Willow: A User-Programmable SSD

Sudharsan Seshadri, Mark Gahagan, Sundaram Bhaskaran, Trevor Bunker, Arup De, Yanqin Jin, Yang Liu, and Steven Swanson

Non-Volatile Systems Laboratory
Computer Science and Engineering
University of California, San Diego
Hard Drives (2006)

640x \rightarrow 2.2x/yr

866 \rightarrow 2.3x/yr

PCIe-PCM (2010)
PCIe-Flash (2007)
PCIe-Flash (2012)
PCIe-Flash (2015?)
Case Study: Programmable GPUs

Hidden Programmability (Firmware)

Full blown Programmability

ParDal Programmability (Shaders)
Modern SSDs Hide Their Programmability

- Fixed interface
  - SATA or NVMe
  - Storage-centric operations

- Flexible hardware
  - Multi-core processors
  - Complex firmware
Candidates for Near-Storage Compute

• Data-intensive computation
  – Database scans
  – Transcoding
  – Analytics

• Data-dependent accesses
  – e.g. pointer chasing

• Semantic extension
  – e.g. transactions

• Privileged execution
  – e.g. OS offload

Modern SSD processors are inadequate

Feasible on modern SSD processors
Specialized SSDs

- **OS Offload**
  - [Zhang’12]
  - [Caulfield’12]

- **OS Bypass**
  - [Bhaskaran’13]
  - [Saxena’12]

- **Caching**
  - [Do’13]
  - [Kang’13]

- **Query Processing**
  - [Balakrishnan’12]
  - [Huang’12]
  - [Josephson’10]

- **Virtualization**
  - [Kang’13]
  - [Coburn’13]
  - [Prabhakaran’08]

- **Transaction Support**
  - [Wang’14]
  - [De’13]

- **Key Value Store**
  - [Caulfield’12]

- **Novel IO Abstractions**
  - [Coburn’13]
  - [Bhaskaran’13]

- **Virtualization**
  - [Saxena’12]

- **Transaction Support**
  - [Josephson’10]
A Programmable SSD Should...

- Provide a flexible interface
  - New arguments, semantics, and operations
  - Programmable in C (or something better)

- Enforce file system permissions

- Allow execution of untrusted code

- Allow multiple specialized functions to coexist

- Allow for reuse and sharing of functions between applications

- Allow applications to invoke operations without a system call.

- Be able to run trusted code
  - The OS can delegate operations to Willow
  - Untrusted applications can to invoke them.
Willow System Overview

Host

- Application
- Trusted Kernel
- Willow Driver
- PCIe Ctrl

PCIe

Willow

- Bridge
- Interconnect
- SPU
- Custom Code
- Emulated PCM

Interconnect

- 4 GB/s

UC San Diego
The Willow Processor Complex

- 125 MHz MIPS processor
- 32 KB of D- and I-mem
- A bank of NVM
- Network interface
- High-bandwidth Data Streamer
Willow Usage Model and SSD Apps

• The programmer creates an “SSD App”
  • The kernel installs “SSDApps” for applications
    – The Willow-resident code
    – A userspace library
    – A kernel module, if needed
• Communication via RPCs
• Host and SSD code can send and receive RPCs
Trust and Protection

- A file system sets protection policy
- RPCs carry an unforgeable ProcessID
- Execution at SPUs is always on behalf of a ProcessID
- The Willow driver installs access rights
- Willow firmware checks permissions on access
Willow Case Studies

- Basic IO
- Direct IO
- Caching
- Transaction processing
- Key-Value Store
- File Append w/o the file system
Transaction Acceleration with MARS

[SOSP’13]
Editable Atomic Writes in Willow

LogWrite(bufA, addrA, lenA, logAddrA);
LogWrite(bufB, addrB, lenB, logAddrB);
LogWrite(bufC, addrC, lenC, logAddrC);
Commit();
Performance Benefits On TPCB

<table>
<thead>
<tr>
<th>Thread Count</th>
<th>Transctions Per Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5000</td>
</tr>
<tr>
<td>2</td>
<td>7000</td>
</tr>
<tr>
<td>4</td>
<td>10000</td>
</tr>
<tr>
<td>8</td>
<td>14000</td>
</tr>
<tr>
<td>16</td>
<td>20000</td>
</tr>
</tbody>
</table>

ARIES-DirectIO
MARS-Willow

1.5x increase in performance for MARS-Willow compared to ARIES-DirectIO.
Observations and Limitations

- SSD App development is relatively easy
- Composability of SSD Apps is very valuable
- Striping data across SPUs increases complexity for some SSD Apps
- Limited instruction and data storage at SPUs is a persistent challenge
The time is ripe for programmable storage

• Fast NVMs increase storage flexibility and performance demands
• Existing SSDs are already “software defined”
• Numerous applications already exist
• Willow provides a clean, flexible interface
  – Smooth integration with existing software
  – Powerful enough for complex applications
  – Preserves file system protections
• Programmable storage can simplify and accelerate applications
Thanks!