

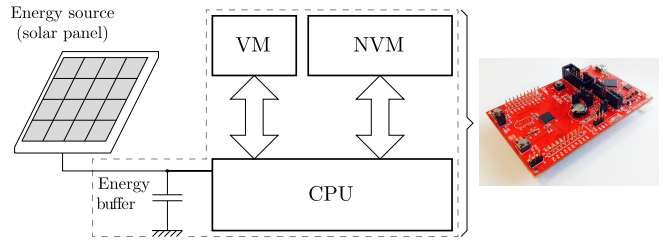
## Battery-less Energy Harvesting IoT nodes

### Eliminating electro-chemical batteries from Energy Harvesting IoT nodes

- Extend lifespan while decreasing needs for maintenance
- Reduce environmental footprint [3]

### Building blocks already available

- Efficient and fast Non-Volatile Memory (NVM)
- Ultra-low-power microcontrollers
- Supercapacitors



### What about energy availability?

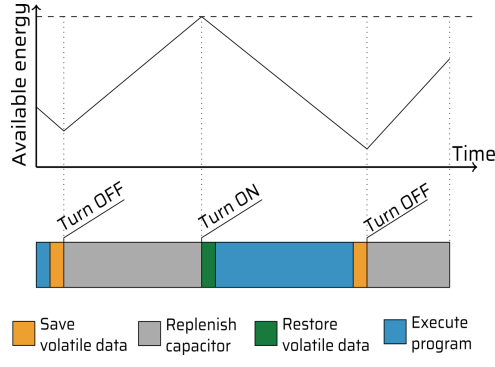
- In limited quantity at a given time
- Is it too limited to perform complex functions (AI, signal processing)?

No! The solution is to weave together computation steps and idle periods to spread the execution over several charging cycles → intermittent computing

## Safe & Efficient Intermittent Computing

### Intermittent system

- Compute when possible
- Deal with intermittency
- Idle when energy is low



### Efficiency

- No useless computations
- Minimize overhead
- No useless checkpoint

### Safe intermittency

- No uncontrolled power failure
- Static guarantees (e.g., atomicity, forward progress)

## RESURRECT: an energy-aware runtime for intermittent systems

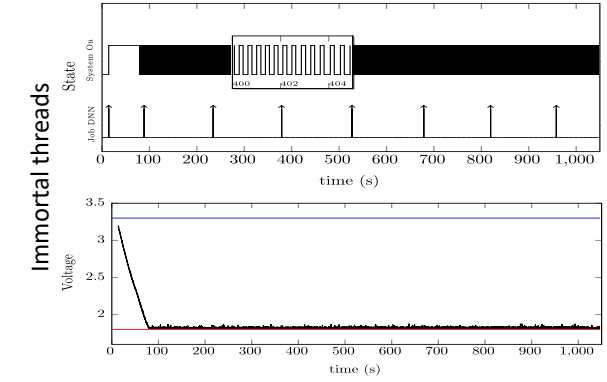
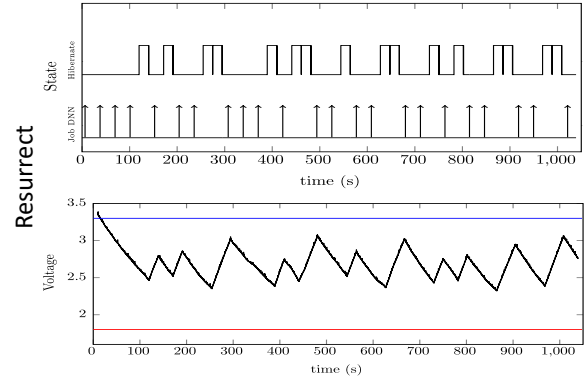
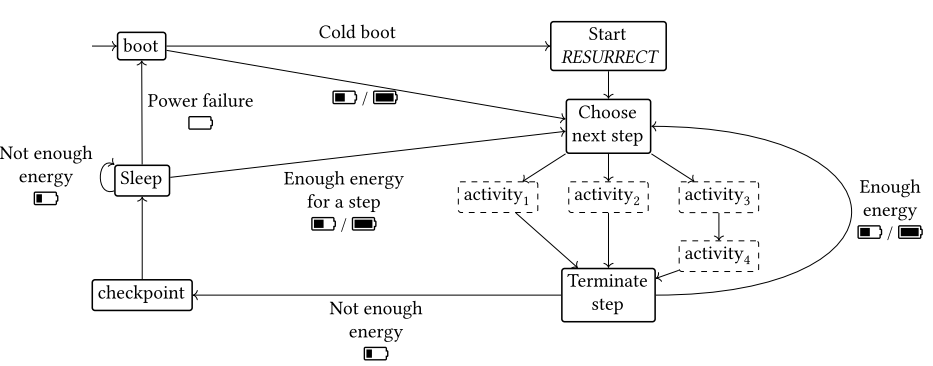
### Objectives

- Hw & Sw tasks, reactive and periodic functions
- Offline computation of optimal schedules [2]
- Weave optimal computation steps and idle periods
- Skip useless checkpoints if enough energy

### Evaluation

- Example of DNN benchmark
- Radio Frequency (RF) energy harvesting setup
- MSP-EXP430FR5994 board

	Number of inference Completed	WCET of inference	Number of inference Completed	WCET of inference
<i>RESURRECT</i>	21	71.6	11	111.6
<i>Immortal Threads</i>	7	151.40	3	324.1
<i>Alpaca</i>	15	72.87	7	159.18



## SCHEMATIC: Compiler-level co-optimization of memory mapping and checkpoint placement [1]

### Objectives

- Split functions that take more than one charge to execute
- Minimize overhead of intermittency management

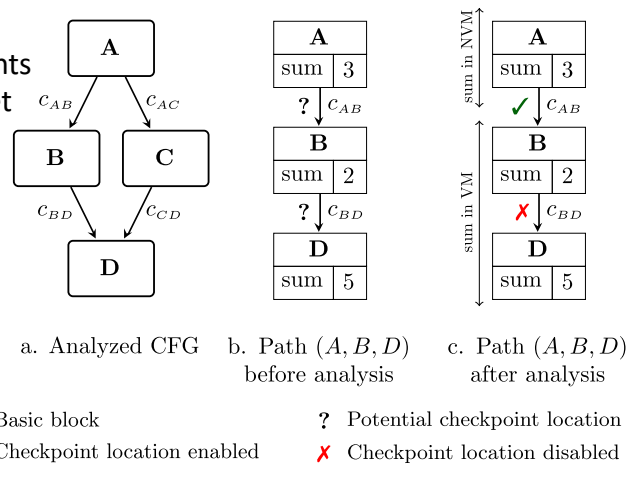
```

int sum = 0;
for(int i = 0; i < SIZE; i++)
    sum += array[i];
/* ..... */
int a = f(sum);
    
```

Annotations: Save checkpoint here?, Store in VM or NVM?, Save checkpoint here?

### SCHEMATIC: Joint checkpoint and memory allocation

- Energy efficient memory mapping between checkpoints
- Insert checkpoints based on worst-case energy budget
- Target the most frequent paths first
- Account for limited size of volatile memory
- Ensure forward progress across load cycles



### Evaluation: SCHEMATIC vs. SotA

- On average: 51% less energy consumed
- No useless computations
- Small number of checkpoints

## Bird Song Recognition Sensor (WIP)

### System design

- Functional design ✓
- Platform design ✓

### Software payload

- Signal acquisition ✓
- Event detection ✓
- Classification (CNN) ✓
- Storage/communication ✓

### Software stack

- LLVM+SCHEMATIC ✓
- RESURRECT ✓

### Hardware platform

- TI MSP-EXP430FR5994 Launchpad ✓
- Solar panel for EH ✓
- Microphone, LPWAN, external NVM ✓
- ✓ Done ✓ In progress ✓ Todo

## Main publications

[1] H. Reymond, J.-L. Béchenec, M. Briday, S. Faucou, I. Puaut, and E. Rohou. 2024. "SCHEMATIC: Compile-time checkpoint placement and memory allocation for intermittent systems." In IEEE/ACM International Symposium on Code Generation and Optimization (CGO).

[2] A. Bernabeu, J.-L. Béchenec, M. Briday, S. Faucou, and O. H. Roux. 2023. "Cost-Optimal Timed Trace Synthesis for Scheduling of Intermittent Embedded Systems." In Discrete Event Dynamic Systems.

[3] V. Lostanlen, A. Bernabeu, J.-L. Béchenec, M. Briday, S. Faucou, and M. Lagrange. 2021. "Energy Efficiency Is Not Enough: Towards a Batteryless Internet of Sounds." In Proceedings of the International Workshop on the Internet of Sounds (IWIS). Best Paper Award.

Contributors: J.-L. Béchenec (LS2N-STR), O. Berder (IRISA-Granit), A. Bernabeu (LS2N-STR), M. Brunet (IETR-ASIC), S. Faucou (LS2N-STR), M. Briday (LS2N-STR), M. Gautier (IRISA-Granit), R. Gerzaguet (IRISA-Granit), M. Méndez Real (IETR-ASIC), S. Pillement (IETR-ASIC), I. Puaut (Inria-PACAP), H. Reymond (Inria-PACAP), E. Rohou (Inria-PACAP)