Heterogeneous- and NUMA-aware Scheduling for Many-Core Architectures

Panayiotis Petrides(*)

Pedro Trancoso (*)(**)

(*) Computer Science Department
University of Cyprus

(**) Computer Science and Engineering Chalmers University of Technology

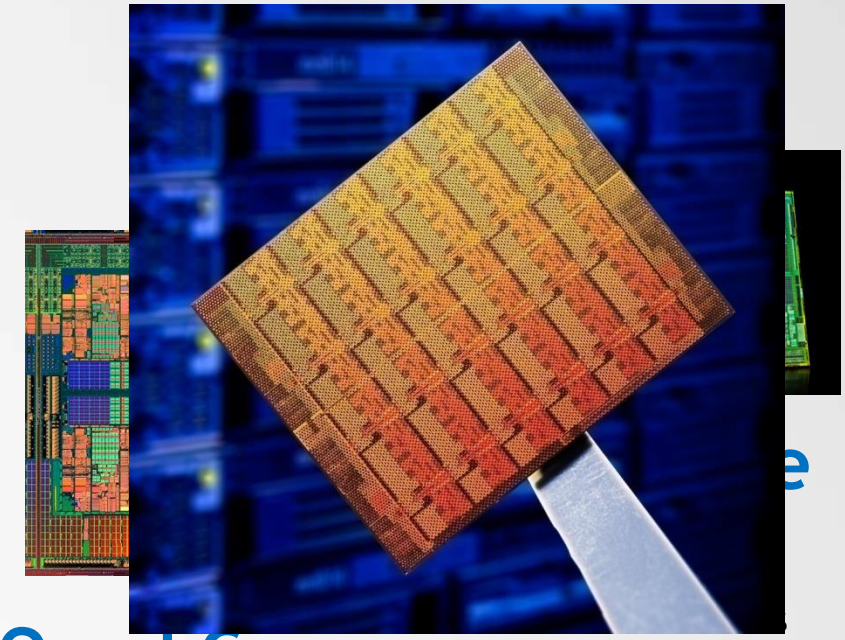
CASPER: Computer Architecture System Performance Evaluation Research

Outline

- Motivation
- Scheduling Policy
- Experimental Results
- Conclusions

Motivation

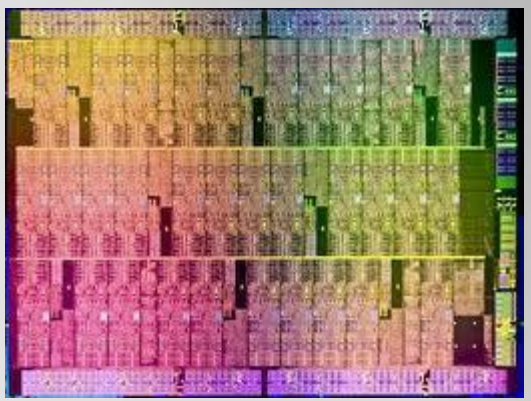
Dual Core



Quad Core

Intel SCC

CMP with 48 low power cores

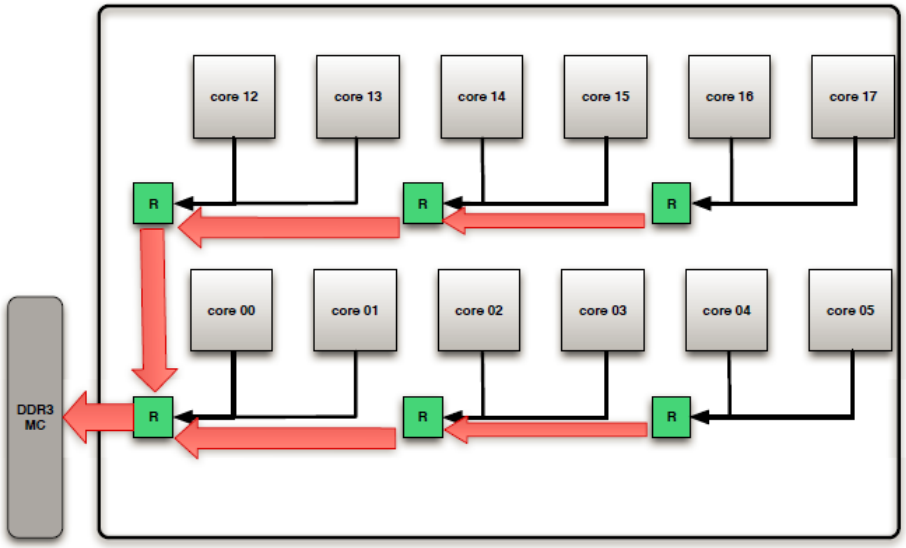


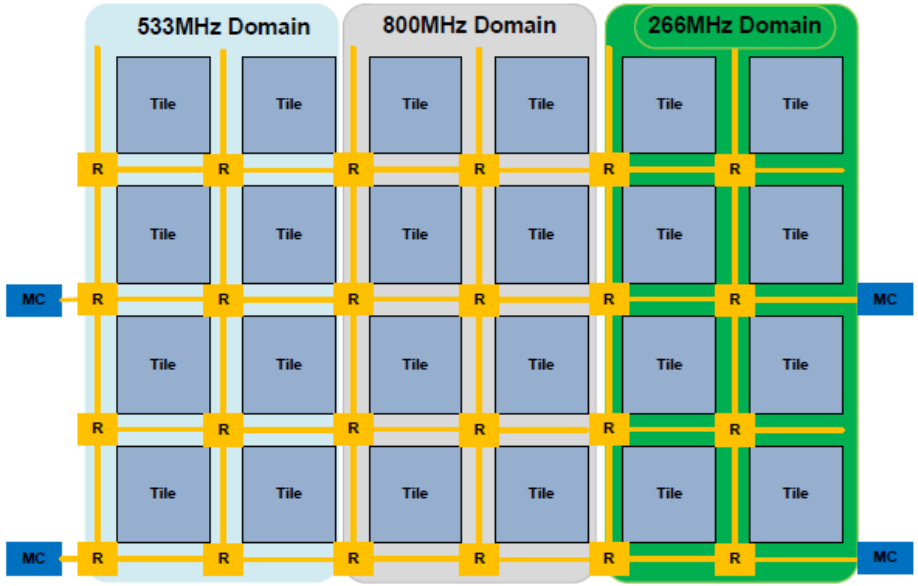
Many-core Array

CMP with ~10s ~100s low power cores

Motivation

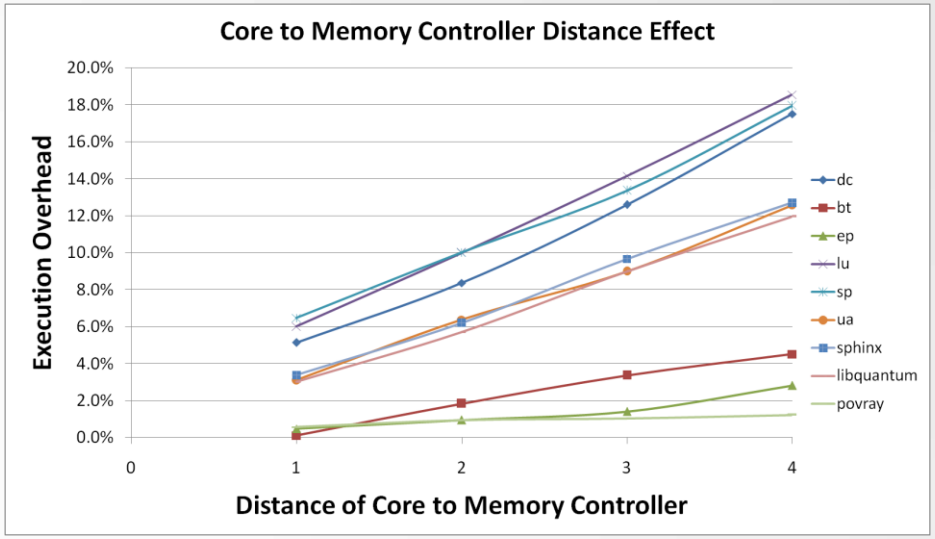
1. Distance of Core to the Memory Controller
   - Non Uniform Memory Access
2. Resources of Different Core Frequency
3. Memory Controller Accesses Contention

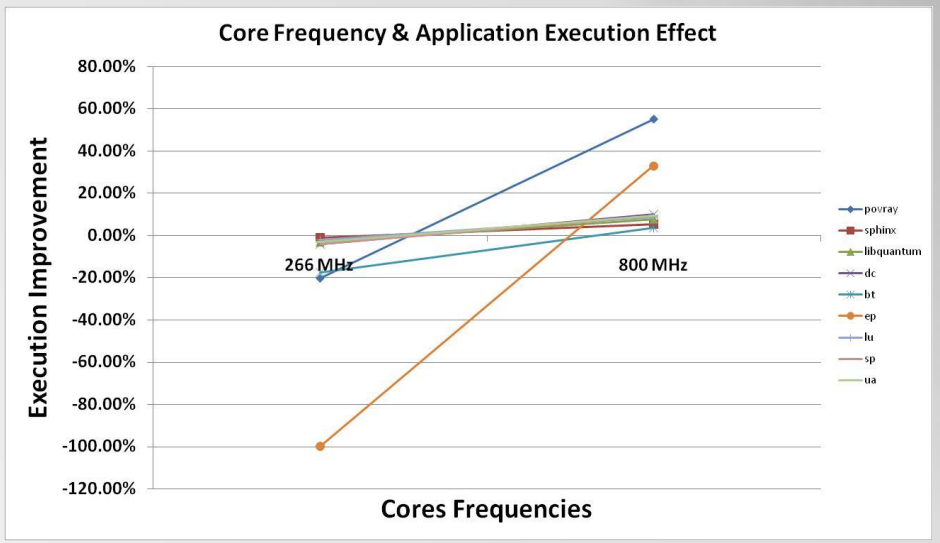


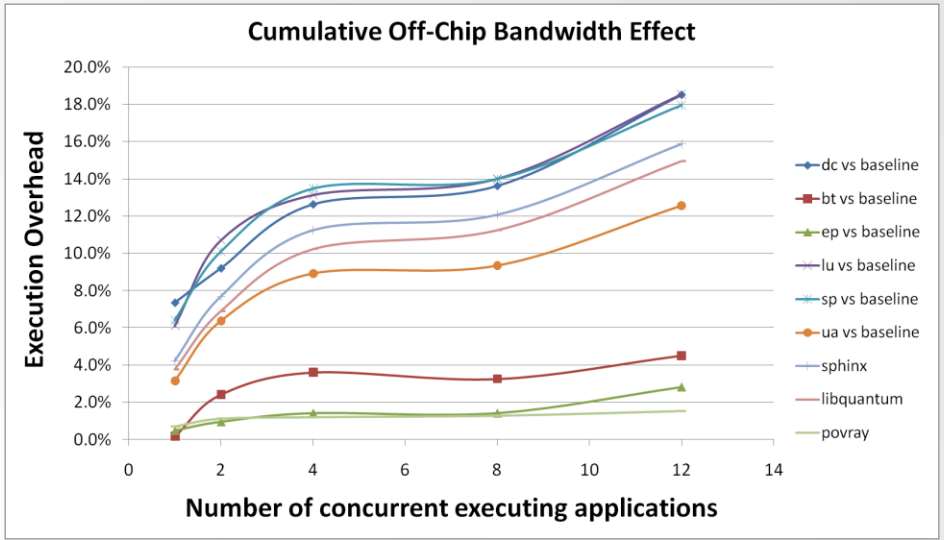


Motivation

Executing SPEC CPU2006 and NAS Benchmark Suites on Intel SCC







Outline

- Motivation
- Scheduling Policy
- Experimental Results
- Conclusions

Scheduling Policy - Characterizing Applications

Determine how the distance and core frequency factors influence applications execution

$$D(H) = a \times H \qquad a = \frac{IPC_x - IPC_y}{H_x - H_y}$$

$$F(f) = b \times f, \qquad b = \frac{IPC_x - IPC_z}{f_i - f_j}$$

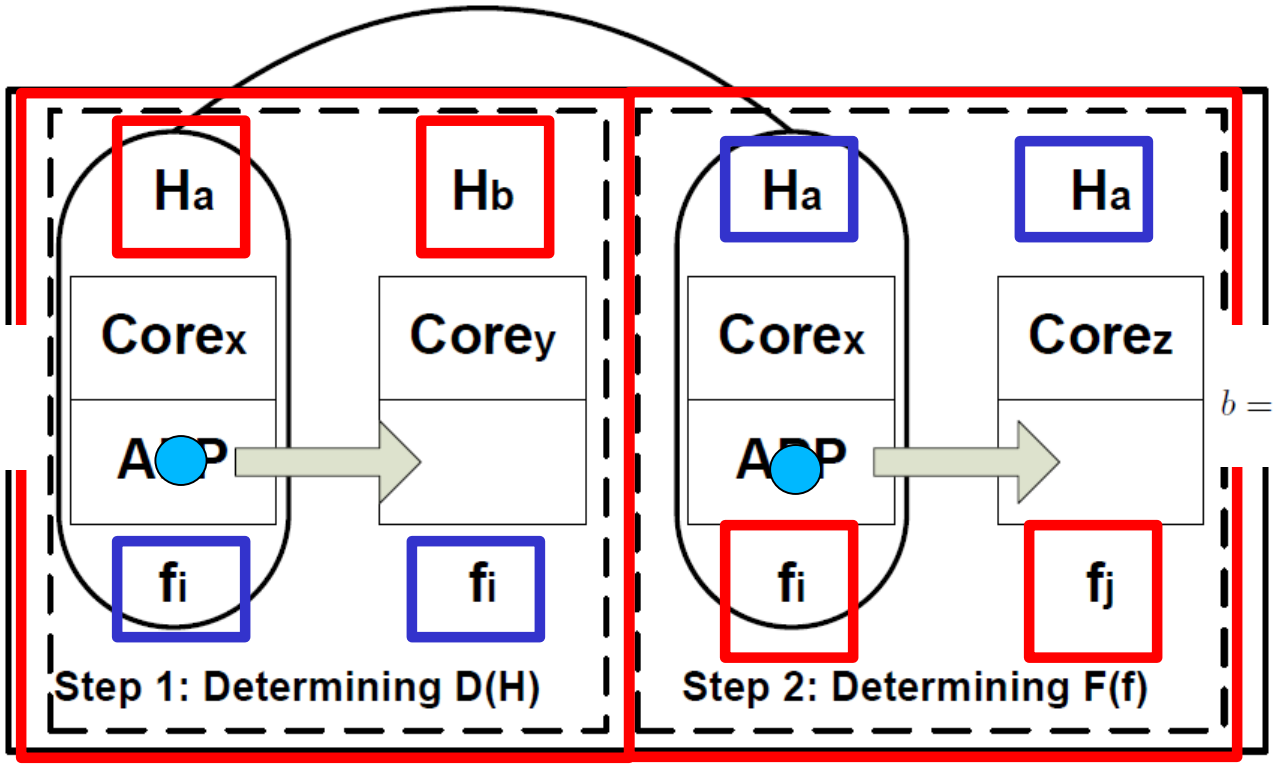
Both Frequency and Distance change in a linear way

Scheduling Policy - Characterizing Applications

System Prerequisites in order to determine factors of influence:

Discrete Couples of cores with one factor varying and the other one constant

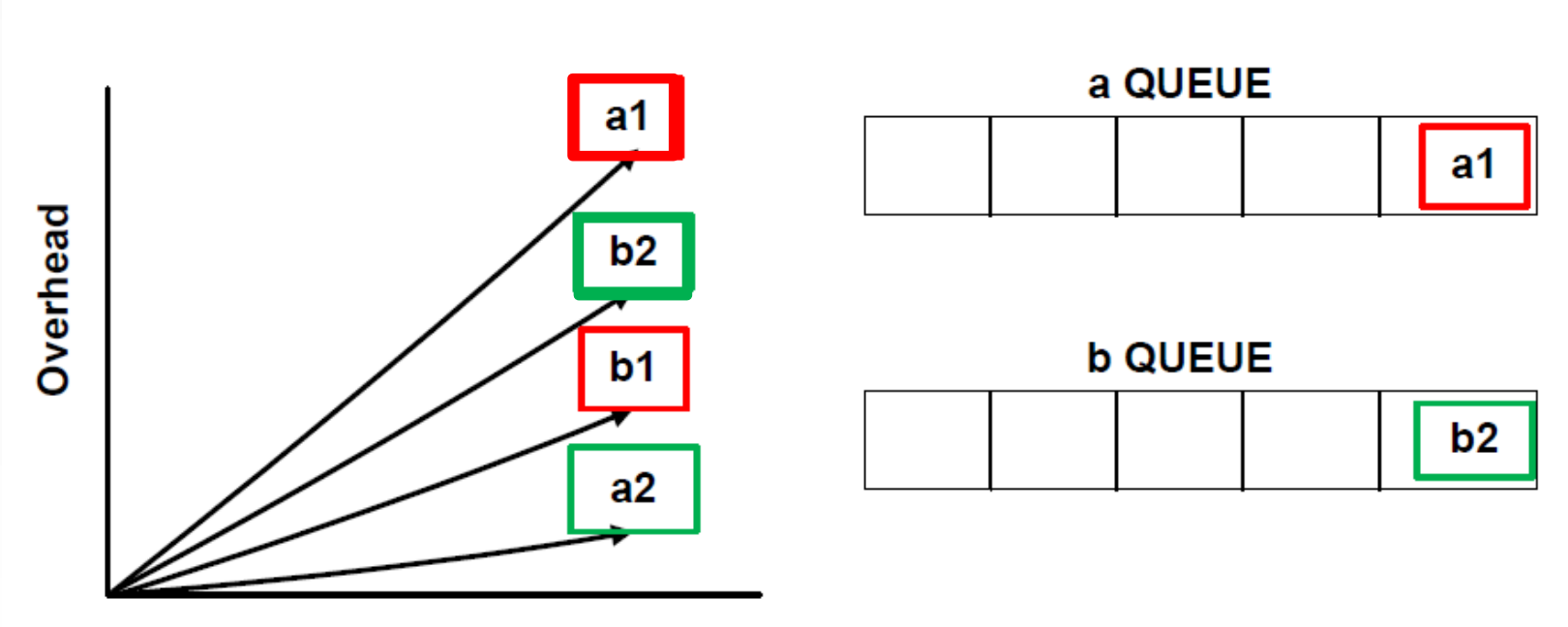
$$a = \frac{IPC_x - IPC_y}{H_x - H_y}$$



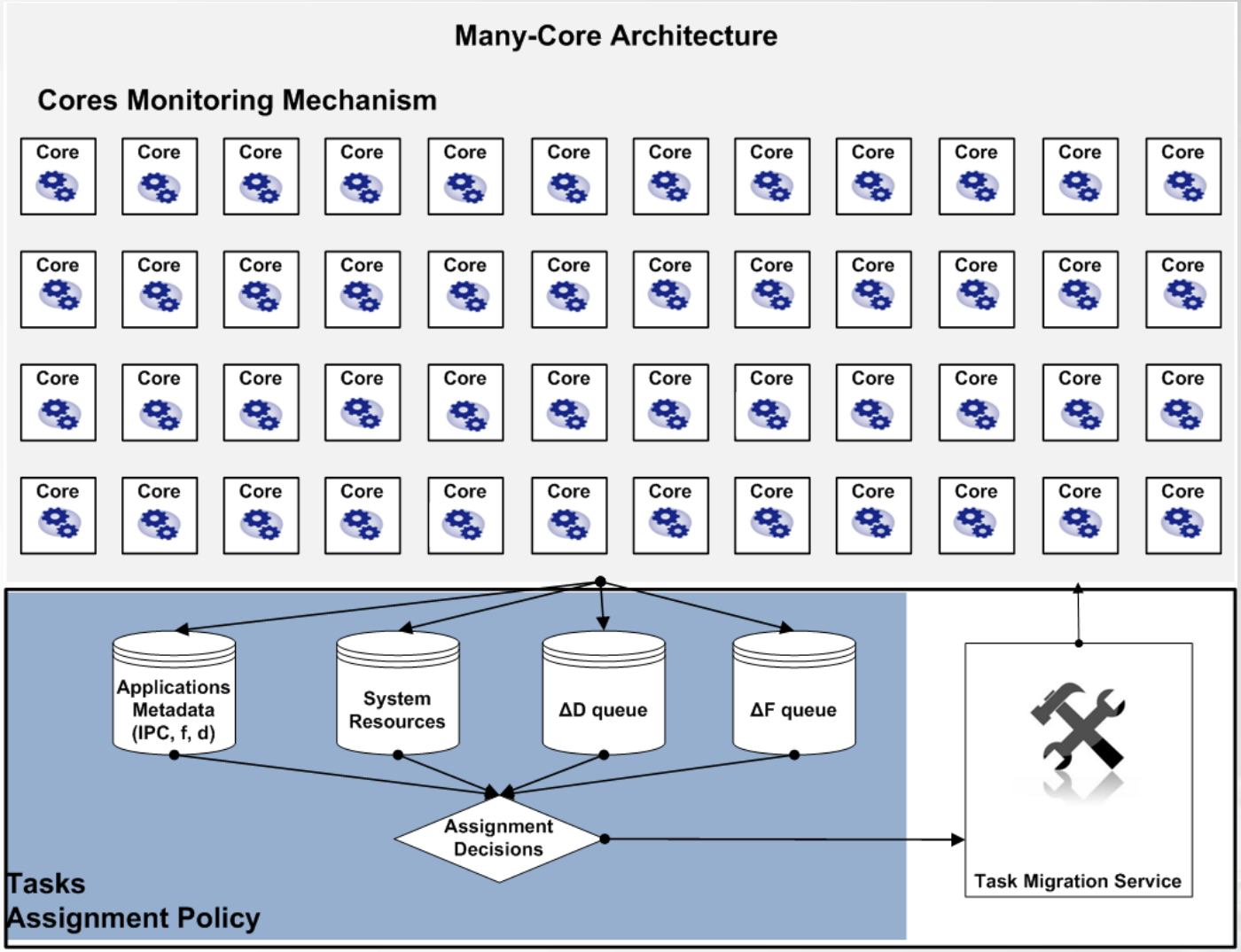
$$b = \frac{IPC_x - IPC_z}{f_i - f_j}$$

Scheduling Policy: Implementation

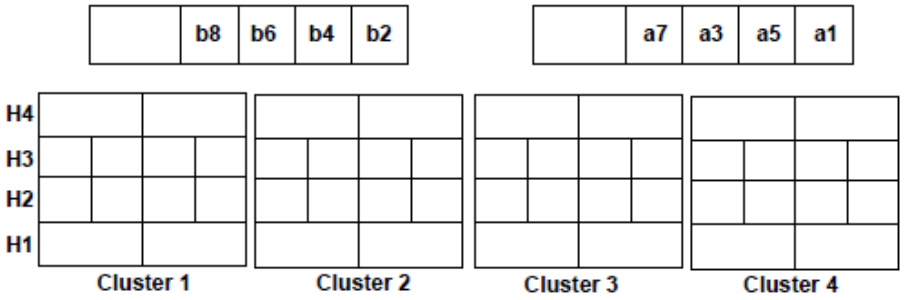
- In order to determine applications behavior we monitor their execution
  - Construct at each monitor phase the corresponding queues of a and b

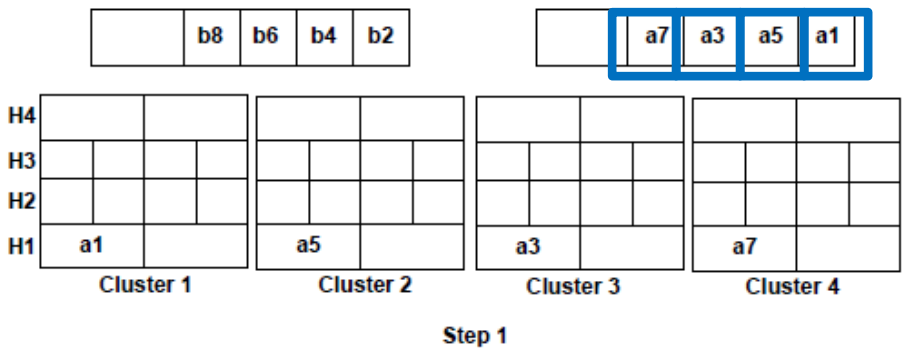


Scheduling Policy: Implementation

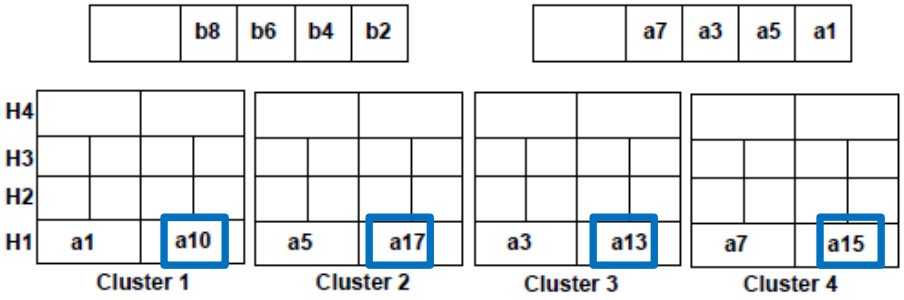


Scheduling Policy: Implementation

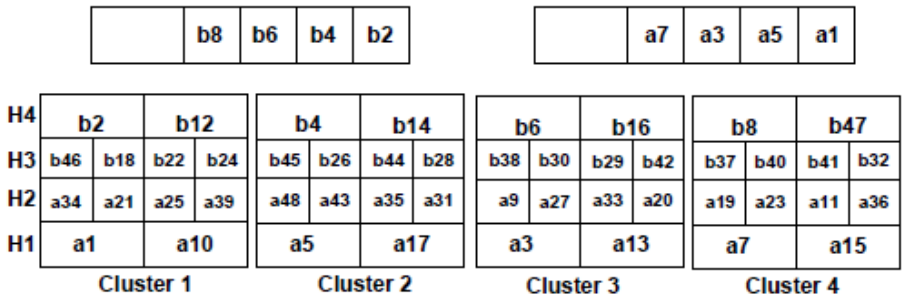




Step 1



Step 2



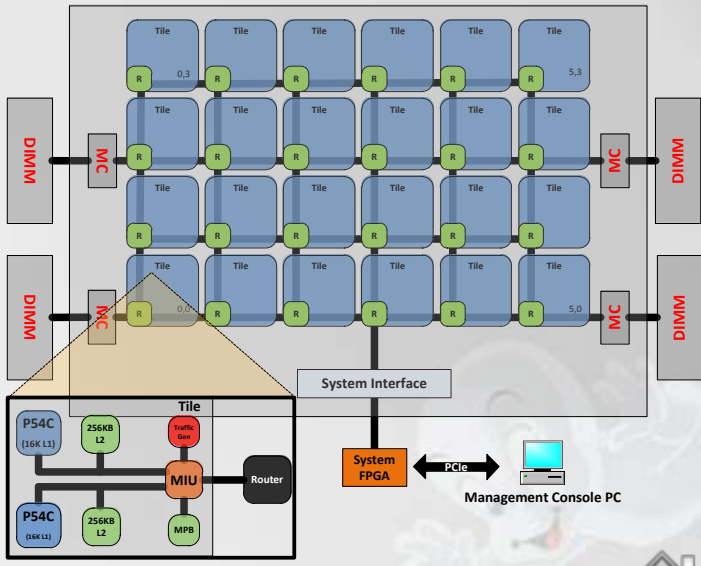
Step n

Outline

- Motivation
- Scheduling Policy
- Experimental Results
- Conclusions

Experimental Setup

- Intel SCC Processor
  - 48-core P54C Core Architecture
  - 4 DDR3 Memory Controllers per 12-cores
  - Linux kernel running at each core
- Applications from SPEC CPU2006 and NAS benchmarks (medium working size sets)
  - Povray (compute-bound)
  - Sphinx (Medium memory-bound)
  - Libquantum (High memory-bound)
- Checkpointing/Resuming using CryoPID library
  - Migration overhead < 1%

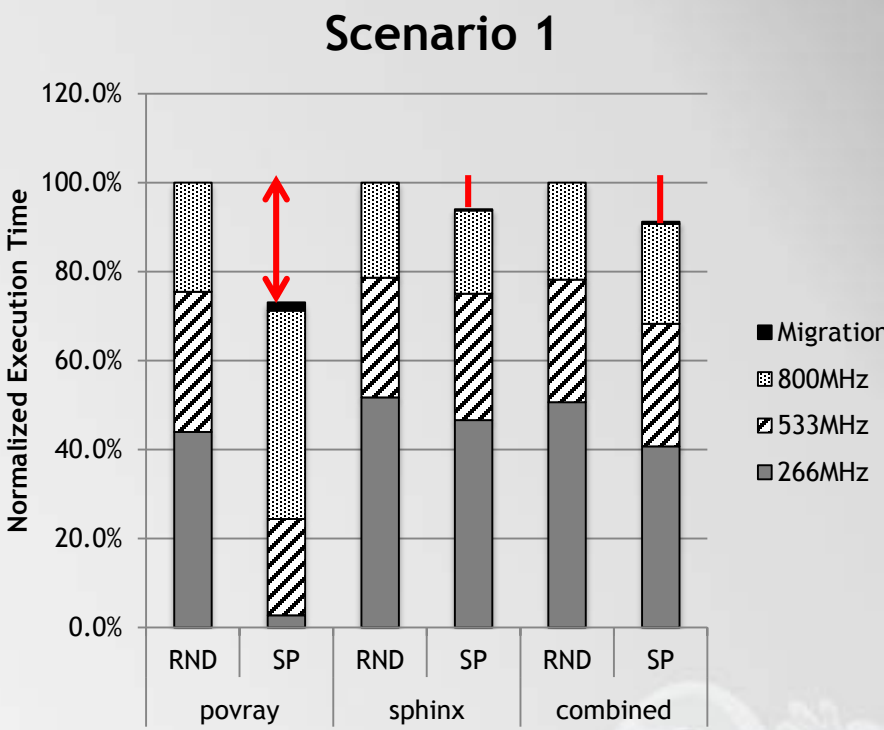


Evaluating Scheduling Policy

Scenario 1: Compute-bound and Memory-bound applications

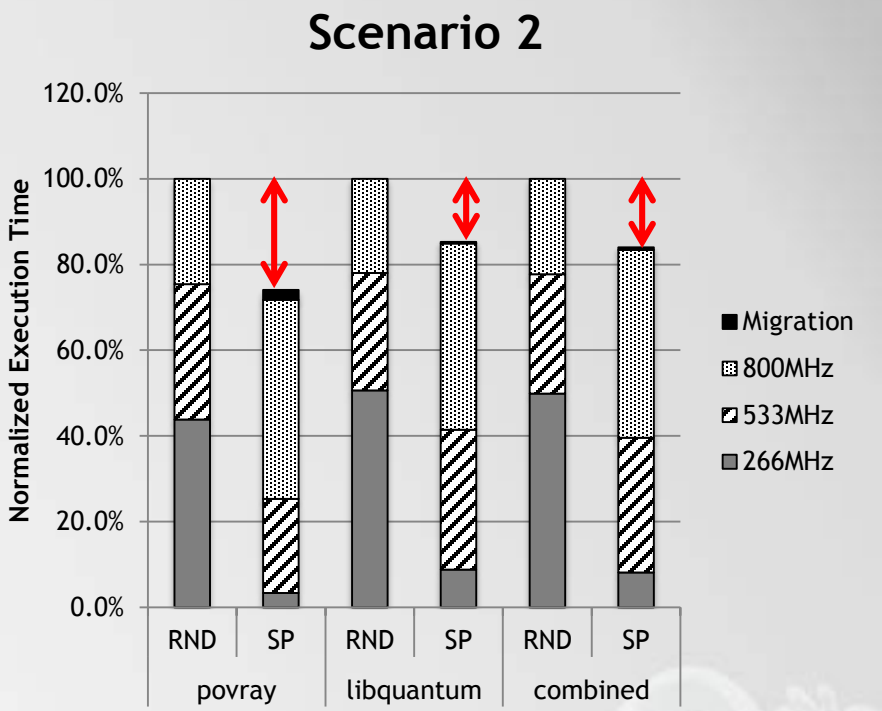
Scheduler Placement

sphinx	sphinx	povray	povray	povray	povray	povray	povray	povray	povray	sphinx	sphinx
36	37	38	39	40	41	42	43	44	45	46	47
sphinx	sphinx	sphinx	sphinx	povray	povray	povray	povray	sphinx	sphinx	sphinx	sphinx
24	25	26	27	28	29	30	31	32	33	34	35
sphinx	sphinx	povray	povray	povray	povray	povray	povray	povray	povray	sphinx	sphinx
12	13	14	15	16	17	18	19	20	21	22	23
sphinx	sphinx	sphinx	sphinx	povray	povray	povray	povray	sphinx	sphinx	sphinx	sphinx
0	1	2	3	4	5	6	7	8	9	10	11
533MHZ				800MHZ				266MHZ			



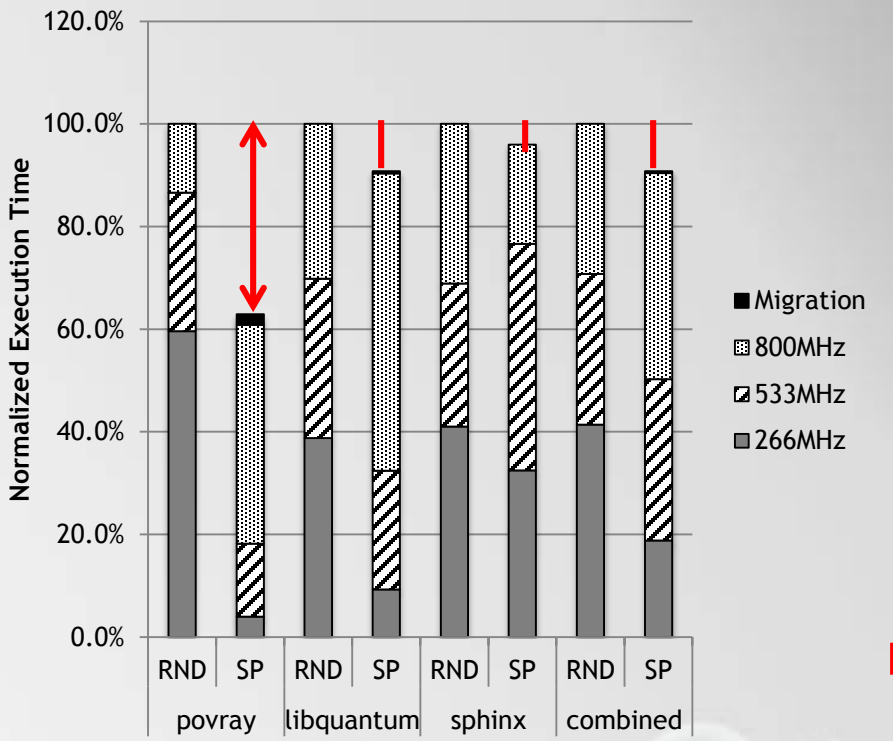
Evaluating Scheduling Policy

Scenario 2: Compute-bound and Memory-bound applications



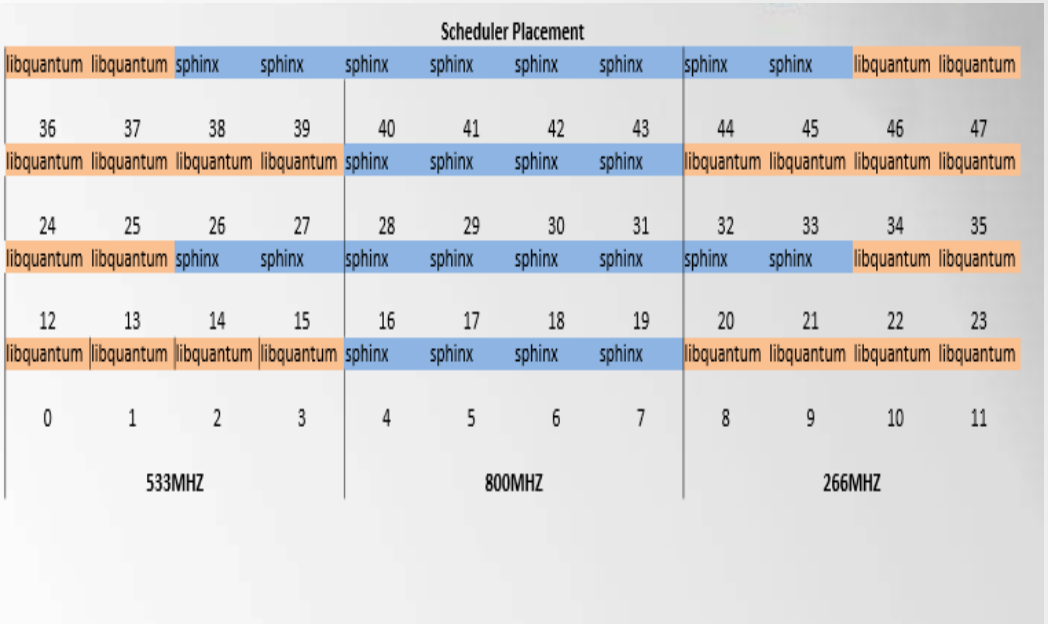
Evaluating Scheduling Policy

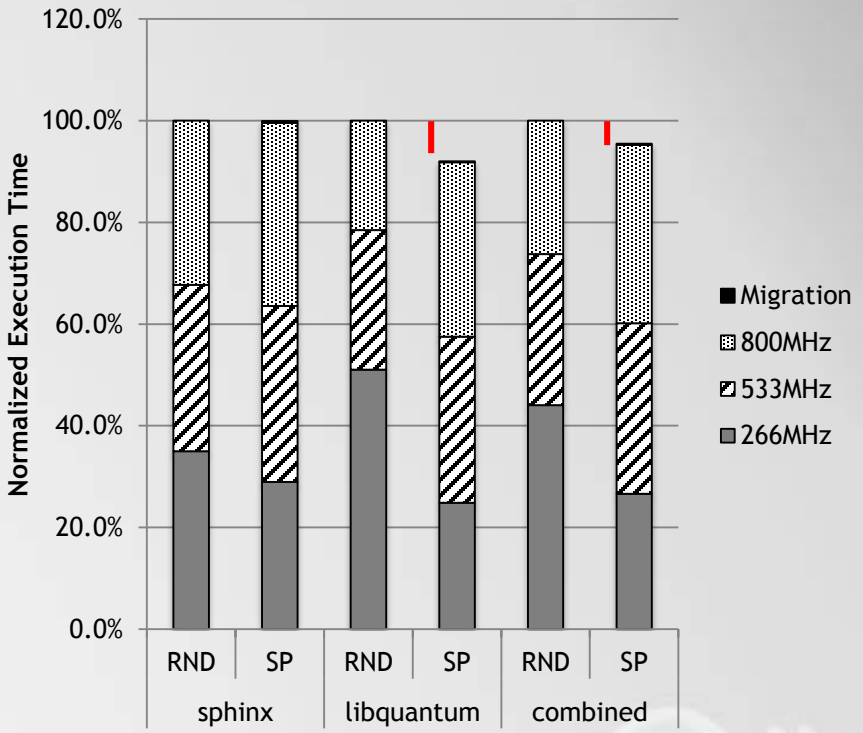
Scenario 3: 1 Compute-bound and 2 Memory-bound applications



Evaluating Scheduling Policy

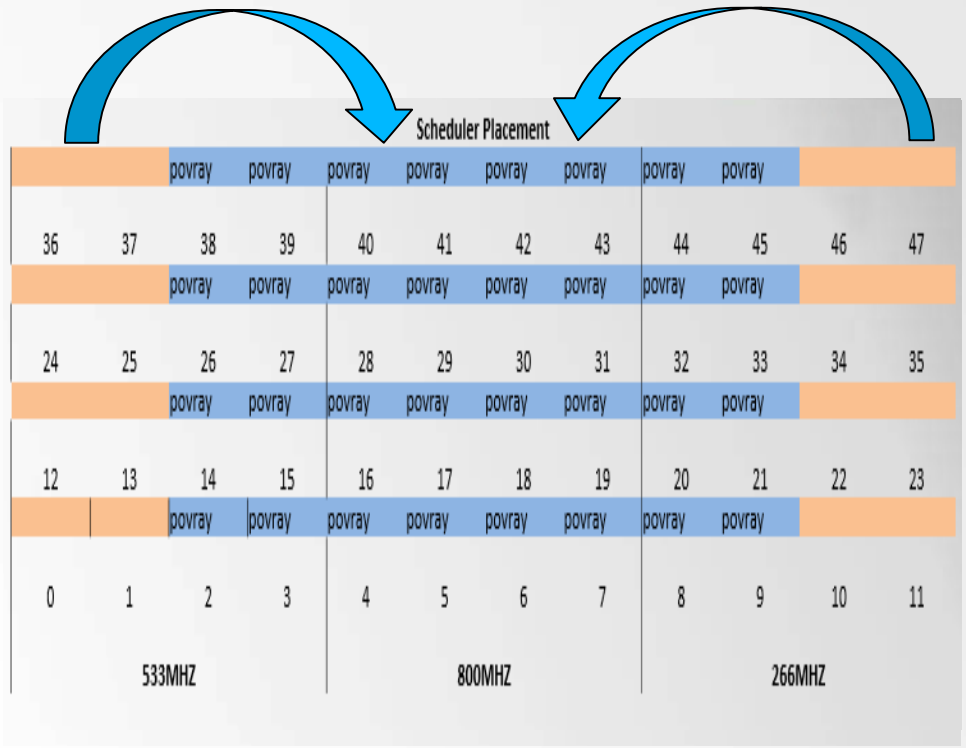
Scenario 4: 2 Memory-bound applications

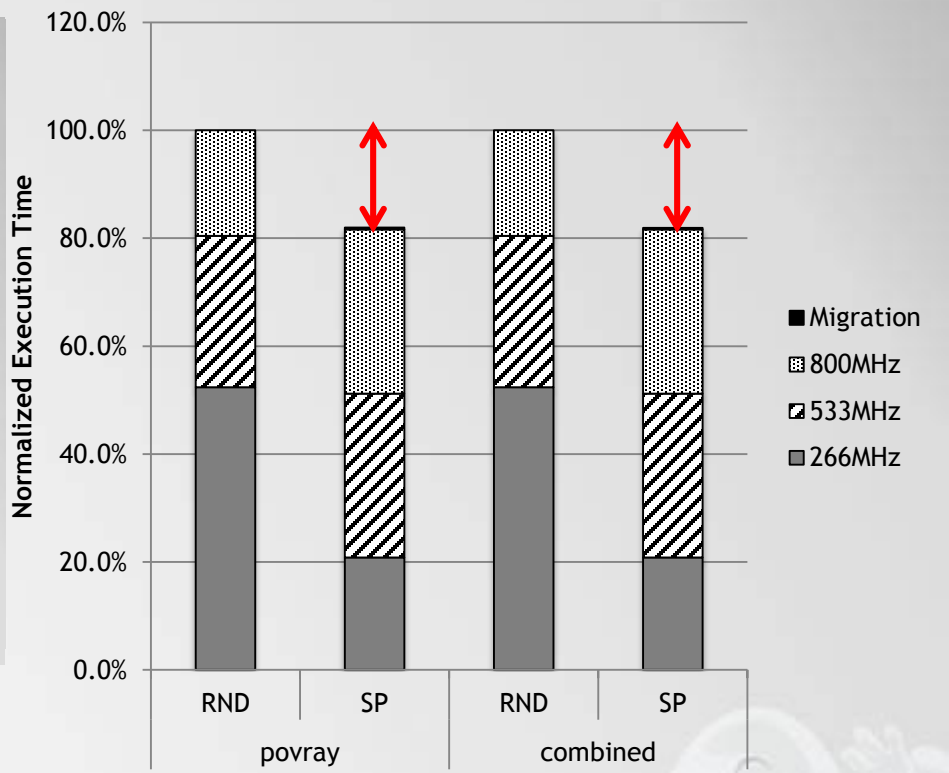




Evaluating Scheduling Policy

Scenario 5: 1 Compute-bound application





Outline

- Motivation
- Scheduling Policy
- Experimental Results
- Conclusions

Conclusions And Future Work

- We proposed an online scheduling policy which addresses application demands and characteristics
- Implementation on a real many-core architecture using real workloads
- Performance Improvement
  - Compute-bound up to 36%
  - Memory-bound up to 15%

Thank You!

CASPER Group
University of Cyprus
Computer Architecture, Systems and Performance Evaluation Research
Visit us:
www.cs.ucy.ac.cy/carch/casper