ARPE

Analyzing the relationship between parameters and effectors





Martina Maggio, Henry Hoffmann
Lund University, University of Chicago

Motivation

- Controlling computing systems is messy
- Often you hear “modeling that system is too complicated, therefore...”

Motivation

- And it’s actually true
- Depending on the model level, creating a model for a computing system might be really complicated

Motivation

- Example:
- Suppose we want to model power consumption of a software piece given the datasheet power consumption of the instruction set

we need to know that the software code translates into a set of atomic operations

Motivation

- This might be really “complicated”
- However, often it is much easier to grasp the relationship between the variable that can be changed and some final effect

Motivation

- This is a first step to respond to the question of how to automatically control the system

Outline

- Methodology
- Experimental evaluation
- Conclusion and future work

Methodology

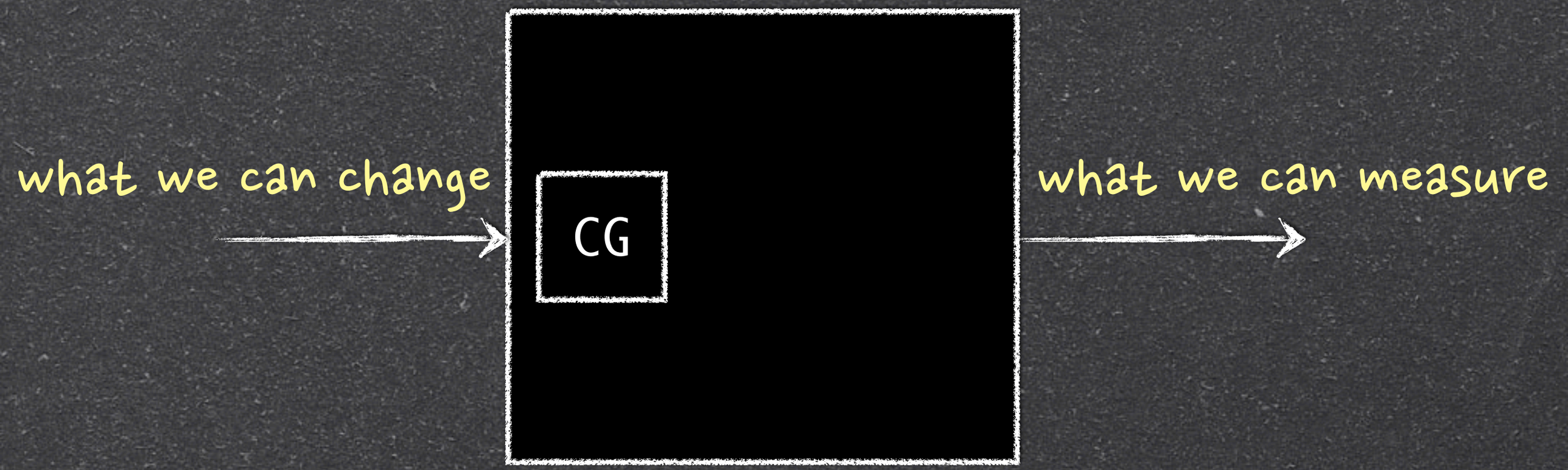
what we can change

ideally:

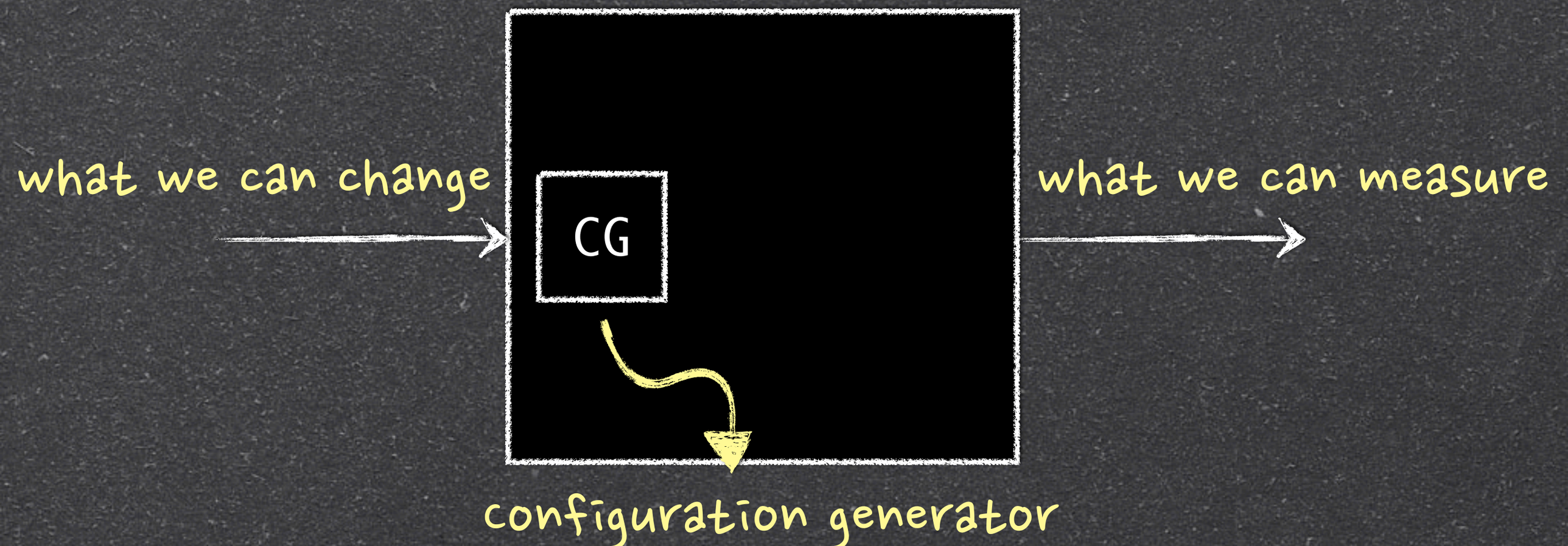
something that generates one or more models

what we can measure

Methodology



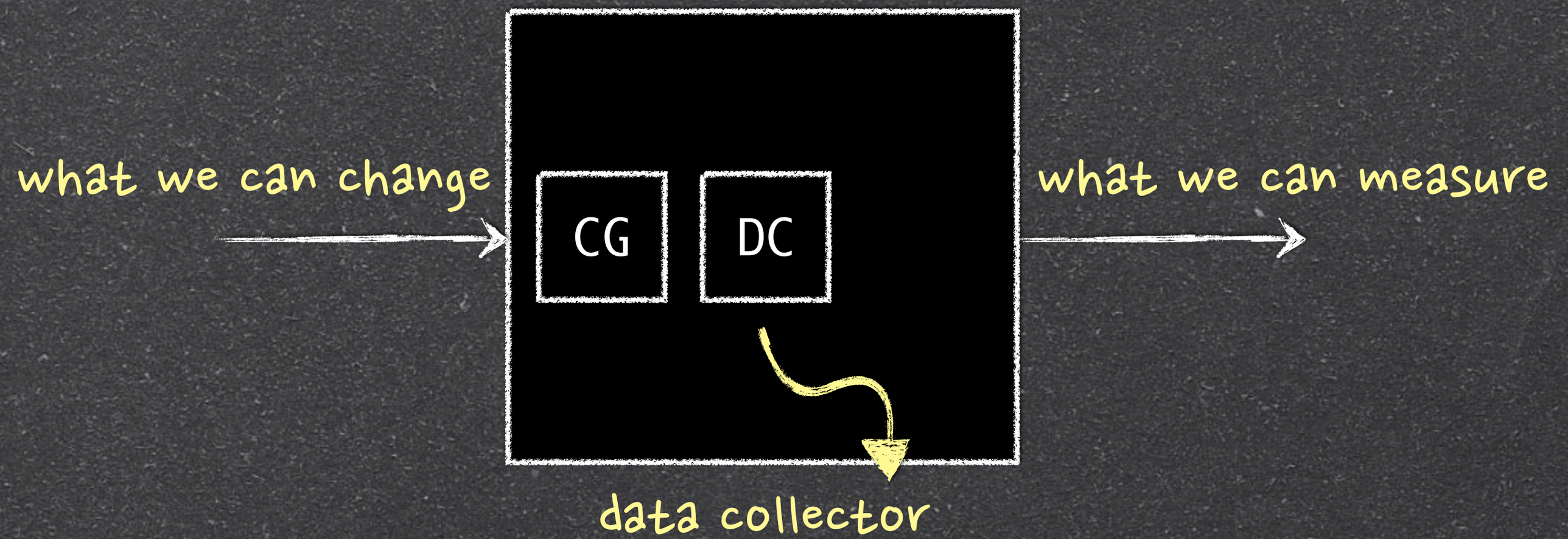
Methodology



Configuration generator

- Selects a set of configurations given the parameter list and their values
- Automatically generates scripts to execute experiments on the system to build the model

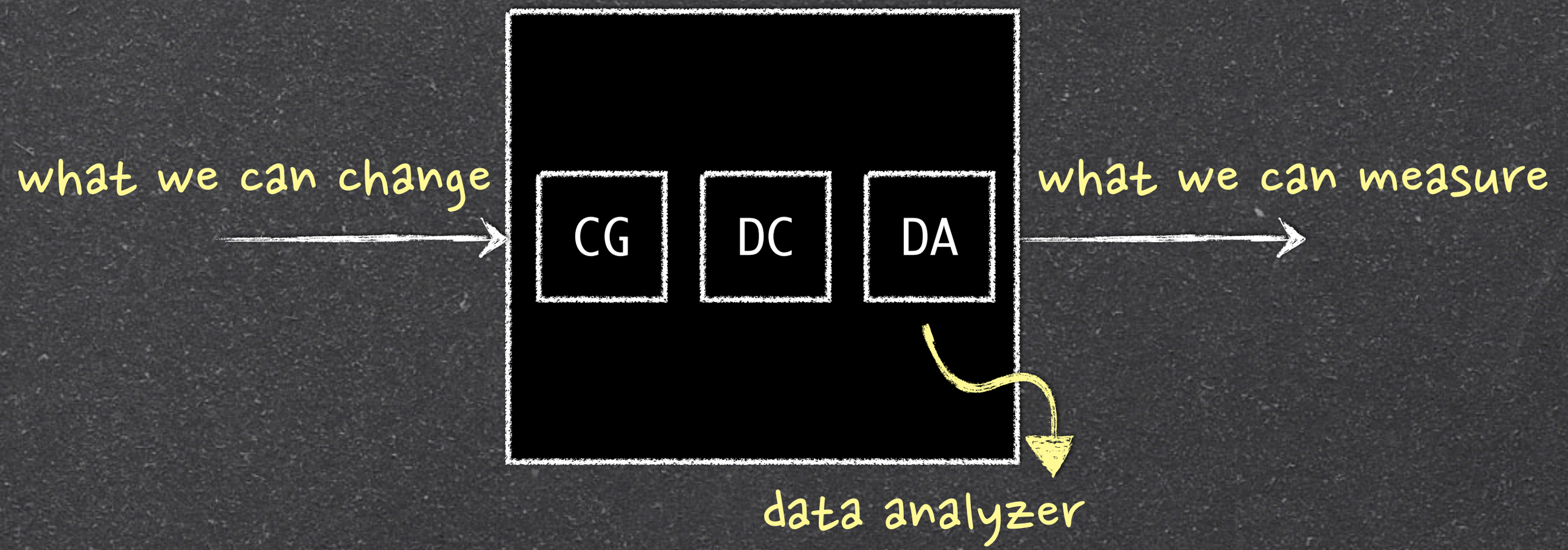
Methodology



Data collector

- Executes experiments on the system collecting relevant data automatically

Methodology



Data analyzer

- Given the collected data generated many different models of the system and selects the less “error prone”

Data analyzer

- Single parameter linear regression to explain the relationship between one parameter and the “measured” effect
- Multiple parameters multivariate linear regression to build a comprehensive model

Data analyzer example

- Execute tomcat from the DaCapo benchmark suite 50 times per configuration
  - Parameters: number of threads (1 to 32), iterations (1 to 10)
  - Model the real execution time of one instance



8 seconds to process the data, low errors

Experimental results

- Execution time
- Energy consumed

Execution time

- Comparison of models for execution time given performance counter or heartbeats

Execution time

- We measure the number of instructions retired per second with OProfile and the number of heartbeats retired per second for instrumeted applications: swaptions and x264 from the PARSEC benchmark suite

swaptions

	heart rate	instructions retired
training set	0.00027 [s]	0.00034 [s]
test set	0.00030 [s]	0.04130 [s]

x264

	heart rate	instructions retired
training set	0.00494 [s]	0.00072 [s]
test set	0.00353 [s]	0.00072 [s]

Execution time

- Results insight:
  - There is no unique parameter to predict execution time, for some benchmark the heart rate does a better job, for others the number of instructions retired is more insightful



future work might be: how do we distinguish?

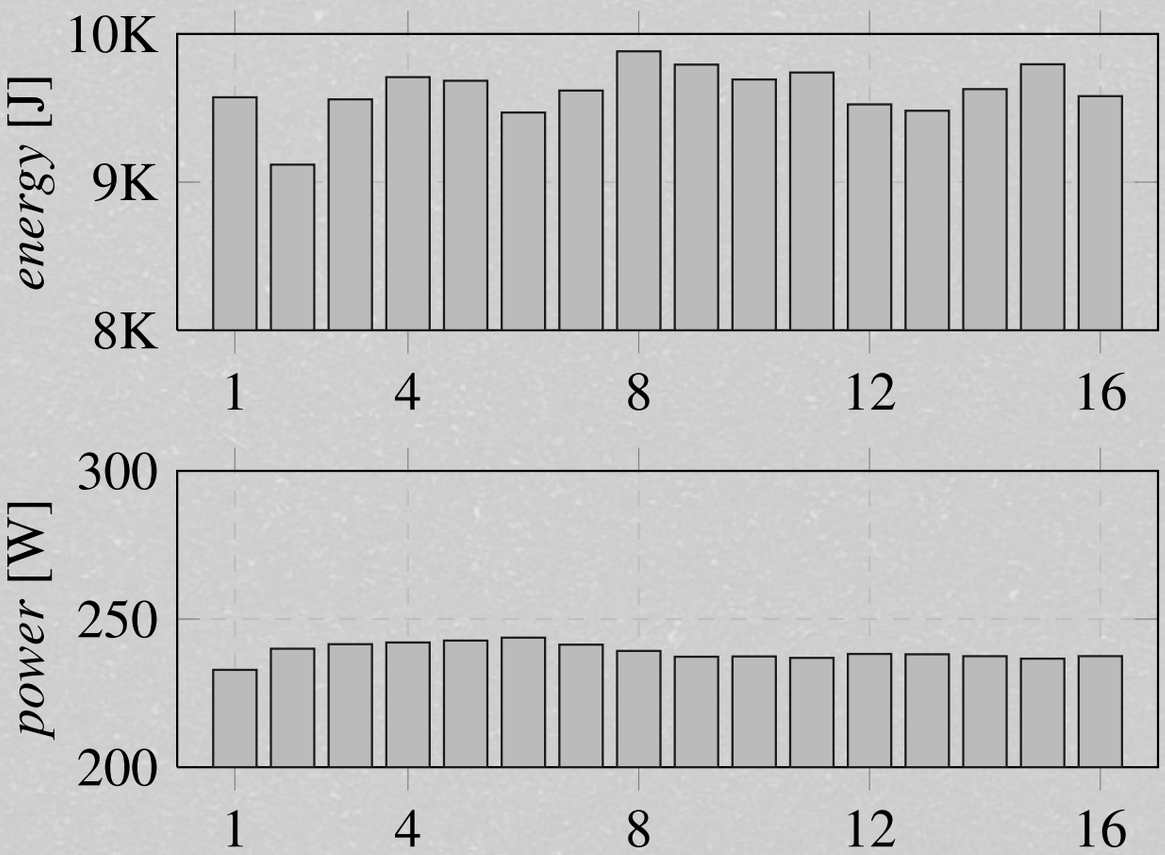
Energy consumption

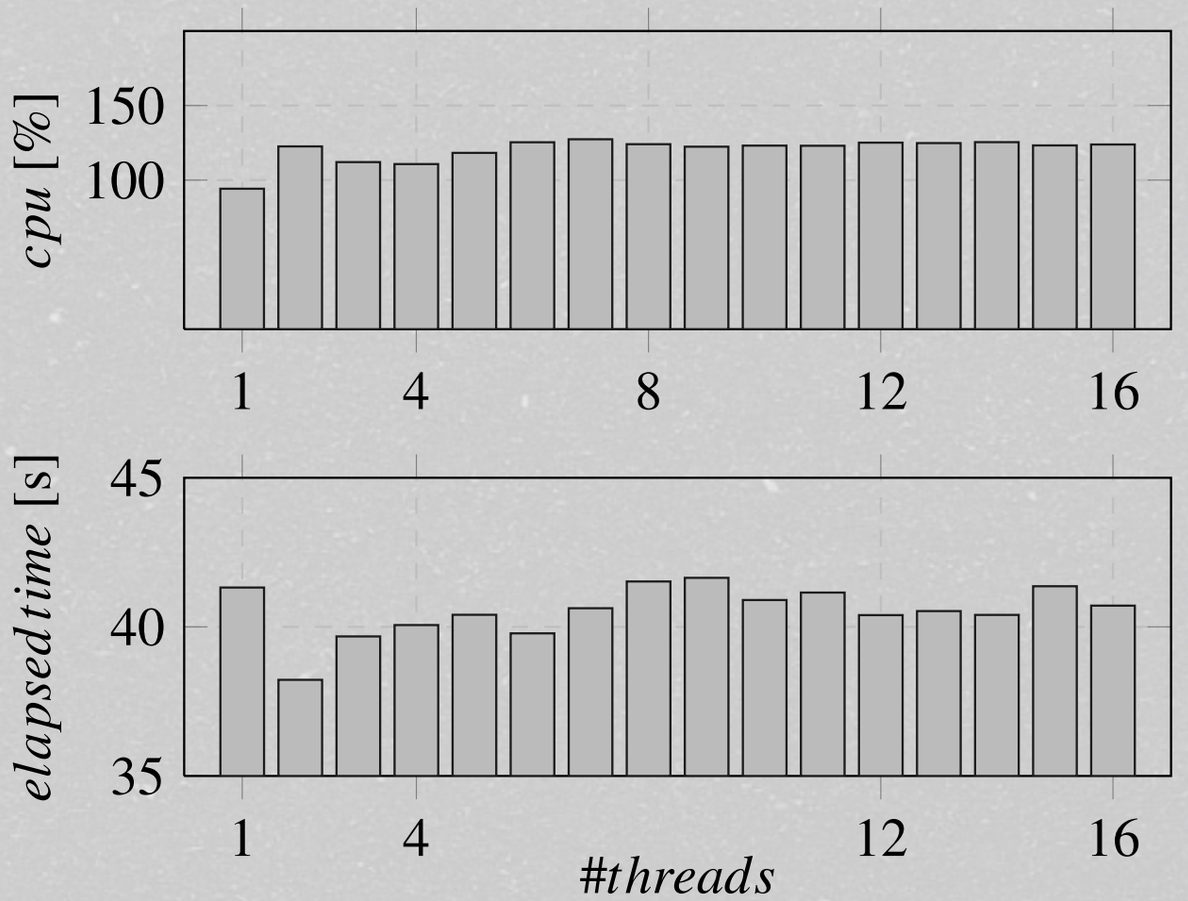
- Do we really need hardware power sensors or can we infer the power consumption of CPU-intensive applications based on the CPU consumption parameter?

Energy consumption

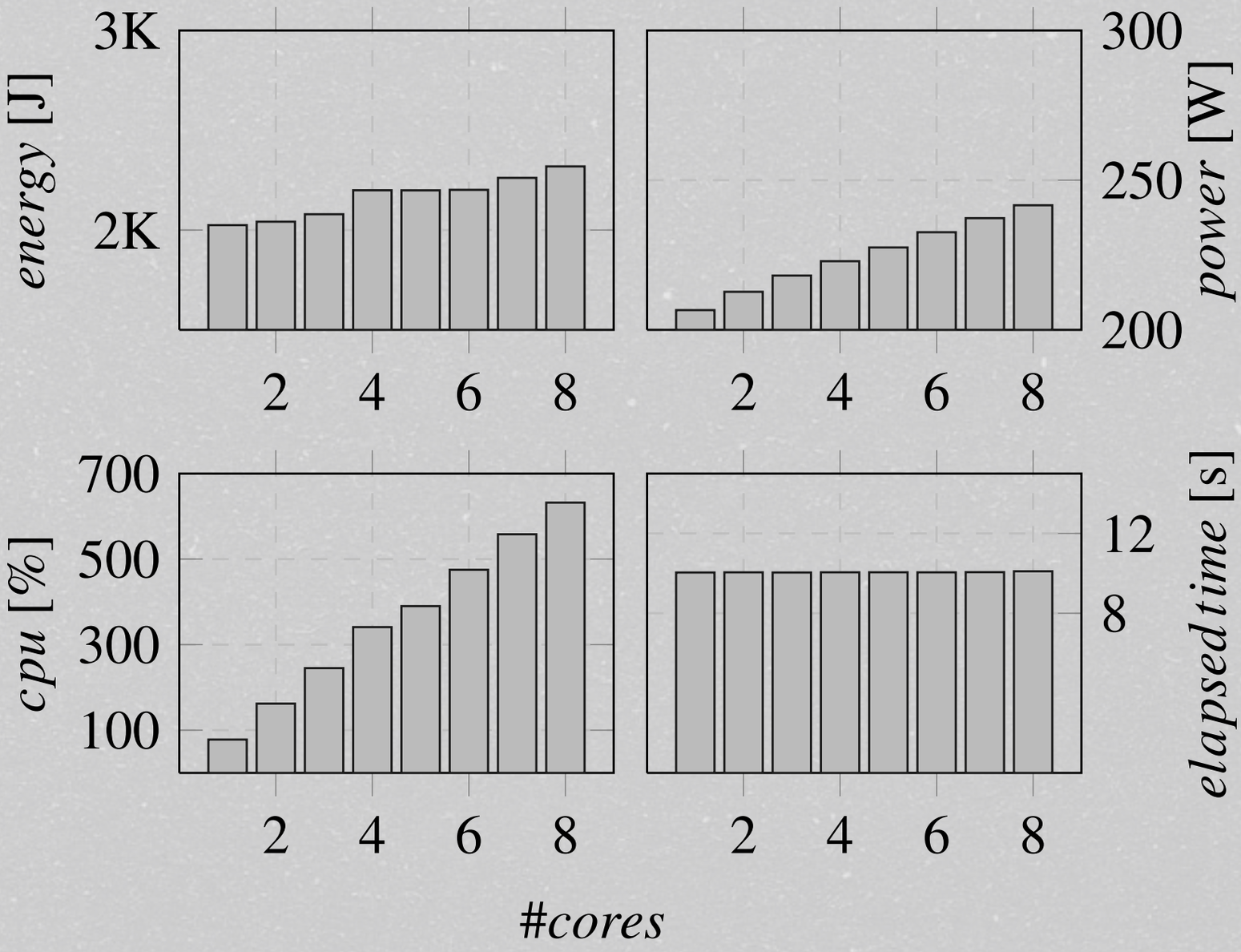
- We measure the execution time, the power consumption and the CPU consumption of sunflow, treadbeans (from the DaCapo benchmark suite) and lookbusy

Energy consumption





Energy consumption



Errors

	power & time	CPU & time
sunflow	16.10 [J]	287.90 [J]
treadbeans	59.99 [J]	126.03 [J]
lookbusy	20.62 [J]	19.48 [J]

Highlight

- Note that all the sensors used are already coded in the released version, for example the tool automatically measures and stores CPU consumption during the benchmark execution without human intervention

Conclusion

- We have developed a tool to build models based on the execution of applications on specific hardware
- The tool is given a parameter list and reasonable values or intervals and executes the application with different configuration producing equation based models

Future work

- Automatic controller generation, based on models
- Exploration of diagnostic features given the output of the tool - is self healing possible?
- Workload diversity checker

Questions?

Thanks for the attention

http://github.com/martinamaggio/arpe