EFOP-3.6.2-16-2017-00013

European Union

SELECTED TOPICS IN GREEN SOFTWARE ENGINEERING

CSABA SZABÓ

FAC. OF EL. ENG. & INF.
TECHNICAL UNIVERSITY OF KOŠICE, SLOVAKIA

SZÉCHENYI 2020

HUNGARIAN GOVERNMENT

European Union
European Social Fund

INVESTING IN YOUR FUTURE

CONTENTS

- What is green software (engineering)?
- Hardware + software + user = ?
- Existing tools
- Selected topics
- Examples on ST
- Conclusions
- Future of GSE

GREEN SOFTWARE ENGINEERING

Green software engineering is a branch of software engineering focusing on energy aspects of software. Please note that software plays here the role of the process, which one's energy consumption can be expressed through the energy consumption of all hardware parts that were used in any way by the examined process. Therefore, when evaluating software greenness, we always examine the usage load on hardware parts during software execution time.

THE ROLE OF HARDWARE

- Display
- Networking (Wi-Fi, Bluetooth), radio
- Processor
- Memory
- Disks
- Battery
- Sensors

THE ROLE OF SOFTWARE

- Operating system (difference between Windows, Linux, macOS, Android, iOS)
- Working software
- Computer games
- Application systems
- Databases

THE ROLE OF THE USER

- The user “drives” the software
- Needs individual training (unlike HW/SW)

- Does (s)he receive it? Where?

- Repairing bad configuration is often done by buying a new device…

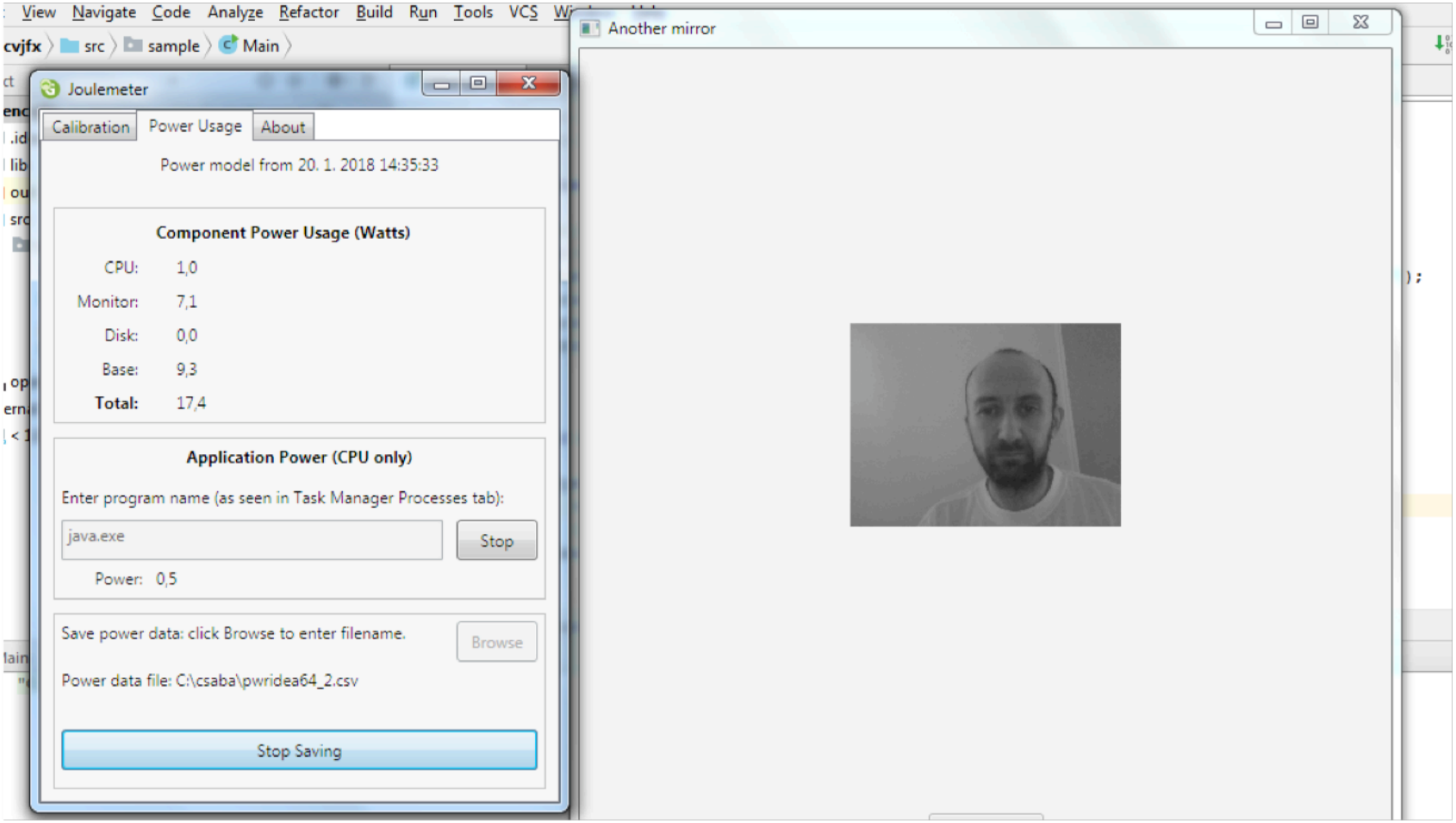
EXISTING TOOLS

- OS-level (top, activity/process monitor, energy saver)
- For developers (Intel RAPL, GreenDroid, Microsoft Joulemeter, Android Studio, Xcode Debug Navigator, Instruments)
- User-space (powermetrics, powertop, Trepn profiler, Intel Power Gadget, Greenify, coffeinate)

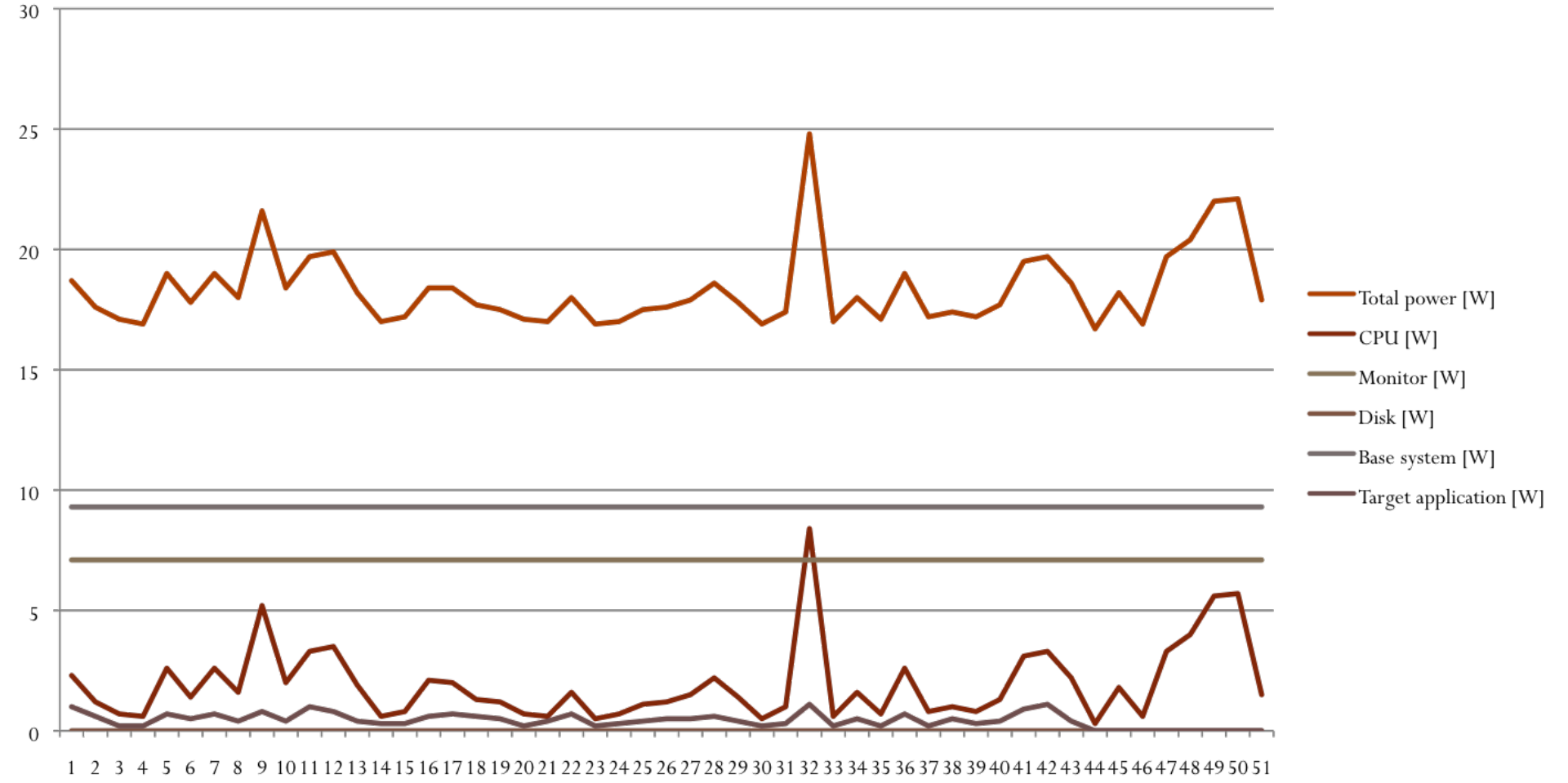
TOPIC SELECTION

- In-development measurement
- Post-development measurement
- Measuring third-party SW
- How does the energy profile of the software development process look like
- Distributed applications measurement

EXAMPLE 1: IN-DEVELOPMENT MEASURING (INTELLECTUAL OUTPUT 1,2 OF THE ERASMUS+ PROJECT 2017-1-SK01-KA203-035402)



EXAMPLE 1: IN-DEVELOPMENT MEASURING (INTELLECTUAL OUTPUT 1,2 OF THE ERASMUS+ PROJECT 2017-1-SK01-KA203-035402)



EXAMPLE 2: POST-DEVELOPMENT MEASURING

- As part of black-box testing
- Acceptance criteria
- Performed manually or automatically

EXAMPLE 2:
POST-DEVELOPMENT MEASURING

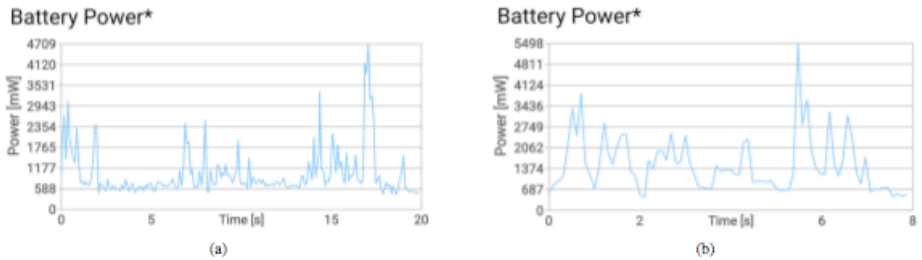


Fig. 5. Comparing (a) no image caching image caching vs (b) image caching concerning energy consumption

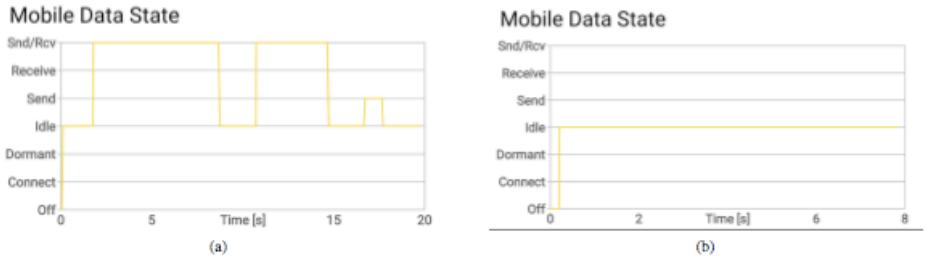


Fig. 6. Comparing (a) no image caching vs (b) image caching concerning network traffic

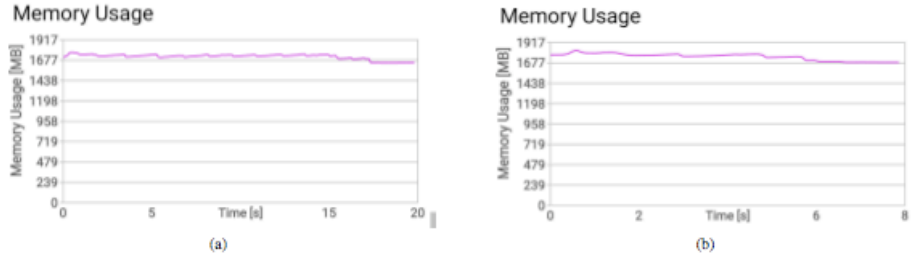
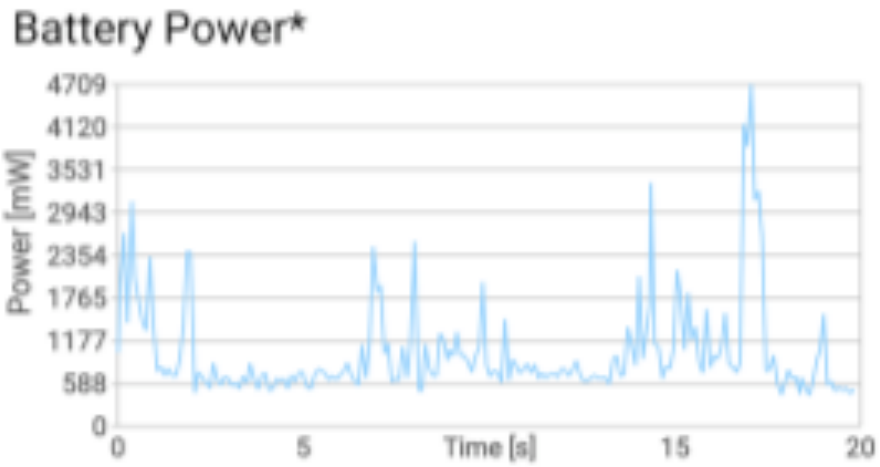
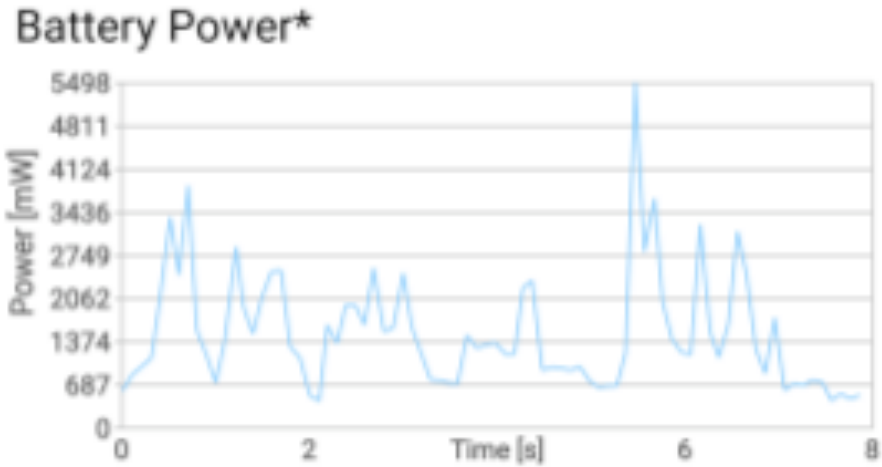


Fig. 7. Comparing (a) no image caching vs (b)image caching concerning memory usage

EXAMPLE 2:
POST-DEVELOPMENT MEASURING



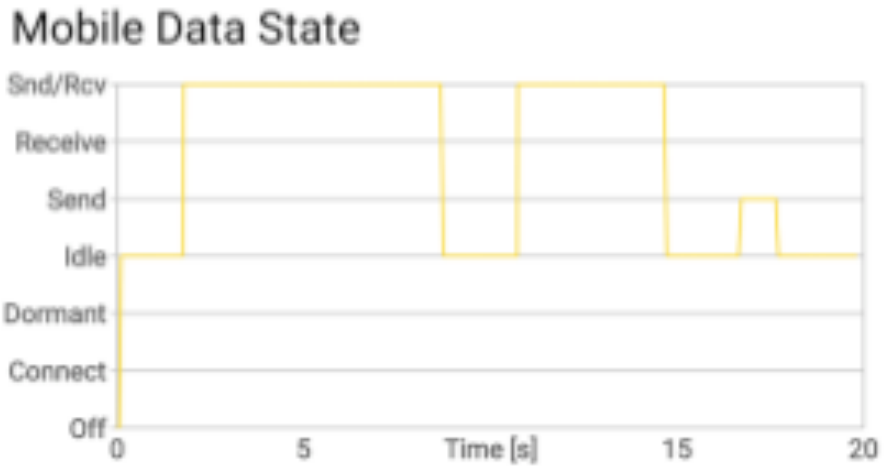
(a)



(b)

Fig. 5. Comparing (a) no image caching image caching vs (b) image caching concerning energy consumption

EXAMPLE 2:
POST-DEVELOPMENT MEASURING



(a)

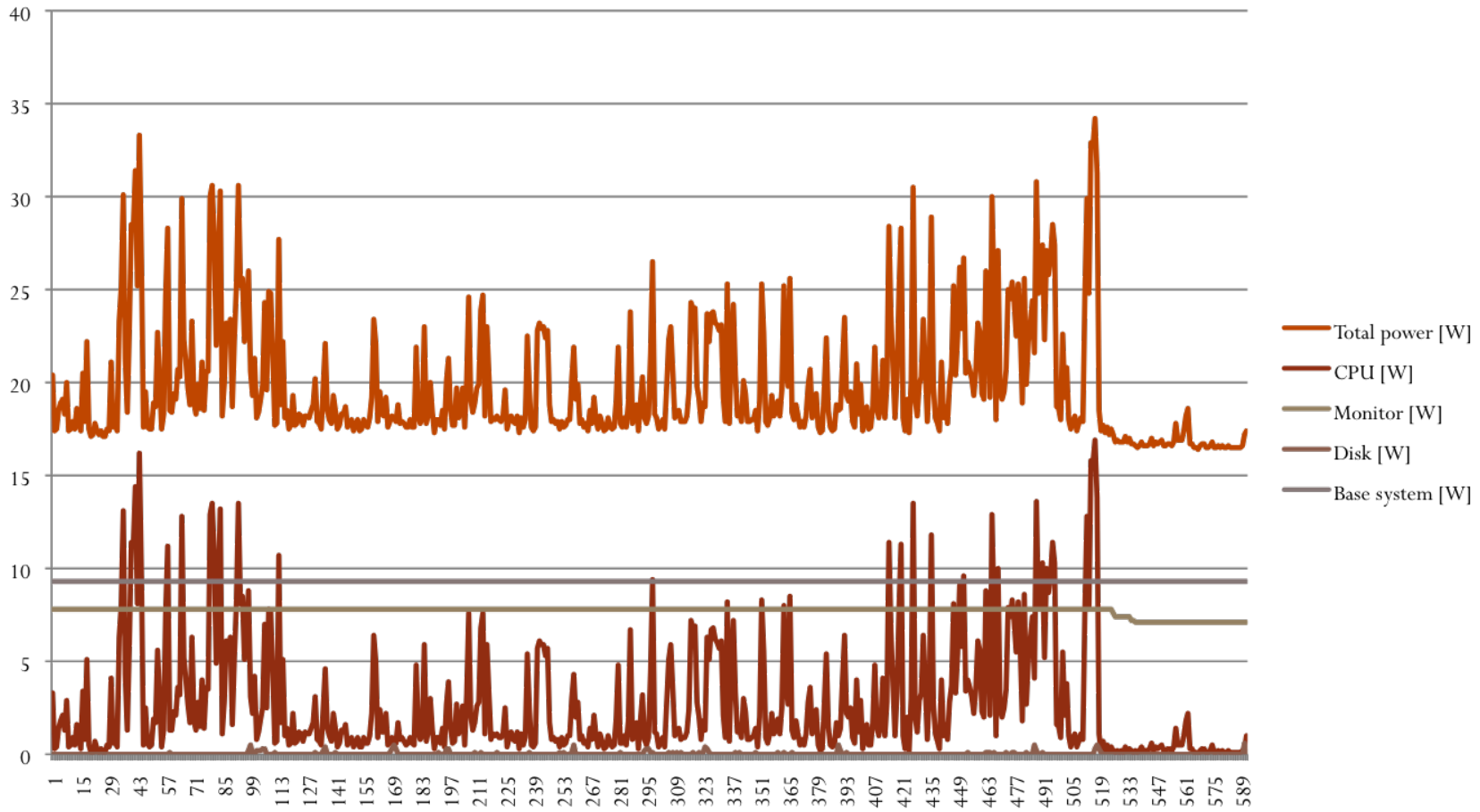
(b)

Fig. 6. Comparing (a) no image caching vs (b) image caching concerning network traffic

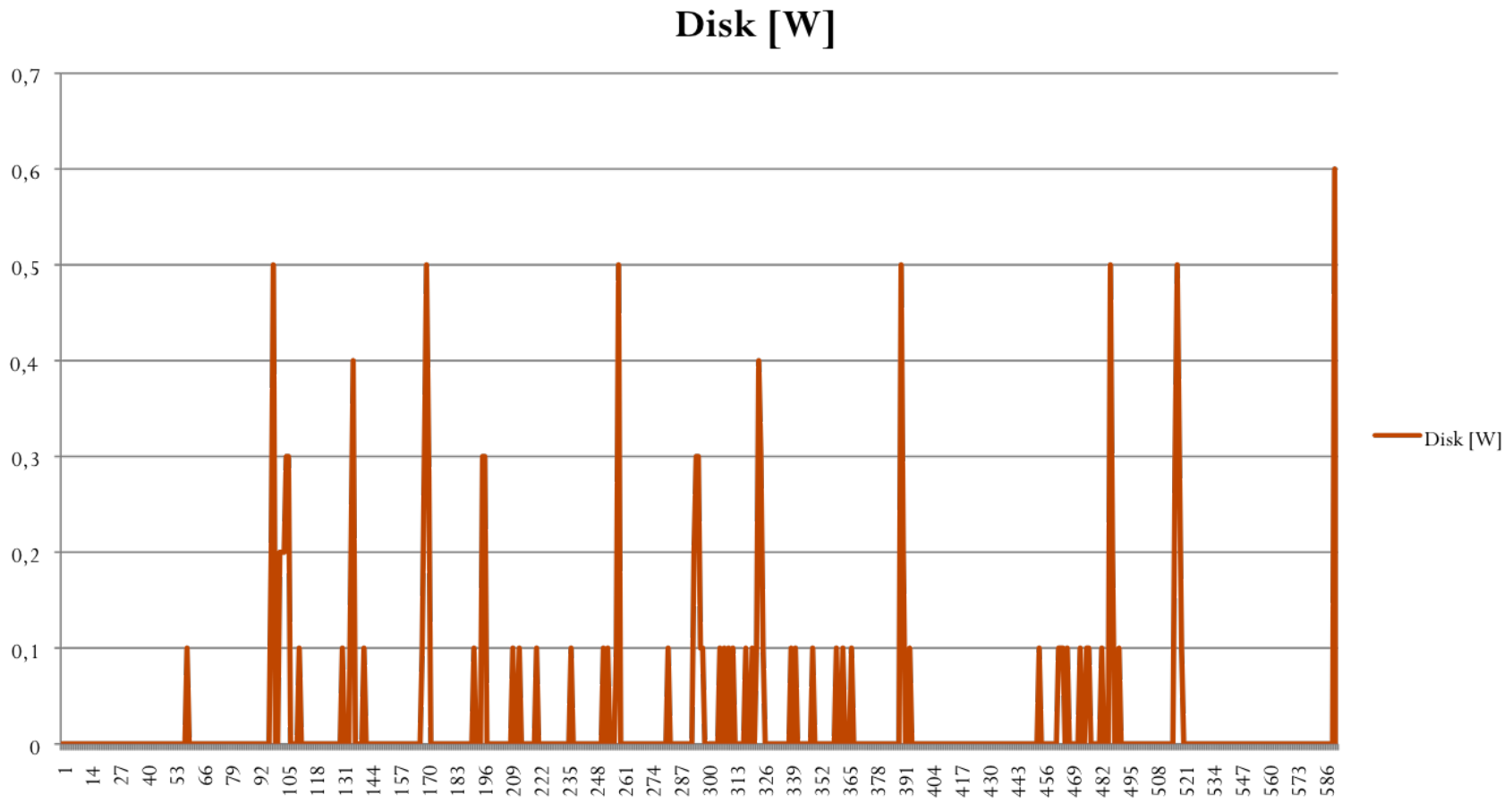
EXAMPLE 3: MEASURING THIRD-PARTY SOFTWARE (INTELLECTUAL OUTPUT 1 OF THE ERASMUS+ PROJECT 2017-1-SK01-KA203-035402)

- To select the best performing alternative (for development or use)
- Preparing energy benchmarks

EXAMPLE 3: MEASURING THIRD-PARTY SOFTWARE (INTELLECTUAL OUTPUT 1 OF THE ERASMUS+ PROJECT 2017-1-SK01-KA203-035402)



EXAMPLE 3: MEASURING THIRD-PARTY SOFTWARE (INTELLECTUAL OUTPUT 1 OF THE ERASMUS+ PROJECT 2017-1-SK01-KA203-035402)

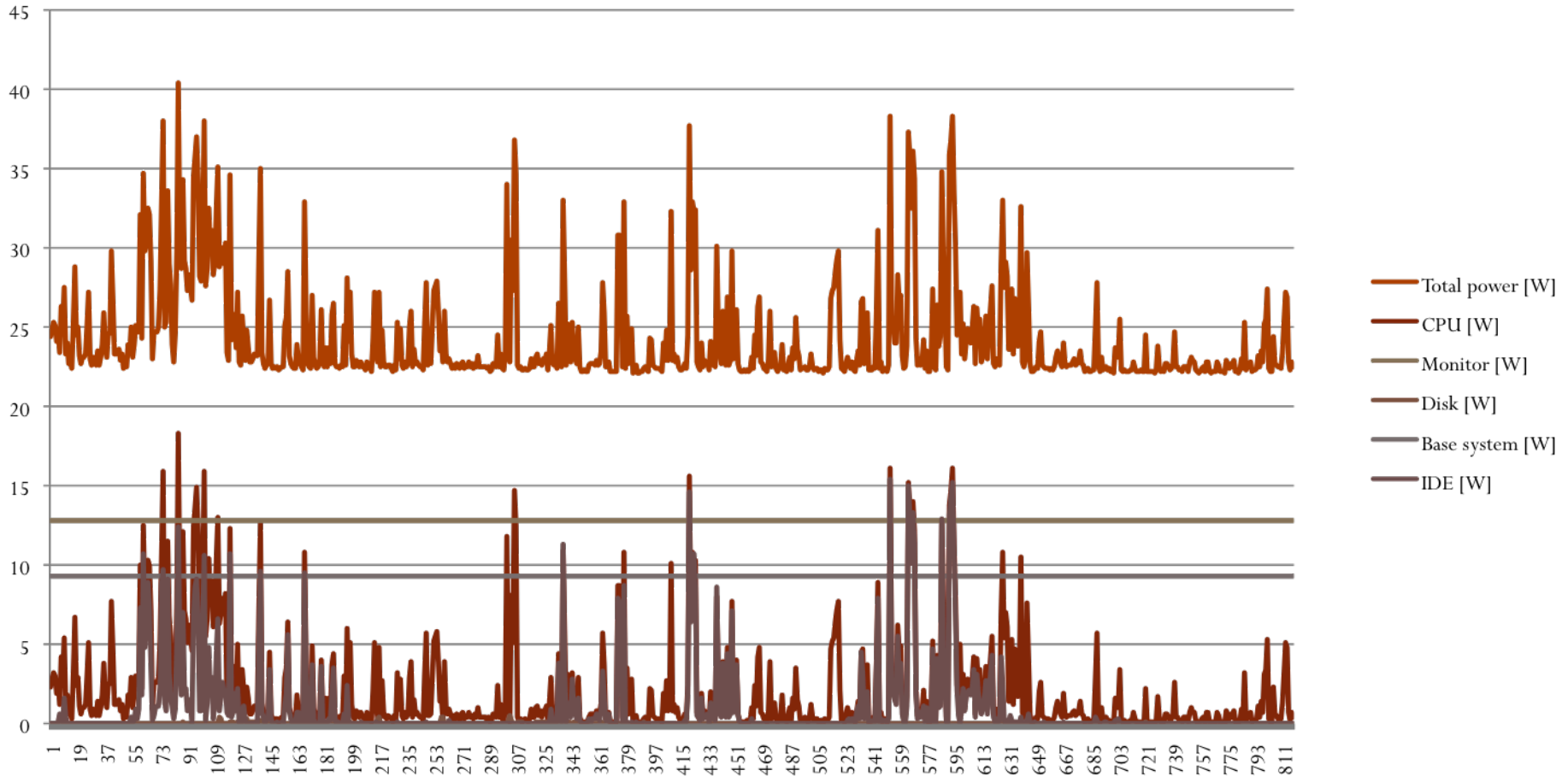


EXAMPLE 4:
MEASURING THE (SIMPLE) DEVELOPMENT PROCESS

- Starting/ending when starting/ending to work
- Considering short breaks as well

- Useful as topic at retrospective meetings and analysis

EXAMPLE 4:
MEASURING THE (SIMPLE) DEVELOPMENT PROCESS



EXAMPLE 5:
MEASURING DISTRIBUTED APPLICATIONS

- All nodes running the application
- Communication network

- Distributed testing
- Performance testing
- User profiles

WHAT TO DO WITH THE MEASUREMENTS' RESULTS?

- E1 application energy consumption: 2.19 Ws
- E2 in total, image cashing consumes about 50% less energy because it is 60% faster, even that there are moments it is consuming more energy
- E3 disk operations' energy consumption: 1.35 Ws, total energy consumption: 1771.03 Ws
- E4 IDE energy consumption: 74.72 Ws
- E4/5 total host system energy consumption: 1983.28 Ws

WHAT TO DO WITH THE MEASUREMENTS' RESULTS?

- Setting up the OS
- Uninstalling bad performing apps
- Fine-tuning the HW/SW/USR ecosystem
- SW evolution
- Software quality measurements
- Project planning to avoid high energy peaks

RECAPITULATION

- In-development (OpenCV+JavaFX+Joulemeter)
- Post-development (Android app+Trepn)
- Measuring third-party sw (IntelliJ IDEA install app+Joulemeter)
- How does the energy profile of the software development process look like (Chrome+IntelliJ+JavaFX SceneBuilder+developed app)
- Distributed applications (clients+servers+route)

FUTURE WORK IN GREEN SOFTWARE ENGINEERING

- Research:
  - Unification of principles
  - Standardization of representation of results
  - Software energy efficiency (label?)
- Industry:
  - Scaling over the path “single developer -> team -> management (project) -> company maturity”
- Academia:
  - Prepare this future by educating the people

European Union

THANK YOU FOR YOUR ATTENTION!

CSABA.SZABO
@TUKE.SK

EFOP-3.6.2-16-2017-00013

SZÉCHENYI 2020

HUNGARIAN GOVERNMENT

European Union
European Social Fund

INVESTING IN YOUR FUTURE