

LEVERAGING HIP ON INTEL PVC

HIP ON AURORA PROJECT TEAM
Colleen Bertoni, Brice Videau

WHAT IS HIP?

Heterogeneous-Computing Interface for Portability

- HIP is AMD's Portable GPU programming model
- Very similar to CUDA
 - CUDA programs can be automatically translated to HIP (HIPIFY)
 - HIP programs run natively on AMD GPUs using ROCm + LLVM/Clang
 - HIP programs run natively on NVIDIA GPUs using CUDA + nvcc
- AMD provides GPU accelerated math libraries
 - hipBLAS, hipFFT, hipSPARSE, ...
 - Wrappers around AMD and Nvidia GPU compute libraries
 - Example: hipBLAS wraps rocBLAS and cuBLAS
- High performance on both AMD and Nvidia GPUs by design

HIP ON AURORA PROJECT OBJECTIVES

Support HIP Applications on Intel GPUs

- 2 Exascale systems feature AMD GPUs (Frontier OLCF, El Capitan LLNL)
 - HIP is a likely target for many exascale applications
 - 3 Pre-exascale systems use Nvidia GPUs (Summit, Perlmutter, Polaris)
- What about Aurora with its Intel GPUs?
 - Bring HIP support to Aurora
- HIP on Aurora project, originally lead by Hal Finkel:
 - Implement HIP on top of Level Zero (Intel native API on Aurora) or OpenCL;
 - Restructure HIP to accept several backend plugins;
 - Create a test suite with code of interest to ECP;
 - Support selected ECP applications;
 - Implement HIP math libraries on top of oneMKL;
 - Investigate CUDA support.

HIP ON AURORA TEAM COLLABORATORS

Team is made up of National Labs, Academia, Industry



- National labs
 - Compiler and runtime researchers focused on HPC community
 - Users and developers of HPC applications



- Academia
 - Compiler and runtime researchers focused on new programming technologies



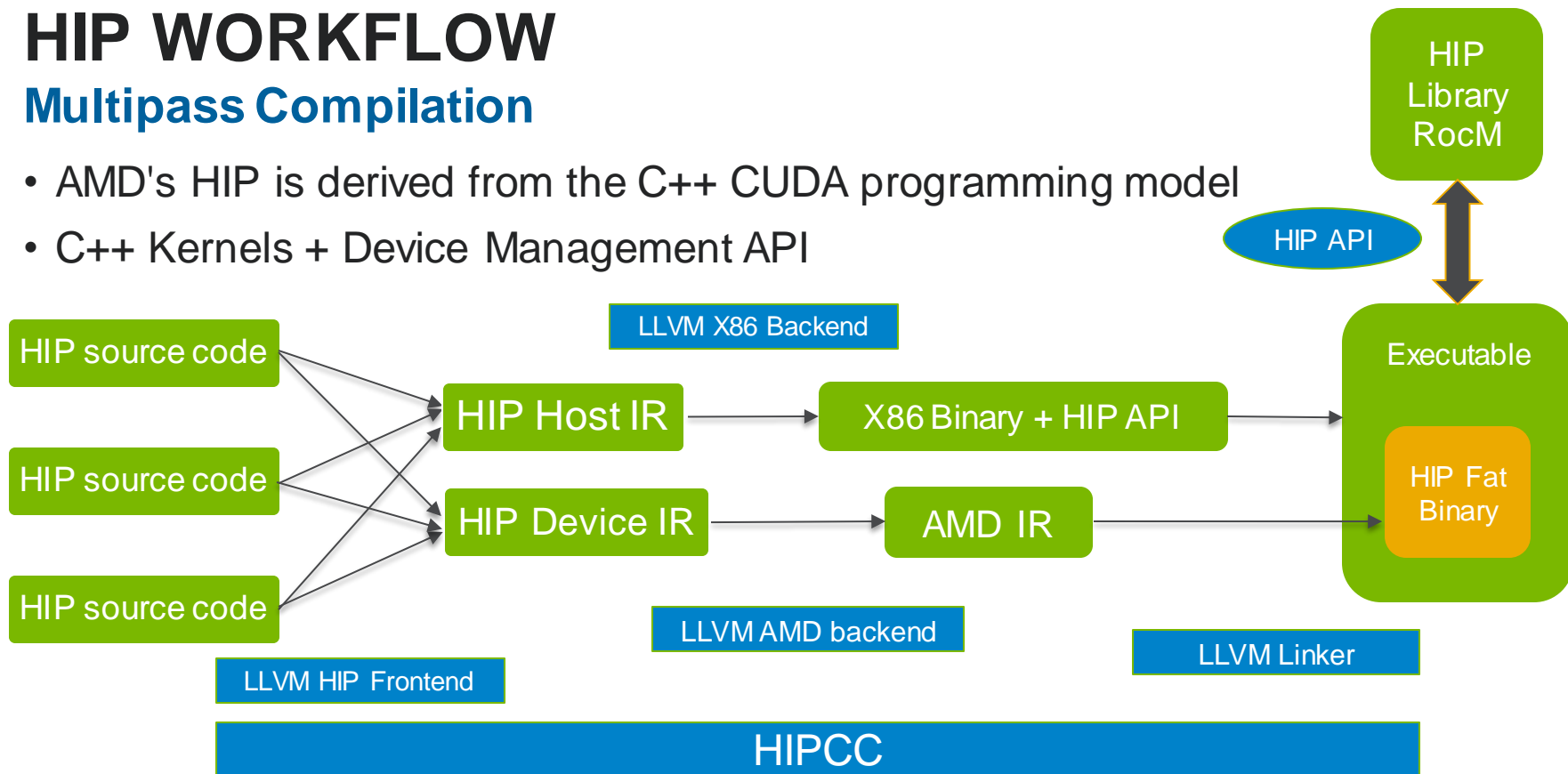
- Industry
 - Developers of HIP
 - Developers of Intel GPUs



HIP WORKFLOW

Multipass Compilation

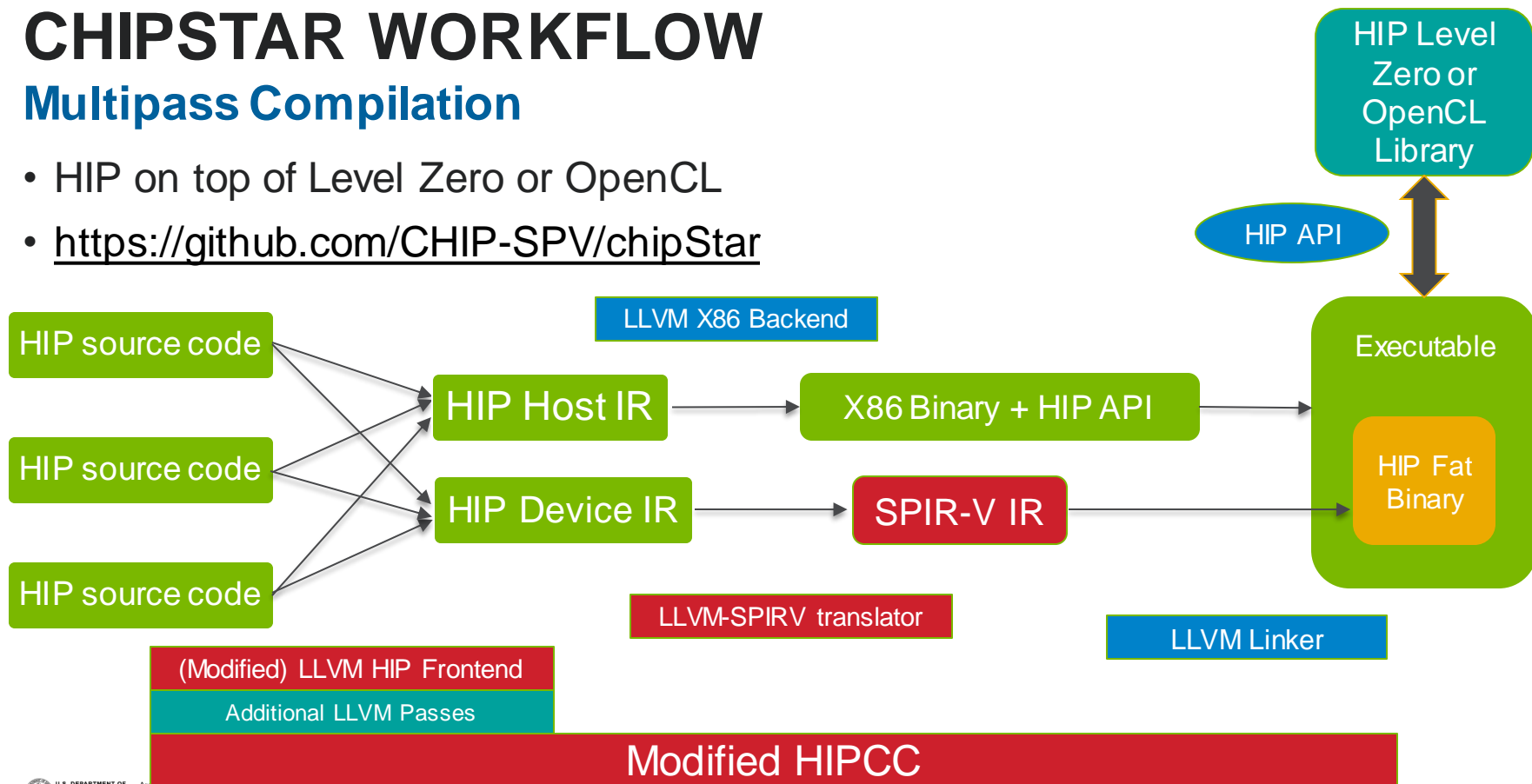
- AMD's HIP is derived from the C++ CUDA programming model
- C++ Kernels + Device Management API

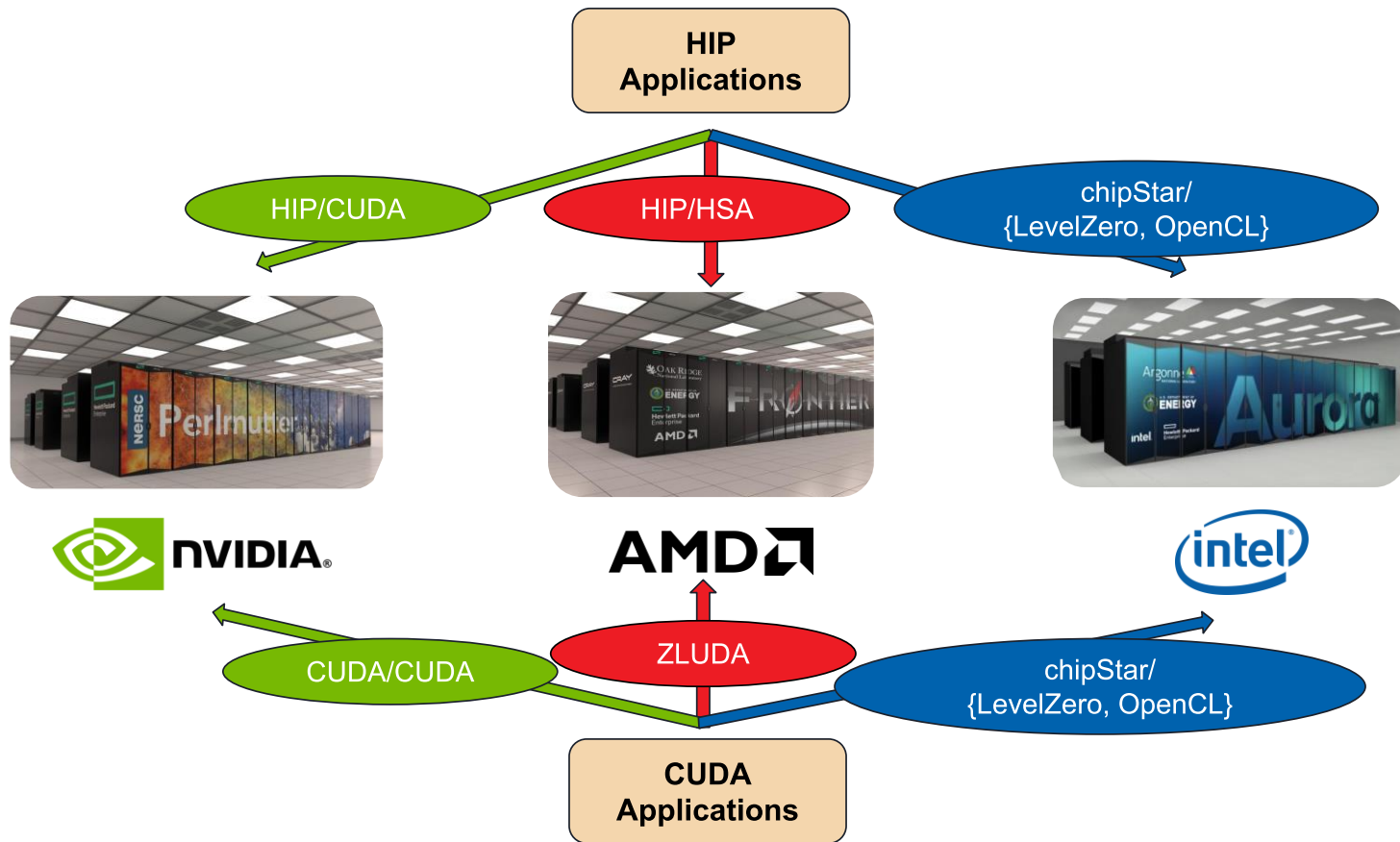


CHIPSTAR WORKFLOW

Multipass Compilation

- HIP on top of Level Zero or OpenCL
- <https://github.com/CHIP-SPV/chipStar>





SUPPORTED SOFTWARE



Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.



CHIPSTAR

A HIP and CUDA Implementation on top of Level Zero and OpenCL

- Available online: <https://github.com/CHIP-SPV/chipStar>
- Built from open source components:
 - HIP Common
 - HIP Tests
 - ROCm-Device-Libs
 - LLVM(16, 17, 18, 19)
 - SPIRV LLVMTranslator
- GitHub action CI leverages AMD unit tests
- Version 1.2.0 (and later 1.2.1) released late 2024, stability and performance improvements, enhanced support for CUDA
- Currently supported by ALCF (through PaganLC) and Intel.
- CUDA is experimentally supported by converting CUDA calls to chipStar calls

SUPPORTED LIBRARIES

Provide HIP Math Libraries for Applications

- Actively supported and built on top of oneMKL, thank to oneAPI SYCL interoperability
- Unified architecture to interface with SYCL oneMKL
- Delivered to ECP
 - hipRTC
 - hipBLAS
 - hipSOLVER
- Work in progress
 - hipFFT
 - hipCUB
- Prospective
 - hipRand

SUPPORTED APPLICATIONS

Help application teams to use HIP on Aurora

- Actively engaged with application teams
- Functional, optimizing performance
 - CP2K (several CP2K apps are supported, more are coming)
 - LibCEED (some quirks exist for some patterns)
 - GAMESS
 - OpenMM
- In progress
 - OpenFanWing INCITE
- Prospective, always looking for more
 - LS_DALTON (TAL_SH)
 - DIRAC (TAL_SH)
 - TRITON

LIMITATIONS

Most Common Encountered Issues

- Kernel limitations (apply to SYCL as well)
 - Some masked group functions are not supported
 - 3 argument shuffles
 - HW limitation, working on emulation
 - Early exit from thread will cause issues
 - HW limitation
 - Working on compiler pass to alleviate simple cases
- Missing libraries
 - Supporting new libraries can be done, some are in progress
 - Intel oneMKL and other libraries usually map one to one with CUDA libraries
- Bugs, they should be reported and the project will address them

BENCHMARKING WITH HECBENCH



Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.



EVALUATING CHIPSTAR WITH THE HECBENCH BENCHMARK SET

- Goals:
 - Assess the functionality of chipStar for HIP and CUDA APIs by looking at the pass rate of the benchmark set
 - Report any bugs found
 - Assess the performance of chipStar by comparing the performance of HIP codes running with chipStar to SYCL codes running with Intel oneAPI on Intel GPUs
- We chose to use HeCBench (<https://github.com/zjin-lcf/HeCBench>) since it is extensive and has the same code written in multiple different languages

BENCHMARKING WITH HECBENCH: HIP API

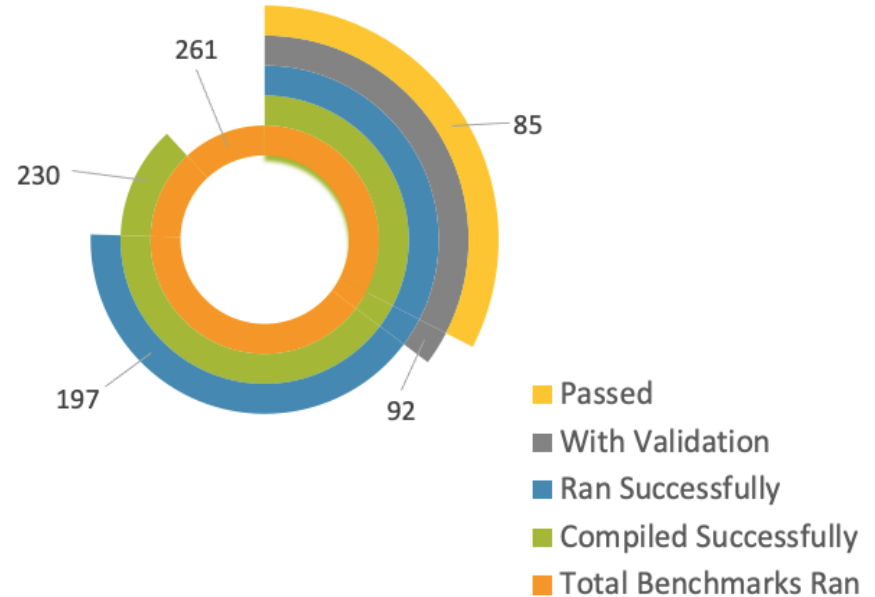


Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.

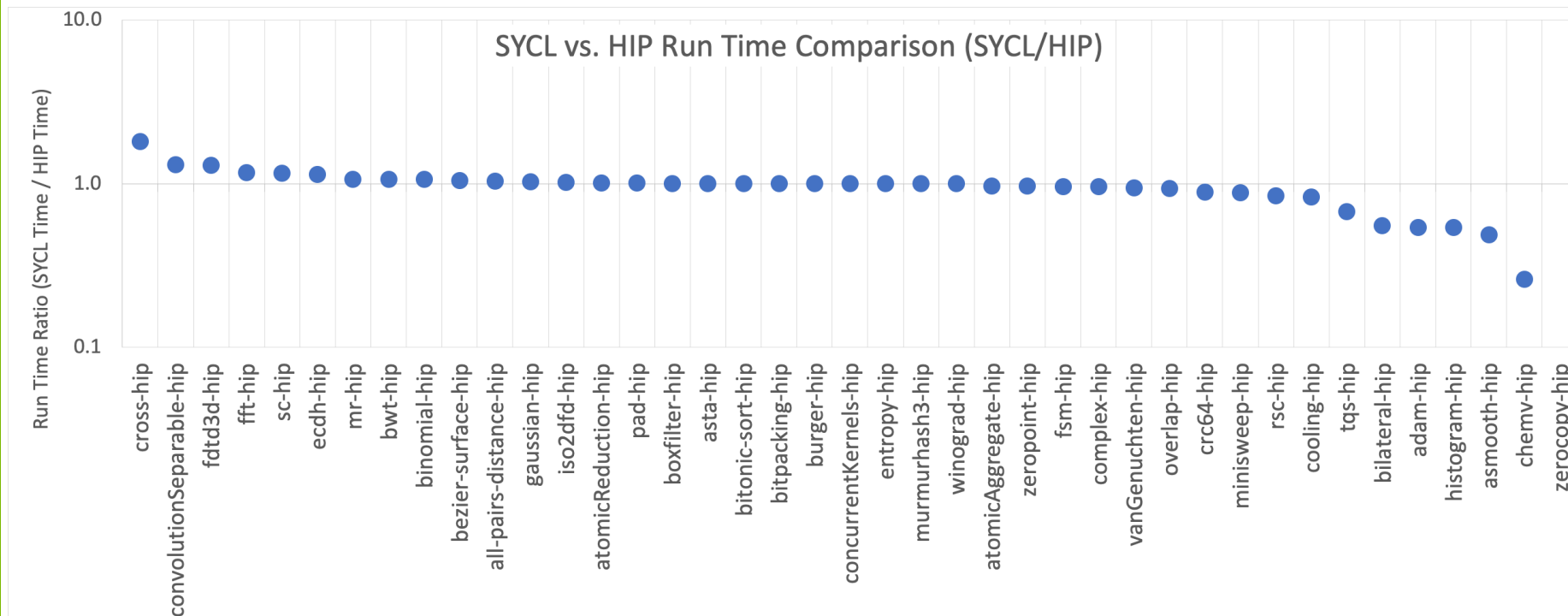


HECBENCHMARK: HIP FUNCTIONALITY

- Running chipStar on PVC, 88.12% of (230/261) HIP benchmarks compiled successfully, and within the compiled ones, 85.65% (197/230) ran successfully. For those ran and had result verification, 92.39% (85/92) passed.
- The errors are a variety of Intel runtime errors, hangs, incorrect answers, and chipStar errors



HECBENCHMARK: HIP PERFORMANCE ON PVC



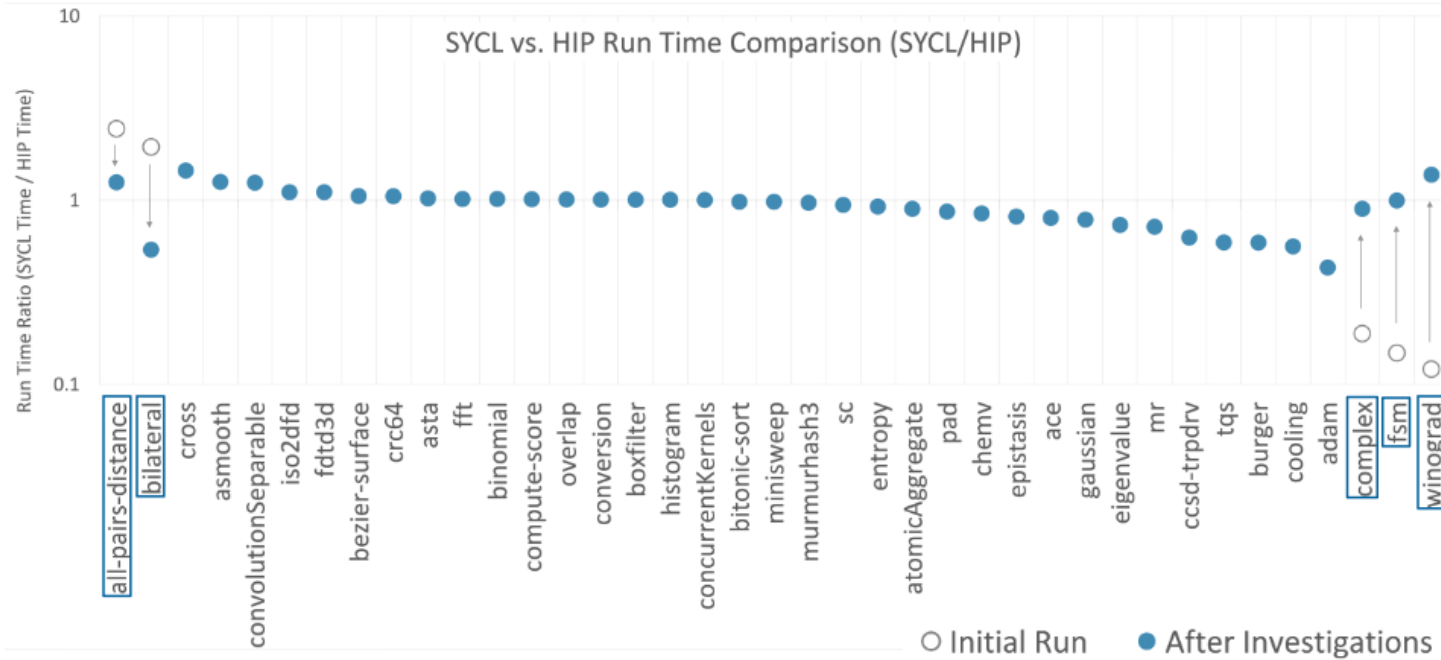
Note:

- > 1.0 means HIP is better
- Close to 1.0 means HIP and SYCL are similar
- < 1.0 means HIP is worse

HECBENCHMARK: PERFORMANCE ON GEN9

PERFORMANCE COMPARISON

Using HeCBench subsets: SYCL version 2023.1 (with icpx compiler) and HIP (with chipStar) on Intel Gen9 GPU. We expect results to be close to 1 for similar performance between the two. Overall, the average run time ratio is around 0.91, indicating similar run time to SYCL, which has a slightly better performance.



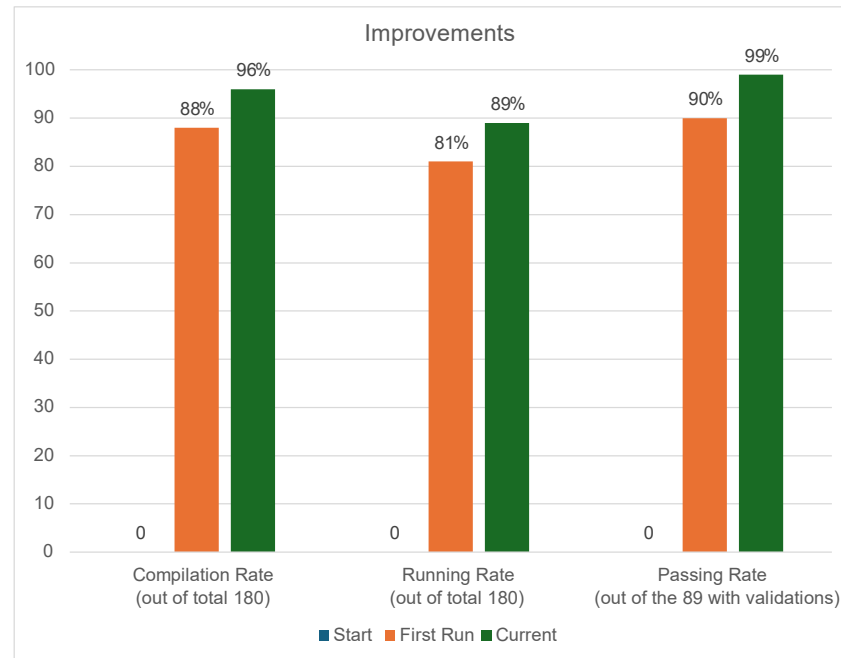
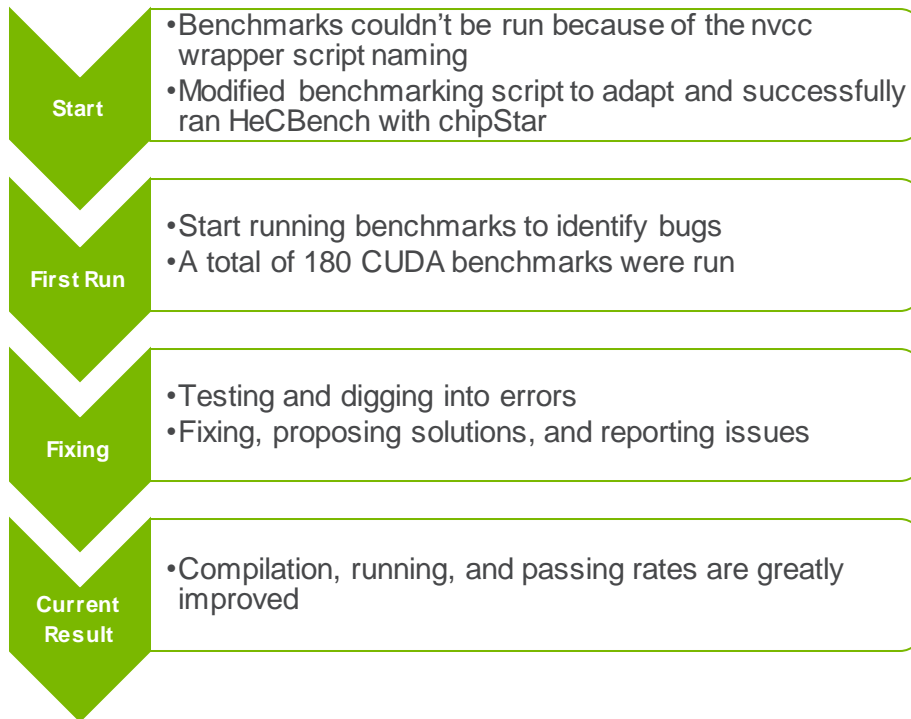
EVALUATING CUDA SUPPORT WITH HECBENCH



Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.

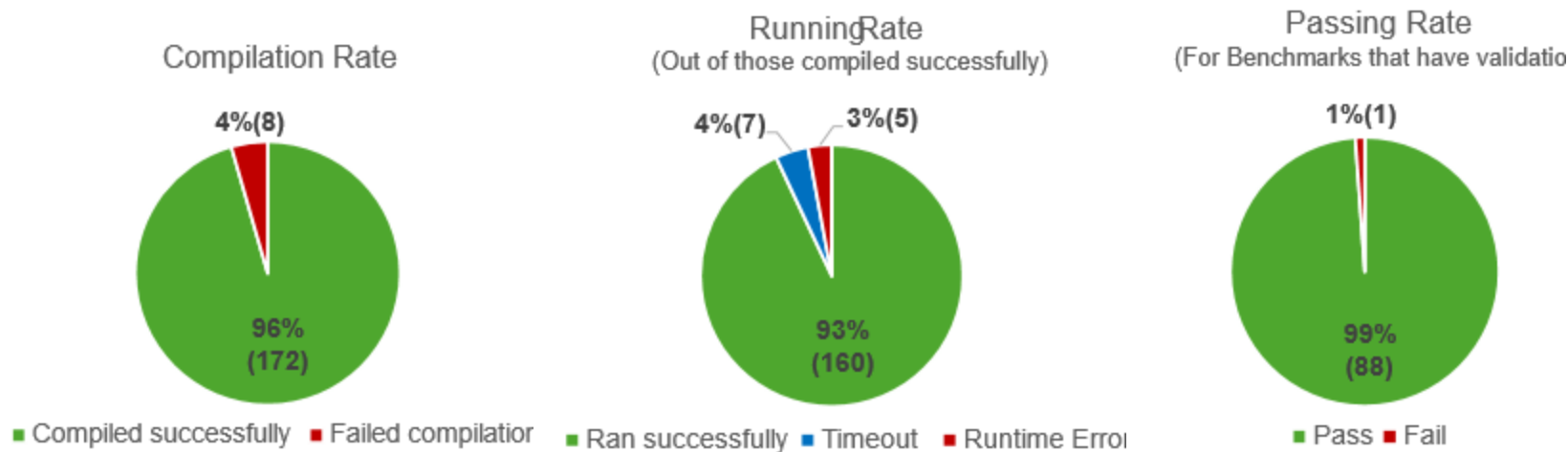


HECBENCHMARK: CUDA FUNCTIONALITY



RESULTS

Overall Results – Compilation, Running, Passing Rates



APPLICATION HIGHLIGHT: GAMESS



Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.



GAMESS: INTRODUCTION

- General Atomic and Molecular Electronic Structure System (GAMESS) is a quantum chemistry software package
 - Contains a CUDA/HIP library targeting Summit and Frontier
- To port to chipStar, we focused on the Hartree-Fock (HF) method.
 - Over **20,000 lines of HIP/CUDA kernel code** and multiple hipBLAS and hipSOVLER routines (**hipblasDscal, hipblasDgemm, hipblasDcopy, hipblasDaxpy, hipblasDdot, hipblasDgemv, hipblasDgeam, hipsolverDsyevd.**)
- The kernels use **shared memory with __syncthreads()** calls to ensure copying values from global memory to shared memory completed for the threadblock before using it.

GAMESS: TIMING COMPARISON

- Porting effort was relatively low
 - Build files are very similar
 - We could use the same cmake for HIP on Nvidia, AMD, and Intel HW
 - One issue was due to kernel thread synchronization: In CUDA group barriers don't count exited threads but in LevelZero and OpenCL this is a undefined behavior and can deadlock

| Hardware | SCF time (s) |
|----------------------|--------------|
| Nvidia A100 | 1.66 |
| AMD MI250 (one GCD) | 4.71* |
| Intel PVC (one tile) | 2.84 |

* This is due to the eigensolve in hipBLAS being relatively slow.

CONCLUSION



Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.

CONCLUSIONS

chipStar is HIP for Aurora

- chipStar 1.0 (delivered almost 50 ECP MS)
 - Working HIP implementation
 - Supporting selected applications and libraries
 - Competitive performances with SYCL
- chipStar 1.1, focusing on performance improvements
- chipStar 1.2, experimental CUDA, improve performance and stability
- Deployed on our platforms, can leverage Level Zero or OpenCL
 - Aurora (PVC)
 - JLSE (PVC, Iris)
- chipStar is used in other projects:
 - HipScript (HIP/Cuda on WebAssembly)
 - RISC-V target

PERSPECTIVES

Exciting Possibilities

- Native CUDA support
 - Experimental but promising
- Support for more
 - Applications: collaborating with new teams
 - Features: HIP is constantly evolving, focusing on Applications requirements
 - Libraries as required by applications
 - Platforms
 - Extensions to OpenCL to better support HIP and CUDA
 - ARM is under investigation
 - Rusticl is working: universal Linux support coming
- Looking for Additional funding

ACKNOWLEDGMENTS

ACKNOWLEDGMENTS

- Thanks to many contributors: Paulius Velesko, Pekka Jaaskelainen, Henry Linjamaki, Michal Babej, Peng Tu, Sarkar Sarbojit, Ben Ashbaugh, Colleen Bertoni, Jenny Chen, Philip C. Roth, Wael Elwasif, Rahulkumar Gaytri, Jisheng Zhao, Karol Herbst, Kevin Harms, Brice Videau
- This research used resources of the Argonne Leadership Computing Facility, a U.S. Department of Energy (DOE) Office of Science user facility at Argonne National Laboratory and is based on research supported by the U.S. DOE Office of Science-Advanced Scientific Computing Research Program, under Contract No. DE-AC02-06CH11357.
- This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative