

ARC3

Mark Dixon

m.c.dixon@leeds.ac.uk



- **What ARC3 is**
- **What it looks like**
- **CPU developments**
- **How is it different to earlier ARC machines**

New system!!!

- Pilot started 1st February 2017
- “Soft” start of service 1st March 2017
- ARC3, physically split between two rooms, consists of:
 - 2 login nodes (`arc3.leeds.ac.uk`)
 - 4056 cores of Intel Xeon “Broadwell” E5-2650v4 (12 cores per CPU) & 22 Tb RAM
 - 4 NVIDIA K80 GPUs
 - 2 High-memory nodes
 - 370 Tb parallel filesystem (`/nobackup`)
 - Mellanox FDR InfiniBand interconnect, 56Gbit/s, 2:1 blocking

ARC3

ARC3 is the replacement system for ARC1.

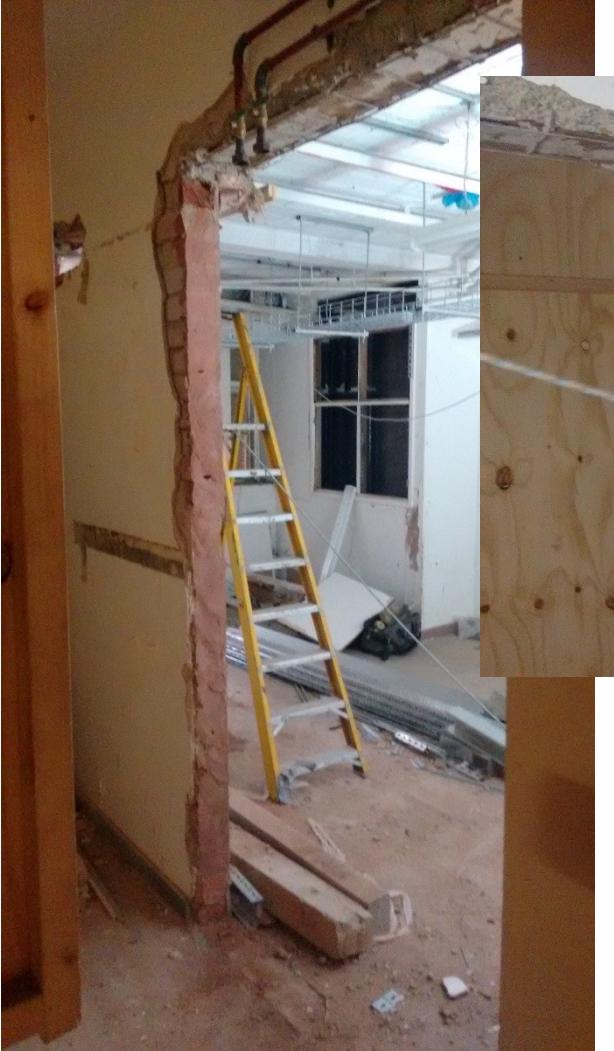
ARC3



UNIVERSITY OF LEEDS







ARC3



UNIVERSITY OF LEEDS



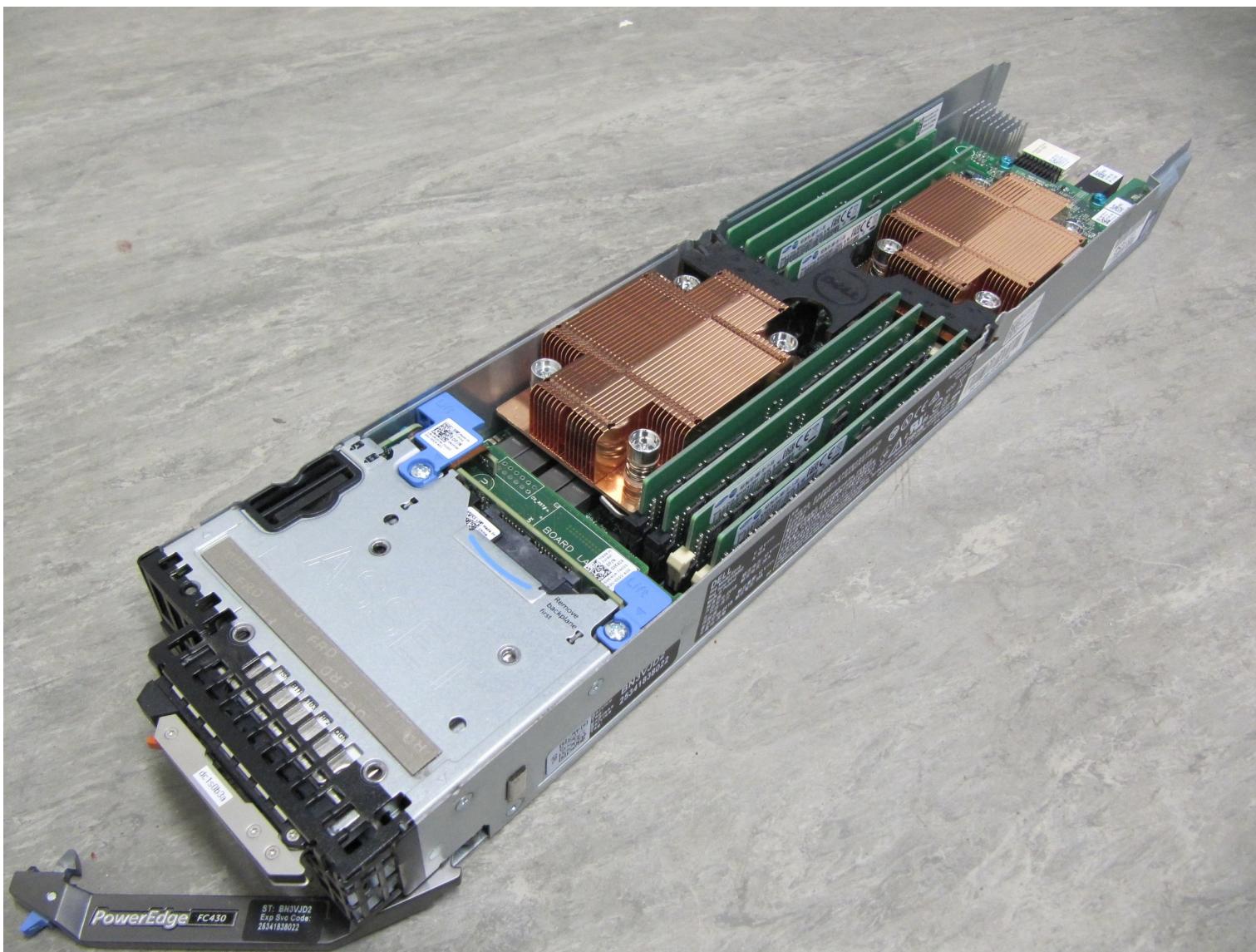


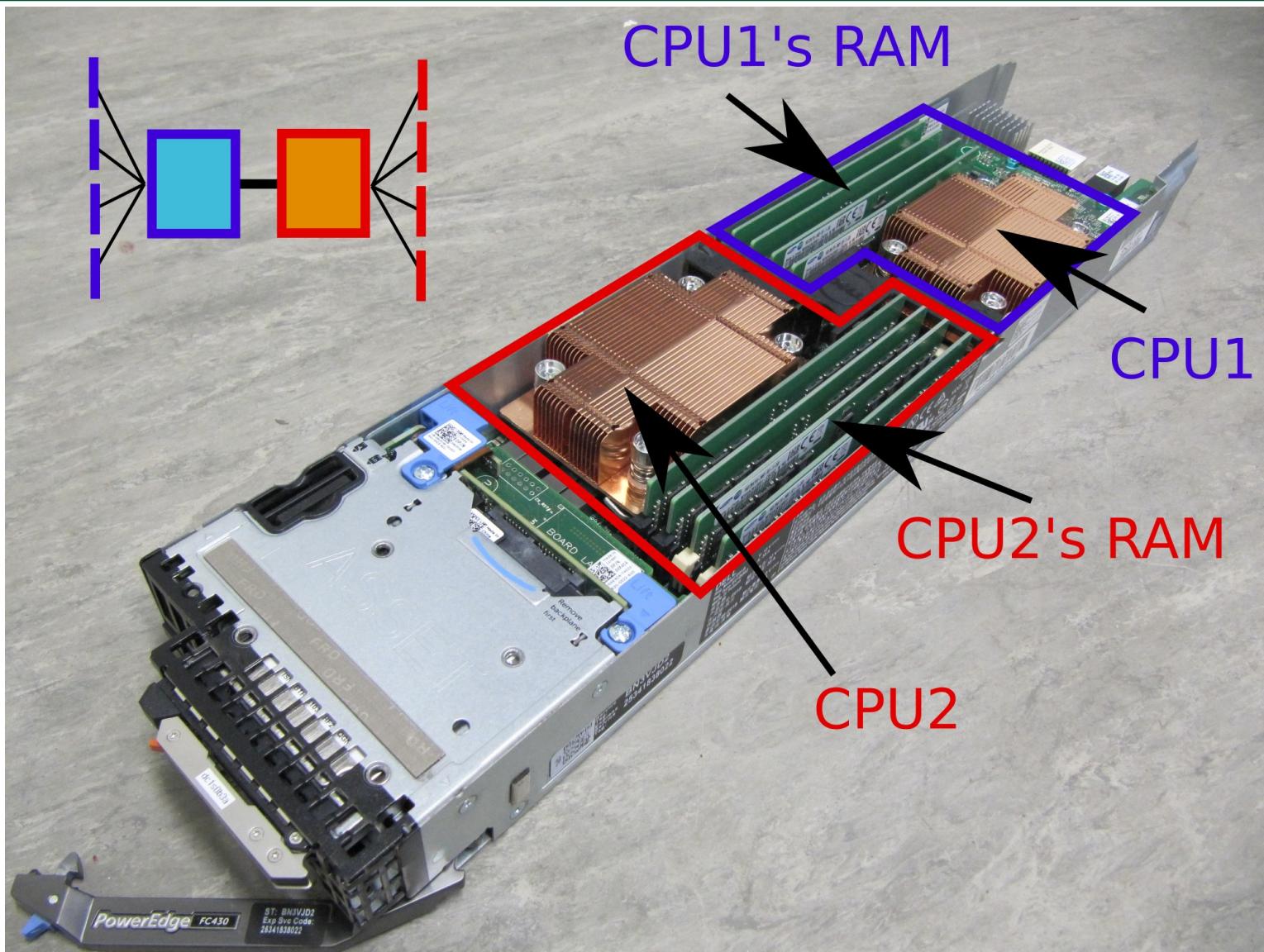


ARC3



UNIVERSITY OF LEEDS







ARC3



UNIVERSITY OF LEEDS



ARC3



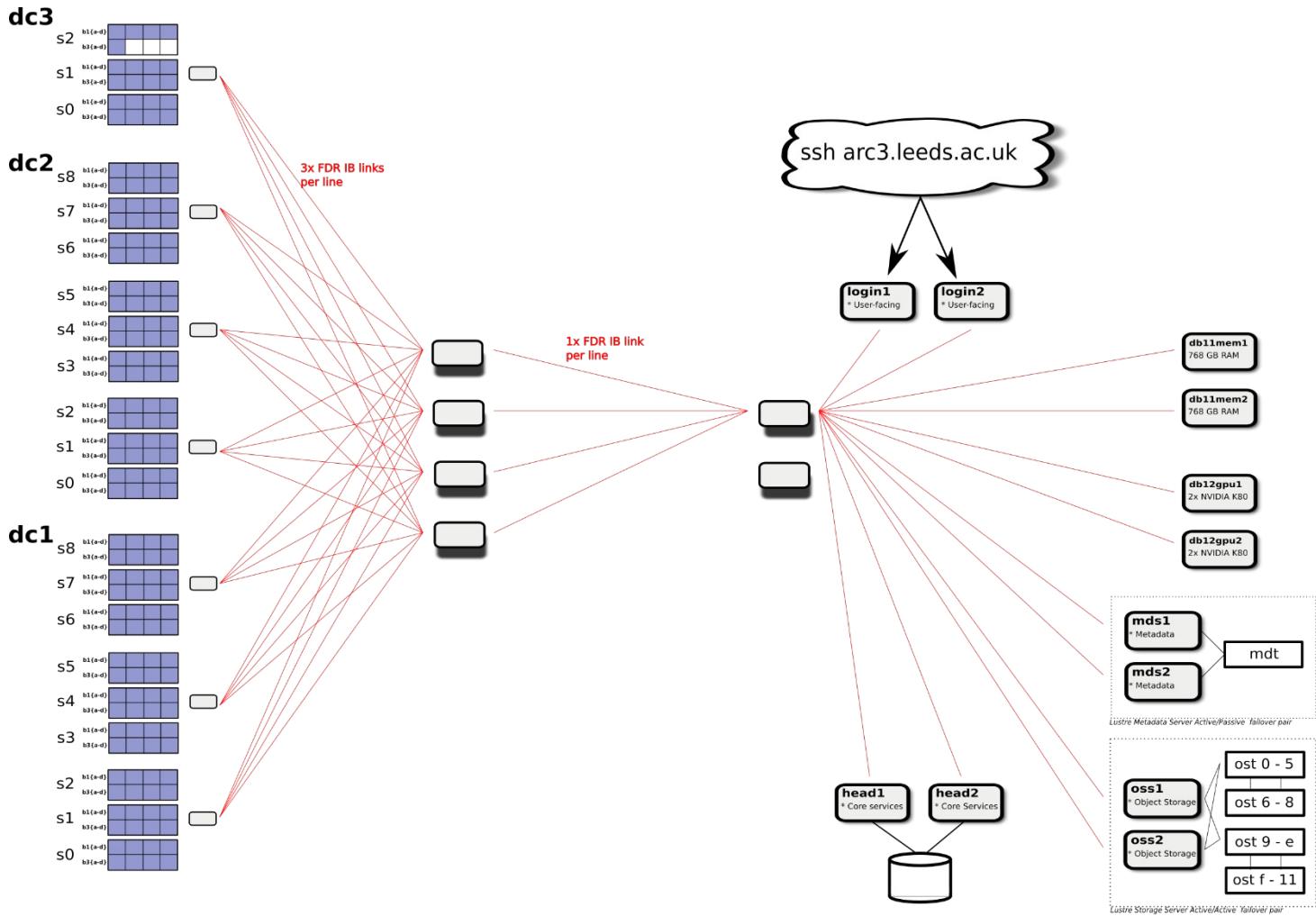
UNIVERSITY OF LEEDS



ARC3



UNIVERSITY OF LEEDS



`/nobackup`
(Lustre parallel filesystem)

Comparison with other ARC machines:

	ARC3 (2016/2017)	ARC2 (2013)	ARC1 (2009)
Overview	152 Tflop/s 4056 cores (AVX2) 22 TB RAM 56Gbit Mel. InfiniBand Hi mem (768GB) and GPU nodes CentOS 7	63 Tflop/s 3040 cores (AVX) 6 TB RAM 40Gbit Mel. InfiniBand CentOS 6	25 Tflop/s 2112 cores (SSE4.2) 3.6 TB RAM 40Gbit Mel. InfiniBand Hi mem (128GB) nodes CentOS 5
Main node_type	24core-128G	16core-32G	8core-12G
CPUs	Intel “Broadwell” 2x E5-2650v4 2.2GHz	Intel “Sandy Bridge” 2x E5-2670 2.6GHz	Intel “Nehalem” 2x X5560 2.8GHz
Memory	8x 16GB 2.4GHz	8x 4GB 1.6GHz	6x 2GB 1GHz (eff.)
Disk	100 GB SSD	400 GB HDD	11 GB HDD
Per core	35.2 Gflop/s 5.3 GB RAM 800 MHz RAM	20.8 Gflop/s 2 GB RAM 800 MHz RAM	11.2 Gflop/s 1.5 GB RAM 750 MHz RAM



Developments in CPU design

- CPU frequencies stopped increasing some years ago (power limits) and is now slowly dropping (power limits)
- Transistors keep getting smaller (fingers crossed), changes in where the transistor budget is spent. Previously used to:
 - Provide richer instruction sets
 - Increase the size of the pipeline (number of instructions processed at once)
 - Move to multiple cores in a single CPU
 - Current focus is on more powerful, “vector”, instructions and more cores
- Memory performance struggling to keep up

Developments in CPU design

Status:

Current focus is on more powerful, “vector”, instructions. Single instruction operates on multiple numbers at once.

What this means:

- The quality of the compiler is increasingly important
- If you are not using the Intel compiler, please consider it
- If you are using the Intel compiler, please consider investigating how it is optimising your code
- Profile your code, be nice to your compiler at bottlenecks

We can help.



Developments in CPU design

Some recommended compiler flags...

Intel:

Compile with: `-xCORE-AVX2` (or `-xHost`)

Diagnostics: `-qopt-report -qopt-report-phase=vec`

GNU:

Compile with: `-march=broadwell`

PGI:

Compile with: `-tp=haswell-64`



General user-visible changes: evolution, not revolution

- Latest version of operating system (CentOS 7)
- Latest versions of compilers, libraries, applications (ask if we've not yet ported across something you need)
- Default node has 24core-128G, two nodes of 24core-768G available (`qsub -l node_type=24core-768G <job>`)
- Features:
 - High memory nodes
 - GPU nodes
 - Compute node storage
 - Remote graphics
 - Containers
 - Better integration with Windows groups

Feature: High memory nodes

- People often have a requirement for large amounts of memory for their application/problem
- Cannot always be distributed across multiple nodes with MPI.
- Standard node type has reasonable amount of memory (128G), but some people may sometimes need more.
- Two high memory nodes are available
 - Each with 24 cores, 768G RAM
 - `qsub -l node_type=24core-768G jobsctipt.sh`

Feature: GPU nodes

- Increasing numbers of people asking us about GPUs
 - More complicated “memory latency hiding” programming model
 - Suited to strong scaling problems requiring low memory sizes
 - Improving application support
- Two nodes are available
 - Each with 24 cores, 128G RAM and 2x NVIDIA K80 cards
 - Each card presents 2x GPUs (2.5k cores and 12G RAM per GPU)
 - Jobs can ask for half or all of the memory, RAM, cards in a node:
 - `qsub -l coproc_k80=1 jobscrip.sh`
 - `qsub -l coproc_k80=2 jobscrip.sh`
- Possible use to pilot accelerated graphics feature in future

Feature: Remote Graphics

- Running an X server on the desktop and running X applications on an ARC system is ...
 - ... Slow (particularly off-site or over the wireless network)
 - ... Increasingly unreliable (particularly for 3D graphics)
- Paraview, VisIt available – designed to work in this type environment
- We're experimenting with X2GO as an alternative
 - Run local client (available for Windows, Linux, Mac)
 - Connects to ARC3 via SSH
 - Runs X server and X applications on same machine
 - Scope for adding hardware-accelerated remote graphics in future

Feature: Containers

- Sometimes want more flexibility in how software is obtained
 - Installing software yourself can be painful
 - Getting ARC team to install can be slower than needed and removes control
- Experimenting with “containers”
 - Bundles an application with its entire environment, i.e. its own copy of the operating system, as a single image
 - “Docker” is currently the most popular type, can download prebuild images for certain tasks from the internet
 - We have installed “Singularity”, which can run Docker images



Feature: Better integration with Windows groups

- ARC3 imports each user's Windows / Active Directory groups each night
- Means that if your research group already has some groups, they can be used on ARC3 to control access to files.



Please see our website for more details:

<http://arc.leeds.ac.uk/>

To apply for an ARC3 account, please go to:

- Existing users:

<http://arc.leeds.ac.uk/apply/requesting-an-arc3-account/>

- New users:

<http://arc.leeds.ac.uk/apply/getting-an-account/>

ARC3

Mark Dixon

m.c.dixon@leeds.ac.uk