enmotus

Adding Analytical Behavioral Intelligence to Block Storage Layer

andy mills, co-founder/ceo

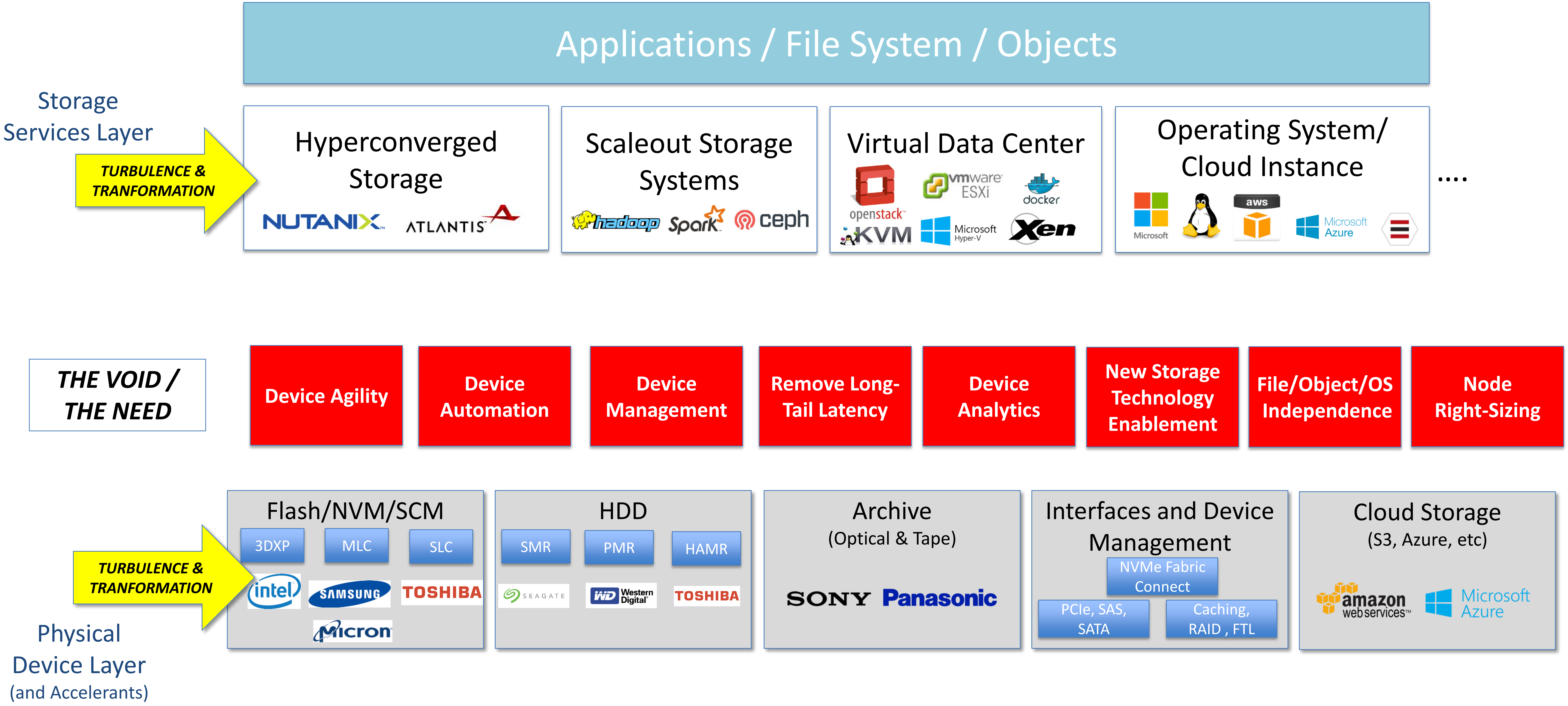
June 2nd 2016

www.enmotus.com

Topics

- Need for a fast, efficient storage device virtualization (SDV) layer
- Behavioral analysis and automation of storage devices
- Enmotus FuzeDrive
- Open REST/JSON for storage device telemetry data collection

Evolving Applications, Stack and Devices



Applications / File System / Objects

Storage Services Layer

TURBULENCE & TRANFORMATION

Hyperconverged Storage

Scaleout Storage Systems

Virtual Data Center

Operating System/ Cloud Instance

....

THE VOID / THE NEED

- Device Agility
- Device Automation
- Device Management
- Remove Long-Tail Latency
- Device Analytics
- New Storage Technology Enablement
- File/Object/OS Independence
- Node Right-Sizing

Flash/NVM/SCM
- 3DXP
- MLC
- SLC

HDD
- SMR
- PMR
- HAMR

Archive
(Optical & Tape)

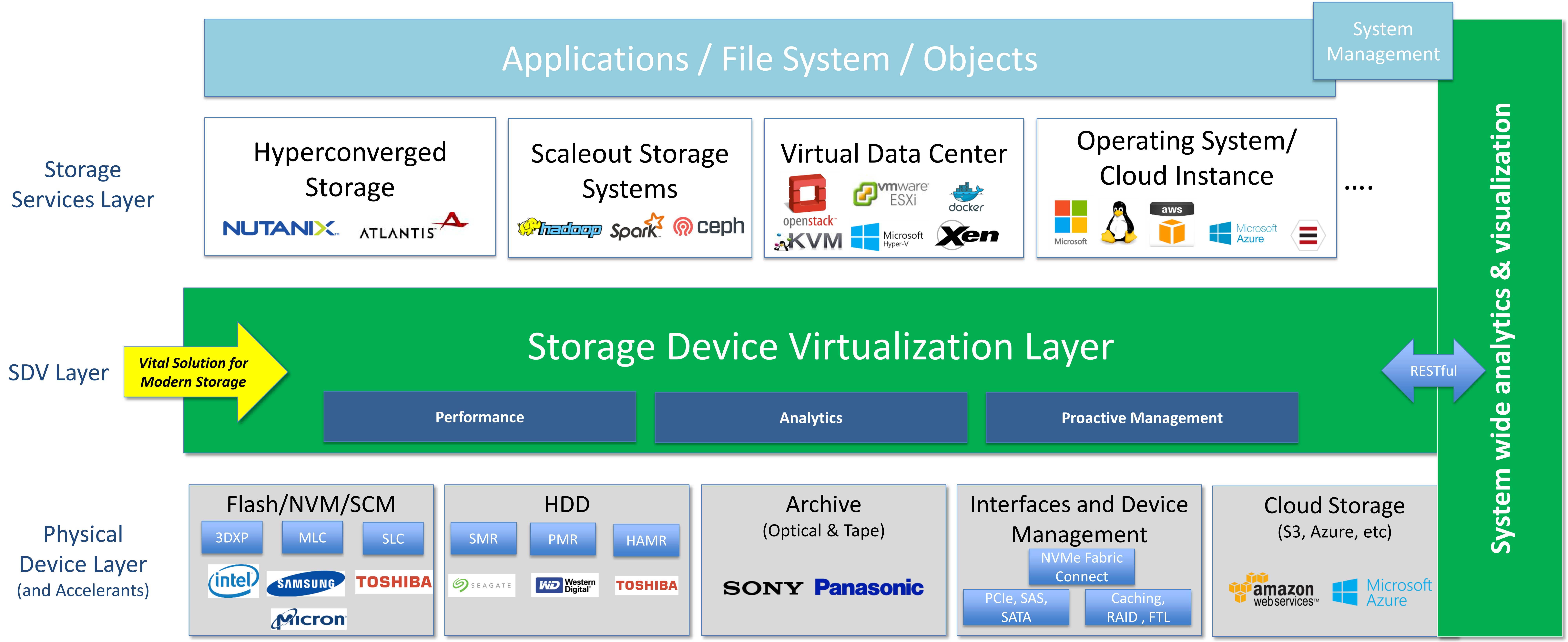
Interfaces and Device Management
- NVMe Fabric Connect
- PCIe, SAS, SATA
- Caching, RAID , FTL

Cloud Storage
(S3, Azure, etc)

TURBULENCE & TRANFORMATION

Physical Device Layer
(and Accelerants)

Evolving Applications, Stack and Devices



Applications / File System / Objects

System Management

System wide analytics & visualization

Storage Services Layer

Hyperconverged Storage

Scaleout Storage Systems

Virtual Data Center

Operating System/ Cloud Instance

....

SDV Layer

Vital Solution for Modern Storage

Storage Device Virtualization Layer

- Performance
- Analytics
- Proactive Management

RESTful

Physical Device Layer (and Accelerants)

Flash/NVM/SCM
- 3DXP
- MLC
- SLC

HDD
- SMR
- PMR
- HAMR

Archive (Optical & Tape)

Interfaces and Device Management
- NVMe Fabric Connect
- PCIe, SAS, SATA
- Caching, RAID , FTL

Cloud Storage (S3, Azure, etc)

enmotus

Storage Device Virtualization

- Intelligent storage device software layer
  - Behavioral approach to mapping devices to application workloads
  - Autonomous and centralized device management
  - Fast translation i.e. minimal impact IO performance and latency
- Benefits
  - Node level – automatically load balance across RAM, SSD, HDD
  - System level – detect and isolate issues such as long tail latency
  - Central collector – analyze and correct device behavior
- Open - provide APIs via JSON/RESTful protocols
  - Connectors to other tools e.g. Splunk or internal management

enmotus

What SDV is Not….

A New Clustered File System

Just Another SSD Cache

Software Defined Storage

- Clustered File System
  - Complementary to SDV
  - Usually requires a separate inter-node communications channel
  - Also used in shared/clustered SSD caching (pseudo file system)
- SSD Caching
  - Optimized around HDD/SAN acceleration hence
    - Up to 80% of the SSD raw performance is lost
  - Often tied to specific vendor SSDs
  - CPU intensive as size and activity levels rise
- Software Defined Storage
  - Complementary to SDV
  - Acts at a high layer – optimized around commodity hardware use and standard operating systems
  - SAN replacement

enmotus

Google’s Disk for Data Centers

Key Problems Identified	Storage Device Virtualization (SDV)
Balanced application of DRAM/SSD/HDD	Automated, intelligent real time block or memory migration between devices
Move cache from disks to hosts	Automatically choose most appropriate cache media RAM, NVRAM, SSD
Hybrid use of CMR and SMR drives	Automatically map to CMR or SMR (all types) based on detected traffic patterns
Host managed retries to contain tail latency	Manage long tail latency through both active and passive behavioral analysis versus than just simple SMART logs reporting
Capture more performance info to manage tail latency	Combined spatial and temporal statistics can better determine where the origins of tail latency lie and enable better automation of fixes
Flash device behavior with respect to uncorrectable events is problematic	Machine level behavioral analysis can automatically correct problematic devices

Source: Google paper, Disks for Data Centers by Lawrence Ying et al. https://research.google.com/pubs/pub44830.html

Relevance to OCP-Storage

- SDV designed to be hardware and SDS agnostic
  - Full blown x86 server-storage platforms
  - Lightweight Honey Badger/ATOM or ARM32/64
  - High performance Knox/Lightening configurations
- Device Flexibility
  - Handle NVMe, SAS, SATA with single stack
  - Path to pmem/NVRAM/SCM class devices
  - NVMe over fabric
- Storage specific management layer

Enmotus Storage Device Virtualization

enmotus

Storage Device Virtualization

Virtualize

- Memory and SSD class performance
- Non-disruptively add, move, change storage devices
- For modern scaleout and conventional environments

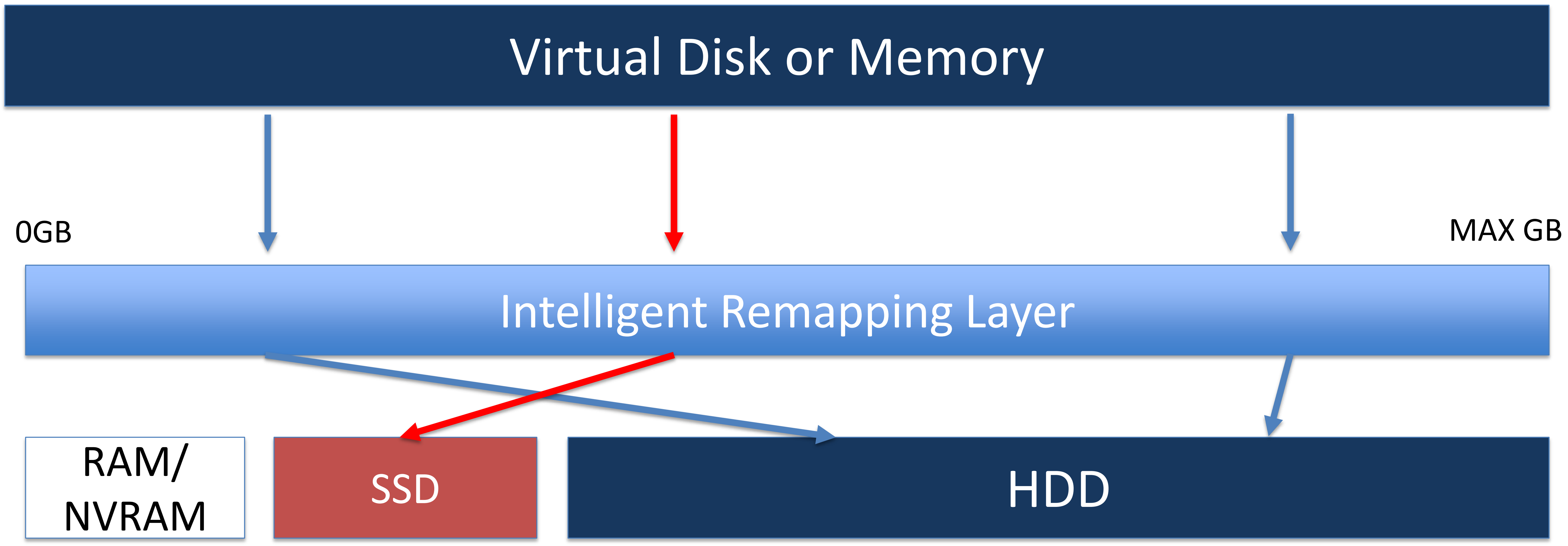
Analyze

- Deep local analysis of device behavior
- Open device storage log repository
- Intelligent centralized reporting
- Uniform, media agnostic reporting
- Spatial and temporal analysis

Optimize

- Real-time balancing of performance, capacity and cost
- Automated device relocation
- Policy driven

Virtualize



Virtual Disk or Memory

0GB

MAX GB

Intelligent Remapping Layer

RAM/ NVRAM

SSD

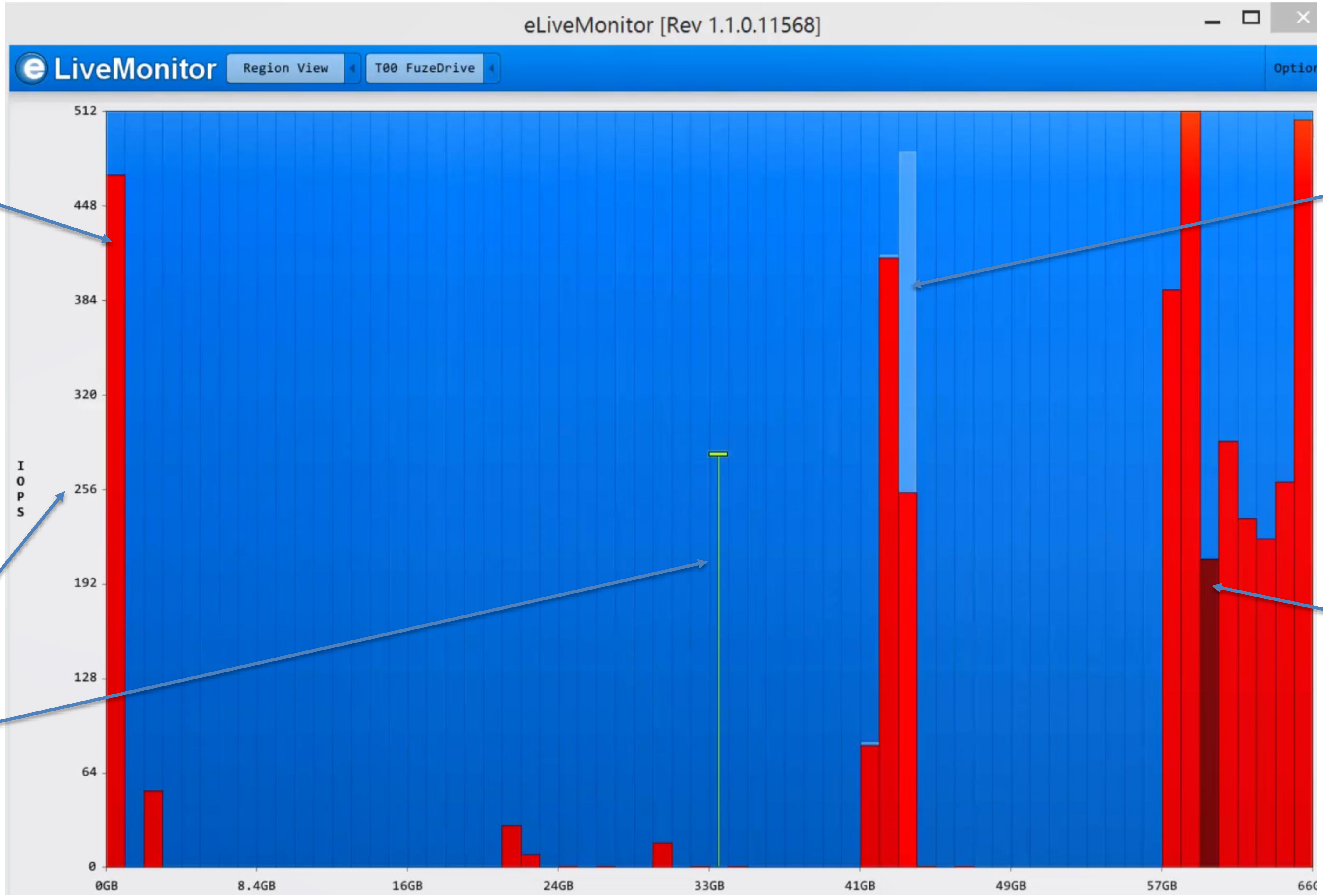
HDD

Block Storage Devices

Video Demo – Real Time Optimization

Device Analytics
by
enmotus

Analyze and Visualize



Percentage of Fast Tier (SSD) mapped in the Capacity/LBA range (red)

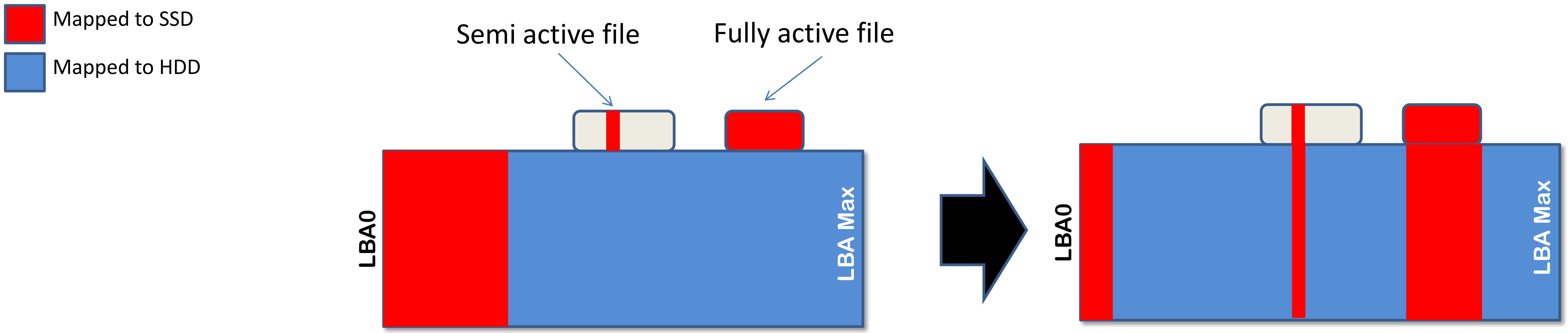
Target range where SSD is to be relocated (pale blue)

Performance (IOPS, MB/s)

Target range where SSD is to be stolen relocated away from (dark red)

Capacity Point/Logic Block Address

Optimize



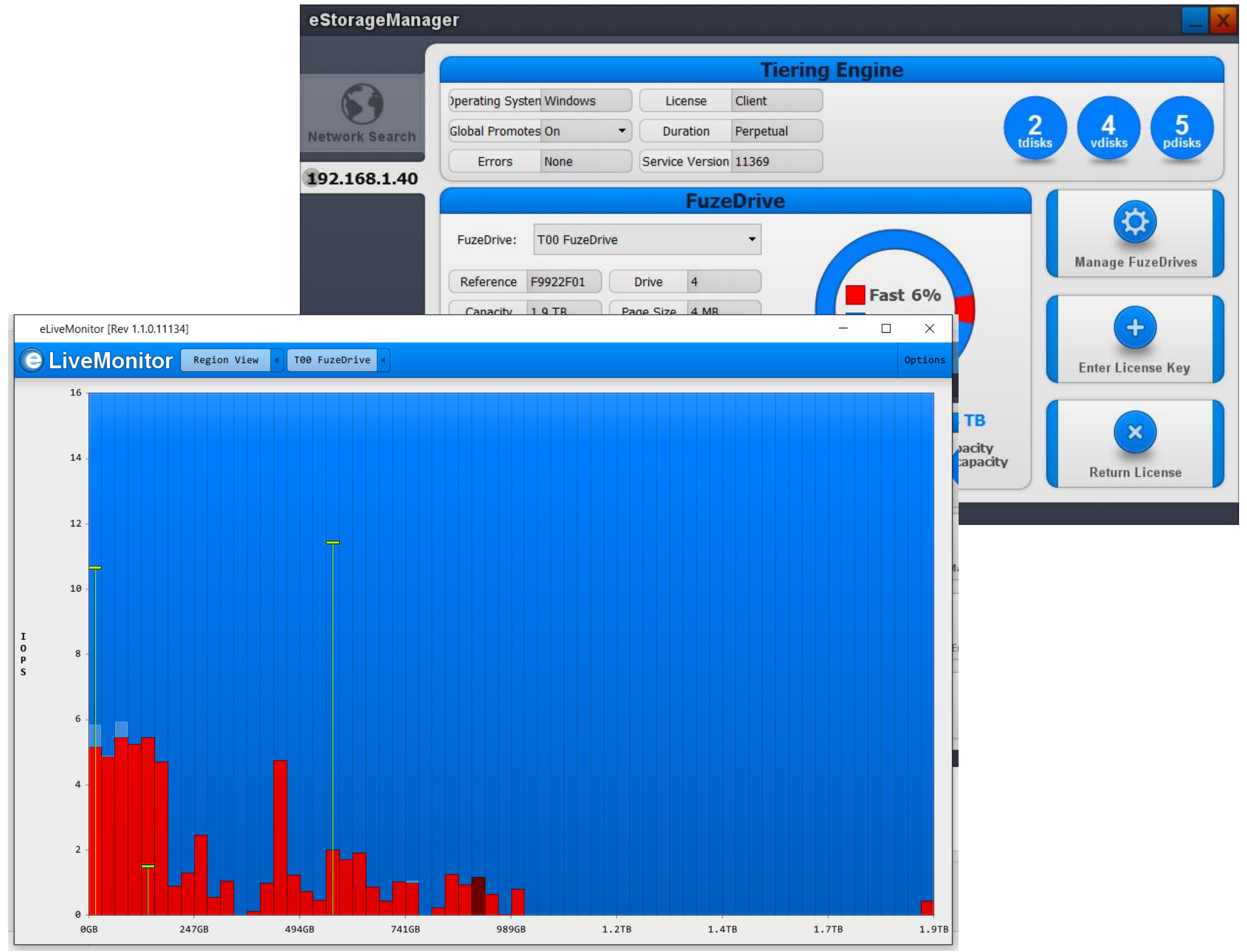
Newly Initialized Tiered Volume

Post Tiered Volume

- The active portions of files **relocate** to the SSD in real time
- 100% block/LBA based decision engine with rigidity controls for each movable page: full float, pin to tier, rigidity setting
- Instant usage at full speed of the SSD (reads and writes) at low LBA ranges: user has an instant SSD experience

Enmotus FuzeDrive

High capacity hybrid SSD class storage for any Intel or ARM class storage server


eStorageManager

Tiering Engine

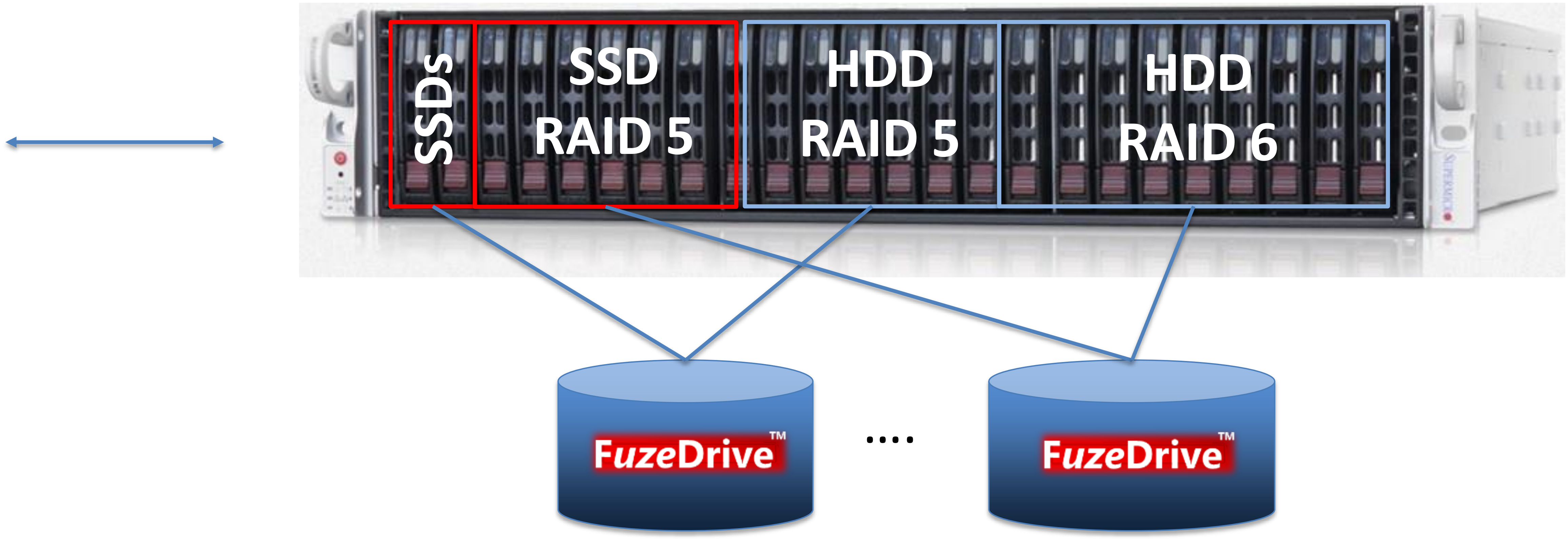
)perating Systen Windows
License Client
Global Promotes On
Duration Perpetual
Errors None
Service Version 11369

192.168.1.40

FuzeDrive

FuzeDrive: T00 FuzeDrive
Reference F9922F01
Drive 4
Capacity 1.9 TB
Page Size 4 MB

eLiveMonitor [Rev 1.1.0.11134]



- FuzeDrive remote management
- Tiered storage provisioning
- Innovative live activity monitor

- Up to 4 high performance virtual disks (FuzeDrives)
- Fully automated block level tiering
- SAS/SATA or PCIe/NVMe SSD plus HDD tiering
- Supports Windows, Hyper-V, Linux, KVM, Xen

FuzeDrive™ Technology

Storage Device Virtualization

- High performance device level virtualization layer
- Add, remove, change SSDs on live volumes
- MicroTiering™ automatically migrates data across 2 levels of storage
- RAM cache for burst traffic up to 20GB/s

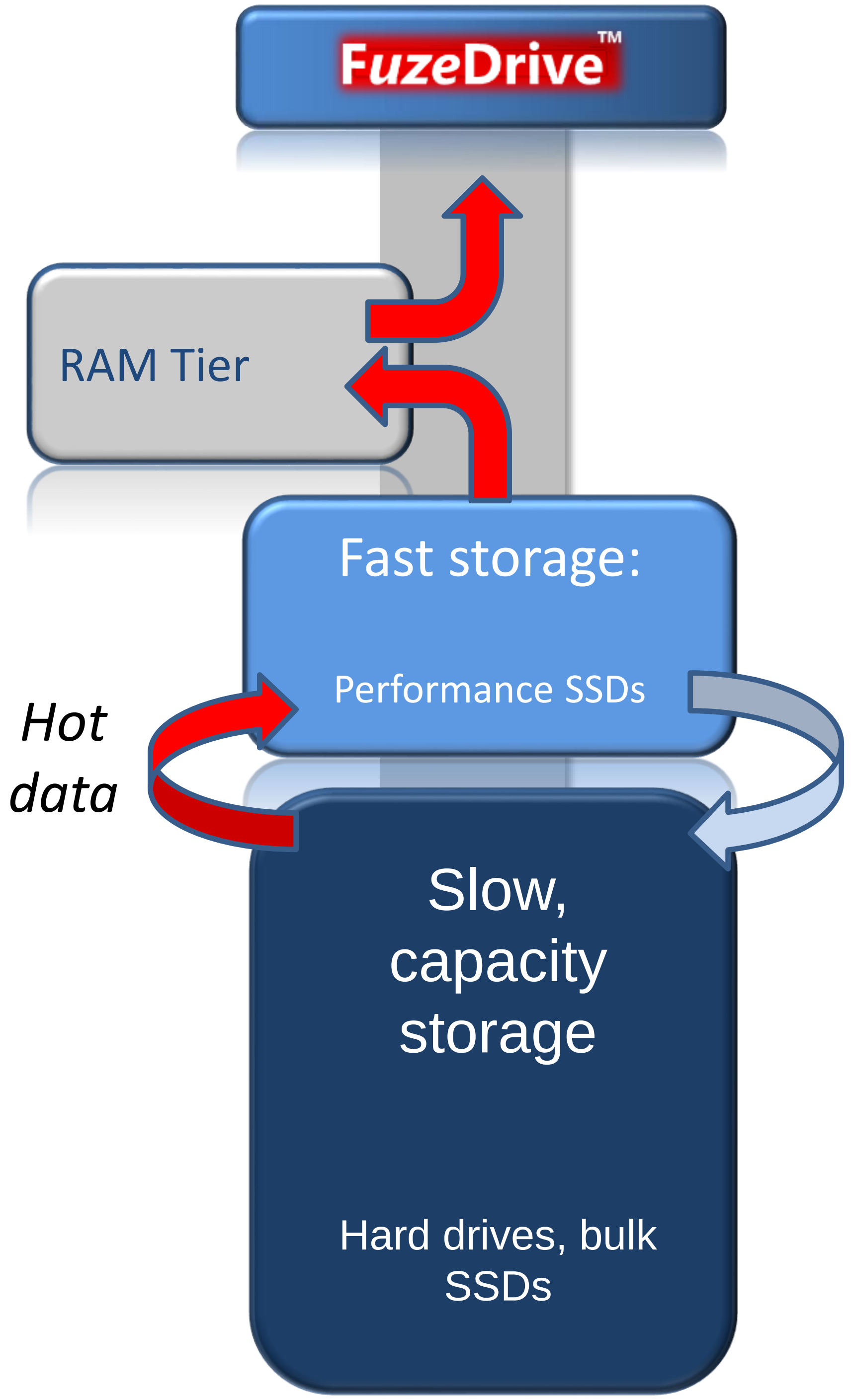
Spans multiple environments

- Virtual servers in both public, private and hybrid clouds
- Embed in to OEM storage solution or standalone software
- All major Linux distros and MS Windows

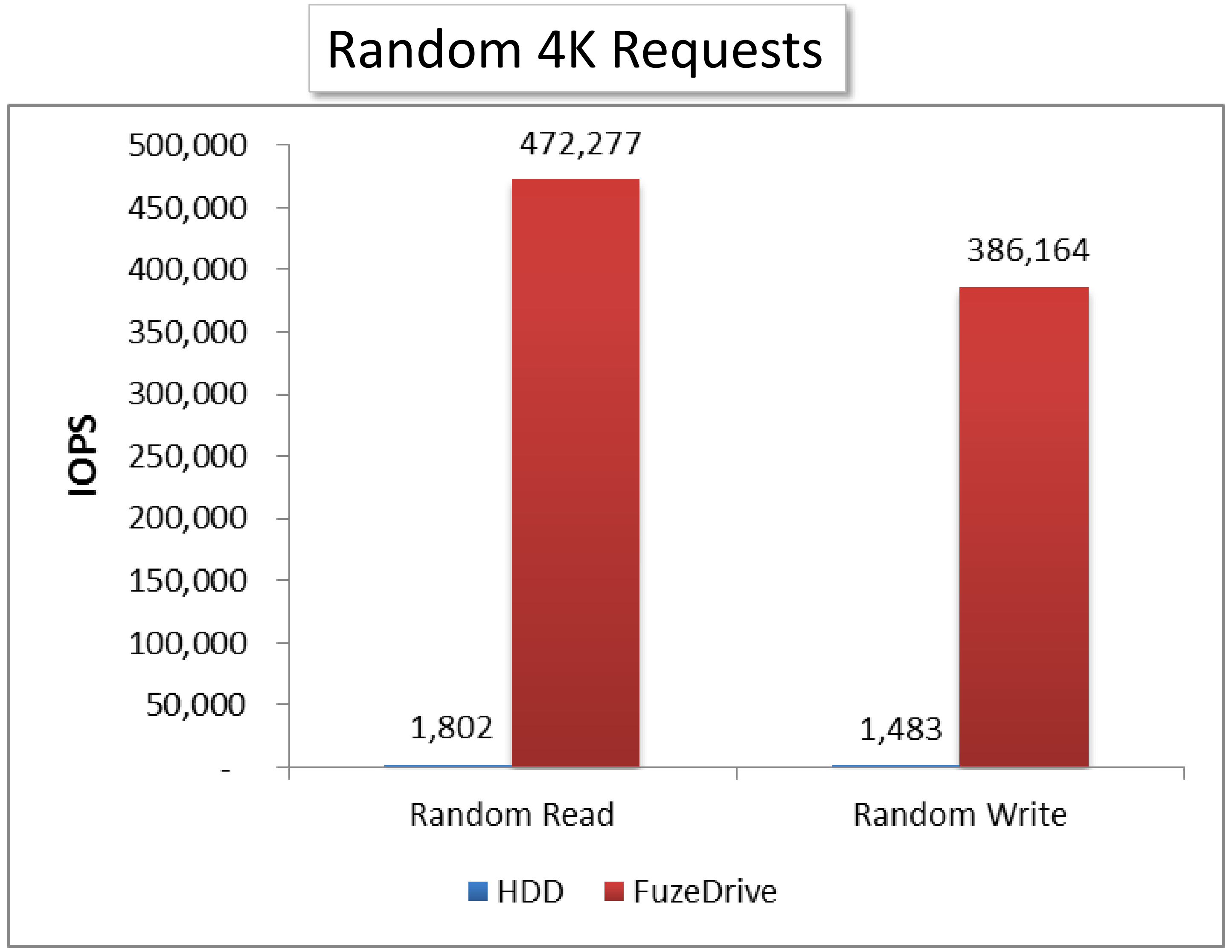
Key Benefits

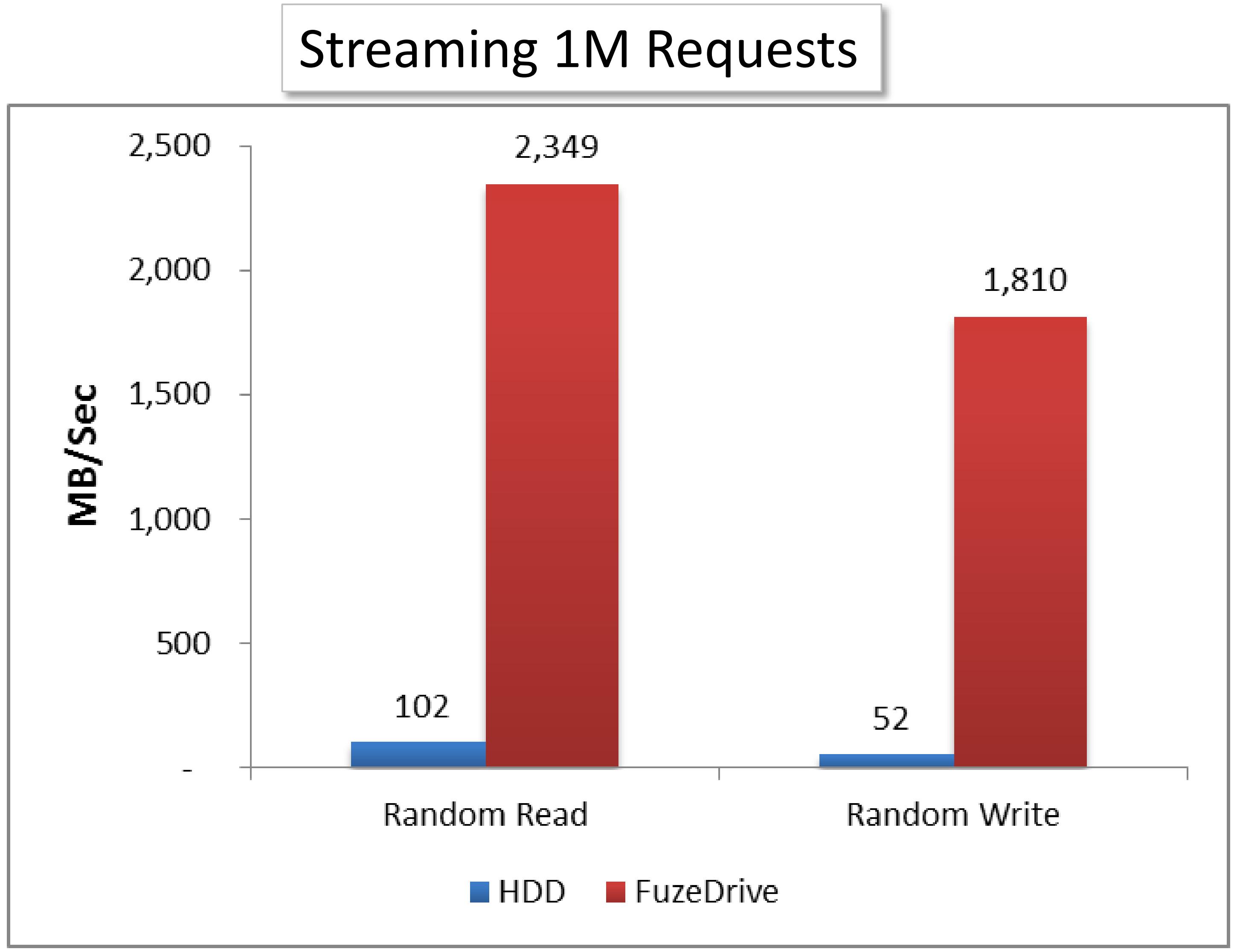
- Operate at full SSD RD-WR rates with HDD capacities
- Streaming and random traffic - >11x faster than SSD caching
- RAM cache up to 20GB/s sequential burst

Appears as standard Windows or Linux block device



Performance: PCIe NVMe Example





- Up to 260x faster in raw performance than RAID 6 for same capacity
- Example shown:
  - Linux CentOS 7 36-bay storage-server
  - Single PCIe NVMe SSD fuzed with RAID6 8-drive 6TB drives

Source: http://www.colfax-intl.com/nd/solutions/enmotus-fuzedrive.aspx

Supported Devices

- Solid State Devices
  - PCIe SSDs: NVMe, Micron P3/4xxx, FusionIO
  - SAS 6/12G: All industry standard devices
  - SATA 3/6G: All industry standard devices

- Memory Class Devices
  - NVDIMM: Micron, SMART, Netlist, Viking
  - Diablo/Sandisk UlltraDIMM

- Virtual Devices Tested
  - Hardware RAID: LSI MegaRAID, Adaptec, Marvell
  - Microsoft storage spaces devices
  - DotHill/Dell PERC S110 software RAID
  - Virtual disk service tiering: AWS, Azure

Enmotus Community Device Telemetry API

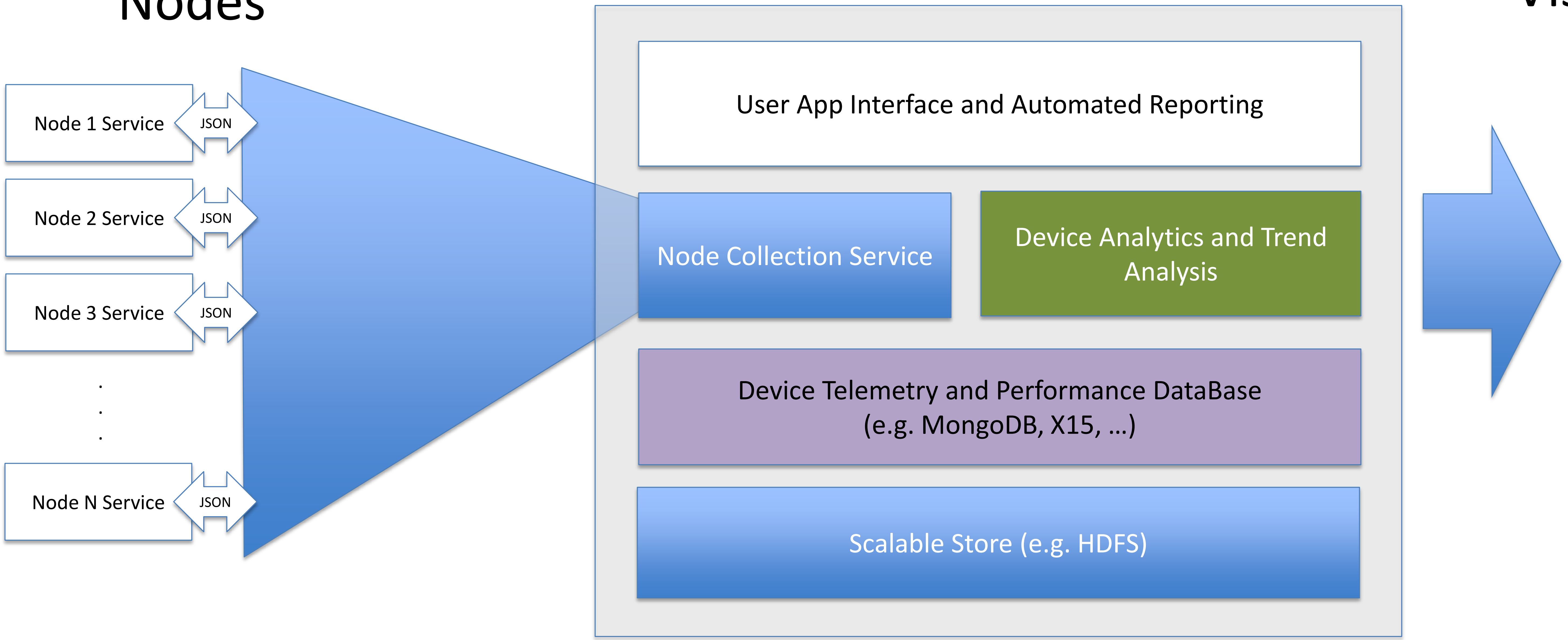
- Provide an open API for device telemetry based on JSON/RESTful
- Enable a standard way to extract SMART, SCSI, NVMe log and performance IO data
- Publish spec for OCP/community in June/July timeframe
- Release free/community JSON/RESTful management agent for several Linux distros

Centralized Collector

Compute/Storage Nodes

Central Device Analytics Server

Visual Reporting Tools



Node 1 Service

Node 2 Service

Node 3 Service

.
.
.

Node N Service

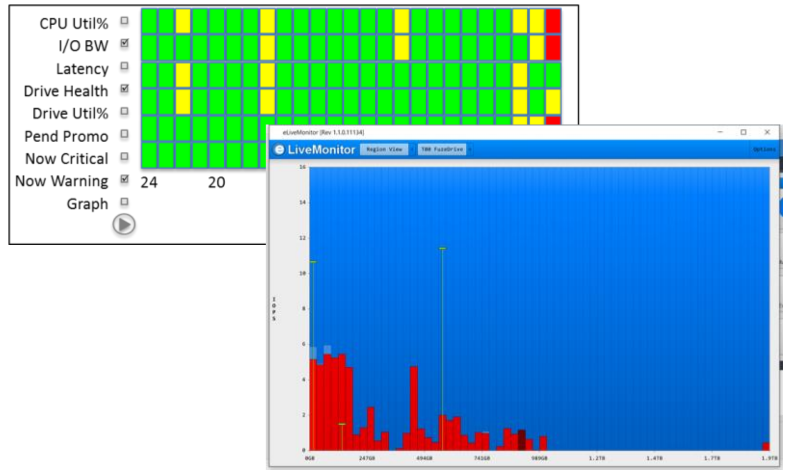
User App Interface and Automated Reporting

Node Collection Service

Device Analytics and Trend Analysis

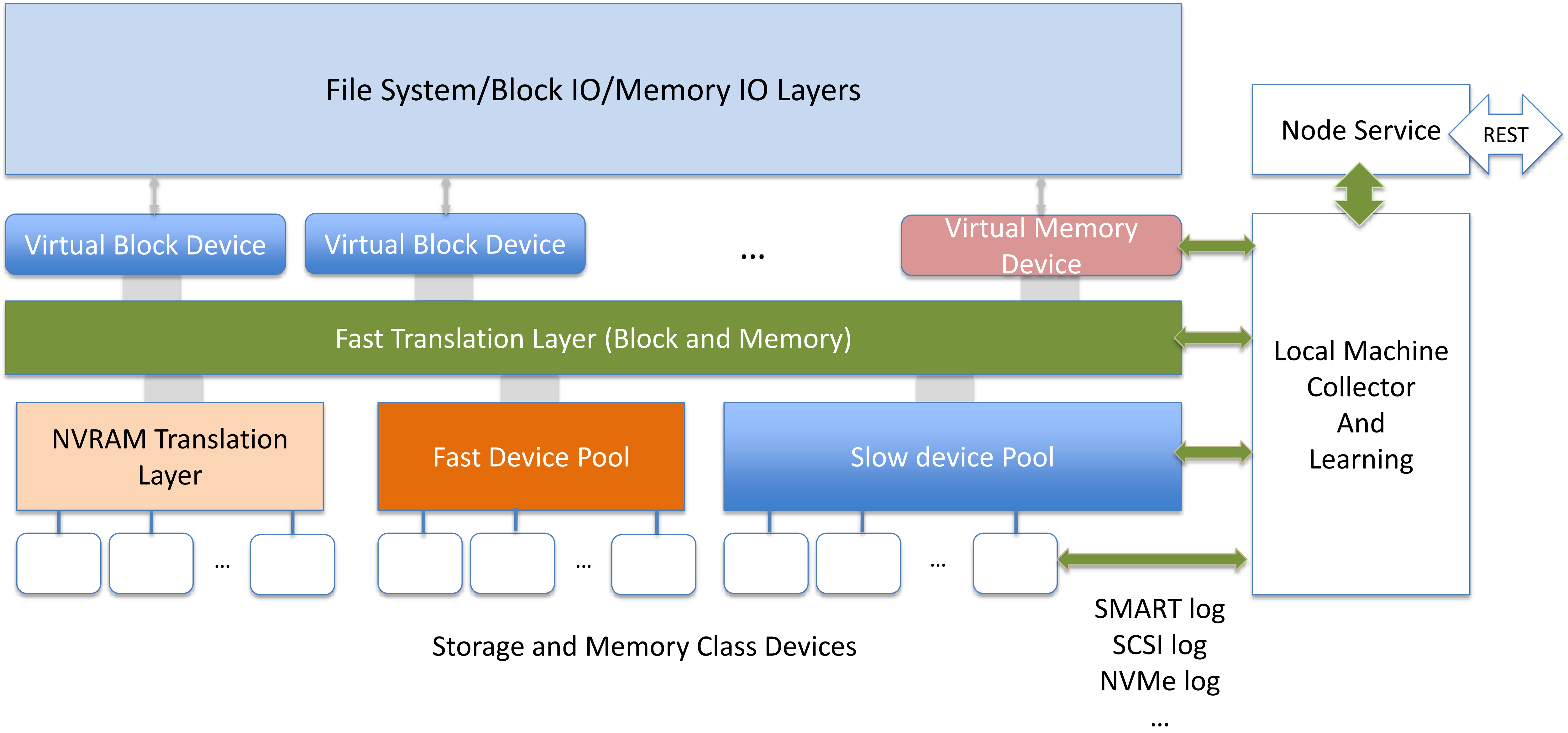
Device Telemetry and Performance DataBase
(e.g. MongoDB, X15, …)

Scalable Store (e.g. HDFS)

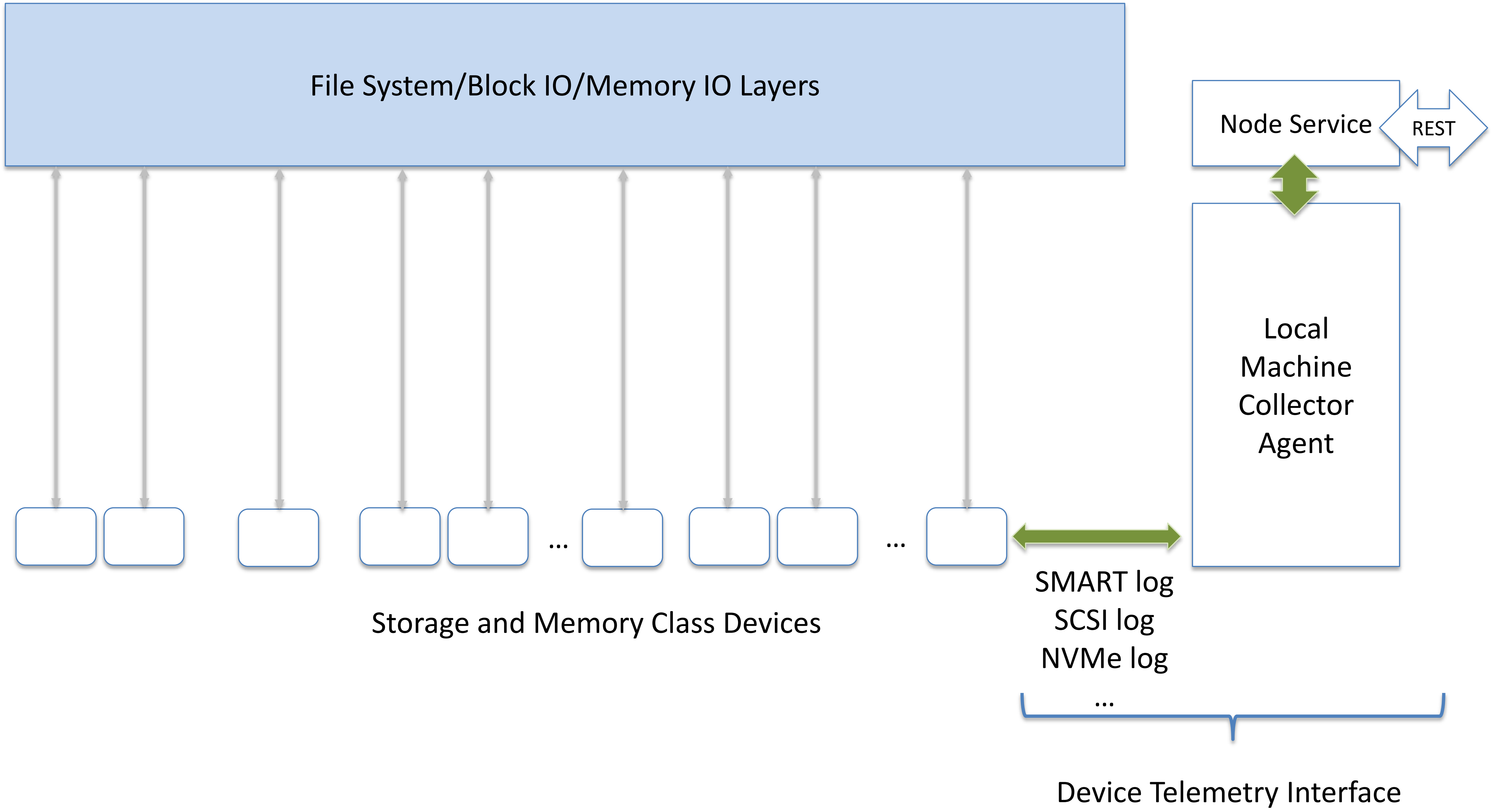


enmotus

Fully Virtualized Nodes



Regular Node



File System/Block IO/Memory IO Layers

Node Service

REST

Local Machine Collector Agent

…

…

Storage and Memory Class Devices

SMART log
SCSI log
NVMe log
…

Device Telemetry Interface

REST API for Storage Telemetry

- API leverages features of the HTTP protocol
  - Drives are modeled as a REST resource, represented as a URI
  - Uses GET method to retrieve drive information
  - Uses HTTP Authentication methods when applicable
- JSON is used to represent the information
- Drive information is retrieved through the API
  - Lists of drives, vdrives, and pdrives are returned with GETs
  - Individual drive, vdrive, or pdrive information is returned using the IDs returned above for virtualized storage nodes

Community Release

- Initial 0.1 release Jun/July
  - RESTful/JSON definition document
  - Example node software (RPM, DEB)
- REST Features
  - Drive list
  - IOstat information by drive
  - Select SMART data by drive

enmotus

IOSTAT Metrics

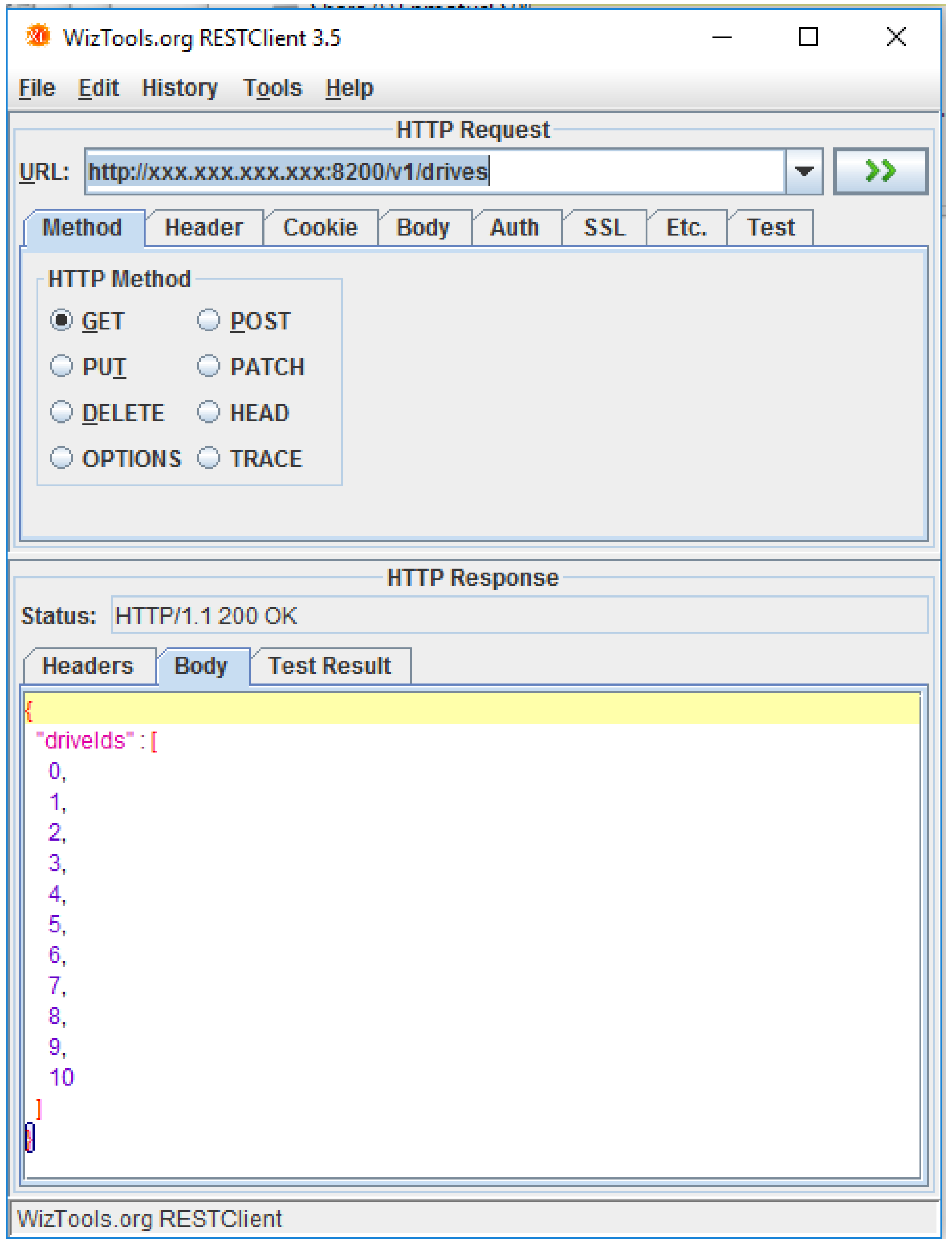
- rrqm/s
  - The number of read requests merged per second that were queued to the device
- wrqm/s
  - The number of write requests merged per second that were queued to the device
- r/s
  - The number of read requests that were issued to the device per second
- w/s
  - The number of write requests that were issued to the device per second
- rkB/s
  - The number of kilobytes read from the device per second
- wkB/s
  - The number of kilobytes written to the device per second
- avgrq-sz
  - The average size (in sectors) of the requests that were issued to the device
- avgqu-sz
  - The average queue length of the requests that were issued to the device
- await
  - The average time (in milliseconds) for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them
- r_await
  - The average time (in milliseconds) for read requests issued to the device to be served
- w_await
  - The average time (in milliseconds) for write requests issued to the device to be served
- %util
  - Percentage of CPU time during which I/O requests were issued to the device (bandwidth utilization for the device). Device saturation occurs when this value is close to 100%

enmotus

SMART Metrics

- Overall health self-assessment test result
- Remaining SMART metrics return current, worst, threshold, and raw values. Supported SMART fields, if available:
  - ID 5 – Reallocated Sector Count
  - ID 172/182 – Erase Fail Count
  - ID 187 – Reported Uncorrectable Errors
  - ID 188 – Command Timeout
  - ID 196 – Reallocation Event Count
  - ID 197 – Current Pending Sector Count
  - ID 198 – Offline Scan Uncorrectable Sector Count

REST API Drive Lists Example


WizTools.org RESTClient 3.5
File Edit History Tools Help
HTTP Request
Method Header Cookie Body Auth SSL Etc. Test
HTTP Response
Headers Body Test Result
WizTools.org RESTClient

URL: http://xxx.xxx.xxx.xxx:8200/v1/drives

HTTP Method

● GET ○ POST
○ PUT ○ PATCH
○ DELETE ○ HEAD
○ OPTIONS ○ TRACE

Status: HTTP/1.1 200 OK

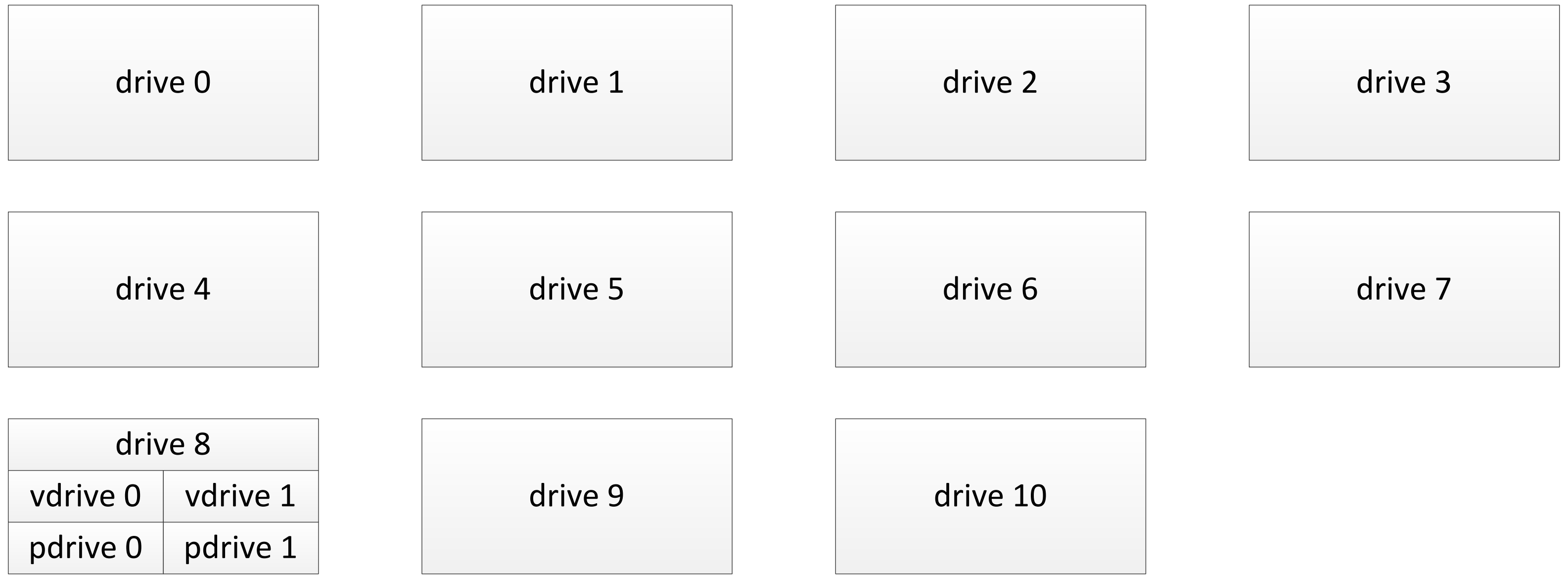
```
{
 "driveIds" : [
   0,
   1,
   2,
   3,
   4,
   5,
   6,
   7,
   8,
   9,
   10
 ]
}
```

GET request: /drives
Response: {driveIds : [driveID1, driveID2, driveID3, . . .]}

GET request: /drives/driveIDX
Response: See next slide

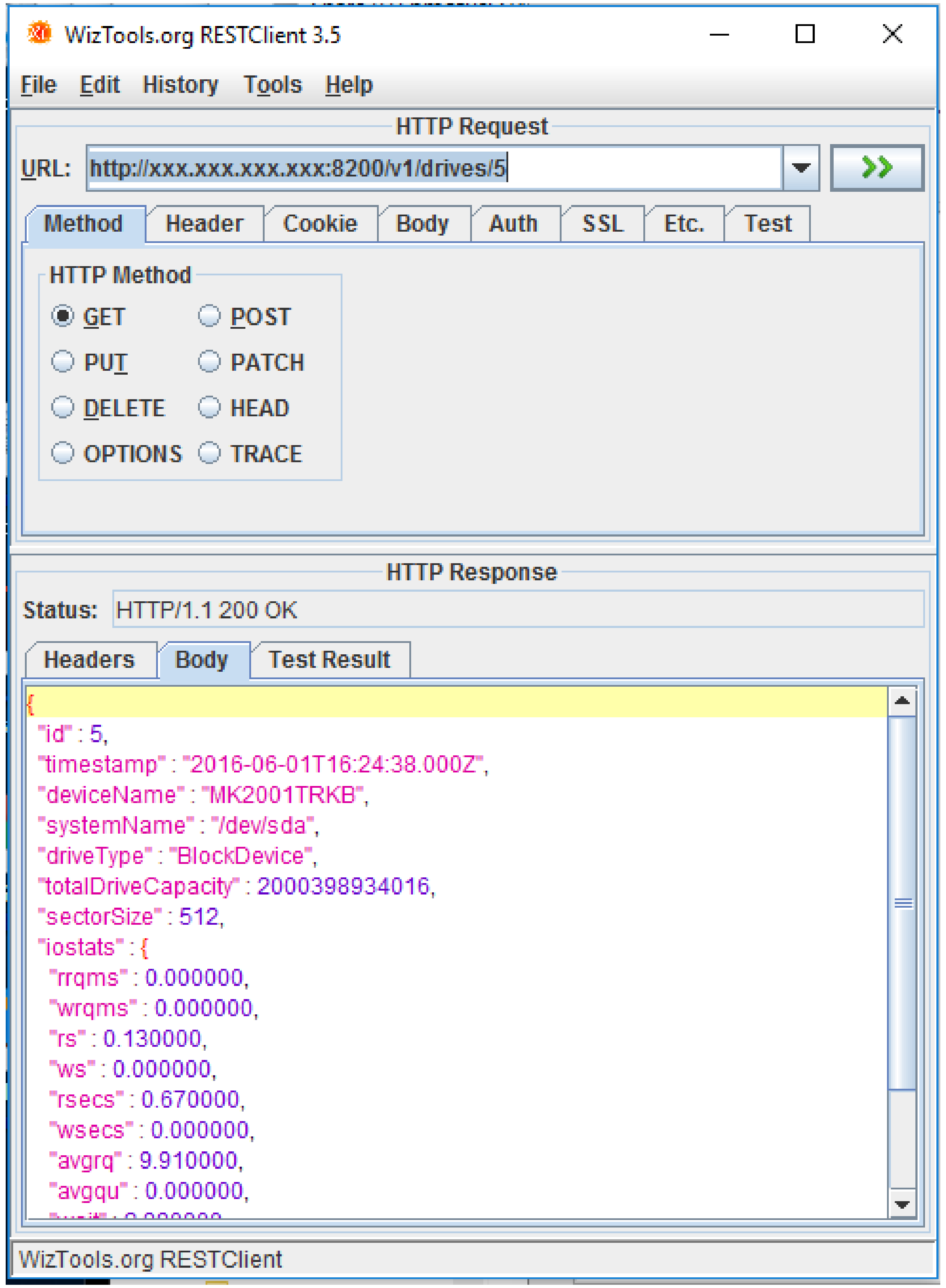
GET Request :/drives/driveIDX/vdrives
Response:{vdriveIds : [vdriveID1, vdriveID2, vdriveID3, . . .]}

GET Request: /drives/driveIDX/vdrives/vdriveIDY/pdrives
Response: {pdriveIds : [pdriveID1, pdriveID2, pdriveID3, . . .]}



enmotus

REST API Block and Virtual Return Example

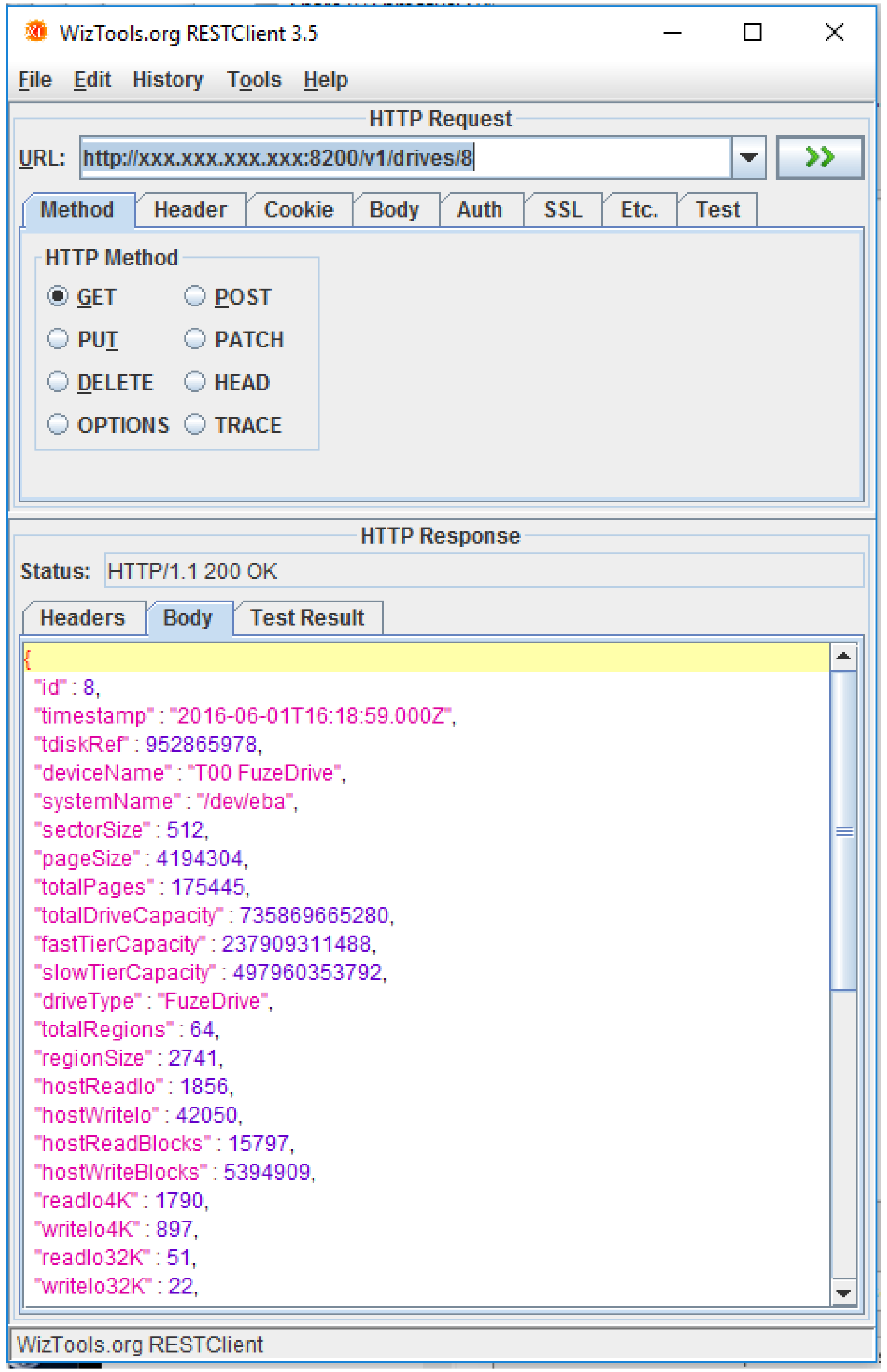


URL: http://xxx.xxx.xxx.xxx:8200/v1/drives/5

● GET

Status: HTTP/1.1 200 OK

```
{
 "id" : 5,
 "timestamp" : "2016-06-01T16:24:38.000Z",
 "deviceName" : "MK2001TRKB",
 "systemName" : "/dev/sda",
 "driveType" : "BlockDevice",
 "totalDriveCapacity" : 2000398934016,
 "sectorSize" : 512,
 "iostats" : {
  "rrqms" : 0.000000,
  "wrqms" : 0.000000,
  "rs" : 0.130000,
  "ws" : 0.000000,
  "rsecs" : 0.670000,
  "wsecs" : 0.000000,
  "avgrq" : 9.910000,
  "avgqu" : 0.000000,
```



URL: http://xxx.xxx.xxx.xxx:8200/v1/drives/8

● GET

Status: HTTP/1.1 200 OK

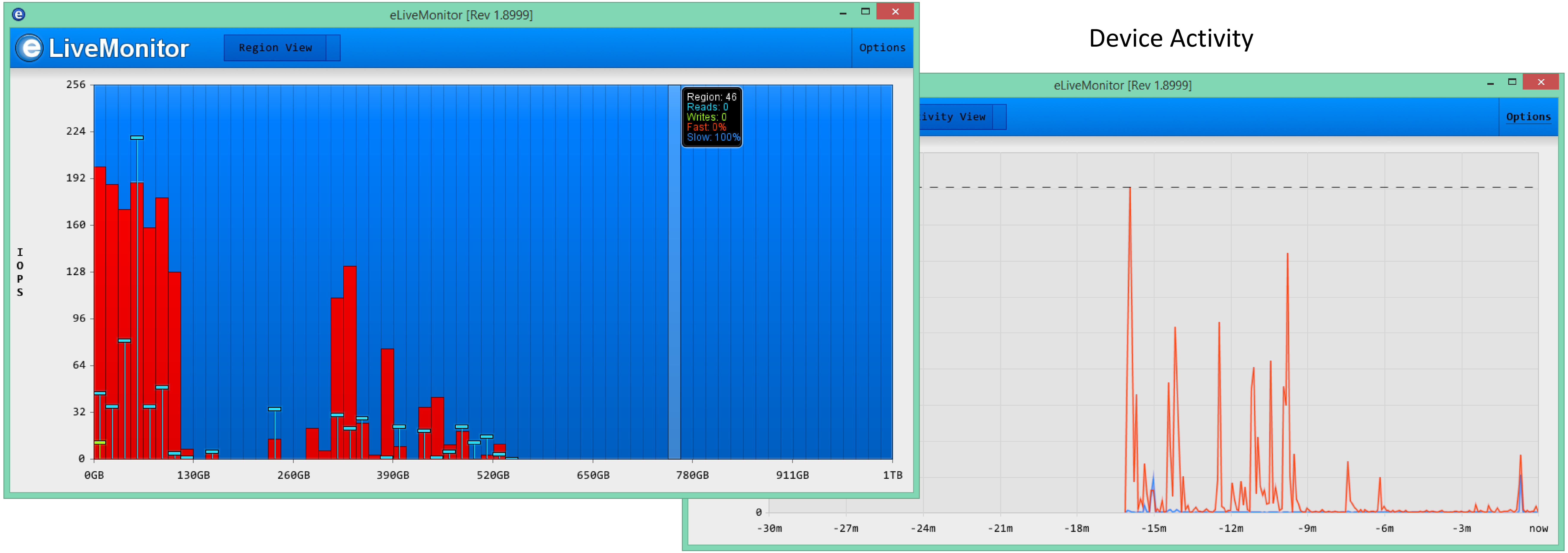
```
{
 "id" : 8,
 "timestamp" : "2016-06-01T16:18:59.000Z",
 "tdiskRef" : 952865978,
 "deviceName" : "T00 FuzeDrive",
 "systemName" : "/dev/eba",
 "sectorSize" : 512,
 "pageSize" : 4194304,
 "totalPages" : 175445,
 "totalDriveCapacity" : 735869665280,
 "fastTierCapacity" : 237909311488,
 "slowTierCapacity" : 497960353792,
 "driveType" : "FuzeDrive",
 "totalRegions" : 64,
 "regionSize" : 2741,
 "hostReadIo" : 1856,
 "hostWriteIo" : 42050,
 "hostReadBlocks" : 15797,
 "hostWriteBlocks" : 5394909,
 "readIo4K" : 1790,
 "writeIo4K" : 897,
 "readIo32K" : 51,
 "writeIo32K" : 22,
```

enmotus

Monitoring Device Activity and Mapping

Mapping vs. Activity (if FuzeDrive Virtual Disk)

Device Activity



enmotus

Thanks!

Please send email to ken.hirata@enmotus.com or andy.mills@enmotus.com if interested in receiving the spec and/or example agent