Autotuning Parallel Application in Heterogeneous Systems

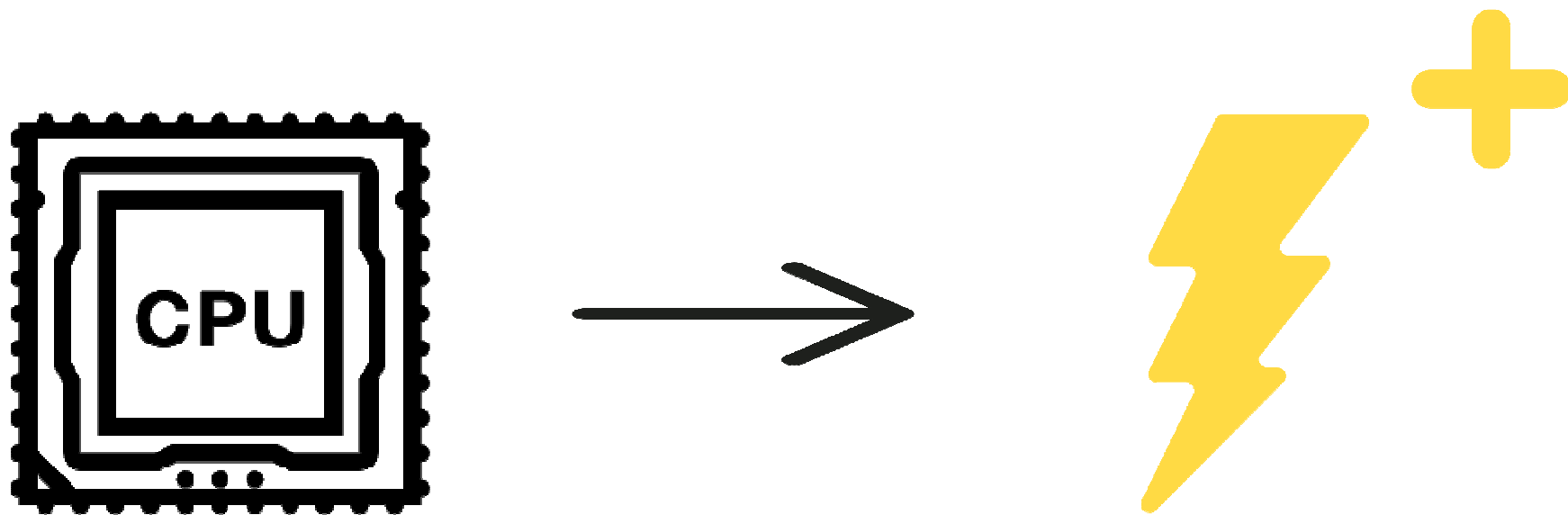João Alberto Trigo de Bordalo Morais
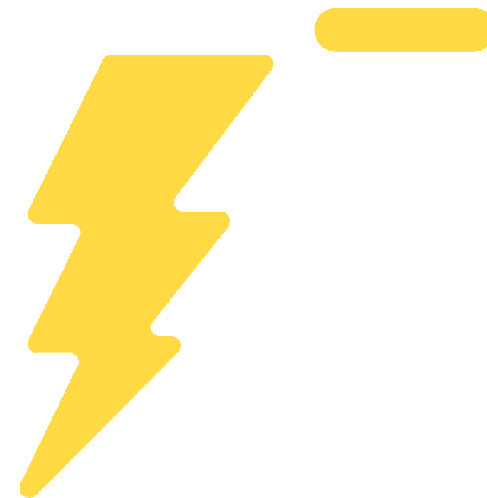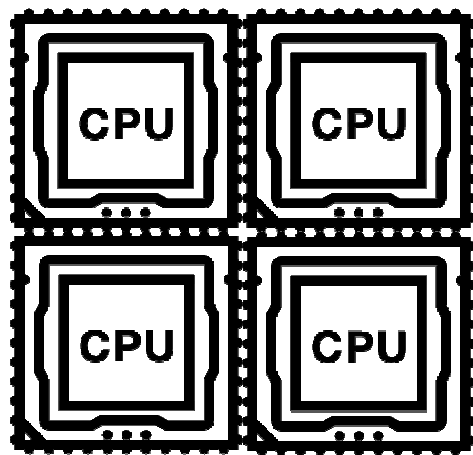
Supervisor:
Jorge Manuel Gomes Barbosa

LIACC

Agenda

- Context
- Achieving the Highest Processing Power
- Problem
- Possible Solution
- Solution's Approach
- Solution's Validation
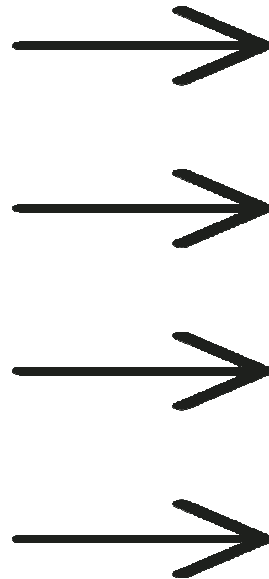- Solution's Methodology
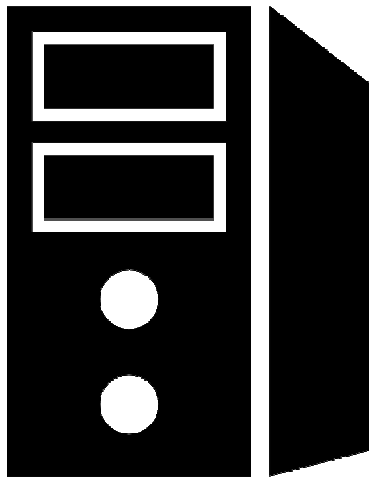- Impact/Goals
- Work plan
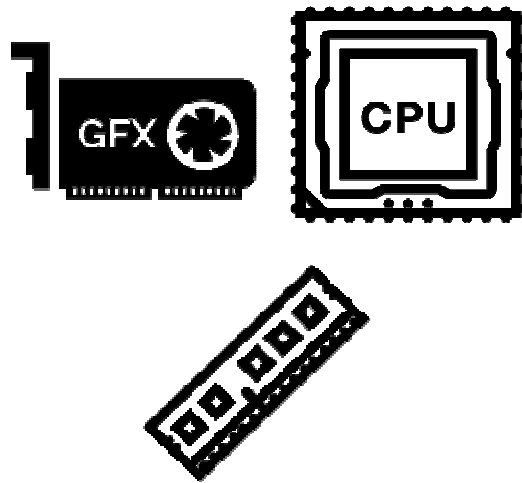
Context (1)

Context (2)
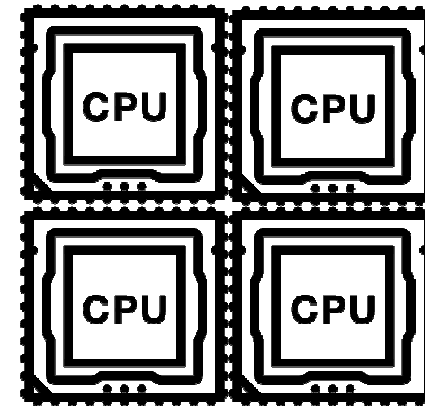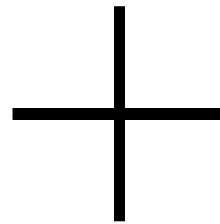
Context (3)

CPU
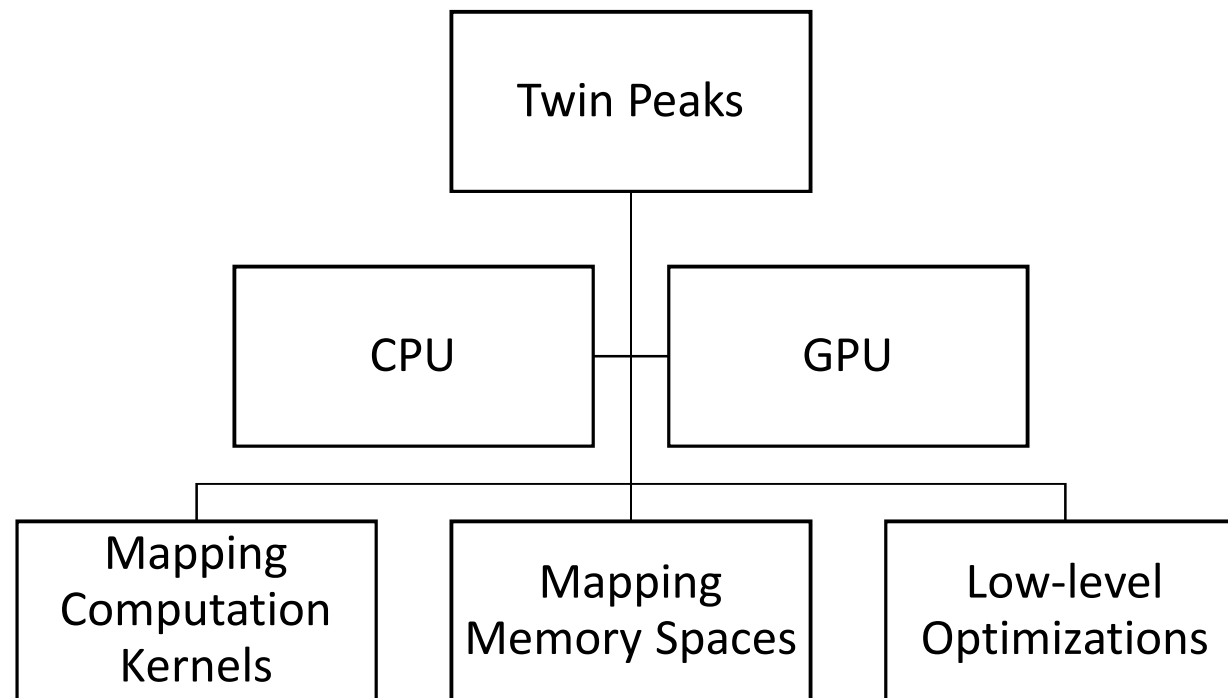
GPU

XEON PHI

FPGA

Achieving the Highest Processing Power

Using Computers'
Heterogeneous Components

Using Code Parallelization

Heterogeneous Components

```
                    ┌─────────────────┐
                    │   Twin Peaks    │
                    └────────┬────────┘
                             │
             ┌───────────────┴───────────────┐
      ┌──────┴──────┐                 ┌───────┴──────┐
      │     CPU     │                 │     GPU      │
      └──────┬──────┘                 └──────────────┘
             │
   ┌─────────┼──────────────────────────────────┐
┌──┴───────┐ ┌──┴──────────┐          ┌──────────┴──┐
│ Mapping  │ │  Mapping    │          │  Low-level  │
│Computation│ │Memory Spaces│          │Optimizations│
│ Kernels  │ │             │          │             │
└──────────┘ └─────────────┘          └─────────────┘
```

Code Parallelization

Kremlin
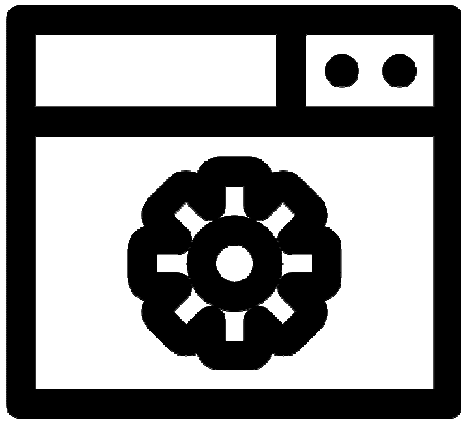- Find possible paralized regions
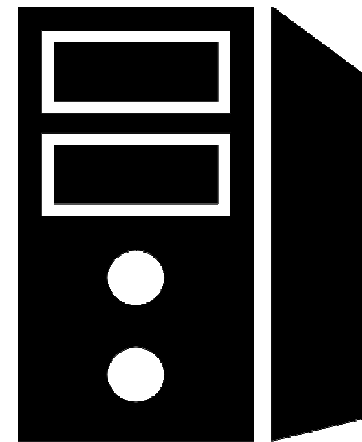
Kismet
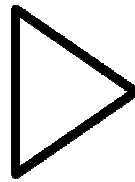- Parallel speedup estimation

Argo
- autotuning framework to dynamically adapt applications in multicore architectures

Problem

Software Application
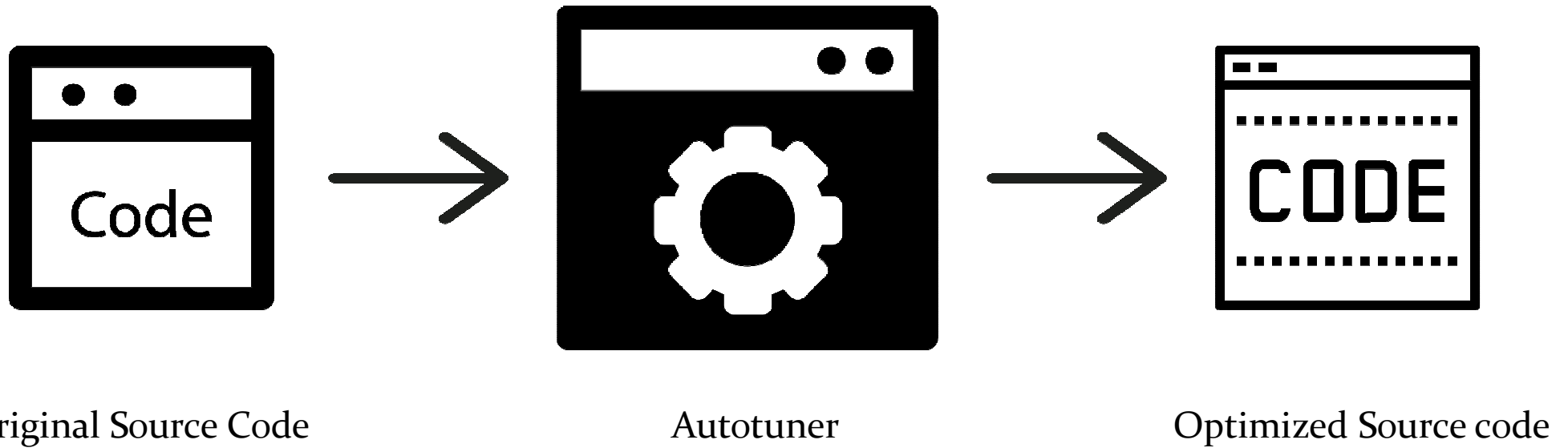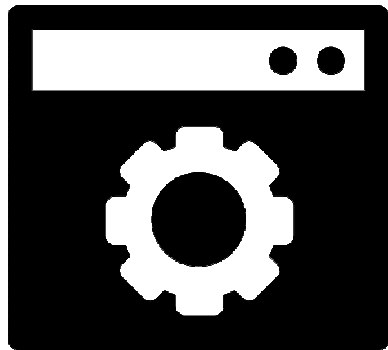
Heterogeneous System

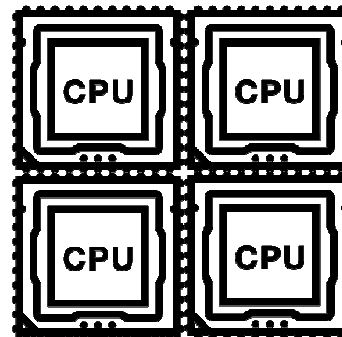# Possible Solution

Original Source Code   Autotuner   Optimized Source code

Solution's Aproach (1)

Autotuner

Kremlin
+
Expert
Parallelization

Solution's Aproach (2)

Kremlin	Expert Parallelization
• Detect Possible Parallized code • Avaliation of its Speed	• Manual Code modifications

Compare best results
+
Find Autotunning's Parameters

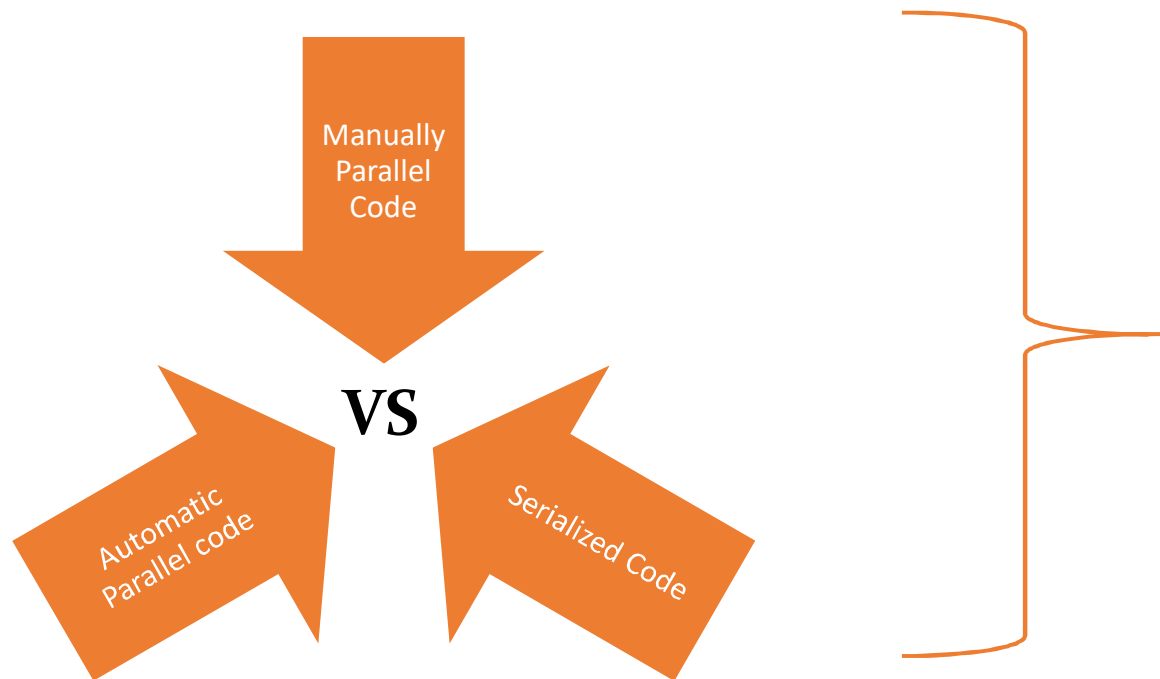# Solution's Methodology

Solution's Validation (1)

2 use cases:

- A biopharmaceutical HPC application for accelerating drug discovery;

&

- A self-adaptive navigation system to be used in smart cities

Solution's Validation (2)

Manually Parallel Code

VS

Automatic Parallel code

Serialized Code

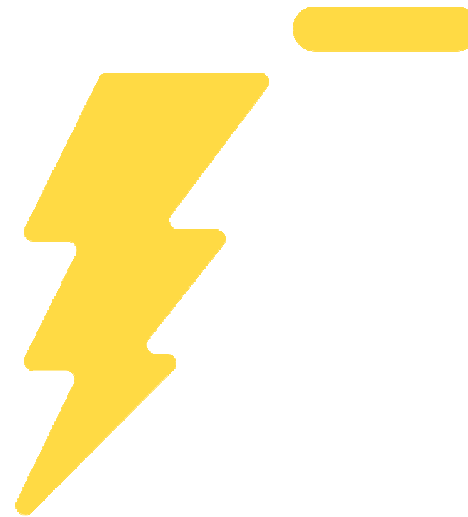**Evaluation Metrics**

Energy Consuming

Execution Time
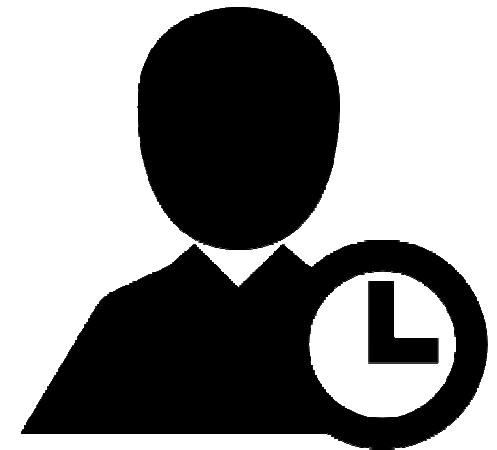
Memory Accesses

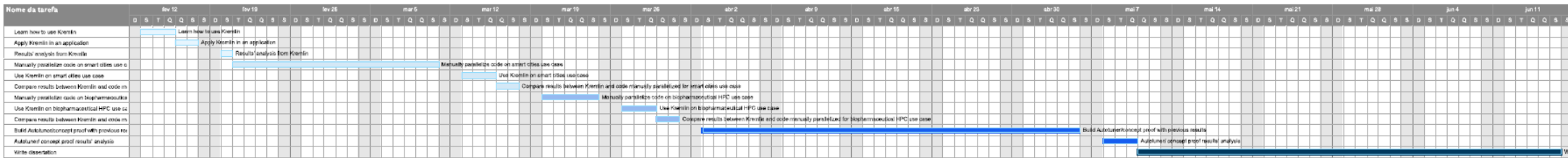Processing Power

Impact/Goals

Highest Performance

Less Energy Consumed

Developers' Time Saved

Work Plan

- **From 13/2 until 16/6**

- **5 distinct phases**

Faculdade de Engenharia da Universidade do Porto
Mestrado Integrado em Engenharia Informática e Computação