FRONTIER Spec Sheet

The Frontier system will be composed of more than 100 Cray Shasta cabinets with high density compute blades powered by HPC and AI- optimized AMD EPYC™ processors and Radeon Instinct™ GPU accelerators purpose-built for the needs of exascale computing. The new accelerator-centric compute blades will support a 4:1 GPU to CPU ratio with high speed AMD Infinity Fabric™ links and coherent memory between them within the node. Each node will have one Slingshot interconnect network port for every GPU with streamlined communication between the GPUs and network to enable optimal performance for high-performance computing and AI workloads at exascale.

To make this performance seamless to consume by developers, ORNL and Cray will partner with AMD to co-design and develop enhanced GPU programming tools designed for performance, productivity and portability, which will tightly integrate with the existing AMD ROCm open compute platform. In addition, Frontier will support many of the same compilers, programming models, and tools that have been available to OLCF users on both the Titan and Summit supercomputers. In fact, Summit is a premier development platform for Frontier.

Hardware

Peak Performance	>1.5 EF
Footprint	> 100 cabinets
Node	1 HPC and AI Optimized AMD EPYC CPU 4 Purpose Built AMD Radeon Instinct GPU
CPU-GPU Interconnect	AMD Infinity Fabric Coherent memory across the node
System Interconnect	Multiple Slingshot NICs providing 100 GB/s network bandwidth Slingshot dragonfly network which provides adaptive routing, congestion management and quality of service.
Storage	2-4x performance and capacity of Summit's I/O subsystem. Frontier will have near node storage like Summit.

Programming Environment

Frontier will support multiple compilers, programming models, and tools, many of which are readily available on Summit today. Below are the compilers, programming languages and models, and additional tools that we are targeting to make available on Frontier. As with any large, complex deployment, the list could change as the project progresses and we get closer to production.

Compilers	Cray PE AMD ROCm GCC

Programming Languages and Models Supported	C, C++, Fortran (for all compilers) OpenMP 5.x (Cray, AMD, and possibly GCC compilers) UPC (Cray and GCC compilers) Coarray Fortran, Coarrary C++ (Cray compilers) AMD HIP Chapel Global Arrays Charm++ GASNet OpenSHMEM
System-level Programming Tools	CrayPat/Apprentice2 Cray Reveal Open | SpeedShop TAU HPCToolkit Score-P VAMPIR
Node-level Programming Tools	GNU gprof PAPI AMD ROCProfiler
Debugging and Correctness Tools	ARM DDT Cray CCDB Stack Trace Analysis Tool Cray GDB4HPC gdb AMD ROCm debug service and GDB-MMI Cray Abnormal Termination Processing
Math Libraries	BLAS LAPACK ScaLAPACK Iterative Refinement Toolkit FFTW or similiar PETSc Trillinos
GUI and Visualization APIs, I/O Libraries	X11 Motif Qt NX, NeatX, or similiar NetCDF HDF5

HIP

The OLCF plans to make HIP available on Summit so that users can begin using it prior to its availability on Frontier. HIP is a C++ runtime API that allows developers to write portable code to run on AMD and NVIDIA GPUs. It is essentially a wrapper that uses the underlying CUDA or ROCm platform that is installed on a system. The API is very similar to CUDA so transitioning existing codes from CUDA to HIP should be fairly straightforward in most cases. In addition, HIP provides porting tools which can be used to help port CUDA codes to the HIP layer, with no loss of performance as compared to the original CUDA application. HIP is not intended to be a drop-in replacement for CUDA, and developers should expect to do some manual coding and performance tuning work to complete the port.

Some key features include:

- HIP is very thin and has little or no performance impact over coding directly in CUDA or hcc "HC" mode.
- HIP allows coding in a single-source C++ programming language including features such as templates, C++11 lambdas, classes, namespaces, and more.
- The "hipify" tool automatically converts source from CUDA to HIP.
- Developers can specialize for the platform (CUDA or hcc) to tune for performance or handle tricky cases

Migration Path from Summit to Frontier

Summit	Frontier	Comments
CUDA C/C++	HIP C/C++	HIP provides tools to help port existing CUDA codes to the HIP layer. HIP is not intended to be a drop-in replacement for CUDA, and developers should expect to do some manual coding and performance tuning work to complete the port.
OpenACC	OpenMP (offload)	OpenACC codes can be migrated to OpenMP (offload) for Frontier. *Direct support for OpenACC on Frontier is still under discussion.*
OpenMP (offload)	OpenMP (offload)	Virtually the same on Summit and Frontier
FORTRAN w/CUDA C/C++	FORTRAN w/HIP C/C++	As with CUDA, this will require interfaces to the C/C++ API calls
CUDA FORTRAN	FORTRAN w/HIP C/C++	As with CUDA, this will require interfaces to the C/C++ API calls

Machine Learning

Like Summit, Frontier will be fine tuned to run AI workloads. The vendor will provide a fully optimized, scalable data science suite. This area is rapidly evolving, so modifications to precise tools/libraries are very likely. The plan is to make widely used frameworks (such as the ones below) available in addition to the Cray Programming Environment Deep-learning plugin. This plugin improves algorithms and performance for training of deep neural networks and provides support for Apache Spark, GraphX, MLib, Alchemist frameworks, and pbdR.

Frameworks	TensorFlow BigDL for Apache Spark PyTorch MXNet Keras MLib scikit-learn OpenCV
Additional Tools	Cray Programming Environment Deep-learning Plugin TensorBoard (notebook-based NN hyper-parameter turning framework)