Today

=> Persistent Storage
    => HDDs
    => Flash-based SSDs

=> Advice
    => Students (undergrads, grads)

=> Wrap-up
    => What did we learn?
Flash-based Solid-state Storage Devices (SSDs)

"new" base technology in 1980's (Toshiba)

basics: flash cell (storage for one/few bits)

=> traps charge in cell

SLC [single-level cell]

high \(\rightarrow 0\), low \(\rightarrow 1\)

multiple bits/cell

MLC
Flash chip

\[ \Rightarrow \text{interface / internals} \]
\[ \Rightarrow \text{store many bytes of data} \]

Device:

- many chips

Interface:

- read (page #)
  \[ \Rightarrow \text{data} \]
- write:
  - weird (SSD)

  \[ \Rightarrow \text{erase (block #)} \]
  \[ \Rightarrow \text{enables us to program pages in block (1 time each)} \]

  \[ \Rightarrow \text{program (page #)} \]
  \[ \text{sets bits of that page to desired content} \]

Internals:

- larger blocks
  - sector: 512 bytes
  - larger blocks: 4KB

- series of blocks:

  0 \hline 1
  \hline
  \hline

\[ \sim 1 \text{ MB} \]

Each block divided into pages (SSD)

\[ \underline{4KB} \]

\[ \underline{\ldots} \]

\[ \underline{\ldots} \]

\[ \underline{\ldots} \]
Performance

- read
- erase (block)
- program

Latency:
- \( \sim 10 \mu s \) (much faster than HDD)
- \( \sim 1 \frac{1}{2} \text{ms} \)
- \( \sim 100 \mu \text{s} \)

Reliability

- Primary problem: wear out
  - charge level
  - time
  - \( \sim 10 \text{k} \rightarrow 100 \text{k} \) erase/program cycles

- Secondary problem: read/program disturbance

- "wear leveling"
  - goal: spread out load writes

Diagram:

- Block
- Pages
- Charge level
Goal: Build HDD replacement w/ same API

HDD API: \( \frac{\text{read}}{\text{write}} \) (blocks)

Map reads/writes

\[ \Rightarrow \text{Flash reads, erase, program} \]

---

Diagram:

- CPU
- Ctrl
- Mem
- Flash-based SSD
- Bus
Naive: Direct map

4KB read, write

Interface:

Flash:

interface read (100) \implies flash read (page=100)

write (block=) \implies \text{(page by page)}

read block \implies \text{Flash SSD memory}

erase block

program each page (mostly with old, new too)
write ( ) =>
  read (block)
  erase (block)
  program (block)

Problem #1: [way too slow]

Problem #2: wear out
  (in control of programmer)
Question:

How to design Flash-based SSD so as to

1) get good write perf?

2) not allow user to quickly wear out device?
interface:

\[ \text{write (address = 100, data = 'a')} \]

\[ \text{block} \]

\[ \text{block} \]

\[ \text{... a b ...} \]

\[ \text{page} \]

\[ \text{write (address = 2000, data = 'b')} \]

\[ \Rightarrow "\text{log structuring"} \]

\[ "\text{logging"} \quad (\text{append only}) \]

Problems: 1) how to find info?
2) run out of space?
1) how to find info?

```
addr: 3000
```

```
[ meta block, block, ...
```

```
write (address = 100, "a")
```

```
write (address = 3000, "b")
```

```
read (addr = 3000) =>
```

```
need more info:
[ mapping table ]
```
Mapping Table: Flash Translation Layer (FTL)

For each page: logical -> physical

\[
\begin{pmatrix}
3000 \rightarrow 2 \\
100 \rightarrow 3 \\
4500 \rightarrow 4
\end{pmatrix}
\]

Problem: what if device loses power?

\rightarrow (many solutions)
What if device fills up?

$\Rightarrow$ need: free "old" copies of data

$\Rightarrow$ reclaiming space: (garbage collection)
Garbage collection:

- How to tell if page is garbage?

<table>
<thead>
<tr>
<th>mem (map)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000 - 2</td>
</tr>
<tr>
<td>1000 - 7100</td>
</tr>
</tbody>
</table>

Meta data:

- addr: 1000
- (live)

Addr: 3000

Garbage

How reclaim free space?

- Move live data to new block
- Erase entire old block + reuse it
CS 354

"The End"

\[ \Rightarrow \text{Admin:} \]

hw5 graded

competition: soon

hw6: [due tomorrow]

"optional" final:

[\Rightarrow \text{late Fri, early Sat:}]

produce grade

final: Monday @ night

\[ \Rightarrow \text{I prepared for this exam.} \]
Exam:

=> prepare

=> 1 page cheat sheet
   (both sides)