# Software development methodology in a Green IT environment

**Hayri ACAR** [a],
Gülfem I. ALPTEKIN [b], Jean-Patrick GELAS [a,c], Parisa GHODOUS [a]

[a] Université Claude Bernard, Lyon 1, LIRIS, France
[c] ENS Lyon, LIP, UMR 5668, France
[b] Galatasaray University, Turkey

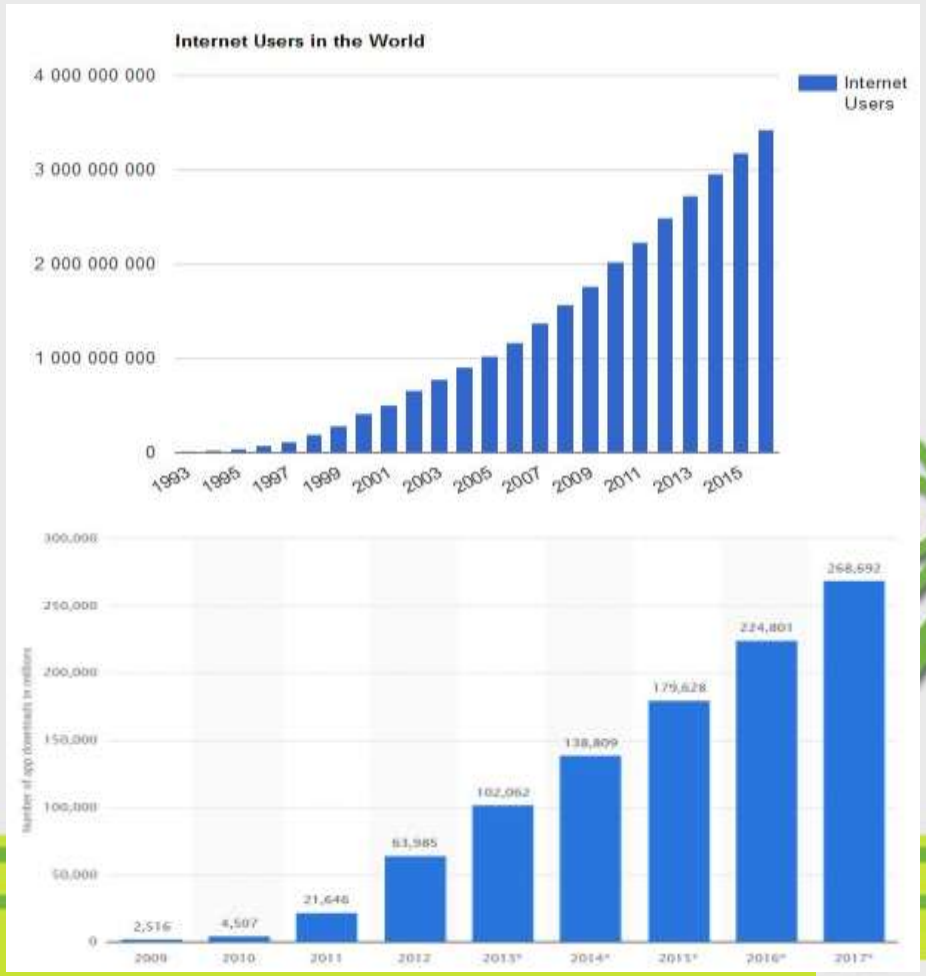
Conférence éco-conception logicielle

03/02/2017

# Usage and users never stop growing…

| | | |
|---|---|---|
| **3,378,456,608** Internet Users in the world | **1,032,410,275** Total number of Websites | **122,427,436,633** Emails sent today |
| **2,571,956,106** Google searches today | **2,387,209** Blog posts written today | **341,019,422** Tweets sent today |
| **5,833,699,214** Videos viewed today on YouTube | **34,182,129** Photos uploaded today on Instagram | **52,968,464** Tumblr posts today |
| **1,666,276,812** Facebook active users | **446,360,937** Google+ active users | **305,326,802** Twitter active users |
| **123,465,035** Pinterest active users | **101,090,689** Skype calls today | **34,998** Websites hacked today |
| **340,860** Computers sold today | **3,579,220** Smartphones sold today | **562,070** Tablets sold today |

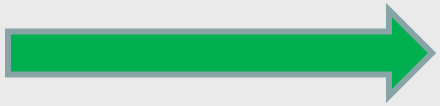
[Internet live stats]



# INTRODUCTION

## USAGE:

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

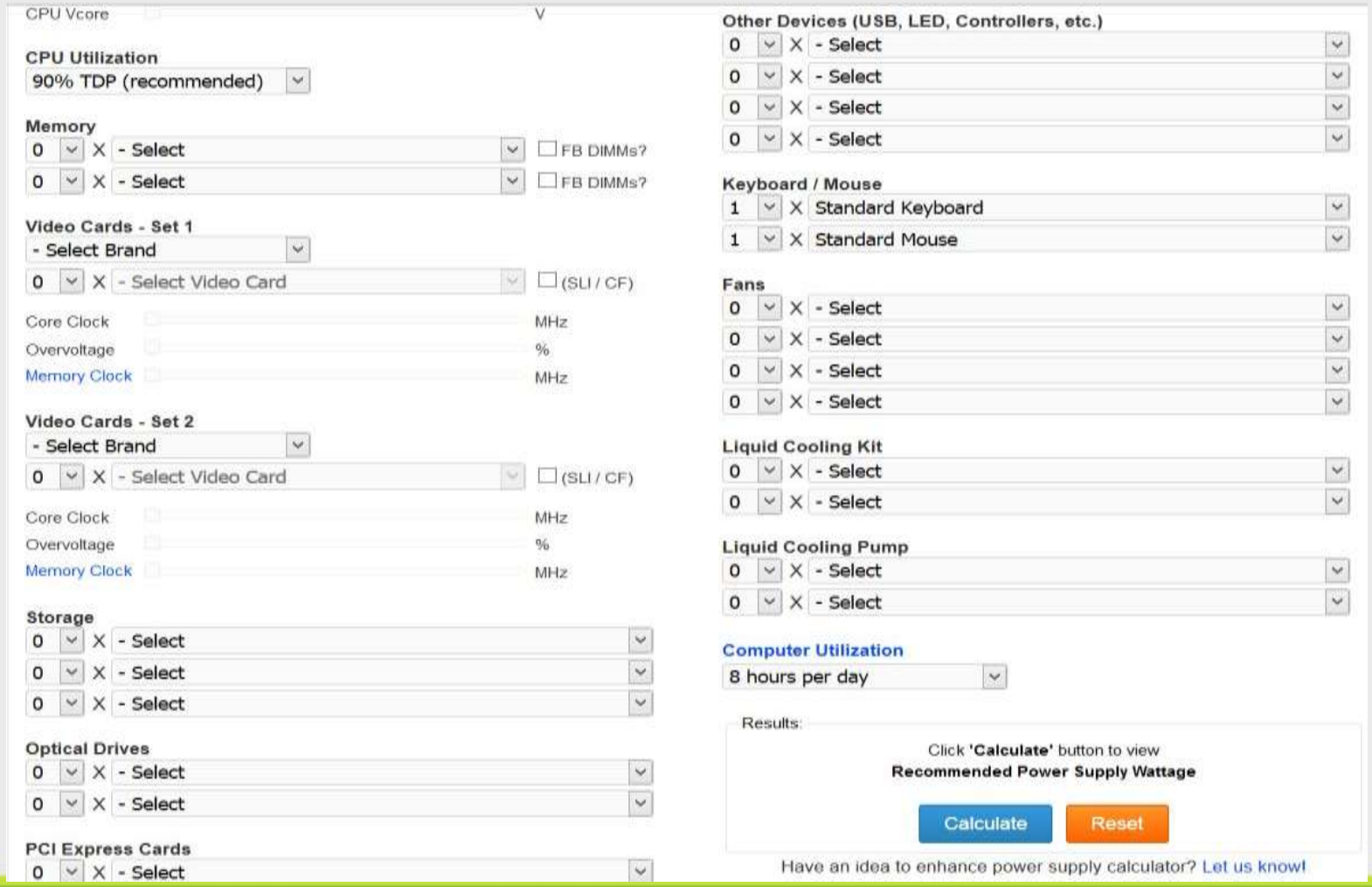


## RESULTS:

Power Consumption
Energy Consumption
Greenhouse Gas Emission
Pollution

**How we can optimize our usage to obtain *green* results ?**

# STATE OF THE ART

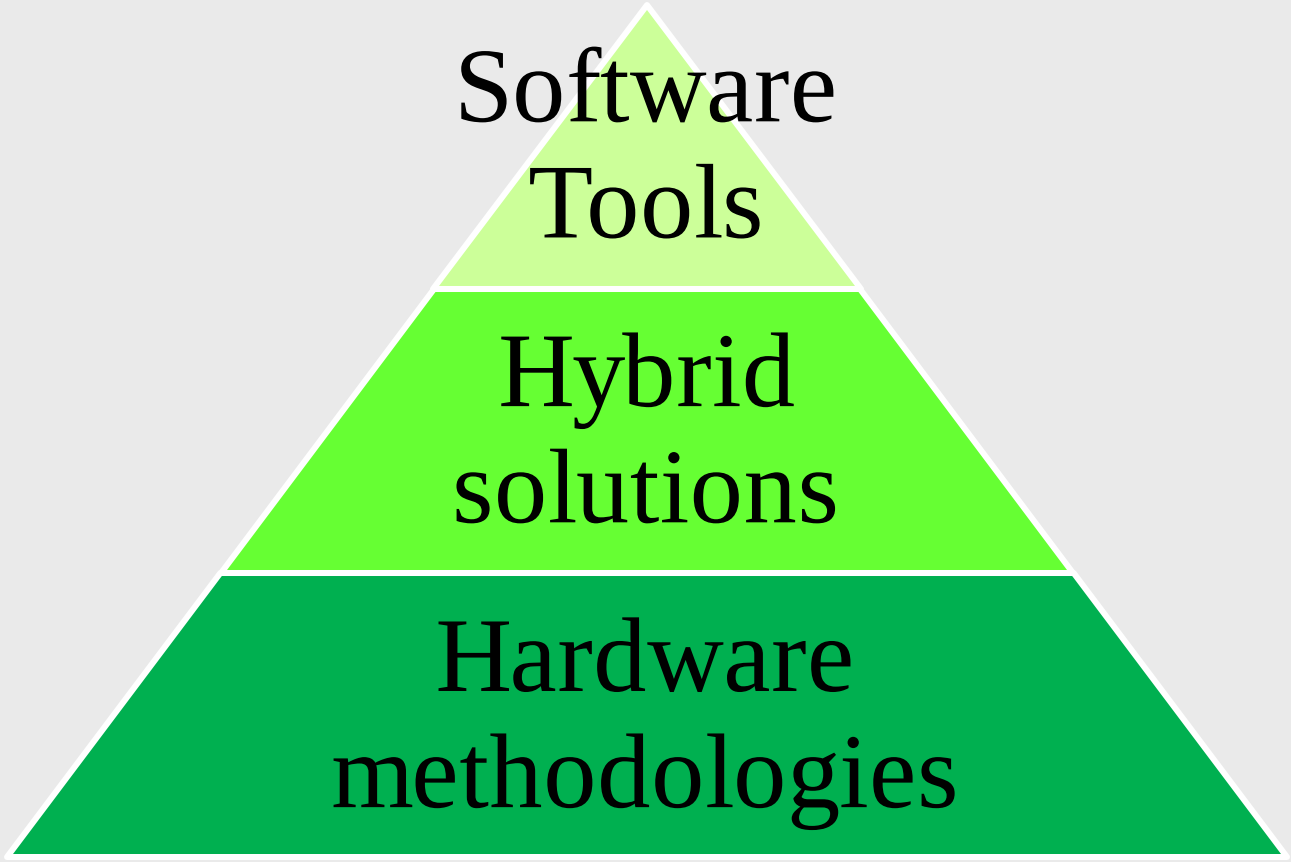


# STATE OF THE ART

POWER CONSUMING COMPONENTS

- CPU
- MEMORY
- HARD DISK
- NETWORK
- OTHERS

# STATE OF THE ART

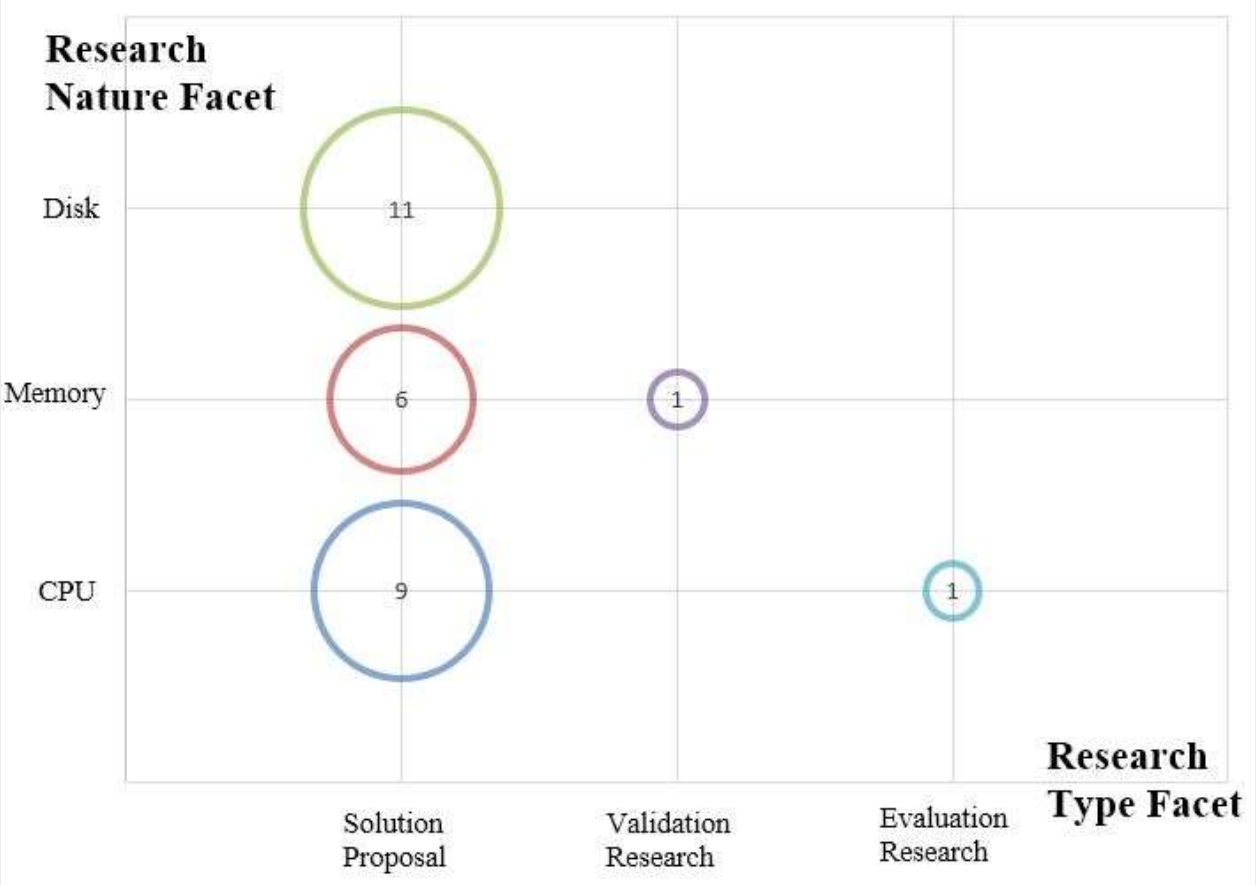


# STATE OF THE ART

- Software tools:

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

# SYSTEMATIC REVIEW



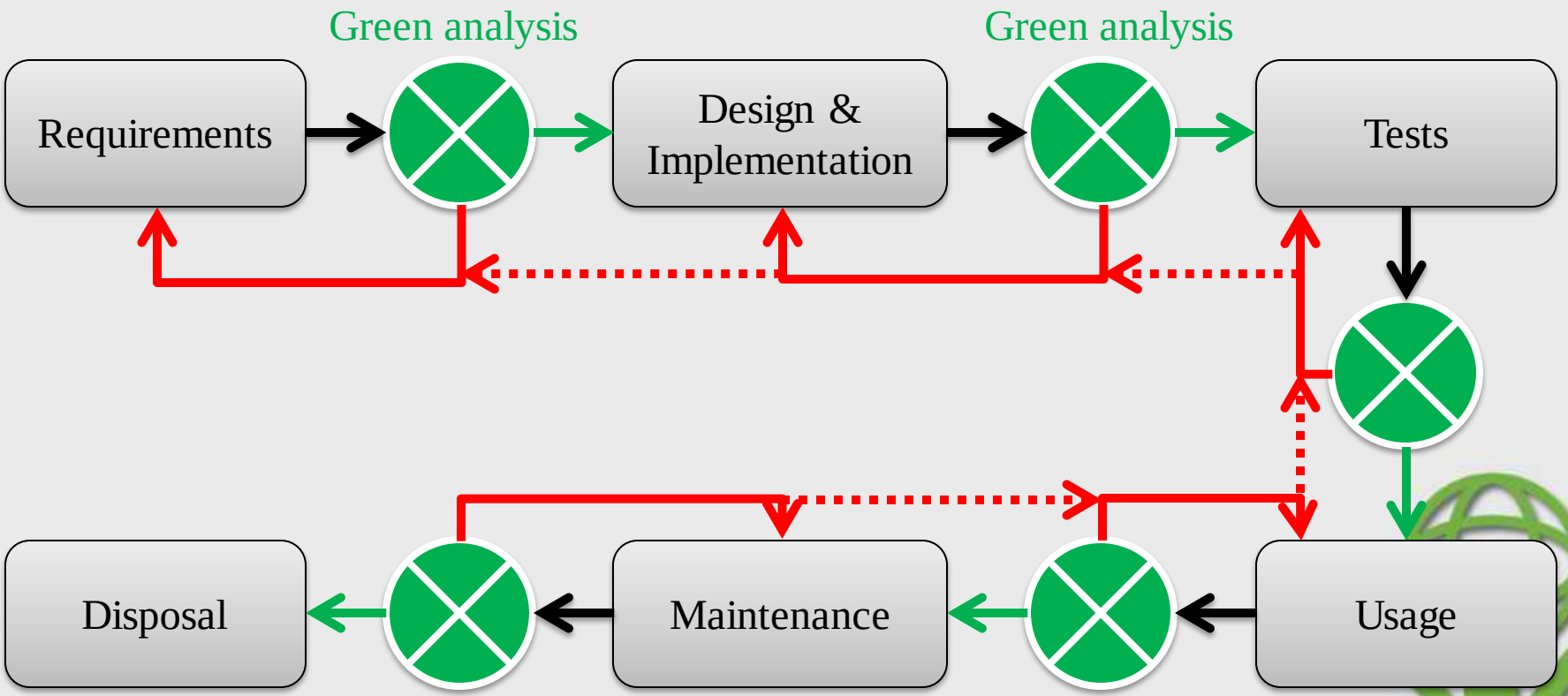
# Some software tools

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

# STATE OF THE ART

| Power saving techniques | |
|---|---|
| Transistor sizing and reordering | Dynamic power consumption of transistors is decreased |
| Network-On-Chip | Improvement over standard bus and crossbar interconnections |
| Reduce memory access | integrate a cache in the classical memory hierarchy of a modern processor |
| **DFS (Dynamic Frequency Scaling)** | Method where frequency of a microprocessor can be adapted automatically |
| **DVS (Dynamic voltage scaling)** | Lowering or increasing the supply voltage of the CPU |

# ENGINEERING PROCESS




**Towards a Green and Sustainable Software**
H. Acar, G.I. Alptekin, J.-P. Gelas and P. Ghodous.
*In Proceedings of the 22nd {ISPE} Inc. International Conference on Concurrent Engineering.*

# POWER MODEL

Dynamic and static power: $P_{Software} = P_{dynamic} + P_{static}$

Separating dynamic and static power:

$$P_{Softwre} = P_{CPU,dynamic} + P_{CPU,static} + P_{Memory,dynamic} + P_{Memory,static} + P_{Disk,dynamic} + P_{Disk,static} + P_{Network,dynamic} + P_{Network,static}$$

# POWER MODEL

**CPU:**

$$P_{CPU} = C \times V^2 \times F \quad \Rightarrow \quad P_{CPU,id} = P_{CPU} \times \frac{N_{id}}{100}$$

**Memory:**

$$P_{DRAM,dyn} = P_{Activate} + P_{Precharge} + P_{Read} + P_{Write}$$

$$\Downarrow$$

$$P_{DRAM,id} = P_{DRAM,dyn} \cdot M_{id}$$

**Hard Disk:**

$$P_{Disk} = P_{Active} + (P_{Idle} + P_{Standby} + P_{Sleep})$$

$$\Downarrow$$

$$P_{Disk,dyn} = P_{Activate} = P_{Read} + P_{Write}$$

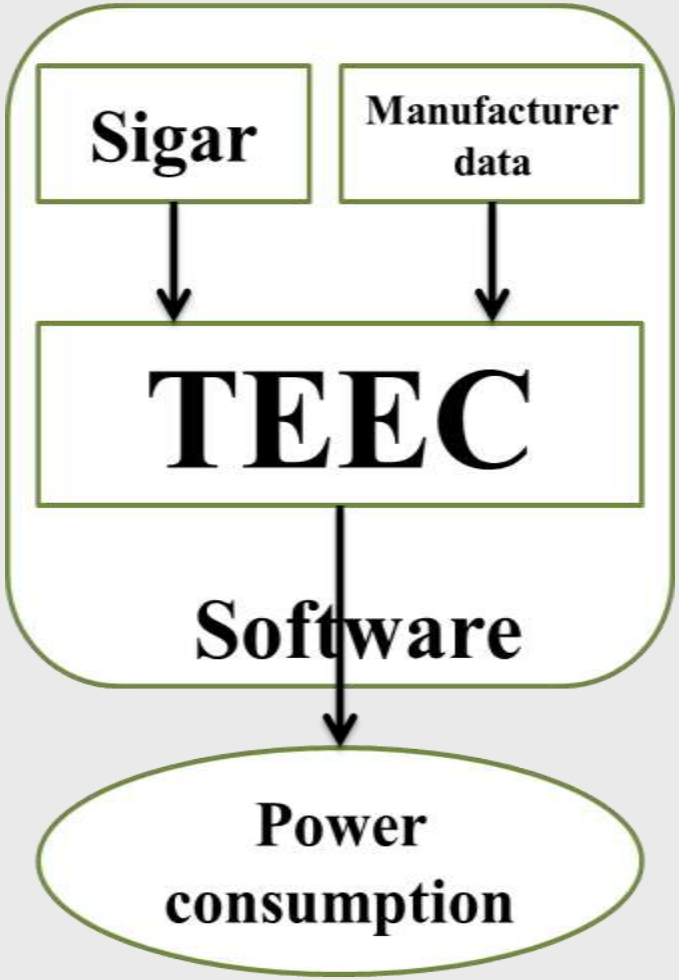
# TOP PROGRAMMING LANGUAGE


| May 2016 | May 2015 | Change | Programming Language | Ratings | Change |
|---|---|---|---|---|---|
| 1 | 1 | | Java | 20.956% | +4.09% |
| 2 | 2 | | C | 13.223% | -3.62% |
| 3 | 3 | | C++ | 6.698% | -1.18% |
| 4 | 5 | ^ | C# | 4.481% | -0.78% |
| 5 | 6 | ^ | Python | 3.789% | +0.06% |
| 6 | 9 | ^ | PHP | 2.992% | +0.27% |
| 7 | 7 | | JavaScript | 2.340% | -0.79% |
| 8 | 15 | ^^ | Ruby | 2.338% | +1.07% |
| 9 | 11 | ^ | Perl | 2.326% | +0.51% |
| 10 | 8 | v | Visual Basic .NET | 2.325% | -0.64% |
| 11 | 13 | ^ | Delphi/Object Pascal | 2.008% | +0.71% |
| 12 | 22 | ^^ | Assembly language | 1.883% | +1.12% |
| 13 | 10 | v | Visual Basic | 1.828% | -0.07% |
| 14 | 4 | vv | Objective-C | 1.597% | -3.80% |

IEEE Spectrum 2015
http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2015

TIOBE INDEX
http://www.tiobe.com/tiobe_index

# Tool for Estimating Energy Consumption

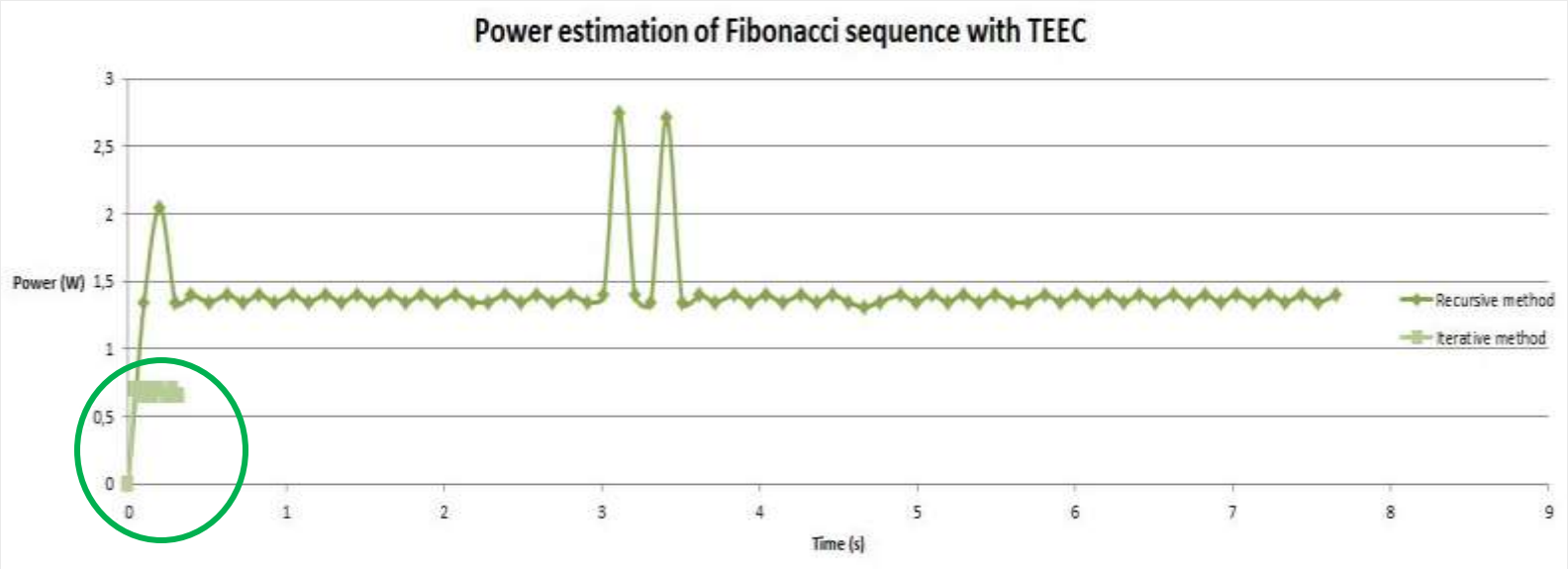


# EXPERIMENTS

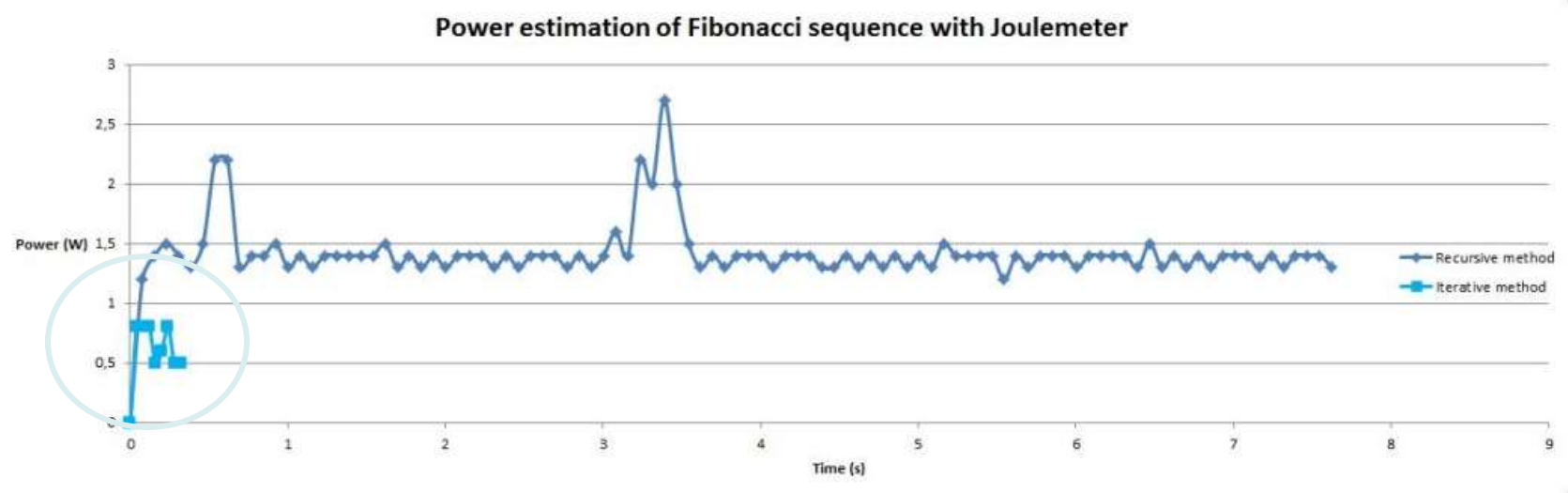
Fibonacci sequence is implemented in Java of 2 ways:

- Recursively: 45 terms are calculed.
- Iteratively: calculation of 5,000 values.

TEEC & Joulemeter results are compared.



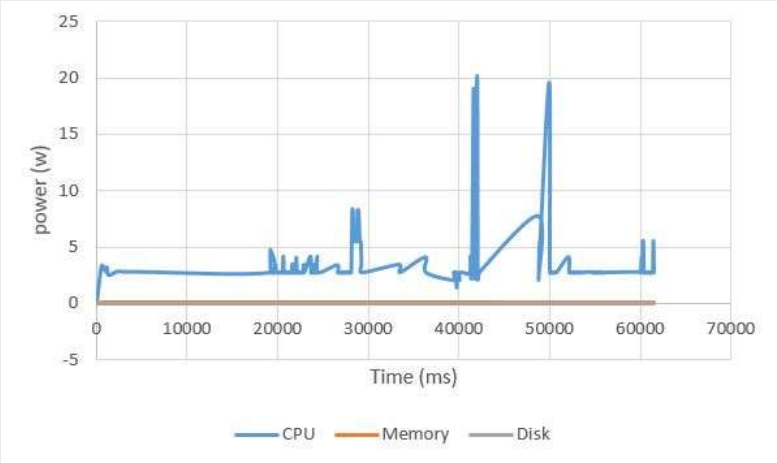
# EXPERIMENTS



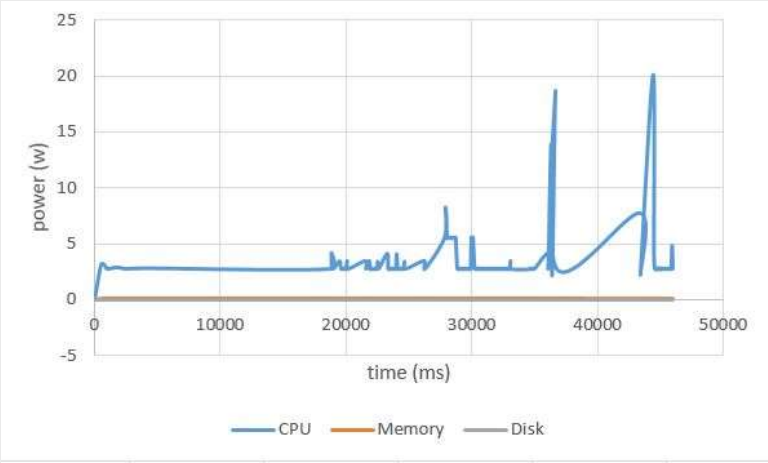

Quite similar results between TEEC and Joulemeter.

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

# EXPERIMENTS



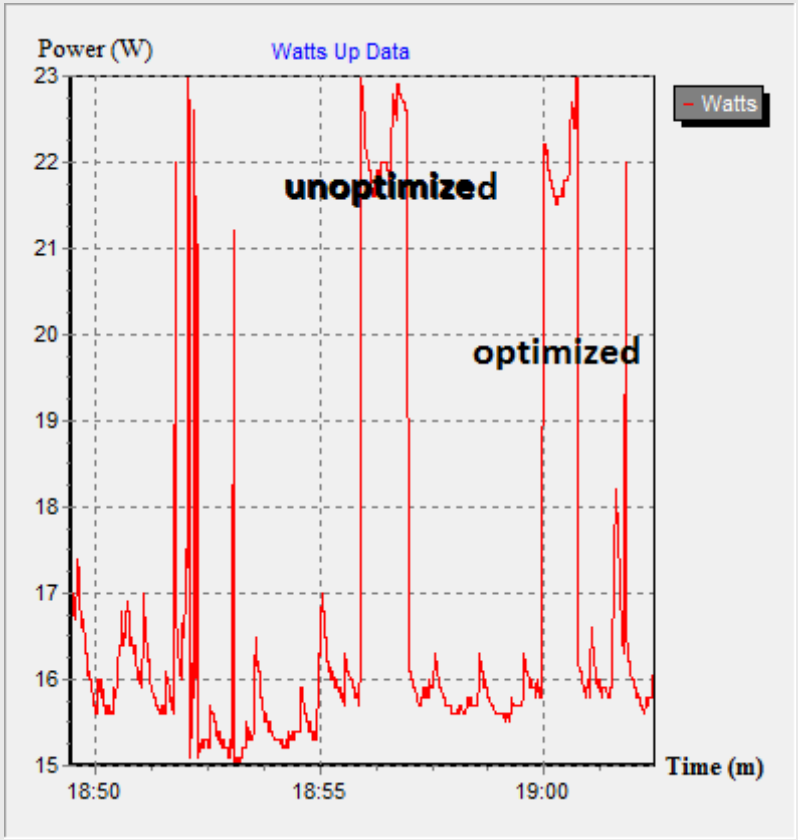

Unoptimized code




Optimized code

| Functions | Unoptimized | Optimized |
|---|---|---|
| | Time (ms) | |
| Locality of reference | 18140 | 17219 |
| Compare array to array list | 22047 | 17297 |
| Compare integer list loop | 7734 | 7391 |
| Char array StringBuilder | 11235 | 2421 |
| Binary search | 2250 | 438 |

# EXPERIMENTS

# EXPERIMENTS

**Nested Loops**

C

Unoptimized

```
char* fn(int N) {
  int i = 0;
  char* v = (char*)malloc(N  * 2);
  while (i < N) {
    int j = 0;
    v[i] = 0;
                i++;
    while (j < N) {
      v[i] += v[j + N];
                j++;
    }
  }
return v;
}
```

Optimized

```
char* fn2(int N) {
  int i = 0;
  char* v = (char*)malloc(N * 2);
  int tmp;
  while (i < N) {
    int j = 0;
    v[i] = 0;
                i++;
                tmp = 0;
    while (j < N) {
      tmp += v[j + N];
                j++;
    }
                v[i] = tmp;
  }
return v;
}
```

Java

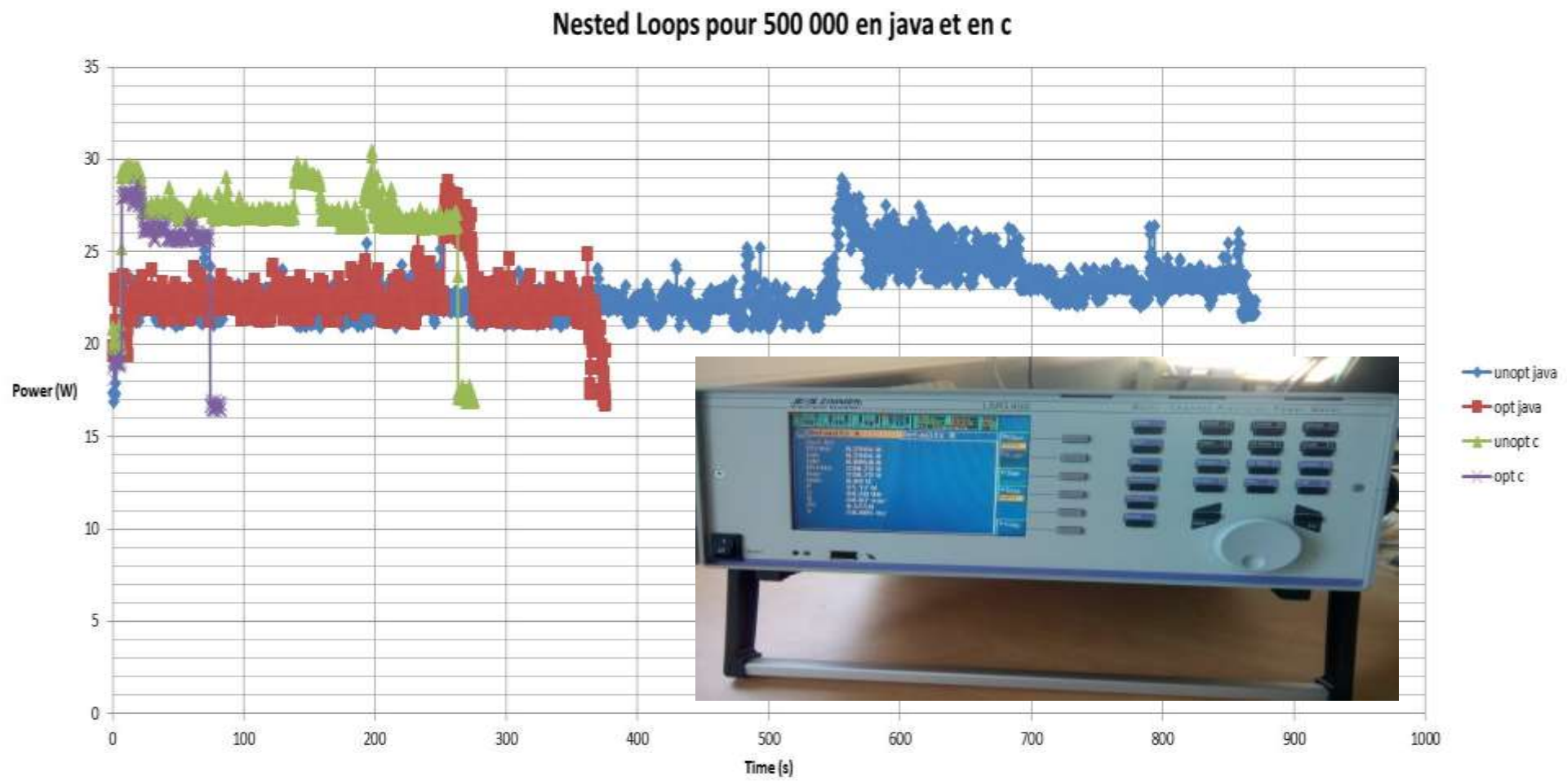
Unoptimized

```
public static char[] fn(int N)
{
int i = 0;
char[] v = new char[N*2];
while (i < N) {
int j = 0;
v[i] = 0;
i++;
while (j < N) {
      v[i] += v[j + N];
for (v + j + N)
                j++;
                }
                }
               return v;
               }
```

Optimized

```
public static char[] fn2(int
N) {
int i = 0;
char[] v = new char[N*2];
int tmp;
while (i < N) {
int j = 0;
v[i] = 0;
i++;
tmp = 0;
while (j < N) {
tmp += v[j + N];
                j++;
}
v[i] = (char) tmp;
                }
              return v;
              }
```

# EXPERIMENTS



# CONCLUSIONS and FUTURE WORKS

- 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 programing best practises (with our expertise).
- Focus on Cloud computing (web services)

# PUBLICATIONS

**The impact of source code in software on power consumption**
H. Acar, G.I. Alptekin, J.-P. Gelas and P. Ghodous.
*International Journal of Electronic Business Management,*

**TEEC : Improving power consumption estimation of software**
H. Acar, G.I. Alptekin, J.-P. Gelas and P. Ghodous.
*In Proceedings of the 30th International Conference on Environmental informatics.*
Berlin, Germany, September 14-16, 2016.

**Beyond CPU: Considering Memory Power Consumption of Software**
H. Acar, G.I. Alptekin, J.-P. Gelas and P. Ghodous.
*In Proceedings of the 5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS).*
Rome, Italy, April 23-25, 2016.

**A Green approach to save energy consumed by software**
H. Acar, G.I. Alptekin, J.-P. Gelas and P. Ghodous.
*In Proceedings of the 3rd International Conference on ICT for Sustainability (ICT4S).*
Copenhagen, Denmark, September 7-9, 2015.

**Towards a Green and Sustainable Software**
H. Acar, G.I. Alptekin, J.-P. Gelas and P. Ghodous.
*In Proceedings of the 22nd {ISPE} Inc. International Conference on Concurrent Engineering.*
Delft, The Netherlands, July 20-23, 2015, pp. 471-480