# Dynamic Autotuning of Algorithmic Skeletons

Chris Cummins

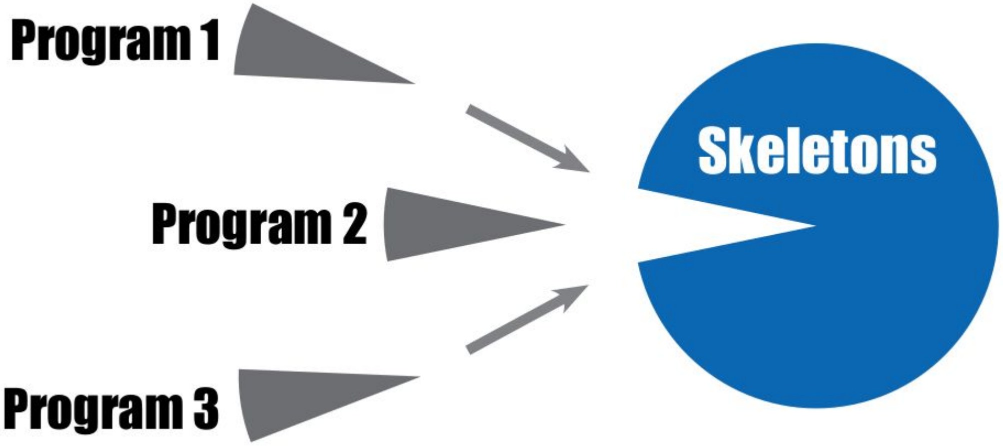
THE UNIVERSITY of EDINBURGH
informatics

EPSRC Centre for Doctoral Training in
Pervasive Parallelism

EPSRC
Engineering and Physical Sciences
Research Council

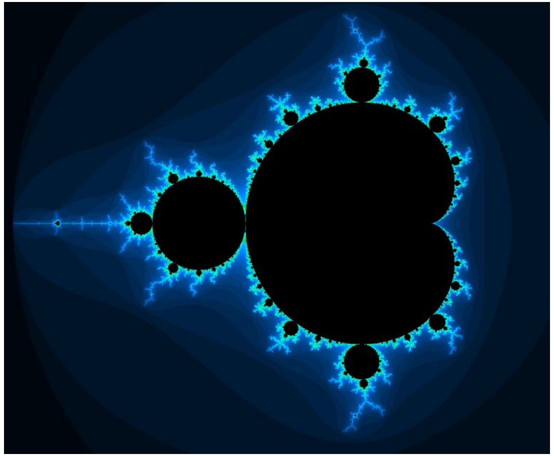
# What are algorithmic skeletons?

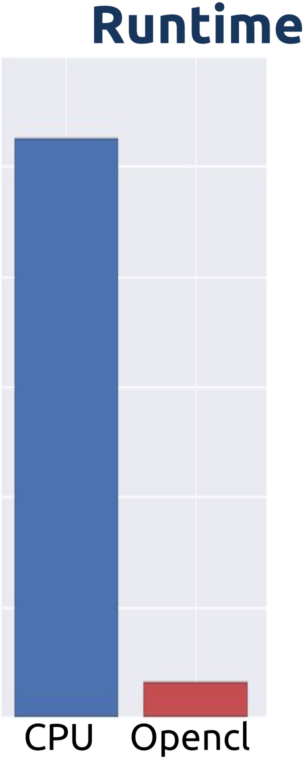
**Parallel** implementations of common **patterns** of computation.

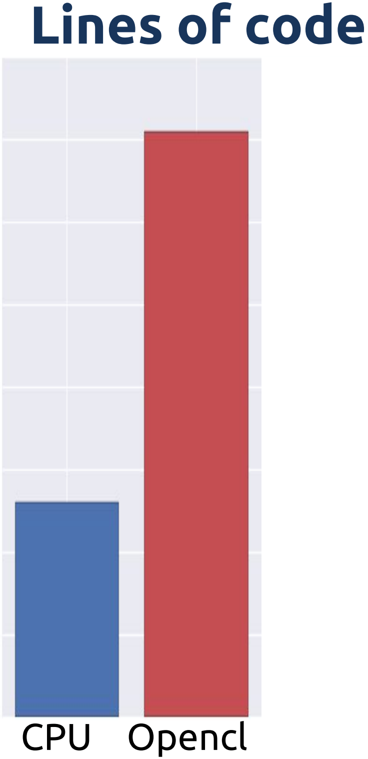


# Why do we need them?

Heterogeneous parallelism offers massive **performance**.





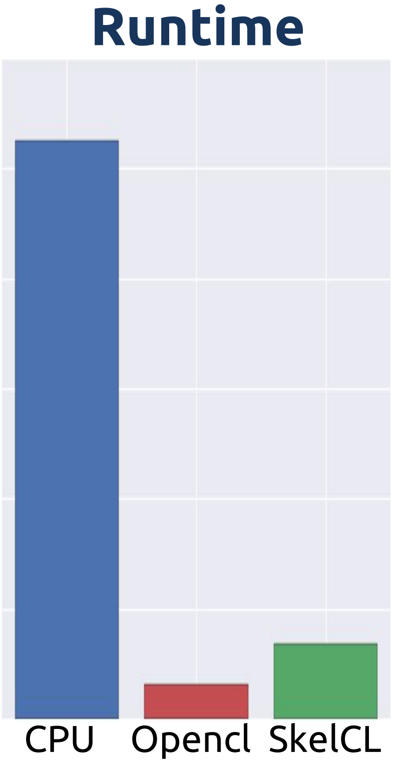


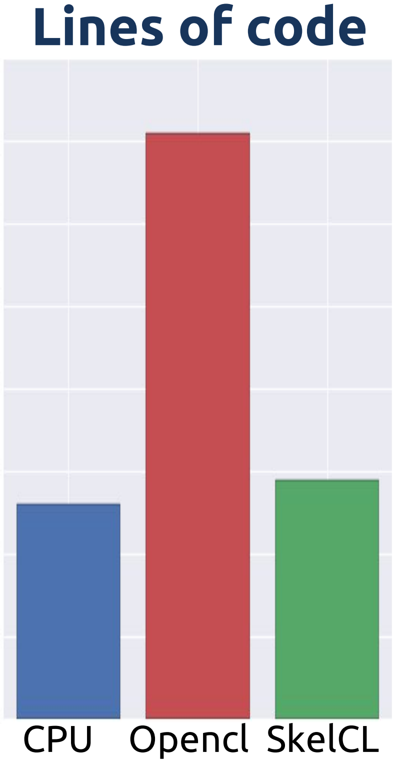
# Why do we need them?

Heterogeneous parallelism offers massive **performance**.

Algorithmic Skeletons offer **ease of use**.

For both performance **and** ease of use, we need autotuning.


Runtime


Lines of code

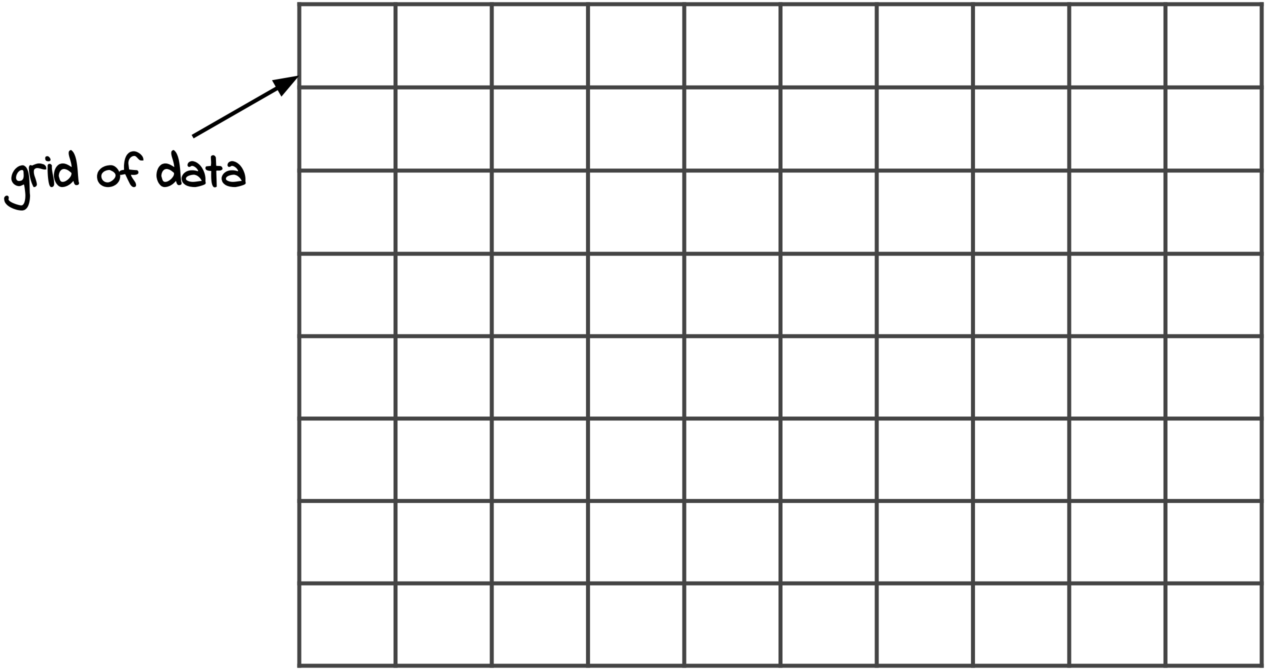
# My Project

Demonstrate **dynamic autotuning** of algorithmic skeletons.

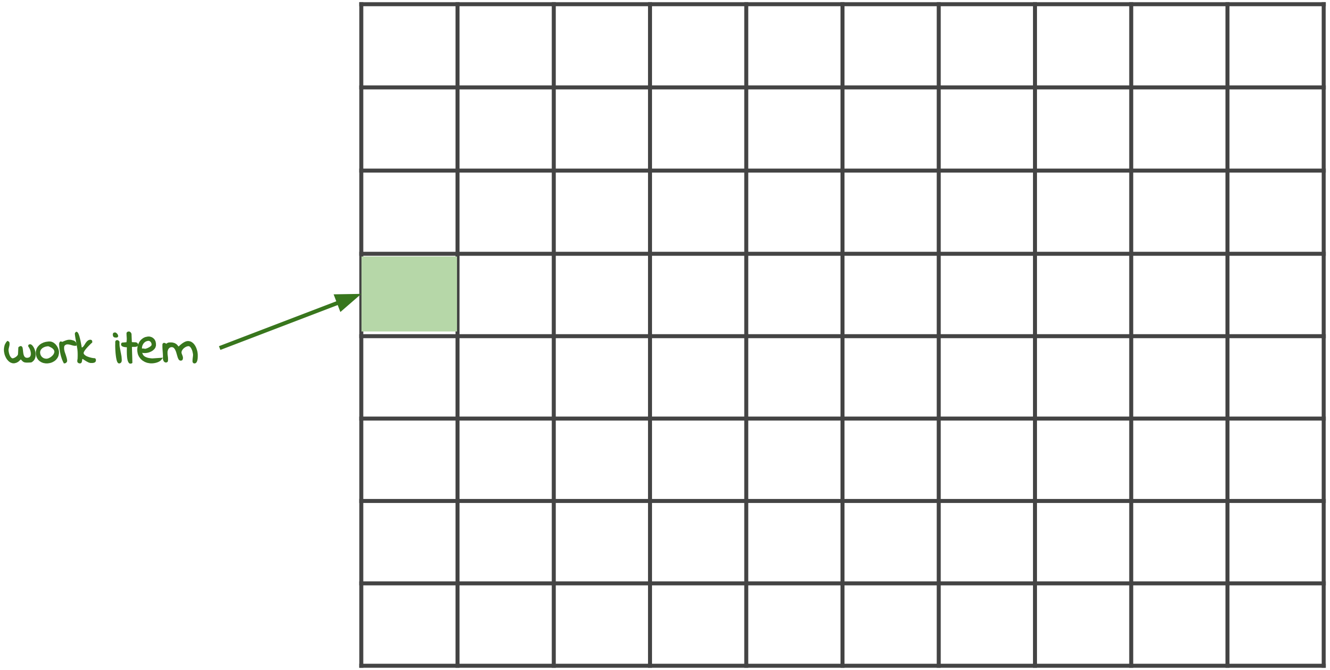
Using the **SkelCL** data-parallel skeleton library.

Targeting **Stencils** applications on GPUs and CPUs.

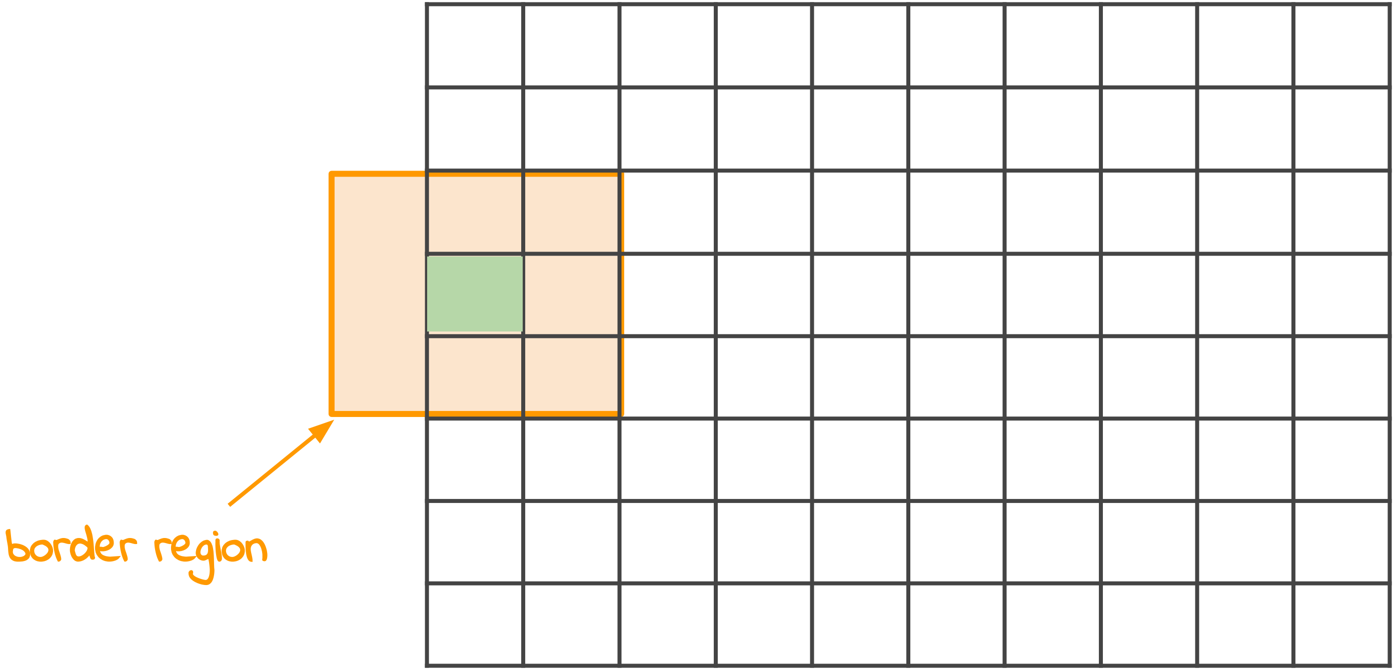
# Stencil codes



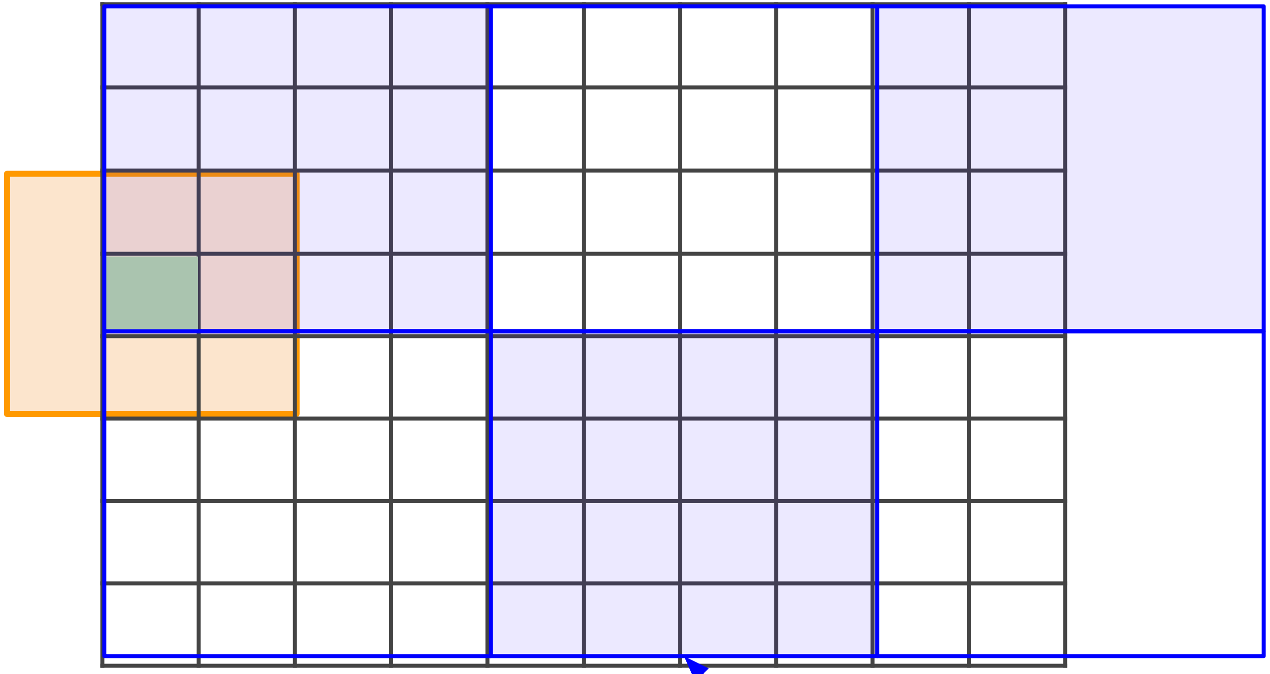
# Stencil codes



# Stencil codes

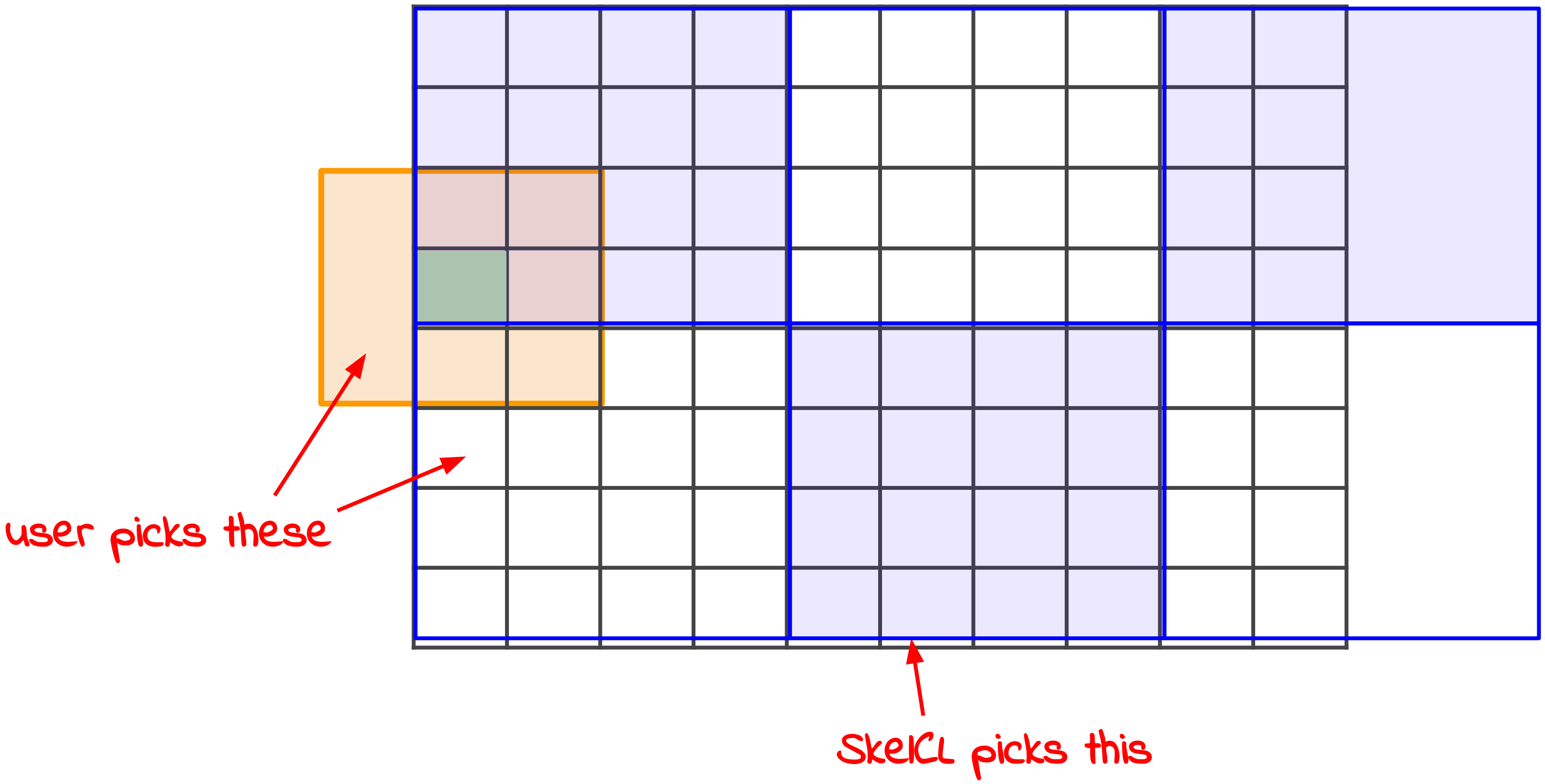


# Stencil codes



divided into workgroups

# Stencil codes



# Exploring optimisation space

Performance of workgroup size depends on:

| | |
|---|---|
| Program | Shape of border region, static instruction counts, … |
| Hardware | Local memory capacity, num processors, … |
| Dataset | Number of elements, data types, … |

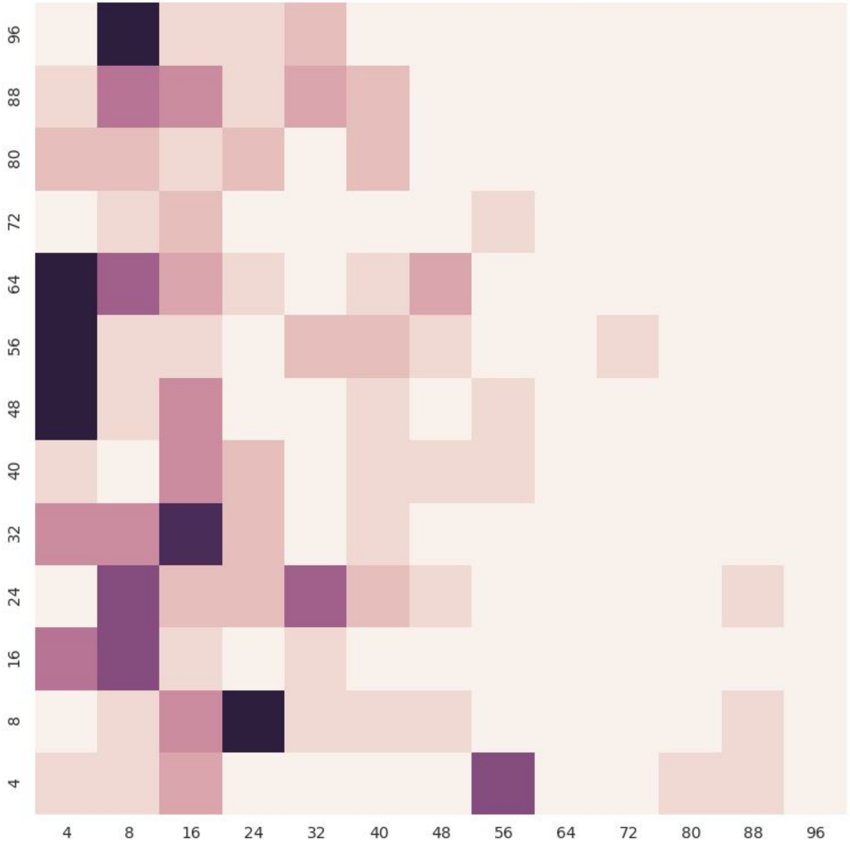
**How can we test this?**

Try a bunch of **synthetic** workloads.

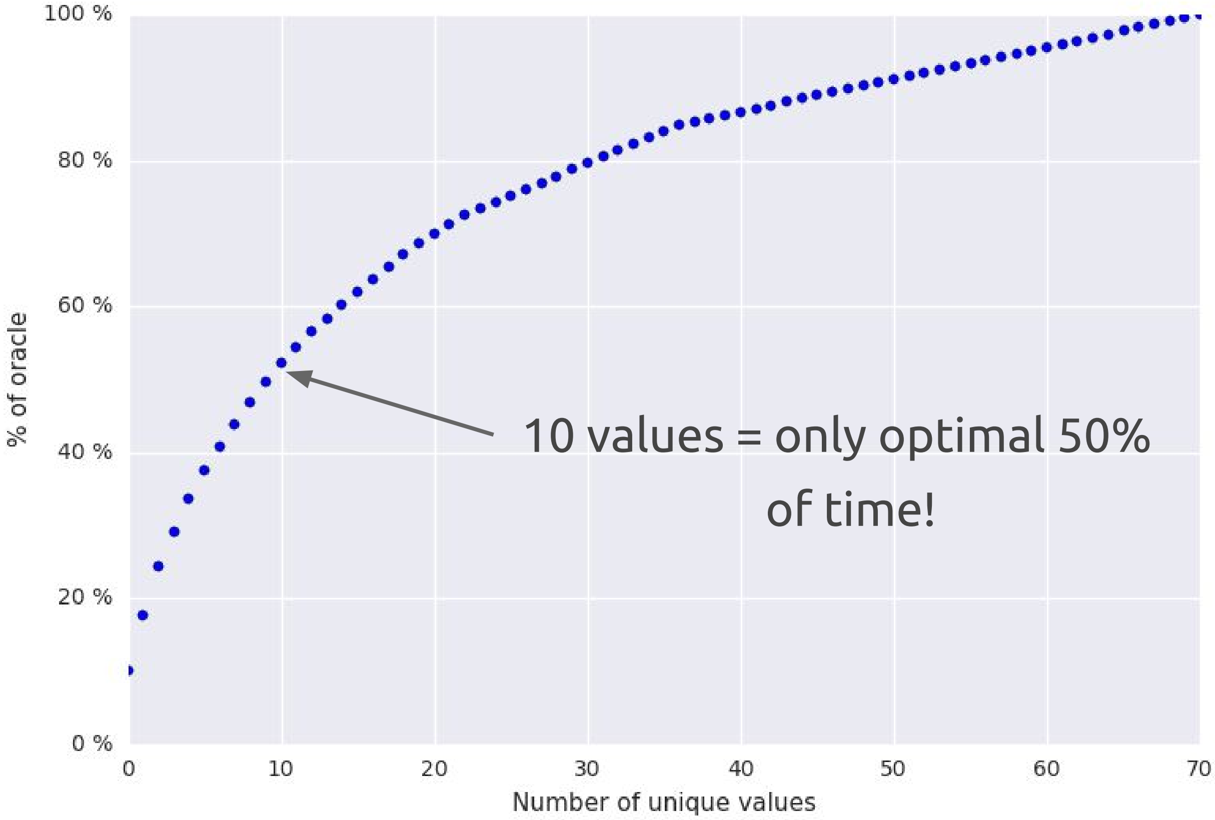
Measure runtime of different workgroup sizes, compare **performance**.

# Distribution of best values

No silver bullet!

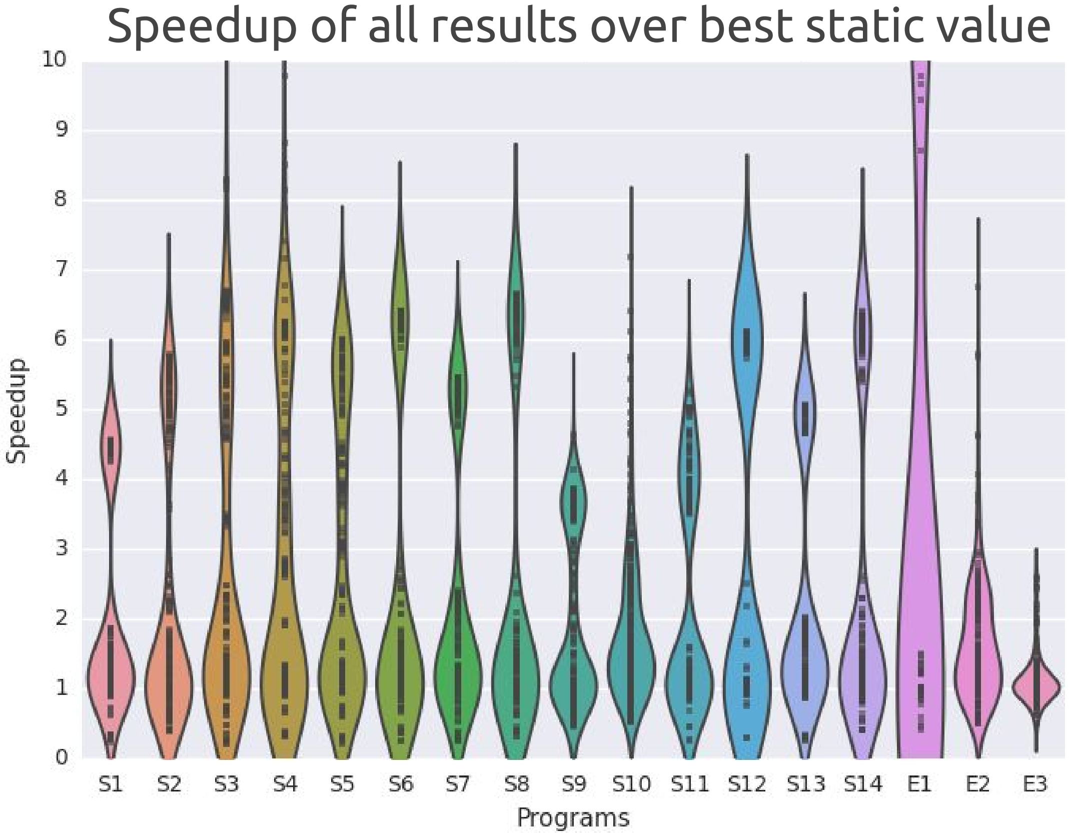


# How many values do we need?



10 values = only optimal 50% of time!

# What's the best we can do *statically*?



# Autotuner design

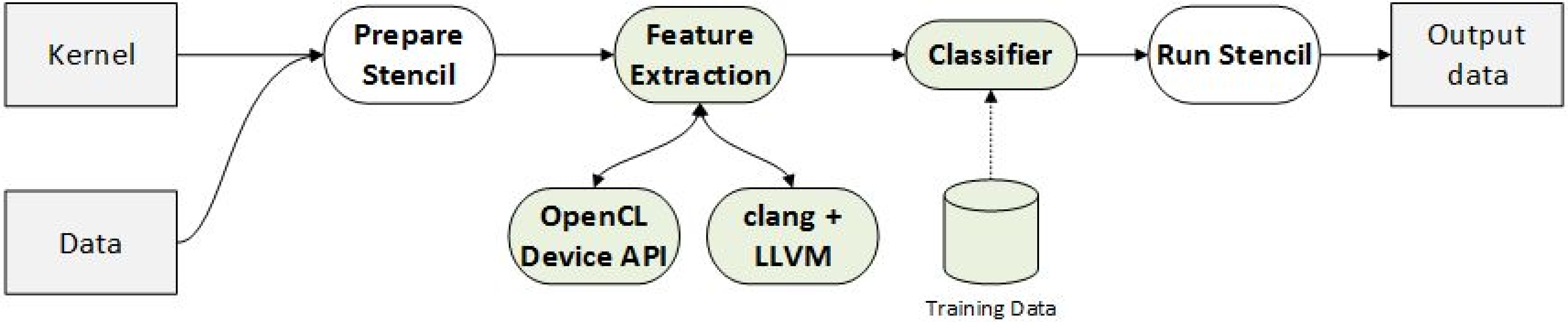
Extract **features** from hardware, program, and dataset.

Use best workgroup sizes as **training data**.

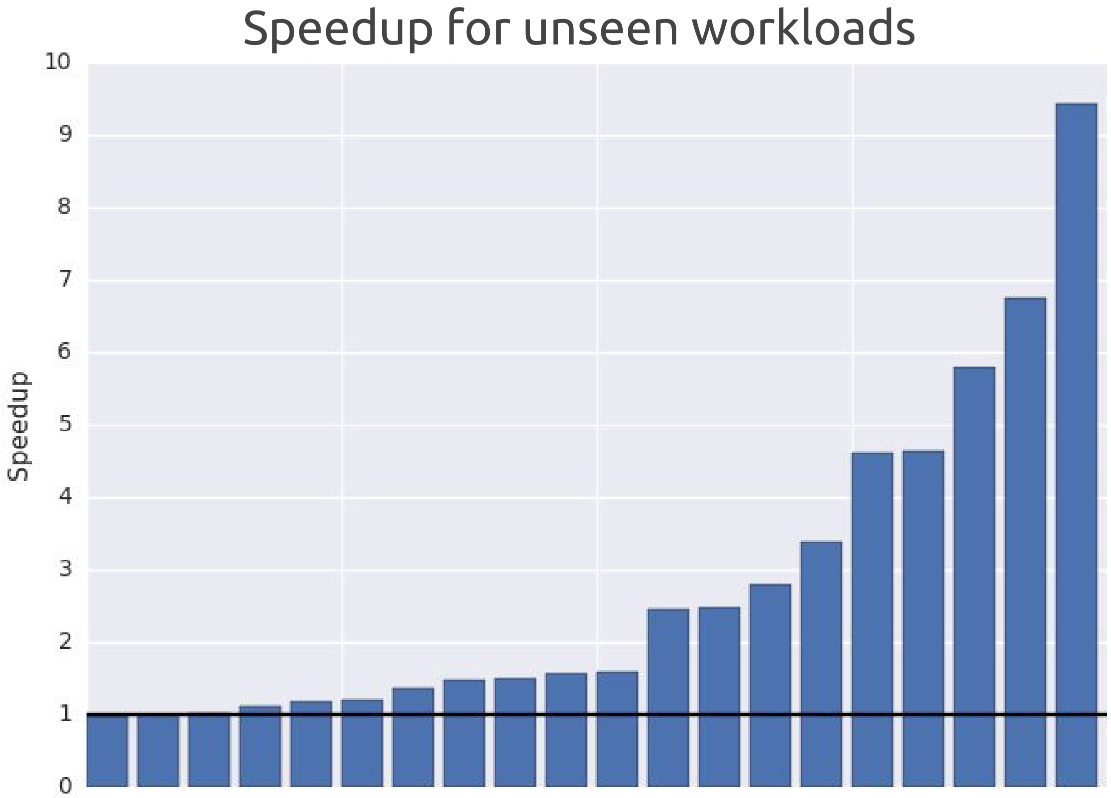
Use **machine learning** to predict:

*f(features) -> (workgroup size)*

# Autotuner design

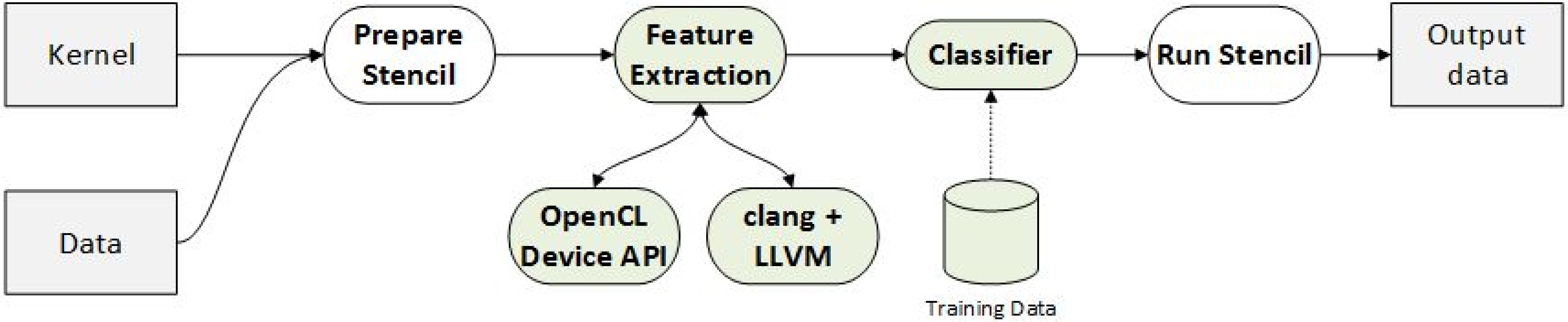


# Autotuner performance

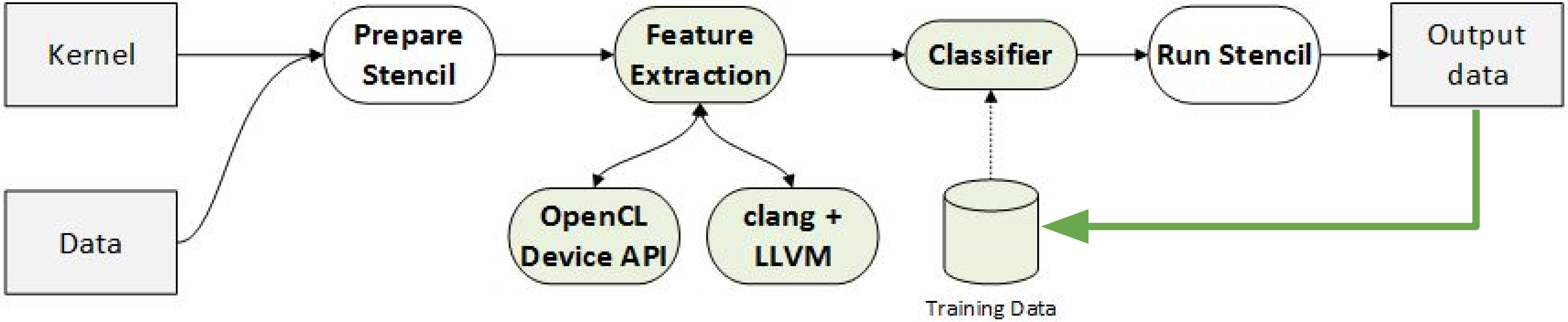


Mean: **2.82x**

# Moving Forward



# Moving Forward



# Conclusions

High level skeletons needed for complexity of GPU programming.

Values used to parameterise these skeletons offer 10x performance margin.

Synthetic benchmarks + runtime features + machine learning
= **2.8x** performance improvement of real programs.