Dynamic Reconfiguration in Sensor Networks with Regenerative Energy Sources

Ani Nahapetian[1], Paolo Lombardo[2], Andrea Acquaviva[3], Luca Benini[2], Majid Sarrafzadeh[1]

[1]Computer Science Department, University of California, Los Angeles (UCLA)

[2]Dipartimento di Elettronica, Informatica e Sistemistica (DEIS), Università di Bologna

[3]Information Science and Technology Institute (ISTI), Università di Urbino

Talk Outline

- Introduction / Related Work
- Problem Formulation / Assumptions
- Statistical Approach
- Simulation Results
- Case Study – MicrelEye
- Conclusion

Perpetual Operation	Fast, Flexible, Energy Efficient
with Dynamic Reconfiguration	
with Regenerative Energy	

Talk Outline

- Introduction / Related Work
- Problem Formulation / Assumptions
- Statistical Approach
- Simulation Results
- Case Study – MicrelEye
- Conclusion

Regenerative Energy

- **Energy Harvesting** or Energy Scavenging - Capturing energy from the environment
- Systems with **Regenerative Energy** Sources - Systems that obtain or supplement their energy supply with energy captured from the environment

Regenerative Energy

- **Energy Harvesting** or Energy Scavenging - Capturing energy from the environment

❖Applications with dependable energy sources
❖Supplementing battery technology
❖Perpetual operation

- Systems with **Regenerative Energy** Sources - Systems that obtain or supplement their energy supply with energy captured from the environment

How are these Systems Different?

Battery-Powered Systems

- Limited total available energy
- Optimize for limited energy availability over time
- Energy consumption optimization based on battery characteristics

Regenerative Energy Systems

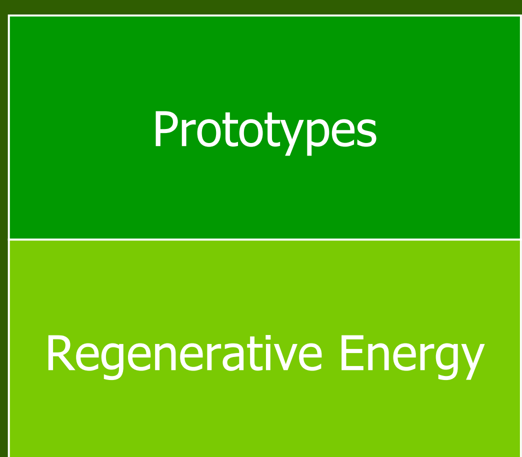
- Perpetual operation may be feasible
- Optimize for limited energy availability at any instance in time
- Instances where better to consume energy
- Considerable variability in energy availability

Regenerative Energy Related Work

- A. Kansal, J. Hsu, M. B. Srivastava, V. Raghunathan, Harvesting Aware Power Management for Sensor Networks. *DAC '06*
- X. Jiang, J. Polastre, and D. Culler, Perpetual Environmentally Powered Sensor Networks. *IPSN/SPOTS*'05

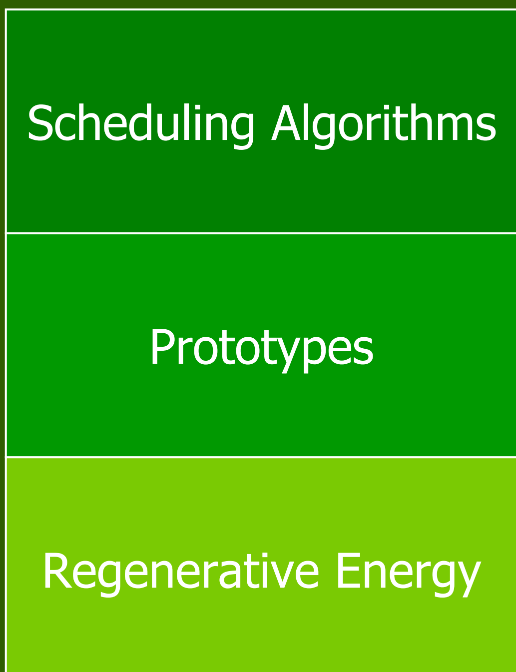
Regenerative Energy

Regenerative Energy Related Work



- Ambulatory motion energy harvesting shoe prototype - MIT
- Vibration energy harvesting – TIMA Labs
- Prometheus project utilizing solar power - Berkeley
- Heliomote project utilizing solar power - UCLA
- Network of mobile nodes roam in search of energy - USC

Regenerative Energy Related Work



- DVS Approach
  - A. Allavena and D. Mossé, Scheduling of Frame-based Embedded Systems with Rechargeable Batteries. *Workshop on Power Management for Real-Time and Embedded Systems* 2001
  - C. Rusu, R. Melhem, and D. Mossé, Multi-version Scheduling in Rechargeable Energy-aware Real-time Systems. *ECRTS '03*
- Online scheduling DVS-independent
  - C. Moser, D. Brunelli, L. Thiele and L. Benini. Real-time Scheduling with Regenerative Energy. *ECRTS '06*
  - C. Moser, D. Brunelli, L. Thiele and L. Benini. Lazy Scheduling for Energy Harvesting Sensor Nodes. *DIPES '06*

Dynamic Reconfigurability with Regenerative Energy Sources

- Low Power
  - Hardware execution more energy efficient
  - Low-power solutions that integrate FPGAs on chip (such as ATMEL)

I. Folcarelli, A. Susu, T. Kluter, G. De Micheli, A. Acquaviva, An opportunistic reconfiguration strategy for environmentally powered devices. *CF '06*

- Limited computational resources in sensor networks
  - Execution of different types of task with the speed and the energy efficiency of hardware.
  - Variety or complexity dictates division into tasks

Talk Outline

- Introduction / Related Work
- Problem Formulation / Assumptions
- Statistical Approach
- Simulation Results
- Case Study – MicrelEye
- Conclusion

Problem Statement

- Intuitively:

  Schedule tasks onto hardware or software for execution, while **manipulating the energy** provided by regenerative sources, while determining when to **reconfigure the FPGA**

  Objective: Ensure the execution of the largest number of tasks, within their availability interval. (In the case of dependencies between tasks, without violating a dependency)

Problem Statement

- Given: Task *i*
  - Arrival time (a_i)
  - Hard deadline (d_i)
  - Energy requirement for execution on hardware (H_i)
  - Energy requirement for execution software (S_i)
  - Type distinguishing reconfiguration profile

Problem Statement

Task types identify whether a reconfiguration is needed between the execution of two consecutive tasks

Possibility of porting reconfiguration data from an external source

- Given: Task *i*
  - – Arrival time (a_i)
  - – Hard deadline (d_i)
  - – Energy requirement for execution on hardware (H_i)
  - – Energy requirement for execution software (S_i)
  - – Type distinguishing reconfiguration profile

Problem Statement

- Given: Task *i*
  - a_i, d_i, H_i, S_i, Type
- Given: Resources
  - Processor on which a software implementation can be executed
  - FPGA with a known reconfiguration cost (or costs)

Problem Statement

- Given: Task *i*
  - a_i, d_i, H_i, S_i, Type
- Given: Resources
  - Processor, FPGA
- Given: Regenerative energy source with an energy buffer
  - Energy loss insignificant
  - External source of energy, which can vary significantly
  - Limited storage capacity

Problem Statement

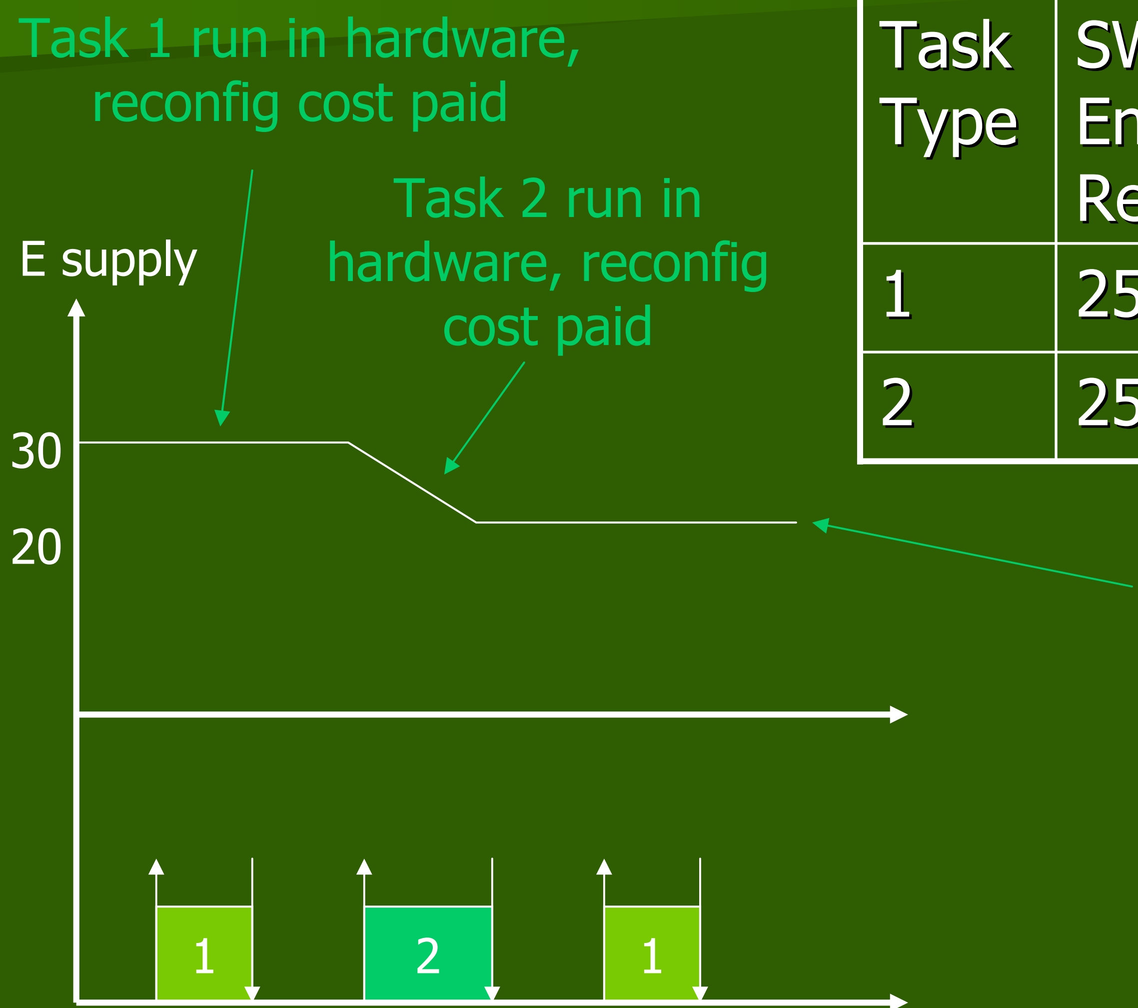
- Given: Task *i*
  - a_i, d_i, H_i, S_i, Type
- Given: Resources
  - Processor, FPGA
- Given: Regenerative energy source with an energy buffer
- Objective: Minimize the number of tasks that miss their deadlines

Assumptions

- Exists both a software and a hardware version of tasks
  - Can handle single implementation, but potential for energy savings is diminished
- Require knowledge of reconfiguration cost, energy consumption of hardware and software task executions
  - Can be profiled

Example

Task Type	SW Energy Req	HW Energy Req	Reconfig Cost
1	25	10	10
2	25	2	



Talk Outline

- Introduction / Related Work
- Problem Formulation / Assumptions
- Statistical Approach
- Simulation Results
- Case Study – MicrelEye
- Conclusion

Key Observations

- Only **last reconfiguration is important** for future reconfigurations and scheduling.
- Reconfiguration is valuable if
  - **IF** large supply of energy (i.e. larger than storage to capacity)
  - **IF** task has large differential between software and hardware execution cost
  - **IF** task is frequent

Expected Energy Calculation

- Evaluate expected energy after some future task executions to determine benefit of reconfiguration now.

$$Exp(E) = E_{current} - R - H_j + Exp(E_A) \cdot F - \left(Exp(E_{type \neq j}) + Exp(E_{type = j}) \right) \cdot F$$

$E_{current}$ – current available E
H_i – HW execution energy
R – Reconfig energy
F – Number of tasks into future

Expected Energy Calculation

- Evaluate expected energy after some future task executions to determine benefit of reconfiguration now.

$$Exp(E) = E_{current} - R - H_j + Exp(E_A) \cdot F$$
$$- \left(Exp(E_{type \neq j}) + Exp(E_{type = j}) \right) \cdot F$$

$E_{current}$ – current available E
H_i – HW execution energy
R – Reconfig energy
F – Number of tasks into future

Expected Energy Calculation

- *Exp(E_{type} ≠ j)* - expected cost of running the next task, of a type other than *j* on SW
- *Exp(E_{type} = j)* - expected cost of running the next task of type *j* on HW, scaled by the likelihood of such a task type occurring.

$$Exp(E_{type \neq j}) = \sum_{l \neq j, l=1}^{TT} \frac{N_l}{\sum_{k=1}^{TT} N_k} S_l$$

$$Exp(E_{type = j}) = \frac{N_j}{\sum_{k=1}^{TT} N_k} H_j$$

TT – Number of task types

N_i – Number of occurrences of task type i

H_i – HW execution energy

S_i – SW execution energy

Extended to Order-2 Statistics

- Consider the possibility of a task following another task.
- Maintain statistics on the pairs of tasks, instead of individual tasks.

$$Exp(E_{type \neq j}) = \sum_{l \neq j, l=1}^{TT} \frac{N_{j,l}}{\sum_{k=1}^{TT} N_k - 1} S_l$$

$$Exp(E_{type = j}) = \frac{N_{j,j}}{\sum_{k=1}^{TT} N_k - 1} H_j$$

TT – Number of task types
$N_{i,j}$ – Number of occurrences of task type j followed by i
N_i – Number of occurrences of task type i
H_i – HW execution energy
S_i – SW execution energy

Extended to Order-2 Statistics

- Consider the possibility of a task following another task.
- Maintain statistics on the pairs of tasks, instead of individual tasks.

$$Exp(E_{type \neq j}) = \sum_{l \neq j, l=1}^{TT} \frac{N_{j,l}}{\sum_{k=1}^{TT} N_k - 1} S_l$$

$$Exp(E_{type = j}) = \frac{N_{j,j}}{\sum_{k=1}^{TT} N_k - 1} H_j$$

TT – Number of task types
$N_{i,j}$ – Number of occurrences of task type j followed by i
N_i – Number of occurrences of task type i
H_i – HW execution energy
S_i – SW execution energy

Expected Additional Energy Computation

- Studied by related work
- Use the product of the expected length of time until the arrival of the next task, D, and the estimated available power, $P_{expected}$

$$Exp(E_A) = P_{expected} \cdot Exp(D)$$

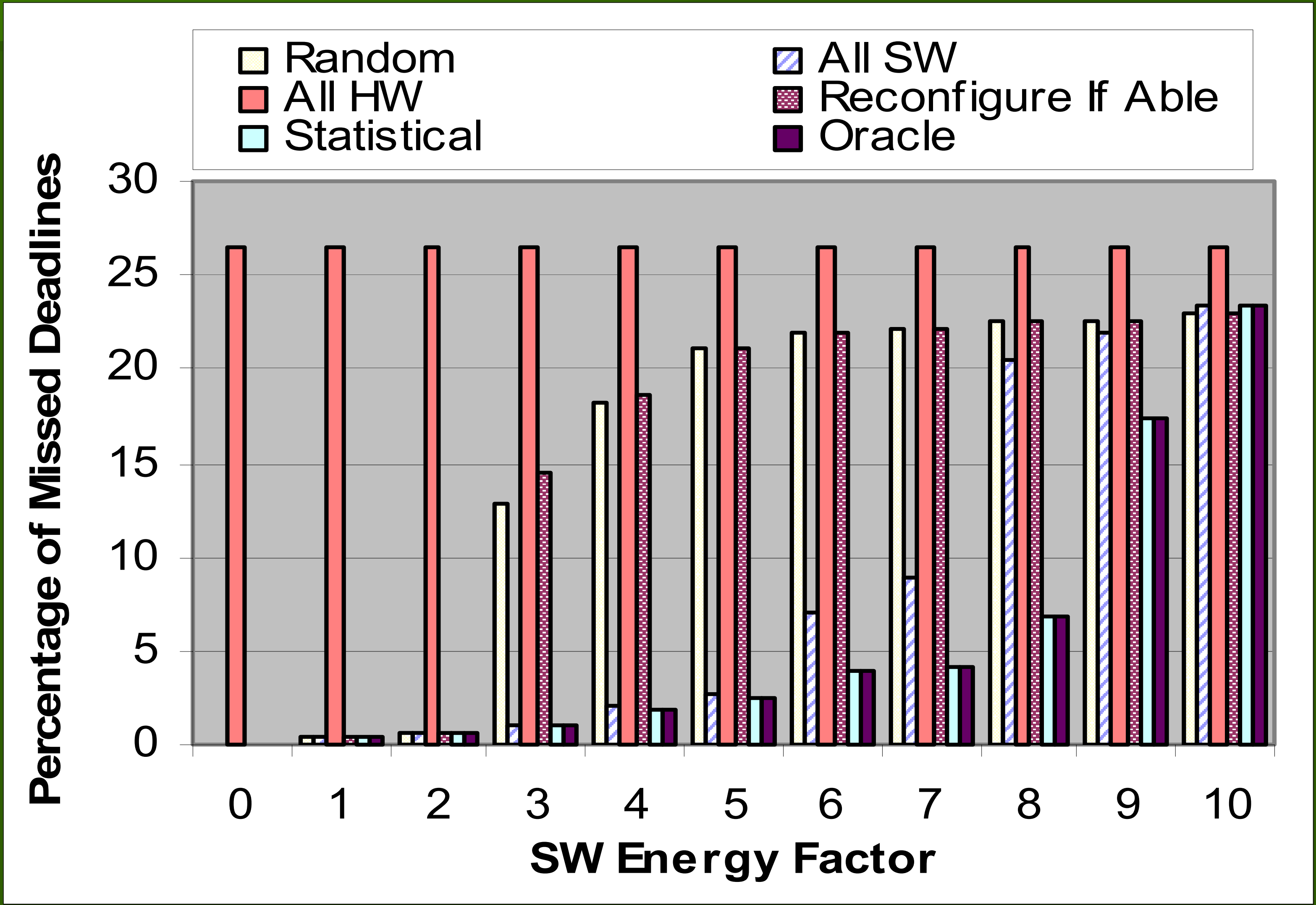
Talk Outline

- Introduction / Related Work
- Problem Formulation / Assumptions
- Statistical Approach
- Simulation Results
- Case Study – MicrelEye
- Conclusion

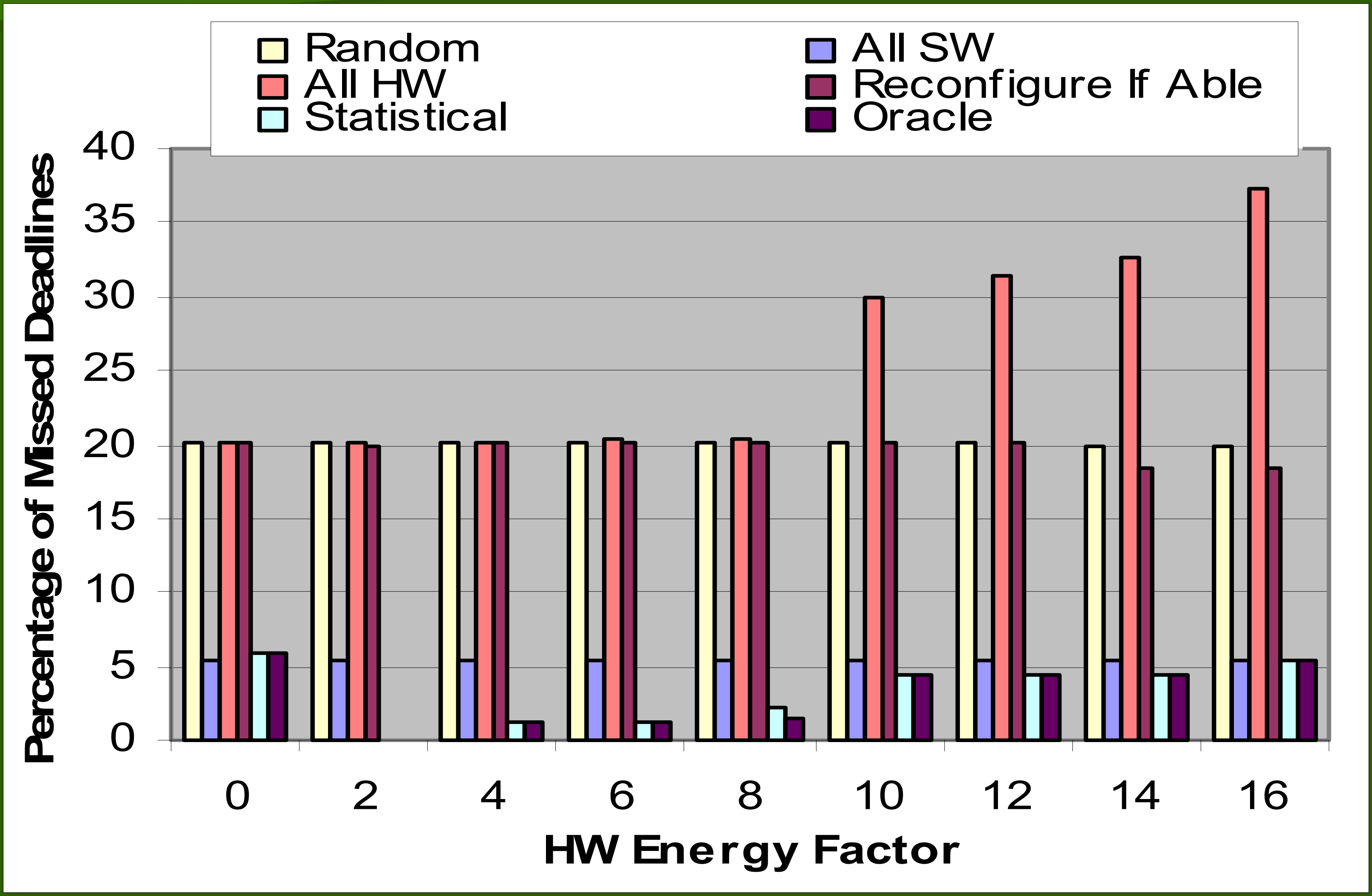
Comparison Approaches

Random	Run task on HW 50% of the time. If not enough energy, run in SW
All-HW	Always run task on HW
Reconfig-if-able	Run task on HW, by reconfiguring if needed. If not enough energy, run in SW
All-SW	Always run task in SW
Statistical	Calculates expected energy after execution of two tasks
Oracle	Aware of immediate harvested energy profile and future tasks

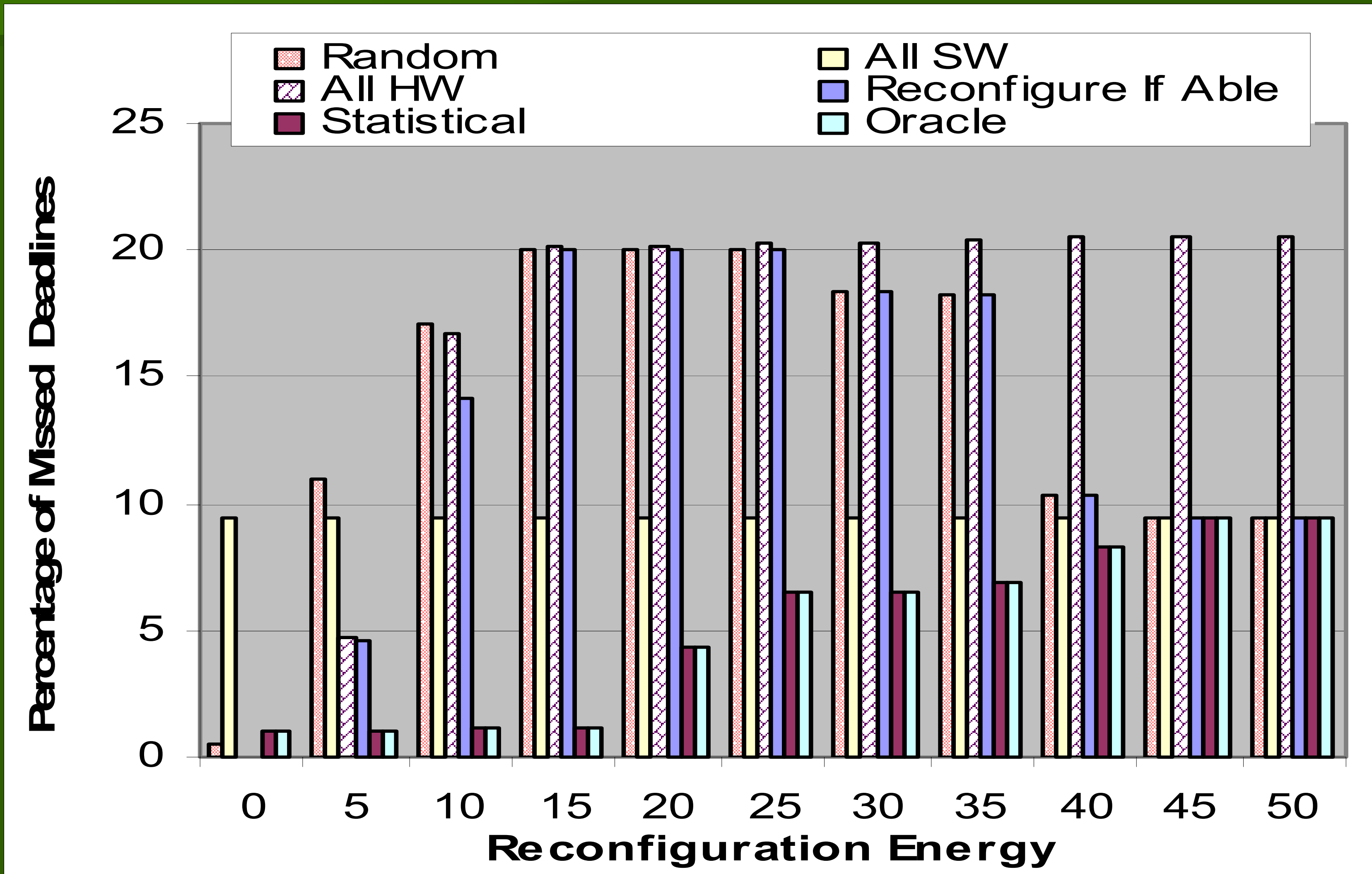
Deadline Misses for Various Software Energy Costs



Deadline Misses for Various Hardware Energy Costs



Deadline Misses for Various Reconfiguration Costs

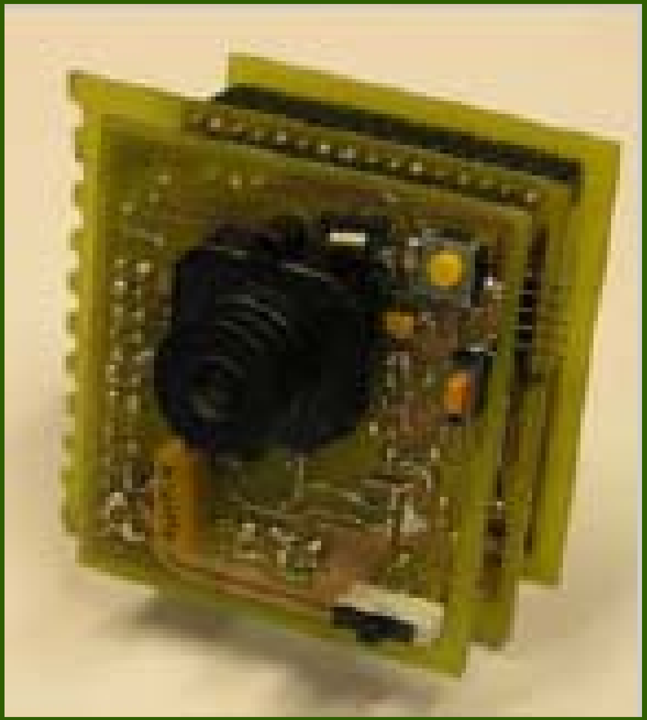


Talk Outline

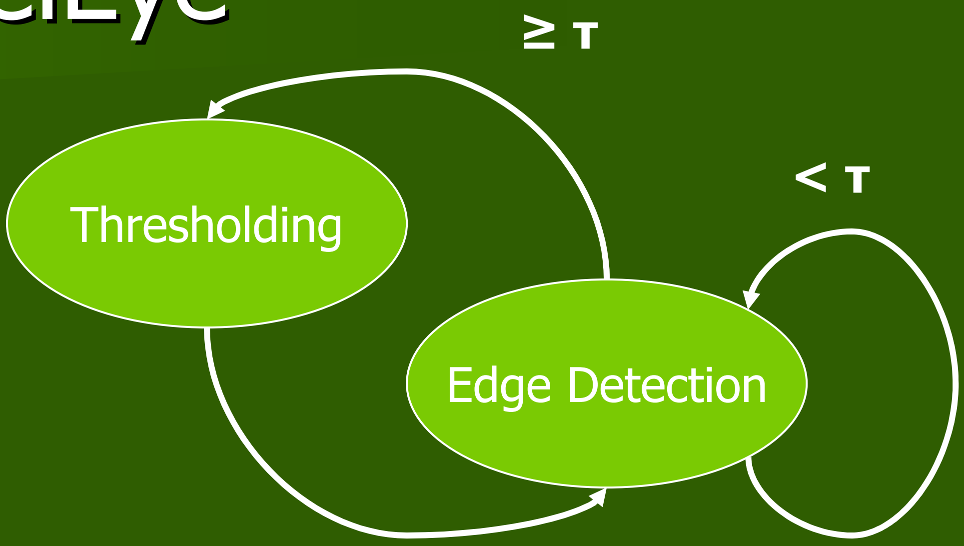
- Introduction / Related Work
- Problem Formulation / Assumptions
- Statistical Approach
- Simulation Results
- Case Study – MicrelEye
- Conclusion

MicrelEye Platform

- Single solar cell and battery
- Omnivision 7640 video sensor
- Bluetooth transceiver
- ATMEL FPSLIC configurable platform, with AVR microcontroller and 40K gate FPGA



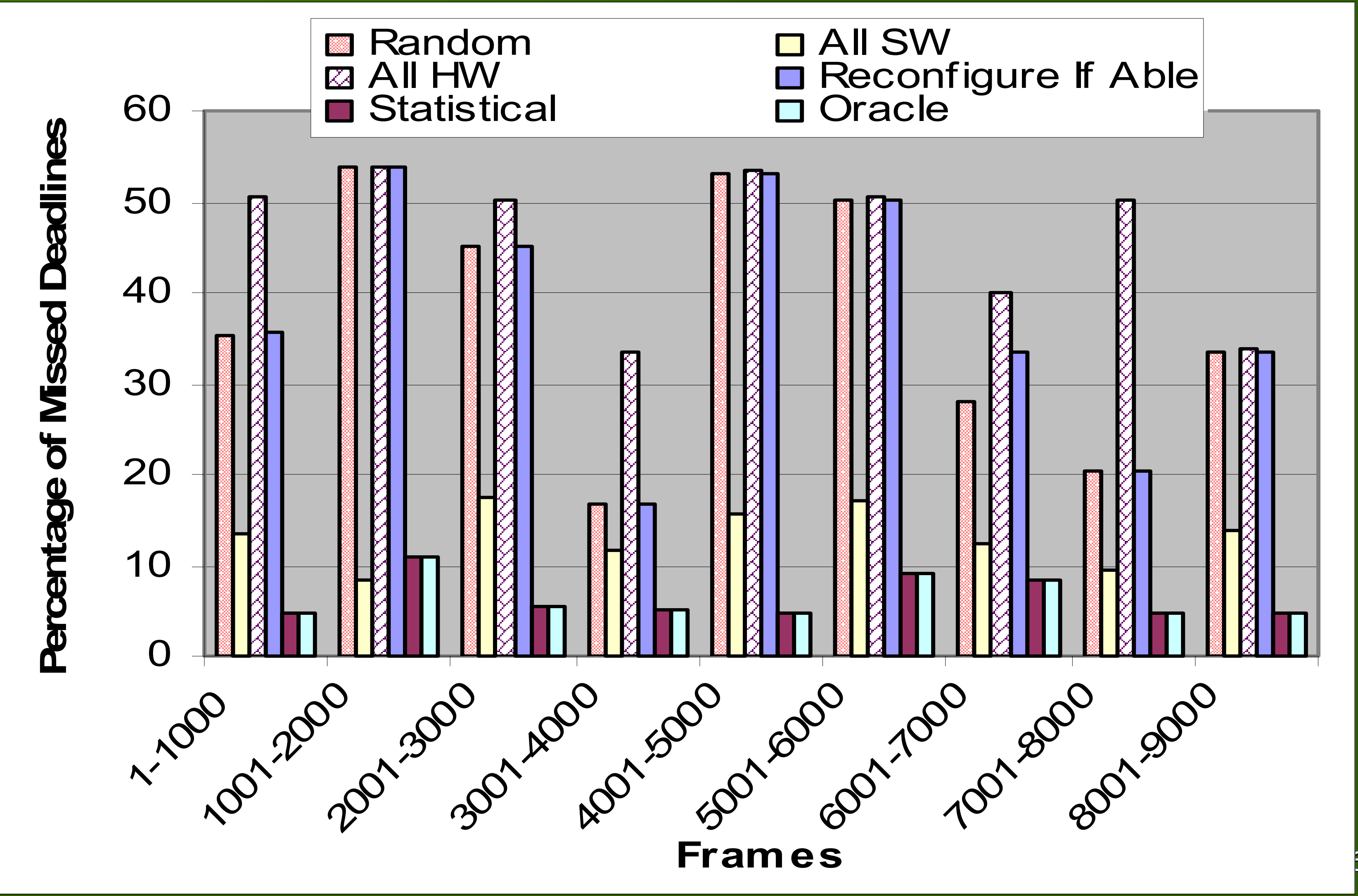
Vision Application Run on the MicrelEye



- Thresholding:
  - Converts a frame from its full 8-bit or 24-bit to a single bit representation for each pixel.
  - Used for object detection.
- Laplacian edge detection:
  - Using Laplacian matrix multiplication
  - Used for tracking

Application	SW E (mJ)	HW E (mJ)	Reconfig E (mJ)
Thresholding	25.0	8.93	4.48
Edge Detection	37.4	28.08	6.60

Deadline Misses for Various Frame Sequences



Conclusion

- **Paradigm shift** caused by regenerative energy sources and need to **integrate reconfigurable devices** into sensor networks nodes
- **Statistically based approach** to schedule tasks
- Evaluation using **simulations** and **MicrelEye prototype system**

<table>
<tr><td>Perpetual Operation</td><td>Fast, Flexible, Energy Efficient</td></tr>
<tr><td colspan="2">with Dynamic Reconfiguration</td></tr>
<tr><td colspan="2">with Regenerative Energy</td></tr>
</table>

Related Work on Regenerative Energy

- Discussion of regenerative energy sources / Sensor networks adapting to perpetual operation
  – A. Kansal, J. Hsu, M. B. Srivastava, V. Raghunathan, Harvesting Aware Power Management for Sensor Networks. *DAC '06*
  – X. Jiang, J. Polastre, and D. Culler, Perpetual Environmentally Powered Sensor Networks. *IPSN/SPOTS* '05

Related Work: Prototypes using Regenerative Energy

- Ambulatory motion energy harvesting shoe prototype
  - J.A. Paradiso, T. Starner, Energy Scavenging for Mobile and Wireless Electronics. Pervasive Computing, pp. 18-27, January-March, 2005
- Vibration energy harvesting
  - Y. Ammar, A. Buhrig, M. Marzencki, B. Charlot, S. Basrour and M. Renaudin, Wireless sensor network node with asynchronous architecture and vibration harvesting micro power generator. *Conference on Smart Objects and Ambient intelligence: innovative Context-Aware Services: Usages and Technologies*, 2005
- Prometheus project utilizing solar power
  - X. Jiang, J. Polastre, and D. Culler, Perpetual Environmentally Powered Sensor Networks. *IPSN/SPOTS* '05
- Heliomote project utilizing solar power
  - http://research.cens.ucla.edu/portal/page?_pageid=56,55124,56_55125&_dad=portal&_schema=PORTAL
- Network of mobile nodes roam in search of energy
  - M. Rahimi, H. Shah, G. Sukhatme, J. Heidemann, and D. Estrin. Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network. *IEEE International Conference on Robotics and Automation*, 2003

Related Work: Scheduling with Regenerative Energy

- Utilize dynamic voltage scaling to approach the problem
  - A. Allavena and D. Mossé, Scheduling of Frame-based Embedded Systems with Rechargeable Batteries. *Workshop on Power Management for Real-Time and Embedded Systems* 2001
  - C. Rusu, R. Melhem, and D. Mossé, Multi-version Scheduling in Rechargeable Energy-aware Real-time Systems. *ECRTS '03*
- Online scheduling approach independent of a dynamic voltage scaling
  - C. Moser, D. Brunelli, L. Thiele and L. Benini. Real-time Scheduling with Regenerative Energy. *ECRTS '06*
  - C. Moser, D. Brunelli, L. Thiele and L. Benini. Lazy Scheduling for Energy Harvesting Sensor Nodes. *DIPES '06*

Related Work: Reconfigurability in Sensor Networks

- Dynamic software reconfiguration in sensor networks
  - T. Tuan, S.F. Li, J. Rabaey. Reconfigurable platform design for wireless protocol processors. *ICASSP '*01
  - S. Kogekar, S. Neema, B. Eames, X. Koutsoukos, A. Ledeczi, and M. Maroti. Constraint-guided dynamic reconfiguration in sensor networks. *IPSN '04*
- Combination of a regenerative energy system with dynamic reconfigurarabilty has first been examined
  - I. Folcarelli, A. Susu, T. Kluter, G. De Micheli, A. Acquaviva, An opportunistic reconfiguration strategy for environmentally powered devices. *CF '06*

Assumptions

- Software execution is more convenient than performing reconfiguration followed by hardware execution

$$S_i \le H_i + R_i$$

H_i – HW execution energy
S_i – SW execution energy
R_i – Reconfig energy
for task i

Assumptions

- Cost of running a task on hardware is less expensive, than running a task on software, ignoring the cost of reconfiguration

$$H_i \leq S_i$$

H_i – HW execution energy
S_i – SW execution energy
R_i – Reconfig energy
for task i