Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

Research paper presentation:
"De-indirection for Flash-based SSDs with Nameless Writes"
Y. Zhang, L. P. Arulraj, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau

Federico Wasserman & Rodolphe Lepigre

MOSIG - Parallel, Distributed and Embedded Systems

December 17, 2012

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

Outline

1. Introduction

2. SSD principles

3. Indirection in SSDs

4. Nameless Writes

5. Evaluation

6. Conclusion

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

1 **Introduction**

2 SSD principles

3 Indirection in SSDs

4 Nameless Writes

5 Evaluation

6 Conclusion

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

What are Nameless Writes?

- New device interface for SSDs
- Remove the need for indirection
- Idea: the device chooses WHERE to write

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

How are Nameless Writes different?

Usual Writes:

- The FS requests the writing of data at some location
- The device performs the write

Nameless Writes:

- The FS requests the writing of data
- The device performs the write
- Address returned to the FS

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

1 Introduction

2 SSD principles

3 Indirection in SSDs

4 Nameless Writes

5 Evaluation

6 Conclusion

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

SSDs need indirection

- Indirection is used to implement wear-leveling
- Absolutely necessary to ensure reasonable lifetime
- Problem: need to store indirection table
- 3 main techniques:
    - Full-page mapping
    - Block mapping
    - Hybrid mapping

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

Full-page mapping

- Each page can be mapped
- Consider 32-bit pointers per 2KB pages
- With 1TB SSD, 2GB indirection table
- Problem: Great space overhead, DRAM is expensive

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

Block mapping

- Mapping at block-level (128 pages)
- 32MB indirection table in the same settings
- Smaller memory overhead
- Problem: high garbage collection cost (Gupta et al.)

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

Hybrid mapping

- Map most data at block level
- Small page-mapped area
- Keeps space overhead low
- Avoids garbage collection overhead
- Problem: garbage collection can still hurt performances
- Problem: very complex FTL (Flash Translation Layer)

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

1 Introduction

2 SSD principles

3 Indirection in SSDs

4 Nameless Writes

5 Evaluation

6 Conclusion

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion

Introduction
SSD principles
Indirection in SSDs
Nameless Writes
Evaluation
Conclusion