Using Non-Volatile Memory

Practical Persistent Memory Programming

Adrian Jackson
EPCC
a.jackson@epcc.ed.ac.uk
@adrianjhpc

Aims

- Understand persistent memory hardware and software
- Understand the different aspects that can impact performance, and the shared/private nature of the resources
- Learn how to program persistent memory
- Get hands on with persistent memory hardware

Aims cont

- Understand data movement and think about application data requirements
- Understanding I/O and data movement, particularly persistence requirements, is hard
- Thinking about different ways you can do I/O or storing data

Format

- Lectures and practicals

- Slides and exercise material available online:
  - https://github.com/NGIOproject/PMTutorial
  - Exercises will be done on remote machine (NEXTGenIO prototype)
  - We will give you accounts on these

Timetable

- Day 1: 15th January
  - 14.00 Introduction
  - 14.15 Hardware, I/O and storage
  - 15.00 Practical: Streams and IOR and Using different mount points
  - 15.30 Low-level persistent memory programming
  - 16.15 Persistent thinking
  - 16.30 Close
  - Homework practical: Persistent memory programming
- Day 2: 22nd January
  - 14.00 Higher-level persistent memory programming
  - 15.00 Practical: Using PMDK key stores
  - 15.30 Other approaches
  - 16.30 Finish

EPCC’s PRACE Training Centre

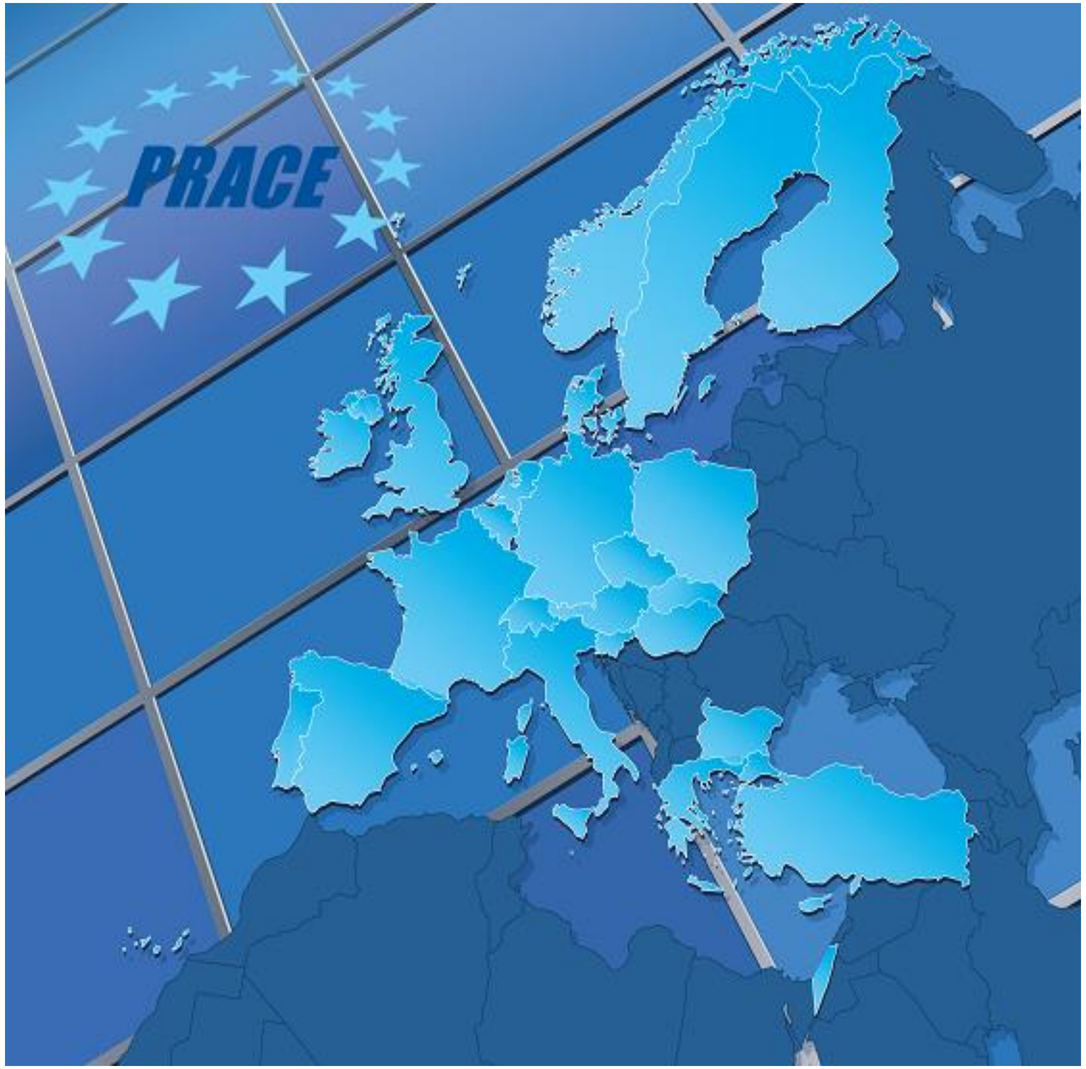
CAPACITIES

e-infrastructure

Currently 25 members

- Austria
- Belgium
- Bulgaria
- Cyprus
- Czech Republic
- Denmark
- Finland
- France
- Germany
- Greece
- Hungary
- Ireland
- Israel
- Italy
- Netherlands
- Norway
- Poland
- Portugal
- Slovakia
- Slovenia
- Spain
- Sweden
- Switzerland
- Turkey
- UK

UK represented by EPCC
(on behalf of EPSRC)



Ten PRACE Training Centres (PTCs)

Hubs for world-class HPC training for researchers in Europe
www.training.prace-ri.eu

- Barcelona Supercomputing Center (Spain)
- CINECA - Consorzio Interuniversitario (Italy)
- CSC - IT Center for Science Ltd (Finland)
- EPCC at the University of Edinburgh (UK)
- Gauss Centre for Supercomputing (Germany)
- Maison de la Simulation (France)
- GRNET (Greece)
- ICHEC (Ireland)
- IT4I (Czech Republic)
- SURFsara (Netherlands)

PRACE support

- PRACE also funds catering and other expenses for PTC courses

- Upcoming courses (at EPCC and throughout Europe)
  - www.archer.ac.uk/training/
  - www.training.prace-ri.eu

- Please fill in the course feedback form!
  - see www.archer.ac.uk/training/feedback/
  - opens on last day of course

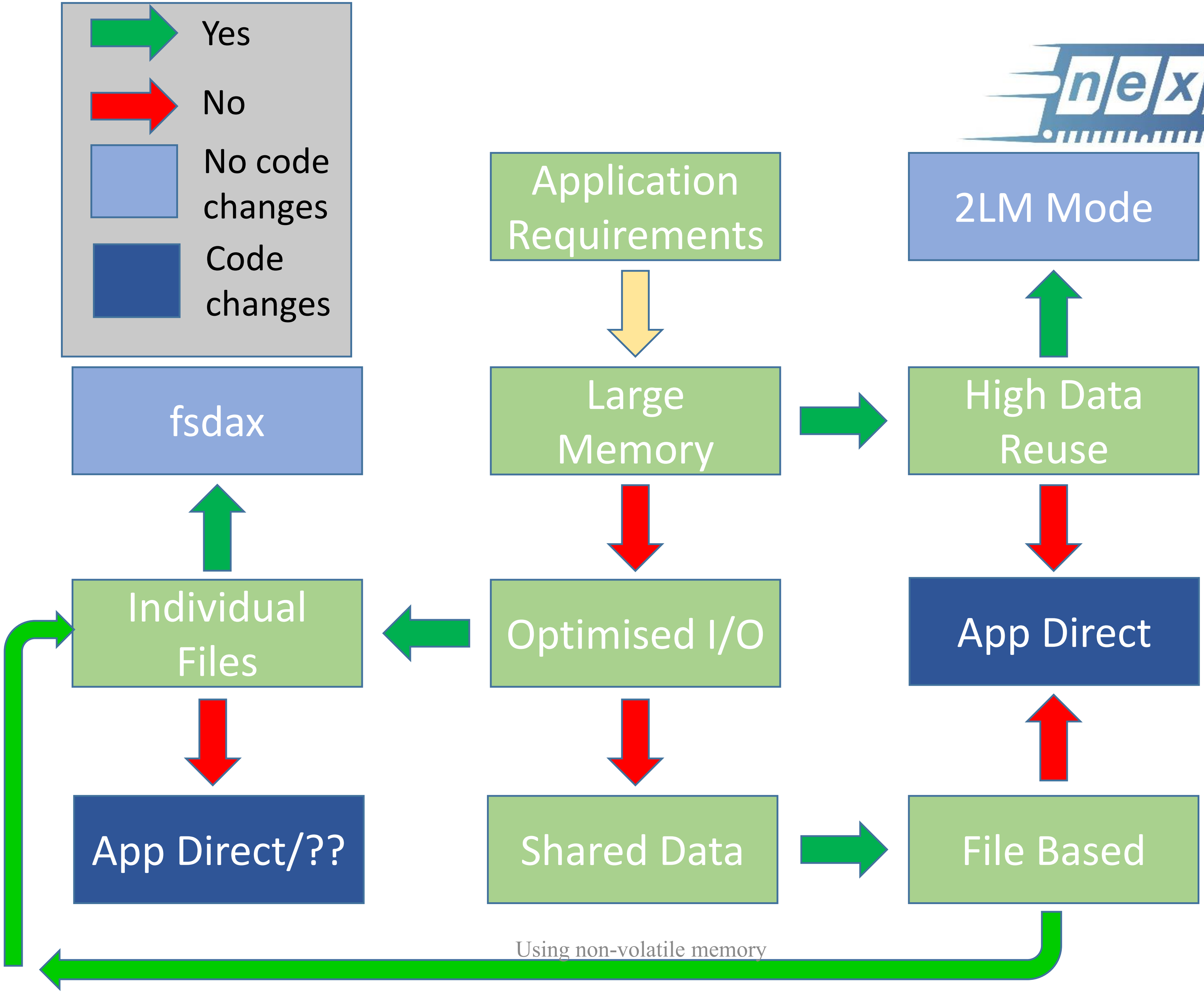
Programming persistent memory

Programming persistent memory

```
double *a, *b, *c;
pmemaddr = pmem_map_file(path, array_length,
                     PMEM_FILE_CREATE|PMEM_FILE_EXCL,
                     0666, &mapped_len, &is_pmem)
a = pmemaddr;
b = pmemaddr + (*array_size+OFFSET)*BytesPerWord;
c = pmemaddr + (*array_size+OFFSET)*BytesPerWord*2;

#pragma omp parallel for
for (j=0; j<*array_size; j++){
   a[j] = b[j]+scalar*c[j];
}
pmem_persist(a, *array_size*BytesPerWord);
```

Programming persistent memory



🟩 Yes

🟥 No

▨ No code changes

■ Code changes

Application Requirements

Large Memory

High Data Reuse

2LM Mode

App Direct

Optimised I/O

Individual Files

fsdax

App Direct/??

Shared Data

File Based

Programming persistent memory

- Design and performance considerations are the challenge
  - Programming the memory is easy
- Design for functionality
  - What is persistent, when is it persistent, what failures can you tolerate, etc..
- Design for performance
  - Memory size, I/O, data access costs, etc…
- Design for hardware configurations
  - NUMA, filesystems, storage, etc…

Summary

- Please don’t hesitate to ask questions!
- There are practicals
- We are aiming at different experience levels so if it’s too easy/you know it already/it’s too difficult let us know