# Using hardware counter data to model performance and energy usage in NUMA systems

**Doctoral Meeting** February 2019

#### Miquel López Becoña

Centro Singular de Investigación en Tecnoloxías da Información





Centro Singular de Investigación

citius uscles

L Context

Motivation and goals Hardware Counters Berkeley Roofline Model Intel RAPL

- Performance analysis and optimization tool
  Introduction
  Optimization strategies
- 3 Energy usage modelling Relation with RM Other events
  - Conclusions and future work



Context Motivation and goals Hardware Counters Berkeley Roofline Model Intel RAPL

2 Performance analysis and optimization tool Introduction Optimization strategies

- 3 Energy usage modelling Relation with RM Other events
- 4 Conclusions and future work



## Motivation and goals

- Memory gap: data locality is a key matter in performance.
  - Specially in NUMA systems.
- Power consumption issues.
- Goals: runtime optimization, performance and energy usage modelling.
- How? Development of a tool that:
  - Reads and gathers hardware counter information about the activity of each thread.
  - Uses of Roofline Model as the basis to characterize the performance.
  - Performs migrations to improve locality.



#### Hardware Counters

- Microprocessor specific registers.
- Intel PEBS: sampling.
- Low overhead and high accuracy.
- Issues while measuring FLOPS: replaced by instructions. More general.
- perf\_events Linux interface to access and extend its information.
- Each sample dumps the core state.
  - Generic fields: timestamp, CPU ID, PID, TID ...
  - Hardware events: what we actually measure.



		PEBS record format											
	Offset	64bit/8 bytes Field											
1	0×0	R/EFLAGS											
	0x8	E/REIP											
	0x10	R/EAX											
	0x18	E/EBX											
	0x20	R/ECX											
	0x28	R/EDX											
	0x30	R/ESI											
	0x38	R/EDI											
	0x40	W/EBP											
	0x48	R/ESP											
	0x50	R8											
	0x58	R9											
	0x60	R10											
	0x68	R11											
	0x70	R12											
	0x78	R13											
	0x80	R14	6										
	0x88	R15											
	0x90	IA32_PERF_GLOBAL_STATUS											
$\setminus$	0x98	Data linear Address											
	0xA0	Data Source encoding	1										
	0xA8	Latency (core cycles)	-										

## Berkeley Roofline Model

- Performance model.
- 2D plot: simple and easy to understand.
- Helps finding bottlenecks.
  - Clues to optimize our applications.
- X axis: operational intensity.

FLOPS bytes\_read\_from\_memory

- Y axis: raw performance (GFLOPS/sec).
- Roof/s: hardware limits.
- Basic classification: side of the graphic.
- Extensions: Dynamic Roofline Model (DyRM), and 3DyRM (useful for NUMA)<sup>2</sup>



<sup>&</sup>lt;sup>2</sup>Óscar García Lorenzo et al. "3DyRM: A dynamic roofline model including memory latency information" CinUS <sup>3</sup>Samuel Williams et al. "Roofline: an insightful visual performance model for multicore architectures"

Context

## Example of generated Roofline Model





4/17

## Which hardware events do we currently use?

- Mainly those which are associated to the Roofline Model.
- Instruction count (performance related).
- Offcore requests: cache lines read from memory (performance related).
- Memory access latency (data locality related).
- Energy usage per node (load balancing related), with the aid of Intel RAPL.



# Intel RAPL (Running Average Power Limit)

- Software interface to estimate energy usage.
- Divides consumption between logical domains:

cores

- pkg: cores + LLC + memory controller + ...
- ▷ ram
- Variable support.
- Set of buffers for each NUMA node.



# Example energy usage plot





Time (s)



1 Context

Motivation and goals Hardware Counters Berkeley Roofline Model Intel RAPL

Performance analysis and optimization tool
 Introduction
 Optimization strategies

3 Energy usage modelling Relation with RM Other events

4 Conclusions and future work



## Performance analysis and optimization tool

#### Main goal

To characterize the performance of parallel programs and performs thread and page migrations accordingly using hardware counters information.

- **Two main modes:** JUST\_PROFILE and DO\_MIGRATIONS.
  - CSV dumping for a posteriori analysis.
- Requires Linux OS and a Intel microprocessor.
- Not very intrusive: low overhead and easy to install and use.
- No root permissions required\*.
- Automatic detection of system topology (relation of CPUs per NUMA node, etc.).
- Language-independent solution (code is not analysed).

# Example of Dynamic Roofline Model plot

Additional R and Python code for analysing data and making neat plots.





## Optimization strategies

- Decisions about which thread/page to migrate and where.
- Selection of a set of migrations (that might be empty) per iteration.
- Modular implementation of the strategies.
- Usually based on classic optimization strategies and search problems.
  - Simulated annealing, genetic algorithm, random, energy-balancing...
- Most of them are in an early stage of development.



1 Context

Motivation and goals Hardware Counters Berkeley Roofline Model Intel RAPL

- Performance analysis and optimization tool
  Introduction
  Optimization strategies
- 3 Energy usage modelling Relation with RM Other events

4 Conclusions and future work



## Energy usage modelling with Roofline Model

- Let's model how energy usage is affected by performance metrics.
- RAPL domains!
- First approach: Roofline Model.
- pkg related to CPU activity (ginsts/s), ram related to memory operations (OI).
- First approach: energetic Roofline.



Local ops per second vs ram



# **Energetic Roofline**



Operational Intensity (Insts/Byte traffic)



## Relation with DyRM

Good results in some cases, but not in all of them...





## Other hardware events

- Maybe we should take into account other/more metrics?
- Search for more correlations...

																			Ev	ent-	en	erg	y u	sa	ge	cor	rela	atic	ons	for	r be	enc	hm	ark	c bt																			
-R					0.56	6 -0.631	0.851				41 0			1504	0.263	0.742	0.0556						0718				0.0	95 01																0.037	n as		81 O					0.552	0.425	
5	0.545	0.787	0.89	0.652	0.926	0.853	0.80	0.42	5 Q.4	a 0.0	42 0	173 0		0.699	0.362	0.749	0.16	0.471	0.855	0.856	0.648	0.916	0.813	0.71	0.73	-0.03	N 031	55 - O.	324 K	.924	246	0.248	0.395	0.241	0.57	0.511	0.373	0.221	-0.44	0.215	0.828	0.823	0.35	0.0153	a o	951 Q.3	41 0	963 O	0 55 1	0.909	0.124	0.653	0.413	
	BE WERE DISC TANEN, CONDITIONLE	ARTHERIN DIV, ACTINE	CIVER ACTIVES YEARING ACTIVE	כאמוני אבאוווע צוארו לי אמאוויני	CICLE, ACTIVITY STOLES, TOTAL	SNORM NOT STINGS ALMUST TO D	DEEDMIE JANTO-ESI-FEALTY_CITLES	OTLIB LOND JINSSES MOSE CAURES A, JINUK	DTLB.LCMD. MISSESIMULK. COMPLETED	38 JH STUSSESSI OHOT STUD		NDL8, LOAD, MESBERINAL, CURATION	CILLE LOND MISSESSMUC CONFLICTIO 44	DRUA_STORE_MESSES STUA_HT	TLB, STORE, MISBES MISS, CAUSES, A, MILLK	36,74, LT2:23056, 36072, LT2	TRUE STORE WESESWALK COMPLETED AK	NOTABLIC NUMBERS AND ADDRESS AND	P_ARTHSCAUR, DOURG	BULLER REPORT	COEH	THE VERSION	CLOEMERS	saniazoba	sdon") "stroug" teat" my ba	struc's on asoba	ngaann		STELO SADO TANÀO	SQUE, TRANS	arcool and savida	5401 222 28 02	san' uai seba	STELC'SADO' BAD' SHORE	subil Swba	STRUME SHOT	BQ_UDPS_NDF_DELNERED.COME	strad saor suba	T DRIMERED CYCLES, D. UDPS, DRIM, CORE	ION, NOT DELIVERED CITCLES, PL, MAS, DK	DELMERED CYCLES JE 2, UP DELM CORE	BEINERED CHORE IE 1 UDV DELN CORE	DELINERED CYCLER, JE. J., JOP., DELIN, CORE	4017165 <sup>°</sup> 01	MOTO-PTO-PLACE	4 AND DECEMBER 199	Annual State	NT NBC SNT STALL	MT_MISCRECOMBIN_CYCLES	NU_MECSECONERY_CHILES_MY	MF_WISCRECONDIN_STALLS_COUNT	สถาชาชาย	TLL, JASSES MES, CAUSES, A, JANUK	
						-0.8										-0	4										0.0										(	),4	00,00		0.00	C.UOM	0,000					0.8						



## Current stage: machine learning

- Join data for lots of events: too many columns in CSV.
- Dimensionality Reduction: PCA (Principal Component Analysis).
- Machine learning model with the most N relevant events.



1 Context

Motivation and goals Hardware Counters Berkeley Roofline Model Intel RAPL

- Performance analysis and optimization tool
  Introduction
  Optimization strategies
- 3 Energy usage modelling Relation with RM Other events
  - Conclusions and future work



## Conclusions

#### Development of a new tool

- To characterize the performance of parallel applications in NUMA systems.
- To optimize memory accesses through page and thread migrations.
- To lower power consumption.

#### Energy usage research work

To explain it based on hardware counter information.



#### Future work

- Progressing in the energy study.
- Algorithm refinement for current strategies.
- Implement new strategies.



miguel.becona@usc.es





Cofinanciado pola Unión Europea

Fondo Europeo de Desenvolvemento Rexional "Unha maneira de facer Europa"