

# Runtime Solutions for Power-Aware Applications



---

**Daniele De Sensi**

Supervisor: Marco Danelutto

***Computer Science Department,  
University of Pisa, Italy***

# Hi!

---



PhD Student at  
University of **Pisa** (Italy)



Working on  
**Power-Aware** computing



Member of the **Parallel  
Programming Models** Group



Work done in the last  
**3 years**

**TARGET  
&  
MOTIVATION**



# SCENARIOS

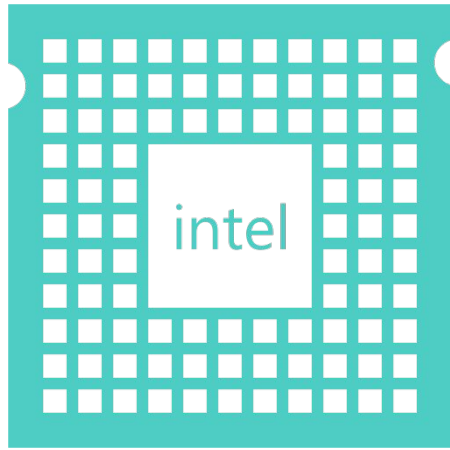
---

<i>Power Cap</i>	
Watts	$\leq 45$

<i>Performance Requirement</i>	
# Frames/Sec	$\geq 20$

# CONFIGURATION

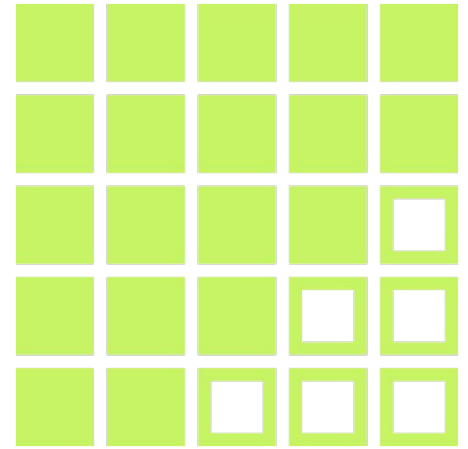
---



#Cores



Clock  
Frequency



Threads  
Mapping

# THESIS STRUCTURE

---

**Science**



**Engineering**



**SCIENCE**



# ONLINE LEARNING

User requirement: 24 Frames/Second

18 Frm/s	42 Frm/s	30 Frm/s
-------------	-------------	-------------

#Cores: 4	#Cores: 9	#Cores: 7
--------------	--------------	--------------



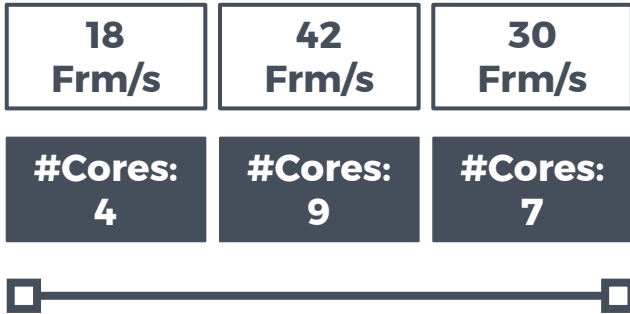
Calibration Phase



Time

# ONLINE LEARNING

User requirement: 24 Frames/Second



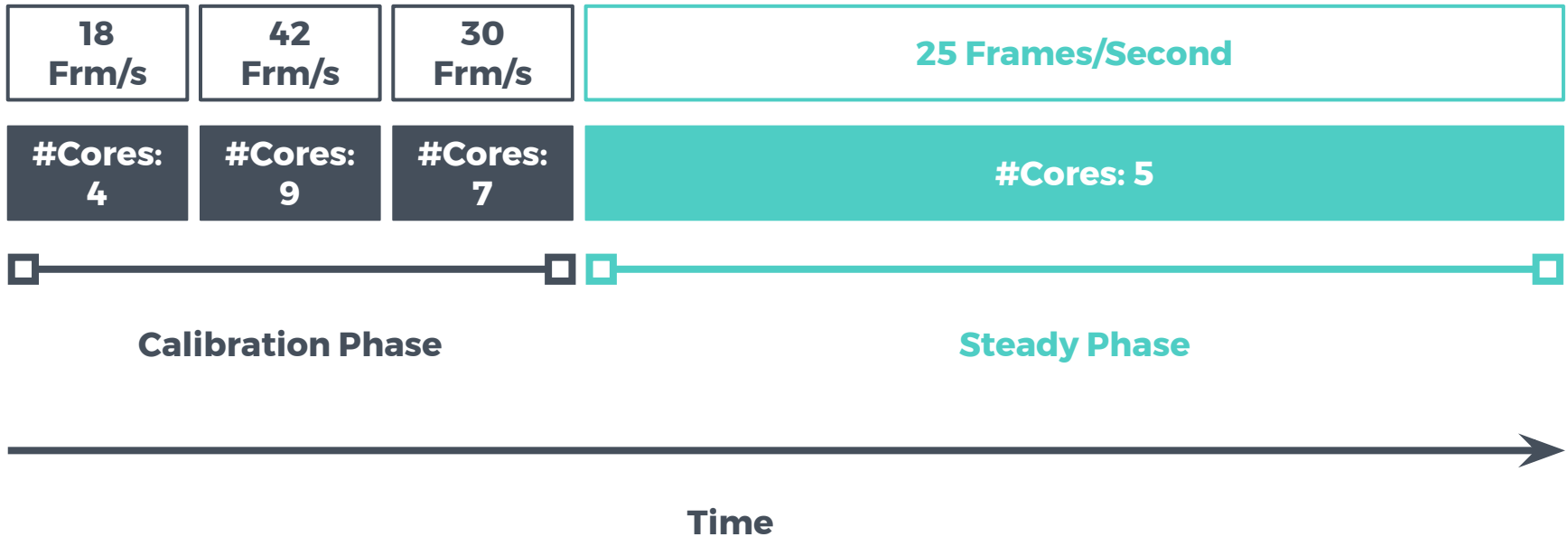
Calibration Phase



Time

# ONLINE LEARNING

User requirement: 24 Frames/Second



# ONLINE LEARNING

---



1. How much data to collect?
2. How expensive to compute model and perform predictions?

# ALGORITHM

---

$$T_S(n, f) = \frac{T_S(n, f_{max}) - T_S(n, f_{min})}{f_{max} - f_{min}} \cdot f + \frac{T_S(n, f_{min}) \cdot f_{max} - T_S(n, f_{max}) \cdot f_{min}}{f_{max} - f_{min}}$$

$$\begin{aligned} P(n, f) &= I_{leak} \left[ \min(n, K) \left( V(f) - v_{min} \right) + K v_{min} \right] + A_c V(f)^2 f n = \\ &= a_0 \left[ \min(n, K) \left( V(f) - v_{min} \right) + K v_{min} \right] + a_1 V(f)^2 f n \end{aligned}$$



# RESULTS: FAILURES

---

Requirement:  $\geq 80$  Frames/Second

All configurations

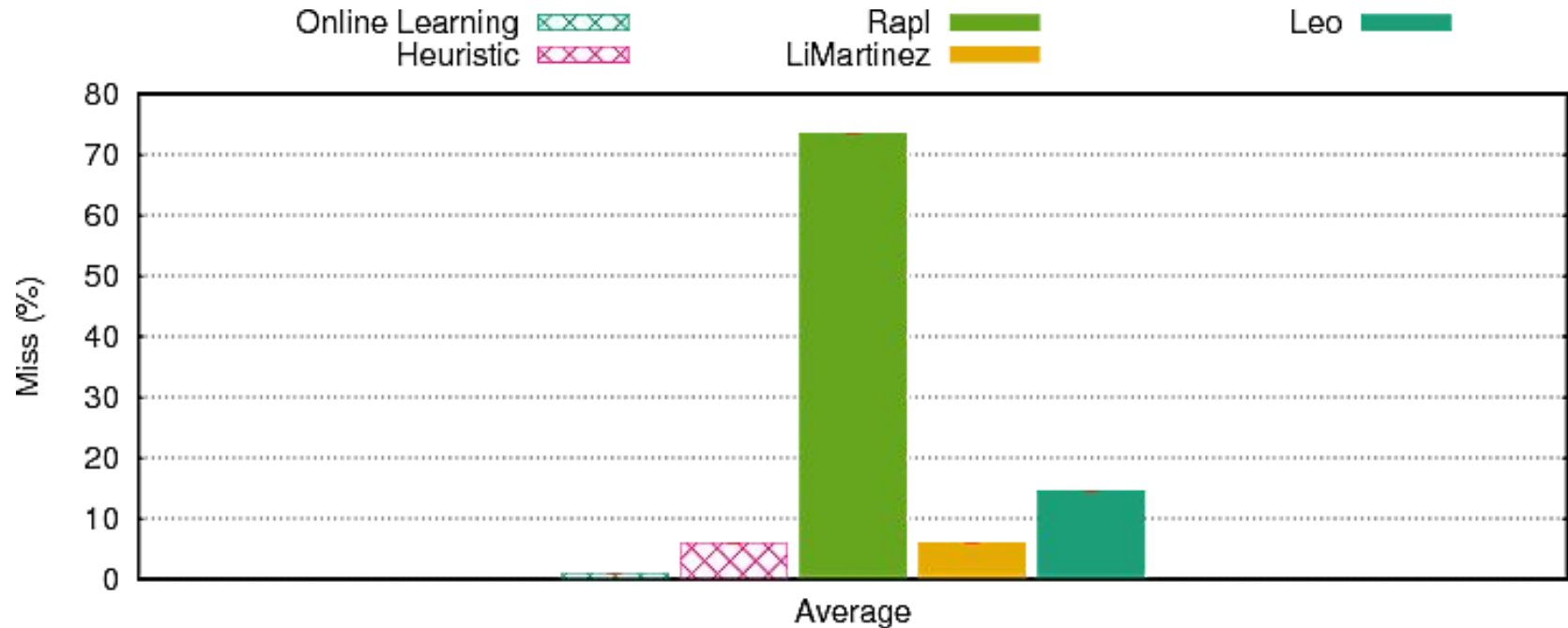
Performance Model

```
graph LR; A[All configurations] -- Performance Model --> B[Configurations with a performance >= 80 requests per second]
```

Configurations with a **performance**  $\geq 80$  requests per second

# RESULTS: FAILURES

Tested on a wide set of power/performance requirements (lower = better)



# RESULTS: OPTIMALITY

Requirement:  $\geq 80$  Frames/Second

All configurations

Performance Model

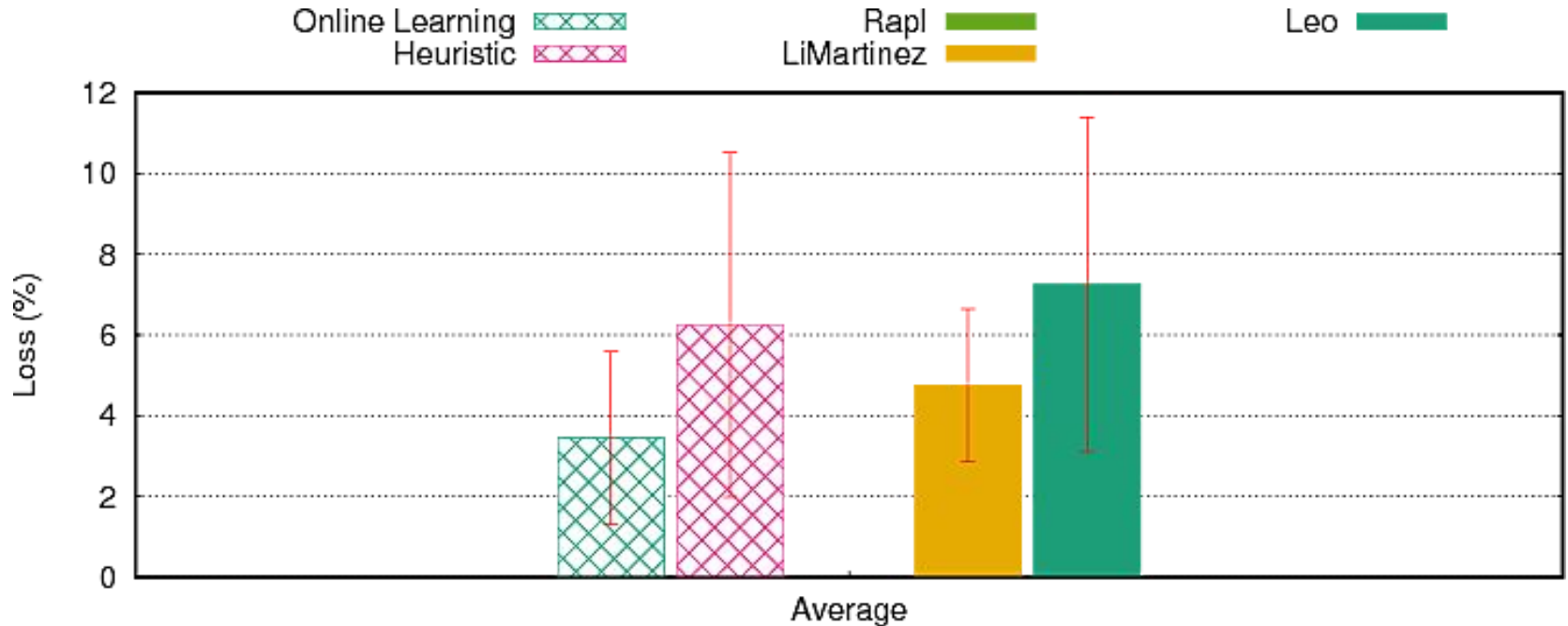
Configurations with a **performance**  $\geq 80$  requests per second

Power Consumption Model

**Lowest power consuming** configuration with a performance  $\geq 80$  requests per second

# RESULTS: OPTIMALITY

Tested on a wide set of power/performance requirements (lower = better)



# HOW MUCH DATA TO COLLECT?

---

Typical Training/Test:

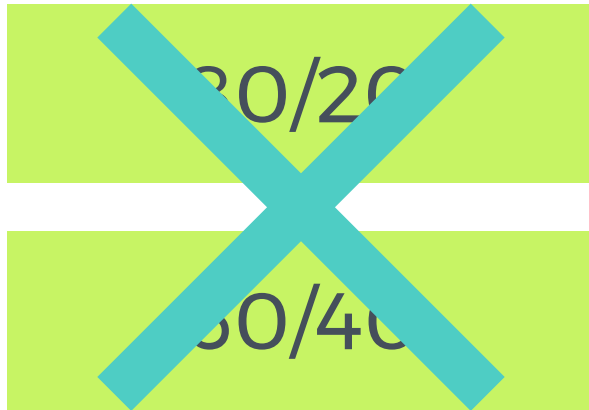
80/20

60/40

# HOW MUCH DATA TO COLLECT?

---

Typical Training/Test:



# HOW MUCH DATA TO COLLECT?

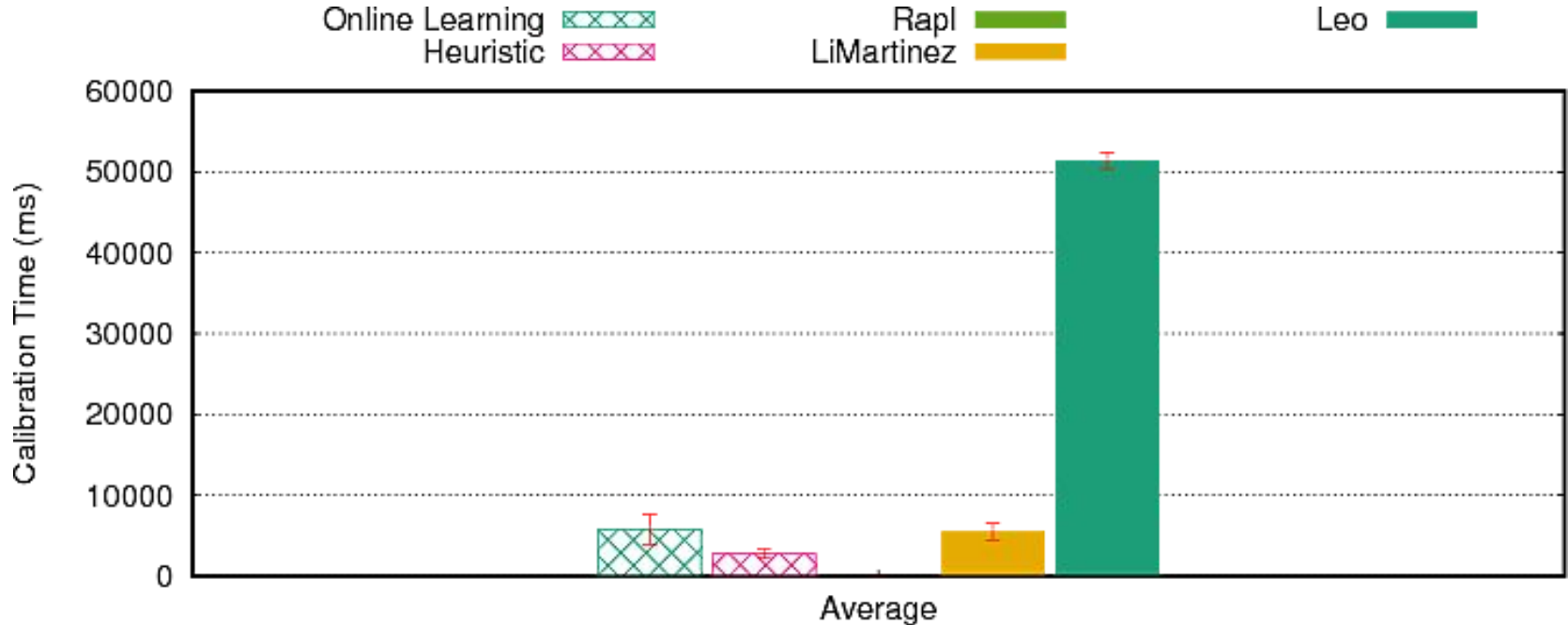
---

Our Algorithm Training/Test:

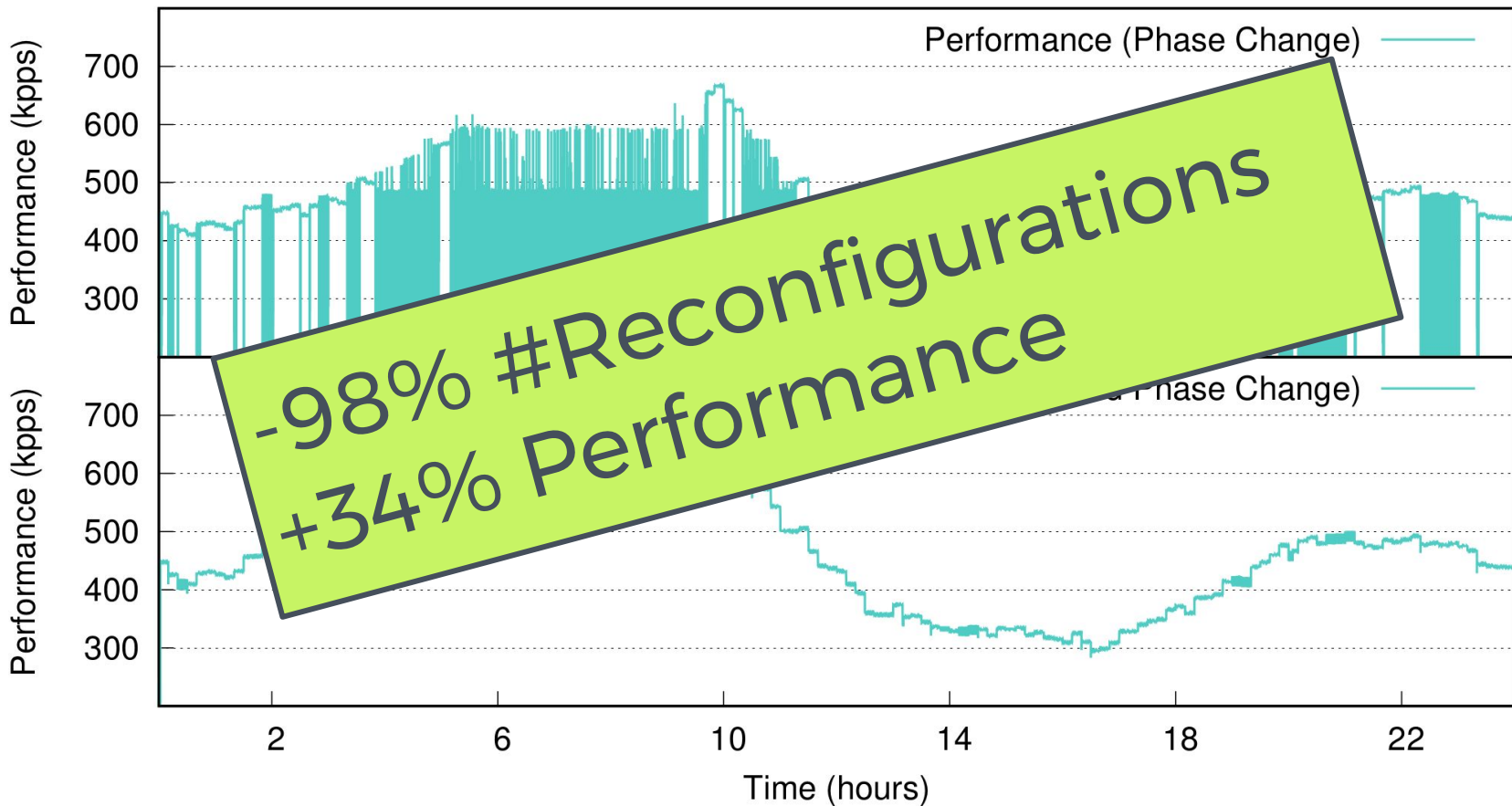
3/97

# RESULTS: OVERHEAD

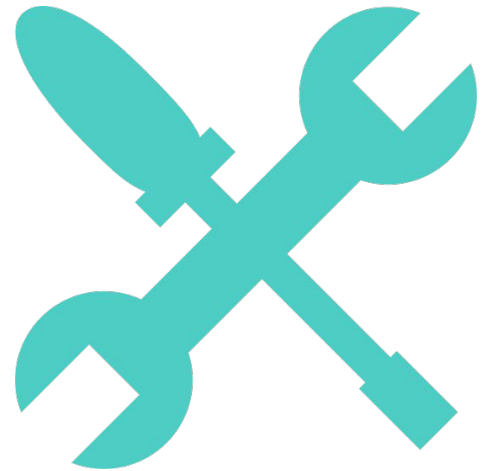
Tested on a wide set of power/performance requirements (lower = better)



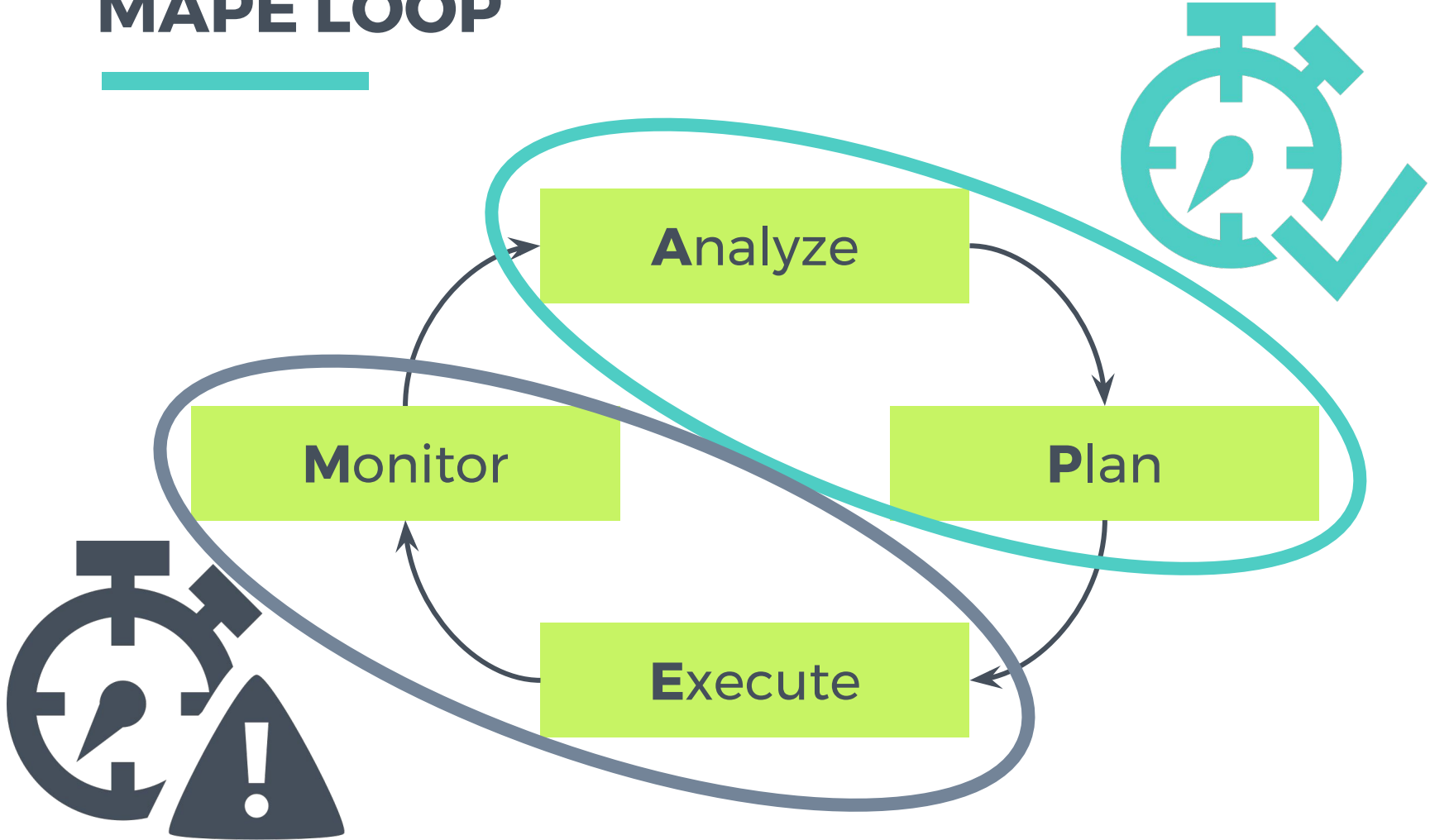
# STABILITY



**ENGINEERING**



# MAPE LOOP



# TARGET

---

Provide a **customizable** C++ framework for building **autonomic** and **power-aware** reconfiguration algorithms on shared memory multicores

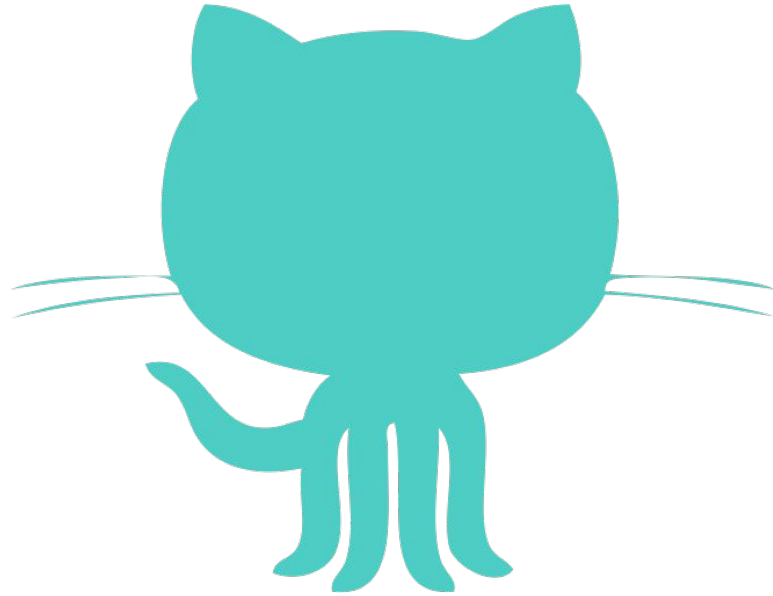
Designer focuses on the **algorithm**, exploiting monitoring infrastructure and executors provided by the framework

Easily **compare** new strategies with state of the art ones

Wide set of applications provided as **testbed**

# NORNIR

---



<http://danieledesensi.github.io/nornir/>

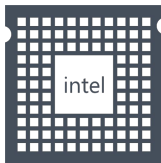
# SOFTWARE STACK

---

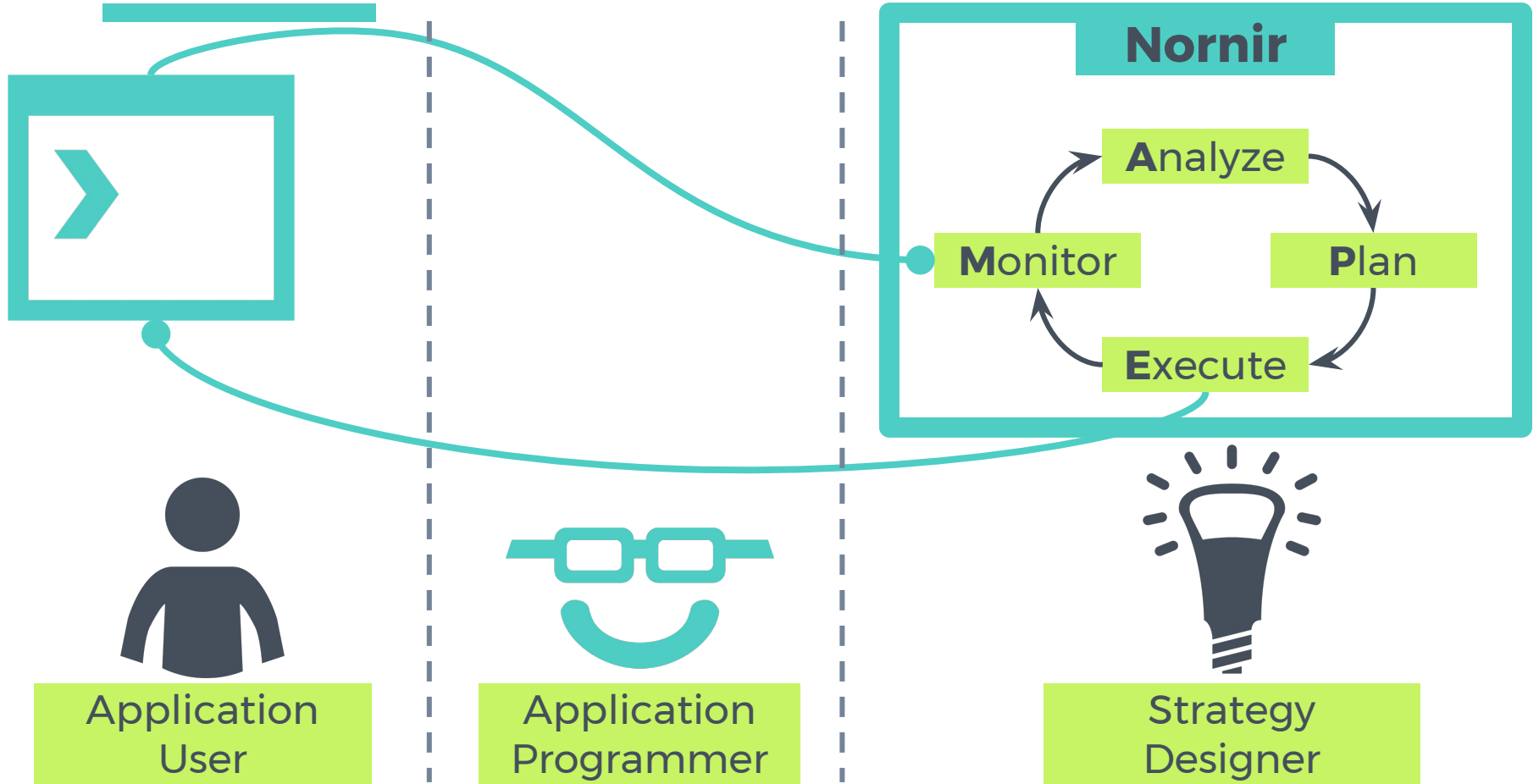
Applications (P<sup>3</sup>ARSEC)

Decision Algorithms (Nornir)

Hardware Abstraction (Mammut)



# INTERACTION



# EXAMPLE

---

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<nornirParameters>
```

```
  <requirements>
```

```
    <powerConsumption>50</powerConsumption>
```

```
    <throughput>MAX</throughput>
```

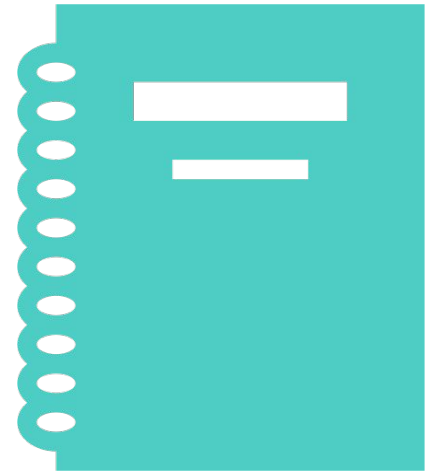
```
  </requirements>
```

```
</nornirParameters>
```

# DASHBOARD



# REFERENCES



# REFERENCES



1. De Sensi D., Torquati M., Danelutto M.  
A Reconfiguration Algorithm for Power-Aware Parallel Applications  
**ACM Transactions on Architecture and Code Optimization, 2016**
2. De Sensi D., De Matteis T., Torquati M., Mencagli G., Danelutto M.  
Bringing Parallel Patterns out of the Corner:  
the P3ARSEC Benchmark Suite  
**ACM Transactions on Architecture and Code Optimization, 2017**
3. De Sensi D., De Matteis T., Danelutto M.  
Nornir: Autonomic and Power-Aware Computing for the Masses  
Submitted to **Future Generation Computer Systems, 2017**
4. De Sensi D., Torquati M., Danelutto M.  
Mammut: High-level management of system knobs and sensors  
**SoftwareX, 2017**
5. Torquati M., De Sensi D., Mencagli G., Aldinucci M., Danelutto M.  
Power-Aware Pipelining with Automatic Concurrency Control  
Submitted to **Concurrency and Computation: Practice and Experience, 2017**
6. Danelutto M., De Sensi D., Torquati M.  
A Power-Aware, Self-Adaptive Macro Data Flow Framework  
**Parallel Processing Letters, 2016**
7. Danelutto M., De Matteis T., De Sensi D., Mencagli G., Torquati M.  
P3ARSEC: Towards Parallel Patterns Benchmarking  
**Proceedings of the 32nd Annual ACM Symposium on Applied Computing (SAC), 2017**
8. Danelutto M., De Matteis T., De Sensi D., Torquati M.  
Evaluating Concurrency Throttling and Thread Packing on SMT Multicores  
**Proceedings of 25th Intl. Conf. on Parallel, Distributed and Network-Based Processing (PDP), 2017**
9. De Sensi D.  
Predicting Performance and Power Consumption of Parallel Applications  
**Proceedings of 24th Intl. Conf. on Parallel, Distributed and Network-Based Processing (PDP), 2016**
10. Danelutto M., De Sensi D., Torquati M.  
Energy Driven Adaptivity in Stream Parallel Computations  
**Proceedings of 23rd Intl. Conf. on Parallel, Distributed and Network-Based Processing (PDP), 2015**
11. De Sensi D., De Matteis T., Danelutto M.  
Nornir: A Customisable Framework for Autonomic and Power-Aware Applications  
**Proceedings of Euro-Par Workshops, 2017**
12. Aldinucci M., Danelutto M., De Sensi D., Mencagli G., Torquati M.  
Towards Power-Aware Data Pipelining on Multicores  
**Proceedings of HLPP: International Workshop on High-Level Parallel Programming, 2017**
13. Danelutto M., De Sensi D., Torquati M.  
A Power-Aware, Self-Adaptive Macro Data Flow Framework  
**Proceedings of HLPP: International Workshop on High-Level Parallel Programming, 2016**
14. De Sensi D., Kilpatrick P., Torquati M.  
State-aware Concurrency Throttling  
**Proceedings of Parallel Computing (ParCo) Conference, 2017**

**CONCLUSIONS  
&  
FUTURE WORK**



# CONCLUSIONS

---

- Algorithms for enforcing requirements on performance and power consumption, improving state of the art
- A framework for implementing **autonomic** and **power-aware** reconfiguration strategies with low programming effort
- A benchmark for testing such algorithms
- ...

# FUTURE WORK

---

Extend to **multiple** applications

Support **distributed memory** architectures

Other **knobs** (e.g. offcore frequency)



# Thanks for your attention

[desensi@di.unipi.it](mailto:desensi@di.unipi.it)



**Daniele De Sensi**