Zuverlässiger paralleler Speicher mit Flash Memories

Kathrin Peter

Zuse Institute Berlin

January 30, 2009
Motivation

- Nonvolatile storage
- Flash memory - Invented by Dr. Fujio Masuoka 1984
- Type of EEPROM
- Usage in a RAID-like configuration instead of hard disk drives

Flash memory based storage: **Alternative to hard disks?**
Outline

1. Flash based memory

2. Parallel and distributed storage based on Flash memories
Application of Flash memories
Pros and Cons

- Less power consumption
- Higher access rates (in some cases)
- Uniform access time for random access - no seeks
- Robustness (extreme temperatures, vibration, shock)

Price
- Limited erase cycles
- Flash management

<table>
<thead>
<tr>
<th>Model</th>
<th>2.5&quot; SATA 3.0 Gbps SSD</th>
<th>2.5&quot; SATA 3.0 Gbps HDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanism type</td>
<td>Solid NAND flash based</td>
<td>Magnetic rotating platters</td>
</tr>
<tr>
<td>Density</td>
<td>64 GByte</td>
<td>80 GByte</td>
</tr>
<tr>
<td>Weight</td>
<td>73 g</td>
<td>365g</td>
</tr>
<tr>
<td>Active Power consumption</td>
<td>1 W</td>
<td>3.86 W</td>
</tr>
<tr>
<td>Operating temperature</td>
<td>0°C-70°C</td>
<td>5°C-55°C</td>
</tr>
<tr>
<td>Acoustic Noise</td>
<td>None</td>
<td>0.3 dB</td>
</tr>
<tr>
<td>Endurance</td>
<td>MTBF &gt; 2M hours</td>
<td>MTBF &lt; 0.7M hours</td>
</tr>
<tr>
<td>Av. access time</td>
<td>0.1 msec</td>
<td>17 msec</td>
</tr>
<tr>
<td>Read performance</td>
<td>100 MB/s</td>
<td>34 MB/s</td>
</tr>
<tr>
<td>Write performance</td>
<td>80 MB/s</td>
<td>34 MB/s</td>
</tr>
</tbody>
</table>
Limited erase cycles and flash management

- Memory cell: floating gate transistor
- Retention and Endurance

Typical page size: 16896 B (=2 KB+64 B spare)
Typical block size: 64 pages = 128 KB
Problem: **How to map logical blocks to flash addresses?**

Disadvantage of linear mapping (one-to-one mapping):
- Frequently-used erase units wear out
- Identity mapping requires lots of copying to and from RAM (fixed block size). Example:

Solution: **Sophisticated block-to-flash mapping** and moving around blocks: **wear leveling, garbage collection**
Mapping of virtual block 5 to physical block 0, page 3

Flash memory

Virtual-to-logical page maps

Logical-to-physical erase unit map

example:

virtual block 5 = logical block 7 = 111

= log. erase unit 1

= phy. erase unit 0

page 3

page 3

page 0

page 1

page 2

block 0

block 1

block 2

virtual−to−logical

page maps

Algorithms and Data Structures for Flash Memories, Gal, Toledo, 2004
Wear leveling - example

flash memory

hot pool
cold pool

order in pools

older

stop aging

On Efficient Wear Leveling for Large-Scale Flash-Memory Storage Systems, Li-Pin Chang, 2007
Garbage collection - example

- Free space drops below a threshold
- Select blocks for reclamation
  - Copy all live pages to free pages somewhere else
  - Change mapping entry, update to new position
  - Erase block and allocate pages as free

Diagram:
- Erase block x
- Erase block y
- Erase block x
- Green: life page
- Black: invalid page
- White: free page
Facts (local) wear leveling

- Performance: Erase operation is slow
- No in-place update
- Controller: Efficient mapping and erase distribution
Flash memory in a RAID-like system

- **Bandwidth aggregation**
- **Reliability** (redundancy for fault tolerance: longer lifetime)
- **Problem**: uneven usage of flash memories, more writes to redundant blocks when data becomes updated
There exist data distribution algorithms to place data evenly on a single memory.

Goal: long lifetime of a single memory.

We work on even distribution of writes across all memories in a memory array.

Goal: even usage of cells on all flash memories, reliable data storage, high throughput.

Method: global wear leveling.
Data Distribution Algorithms

- Staggered Striping
- Hierarchical dual-pool Algorithm

- Different starting points of algorithms
  - Explicit placing of data
  - Data movement after storing and local wear leveling

Kathrin Peter
Reliable Parallel Storage on Flash Memories
Summary

- Use aggregated **bandwidth and fault-tolerance (longer lifetime)**
- Next steps: Implementation and evaluation of the global wear leveling algorithm
- Further appl.: application server, dual devices (flash, HDD) databases, mobile devices, flash file systems
Danke!