Exploring adaptive program behavior

Bonnichsen, Lars Frydendale; Probst, Christian W.

Publication date: 2014

Document Version
Publisher’s PDF, also known as Version of record

Citation (APA):

DTU Library
Technical Information Center of Denmark
Exploring adaptive program behavior
Lars Frydendal Bonnichsen and Christian W. Probst
Technical University of Denmark
Project number #295440

Motivation
▶ Modern computer systems are ever more complex.
▶ Their bottlenecks can change at run time.
▶ The dynamic nature makes it more difficult to make good optimization decisions.
▶ Adaptive behavior has been suggested as a solution.
▶ Decisions are made at runtime by switching between program versions.

Version: implementation of part of a program.

Adaptivity framework
▶ We will optimize based on feedback, both at run time and compile time.
▶ We will use a uniform design to explore adaptive program behavior, as outlined in below:

Adaptivity
Evaluates behavior of
Program
Adaptivity
Adapts behavior of
System state
System
Evaluated on

Adaptivity operation
▶ The adaptivity block decides when to adapt, selects a version to use, and switches to it.
▶ The actions form a wide design space.

Design space exploration
▶ We characterize the adaptivities design space by 4 factors.


Approach
▶ There are different costs and gains for different forms of adaptive behavior:
▶ For instance, recompilation, dispatchers, and iterative compilation.
▶ We want to integrate and evaluate different forms of adaptive behavior.
▶ Our strategy has 3 steps:
1. Design an extensible adaptivity framework.
2. Expose adaptivity uniformly.
3. Use the extensible nature of the framework to support a wide range of adaptive behavior.

Contributions
▶ Designed an extensible adaptivity framework.
▶ Implemented a function level adaptivity framework.
▶ Outlined how the framework can be expanded to the whole design space.
▶ Defined criterias to evaluate adaptive program behavior.

Future work
▶ Using the outlined structure, we will integrate different forms of adaptive behavior to determine:
1. How much can be gained from the adaptive behavior. (∼ 5 – 16% [1])
2. The cost of switching versions.
3. The overhead of supporting switches. (Best case ∼ 2% [3])
4. How should versions be selected.
At a high level, we want to determine when the benefits of adaptive behavior outweigh the disadvantages.

Related Work

Conclusion
▶ Adaptive behavior allows for further optimization, but it comes at a cost.
▶ We need to determine when adaptive program behavior is worth the cost.
▶ We have a plan how to integrate different forms of adaptivity.