Mitigating Extremely Slow Corner Cases in the GNU Math Libraries on TLCC2 and BG/Q

LC User’s Meeting
Nov 13, 2014

John Gyllenhaal, Livermore Computing
In ~2008 the GNU math library decided to sacrifice performance for accuracy

- Drops into 768 bit emulation mode for some inputs
  - For BG/Q (Seq) bgxlc -O3 -qstrict or bgxlc -O2 or bggcc -O3
    - pow (0.9999999999999998L, 1.5) -> 898 cycles
    - pow (0.9999999999999999L, 1.5) -> 8,675,748 cycles
    - exp (0.111E-15L) -> 374 cycles
    - exp (0.11102230246E-15L) -> 1,675,327 cycles
  - Hits TLCC2 (Zin) also for gcc (all optimization levels)
    - pow (0.9999999999999997L, 1.5) -> 1,127,464 cycles
    - exp (0.11102230246E-15L) -> 203,397 cycles

- Recently caused unexplained 4X slowdown on BG/Q

- Diagnostic: STAT & profiling shows descriptive names
  - __slowexp, __slowpow, __mpexp (called by slowexp & slowpow)
Mitigation Strategy:
Don’t Use GNU Math Library!

- On TLCC2 (Zin) with gcc, link to Intel’s Math Libraries
  - Automatic for Intel Compilers (why we haven’t seen much)
  - However gives slightly different answers for these corner cases

- On BG/Q (Seq) link to IBM’s MASS Library
  - -L/opt/ibmcmp/xlmass/bg/7.3/bglib64 -lmass
  - Automatically inlined for IBM Compilers with -O3 if don’t specify -qstrict
  - Slight differences in accuracy possible for ‘exceptional’ values

- Surprisingly tricky to get link line right
  - Must be placed just before first -lm or -lxlfmath
  - Must be placed after everything else (.o files, etc.)

- Link line check for library that provided exp
  - -Wl,-y,exp -> want exp found first in libimf or libmass
  - I can help -> gyllen@llnl.gov, 4-5485

“Mr. Osborne, may I be excused? My brain is full.”