

### Motivation

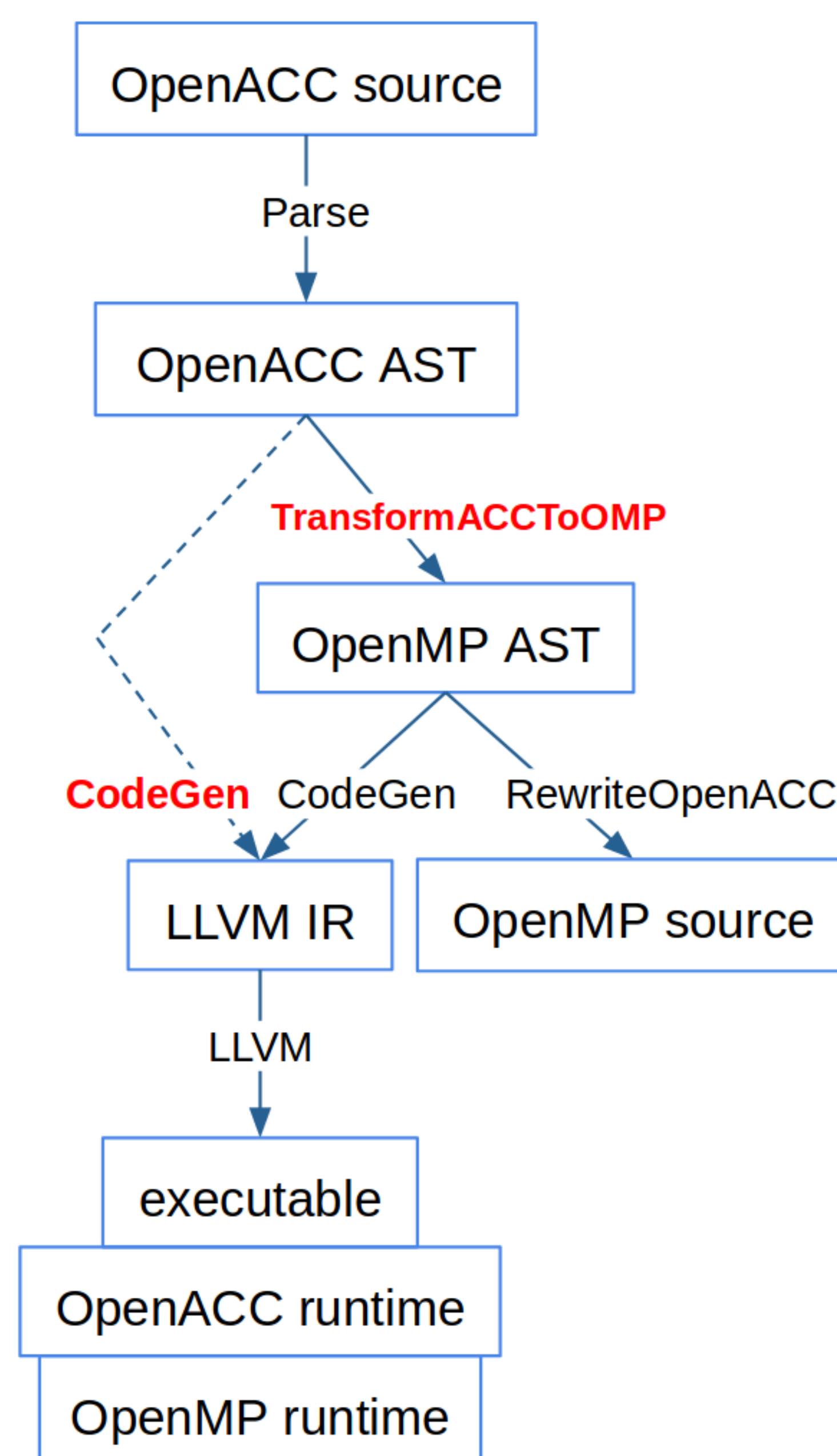
- OpenACC launched in 2010 as portable directive-based programming model in C, C++, and Fortran for heterogeneous accelerators
- OpenACC 3.0 vs. OpenMP 5.0
  - 149 pages vs. 666 pages (182 pages are tooling libs)
  - Descriptive vs. prescriptive: requires less performance tuning from app developers
  - OpenMP features continue to be inspired by OpenACC
  - OpenACC remains an important, more tractable alternative for heterogeneous programming
- Lack of extensible, open-source, production-quality OpenACC compilers
  - Needed for HPC app development and OpenACC adoption and evolution
  - GCC support is improving, but LLVM is known for its extensible ecosystem and license model

### Objectives

- Develop production-quality, standard-conforming traditional OpenACC compiler and runtime support by extending Clang and LLVM
- Enable research and development of source-level OpenACC tools
  - Design compiler to leverage Clang/LLVM ecosystem extensibility
  - E.g., Pretty printers, analyzers, lint tools, and debugger and editor extensions
- As matures, contribute OpenACC support to upstream Clang and LLVM
- Throughout development
  - Actively contribute upstream all Clang and LLVM improvements that are mutually beneficial
  - Actively contribute to the OpenACC specification
  - Many contributions already made from Clacc

### Design: Translate OpenACC to OpenMP

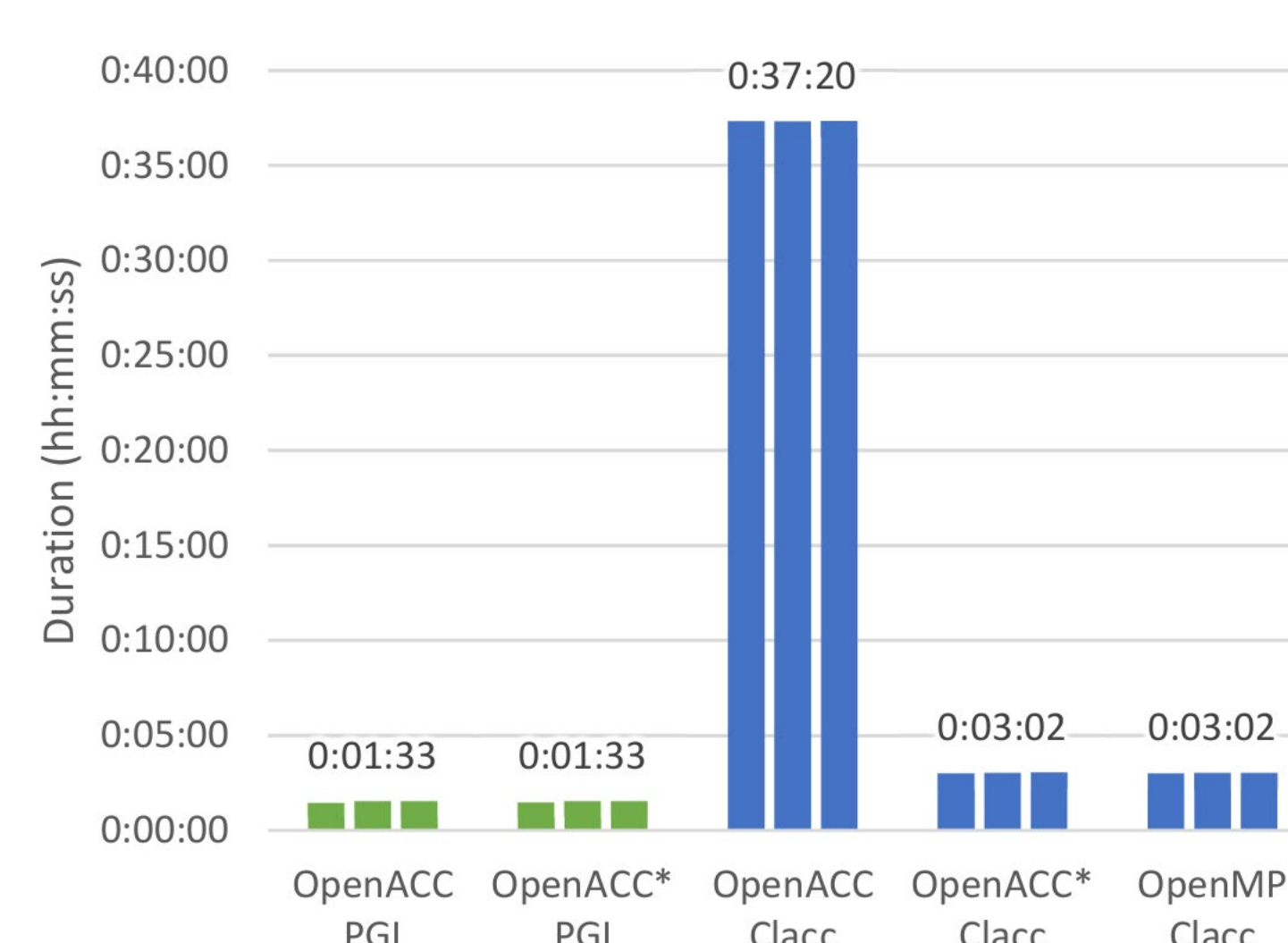
- Primarily a pragmatic choice for traditional compilation
  - Builds on existing OpenMP implementation
  - Lowering representation: descriptive to prescriptive
  - OpenMP acts as IR: OpenMP diagnostics are not exposed to users
- Creates potential for non-traditional user-level features
  - Reuse/extend existing OpenMP tools
  - Source-to-source translation: automatic migration to OpenMP
  - Semantics and implementation for mixed OpenACC/OpenMP
  - OpenMP as debug representation of compiler optimization decisions
- Possibly use MLIR or LLVM IR extensions in the future



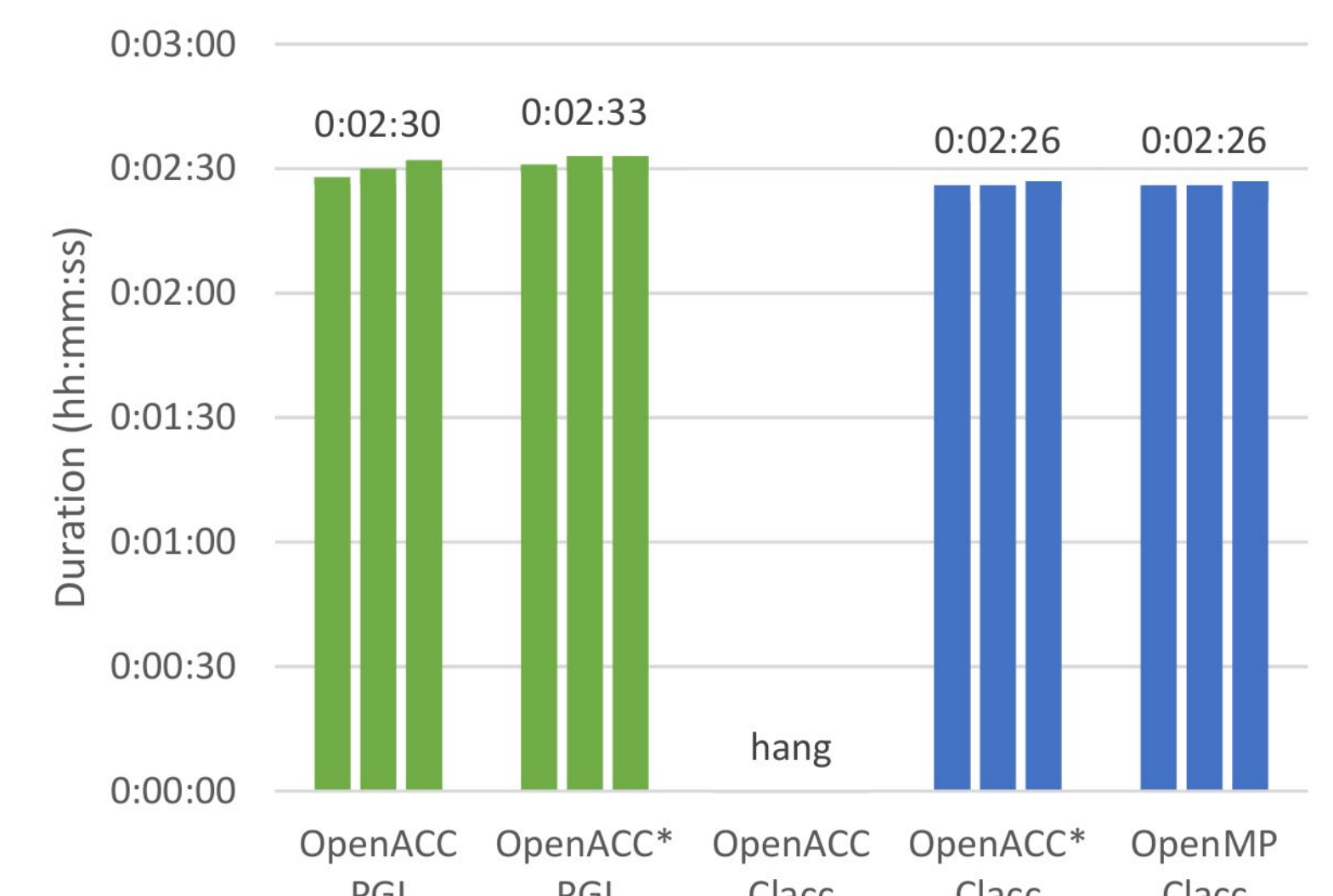
### Preliminary GPU Evaluation

- Clacc status: C, prescriptive interpretation, GPU and multicore
- Ubuntu 18.04.3, 32 GB DRAM
- Intel Core i7-7700HQ 2.8GHz CPU (8 threads)
- NVIDIA GeForce GTX 1050
- PGI Community Edition 19.10
- SPEC ACCEL 1.3, ref workload, non-compliant

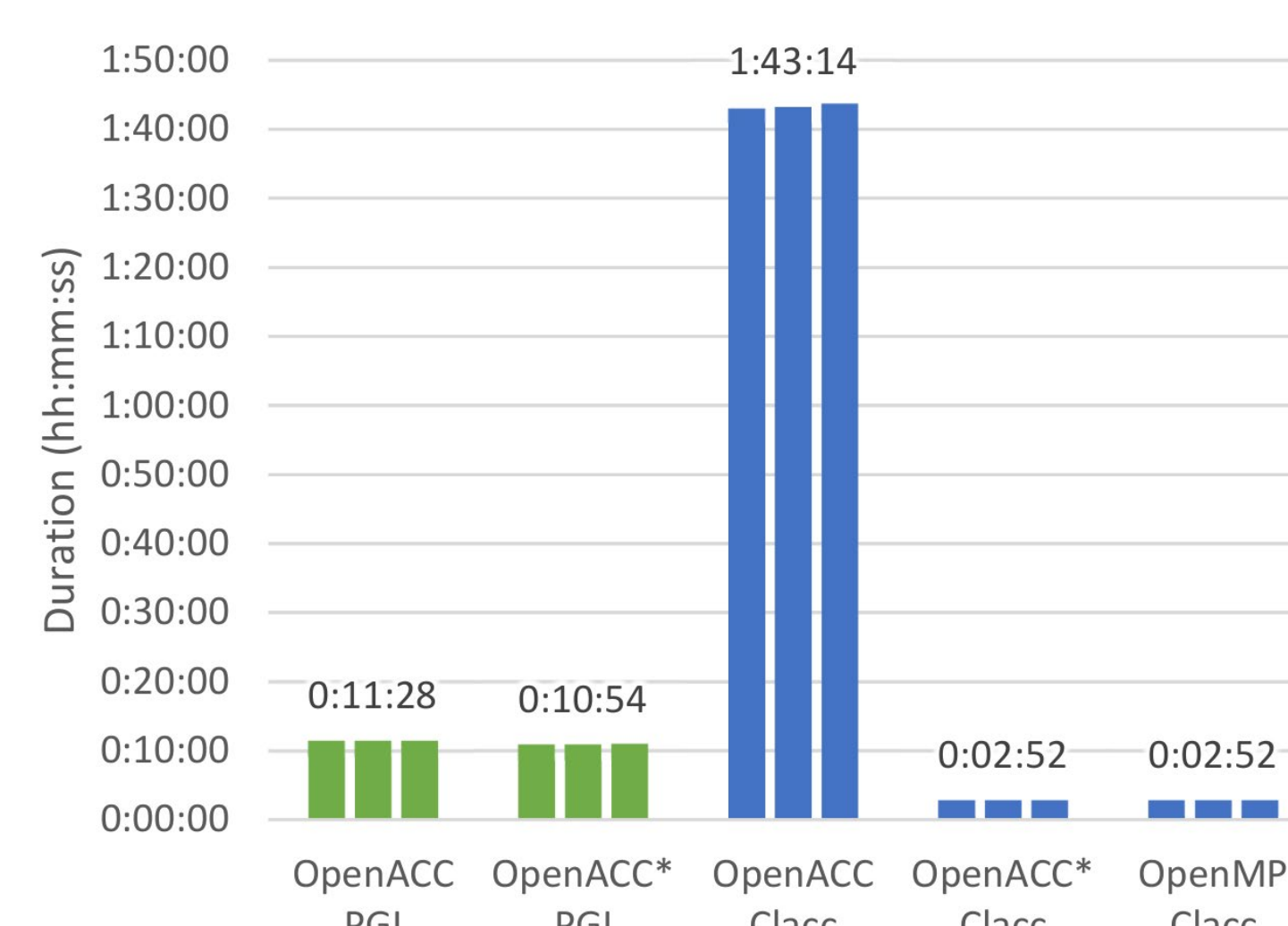
303.ostencil and 503.postencil



304.olbm and 504.polbm



314.omriq and 514.pomriq



- OpenACC = 3xx benchmark
- OpenMP = 5xx benchmark (OpenMP offload counterpart)
- OpenACC\* = 3xx benchmark modified to have same levels of parallelism and private clauses as 5xx benchmark

Need descriptive OpenACC interpretation