Automatic Benchmark Generation from HPC I/O Traces

Proposal for an internship/project
Advisors: Francieli Boito (francieli.zanon-boito@u-bordeaux.fr) and Luan Teylo (luan.teylo@inria.fr)
Institution: Centre Inria de l'Université de Bordeaux, France
September 2023

1. Motivation

High-Performance Computing (HPC) applications are designed to solve real-world problems, such as aerospace simulations and weather prediction models. These applications require vast computational resources and are executed on what are known as Supercomputers: massive clusters comprising hundreds to thousands of computing nodes with high computing, network, and storage power. HPC applications often generate and consume a huge amount of data. To manage all this data, supercomputers employ Parallel File Systems (PFS), which are shared file systems with multiple storage servers. These are designed for high throughput and concurrent access from multiple applications. Thus, a PFS in a supercomputer handles a large volume of I/O (Input/Output) requests, including various concurrent operations such as opening and reading one or multiple files.

To evaluate and study a PFS, benchmark tools are used. A benchmark tool is an application specifically designed to assess the performance of hardware, software, or entire systems under various conditions. The main goal of benchmarking is to generate quantifiable metrics that can be used for comparing different configurations or systems. Essentially, a benchmark runs a series of tests, that emulates various operations performed by the hardware or software. After running these tests, the benchmark produces results that offer insights into performance characteristics such as speed, latency, throughput, and resource utilization. These metrics serve as valuable tools for system tuning and optimization.

In this project, we aim to develop a benchmark tool focused on the I/O operations of HPC applications and the evaluation of Parallel File Systems. Unlike "general-purpose" tools, such as IOR [1], our objective is to create a tool that is capable of mimicking the actual behavior of a real HPC application. In order to do that, the tool will use traces obtained from previous executions. This approach will allow us to assess how a system performs under realistic conditions.

2. Goals

The goal of this internship is to develop a method (and implement it) to, given a job I/O trace, execute a benchmark that mimics that job's I/O behavior. A job I/O trace is a file containing the bandwidth (bytes per second) measured by the file system according to its I/O activity periodically (e.g. every 10 seconds).

To execute a benchmark, the proposed route is to use an existing benchmarking tool, called IOR, which allows for scripting different I/O phases. More concretely, for example, if the application accesses 10GB, then does no I/O for 10 seconds and then accesses 5GB, we want the tool to create an IOR script that does exactly that. This is the first step and minimal result expected from the project.

Further steps will be proposed depending on the progress of the person. Some examples:

- Modify IOR (or another tool, if another tool is used) to export more information, such as I/O phase

limits and compute metrics such as stretch. These metrics are useful for scheduling experiments, for example [2].
- Generate multiple benchmark versions covering different access patterns, e.g. contiguous or random access to files, single file or multiple ones, etc.
- Use frequency tools, such as FTIO [3], to simplify the generated script for periodic jobs.
- Generate a benchmark from multiple traces, by combining and generalizing their behavior (for example, with averages or, more sophisticated, by fitting some distribution).
- Handle noisy traces, where the limits of I/O phases are not very clear.

3. Please notice

The meetings related to this project will be held in English.

3. References

[1] https://github.com/hpc/ior
[2] Francieli Boito, Guillaume Pallez, Luan Teylo, Nicolas Vidal. IO-SETS: Simple and efficient approaches for I/O bandwidth management. IEEE Transactions on Parallel and Distributed Systems, 2023, 34 (10), pp.2783 - 2796. 10.1109/TPDS.2023.3305028. https://inria.hal.science/hal-03648225/
[3] Tarraf, A., Bandet, A., Boito, F.Z., Pallez, G., & Wolf, F.A. (2023). FTIO: Detecting I/O Periodicity Using Frequency Techniques. https://arxiv.org/abs/2306.08601