
# Understanding & Improving I/O Performance on HPC Systems

**Adrian Jackson** EPCC
**Keeran Barabazon** ARM/Allinea
**Tomislav Šubić** Arctur
**Muhammad Sarim Zafar** TU Dresden
a.jackson@epcc.ed.ac.uk
@adrianjhpc

# Aims


- Understand parallel/high performance I/O hardware and software
- Understand the different aspects that can impact performance at different scales, and the share nature of the resources
- Learn different parallel I/O strategies
- Understand profiling and analysing I/O performance for parallel programs
- Get hands on with parallel I/O profiling

# Real Aims


- Understanding I/O performance varies significantly
- Understanding I/O is hard
- Thinking about different ways you can do I/O
- Understand some tools that can help you

# Format

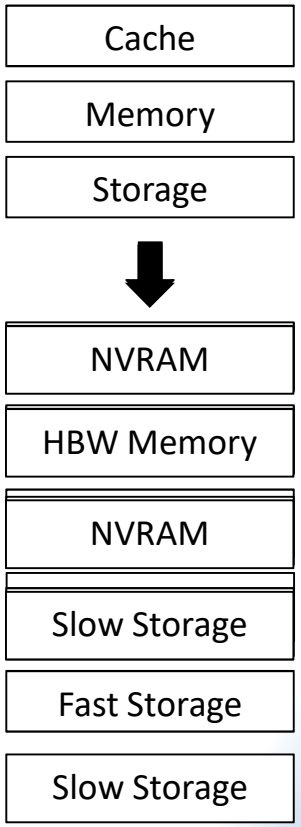

- Lectures and practicals
- Slides and exercise material available online:
  - https://
- Exercises will be done on remote machine
  - ARCHER (https://www.archer.ac.uk)
  - We will give you accounts on these
- Some programming for those who want to, for others can just run programs
- Some setting up of profiling tools

# Timetable


- 09.00 Large scale computing system I/O hardware and software libraries
- 09.45 Systemwide performance and implications
- 11.00 Break
- 11.30 Systemwide data and benchmarking practical
- 12.15 Parallel I/O strategies and libraries
- 13.00 Lunch
- 14.00 Parallel I/O practical
- 15.00 Profiling application I/O
- 16.00 Break
- 16.30 Hands on Profiling
- 17.30 Summary and discussion

# New Memory Hierarchies


- High bandwidth, on processor memory
  - Large, high bandwidth cache
  - Latency cost for individual access may be an issue
- Main memory
  - DRAM
  - Costly in terms of energy, potential for lower latencies than high bandwidth memory
- NVRAM main memory
  - High capacity, ultra fast storage
  - Low energy (when at rest) but still slower than DRAM

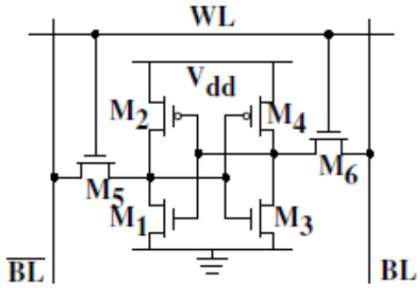


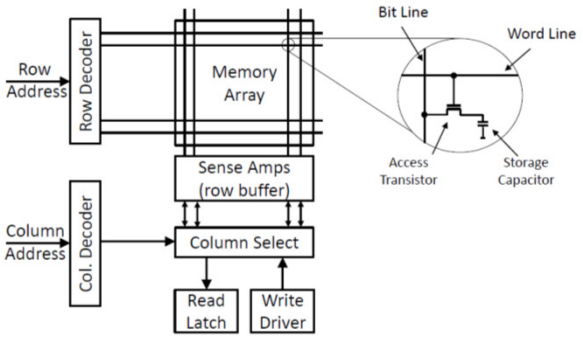
# Non-volatile memory

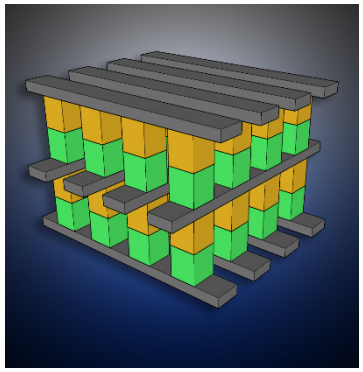

- Non-volatile RAM
  - 3D XPoint technology
  - STT-RAM
- Much larger capacity than DRAM
  - Hosted in the DRAM slots, controlled by a standard memory controller
- Slower than DRAM by a small factor, but significantly faster than SSDs
- STT-RAM
  - Read fast and low energy
  - Write slow and high energy
    - Trade off between durability and performance
    - Can sacrifice data persistence for faster writes

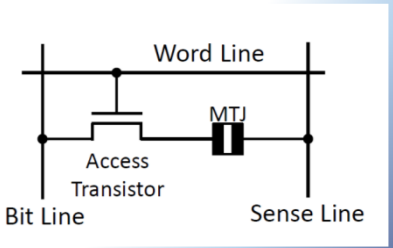
# SRAM vs NVRAM

- SRAM used for cache
- High performance but costly
  - Die area
  - Energy leakage
- DRAM lower cost but lower performance
  - Higher power/refresh requirement
- NVRAM technologies offer
  - Much smaller implementation area
  - No refresh/ no/low energy leakage
  - Independent read/write cycles





# NEXTGenIO - key facts

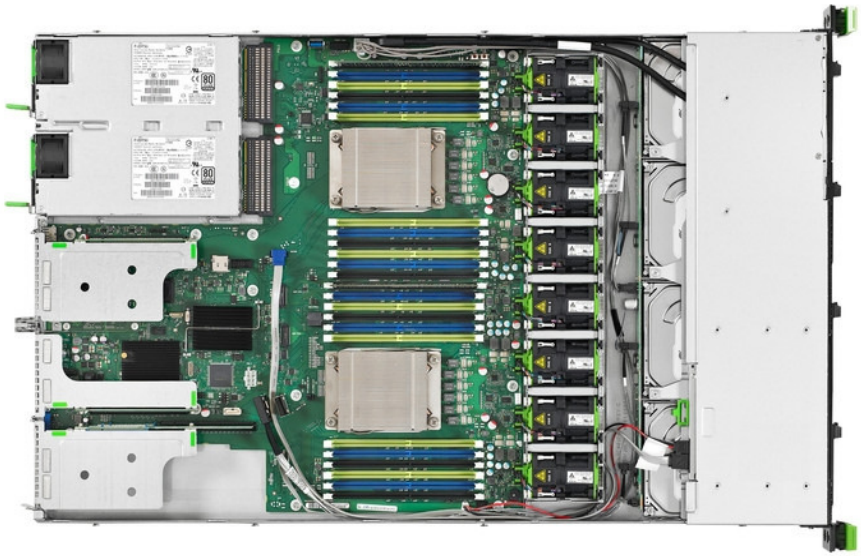
- Research & Innovation Action
- 36 month duration
- €8.1 million
- Approx. 50% committed to hardware development
- Prototype system part of the project

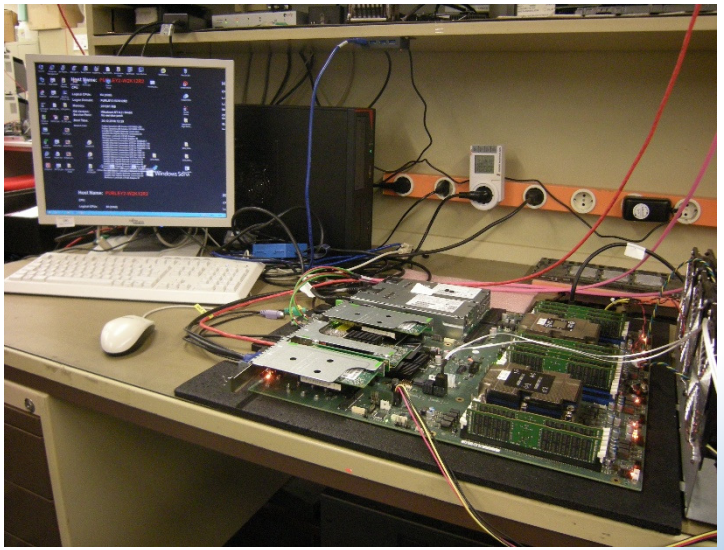
# Project objectives

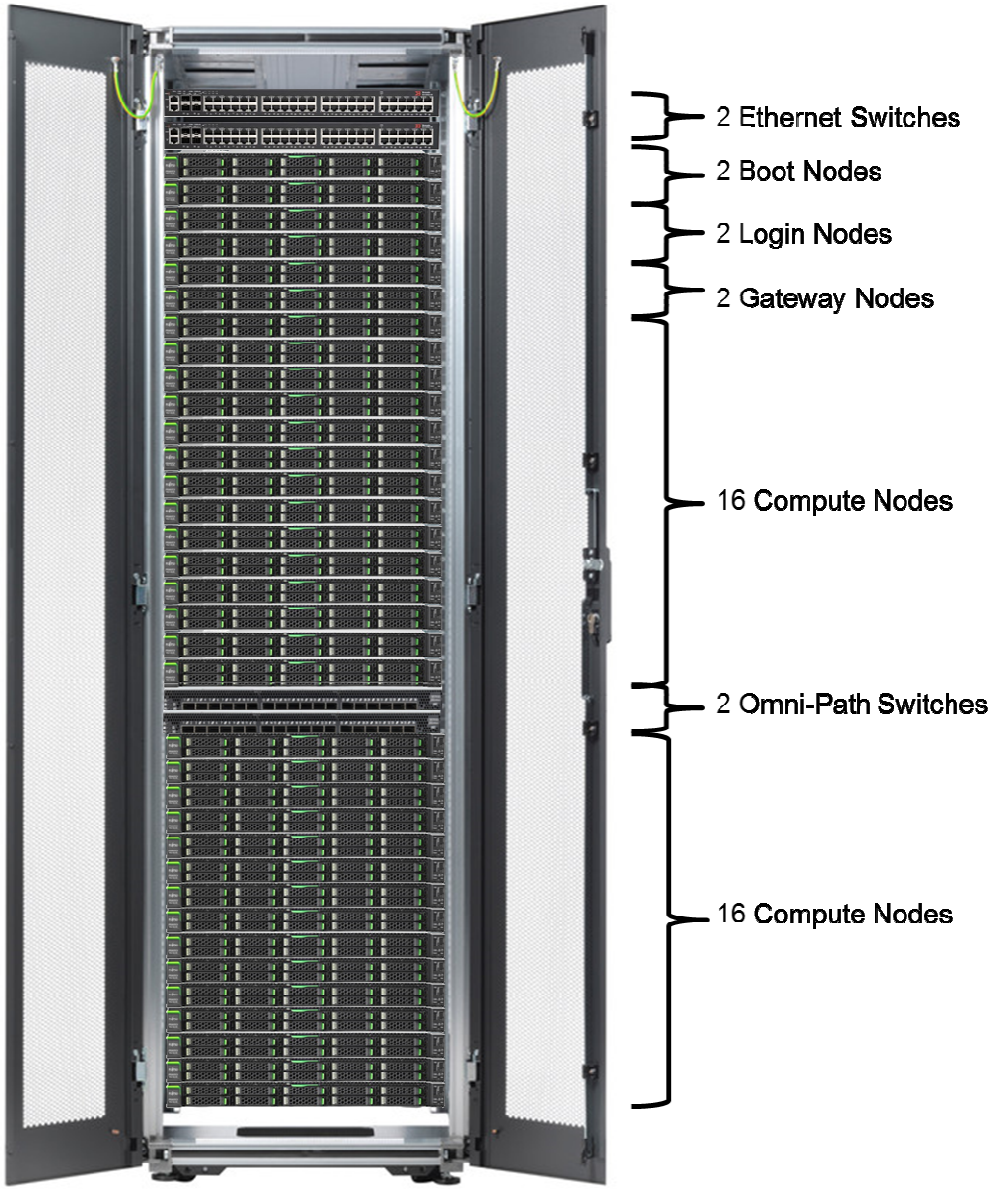

- Hardware platform prototype
  - Demonstrating the prototype's broad applicability for both HPC and data centric applications
- Exascale I/O investigation
  - Understanding how best to exploit NVRAM
- Systemware development:
  - Producing the necessary software to enable Exascale application execution on the hardware platform
- Application co-design
  - Understanding individual application I/O profiles and typical I/O workloads on shared systems running multiple different applications

# Hardware

- Motherboard developed at Fujitsu factory in Augsburg







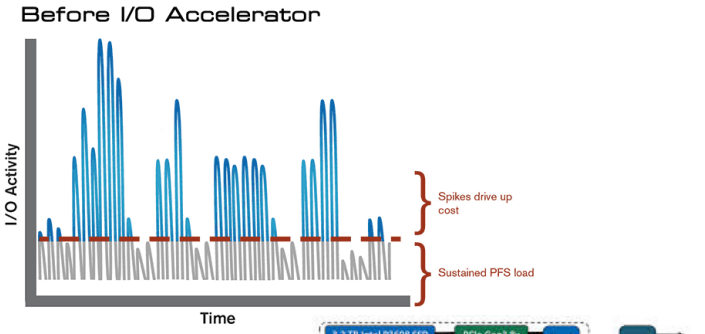

Final configuration to be confirmed, but will likely be split over two racks.

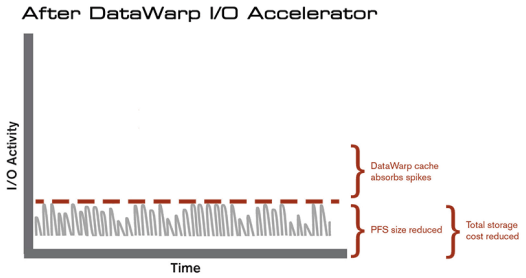
# Systemware

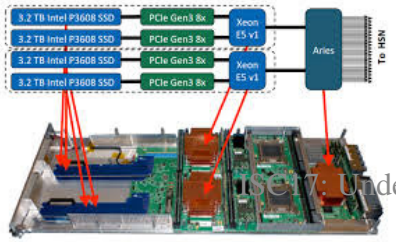

- Work on adapting job scheduler (SLURM)
- Development of a data scheduler
- Object stores as alternatives to file systems
  - DAOS (Distributed Application Object Storage)
  - dataClay
- Multi-node NVRAM file system
  - echoFS

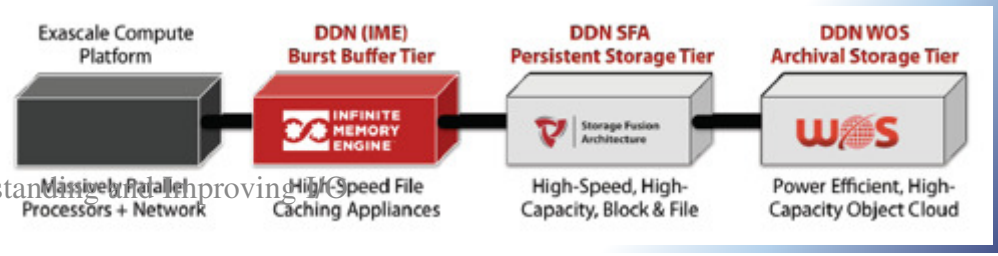
# Burst Buffer

- Non-volatile already becoming part of HPC hardware stack
- SSDs offer high I/O performance but at a cost
  - How to utilise in large scale systems?
- Burst-buffer hardware accelerating parallel filesystem
  - Cray DataWarp
  - DDN IME (Infinite Memory Engine)

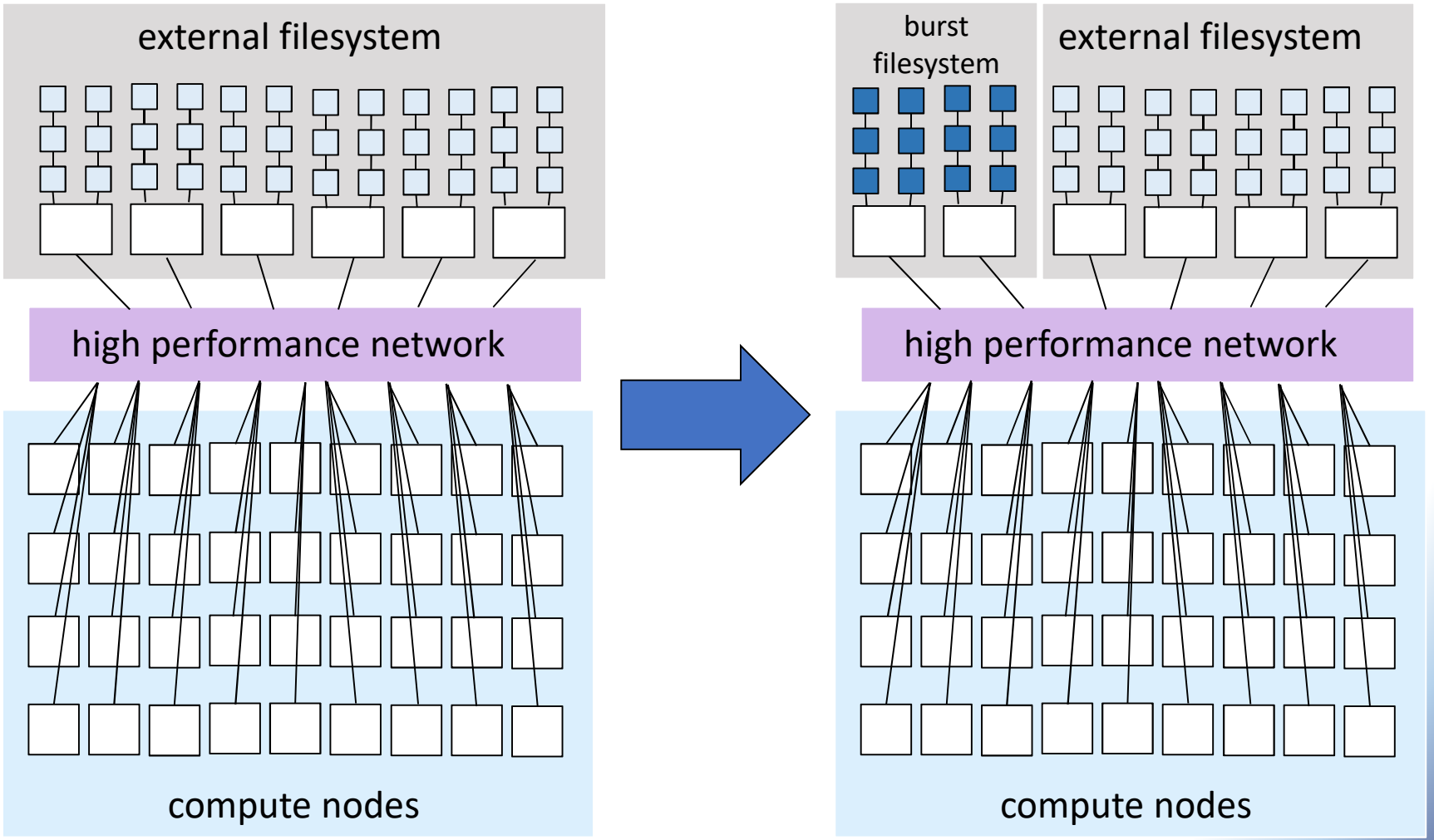


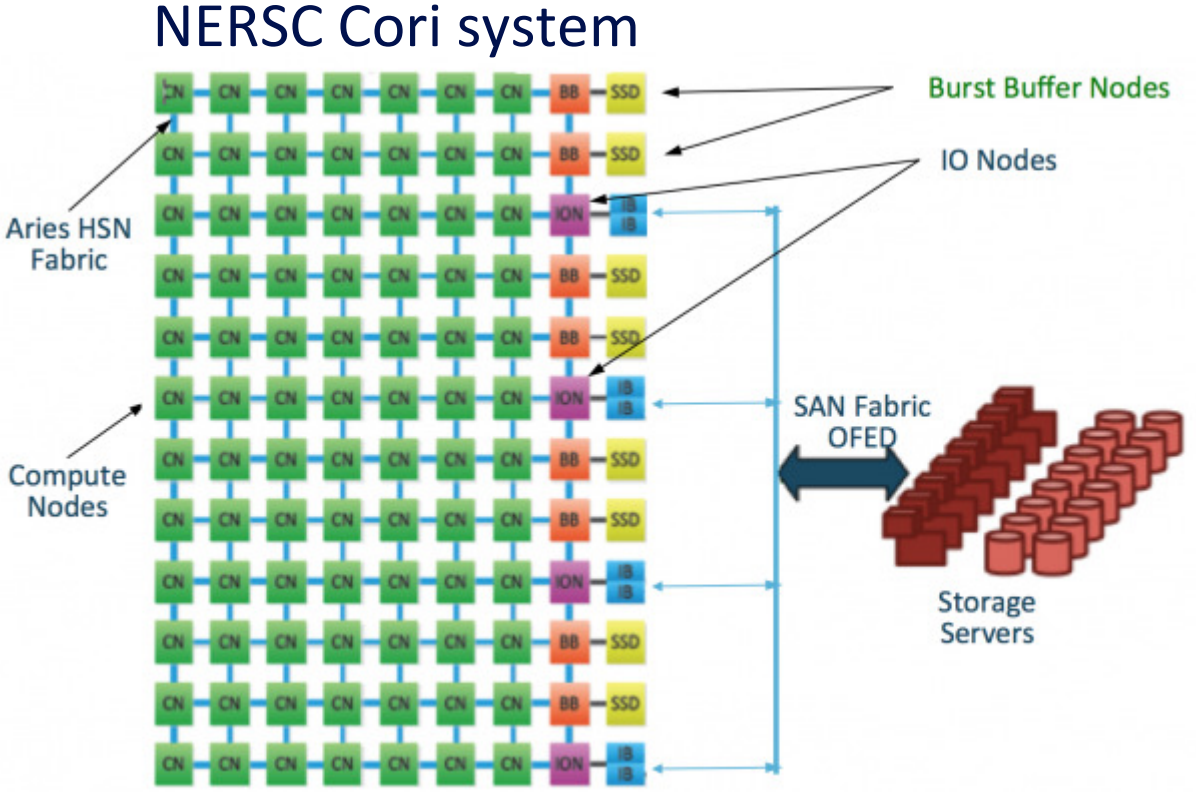



# Burst buffer



# Burst buffer

- Burst buffer is (generally) separate resource to filesystem
- Managing usage/allocation/data movement is user responsibility
- Storage has compute
- Resource is external
- Scheduling can be separate
  - Usage not required



# Summary


- Please don’t hesitate to ask questions!
- We are doing practicals
  - But if you’re not confident in programming we have other options
- We are aiming at different experience levels so if it’s too easy/you know it already let us know