7.0 Compilation based on dynamic information

7.4 Types of profiles

7.4.1 Frequency of method invocation

- absolute count
  - sampling identifies critical methods

  - gprof (unit profiler)
    (Intel: vtune)

* important extension
  absolute count of invocation in some context, e.g., f \rightarrow g, h \rightarrow g

* hot, high freq. method
Example - flow graph of some method pro
Looking inside a method

7.4.2 Frequency of basic block execution

- count (estimate?) for each bb
- "block profile"
- easily obtained by inserting counting instructions
  - could use sampling
  - subject to optimization
7.4.3 Frequency of transitions from Block $B_j$ to Block $B_i$

- "edge profile"
- weight for each edge in control flow graph
  (number of times edge traversed)
- blue in slide 2
- counters are needed
  - optimize certain opportunities
7.4.4 Frequency of paths taken

- how often are the paths executed

- sequence of basic blocks — implies edges

- "path profile"

- red in slide 2

\[ P_1 : B_1 - B_3 - B_5 - B_6 - B_8 - B_9 : 90 \]
\[ P_2 : B_1 - B_4 - B_5 - B_7 - B_8 - B_9 : 10 \]
\[ P_3 : B_2 - B_4 - B_5 - B_6 - B_8 - B_9 : 10 \]
\[ P_4 : B_2 - B_4 - B_5 - B_7 - B_8 - B_9 : 90 \]
Path profile (in general case) cannot be constructed from edge/block profile alone.

Additional counters are needed.

Potentially expensive.

- Maybe information on top k paths is sufficient for compiler.
- Limit interest data gathering to paths at length N.
- Reduce instrumentation.

Info could be useful for register allocation, scheduling.
7.5 Value profile

Value profile captures information on values used or computed in a program.

Consider loading a variable:
compute base address
load base + offset → register

\[
\text{movl} (\% \text{eax}, \% \text{edx}), \% \text{ebx}
\]

If offset is always "0": Simplify computation
no need for addition

obtain profile

```
0 1 2 3 4 ...
```

100% 35% 10% 5% other
use of value profiles...

- expect multiple uses of "offset" in a method
- specialize method bodies
  - offset == 0
  - offset != 0
  - decide at run-time on path to be taken