UCX Support for RISC-V 64

TACTICAL COMPUTING LABORATORIES

WWW.TACTCOMPLABS.COM

Christopher Taylor
Senior Research Scientist

RISC-V

- What is RISC-V?
  - Free Instruction Set Architecture (ISA)
  - RISC-V has a ‘core’ suite of extensions (G or RV64G)
  - Other extensions for caches, vector units, virtualization, etc
  - Extensions are “building blocks”
  - Popular in “Internet of Things” (IoT) space

Bottom Line Up Front

- UCX now supports RISC-V 64!
  - PR #8246, “UCX: Adding support for RISC-V architecture; RV64G”
  - SiFive Unmatched cluster with Connect-X 3 & 4
  - OpenMPI, OSSS-UCX verified over Ethernet and IPoIB
  - Ubuntu, Slurm
- Verbs/IB Support Currently Blocked on RISC-V
  - Canonical announced Ubuntu support for RISC-V in 2022
  - Verbs/IB requires kernel support for 32-bit compatibility
  - 32-bit compatibility for 64-bit kernel in-progress; (completed next stable release)

RISC-V

- 32 bit instructions for 64 bit data
  - LUI 20 bits, ADDI 12 bits means 2 instructions to load a 32 bit value
  - 64 bit value takes…4 instructions!
  - Automatic sign-extensions
  - Provides for 16-bit compressed instructions
- Weak memory consistency model
  - Data and instruction caches can be explicitly flushed
  - Performance optimization opportunities
- Base ISA has 32 integer registers
- Floating point (fp) extension provides 32 fp registers
- 16-byte stack frames

Implementation Notes

- UCX requires self-modifying code
  - system calls for memory are binary instrumented
  - memory system calls all have sufficient space to implement instrumentation (`syscall`/`ecall`)
  - ELF relocation support for RISC-V in-progress
- Program Counter (PC) relative jumps
  - RISC-V does not permit direct modification of PC
  - RISC-V specification encourages using AUIPC and JALR
  - AUIPC/JALR resulted in issues with return address (call stack)
  - UCX's binary instrumentation required using ADDI and JAL
  - Special care due to automatic sign extensions

Implementation Notes

- UCX requires self-modifying code
- UCX uses `constructor` attribute
- Memory consistency issues
  - RVG64 base ISA does not have data or instruction cache flush instructions
  - Only provides fences; data cache fences are supervised, instruction cache flushes are unsupervised
  - GCC supports instruction cache flushes; not quite, wraps instruction cache fence instruction
  - Instruction cache fence induces a page table/TLB update
  - Memory management system is implemented in software; MMU hardware would fix this issue
  - POSIX cache flushes unsupported in Ubuntu's RISC-V

Long Term Solutions & Next Steps

- CMO extension
  - RISC-V's CMO extension provides user-land data cache flush
  - CMO extension is not supported in current SiFive HiFive Unmatched
  - CMO extension should be a requirement for HPC oriented RISC-V processors
- File bug reports upstream
  - glibc, clang-libc
  - gcc, clang
  - GNU/Linux kernel

Key Takeaway: UCX currently works on RISC-V for Ethernet/IPoIB and future HW extensions and OS support will bring full functionality!


TCL