Detecting Variable-Length Phases in Utility Programs

Xipeng Shen, Chen Ding, Sandhya Dwarkadas, and Michael L. Scott

Department of Computer Science, University of Rochester

INTRODUCTION

Utility Programs

Motivations

The benefit of the adaptation depends on the accuracy of phase partitioning and behavior characterization.

Digital Alpha machine.

Utility Programs pass big challenges to previous techniques.

CoV ( % )

Limitations of Previous Methods

Examples:

Digital Alpha machine.

Adaptive Profiling: Convert Challenges to Opportunities

EVALUATION

Methodology

Comparison with Procedure and Interval Phases

Finding Phase Markers

Active-profiling, procedure, interval and interval-cluster phase CoVs

Finding phase markers for regular inputs

Evaluation on GCC

A phase, especially the outermost phase, often represents a memory usage cycle, in which temporary data are allocated and then collected.

Uses in Memory Management

Adaptive Profiling: Convert Challenges to Opportunities

Adaptive Profiling: Convert Challenges to Opportunities

Comparisons

A two-step technique for detecting input-dependent phase behavior in utility programs.

Conclusions

Finding phase markers for regular inputs

Comparisons

A two-step technique for detecting input-dependent phase behavior in utility programs.

Conclusions

Comparisons

A two-step technique for detecting input-dependent phase behavior in utility programs.

Conclusions

Comparisons

A two-step technique for detecting input-dependent phase behavior in utility programs.

Comparisons

A two-step technique for detecting input-dependent phase behavior in utility programs.

Comparisons

A two-step technique for detecting input-dependent phase behavior in utility programs.