Optimisation of Parallel Scientific Applications on Highly Heterogeneous Modern Multicore and Multi-GPU Computing Nodes

Vladimir Rychkov Ziming Zhong

Heterogeneous Computing Laboratory
School of Computer Science and Informatics
University College Dublin

May 11, 2012

Introduction

- **Hybrid multi-CPU/GPU architectures in HPC**
  How to utilise highly heterogeneous hardware and software stack?
- **Target platforms and applications**
  Dedicated multicore and multi-GPU nodes and clusters
  Data-parallel applications dependent on data locality

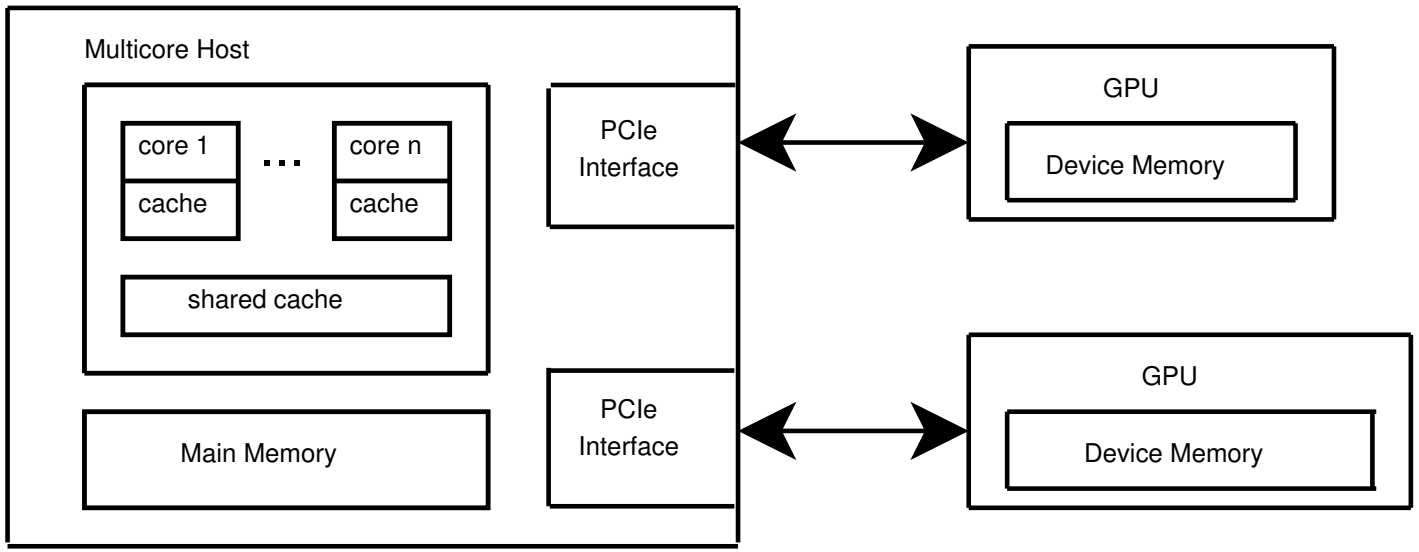
Introduction

- **Hybrid multi-CPU/GPU architectures in HPC**
  How to utilise highly heterogeneous hardware and software stack?
- **Target platforms and applications**
  Dedicated multicore and multi-GPU nodes and clusters
  Data-parallel applications dependent on data locality
- **Performance modeling and model-based optimisation**
  Realistic models of processors executing data-parallel application
  Accurate measurement of performance on each processor
  Optimal data partitioning based on the models
- **Functional performance models**
  Heterogeneous uniprocessor clusters

Introduction

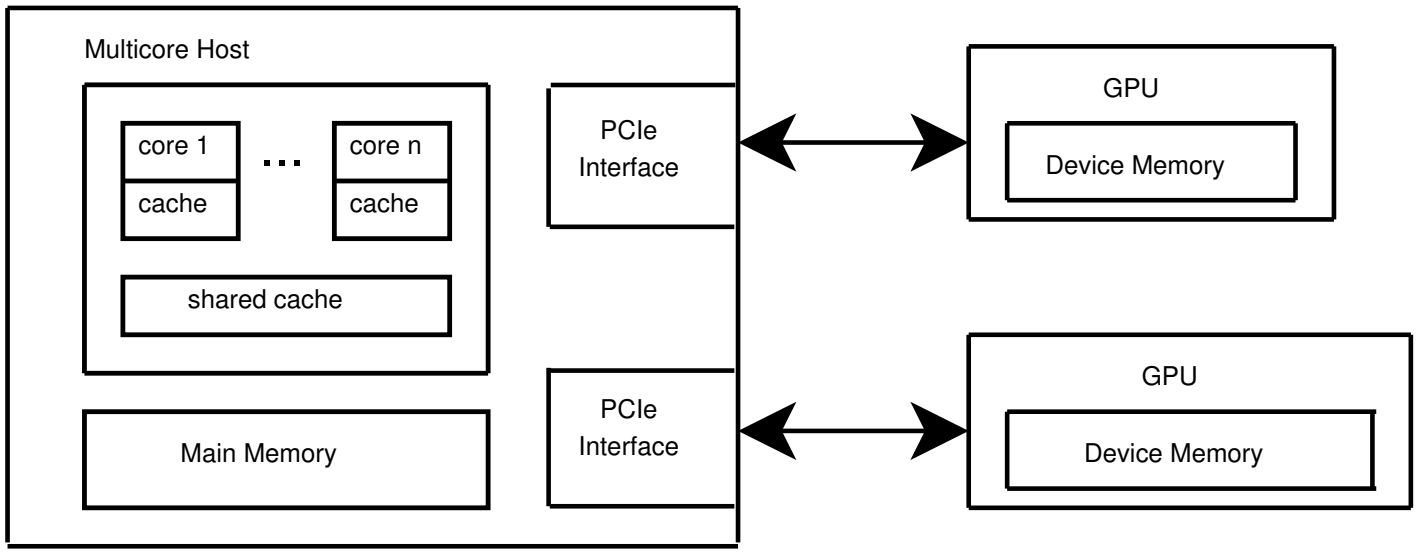
- **Hybrid multi-CPU/GPU architectures in HPC**
  How to utilise highly heterogeneous hardware and software stack?
- **Target platforms and applications**
  Dedicated multicore and multi-GPU nodes and clusters
  Data-parallel applications dependent on data locality
- **Performance modeling and model-based optimisation**
  Realistic models of processors executing data-parallel application
  Accurate measurement of performance on each processor
  Optimal data partitioning based on the models
- **Functional performance models**
  Heterogeneous uniprocessor clusters
- **How to model performance of devices on a hybrid platform?**
  Different programming models and software
  Shared resources of different speed and capacity

Hybrid HPC Platform



- Highly heterogeneous devices
- Memory: shared and distributed
- Resource contention: memory, PCI Express

Hybrid HPC Platform



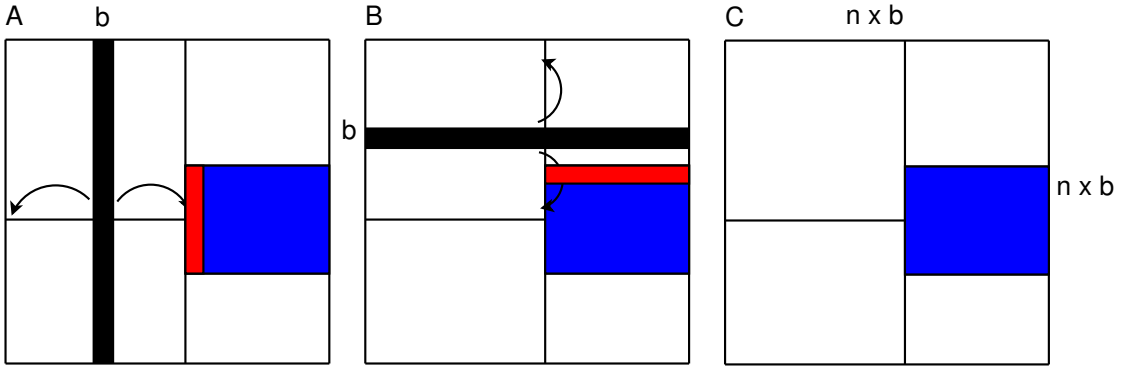
- Highly heterogeneous devices
- Memory: shared and distributed
- Resource contention: memory, PCI Express
- **Can be modelled as a heterogeneous distributed-memory system**
  Functional performance models of devices
  Intra-node data partitioning between devices

Outline

- Functional performance models of multiple cores and GPUs
- Accurate measurement of performance of devices
- Optimisation of computational kernels
- Intra-node data partitioning based on FPMs of devices

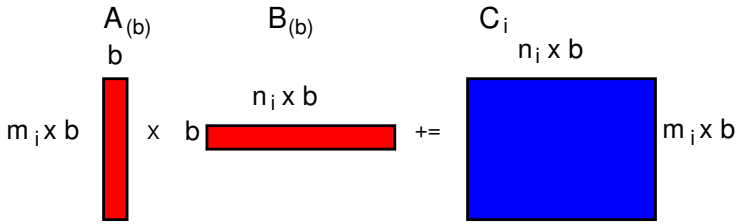
Heterogeneous Data-Parallel Application

- Heterogeneous matrix multiplication



Beaumont, O., Boudet, V., Rastello, F., Robert, Y.: Matrix Multiplication on Heterogeneous Platforms. IEEE Trans. Parallel Distrib. Syst. 12(10), 1033-1051 (2001)

- Representative kernel, GEMM: $C_i += A_{(b)} \times B_{(b)}$



Computation is proportional to the area of submatrix C_i
The same memory access pattern as the whole application

Functional Performance Models of Devices

- Optimised GEMM routines: MKL/ACML(CPU), CUBLAS(GPU)

Functional Performance Models of Devices

- Optimised GEMM routines: MKL/ACML(CPU), CUBLAS(GPU)

- CPU cores compete with each other within a socket
  $s_c(x)$ - speed of a socket executing CPU GEMM on c cores, with submatrices x/c

Functional Performance Models of Devices

- Optimised GEMM routines: MKL/ACML(CPU), CUBLAS(GPU)
- CPU cores compete with each other within a socket
  $s_c(x)$ - speed of a socket executing CPU GEMM on c cores, with submatrices x/c
- GPU depends on a host process, which handles data transfers
- GPU performance can be measured only within some range
  $g(x)$ - combined speed of a GPU and its dedicated core executing CUBLAS, can be defined at $[0, \infty)$ for out-of-core GEMM

Performance Measurement on Hybrid Platforms

- Bind processes to cores
  - to avoid potential performance degradation resulting from automatic rearranging of processes by operating system
- Synchronise processes
  - to ensure resources are shared between processes
- Repeat experiments multiple times
  - until the results are proved statistically reliable

Performance Measurement on Hybrid Platforms

- Bind processes to cores
  - to avoid potential performance degradation resulting from automatic rearranging of processes by operating system
- Synchronise processes
  - to ensure resources are shared between processes
- Repeat experiments multiple times
  - until the results are proved statistically reliable

- Performance of multiple cores and GPUs can be measured separately
- Simultaneous benchmark of multiple cores on a socket
- Combined benchmark for a GPU and its dedicated core

Experimental Platform

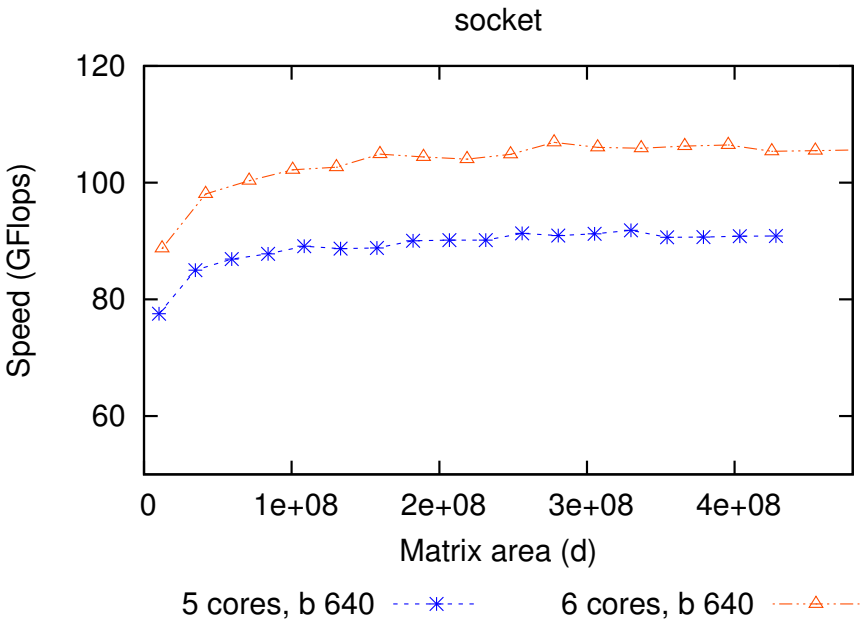
Multicore multi-GPU node ig.eecs.utk.edu

Innovative Computing Laboratory, University of Tennessee, USA

	CPU (AMD)	GPUs (NVIDIA)	
Architecture	Opteron 8439SE	GeForce GTX480	Tesla C870
Core Clock	2.8 GHz	700 MHz	600 MHz
Number of Cores	8×6 cores	480 cores	128 cores
Memory Size	64 GB	1536 MB	1536 MB
Memory Bandwidth		177.4 GB/s	76.8 GB/s

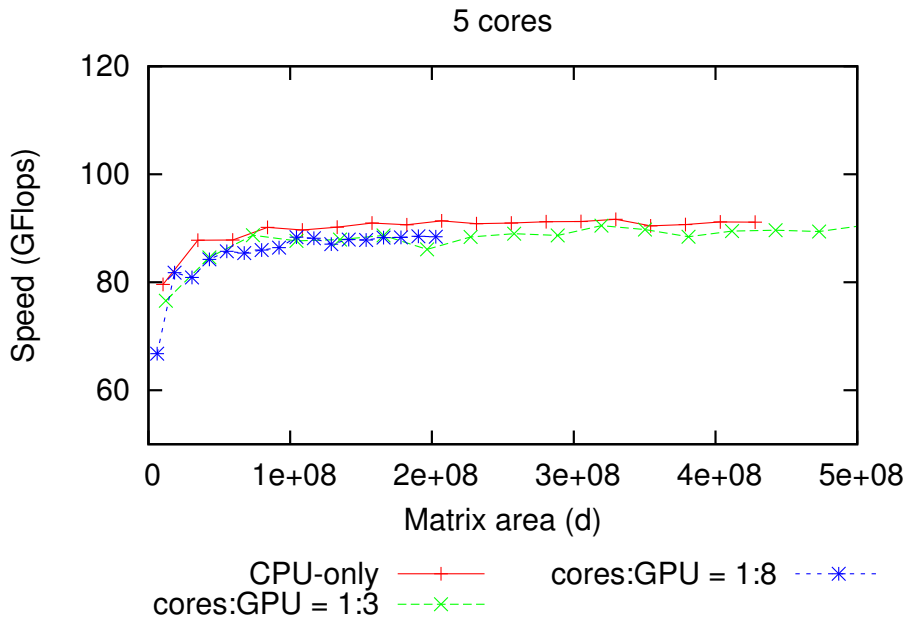
Functional Performance Models of Multiple Cores

Speed functions of multiple cores



- $s_5(x)$, $s_6(x)$ - speed functions for 5 and 6 cores on a socket

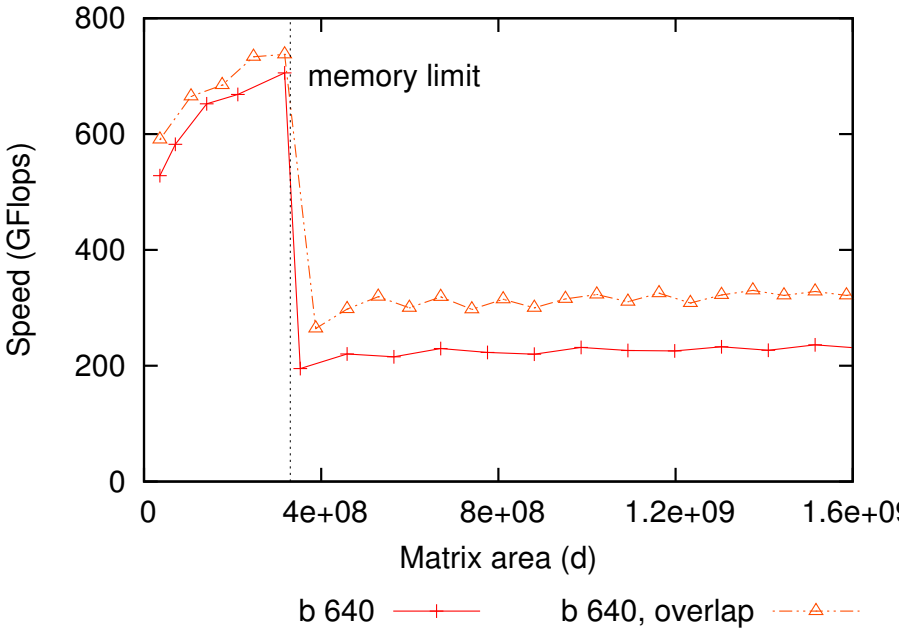
Resource contention with GPU



- $s_5(x)$ remains stable when sharing resource with GPU

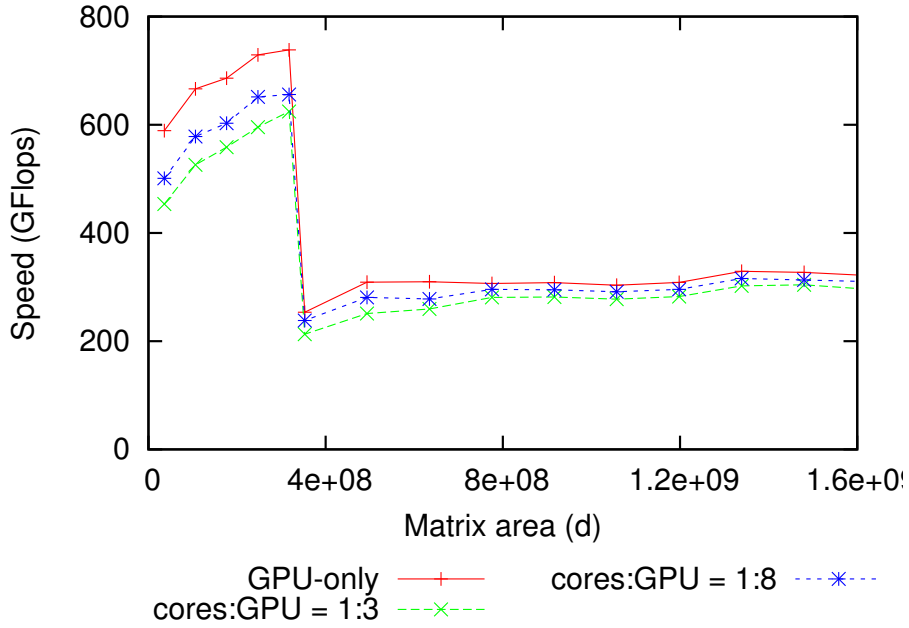
Functional Performance Models of GPUs

Speed functions of GeForce GTX480



- Out-of-core implementation extends the range of problem sizes
- Overlapping improves performance

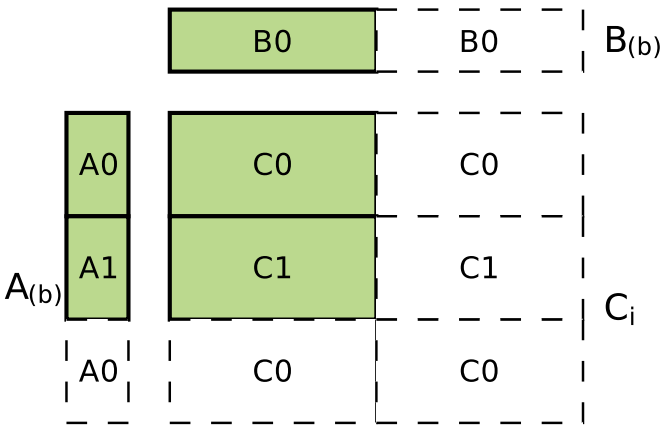
Resource contention



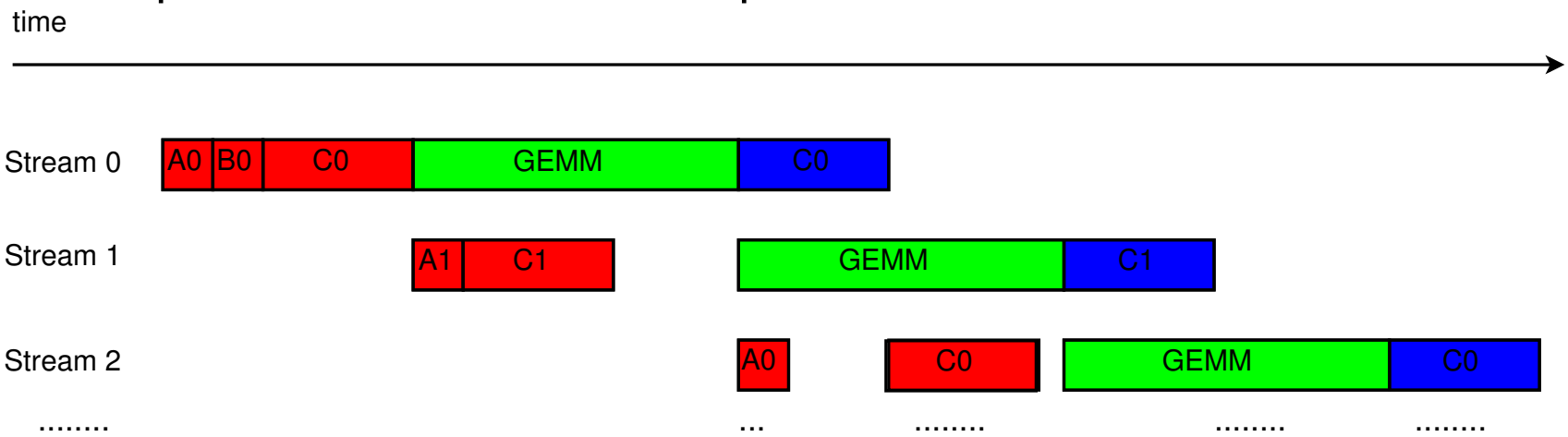
- Performance decreases around 10% from resource contention

Optimisation of the Kernel

- Out-of-core computations



- Overlap data transfers and computation



FPM-based Data Partitioning on Hybrid Platform

Execution time of the parallel matrix multiplication application on different configurations on the hybrid server

Matrix size	Execution time (sec) CPUs	GPUs	CPUs+GPUs
12800×12800	14.6	10.5	5.8
19200×19200	43.4	32.5	16.2
25600×25600	99.8	147.9	38.2
32000×32000	189.2	265.3	114.1

Ongoing and Future Study

Target platform: multi-GPU servers - more resource contention

- Accurate measurement technique
- FPM of multiple GPUs on the same server
- Design of computational kernel for multi-GPU servers
- Experiments on data partitioning with different models

Publications

- Zhong, Z., Rychkov, V., Lastovetsky, A. "Data Partitioning on Heterogeneous Multicore Platforms", Cluster 2011, Austin, Texas, USA, IEEE Computer Society, pp. 580-584 (2011).
- Zhong, Z., Rychkov, V., Lastovetsky, A. "Functional Performance Models of Scientific Applications for Heterogeneous Multicore and Multi-GPU Systems" (submitted to Euro-Par 2012, Rhodes Island, Greece).

Output

Project web page: `http://hcl.ucd.ie/project/fupermod`

Software

- Implemented in the FuPerMod package, developed at HCL
- Based on system and mathematical software: C/C++, MPI, Autotools, GNU Scientific Library, Boost C++ libraries, BLAS, CUDA Toolkit

Team

- 1 postdoctoral researcher: Vladimir Rychkov
- 2 PhD students: David Clarke, Ziming Zhong

Collaboration

Hardware

- Multicore multi-GPU servers (Innovative Computing Laboratory, University of Tennessee, USA)
- Multicore multi-GPU cluster (High Performance Computing & Architectures group, University Jaume I, Spain)
- Grid'5000 (INRIA, CNRS, RENATER, France)

Collaboration

- Leonel Sousa, Alexandar Ilic (Institute of System Engineering, Computer Research and Development, Portugal)
- Enrique Quintana Orti (High Performance Computing & Architectures, University Jaume I, Spain)

Financial Support

Science Foundation Ireland

UCD CSI

China Scholarship Council

Complex HPC action COST