Addressing Shingled Magnetic Recording drives with Linear Tape File System

Albert Chen and Jim Malina

SDC – Storage Developer's Conference 2013

SNIA

absolutely™ WD

Agenda

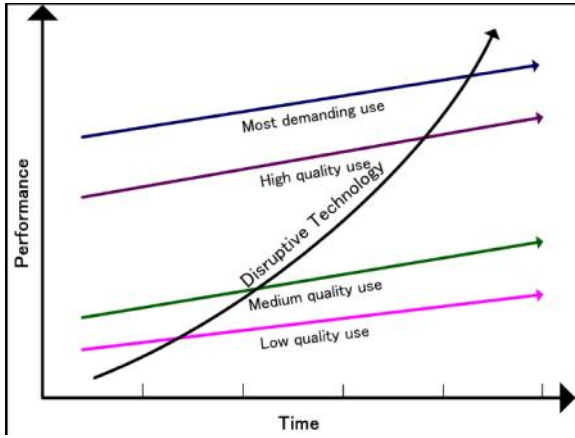
1. Host Managed SMR Philosphy
2. Problem Statement - Opportunity
3. Approaches
4. Hardware: Archive / Cloud drive
5. Software: Filesystem
6. Development
7. Validation: Testing & Performance
8. Lesson’s Learned / Feedback

absolutely™ WD®

Host Managed SMR

Philosophy

- **Shingle Magnetic Recording** is a **disruptive technology** – Innovation that results in **variable performance** but adds **new value** proposition. Fringe customers may benefit and over time performance may increase.



- Identify markets ripe to benefit from SMR
  - Able to modify their storage stack
  - Write 100% sequentially (tape like)
  - Benefit from lower $/GB
- "You've got to start with the customer experience and work backwards to the technology. You can't start with the technology and try to figure out where you're going to sell it" Steve Jobs

absolutely™ WD®

Source available at http://en.wikipedia.org/wiki/File:Disruptivetechnology.gif"

Problem Statement / Opportunity

- Background:
  - Archive / Cold storage is a new market.
    - Write once and forget. (read seldom, if ever).
    - Primary care about: TCO
  - Shingle Magnetic Recording is a new technology
    - Sequential write (only)
    - Path to lowest $/GB if host can abide by one simple rule
  - Industry has discussed two approaches for SMR
    - Drive does data management – similar to FTL
    - Host does data management – via a SMR file system
- Problem Statement
  - Solve the new market opportunity, Archive / cold storage with new recording technology that provides lowest $/GB

Host Managed SMR

Approaches

- Drive managed SMR
  - Similar to Flash Translation layers, data management is complicated.
    - Metadata, garbage collection, over provisioning, write amplification, electronics resources, variable performance, validation.
- Host Managed SMR
  - Similar to Flash File Systems, data management is complicated but there may be opportunity to leverage mature and popular file systems that write sequentially.
    - Drive will not be burdened by the resource cost of a drive managed system

Hardware: Archive cloud drive

- Host Managed SMR, Single Zone, Host writes 100% sequentially – **Simplest SMR implementation**
- Intended for “Big Data”, Cold/Archive, Tape Replacement
- Use Case:
  - Write once, read seldom
  - Garbage Collect from Device to Device
- We built it
  - Proved AD gain, Build & test process
  - Proved Device Architecture, Reliability & Error handling

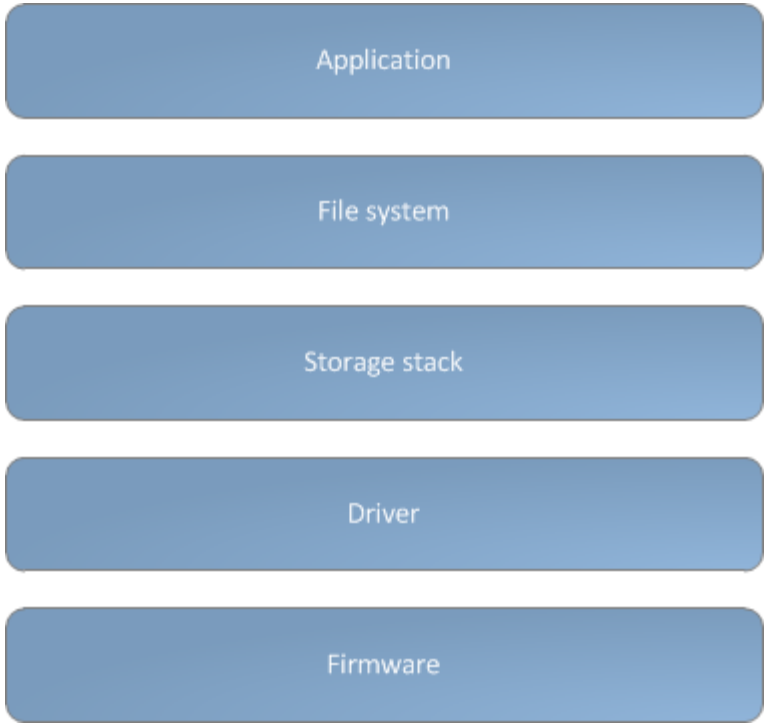
Software

absolutely™ WD®

Host I/O

- Sequential write only.
- No overwrite allowed.
- Garbage collection.
- Handle out of bounds access failure.

Abstraction layer



File system

- Industry usage/support.
- Open source/specification.
- User vs kernel space
  - Minimize risk
  - Ease of development/debug

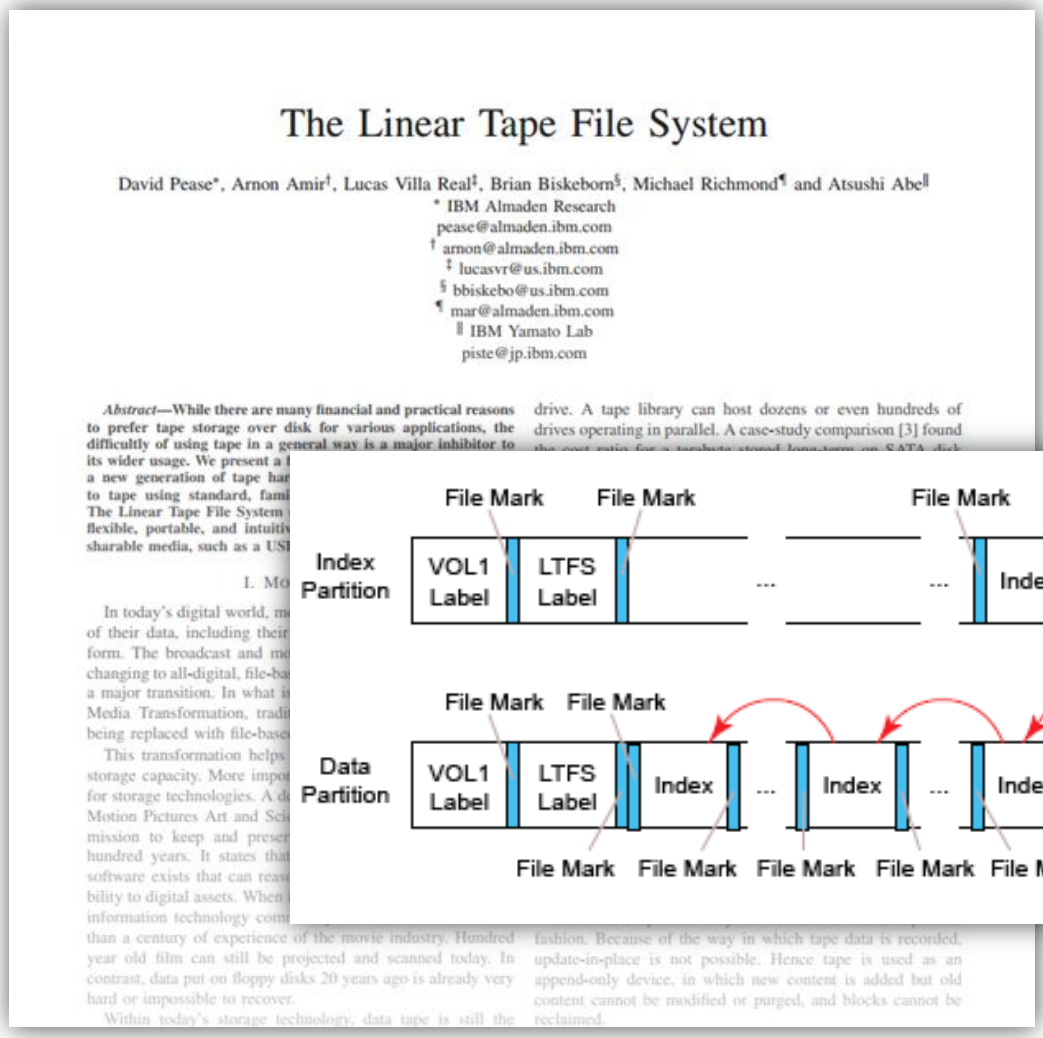
LogFS

“LogFS is a Linux log-structured and scalable flash file system, intended for use on large devices of flash memory. It is written by Jörn Engel and in part sponsored by the CE Linux Forum.

LogFS is included in the mainline Linux kernel and was introduced in version 2.6.34, released on May 16, 2010.”

- Wikipedia (http://en.wikipedia.org/wiki/LogFS)

LTFS



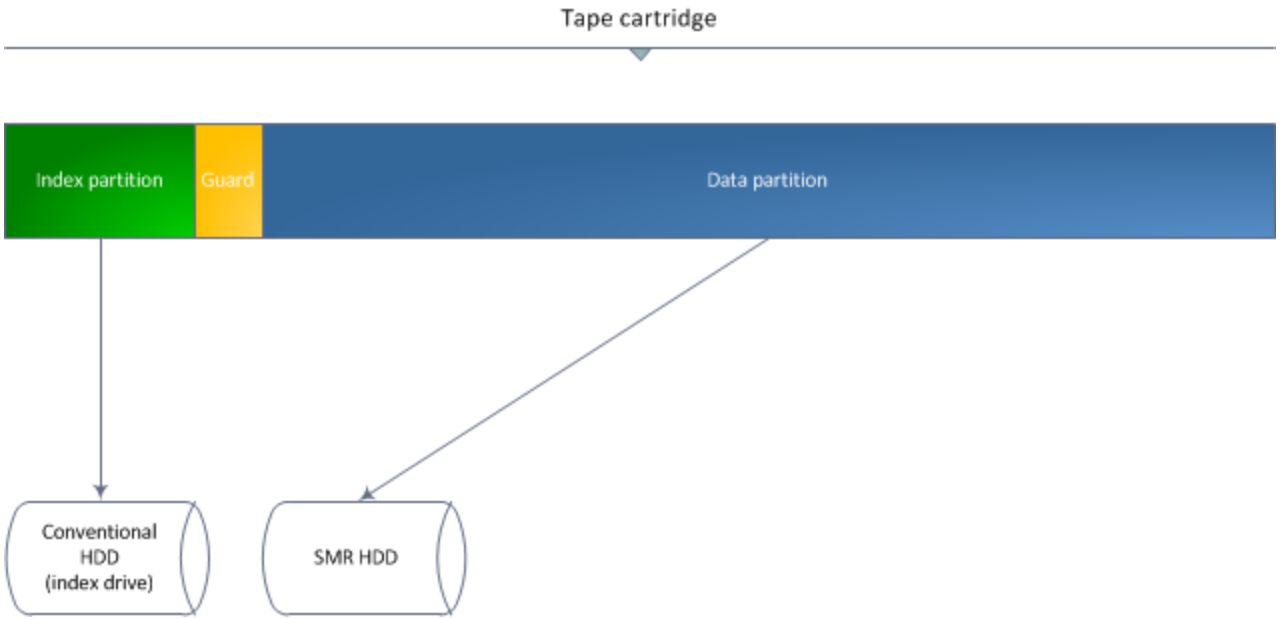
The Linear Tape File System

David Pease*, Arnon Amir†, Lucas Villa Real‡, Brian Biskeborn§, Michael Richmond¶ and Atsushi Abe‖
* IBM Almaden Research
pease@almaden.ibm.com
† arnon@almaden.ibm.com
‡ lucasvr@us.ibm.com
§ bbiskebo@us.ibm.com
¶ mar@almaden.ibm.com
‖ IBM Yamato Lab
piste@jp.ibm.com

***Abstract*—While there are many financial and practical reasons to prefer tape storage over disk for various applications, the difficultly of using tape in a general way is a major inhibitor to its wider usage. We present a t a new generation of tape har to tape using standard, fami The Linear Tape File System flexible, portable, and intuitiv sharable media, such as a USI**

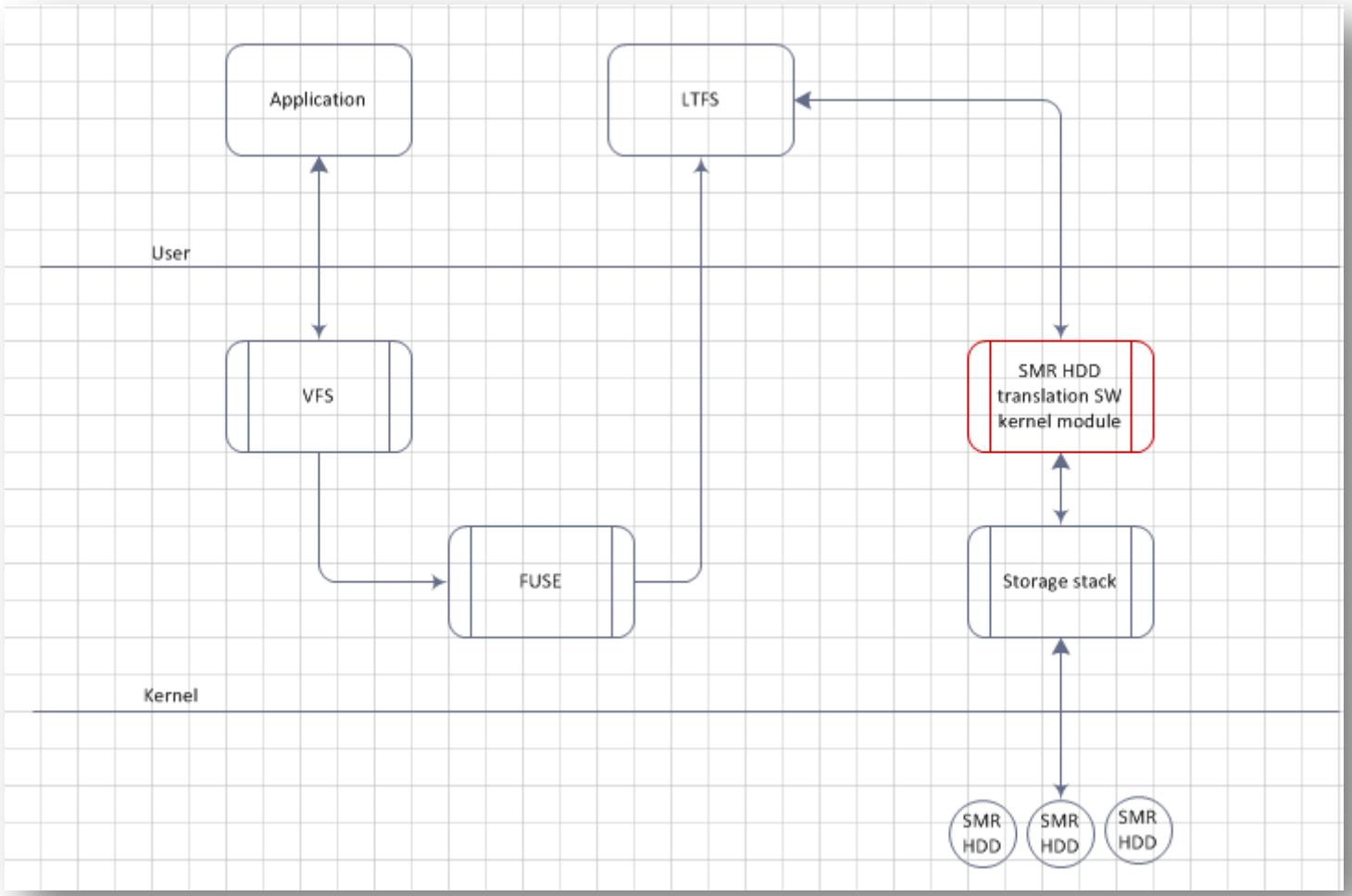
absolutely™ WD

Partition



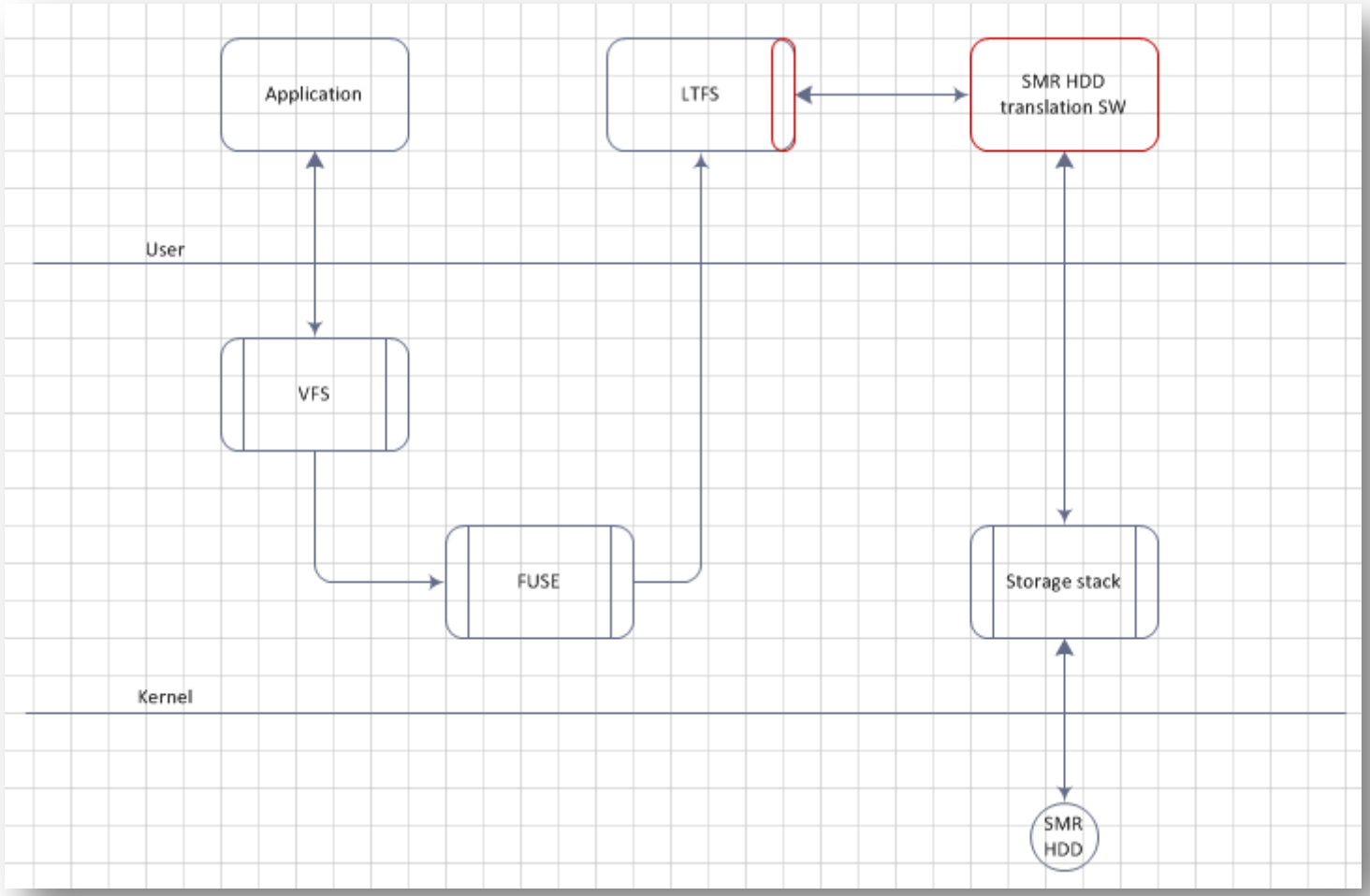
absolutely™ WD

Kernel module



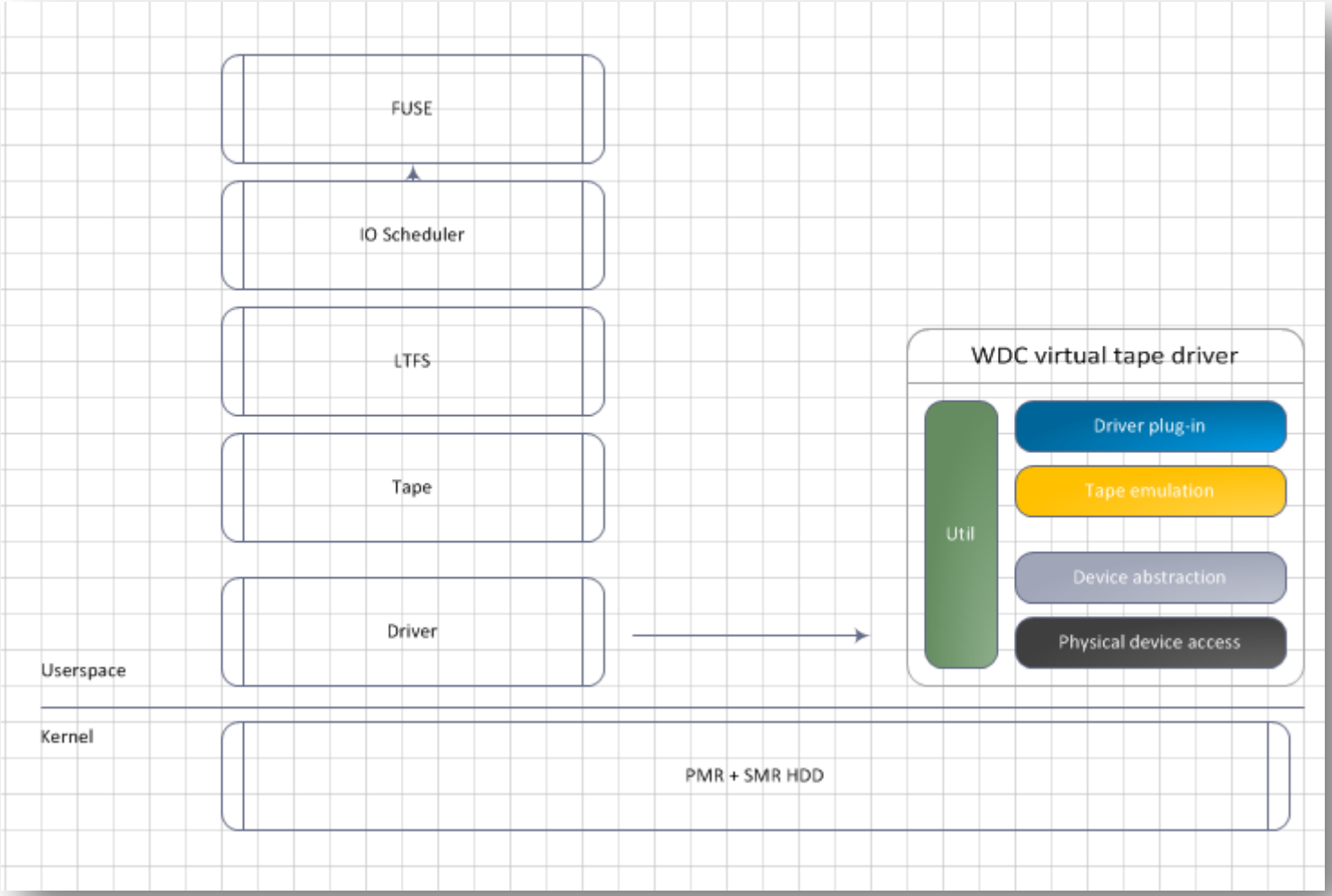
absolutely™ WD

User space



absolutely™ WD

Code layout



Development

LTFS 1.2.5 with Ubuntu 10.04.4 LTS Lucid Lynx

absolutely™ WD

Coding

- No changes to LTFS source
- Backend tape driver
- ~5000 LOC in 25 files
- 2 month POC
- Work in progress

Validation:
Testing & Development

absolutely™ WD®

Validation at each level

- Directory
- File
- Device

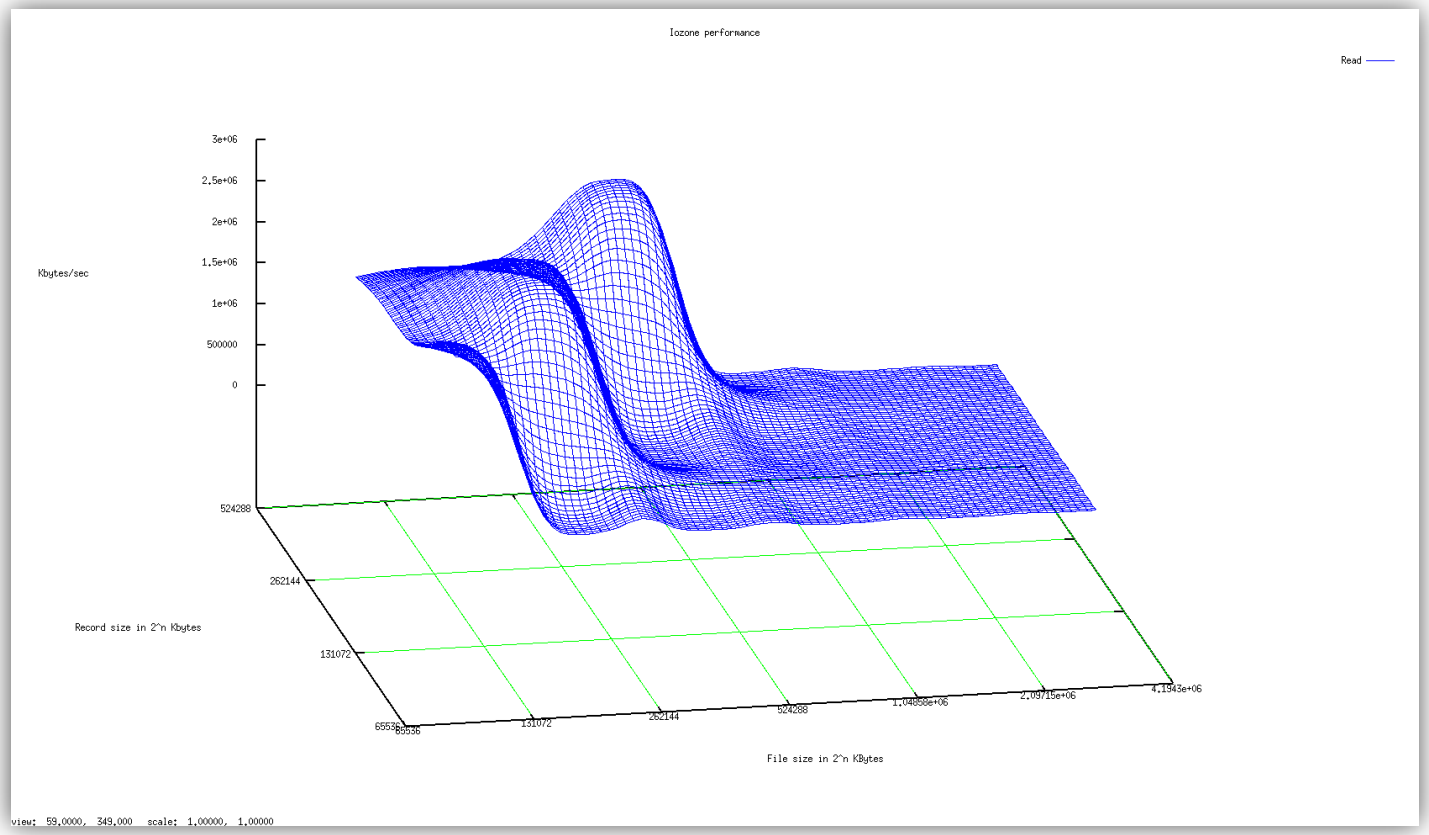
POC setup

absolutely™ WD

Parameters

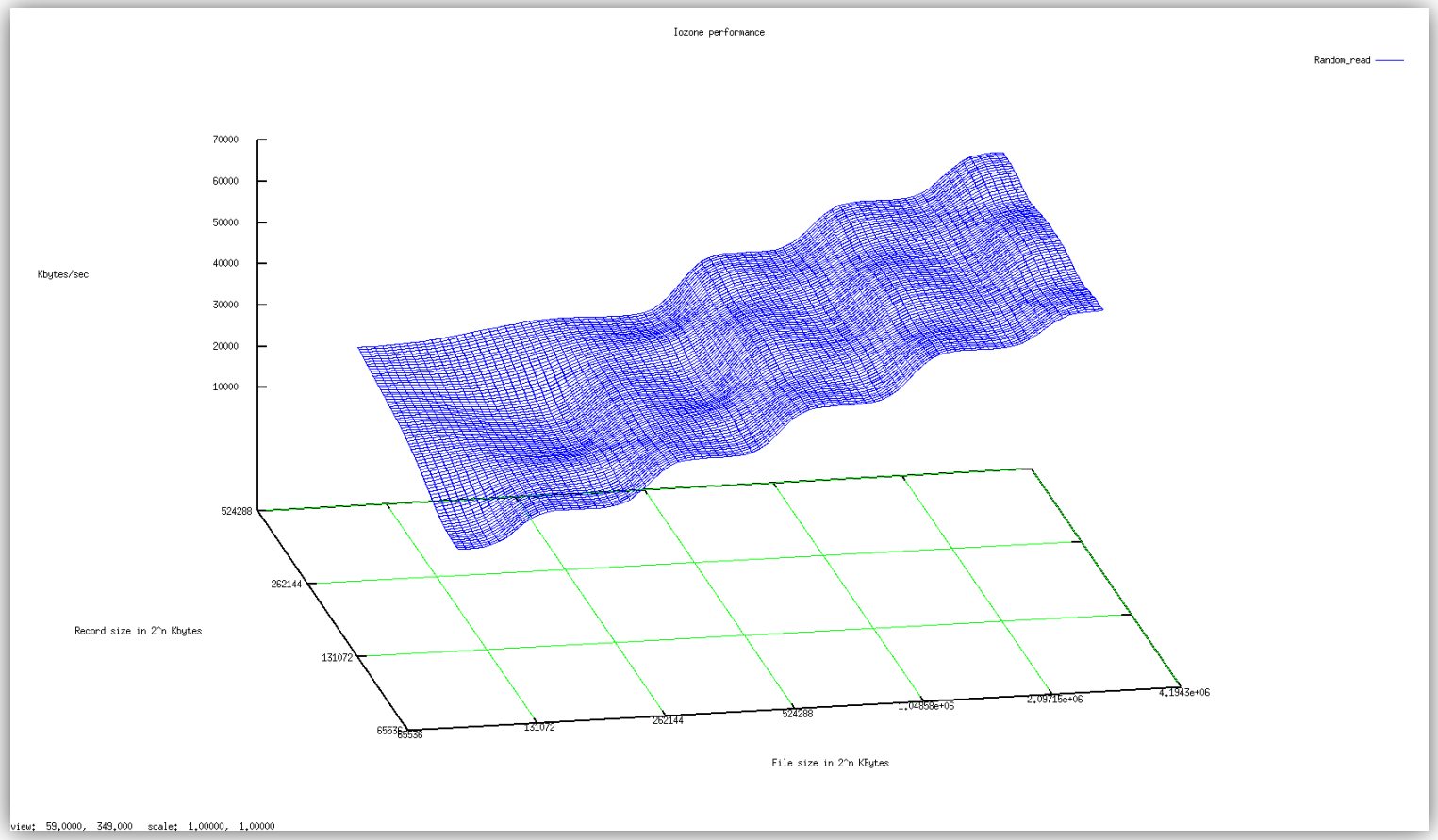
- File size: 500MB to 4GB
- Record size: 100MB to 500MB
- O_SYNC
- Purge processor cache before each file operation
- Unmount LTFS between each test
- Intel Corporation 5 Series/3400 Series Chipset 6 port SATA AHCI Controller (rev 06)

Performance – Sequential Read



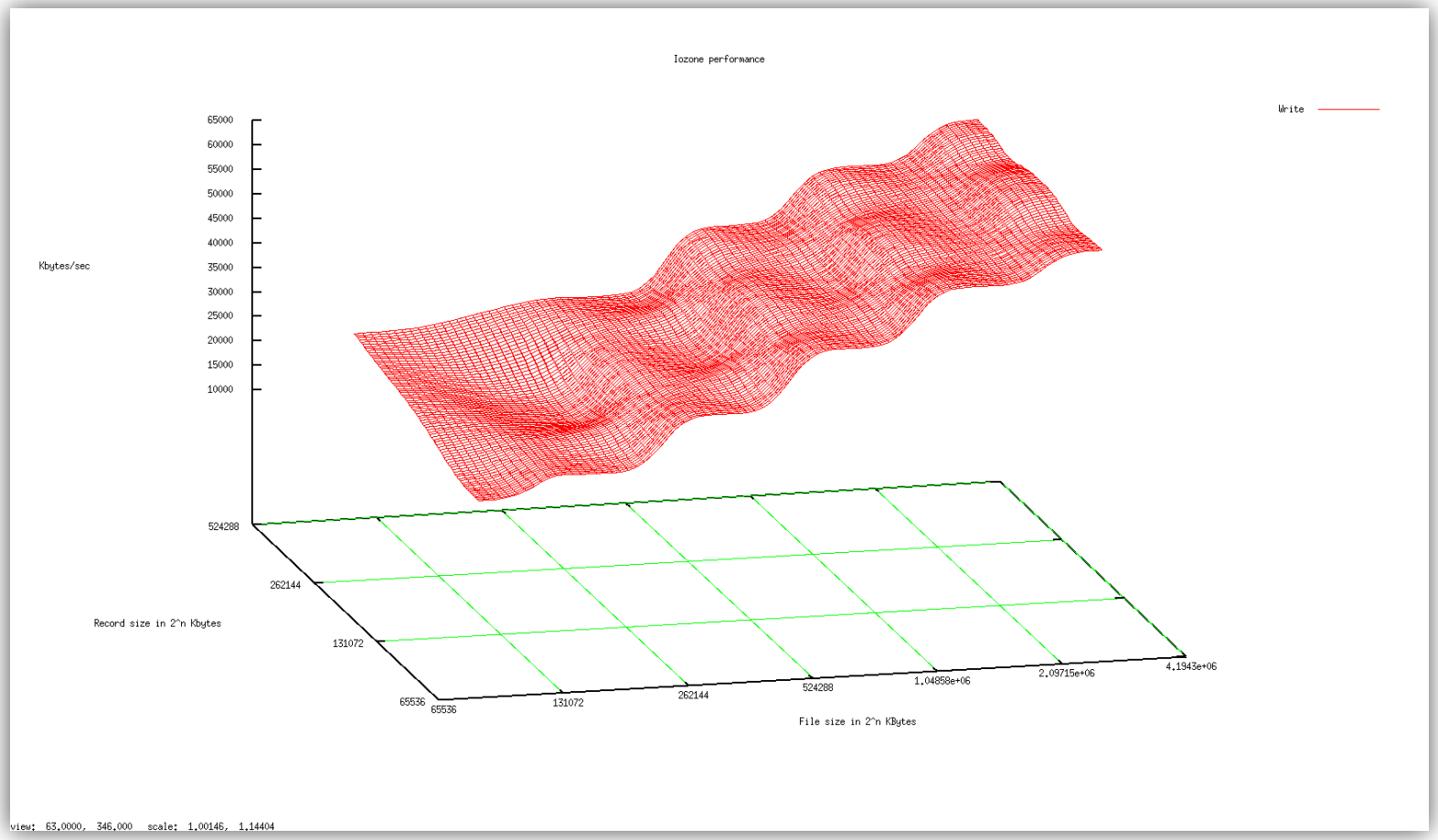
absolutely™ WD

Performance – Sequential Write



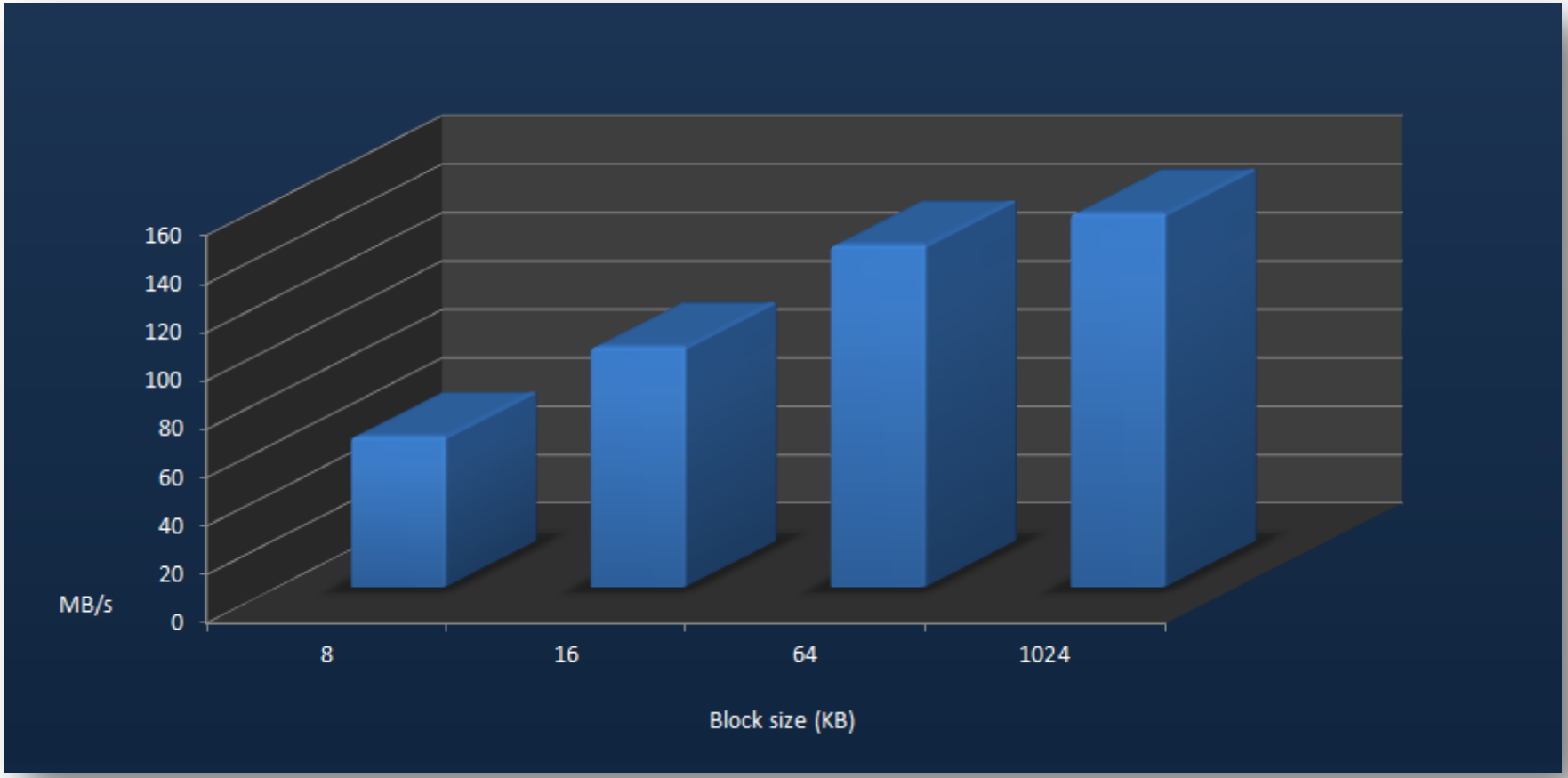
absolutely™ WD

Performance – Sequential Write



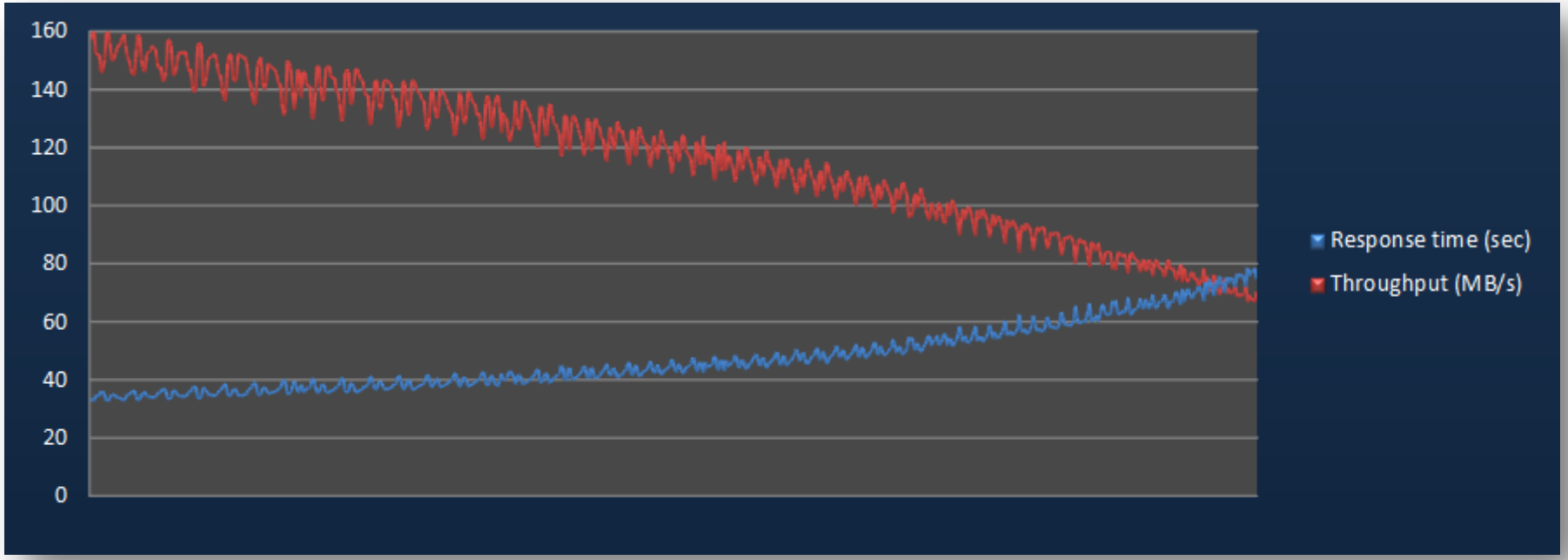
absolutely™ WD

Throughput *f*(block size)



absolutely™ WD

Throughput *f*(LBA)

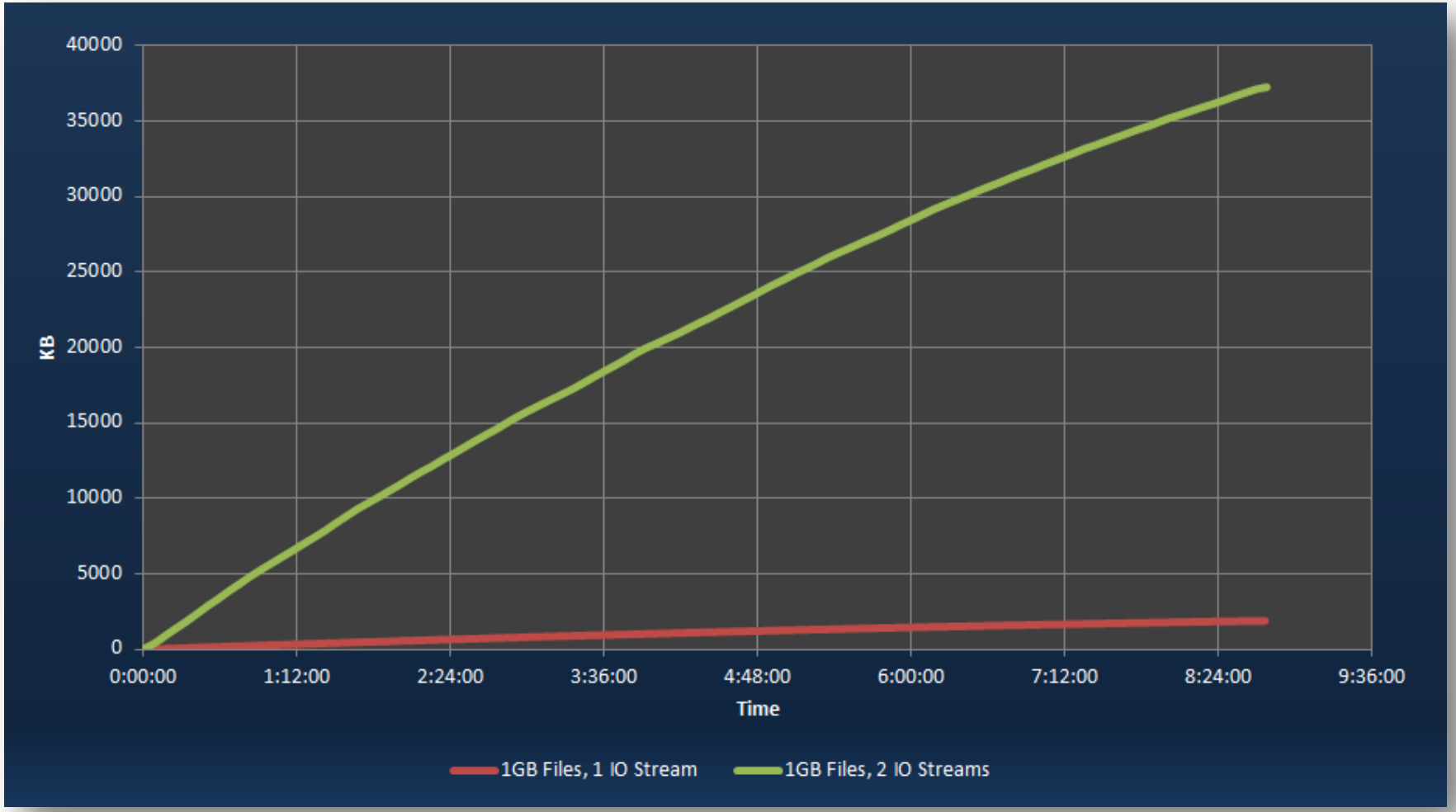


absolutely™ WD

Write sequential: ~116MB/s

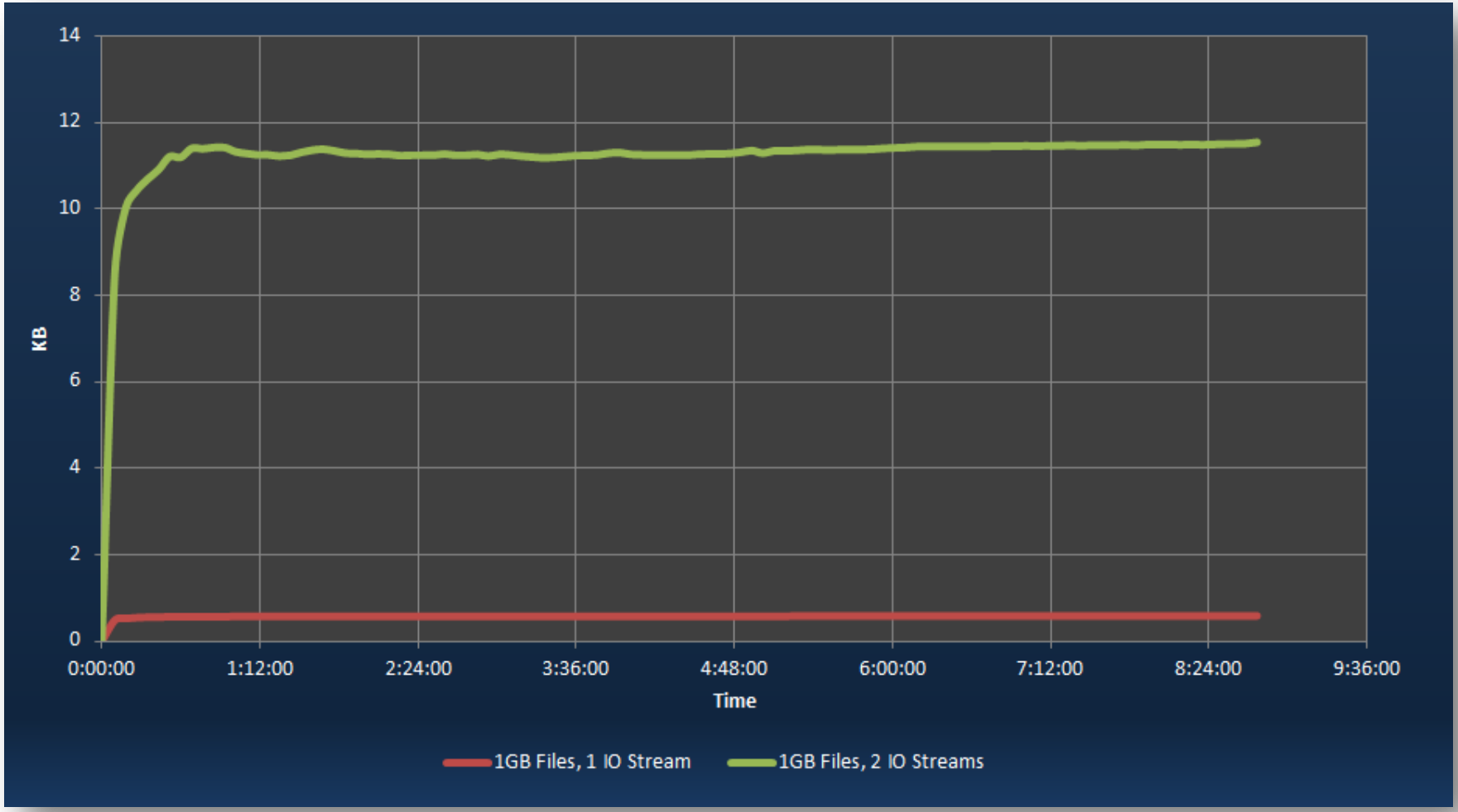
Metadata

Index size growth *f*(t)



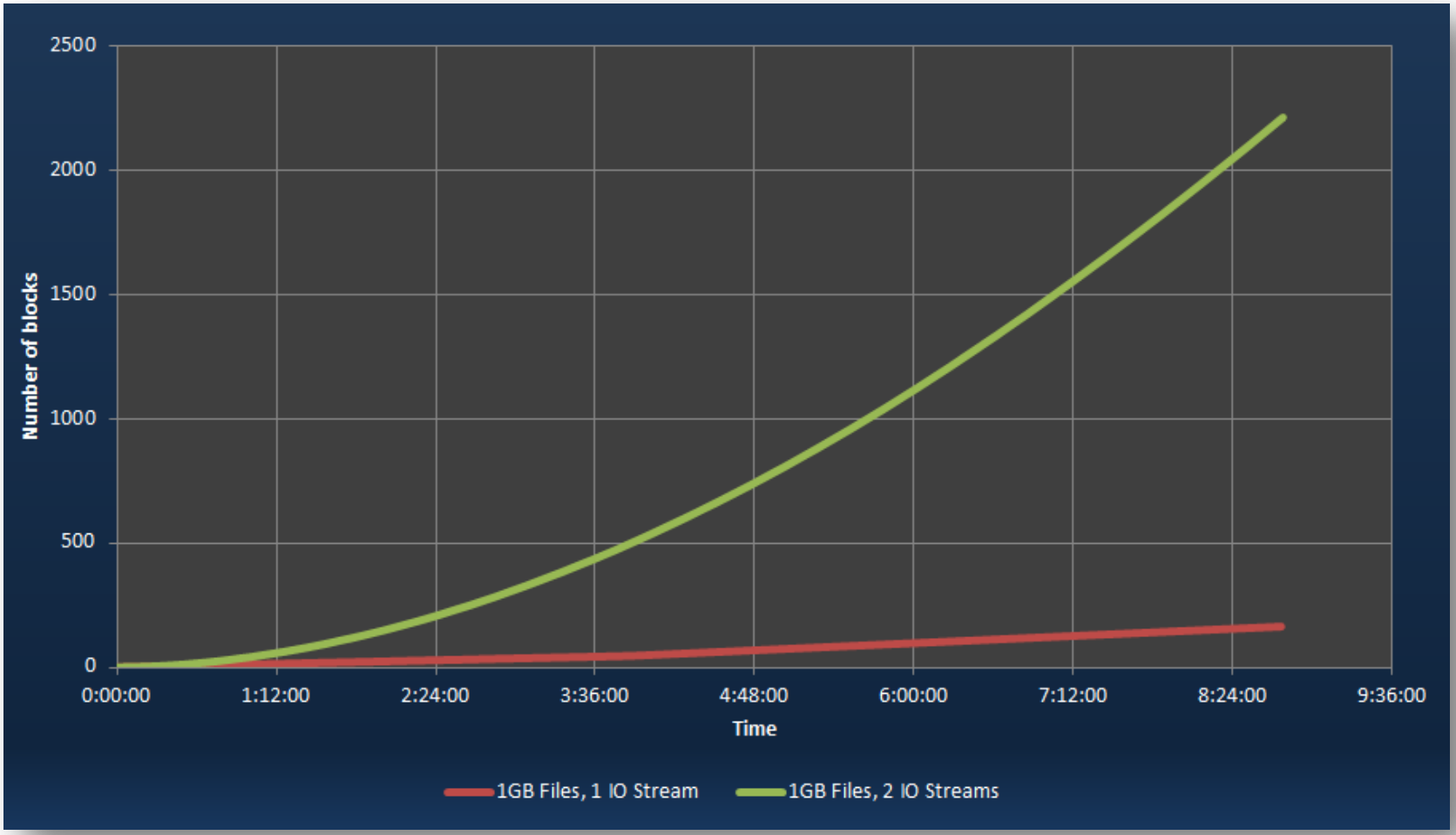
absolutely™ WD

Index size/file *f*(t)



absolutely™ WD

Index block allocation (1MB) *f*(t)



absolutely™ WD

System information

- Dell PowerEdge T110
- 2GB 1333 MHz DRAM
- Intel Xeon E3-1220 CPU
- LSI SAS9207-8e HBA

CPU and memory usage *f*(t)

absolutely™ WD

What have we learned?

1. LTFS interoperability
2. Performance
3. Testing

Host managed SMR

Feedback

- Customers “dig” it - easy to use
- Writes are 100% sequential – fastest way to store data on HDD
- Read access at full HDD speeds (random and sequential)
- Smallest overprovisioned solution – **lowest $/GB**
- Simple design – lower cost – higher reliability