# SIMULATE & DUPLICATE

or

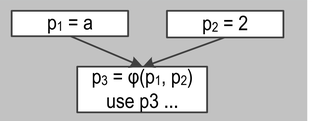
*"Simulation-based Code Duplication for Enhancing Compiler Optimizations"*

David Leopoldseder
david.leopoldseder@jku.at

JOHANNES KEPLER UNIVERSITY LINZ

## Problem

⇒ Control-flow **prohibits** many optimizations




## Solution: Code Duplication

⇒ **Copy** merge block into **predecessors**

⇒ **Eliminate** merges **=>** Increase **code size**

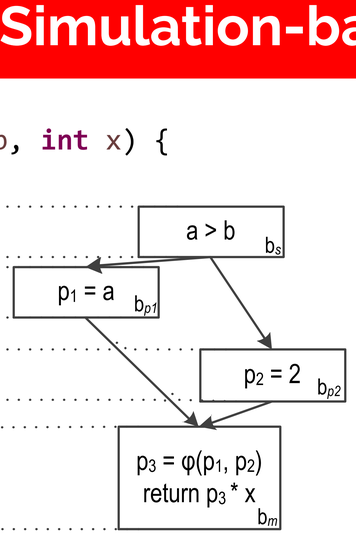
## Cost/Benefit Analysis

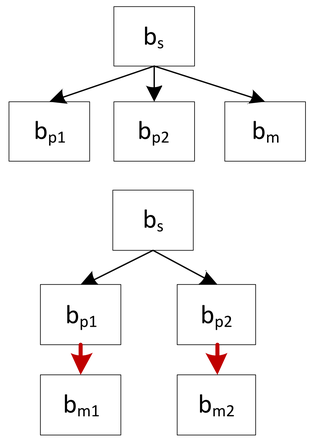
⇒ Find **opportunities prior** to duplication

⇒ Duplicate as **little as possible**

## Approach: Simulation-based Code Duplication

```
int f(int a, int b, int x) {
  int p;
  if (a > b) {
    p = a;
  } else {
    p = 2;
  }
  return p * x;
}
```

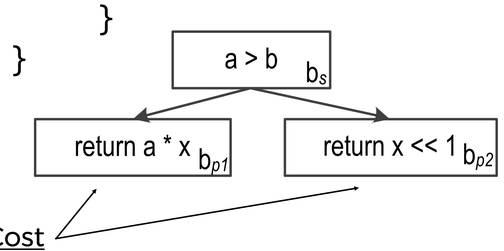






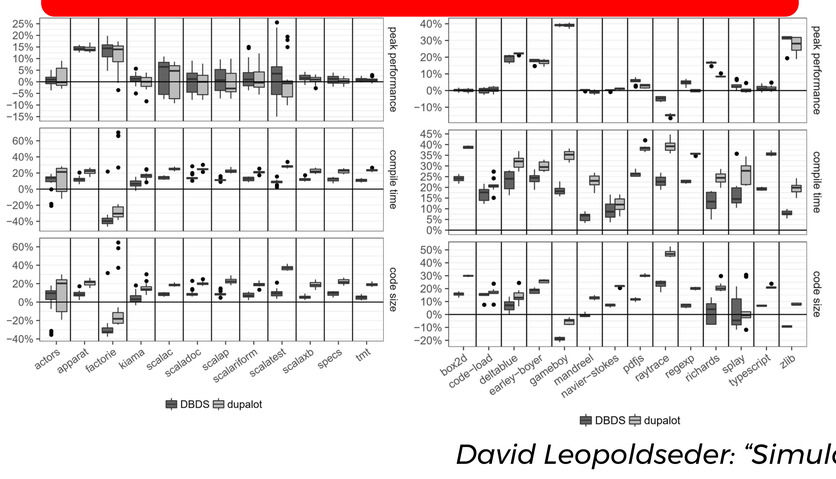
```
int f(int a, int b, int x) {
    if (a > b) {
        return a * x;
    } else {
        return x << 1;
    }
}
```




1. **Simulate** Duplication
2. Find Optimization **Opportunities** & **Trade-Off**
   => **Quantify** Opportunities
   => **Cost / Benefit Analysis:** Code Size vs Peak
3. Duplicate & **Optimize**

**Why simulate ?**
=> No backtracking
=> Make compile-time overhead feasible
=> No cleanup necessary

## Performance




## Try it out now !

graalvm.github.io

*David Leopoldseder: "Simulation-based Code Duplication for Enhancing Compiler Optimizations".* **SPLASH '17**