

What Are Your Programming Language's Energy-Delay Implications?



**Stefanos
Georgiou**



**Maria
Kechagia**



**Panos
Louridas**



**Diomidis
Spinellis**



PERCCOM summer school 2018
Lappeenranta, Finland

Introduction



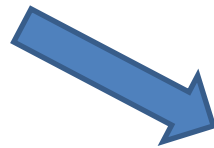
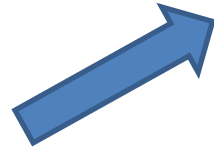
15%



2.3%

Motivation

Ruby
C# Rust Go
Python Java
Visual Basic.NET
R JavaScript
Perl Swift
C++ C PHP



Research Gap

Existing Work

- Energy/Performance investigation
- Examination of running applications
- Experiments on a single platform

Ours

- Energy Delay Product measurements
- Comparison of 14 languages
- Analysis of small programming tasks
- Experiments on 3 platforms

Energy Delay Product

$$\text{EDP} = E \times T^w$$

$w = 1$

$w = 2$

$w = 3$



Research Questions



RQ1: Which programming languages are the most EDP efficient and inefficient for particular task?

RQ2: Which type of programming languages are, on average, more EDP efficient and inefficient for each of our selected platforms?

RQ3: How much does the EDP of each programming language differ among the selected platforms?

Programming Languages



- Monthly index rating based on languages popularity
- Data retrieved from 25 search engines using search query
- Programming Languages criteria:
 1. The least, 5000 hits on Google
 2. Turing complete
 3. Wikipedia page

Selected Programming Languages

Categories	Programming Languages	Compilers & Interpreters			Optimization
		Embedded	Laptop	Server	Flags
Compiled	C	6.3.0	6.4.1	6.4.1	-O3
	C++	6.3.0	6.4.1	6.4.1	-O3
	Go	1.4.3	1.7.6	1.7.6	--
	Rust	1.20.0	1.18.0	1.21.0	-O
	Swift	3.1.1	3.0.2	3.0.2	-O
Semi-Compiled	C#	4.6.2	4.6.2	4.6.2	-optimize+
	VB.NET	4.6.2	4.6.2	4.6.2	-optimize+
	Java	1.8.0	1.8.0	1.8.0	--
Interpreted	JavaScript	9.0.4	8.9.3	8.9.3	--
	Perl	5.24.1	5.24.1	5.24.1	--
	PHP	5.6.30	7.0.25	7.0.25	--
	Python	2.7.23	2.7.13	2.7.13	--
	R	3.3.3	3.4.2	3.4.2	--
	Ruby	2.4.2	2.4.1	2.4.1	--

Rosetta Code Repository



ROSETTACODE.ORG

[Create account](#) [Log in](#)



[Community](#) ▾

[Explore](#) ▾

[Main page](#) [Discussion](#) [View source](#) [History](#)

Rosetta Code

Rosetta Code is a [programming chrestomathy](#) site. The idea is to present solutions to the same task in as many different languages as possible, to demonstrate how languages are similar and different, and to aid a person with a grounding in one approach to a problem in learning another. Rosetta Code currently has 872 tasks, 210 [draft tasks](#), and is aware of 682 [languages](#), though we do not (and cannot) have solutions to every task in every language.

Contents [\[hide\]](#)

1 [Places to start](#)

1.1 [Recently-Updated Tasks](#)

1.2 [Ways to Contribute](#)

1.3 [Recent Discussions](#)

Data Set

Categories	Tasks
Arithmetic	<i>exponentiation-operator and numerical-integration</i>
Compression	<i>huffman-coding and lzw-compression</i>
Concurrent	<i>concurrency-computing and synchronous-concurrency</i>
Data structures	<i>array-concatenation and json</i>
File handling	<i>file-input-output</i>
Recursion	<i>Factorial, ackermann-function and palindrome-detection</i>
Regular Expression	<i>regular expression</i>
Sorting algorithms	<i>selection, insertion, merge, bubble, and quick</i>
String manipulation	<i>url-encoding/decoding</i>
Object-Oriented	<i>inheritance single/multiple, class, and call-an-object-method</i>
Functional	<i>function-composition</i>

Amendments

- ✓ Wrapped tasks in a loop (more than a sec.)
- ✓ Wrote from scratch missing implementations
- ✓ Removed multiple implementations of the same languages
- ✓ Made tasks execution loop dependent and added *volatile* variables
- ✓ 2000 approx. BASH script to manage data set

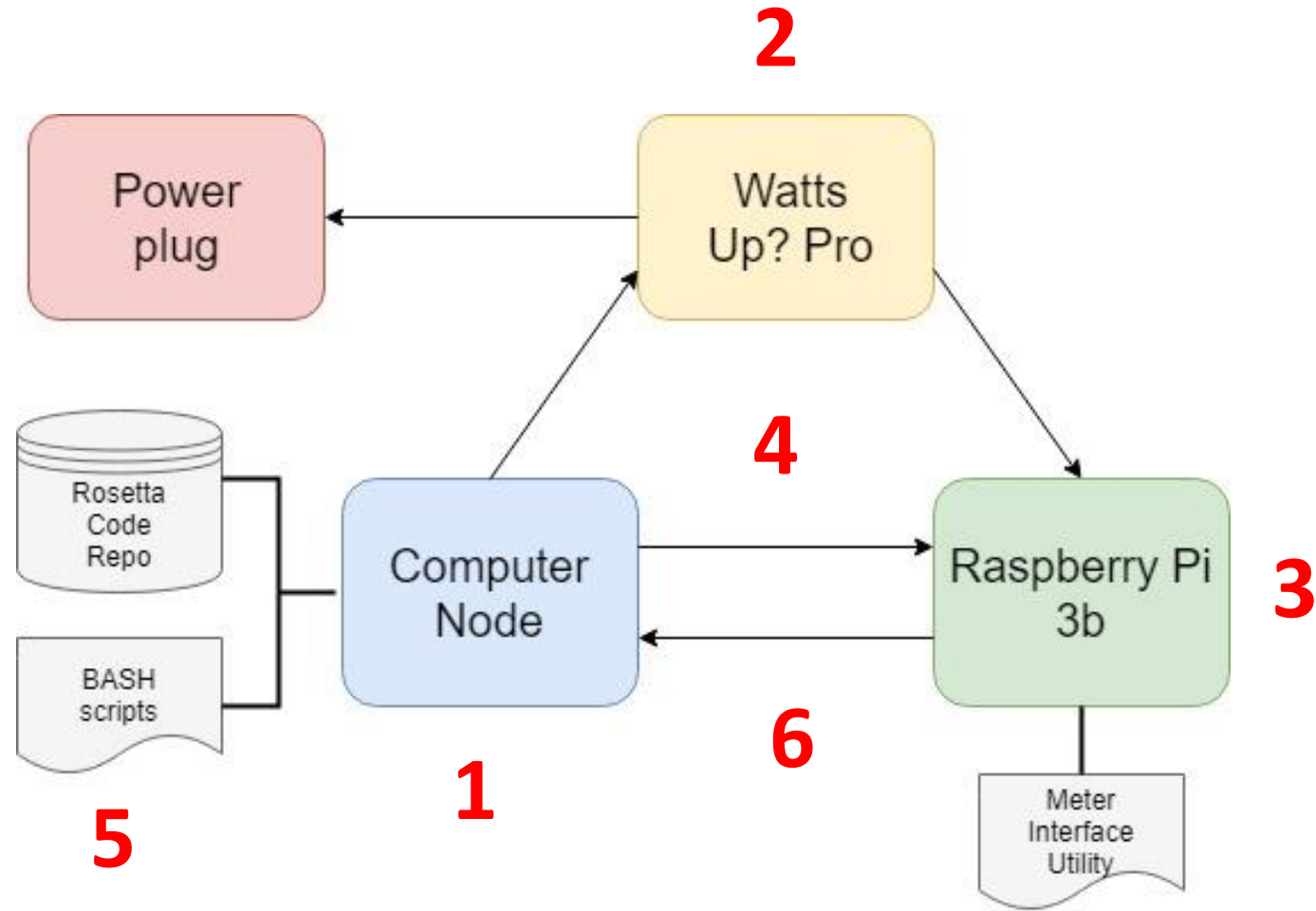
Automated Script for the experiment

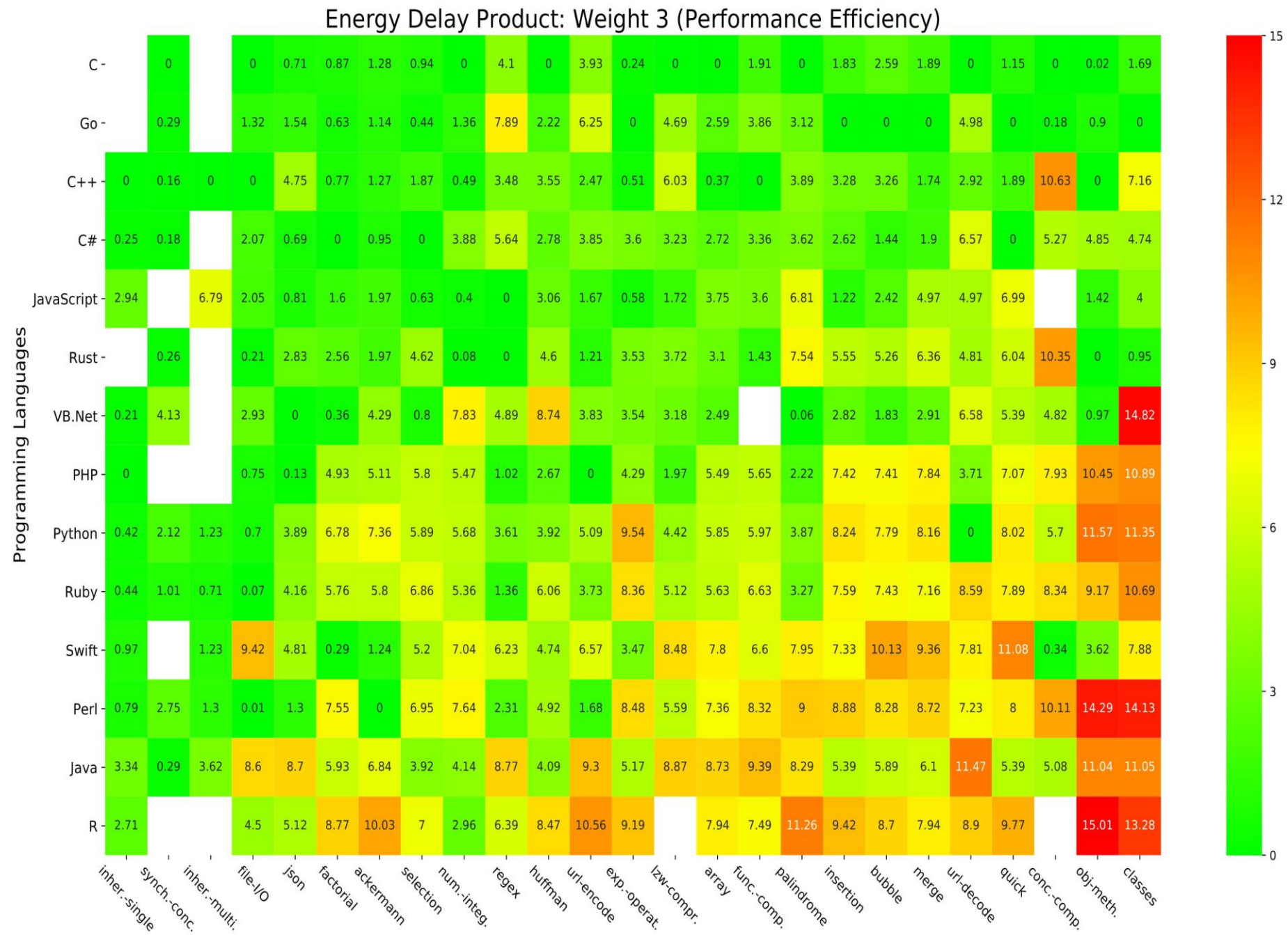


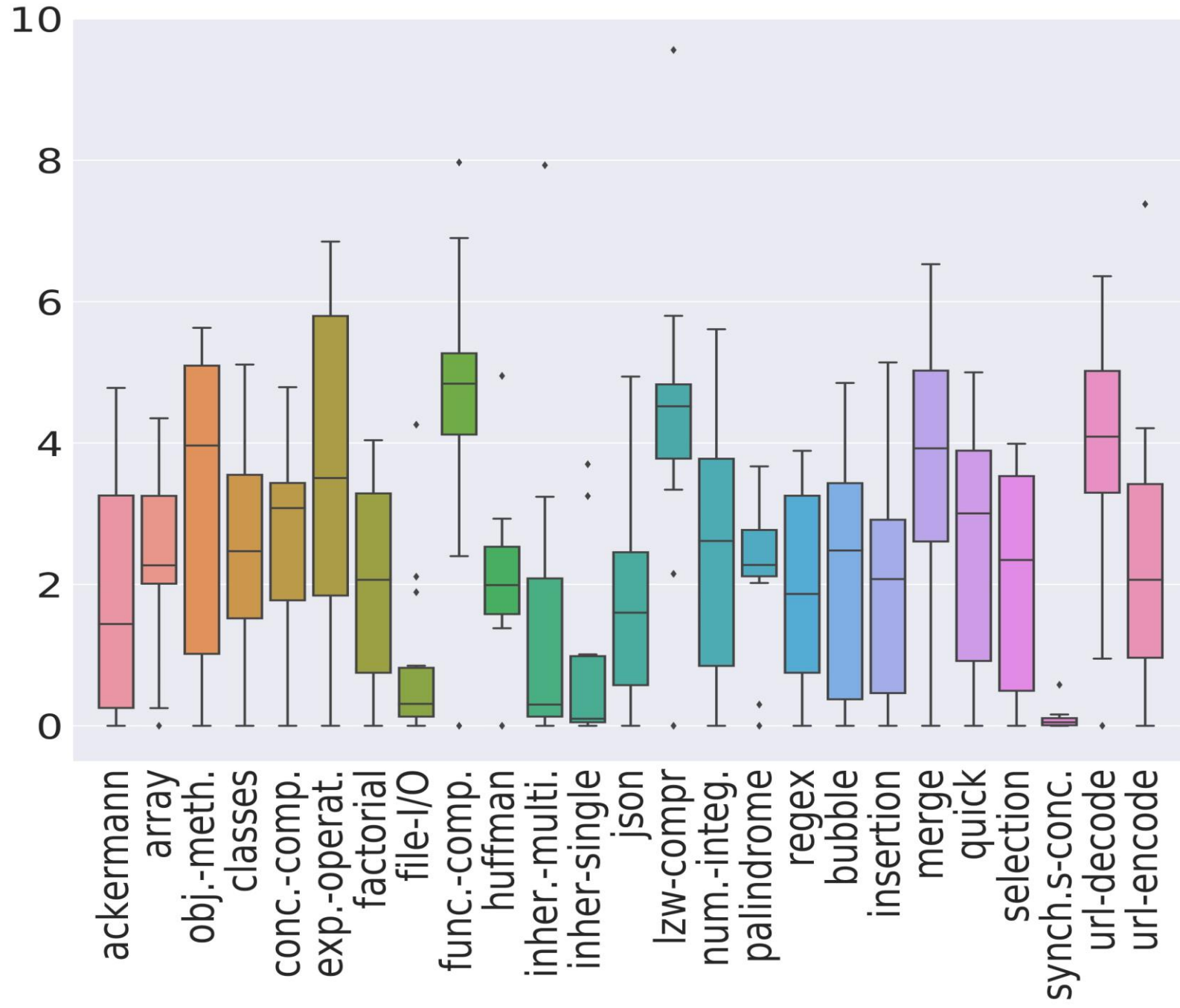
Experimental Platform



Execution Process





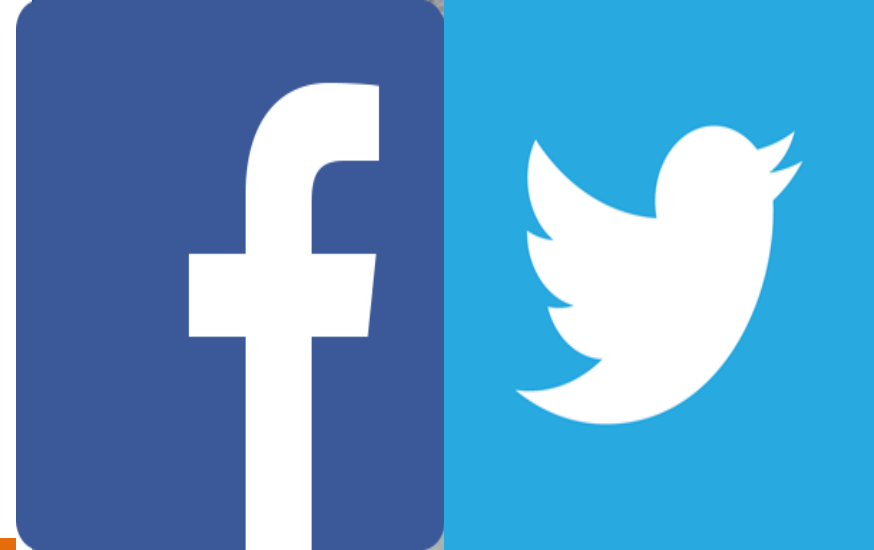


RQ1. Which programming languages are the most EDP efficient and inefficient for particular tasks?

Task categories	Most efficient/inefficient	
Arithmetic	C/R, VB.NET	Computation speed
Compression	C/VB.NET, Java	
Concurrent	C/VB.NET, Perl	OpenMP + libco
File Handling	Rust/VB.NET	Sys Calls + I/O buffers
Regular Expressions	JavaScript/Java	V8 engine + RegEx
Sorting	Go/Swift, R	Memory + Speed
Functional	C++/Swift, Perl	Meta-programming

Possible Applications

Task categories	Most efficient/inefficient
Arithmetic	C/R, VB.NET
Compression	C/VB.NET, Java
Concurrent	C/VB.NET, Perl
File Handling	Rust/VB.NET, Swift
Regular Expressions	JavaScript/Java
Sorting	Go/Swift, R
Functional	C++/Swift, Perl



RQ2. Which types of programming languages are, on average, more EDP efficient and inefficient for each of the selected platforms?

Rank	Embedded	Laptop	Server
1	C	C	C
2	C++	Go	Go
3	Go	C++	C++
4	Rust	JavaScript	C#
5	JavaScript	Rust	JavaScript
6	C#	C#	Rust
7	VB.NET	VB.NET	VB.NET
8	PHP	PHP	PHP
9	Ruby	Ruby	Python
10	Python	Swift	Ruby
11	Perl	Python	Swift
12	Java	Perl	Perl
13	Swift	Java	Java
14	R	R	R

RQ3. How much does the EDP of each programming language differ among the selected platforms?

- *Hypothesis H_0 : A programming language's average EDP, does not have a statistically important difference between the measurement platforms.*

There is a significant difference between the average EDP, in some case, of the embedded and laptop platforms.

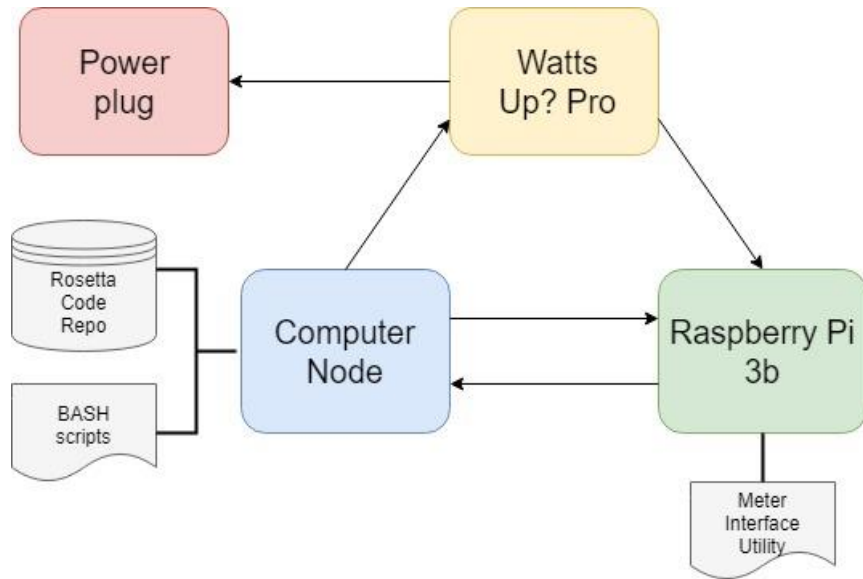
Conclusions

- Proper selection of programming languages implementation offer EDP efficiency.
- Compiled are more EDP efficient than semi-compiled and interpreted, in most of cases.
- There is no statistical difference among the same implementations on different platforms.

Future work

- More and more realistic test inputs
- More categories e.g., networking
- Different CPU clock frequencies

Take-aways



Thank you!

- a customized and extended data set that can be used as a benchmark for similar studies
- a set of tools for measuring the EDP of various programming languages and tasks
- an empirical study on programming language EDP implications,
- a programming language-based ranking catalogue, in the form of heat maps, where developers can find which programming language to pick for particular tasks

Contact details: sgeorgiou@aueb.gr

sgeorgiou@singularlogic.eu

<https://github.com/stefanos1316>

References

- <https://mashable.com/2017/12/12/qualcomm-snapdragon-845-phone-innovations/>
- <http://emsys-design.com/data-centers.php>
- <https://www.ziprecruiter.com/blog/software-developer-job-description-sample-template/>
- <http://www.lincoder.com/using-strace-to-trace-system-calls-and-signals-in-linux/>
- [https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))

EDP how it works

Consider the following case where we have applications A1 and A2.

A1 has 2 seconds of execution time and 5 joules of energy consumption.

A2 has 3 seconds of execution time and 2 joules of energy consumption.

By increasing their weights ($E \times T^w$ where $w = 1, 2, 3$) we can see the following:

$w = 1$: A1 2s 5j \Rightarrow 10 EDP

A2 3s 2j \Rightarrow 6 EDP

$w = 2$: A1 4s 5j \Rightarrow 20 EDP

A2 9s 2j \Rightarrow 18 EDP

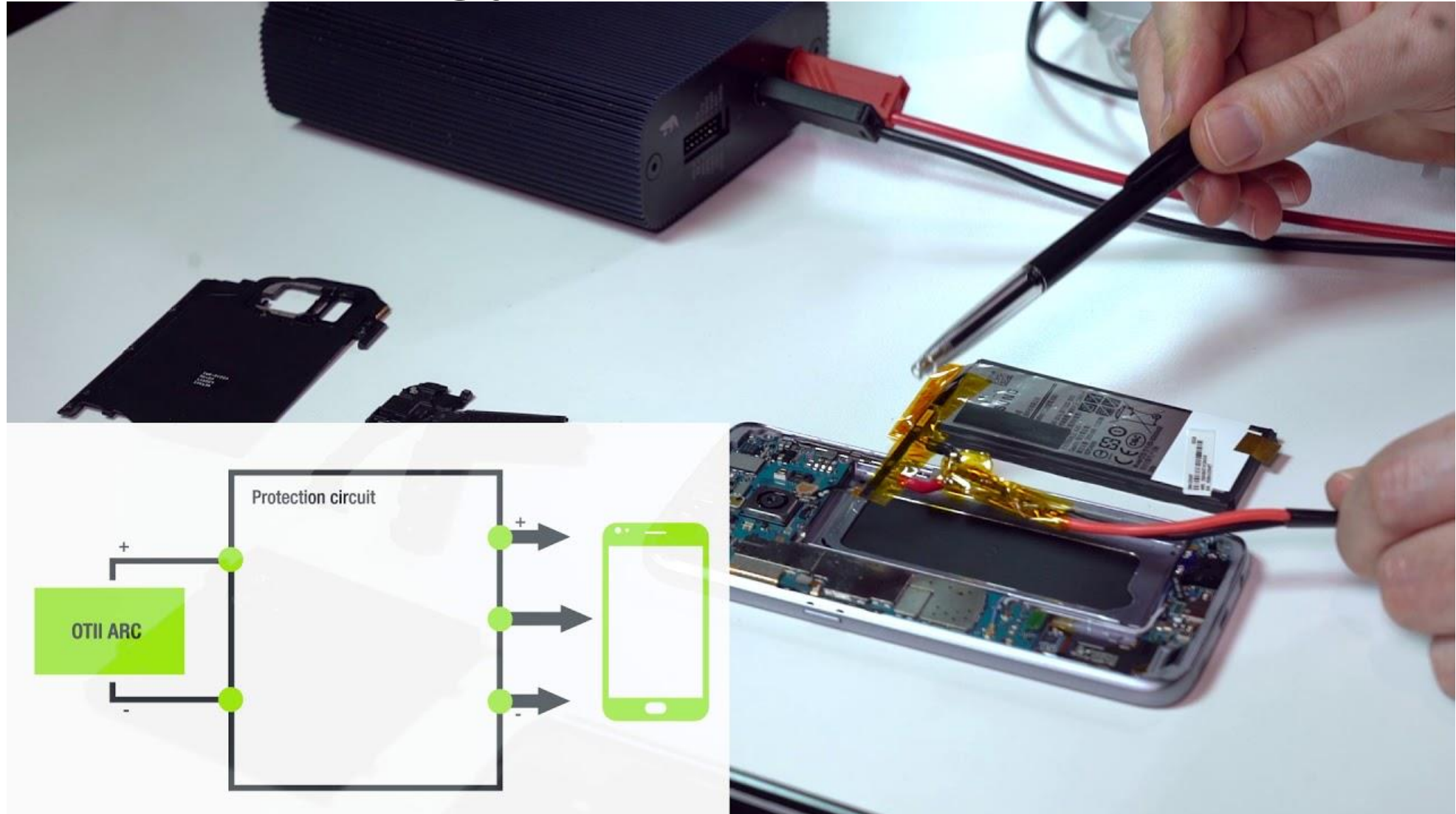
$w = 3$: A1 8s 5j \Rightarrow 40 EDP

A2 27s 2j \Rightarrow 54 EDP

Get ready, questions are coming!!!



Which are the different venues for measuring software's energy consumption?



What are their pitfalls of each approach?



Tools for measuring energy and power consumption inside a Linux environment.

[Edit](#)[energy-monitor](#)[energy-consumption](#)[Manage topics](#)

📄 84 commits

🌿 1 branch

📦 0 releases

👤 2 contributors

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download ▾](#)

stefanos1316 Minor changes in README

Latest commit 99c176e 8 days ago

📁 applications	Added resource usage feature	9 months ago
📁 img	Added resource usage feature	9 months ago
📁 jRAPL @ ecd45b3	add jRAPL	a year ago
📁 jalen @ 2c1834a	Adding submodules	a year ago
📁 jolinar @ 12f4e3a	Adding submodules	a year ago
📁 likwid @ 4e1a04e	Adding likwid	18 days ago
📁 papi @ db6bcc5	adding minor changes	8 days ago
📁 powerapi-cli	Added resource usage feature	9 months ago
📁 scripts	adding minor changes	8 days ago
📁 uarch-configure @ 788f97d	Adding submodule rapl	a year ago
📁 watts-up @ 9bee506	Added watts up	10 months ago
📄 .gitignore	adding minor changes	8 days ago
📄 .gitmodules	adding minor changes	8 days ago