FPGAs in Servers

Proposal for discussion

3/24/2015

Jamey Hicks, John Ankcorn, Myron King
Connectal.org

Business Context

- Current data centers built on heavy scaling of generic x86 servers
- But, power envelope limits performance per package and per unit volume
- The processing required for big data applications is often a poor fit for general purpose CPUs
- Therefore, for future growth in data center performance, use of FPGAs seems promising

Problem Statement

- Current FPGA tools assume dedicated use of board by single user, even for trivial designs
- For realistic deployment on servers, infrastructure needed that supports
  - Protected, multi-user use of disjoint regions of FPGA
  - Floor planning of regions based on actual workload (number of LUTs per region, ...)
- To receive wide deployment in the industry, open source reference implementation needs to be widely available

Req’ts for Open Source Reference

- Altera
  - Stratix V, Stratix 10
  - DDR3, DDR4, PCIe gen2, PCIe gen3, 10G ethernet
- Xilinx
  - Virtex7, Virtex Ultra
  - DDR3, DDR4, PCIe gen2, PCIe gen3, 10G ethernet
- Linux
  - Generic device driver, process model, hypervisor integration
- Windows
  - Generic device driver, process model, hypervisor integration
- X86, ARM, other instruction sets?
- Performance within 95% of non-shared deployment

Technical Approach

- Application process can request a coprocessing tile to be loaded
  - Process startup triggers tile load
  - Tile deactivated when process ends
- Tile manager on the FPGA
  - Mediates access to host bus and locally connected components
  - Responsible for QoS, Fairness, Arbitration of resource use by tiles.
  - Misbehaviour in a tile cannot affect other tiles:
    - PCIe transactions buffered. If tile logic not ready, then TLP deleted.
    - If worried about integrity of bitfile contents, perhaps only load ones that have been signed by an authorized build server
- Virtualization
  - Device-driver in the host OS controls the actual hardware
  - Device-driver in the guest OS requests resources from the host OS, i.e., a tile

Work Flow

- Joint discussion to get approximate agreement on problem/approach
- Quarterly review, source always publicly available
- Demonstration implementations are the basis for further discussions
- Refine design and implementation

Tasks

- Construct tile manager with fixed per-tile interface ports
  - Per-tile ports disabled during tile reconfiguration
- Tile manager
  - Configuration and floor planning
  - Partial reconfiguration to load a tile
  - Disable ports without locking PCIe
- Implement tile loader

Releases

- Release 0: Single process Linux (available)
  - Virtex7, DDR3, PCIe gen2
  - Generic Linux driver
  - Single process use of FPGA
- Release 0.1: Tiled Linux release
  - Adds multi-tile use of FPGA
- Release 0.2: Reconfigurable tiles
  - Adds dynamic reconfiguration of tiles
- Release 0.3: Altera support
  - Stratix V, DDR3, PCIe gen2
- Release 0.4: Hypervisor support
  - Xen PVH integration
- Release 0.5: Windows Server 2012
  - Process model, generic device driver, mem mapping
- Release 0.6: Windows hypervisor
  - Which hypervisor?

Topics for Future Features

- Transparent cache support
  - E.g., IBM CAPI
- Multi-FPGA solutions
  - Networking, e.g., Interlaken
  - Deployment and orchestration
- Time slicing of FPGA tiles
  - E.g., IBM CAPI process model
- Debug and trace tools
- Demonstration examples
  - Memcached
  - Filesystem
  - Database acceleration
  - Convolutional Neural Network, other DBNs
  - Genomic applications

Questions

- Do we agree on the problem statement?
- Do we agree on the approach?
- Spatial vs time division multiplexing of FPGAs?
- How deep to integrate with OS?
- Which virtualization modes / hypervisors?