PRiME Power-efficient, Reliable, Many-core Embedded systems

EPSRC Engineering and Physical Sciences Research Council

# Software-Assistant Runtime Power Management on SpiNNaker

D. Shang, I. Sugiarto, S. Furber, J. Garside
PRiME – Theme 4: Platforms, Applications, Demonstrations

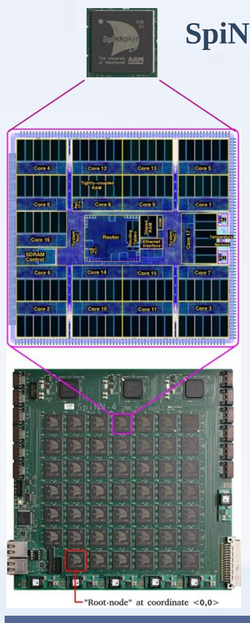
## 1. INTRODUCTION

As one of the most significant scientific inventions of the 20th century, microprocessors and computing systems have tremendous positive impact on everyday life, especially the embedded computing systems – many of which will be low-power mobile devices – in the future. Continuing advances in microprocessor and embedded system design are the keys. Computing systems, however, are facing a once-in-a-generation technical challenge: the relentless increase in processor speed to improve performance of the past 50 years has come to an end.

So，computing systems are being forced to switch from a focus on performance-centric serial computation to energy-efficient parallel computation. This switch is driven by the higher energy-efficiency of using many slower parallel processor cores instead of a single high-speed one. It has attracted worldwide attention and the term “multi-core”, and subsequently “many-core” came into widespread use to generally describe the vision of computing systems with 100s of processor cores, for example the SpiNNaker machine.This motivates us to explore power/energy efficiency and runtime power management on the SpiNNaker machine, a neuromorphic platform with massive parallelisms.

## 2. OVERVIEW of SPINNAKER

The SpiNNaker machine is a novel massively parallel computer architecture, inspired by the fundamental structure and function of the human brain. Each SpiNNaker chip comprises 18 identical ARM968 cores, each core with its own local tightly-coupled memory for storing data (64KB) and instructions (32KB). All cores have access to a shared off-die 128MB SDRAM through a self-timed system network-on-chip (NoC). The full version of the SpiNNaker machine will contain 1,036,800 cores.





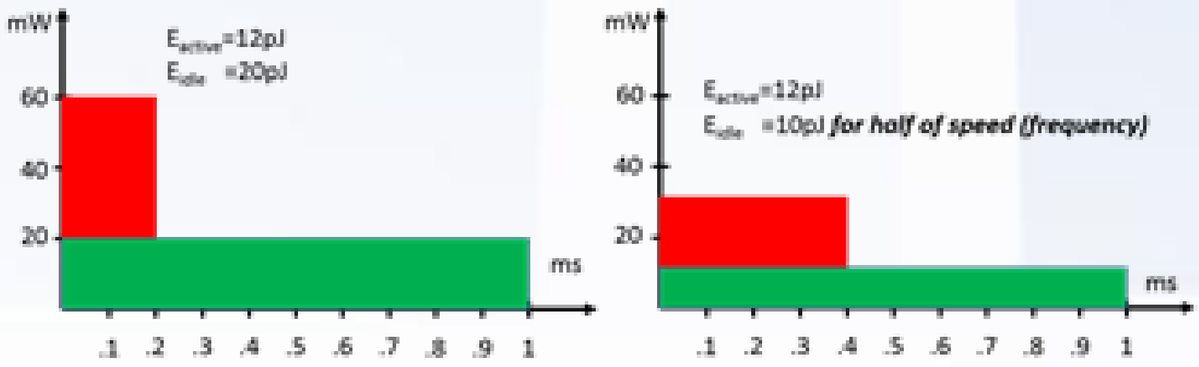
## 3. RUN-TIME MANAGEMENT -- RTM

- Why RTM?
  - Optimizes system performance
  - Maintains long-term reliability
- How does it work?
  - Common: OS-managed (via governor)
    - Reads hardware sensor and program performances
    - Applies some control mechanisms, e.g., PMC-based, ML.
  - Hardware Level: DVFS
    - Increasingly explored in multi-core embedded systems
- What about SpiNNaker?
  - It doesn’t have OS, provides only DFS and thermal sensors
  - **need PMCs defined by software**!

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-(t \cdot A)^\beta} dt$$

## 4. THEORY & AN EXAMPLE

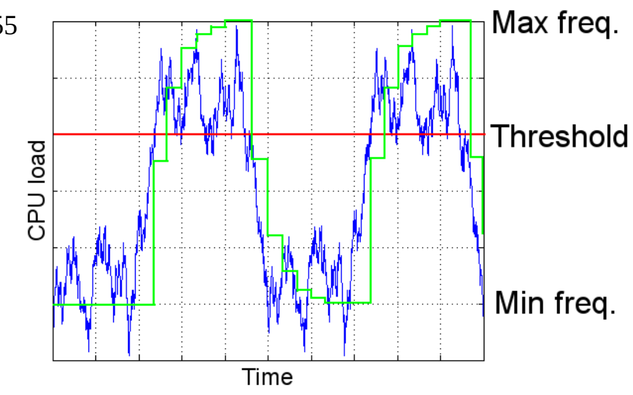
$$E_{total} = E_{active} + Ei_{dle} = Pa_{ctiveTactive} + Pi_{dleTidle};$$
$$T_{total} = T_{active} + T_{idle}$$

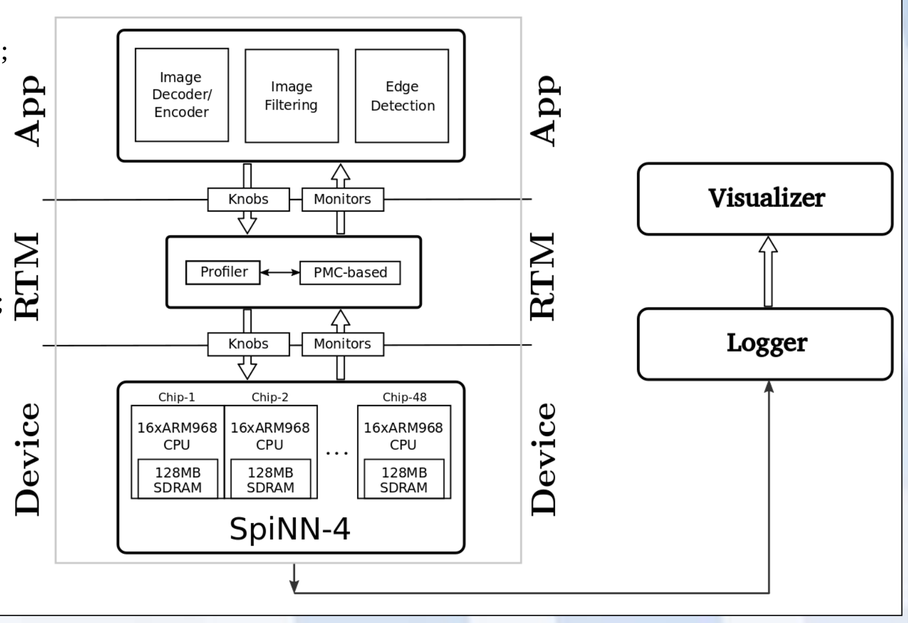



## 5. PROPOSED GOVERNOR & EXPERIMENTS

**Algorithm:** Improved Conservative Governor

```
for every thermal tick do
    if TVC < TVC_THRESHOLD & maxFreq < 255
    then
        maxFreq+ = 5;
    endif
endfor
for every governing tick do
    if TVC > TVC_THRESHOLD
    then
        maxfreq- = 5;
    endif
    get cputil {utilization since last check};
    get cf {current frequency};
    if cputil < CPU_THRESHOLD
    then
        step = (maxFreq-cf)/2;
        if cf <maxFreq
        then
            cf+ = step;
        else
            cf = maxFreq;
        endif
    else
        step = (cf-minFreq)/2;
        if cf > minFreq
        then
            cf- = step;
        else
            cf = minFreq;
        endif
    endif
endfor
return cf;
```




TIMING MEASUREMENT RESULT (IN MILLISECONDS).

| App. | Res. | Governor G1 | G2 | G3 | G4 |
|---|---|---|---|---|---|
| A1 | vga | 955 | 976 | 976 | 975 |
| | svga | 1490 | 1522 | 1522 | 1523 |
| | xga | 2444 | 2498 | 2498 | 2498 |
| A2 | vga | 2670 | 3080 | 3080 | 3080 |
| | svga | 4408 | 4737 | 4737 | 4737 |
| | xga | 7114 | 7342 | 7342 | 7342 |
| A3 | vga | 437 | 454 | 454 | 451 |
| | svga | 674 | 696 | 696 | 697 |
| | xga | 1111 | 1150 | 1150 | 1150 |

Fastest

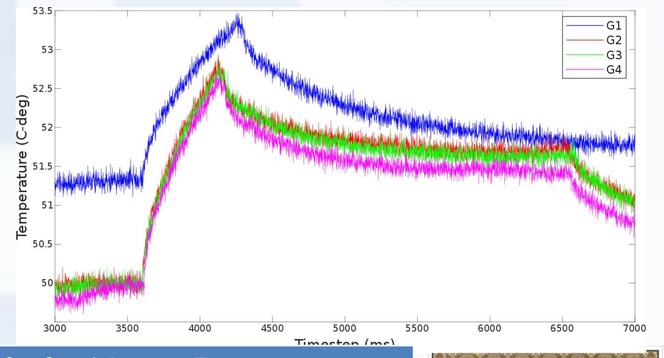
ENERGY CONSUMPTION (IN JOULE).

| App. | Res. | Governor G1 | G2 | G3 | G4 |
|---|---|---|---|---|---|
| A1 | vga | 2.76 | 1.98 | 2.11 | 2.27 |
| | svga | 6.40 | 5.06 | 5.05 | 5.12 |
| | xga | 17.79 | 13.74 | 13.82 | 13.74 |
| A2 | vga | 8.24 | 6.84 | 7.16 | 7.06 |
| | svga | 22.20 | 16.46 | 17.02 | 16.13 |
| | xga | 58.72 | 39.29 | 40.95 | 39.29 |
| A3 | vga | 9.41 | 7.16 | 7.17 | 6.62 |
| | svga | 17.07 | 13.01 | 13.08 | 11.90 |
| | xga | 48.30 | 37.19 | 36.87 | 33.92 |

Lowest

TEMPERATURE INCREASE (IN DEGREE CELCIUS) DURING A PROGRAM EXECUTION.

| App. | Res. | Governor G1 | G2 | G3 | G4 |
|---|---|---|---|---|---|
| A1 | vga | 0.19 | 0.72 | 0.75 | 0.71 |
| | svga | 0.3 | 0.95 | 0.93 | 0.87 |
| | xga | 0.22 | 1.08 | 1.05 | 1.05 |
| A2 | vga | 0.36 | 1.39 | 1.21 | 1.3 |
| | svga | 0.34 | 1.39 | 1.34 | 1.43 |
| | xga | 0.5 | 1.72 | 1.66 | 1.55 |
| A3 | vga | 1.07 | 1.2 | 1.08 | 1.15 |
| | svga | 1.2 | 1.36 | 1.34 | 1.27 |
| | xga | 1.34 | 1.54 | 1.52 | 1.41 |




| Scenario | Description |
|---|---|
| A1 | JPEG image encoding |
| A2 | JPEG image decoding |
| A3 | Filtering + edge detection |

| | |
|---|---|
| G1 | User defined and static/constant Frequency. |
| G2 | On-demand, the higher loads , the higher frequency |
| G3 | Conservation, frequency changes at fixed step interval |
| G4 | This work, freq. step interval is half compared to G3 |


This work was supported by EPSRC through the Graceful project (EP/L000563/1), the PRiME project (EP/K034448/1), and the SpiNNaker project (EP/D07908X/1, EP/G015740/1). The work was also supported by the EU ICT Flagship HBP project (FP7-604102 and H2020-720270) at the University of Manchester.

Imperial College London · intel · arm · Newcastle University · NXP · Imagination · The University of Manchester · Microsoft Research · University of Southampton · nmi Semiconductors to Systems · CONNECTED COMMUNITIES · Knowledge Transfer Network ICT