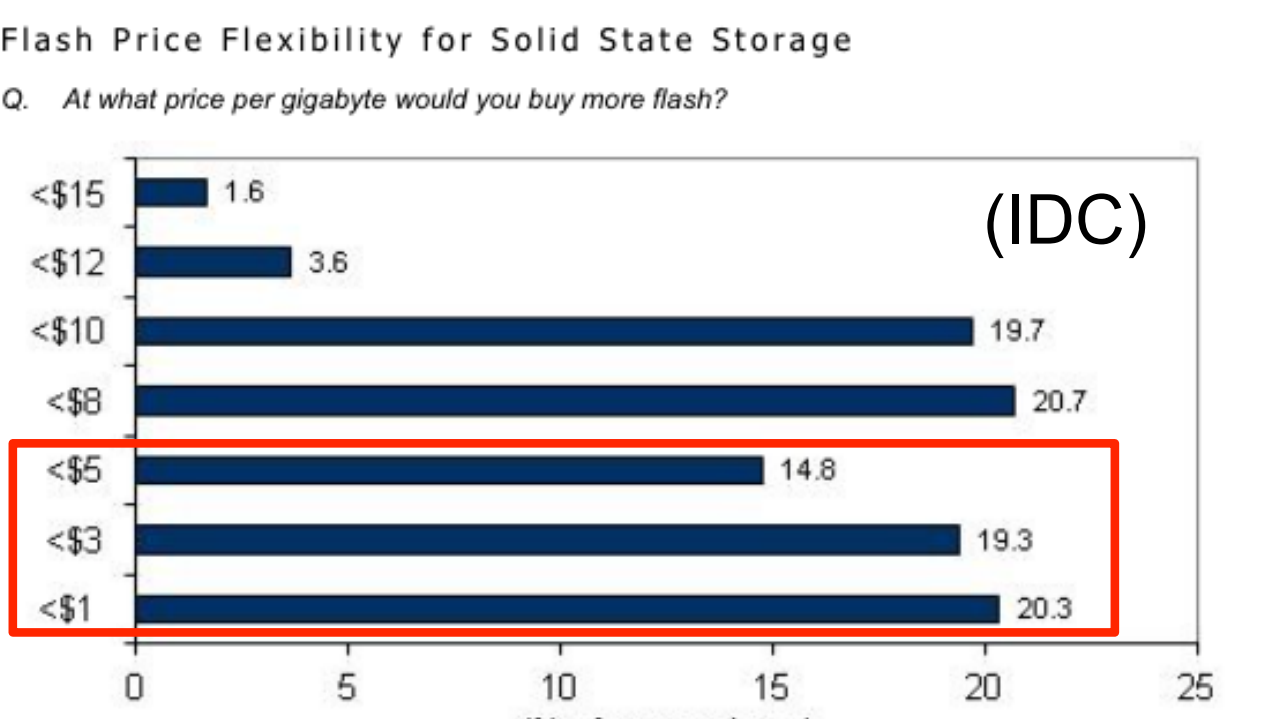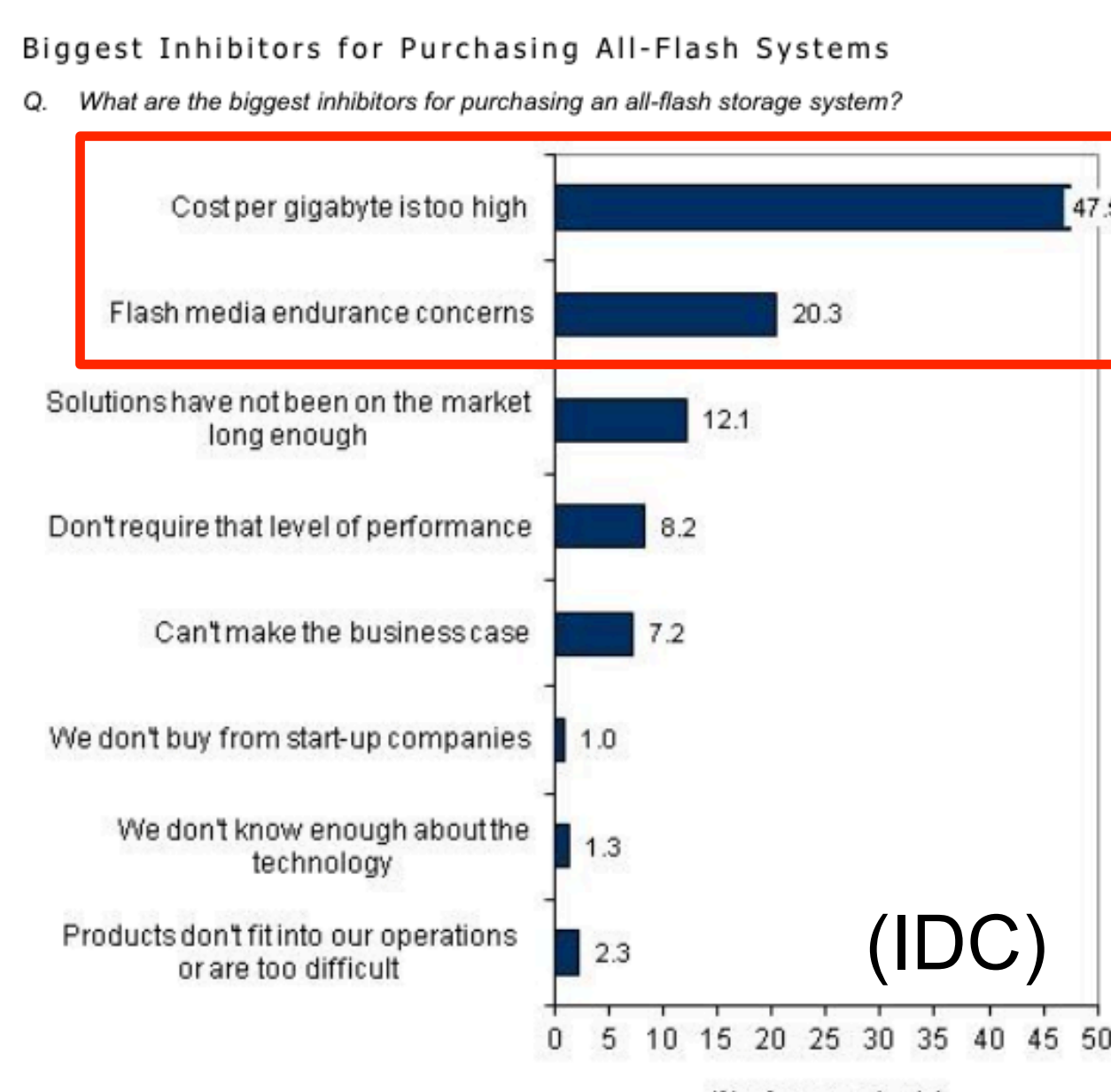# IBM

SALSA: treating the weaknesses of low-cost Flash in software

Nikolas Ioannou, Ioannis Koltsidas, Roman Pletka, Sasa Tomic, Radu Stoica, Thomas Weigold, Evangelos Eleftheriou
{nio, iko, rap, sat, rst, twe, ele}@zurich.ibm.com
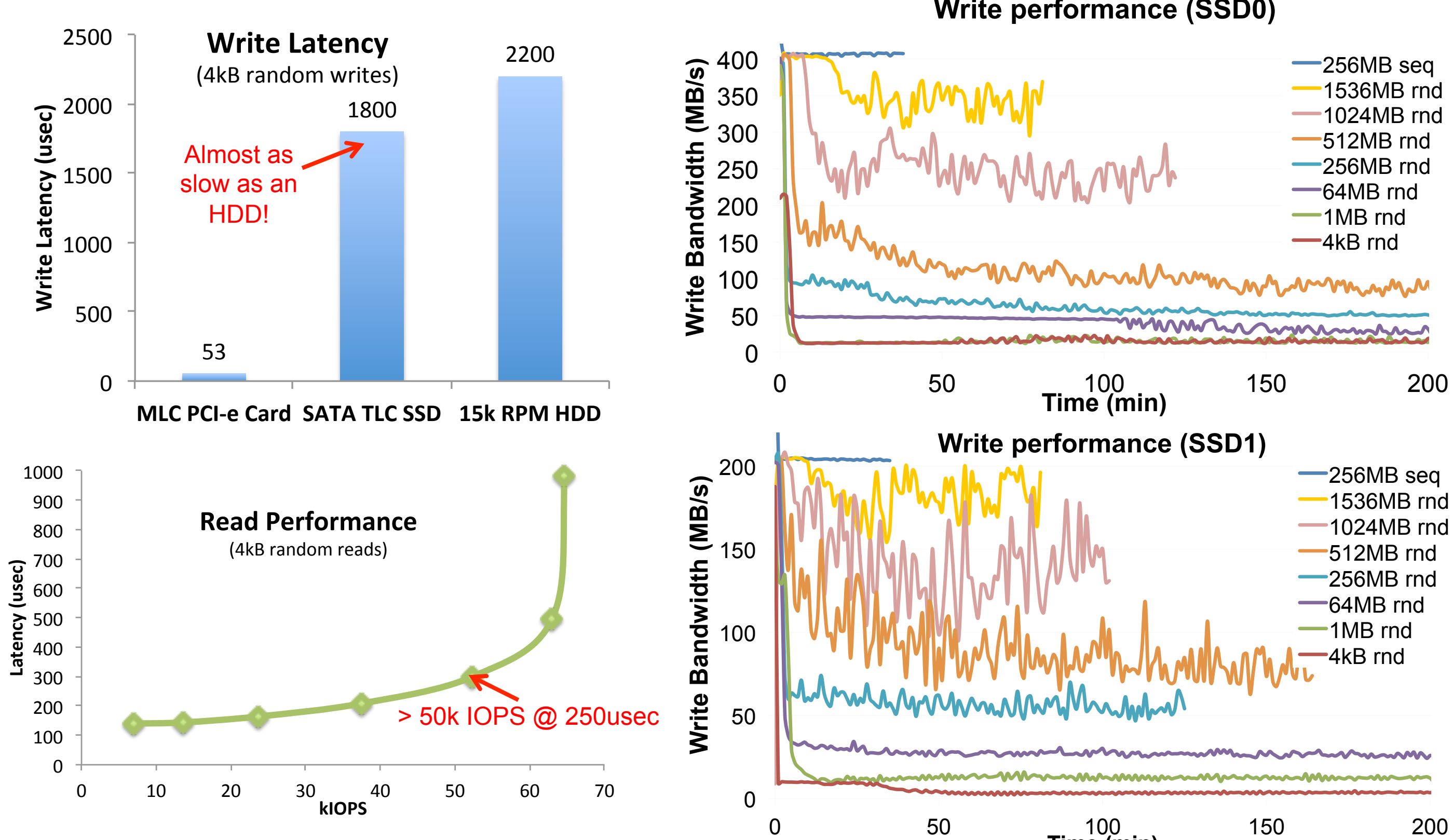**IBM Research - Zurich**

Low-cost Flash in high demand

Biggest Inhibitors for Purchasing All-Flash Systems

Q. *What are the biggest inhibitors for purchasing an all-flash storage system?*

(IDC)

Forbes
Cold Flash And A New Dimension For Flash Memory

The Register
Facebook's request to the flash industry: 'Make the worst flash possible'
Counterintuitive, perhaps – but eminently sensible

eBay: "We could live with 1/3rd the number of writes that normal flash supports as long as we could get it for 1/4th the price."
(thememoryguy.com)

Flash Price Flexibility for Solid State Storage
Q. *At what price per gigabyte would you buy more flash?*

(IDC)

Also according to IDC:

- Over 67% are looking into taking **a self-integrated approach** to deploying Flash in their servers
- Over 77% are looking into or are in the process of adopting flash storage **in the Cloud**
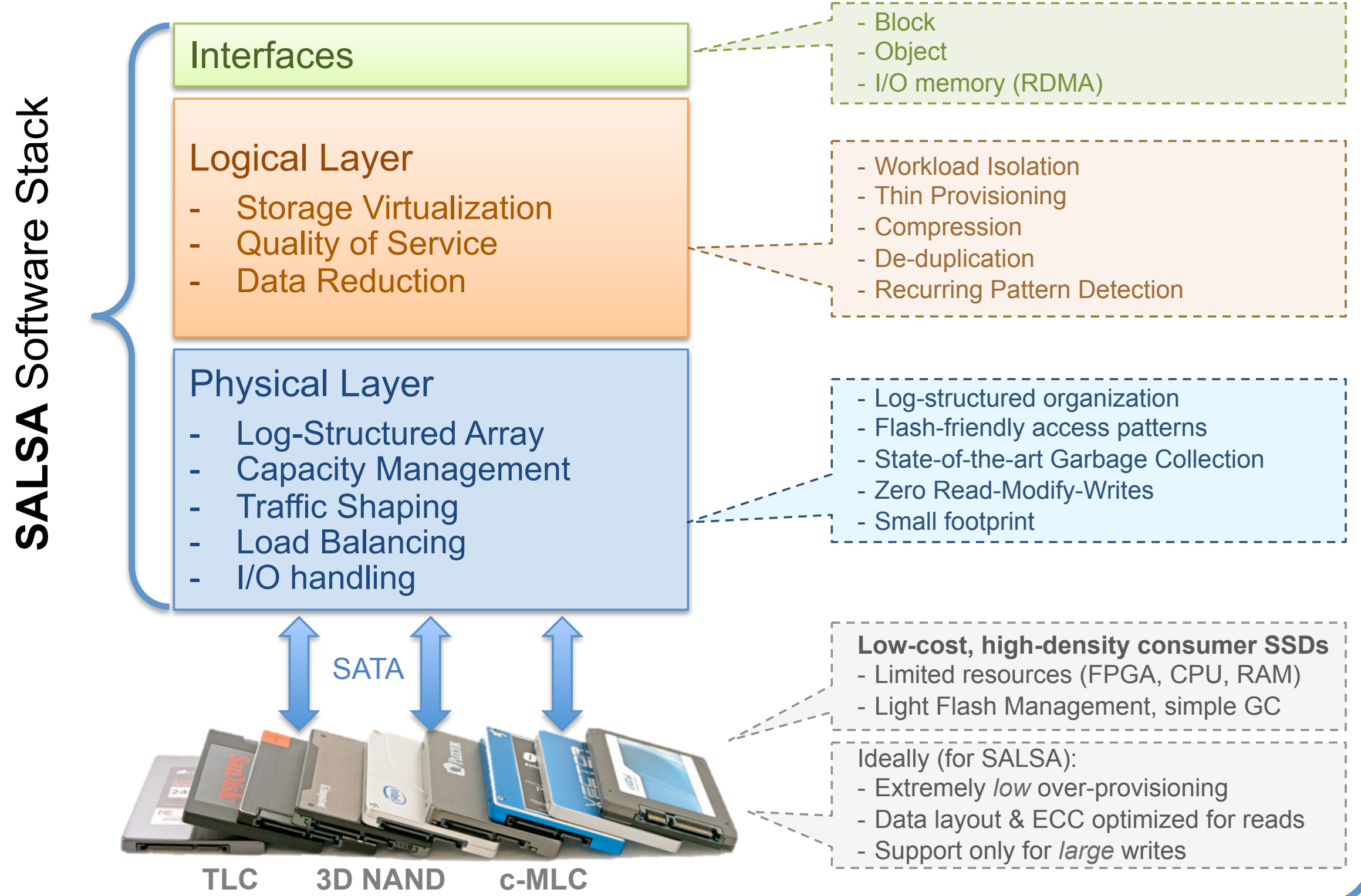
Low-cost SSD performance

Write Latency (4kB random writes)

Almost as slow as an HDD!

Read Performance (4kB random reads)

> 50k IOPS @ 250usec

Write performance (SSD0)

Write performance (SSD1)

- SSD0: Off-the-shelf SATA SSD, TLC Flash, 1TB, 2.5" drive
- SSD1: Off-the-shelf SATA SSD, TLC Flash, 480GB, 2.5" drive
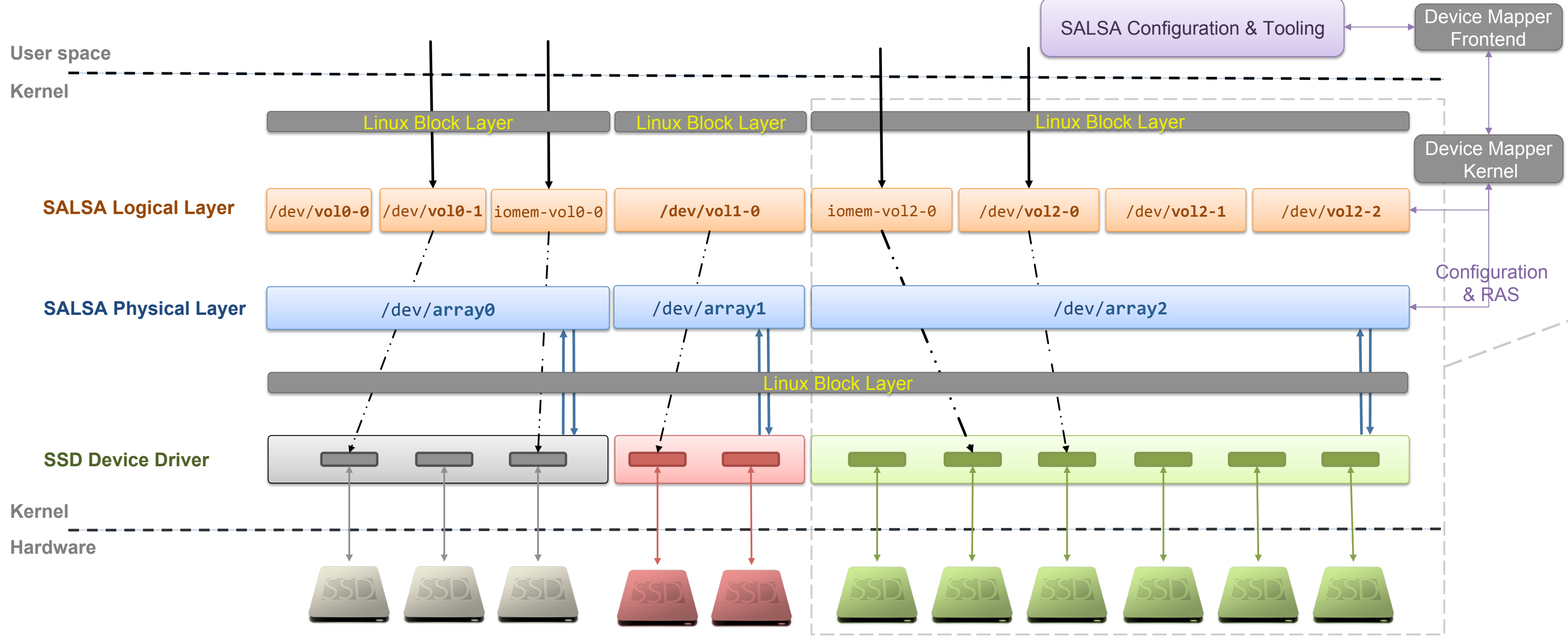- **Price per GB: <$0.5**

Cloud workloads

- Many write-once-read-many workloads
  - Data is often immutable
  - Don't need the write endurance of high-end Flash
  - Examples:
    - Data Warehousing / Analytics
    - Active Archives
    - Social

- Focus on density, cost, and read performance:
  - Read performance: high IOPS & bandwidth, low latency
  - Low cost: consumer-level
  - Enterprise-grade performance and high availability
  - High data ingest rate that is non-disruptive to Reads

- Existing I/O stacks and architectures are not a good fit
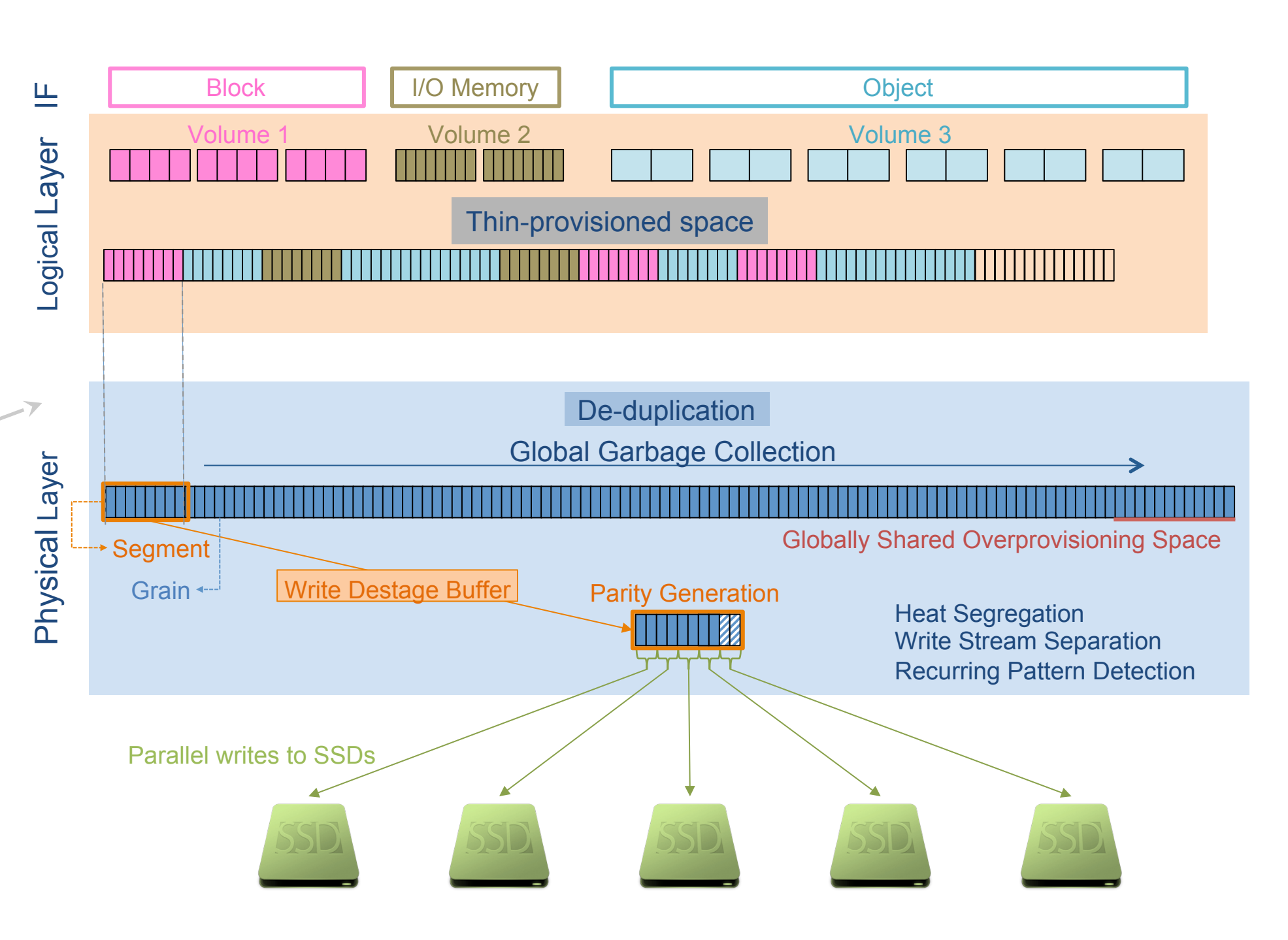 - A Workload-Optimized solution is needed

SoftwAre Log-Structured Array

Key Ideas:

1. Use high-density, low-endurance Flash
2. Move complexity **from hardware to software**
3. Optimize end-to-end for low Write Amplification and data durability
4. Employ data reduction to further reduce cost per GB
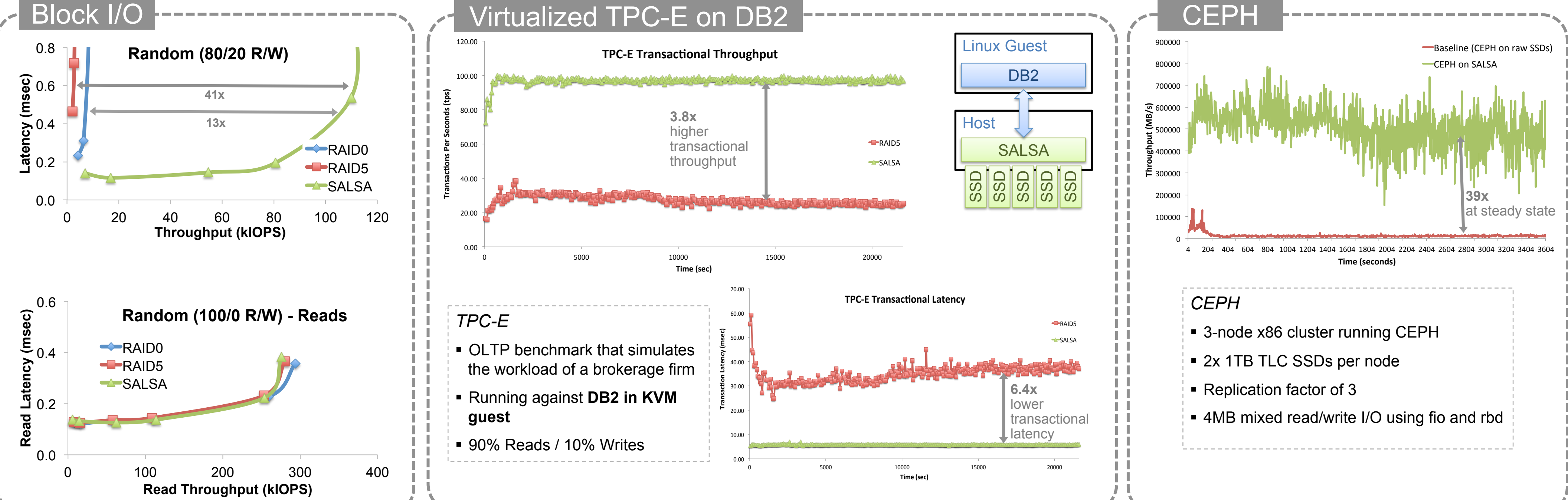5. Natively support Object Storage

Performance
Capacity
Device Lifetime

Write Amplification

SALSA Software Stack

Interfaces
- Block
- Object
- I/O memory (RDMA)

Logical Layer
- Storage Virtualization
- Quality of Service
- Data Reduction
  - Workload Isolation
  - Thin Provisioning
  - Compression
  - De-duplication
  - Recurring Pattern Detection

Physical Layer
- Log-Structured Array
- Capacity Management
- Traffic Shaping
- Load Balancing
- I/O handling
  - Log-structured organization
  - Flash-friendly access patterns
  - State-of-the-art Garbage Collection
  - Zero Read-Modify-Writes
  - Small footprint

SATA

Low-cost, high-density consumer SSDs
- Limited resources (FPGA, CPU, RAM)
- Light Flash Management, simple GC

Ideally (for SALSA):
- Extremely *low* over-provisioning
- Data layout & ECC optimized for reads
- Support only for *large* writes

TLC   3D NAND   c-MLC

SALSA Data Path

SALSA elevates low-cost Flash to meet Cloud storage requirements

SALSA Virtual Array

Preliminary performance evaluation

Block I/O

Random (80/20 R/W)
41x
13x
RAID0
RAID5
SALSA

Random (100/0 R/W) - Reads
RAID0
RAID5
SALSA

Virtualized TPC-E on DB2

TPC-E Transactional Throughput
3.8x higher transactional throughput
RAID5
SALSA

Linux Guest
DB2
Host
SALSA
SSD SSD SSD SSD SSD

TPC-E Transactional Latency
6.4x lower transactional latency

TPC-E
- OLTP benchmark that simulates the workload of a brokerage firm
- Running against **DB2 in KVM guest**
- 90% Reads / 10% Writes

CEPH

Baseline (CEPH on raw SSDs)
CEPH on SALSA
39x at steady state

CEPH
- 3-node x86 cluster running CEPH
- 2x 1TB TLC SSDs per node
- Replication factor of 3
- 4MB mixed read/write I/O using fio and rbd

Conclusions

- Use commodity Flash in the cloud
- Shift complexity from hardware to software
- SALSA: a storage virtualization stack for Flash
  - Workaround FTL unpredictability
  - Elevate the performance of commodity Flash
  - Optimize end-to-end for low Write Amplification
    - Parity protection without the RMW penalty
    - Stream separation and heat segregation
  - Workload consolidation and QOS
- Next steps:
  - Integrate with Open-Channel SSDs
  - Automate profiling and configuration for SSDs