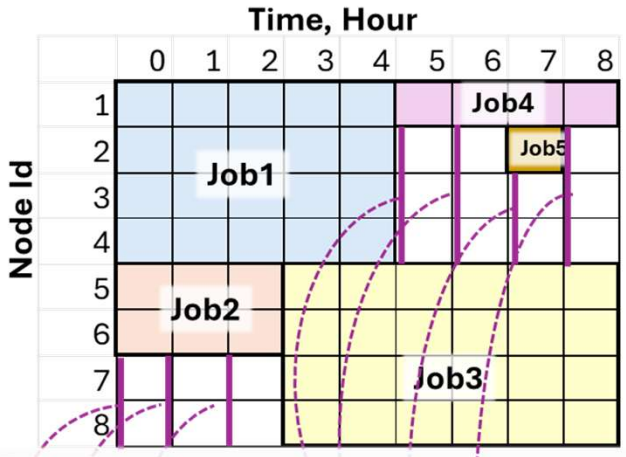
The control region is calculated from multiple runs (typically 10-20)

A region three standard deviations away from the mean in both directions is an **in-control region**

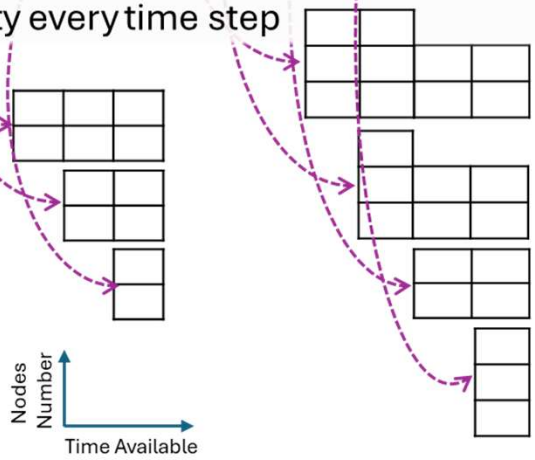
If the five-point running average (five most recent runs) falls outside the in-control region, the last run is marked as either under or over-performing, depending on the direction

Can It Run Without Affecting User Jobs?

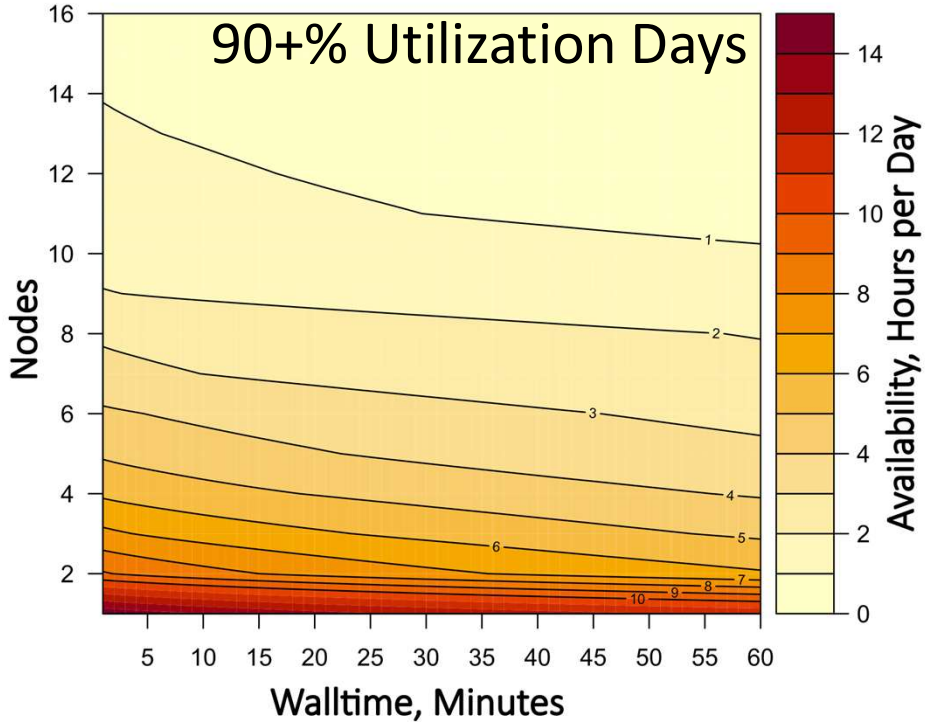
1. Calculate individual node utilization over tim



2. Extract available nodes and duration availability every time step

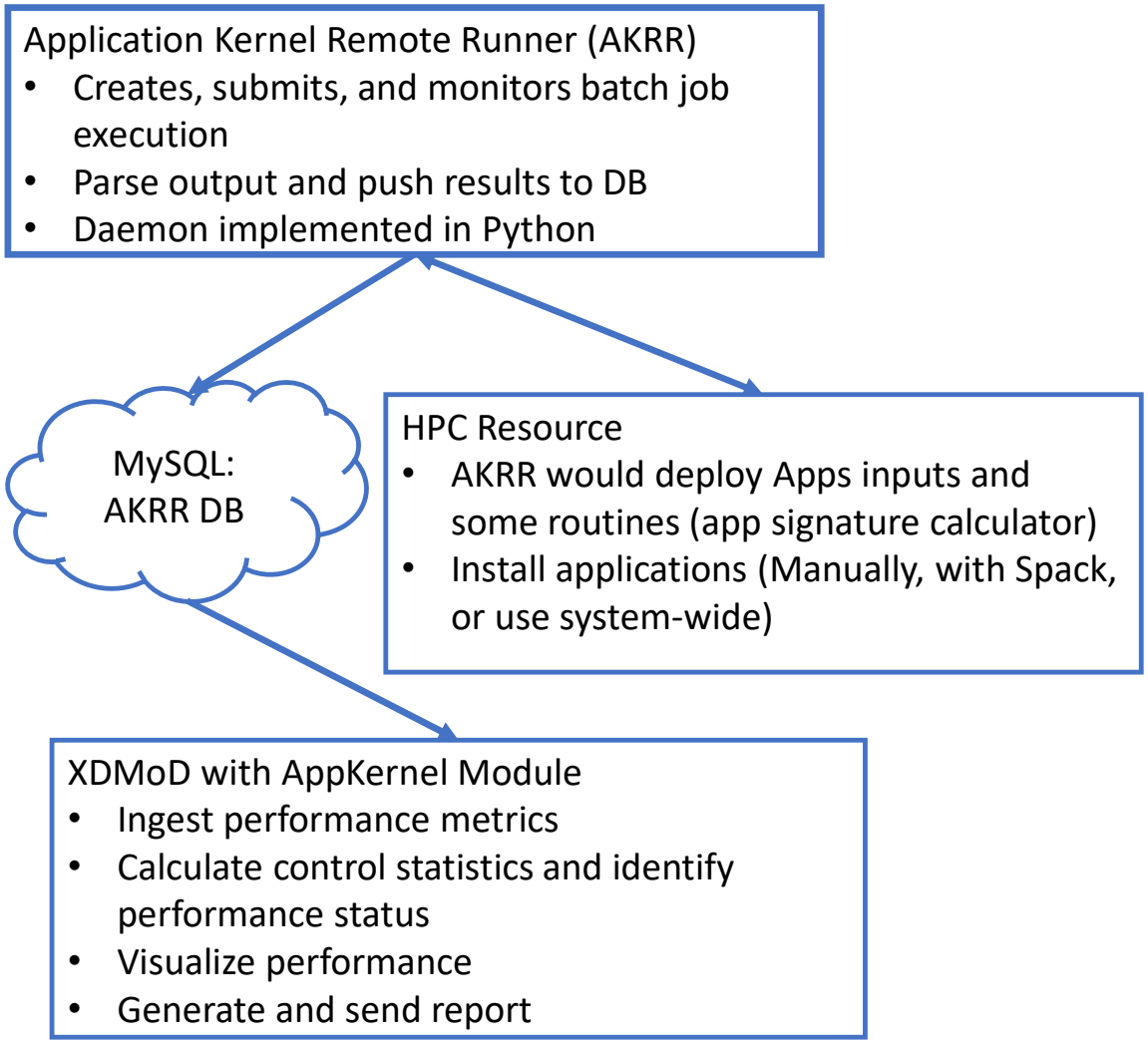


A. UBHPC, CL, 118 Nodes



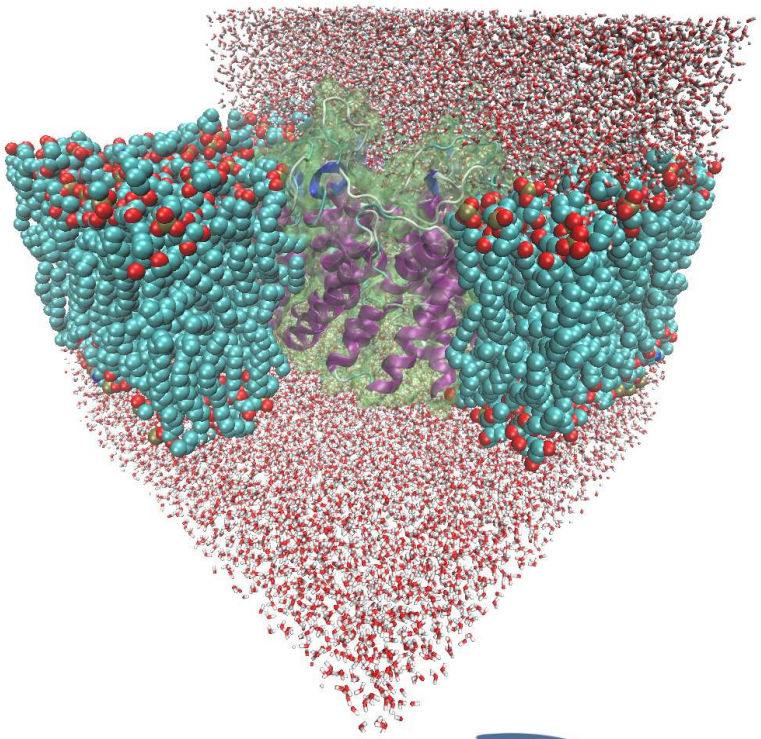
- Visit our poster at PEARC25
- With Slurm Simulator or HPCMod, we will study strategies on how to use these gaps (lower priority, preemption)

Continuous Performance Monitoring with XDMoD



- Task – single execution of benchmark or application
  - Set to start at a specific time with desired periodicity (daily, weekly and so on)
- Supported HPC resources
  - Slurm
  - Single Linux machine
  - *PBS/Torque*
  - *OpenStack*
- Applications and Benchmarks
  - **HPCC**
  - **IMB**
  - **IOR/MDTest**
  - Graph500
  - HPCG
  - AI-Benchmark-Alpha
  - NAMD
  - **Gromacs**
  - ***NWChem***
  - ENZO
  - OpenFOAM

GROMACS: Molecular Dynamics of Biomolecular Systems



GROMACS is molecular dynamics simulation of biomolecular systems

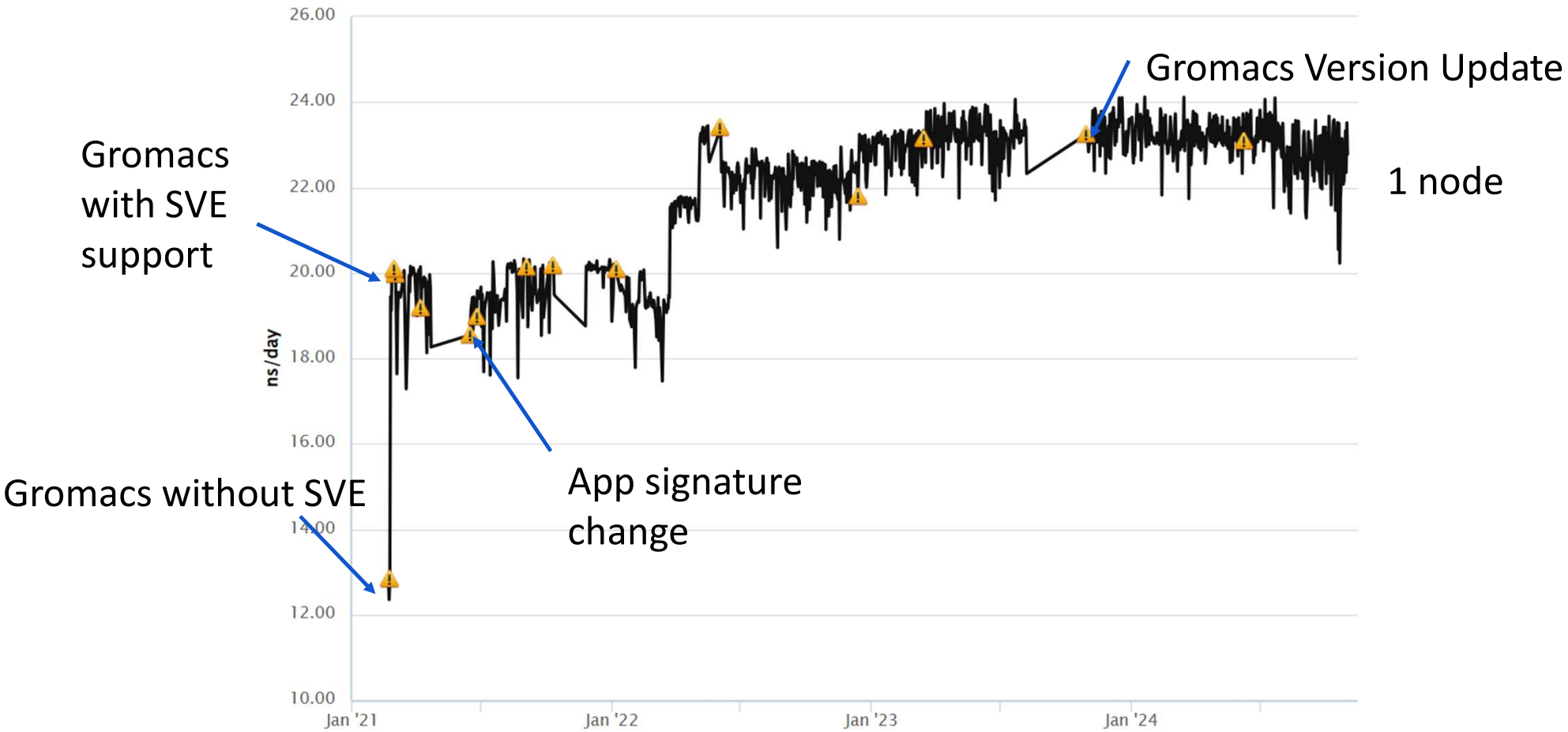
Application computational characteristics:

- Solve ODE (second Newton law)
- Particle interactions
  - Short range/long range
- FFT

Test case:

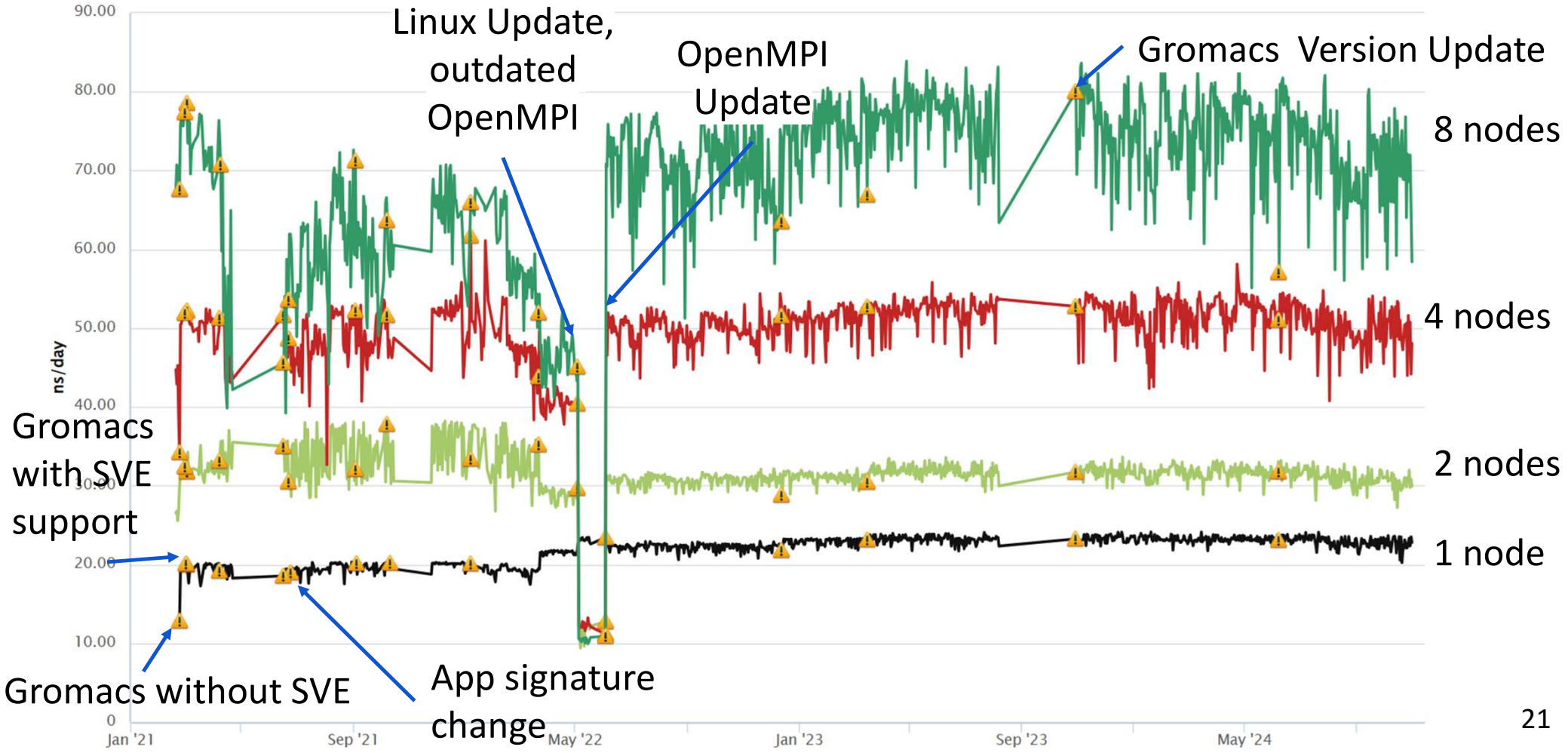
- Membrane protein
- 82k atoms system

Performance over Time: GROMACS



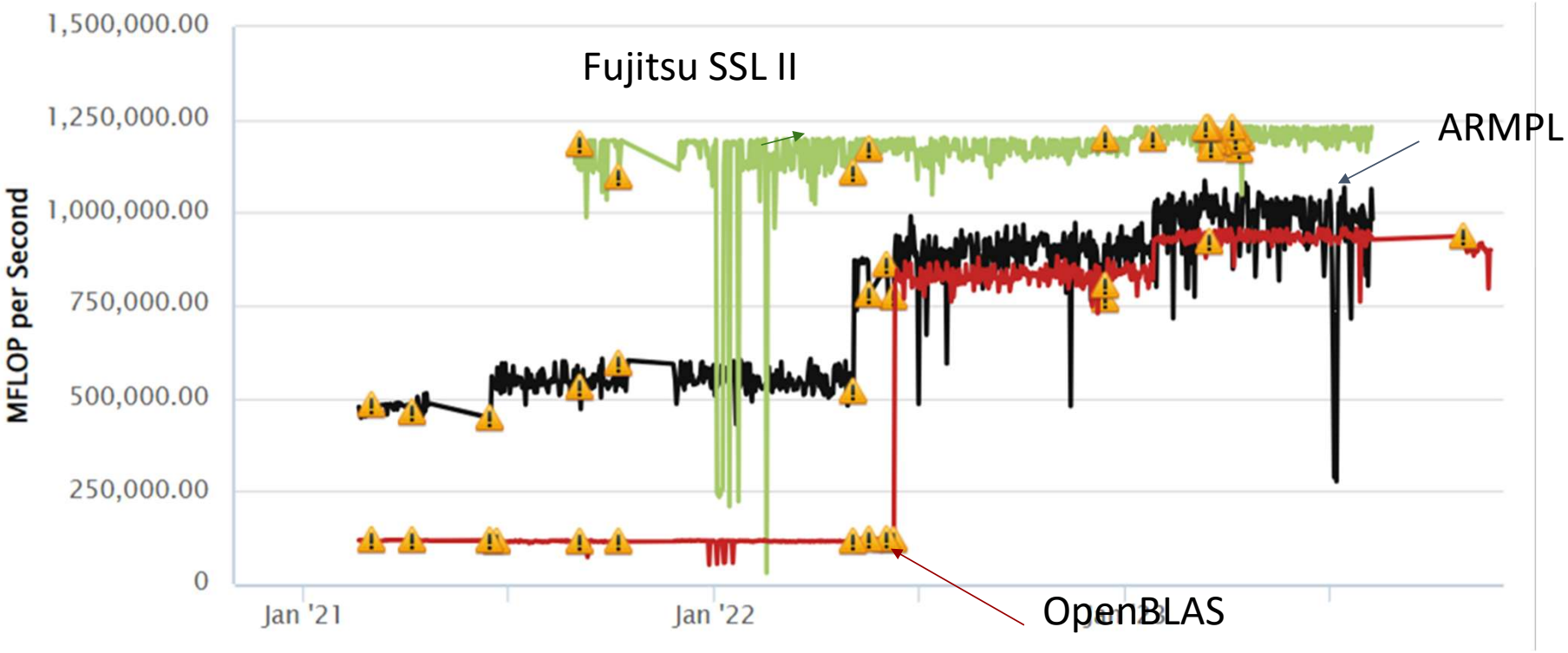
Large improvement on first implementation followed by smaller incremental change

Performance over Time: GROMACS



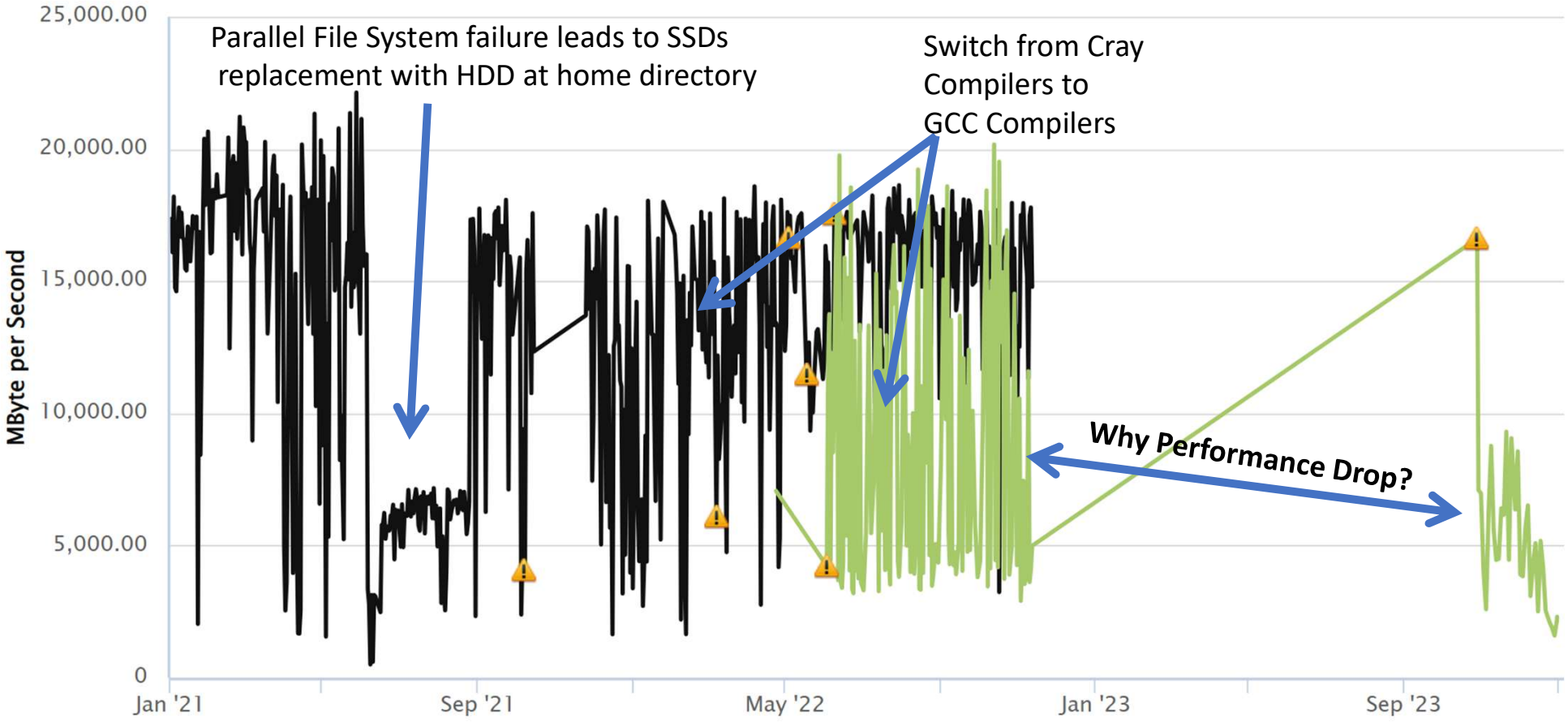
HPCC: Linear Algebra Library Performance

Linpack performance by Fujitsu SSL II, ARMPL and OpenBLAS



Open Source Library Catching up with Proprietary Solutions

IOR: Paral Filesystem Performance



Benchmarking Ookami

HPCC: HPC challenge benchmark

HPC Challenge Benchmark combine multiple benchmarks together

- High Performance LINPACK, which solves a linear system of equations and measures the floating-point performance
- Matrix-matrix multiplication
- Fast Fourier Transform
- Stream: memory bandwidth
- Parallel Matrix Transpose
- MPI Random Access

HPCC: HPC challenge benchmark

CPU/System	Cores	Matrix Multiplication			LINPACK			FFT		
		GFLOPS	GFLOPS/Core		GFLOPS		GFLOPS /Core	GFLOPS		GFLOPS/ Core
ARM Fujitsu A64FX, SVE 512b (SBU-Ookami, FJ)	48	1978	41.2	± 0.2	1177	± 19	24.5	24.4	± 0.9	0.51
ARM Amazon Graviton 3, Neoverse V1, SVE 256b (AWS)	64	1158	18.1	± 0.0	965	± 1	15.1	71.0	± 0.7	1.11
x86 AMD EPYC 7643 Zen3(Milan), AVX2 (SBU)	96	2775	28.9	± 0.9	1493	± 16	15.6	42.6	± 1.0	0.44
x86 AMD EPYC 7763 Zen3(Milan), AVX2 (Purdue Anvil)	128	3046	23.8	± 1.6	2176	± 100	17.0	54.7	± 4.8	0.43
x86 Intel Xeon Gold 6148, Skylake-X, AVX512 (SBU)	40	1559	39.0	± 8.1	981.22	± 109	24.5	33.4	± 2.4	0.84
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2)	48	2122	44.2	± 1.7	1158	± 34	24.1	35.8	± 1.9	0.75
x86 Intel Xeon Plat. 8380, Ice Lake, AVX512 (TACC-Stampede 2)	80	3824	47.8	± 0.6	1713	± 5	21.4	76.4	± 2.0	0.96
x86 Intel Xeon Max 9468, Sapphire Rapids, DDR mode (SBU)	96	4787	49.9	± 2.7	2211	± 182	23.0	129.0	± 15.1	1.34
x86 Intel Xeon Max 9468, Sapphire Rapids, HBM mode (SBU)	96	5392	56.2	± 4.2	2862	± 36	29.8	143.1	± 24.4	1.49
NVIDIA Grace CPU Superchip ES, ARMPL	144	4089	28.4	± 0.1	3124	± 12	21.7	5.5	± 0.1	0.04
NVIDIA Grace CPU Superchip ES, OpenBLAS, FFTW	144	4461	31.0	± 0.1	3120	± 15	21.7	134.2	± 1.7	0.93

- **In Matrix multiplication and LINPACK wider SIMD has higher performance**
- **In Matrix multiplication and LINPACK wider SIMD NVIDIA Grace performed similar or better to AMD Millan in per core performance. Adding high core counts lead to higher per node performance in LINPACK**
- **For FFT per core performance of Grace is similar to Skylake-X and per node is between different memory modes for Sapphire Rapids**

HPCG: The High-Performance Conjugate Gradients

- The High-Performance Conjugate Gradients (HPCG) benchmark is an alternative to the HPL benchmark (used in HPCC) and utilizes methods and patterns commonly used in many PDE solvers
- Unlike HPCC, HPCG does not rely on external libraries but requires vendors to optimize their own version of HPCG.
- Thus for x86 machines, we used the Intel version of HPCG, for the A64FX Cray version and for AMD the reference version.

HPCG: The High-Performance Conjugate Gradients

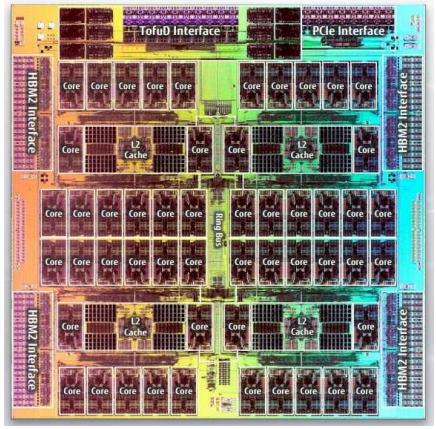
CPU/System	Cores	HPCG Version	HPCG		
			GFLOPS		GFLOPS/ Core
ARM Fujitsu A64FX, SVE 512b	48	Cray	64.4	± 2.8	1.34
x86 Intel Xeon Gold 6148, Skylake-X, AVX512 (SBU)	40	Intel	36.4	± 0.3	0.91
x86 AMD EPYC 7643 Zen3(Milan), AVX2 (SBU)	96	Intel	53.0	± 2.0	0.55
x86 Intel Xeon Max 9468, Sapphire Rapids, DDR mode (SBU)	96	Intel	83.6	± 1.1	0.87
x86 Intel Xeon Max 9468, Sapphire Rapids, HBM mode (SBU)	96	Intel	197.5	± 2.1	2.06
NVIDIA Grace CPU Superchip ES	144	Unoptimized	106.5	± 0.1	0.74

- **Even unoptimized, Grace show higher performance per core than AMD Millan and per node performance is higher than Sapphire Rapids in DDR mode**

GROMACS: Molecular Dynamics of Biomolecular Systems

System	Cores	Simulation Speed, ns/day	Simulation Speed per Core, ns/day/core	Power, W	Energy Efficiency, ns/kWh
CPU Only Calculation					
ARM Futjitsu A64FX, SVE 512bit (SBU-Ookami, Fujitsu)	48	22.8 ± 0.3	0.48	105 ± 5	9.1 ± 0.4
ARM Cavium ThunderX2 (SBU-Ookami)	64	28.8 ± 4.2	0.45		
ARM Amazon Graviton 2, Neoverse N1 (AWS)	48	37.8 ± 0.1	0.79		
ARM Amazon Graviton 3, Neoverse V1, SVE 256bit (AWS)	64	71.4 ± 1.0	1.12		
ARM Ampere Altra, Neoverse N1 (Azure)	64	56.5 ± 0.6	0.88		
ARM Ampere One A192-32X, Neoverse N1 (Ampere)	192	172.1 ± 2.3	0.90	512 ± 5	14.0 ± 0.1
ARM NVIDIA Grace, Neoverse V2, SVE 128bit (SBU)	144	235.2 ± 0.4	1.63	709 ± 41	18.9 ± 0.8
x86 AMD EPYC 7742 Zen2(Rome), AVX2 (PSC Bridges-2)	128	109.6 ± 4.8	0.86		
x86 AMD EPYC 7763 Zen3(Milan), AVX2 (Purdue Anvil)	128	169.9 ± 4.4	1.33		
x86 Intel Xeon Plat. 8160, Skylake-X, AVX512 (TACC-Stampede 2)	48	70.4 ± 0.8	1.47		
x86 Intel Xeon Plat. 8380, Ice Lake, AVX512 (TACC-Stampede 2)	80	133.3 ± 6.0	1.67		
x86 Intel Xeon Gold 6130, Skylake-X, AVX512 (UBHPC)	32	39.3 ± 0.9	1.23	367 ± 35	4.5 ± 0.5
x86 Intel Xeon Gold 6330, Ice Lake, AVX512 (UBHPC)	56	103.0 ± 2.0	1.84	619 ± 17	6.9 ± 0.2
x86 Intel Xeon Max 9468, Sapphire Rapids, AVX512 (SBU)	96	193.08 ± 2.3	2.01	820 ± 7	9.8 ± 0.1
CPU-GPU Calculations					
x86 Intel Xeon Gold 6130, NVIDIA V100x2 (UBHPC)	32	145.1 ± 2.8		435 ± 7	13.9 ± 0.3
x86 Intel Xeon Gold 6330, NVIDIA A100x2 (UBHPC)	56	236.5 ± 10.8		707 ± 9	13.9 ± 0.8
AMD Ryzen 9 7950X (16 Cores Used)/NVIDIA RTX 4090	16	284.82			
NVIDIA Grace Hopper Superchip ES	72	429			

Conclusions



- We have monitored the utilization and performance change for over four years
- The team has gained significant insights into tracking and gathering meaningful metrics to assess the impact of the HPC testbed
- Ookami was overall a very successful project
- Utilization monitoring helps us to better understand the user's activity on the system
- Continuous performance monitoring allows us to see the performance improvements of applications as the new technology gains higher adoption

- Visit our Ookami Talk at ISC25 MODA Workshop

XDMoD
METRICS ON DEMAND

ACCESS Metrics

NSF

Acknowledgements

NSF OAC Awards: 1927880 and 2137603

- This work is supported by the National Science Foundation under award 1927880 and 2137603.

- Access Ookami through ACCESS-CI

XDMoD
METRICS ON DEMAND

Stony Brook University

iACS
INSTITUTE FOR ADVANCED COMPUTATIONAL SCIENCE

University at Buffalo
Center for Computational Research