System architecture for network-attached FPGAs in the Cloud using partial reconfiguration

Burkhard Ringlein, Francois Abel, Alexander Ditter, Beat Weiss, Christoph Hagleitner, and Dietmar Fey
1) **Standalone** FPGAs must control themselves
2) Abstract the physical FPGA
3) No disclose of sources required
4) Guarantee **integrity** of the DC
5) Build **clusters** dynamically
6) Reuse / Integrate with existing DC services
System Architecture: Requirements and Proposal

1) **Standalone** FPGAs must control themselves
2) Abstract the physical FPGA
3) No disclose of sources required
4) Guarantee **integrity** of the DC
5) Build **clusters** dynamically
6) Reuse / Integrate with existing DC services

1) **TCP/IP and REST** (Representational State Transfer)
   - have proven to **scale**
   - are **hardware agnostic**
2) FPGA must reconfigure itself
3) Separation of privileges and network interfaces within the FPGA

→ using **partial reconfiguration** via a **RESTful API** based on TCP/IP
System Architecture: Divide and Conquer

• Three levels of Management:
  1) Data Center
  2) per Sled/Chassis
  3) per FPGA

• Results:
  1) < 2% of a Xilinx Kintex
  2) deployment time independent of cluster/application size
  3) single application bit-stream for “zillions” of FPGA nodes
  4) fast configuration
System Architecture: Divide and Conquer

• **Three levels of Management:**
  1) Data Center
  2) per Sled/Chassis
  3) per FPGA

• **Results:**
  1) < 2% of a Xilinx Kintex
  2) deployment time independent of cluster/application size
  3) single application bit-stream for “zillions” of FPGA nodes
  4) fast configuration

Thank you...

ngl@zurich.ibm.com
zurich.ibm.com/cci/cloudFPGA/