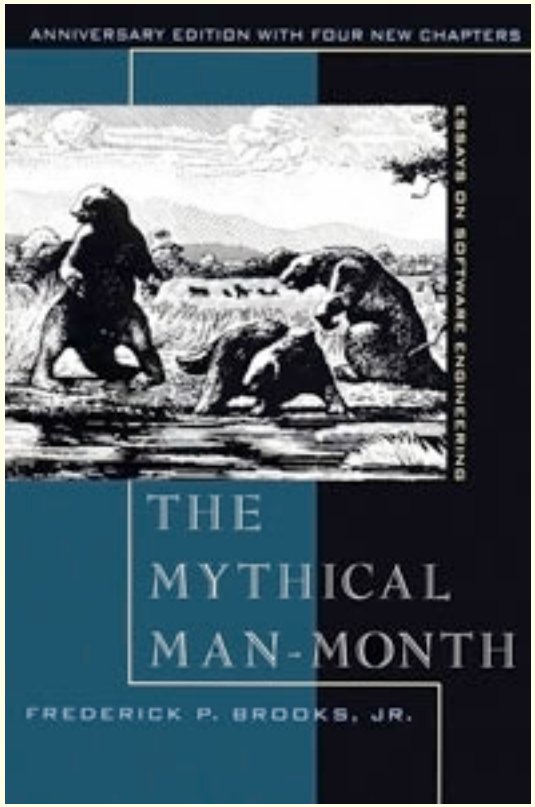
Fortran GPU Compilers: Improving But No Silver Bullet

Tom Henderson
NOAA Global Systems Division
Thomas.B.Henderson@noaa.gov

Mark Govett, Jacques Middlecoff
Paul Madden, James Rosinski,
Craig Tierney

Thunder Stolen?

- I was planning a rant but…
  - Compiler vendors are already responding to my whining!
- Bullet is not silver yet…
  - Forecast: more whining



Outline

- The despair of “Tightly-Nested Outer Loops” (TNOL)
- The joy of bitwise-exact comparison
- The ongoing agony of data transfers

TNOL

- Commercial directive-based Fortran GPU compilers require(d) “tightly-nested outer loops” (TNOL)
  - Forces extensive restructuring of legacy codes
  - Restructuring may promote arrays increasing memory footprint
  - Not a limitation for F2C-ACC

```
! This is OK
do ipn=1,nip
  do k=1,nvl
    <statements>
  enddo
enddo
```

```
! This is NOT OK
do ipn=1,nip
  <statements>
  do k=1,nvl
    <statements>
  enddo
enddo
```

TNOL

- Created NIM “vdmintv” stand-alone test
  - Key NIM subroutine (25% of wall-clock time)
- TNOL requires promotion of temporary arrays to higher rank
  - 2.5x slow down on CPU!

TNOL

- GPU optimization via F2C-ACC and NVIDIA's Paulius Micikevicius
  - Paulius identified best possible CUDA solution
- TNOL costs ~15% on GPU
  - Comparing fastest schemes using mixes of GPU "shared" and "local" memory

Compiler Vendor Response

- Strong response for TNOL
  - Cray
  - CAPS
  - PGI
  - (PathScale)
- Eventual fix in OpenACC
  - Expect approaches to converge

Compiler Vendor Response

- Some progress on more advanced optimizations like shared/local memory
  - Cray
    - Shared memory, close to F2C-ACC performance
  - CAPS
    - Shared memory, close to F2C-ACC performance
  - PGI
- Different approaches now

Outline

- The despair of “Tightly-Nested Outer Loops” (TNOL)
- The joy of bitwise-exact comparison
- The ongoing agony of data transfers

Bitwise-Exact Comparison

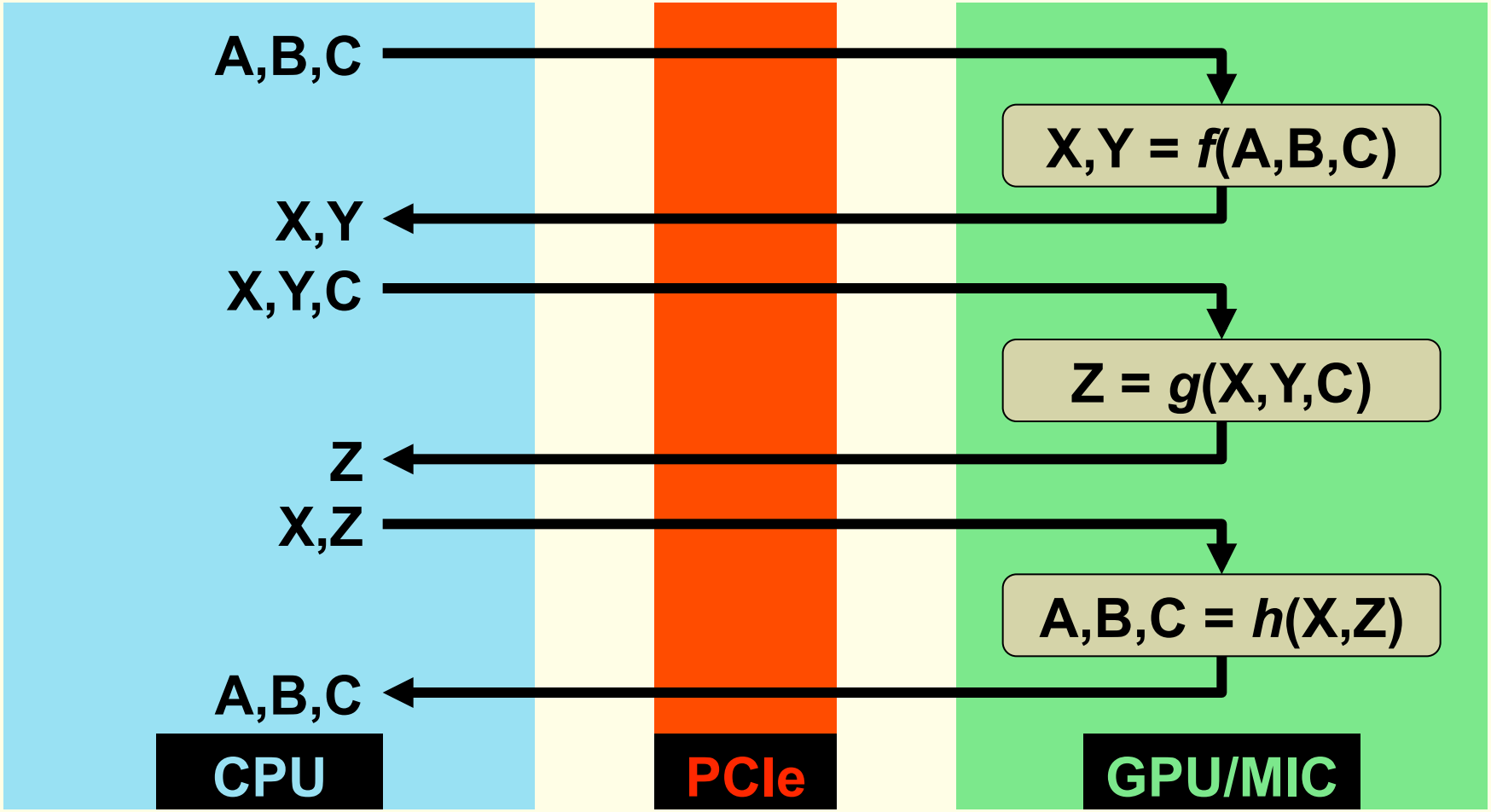
- As of CUDA v4.2 and F2C-ACC v4.2 bitwise-exact comparison between CPU and GPU can be achieved!
  - nvcc compiler flags
    - “-ftz=true –fmad=false”
  - Avoid library functions including “pow” (**)
- Greatly speeds up debugging
  - NIM now has a run-time switch to run “**” operations on CPU for automated testing

Outline

- The despair of “Tightly-Nested Outer Loops” (TNOL)
- The joy of bitwise-exact comparison
- The ongoing agony of data transfers

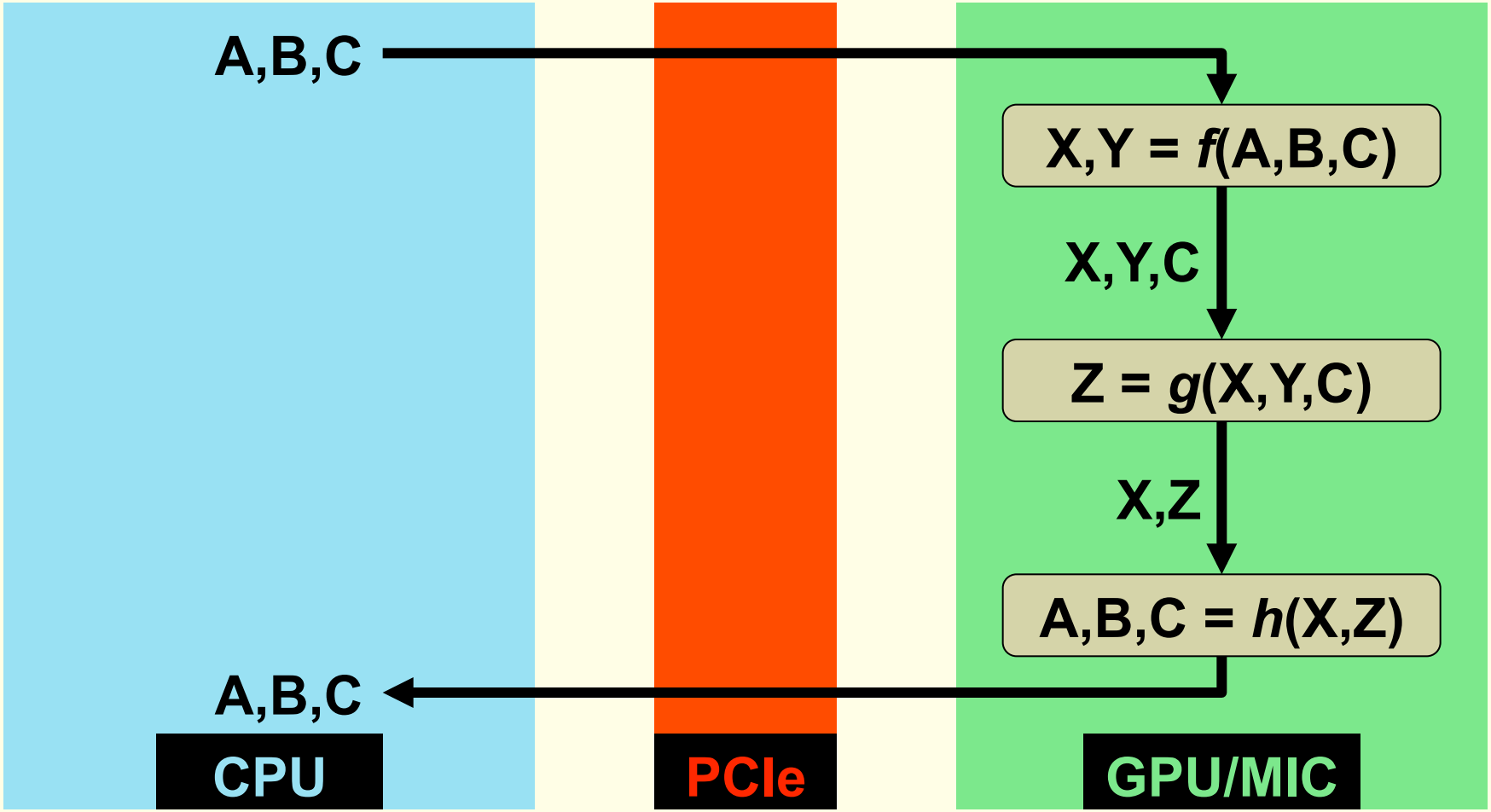
Host-Device Data Transfers

- "Accelerator" model is well-supported



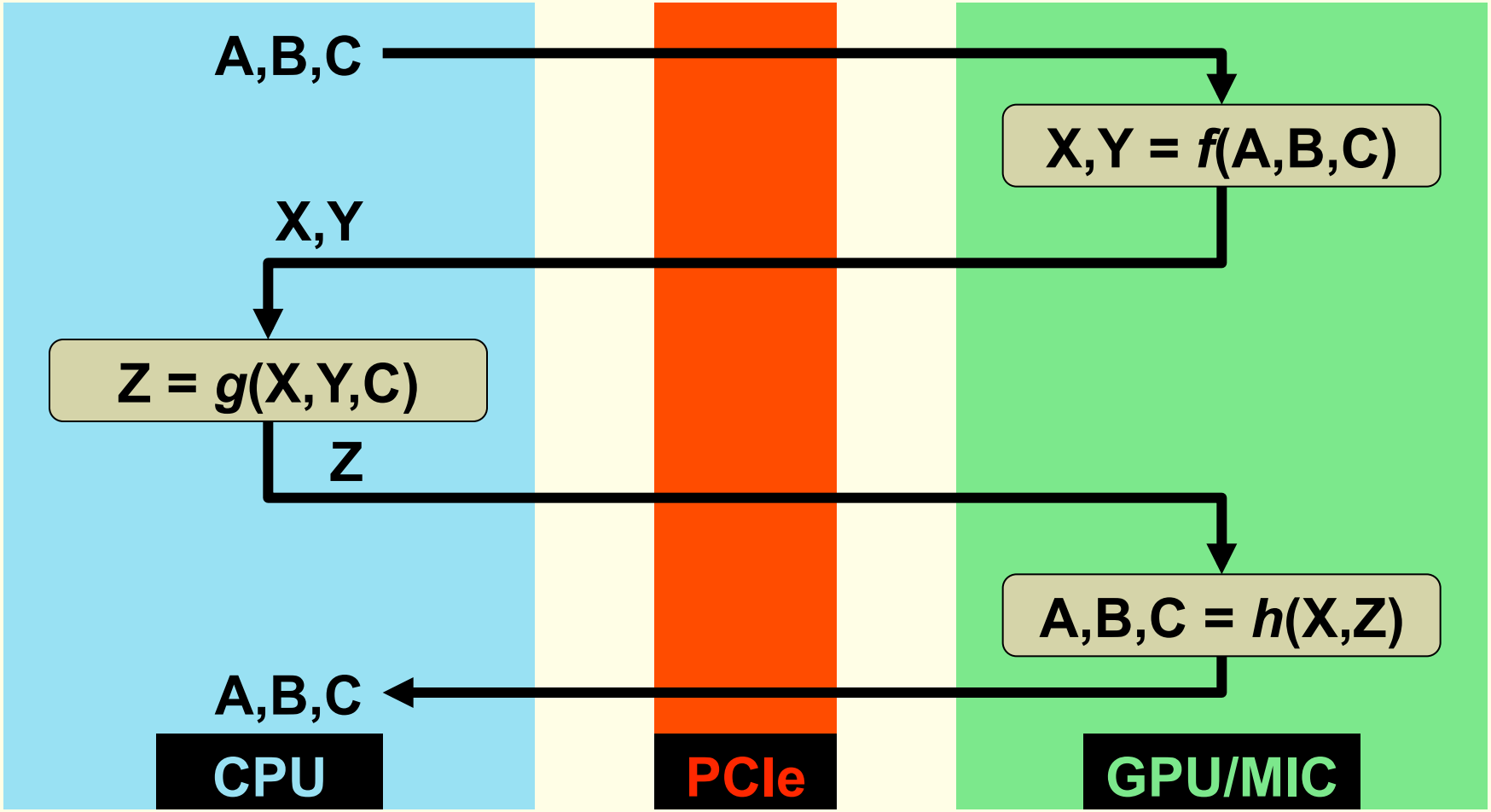
Host-Device Data Transfers

- "State on Accelerator" is a bit harder



Host-Device Data Transfers

- Per-kernel validation is painful!



Please Make Data Transfers Easier

- Compiler has all the information it needs via directives
- User should be able to say “data lives here, run the kernel there”
  - Reduce “accidental complexity”
- Stop the whining!

Thanks to…

- Francois Bodin, Guillaume Poirier, and others at CAPS for assistance with HMPP
- Pete Johnsen at Cray for assistance with Cray OpenACC GPU compiler
- Dave Norton and others at PGI for assistance with PGI Accelerator
- Paulius Micikevicius at NVIDIA
- We want to see multiple successful commercial directive-based Fortran compilers for GPU/MIC

Thank You