Energy Efficiency of Software TM on heterogeneous multi-core CPUs

Emilio Villegas, **Alejandro Villegas**, Angeles Navarro, Rafael Asenjo, Oscar Plata

Dept. Arquitectura de Computadores

Universidad de Málaga

Yash Ukidave

Advanced Micro Devices

Outline

- Motivation & Background
- TM Applications on Heterogeneous CPU
- Evaluation
- Conclusions and future work

Outline

- Motivation & Background
- TM Applications on Heterogeneous CPU
- Evaluation
- Conclusions and future work

Motivation

- TM as optimistic mechanism for mutual exclusion
- Is energy efficient?
  - Serialization [ispled-2005, sbac-pad-2010, pdp-2012] and clock gating [ipdps-2010] as means to improve energy efficiency in HTM
  - DVFS techniques to improve energy efficiency in STM [cal-2009, sbac-pad-2010]

- Heterogeneous CPUs
  - ARM big-little
    - » Powerful big cores
    - » Energy efficient little cores

TM may benefit from heterogeneous CPUs in terms of energy and performance

Motivation

- TM analysis on heterogeneous CPUs
  - Simulation platforms
  - Hardware: Odroid-XU3
- Research space
  - TM for heterogeneous CPU
    - » Optimize STM libraries
    - » Hardware support
  - TM application design
  - Scheduling TM applications
  - …

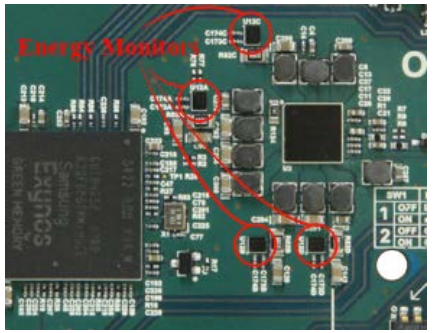
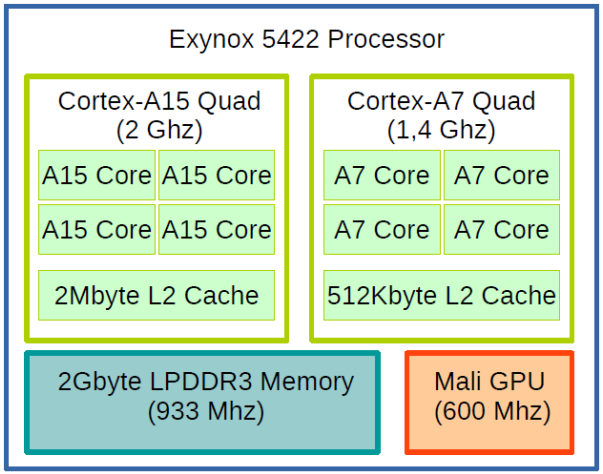


Image Source: http://www.hardkernel.com/main/products/prdt_info.php?g_code=g140448267127



Motivation

- We plan to create a scheduler for TM applications on heterogeneous CPUs using:
  - Energy metrics
  - TM metrics (i.e., commit ratio…)
  - Different policies (i.e., minimize energy consumption, minimize execution time…)

1. Instrument applications
2. Analyze applications

} Work in progress!

3. Use metrics to build an energy-efficient scheduler

Outline

- Motivation & Background
- TM Applications on Heterogeneous CPU
- Evaluation
- Conclusions and future work

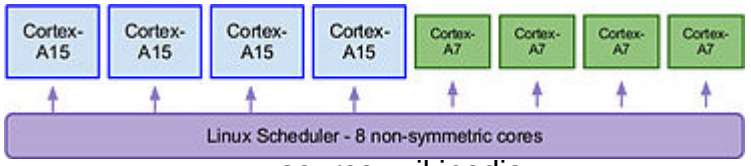
Instrumenting TM applications

- The energy meter library accesses the hardware sensors and integrate current over time
- Replace timing instrumentation with energy meter instrumentation
  - The energy meter library includes a timer
- We add the energy consumption of all sensors into a single counter:
  - We plan to measure big cores, little cores and memory energy consumption separately.
  - This will provide more precise information on the application running on a given core

```
MAIN(argc, argv)
{
    /*
     * Initialization
     */
...
    Energy * e;
    e = energy_create();
    energy_init(e);
    /*
     * Run transactions
     */
...
    energy_start(e);
#ifdef OTM
#pragma omp parallel
    {
        router_solve((void *)&routerArg);
    }
#else
   thread_start(router_solve,(void*)&routerArg);
#endif
energy_stop(e);...
    /*
     * Check solution and clean up
     */
    energy_printf(e, stderr);
    energy_destroy(e);
}
```

Instrumenting TM applications

- We selected TinySTM as software TM
  - We plan to evaluate other STM systems.
  - Review comments:
    - » Correctness / Memory consistency
- Default scheduler: global task scheduling
  - By default, use the big cores:
    - » Threads 1 to 4 are allocated on the big cluster
    - » Threads 5 to 8 are allocated on the little cluster
  - To set affinity, we use *taskset*
    - » However, it ignores the use of mixed cores (i.e., 2 big and 2 little)



source: wikipedia

- We tried to instrument the TinySTM operations
  - The execution time of these functions is too low compared to the resolution of the sensors

Outline

- Motivation & Background
- TM Applications on Heterogeneous CPU
- Evaluation
- Conclusions and future work

Evaluation

- Some applications did not run:
  - Bayes/Yada/Genome/Kmeans(seq)
  - Applications that execute correcty also pass the verification tests
- Secuential execution usually performed better in terms of energy and execution time
  - Only Labyrinth performed better using TinySTM

App	ExTime(A15)	Energy(A15)
Intruder	42s	80j
Labyrinth	97s	273j
Ssca2	19s	35j
Vacation	44s	84j

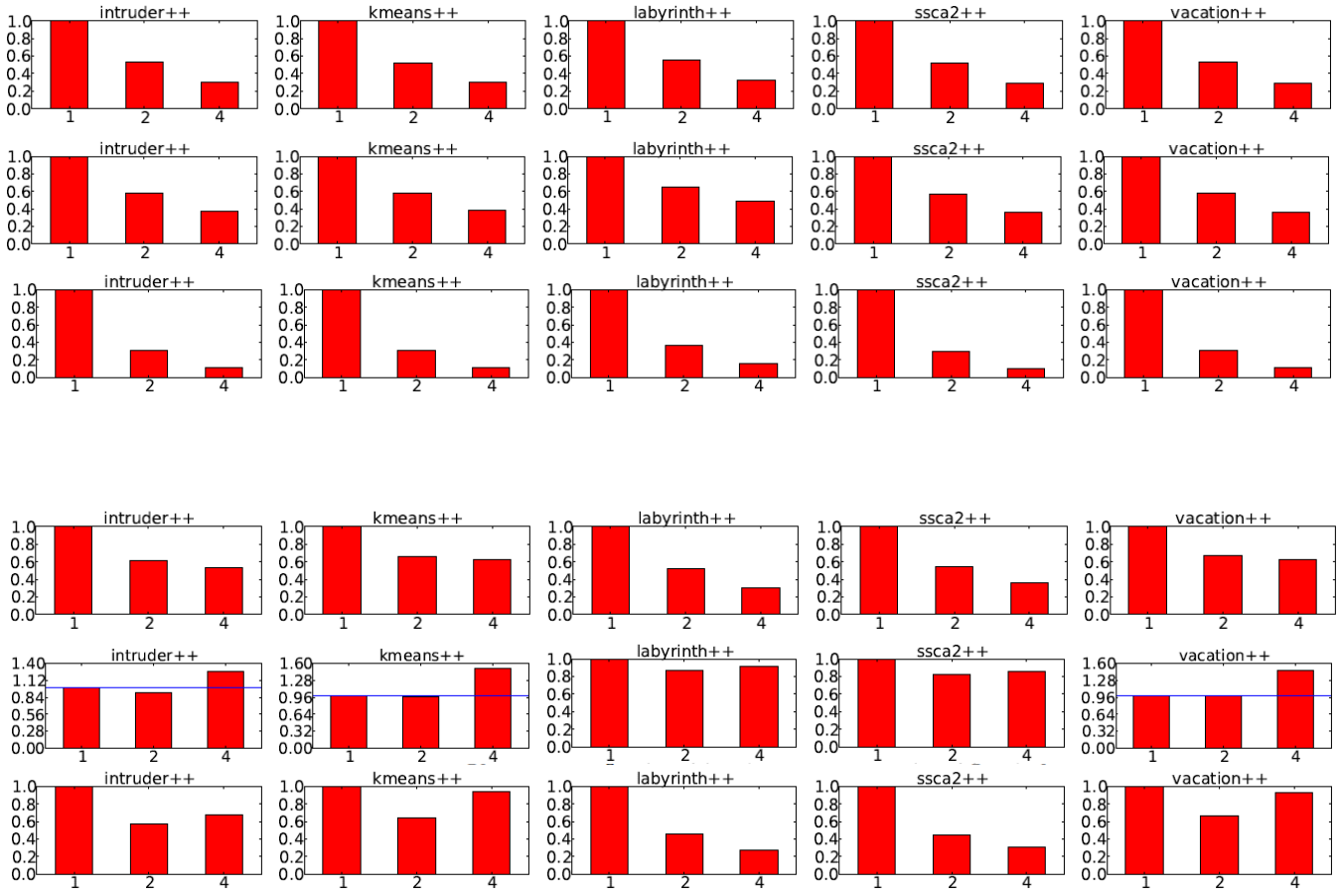
Big

App	ExTime(A7)	Energy(A7)
Intruder	63s	71j
Labyrinth	268s	314j
Ssca2	30s	34j
Vacation	65s	73j

Little

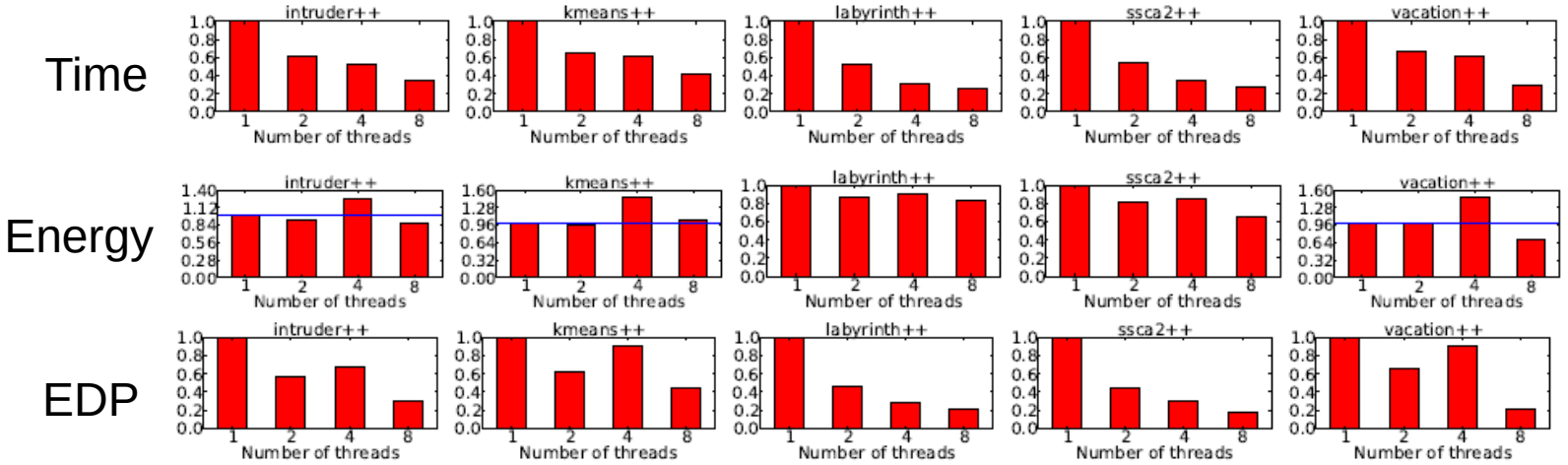
Evaluation

- Little and big cores perform differently



Evaluation

- The OS scheduler quickly moves the workload to the big cores
- Up to 4 threads, the scheduler fills first the big cluster
- 8 threads are able to fill the big and little clusters



Evaluation

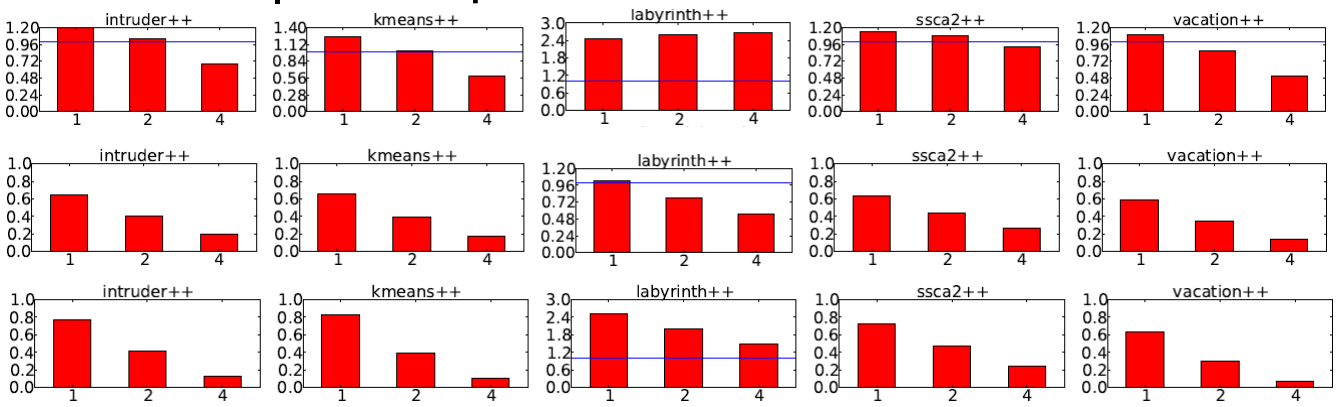
- Little/Big cores comparison
  - >1: A15 performs better | <1: A7 performs better

$$\frac{A7}{A15}$$

Time

Energy

EDP



- We are investigating why some applications perform better with 2 and 4 threads in the little cluster

Aborts $\frac{A7}{A15}$

Intruder	Kmeans	Labyrinth	Ssca2	Vacation
0.74	0.58	0.96	0.63	0.95

We are working on getting other metrics (load imbalance, cache misses, …)

Outline

- Motivation & Background
- TM Applications on Heterogeneous CPU
- Evaluation
- Conclusions and future work

Conclusions

- We analyzed the behavior of TinySTM and STAMP running on a big.LITTLE architecture:
  - Presents better scalability on the little cluster.
  - Demanding applications such as labyrinth present better performance on the big cluster.
- We got useful comments from the reviewers:
  - Analyze correctness, study other TMs, add more metrics

Future work

- Add more metrics for a better understanding of TinySTM on the heterogeneous processors
- Study other TMs and try to get general results
- Extend current task scheduling techniques to consider TM metrics
- Other research directions:
  - Modify previous energy-aware TMs to work on heterogeneous processors
  - Design multi-threaded applications that use TM on heterogeneous CPUs

Some heterogeneous CPUs schedulers:

Fedorova et al. [EuroPar 07]
Chen et al. [ISPLED 12]
Hyatt et al. [CSCI 14]
Yu et al. [ISOCC 13]
Banerjee et al. [IC3 15]
Del Sozzo et al. [DATE 16]
Li et al. [SC 07]

Energy Efficiency of Software TM on heterogeneous multi-core CPUs

Emilio Villegas, **Alejandro Villegas (avillegas@uma.es)**,
Angeles Navarro, Rafael Asenjo, Oscar Plata
Dept. Arquitectura de Computadores
Universidad de Málaga

Yash Ukidave
Advanced Micro Devices