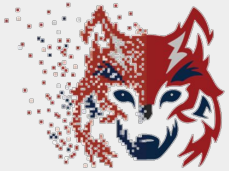


Ookami Webinar

Th 2/29/2024, 2pm EST

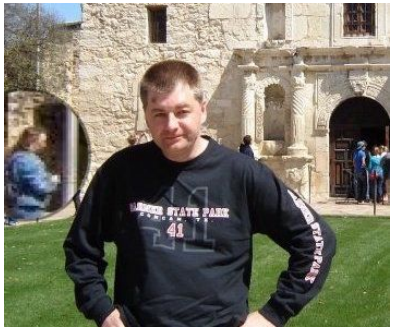
NSF OAC 1927880

Stony Brook University

iACS INSTITUTE FOR ADVANCED COMPUTATIONAL SCIENCE

University at Buffalo
The State University of New York

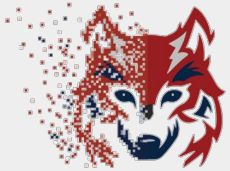
Eva Siegmann
Lead Research Scientist



Tony Curtis
Programming Project Leader

Agenda

- Part I: Ookami overview
- Part II: Get an allocation
- Part III: Getting started on Ookami
- Part IV: Getting the best out of Ookami
- Part V: Upcoming events
- Part VI: Support mechanisms



Part I

Ookami Overview

Target Audience: Researchers interested in using Ookami

Motivation - Fugaku

Fugaku
#1 Fastest computer in the world until 6/2022

First machine to be fastest in all 5 major benchmarks

- Green-500 benchmark
- Top-500 benchmark
- HPCG benchmark
- HPL-AI benchmark
- Graph-500 benchmark

- 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

Ookami

OOKAMI

1.5 mil node hours per year

Node	
Processor	A64FX 700
#Cores	48
Peak DP	2.76 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

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
- Operated by Stony Brook in cooperation with the University at Buffalo

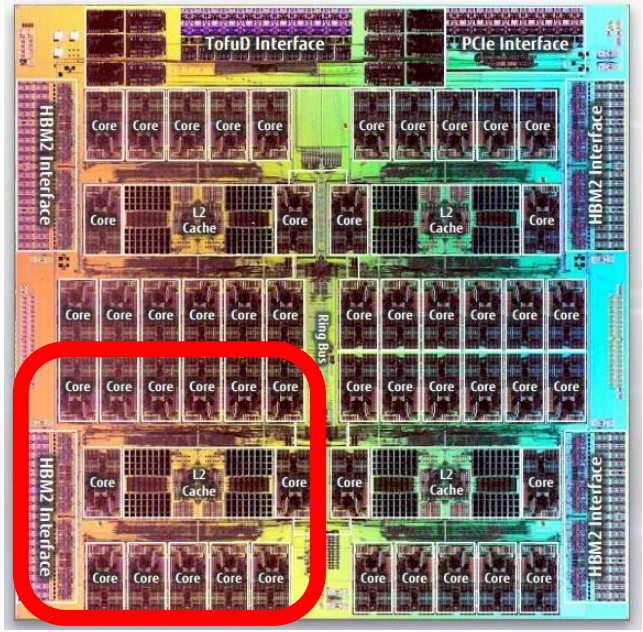
Stony Brook University

iACS INSTITUTE FOR ADVANCED COMPUTATIONAL SCIENCE

University at Buffalo
The State University of New York

A64FX NUMA Node Architecture

- Arm V8-64bit
- Supports high calculation performance and low power consumption
- 32 (4x8) GB HBM @ 1TB/s
- Supports Scalable Vector Extensions (SVE) with 512-bit vector length
- 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$
- PCIe 3 (+ Tofu-3) network



Ookami Environment

- Rocky Linux 8
- Lustre file system providing ~800TB
- Slurm workload manager
- Full bisection bandwidth HDR100
- Module environment
    - > 400 modules
- Various compilers, profilers & debuggers (GCC, Arm, CPE, forge, OSACA, etc.)
- Available via Open OnDemand

Part II

Getting an allocation on Ookami

Target Audience: Researchers interested in using Ookami

Getting an allocation / account

- Ookami is an ACCESS resource provider - https://access-ci.org/

  *ACCESS is a program established and funded by the National Science Foundation to help researchers and educators, with or without supporting grants, to utilize the nation's advanced computing systems and services – at **no cost.***

- ACCESS allocations grant you service units (SUs) which can be exchanged against node hours / core hours / GPU hours / storage on a resource provider

  https://support.access-ci.org/video-learning-center

Getting an allocation / account

CREATE ACCOUNT

SELECT OPPORTUNITY

REQUEST ALLOCATION

RECEIVE CREDITS

EXCHANGE CREDITS

- Create an ACCESS account
- Start with requesting an EXPLORE allocation
  - Title
  - Overview
  - CV (3 page max)
- Reviewing usually takes around one business day
- Receive email regarding the allocation
- Transfer SUs (Service Units) to Ookami - Use the exchange calculator
- We will contact you regarding your account creation
- Optional: add users to your allocation

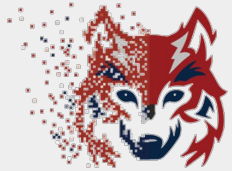
Reasons for choosing Ookami for your computations

- Great resource to test A64FX and do benchmarking
- 174 compute nodes in SLURM, 8352 cores
  - 2 interactive debug A64FX nodes
- If you don’t care about the technical details:

  A lot of preinstalled software and many commonly used science codes

  > *400 modules* (e.g. OpenFOAM, lammps, QuantumEspresso, ROMS)

  additional software installs can be requested
- Dedicated support team
- Usually short queueing times
- No costs for using it



Part III

Getting started on Ookami

Target Audience: Ookami newcomers

Getting started on Ookami

- Commonly supported high-performance related languages
  - C
  - C++
  - Fortran
  - Python & infrastructure
- Compiler suites
  - Fujitsu
  - Cray
  - ARM
  - LLVM
  - GCC
    - System version old, use newer module to get any real optimization
  - NVIDIA

Getting started on Ookami

- Modules
  - Ookami has tons of packages available through the `module` command
  - List them with

```
$ module avail
```

  - Find specific packages, e.g.

```
$ module av openmpi
```

  - TAB completion!
  - Our packages will load their dependencies for you

Getting started on Ookami

- MPI
  - Message Passing Interface = distributed parallelism (multi-node)
  - Communication over Infiniband and / or in-memory
- Implementations
  - Fujitsu MPI
  - Open-MPI
  - MVAPICH 2 (3 in progress)
  - MPICH
- We make every effort to keep these up to date

- MPI / Compiler combinations
  - Various MPIs built with
    - Fujitsu
    - Cray
    - ARM
    - GCC
    - NVIDIA
  - `module` command will find these

Getting started on Ookami

- Python and friends
  - Conda
  - Pypy
  - mpi4py

Getting started on Ookami

- OpenMP
  - Purely intra-node (one single process) parallelism with threads / tasks
  - Supported by all compilers
  - FAQ shows how to enable for each compiler suite
    - https://www.stonybrook.edu/commcms/ookami/support/faq/Vectorization_Flags.php
  - SBU member of language / specification committee

Getting started on Ookami

- MPI ***and*** OpenMP ?
  - Yes!
  - Quite common, e.g. national labs, DoE
  - In fact, this is probably the best path to distributed / hybrid performance on Ookami

Getting started on Ookami

- Debuggers
  - `gdb` default option under Linux
  - `lldb` (ell-ell-dee-bee) is part of LLVM
  - DDT is part of Linaro Forge
    - Includes remote GUI client
    - If interested, we have a walk-through presentation

Getting started on Ookami

- Profilers
  - MAP and `perf-report` are part of Linaro Forge
    - MAP has remote client like DDT
  - `perf` is standard Linux tool
  - PAPI

Part IV

Getting the best out of Ookami

Target Audience: More experienced Ookami users

Getting the best out of Ookami

- Optimization in compilers
  - To take advantage of optimization for a64fx may require special flags, e.g.

```
$ armflang -Ofast -armpl -mcpu=a64fx …
```

  - Vendor compilers come with optimized scientific libraries
    - E.g. BLAS, LAPACK
    - Likely to be much better than “vanilla” `netlib.org` downloads
    - Beware packages with Makefiles etc. trying to link the vanilla ones!
      - Our compilers might well have better options

Getting the best out of Ookami

- Vectorization in compilers
  - Performance on A64fx comes from 2 sources:
    - Parallelism
    - Vectorization
    - (and the 2 are complementary)
  - Telling compilers about vectorization can require command-line options
    - `-mcpu=a64fx` is common to GCC- and LLVM-based ones
    - NVIDIA wants `-tp=native` on e.g. fj-debug nodes

Getting the best out of Ookami

- Getting reports out of compilers
  - You can ask compilers to tell you what they are doing
    - Optimization, vectorization
  - Or not doing
    - Missing optimization, missing vectorization
  - Our comprehensive FAQ lists these options
    - https://www.stonybrook.edu/commcms/ookami/support/faq/Vectorization_Flags.php

Getting the best out of Ookami

- NUMA and process / thread placement
  - A64fx can be sensitive to where processes / threads run
  - 4 NUMA nodes (a.k.a. Core Management Groups)
    - Each with 12 cores sharing HBM
    - 4 MPI ranks with up to 12 threads per rank is a good first attempt
      - Filling the node completely might not be a good idea
      - Also need to consider memory usage
      - ranks * threads ≤ 48
  - Open-MPI has various command-line options to tightly control placement e.g. `-map-by`
    - Also `numactl` command

Getting the best out of Ookami

- Newer languages / environments
  - OpenACC
    - No GPUs on ookami, but it works (NVIDIA compiler)
    - SBU member of specification team
  - Kokkos / SYCL
    - No GPUs on ookami, but it works with OpenMP (LLVM C++) backend

- Newer languages / environments

  - Chapel
    - We have good connections with the Cray development team
  - Julia
    - Supports a64fx
    - Developers use ookami. Also will answer questions on our slack
  - Rust, Go

Getting the best out of Ookami

- Less common languages / environments
  - OpenSHMEM
    - Partitioned Global Address Space library for decoupled parallel communication
    - Available in Open-MPI
  - Coarrays (Fortran)
    - Available in Fujitsu and Cray: built-in
    - Also can be done through GCC and OpenCoArrays library
  - (Just in case anyone is interested)

Getting the best out of Ookami

- What's up with LLVM and vendors?
  - We have vendor compilers from e.g. Fujitsu, Cray, ARM, NVIDIA on ookami. All of these are (slowly?) migrating to using LLVM
    - llvm.org
  - as in `clang`, `clang++`, `flang` in various ways
  - So we are seeing a unification of compiler architectures over time
  - This is nice for users, as the command-line options are likely to be similar or identical
    - Interim pain as historical Makefiles break

Part V

Upcoming events

Target Audience: Everybody interested

Upcoming Events

- **Ookami Open OnDemand Webinar - Th 3/7/2024 at 2pm EST**
  Ookami is now available via Open OnDemand, an intuitive, innovative, and interactive interface to remote computing resources. Open OnDemand helps computational researchers and students efficiently utilize remote computing resources by making them easy to access from any device. In this webinar we will show you how to use Open OnDemand on Ookami.
  Register for the Open OnDemand Webinar

- **3rd Ookami User Group Meeting - Th 3/21/2024 at 2pm EST**
  The purpose of this meeting is to share experiences, learn from each other, get feedback and have a fruitful discussion about A64FX and Ookami.The meeting will take place from 2 - 4 pm EST via Zoom and will feature several 5 to 10 minute talks from users. If you are interested in presenting, please indicate so during in the registration form. Have a look at our previous user group meetings (2022, 2023)
  Register for the User Group Meeting

Part VI

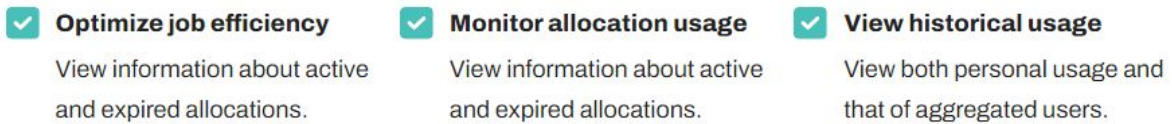
Support

Target Audience: Everybody

Tools

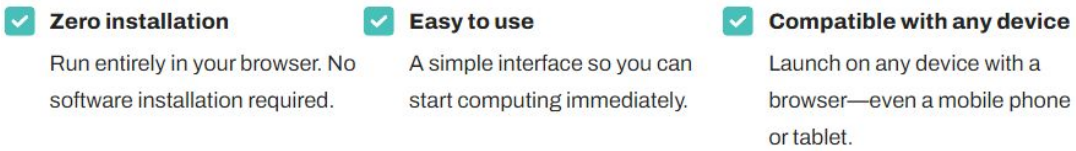
- **XDMoD**

  View information about allocations, usage data, and usage, as well as specific information about your usage and general historic usage of the ACCESS allocated resources.



- **ACCESS OnDemand**

  Easy-to-use web portal to allow researchers to compute from anywhere without client software or command-line interface, and significantly speed up the time to science



Ookami support

There are several Ookami support mechanisms and a dedicated project team that is happy to support you!

- Online documentation & FAQ
- Slack channel
- Virtual office hours
  every Tue 10am - noon EST, and Th 2pm - 4pm EST
- Ticketing system
- Regular webinars
- Email: ookami_computer@stonybrook.edu