Science at Exascale and Beyond

Enabling Breakthroughs in Science and Engineering

Katherine Riley
Director of Science, Argonne Leadership Computing Facility

Sept 28, 2021

Argonne NATIONAL LABORATORY

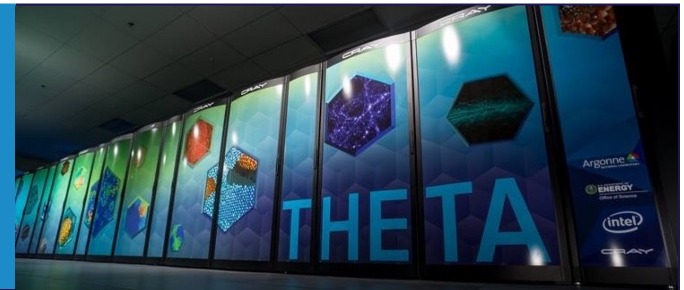
DOE SC

Advanced Scientific Computing Research User Facilities

The Advanced Scientific Computing Research (ASCR) program leads the nation and the world in supercomputing, high-end computational science, and advanced networking for science.

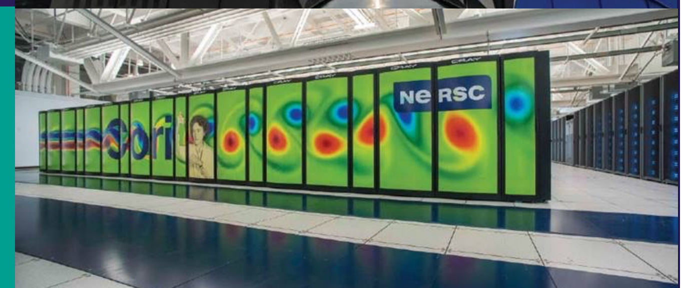
ALCF and OLCF make up the DOE Leadership Computing Facility

Argonne Leadership Computing Facility (ALCF)



Oak Ridge Leadership Computing Facility (OLCF)

National Energy Research Scientific Computing Center (NERSC)



Energy Sciences Network (ESnet)

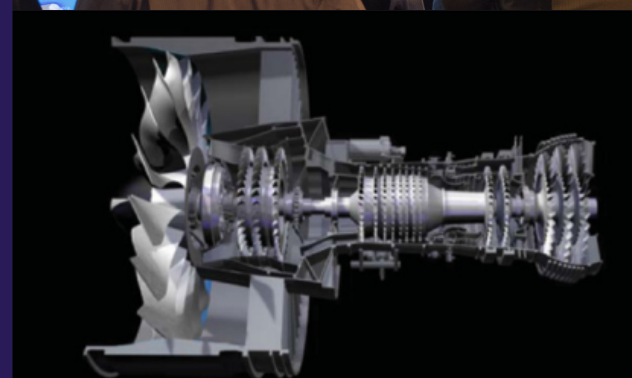
DOE Leadership Computing Facility

- Established in 2004 as a collaborative, multi-lab initiative funded by DOE's ***Advanced Scientific Computing Research*** program
- Operates as **one facility** with two centers, at Argonne and at Oak Ridge National Laboratory
- Deploys and operates at least two advanced architectures that are **10-100 times more powerful** than systems typically available for open scientific research
- **Fully dedicated** to open science to address the ever-growing needs of the scientific community

Shaping the Future of Supercomputing

ALCF resources help the research community advance our knowledge of how things work, provide technological solutions to problems, and keep the nation safe and competitive.

Contributions to Science



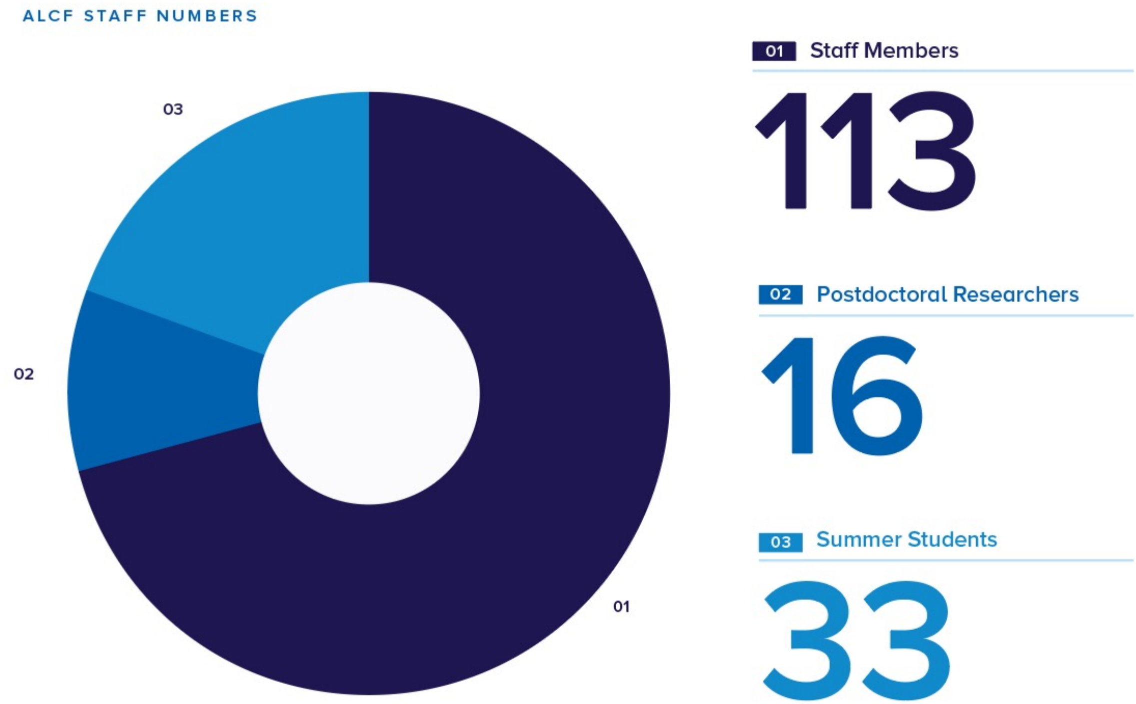
Economic Benefits

Educational Outreach

ALCF Staff

To ensure facility users are able to get the most out of its supercomputers, the ALCF has assembled an exceptional team of:

- HPC system and network administrators
- computational scientists,
- computer scientists
- data scientists
- performance engineers
- visualization experts
- software developers
- user support staff



ALCF at a Glance in 2020

- Users pursue scientific challenges
- In-house experts to help maximize results
- Resources fully dedicated to open science

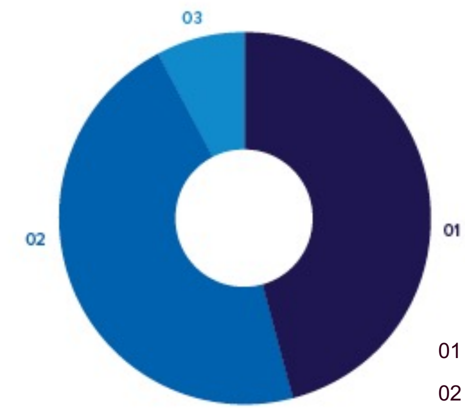
101M node-hours of compute time

369 active projects

1,174 facility users

246+ publications

2020 ALCF Users by Affiliation

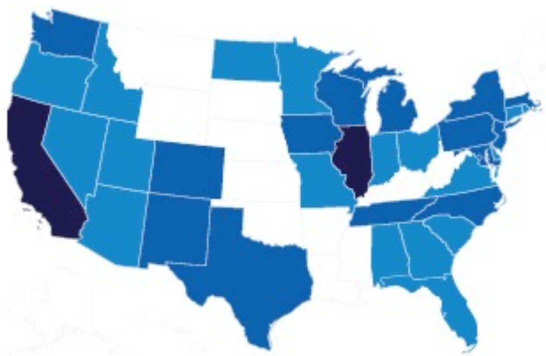


01 | Academia: 545

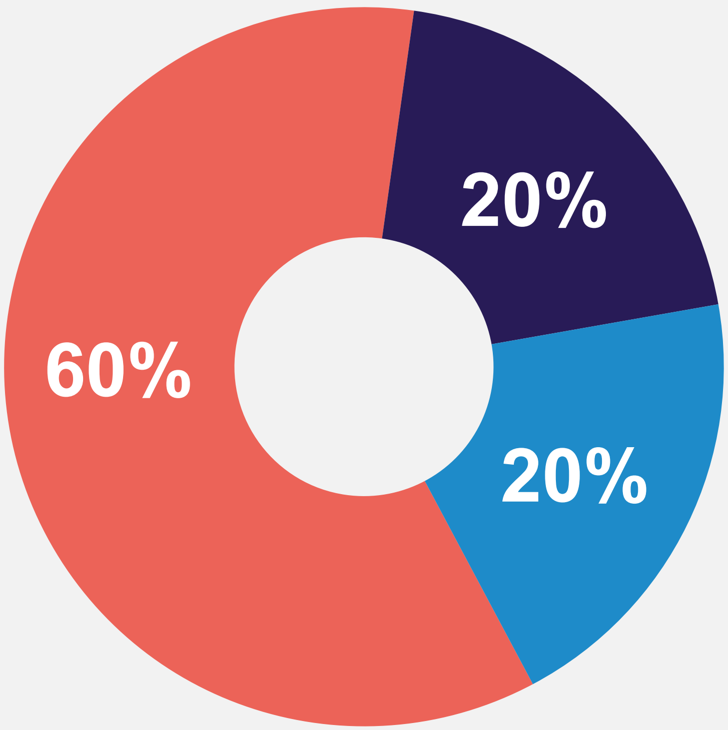
02 | Government: 539

03 | Industry: 90

2020 U.S. ALCF Users by State



ALCF Allocation Programs



INCITE: Innovative and Novel Computational Impact on Theory and Experiment

- Yearly call with computational readiness and peer reviews
- Open to all domains and user communities

ALCC: ASCR Leadership Computing Challenge

- Yearly call with peer reviews
- Focused on DOE priority

DD: Director's Discretionary Program

- Rapid allocations for project prep and immediate needs
  - Early Science Program (ESP)
  - Exascale Computing Project (ECP)
  - ALCF Data Science Program (ADSP)
  - Proprietary Projects

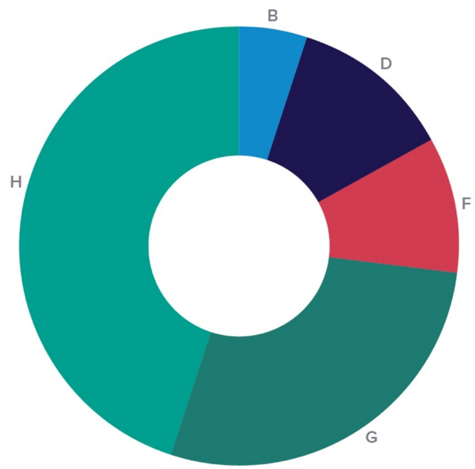
Accessing ALCF Resources for Science

As a national user facility dedicated to open science, any researcher in the world with a large-scale computing problem can apply for time on ALCF computing resources.

2020 INCITE

17.8M NODE HOURS

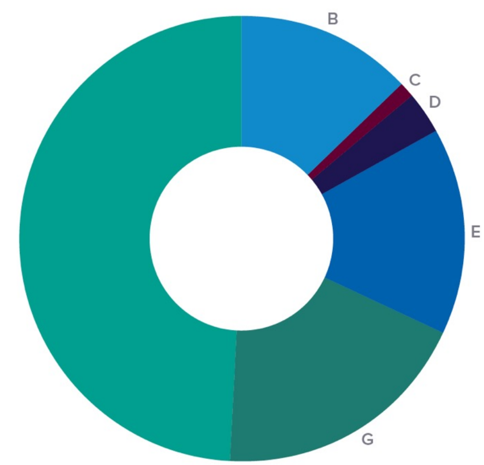
A	Biological Sciences	– %
B	Chemistry	5
C	Computer Science	–
D	Earth Science	12
E	Energy Technologies	–
F	Engineering	10
G	Materials Science	28
H	Physics	45



2020 ALCC

5.87M NODE HOURS

A	Biological Sciences	– %
B	Chemistry	13
C	Computer Science	1
D	Earth Science	3
E	Energy Technologies	15
F	Engineering	–
G	Materials Science	19
H	Physics	49



ALCC data are from calendar year 2020.

LCF Growth and Impact of the INCITE Program

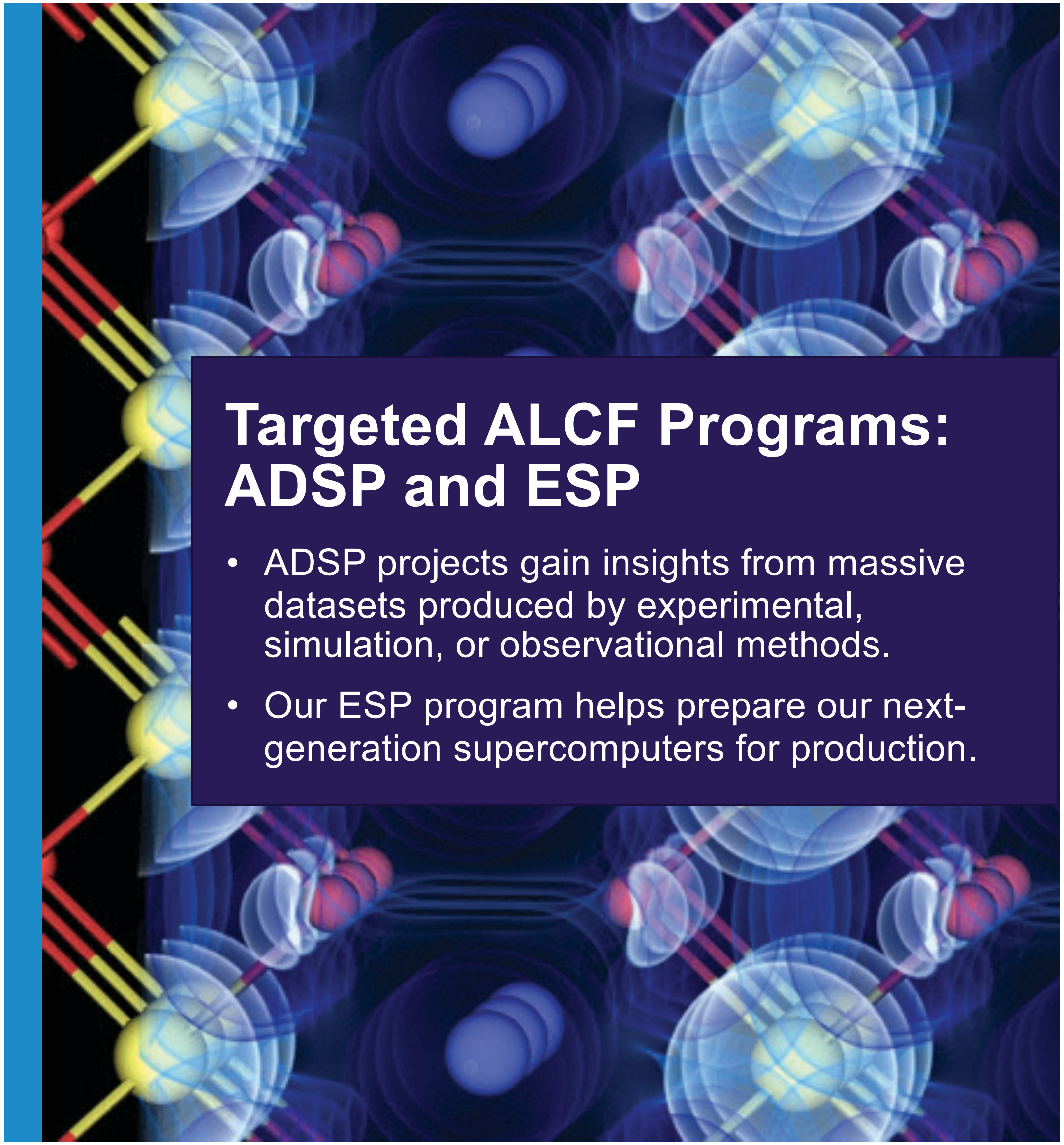
~2X per year (2008–2013) ~3X per year (2014–2018) ~4X per year (2019–2020)

	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019*	2020*
Hours	4.9M	6.5M	18.2M	95M	268M	889M	1.6B	1.7B	1.7B	4.7B	5.8B	5.8B	5.8B	5.8B	5.9B	71M	38.5M
Projects	3	3	15	45	55	66	69	57	60	61	59	56	56	55	55	62	47

2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

- Researchers solved the 2D Hubbard model and presented evidence that it predicts HTSC behavior. ***Phys. Rev. Lett*** (2005)
- Modeling of molecular basis of Parkinson's disease named #1 computational accomplishment. ***Breakthroughs*** (2008)
- Largest simulation of a galaxy's worth of dark matter, showed for the first time the fractal-like appearance of dark matter substructures. ***Nature*** (2008), ***Science*** (2009)
- World's first continuous simulation of 21,000 years of Earth's climate history. ***Science*** (2009)
- Largest-ever LES of a full-sized commercial combustion chamber used in an existing helicopter turbine. **Compte Rendus Mecanique** (2009)
- Unprecedented simulation of magnitude-8 earthquake over 125-square miles. **Proc. SC10**
- OMEN breaks the petascale barrier using more than 220,000 cores. **Proc. SC10**
- New method to rapidly determine protein structure, with limited experimental data. ***Science*** (2010), ***Nature*** (2011)
- Calculation of the number of bound nuclei in nature. ***Nature*** (2012)
- NIST proposes new standard reference materials from LCF concrete simulations.
- Recovery from slow inactivation in potassium channels controlled by H_2O. ***Nature*** (2013)
- Macroscale superlubricity enabled by graphene nanoscroll formation. ***Science*** (2015)
- Carbon-based tribofilms from lubricating oils. ***Nature*** (2016)
- Quantitative 3D evolution of colloidal nanoparticle oxidation in solution. ***Science*** (2017)
- Ultra-selective high-flux membranes from directly synthesized zeolite nanosheets. ***Nature*** (2017)
- Overcame limitations modeling massive stars. ***Nature*** (2018)
- Microscope-in-a-computer to help find early cancer. ***Nature*** (2019)
- Accelerate vaccine and drug identification for COVID-19. Proc. ***SC20***

*change allocation unit



Targeted ALCF Programs: ADSP and ESP

- ADSP projects gain insights from massive datasets produced by experimental, simulation, or observational methods.
- Our ESP program helps prepare our next-generation supercomputers for production.

ADSP: ALCF Data Science Program

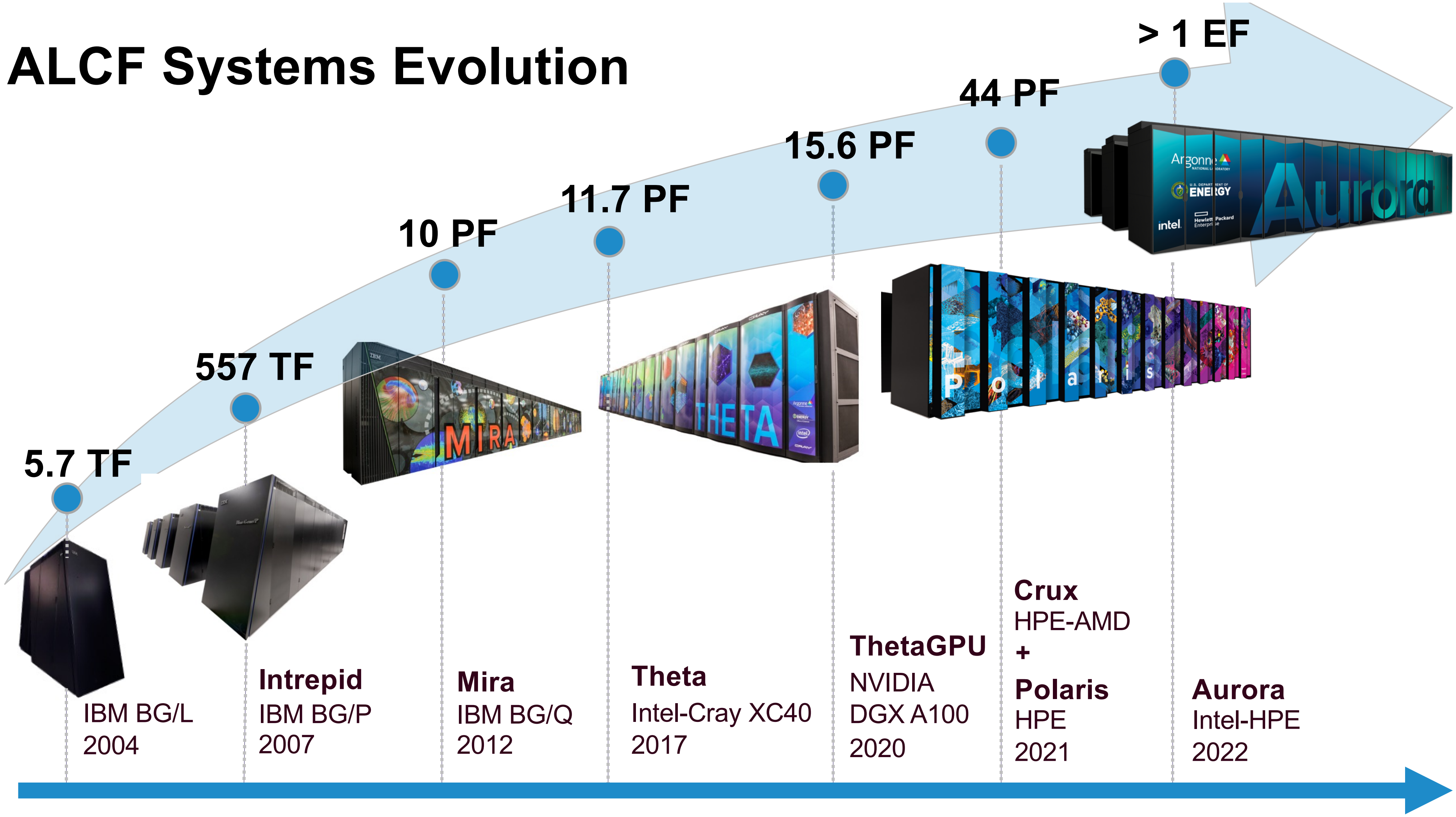
- Award Cycle: October 1 to September 30
- Award size: millions of compute-hours
- Award duration: 2 years
- ALCF resources: Varies according to project needs

ESP: Early Science Program

- Award Cycle: Determined by production timeline
- Successful program adopted by other leadership facilities.
- Helps bring each new supercomputer into production.
- PI-led projects represent the most challenging applications for a new architecture.

Argonne NATIONAL LABORATORY

ALCF Systems Evolution



Computing Resources

Polaris
- HPE
- AMD processors/NVIDIA GPUs
- 44 petaflops

Theta

KNL NODES
- Intel-Cray XC40
- 11.69 petaflops
- 4,392 nodes
- 281,088 cores
- 843 TiB of memory

GPU NODES
- NVIDIA DGX A100
- 3.9 petaflops
- AMD EPYC 7742
- 24 nodes 24 TB of DDR4 memory
- 7, 680 GB of GPU memory

Cooley
- Cray/NVIDIA
  126 nodes
- 1512 Intel Haswell CPU cores
- 126 NVIDIA Tesla K80 GPUs
- 48 TB RAM / 3 TB GPU

Iota
- Intel/Cray XC40 architecture
- 117 teraflops
- 44 nodes
- 2,816 cores
- 12.3 TB of memory

JLSE Experimental Testbeds
- 150 nodes
- Intel/AMD/IBM/Marvell/GPGPU
- EDR/100GbE/OPA
- Lustre/GPFS/DAOS

Grand and Eagle (Storage)
Each system has:
- HPE ClusterStor E1000
- 100 petabytes of usable capacity
- 8,480 disk drives
- Lustre filesystem
  - 160 Object Storage Targets
  - 40 Metadata Targets
- HDR Infiniband network
- 650 GB/s rate on data transfers

Community Data Sharing with Grand and Eagle

- A global filesystem deployed to bring larger and more capable production-level file sharing to facility users
- A space for broader distribution of reassembled data acquired from various experiments
  - Data originating at the ALCF
  - Greater scientific community
- Science community can access uploaded data, and ALCF users are able to directly access the data for analysis
- Designed to foster experimentation
  - Analysts are able to write new algorithms to attempt analyses that have never been performed

- **HPE ClusterStor E1000**
- **200 petabytes of usable capacity**
- **16,960 disk drives**
- **Lustre filesystem**
- **320 Object Storage Targets**
- **80 Metadata Targets**
- **HDR Infiniband network**
- **650 GB/s rate on data transfers per filesystem**

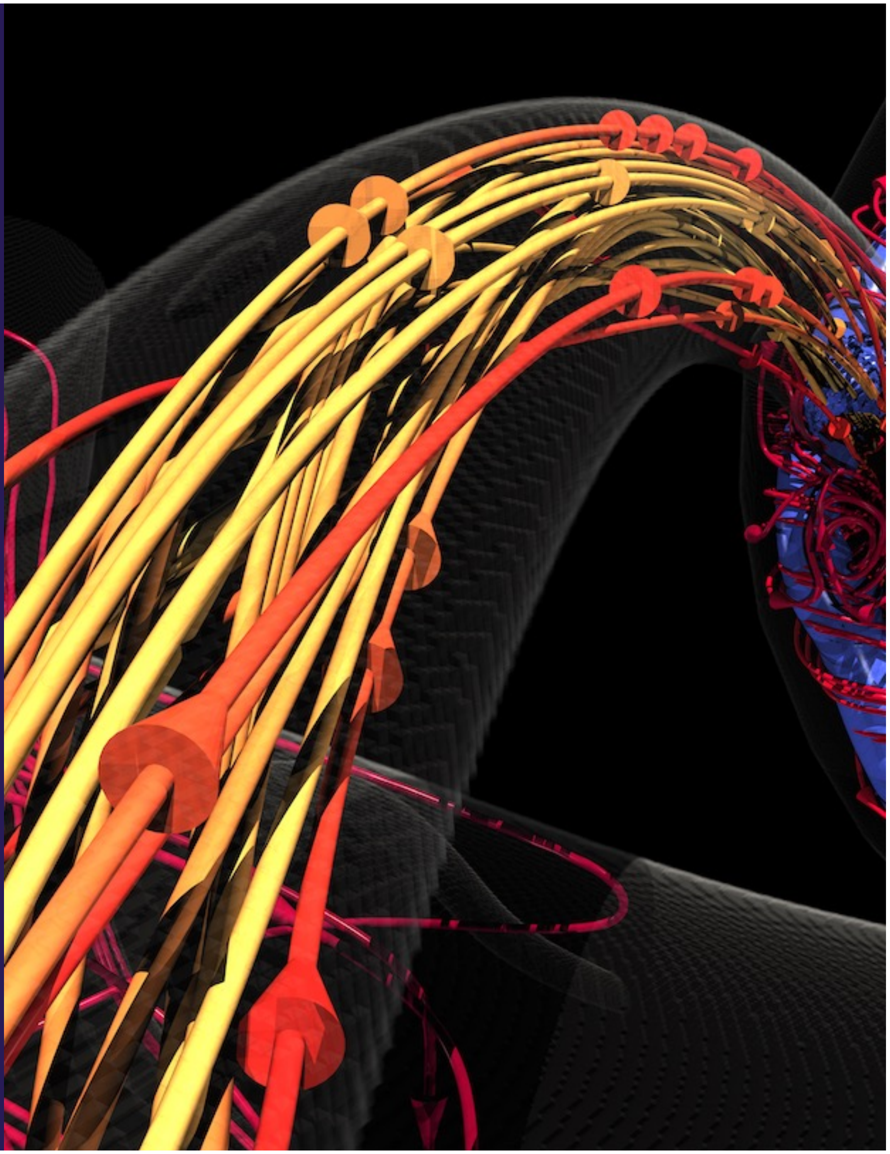
Charging Ahead: AI Testbed

Advancing science with HPC

- ALCF AI pathfinding effort provides insights on cutting-edge AI technology and how it improves science outcomes
- Evaluates the usability and performance of machine learning-based applications running on these accelerators
  - a deep learning accelerator, reconfigurable dataflow units, intelligent processing unit- (IPU) based systems
- Ongoing work is guiding the facility toward a future marked by extreme heterogeneity in the compute: CPUs, GPUs, AI, and other accelerators

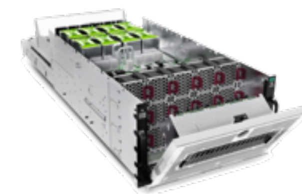
The following testbed hardware is deployed:

- SambaNova
- GraphCore
- Groq



System Configuration	Polaris
# of River Compute Racks	40
# of Apollo Gen10+ Chassis	280
# of Nodes	560
# of AMD EPYC 7532 CPUs	560
# of NVIDIA A100 GPUs	2240
Total GPU HBM2 Memory	87.5 TB
Total CPU DDR4 Memory	280 TB
Total NVMe SSD Capacity	1.75 PB
Interconnect	HPE Slingshot
# of Cassini NICs	1120
# of Rosetta Switches	80
Total Injection BW (w/ Cassini)	28 TB/s
Total GPU DP Tensor Core Flops	44 PF
Total Power	1.8 MW

Single Node Configuration	Polaris
# of AMD EPYC 7532 CPUs	1
# of NVIDIA A100 GPUs	4
Total HBM2 Memory	160 GB
HBM2 Memory BW per GPU	1.6 TB/s
Total DDR4 Memory	512 GB
DDR4 Memory BW	204.8 GB/s
# of NVMe SSDs	2
Total NVMe SSD Capacity	3.2 TB
# of Cassini NICs	2
Total Injection BW (w/ Cassini)	50 GB/s
PCIe Gen4 BW	64 GB/s
NVLink BW	600 GB/s
Total GPU DP Tensor Core Flops	78 TF



Apollo 6500 Gen10+

NVIDIA HGX A100 4-GPU

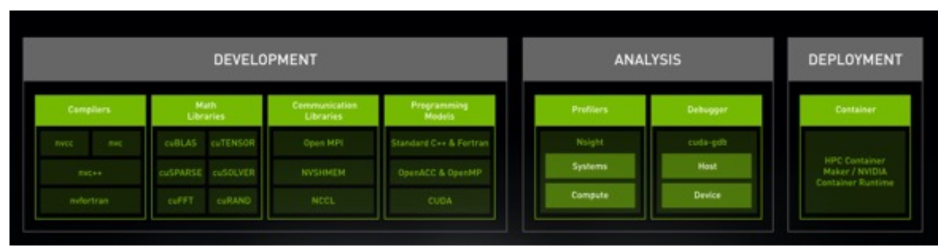
ALCF Resources - Polaris

Software Overview

- Provides an excellent platform for preparing application codes for Aurora
  - All programming models available on Aurora can be tested
  - Features HPE Cray (PE) Programming Environment
  - Built with HPE HPCM system software
- Provides excellent capabilities in simulation, data and learning using Nvidia's existing HPC SDK
- Support for HPE Cray MPI and MPICH via libfabric using Slingshot provider
  - Initial SS10 feature support
  - later full SS11 feature support for testing all MPI features available on Aurora

Programming Environment

- HPE Cray PE for Polaris
  - HPE Cray MPI support for PGI offload to A100 for Multi-NIC and Multi-GPU support
  - Full Rome and Milan support
- SYCL/Data Parallel C++ provided via
  - CodePlay computecpp compiler with Nvidia support
  - LLVM via Intel DPC++ branch which supports offload to Nvidia GPUs as well as Intel GPUs
- Next NVIDIA HPC SDK will provide primary support for programming A100



Bridge to Aurora – Bringing Programming Models Together

- Polaris will provide a platform for preparation for Aurora
- Polaris and Aurora will have many similarities at the system and user level

Component	Polaris	Aurora
System Software	**HPCM**	HPCM
Programming Models	**MPI, OpenMP, DPC++, Kokkos, RAJA, HIP**, CUDA, OpenACC	**MPI, OpenMP, DPC++, Kokkos, RAJA, HIP**
Tools	**PAT, gdb, ATP,** NVIDIA Nsight, cuda-gdb	**PAT, gdb, ATP,** Intel Vtune
MPI	**HPE Cray MPI, MPICH**	**HPE Cray MPI, MPICH,** Intel MPI
Multi-GPU	1 CPU : 4 GPU	2 CPU : 6 GPU
Data and Learning	**DL frameworks, Cray AI stack, Python/Numba, Spark, Containers,** Rapids	**DL frameworks, Cray AI stack, Python/Numba, Spark, Containers,** oneDAL
Math Libraries	cu* from CUDA	oneAPI

Aurora

Argonne's upcoming exascale supercomputer will leverage several technological innovations to support machine learning and data science workloads alongside traditional modeling and simulation runs.

SUSTAINED PERFORMANCE
≥1 Exaflop DP

X^e ARCHITECTURE-BASED GPU
Ponte Vecchio

INTEL XEON SCALABLE PROCESSOR
Sapphire Rapids

PLATFORM
HPE Cray EX

Compute Node
2 Intel Xeon scalable "Sapphire Rapids" processors; 6 X^e arch-based GPUs; Unified Memory Architecture; 8 fabric endpoints; RAMBO

GPU Architecture
X^e arch-based "Ponte Vecchio" GPU; Tile-based chiplets, HBM stack, Foveros 3D integration, 7nm

CPU-GPU Interconnect
CPU-GPU: PCIe
GPU-GPU: X^e Link

System Interconnect
HPE Slingshot 11; Dragonfly topology with adaptive routing

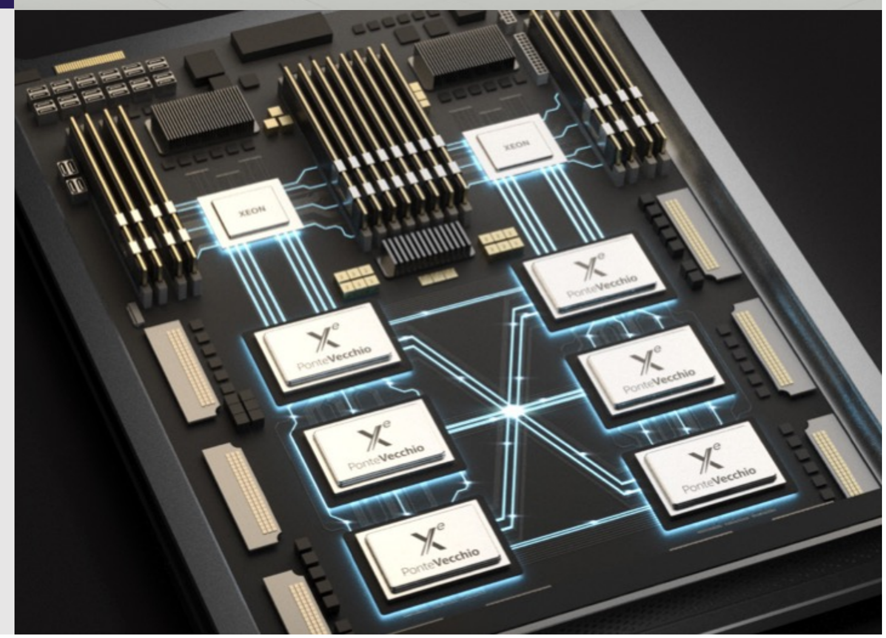
Network Switch
25.6 Tb/s per switch, from 64–200 Gbs ports (25 GB/s per direction)

High-Performance Storage
≥230 PB, ≥25 TB/s (DAOS)

Programming Models
Intel oneAPI, MPI, OpenMP, C/C++, Fortran, SYCL/DPC++

Node Performance
>130 TF

System Size
>9,000 nodes



ALCF Resources - Aurora

- Platform
  - HPE Cray XE
- Software Stack
  - HPE Cray XE software stack + Intel enhancements + Data and Learning
- Compilers
  - Intel, LLVM, GCC
- Programming Models
  - Intel oneAPI, OpenMP, DPC++/SYCL
- Programming Languages and Models
  - Fortran, C, C++, OpenMP 5.x (Intel, Cray, and possibly LLVM compilers), UPC (Cray), Coarray Fortran (Intel), Data Parallel C++ (Intel and LLVM compilers), Open SHMEM, Python, Numba, MPI, OpenCL
- Programming Tools
  - Open|Speedshop, TAU, HPCToolkit, Score-P, Darshan, Intel Trace Analyser and Collector, Intel Vtune, Advisor, and Inspector, PAPI, GNU gprof
- Debugging and Correctness Tools
  - Stack Trace Analysis Tool, gdb, Cray Abnormal Termination Processing
- Math Libraries
  - Intel MKL, Intel MKL-DNN, ScaLAPACK
- GUI and Viz APIs, I/O Libraries
  - X11, Motif, QT, NetCDF, Parallel, NetCDF, HDF5
- Frameworks
  - TensorFlow, PyTorch, Scikit-learn, Spark Mllib, GraphX, Intel DAAL, Intel MKL-DNN

https://aurora.alcf.anl.gov

Preparing researchers for Aurora

ALCF is engaged in several training and outreach activities designed to prepare the HPC community for science in the exascale era.

Public Training

- Aurora Early Adopter Series and Developer Sessions (webinars)
- ALCF Simulation, Data, and Learning Workshop
- ALCF Computational Performance Workshop
- Intel oneAPI Webinars
- ECP Workshops and Webinars
- ALCF-NVIDIA GPU Hackathon

Currently ESP/ECP Focused

- Aurora COE Workshops
- Aurora ESP Hackathons

Join in:
https://www.alcf.anl.gov/events

Outreach/Informational Services

- Best Practices for GPU Code Development (article series)
- Let's Talk Exascale Code Development (podcast series w/ ECP)
- Aurora Software Development (articles series on ALCF staff efforts)
- Aurora Early Science Program Project Profiles (article series)
- Exascale-Themed Social Media Activities (Twitter Chat, Instagram Live)

Preparing Users for Exascale

Early Science Program

- Ensure the facility's next-generation systems are ready for science on day one – 15 (+5) projects

- Provides research teams with critical pre-production computing time and resources
  - collaboration with ALCF staff and post-doc
  - prepares applications for the architecture and scale of a new supercomputer
  - solidifies libraries and infrastructure for other
  - production applications to run on the system

Exascale Computing Program HI/AD

- Ensure ECP application success (~14 projects)
- Funds ALCF staff to collaborate on readiness for Aurora

2020 Training Activities
- 8 Workshops
- 12 Webinars
- 5 Hackathons

ESP Projects Software Dependencies

PI	Pillar	Compiled Codes	Compiled Languages	Numerical Libraries	Productivity Languages	ML/DL Frameworks	Other Apps/Packages
Benali	Simulation	QMCPACK	C++	MKL, FFTW, BLAS/LAPACK			HDF5, ADIOS, libXML
Chang	Simulation	XGC	F90	PETSc, PSPLINE, LAPACK			ADIOS, parMETIS
Dunning	Simulation	NWChemEx	C++17				
Heitmann	Simulation	HACC	C++				Thrust (analytics)
Jansen	Simulation	PHASTA	F90, C, C++	PETSc			PUMI, Zoltan, parMETIS
Berzins	Simulation	UINTAH	C++	Hypre			Kokkos
Lele	Simulation	SU2, PadeOps	C++, Fortran	FFTW			HDF5
Christ	Simulation	USQCD codes*	C, C++				
Nakano	Simulation	NAQMD, RMD	F90, C++	FFT, BLAS			
Roux	Simulation	NAMD	C, C++	FFTW/MKL			Charm++

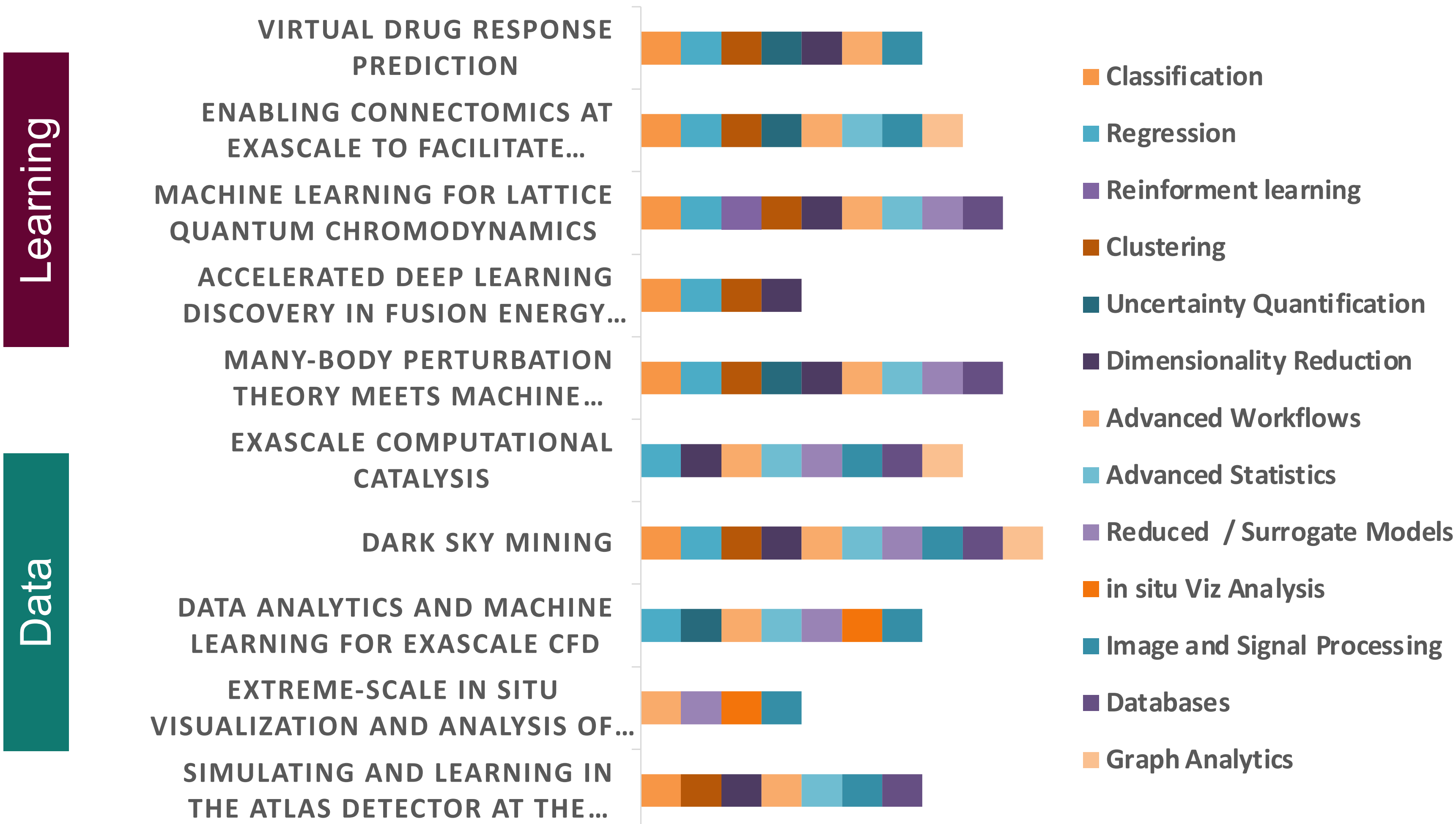
*MILC, CPS, Chroma/Redstar, RBC/Bielefeld

ESP Projects Software Dependencies

PI	Pillar	Compiled Codes	Compiled Languages	Numerical Libraries	Productivity Languages	ML/DL Frameworks	Other Apps/Packages
Bross	Data	NWchemEx	C++17		Python		NumPy, SciPy, Cython, Balsam
Habib	Data	HACC	C++	Thrust, FFTW	Python, R	TensorFlow, Keras, Scikit-Learn, PyTorch, mufnn, LaGP	NumPy, Pandas
Jansen	Data	PHASTA	F90, C, C++	PETSc			
Proudfoot	Data	Athena	C		Python	TensorFlow	CERN ROOT
Randles	Data	HARVEY	C++	SENSEI			
Detmold	Learning	USQCD codes*	C, C++		Python	TensorFlow, Deep Hyper	Balsam
Ferrier	Learning				Python	Tensorflow, Horovod	Tomosaic
Marom	Learning	BerkeleyGW, Quantum Espresso, SISSO	F90	BLAS, LAPACK, ScaLAPACK, FFTW, ELPA	Python	TensorFlow, Pytorch, cuDNN	CUDA
Stevens	Learning				Python	TensorFlow, Keras	CANDLE
Tang	Learning				Python	TensorFlow, Keras	FRNN

*MILC, CPS, Chroma/Redstar, RBC/Bielefeld

AURORA ESP Data and Learning Methods



Early Science Project Metric for Success

- INCITE Readiness is driver for ESP Success on Aurora
  - —Scaling to 20% of Aurora
  - —Effective use of hardware (use GPUs)
  - —Ready to run science problem within 3 months

- Progress tracked
  - —Staff engagement
  - —Quarterly report

20 ESP projects driving debug of the hardware, software, and policies for Aurora

ECP APPLICATIONS WITH ALCF HI EFFORT

Project	Codes	Languages	Programming Models	Libraries	Staff POC
GAMESS	GAMESS	Fortran, C++	OpenMP, SYCL		Coleen Bertoni, Yuri Alexseev
NWChemEx	NWChemEx	C++	DPC++	BLAS, LAPACK, cuTT, Umpire, Cereal, BLIS	Victor Anisimov, Abhishek Bagusetty
QMCPack	QMCPack	C++	OpenMP	MKL, HDF5, boost	Thomas Applencourt
ExaSMR	NekRS, OpenMC	Fortran, C++	OCCA (OpenMP/SYCL), OpenMP	HYPRE	Kris Rowe, Saumil Patel
CANDLE	CANDLE	Python		TensorFlow, PyTorch	Murali Emani
ExaSky	HACC, Nyx	C++	OpenCL, OpenMP, SYCL	AMREx	Vitali Morozov, Esteban Rangel
LatticeQCD	Chroma, CPS, MILC	C, C++	DPC++, OpenMP		James Osborn
WDMApp	GENE, GEM, XGC	Fortran, C++	OpenMP, DPC++, Kokkos	PETSc, Cabana, gtensor, EFFIS	Tim Williams
E3SM-MMF	E3SM	Fortran, C++	OpenMP, DPC++	PNETCDF, HDF5	Abhishek Bagusetty
EXAALT	LAMMPS, LATTE, ParSplice	C++, Fortran	OpenMP, Kokkos	MKL, BML, PROGRESS	Yasaman Ghadar, Chris Knight
ExaWind	Nalu-wind, AMR-Wind, OpenFAST	C++, Fortran	Kokkos	Trilinos, HYPRE, HDF5, PNETCDF, YAML-CPP	JaeHyuk Kwack
EQSim	SW4	C++	Raja	EXAIO, UMPIRE, ZPF	Brian Homerding
ExaStar	Flash, Castro, Thornado	Fortran, C++	OpenMP	MKL, HDF5, AMReX	Brice Videau
ExaFEL	spiniFEL	C++, Python		numpy, FINUFFT	Servesh Muralidharan

ECP APPLICATIONS WITH ALCF HI EFFORT

Project	Codes	Languages	Programming Models	Libraries	Staff POC
GAMESS	GAMESS	Fortran, C++	OpenMP, SYCL		Coleen Bertoni, Yuri Alexseev
NWChemEx	NWChemEx	C++	DPC++	BLAS, LAPACK, cuTT, Umpire, Cereal, BLIS	Victor Anisimov, Abhishek Bagusetty
QMCPack	QMCPack	C++	OpenMP	MKL, HDF5, boost	Thomas Applencourt
ExaSMR	NekRS, OpenMC	Fortran, C++	OCCA (OpenMP/SYCL), OpenMP	HYPRE	Kris Rowe, Saumil Patel
CANDLE	CANDLE	Python		TensorFlow, PyTorch	Murali Emani
ExaSky	HACC, Nyx	C++[illegible]	[illegible]	[illegible]	[illegible]
LatticeQCD	Chroma, CPS, MILC	C, C[illegible]	[illegible]	[illegible]	[illegible]
WDMApp	GENE, GEM, XGC	Fortran, C++	OpenMP, DPC++, Kokkos	PETSc, Cabana, gtensor, EFFIS	Tim Williams
E3SM-MMF	E3SM	Fortran, C++	OpenMP, DPC++	PNETCDF, HDF5	Abhishek Bagusetty
EXAALT	LAMMPS, LATTE, ParSplice	C++, Fortran	OpenMP, Kokkos	MKL, BML, PROGRESS	Yasaman Ghadar, Chris Knight
ExaWind	Nalu-wind, AMR-Wind, OpenFAST	C++, Fortran	Kokkos	Trilinos, HYPRE, HDF5, PNETCDF, YAML-CPP	JaeHyuk Kwack
EQSim	SW4	C++	Raja	EXAIO, UMPIRE, ZPF	Brian Homerding
ExaStar	Flash, Castro, Thornado	Fortran, C++	OpenMP	MKL, HDF5, AMReX	Brice Videau
ExaFEL	spiniFEL	C++, Python		numpy, FINUFFT	Servesh Muralidharan

14 projects on this list, though the list grows and shrinks
Adding to the ESP projects preparing Aurora

ExaSMR

Overview

- ALCF Catalyst: Kris Rowe, Saumil Patel
- Overview
  - Simulation of a full small modular reactor (SMR) core
  - Multiphysics solver—coupling neutronics and computational fluid dynamics (CFD)
  - Science problem will require 22 billion degrees of freedom for CFD, 10 billion particles for neutronics
- Key Components:
  - **CFD: NekRS**
    - A port of Nek5000 to GPU architectures
    - C/C++, F77, MPI, OCCA framework
  - **Neutronics: OpenMC**
    - C++, MPI, OpenMP offload

Activities and Results

- OpenMC built by Intel on nightly basis to accelerate
- A fork of OCCA was created on the ALCF GitHub in order to share the most up-to-date version of the OCCA DPC++ backend
- ALCF staff successfully ran NekRS test case involving a single reactor core pin to compare the performance of the OCCA DPC++ and OpenCL backends on Intel GPUs.
- As part of ongoing collaboration with Intel, integration of math libraries – such as oneMKL – with OCCA is being investigated.
- Recent presentations on the OCCA DPC++ backend:
  - P3HPC forum during the ECP annual meeting (04/14/21)
  - oneAPI Developers Summit at ISC'21 (06/22/21)
  - CEED Annual Meeting (8/3/21)
  - ALCF Aurora Software Development Series
    - https://www.alcf.anl.gov/news/aurora-software-development-bringing-occa-open-source-library-exascale

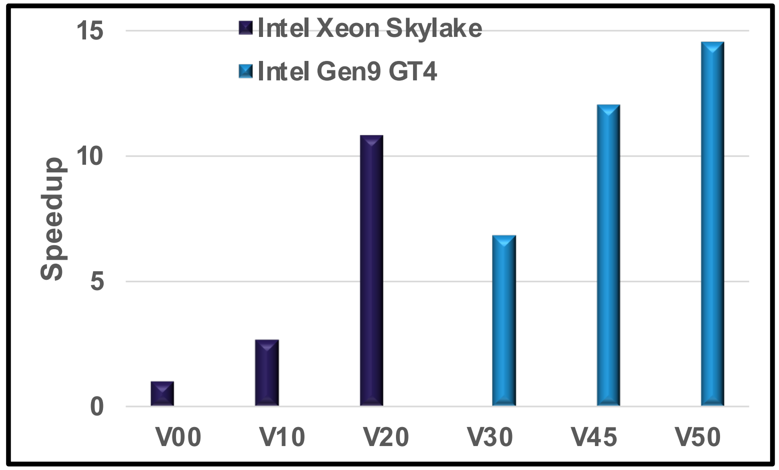
GAMESS

Overview

- ALCF Catalyst: Colleen Bertoni
- Goal is to enable quantum chemistry on extremely large systems of interest in catalysis and energy research.
  - Science challenge is to compute energies/reaction pathways of catalysis reactions in a large silica nanoparticle including solvent.
  - Plan is to use various fragmentation methods in GAMESS ((E)FMO/RI-MP2, (E)FMO/CCSD(T), (E)FMO RI-CR-CC(2,3)).
- Programming models:
  - Linear algebra libraries
  - CUDA
  - Plans for HIP/DPC++/OpenMP
- Key physics modules
  - GAMESS Fortran code (fragmentation framework and input reader)
    - MPI/OpenMP threading for CPU with OpenMP offload for part of the RI-MP2 code for GPU
  - GAMESS C/C++ integral libraries (LibAccInt, which contains GPU code for integrals)
    - CUDA code, OpenMP offload port in progress, also considering DPC++

Activities and Results

- Porting RI-MP2 mini-app to Intel GPUs with OpenMP offload
  - Series of progressive optimizations, including OpenMP threading (V10), porting to MKL (V20), offloading to GPU (V30), restructuring loops (V45), and enabling concurrent CPU+GPU computation (V50)

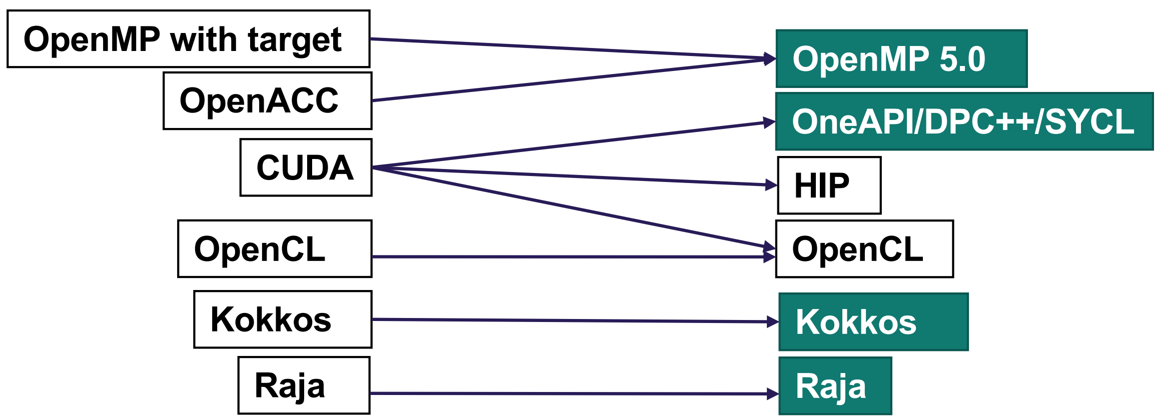


- Fortran development
  - Updated the Fortran GAMESS code to the latest OpenMP offload from the team and investigating runtime issues
  - Working with Intel team on resolving compiler/runtime issues

Argonne NATIONAL LABORATORY

Portable approaches for Aurora and Exascale

- Aurora's typical production use-cases will combine across simulation, data, and learning
- Porting and optimizing a compiled C/C++/Fortran code is an incomplete paradigm
- Analyze workflow
  - —Does it run on a supercomputer?
  - —Is I/O based on a standard?
    - Consider a standard HDF/MPI-IO/Spark-RDD/SQL/Apache Arrow/…. *or* port optimally to DAOS
  - —GPU implementations?
    - Consider the architectural impact on application
    - OpenMP might be first

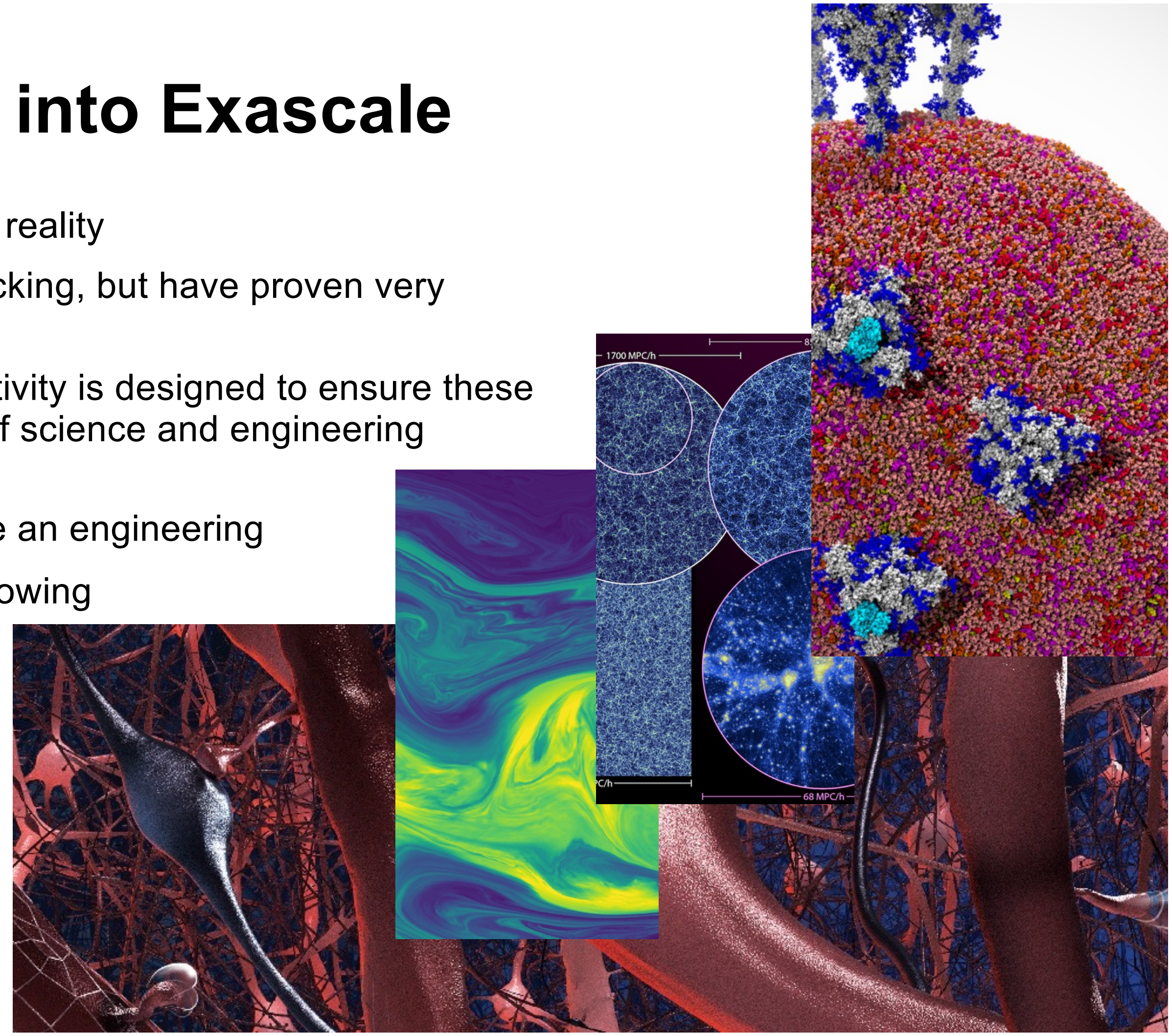


Best Practices so Far – Good Software Practices

- Good software engineering and practices will enable smoother development of scientific applications
- New acceleration(with GPUs) are likely to need refactoring. Could benefit from new algorithms.
- Choose an approach that aligns with how much effort the team has to put into the code
- Identify a portability strategy from the start
  - —CPU: Intel/AMD/Arm
  - —GPU: Intel/AMD/NVIDIA
- High-level portability layers are worth considering (e.g. Kokkos)
  - —Work with developers of that layer if you can
- If targeting a future GPU, easiest development is on an earlier generation
- Optimization
  - —First - Focus on functionality, correctness and portability first
  - —Use a physically relevant problem
- Specific optimization approaches and parameters will be made public as we can make them public

ALCF/Aurora moving into Exascale

- In coming year, exascale will be come a reality
- Current best practices might not be shocking, but have proven very important
- An unprecedented level of readiness activity is designed to ensure these systems are productive for wide range of science and engineering problems
- Aurora will be a powerful tool for science an engineering
- Complexity is scientific HPC needs is growing
- ALCF is planning around those needs



Questions