Exploring Large Profiles with Calling Context Ring Charts

Philippe Moret  Walter Binder
Danilo Ansaloni  Alex Villazón

University of Lugano, Switzerland
Profiling

- Analysis of program behavior
- Detection of hot spots
- Calling context profiling
  - Dynamic metrics for each calling context
  - Data structure: Calling Context Tree (CCT)
void f() {
    int i;
    for (i=1;i<=10;++i) {
        h();
        g(i);
    }
}

void g(int i) {
    int j;
    for (j=1;j<=i;++j) {
        h();
    }
}

void h() { return; }

P. Moret, W. Binder, D. Ansaloni, and A. Villazón
Exploring Large Profiles with Calling Context Ring Charts
Problem Statement

- CCTs for typical applications can be huge
  - Up to millions of nodes
  - Maximal depth up to 450
- Need a way to present the data to the developer
  - Support for exploring deep trees
  - Locate hot spots
Space Filling Visualization Techniques

- **Treemaps**
  - Rectangular layout
  - 100% space utilization
  - Size can be proportionnal to a metric for each represented node
  - Limited representation of hierarchy

- **Calling Context Ring Charts (CCRC)**
  - Circular layout, similar to Sunburst
  - Navigation by changing the root node
  - Hierarchy is well visualized
  - Zooming (changing tree depth)
Nodes of equal size

- **f()**
  - Invocations = 1
  - Bytecodes = 106

- **h()**
  - Invocations = 10
  - Bytecodes = 10

- **g(int)**
  - Invocations = 10
  - Bytecodes = 445

- **h()**
  - Invocations = 55
  - Bytecodes = 55
Angle proportional to bytecode consumption

- $f()$: Invocations = 1, Bytecodes = 106
- $g(int)$: Invocations = 10, Bytecodes = 445
- $h()$: Invocations = 55, Bytecodes = 55
Angle and area proportional to bytecode consumption

```
f()
  Invocations = 1
  Bytecodes = 106

h()
  Invocations = 10
  Bytecodes = 10

  g(int)
  Invocations = 10
  Bytecodes = 445

  h()
    Invocations = 55
    Bytecodes = 55
```
Demo

Setting:
- Aspect for CCT creation
- Number of executed bytecodes as metric
- Incremental updates sent every second through a socket
- Display is updated when an update packet is received
Conclusion

- Hierarchy represented using a circular layout
- Exploration of the tree by changing the root node
- Size of the segment according to a metric helps locating hotspots
- Future Work
  - Use a color scheme to represent additional data (other metrics, . . .)
  - Add an advanced search mechanism