# Performance Impact of –fno-omit-frame-pointer on IA

Intel

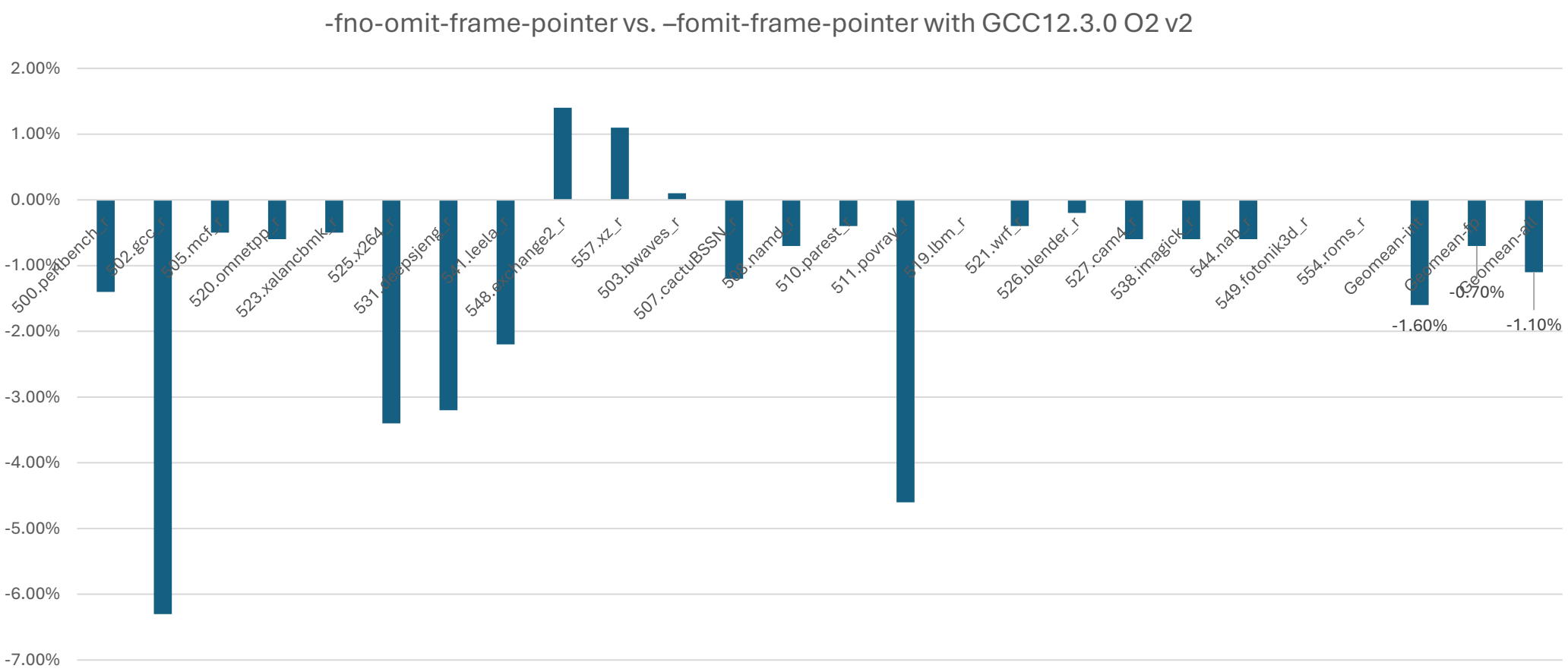
Annita Zhang

# Background

https://fedoraproject.org/wiki/Changes/fno-omit-frame-pointer

- Problem Statement
  - Fedora added –fno-omit-frame-pointer and –mno-omit-leaf-frame-pointer to the default C/C++ compilation flags
- Purpose:
  - Improve the effectiveness of profiling and debugging
- Status:
  - It was added into Fedora Linux 38 and 40
  - Ubuntu 24.04 LTS enabled it by default
- Cost:
  - Save, setup and restore frame pointer in each function => 3 instructions
  - Lose 1 GPR register for register allocation
- Performance Impact:
  - Some performance loss is expected

# SPEC2017 performance impact on EMR

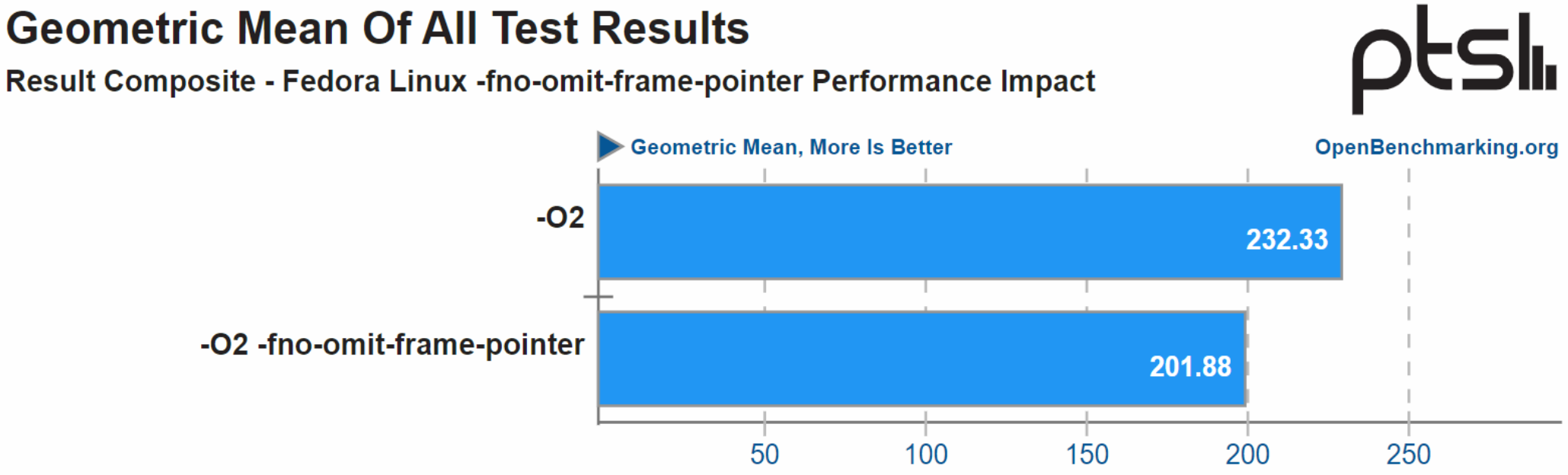


# Performance Impact by Fedora and Suse

- Benchmark Measurement Reported by Fedora
  - Compiling the kernel with GCC is 2.4% slower with frame pointers
  - Running Blender to render a frame is 2% slower on our specific testcase
  - openssl/botan/zstd do not seem to be affected significantly when built with frame pointers
  - The impact on CPython benchmarks can be anywhere from 1-10% depending on the specific benchmark
  - Redis benchmarks do not seem to be significantly impacted when built with frame pointers
- Benchmark reported by Suse
  - Possible 5~10% range depending on the particular software

# Performance Impact reported by Phoronix

- The geomean performance penalty is 14% out of 100 tests




https://www.phoronix.com/review/fedora-frame-pointer/5

# Proposal

- Given the performance impact, suggest not to enable –fno-omit-frame-pointer by default in GCC for OpenEuler releases
- Explore alternative approaches to improve profiling and debugging
  - Dwarf
  - ORC
  - LBR
  - CTF Frame
  - Shadow Stack
- Redhat is developing “eu-stacktrace” w/o frame pointer
  - https://www.phoronix.com/news/Red-Hat-eu-stacktrace