Selection of the Register File Size and the Resource Allocation Policy on SMT Processors

J. Alastruey[1], T. Monreal[1], F. J. Cazorla[2], V. Viñals[1], M. Valero[2,3]

[1]gaZ – I3A - Universidad de Zaragoza

[2]BSC, Barcelona Supercomputing Center

[3]DAC, Universitat Politècnica de Catalunya

HiPEAC - High-Performance Embedded Architecture and Compilation

International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)
Campo Grande, Brasil, 2008

Introduction

- Context: SMT processors
  - Private resources: PC, Map Table,
  - Shared resources: Register File, IQ, …

managed through resource allocation policies

Introduction

- Context: SMT processors
  - Private resources: PC, Map Table,
  - Shared resources: Register File, IQ, …

managed through resource allocation policies

- Physical Register File (PRF) shared among all threads

Supplies operands

Stores commited and speculative states

PRF

⇑ ports

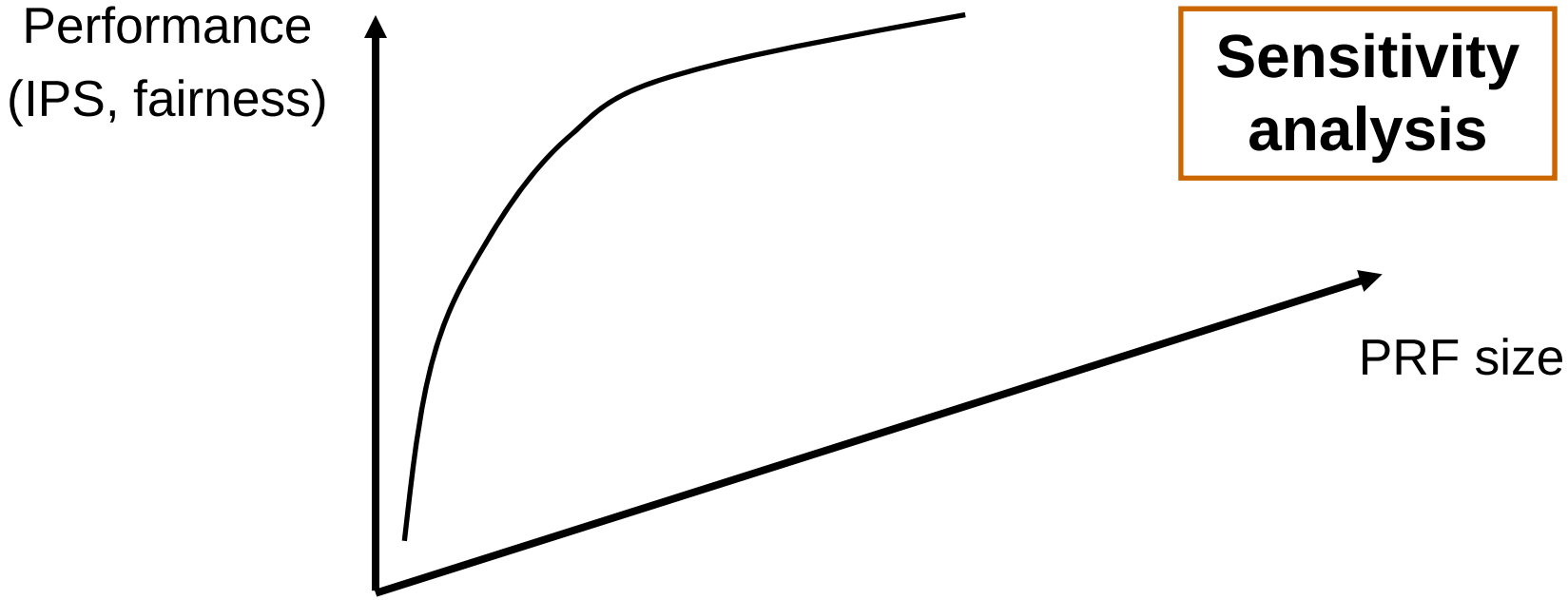
⇑ physical registers

⇑ energy ⇑ area ⇑ access time

Motivation

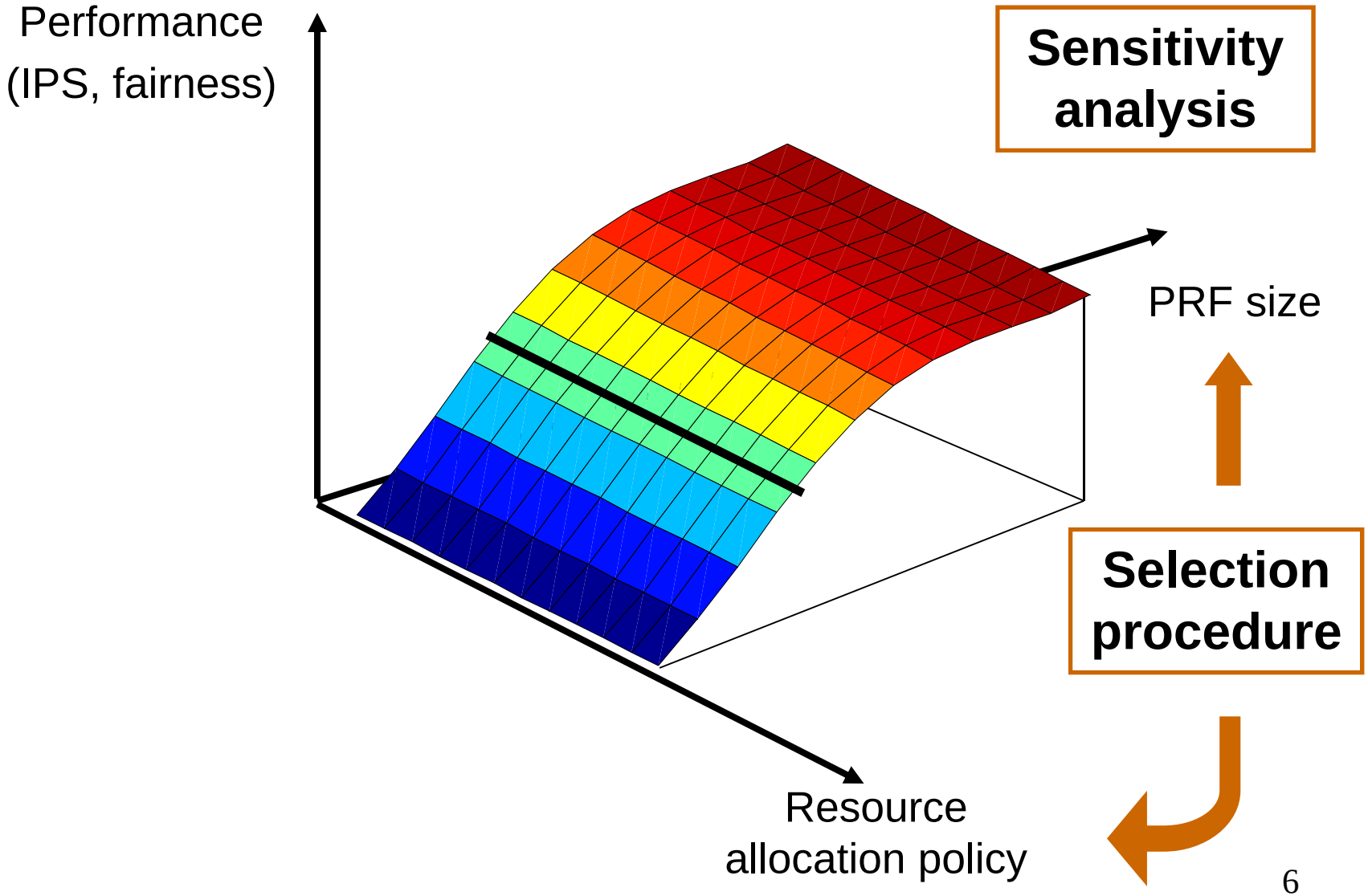
- Two critical design issues in SMT
  - PRF sizing
    - Large PRFs
      - ⇓ rename stalls → ⇑ IPC
    - Small PRF
      - ⇑ frequency → ⇑ IPS
  - Resource allocation policy
    - Multiple choices

Contributions



Sensitivity analysis

Contributions



Outline

- **Resource Allocation Policies (RAP)**
- Experimentation
- Performance sensitivity to PRF size and RAP
- PRF size and RAP selection procedure
- Conclussions

Resource allocation policies

- Distribution of SMT shared resources driven by
  - Fetch policy
    - Which thread fetch instructions
  - Resource usage constraining policy
    - Trigger events: L2 misses, threshold crossing …
    - Constraining actions: fetch-stall, flush.

- Many approaches !!

- Classification

Resource allocation policies

Fetch bandwith distribution among threads

fetch policies

resource usage constr. policies

- event
- action(s)

resource allocation policies

Resource allocation policies

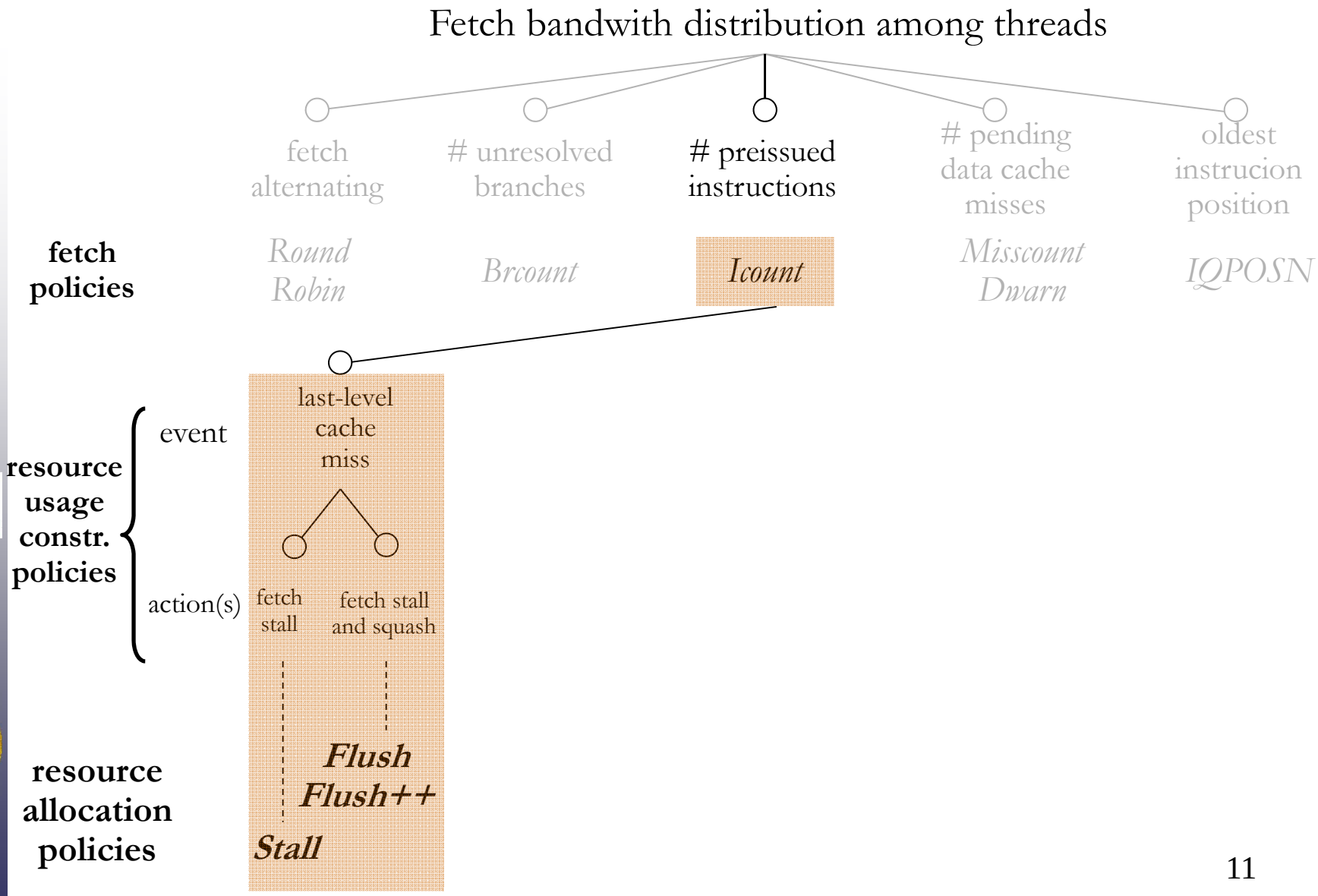
Fetch bandwith distribution among threads

	fetch alternating	# unresolved branches	# preissued instructions	# pending data cache misses	oldest instrucion position
fetch policies	*Round Robin*	*Brcount*	*Icount*	*Misscount Dwarn*	*IQPOSN*

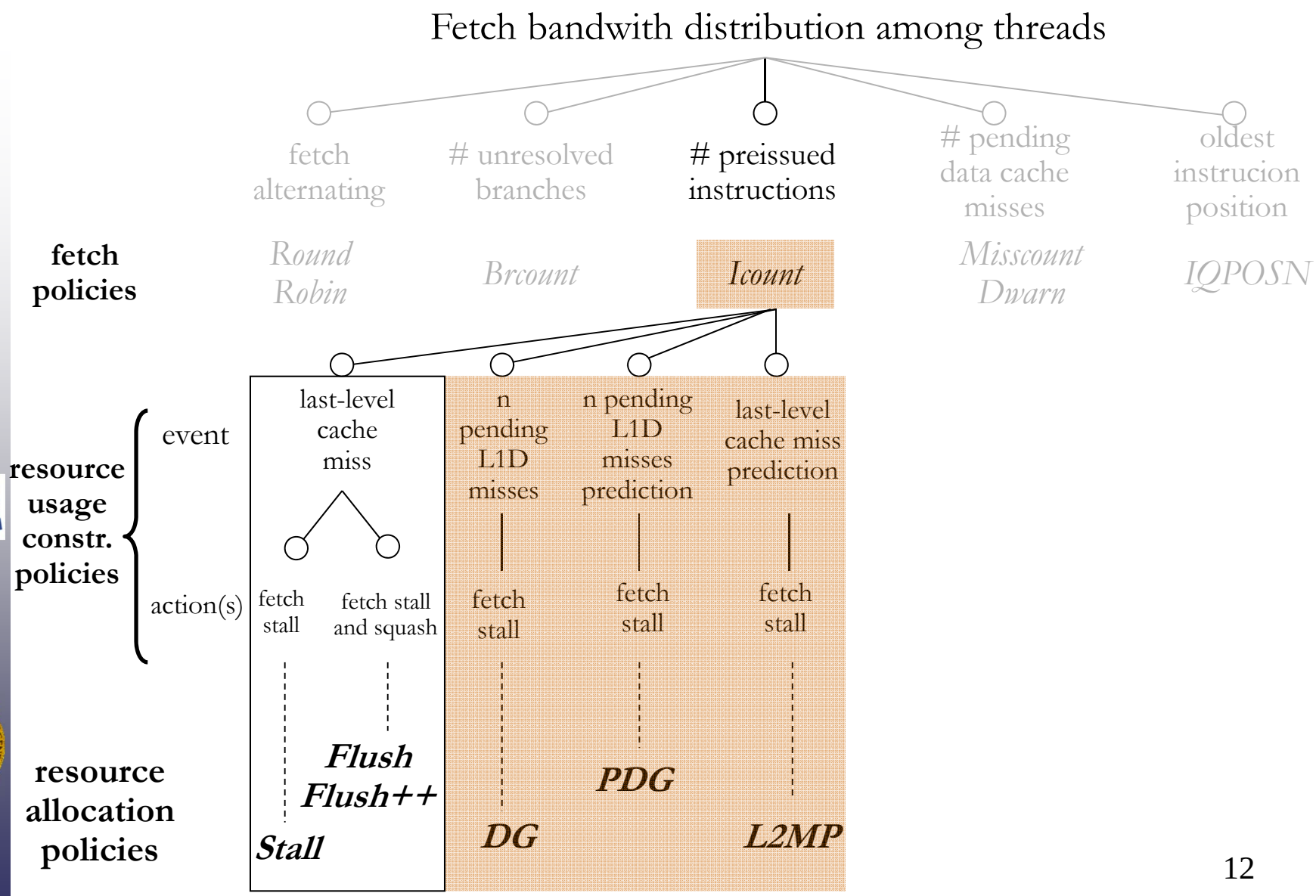
resource usage constr. policies
- event
- action(s)

resource allocation policies

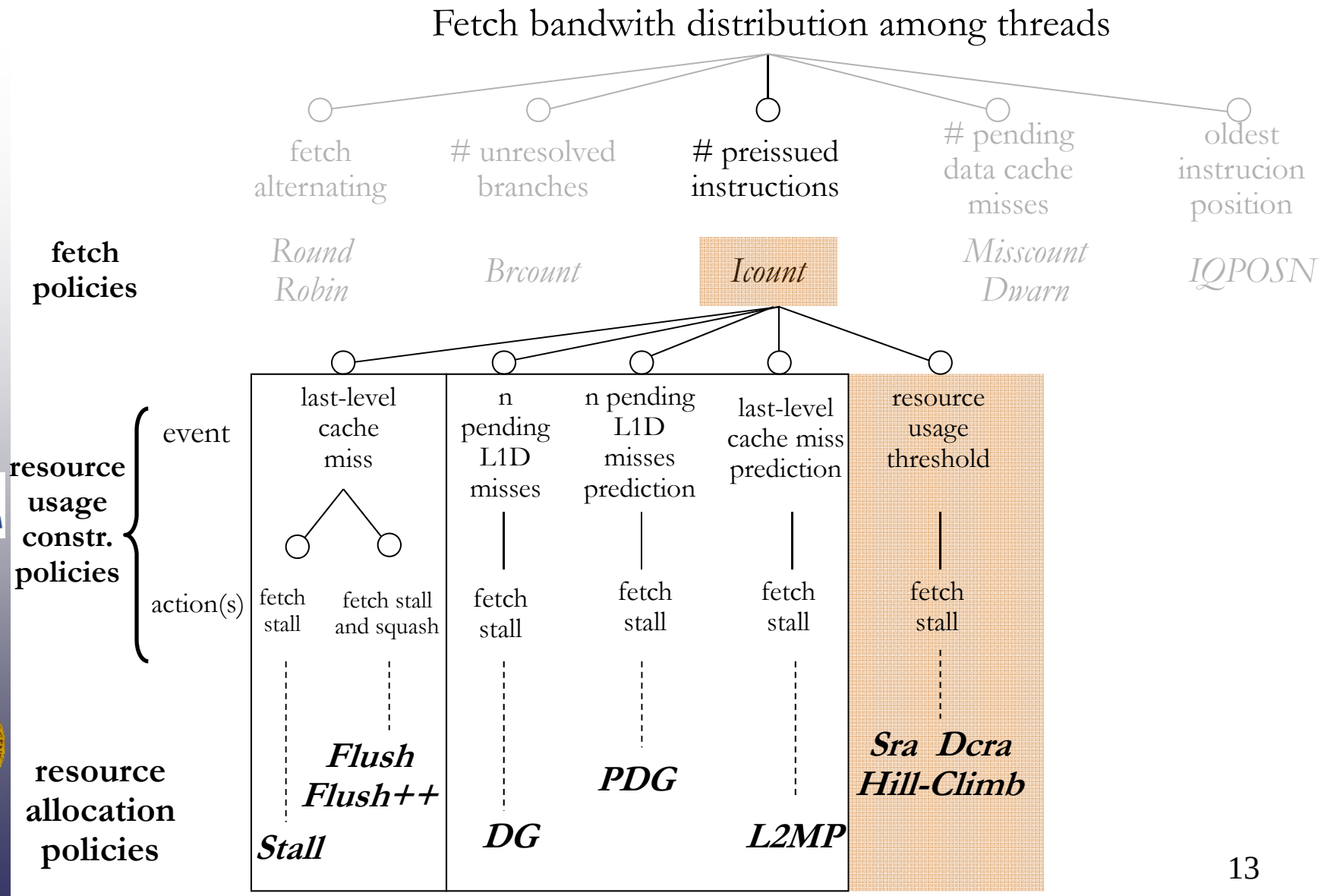
Resource allocation policies



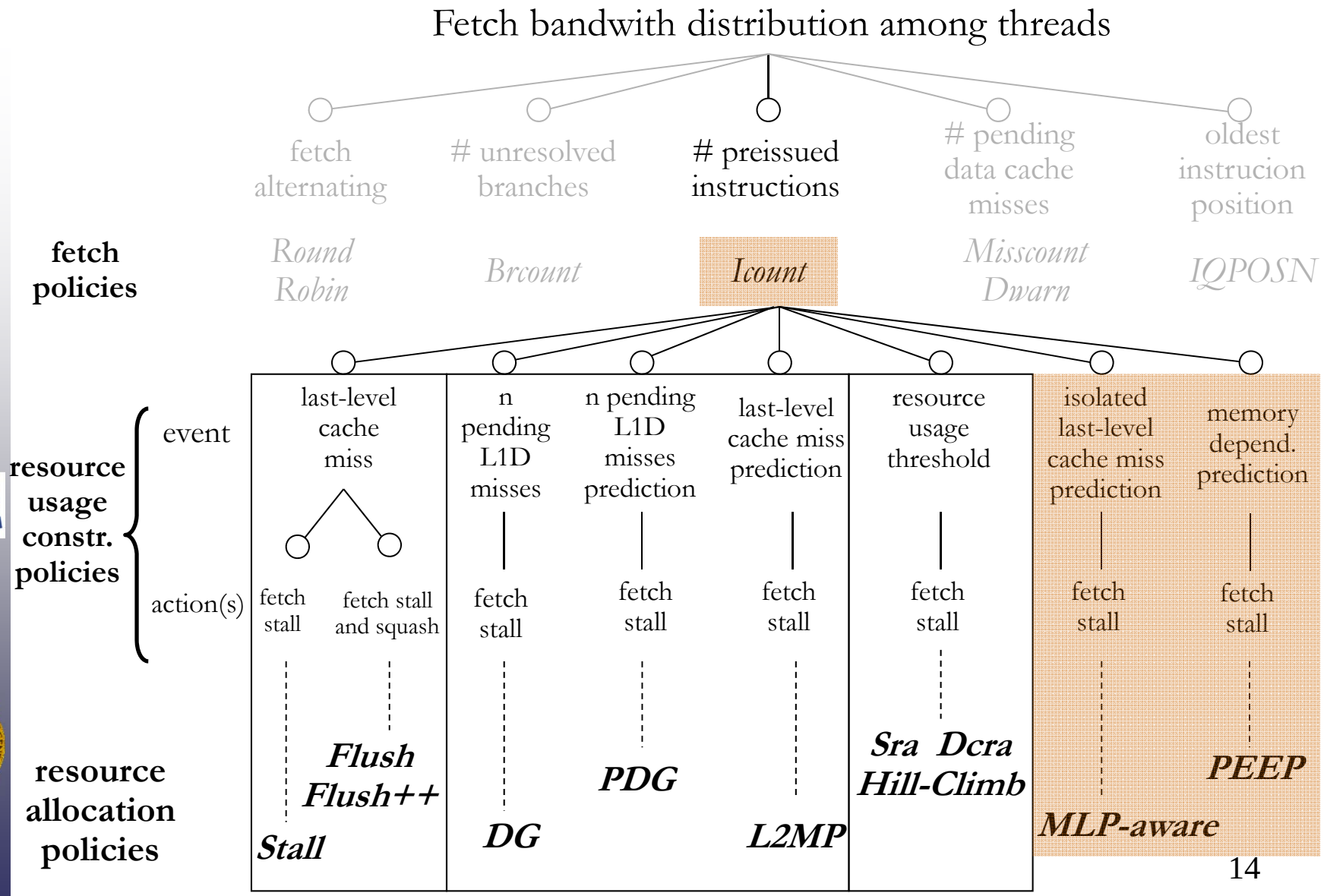
Resource allocation policies



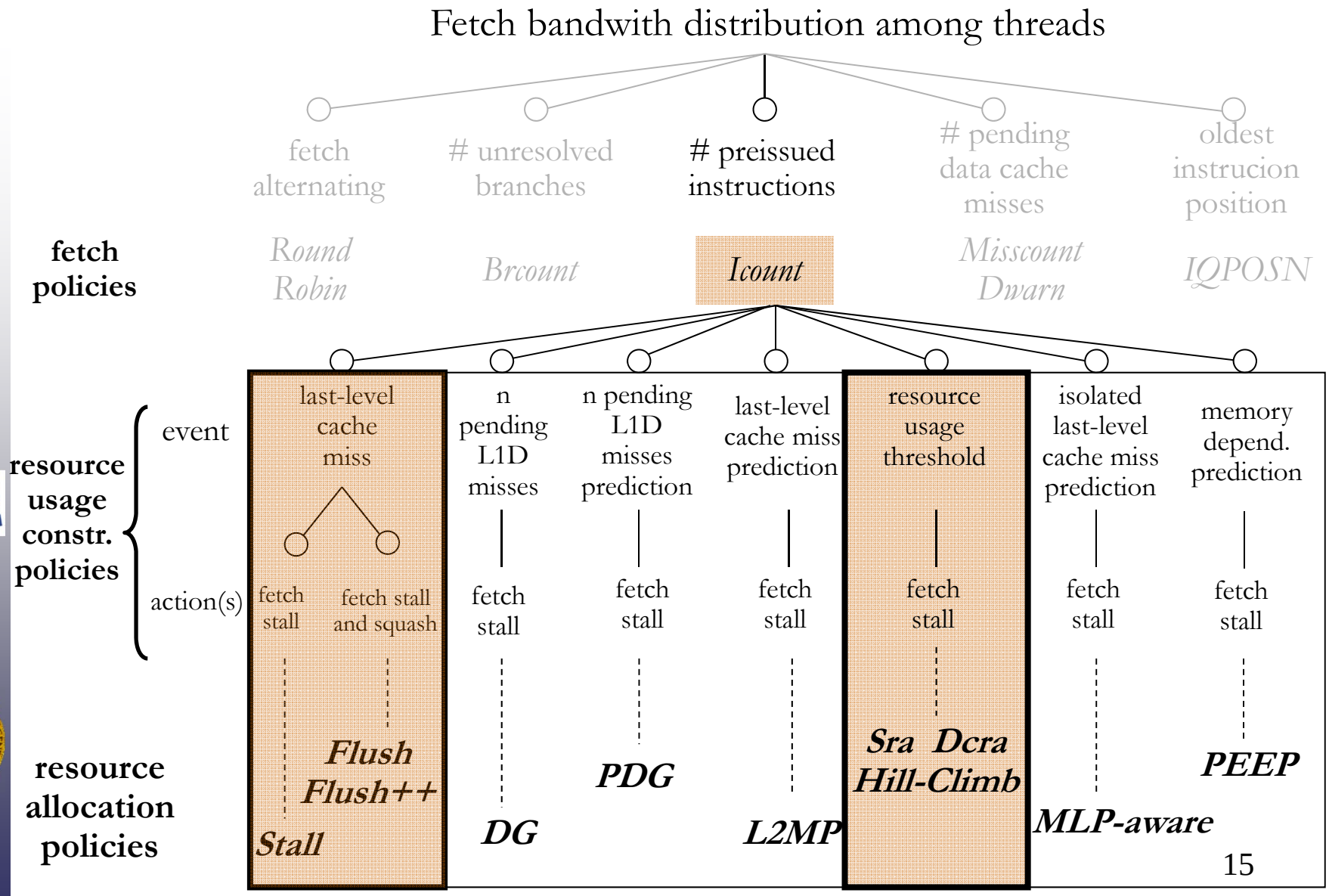
Resource allocation policies



Resource allocation policies



Resource allocation policies

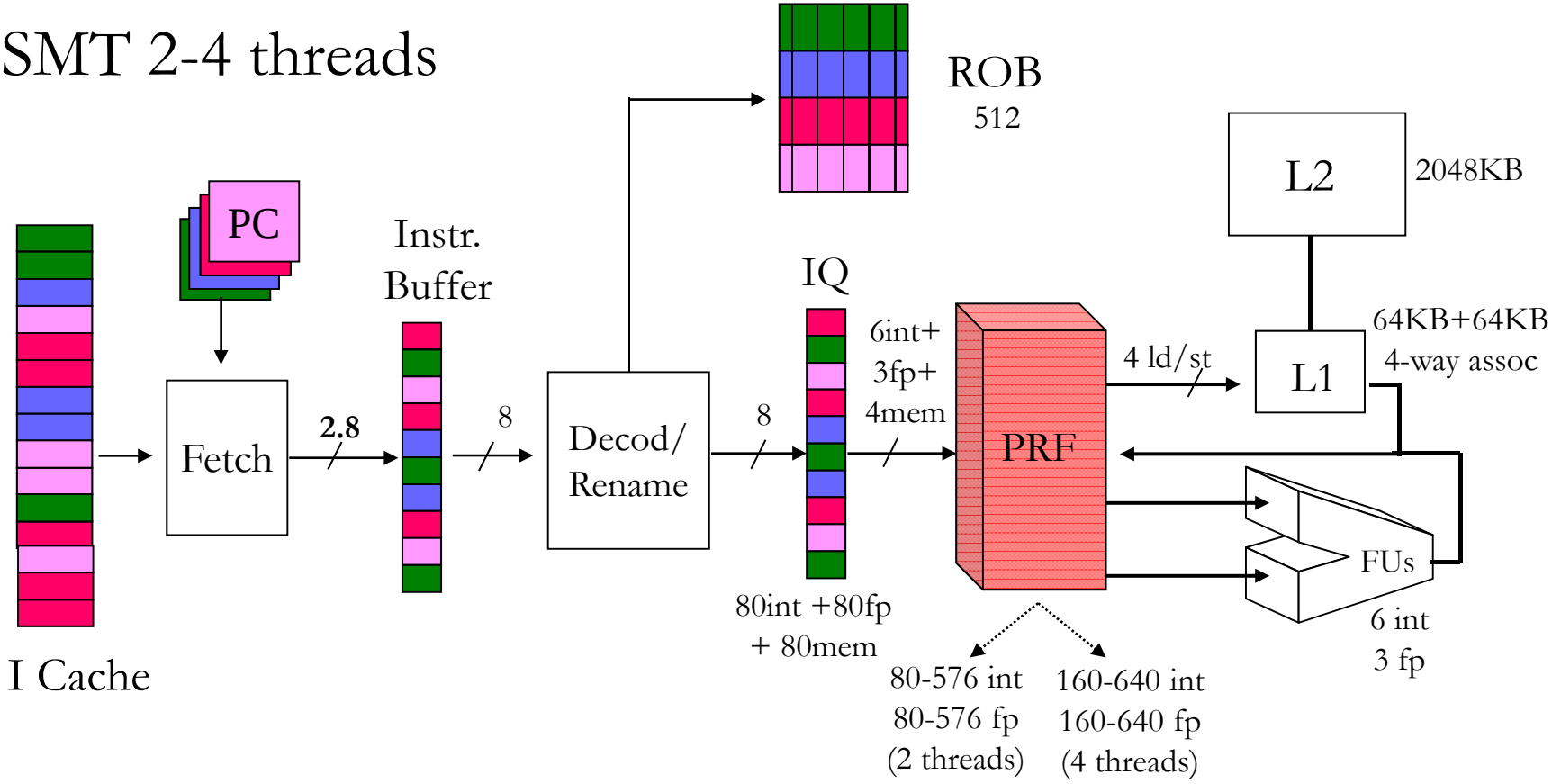


Outline

- Resource allocation policies (RAP)
- **Experimentation**
- Performance sensitivity to PRF size and RAP
- PRF size and RAP selection procedure
- Conclussions

Processor model

- SMT 2-4 threads



Except PRF, same configuration for 2 and 4 threads

smtsim[1] based simulator

[1] D.Tullsen, S. Eggers, H. Levy. "Simultaneous multithreading: Maximizing on-chip parallelism". *ISCA* 1995.

Workload

- SPEC2000
  - 12 int + 13 fp (all but fma3d)
  - Representative parts, 300M
- Workload
  - 2-SMT: 24 pairs, 4-SMT: 12 quartets
  - Balanced composition of int-fp and high-low ilp benchmarks
- End of simulation: *last*

	high-ILP	low-ILP	mix
int	bzip2-eon, gzip-gcc	vpr-mcf, vortex-twolf	perlbmk-vortex, gcc-gap
int-fp	perlbmk-apsi, crafty-galgel	gap-swim, parser-mgrid	crafty-art, gzip-mgrid
	bzip2-mesa, eon-sixtrack	vpr-lucas, mcf-equake	twolf-galgel, parser-ammp
fp	mesa-sixtrack, ammp-wupwise	lucas-equake, applu-art	apsi-applu, facerec-swim

Metrics

- Raw performance

$$IPS_{total} = \frac{IPC_{total}}{Tcycle}$$

- Fairness[1]

$$H_{mean-wIPC} = \frac{N}{\sum_{i=1}^{N} (rIPC_i)^{-1}}$$

$$IPC_{total} = \sum_{i=1}^{N} IPC_{SMT_i}$$

N = #threads

$$rIPC_i = \frac{IPC_{SMT_i}}{IPC_{ST_i}}$$

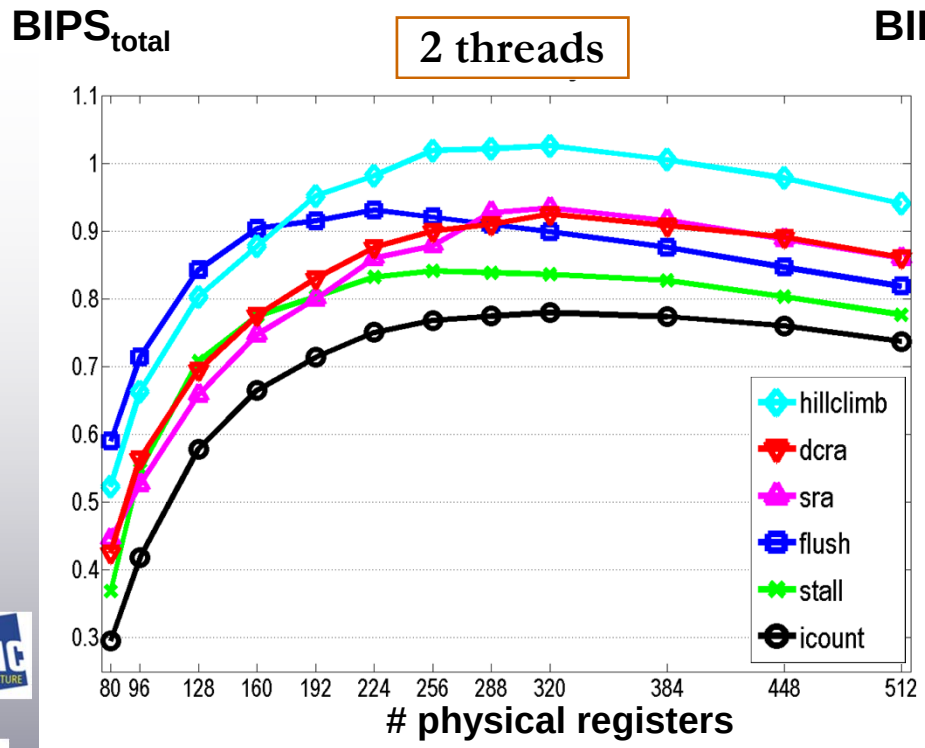
ST: single-thread mode

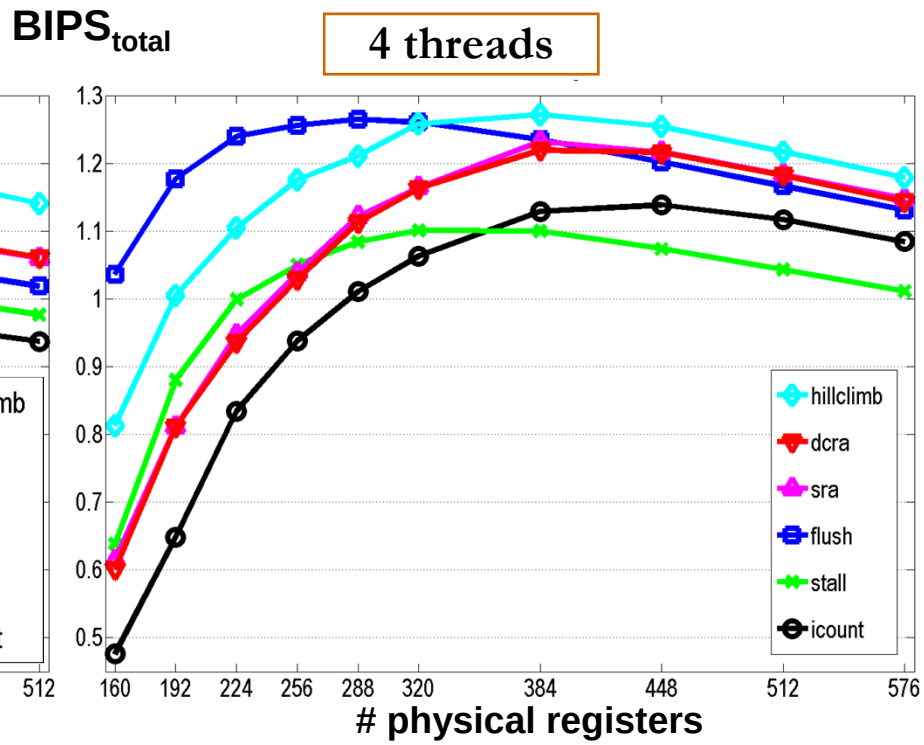
[1]K. Luo, J. Gummaraju and M. Franklin. "Balancing throughput and fairness in SMT processors", *ISPASS* 2001.

Outline

- Resource allocation policies (RAP)
- Experimentation
- **Performance sensitivity to PRF size and RAP**
- PRF size and RAP selection procedure
- Conclussions

Performance: IPS

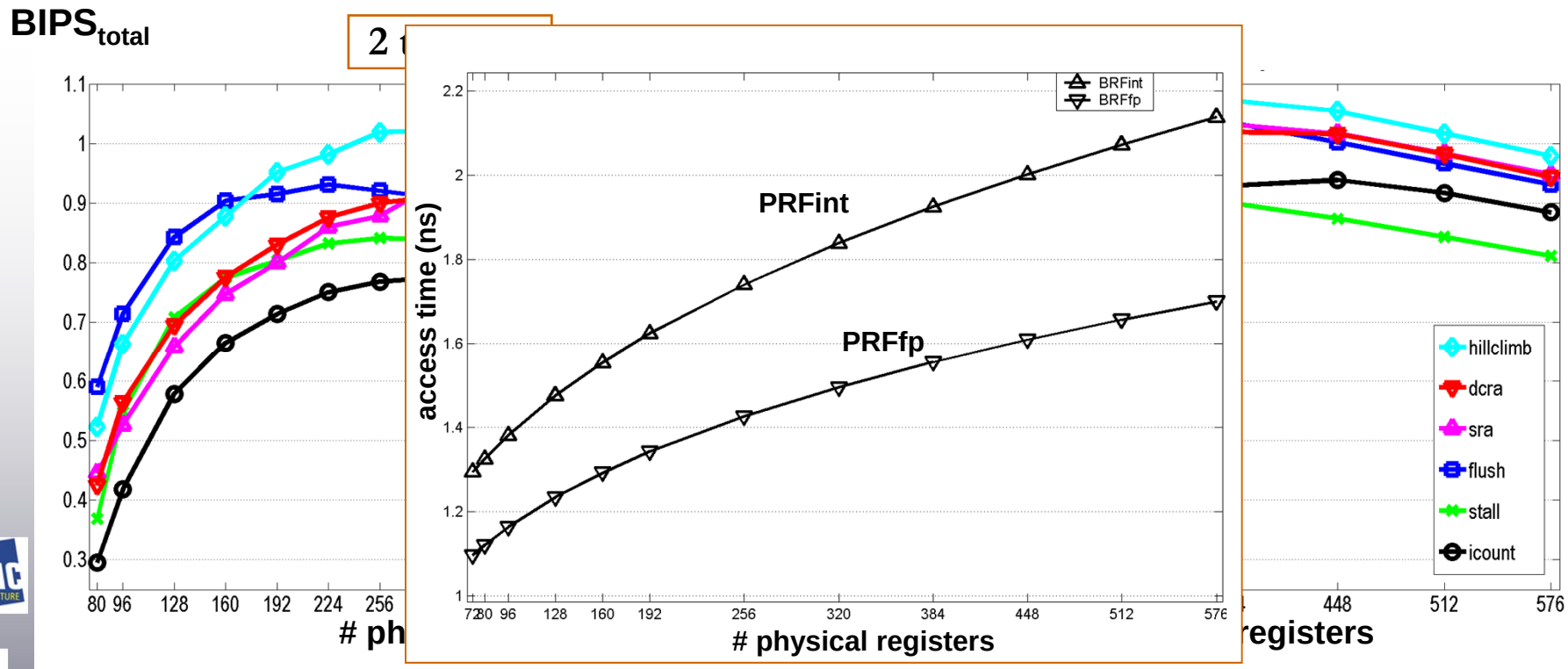




Tcycle = Taccess PRF

1. PRF size sensitivity
2. Resource allocation policy sensitivity
3. 2-4 threads comparison

Performance: IPS

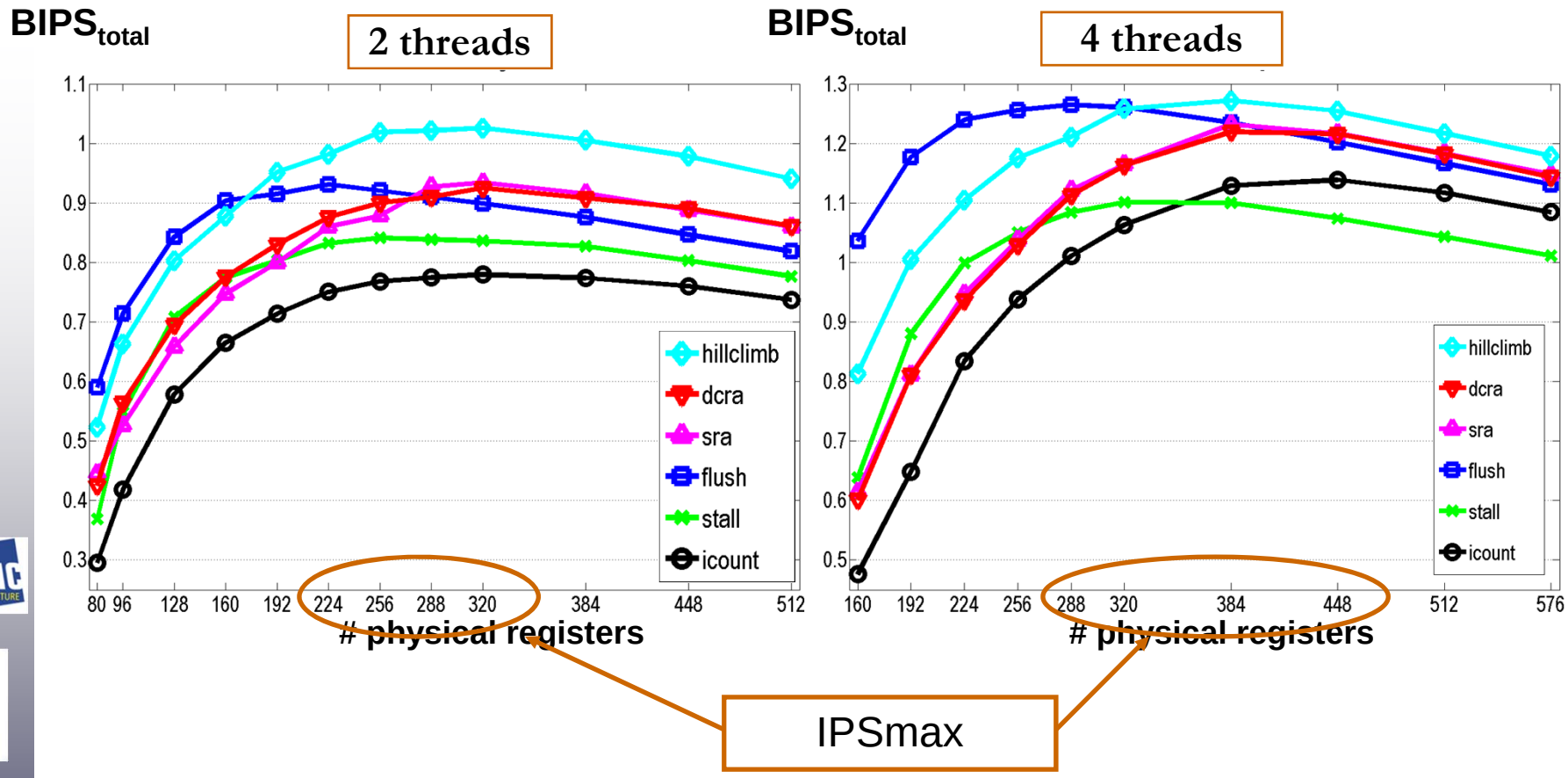


Tcycle = Taccess PRF[1]

1. PRF size sensitivity
2. Resource allocation policy sensitivity

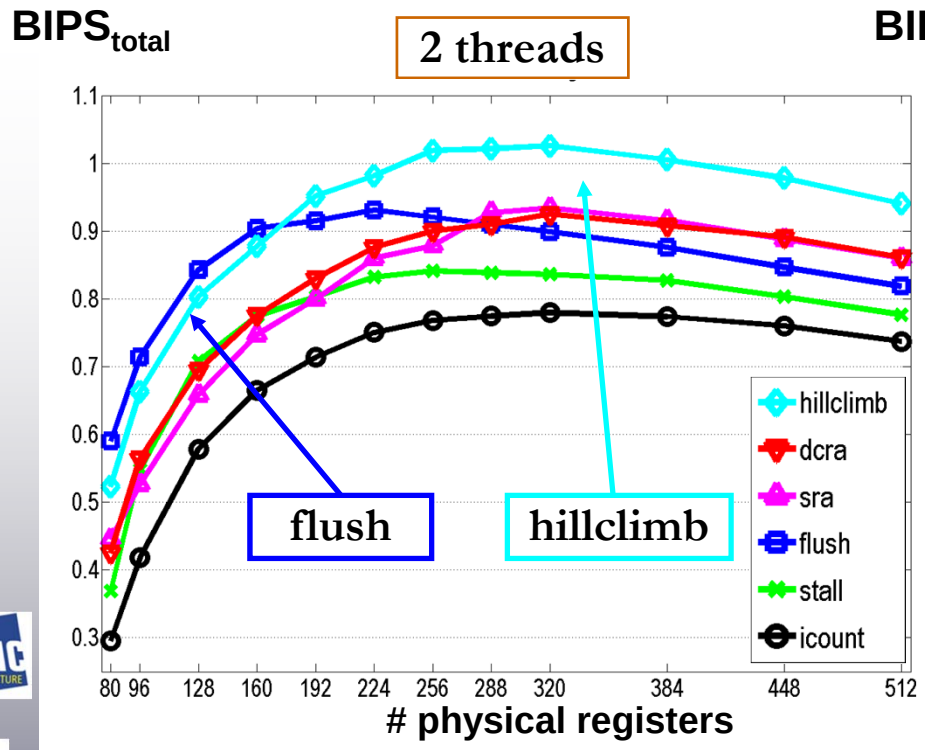
[1] S. Rixner, W. Dally, B. Khailany, P. Mattson, U. Kapasi and J.Owens, "Register Organization for Media Processing", *HPCA* 2000.

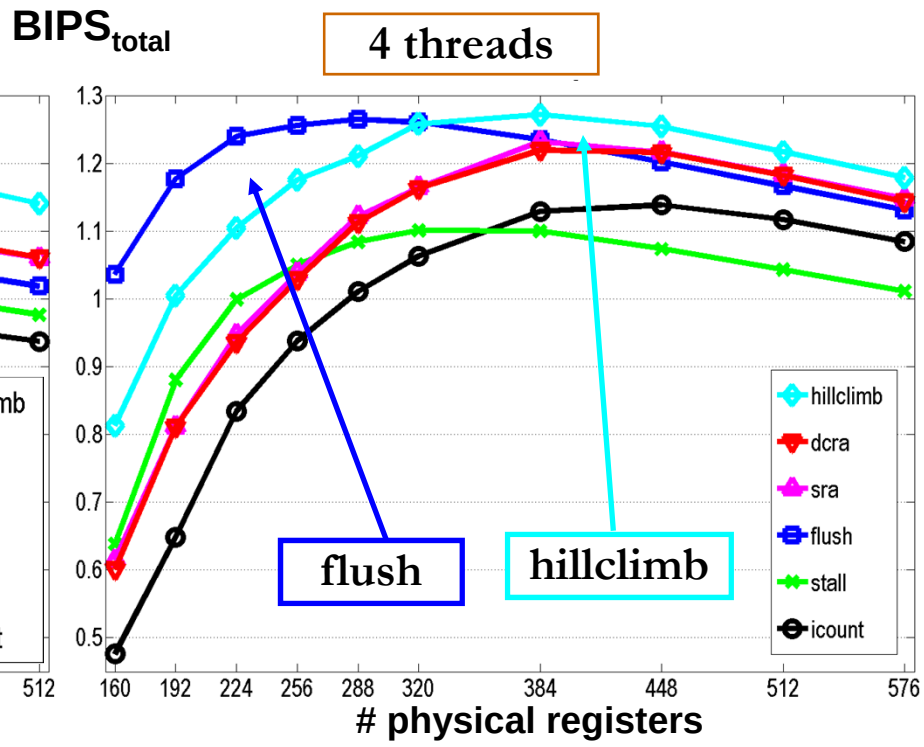
Performance: IPS



1. PRF size sensitivity
2. Resource allocation policy sensitivity
3. 2-4 threads comparison

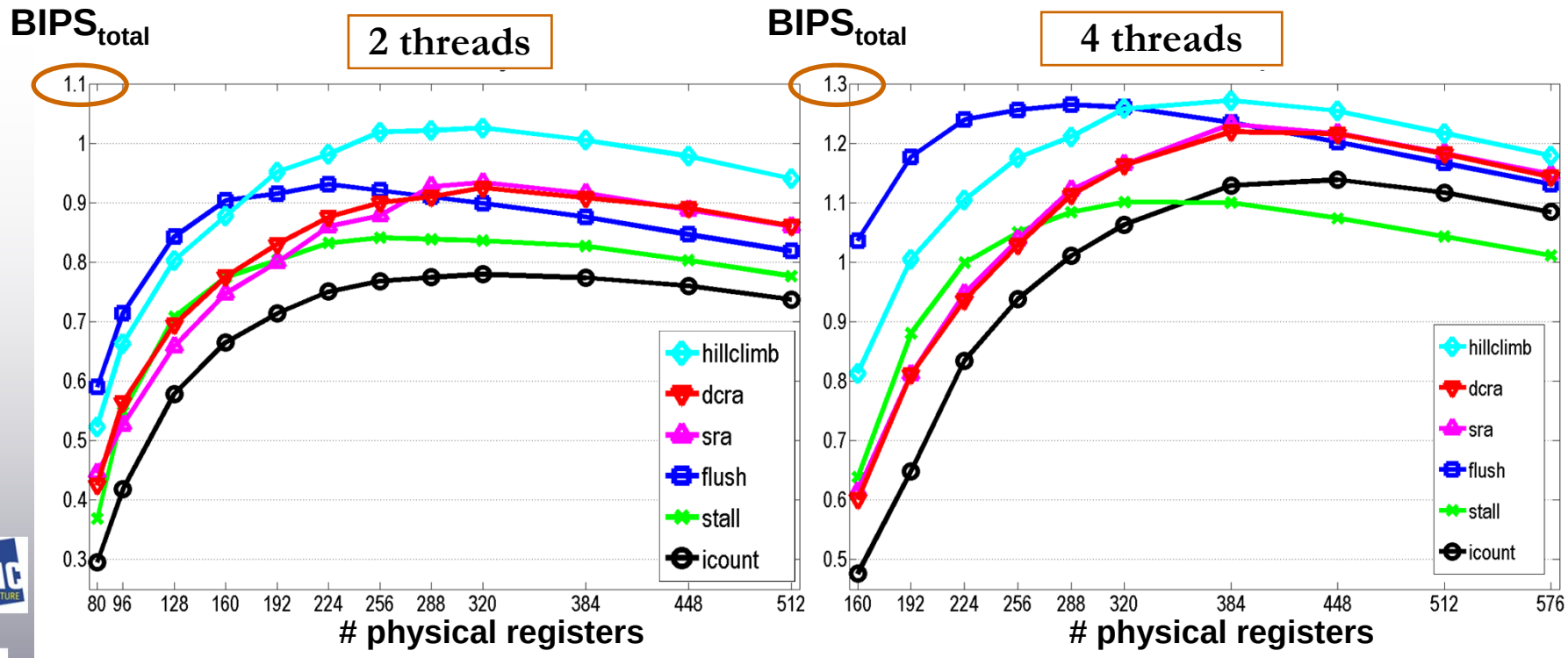
Performance: IPS





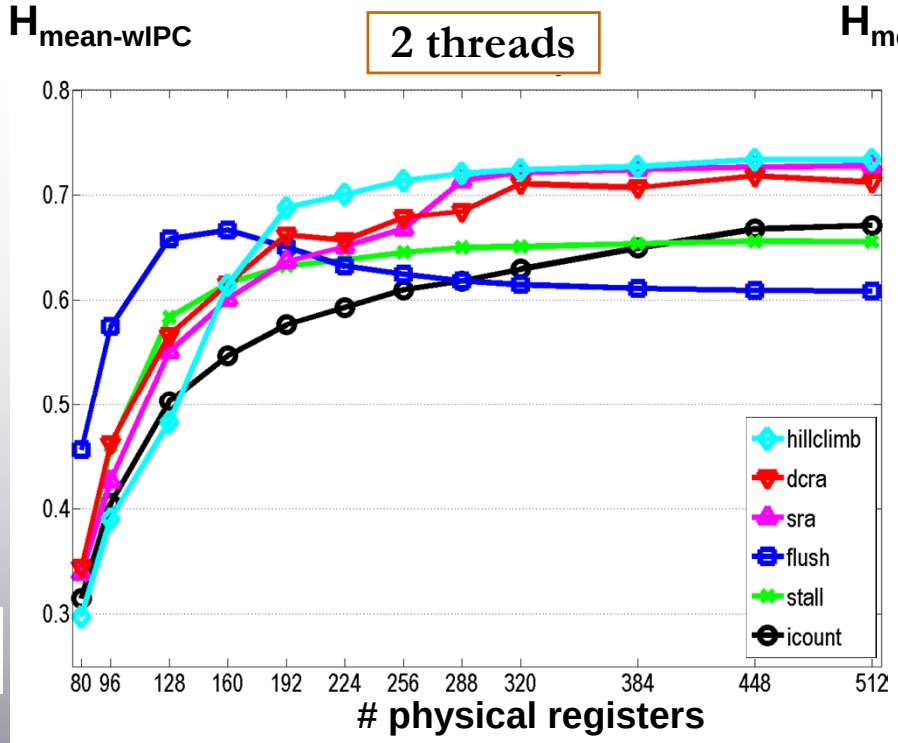
1. PRF size sensitivity
2. Resource allocation policy sensitivity
3. 2-4 threads comparison

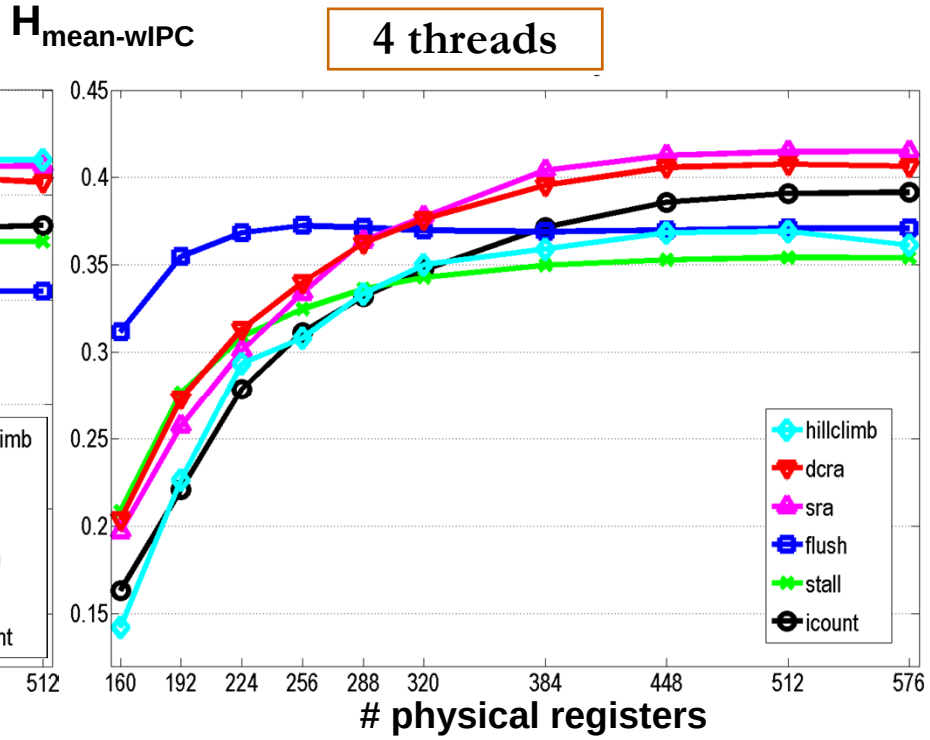
Performance: IPS



1. PRF size sensitivity
2. Resource allocation policy sensitivity
3. 2-4 threads comparison

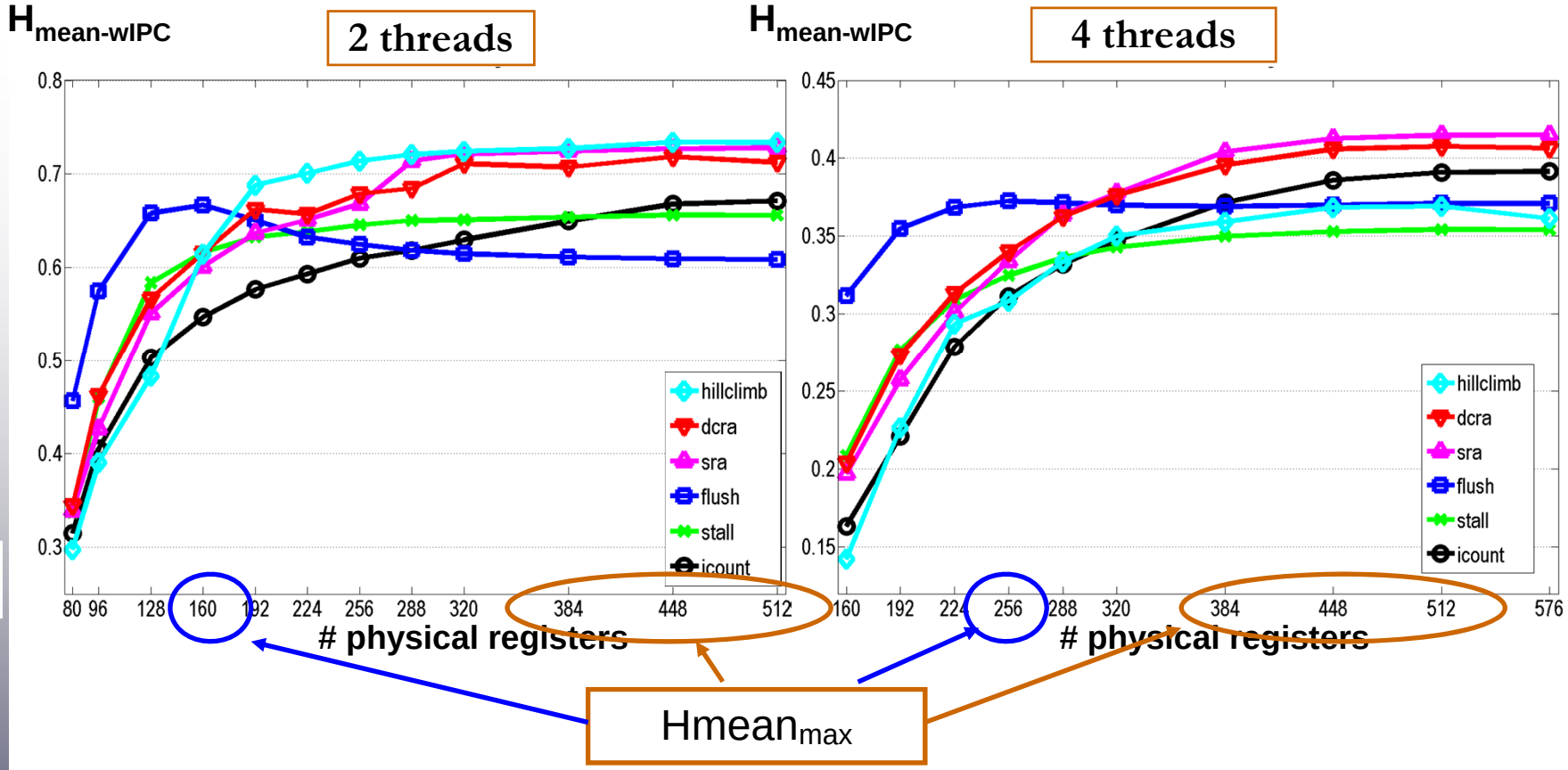
Fairness





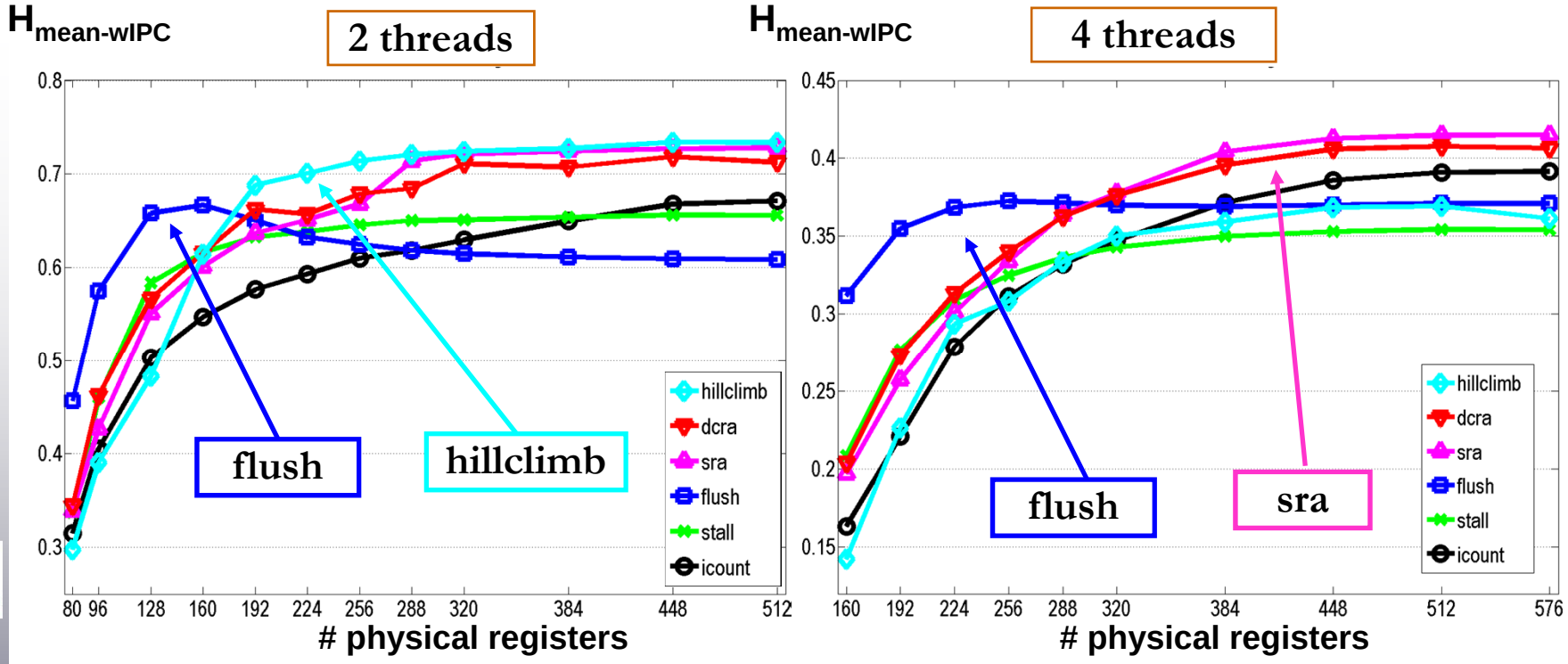
1. PRF size sensitivity
2. Resource allocation policy sensitivity
3. 2-4 threads comparison

Fairness



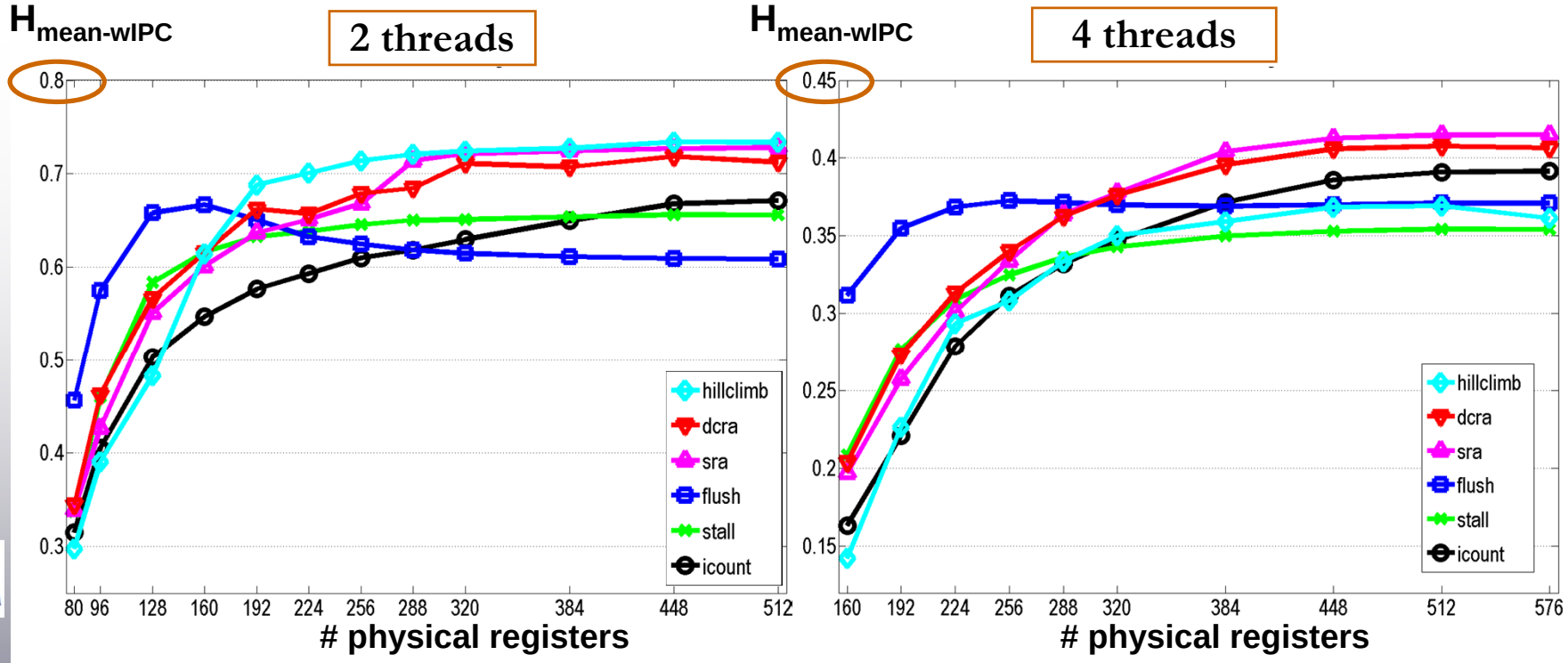
1. PRF size sensitivity
2. Resource allocation policy sensitivity
3. 2-4 threads comparison

Fairness



1. PRF size sensitivity
2. Resource allocation policy sensitivity
3. 2-4 threads comparison

Fairness



1. PRF size sensitivity
2. Resource allocation policy sensitivity
3. 2-4 threads comparison

Outline

- Resource allocation policies
- Experimentation
- Performance sensitivity to PRF size and RAP
- **PRF size and RAP selection procedure**
- Conclussions

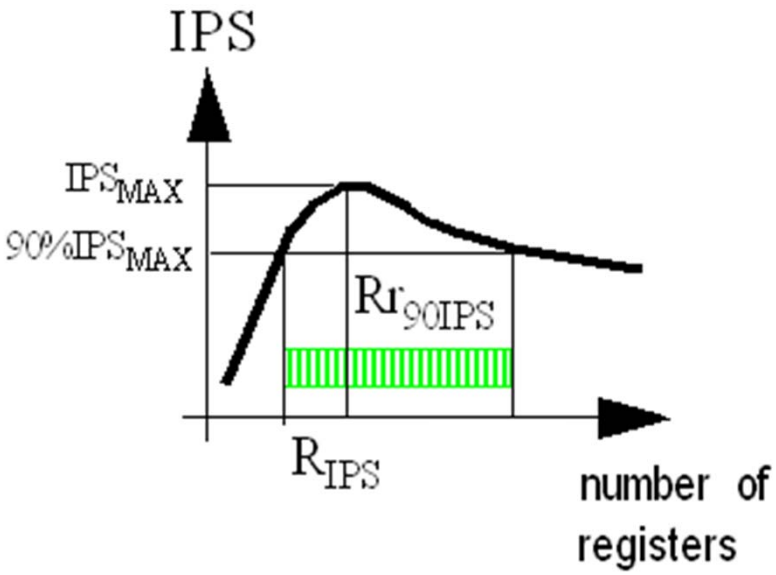
Selection of PRF size and resource allocation policy

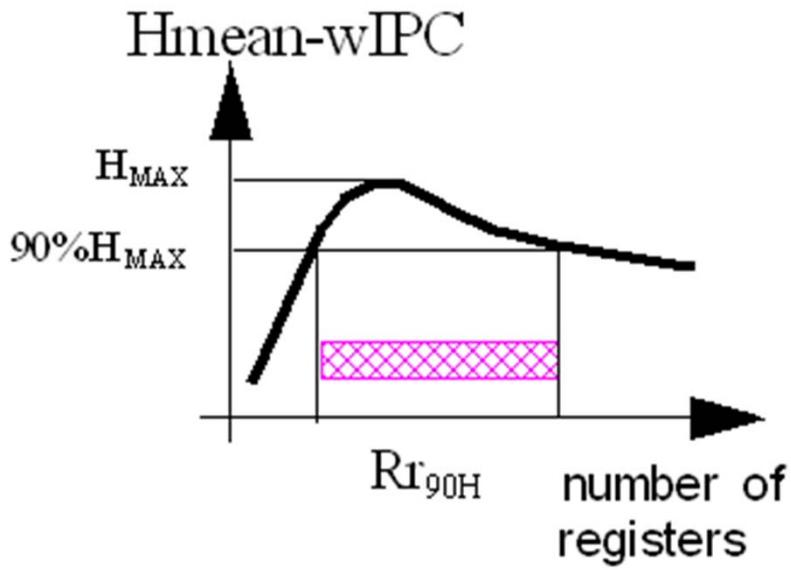
- Goal: select PRF size and resource allocation policy that maximizes performance
  - IPS vs. fairness tradeoff

- Procedure
  1. For each policy, selects the best performing PRF size
  2. Comparison of obtained design points

PRF size selection

- Goal
  - Maximize IPS and Fairness
- Observation (all RAPs)
  - Rr_{90IPS} and Rr_{90H} overlap
    - Several PRF sizes for each RAP
- Many design targets
  - Hardware cost
  - Energy consumption
  - Raw performance
    - closest size to R_{IPS}





Selection of PRF size and resource allocation policy

	2-threads			4-threads		
Policy	**R**			**R**		
Hill-climbing	320			384		
Dcra	320			384		
Sra	320			384		
Flush	224			288		
Stall	256			320		
Icount	320			448		

R_{IPS} for all resource allocation policies

Selection of PRF size and resource allocation policy

	2-threads			4-threads		
Policy	**R**	**BIPS**	**Hmean**	**R**	**BIPS**	**Hmean**
Hill-climbing	320			384		
Dcra	320			384		
Sra	320			384		
Flush	224			288		
Stall	256			320		
Icount	320			448		

Comparison among RAPs

Selection of PRF size and resource allocation policy

	2-threads			4-threads		
Tipo	**R**	**BIPS**	**Hmean**	**R**	**BIPS**	**Hmean**
Hill-climbing	**320**	**1.03**	**0.72**	384	1.27	0.36
Dcra	320	0.92	0.71	384	1.22	0.40
Sra	320	0.93	0.72	**384**	1.23	**0.40**
Flush	224	0.93	0.63	**288**	**1.27**	0.37
Stall	256	0.84	0.65	320	1.10	0.34
Icount	320	0.78	0.63	448	1.14	0.39

2 threads: Hill-climbing with 320r

4 threads

- IPS: Flush-288r
- Hmean: Sra-384r

Outline

- Resource allocation policies
- Experimentation
- Performance sensitivity to PRF size and RAP
- PRF size and RAP selection procedure
- **Conclussions**

Conclussions

- Combined performance sensitivity to PRF size and resource allocation policy
  - Small PRFs: Flush
  - Large PRFs
    - IPS: Hill-climbing
    - Fairness: Sra or Dcra
- Selection of PRF size and resource allocation policy procedure based on two metrics (IPS, fairness)
  - 2 threads: Hill-climbing-320r
  - 4 threads
    - IPS: Flush-224r
    - Fairness: Sra-384r
- From 2 to 4 threads
  - Better IPS (23%) but worst fairness(-44%)

Selection of the Register File Size and the Resource Allocation Policy on SMT Processors

J. Alastruey[1], T. Monreal[1], F. Cazorla[2], V. Viñals[1], M. Valero[2,3]

[1]gaZ – I3A - Universidad de Zaragoza

[2]BSC, Barcelona Supercomputing Center

[3]DAC, Universitat Politècnica de Catalunya

HiPEAC - High-Performance Embedded Architecture and Compilation

International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)
Campo Grande, Brasil, 2008