Introduction to Floating-Point Analysis and Reproducibility

Ignacio Laguna, Harshitha Menon
Lawrence Livermore National Laboratory

Michael Bentley, Ian Briggs, Pavel Panchekha, Ganesh Gopalakrishnan
University of Utah

Hui Guo, Cindy Rubio González
University of California at Davis

Michael O. Lam
James Madison University

http://fpanalysistools.org/

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-PRES-780623).

Topics in this full-day tutorial

- Floating-point analysis
  - For improving the performance of the code
- Non-Reproducibility
  - Caused by program execution
  - Caused by floating-point result nondeterminism
- Why combine FP analysis and Non-Reproducibility in one tutorial?
    - Because, separating them through external symptoms alone is difficult in practice!

Reproducibility

The hope/expectation that a subsequent run of the program under “identical conditions” will produce the same answer

http://www.clipartpanda.com/clipart_images/elevator-clipart-69356901

Clipart Info

Use this image on your Powerpoints, School Projects, Reports and More!

Life works around reproducibility

Even babies expect it

Lecun’s Turing lecture

Reasons for Lack of Reproducibility in Numeric Programs, and Tools We Present

- Nondeterministic MPI message matching
  - Send messages that race to match a wildcard receive
    - TOOL: **ReMPI**
- Data races that change the space of executions
  - “Pink-elephant values” [Sutter]
    - TOOL: **Archer**
- Compiler optimizations that change the binaries that get linked
  - Makes code non-portable across platforms
    - TOOL: **FLiT**
- FP Exceptions that are thrown
  - Makes the execution unexpectedly terminate
    - TOOL: **FPChecker**

Floating-point Analysis Topics and Tools

- Understanding and Benchmarking Floating-Point Codes
  - So that we have an objective truth to tool behavior, comparisons
    - FRAMEWORK: FPBench
- Precision Tuning
  - Determine precision changes that improve performance
    - TOOLS: Precimonious, HiFPTuner
- Framework for source-to-source precision tuning & analysis
  - Enables community to conduct precision tuning research
    - TOOLS: ADAPT, FloatSmith

A quick introduction to today's major topic

Some basics of Floating-Point Arithmetic

The Floating-Point number system is not new

Then

Zuse Z1 (~1938)

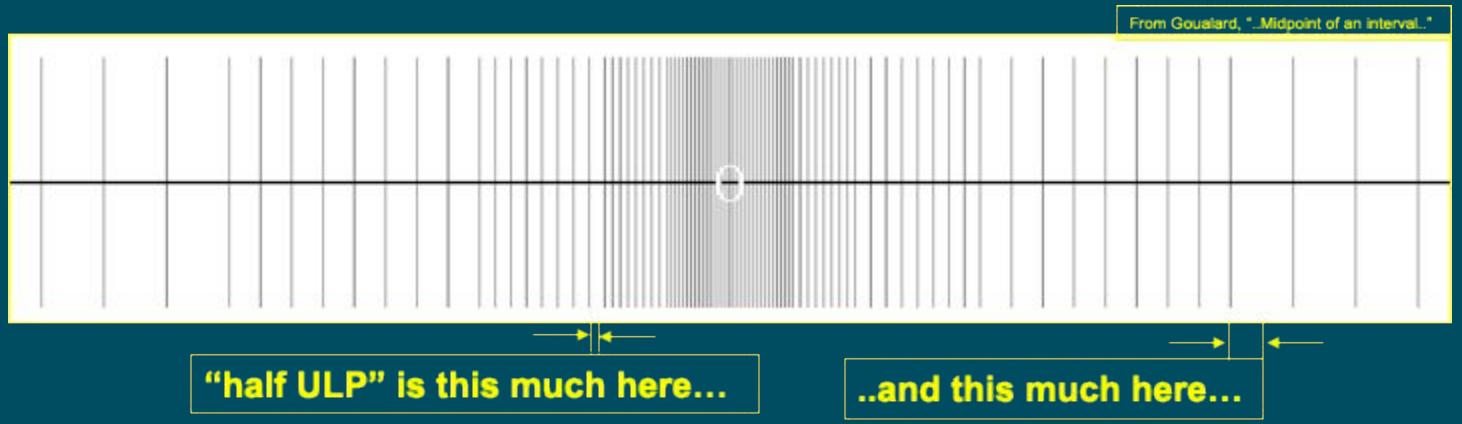
Now

IEEE Standard for Floating-Point Arithmetic

Floating-Point approximates Reals

- Because of rounding, (x+y)+z != x+(y+z)
  - And many more such identities are violated
- Compilers can change your math
  - x/y → x * (1/y)
- Rounding errors are non-intuitive
  - Because of the uneven FP number scale

The Floating-Point Rounding is Non-Intuitive



The FP number system tries to span a large range using an “insufficient number of bits”

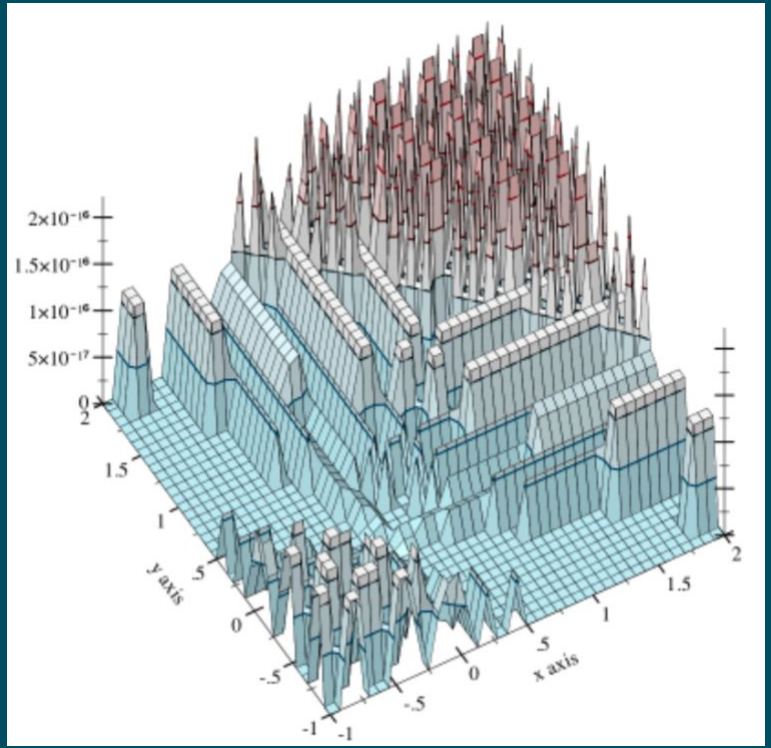
The FP error function is highly non-intuitive

E.g.

Rounding error of (x+y) as a function of x and y

IEEE Model of error:

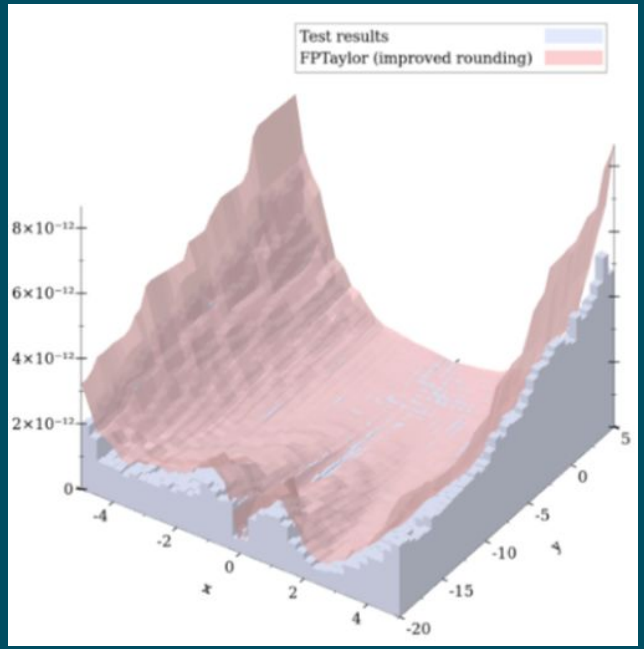
(x+y) . half-ULP



The FP error function is highly non-intuitive even at a more macroscopic level… Heuristic solutions recommended in practice!

E.g. Rounding error of the JetEngine benchmark of Darulova and Kuncak as plotted by Solovyev using his FPTaylor tool, with ground-truth obtained thru shadow-value simulation.

```
double jetEngine(double x1, double x2) {
  double t = 3 * x1 * x1 + 2 * x2 - x1
  double q = x1 * x1 + 1
  double p = t / q
  double s1 = 2 * x1 * p * (p - 3)
  double s2 = x1 * x1 * (4 * p - 6)
  double s3 = 3 * x1 * x1 * p
  double s4 = x1 * x1 * x1
  double s5 = 3 * p
  return x1 + ((s1 + s2) * q + s3 + s4 + x1 + s5)
}
```



http://fpanalysistools.org/

Kahan’s observation

> 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 Analysis is Suddenly “Front and Center” in HPC + many other areas

- Allocating needlessly high precision increases data movement
  - Multiple precision types are on the rise
    - Often driven by ML
- The variety of hardware is increasing
  - GPUs and other accelerators
    - Their normal behaviors as well as EXCEPTIONS are on the rise
- Compilers exploit floating-point in an increasing number of ways
  - Compiler flags mean different things
    - Compilers may heed your flags selectively

Frenetic pace of FP research now

- Multiple conferences
- Many sessions per conference
- Many different issues

Very little that is tangible for a practitioner to try some of these out

PURPOSE OF THIS TUTORIAL: Change this!

Goals of this Tutorial

- Introduce FOUR mileposts in your repertoire of knowledge
  - Four tools you can practice during the tutorials
  - You can apply them in your own projects!
- We are a resource you can count on during your future work
  - We are invested in multiple research projects in this area
  - We know many more researchers and practitioners whose work we can refer

We hope to build a community of researchers and practitioners

Access to AWS Instances (changeme)

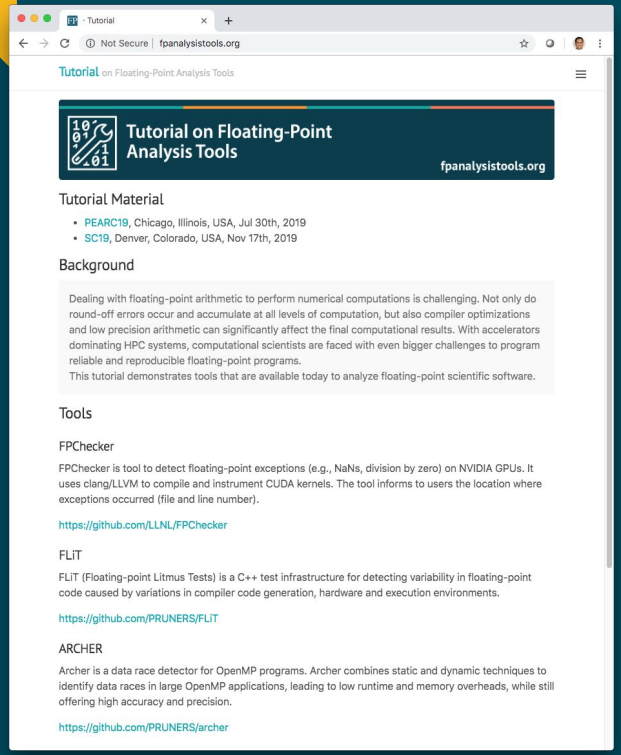
- You will be given access to AWS instances
  - User, password, and IP address will be provided
- How to access your instance:

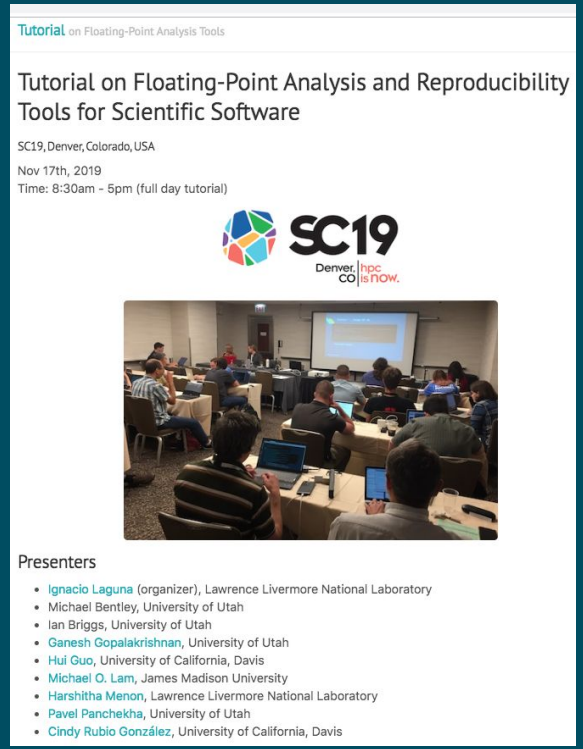
```
ssh user@1.2.3.4
```

- Exercises for each module located in user's /home directory

```
/home/user1/
   |---Module-TOOL1
      |---exercise-1
      |---exercise-2
      |---exercise-3
   |---Module-TOOL2
      |---exercise-1
      |---exercise-2
      |---exercise-3
   ...
```

Website & Schedule: fpanalysistools.org





http://fpanalysistools.org/