TOOLS TO DIAGNOSE AND REPAIR FLOATING-POINT ERRORS IN HETEROGENEOUS COMPUTING HARDWARE AND SOFTWARE

An SC’24 Half-Day Tutorial

ODYSSEY

Ben Wang
Ed Misback,
University of Washington

CIEL

Cindy Rubio-Gonzalez
Dolores Miao,
University of California, Davis

GPU-FPX
FTTN

Ganesh Gopalakrishnan
University of Utah
Xinyi Li
Pacific Northwest National Laboratory

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

UC DAVIS
UNIVERSITY OF CALIFORNIA

KAHLERT SCHOOL OF COMPUTING
THE UNIVERSITY OF UTAH

Lawrence Livermore National Laboratory

Pacific Northwest
NATIONAL LABORATORY

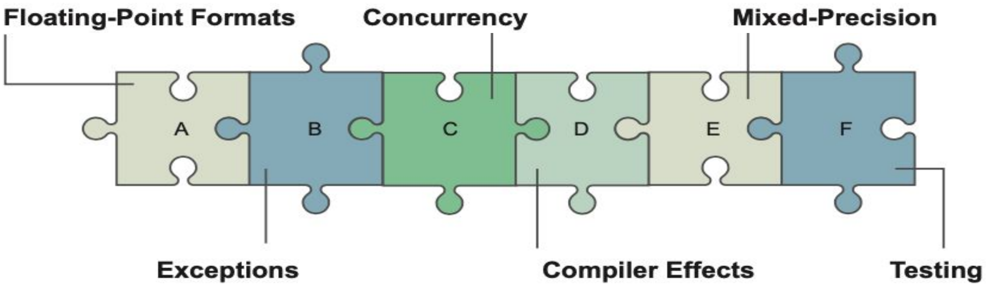
Project Details, Funding Support

ComPort is a DOE XStack Research Project supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research ComPort: Rigorous Testing Methods to Safeguard Software Porting under Award Numbers

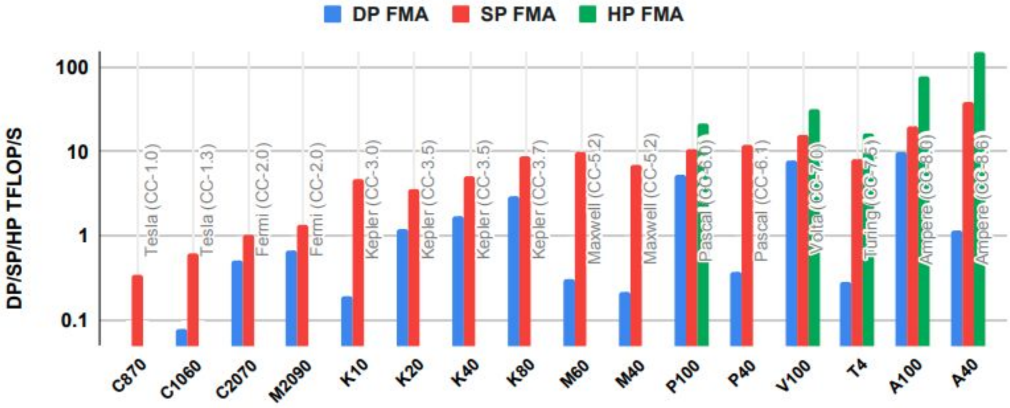
- DE-SC0022252 (University of Utah, PI: Ganesh Gopalakrishnan, and Co-PI: Pavel Panchekha)
- SCW 1743 (Lawrence Livermore National Laboratory, PI: Ignacio Laguna)
- SCW 78284 (Pacific Northwest National Laboratory, PI: Ang Li)
- DE-SC0022182 (University of California, Davis, PI: Cindy Rubio-Gonzalez)
- DE-SC0022081 (University of Washington, PI: Zachary Tatlock)

- Webpage for our Project: https://bit.ly/XStack-ComPort-Project
- Question? Email ganesh@cs.utah.edu

Concerns in the Floating-Point Arithmetic



"Guarding Numerics Against Rising Heterogeneity", SC Correctness Workshop 2021 by the PIs



- Basic misunderstandings about FP prevail
  - Need tools to understand, improve code
- Heterogeneity (CPUs/GPUs) the norm
  - Rapidly changing in features
  - AMD GPUs also on the rise
    - Unknown repro when porting
- Mostly undocumented building-blocks
  - Libraries are binary-only (in undocumented assembly-level ISA)
  - Compilers differ, especially across optimization levels
- Various Precision Choices
  - FP16, FP8
- Built-in Acceleration for matrix operations
  - Tensor cores (NVIDIA)
    - Not fully IEEE compatible
  - Matrix cores (AMD)
- No hardware trapping of exceptions in NVIDIA
  - AMD can trap
    - Effort to use it is non-trivial

Units and Presenters, other PIs Involved

- Odyssey
    - An Interactive Workbench for Floating Point Analysis
        - Zachary Tatlock, PI at UW
        - Pavel Panchekha, PI at Utah
        - **Presenters: Edward Misback and Benjamin Wang, UW**
- Ciel
    - Ciel: Expression Isolation of Compiler-Induced Numerical Inconsistencies in Heterogeneous Code
        - Cindy Rubio-Gonzalez, PI at UC Davis
        - Ignacio Laguna, PI at LLNL
        - **Presenters: Dolores Miao, Cindy Rubio-Gonzalez, UC Davis**
- GPU-FPX and FTTN
    - A Low-Overhead tool for Floating-Point Exception Detection in NVIDIA GPUs
    - Feature-Targeted Testing of Numerics
        - Ganesh Gopalakrishnan, PI at Utah
        - Ang Li, PI at PNNL
        - **Presenters: Ganesh Gopalakrishnan, Xinyi Li, PNNL**

Goals : Understand Floating-Point Arithmetic
Mitigate Ill-Effects of FP Error, Exceptions

Numerical errors are rare,

rare enough not to care about them all the time,

but yet not rare enough to ignore them.

– William M. Kahan

Floating-Point Behavior Can Be Confusing

> Numerical errors are rare, rare enough
> not to care about them all the time,
> but yet not rare enough to ignore them.
> — William M. Kahan

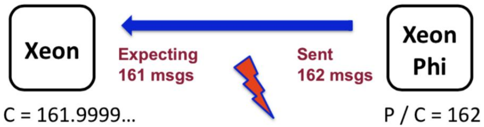
Examples of recent FP errors

- Simulation in the Large Hadron Collider
  - Need to track charged particles with exquisite precision
    - **10 microns** over **10 meters**
  - Round-off resulted in **missed** / **mis-identified** collisions
    - (cf. Bailey and Borwein)

P = 0.421874999999999944488848768742172978818416595458984375
C = 0.0026041666666666666522106377001932742132339626550674438476562 5

Compute: floor(P / C)

From a Uintah paper by Meng Humphreys Berzins. Caused MPI Deadlock

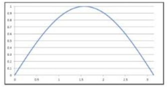


P / C = 161.9999...
floor(P / C) = **161**

P / C = 162
floor(P / C) = **162**

- Intel issues “specification update” for trig library
  - Originally guaranteed to have a **one ULP error**
  - Measured error was **164-billion ULPs**
    - **37 bits** of the mantissa were **wrong**

(Bruce Dawson's blog)



This is buggy in FP: $\#\mathrm{define}\ MAX(x,y)\ ((x) \geq (y) ? (x) : (y))$

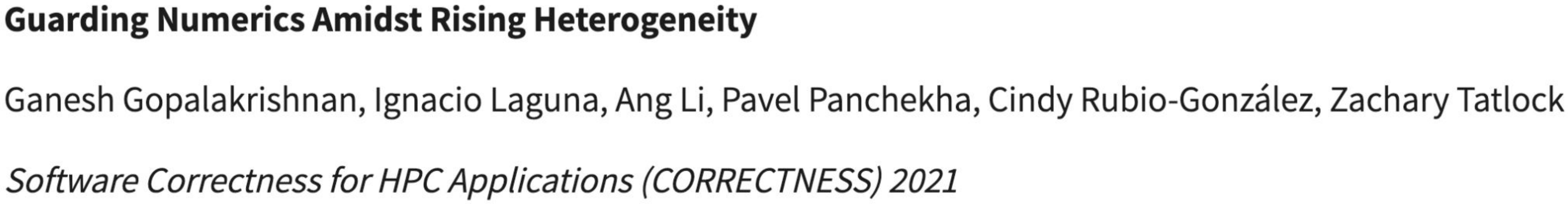
This iterates 10 times in FP32 but 11 times in FP64

```
for (i=0.0; i < 1.0; i+=0.1) { … }
```

it goes 10, 11, 10, 11, … till about 100 bits of precision

With rising heterogeneity, they can rear their heads more frequently

https://ieeexplore.ieee.org/document/9651291



Guarding Numerics Amidst Rising Heterogeneity

Ganesh Gopalakrishnan, Ignacio Laguna, Ang Li, Pavel Panchekha, Cindy Rubio-González, Zachary Tatlock

Software Correctness for HPC Applications (CORRECTNESS) 2021

U.S. DEPARTMENT OF ENERGY

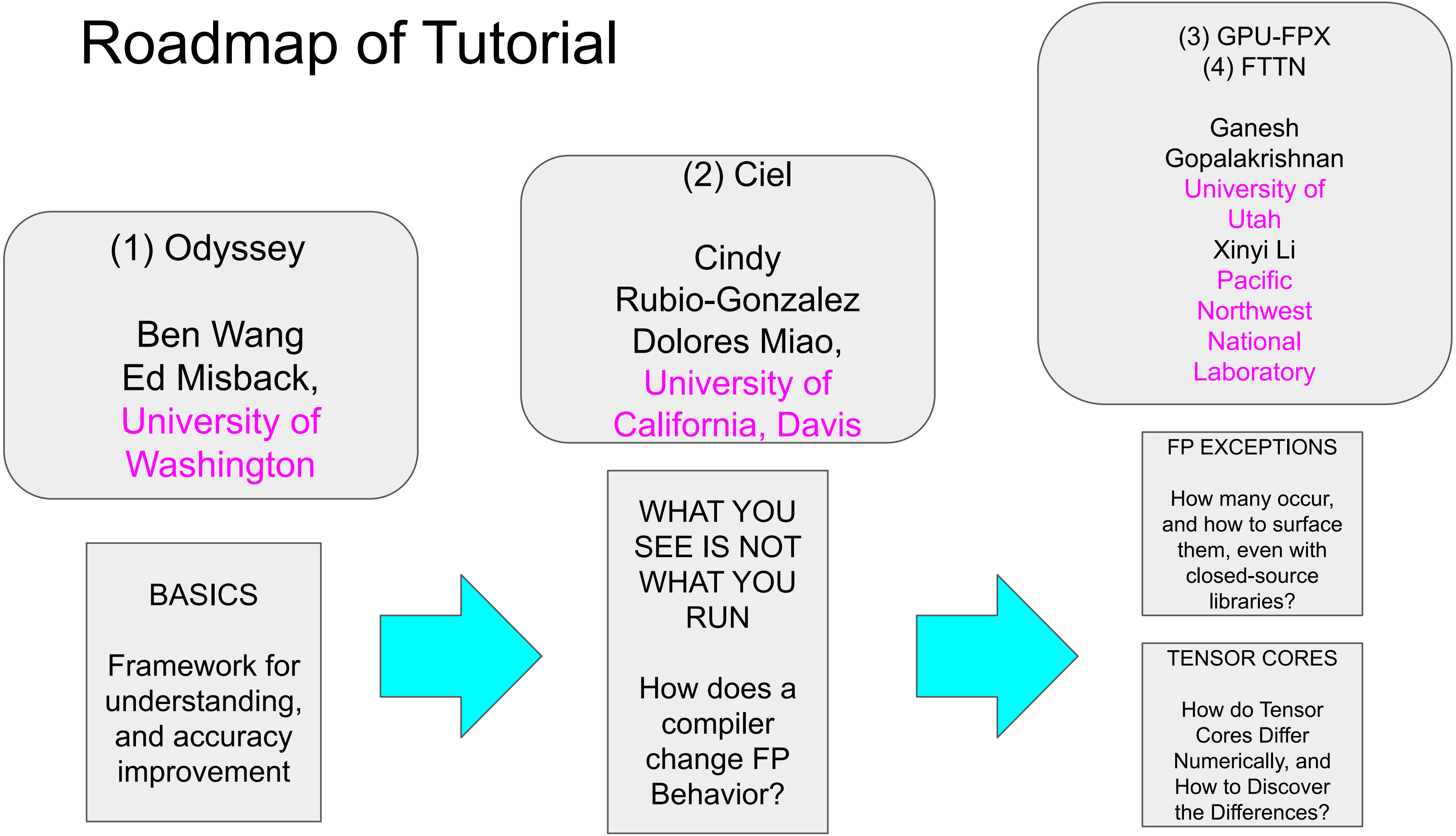
National Science Foundation

DOE/NSF Workshop on Correctness in Scientific Computing

June 17, 2023
Orlando, FL

https://arxiv.org/pdf/2312.15640

Roadmap of Tutorial



(1) Odyssey

Ben Wang
Ed Misback,
University of Washington

BASICS

Framework for understanding, and accuracy improvement

(2) Ciel

Cindy Rubio-Gonzalez
Dolores Miao,
University of California, Davis

WHAT YOU SEE IS NOT WHAT YOU RUN

How does a compiler change FP Behavior?

(3) GPU-FPX
(4) FTTN

Ganesh Gopalakrishnan
University of Utah
Xinyi Li
Pacific Northwest National Laboratory

FP EXCEPTIONS

How many occur, and how to surface them, even with closed-source libraries?

TENSOR CORES

How do Tensor Cores Differ Numerically, and How to Discover the Differences?