Chris Cummins

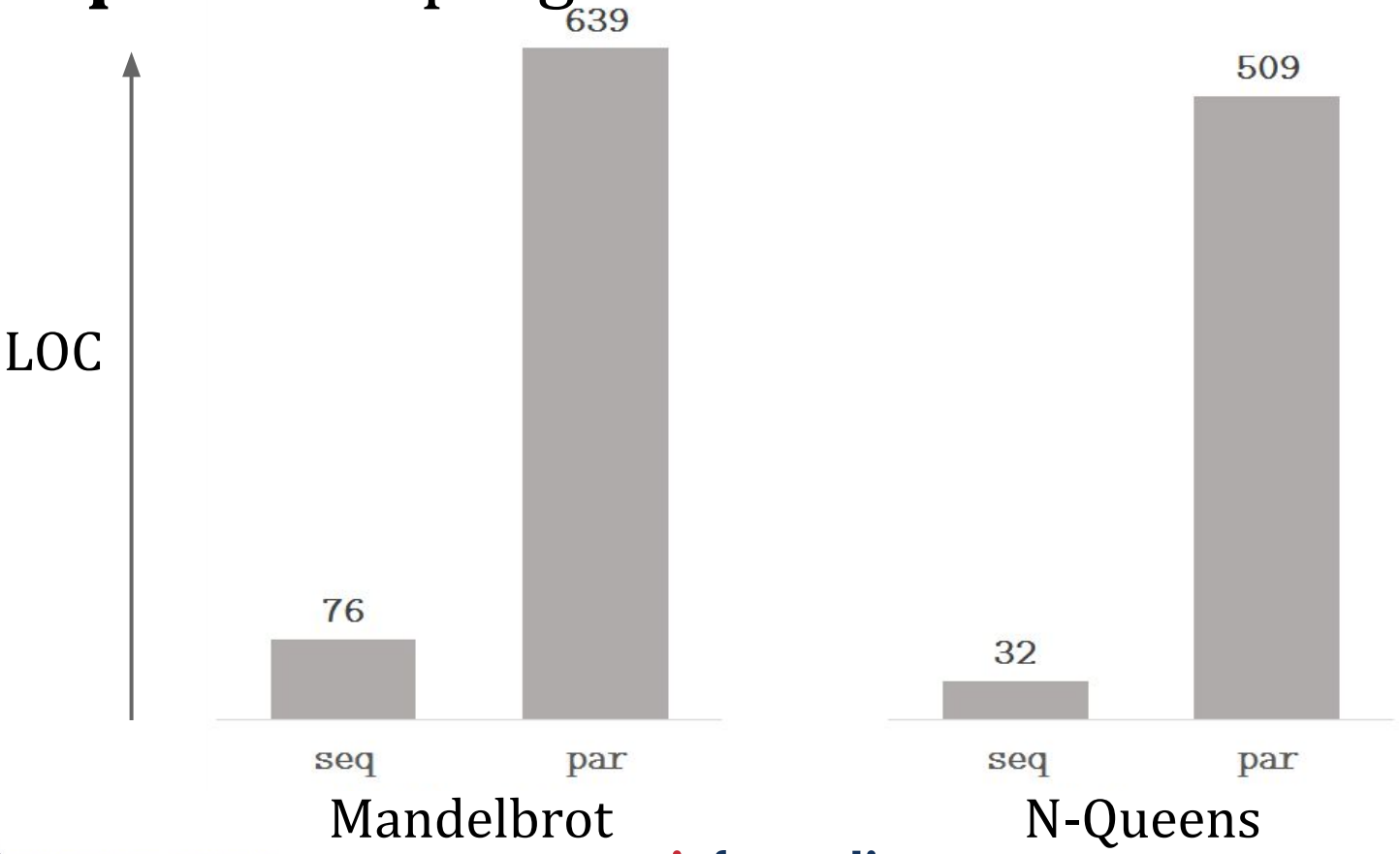
MEng Electronic Eng. & Computer Science

Intel Open Source Technology Center

Research: Automated Parallelism

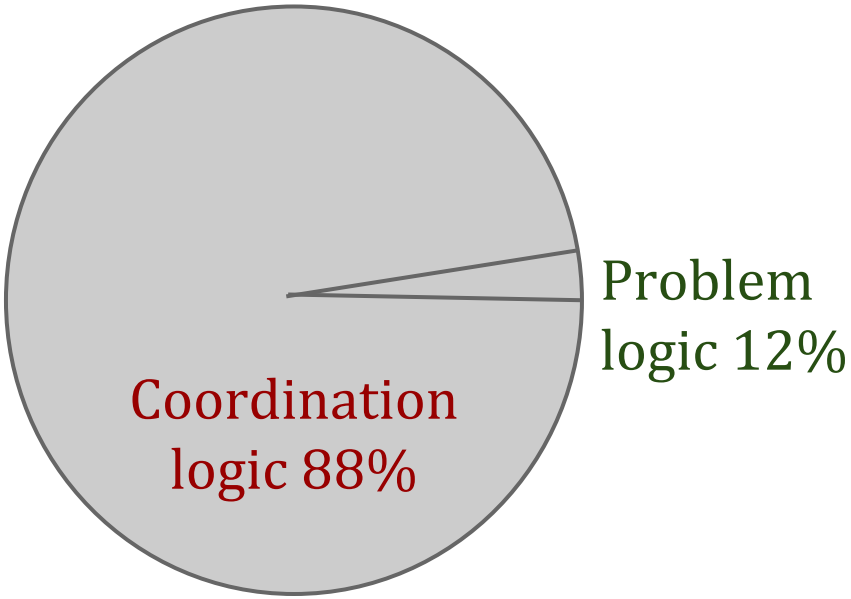
The problem with parallelism

Typically 10-20x more code for functionally **equivalent** programs.

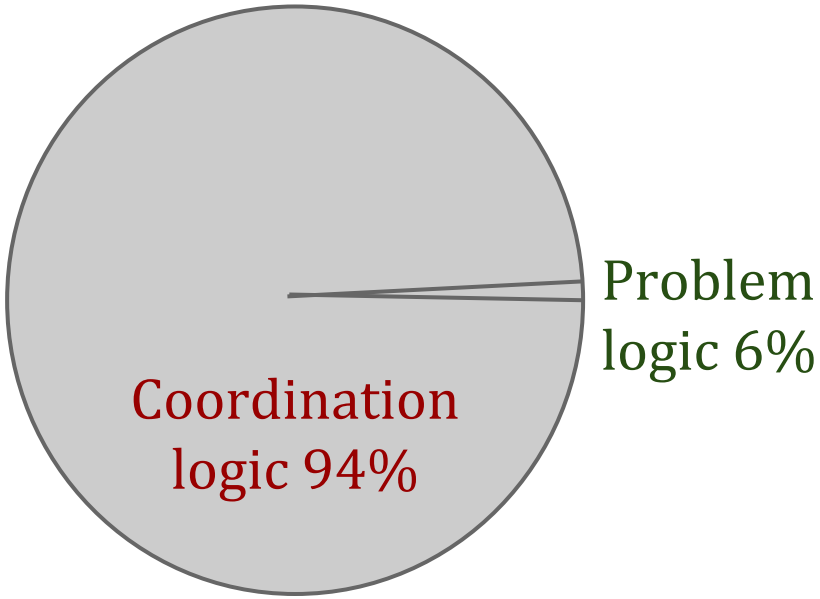


The problem with parallelism

How much of that is **relevant** to the problem?



Mandelbrot



N-Queens

Algorithmic Skeletons

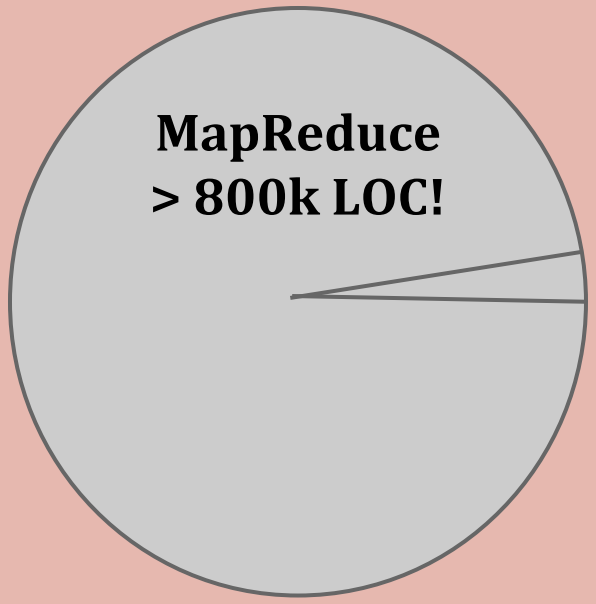
Programmers repeatedly use generic **patterns**.

Abstract these patterns, and **parallelise** them.

Let users provide the **problem logic**.

Algorithmic Skeletons

Skeletal domain



Problem domain

Web indexing.

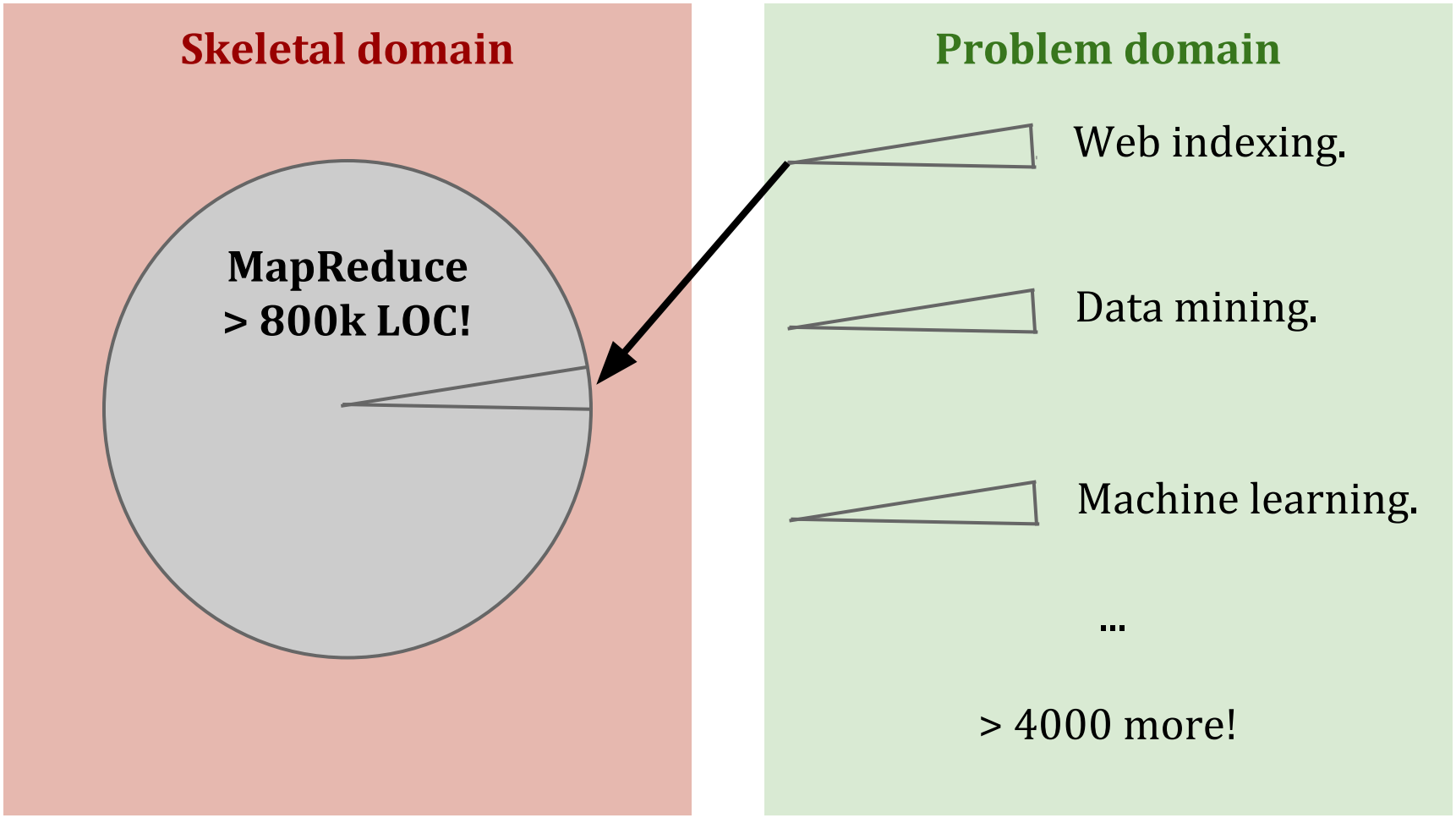
Data mining.

Machine learning.

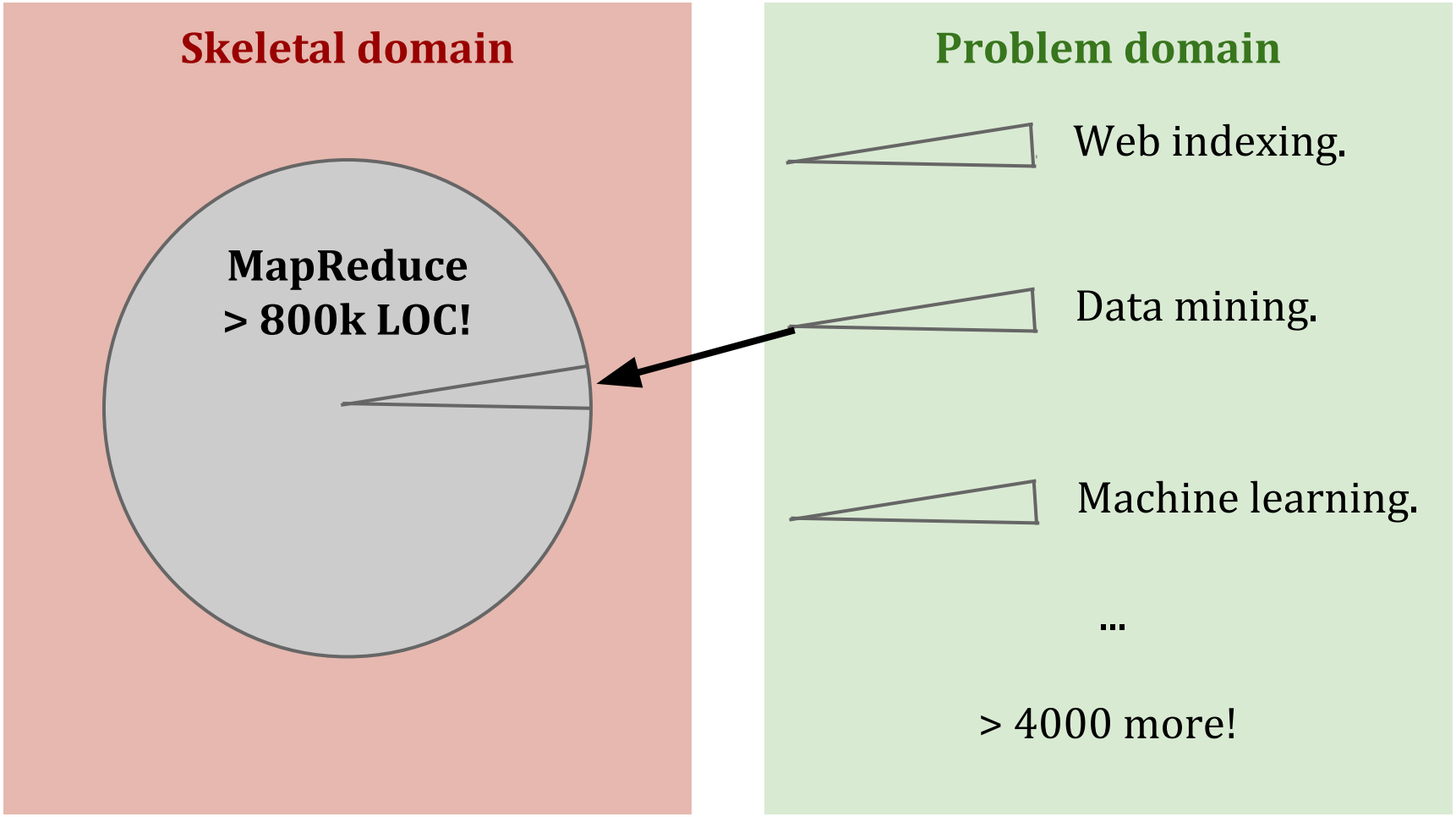
...

> 4000 more!

Algorithmic Skeletons



Algorithmic Skeletons



Algorithmic Skeletons

How do we **expose** these abstractions?

- New languages?
- Tool boxes and libraries?

How much do we **automate**?

- Completely black box?
- User-defined parallelism?

How do we get people to **use** skeletons?

- High Performance?
- Ease of Use?