Key Value SSD Explained – Concept, Device, System, and Standard

09/14/2017

Yang Seok Ki
Director of Memory Solutions Lab
Samsung Semiconductor Inc.

Disclaimer

This presentation and/or accompanying oral statements by Samsung representatives collectively, the “Presentation”) is intended to provide information concerning the SSD and memory industry and Samsung Electronics Co., Ltd. and certain affiliates (collectively, “Samsung”). While Samsung strives to provide information that is accurate and up-to-date, this Presentation may nonetheless contain inaccuracies or omissions. As a consequence, Samsung does not in any way guarantee the accuracy or completeness of the information provided in this Presentation.

This Presentation may include forward-looking statements, including, but not limited to, statements about any matter that is not a historical fact; statements regarding Samsung’s intentions, beliefs or current expectations concerning, among other things, market prospects, technological developments, growth, strategies, and the industry in which Samsung operates; and statements regarding products or features that are still in development. By their nature, forward-looking statements involve risks and uncertainties, because they relate to events and depend on circumstances that may or may not occur in the future. Samsung cautions you that forward looking statements are not guarantees of future performance and that the actual developments of Samsung, the market, or industry in which Samsung operates may differ materially from those made or suggested by the forward-looking statements in this Presentation. In addition, even if such forward-looking statements are shown to be accurate, those developments may not be indicative of developments in future periods.

.

Agenda

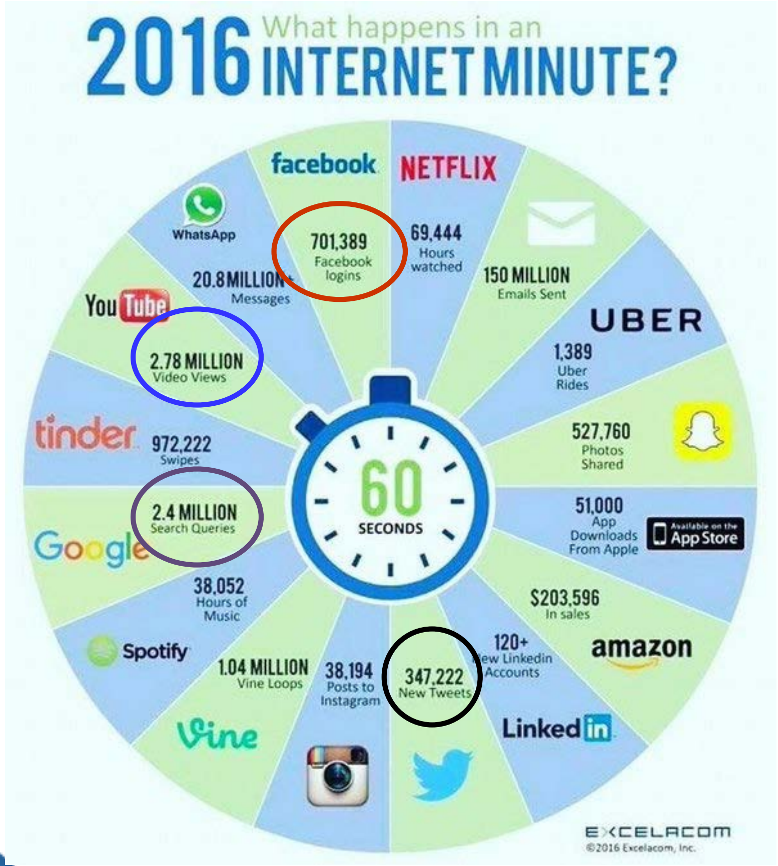
- **Cloud: A New Era**
- **Scalability: A New Challenge**
- **Key Value SSD: A New Technology**
  - Samsung Key Value SSD
- **Ecosystem**
- **Use Case and Performance Studies**
- **Q&A**

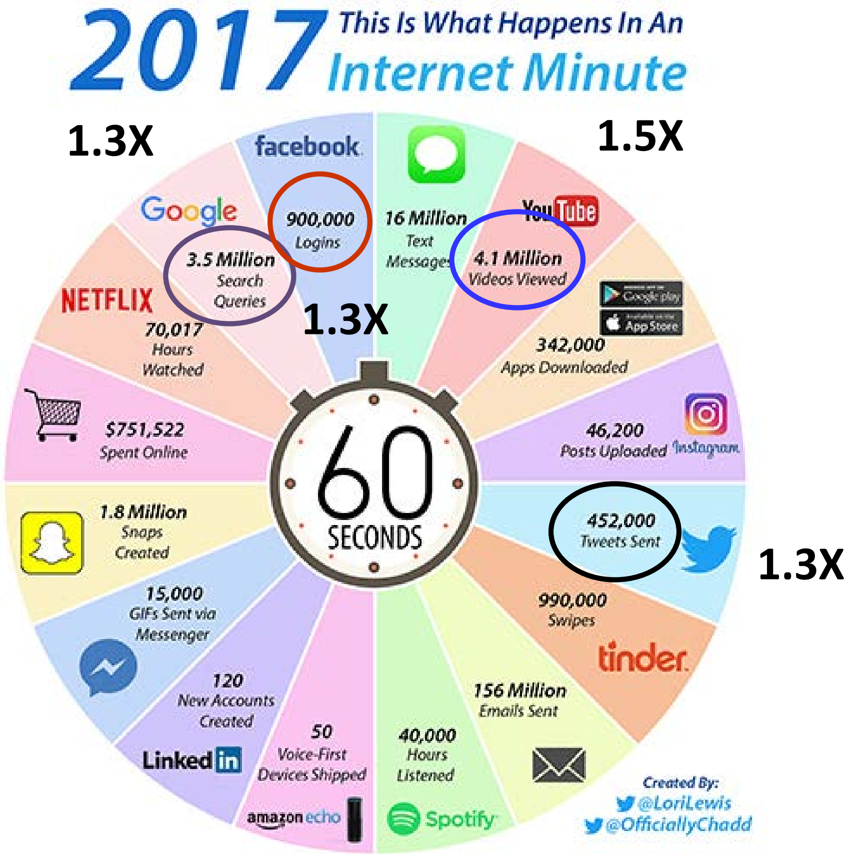
Agenda

- Background
- Concept
- Key Value SSD
- Ecosystem
- Use Case and Performance Studies
- Standards
- Q&A

Cloud: A New Era & Challenges

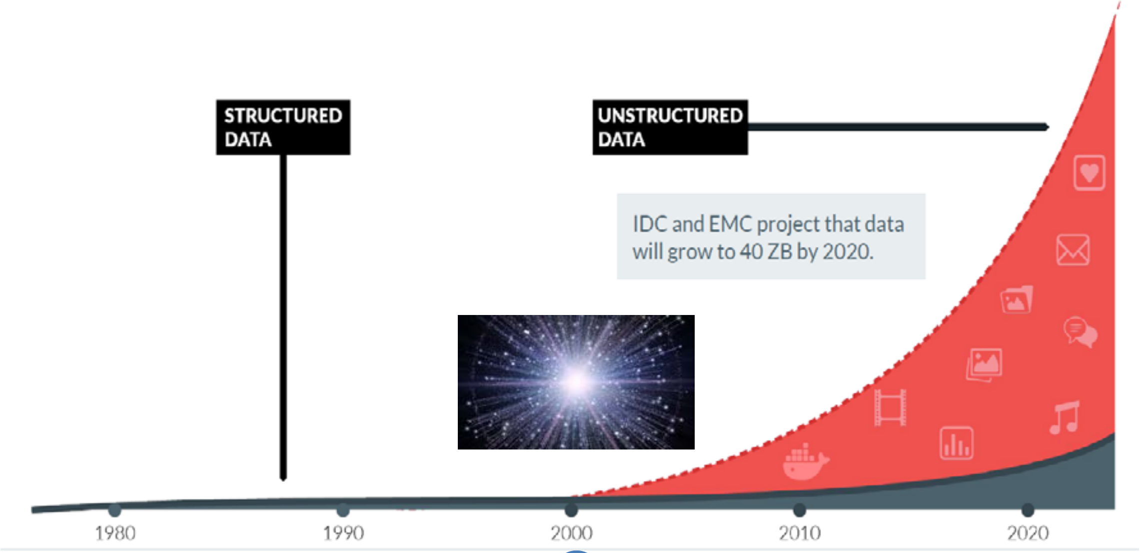
What happens in an internet minute?





BC/AD in IT

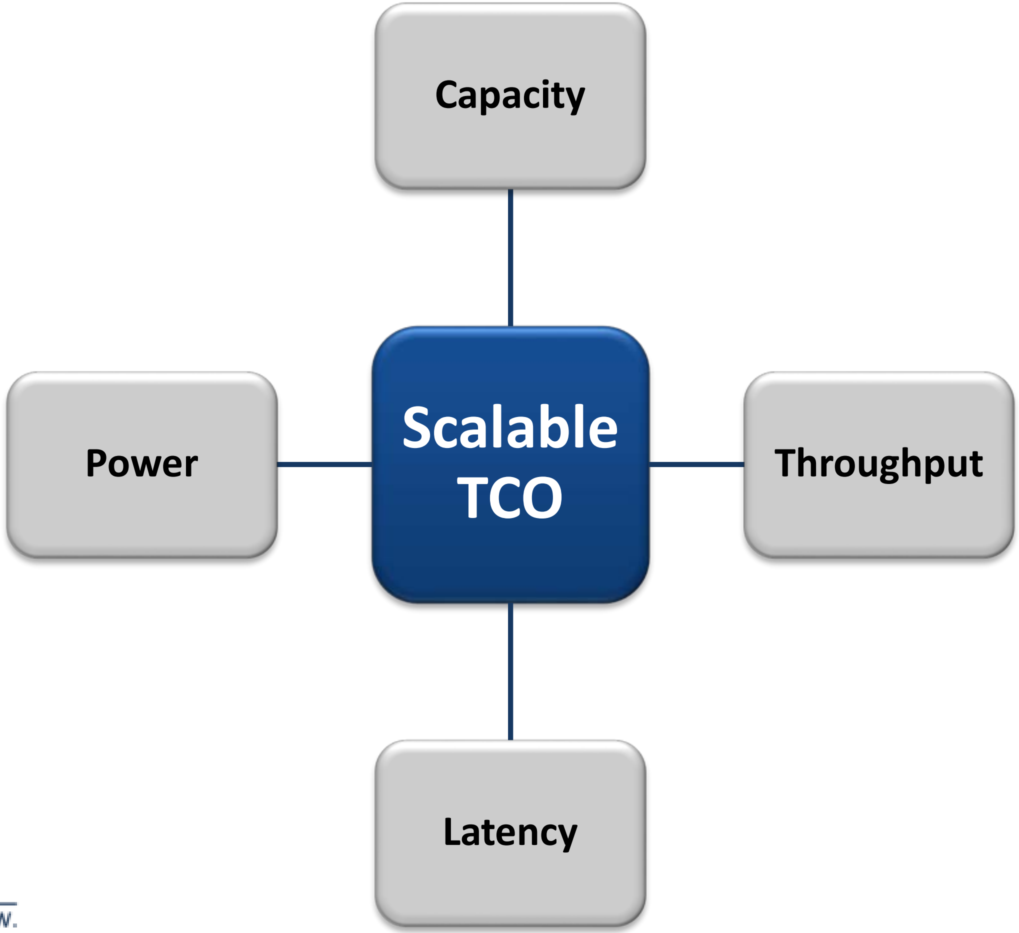
Source: Human Computer Interaction % Knowledge Discovery



Before Cloud

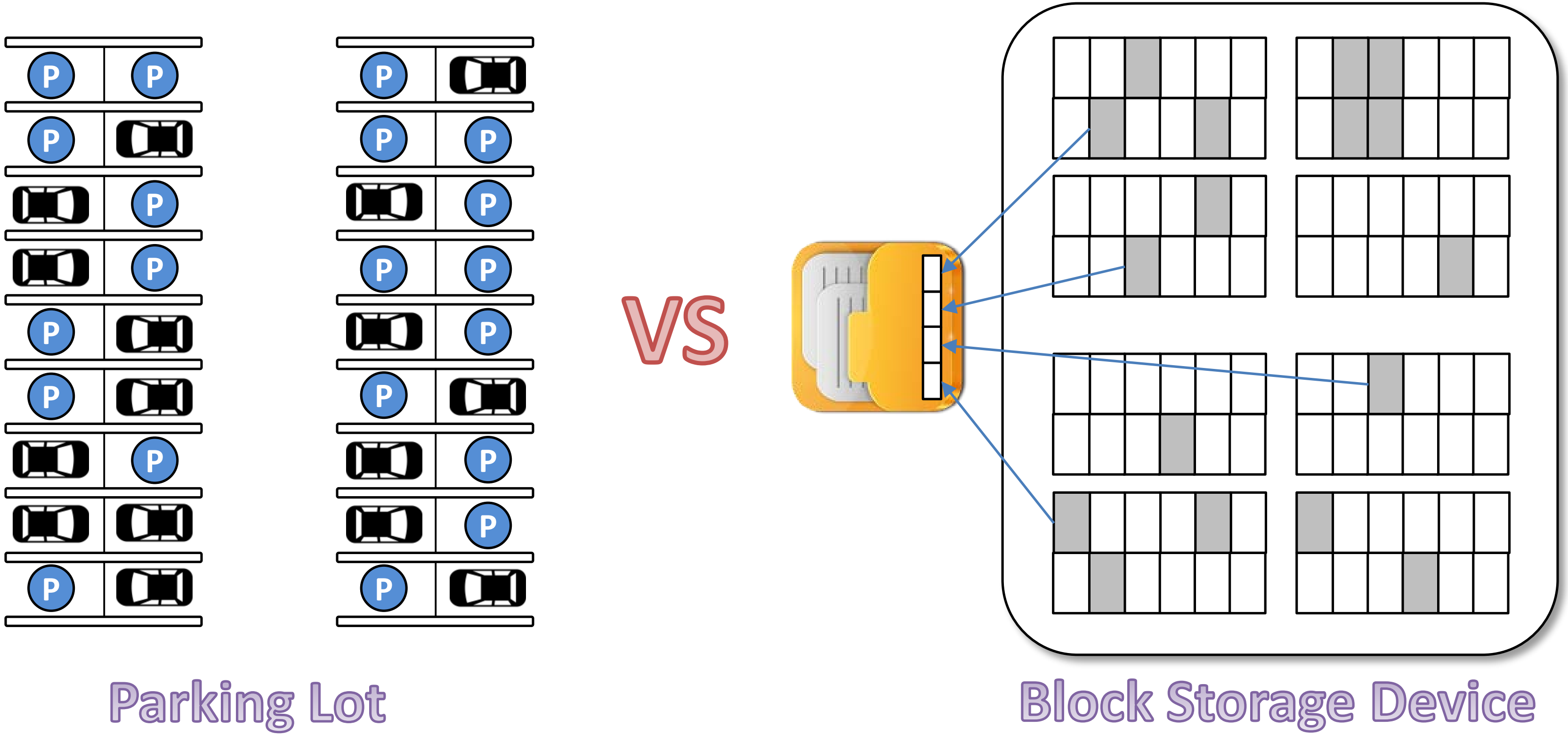
Anno Datum

Challenges in Cloud Era



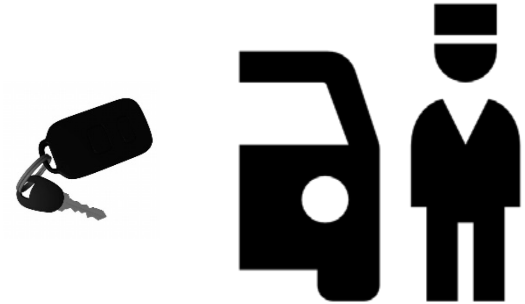
Block: Parking Lot/Structure

- A driver (host) is responsible for parking (data management)

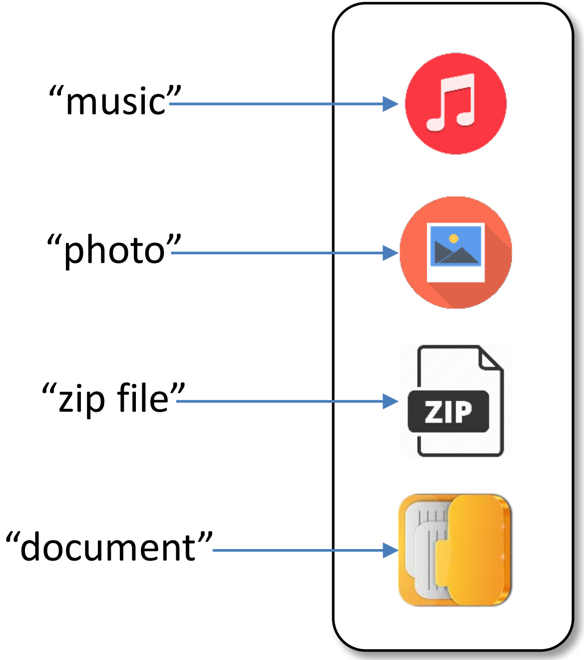


Object: Valet Parking

- **A parking facility (storage) is responsible for parking (data management)**



VS



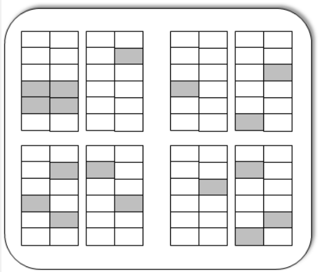
Valet Parking

Object Storage Device

Key Value SSD: New Scalable Technology

Everything is object!

Block



VS

Key Value

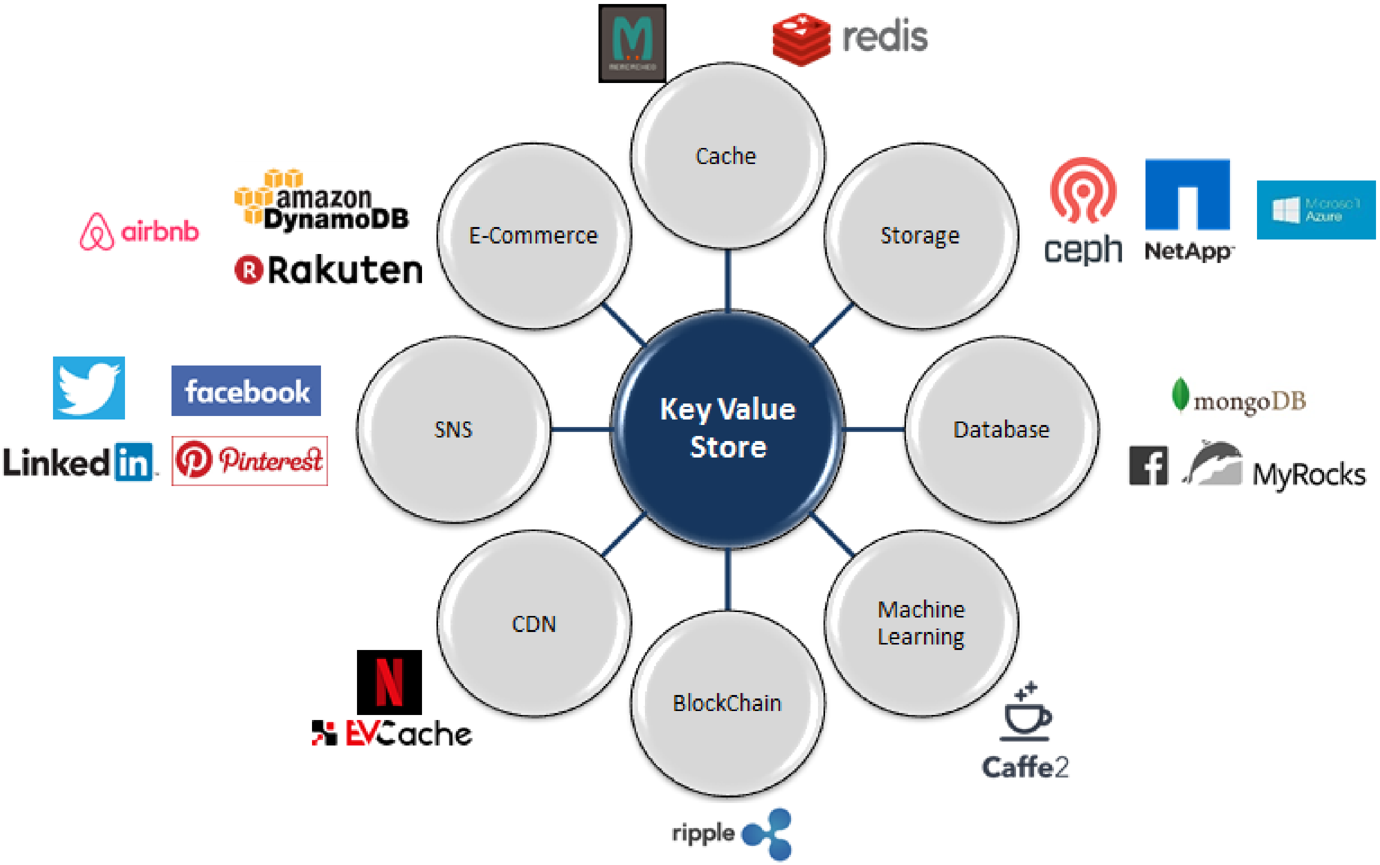
OSD Object Storage

ID	Attributes	User Data

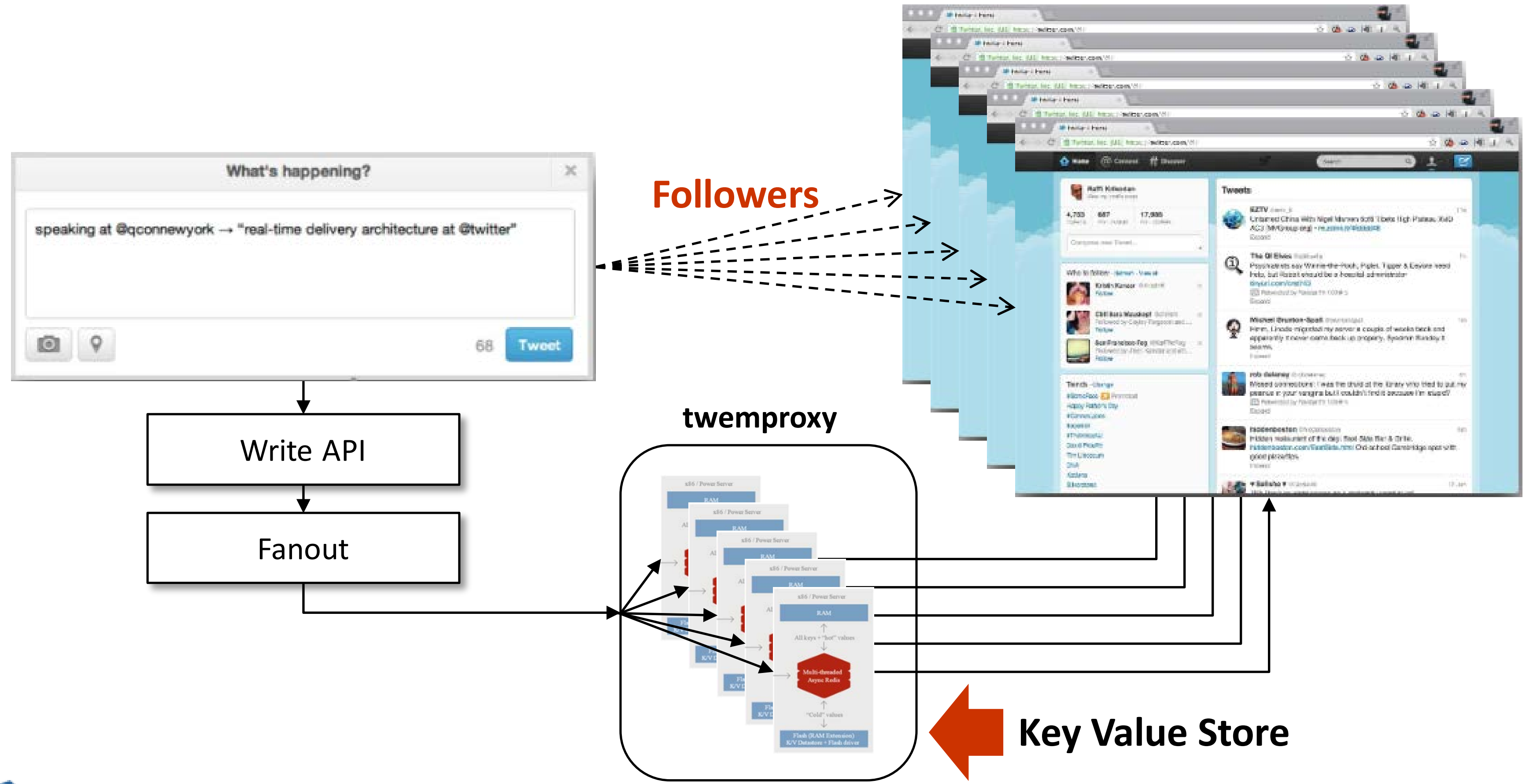
KV Storage

Key	Value

Key Value Stores are Common in Systems at Scale

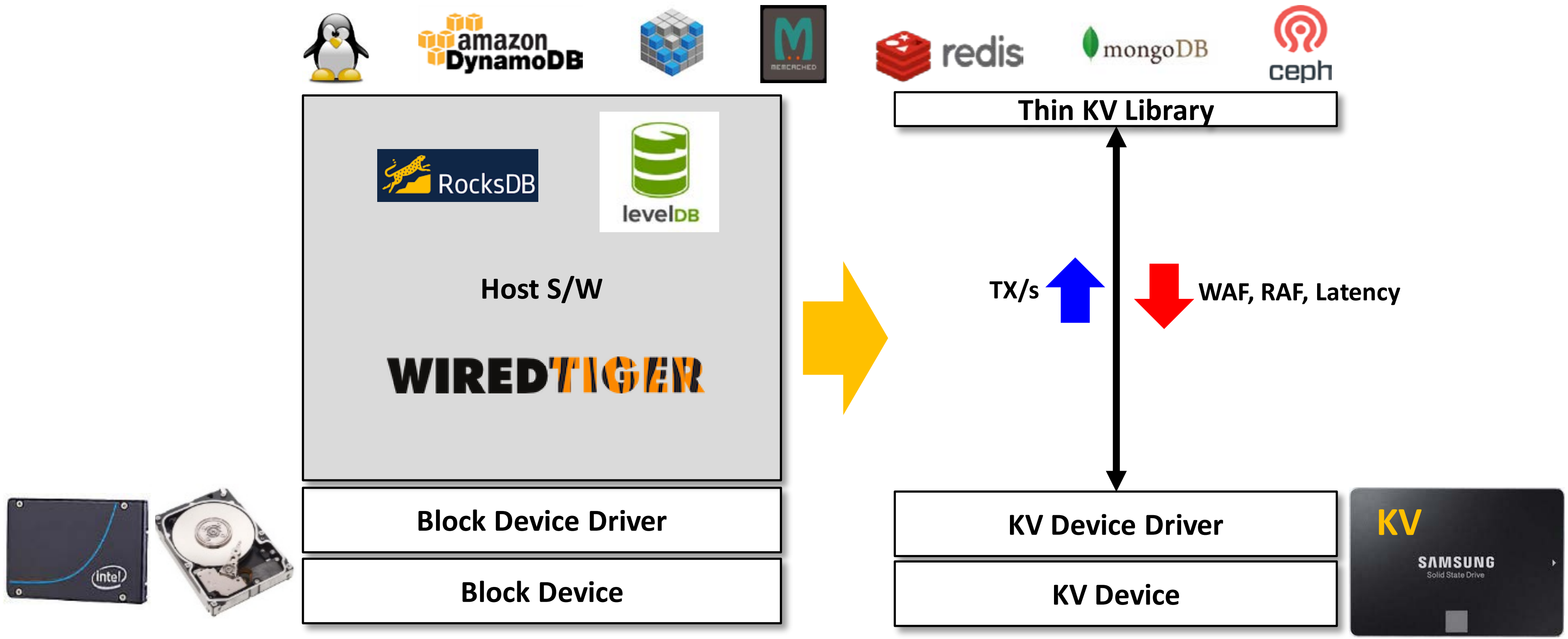


Key Value in Systems at Scale: Twitter Timeline Service



Key Idea

Key Value Store is everywhere!

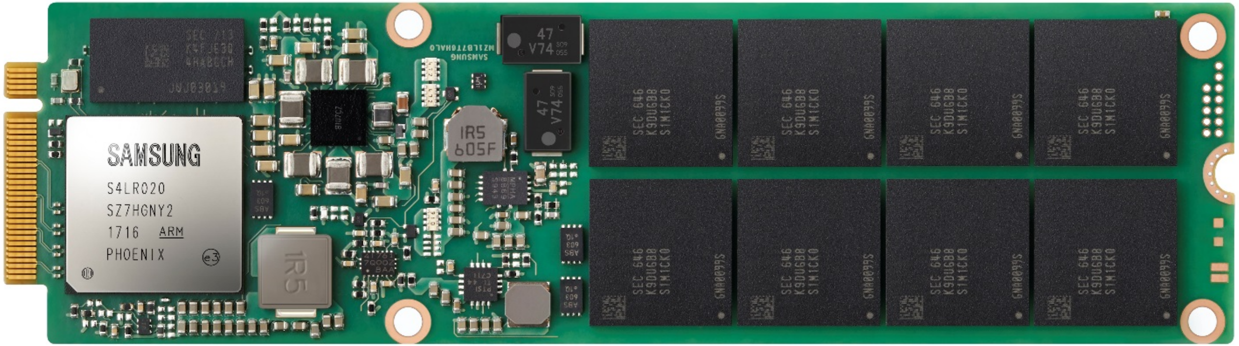


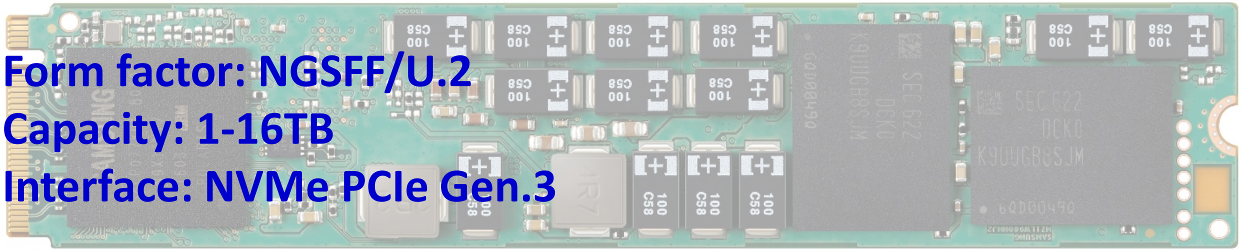
Traditional KV Store

KV Stacks

Samsung KV-PM983 Prototype

NGSFF KV SSD

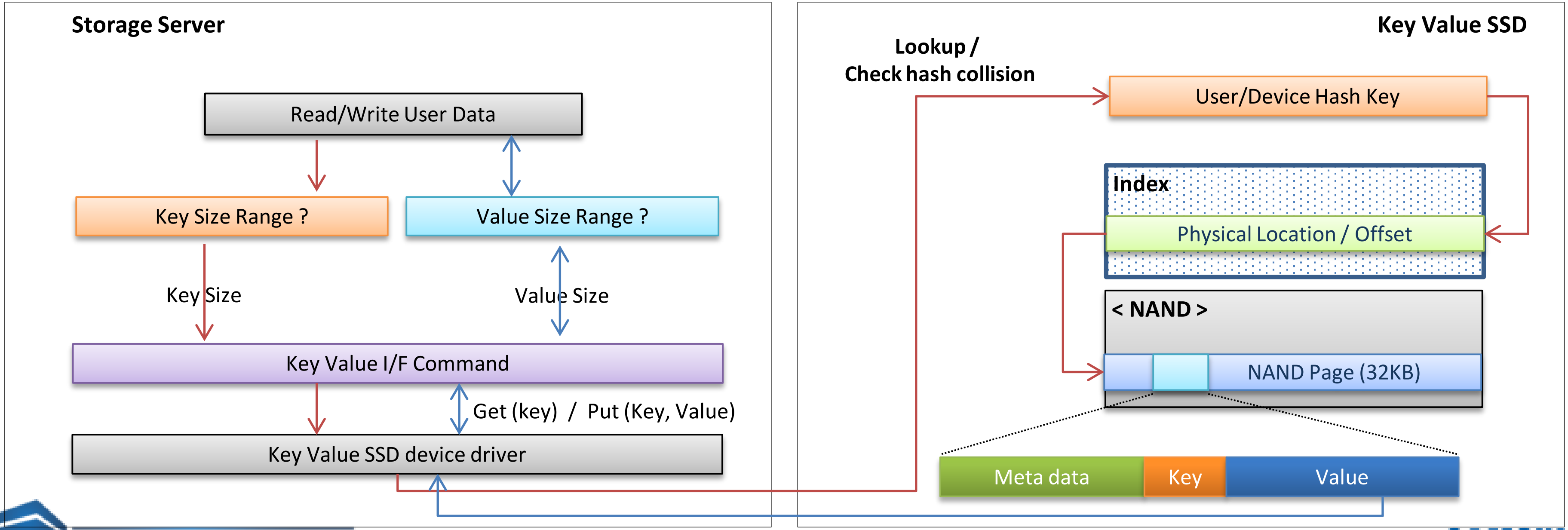




Form factor: NGSFF/U.2
Capacity: 1-16TB
Interface: NVMe PCIe Gen.3

KV SSD Design Overview

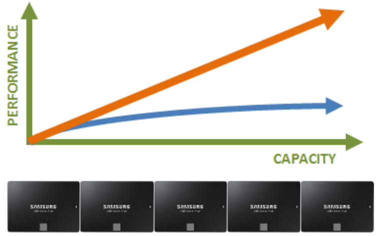
- **Key/Value Range**
  - Key : 4~255B
  - Value : 64B~2GB (32B granularity)
  - The large value is stored into multiple NAND pages



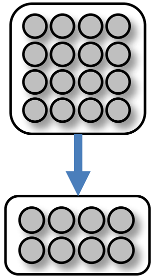
Key Value SSD is a Scalable Solution with Better TCO

KV SSD
- Performance
- Capability

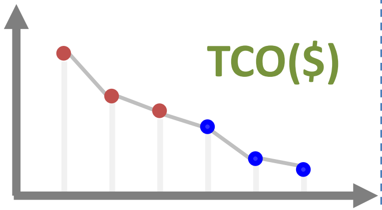
Scale-Up
- Capacity
- Performance



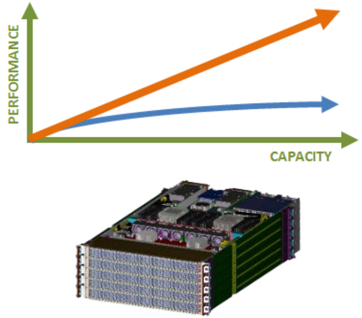
Scale-In
- CPU
- Server



Scale-Down
- TCO
- Power

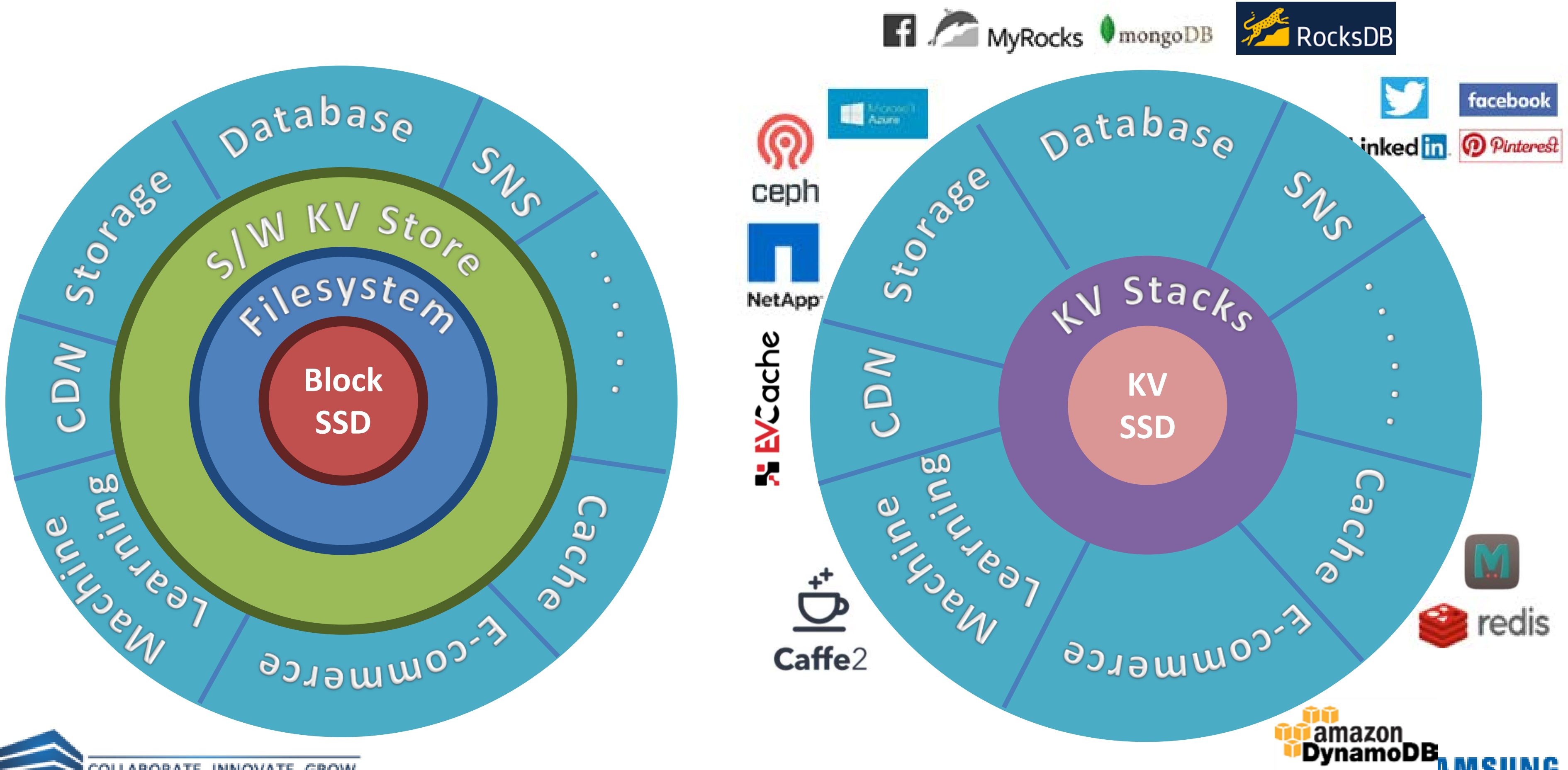


Scale-Out
- Capacity
- Performance

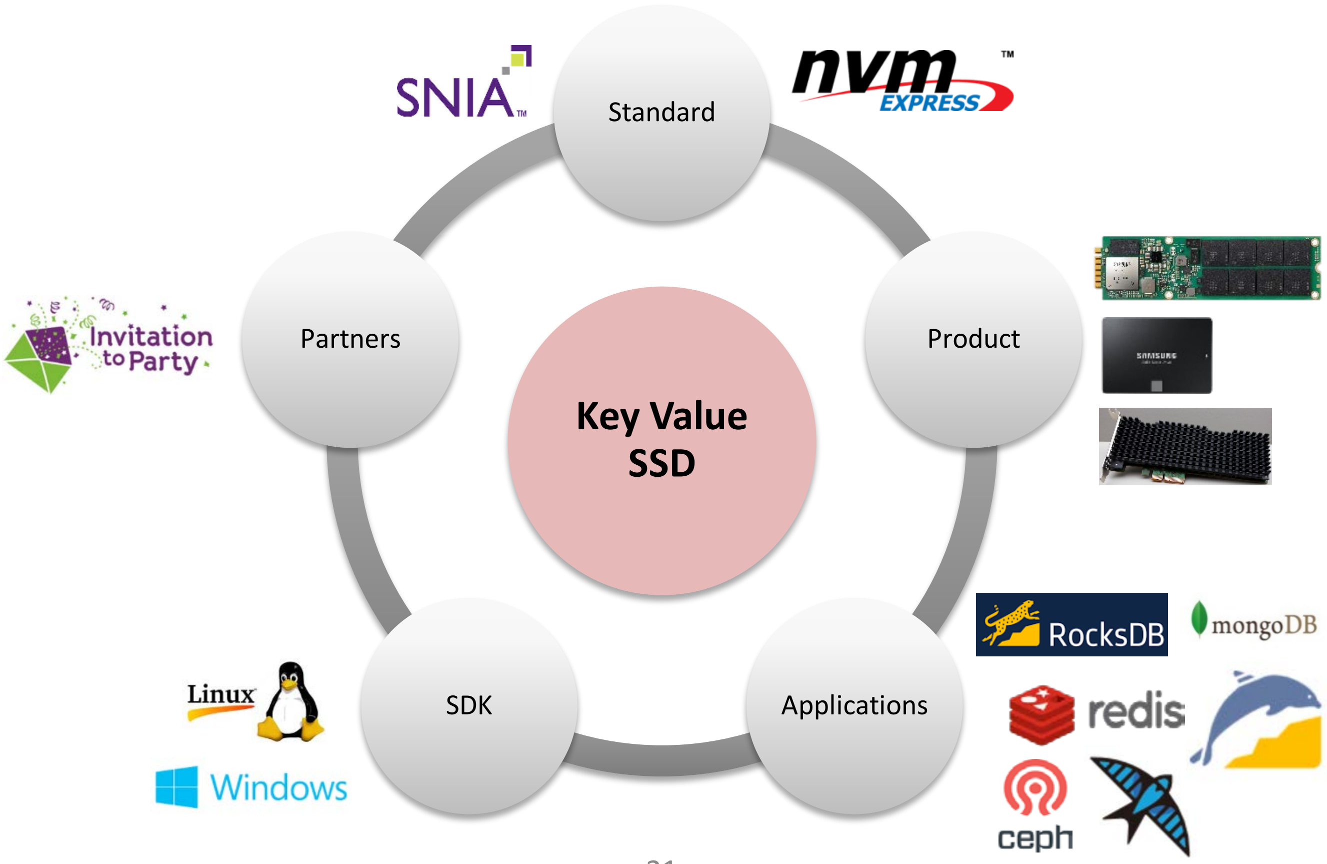


KV SSD Ecosystem

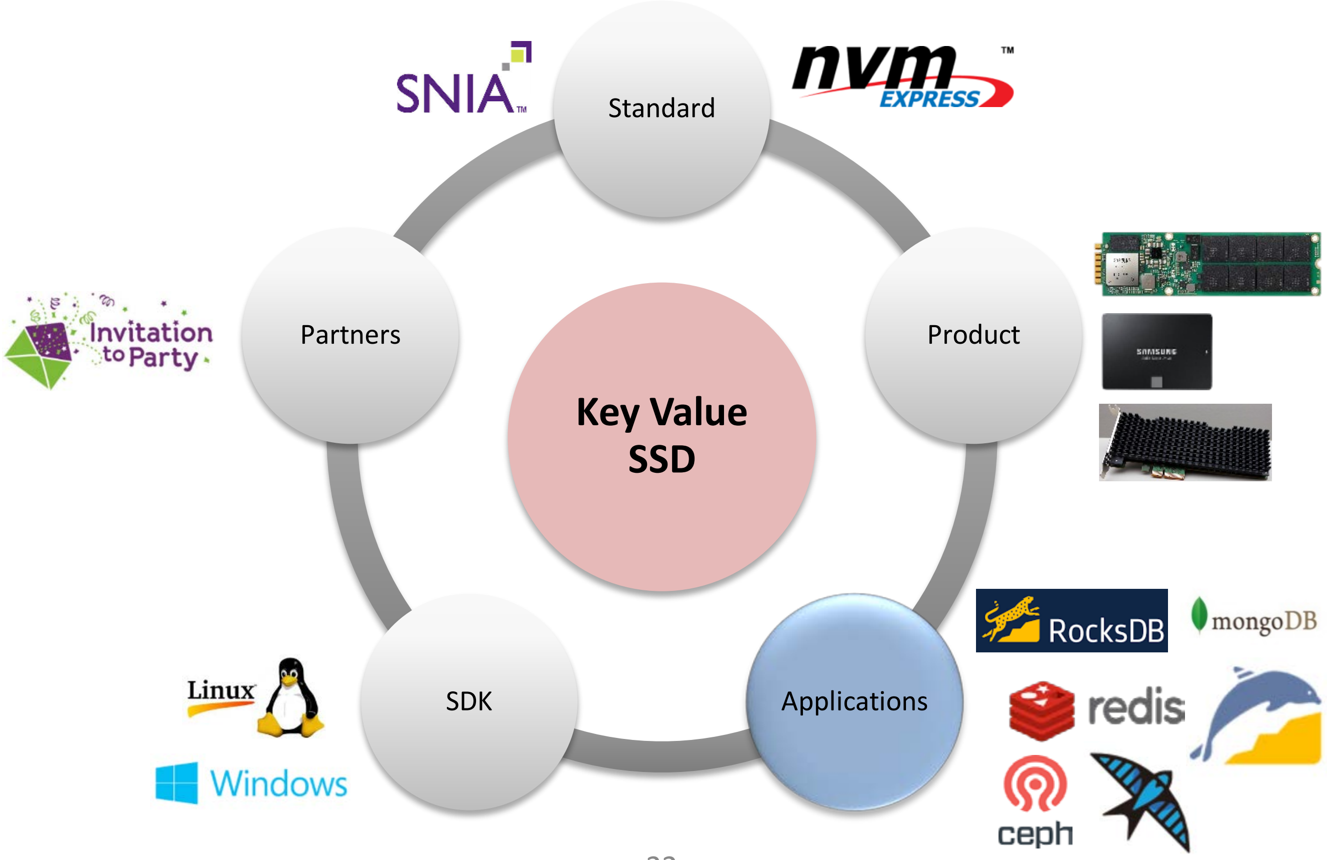
Ecosystem in Block and KV Device Era



KV SSD Ecosystem



KV SSD Ecosystem



Applications for KV SSD

NoSQL DB

KV Adapter

API

KV Stacks

KV Device

Distributed DB

Storage Engine

KV Adapter

API

KV Stacks

KV Device

Object Storage System

Swift API

OSD

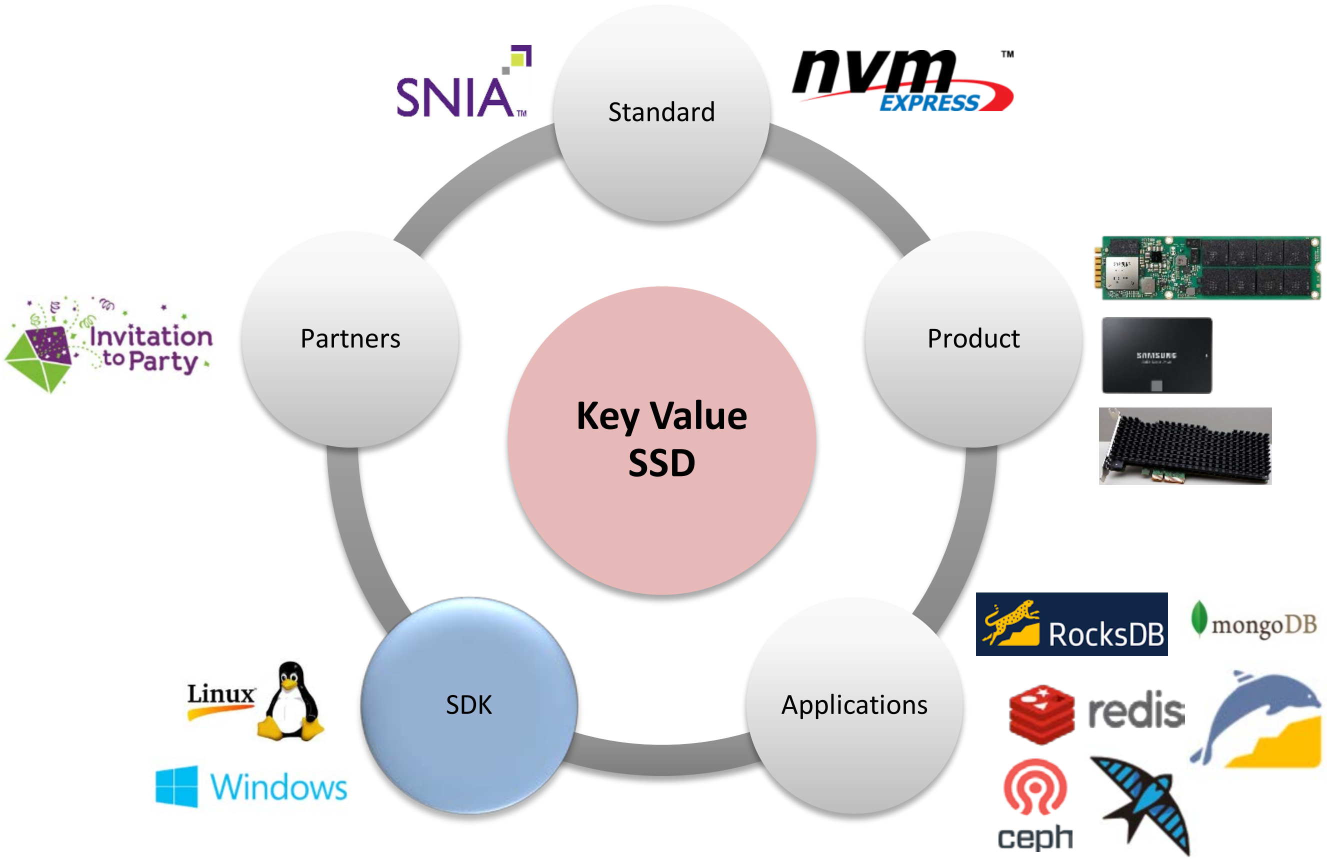
Storage Engine

API

KV Stacks

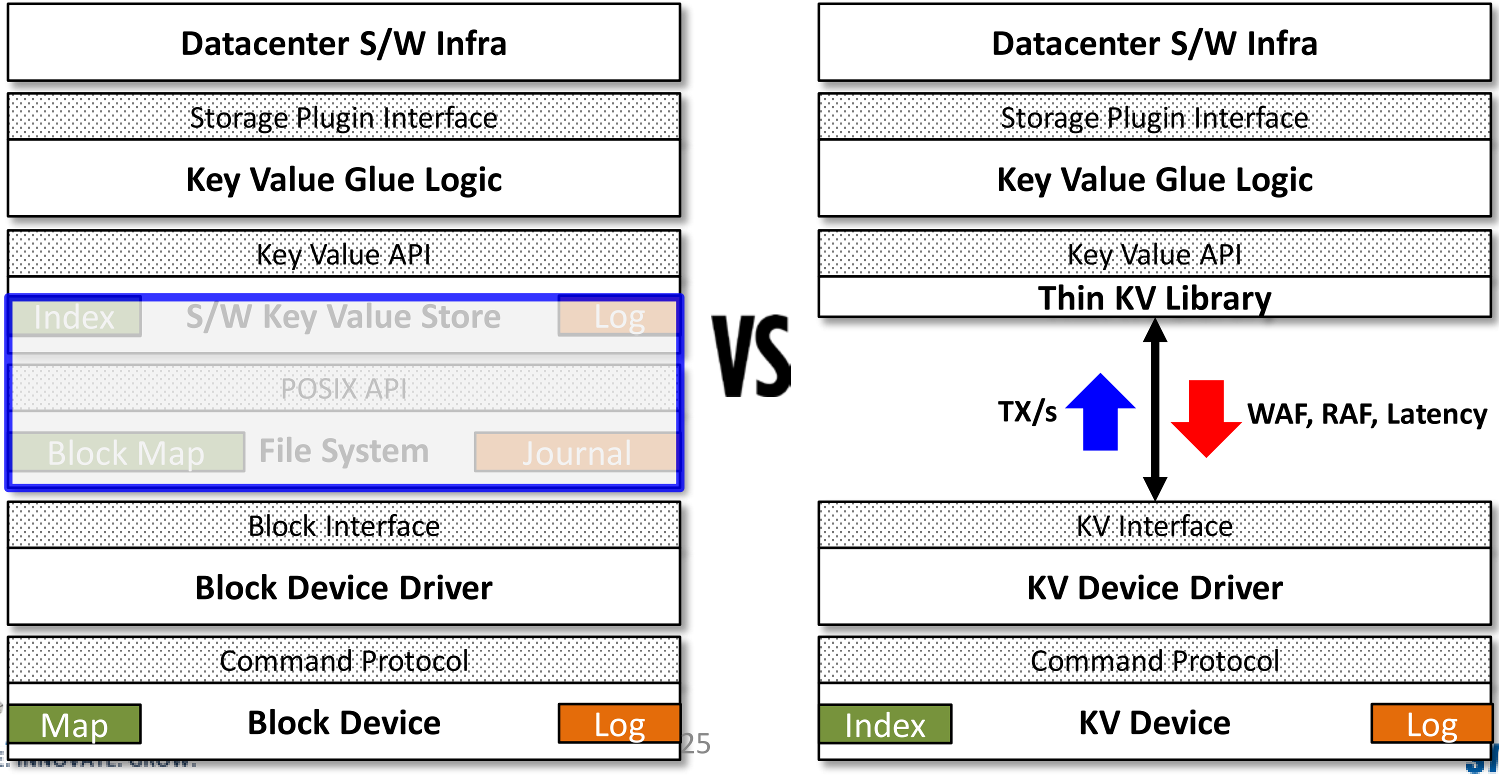
KV Device

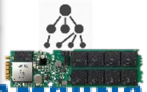
KV SSD Ecosystem



Key Value SW Stacks

- **SSD with native key value interface through hardware software co-design**





Key Value Software Development Stacks

RocksDB redis mongoDB ceph

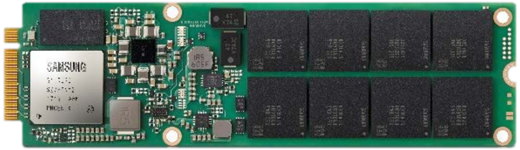
Key Value Library & Tools

Cache	AIO	Multi-Queue	Multi-Device	Memory Manager	Tools

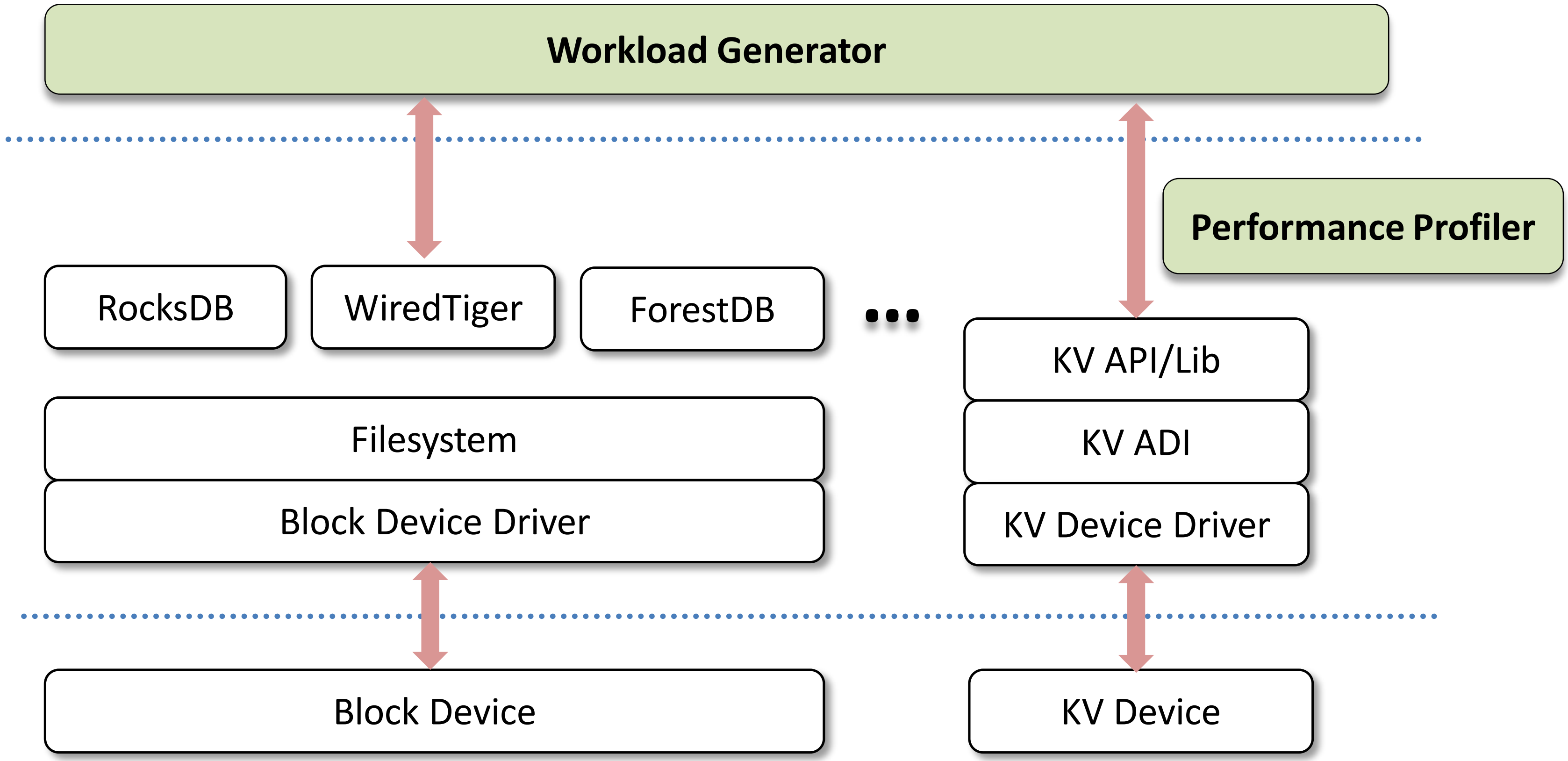
KV Abstract Device Interface (ADI)

store/retrieve/delete/exist	KV Pair	namespace

Linux Kernel Device Driver	Linux User-space Device Driver	Windows Device Driver



kvbench: Key Value Benchmark Suite



KV Virtualization

Application

RocksDB

Spark

redis

ceph

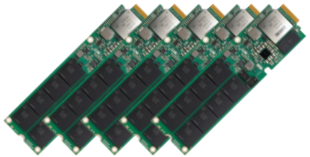
KV Virtualization

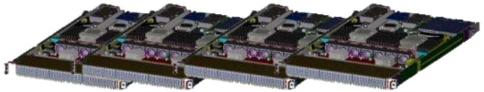
Capacity Management

Key Space Distribution

Load Balancing

KV devices





Key Value Software Development Stacks

RocksDB redis mongoDB ceph

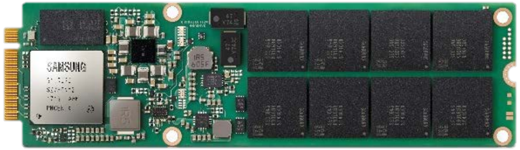
Key Value Library & Tools

- Cache
- AIO
- Multi-Queue
- Multi-Device
- Memory Manager
- Tools

KV Abstract Device Interface (ADI)

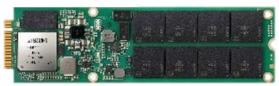
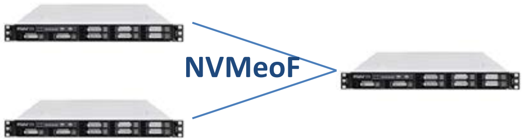
- store/retrieve/delete/exist
- KV Pair
- namespace

Linux Kernel Device Driver	Linux User-space Device Driver	Windows Device Driver

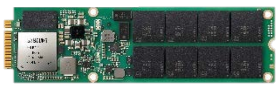
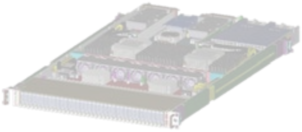
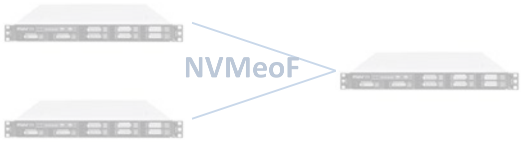


Key Value SSD Use Case Studies

Use Case Study

	Single	Scale-Up	Scale-Out
Benchmark	KVBench		
Key Value Store	RocksDB vs KV Stacks		
Device			

Use Case Study

	Single	Scale-Up	Scale-Out
Benchmark	KVBench		
Key Value Store	RocksDB or KV Stacks		
Device			

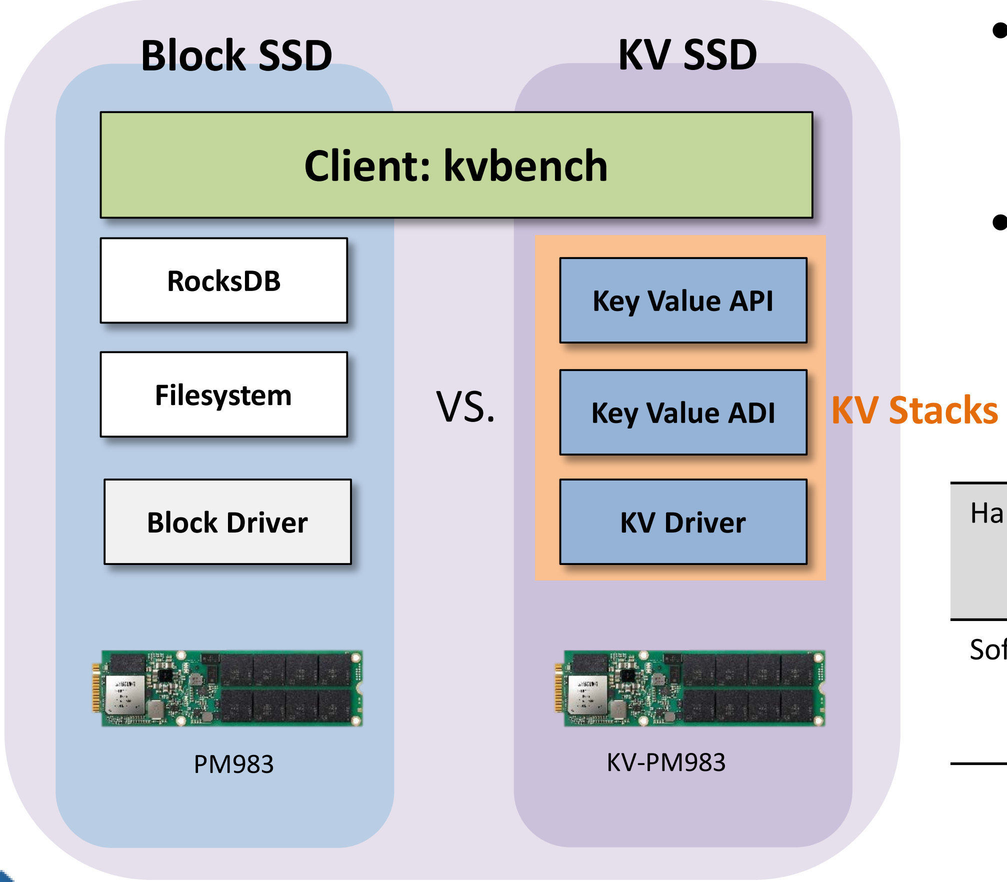
Single Component Performance: RocksDB vs. KV Stacks

- **RocksDB**
  - Originated by Facebook and Actively used in their infrastructure
  - Most popular embedded NoSQL database
  - Persistent Key-Value Store
  - Optimized for fast storage (e.g., SSD)
  - Uses Log Structured Merge Tree architecture

- **KV Stacks on KV SSD**
  - Benchmark tool directly operates on KV SSD through KV Stacks

RocksDB

RocksDB vs. KV Stacks Performance Measurement

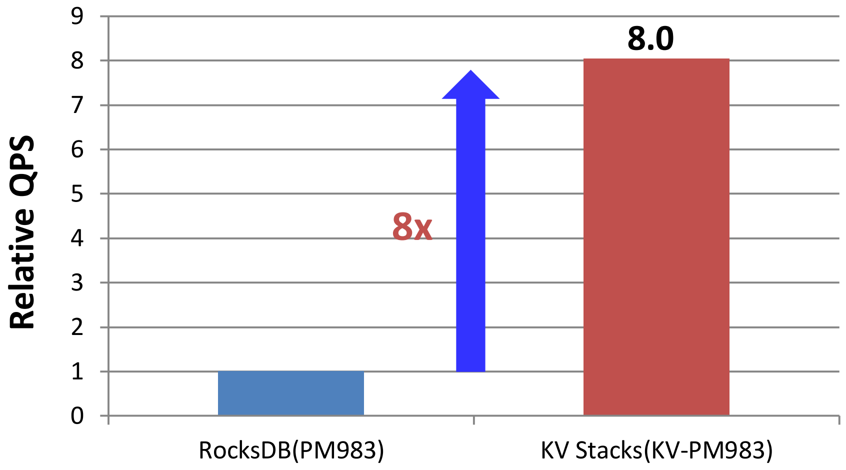


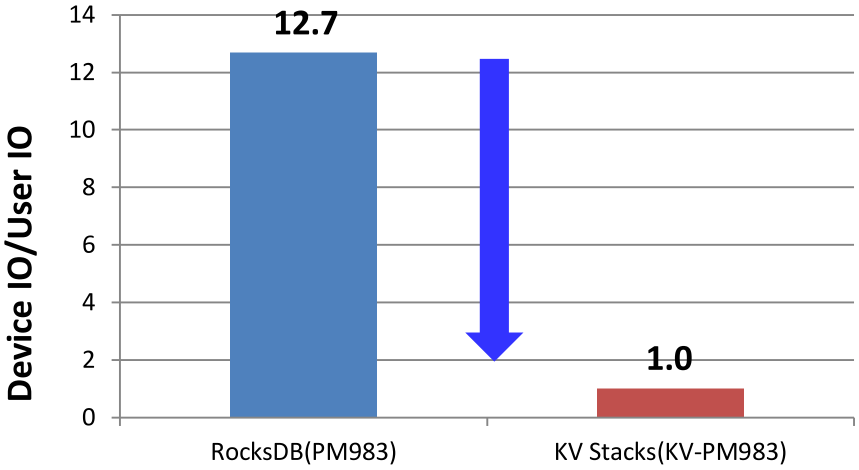
- **Better Performance**
  - Lean software stacks
  - Overhead moved to device
- **IO Efficiency**
  - Reduction of host traffic to devices

Hardware	Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz 96 GB RAM PM983(Block) & KV-PM983 SSD
Software	Ubuntu 16.04 RocksDB v5.0.2 on XFS 50M records, 16B Key, 4KB value

Performance: Random PUT

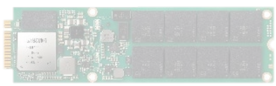
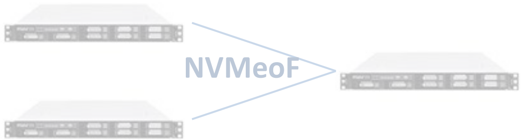
- 8x more QPS (Query Per Second) with KV Stacks than RocksDB on block SSD
- 90+% less traffic goes from host to device with KV SSD than RocksDB on block device



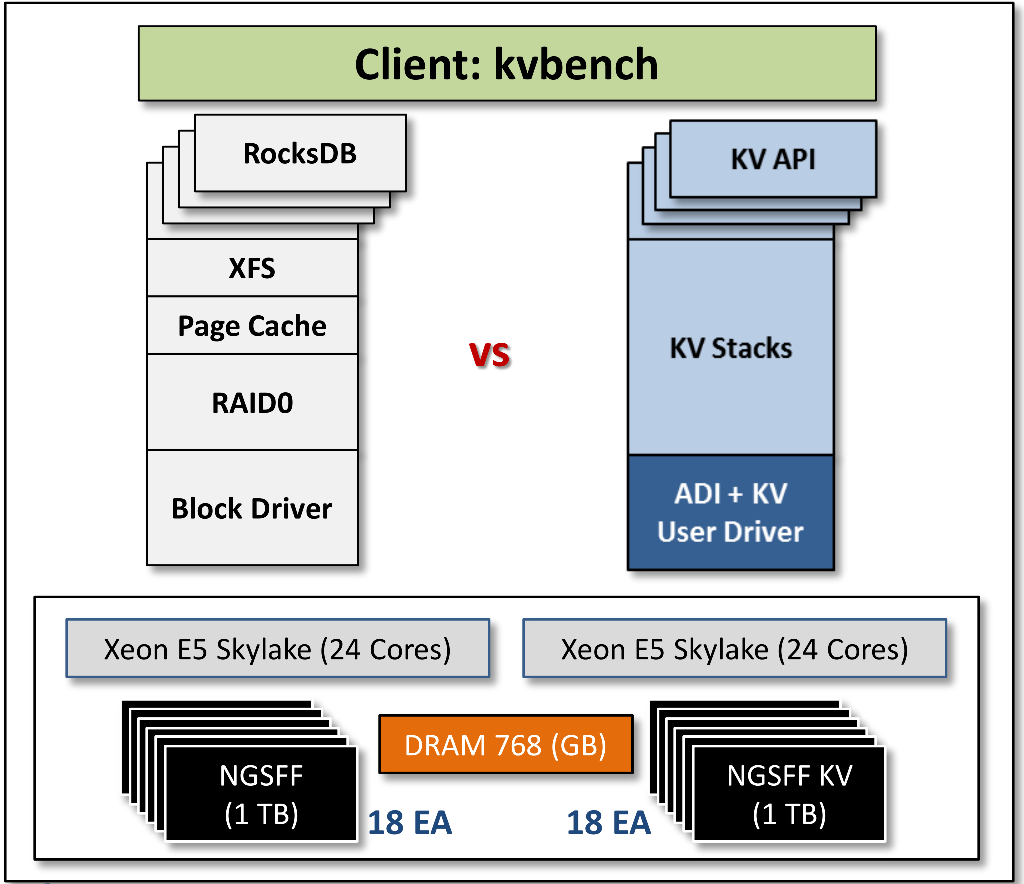


* Workload: 100% random put, 16 byte keys of random uniform distribution, 4KB-fixed values on single PM983 and KV-PM983 in a clean state

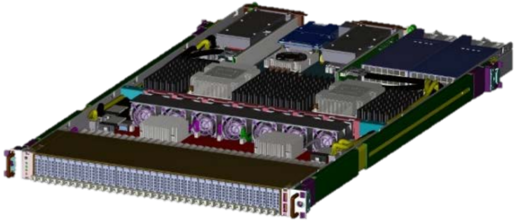
Use Case Study

	Single	**Scale-Up**	Scale-Out
Benchmark		KVBench	
Key Value Store		RocksDB vs KV Stacks	
Device			

Scale-Up Storage: RocksDB

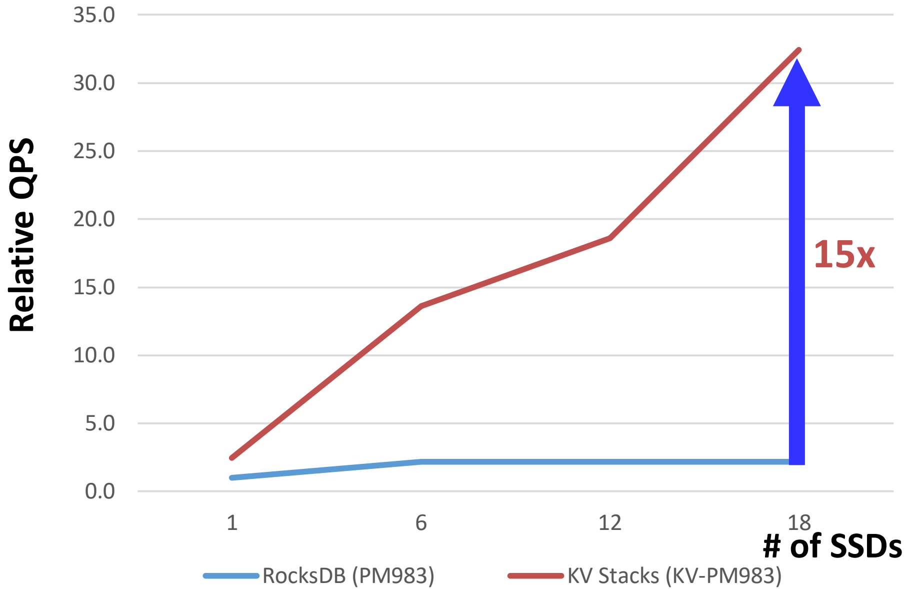


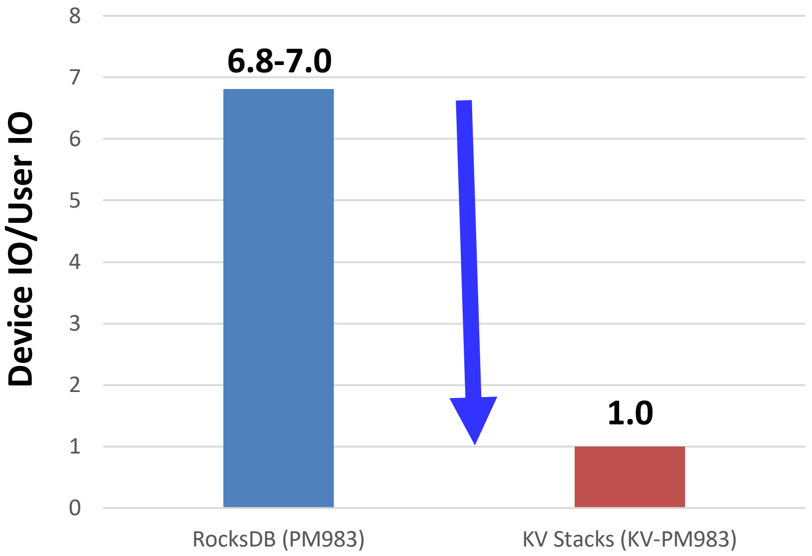
- **Linear Scaling**
  - More devices, more throughput and capacity
- **IO Efficiency**
  - Reduction of host traffics to devices
- **Less CPU utilization**
  - Small number of cores or less CPU utilization for performance



Scale-up Performance: Random Key PUT

- 15x IO performance over S/W key value store on block devices

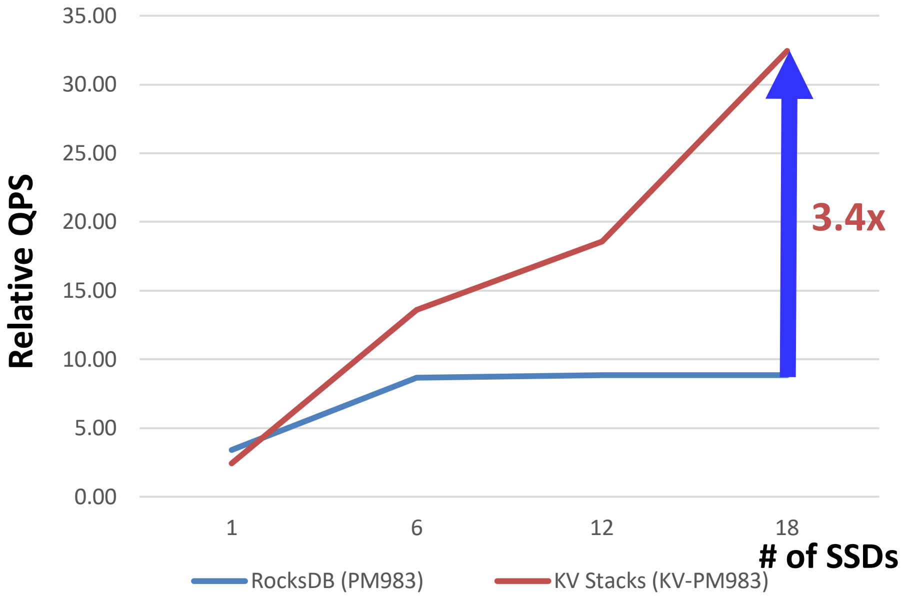


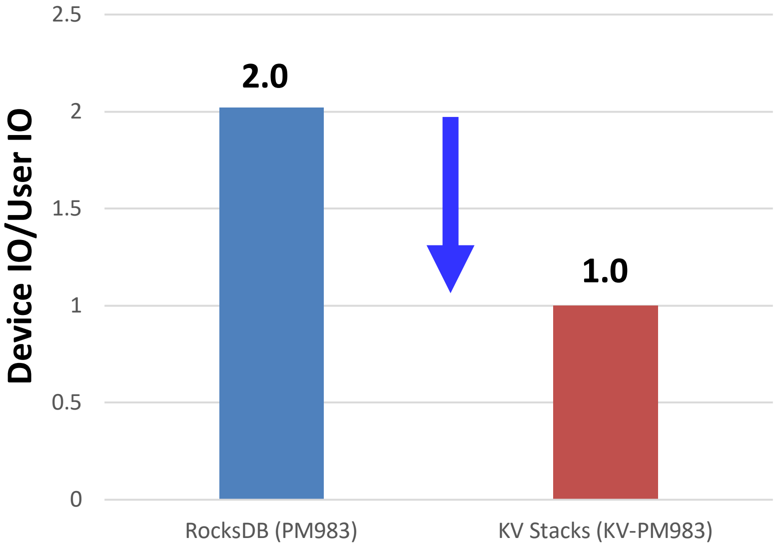


Relative performance to the maximum aggregate RocksDB random Put QPS for 1 SSD with a default configuration for 1 PM983 SSD in a clean state.
System: Ubuntu 16.04.2 LTS, , Ext4, RAID0 for block SSDs, Actual CPU utilization could be 70-90% at CPU saturation point.
Workload: 100% puts, 16 byte keys of random uniform distribution for RocksDB v. 5.0.2, 4KB-fixed values, 24 RocksDB instances with 4 client threads, 50GB/Instance or 1.2TB Data is used

Scale-up Performance: Sequential Key PUT

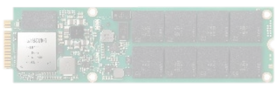
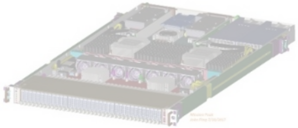
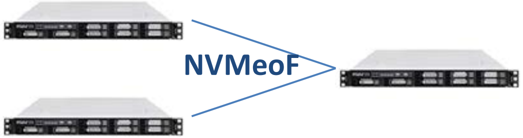
- **3.4x IO performance over S/W key value store on block devices**



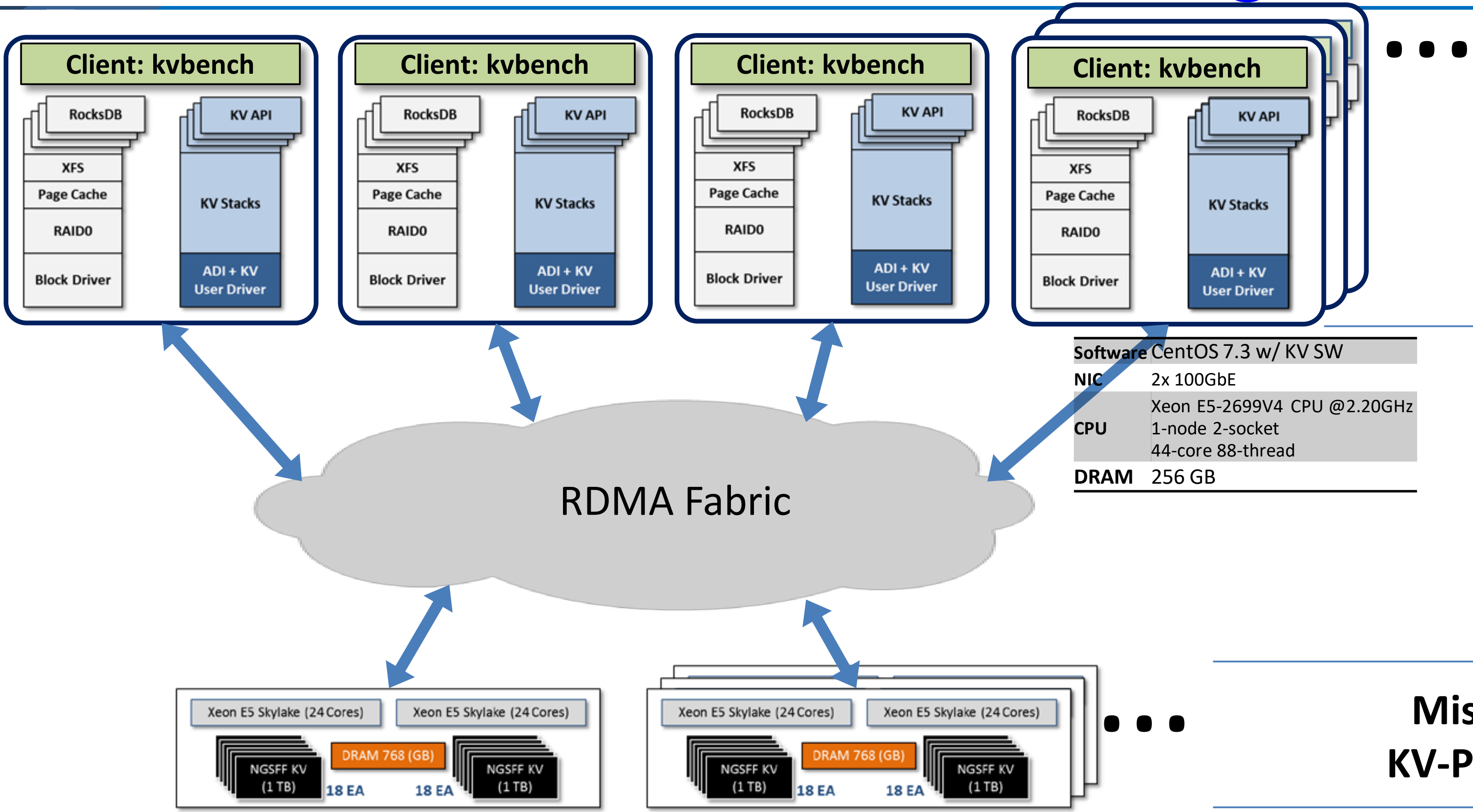


Relative performance to the maximum aggregate RocksDB random Put QPS for 1 SSD with a default configuration for 1 PM983 SSD in a clean state.
System: Ubuntu 16.04.2 LTS, , Ext4, RAID0 for block SSDs, Actual CPU utilization could be 90% at CPU saturation point.
Workload: 100% puts, 16 byte keys of random uniform distribution for RocksDB v. 5.0.2, 4KB-fixed values, 36 RocksDB instances with 1 client thread, 34GB/Instance or 1.2TB Data is used

Use Case Study

	Single	Scale-Up	Scale-Out
Benchmark	KVBench		
Key Value Store	RocksDB vs KV Stacks		
Device			

Scale-Out: RocksDB & KV Stacks Configuration



Software	CentOS 7.3 w/ KV SW
NIC	2x 100GbE
CPU	Xeon E5-2699V4 CPU @2.20GHz 1-node 2-socket 44-core 88-thread
DRAM	256 GB

RocksDB vs KV Stacks

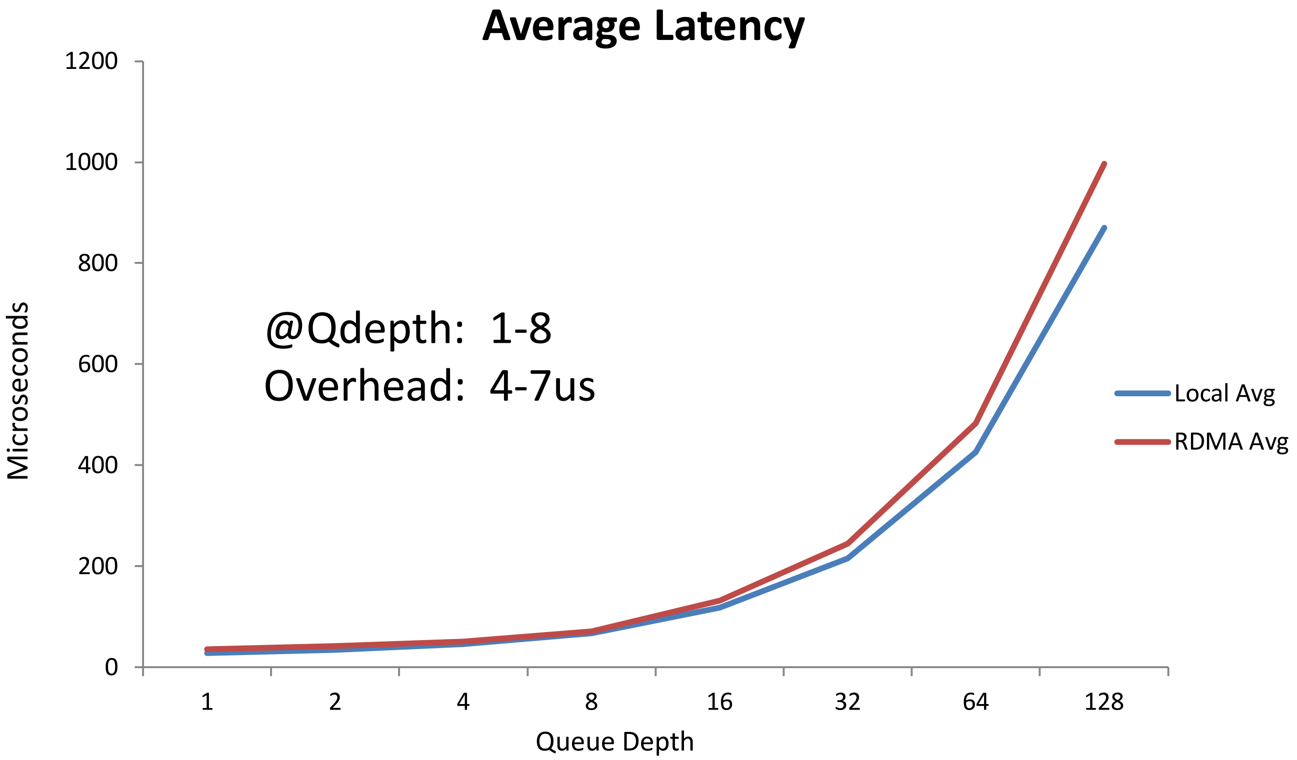
NVMeoF over RDMA

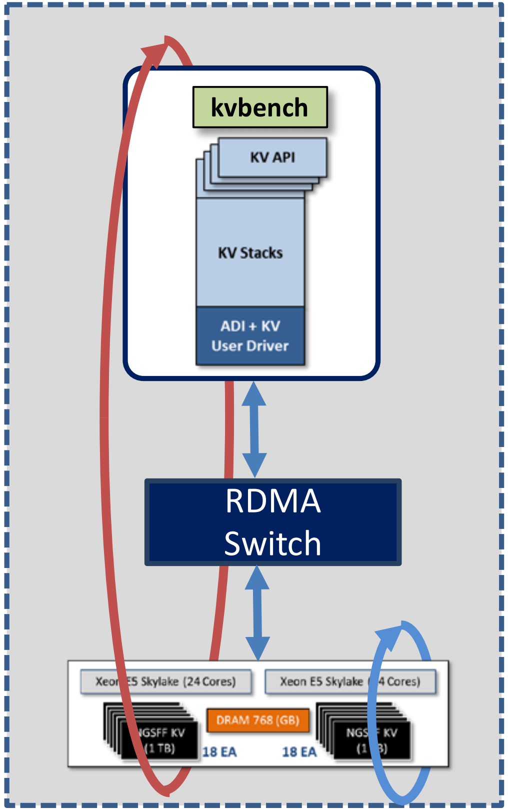
Mission Peak KV-PM983 SSDs

CPU	Xeon 8160 CPU @2.10GHz 1-node 2-socket, 48-core 96-thread
# SSDs	36x 1TB

Software	CentOS 7.3 w/ KV Target
NICs	2x 100GbE + 2x 50GbE

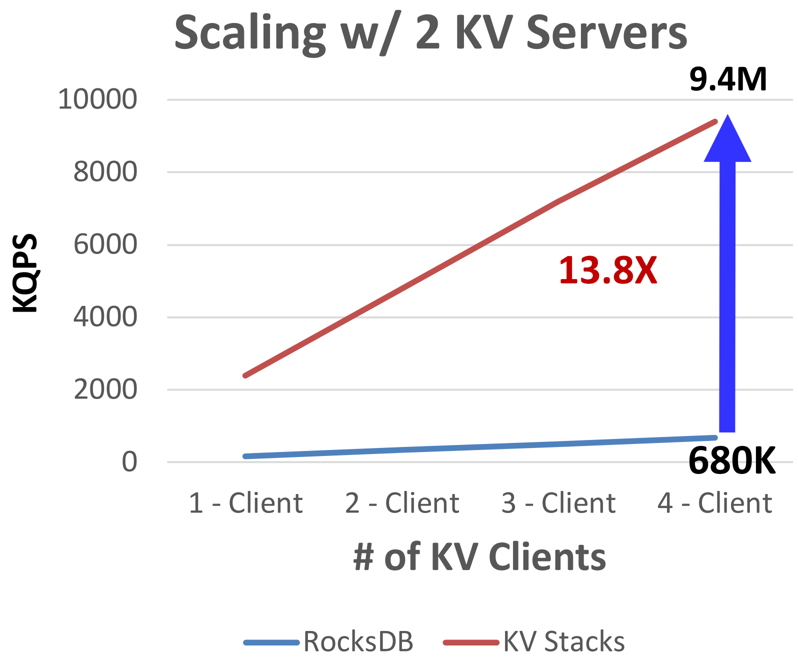
Local vs NVMeoF PUT Latency



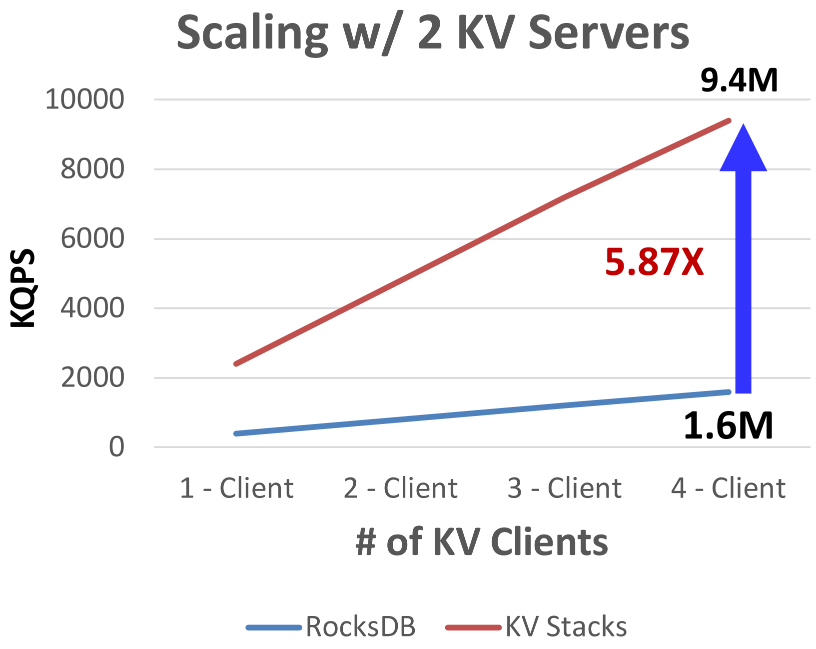


Performance and Capacity Scale-Out: PUT Throughput

Fill Random



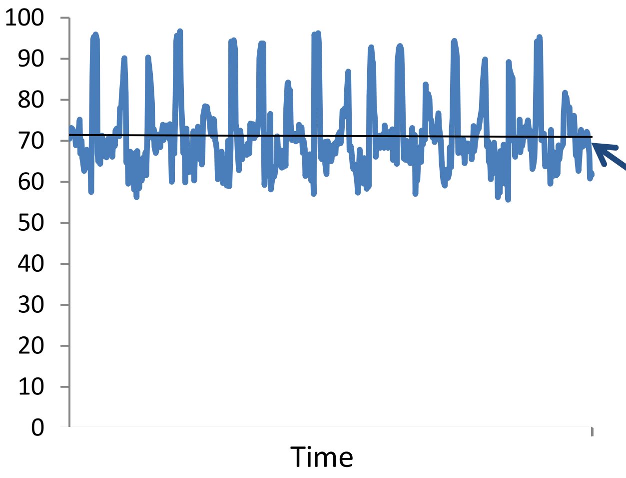
Fill Sequential



Client RocksDB: CentOS 7.3, Ext4, RAID0 for block SSDs,
Workload: 100% puts, 16 byte keys of random uniform distribution for RocksDB, 4KB-fixed values, 24 RocksDB instances with 8 client threads, 50GB/Instance or 1.2TB Data is used,
Client KV Stacks: CentOS 7.3, KV Load Generator, 100% 4K PUTs, 16 byte keys,
KV Server: Mission Peak w/ NVMeoF KV Target

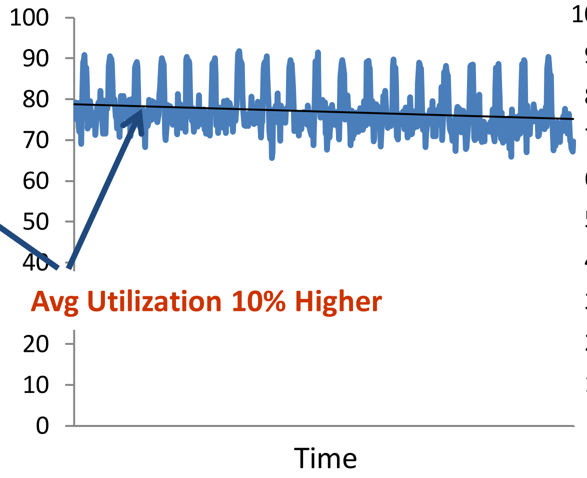
CPU Utilization for Clients

Fill Random



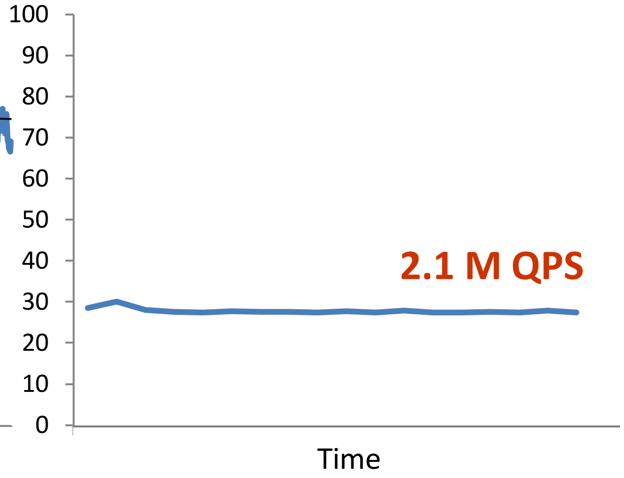
Avg 170K QPS@72% CPU

Fill Sequential



Avg 400K QPS@80% CPU

KV Stacks

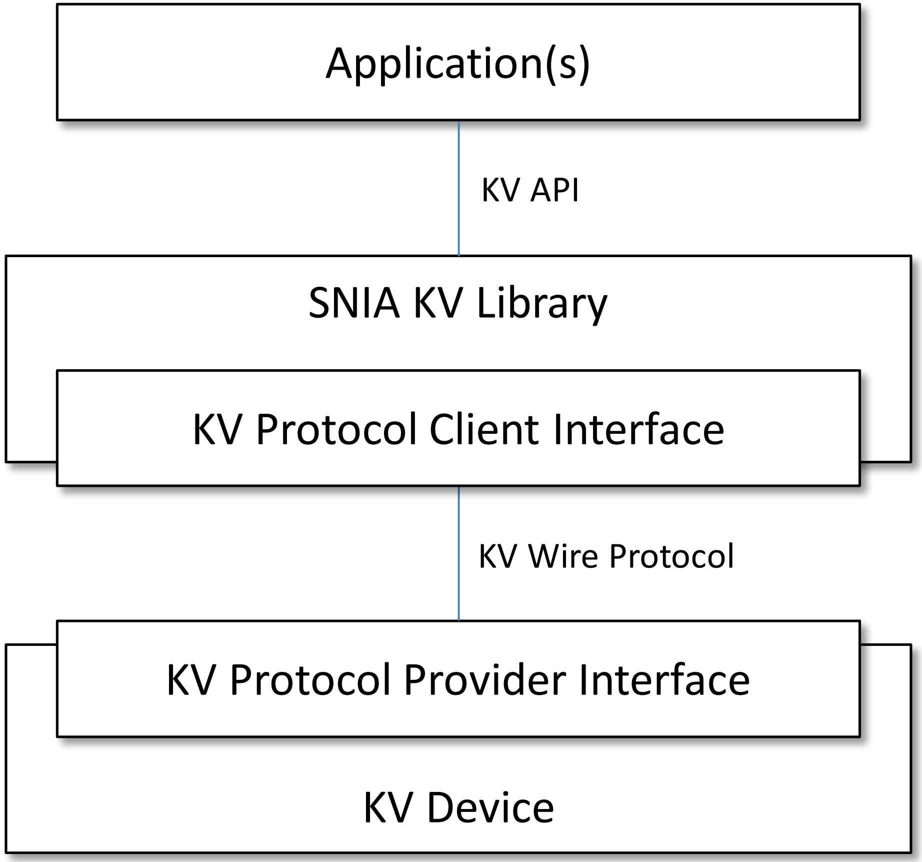


Avg 2.1M QPS@30% CPU

Key Value SSD Standards

Bill Martin
Principal Engineer
Memory Solutions Lab

Key Value SSD layers



e.g. SNIA KV API

e.g. C Library, Java, etc. Written by vendors, open source, etc.

e.g. NVMe KV commands

e.g. KV SSD

Key Value SSD Standard Activities

- **NVMe**
  - Work on a technical proposal is being discussed by the NVMe working group
  - The group is defining the scope of the work
  - This will be a new device type
- **SNIA**
  - A proposal for a Key Value API has been submitted to the SNIA Object Drive Technical Working Group
  - Discussion on the minimum necessary commands to meet basic Key Value needs is progressing

Key Value, not Object Drive

- **Both standards efforts are focused on Key Value SSD not Object Drive**
  - Key Value is a means to submit a Key and put or get a Value
  - Object Drive would include more extensive commands to query the Key Value database

NVMe Extension for Key Value SSD

- Defines a new device type for a Key Value device
- A controller performs either KV or traditional block storage commands

New Key Value Commands	PUT	GET	DELETE	EXISTS

Existing Command Extension	Admin command	Identify commands for KV	Other non-block specific commands

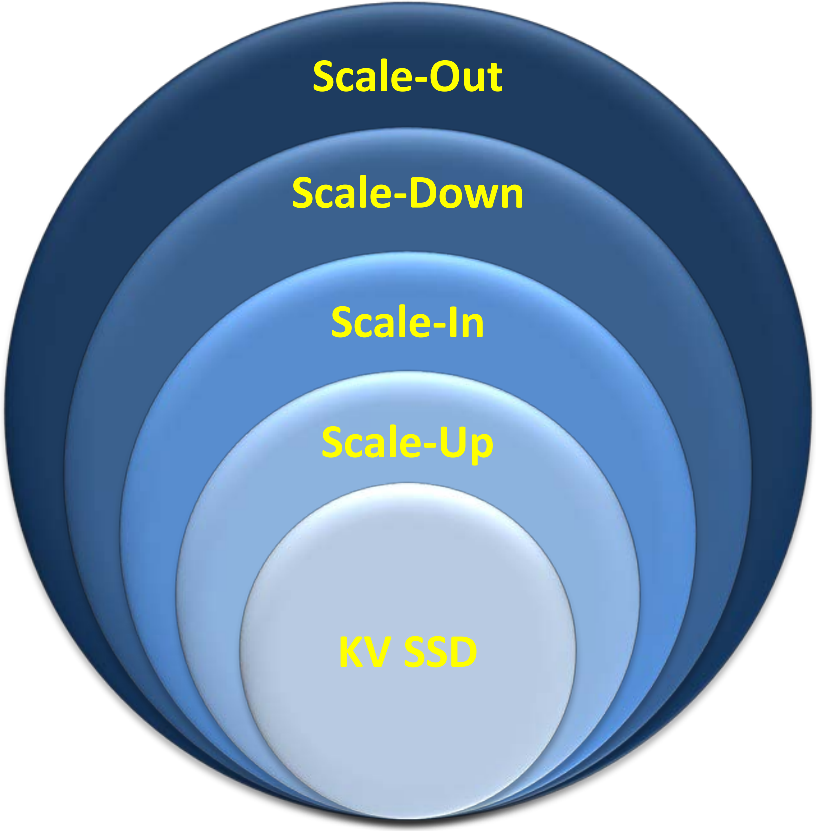
SNIA Key Value API

- **The Key Value API (Application Programming Interface) has been presented to SNIA for consideration in the Object Drive Technical Working Group**
- **Defines a Tuple**
  - Key
  - Value
- **Defines KV specific constants**
  - Max Key Lebngth
  - Alignment Unit
- **Key type supported**
  - 4 byte fixed
  - 8 byte fixed
  - Variable length character string
  - Variable length binary string
- **The API defines the calls that an application may make to the Key Value device interface**
  - These calls are independent of any specific implementation
  - These calls support the basic commands proposed for the NVMe standard
    - Open/Close
    - Store/Retreive
    - Exist
    - Delete
    - Containers/groups

Call for Participation

- **NVMe work is proceeding in the NVMe working group**
  - www.nvmexpress.org
    - Contributors and Promoters have access to working proposals
- **SNIA work is proceeding in SNIA Object Drive Technical Working group**
  - www.snia.org
    - Members may join the Object Drive TWG and have access to working proposals

Key Value SSD is a Scalable Solution with Better TCO



- Linear performance and capacity scaling
- TCO reduction
- CPU or server reduction
- Dense performance and capacity scaling
- Lean host software stacks

Questions?

kvssd@ssi.samsung.com