The Programming Environment for Frontier

Luiz DeRose
Sr. Principal Engineer
Programming Environment Director

a Hewlett Packard Enterprise company

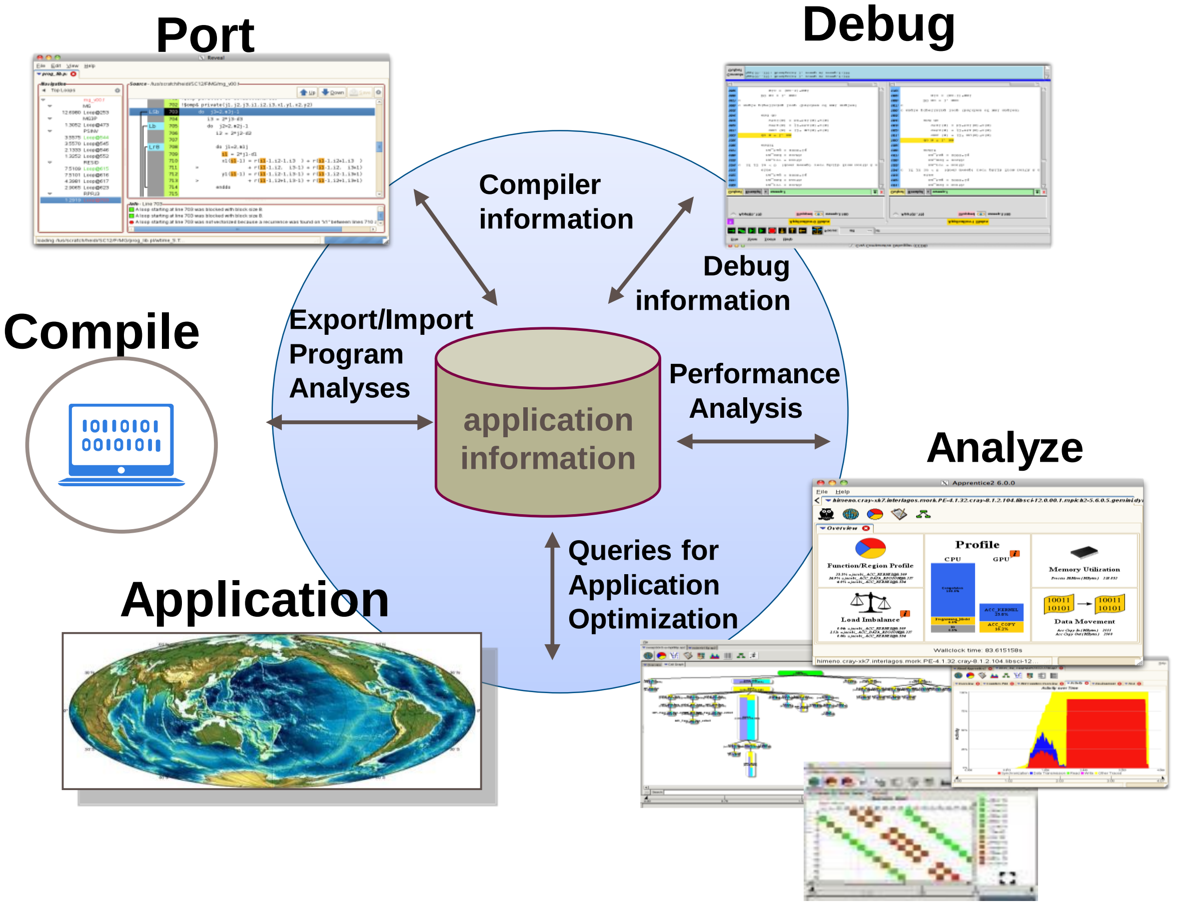
The Cray Programming Environment Mission

- Provide **scalable performance, portability, and programmability** on homogeneous and heterogeneous Cray systems

- Provide the best environment to develop, debug, analyze, and optimize applications for **production supercomputing** with **tightly coupled compilers, libraries, and tools**

- Address issues of scale and complexity of HPC systems

- Close **interaction with users** and **quick turnaround on customer issues**



Cray’s PE Vision for Accelerated Computing

- **Most important hurdle for widespread adoption of accelerated computing in HPC is programming difficulty**
  - Need a single programming model that is **portable across machine types**
    - **Portable** expression of heterogeneity and multi-level parallelism
    - Programming model and optimization should not be significantly difference for “accelerated” nodes and multi-core x86 processors
    - **Allow users to maintain a single code base**

- **Cray’s approach to Accelerator Programming is to provide an ease of use tightly coupled high level programming environment with compilers, libraries, and tools that can hide the complexity of the system**

- **Ease of use is possible with**
  - Compiler making it **feasible for users** to write applications in **Fortran, C, and C++**
  - Tools to help users port and optimize for hybrid systems
  - Auto-tuned scientific libraries

Building a Software Stack for Frontier

- ORNL, LLNL, Cray, and AMD working together to deliver full software stack
  - CORAL-2 NRE
  - Provides Compiler and library choice
  - Includes:
    - Multiple programming environments
    - Performance and correctness tools
    - Optimizations such as:
      - Cray MPI GPU-to-GPU data movement
      - libsci_acc
      - Cray PE DL Plugin
    - Compiler interoperability

Cray® Programming Environment

Cray Compiling Environment (CCE)

Reveal

AMD Programming Environment

AMD OpenMP compiler (AOCC)

Heterogeneous-compute Interface for Portability (HIP)

GNU Programming Environment

GNU Compiler Collection

Cray Message Passing Toolkit (CMPT)

Cray Performance Measurement & Analysis Tools (CPMAT)

Cray Debugger Support Tools (CDST)

Cray Scientific and Math Libraries (CSML)

Cray Environment Setup and Compiling Support (CENV)

Cray Chapel

Third Party Products

The Cray Compiling Environment (CCE)

- **Cray technology designed for real scientific applications, not just for benchmarks**
  - Arguably the most complete vectorization capabilities in the industry
    - **Full automatic loop vectorization** with **automatic outer loop vectorization**
      - no need for directives and source code modification
  - **Automatic optimizations** deliver performance for a new target through a simple recompile
  - **Compiler optimization feedback** for users with annotated listing of source code
  - **Customized for our users**

- **Fully integrated heterogeneous optimization capability**

- **Fully Integrated and optimized PGAS**

- **Support multiple platforms**
  - **X86 (Intel or AMD)**
  - **ARM SVE+HBM (when available)**
    - **NSP-1 and ThunderX**
  - **AMD GPUs**

C++ 17 | **Fortran 2018** | **OpenMP 5.0**

C11 | **UPC 1.3**

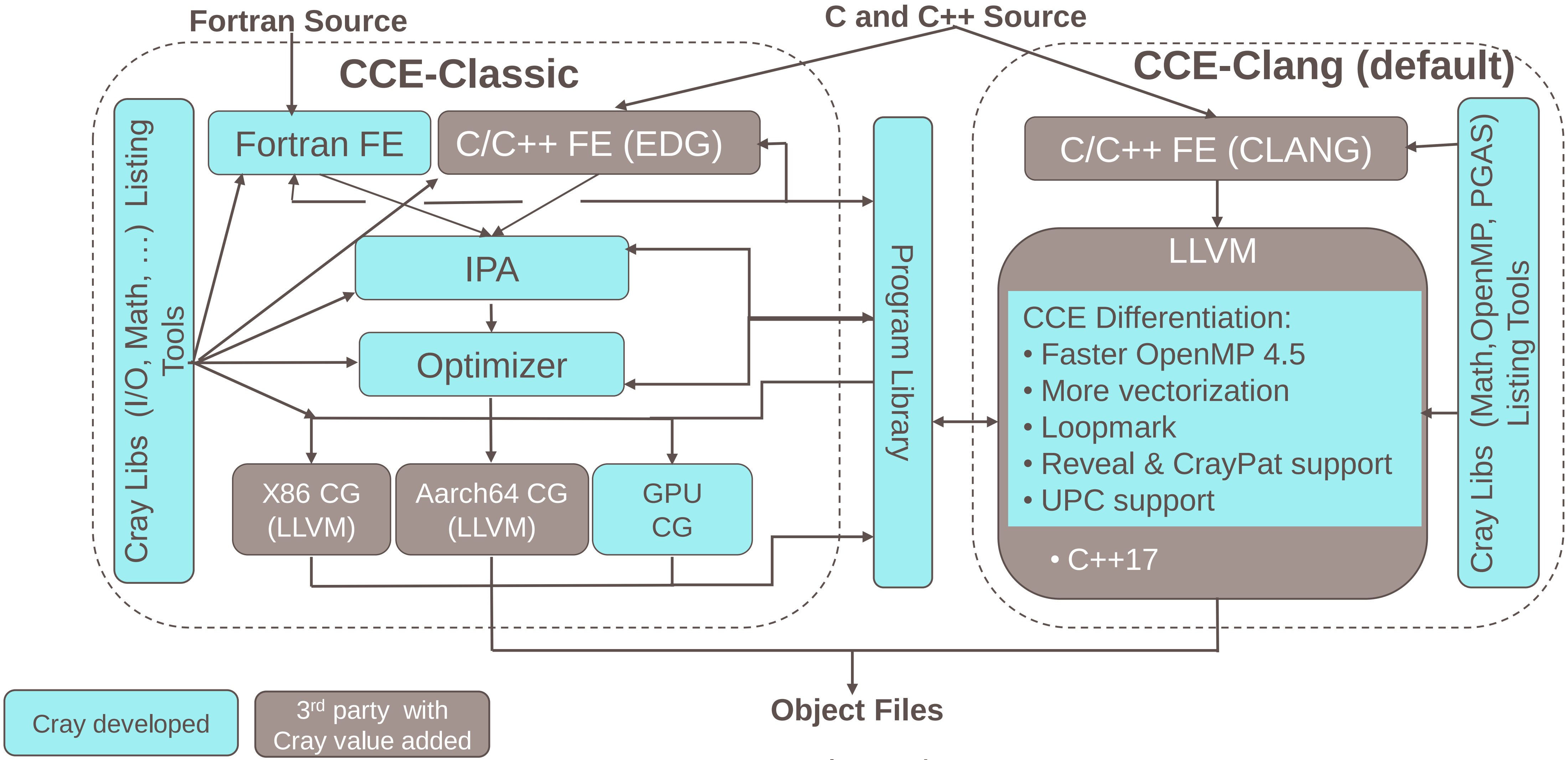
State of Cray's Existing C++ Compiler

- **C++ Trends**
  - C++ usage in HPC has accelerated
  - C++ language changes have accelerated too
    - Standards: 1998, 2003, 2011, 2014, 2017, 2020, …
    - C++20 is expected to be a major change like C++11
  - **Clang leads C++ compilers in standards support and diagnostics**

- **Customer's view**
  - Compile times are slow, especially when templates are used
  - C++11 was provided late (2015), whereas other compilers had full C++11 support earlier (2013) and experimental support before it was ratified

- **Cray's view**
  - Significantly more pressure to have a competitive C++ compiler!
    - Difficult to reduce compile time while maintaining same performance
    - Dependent on EDG schedule for supporting new standards
    - Lots of developer time spent maintaining non-differentiating passes common to all compilers

Cray Compiling Environment – CCE 9.0



CCE 9.0 released on June 2019

Use ***$module swap cce cce/9.0-classic*** to change compilers

CCE-Clang: What Will Be the Impact on Users?

- **Cray will track Clang closely**
  - Weekly or biweekly merges

- **CCE major releases will follow Clang releases (about 2 months lag)**
  - CCE will have more frequent major releases
  - Will follow the same version number (almost coincidentally)

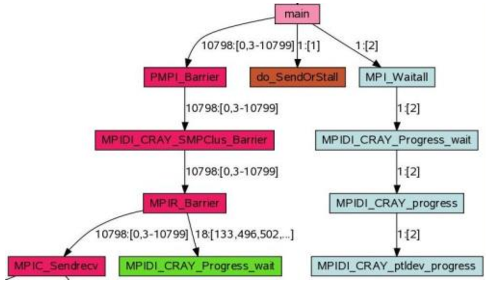
- **Cray-Clang is not Clang**
  - Cray will support Cray-Clang

- **What steps, if any, are we taking to open source programming environment tools like compilers, debuggers and profilers for existing and future accelerators?**

Scalable Debugging on Cray Systems

- Systems with thousands of threads of execution need a new debugging paradigm
  - Cray's focus is to build tools around traditional debuggers with innovative techniques for productivity and scalability
  - Scalable Solutions based on MRNet from University of Wisconsin

- **STAT - Stack Trace Analysis Tool**
  - Scalable generation of a single merged stack backtrace for the application
  - GUI based tool (stat-gui/stat-view) along with cli tools (stat-cl)
  - Gain insight into application behavior at a function level



- **ATP - Abnormal Termination Processing**
  - Scalable core file generation and analysis when application crashes
  - Generates a merged stack backtrace akin to stat
  - Selection algorithm to dump unique core files

- **gdb4hpc**
  - Conventional CLI based interactive parallel debugger
  - Look and feel of gdb – syntax is inspired by gdb!
  - Debug your application at scale

- **CCDB - Comparative debugging**
  - **A data-centric paradigm**
  - Compare two applications side-by-side
    - Focus on the data – not state and internal operations
  - GUI tool that interacts with gdb4hpc

- **Valgring4hpc**
  - Parallel valgrind based debugging tool (memcheck)
  - Aids in detection of memory leaks and errors in parallel applications
  - Aggregates like errors across PEs/threads

Cray Performance Analysis Tools on GPUs

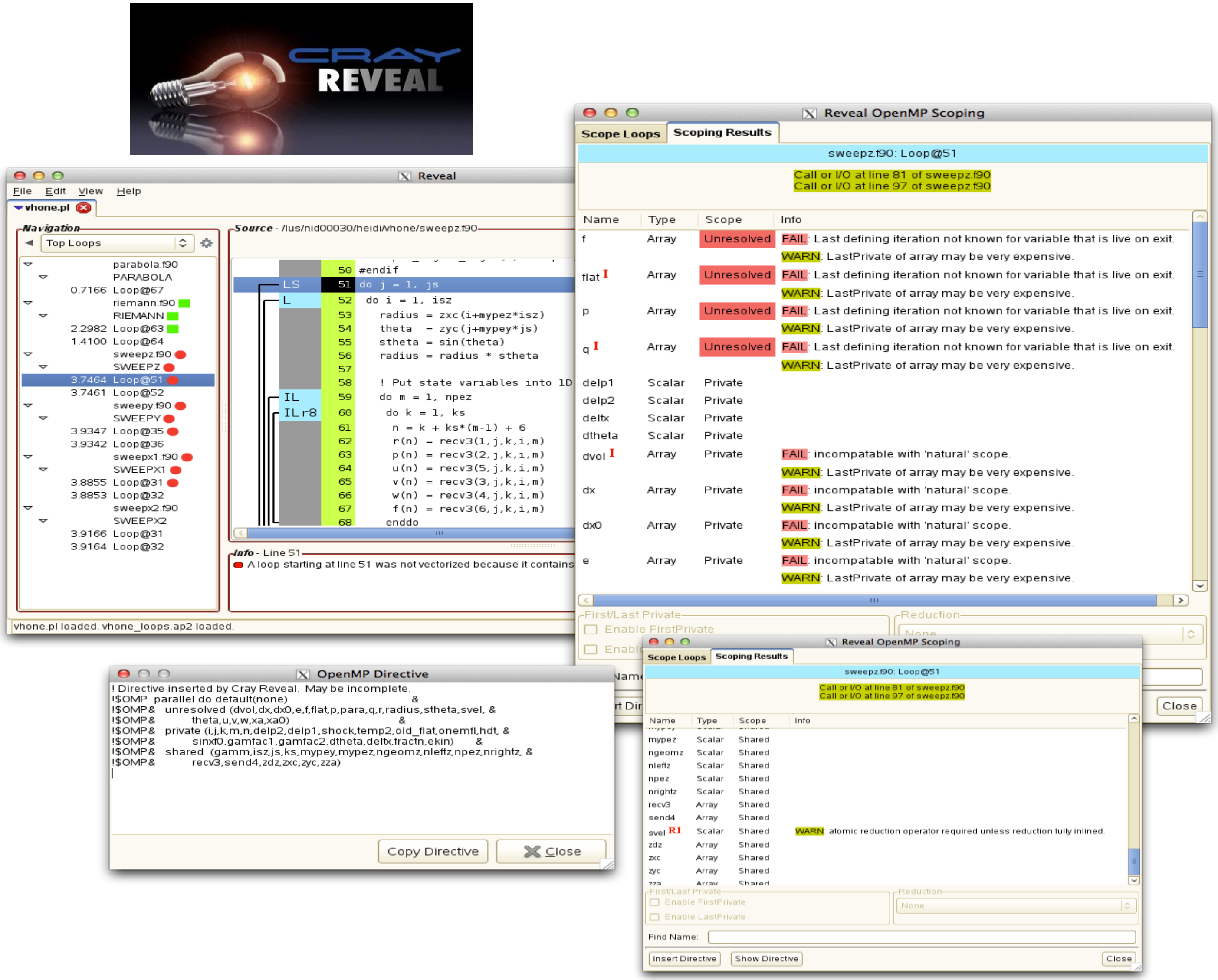
a Hewlett Packard Enterprise company

Cray Performance Tools profiled production applications with over 256,000 ranks

- **Provide whole program performance analysis**
  - Single tool for CPU and GPU performance analysis
    - Single report can include **statistics for both the host and the accelerator**
    - Performance statistics includes accelerator time, host time, and amount of data copied to/from the accelerator
- Performance **statistics mapped back** to the user **source** by line number
  - Performance statistics grouped by accelerator directive
- **Scaling** (running big jobs with a large number of GPUs)
  - **Results** summarized and **consolidated** in one place

Simplifying the Parallelization Task with Reveal

a Hewlett Packard Enterprise company



- **Reduce effort** associated with adding OpenMP to MPI programs
- **Identify parallelization and scoping issues**
- Get **feedback on issues down the call chain** (shared reductions, etc.)
- Get **insight into optimizations** performed by the Cray compiler
- Track requests to memory and evaluate the bandwidth contribution of objects within a program

How About OpenACC?

- **Are there any large applications in production with OpenMP for accelerators?**

- **How are we going to support existing OpenACC codebases, when we consider that an automatic tool will struggle to convert hundreds of thousands of lines of Fortran full of hand-written, multi-gen OpenACC directives to OpenMP?**

- **We are seeing that CUDA codes are more portable to AMD GPUs via HIP than OpenACC codes. What does this tell us about the purported portability benefits of directives?**

Developer Environment for Frontier

a Hewlett Packard Enterprise company

Programming Languages

Languages
- Fortran
- C
- C++
- Chapel
- Python 3
- R

Programming Models

Distributed Memory
- Cray MPI SHMEM

Shared Memory / GPU
- OpenMP MPI GPU2GPU
- HIP / CUDA

PGAS & Global View
- UPC Fortran coarrays Coarray C++ Chapel
- GA

Programming Environments

PrgEnv-
- Cray Compiling Environment PrgEnv-cray
- GNU PrgEnv-gnu
- 3rd Party compilers PrgEnv-AMD

Optimized Libraries

Scientific Libraries
- BLAS
- LAPACK
- ScaLAPACK
- Iterative Refinement Toolkit
- libsci_acc
- FFTW

I/O Libraries
- NetCDF
- HDF5
- ADIOS

Tools

Environment setup
- Modules / Lmod
- Tool Enablement (Spack, CMake, EasyBuild, etc)

Performance Analysis
- CrayPAT
- Cray Apprentice²

Porting
- Reveal
- CCDB

Tools (continued)

Debuggers
- gdb4hpc
- TotalView
- DDT

Debugging Support
- Valgrind4hpc
- STAT
- Abnormal Termination Processing (ATP)

Tools Interface
- CTI

Analytics / AI

AI Toolboxes
- Cray Urika AI - Analytics
- Chapel AI

DL Frameworks
- Cray PE DL Scalability Plugin

Cray Developed | Cray added value to 3rd party | 3rd party packaging | Licensed ISV SW

Summary

- **Cray/HPE will provide a high level programming environment for ORNL Frontier**
  - Fortran, C, and C++ compilers
    - **OpenMP Target directives**
    - Compiler optimizations to take advantage of accelerator and multi-core X86 hardware appropriately
    - Support for new programming frameworks, such as **RAJA and Kokkos**
  - Cray **Reveal**
    - **Scoping analysis** tool to assist user in understanding their code and taking full advantage of SW and HW system
  - **Cray Performance Measurement and Analysis toolkit**
    - Single tool for GPU and CPU performance analysis with statistics for the whole application
  - **Parallel debugger support** with allinea DDT, Rogue Wave TotalView, or **Cray gdb4hpc** & **CCDB**
  - optimized on-node, GPU-GPU data movement operations and data movement operations between CPU- and GPU-attached memory regions using the ROCm interface directly
  - Auto-tuned Scientific Libraries support
    - Getting performance from the system … **no assembly required**

FORWARD LOOKING STATEMENTS

This presentation may contain forward-looking statements that involve risks, uncertainties and assumptions. If the risks or uncertainties ever materialize or the assumptions prove incorrect, the results of Hewlett Packard Enterprise Company and its consolidated subsidiaries ("Hewlett Packard Enterprise") may differ materially from those expressed or implied by such forward-looking statements and assumptions. All statements other than statements of historical fact are statements that could be deemed forward-looking statements, including but not limited to any statements regarding the expected benefits and costs of the transaction contemplated by this presentation; the expected timing of the completion of the transaction; the ability of HPE, its subsidiaries and Cray to complete the transaction considering the various conditions to the transaction, some of which are outside the parties' control, including those conditions related to regulatory approvals; projections of revenue, margins, expenses, net earnings, net earnings per share, cash flows, or other financial items; any statements concerning the expected development, performance, market share or competitive performance relating to products or services; any statements regarding current or future macroeconomic trends or events and the impact of those trends and events on Hewlett Packard Enterprise and its financial performance; any statements of expectation or belief; and any statements of assumptions underlying any of the foregoing. Risks, uncertainties and assumptions include the possibility that expected benefits of the transaction described in this presentation may not materialize as expected; that the transaction may not be timely completed, if at all; that, prior to the completion of the transaction, Cray's business may not perform as expected due to transaction-related uncertainty or other factors; that the parties are unable to successfully implement integration strategies; the need to address the many challenges facing Hewlett Packard Enterprise's businesses; the competitive pressures faced by Hewlett Packard Enterprise's businesses; risks associated with executing Hewlett Packard Enterprise's strategy; the impact of macroeconomic and geopolitical trends and events; the development and transition of new products and services and the enhancement of existing products and services to meet customer needs and respond to emerging technological trends; and other risks that are described in our Fiscal Year 2018 Annual Report on Form 10-K, and that are otherwise described or updated from time to time in Hewlett Packard Enterprise's other filings with the Securities and Exchange Commission, including but not limited to our subsequent Quarterly Reports on Form 10-Q. Hewlett Packard Enterprise assumes no obligation and does not intend to update these forward-looking statements.

QUESTIONS?

CRAY®

a Hewlett Packard Enterprise company