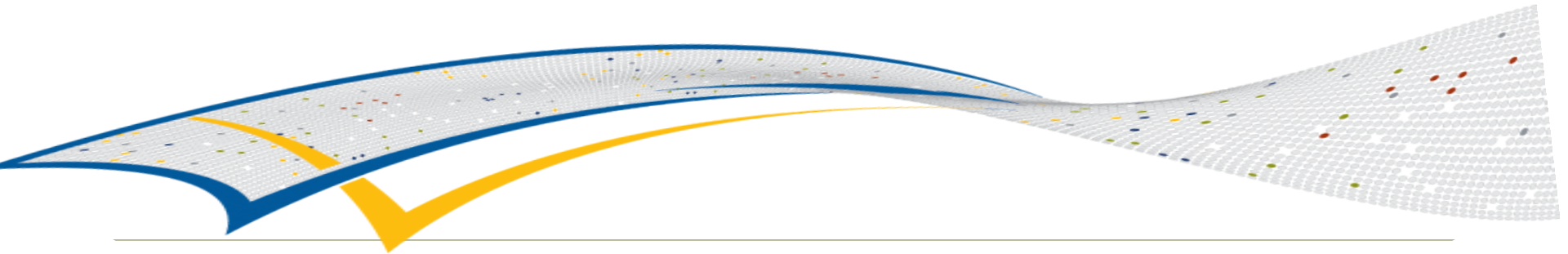
Expanding Stack Coverage In HPC Systems

David Race



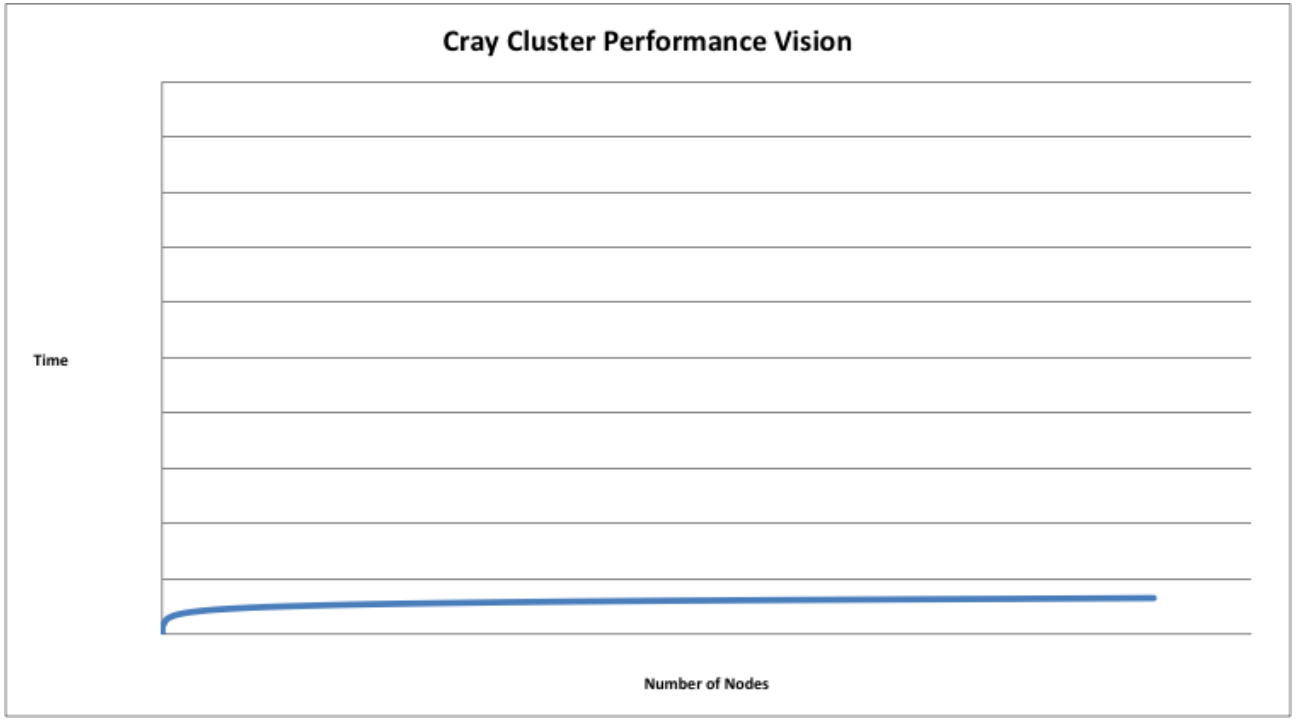
Safe Harbor Statement

This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts. These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.

Cray Is Scalability

Admin

Monitoring



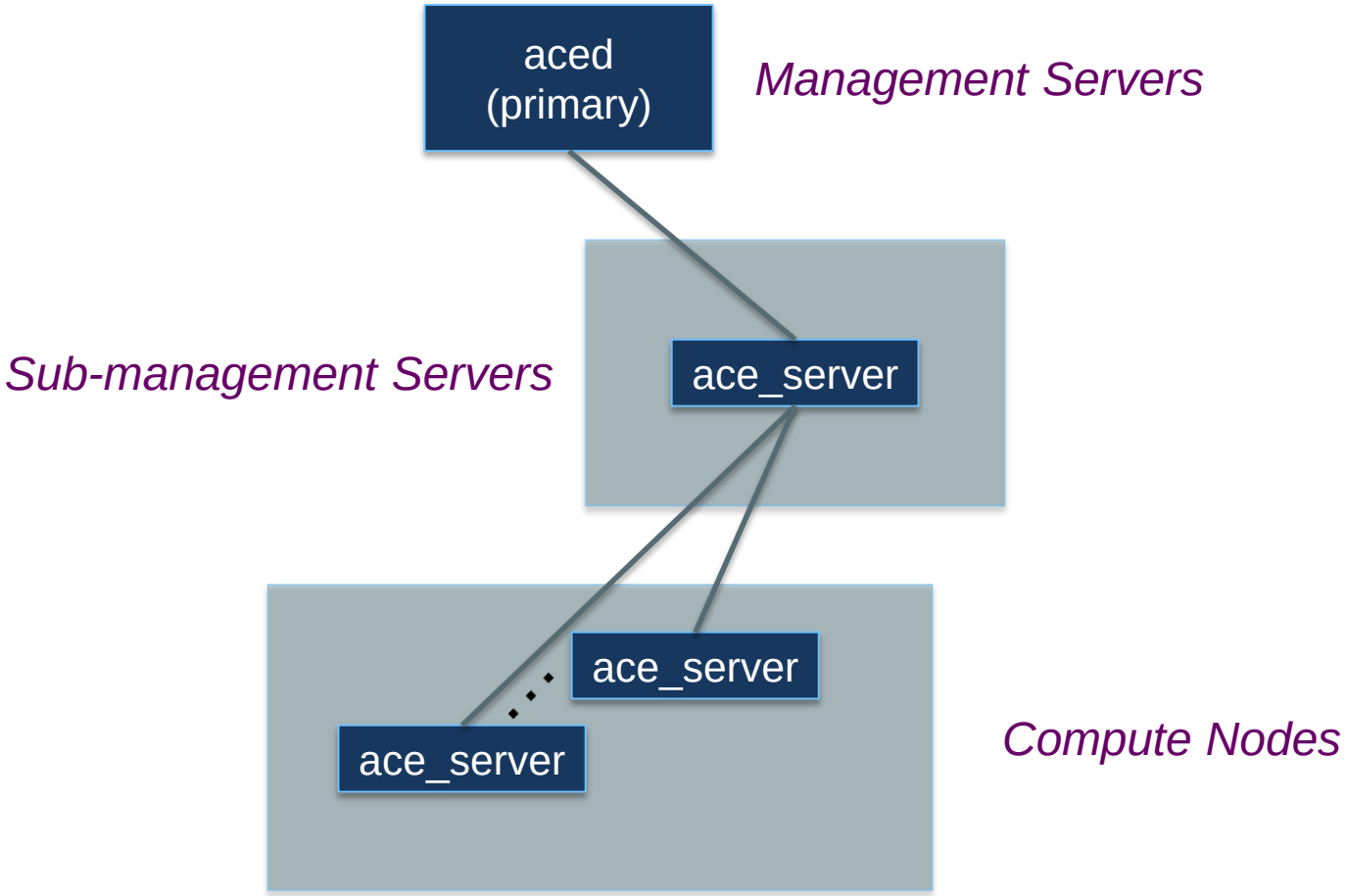
Runtime

Support

KEY VECTORS

Cluster should “feel” like a linux workstation!

ACE Foundation Architecture



Cray Cluster Software Stack

HPC Programming Tools	**Development & Performance Tools**	**Cray® PE on CS ***	Intel® Cluster Studio	PGI Cluster Development Kit	GNU toolchain	CUDA
	Application Libraries	**Cray® LibSci, LibSci_ACC**	Intel® MPI	Platform MPI	MVAPICH2	OpenMPI
	Debuggers	Rogue Wave TotalView	Allinea DDT, MAP	Intel IDB	PGI PGDBG	GNU GDB
Schedulers File Systems and Management	**Resource Management / Job Scheduling**	SLURM	Adaptive Computing: MOAB / Maui / Torque	Altair PBSPro	IBM Platform LSF	Grid Engine
	File Systems	Lustre	NFS	GPFS	PanFS	Local (ext3, ext4, XFS)
	Cluster Management	**Cray® Advanced Cluster Engine (ACE™) Management Software****				
Operating Systems and Drivers	**Drivers & Network Mgmt.**	Accelerator software stack & drivers			OFED™	
	Operating Systems	Linux (RedHat, CentOS, SUSE) **				

Stack Components Need To Be Integrated Into A Scalable Base

MVAPICH2 Challenges

- **Common Runtime**
  - Job Startup
    - Needs to be flat and fast for all programming models
    - Needs to leverage the hierarchical environment
    - Obtain communication characteristics from the scheduler
      - Topology aware scheduling is already available
  - Expanding the programming model to target exascale
    - How many programming models can leverage a common runtime?
  - Too many optimizations are only available to one part of the stack
    - How can the admin/support leverage the runtime environment?
    - How far can the common runtime extend?
      - Hadoop for HPC
      - Next generation analysis
  - Open OFED made the stack easier
    - How do we handle the burgeoning number of versions of OFED
    - Lustre + OFED creates a large challenge – almost nightmarish
      - Shared storage + different interconnects + Lustre causes an explosion of versions that have to work together
      - Once a Lustre server is working, it is never touched
- **Performance Additions Environments**
  - Custom common runtime algorithms for particular installations
    - How many have used the PMPI interface for custom algorithms?
      - This causes problems with profiling tools
    - Adding value with special runtime algorithms, but keeping the core of the mvapich2
  - Library development tools that leverage the cluster dynamic linking
    - Eclipse tools for developing value added algorithms
- **Is it time to begin having "standardized" runtime environments beyond PMPI and MPIT?**

Other Challenges For Required Improvements

- **Rather Obvious Challenges**
  - Leveraging the Advantages of Linux
    - Dynamic Linking
    - Enhanced Programming Environments Beyond vi☺
  - Analysis of Data
    - Bringing “Google-like” analysis to HPC Clusters
  - Expanding analytic models to access HPC Data
    - Structured Data
    - Time Series 3D Data
    - Common storage
    - Old data storage locations run old versions of Lustre

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, URIKA, and YARCDATA. The following are trademarks of Cray Inc.: ACE, APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, THREADSTORM. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.