NATURAL ENVIRONMENT RESEARCH COUNCIL

EPSRC

ARCHER Hardware

Overview and Introduction

Slides contributed by Cray and EPCC

epcc

THE UNIVERSITY OF EDINBURGH

Nodes: The building blocks

The Cray XC30 is a Massively Parallel Processor (MPP) supercomputer design. It is therefore built from many thousands of individual nodes.

There are two basic types of nodes in any Cray XC30:

- Compute nodes
  - These only do user computation and are always referred to as “Compute nodes”
- Service/Login nodes
  - These provide all the additional services required for the system to function, and are given additional names depending on their individual task:
    - Login nodes – allow users to log in and perform interactive tasks
    - PBS Mom nodes – run and managing PBS batch scripts
    - Service Database node (SDB) – holds system configuration information
    - LNET Routers - connect to the external filesystem.

There are usually many more compute than service nodes

Differences between Nodes

Service/Login Nodes

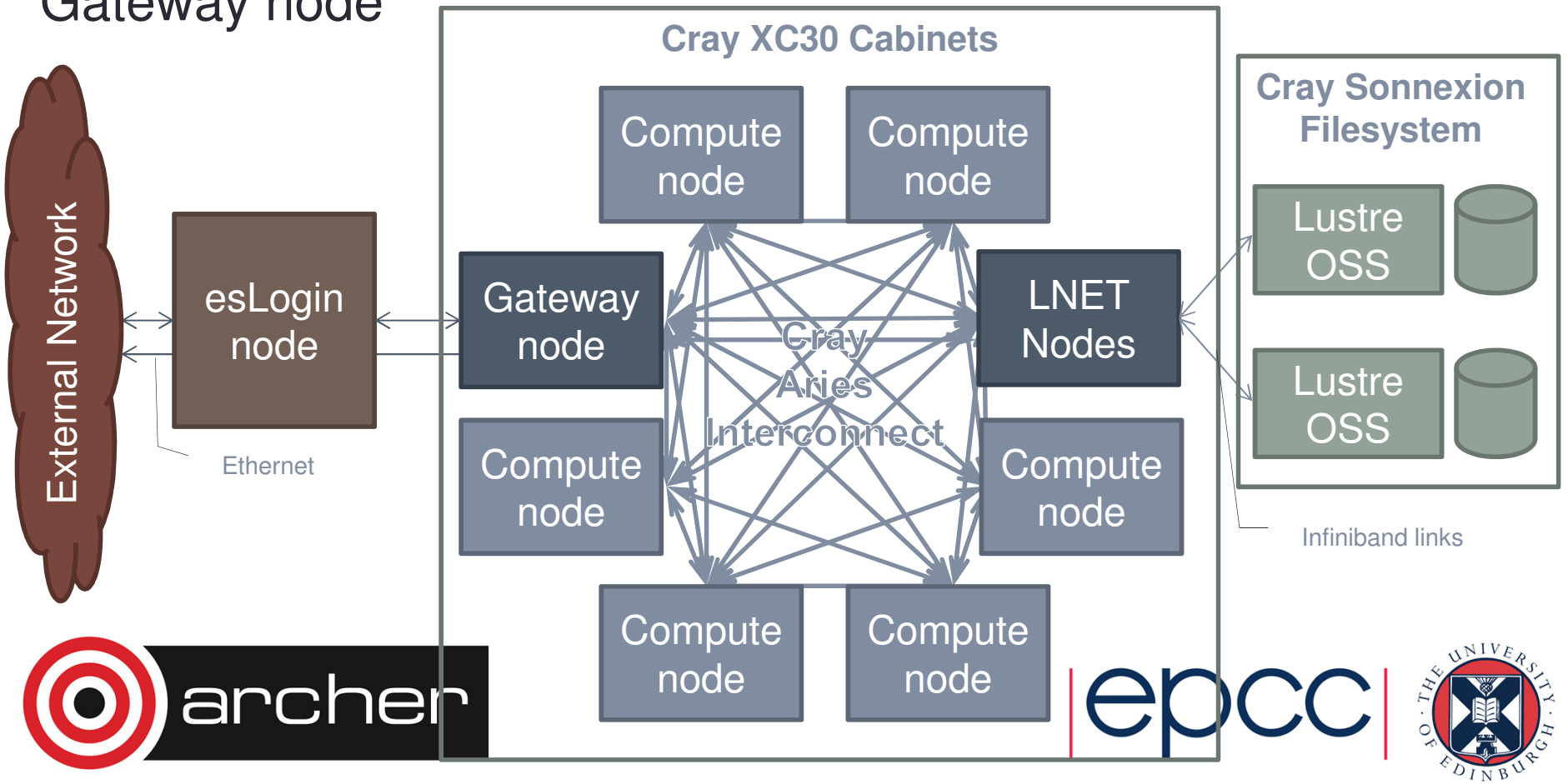
- This is the node you access when you first log in to the system.
- They run a full version of the CLE operating system (all libraries and tools available)
- They are used for editing files, compiling code, submitting jobs to the batch queue and other interactive tasks.
- They are shared resources that may be used concurrently by multiple users.
- There may be many service nodes in any Cray XC30 and can be used for various system services (login nodes, IO routers, daemon servers).

Compute nodes

- These are the nodes on which production jobs are executed
- They run Compute Node Linux, a version of the OS optimised for running batch workloads
- They can only be accessed by submitting jobs through a batch management system (PBS Pro on ARCHER)
- They are exclusive resources that may only be used by a single user.
- There are many more compute nodes in any Cray XC30 than login or service nodes.

Interacting with the system

Users do not log directly into the system. Instead they run commands via an esLogin server. This server will relay commands and information via a service node referred to as a "Gateway node"



archer

epcc

THE UNIVERSITY OF EDINBURGH

NATURAL ENVIRONMENT RESEARCH COUNCIL

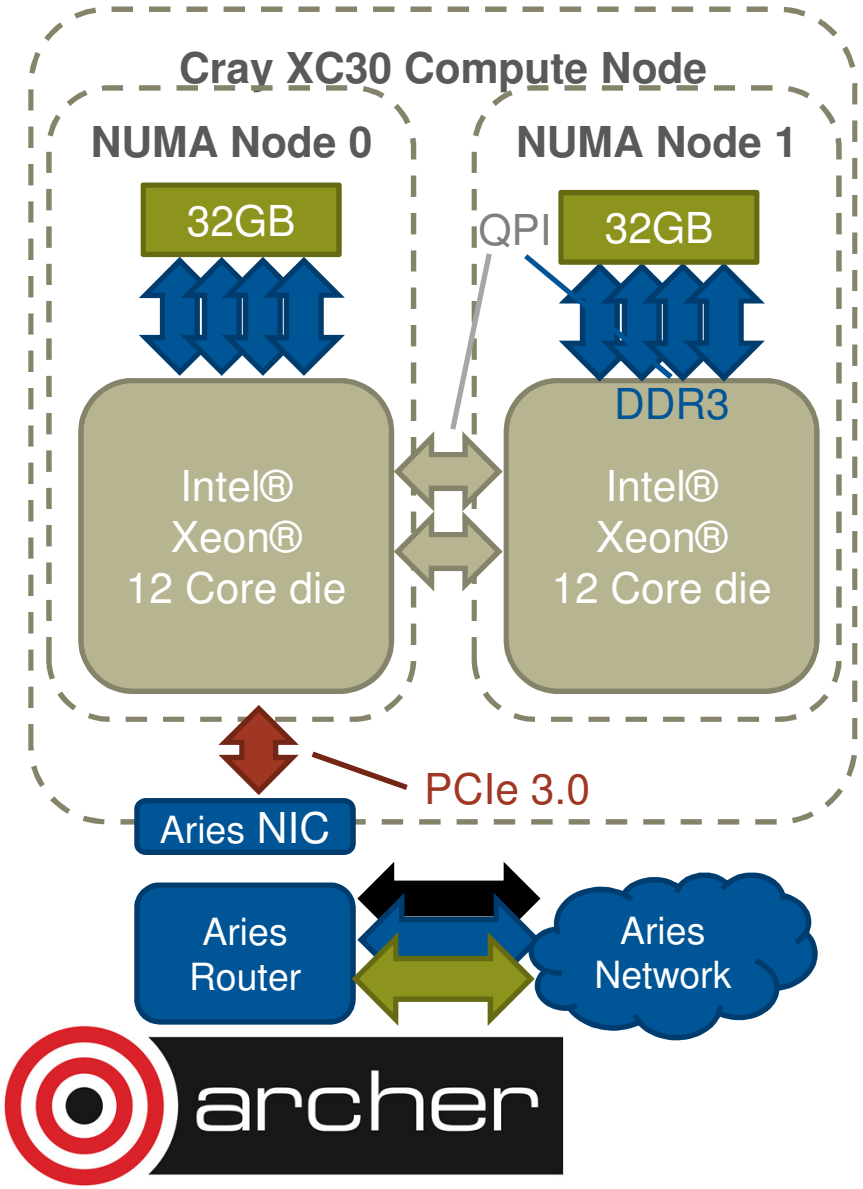
EPSRC

ARCHER Layout

Compute node architecture and topology

epcc

Cray XC30 Intel® Xeon® Compute Node



The XC30 Compute node features:

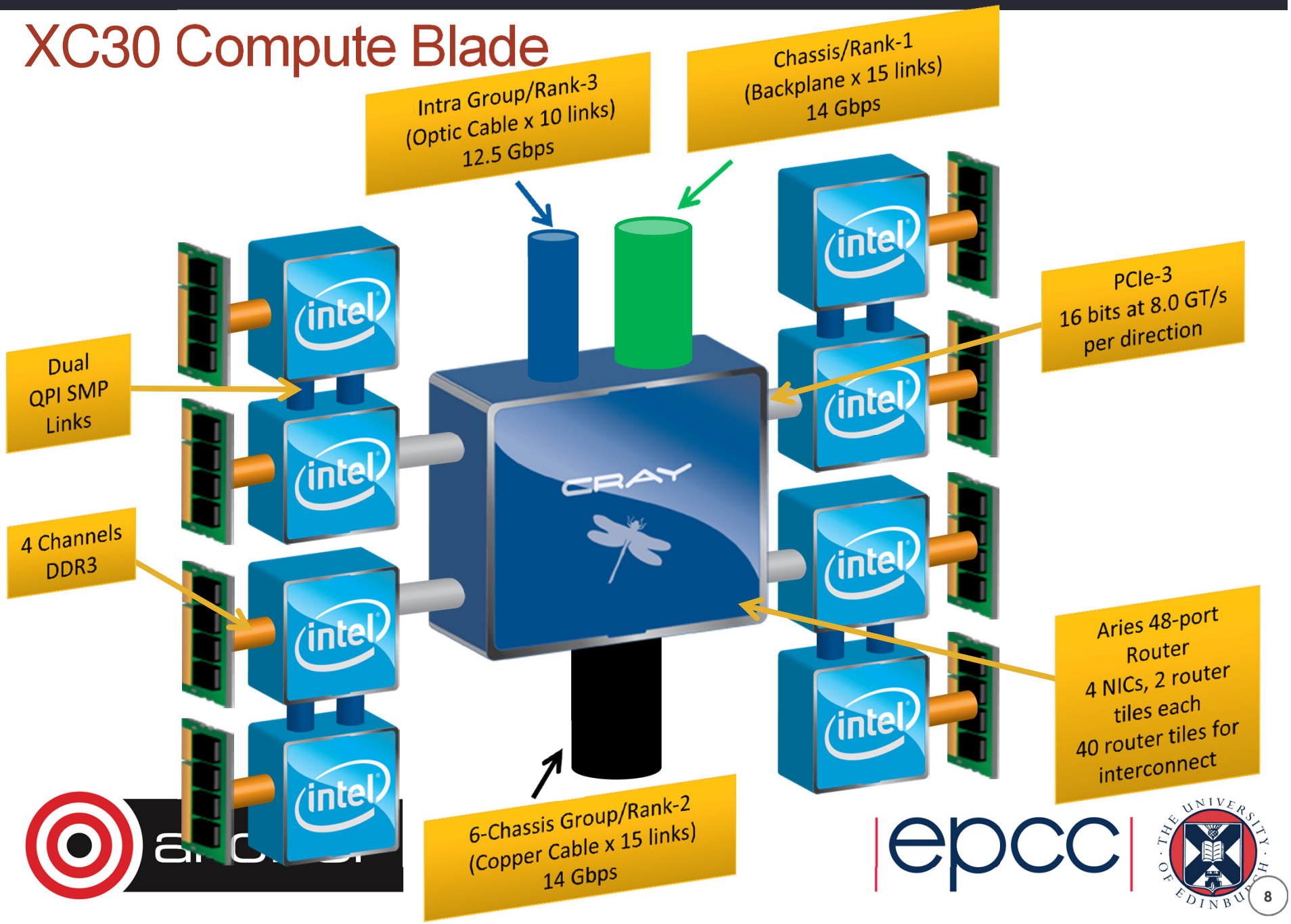
- 2 x Intel® Xeon® Sockets/die
  - 12 core Ivy Bridge
  - QPI interconnect
  - Forms 2 NUMA nodes
- 8 x 1833MHz DDR3
  - 8 GB per Channel
  - 64/128 GB total
- 1 x Aries NIC
  - Connects to shared Aries router and wider network
  - PCI-e 3.0

epcc | THE UNIVERSITY OF EDINBURGH

Terminology

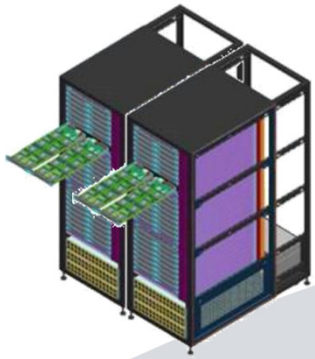
- A *node* corresponds to a single Linux OS
  - on ARCHER, two sockets each with a 12-core CPU
  - all cores on a node see the same shared memory space
  - i.e. maximum extent of an OpenMP shared-memory program
- Nodes are explicit to the user
  - resources allocated in quanta of nodes
  - use given exclusive access to all cores on a node
  - ARCHER resources requested in multiples of nodes
- All the following higher levels not explicit to user
  - but may have performance impacts in practice

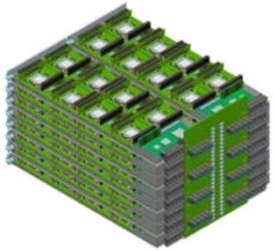
XC30 Compute Blade

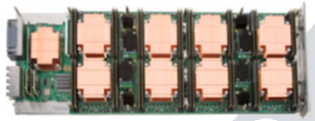


epcc

ARCHER System Building Blocks







Compute Blade

4 Compute Nodes

Chassis

Rank 1 Network

16 Compute Blades

No Cables

64 Compute Nodes

Group

Rank 2 Network

Passive Electrical Network

2 Cabinets

6 Chassis

384 Compute Nodes

System

Rank 3 Network

Active Optical Network

8 Groups

3008 Compute Nodes

archer

epcc

THE UNIVERSITY OF EDINBURGH

NATURAL ENVIRONMENT RESEARCH COUNCIL

EPSRC

CRAY XC30 DRAGONFLY TOPOLOGY + ARIES

archer

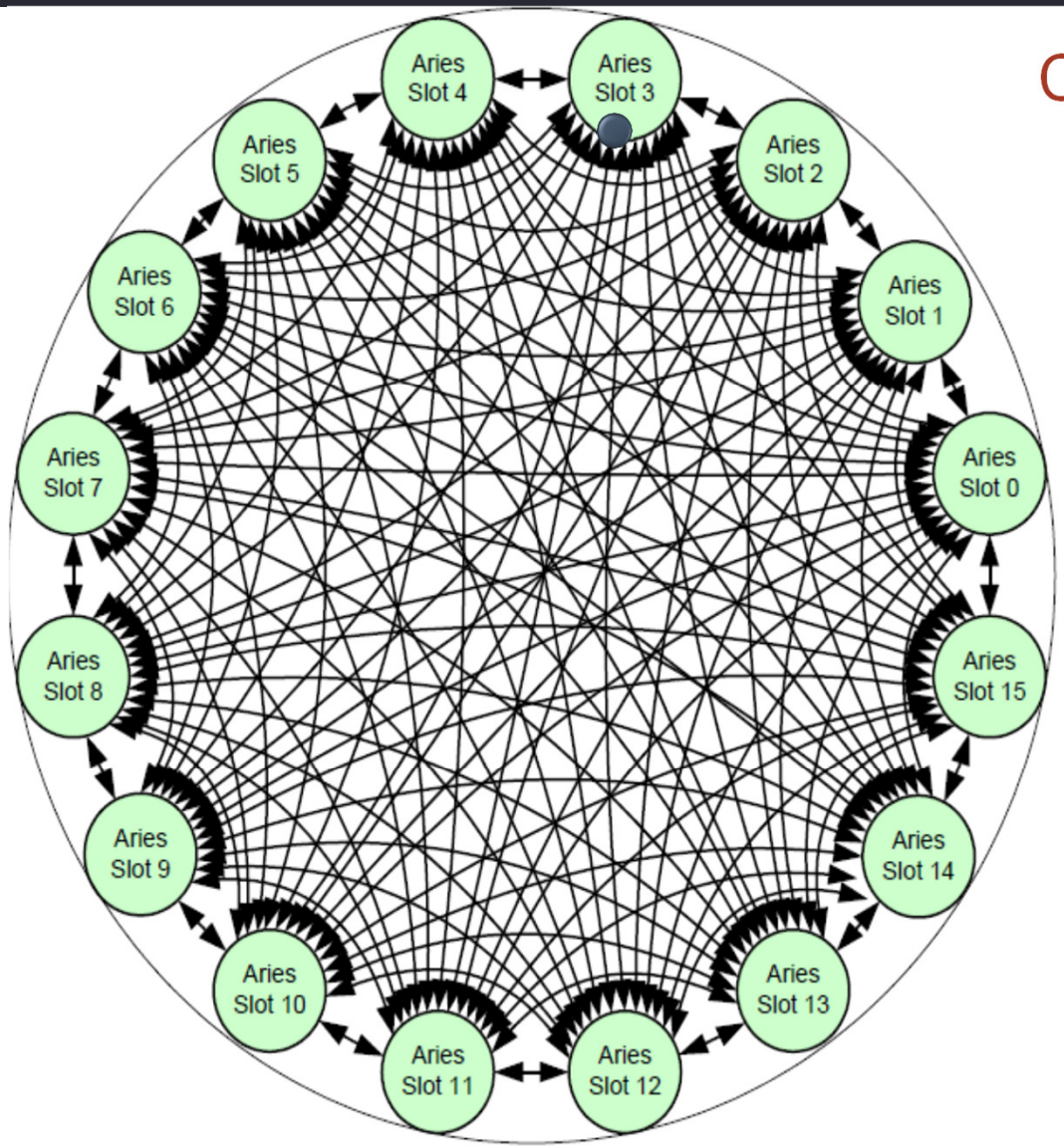
epcc

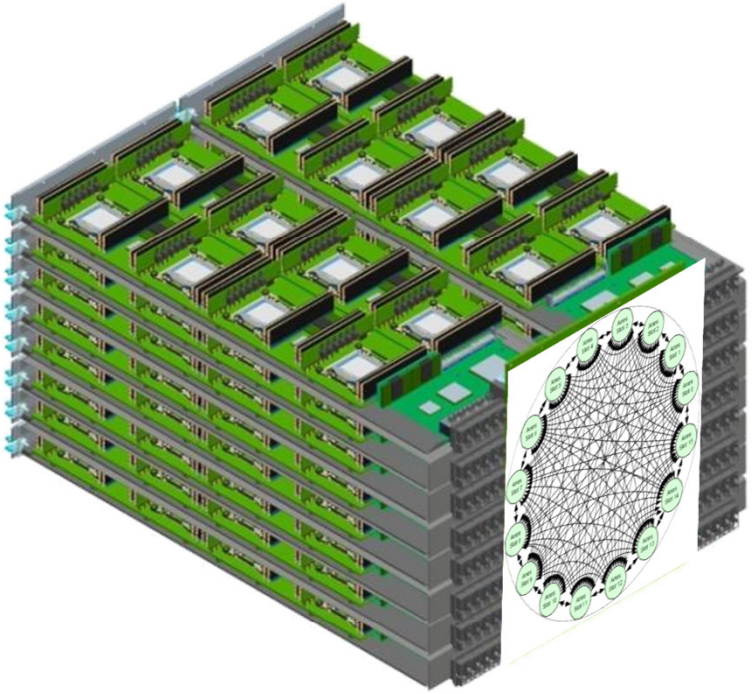
THE UNIVERSITY OF EDINBURGH

Cray Aries Features

- Scalability to > 500,000 X86 Cores
  - Cray users run large jobs – 20-50% of system size is common
  - Many examples of 50K-250K MPI tasks per job
  - Optimized collectives MPI_Allreduce in particular
- Optimized short transfer mechanism (FMA)
  - Provides global access to memory, used by MPI and PGAS
  - High issue rate for small transfers: 8-64 byte put/get and amo in particular
- HPC optimized network
  - Small packet size 64-bytes
  - Router bandwidth >> injection bandwidth
  - Adaptive Routing & Dragonfly topology
- Connectionless design
  - Doesn't depend on a connection cache for performance
  - Limits the memory required per node
- Fault tolerant design
  - Link level retry on error
  - Adaptive routing around failed links
  - Network reconfigures automatically (and quickly) if a component fails
  - End to end CRC check with automatic software retry in MPI

Cray XC30 Rank1 Network





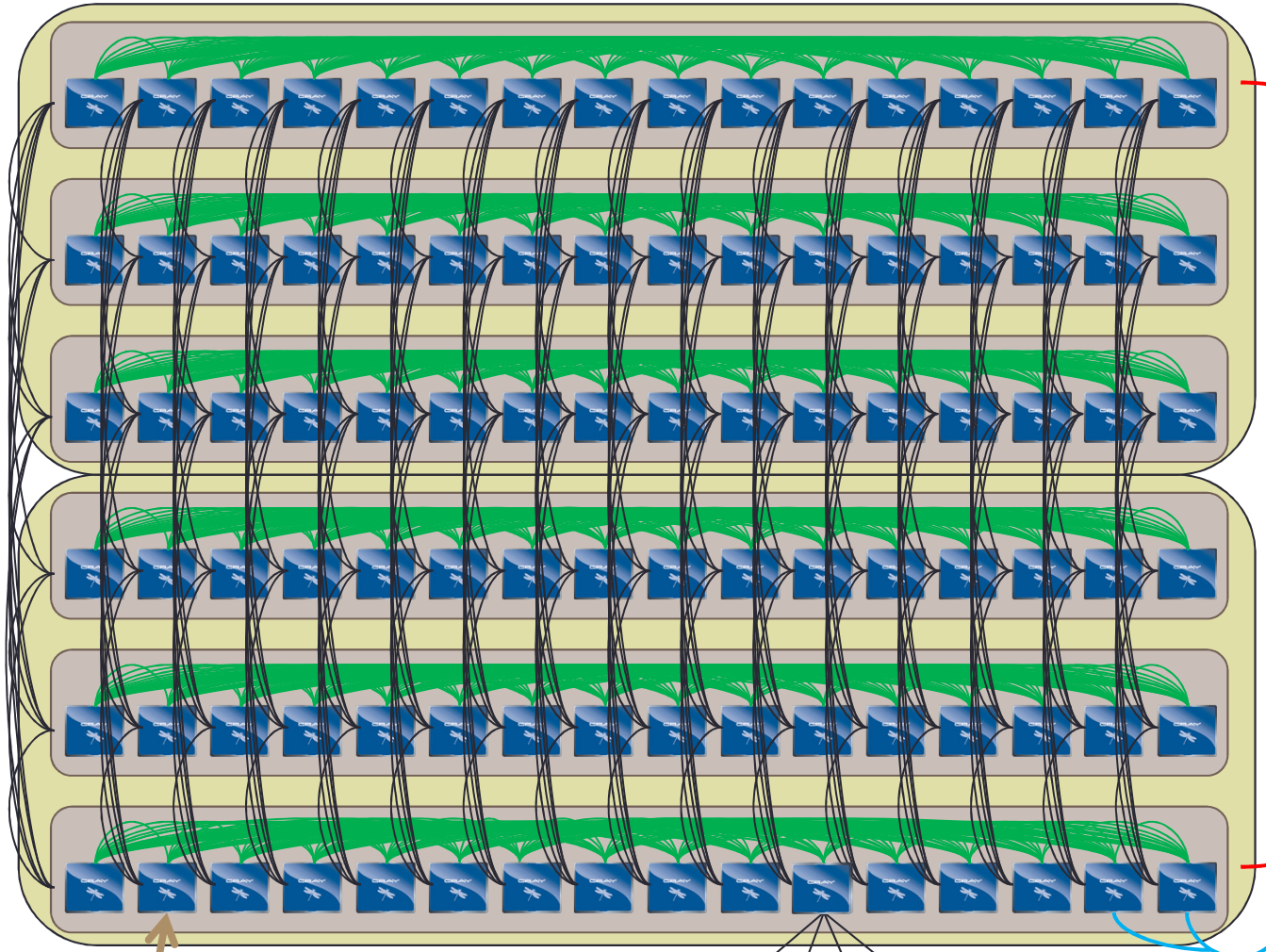
- **Chassis with 16 compute blades**
- **128 Sockets**
- **Inter-Aries communication over backplane**
- **Per-Packet adaptive Routing**

archer

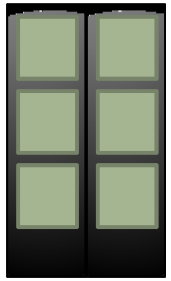
epcc

THE UNIVERSITY OF EDINBURGH

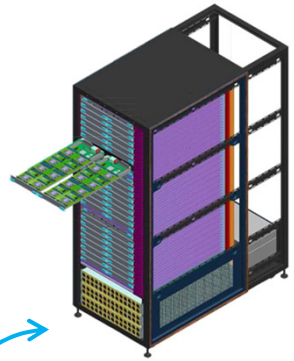
Cray XC30 Rank-2 Copper Network



2 Cabinet Group
768 Sockets

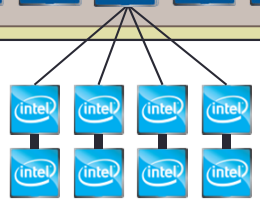


6 backplanes connected with copper cables in a 2-cabinet group:



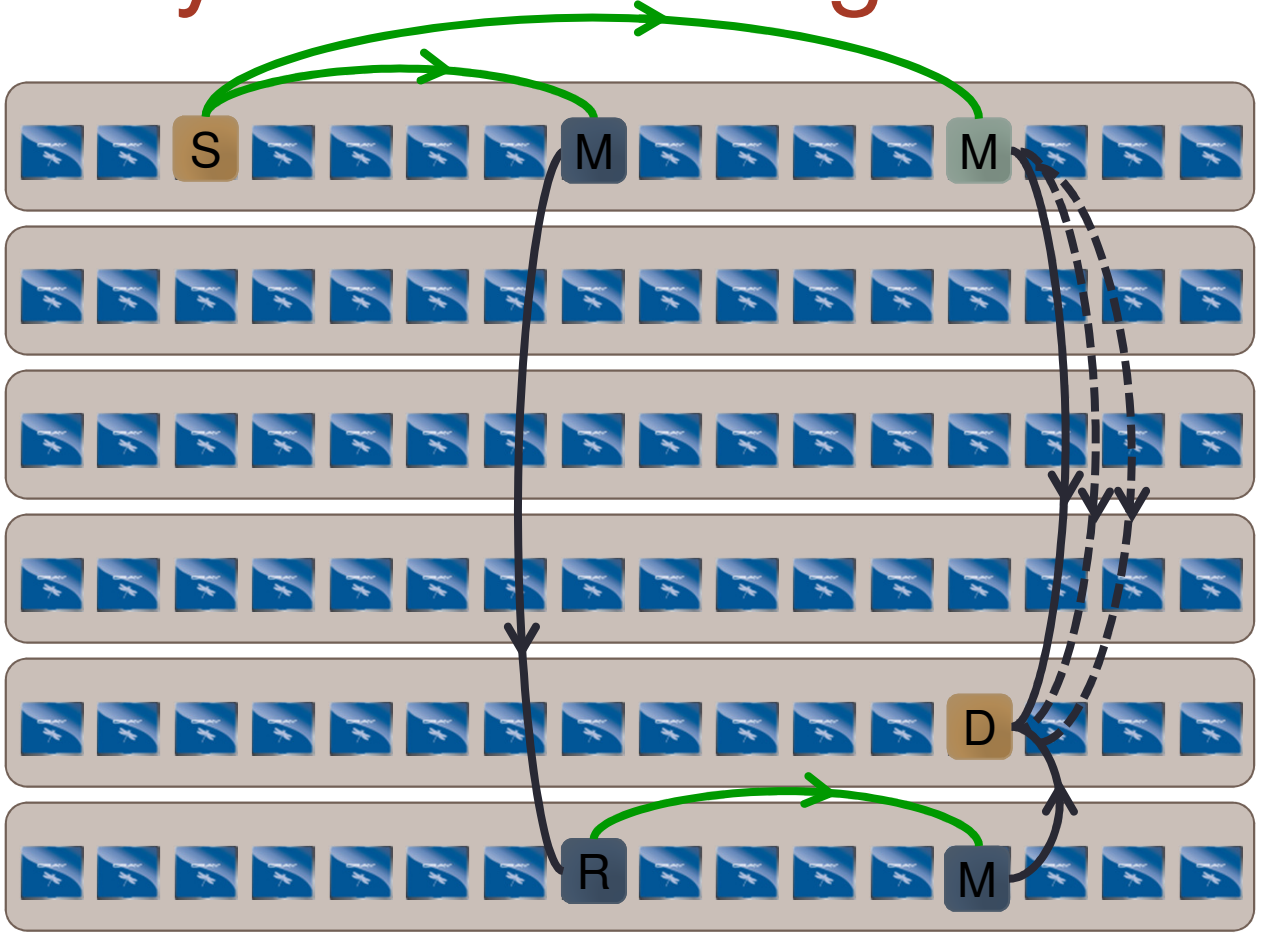
Active optical cables interconnect groups

16 Aries connected by backplane



4 nodes connect to a single Aries

Cray XC30 Routing



Minimal routes between any two nodes in a group are just two hops

Non-minimal route requires up to four hops.

With adaptive routing we select between minimal and non-minimal paths based on load

The Cray XC30 Class-2 Group has sufficient bandwidth to support full injection rate for all 384 nodes with non-minimal routing

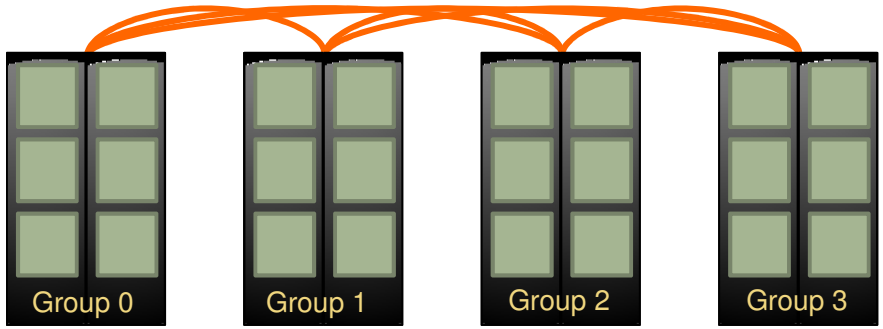
archer

epcc

THE UNIVERSITY OF EDINBURGH

Cray XC30 Network Overview – Rank-3 Network

- An all-to-all pattern is wired between the groups using optical cables (blue network)
- Up to 240 ports are available per 2-cabinet group
- The global bandwidth can be tuned by varying the number of optical cables in the group-to-group connections



Example: An 4-group system is interconnected with 6 optical "bundles". The "bundles" can be configured between 20 and 80 cables wide

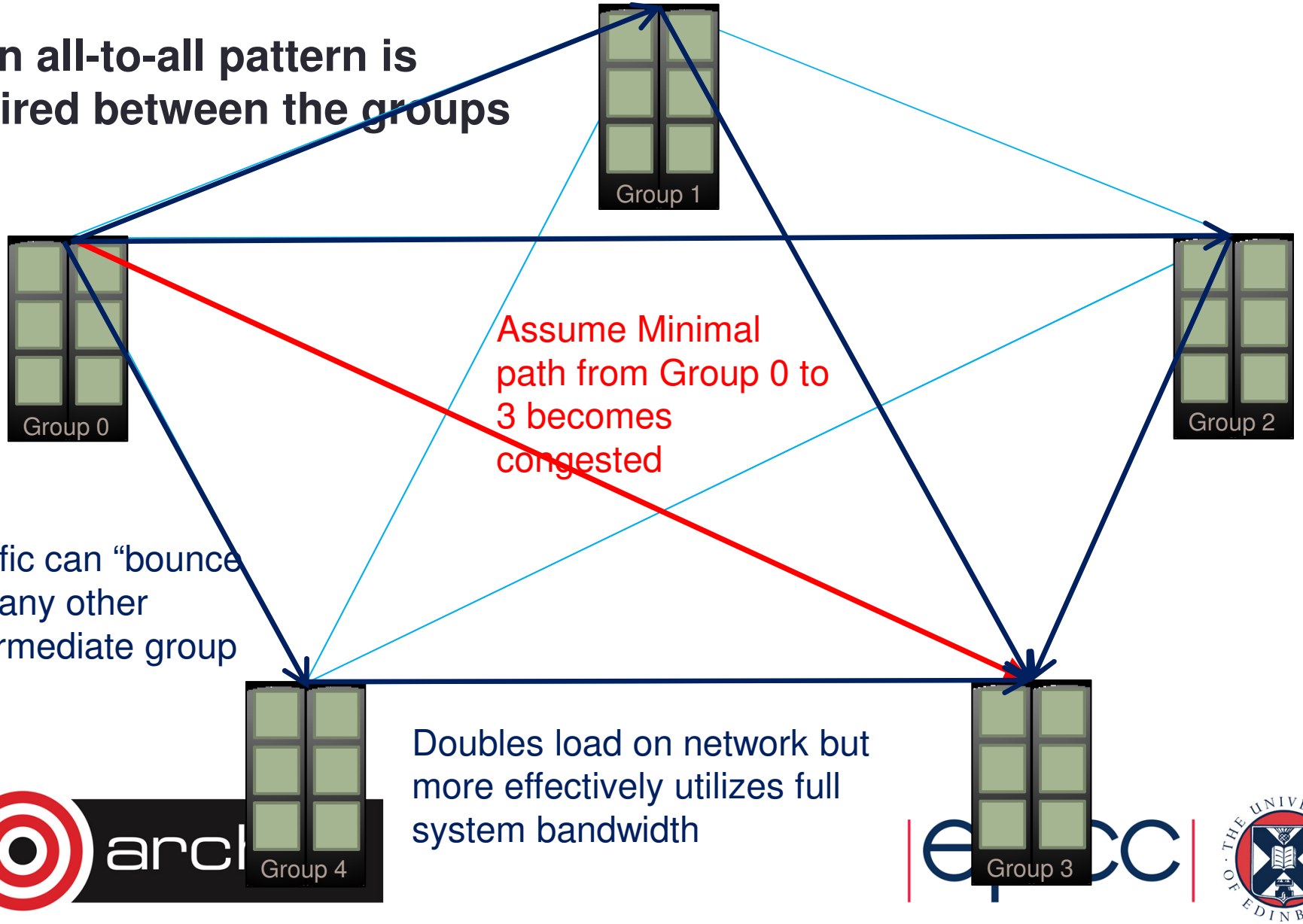
archer

epcc

THE UNIVERSITY OF EDINBURGH

Adaptive Routing over optical network

- **An all-to-all pattern is wired between the groups**



Filesystems

- /home – NFS, not accessible on compute nodes
  - For source code and critical files
  - Backed up
  - > 200 TB total
- /work – Lustre, accessible on all nodes
  - High-performance parallel filesystem
  - Not backed-up
  - > 4PB total
- RDF – GPFS, not accessible on compute nodes
  - Long term data storage

Filesystems

- No /tmp on backend nodes
  - GNU Fortran, file OPEN statements with STATUS='SCRATCH‘
  - export GFORTRAN_TMPDIR=/work/[project]/[group]/[username]/tmp
- Users assigned to projects
  - Filesystems configured around projects:
    - /home/projectcode/projectcode/username
    - /work/projectcode/projectcode/username
- Group permissions also done per project
  - Possible to access files on group permissions with projects but beyond a project would need world readable files
- Sharing data
  - Within projects
    - /work/projectcode/projectcode/shared
  - Between projects
    - /work/projectcode/shared

Summary of ARCHER

- Each nodes contains 24 Intel IvyBridge cores
- 3008 Compute Nodes connected by Aries network
  - 64 GB per node; 1/8th of the nodes (one group) have 128 GB
- Total of 72,192 cores
  - over 200 TB memory
- Peak performance of 1.6PF
- ARCHER is *not* a Linpack engine, but benchmarked at 1.367 PF
  - #19 in November 2013 top 500 list
  - #25 in July 2014 top 500 list
  - Upgrade planned soon
- Expected to provide nearly four times the scientific throughput of its predecessor HECToR
  - HECToR #49 in top 500 with 0.8 PF

NATURAL ENVIRONMENT RESEARCH COUNCIL

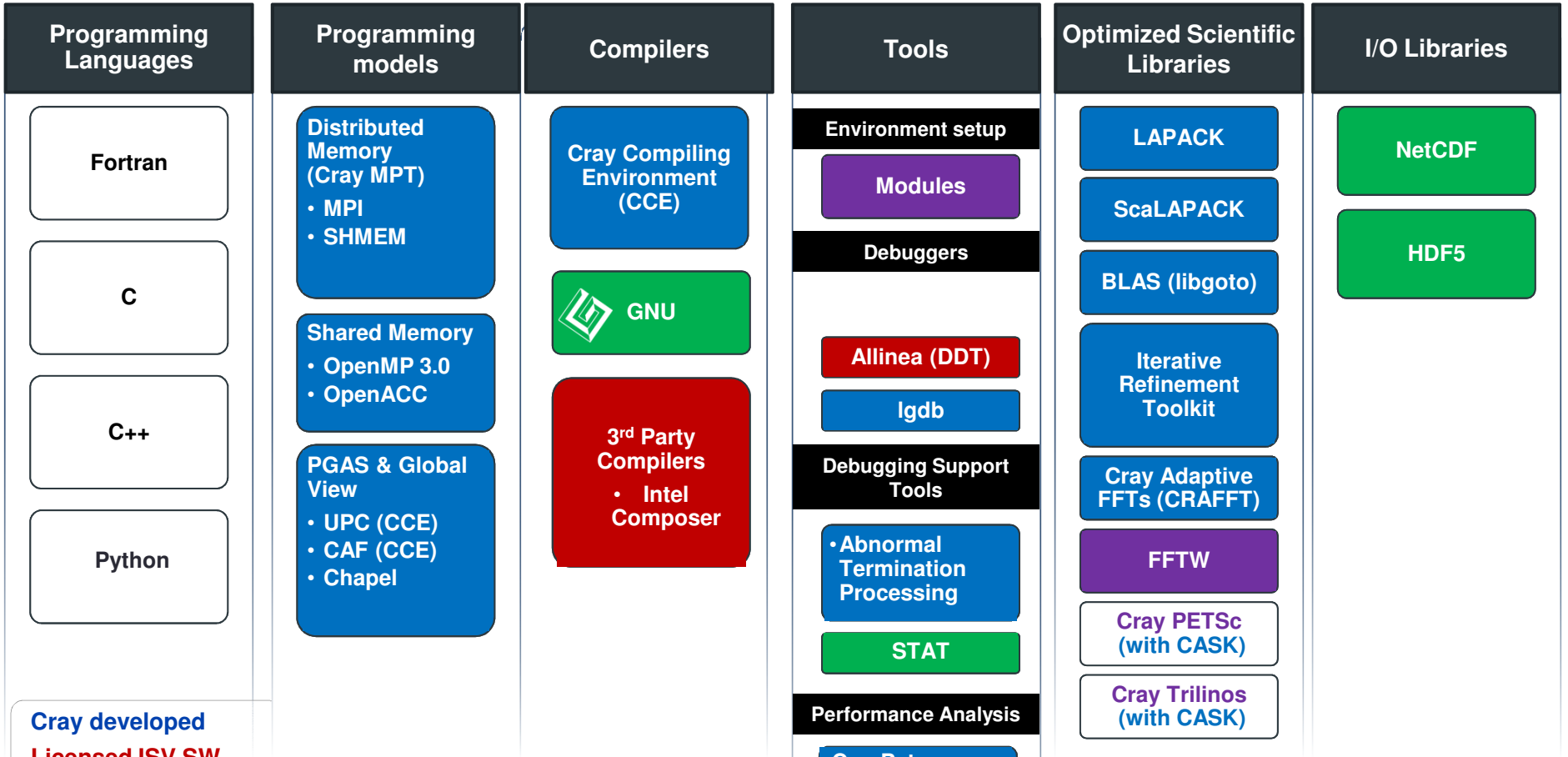
EPSRC

ARCHER Software

Brief Overview

epcc

Cray's Supported Programming Environment



archer

epcc

THE UNIVERSITY OF EDINBURGH

Cray MPI & SHMEM

- Cray MPI
  - Implementation based on MPICH2 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-2 support with the exception of
    - Dynamic process management (MPI_Comm_spawn)
  - MPI-3 support coming soon

- Cray SHMEM
  - Fully optimized Cray SHMEM library supported
    - Fully compliant with OpenSHMEM v1.0
    - Cray XC implementation close to the T3E model

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.
- Support for traditional debugging mechanism
  - Allinea DDT 4.0.1
  - gdb

User administration

- SAFE website used for user administration
  - https://www.archer.ac.uk/safe
- Apply for accounts
- Manage project resources
- Report on usage
- View queries
- Etc….

NATURAL ENVIRONMENT RESEARCH COUNCIL

EPSRC

QUESTIONS?

epcc

THE UNIVERSITY OF EDINBURGH