Assignment7

Please complete the report and submit `report.pdf` through Blackboard

Performance is an important factor to consider for operating systems. Asterinas tracks the performance gap with Linux on a daily basis by employing automated testing (website). In this assignment, you will experience **the process of performance optimization** (or, decrease the performance), more precisely, this assignment will cover the following parts (If you choose to do performance optimization):

1. Identify performance gaps by using benchmarking tools.
2. Flame graph generation, recognized performance bottlenecks.
3. Try to optimize performance.
4. Retry steps 1~3 until the performance is good enough.

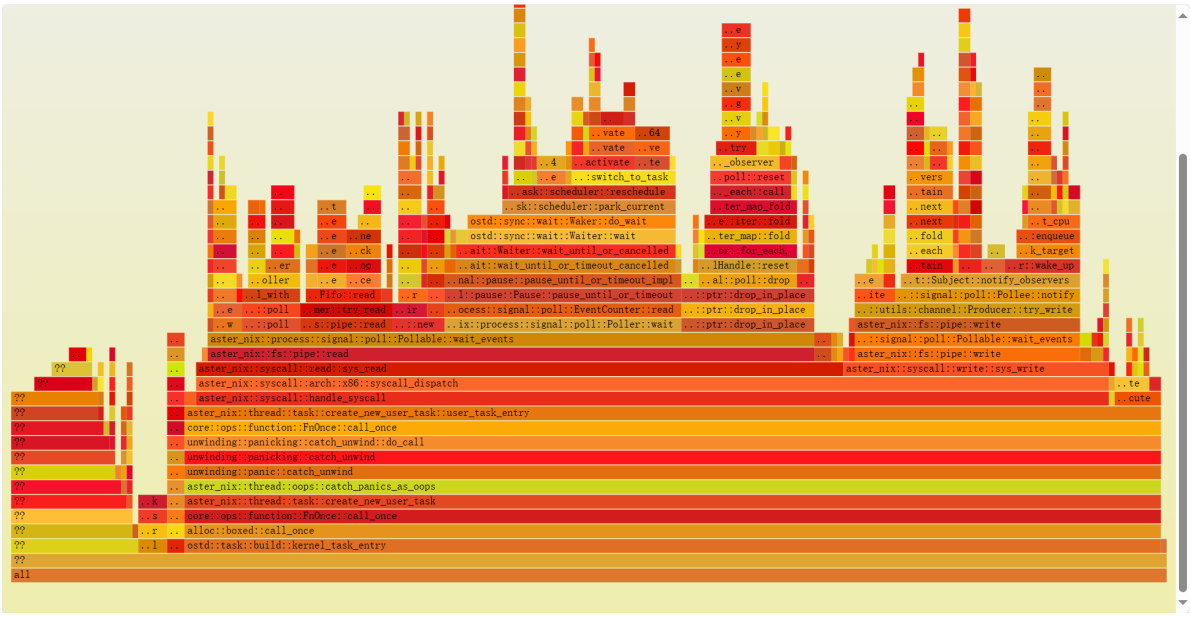
Background

LMbench

From LMbench: "lmbench is a suite of simple, portable, ANSI/C microbenchmarks for UNIX/POSIX. In general, it measures two key features: latency and bandwidth. lmbench is intended to give system developers insight into basic costs of key operations."

Flame graph

From Brendan Gregg: "Flame graphs are a visualization of hierarchical data, created to visualize stack traces of profiled software so that the most frequent code-paths to be identified quickly and accurately." Here is the screenshot of a flame graph analyzing the performance bottleneck of `lat_pipe` in Asterinas:



What to do

In this assignment, you need to do the following things:

(1) Identify performance gaps

1. Select an LMbench script(e.g. `lat_pipe`, `lat_select file`) based on the performance website.
2. In container (root privileges required, use `sudo podman/docker run xxx` to start the container), run `test/benchmark/bench_linux_and_aster.sh lmbench/{Benchmark script directory}` to run the script.
3. You will get `result_lmbench-{Benchmark script directory}.json` in the base directory.

(2) Generate flame graph

The following steps should be performed in container.

1. Start another shell with command `sudo podman/docker exec -it {container name/id} /bin/bash`
2. In shell A, run `make profile_server RELEASE=1` then run the benchmark script in the guest OS (the number of repetitions should be large enough, e.g. 1,000,000, `{Benchmark script} --help` will be helpful).
3. After running the script on the guest OS, execute `make profile_client RELEASE=1` in shell B.
4. After shell B is completes, you will get `aster-nix-profile-xxxx.svg`, open the file using a browser.

(3) Increase OR decrease performance

1. You have two options:
    - Increase performance: Identify performance bottlenecks based on the flame graph. (**Dummy optimization is acceptable**, i.e. remove some useless logic)
    - Decrease performance: Add some time-consuming logic to one of the functions in the flame graph.
2. Try again (1).2 and (1).3 to prove that your optimization / time-consuming logic is effective.

Report

Your report should cover:

1. **[15pts]** What is your chosen LMbench script
2. **[15pts]** The screenshot of `result_lmbench-{Benchmark script directory}.json`
3. **[20pts]** The screenshot of `aster-nix-profile-xxxx.svg` (Use browser)

IF you choose to do optimization, then you need to:

1. **[30pts]** The optimization you did and why you did it. **Please refer to the previous flame graph**
2. **[20pts]** After optimization, try (1).2 and (1).3 again, then show the screenshot of `result_lmbench-{Benchmark script directory}.json`

IF you choose to add time-consuming logic, then you need to:

1. **[20pts]** The time-consuming logic you added.
2. **[30pts]** Generate flame graph again to show the time-consuming logic works.