Formal Methods for HPC: Correct Dosage

Ganesh Gopalakrishnan
School of Computing, University of Utah

http://www.cs.utah.edu/fv

Formal Methods

An apologetic term for using math when most people aren’t

“I am computing the Happens Before of this MPI execution!”

Turns out that there are **four** event types: Issue, Return, Match, Complete

...and MPI’s main ordering relation is “Matches Before”

Formal Methods

A redundant term to be using when everyone is using math

“This bridge was designed using math”

From Wikipedia

John Rushby’s wish: “Disappearing Formal Methods”

(nobody writes anymore “this phone has 1B transistors”)

Formal Methods (Formal Verification, or “FV”)

An unquestioned design validation phase for any chip today!

Hardware failures deemed too expensive that FV pays off!

From Wikipedia

A - (A/B) * B   not always 0

Even for some FP values A and B without any fractional part, it was as high as 256 !! :-(

Formal Methods (Formal Verification, or “FV”)

A missed opportunity in many of today’s SW design ventures

JavaScript was designed in 10 days --- let alone, formalized

(“You can hit ‘send’ to an email and the browser may say

the email went, when it did not” -- Pattabiraman [UBC]”)

Alan Turing was the first to use Formal Methods

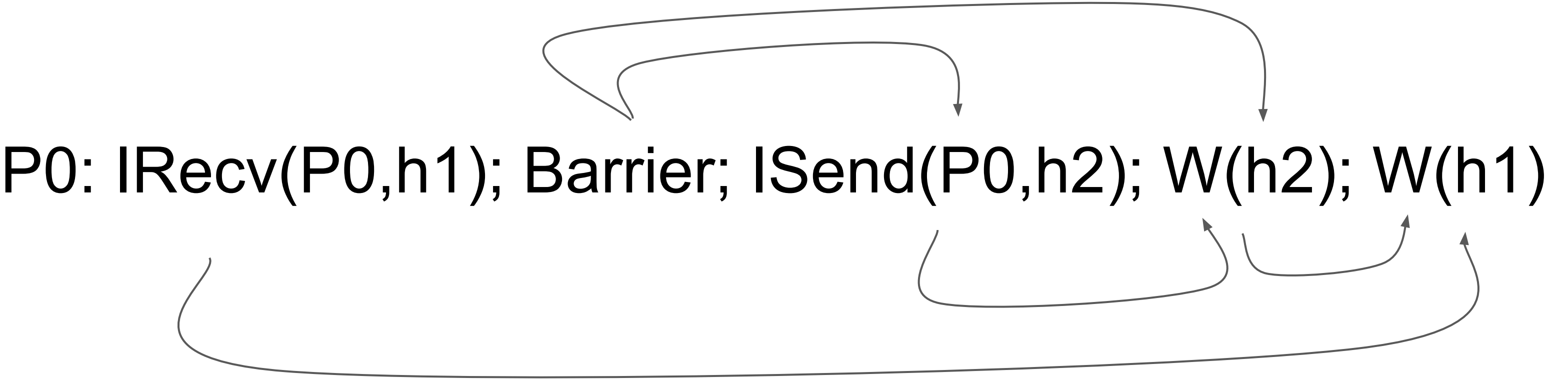
- Described a universal model for computing

- The first to write a hand-proof for a program!

First formal methods exercise wrt an algorithm

Formal Methods Success Stories in HPC (< 2010)

- Systematic analysis of MPI programs
  - Lusk et al, Siegel / Avrunin : Models of MPI
  - Vakkalanka / Vo / Sharma / Gopalakrishnan
    - Developed a “happens before” model for MPI
    - Can calculate the behavior of MPI programs!



P0: IRecv(P0,h1); Barrier; ISend(P0,h2); W(h2); W(h1)

Matches-before relation for this short MPI program

Formal Methods Success in MPI

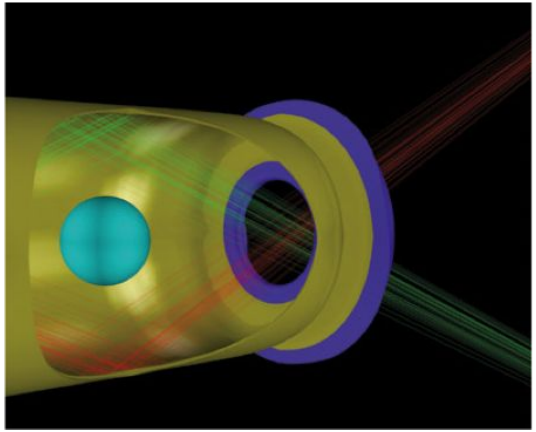
(Demo of Lamport’s Happens-Before based Active Testing

One minute)

Recent Successes (1)

Data Race Checking of OpenMP Programs

- Archer tool released (see http://www.cs.utah.edu/fv)
- Found race in Hydra
  - Overall 1M LOC
  - Issue found in Hypre component
  - Designers had tried to parallelize
  - Used to fail under higher optimization
    - Turned off parallelization!
  - Archer spotted "Benign Race"
    - Was failing under optimization



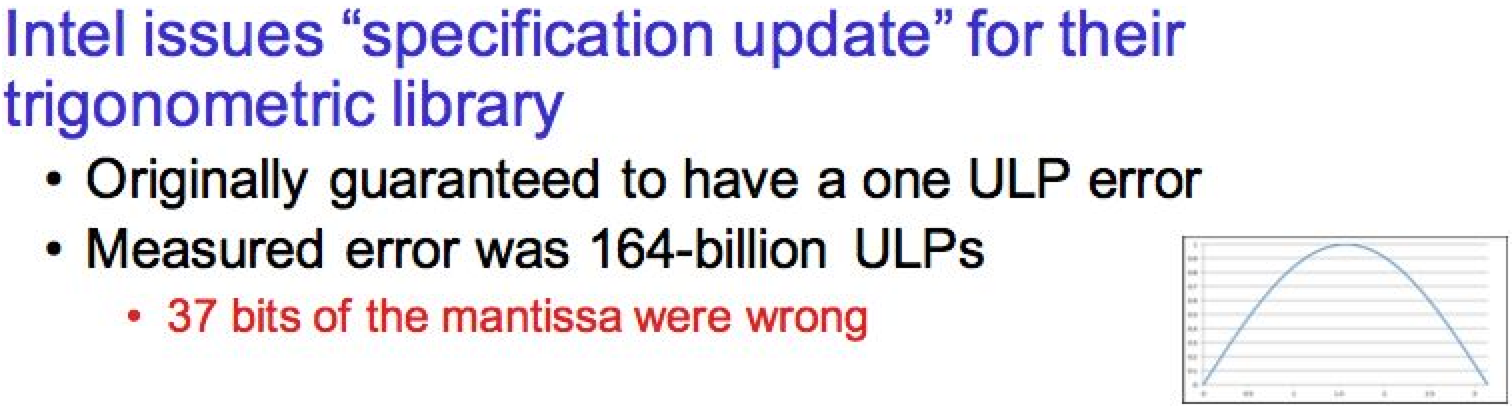
Other recent successes (2, 3, and 4)

- Siegel’s CIVL language
  - Formal intermediate language for many concurrency models
  - Found bug in “fixed” LLNL OpenMP tutorial material
- Session types for MPI
  - Interesting work by Yoshida (Imperial) and Vasconales (U of Lisbon)
- Modeling and analysis of Remote Memory Access Programming (Dan, Lam, Hoefler, Vechev, OOPSLA’16)
- GPU Race Checking
  - GKLEE (Utah), GPUVerify (Imperial)

Dosage of “formal” varies...

Recent Successes (5/5)

Floating-point Precision Tuning (Chiang, Solovyev, Briggs, Baranowski, Gopalakrishnan, Rakamaric’, POPL’17)



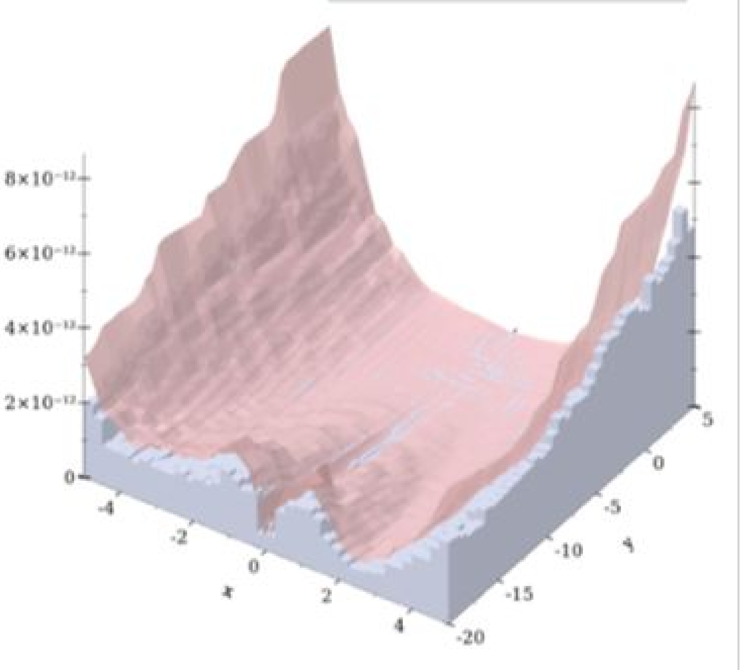
Intel issues “specification update” for their trigonometric library

- Originally guaranteed to have a one ULP error
- Measured error was 164-billion ULPs
  - 37 bits of the mantissa were wrong

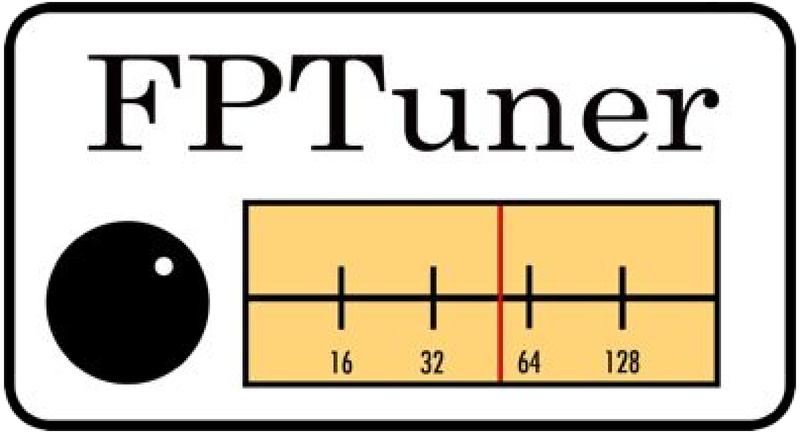
Formally compute error surface

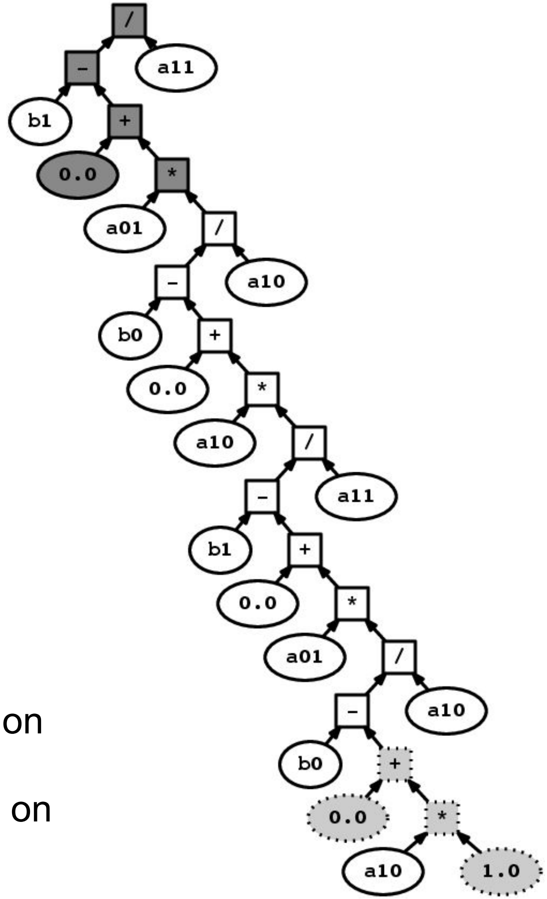
then allocate precision using QCQP

```
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)
}
```



Our Formal FP Precision Tuner FPTuner (POPL’17)





Other tools (Precimonious, Craft, …) tune for given 100 (or so) training points. We tune for entire input intervals, and Guarantee that the error bounds are met!

Payoff is due to FORMAL modeling of floating-point using Symbolic Taylor Forms and the use of Rigorous Global Optimization

We also show that energy consumption is reduced (by measuring on real HW)

Concluding Remarks

- Pedagogical value is unquestionable!
  - Just having the mind-set of calculating races and deadlocks using an understanding of Happens Before is uplifting
    - Value cannot be under-estimated
- It is catching on!
  - And many conferences are welcoming formal modeling of HPC
- No software ENGINEERING without engineering math
  - Formal methods are the engineering math!
- For details from us, please visit
  - http://www.cs.utah.edu/fv