

Whole Program Paths

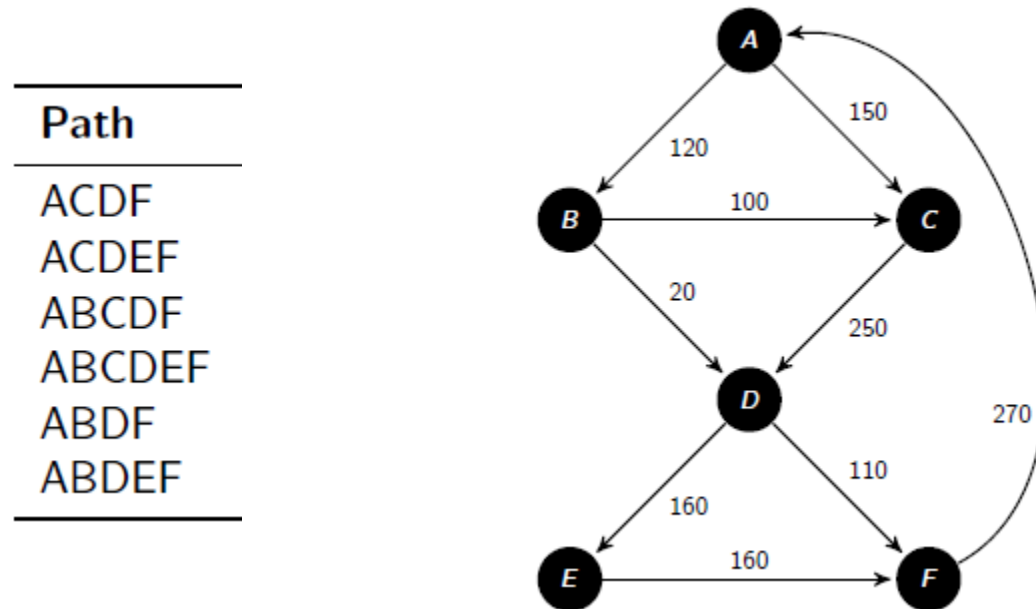
James R. Larus
Microsoft Research
(PLDI'99)

Presented by Haipeng Cai

10/06/2015

What is a *path*

- A dynamic control-flow trace (of a procedure)
- Representation: sum of edge encoding (int)



What is a *whole program path*

- A dynamic control-flow trace (of the entire program)
- Representation: a context-free grammar (as a DAG)

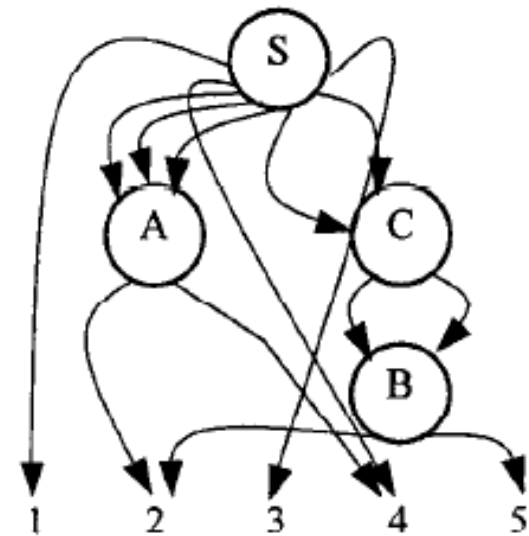
14242424
252525253

$S \rightarrow 14AAACC3$

$A \rightarrow 24$

$B \rightarrow 25$

$C \rightarrow BB$

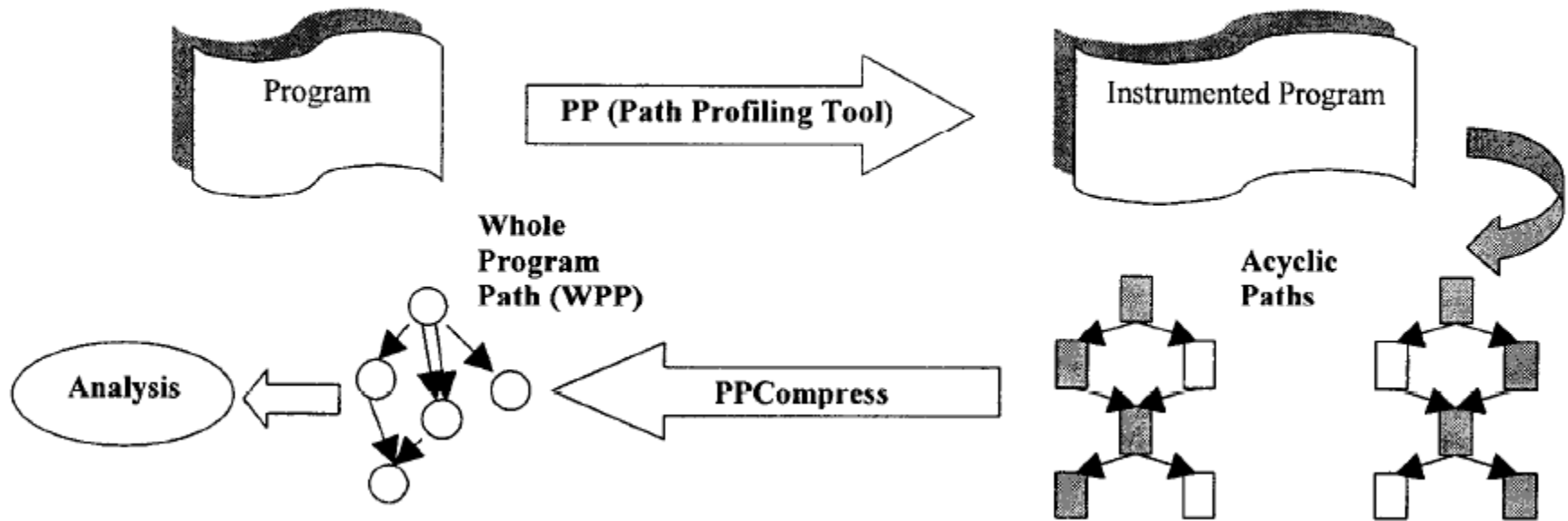




Path Profiling vs Whole Program Path

- EPP
 - Acyclic paths: approximation only for loops
 - Intraprocedural profiling
- WPP
 - Path crossing procedure boundaries
 - Interprocedural profiling

WPP profiling: overview





Phase 1: collect acyclic path traces

- A sequence of opcode-operand pairs

OpCode(Operand)	Meaning
<code>EnterRoutine (ID)</code>	Subsequent paths execute in routine ID
<code>LeaveRoutine ()</code>	Leave current routine and return to previous one
<code>NewPath (ID)</code>	Path ID executed
<code>EnterThread (ID)</code>	Subsequent paths execute in thread ID

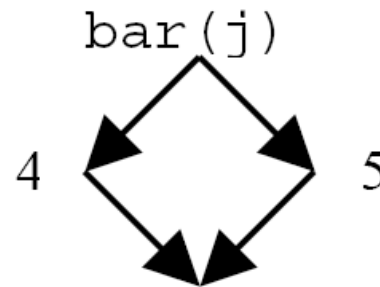
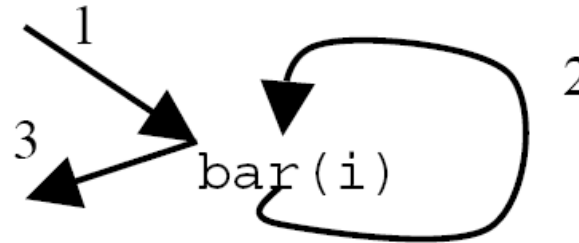
Phase 1: collect acyclic path traces

Code

```
for (i=0; i<9; i++)  
  bar(i);
```

```
int bar(int j) {  
  if (j < 5)  
    return j;  
  else  
    return 0;  
}
```

Paths



Acyclic Path Trace

14242424
25252525253

Phase 2: trace compression

- Explain output string as context-free grammar:

- Efficient compression
- Automatic subsequence grouping

Acyclic
Path Trace

SEQUITUR
Grammar

- Grammar creation

- Append symbols to start rule
- Diagrams appear at most once
- Rules must be used at least twice

14242424
25252525253

$S \rightarrow 14AAACC3$

$A \rightarrow 24$

$B \rightarrow 25$

$C \rightarrow BB$



Phase 2: trace compression

- Enhanced SEQUITUR
 - Looking ahead a symbol before introducing a new rule
 - Trying to apply existing rules

Phase 2: trace compression

- SEQUITUR original

string: 11111211111

Start Rule

S → 1111

S → AA1211

S → AA12A1

S → AB2BA

Action

create A → 11

apply A → 11

create B → A1

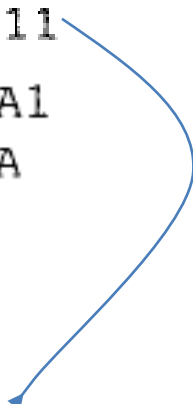
- SEQUITUR (1)

S → AA12AA

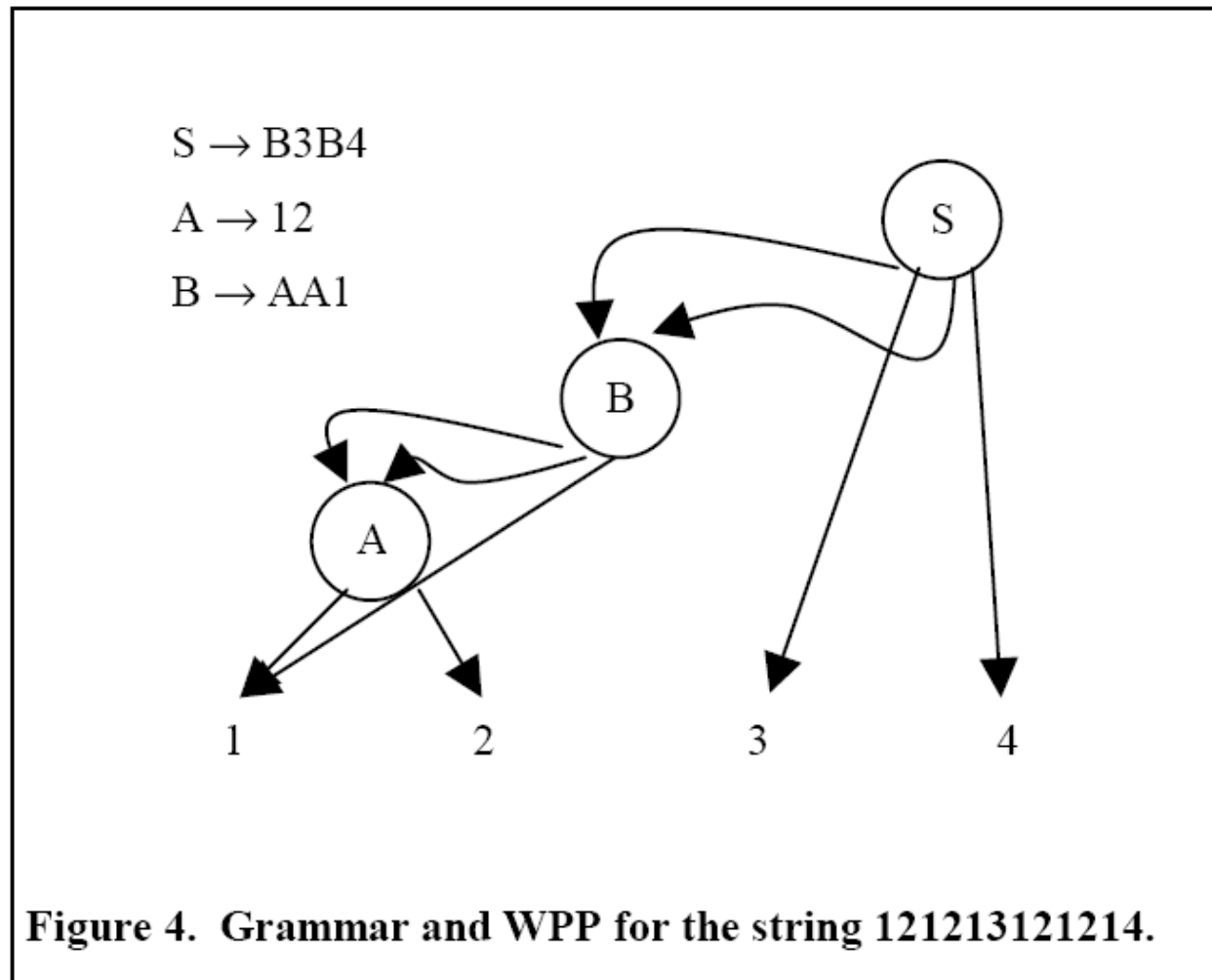
S → C2C

A → 11

C → AA1



Phase 2: trace compression





Application of WPP: hot subpaths

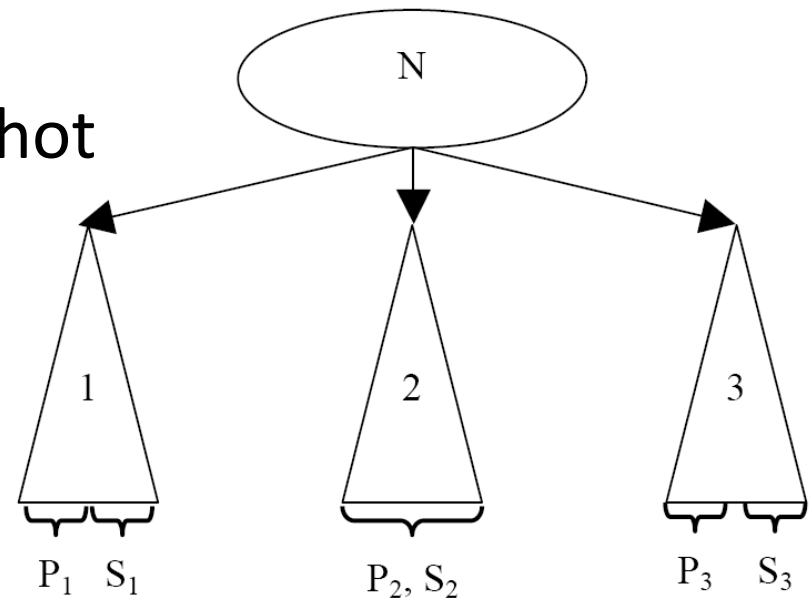
- A hot subpath
 - A sequence of L or fewer consecutively executed acyclic paths that incur a cost of C or more.
- A minimal hot subpath
 - The shortest prefix of a subpath incurring cost C or more

Application of WPP: hot subpaths

- Cost measurement
 - Subpath cost = execution frequency x sum of costs of constituent acyclic paths
 - Single acyclic path cost = number of instructions along the acyclic path

Application of WPP: hot subpaths

- Find hot subpaths
 - Post-order traversal of WPP (DAG)
 - Recursively find minimal hot subpaths at descendant levels
 - Examine concatenated descendant subpaths



Evaluation: WPP

- Programs
 - 8 SPECINT95 benchmarks
 - Two commercial apps
- Metrics
 - Computation time
 - Storage cost

Evaluation: WPP

Benchmark	Time (sec)	Trace Size (MB)	Trace/Sec	WPP Size (MB)	WPP/Sec	Trace/WPP	Num Threads	Num Acyclic Paths	Num Rules	Rules/Path
099.go	90.1	2176.6	24.15	141.1	1.57	15.4	1	17,321	2,760,820	159.4
124.m88ksim	3.0	115.0	38.33	0.3	0.10	392.8	1	1,169	7,927	6.8
126.gcc	9.0	254.3	28.25	23.7	2.64	10.7	1	20,739	489,287	23.6
129.compress (train)	0.0	8.3	22230.90	0.2	632.59	35.1	1	364	5,857	16.1
130.li	4.0	300.4	75.08	2.6	0.64	116.9	1	966	62,076	64.3
132.jpeg	3.0	47.8	15.94	6.6	2.19	7.3	1	1,637	136,816	83.6
134.perl (jumble)	17.0	605.0	35.59	15.0	0.88	40.3	1	2,115	238,893	113.0
147.vortex	48.0	1598.8	33.31	6.6	0.14	241.9	1	5,310	136,269	25.7
SQL	120.0	628.7	5.24	21.1	0.18	29.9	22	193845	404110	2.6
WinWord	8.0	73.3	9.20	6.8	0.85	10.8	4	54254	139073	2.7

Evaluation: WPP

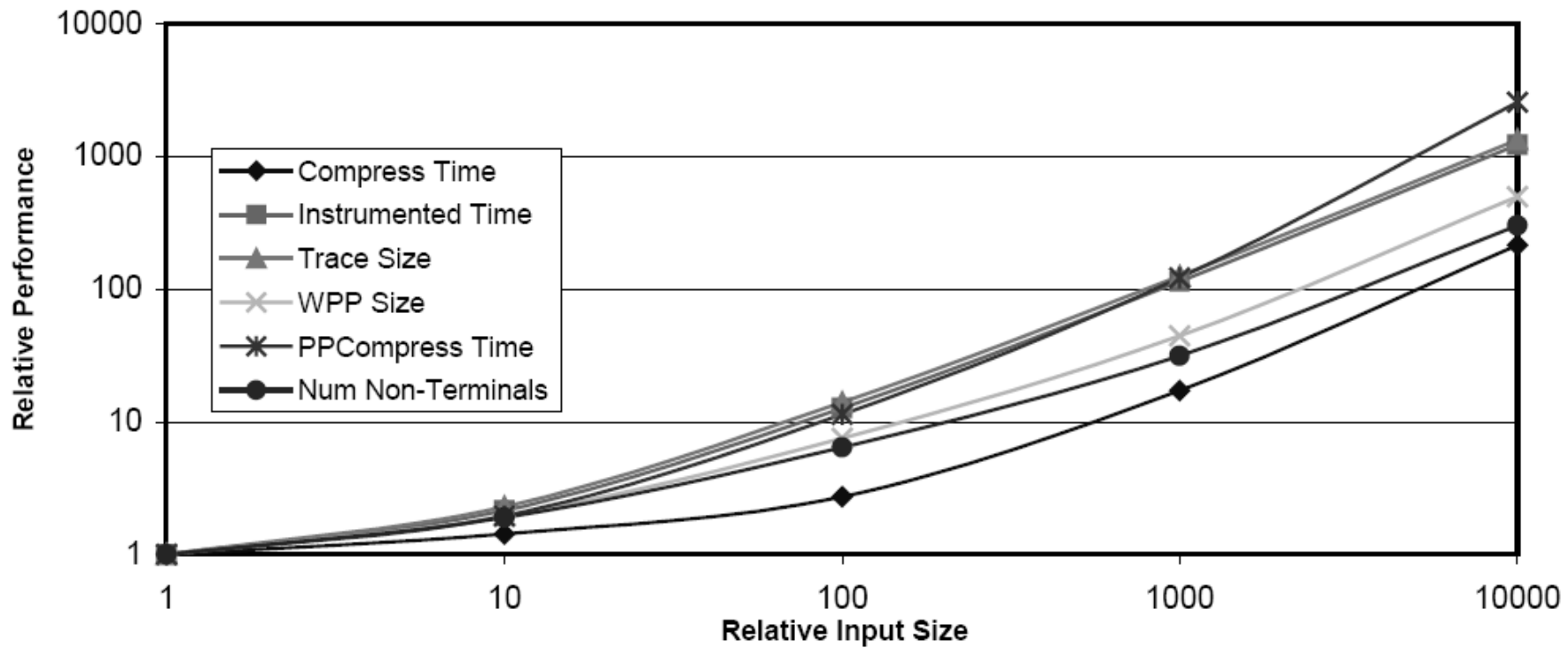


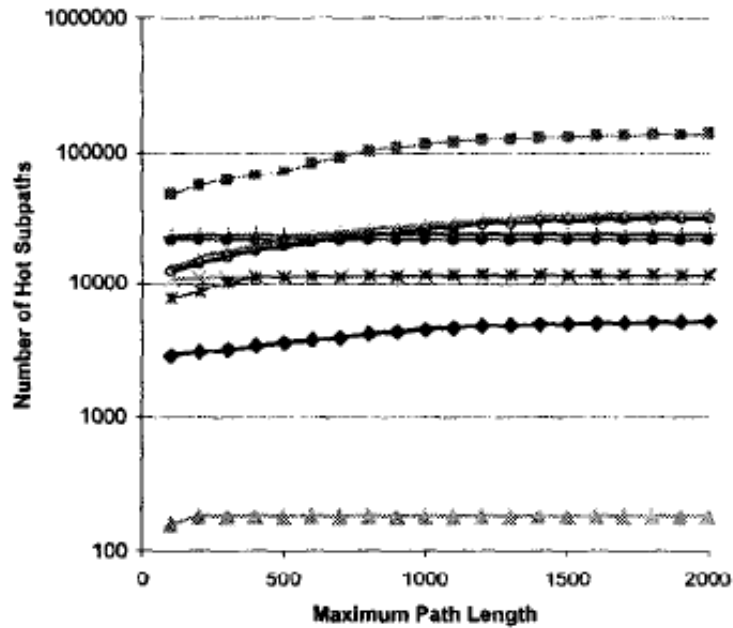
Figure 8. Whole Program Path performance running compress benchmark with various size input files.



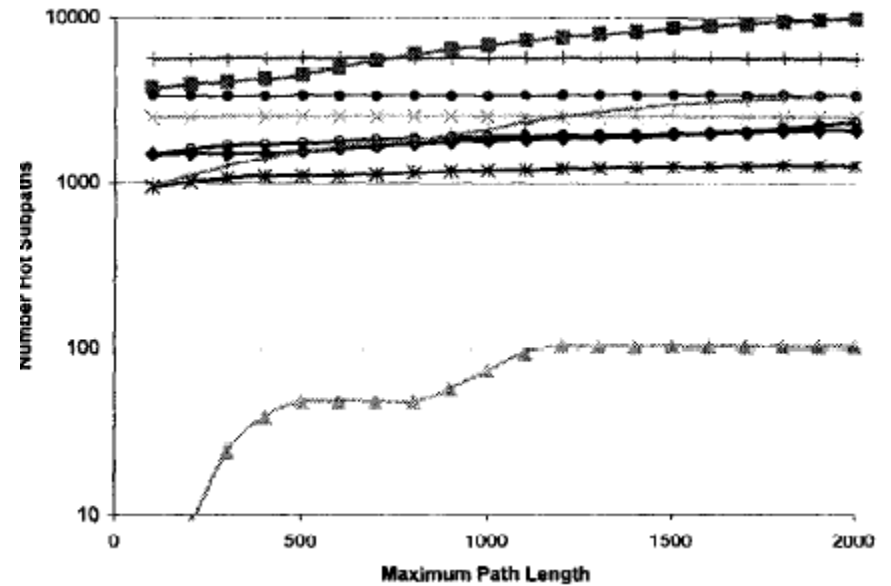
Evaluation: hot subpaths

- Same benchmark programs
- Metrics
 - Maximum hot subpath length
 - Number of minimal hot subpaths
- Compare between two cost thresholds
 - 10,000
 - 100,000

Evaluation: hot subpaths



C = 10,000



C = 100,000



Summary

- Contributions
 - Produce longer paths across loop and procedure boundaries
 - Compress traces for time and cost efficiency
 - Application: hot subpath identification
- Limitations
 - Runtime overhead
 - Profiling duration constrained by space limits
 - Control-flow profile only