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

Panayiotis Petrides(*) Pedro Trancoso (*)(**)

(*) Computer Science (**) Computer Science and Department Engineering Chalmers University of Cyprus University of Technology CASPER: Computer Architecture System Performance Evaluation Research

Outline

Motivation

- Scheduling Policy
- Experimental Results
- Conclusions

2

Motivation





Many-core Array CMP with -10s -100s low power cores



Dual Core

CMP with 48 low power cores

Intel SCC

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





- povrav

- sphinx

-ht

800 MHz

- libguantum

5

Outline

- Motivation
- Scheduling Policy
- Experimental Results
- Conclusions

6

Scheduling Policy - Characterizing Applications

Determine how the distance and core frequency factors influence applications execution

$$D(H) = a \times H$$

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

$$F(f) = b \times f,$$

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

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



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

Many-Core Architecture



Cores Monitoring Mechanism

Scheduling Policy: Implementation



a34

a1

a10

Cluster 1

a5

H1





Step n

a3

Cluster 3

a13

a17

Cluster 2

a11

Cluster 4

a15

a7

Outline

- Motivation
- Scheduling Policy
- Experimental Results
- Conclusions

12

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%



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	povrav	povray	povray	povray	povray	povray	povray	sphinx	sphinx	
				· ·		· (· · ·				
12	13	14	15	16	17	18	19	20	21	22	23	
cohiov	sphiny	cohiov	sphipy	nouray	nouray	nouray	nourau	cohiny	cohiny	sphiny	sphiny	
spinity	ppinitx	ppinitx	ppinin	povidy	povidy	povidy	povidy	sprink	зріппл	sprink	эрппл	
0	1	2	3	4	5	6	7	8	9	10	11	
	533MHZ				800MHZ				266MHZ			



Migration
800MHz
533MHz
266MHz

Scenario 2: Compute-bound and Memory-bound applications

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



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

Scheduler Placement											
ibquantum	sphinx	sphinx	povray	Povray	povray	povray	sphinx	sphinx	libquantum	libquantum	
37	38	39	40	41	42	43	44	45	46	47	
libquantum	sphinx	sphinx	povray	povray	povray	povray	sphinx	sphinx	libquantum	libquantum	
25	26	27	28	29	30	31	32	33	34	35	
ibquantum	sphinx	sphinx	povray	povray	povray	povray	sphinx	sphinx	libquantum	libquantum	
13	14	15	16	17	18	19	20	21	22	23	
ibquantum	sphinx	sphinx	povray	povray	povray	povray	sphinx	sphinx	libquantum	libquantum	
1	2	3	4	5	6	7	8	9	10	11	
533N	IHZ		800MHZ				266MHZ				
li	ibquantum 37 ibquantum 25 ibquantum 13 ibquantum 1 533M	ibquantum sphinx 37 38 ibquantum sphinx 25 26 ibquantum sphinx 13 14 ibquantum sphinx 1 2 533MHZ	ibquantum sphinx sphinx 37 38 39 ibquantum sphinx sphinx 25 26 27 ibquantum sphinx sphinx 13 14 15 ibquantum sphinx sphinx 1 2 3 533MHZ	ibquantum sphinx sphinx povray 37 38 39 40 ibquantum sphinx sphinx povray 25 26 27 28 ibquantum sphinx sphinx povray 13 14 15 16 ibquantum sphinx sphinx povray 1 2 3 4 533MHZ	ibquantum sphinx sphinx povray Povray 37 38 39 40 41 ibquantum sphinx sphinx povray povray 25 26 27 28 29 ibquantum sphinx sphinx povray povray 13 14 15 16 17 ibquantum sphinx sphinx povray povray 13 14 15 16 17 ibquantum sphinx sphinx povray povray 1 2 3 4 5 533MHZ 800	ibquantumsphinxsphinxpovrayPovraypovray373839404142ibquantumsphinxsphinxpovraypovraypovray252627282930ibquantumsphinxsphinxpovraypovraypovray131415161718ibquantumsphinxsphinxpovraypovraypovray123456533MHZ533MHZ555	ibquantumsphinxsphinxpovrayPovraypovraypovray37383940414243ibquantumsphinxsphinxpovraypovraypovray25262728293031ibquantumsphinxsphinxpovraypovraypovray13141516171819ibquantumsphinxsphinxpovraypovraypovraypovray13141516171819ibquantumsphinxsphinxpovraypovraypovraypovray1234567533MHZ800MHZ800MHZ	ibquantum ibquantumsphinxpovraypovraypovraypovraypovraypovraysphinx3738394041424344ibquantum ibquantumsphinxsphinxpovraypovraypovraypovraysphinx2526272829303132ibquantum ibquantumsphinxsphinxpovraypovraypovraypovraypovray1314151617181920ibquantum ibquantumsphinxsphinxpovraypovraypovraypovraysphinx12345678533MHZ800MHZsphinxsphinxsphinx	ibquantum ibquantumsphinxspovraypovraypovraypovraysphinxsphinxsphinx373839404142434445ibquantum ibquantumsphinxsphinxpovraypovraypovraypovraysphinxsphinx252627282930313233ibquantum ibquantumsphinxsphinxpovraypovraypovraypovray131415161718192021ibquantum ibquantumsphinxsphinxpovraypovraypovraypovraysphinx123456789533MHZ800MHZ266I	ibquantum ibquantumsphinxsphinxpovrayPovraypovraypovraypovraysphinxsphinxlibquantum37383940414243444546ibquantum ibquantumsphinxsphinxsphinxpovraypovraypovraypovraysphinxsphinxsphinxlibquantum25262728293031323334ibquantum ibquantumsphinxsphinxsphinxpovraypovraypovraypovraysphinxsphinxlibquantum13141516171819202122ibquantumsphinxsphinxpovraypovraypovraypovraysphinxsphinxlibquantum12345678910533MHZ800MHZ266MHZ266MHZ	





Scenario 4: 2 Memory-bound applications





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