Project Proposal

Evaluation of the Transferability of Google Thesios Synthesized I/O Traces Across Different Systems

September 06th 2024

Vaishnavi Papudesi Babu

A20527963

1. Introduction

I/O traces play a crucial role in designing efficient storage systems, as they provide insights into the performance and behaviour of existing hardware, configurations, and policies. However, capturing these traces in complex, large-scale distributed systems can be challenging[1] due to the diversity of hardware, the volume of operations, and the interference caused by the trace collection process itself. To address this challenge, Google introduced a methodology called *Thesios*[2], which synthesizes I/O traces by combining down-sampled I/O data from multiple disks and storage servers. Rather than relying on full-resolution traces, which are expensive to capture and store, *Thesios* reconstructs accurate representations of I/O activity by carefully selecting and merging data samples from disks with similar characteristics, such as capacity, utilization, and data ratio. This enables the creation of realistic traces without the need for costly and risky real-world deployments.

This project aims to evaluate the accuracy and transferability of I/O traces synthesized by *Thesios*, ensuring that they can effectively represent the full I/O experience and be applied to future storage system designs. By assessing how well these synthetic traces align with actual system performance and how adaptable they are to various configurations, we can determine whether *Thesios* I/O traces are reliable for informing storage system policies and improvements.

2. Background Information

Storage systems rely heavily on I/O traces for design optimization and strategy development. These traces provide invaluable insights into real-world usage patterns, allowing to make informed decisions about hardware configurations, caching mechanisms, and data placement strategies. By analyzing I/O traces, designers can identify bottlenecks, optimize read and write patterns, and improve overall system performance. Furthermore, these traces enable the simulation and testing of new storage architectures and algorithms under realistic workload conditions, facilitating the development of more efficient and scalable storage solutions. The ability to capture and analyze I/O behavior at scale is particularly crucial in high-performance computing (HPC) environments, where even minor inefficiencies can significantly impact overall system performance and scientific output.

In this project, we discuss various available tools to understand whether synthesized I/O traces can be reliably used to make informed decisions in storage system design. The ability to accurately simulate or capture I/O activities can greatly impact how efficiently storage systems are designed and optimized. The following are the main components involved:

IOR benchmarking [3] is a tool used in HPC environments to measure parallel I/O performance. By using IOR, researchers can generate consistent and comparable I/O metrics, enabling the evaluation of system performance across different architectures and configurations. This standardized approach to benchmarking is crucial for understanding how different storage systems and I/O strategies perform under controlled, reproducible conditions.

Darshan [4] is an open-source I/O characterization tool widely used in high-performance computing (HPC) environments. It is designed for lightweight, continuous monitoring of I/O workloads with minimal overhead, making it suitable for production use. Darshan operates by intercepting I/O function calls at multiple levels of the I/O stack, including POSIX, MPI-IO, and high-level libraries. It captures key I/O statistics and provides insights into application I/O patterns, helping researchers and system administrators identify performance bottlenecks and optimize I/O strategies.

Thesios[2] is designed by Google to synthesize full-resolution I/O traces by carefully combining sampled I/O traces. It addresses the challenge of capturing comprehensive I/O behavior in large-scale systems where full-resolution tracing is often impractical due to performance overheads. Thesios aims to generate representative traces with high accuracy for key metrics such as utilization, number of requests, caching hits, and latency. A unique feature of Thesios is its ability to explore "what-if" scenarios by synthesizing realistic counterfactual traces, allowing for the simulation of anticipated hardware or workload changes without costly real-world experiments.

3. Problem Statement

Thesios, developed to synthesize full-resolution I/O traces by combining sampled traces, has shown promise in aiding system design decisions for storage systems. It claims to accurately generate representative traces for key performance metrics like utilization, request numbers, cache hits, and latency. By enabling what-if analyses that simulate hypothetical hardware or workload changes, Thesios provides an alternative to costly and time-consuming real-world experiments. With its integration of server simulators and emulators, Thesios allows for exploring counterfactual scenarios in distributed storage environments, making it highly applicable to large-scale data centers that already utilize I/O sampling tools.

While Thesios offers an innovative approach to generating realistic I/O traces, there remains a significant need to verify the accuracy and reliability of these synthesized traces before they can be used to make critical system design decisions. Synthetic traces must closely mimic the actual performance characteristics of storage systems to be considered trustworthy for guiding hardware upgrades, policy changes, or data placement strategies. Without careful validation, there is a risk of making erroneous conclusions, potentially leading to suboptimal design choices or performance bottlenecks. Therefore, the focus of this project is to thoroughly evaluate the accuracy of the I/O traces generated by Thesios and determine their applicability to actual storage system designs. Ensuring that these synthetic traces accurately reflect real system behaviors is essential for their adoption as a reliable tool in storage system planning and optimization.

4. Related Work

Several prior studies have focused on I/O workload characterization, tracing, and synthetic workload generation. These efforts provide valuable insights into how synthesized I/O traces can be used for system design decisions. One of the core challenges in storage system evaluation has been capturing and understanding I/O traces at scale. Tools such as Stardust[5] were developed to capture disk-level I/O activity in distributed systems, but scaling these approaches to exascale data centers has proven difficult. For instance, while Stardust can capture traces across a large network, its ability to scale and maintain granularity across components like disks and servers remains limited.

In parallel, synthetic workload generation has been a focal point for I/O trace analysis. Research on tools like FIO[6] has explored methods to generate realistic I/O workloads by simulating specific aspects of disk behavior, such as read/write patterns.

Darshan and Datacrumbs represent important contributions by providing in-depth I/O characterization at both the application and system levels. Darshan, in particular, focuses on HPC workloads, offering insights into file system usage, access patterns, and scalability across different clusters. Its integration with the IOR benchmarking tool enables the generation of realistic workloads for evaluation purposes.

On the other hand, Thesios offers a novel approach by stitching together down-sampled traces from distributed storage servers, allowing for counterfactual analysis and detailed performance evaluations. Thesios' ability to handle heterogeneous storage clusters and provide high-fidelity synthetic traces is a key step toward scalable I/O tracing, making it highly relevant for modern data center environments.

5. Proposed Solution

In this paper, we propose to evaluate the accuracy and transferability of synthesized I/O traces provided by Google's Thesios by comparing them against real-time I/O traces generated by tools such as Darshan or Datacrumbs. The goal of this comparison is to determine how closely the synthesized traces reflect real-world workloads and whether they can be used reliably to make storage system design decisions. Thesios offers the ability to synthesize traces based on sampled data, which is valuable for simulating potential future workloads or hardware changes. However, to ensure its effectiveness, it is crucial to compare these synthetic traces with actual traces collected from live systems.

To achieve this, we will generate workloads using the IOR benchmarking tool, which allows us to create representative I/O patterns that stress different aspects of the storage system, such as read/write performance, metadata operations, and caching behavior. By applying IOR to various test cases on actual systems, we can capture real-time I/O traces using Darshan or Datacrumbs, two established profiling tools that offer detailed insights into I/O behavior at different layers of the stack. The real-time traces will then serve as a baseline for comparing the accuracy and fidelity of Thesios-generated traces.

The implementation of this solution will require scripting to automate the process of workload generation, trace capture, and comparison. For this purpose, we will use Python or shell scripting to develop scripts that target specific clusters, such as the Ares cluster or the Chameleon cluster. These clusters provide the necessary computational and storage resources to run large-scale I/O benchmarking and trace collection in a distributed environment. By conducting experiments on these clusters, we aim to evaluate the performance of Thesios in realistic data center settings, ensuring that the synthetic traces generated are both accurate and transferable across different environments.

6. Evaluation

In the Evaluation of this project, we will begin by thoroughly analyzing the I/O traces generated by Google's Thesios, identifying patterns such as common file transfer sizes and specific I/O behavior trends. By examining these characteristics, we can generate benchmarking loads using tools like IOR, simulating similar workloads to evaluate system performance. Alongside this, we will use Darshan to capture real-time I/O traces and compare them with the synthesized traces from Thesios. This comparison will allow us to measure the accuracy and transferability of the synthesized data to real-world scenarios.

Key metrics for this evaluation include file size distribution, throughput, latency, and cache hit rates. We will focus on understanding how well the system performance converges as file sizes increase and observe whether similar patterns emerge in both the synthesized and real-time traces. By doing so, we will determine if the I/O behavior from Thesios accurately reflects real workloads captured by Darshan.

Although the test environment for these evaluations is yet to be finalized, we plan to replicate the configuration used by Thesios, either using the Ares or Chameleon clusters. If we find that IOR benchmarking does not fully capture the intricacies of the workloads or fails to stress the system adequately, we will consider switching to FIO as an alternative tool for generating I/O workloads. This comprehensive approach will allow us to evaluate how closely the synthesized traces match real-world I/O patterns, ensuring their validity for future storage system design decisions.

7. Conclusions

In conclusion, this paper aims to assess the accuracy and applicability of synthesized I/O traces generated by Google's Thesios for making informed decisions in storage system design. By comparing these synthesized traces with real-time data captured by tools such as Darshan, we will evaluate the viability of using synthesized I/O patterns to replicate real-world system behaviors. The benchmarking loads generated through IOR, or similar benchmarking tool, will provide a solid foundation for observing key performance metrics, such as file size distribution and throughput, allowing for a robust comparison of the two data sets.

Our proposed methodology will help verify the accuracy of Thesios-generated traces and test their transferability across different environments. This study has the potential to provide valuable insights for improving storage system designs and offer a reliable, cost-effective alternative to costly real-time experimentation. Through thorough analysis and evaluation, we hope to determine if synthesized I/O traces can serve as a practical and reliable solution for future research and development in distributed storage systems.

8. Additional Resources

8.a) Timeline

Week	Task
1	Study relevant papers, gather all the required information
2	Evaluate and generate metrics from Google Thesios I/O traces
3	Code necessary scripts to generate realtime I/O traces
4	Present intermediate report(midterm)
5	Do the experiment evaluation
6	Do the experiment evaluation
7	Final report
8	Final presentation

8.b) Deliverables

- One final report in PDF form.
- One final Powerpoint presentation.
- Source code.

References

[1] Gregory R Ganger. Generating representative synthetic workloads: An unsolved problem. In Proc. Computer Measurement Group (CMG) Conference, Dec. 1995, 1995.

[2] Phitchaya Mangpo Phothilimthana, Saurabh Kadekodi, Soroush Ghodrati, Selene Moon, and Martin Maas. 2024. *Thesios: Synthesizing Accurate Counterfactual I/O Traces from I/O Samples*. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS '24), Vol. 3. Association for Computing Machinery, New York, NY, USA, 1016–1032. https://doi.org/10.1145/3620666.3651337

[3] Lawrence Livermore National Laboratory, "IOR benchmark," https://github.com/chaos/ior, 2015.

[4] S. Snyder, P. Carns, K. Harms, R. Ross, G. K. Lockwood and N. J. Wright, "Modular HPC I/O Characterization with Darshan," 2016 5th Workshop on Extreme-Scale Programming Tools (ESPT), Salt Lake City, UT, USA, 2016, pp. 9-17, doi: 10.1109/ESPT.2016.006. keywords: {Instruments;Libraries;Buffer storage;Production systems;Writing;Runtime;Registers},

[5] Eno Thereska, Brandon Salmon, John Strunk, Matthew Wachs, Michael Abd-El-Malek, Julio Lopez, and Gregory R Ganger. Stardust: Tracking activity in a distributed storage system. ACM SIGMETRICS Performance Evaluation Review, 34(1):3–14, 2006.

[6] JensAxboe.FlexibleI/OTester,https://fio.readthedocs.io.