



OOKAMI

Initial experiences with the Ookami A64FX testbed

Andrew Burford¹, Alan C. Calder¹, David Carlson¹, Barbara Chapman¹, Firat Coşkun¹, Tony Curtis¹, Catherine Feldman¹, Robert J. Harrison¹, Yan Kang¹, Benjamin Michalowicz¹, Eric Raut¹, Eva Siegmann¹, Daniel G. Wood¹, Robert L. DeLeon², Mathew Jones², Nikolay A. Simakov², Joseph P. White², Dossay Oryspayev³

¹Institute for Advanced Computational Science, USA

²Center for Computational Research, USA

³Brookhaven National Laboratory, USA



Ookami - 狼



OOKAMI

- Ookami is Japanese for wolf
- A computer technology testbed supported by NSF
- Available for researchers worldwide
(excluding ITAR prohibited countries & restricted parties on the EAR entity list)
- Usage is free for non-commercial and limited commercial purposes

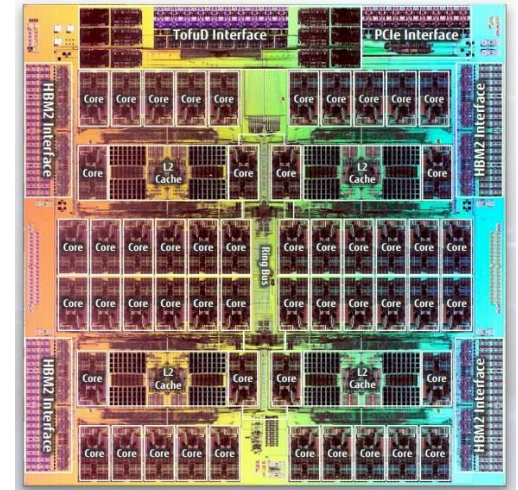


What is Ookami



OOKAMI

- 174 1.8Ghz **A64FX** compute nodes each with 32GB of high-bandwidth memory and a 512 GB SSD
 - Same as in currently fastest machine worldwide, Fugaku
 - First deployment outside Japan
 - HPE/Cray Apollo 80
- Ookami also includes:
 - 1 node with dual socket AMD Rome (128 cores) with 512 Gbyte memory
 - 2 nodes with dual socket Thunder X2 (64 cores) each with 256 Gbyte memory and 2 NVIDIA V100 GPU
 - Intel Sky Lake Processors (32 cores) with 192 Gbyte memory
- Delivers ~ 1.5M node hours per year



Fugaku #1

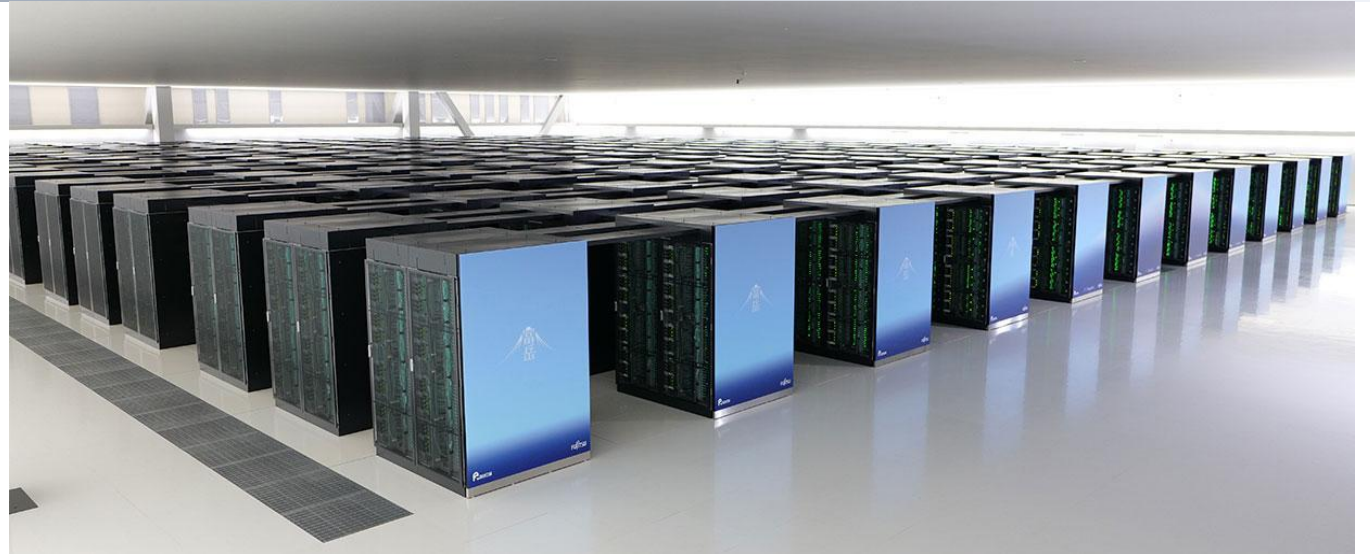
Fastest computer in the world



OOKAMI

First machine to be fastest in
all 5 major benchmarks:

- Green-500
- Top-500 – 415 PFLOP/s in double precision – nearly 3x Summit!
- HPCG
- HPL-AI
- Graph-500



- 432 racks
- 158,976 nodes
- 7,630,848 cores
- 440 PF/s dp (880 sp; 1,760 hp)
- 32 Gbyte memory per node
- 1 Tbyte/s memory bandwidth/node
- Tofu-2 interconnect

<https://www.r-ccs.riken.jp/en/fugaku>

A64fx at a Glance



- ARM V8 64-bit
- 512-bit SVE
- 48 compute cores
- 4 NUMA regions
- 32 (4x8) GB HBM @ 1 TB/s
- PCIe 3 (+ Tofu-3) network



A64fx NUMA Node Architecture



OOKAMI

- Supports high calculation performance and low power consumption
- Supports Scalable Vector Extensions (SVE)
- **4 Core Memory Groups (CMGs)**
 - 12 cores (13 in the FX1000)
 - 64KB L1\$ per core
 - 256b cache line
 - 8MB L2\$ shared between all cores
 - 256b cache line
 - Zero L3\$
 - 8 GB HBM at 256GB/s

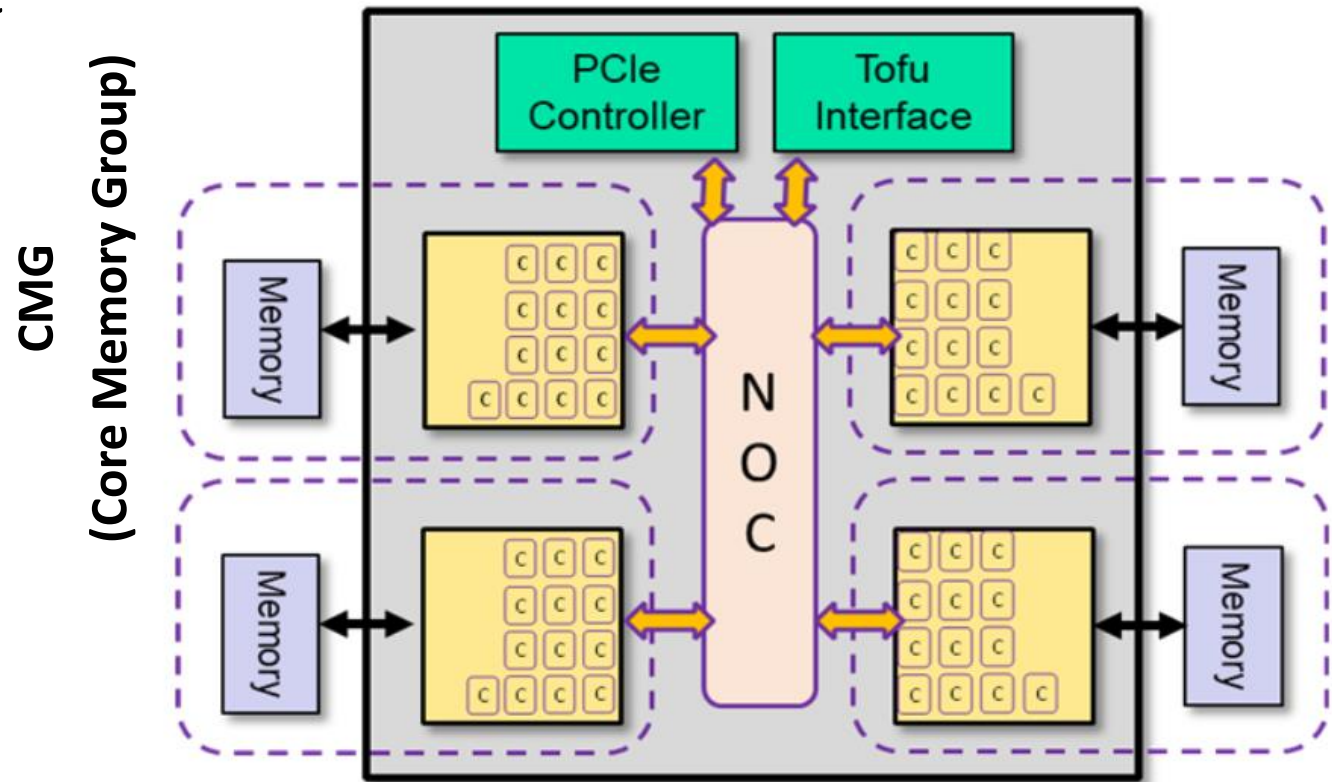


Diagram is the „1000“ chip.
We have „700“ chip, i.e. no assistant cores and no Tofu interface

SVE (Scalable Vector Extensions)



OOKAMI

- Enables Vector Length Agnostic (VLA) programming
 - VLA enables portability, scalability, and optimization
 - The actual vector length is set by the CPU architect
 - Any multiple of 128 bits up to 2048 bits
 - May be dynamically reduced by the OS or hypervisor
- Predicate-centric architecture
- SVE was designed for HPC and can vectorize complex structures
 - Gather-load and scatter-store; horizontal reductions
 - SVE begins to tackle traditional barriers to auto-vectorization
- Support from open source and commercial tools

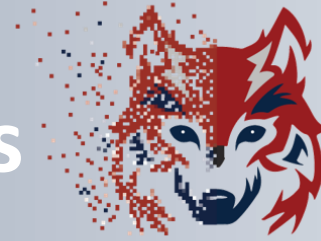
	1	2	3	4
+	5	5	5	5
<i>pred</i>	1	0	1	0
=	6	2	8	4

```
for (i = 0; i < n; ++i)
INDEX i
CMPLT n
```

	n-2	n-1	n	n+1
	1	1	0	0

	1	2		
+	1	2	0	0
<i>pred</i>	1	1	0	0

Memory Statistics of Typical Jobs

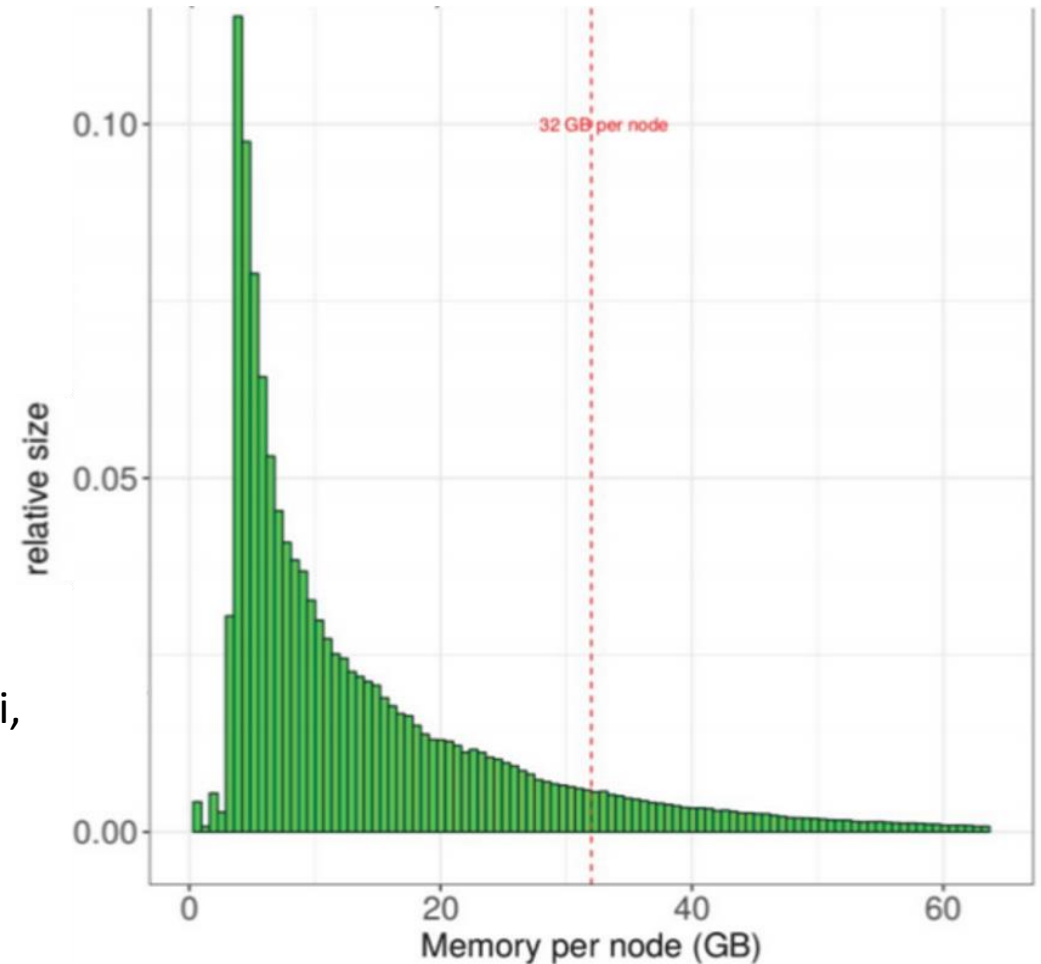


OOKAMI

2017 analysis of XSEDE workload revealed
86% of all jobs need less than 32 GB / node

These 86% of jobs correspond to 85% of the
total XSEDE cpu-hour usage

Simakov, White, DeLeon, Gallo, Jones, Palmer, Plessinger, Furlani,
"A Workload Analysis of NSF's Innovative HPC Resources Using
XDMoD," arXiv:1801.04306v1 [cs.DC], 12 Jan 2018





OOKAMI

“Programmability of a CPU, performance of a GPU”

Satoshi Matsuoka (Head of RIKEN, home of Fugaku)



- Blazing fast memory
- Easily accessed performance
- New technology path to exascale

What else



OOKAMI

- CentOS 8 operating system
- DUO Authentication
- High-performance Lustre file system (~800TB of storage)
- Slurm workload manager
- Compilers: GNU, Arm, Cray, Nvidia, Fujitsu (soon)
- Continuous growing stack of preinstalled software
 - MPI implementations
 - Toolchains
 - Math libraries
 - Performance analysis & debugging:
(arm Forge, Cray, GNU, TAU, ..)

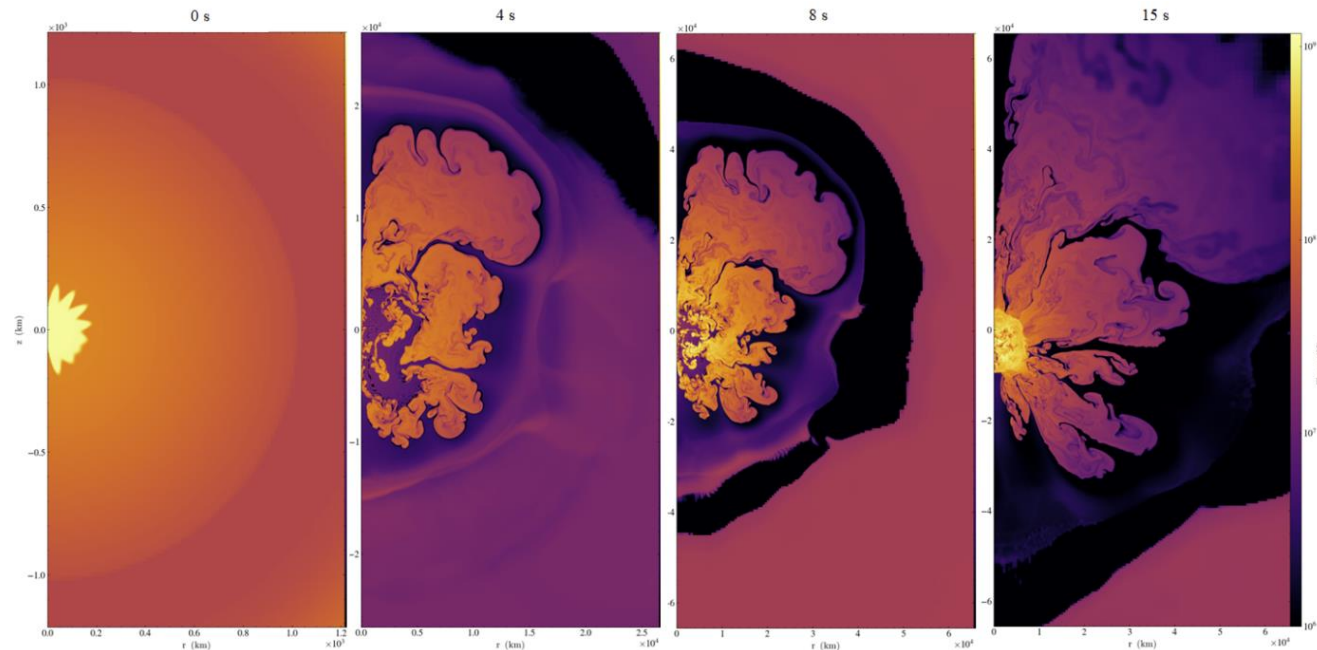
```
----- /cm/local/modulefiles -----
cluster-tools/9.0  gcc/9.2.0      null          shared
cmd               ipmitool/1.8.18  openldap     slurm/slurm/19.05.7
cmjob            lua/5.3.5      openmpi/mlnx/gcc/64/4.0.3rc4
dot              module-git      python3
freeipmi/1.6.4   module-info     python37
----- /cm/shared/modulefiles -----
cm-pmix3/3.1.4  hdf5/1.10.1  hwloc/1.11.11  ucx/1.6.1
----- /lustre/shared/modulefiles -----
anaconda/3      gnuplot/5.4.1  ncurses/6.2
archiconda/3    go/1.16.3      ncurses/arm/gcc/6.2
arm-modules/20  htop/3.0.2     ninja/1.10.2
arm-modules/21  hwloc/2.4.1    nvidia/nvhpc-byo-compiler/21.3
cmake/3.19.0    intel/compiler/64/2020/20.0.2  nvidia/nvhpc-nompi/21.3
CPE-nosve/20.10  intel/mkl/64/2020/20.0.2  nvidia/nvhpc/21.3
CPE-nosve/21.03  intel/mpi/64.2020/20.0.2  openblas/0.3.10
CPE/20.10       intel/tbb/64/2020/20.0.2  openmpi/arm21/4.1.0
CPE/21.03       internal/template  openmpi/gcc8/4.1.0
cuda/toolkit/11.2  julia/1.6.0    openmpi/gcc10/4.1.0
curl/7.73.0     lapack/3.9.0   openssl/1.1.1h
doxygen/1.8.20  libfabric/1.12.1  p7zip/16.02
gcc-10.3.0-openacc  libgd/gcc/2.3.1  pax-utils/1.2.9
gcc/10.2.0      libpng/gcc/1.6.37  tau/2
gcc/10.3.0     likwid/5.1.1    ucx/1.10.0
gcc/11.1.0     mvapich2/arm21/2.3.5  util-linux/2.37
git/2.29        mvapich2/gcc8/2.3.5  xpmem/2.6.3
gnuplot/5.4.0  mvapich2/gcc10/2.3.5  zsh/5.8
```

Initial Experiences



OOKAMI

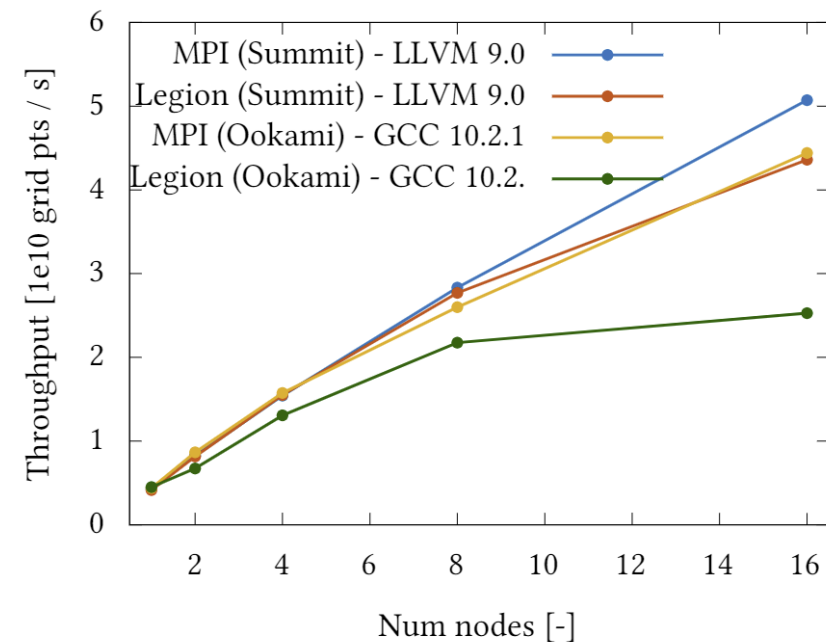
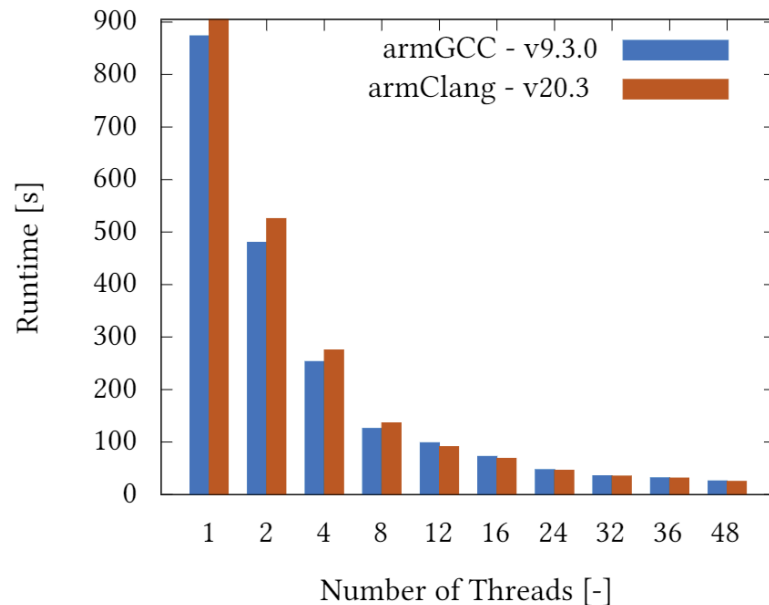
- Most applications run out of the box
- Obtaining high performance is more complex



Minimod



- seismic modeling mini-app developed by Total
- extracts the stencil computation from a production seismic imaging application
- stencil is used to numerically solve the acoustic wave equation
- benchmark to test new and emerging hardware and programming models for geophysics applications

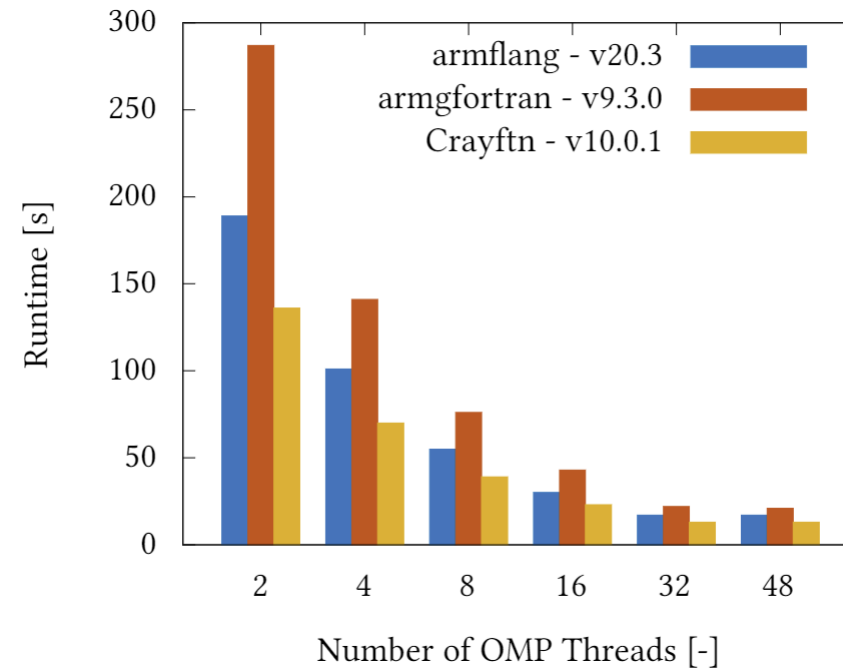


SWIM



OOKAMI

- Part of the SPEC CPU2000 Benchmark suite
- weather forecasting benchmark (FORTRAN OpenMP)
- solves the shallow-water equations using finite differences

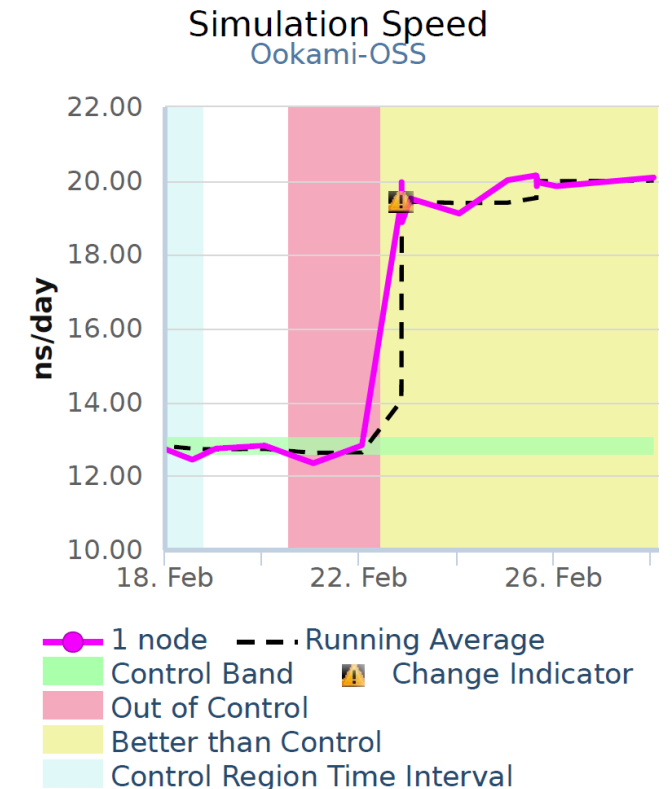
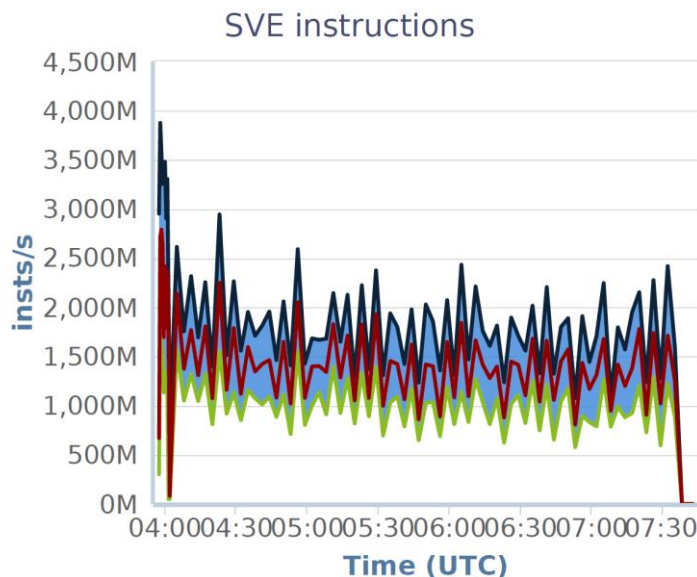


XDMoD



OOKAMI

- Ookami is monitored with XDMoD
- XDMoD software modify to monitor A64FX-specific metrics
- application kernels are used to proactively monitor HPC resource performance by daily benchmarks
- goal is to see how the performance of benchmarks and real applications change as the compiler toolchains improve



Getting Accounts



OOKAMI

- Submit a project request (templates on our website)
 - **Testbed:**
 - Porting and tuning software
 - Benchmarking
 - Limited production calculations to demonstrate capability
 - Significantly less than 15,000 node hours per year
 - First two project years
 - **Production:**
 - Less than 150K node hours per year
 - Lower priority during the first two project years
- **Requests must include:**
Title, date, PI, usage description, computational resources, grant number (if funded)

Getting Accounts



OOKAMI

- Getting access:
 - Create a project request and submit it through ticketing system:
<https://iacs.supportsystem.com/>
 - Requests will be reviewed & published
 - If you are not affiliated to SBU: Fill a volunteer demographic form

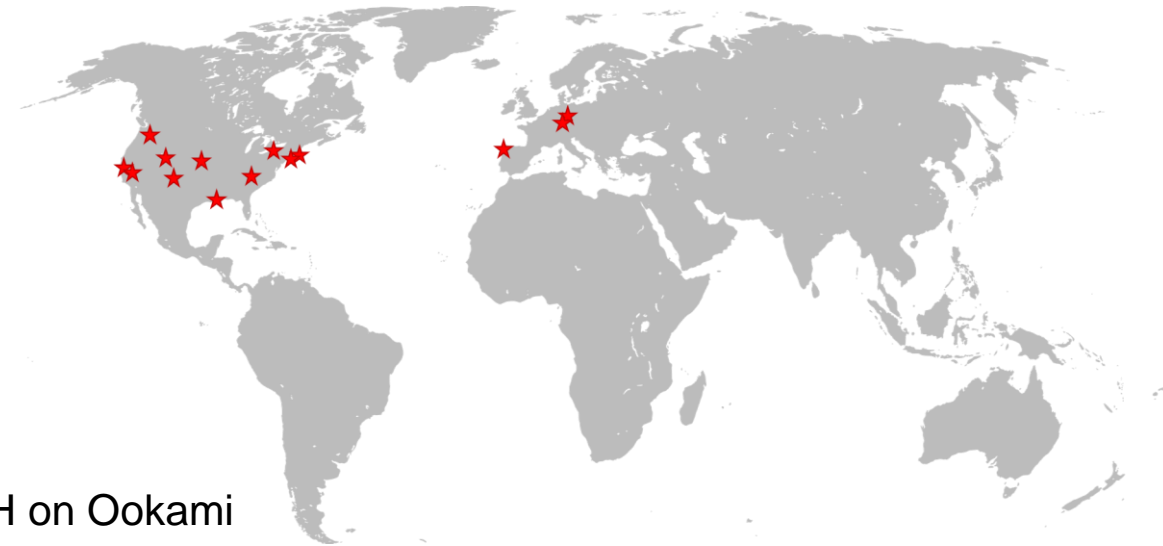
<https://www.stonybrook.edu/ookami/>

Current Status




OOKAMI

- ~ 30 testbed projects (USA & Europe)
- ~ 100 users
- Several trainings & webinars
- Talks about Ookami in this session:
 - Lessons Learned: An In-depth Look at Running FLASH on Ookami
Alan C. Calder, Catherine Feldman, and Benjamin Michalowicz
 - Performance Engineering using SVE
Robert J. Harrison



Get in Contact



- <https://www.stonybrook.edu/ookami/>
- Bi-weekly Hackathon
 - Tue 10am – noon EST
 - Thu 2pm – 4pm EST
- Slack Channel for users #OOKAMI 

Acknowledgement:

- The whole Ookami team
- NSF (grant grant OAC 1927880)

Eva Siegmann

Lead Research Scientist

eva.siegmann@stonybrook.edu

