The Nuts & Bolts of Computational Storage Platform

Presented By

Gopi Jandhyala
Deboleena Minz Sakalley
Seong Kim
Sonal Santan

October 2\textsuperscript{nd}, 2018
Agenda

- Today’s Challenges
  - Problem Statement
  - Solution Proposal
  - Solution Illustration

- Computational Storage Platform
  - Infrastructure
  - Developer Tools
  - Applications

- Solution Proof Points
  - Postgres DB Acceleration
  - Compression

- Summary
Today’s Challenges

- Exponential data growth driven by unstructured data, Eg. video

- **The new brick wall:** Performance bottlenecks & power implications of moving data back and forth to compute

- Computing hitting one brick wall after another (the end of Moore’s law, Dennard Scaling, Amdahl’s law)

- Inevitable evolution towards Heterogeneous Computing

- **Accelerators** critical to scaling performance cost/power efficiently
The New Brick Wall

Moving Data to Compute ….

1-10 TB / server

- Non data-intensive acceleration
- Data-intensive acceleration performance

Performance ✅ Power ✅

10-100 TB / server

- Non data-intensive acceleration
- Data-intensive acceleration performance

Performance ✅ Power ✗

100-1000 TB / server

- Non data-intensive acceleration
- Data-intensive acceleration performance

Performance ✗ Power ✗
The solution

Moving Compute to Data ....

> “Offload” storage centric workloads
> “Split personality” – offload or storage

> Accelerate storage services “Inline”
  >> Encryption, compression, hashing

> Tighter integration
> Compute near Storage
  >> Inline, offload and more
  >> Search, Bigdata
**Finding The Needle In the Haystack**

> Search for an image across 100TB
> Sequential scan of 100TB drive data into CPU and ML based image classification accelerator
  >> I/O and processing bottlenecks
  >> High power consumption

> For same search embedded ML based image classification accelerator scans drive data locally and responds with match/not match
  >> Eliminate I/O and processing bottlenecks
  >> Localized data scanning optimizes system power
Count the total twitter feeds for a key word in the last 24 hours

Hadoop job across multiple data nodes with each node CPU sequentially scanning the saved data
  - I/O and processing bottlenecks
  - High power consumption

Push the counting work to each drive within data node thus enabling higher number of jobs each data node can undertake

Drive responds with the individual count from each job that the CPU can simply aggregate
  - Eliminate I/O and processing bottlenecks
  - Localized data scanning optimizes system power
Agenda

Today’s Challenges
- Problem Statement
- Solution Proposal
- Solution Illustration

Computational Storage Platform
- Infrastructure
- Developer Tools
- Applications

Solution Proof Points
- Postgres DB Acceleration
- Compression

Summary
Xilinx Computation Storage Platform

Applications
- Database Services
  Spark, Postgres, Maria DB, Hadoop
- Storage Services
  Compression, Dedup, Erasure Codes
- Rich Media Services
  Image Search, Video Transcoding

Developer
- Profiler & Debugger
- SDAccel Compiler
- HW & SW Emulation
- OpenCL, C, C++, Verilog

Infrastructure
- High Speed Interconnect
- Device Memory
- ARM Processors
- System Mgmt
- Partial Reconfig

Adaptable Platform
Platform Architecture

Device Memory

- Exposed to CPU as PCIe BAR
- Used by SDAccel APIs to create buffers directly in the device (instead of Host DDR)
- Can be used as P2P buffer target for direct data transfer from the storage
- Shared by local ARM cores and the HW accelerators
> ARM Cores

- Quad Core ARM Cortex A53 up to 1.3GHz
- Enable SW-Assist libraries for HW accelerators (Kernels)
  - Allows for SW/HW partitioning of kernels
- Help run scheduling activities across multiple kernel instances
- Can be used for authentication of end user accelerator binaries
- Enable board management controls
Platform Architecture

> **System Management**
  - Device sensors to monitor current, temperature, voltages etc.
  - System monitoring alerts to Host

> **Debug Hooks**
  - Enables performance monitoring
  - Provides insight on the interconnect/kernel efficiency
  - Stall monitoring
  - ILA/Chipscope for signal level debug
> **Kernels**

- Can be written in Verilog/VHDL, C, C++ or OpenCL
- Dynamically programmed via PR (Partial reconfiguration) during runtime
- Multiple instances of kernels can be programmed for higher throughput
  - Auto-stitched by SDAccel compiler/linker
- Use device memory for input and output buffers
High-Speed Interconnect
- High Speed fabric connects all critical components
  - PCIe interface to the Host and Storage
  - High speed internal AXI bus to memory and kernels
- BW greater than the storage medium

Storage
- Tight integration with the platform
- Direct data transfer to/from the FPGA buffers to enable faster and efficient kernel access
  - Bypass host DDR!
Infrastructure

Shell, Role & Partial Reconfiguration

Platform “Shell”

Accelerator “Role”

Computational Storage Platform
Tools and Infrastructure

> Compiler
  >> Powerful SDAccel compiler
  >> Many options to choose from – OpenCL, C++, C and Verilog/VHDL

> Partial Reconfiguration
  >> Compiler generates image for the Role in the Platform

> Open Source Xilinx RunTime
  >> Feature full runtime stack with user space and Linux kernel drivers
  >> Multi-threading and Multi-process safe

> Platform Design
  >> Platform details including Shell captured as DSA
Developer Productivity Tools

> HW & SW Emulation
  > Emulation flows for faster debug
  > SW emulation
    - Debug kernel functionality
    - Sequential execution
  > HW emulation
    - Synthesized kernel emulated
    - Parallel/event based execution

> Profiler & Debug
  > Performance, Stall and Execution profiling/monitoring
  > ILA/Chipscope level debug also supported

> Mgmt Tools
  > Query board status
  > Perform board admin operations
PostgreSQL Query6

Solution Proofpoint #1

> PostgreSQL

- PostgreSQL Query6 accelerated by SDAccel stack on computational storage platform
- Minor modifications of PostgreSQL code allowed the data movement from storage device directly to the accelerator, thus eliminating host DDR copies altogether

> Results

- Measurements show 3-5x on idle machines and > 10x on machines dealing with memory intensive apps
Solution Proofpoint #1

**PostgreSQL Query6 – x86 DDR BW Comparison**

DDR B/W comparison – x86 vs. Compute Offload vs Computational Storage; 1GB dataset

Enabling Complete Compute Offload and Minimize Memory Usage
Compression

> Comparison among FPGA acceleration, ASIC (Intel QAT), CPU

> In addition >8x lower latency than CPU, e.g. 1MB file

  >> FPGA Acceleration: 1.99ms
  >> ZLIB-1: 16.66ms

<table>
<thead>
<tr>
<th>Engine</th>
<th>Compression Ratio</th>
<th>Throughput</th>
<th>Compression Ratio</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZLIB-1 on CPU [2]</td>
<td>2.622</td>
<td>81 MB/s</td>
<td>29.564</td>
<td>340 MB/s</td>
</tr>
<tr>
<td>QAT-8955 [3]</td>
<td>2.597</td>
<td>1463 MB/s</td>
<td>7.299</td>
<td>2850 MB/s</td>
</tr>
<tr>
<td>FPGA – High Comp*</td>
<td>2.224</td>
<td>2039 MB/s</td>
<td>35.809</td>
<td>2973 MB/s</td>
</tr>
<tr>
<td>FPGA – High Thruput*</td>
<td>2.116</td>
<td>2187 MB/s</td>
<td>27.934</td>
<td>3137 MB/s</td>
</tr>
</tbody>
</table>

* Partner Data on Xilinx FPGA
Compression – GZIP x86 DDR BW Comparison

Solution Proofpoint #2

Compute Offload Data Transfer

Computational Storage Data Transfer
Agenda

Today's Challenges
- Problem Statement
- Solution Proposal
- Solution Illustration

Computational Storage Platform
- Infrastructure
- Developer Tools
- Applications

Solution Proof Points
- Postgres DB Acceleration
- Compression

Summary
Summary

- **Exponential data growth** driving the computational storage opportunity to offload compute functions **closer to memory and storage**

- FPGA enabled **adaptable storage** will enable **differentiation** and unlock **efficiency** for storage workloads

- Building on the success of Xilinx compute acceleration platform, **Xilinx Computational Storage Platform** provides ease of application portability and tremendous returns for workloads that are have storage affinity