Refactoring for Energy Efficiency: A Reflection on the State of the Art

Gustavo Pinto
Francisco Soares
Fernando Castor
{ghlp, fmssn, castor}@cin.ufpe.br
Infra x Apps

- Application Level
- System Level
- Hardware Level
Infra x Apps

Application Level

System Level

Hardware Level

IEEE JSSC’92, ISLPED’94

SOSP’03, EuroSys’06
Infra x Apps

<table>
<thead>
<tr>
<th>Level</th>
<th>Conference(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Level</td>
<td>ICSE’15, OOPSLA’14</td>
</tr>
<tr>
<td>System Level</td>
<td>SOSP’03, EuroSys’06</td>
</tr>
<tr>
<td>Hardware Level</td>
<td>IEEE JSSC’92, ISLPED’94</td>
</tr>
</tbody>
</table>
Infra x Apps

- Application Level
- System Level
- Hardware Level

ICSE’15, OOPSLA’14
SOSP’03, EuroSys’06
IEEE JSSC’92, ISLPED’94
What is the problem?

I have no idea on how to improve this parallel code to be more energy efficient :(

[Image of people working on computers]
What is the problem?

I have no idea on how to improve this parallel code to be more energy efficient :(

Is there any tool that can help us to improve our system to consume less energy?
There is a lack of tools for

- Measurement
- Identify opportunities
- Refactoring & Reengineering
- Testing & debugging

There is a lack of tools for

• Measurement
• Identify opportunities
• Refactoring & Reengineering
• Testing & debugging

We are not the only ones!

“This research agenda argues software reengineering tools and techniques, like static and dynamic program analysis, and systematic code transformations like refactoring, can be used to obtain more energy efficient applications.”

Can Refactoring be used to improve the energy efficiency of a software system?
Changing a Map implementation

Changing a Map implementation


Changing a thread management construct

Can software energy consumption research be instantiated in refactorings?
“Power” or “Energy”
“Power” or “Energy”
"Power" or "Energy"
“Power” or “Energy”
20 Selected Papers
19 Selected Papers

No case study
17 Selected Papers

Do not change the source code
14 Selected Papers

Little to do with refactoring
Research Question

• What are the opportunities, and their inherent challenges, to derive new refactorings focusing on improving the energy efficiency of a software system?
Mobile Apps

User Interfaces
• An average of 40% of energy saving when using darker instead of lighter colors.

Challenges
• Color can be dynamically generated.
• Analyze different scattered files.

Mobile Apps

CPU Offloading:
- Can reduce the overall energy consumption of a mobile application by **up to 50%**

Challenges:
- Decide when to refactor;
- Setup the cloud environment;

Concurrent/Parallel Programming

Excessive Copy Chain

- **Energy saving of 15.38%** when the data is shared instead of copied

Challenges

- Identify the copy pattern;

Concurrent/Parallel Programming

Embrace Parallelism

- Parallel solutions can save up to 80% of energy consumption

Challenges

- Not all kinds of problems can be fully parallelizable;
- Energy efficiency can degrade as the user embraces multi-core CPUs;

DVFS Techniques

Stream Programming:
• An average CPU energy saving of 28%

Challenges:
• There is no prior support for refactoring for stream programming;

DVFS Techniques

Energy Types:
• An energy saving of 30% up to 50% on CPU

Challenges:
• When to refactor;
• Take care of new language constructs;

What is next?

• Implement some refactorings
• Integrate the refactoring in existing IDEs
• Evaluate the effectiveness of the refactoring
  • Controlled experiment with practitioners
Refactoring for Energy Efficiency: A Reflection on the State of the Art

Mobile Apps

User Interfaces
- An average of 40% of energy saving when using darker instead of lighter colors

Challenges
- Color can be dynamic generated
- Analyze different scattered files


Concurrent/Parallel Programming

Excessive Copy Chain
- Energy saving of 15.38% when the data is shared instead of copied

Challenges
- Identify the copy pattern;


DVFS Techniques

Stream Programming:
- An average CPU energy saving of 28%.

Challenges:
- There is no prior support for refactoring for stream programming