A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services

Original Authors:
Andrew Putnam. Adrian M. Caulfield, Eric S. Chung, Derek Chiou, Kypros Constantinides, John Demme, Hadi Esmaeilzadeh, Jeremy Fowers. Gopi Prashanth Gopal, Jan Gray, Michael Haselman, Scott Hauck. Stephen Heil, Amir Hormati, Joo-Young Kimv, Sitaram Lanka, James Larus, Eric Peterson, Simon Pope, Aaron Smith, Jason Thong, Phillip Yi Xiao. Doug Burger

Presented By: S.Venkatesh

1. Overview
2. Challenges and Solution.
3. Introduction to FPGA
4. Requirement and Architecture.
5.Infrastructure and Platform architecture.
   5.1 Debugging support.
 5.2 Failure detection and Recovery.
    5.3 Correct operation.
    5.3 Software Infrastructure

6. Application case study.
    6.1 Micro-pipeline.
    6.2 Queue Manager and Model Reload.
    6.3 Feature extraction.
    6.4 Free Form Expression.
7. Evaluation

Overview

- Demands for datacenter workloads:
  - High computation capabilities.
  - Flexibility
  - Power efficiency
  - Low Cost

CHALLENGE : Hard to improve all factors simultaneously.

Solution

- Composable, reconfigurable fabric to accelerate portions of large-scale software services.
- One fabric consists of:
- (a.) 6x8 2-Dtorus of high-end Stratix V FPGA
- (b.) Embedded into a half-rack of 48 machines.
- (c.) Each server has one FPGA.
- (d.) Wired to other FPGAs with pair of 10 Gb SAS Cables
- (e.) Accessed through PCIe.

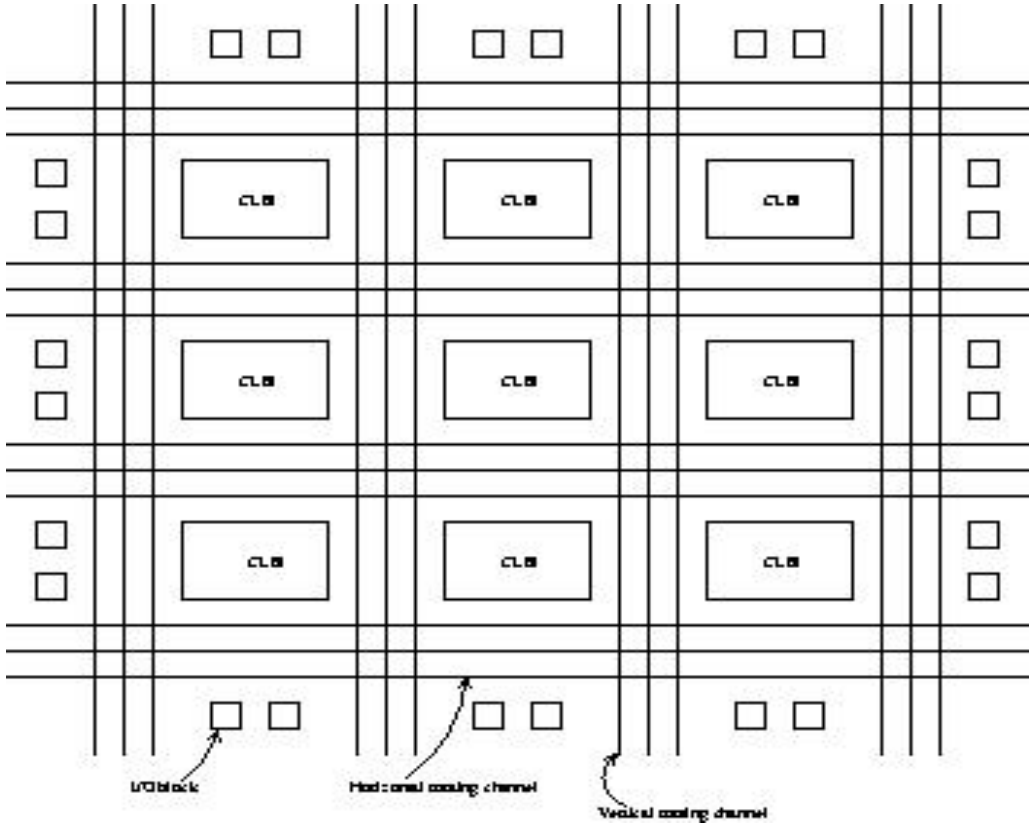
FPGA

- FPGA is one universal chip.
- Initially it does not have any intended logic.
- FPGA can be converted into microcontroller, digital signal processor.
- **Components**
  - Contains large number of configurable logic blocks.
  - CLB can implement any basic function.

FPGA

- **Components:**
  - Multiple CLB can be configured to perform complex digital function.
  - Each CLB contain flip-flops and lookup tables.
  - **Input Output Block** can be programmed to act as input and output ports.
  - Input Output Block can be connected to internal matrix.

FPGA



Requirement And Architecture

- Larger datacenter needs homogeneity to reduce management issues.
- Datacenter evolve rapidly.
- Non-programmable hardware is not sufficient.
- **SOLUTION:**
  - Field Programmable Gate Arrays (FPGA)
  - Use FPGA as computer accelerators.

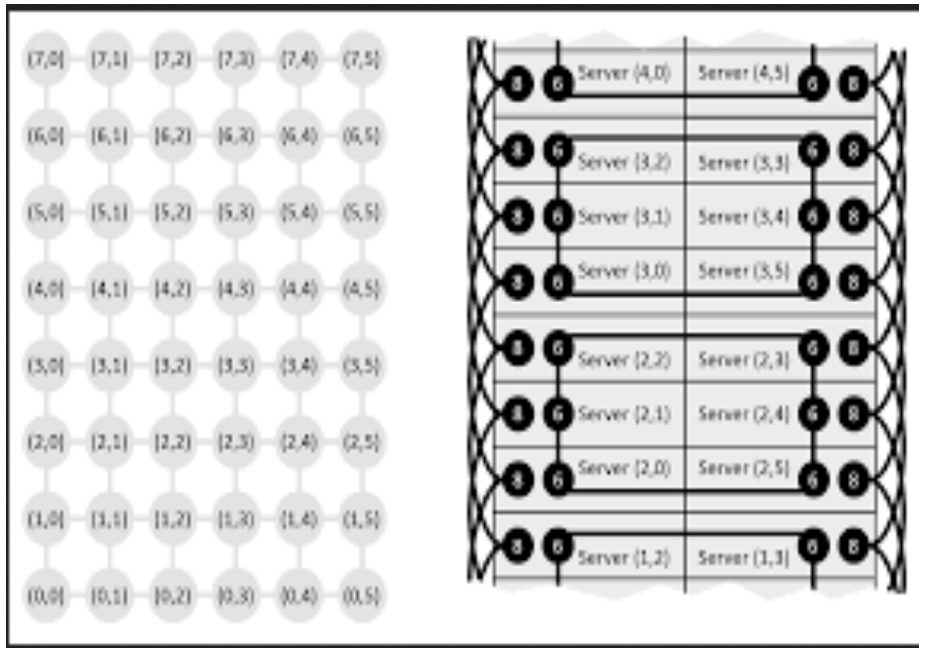
Requirement And Architecture

- **Challenges with FPGA**
  - Standard FPGA reconfiguration time is slow at run-time.
  - Multiple FPGA cost more and consume more power.
  - Single FPGA per server restricts sufficient workload acceleration.

Requirement And Architecture

- **Architecture:**
  - For half-rack consists of 48 server
  - Medium size FPGA and local DRAM for each server.
  - FPGAs are directly wired to each other.

Architecture



Infrastructure and platform Architecture

- Robust software stack for failure detection.
- Three categories of infrastructure:
  - API for interfacing software with the FPGA.
  - Interface between FPGA application logic and board-level functions.
  - Support for resilience and debugging

Debugging support

- Flight data Recorder
- Capture important information about FPGA at run-time.
- Initially stored on-chip memory.
- During health check, it is streamed out.
- Circular buffer: head and tail flits of network packets.

Debugging support

- **Useful to debug**
  - Rare dead lock event.
  - Untested input resulting in hang.
  - Server reboots.
  - Unreliable SL3 links.

Software Interface

- Communication between FPGA and host CPU design goal:
  - Interface must incur low latency.
  - Interface must be multi-threading safe.

- FPGA is provided pointer to user space buffer space.
- Buffer space is divided into 64 slots.
- Each thread has exclusive access to slots.
- To send data to FPGA, fill slot and set flag.

Failure Detection And Recovery

- Monitor server notice unresponsive servers.
- Health monitor contact each machine to get status.
- Execute sequence of soft reboot, hard reboot or manual intervention.
- Healthy service sends status of local FPGA.

Failure Detection And Recovery

- Health monitor update machine list of failed servers.
- Mapping manager moves the application.
- Movement is done based on the location and type of failure.

Correct operation

- FPGA reconfiguration may cause instability in system.
- **Reason:**
  - Reconfiguration can appear as failed PCI
    It triggers non-maskable interrupt bringing instability.
  - Reconfiguring FPGA can send random traffic to neighbor.
    This traffic may appear valid.

Correct operation

- **Solution:**
  - Disable non-maskable for the specific PCI device.
  - Send "TX Halt" message. Meaning ignore all message until link establishes

Shell Architecture

- Apart from application developer needs to write:
  - Host to FPGA communication.
  - Functions required for data marshaling.

  - **Challenges:**
    - Significant burden on developer.
    - These changes require portability.
- **Solution:** Partition all programmable logic into partition.
  - (a) Shell (b) Role

Shell Architecture

- **Solution:**
  - **Shell**
    - Programmable logic common across all applications.
    - Shell consume 23% of FPGA
  - **Features:**
    - Double bit error detection and single bit error correction in DRAM controller.
    - Scrubber runs continuously to remove configuration errors.

Software Infrastructure

- Software works at datacenter level and server level.
- **It needs:**
  - Ensure correct operation.
  - Failure detection.
  - Recovery and debugging.

- **Solution:**
  - Mapping Manager
  - Health Monitor.

Application

- Used in Bing's ranking engine.

- **Overview:**

  - If possible, query is served from front end cache.
  - TLA (Top level aggregator) send query to large number of machines.
  - These machine find documents.
  - It send it to machine running ranking service.

Application

- **Overview:**
  - Ranking service assign score to each document.
  - TLS sort scores and generate result.
  - Features: No of time query word occurred in each document.

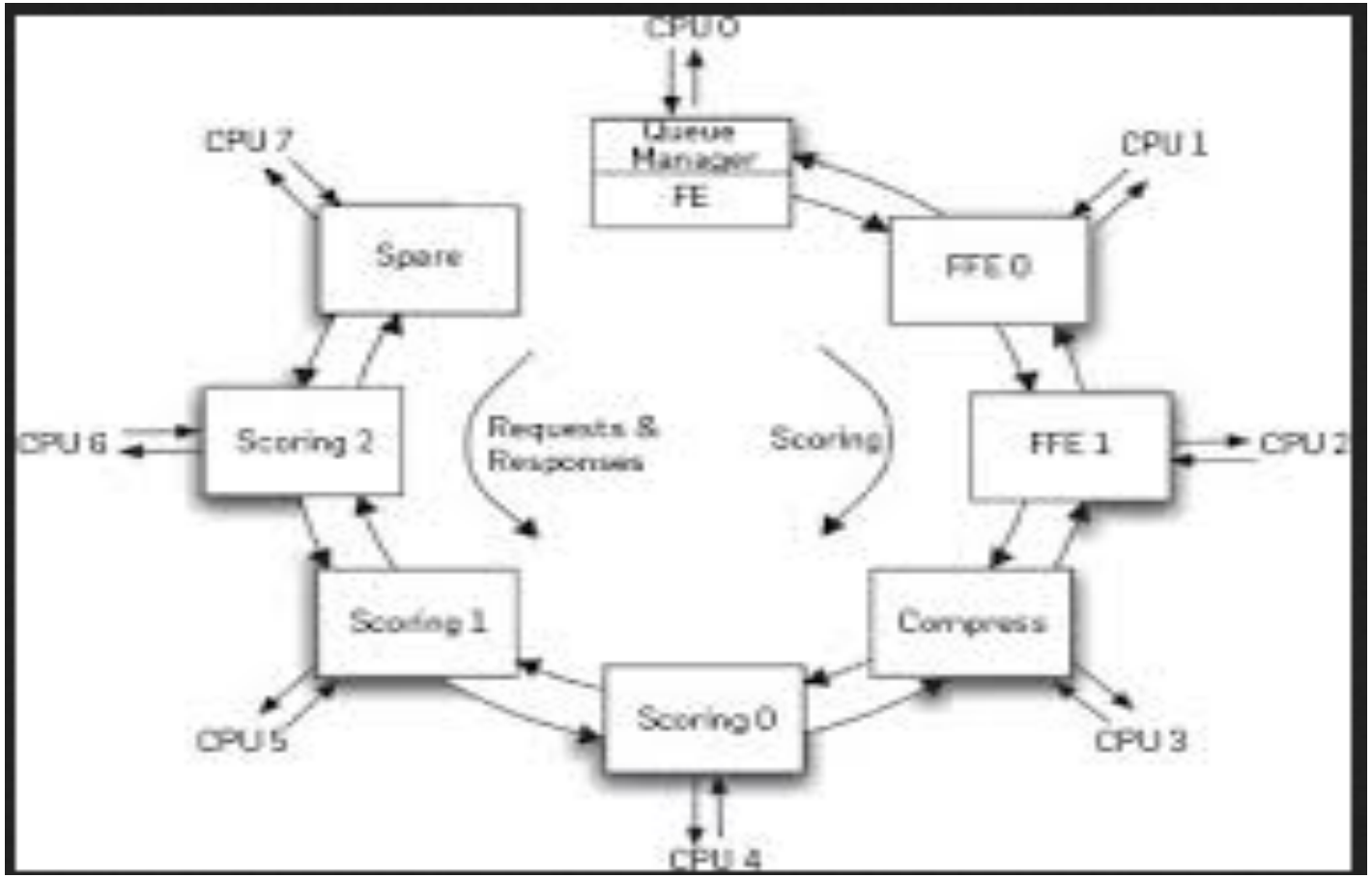
Application

- Similarly many features are sent to machine-learning model.
- Model generate score.
- FPGAs perform: feature computation and machine learning model.

Micropipeline

- Process pipe line is divided into macro-pipeline stages.
- Time limit for micro-pipeline is 8 micro seconds.
- It is 1600 FPGA clock cycles.
- Tasks are distributed in this fashion:
  - 1 FPGA for feature extraction.
  - 2 FPGA for free form expression.
  - 1 FPGA for compression
  - 3 FPGA to hold machine learning models.
  - 1 FPGA is a spare in case of machine failure.

Micropipeline



Queue Manager and Model Reload

- Multiple Models.
- Can be selected based on query type or language etc.
- DRAM contains all queries for a given model in queue.
- Queue Manager selects a queue and reads queries.
- Switch queue when queue is empty.

Queue Manager and Model Reload

- On switching queue send "**Model Reload**" command.
- Model Reload takes less than 250 micro seconds.
- It is relatively slower than document processing time.

Feature Extraction

- On FPGA accelerator, feature extraction runs in parallel.
- Implemented in the form of feature extraction state machine.
- Support for running state machine in parallel on same input data.

Free Form Expression

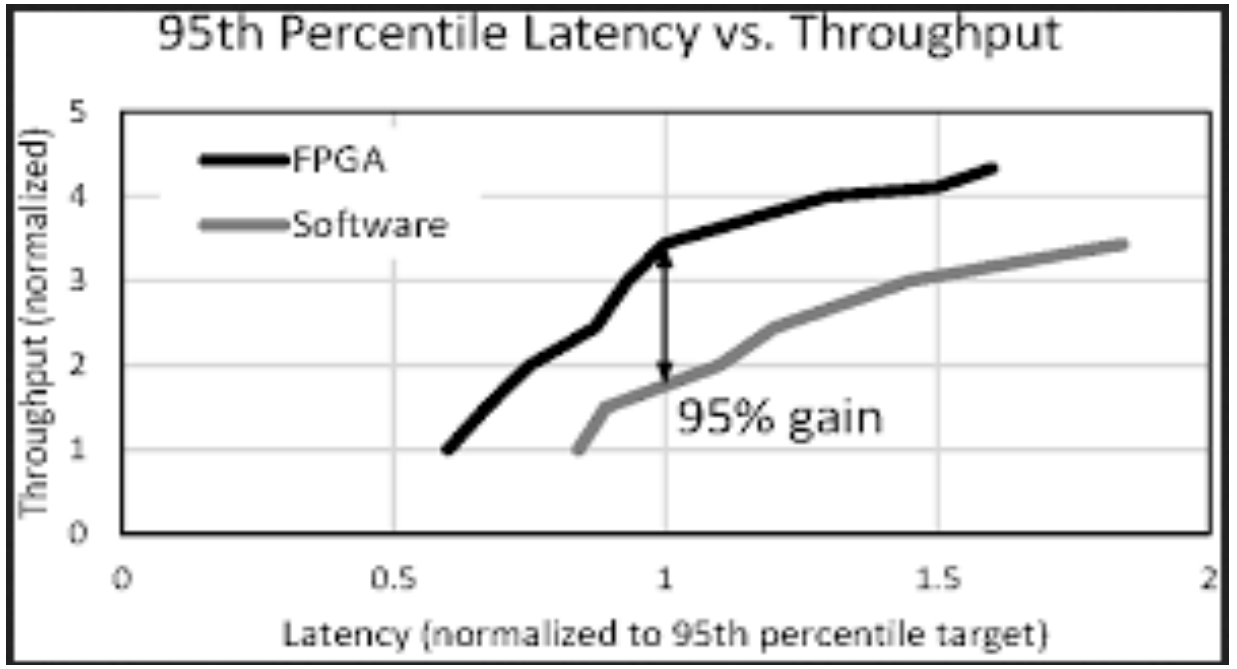
- Mathematical combination of features.
- **Example:** Adding two features.
- **Example:** Can include complex floating point operation
- Custom multicore processor with huge multithreading support.

Free Form Expression

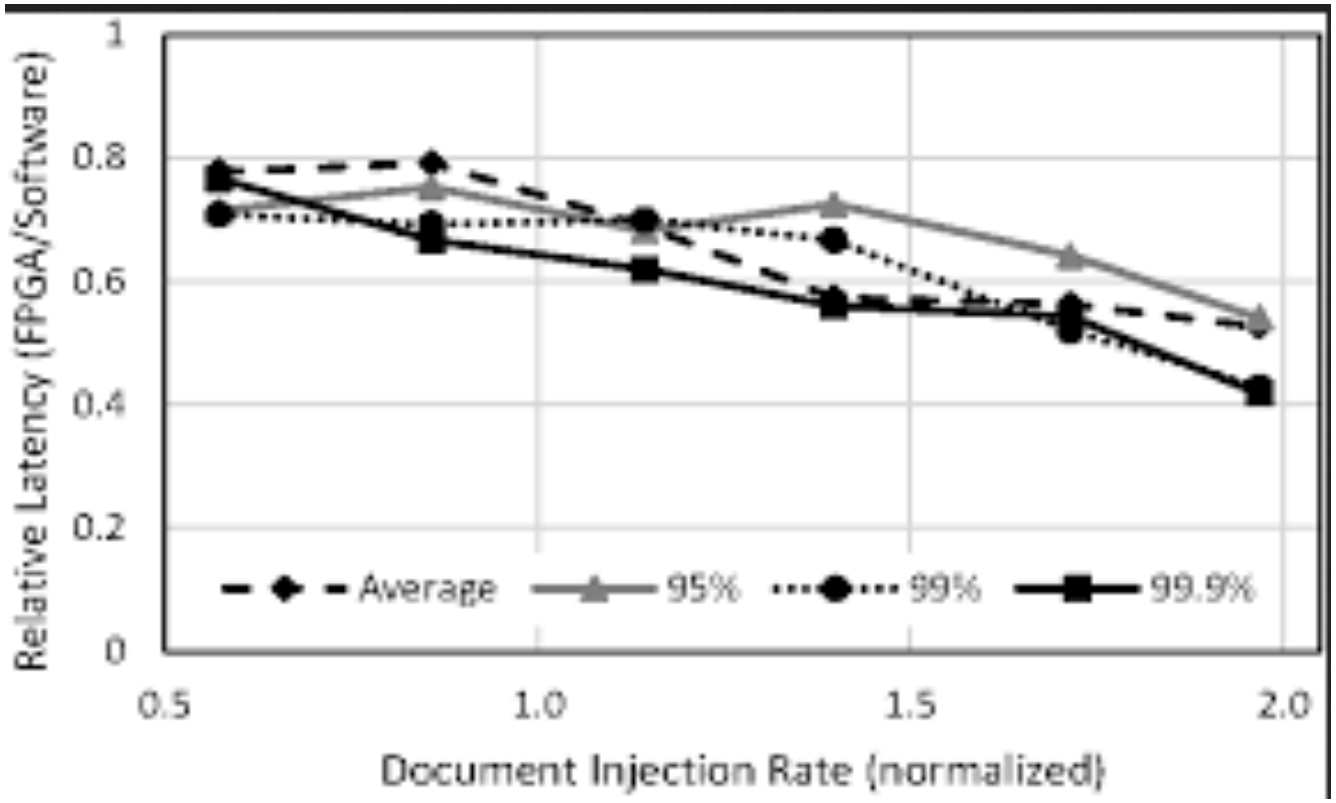
- Implemented on FPGA.
- Long latency expression split across multiple FPGA.
- Single complex FPGA block for ln, fpdiv, exp and float-to-int.

Evaluation

- Node level Experiment:
  - Significant variation in throughput across all stages.
  - Throughput limited by FE.

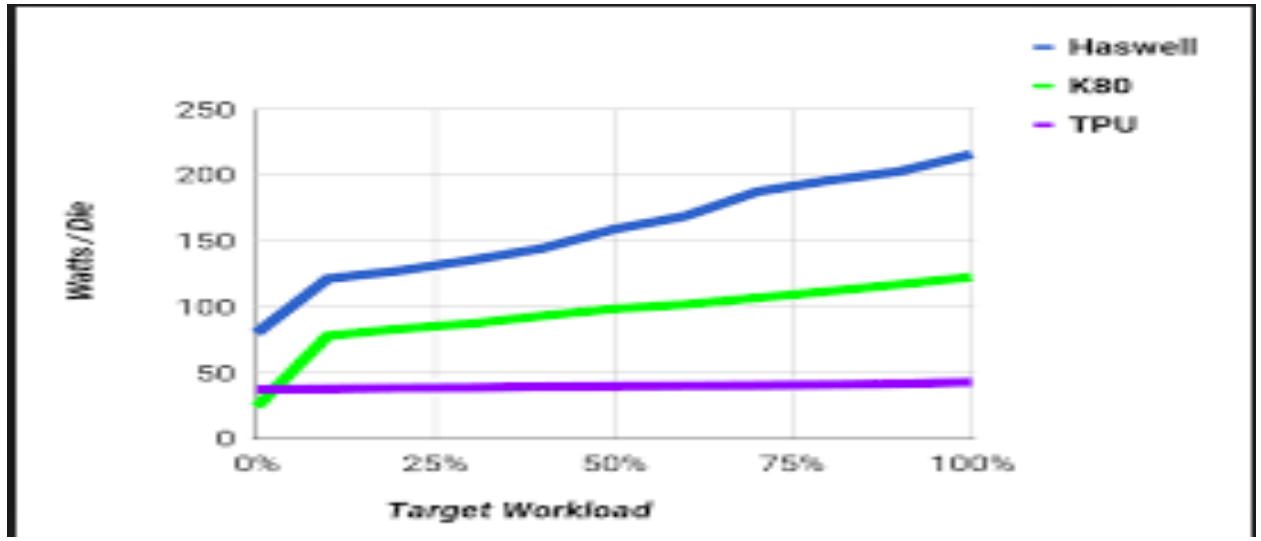


Evaluation


Relative Latency (FPGA/Software)
1
0.8
0.6
0.4
0.2
0
Average
95%
99%
99.9%
0.5
1.0
1.5
2.0
Document Injection Rate (normalized)

Similarity: Using TPUs in datacenter.

- Power consumption compared to GPU is much more than TPUs.



- Same observation is performed for datacenters using FPGAs. Maximum power overhead of FPGAs to our server is of 22.7 W.

Reference

- A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services