Ookami Webinar, 04/28/2021

Ookami - 狼

- Ookami is Japanese for wolf
  - Homage to the origin of the processor and the Stony Brook mascot
- 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 **A64FX** compute nodes each with 32GB of high-bandwidth memory and a 512 Gbyte 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

Benefits for users

OOKAMI

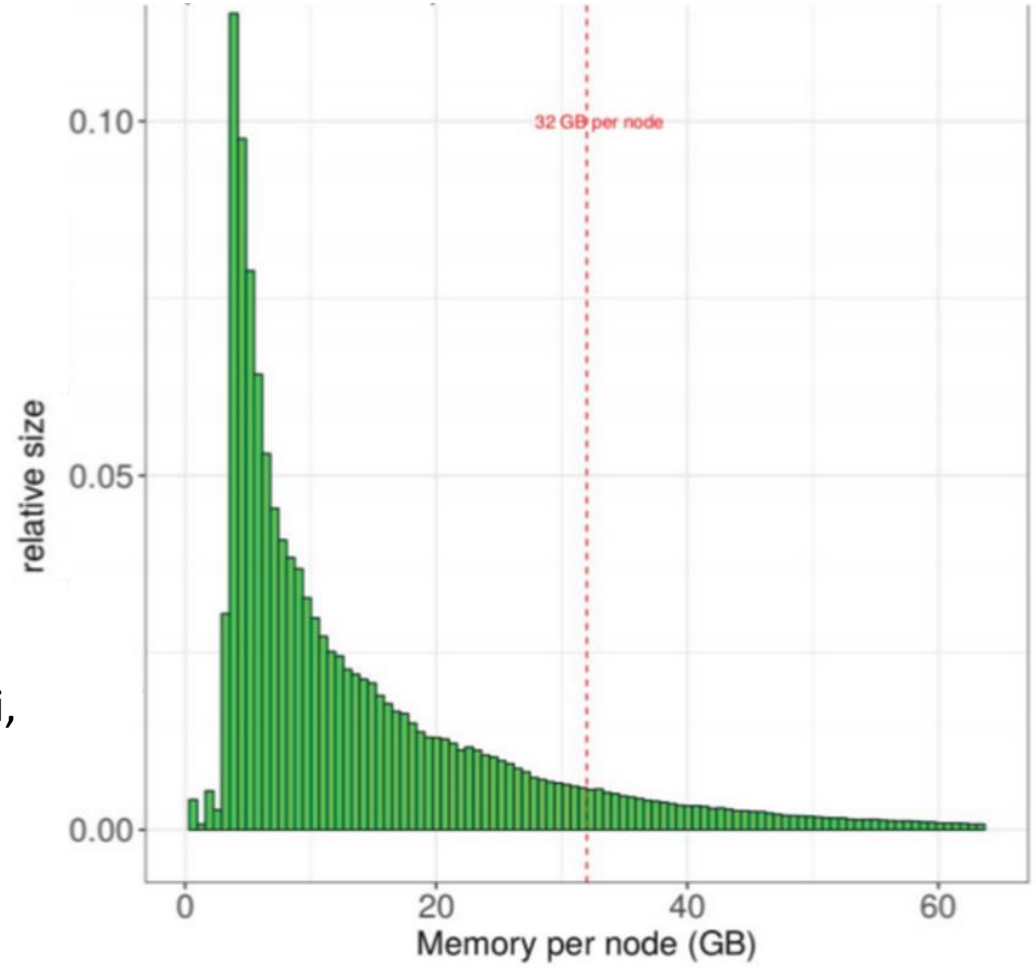
- Access and evaluate state-of-the-art computing technology
- Conduct your own research on newest processors
- Port, tune, and optimize your code in preparation for a new generation of supercomputers
- Secure environment with system maintenance done by the Ookami team

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

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

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

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

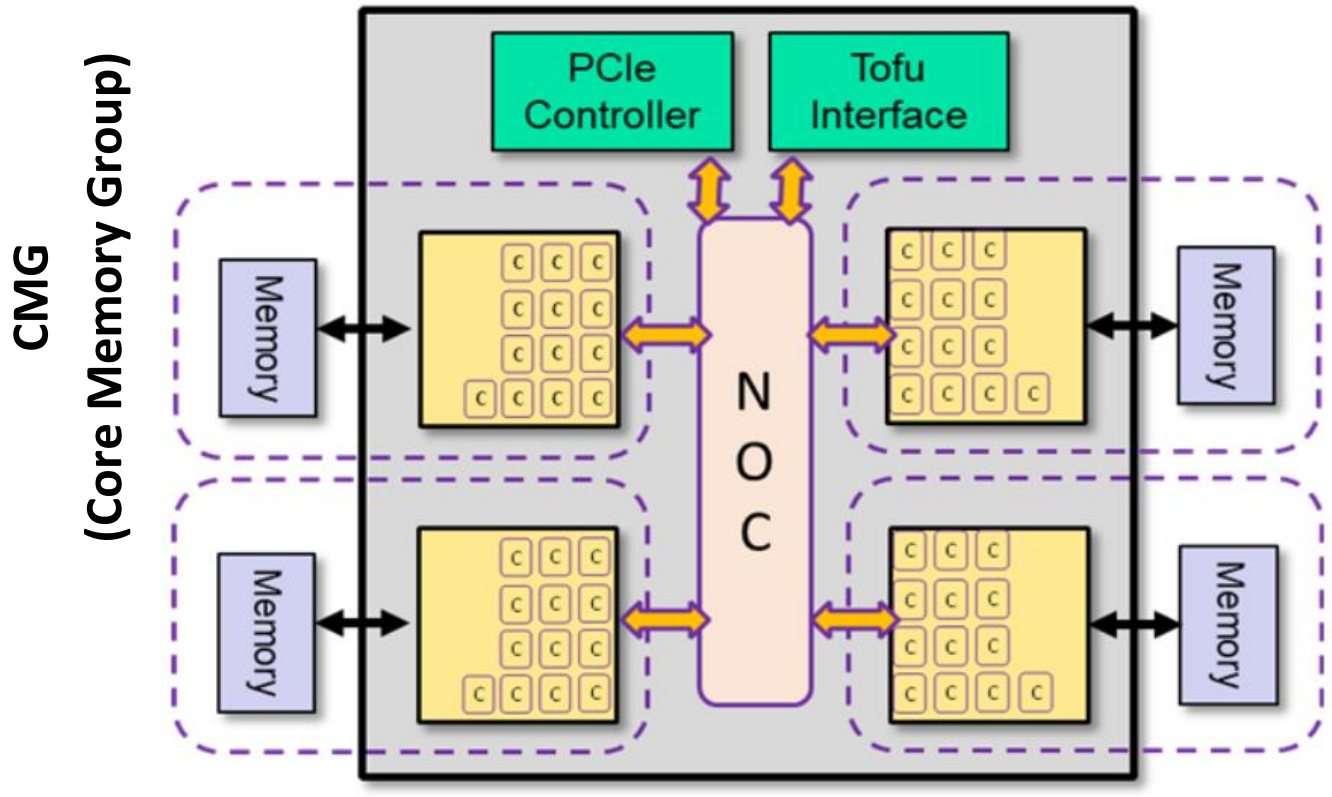
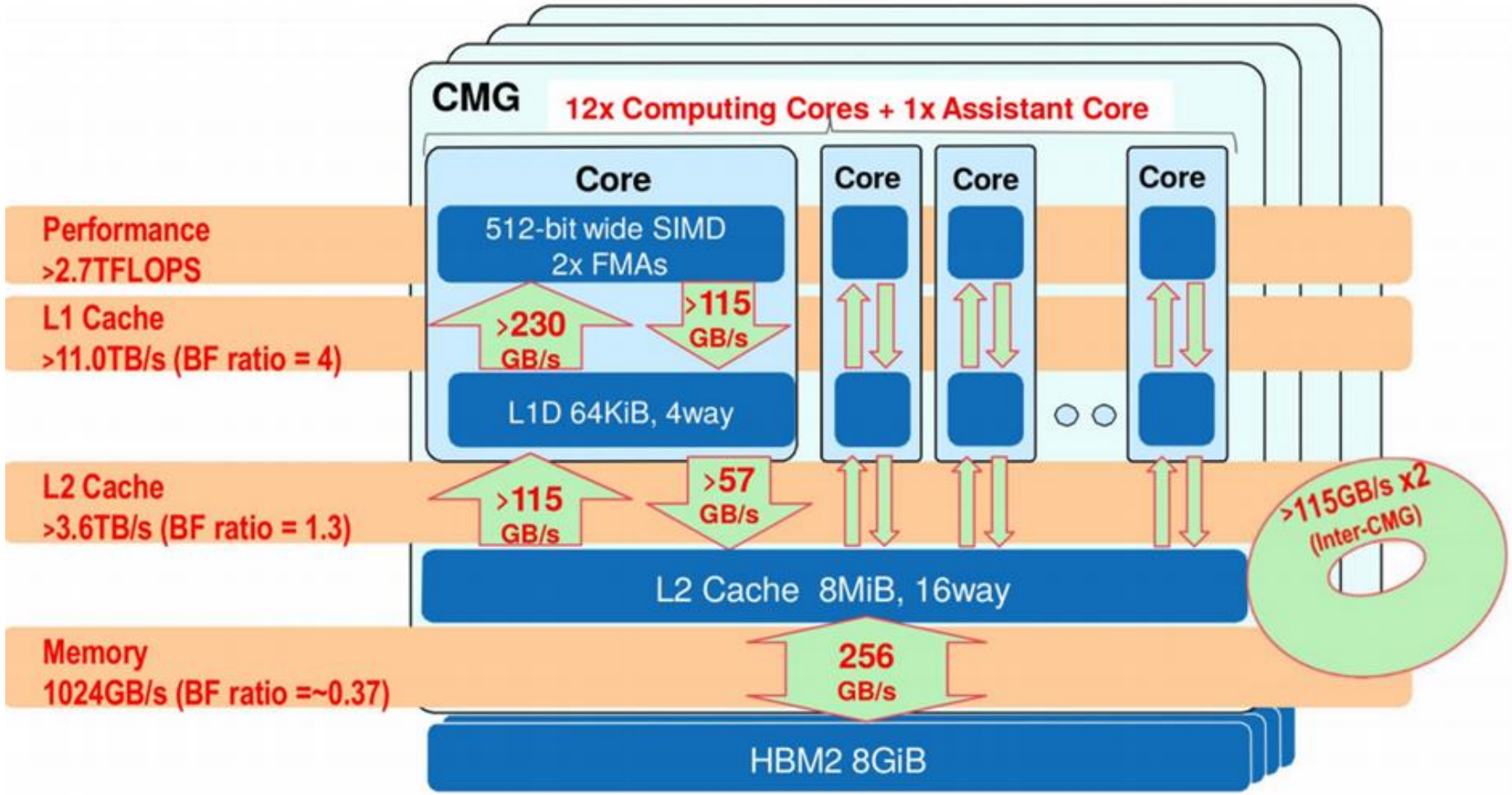


Diagram is the „1000“ chip.
We have „700“ chip, i.e. no assistant cores and no Tofu interface

http://www.jicfus.jp/jp/wp-content/uploads/2018/11/msato-190109.pdf

A64fx Core Memory Group

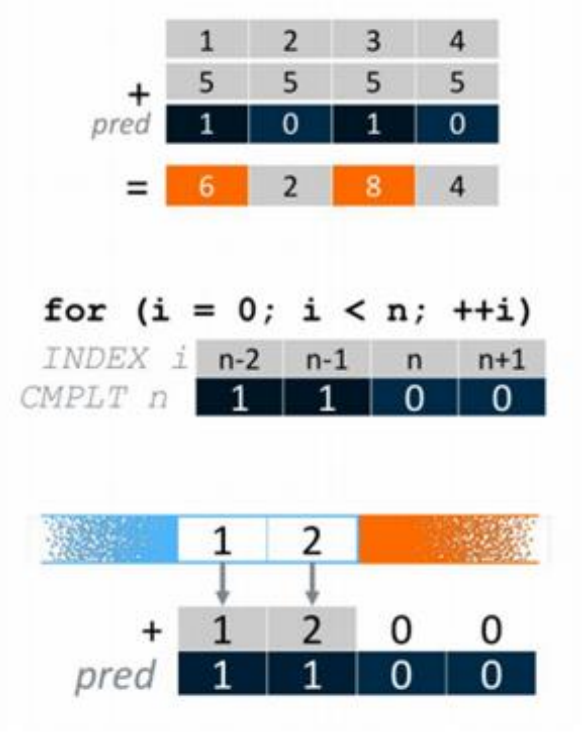


http://www.jicfus.jp/jp/wp-content/uploads/2018/11/msato-190109.pdf

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
  - Predicates are central, not an afterthought
  - Support complex nested conditions and loops
  - Predicate generation also sets condition flags
  - Reduces vector loop management overhead
- 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
    - Software-managed speculative vectorization allows uncounted loops to be vectorized.
    - In-vector serialized inner loop permits outer loop vectorization in spite of dependencies.
- Support from open source and commercial tools



SVE vs Traditional ISA

How do we compute data which has ten chunks of 4-bytes?

OOKAMI

Aarch64 (scalar)

- Ten iterations over a 4-byte register

NEON (128-bit vector engine)

- Two iterations over a 16-byte register + two iterations of a drain loop over a 4-byte register

SVE (128-bit VLA vector engine)

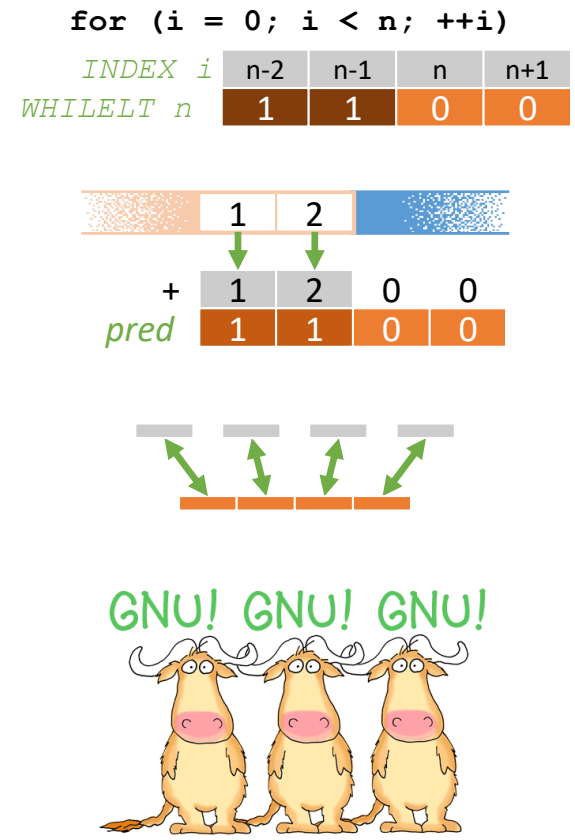
- Three iterations over a 16-byte **VLA register** with an adjustable **predicate**

arm

Summarized

OOKAMI

- SVE enables Vector Length Agnostic (VLA) programming
- VLA enables portability, scalability, and optimization
- Predicates control which operations affect which vector lanes
  - Predicates are not bitmasks
  - You can think of them as dynamically resizing the vector registers
- 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
- SVE was designed for HPC and can vectorize complex structures
- Many open source and commercial tools currently support SVE



arm

What else

OOKAMI

- CentOS 8 operating system
- DUO Authentication
- High-performance Lustre file system (~800TB of storage)
  - 30GB home directory
  - 30TB scratch (cleared every 14days)
  - Project directories on request up to 8TB (temporary larger directories are possible)
- Slurm workload manager

Partition	Time Limit	Min Nodes	Max Nodes
short	4h	1	32
large	8h	24	80
long	2d	1	8
extended	7d	1	2

What else

- Compilers: GNU, Arm, Cray, Nvidia, Fujitsu (soon)
- Ticketing system for any kind of issues / questions /requests
- Continuous growing stack of preinstalled software – module environment
  - MPI implementations
  - Toolchains
  - Math libraries
  - …

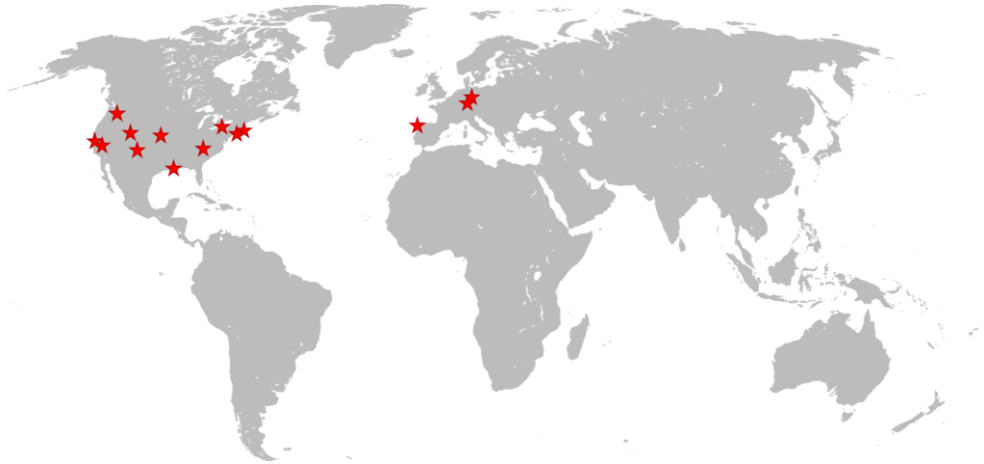
```
------------------------------------------- /cm/local/modulefiles ------------------------------------------
cluster-tools/9.0  freeipmi/1.6.4   module-git   openmpi/mlnx/gcc/64/4.0.3rc4  slurm/slurm/19.05.7
cmd                gcc/9.2.0        module-info  python3                       slurm/slurm/no-version
cmjob              ipmitool/1.8.18  null         python37
dot                lua/5.3.5        openldap     shared

------------------------------------------- /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          gcc/10.2.0                    julia/1.6.0                    nvidia/nvhpc/21.3
archiconda/3        gcc/10.3.0                    lapack/3.9.0                   openblas/0.3.10
arm-modules/20      gcc/11-git                    libgd/gcc/2.3.1                openmpi/arm21/4.1.0
arm-modules/21      git/2.29                      libpng/gcc/1.6.37              openmpi/gcc8/4.1.0
cmake/3.19.0        gnuplot/5.4.0                 likwid/5.1.1                   openmpi/gcc10/4.1.0
CPE-nosve/20.10     gnuplot/5.4.1                 mvapich2/arm21/2.3.5           openssl/1.1.1h
CPE-nosve/21.03     htop/3.0.2                    mvapich2/gcc8/2.3.5            p7zip/16.02
CPE/20.10           hwloc/2.4.1                   mvapich2/gcc10/2.3.5           pax-utils/1.2.9
CPE/21.03           intel/compiler/64/2020/20.0.2 ncurses/6.2                    tau/2
cuda/toolkit/11.2   intel/mkl/64/2020/20.0.2      ncurses/arm/gcc/6.2            ucx/1.10.0
curl/7.73.0         intel/mpi/64.2020/20.0.2      ninja/1.10.2                   util-linux/2.37
doxygen/1.8.20      intel/tbb/64/2020/20.0.2      nvidia/nvhpc-byo-compiler/21.3 xpmem/2.6.3
gcc-10.3.0-openacc  internal/template             nvidia/nvhpc-nompi/21.3        zsh/5.8
```

Current Status

OOKAMI



- ~ 30 testbed projects (USA & Europe)
- ~ 100 users
- Trainings & Webinars:
  - SVE Hackathon (February) - Hands-on session with arm
  - XDMOD (March) - Tool for the Comprehensive Management of HPC Resources
  - TAU (April) - profiling and tracing toolkit for performance analysis of parallel programs
  - **Upcoming:**

    Webinar in cooperation with NHR@FAU - runtime environment, performance models, likwid, OSACA, etc.

Getting Accounts

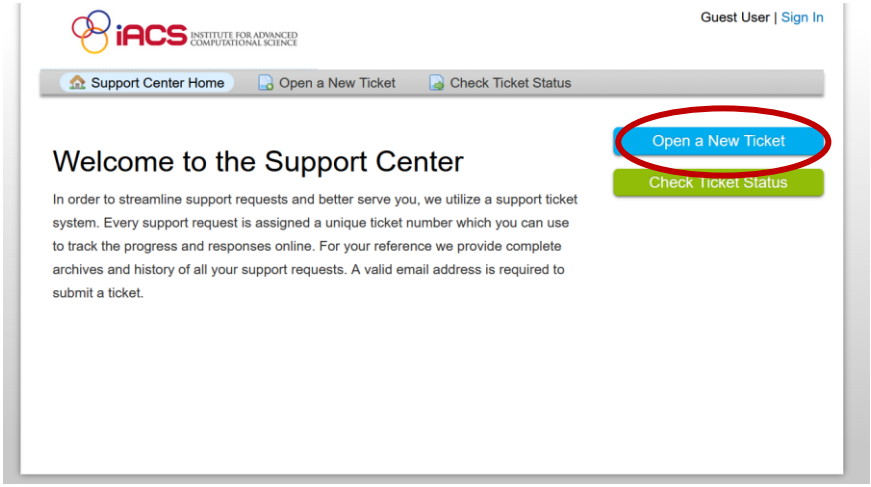
- Submit a project request (templates on our website)
  - **Testbed**:
    - Porting and tuning software
    - Benchmarking
    - Limited production calculations to demonstrate capability
    - Significantly less than 15,000 node hours per year
    - First two project years
  - **Production**:
    - 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
  - All members of a project will get accounts



Get in Contact

OOKAMI

- https://www.stonybrook.edu/ookami/
- Ticketing system: https://iacs.supportsystem.com/
  - Technical questions / issues
  - Project / account requests
- Ookami_computer@stonybrook.edu
  - For general questions
- Bi-weekly Hackathon (Tue 10am – noon, Thu 2 – 4pm)
- Slack Channel #OOKAMI