AdaMICA: Adaptive Multicore Intermittent Computing

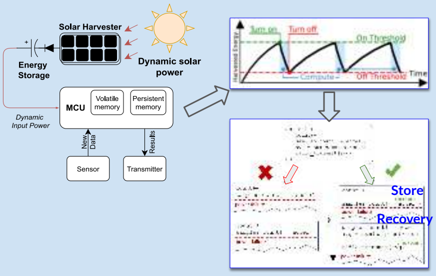
Khakim Akhunov, Kasim Sinan Yildirim

#1 BACKGROUND

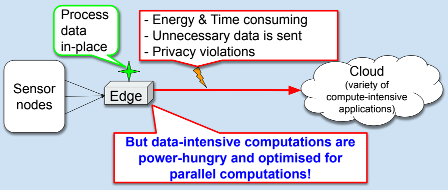
Why do we need to be battery-free?

- 2030 y. -> 50 billion of IoT devices worldwide
- No batteries -> New environments !!!

This is how batteryless systems work



Application requirements are growing



Process data in-place

- Energy & Time consuming
- Unnecessary data is sent
- Privacy violations

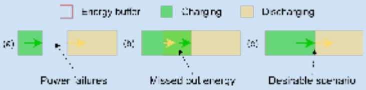
But data-intensive computations are power-hungry and optimised for parallel computations!

#2 PROBLEM

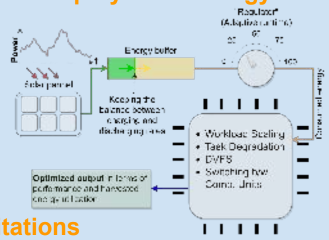
Where to get extra energy?

Ambient Power can reach hundreds of milliwatts, but the energy buffer is small

To get it, intermittent systems need to be "smart" enough, e.i. to be adaptive



How to employ extra energy?

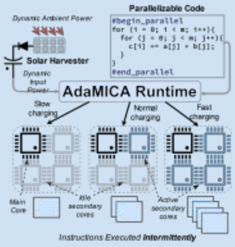

"Regulator" (Adaptive runtime)
Power
Solar panel
Energy buffer
Keeping the balance between charging and discharging area
Workload Scaling
Task Degradation
DVFS
Switching h/w Comp. Units
Optimized output in terms of performance and harvested energy utilization
Consumed energy

Limitations

- Single-core solutions
- Task-specific accelerators
- No generic parallelism exploitation & support
- No computational flexibility

#3 GOAL AND CONTRTIBUTION

The concept of a parallel adaptive system



Goal:

Our objective is to enable the efficient intermittent execution of highly parallelizable computations under dynamic environmental energy.

Challenges:

1. Intermittent computing unique factors (such as the store and recovery overheads) might shade parallelization benefits;
2. Parallel programming model can be further complicated by intermittent programming.

Research Impact:

1. Multicore Intermittent Computing. We introduce the missing software support that enables, for the first time, parallel intermittent computing over multiple cores;
2. Power-scaling Runtime. We introduce the first intermittent runtime that provides the missing parallel programming language constructs and adaptively reconfigures the multicore system concerning the environmental power strength.

#4 METHODOLOGY

Models exploration and extension

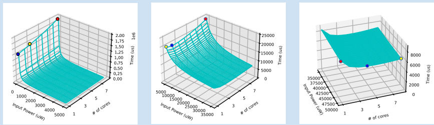
[illegible]

[illegible]

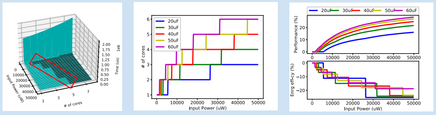
$$\min_{\substack{n \in [1, n_{max}] \\ P_{input} \in \Re_{\geq 0}}} \left(\frac{T_{systemMC}}{T_{systemSC}}\right) < 1$$

Condition when parallelism is desired

High-level simulations



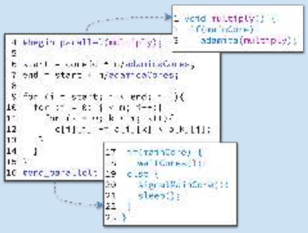
Three parts of the performance dependence on the incoming power and the number of cores working on a task. Red, yellow, blue dots represent high, medium, and low point of the marked line respectively.

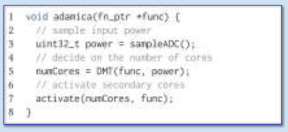


Comparison of the speed up and the energy efficiency drop of multicore intermittent systems.

#5 ADAMICA ARCHITECTURE

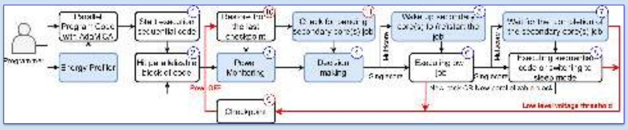
Programming model





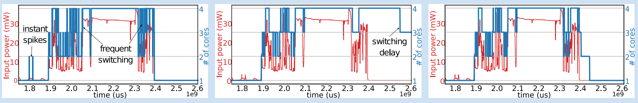
```
void adamica(fn_ptr *func) {
  // sample input power
  uint32_t power = sampleADC();
  // decide on the number of cores
  numCores = DMT(func, power);
  // activate secondary cores
  activate(numCores, func);
}
```

Operating overview


Programmer
Parallel Program Code with AdaMICA
Energy Profiler
Start execution sequential code
Hit parallelizable block of code
Restore from the last checkpoint
Power Monitoring
Check for pending secondary core(s) job
Decision making
Wake up secondary core(s) to initialize the job
Executing the job
Wait for the completion of the secondary core(s) job
Executing sequential code or switching to sleep mode
Checkpoint
Power OFF
Low level voltage threshold

Decision-making efficiency

$$P_{predict} = \frac{W_L * P_{current} + P_{past}}{W_L + 1}$$



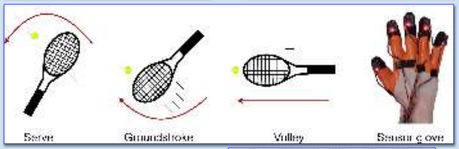
Static decisions

Dynamic decisions

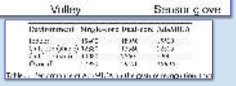
Smart dynamic decisions

#6 EVALUATION AND RESULTS

Experimental setups



Gesture recognition application

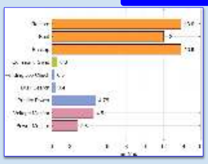


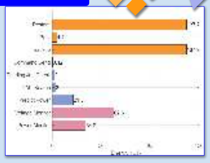
Environment	Single-core	Dual-core	AdaMICA
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
Overall	[illegible]	[illegible]	[illegible]

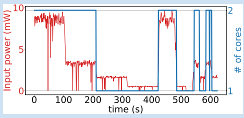
Table [illegible]

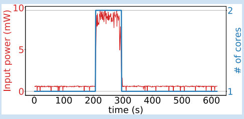
Results

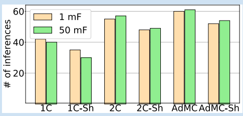
40% speedup

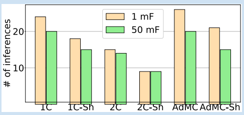




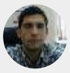








UNIVERSITY OF TRENTO - Italy
Department of Information Engineering and Computer Science



Khakim Akhunov
khakim.akhunov@unitn.it