SSD Firmware Implementation Project
- Lab. #1 -

Sang-Phil Lim (lsfeel0204@gmail.com)
SKKU VLDB Lab.

2011-03-24
Contents

• Project Overview
• Lab. Time Schedule
• Project #1 Guide – FTL Simulator Development
Project Overview

• Project #1: FTL Simulator Development
  – Implement a popular FTL scheme on the simple FTL simulator
  – Perform FTL simulations

• Project #2: SSD Firmware Implementation
  – Porting own FTL code on actual SSD platform
  – Evaluate SSD performance with benchmarking tool
Project Overview

• SSD Firmware Implementation Project
• Goal: “Achieve in-depth knowledge of embedded software design and practical experience”
# Lab. Time Schedule

<table>
<thead>
<tr>
<th>Lab.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>FTL Simulator Development Guide</td>
</tr>
<tr>
<td>#2</td>
<td>FTL Simulation Guide</td>
</tr>
<tr>
<td>#3</td>
<td>Project 1 Presentation</td>
</tr>
<tr>
<td>#4</td>
<td>Jasmine OpenSSD platform tutorial #1</td>
</tr>
<tr>
<td>#5</td>
<td>Jasmine OpenSSD platform tutorial #2</td>
</tr>
<tr>
<td>#6</td>
<td>FTL Porting Guide</td>
</tr>
<tr>
<td>#7</td>
<td>Firmware Debugging Guide</td>
</tr>
<tr>
<td>#8</td>
<td>SSD Performance Evaluation Guide</td>
</tr>
<tr>
<td>#9</td>
<td>Project 2 Presentation</td>
</tr>
</tbody>
</table>
Jasmine OpenSSD Platform

• Based on Indilinx Barefoot™ SSD controller
  – 96KB SRAM, 64MB DRAM, SATA 2.0 host interface
  – maximum 256GB capacity
Project #1 – FTL Simulator Development
Project Guide Line

• First, each team investigate a popular FTL schemes such as BAST, FAST, LAST, DAC, etc.
• Next, implement the FTL scheme on the simple FTL simulator
• (Generate I/O workloads for simulation)
• Simulate FTL algorithm and evaluate the performance results
Development Environment

• OS: Windows
• Build tool: Microsoft Visual Studio 2010 Express edition (free)
FTL Simulator Design Principles

• Basic architecture
  – Single chip, Synch IO (Not support I/O parallelism)
• DRAM
  – Sufficient DRAM
  – All metadata cached in DRAM
• NAND flash
  – NAND Flash NOP(Number Of Programming) = 1
  – Only measuring NAND flash’s chip-level overhead
  – Only count NAND flash’s primitive operations
    • Page read/write
    • Block erase
Logical View of NAND Flash

- Single chip basis
- Not contain actual user data

![Diagram of NAND Flash layout]

- Flash chip
- Block 0
  - Page 0
  - Page 1
  - ...
  - Page m-1
- Block 1
  - Page 0
  - Page 1
  - ...
  - Page m-1

... (Repetitive pattern)

- Block n-1
  - Page 0
  - Page 1
  - ...
  - Page m-1

- Page
  - start_lsn

- Check data integrity
FTL Simulator Overview

- `<R/W, LSN, sector_count>`
- `ftl_read/ftl_write`
- `nand_page_read/nand_page_write`
- `nand_block_erase`

I/O Traces

FTL

NAND Flash (dummy)
Basic Read Operation

Host

FTL

NAND Flash

<\(R, 0, 100\)>

<table>
<thead>
<tr>
<th>LPN</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

per-page

Sending to host
Basic Write Operation

```
Host

<W, 0, 100>

---

FTL

per-page

<table>
<thead>
<tr>
<th>LPN</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

programming host data

---

NAND Flash

```
Notice for Lab #2

• Each team should investigate an FTL scheme
  – Fully understand the target FTL include mapping algorithm and NAND usage
    ! Reading list related to FTLs is downloadable @ http://csl.skku.edu/ICE3028S11/Resources

• Next Lab time, you should summarize key idea of target FTL and present within 5 slides
Paper Reading List

Contact with TA

- By e-mail: lsfeel0204@gmail.com
- By phone: 031-290-7988
- By twitter: @ice3028skku
Any Questions?