The Cray Programming Environment

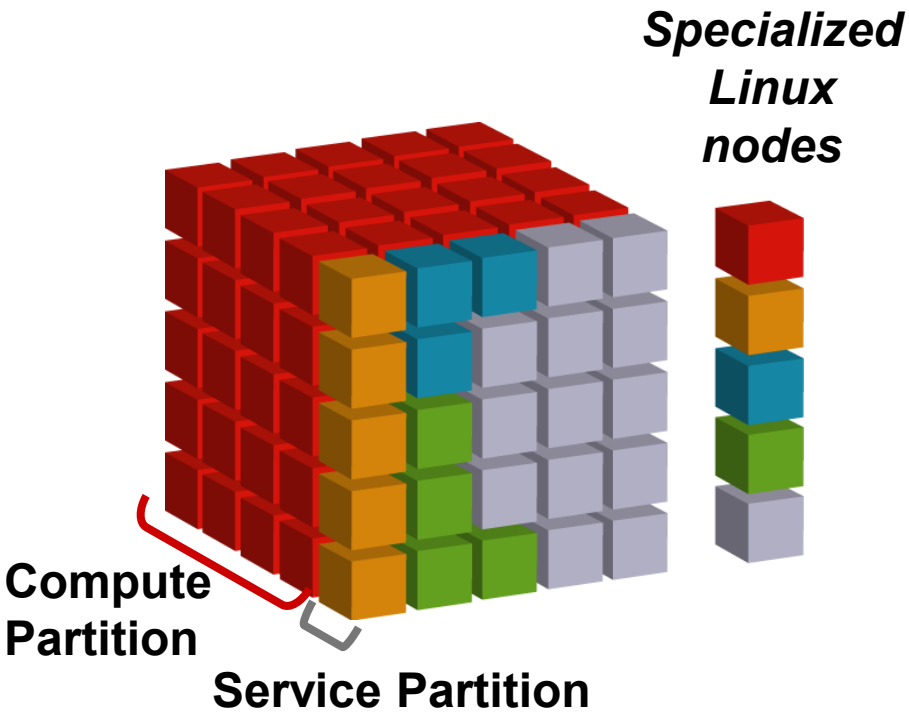
An Introduction

Vision

- **Cray systems are designed to be High Productivity as well as High Performance Computers**
- **The Cray Programming Environment (PE) provides a simple consistent interface to users and developers.**
  - Focus on improving scalability and reducing complexity
- **The default Programming Environment provides:**
  - the highest levels of application performance
  - a rich variety of commonly used tools and libraries
  - a consistent interface to multiple compilers and libraries
  - an increased automation of routine tasks
- **Cray continues to develop and refine the PE**
  - Frequent communication and feedback to/from users
  - Strong collaborations with third-party developers

Scalable Software Architecture

Scalable Software Architecture: CLE



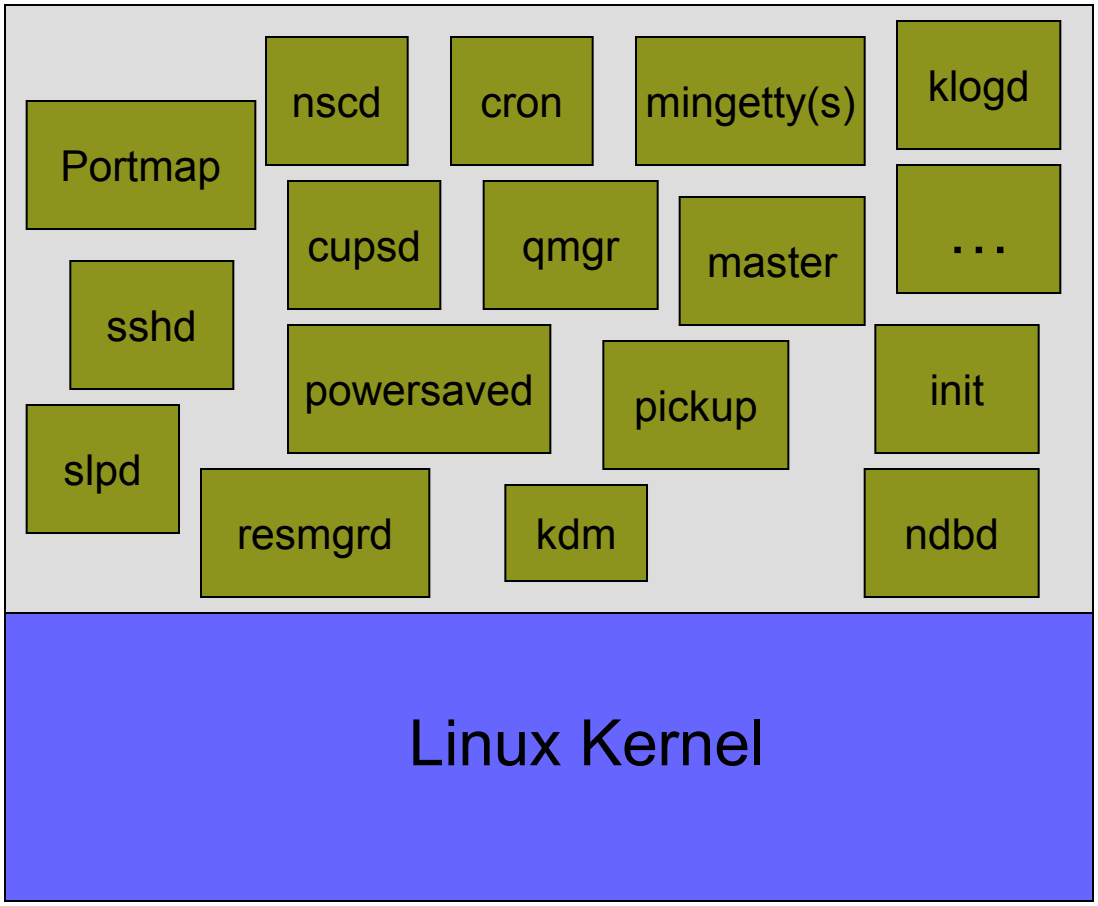
Microkernel on Compute nodes, full featured Linux on Service nodes.

Service PEs specialize by function

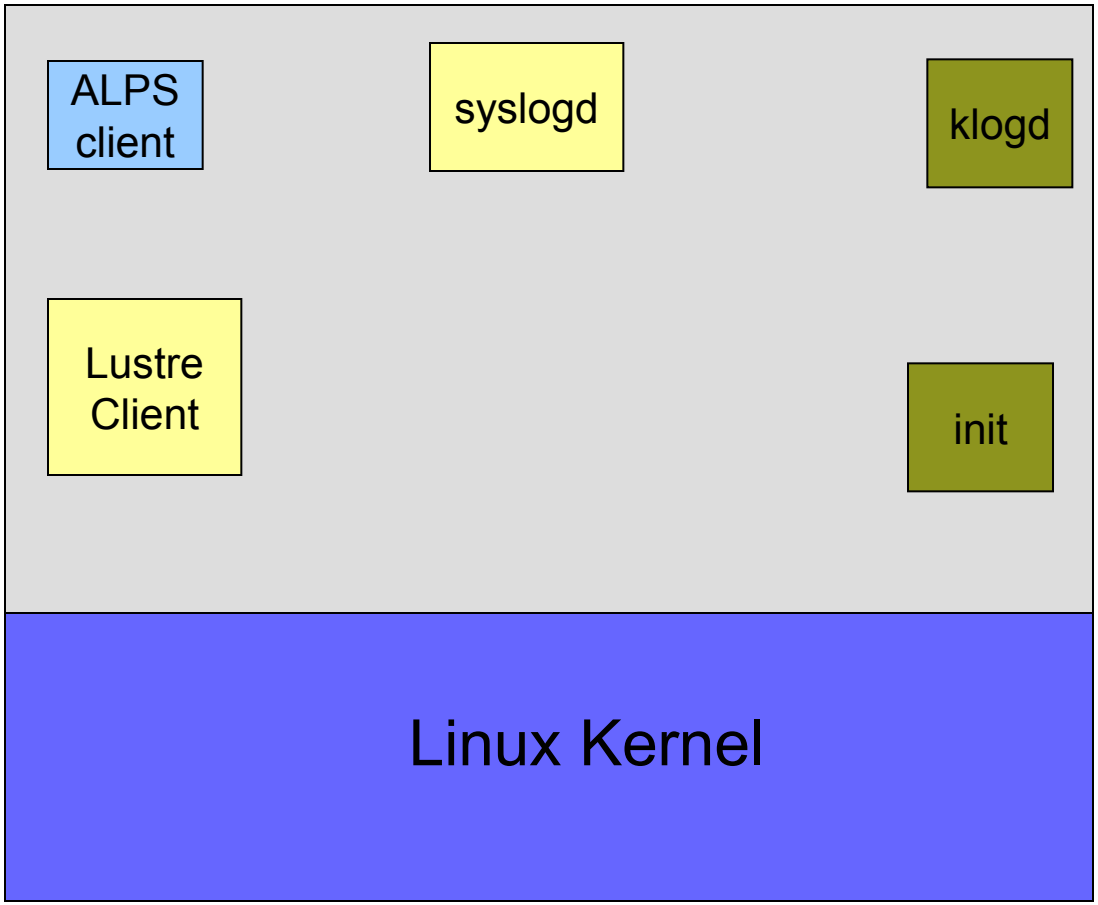
Software Architecture eliminates OS "Jitter"

Software Architecture enables reproducible run times

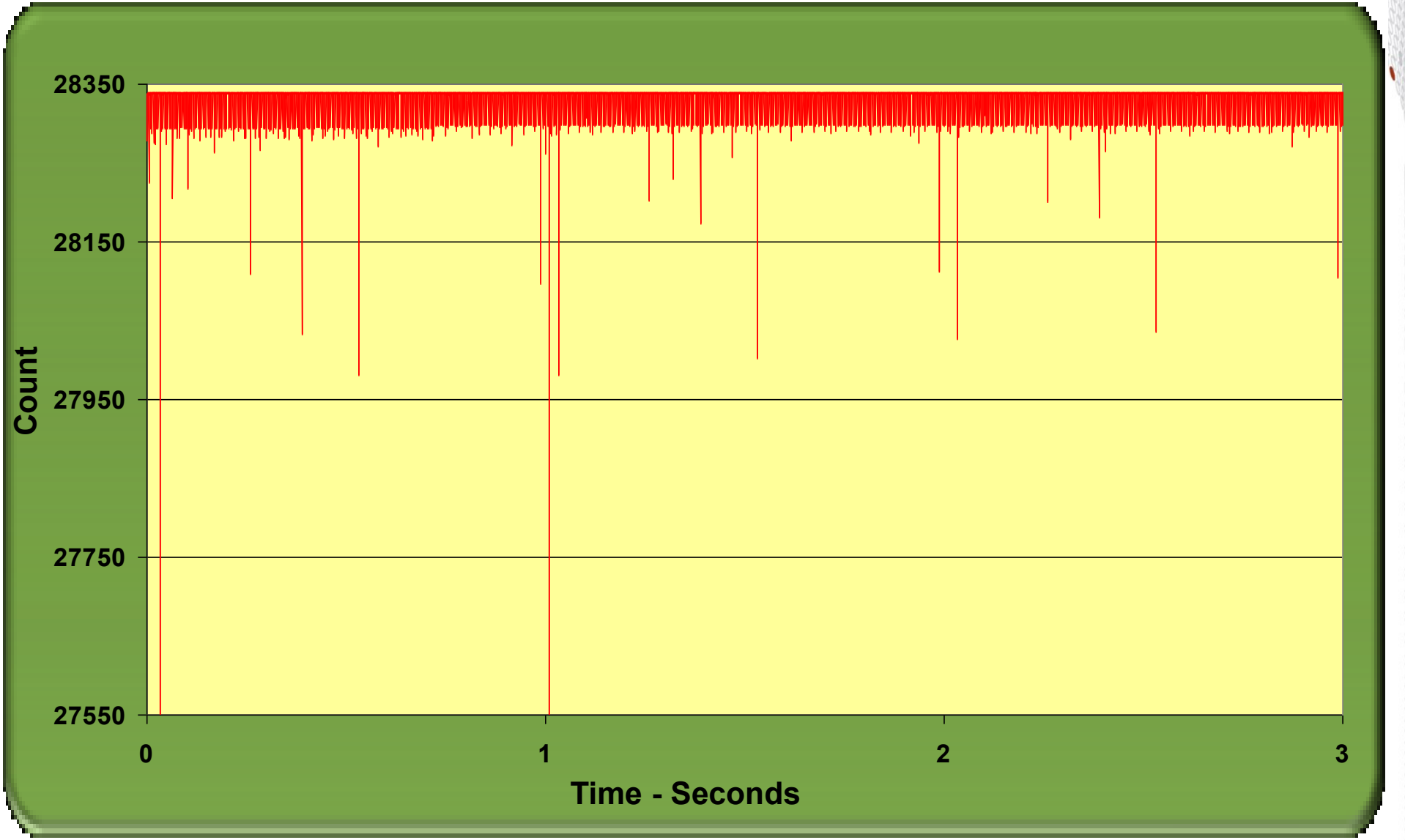
Trimming OS – *Standard Linux Server*



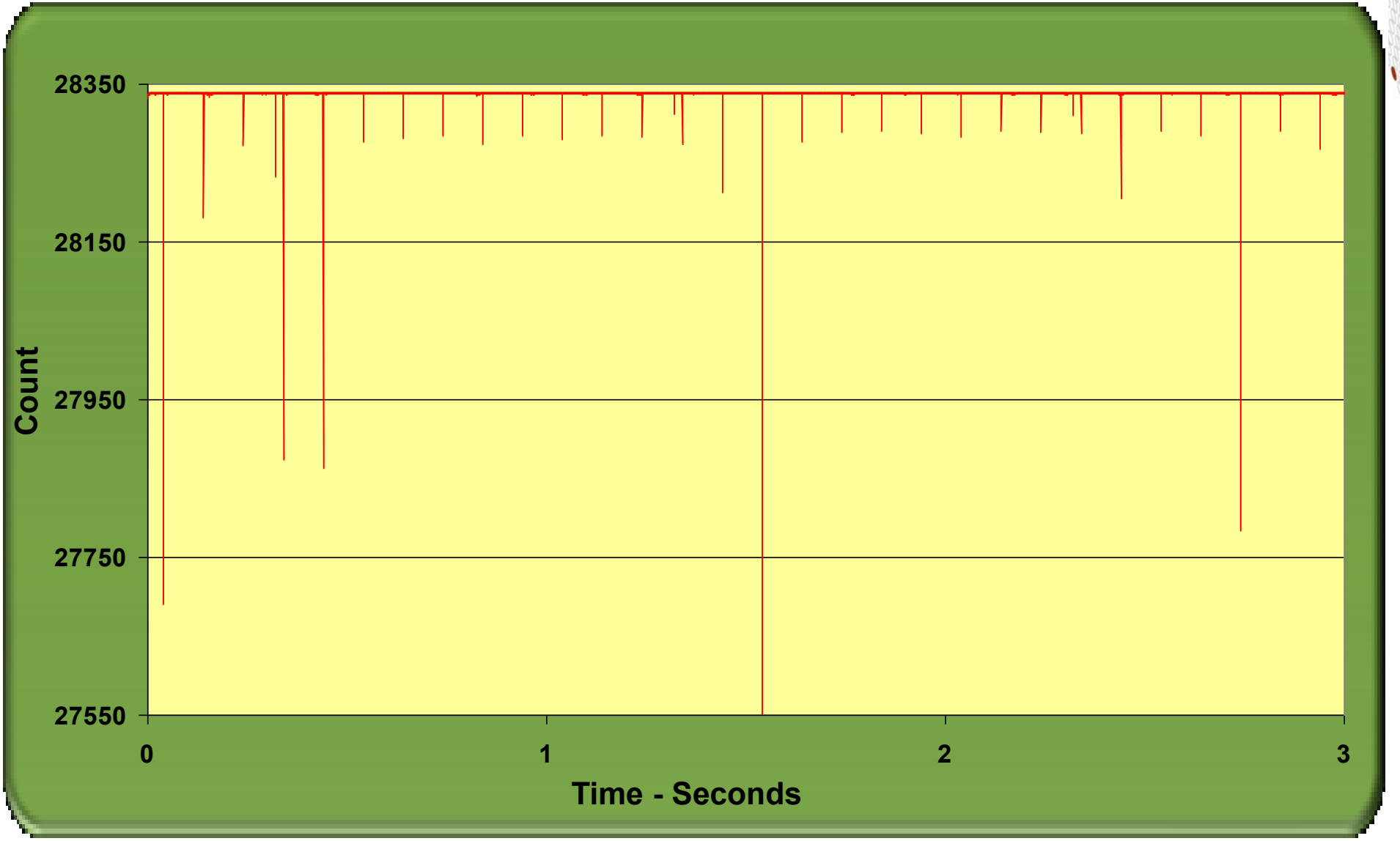
Linux on a Diet – *CLE*



FTQ Plot of Stock SuSE (most daemons removed)



FTQ plot of CNL



Cray Software

Cray's Supported Programming Environment

Programming Languages

- Fortran
- C
- C++
- Python

Programming models

- Distributed Memory (Cray MPT)
  - MPI
  - SHMEM
- Shared Memory
  - OpenMP 3.0
  - OpenACC
- PGAS & Global View
  - UPC (CCE)
  - CAF (CCE)
  - Chapel

Compilers

- Cray Compiling Environment (CCE)
- GNU
- 3rd Party Compilers
  - Intel Composer
  - PGI

Tools

Environment setup
- Modules

Debuggers
- Allinea (DDT)
- lgdb

Debugging Support Tools
- Abnormal Termination Processing
- STAT

Performance Analysis
- CrayPat
- Cray Apprentice²

Scoping Analysis
- Reveal

Optimized Scientific Libraries

- LAPACK
- ScaLAPACK
- BLAS (libgoto)
- Iterative Refinement Toolkit
- FFTW
- Cray PETSc (with CASK)
- Cray Trilinos (with CASK)

I/O Libraries

- NetCDF
- HDF5

Cray developed
Licensed ISV SW
3rd party packaging
Cray added value to 3rd party

The Cray Compilation Environment (CCE)

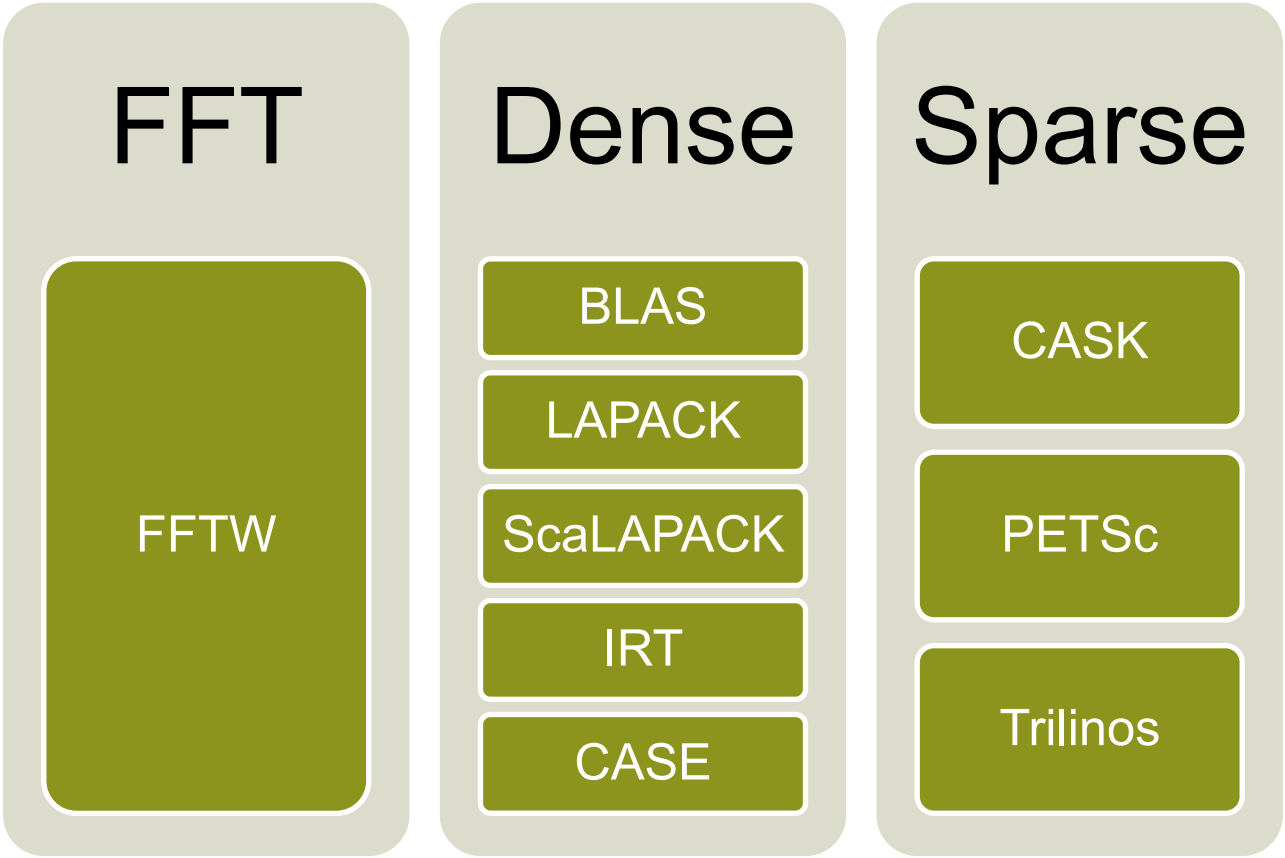
- **The default compiler on XE and XC systems**
  - Specifically designed for HPC applications
  - Takes advantage of Cray's experience with automatic vectorization and shared memory parallelization
- **Excellent standards support for multiple languages and programming models**
  - Fortran 2008 standards compliant
  - C++98/2003 compliant, working on C++11
  - OpenMP 3.1 compliant, working on OpenMP 4.0
  - OpenACC 2.0 compliant

- **Full integrated and optimised support for PGAS languages**
  - UPC 1.2 and Fortran 2008 coarray support
  - No preprocessor involved
  - Full debugger support (With Allinea DDT)
- **OpenMP and automatic multithreading fully integrated**
  - Share the same runtime and resource pool
  - Aggressive loop restructuring and scalar optimization done in the presence of OpenMP
  - Consistent interface for managing OpenMP and automatic multithreading

Cray MPI & SHMEM

- **Cray MPI**
  - Implementation based on MPICH3 source from ANL
  - Includes many improved algorithms and tweaks for Cray hardware
    - Improved algorithms for many collectives
    - Asynchronous progress engine allows overlap of computation and comms
    - Customizable collective buffering when using MPI-IO
    - Optimized Remote Memory Access (one-sided) fully supported including passive RMA
  - Full MPI-3 support with the exception of
    - Dynamic process management (eg. MPI_Comm_spawn)
    - MPI_LONG_DOUBLE and MPI_C_LONG_DOUBLE_COMPLEX for CCE
  - Includes support for Fortran 2008 bindings (from CCE 8.3.3)
- **Cray SHMEM**
  - Fully optimized Cray SHMEM library supported
    - Fully compliant with OpenSHMEM v1.0
    - Cray XC implementation close to the T3E model

Cray Scientific Libraries



IRT – Iterative Refinement Toolkit
CASK – Cray Adaptive Sparse Kernels
CASE – Cray Adaptive Simplified Eigensolver

Cray Performance Analysis Tools (PAT)

- **From performance measurement to performance analysis**
- **Assist the user with application performance analysis and optimization**
  - Help user identify important and meaningful information from potentially massive data sets
  - Help user identify problem areas instead of just reporting data
  - Bring optimization knowledge to a wider set of users
- **Focus on ease of use and intuitive user interfaces**
  - Automatic program instrumentation
  - Automatic analysis
- **Target scalability issues in all areas of tool development**

Debuggers on Cray Systems

- **Systems with hundreds of thousands of threads of execution need a new debugging paradigm**
  - 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 tree
        - running at 216K back-end processes
      - **ATP** - Abnormal Termination Processing
        - Scalable analysis of a sick application, delivering a STAT tree and a minimal, comprehensive, core file set.
    - **Fast Track Debugging**
      - Debugging optimized applications
      - Added to Allinea's DDT 2.6 (June 2010)
    - **Comparative debugging**
      - A data-centric paradigm instead of the traditional control-centric paradigm
      - Collaboration with Monash University and University of Wisconsin for scalability
  - Support for traditional debugging mechanism
    - DDT, gdb, and TotalView

An introduction to modules

What are Environment Modules?

- The Environment Modules package provides for the dynamic modification of a user's environment via modulefiles.
- Each modulefile contains the information needed to configure the shell for an application.
  Typically modulefiles instruct the module command to alter or set shell environment variables such as PATH, MANPATH, etc.
- Modules can be **loaded** and **unloaded** dynamically and atomically, in an clean fashion. All popular shells are supported, including *bash*,*ksh*, *zsh*, *sh*, *csh*, *tcsh*, as well as some scripting languages such as *perl* and *python*.
- Modules are useful in managing different versions of applications. Modules can also be bundled into **metamodules** that will load an entire suite of different applications
- Check **http://modules.sourceforge.net/**

Environment Setup

- **The Cray XC system uses modules in the user environment to support multiple software versions and to create integrated software packages**
  - As new versions of the supported software and associated man pages become available, they are added automatically to the Programming Environment as a new version, while earlier versions are retained to support legacy applications
  - You can use the default version of an application, or you can choose another version by using Modules system commands

The module tool on the Cray XC

- **How can we get appropriate Compiler, Tools, and Libraries?**
  - The modules tool is used to handle different versions of packages
    - e.g.: module load compiler_v1
    - e.g.: module swap compiler_v1 compiler_v2
    - e.g.: module load perftools
- **Taking care of changing of PATH, MANPATH, LM_LICENSE_FILE,.... environment**
  - Modules also provide a simple mechanism for updating certain environment variables, such as PATH, MANPATH, and LD_LIBRARY_PATH
  - In general, you should make use of the modules system rather than embedding specific directory paths into your startup files, makefiles, and scripts.
- **It is also easy to setup your own modules for your own software**

Useful module commands

- **Which modules are available?**
  - module avail, module avail cce
- **Which modules are loaded?**
  - module list
- **Load software**
  - module load perftools
- **Change programming environment**
  - module swap PrgEnv-cray PrgEnv-gnu
- **Change software version**
  - module swap cce/8.0.2 cce/7.4.4

Which SW Products and Versions Are Available

- **avail [avail-options] [path...]**
  - List all available modulefiles in the current MODULEPATH
- **Useful options for filtering**
  - -U, --usermodules
    - List all modulefiles of interest to a typical user
  - -D, --defaultversions
    - List only default versions of modulefiles with multiple available versions
  - -P, --prgenvmodules
    - List all PrgEnv modulefiles
  - -L, --librarymodules
    - List all library modulefiles
  - % **module avail <product>**
    - List all <product> versions available

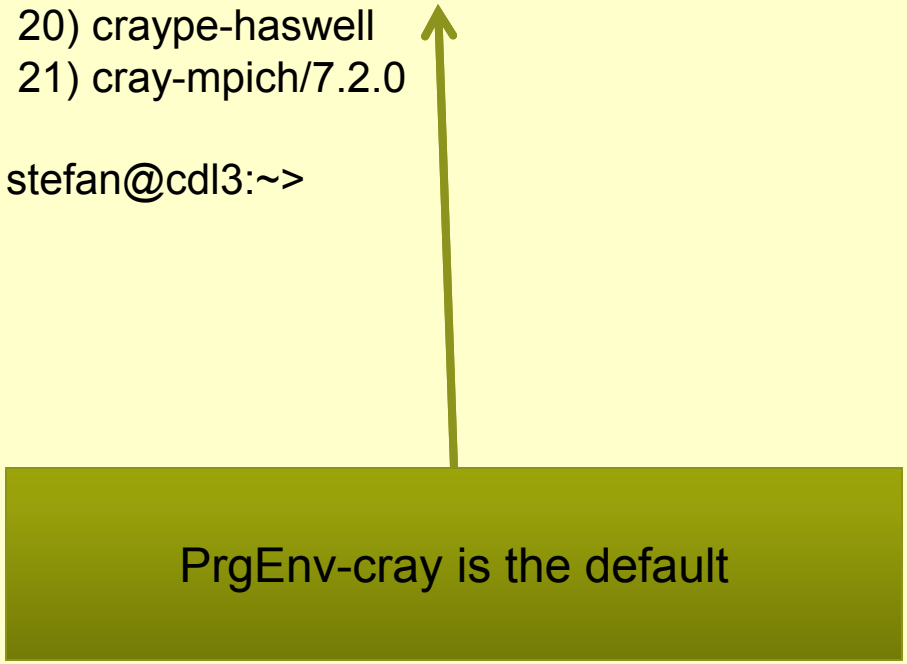
      Note : no real 'regular expressions possible, but leading characters is possible

      > module avail Prg
      ---------- /opt/cray/modulefiles ----------
      PrgEnv-cray/5.2.40(default) PrgEnv-intel/5.2.40(default)
      PrgEnv-gnu/5.2.40(default)

Default module list at KAUST Shaheen II

```
stefan@cdl3:~> module list
Currently Loaded Modulefiles:
  1) modules/3.2.10.3
  2) eswrap/1.1.0-1.020200.1231.0
  3) switch/1.0-1.0502.54233.2.96.ari
  4) craype-network-aries
  5) craype/2.3.0
  6) cce/8.3.10
  7) cray-libsci/13.0.3
  8) udreg/2.3.2-1.0502.9275.1.12.ari
  9) ugni/5.0-1.0502.9685.4.24.ari
 10) pmi/5.0.6-1.0000.10439.140.2.ari
 11) dmapp/7.0.1-1.0502.9501.5.219.ari
 12) gni-headers/3.0-1.0502.9684.5.2.ari
 13) xpmem/0.1-2.0502.55507.3.2.ari
 14) job/1.5.5-0.1_2.0502.54585.3.66.ari
 15) dvs/2.5_0.9.0-1.0502.1873.1.145.ari
 16) alps/5.2.1-2.0502.9041.11.6.ari
 17) rca/1.0.0-2.0502.53711.3.127.ari
 18) atp/1.8.1
 19) PrgEnv-cray/5.2.40
 20) craype-haswell
 21) cray-mpich/7.2.0

stefan@cdl3:~>
```



“Meta”-Module PrgEnv-X

PrgEnv-X is a “meta”-module loading several modules, including the compiler, the corresponding mathematical libs, MPI, system setup needed for the compiler wrappers

```
> module show PrgEnv-cray
-------------------------------------------------------------------
/opt/cray/modulefiles/PrgEnv-cray/5.2.40:

conflict        PrgEnv
conflict        PrgEnv-gnu
conflict        PrgEnv-intel
…
setenv          PE_ENV CRAY
setenv          XTOS_VERSION 5.2.40
setenv          CRAYOS_VERSION 5.2.40
setenv          CRAY_PE_TARGET x86-64
prepend-path    PE_PRODUCT_LIST CRAY
module          swap craype/2.2.1
module          load cce/8.3.8
module          load cray-libsci
module          load dmapp/7.0.1-1.0502.9501.5.219.ari
module          load gni-headers/3.0-1.0502.9684.5.2.ari
module          load alps/5.2.1-2.0502.9041.11.6.ari
module          load rca/1.0.0-2.0502.53711.3.127.ari
module          load atp
…
setenv          CRAY_PRGENVCRAY loaded
-------------------------------------------------------------------
```