May 2019

Performance Annotation Framework

Quentin Jensen  
*Western Washington University*

Chloe Dawson  
*Western Washington University*

Follow this and additional works at: https://cedar.wwu.edu/scholwk

Part of the Higher Education Commons

Large scale applications developers have many tools at their disposal to optimize and verify their software. One of which is Caliper, an annotation-based performance measurement tool. Caliper is very powerful and versatile, however, can be cumbersome to apply to complex applications. To solve this problem, we have created a framework to automatically prepare an application for performance measurement. This framework provides a layer of abstraction between the user and the source-code annotations and library linking. As a result, the process of measuring the performance of an application can be fully automated away—a huge step towards automatic software optimization.

**Goal**

Streamline the process of generating hardware performance data

**Approach**

1. **Project Code**
   - `file1.c`
   - `func1()
2. **Annotator**
3. **Caliperizer**
4. **Makefile**
   - `... goc -o file1.c ...
5. **output.cali** (sampled)
   - ```json
   {}
   ```

**Preliminary Results**

Data observations: We can see that for this sample data set, cache misses were lowest with input 1 and highest for input 3. This would indicate that input 1 is smaller and fits better within L1, L2, and L3 cache. For this program, we can see that all inputs for all functions have fewest cache misses for L3 cache, especially function 2. This indicates that the inputs are small enough to mostly fit within L3 cache.

**Future Plans**

Analyze hardware performance data of proxy applications for validation. Add functionality to annotator, such as inserting annotations around loops.