Dynamic Reconfiguration in Sensor Networks with Regenerative Energy Sources

Ani Nahapetian\textsuperscript{1}, Paolo Lombardo\textsuperscript{2}, Andrea Acquaviva\textsuperscript{3}, Luca Benini\textsuperscript{2}, Majid Sarrafzadeh\textsuperscript{1}

\textsuperscript{1}Computer Science Department, University of California, Los Angeles (UCLA)
\textsuperscript{2}Dipartimento di Elettronica, Informatica e Sistemistica (DEIS), Università di Bologna
\textsuperscript{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

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

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


- X. Jiang, J. Polastre, and D. Culler, Perpetual Environmentally Powered Sensor Networks. *IPSN/SPOTS’05*
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**
  - 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*
Dynamic Reconfigurability with Regenerative Energy Sources

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

- **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

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

- **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

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

Possibility of porting reconfiguration data from an external source.
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 1 run in hardware, reconfig cost paid

Task 2 run in hardware, reconfig cost paid

Task 3 can not execute in either hardware or software

<table>
<thead>
<tr>
<th>Task Type</th>
<th>SW Energy Req</th>
<th>HW Energy Req</th>
<th>Reconfig Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

E supply

19
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.

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

- $E_{\text{current}}$ – current available E
- $H_i$ – HW execution energy
- $R$ – Reconfig energy
- $F$ – Number of tasks into future
Expected Energy Calculation

- $\text{Exp}(E_{\text{type} \neq j})$ - expected cost of running the next task, of a type other than $j$ on SW

- $\text{Exp}(E_{\text{type} = j})$ - expected cost of running the next task of type $j$ on HW, scaled by the likelihood of such a task type occurring.

\[
\text{Exp}(E_{\text{type} \neq j}) = \frac{1}{TT} \sum_{l \neq j, l=1}^{TT} \frac{N_l}{\sum_{k=1}^{TT} N_k} S_i
\]

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

- $TT$ – 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.

\[
\begin{align*}
\text{Exp}(E_{\text{type}\neq j}) &= \sum_{l \neq j, l=1}^{\text{TT}} \frac{N_{j,l}}{\sum_{k=1}^{\text{TT}} N_k - 1} S_l \\
\text{Exp}(E_{\text{type}= j}) &= \frac{N_{j,j}}{\sum_{k=1}^{\text{TT}} N_k - 1} H_j
\end{align*}
\]

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
Consider the possibility of a task following another task.

Maintain statistics on the pairs of tasks, instead of individual tasks.

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

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

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, $P_{\text{expected}}$

\[ \text{Exp}(E_A) = P_{\text{expected}} \cdot \text{Exp}(D) \]
Talk Outline

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

<table>
<thead>
<tr>
<th>Approach</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>Run task on HW 50% of the time. If not enough energy, run in SW</td>
</tr>
<tr>
<td>All-HW</td>
<td>Always run task on HW</td>
</tr>
<tr>
<td>Reconfig-if-able</td>
<td>Run task on HW, by reconfiguring if needed. If not enough energy, run in SW</td>
</tr>
<tr>
<td>All-SW</td>
<td>Always run task in SW</td>
</tr>
<tr>
<td>Statistical</td>
<td>Calculates expected energy after execution of two tasks</td>
</tr>
<tr>
<td>Oracle</td>
<td>Aware of immediate harvested energy profile and future tasks</td>
</tr>
</tbody>
</table>
Deadline Misses for Various Software Energy Costs

- Random
- All HW
- Statistical
- All SW
- Reconfigure If Able
- Oracle

Percentage of Missed Deadlines vs. SW Energy Factor

- Bar chart showing the percentage of missed deadlines for different energy factors ranging from 0 to 10.
- The chart compares various strategies for handling missed deadlines, including random, specific hardware (HW), statistical methods, and reconfiguration if possible, with Oracle as a reference.
Deadline Misses for Various Hardware Energy Costs

The graph shows the percentage of missed deadlines for different hardware energy factors. The bars represent various scenarios:

- Random
- All SW
- Statistical
- All HW
- Reconfigure If Able
- Oracle

The x-axis represents the HW Energy Factor, ranging from 0 to 16, while the y-axis represents the Percentage of Missed Deadlines, ranging from 0 to 40. Each scenario is color-coded for easy identification.
Deadline Misses for Various Reconfiguration Costs

![Graph showing deadline misses for various reconfiguration costs. The x-axis represents reconfiguration energy, and the y-axis represents the percentage of missed deadlines. Different lines represent different strategies: Random, All HW, Reconfigure If Able, Statistical, All SW, and Oracle.](image-url)
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
- ATMEEL 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

<table>
<thead>
<tr>
<th>Application</th>
<th>SW E (mJ)</th>
<th>HW E (mJ)</th>
<th>Reconfig E (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thresholding</td>
<td>25.0</td>
<td>8.93</td>
<td>4.48</td>
</tr>
<tr>
<td>Edge Detection</td>
<td>37.4</td>
<td>28.08</td>
<td>6.60</td>
</tr>
</tbody>
</table>
Deadline Misses for Various Frame Sequences

![Graph showing percentage of missed deadlines for different frame sequences and policies.
- Random
- All SW
- All HW
- Statistical
- Reconfigure If Able
- Oracle

The x-axis represents the range of frames (e.g., 1-1000, 1001-2000), while the y-axis shows the percentage of missed deadlines. The graph compares the performance of different scheduling policies across 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
Related Work on Regenerative Energy

- Discussion of regenerative energy sources / Sensor networks adapting to perpetual operation
  - 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

- 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
Related Work: Scheduling with Regenerative Energy

- Utilize dynamic voltage scaling to approach the problem
  - 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*
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*

- Combination of a regenerative energy system with dynamic reconfigurability 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

\[ S_i \leq H_i + R_i \]

\( H_i \) – 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