Enabling Lightweight Performance Analysis of Complex Scientific Workflows with PerfFlowAspect

Aliza Lisan*, Tapasya Patki†, Stephanie Brink†, Brian Gunnarson†, Hank Childs*

**University of Oregon*, †*Lawrence Livermore National Laboratory*

ABSTRACT

Artificial intelligence and/or Machine learning (AI/ML) are appearing increasingly often in scientific workflows on supercomputers, due to their ability to solve more complex problems. Such workflows create new requirements. Here, we consider changes necessary to support performance analysis of such workflows. In particular, we describe PerfFlowAspect, which approaches this problem via reduced instrumentation costs, support for multi-binary, multi-cluster workflows, and trace formats that support multiple workflow components running on heterogenous HPC infrastructures.

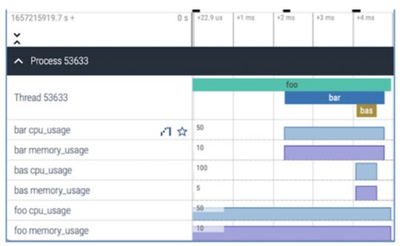
PROBLEMS WITH CURRENT TOOLS

- Lack of support for multi-binary, multi-cluster workflows and ensembles
- How much data to collect (sampling vs instrumentation)
- “Pollution” of application’s codebase
- Limited features for processing and visualizing trace formats from both traditional physics and AI/ML codes
- Lack of built-in support for multiple languages (AI/ML workflows often use Python in addition to C/C++)

PERFFLOWASPECT

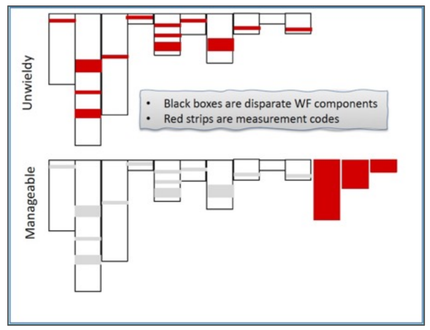
```python
@perfflowaspect.aspect.critical_path()
def foo(msg):
    time.sleep(0.001)
    if msg == "hello":
        return 1
    return 0
```

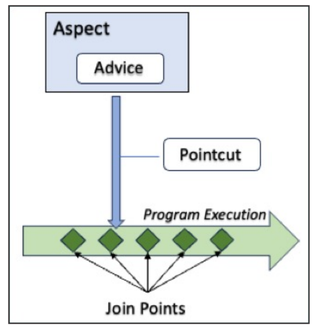
```cpp
__attribute__((annotate("@critical_path()")))
int foo(const std::string &str)
{
    usleep(1000);
    if (str == "hello"){
        return 1;
    }
    return 0;
}
```



- User-friendly and non-intrusive instrumentation
- Annotation-based, enabling selective analysis of interesting parts of the code in C/C++, Python, CUDA
- Support for coarse- and fine-grained analysis
- Support for composing data from disparate sources
- Output is human-readable and in Chrome Tracing Format
- Interactive visualization with Perfetto and Hatchet

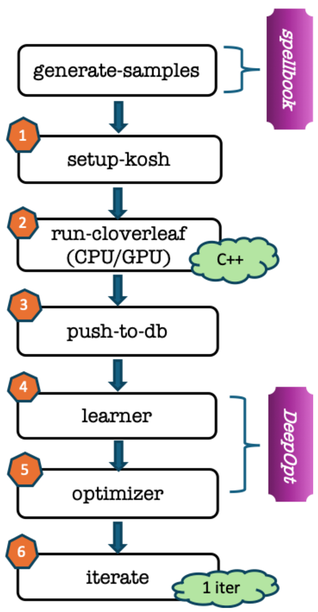
ASPECT-ORIENTED APPROACH





- AOP: programming paradigm that aims to increase modularity by allowing the separation of cross-cutting concerns.
- Enables customization with ‘actions’ or ’advices’: timing, performance counters, memory usage, etc. can be added.
- AOP enables single line annotation on functions.
- Implemented using LLVM Module Pass (C/C++) and Decorators (Python) by instrumenting functions.

DEMONSTRATION: ICECAP WORKFLOW



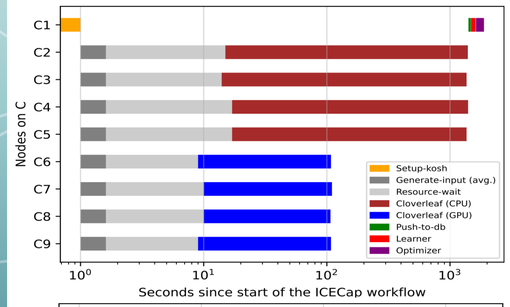
ICECap is a ML-assisted workflow demonstrating the optimization of a 17-parameter National Ignition Facility experiment. It is based on the Merlin framework, which is open-source and designed for large-scale ML workflows.

It is a **multi-binary** and **multi-cluster** workflow consisting of 6 steps involving:

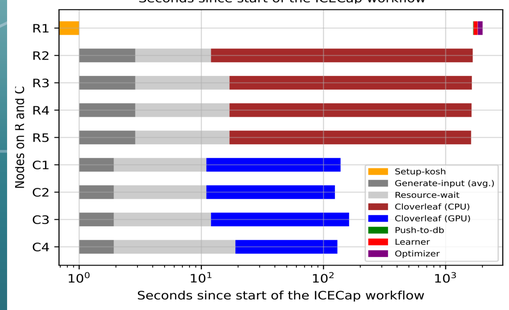
- 4 distinct Python programs
- 1 C++ application (**Cloverleaf**)
- 2 Python libraries (**DeepOpt**, **spellbook**)

ICECAP RESULTS

ICECap experiments were run on LLNL’s Corona (CPU+GPU) and Ruby (CPU-only) machines. Single cluster and multi-cluster experiments were performed, each with 9 nodes. 4 out of 8 Cloverleaf runs per experiment ran on the GPU.

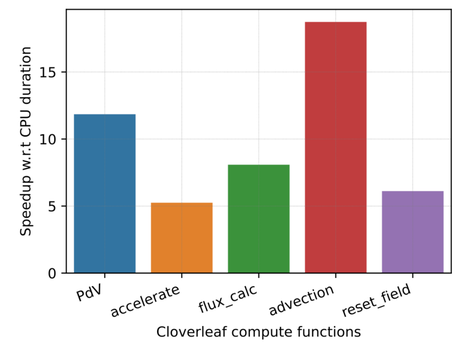


8 GPUs per node from C2-C5 remain idle while Cloverleaf-CPU runs on Corona nodes.



Cloverleaf-CPU runs on the CPU-only nodes (R2-R5) on Ruby, favoring the multi-cluster scenario for better resource utilization.

Function-level speedup comparison of selected GPU kernels w.r.t. their CPU counterparts from a single Cloverleaf iteration on Corona, using data collected by PerfFlowAspect.



CONCLUSIONS AND FUTURE WORK

- Extension to Thicket visualization and analysis to enable comparison across hundreds of configurations
- Enable Fortran support
- Collection of additional performance data such as power, I/O, and network counters
- Analyze workflows in HPC+cloud settings

PerfFlowAspect enables support for multi-binary, multi-cluster, and AI/ML assisted workflows

Lawrence Livermore National Laboratory | COMPUTING

National Nuclear Security Administration

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-POST-XXXXX