3D Flamegraphs for Performance Analysis

Undergraduate student: Alex Boots
Advisor: Xu Liu
North Carolina State University
Raleigh, USA
ajboots@ncsu.edu

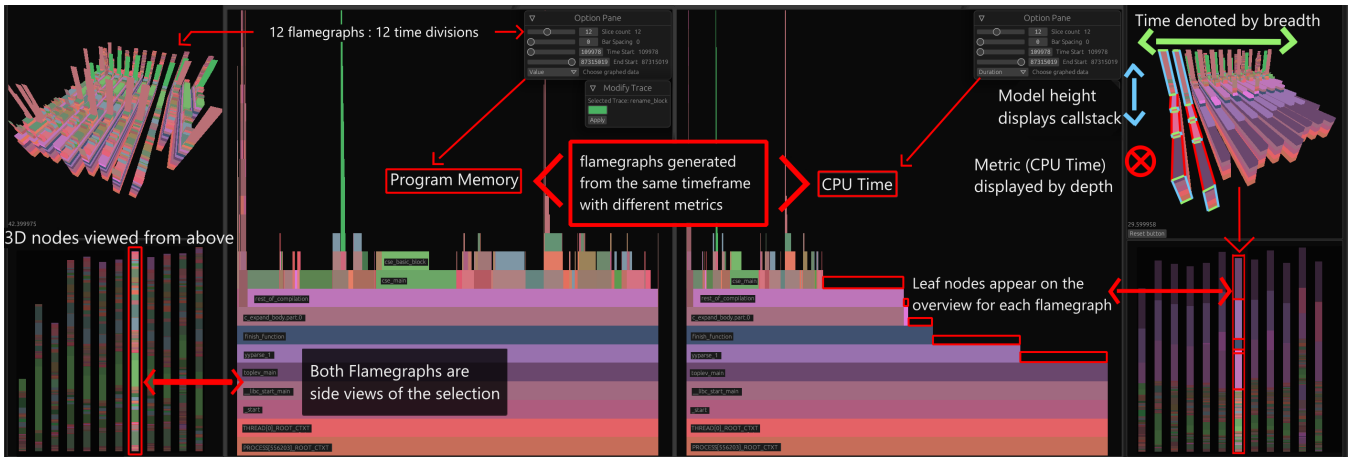

Figure 1. The screenshot of EasyFlame. EasyFlame supports 3D flamegraphs; the three dimensions show timeline, call stacks, and memory transactions, which enable bandwidth analysis in this example. Moreover, EasyFlame is highly configurable. It can slice timeline in different granularities and reduce to different 2D views.

Abstract

Flamegraphs are standard profiling tools in both industry and academia used to visualize large amounts of performance data. Existing flamegraphs are two dimensional, which limits them to showing one metric at once. This project, EasyFlame, is a performance analysis framework based on 3D flamegraphs. EasyFlame outperforms existing approaches through its inclusion of an extra dimension that shows how a metric changes across program execution. challenges.

CCS Concepts: • **Information systems** → **Data structures**.

Keywords: flamegraphs, performance analysis, profiling, tracing

1 Introduction

Data visualization is often performed through profilers and tracers to gain performance insights. Flamegraphs are the de facto method used in both industry and academia to visualize call-path profiles and traces. Both can be collected by mainstream performance tools, such as Perf [3], VTune [7], HPCToolkit [4], DrCCTProf [9]. Existing visualizations use a 2D model with one dimension representing the call stack and the other showing either an aggregated metric value or the call stack and a time dimension. Some popular metrics, such as instructions per cycle (IPC), memory bandwidth consumption, and power dissipation cannot be easily represented by flamegraphs because understanding these metrics requires the call stack, timing, and the metric (e.g., number of instructions, number of memory transactions, or energy consumed). In this paper, we address this challenge by developing EasyFlame, a 3D flamegraph framework, which enables adjustable, insightful analysis with its additional dimension. We implement and optimize EasyFlame with Web Assembly so EasyFlame is performant handling voluminous performance data through the web.

2 Related Work

Flamegraph frameworks developed by both industry and academia, such as Flamescope [8], DataDog continuous profiler [5], GProfiler [2], and a complete list of tools with flamegraphs [6], have tried to address the difficulty of graphing additional metrics. None of the existing approaches utilize the 3D flamegraphs as EasyFlame does.

3 Methodology and Implementation

EasyFlame is implemented with web front end techniques. It enables efficient data processing via WebAssembly and WebGL, which allow EasyFlame to process large data files near at native speed. This speed is necessary when regenerating hundreds of flamegraphs in real time. The usefulness of this approach depends on developing intuition through

regenerating the plot at different granularities. Using a Rust backend allows EasyFlame to utilize low level WebGL bindings to achieve high performance rendering. Performance data is read in through the standard Chrome Trace format [1] and processed into master tree. This tree allows for a single traversal to generate meshes for all stack traces regardless of the granularity.

4 Evaluation

Figure 1 shows one use case of EasyFlame for visualizing the profiles from SPECPU2006 gcc. Displayed are two graphs for the same trace. The left graph displays the CPU usage over time while the right graph shows memory usage over time. These graphs in conjunction give a complete picture of how the CPU and memory usage relate throughout the trace, especially for the memory bandwidth consumption.

5 Conclusions

EasyFlame is the first 3D flamegraph framework. EasyFlame accompanies its innovative 3D view with familiar 2D projections to aid navigation and intuition. Options are available to change the granularity of time slices, interact with the 3D model, and to recolor nodes. Implemented as a web application and optimized through WebAssembly and WebGL, EasyFlame is performant and ready for deployment.

References

[1] Chrome trace format, 2022. [Accessed Dec 14, 2022].
[2] Gprofiler, 2022. [Accessed Dec 14, 2022].
[3] Linux perf, 2022. [Accessed Dec 6, 2022].
[4] L. Adhianto, S. Banerjee, M. Fagan, M. Krentel, G. Marin, J. Mellor-Crummey, and N. R. Tallent. Hpctoolkit: Tools for performance analysis of optimized parallel programs http://hpctoolkit.org. *Concurr. Comput.: Pract. Exper.*, 22(6):685–701, apr 2010.
[5] DataDog. Continuous profiler, 2022. [Accessed Dec 14, 2022].
[6] B. Gregg. Yow! 2022: Visualizing performance: The developer's guide to flame graphs, 2022. [Accessed Dec 14, 2022].
[7] Intel. Vtune, 2022. [Accessed Dec 14, 2022].
[8] Netflix. Flamescope, 2022. [Accessed Dec 14, 2022].
[9] Q. Zhao, X. Liu, and M. Chabbi. Drcctprof: A fine-grained call path profiler for arm-based clusters. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020.