Software development methodology in a Green IT environment

Hayri ACAR\textsuperscript{a}, Gülfem I. ALPTEKIN\textsuperscript{b}, Jean-Patrick GELAS\textsuperscript{a,c}, Parisa GHODOUS\textsuperscript{a}

\textsuperscript{a} Université Claude Bernard, Lyon 1, LIRIS, France
\textsuperscript{c} ENS Lyon, LIP, UMR 5668, France
\textsuperscript{b} Galatasaray University, Turkey
Usage and users never stop growing…

Internet live stats
INTRODUCTION

USAGE:
Applications
Software
Mobile devices
(Smartphone, Tablet,…)
Web Services
Cloud
Internet

RESULTS:
Power Consumption
Energy Consumption
Greenhouse Gas
Emission
Pollution

How we can optimize our usage to obtain green results?
STATE OF THE ART
STATE OF THE ART

POWER CONSUMING COMPONENTS

- CPU
- MEMORY
- HARD DISK
- NETWORK
- OTHERS
STATE OF THE ART

Software
Tools
Hybrid solutions
Hardware methodologies
STATE OF THE ART

- Software tools:

  - Based on Mathematical formula to estimate Energy consumption.
  - Very few tools in this area of study.
  - Tools are not exhaustive.
SYSTEMATIC REVIEW

Research Nature Facet

Disk

Memory

CPU

Research Type Facet

Solution Proposal

Validation Research

Evaluation Research
Some software tools

<table>
<thead>
<tr>
<th>Tools</th>
<th>Power Model</th>
<th>Appreciations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joulemeter</td>
<td>$E_{\text{system}} = E_{\text{CPU}} + E_{\text{Memory}} + E_{\text{Disk}}$</td>
<td>Globally, estimates energy consumption of all components, but for a given process estimates only energy consumption of CPU.</td>
</tr>
<tr>
<td>vEC</td>
<td>$E = E_{\text{bus}} + E_{\text{cell}} + E_{\text{pad}} + E_{\text{main}}$</td>
<td>Estimates only energy consumption due to memory.</td>
</tr>
<tr>
<td>Orion</td>
<td>$E = E_{\text{read}} + E_{\text{write}}$</td>
<td>Communication components are considered.</td>
</tr>
<tr>
<td>Span</td>
<td>$P(t_j,f_i)<em>{pret} = \Delta P(t_j,f_i)</em>{pret} + P(f_i)$</td>
<td>In the Software code, manually code can be added to show the parts of code involved on the power consumption.</td>
</tr>
<tr>
<td>PowerAPI</td>
<td>$P_{\text{Software}} = P_{\text{comp}} + P_{\text{com}}$</td>
<td>Only CPU and network have been considered.</td>
</tr>
</tbody>
</table>
## STATE OF THE ART

### Power saving techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transistor sizing and reordering</td>
<td>Dynamic power consumption of transistors is decreased</td>
</tr>
<tr>
<td>Network-On-Chip</td>
<td>Improvement over standard bus and crossbar interconnections</td>
</tr>
<tr>
<td>Reduce memory access</td>
<td>Integrate a cache in the classical memory hierarchy of a modern processor</td>
</tr>
<tr>
<td><strong>DFS (Dynamic Frequency Scaling)</strong></td>
<td>Method where frequency of a microprocessor can be adapted automatically</td>
</tr>
<tr>
<td><strong>DVS (Dynamic voltage scaling)</strong></td>
<td>Lowering or increasing the supply voltage of the CPU</td>
</tr>
</tbody>
</table>

03/02/2017
Towards a Green and Sustainable Software
In Proceedings of the 22nd (ISPE) Inc. International Conference on Concurrent Engineering.
POWER MODEL

Dynamic and static power: \( P_{Software} = P_{\text{dynamic}} + P_{\text{static}} \)

Separating dynamic and static power:
\[
P_{Software} = P_{\text{CPU,dynamic}} + P_{\text{CPU,static}} +
P_{\text{Memory,dynamic}} + P_{\text{Memory,static}} +
P_{\text{Disk,dynamic}} + P_{\text{Disk,static}} +
P_{\text{Network,dynamic}} + P_{\text{Network,static}}
\]
POWER MODEL

CPU:

\[ P_{CPU} = C \times V^2 \times F \]

\[ P_{CPU,id} = P_{CPU} \times \frac{N_id}{100} \]

Memory:

\[ P_{DRAM,dyn} = P_{Activate} + P_{Precharge} + P_{Read} + P_{Write} \]

\[ P_{DRAM,id} = P_{DRAM,dyn} \cdot M_id \]

Hard Disk:

\[ P_{Disk} = P_{Active} + (P_{Idle} + P_{Standby} + P_{Sleep}) \]

\[ P_{Disk,dyn} = P_{Activate} = P_{Read} + P_{Write} \]
## Top Programming Languages

<table>
<thead>
<tr>
<th>Rank</th>
<th>Change</th>
<th>Programming Language</th>
<th>May 2016</th>
<th>May 2015</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Java</td>
<td>100.0</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>C</td>
<td>99.9</td>
<td>99.9</td>
<td>-0.09%</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>C++</td>
<td>99.6</td>
<td>99.6</td>
<td>-0.09%</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Python</td>
<td>95.8</td>
<td>95.8</td>
<td>+0.06%</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>C#</td>
<td>91.8</td>
<td>91.8</td>
<td>-0.78%</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>PHP</td>
<td>84.7</td>
<td>84.7</td>
<td>+0.27%</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>JavaScript</td>
<td>84.5</td>
<td>83.0</td>
<td>-1.79%</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>iRuby</td>
<td>83.0</td>
<td>83.0</td>
<td>+0.06%</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Perl</td>
<td>75.3</td>
<td>75.3</td>
<td>+0.51%</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Visual Basic .NET</td>
<td>72.4</td>
<td>72.4</td>
<td>-0.64%</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Delphi/Object Pascal</td>
<td>71.4</td>
<td>71.4</td>
<td>+0.12%</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Assembly language</td>
<td>70.9</td>
<td>70.9</td>
<td>-3.80%</td>
</tr>
</tbody>
</table>

Sources:
- TIOBE INDEX: [http://www.tiobe.com/tiobe_index](http://www.tiobe.com/tiobe_index)
Tool for Estimating Energy Consumption

Sigar

Manufacturer data

TEEC

Software

Power consumption
EXPERIMENTS

Fibonacci sequence is implemented in Java of 2 ways:
- Recursively: 45 terms are calculated.
- Iteratively: calculation of 5,000 values.

TEEC & Joulemeter results are compared.
Quite similar results between TEEC and Joulemeter.

Recursive method consumes more power in amplitude and time than the iterative method.
EXPERIMENTS

Unoptimized code

<table>
<thead>
<tr>
<th>Functions</th>
<th>Unoptimized</th>
<th>Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locality of reference</td>
<td>18140</td>
<td>17219</td>
</tr>
<tr>
<td>Compare array to array list</td>
<td>22047</td>
<td>17297</td>
</tr>
<tr>
<td>Compare integer list loop</td>
<td>7734</td>
<td>7391</td>
</tr>
<tr>
<td>Char array StringBuilder</td>
<td>11235</td>
<td>2421</td>
</tr>
<tr>
<td>Binary search</td>
<td>2250</td>
<td>438</td>
</tr>
</tbody>
</table>
EXPERIMENTS
## EXPERIMENTS

### Nested Loops

<table>
<thead>
<tr>
<th>C Unoptimized</th>
<th>C Optimized</th>
<th>Java Unoptimized</th>
<th>Java Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>char* fn(int N) {</td>
<td>char* fn2(int N) {</td>
<td>public static char[] fn(int N) {</td>
<td>public static char[] fn2(int N) {</td>
</tr>
<tr>
<td>int i = 0;</td>
<td>int i = 0;</td>
<td>int i = 0;</td>
<td>int i = 0;</td>
</tr>
<tr>
<td>char* v = (char*)malloc(N * 2);</td>
<td>char* v = (char*)malloc(N * 2);</td>
<td>char[] v = new char[N*2];</td>
<td>char[] v = new char[N*2];</td>
</tr>
<tr>
<td>while (i &lt; N) {</td>
<td>int tmp;</td>
<td>while (i &lt; N) {</td>
<td>int tmp;</td>
</tr>
<tr>
<td>int j = 0;</td>
<td>while (i &lt; N) {</td>
<td>int j = 0;</td>
<td>while (i &lt; N) {</td>
</tr>
<tr>
<td>v[i] = 0;</td>
<td>v[i] = 0;</td>
<td>v[i] = 0;</td>
<td>v[i] = 0;</td>
</tr>
<tr>
<td>i++;</td>
<td>i++;</td>
<td>i++;</td>
<td>i++;</td>
</tr>
<tr>
<td>while (j &lt; N) {</td>
<td>tmp = 0;</td>
<td>while (j &lt; N) {</td>
<td>tmp = 0;</td>
</tr>
<tr>
<td>v[i] += v[j + N];</td>
<td>v[i] = v[j + N];</td>
<td>v[i] = v[j + N];</td>
<td>v[i] = v[j + N];</td>
</tr>
<tr>
<td>j++;</td>
<td>j++;</td>
<td>j++;</td>
<td>j++;</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
<td>}</td>
<td>}</td>
</tr>
<tr>
<td>return v;</td>
<td>v[i] = tmp;</td>
<td>return v;</td>
<td>return v;</td>
</tr>
</tbody>
</table>
EXPERIMENTS

Nested Loops pour 500 000 en java et en c

Power (W)

Time (s)
- Currently, energy estimation tools aren’t exhaustive.
- TEEC: now estimates CPU, memory, hard disk and network interface card.
- Java agents to determine source code hotpoints.
- Propose green programming best practices (with our expertise).
- Focus on Cloud computing (web services)
PUBLICATIONS

The impact of source code in software on power consumption
*International Journal of Electronic Business Management*,

**TEEC : Improving power consumption estimation of software**
*In Proceedings of the 30th International Conference on Environmental informatics.*
Berlin, Germany, September 14-16, 2016.

**Beyond CPU: Considering Memory Power Consumption of Software**
*In Proceedings of the 5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS).*

**A Green approach to save energy consumed by software**
*In Proceedings of the 3rd International Conference on ICT for Sustainability (ICT4S).*
Copenhagen, Denmark, September 7-9, 2015.

**Towards a Green and Sustainable Software**
*In Proceedings of the 22nd {ISPE} Inc. International Conference on Concurrent Engineering.*