Emerging CS SSD Architectures

Ramdas Kachare, Sr Director, System Architecture
Memory Solutions Lab (MSL), Samsung Semiconductor Inc

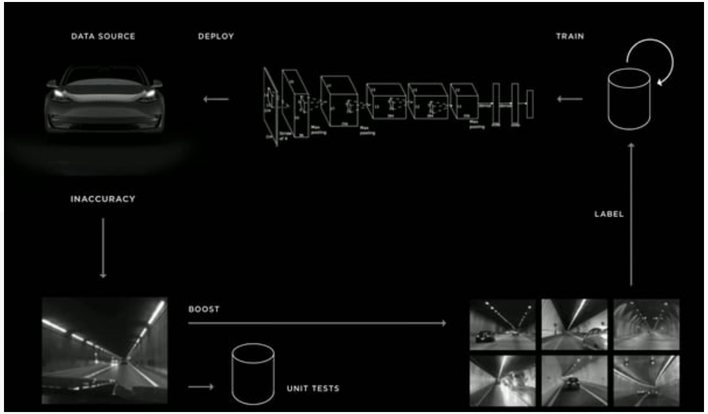
Abstract

Large amounts are data being generated and stored by various applications such as social networks, autonomous driving, and IOT devices. Such vast amounts of data can be processed to gain insights into the applications needs and thus improve overall productivity. Processing data consumes significant resources such as CPU cycles, power, and memory bandwidth. Computational Storage is a emerging technology that facilitates processing of some of the data closer to the storage. Thus it attempts to reduce data movements and thereby reduce data processing costs. This talk looks at the current Samsung CS SSD architectures, observations, and learnings and explores some potential future architecture directions.

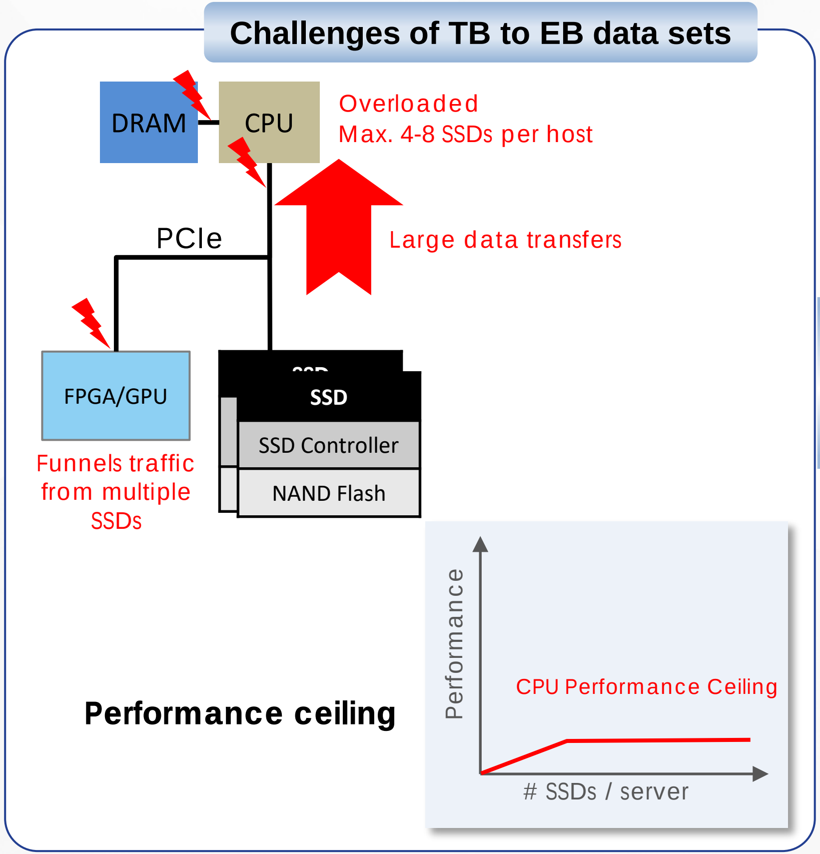
Background

- Data explosion
  - Humongous amount (~60 ZB in 2020), and keeps growing (~20% CAGR, 2020-2025)
- Data driven everything!
  - Improve application productivity using data
- Efficient data processing
  - Compute resources cost – CPU, Memory, Network, energy
  - Performance – latency, throughput, jitter
- Energy consumption
  - Becoming significant portion of overall power consumption

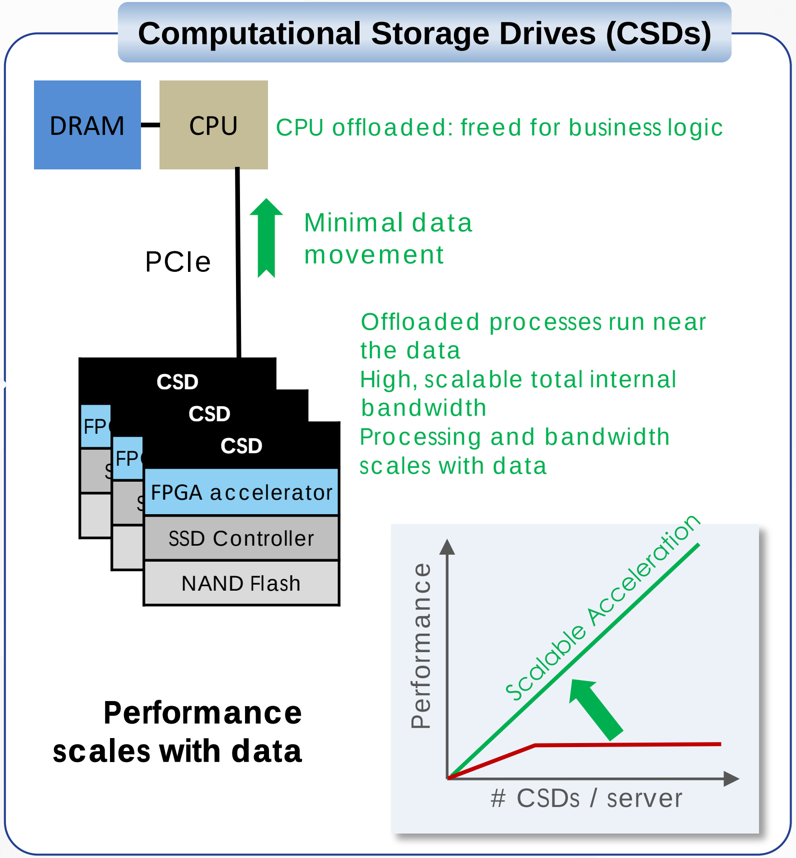
Tesla Data Engine



Why Computational Storage?

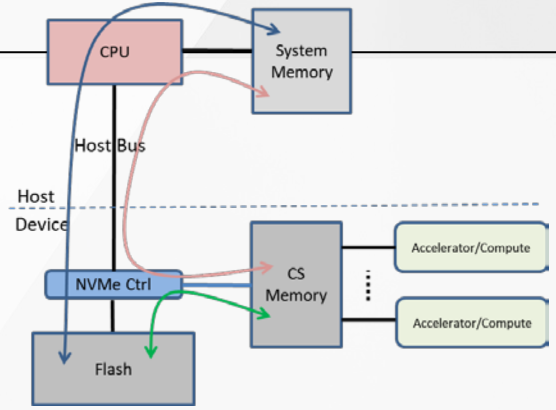


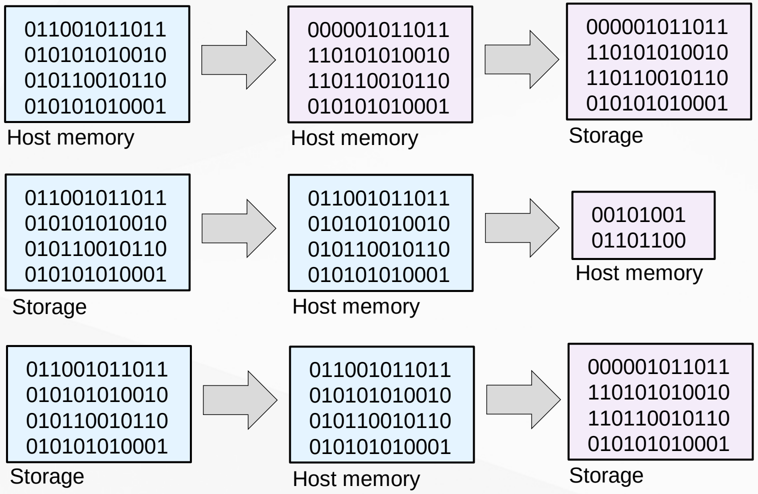
Move compute to the data



Computational Storage – basic premise

- Three phases for every use case/application
  - Load data for processing
  - Process data
  - Get the results
- Reduce unnecessary data transfers
  - Process data in or closer to storage device when optimal
  - Offload data processing
- Reduce latency of computation as seen by applications
  - Start data processing at the earliest
  - Eliminate data hop
- Moving data to Host for processing is expensive
  - CPU cycles, host bus bw, system memory size/bw
  - Power consumption – processing, cooling



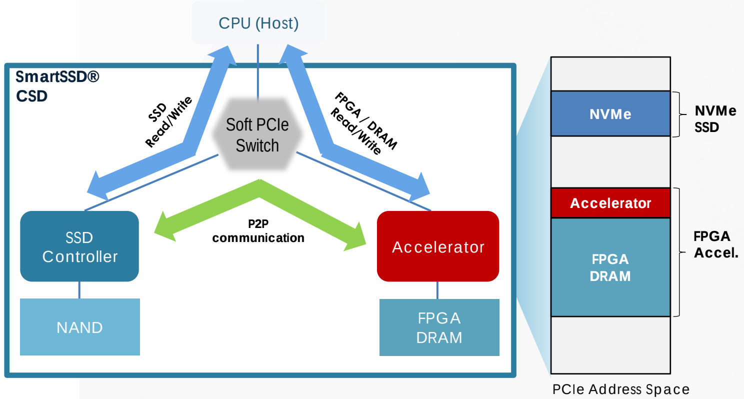


Computational Storage – example applications

- Search in storage
  - Regex, text, objects, files
- Database scan and filter queries
  - Scan heavy
  - Analytics
- Video processing
  - Object detection
  - Transcoding
- Storage services
  - Compression
  - Encryption
  - Media management

Gen I – Samsung SmartSSD™ 1.0

- Search in storage
- Database scan and filter queries
- Video transcoding, processing
- Financial analytics
- Storage services
  - Compression, Encryption



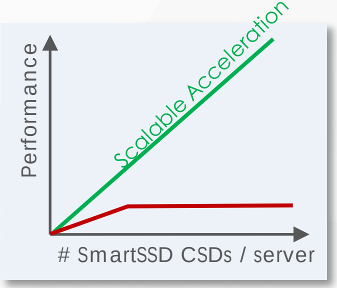
High performance Samsung Enterprise SSD controller (PM1733)

Scalable, high-speed internal data path

Low-power KU15P FPGA for data processing and acceleration: >300k LUTs, 4GB DRAM

Only processed data is moved to host

PCIe Gen3x4 host interface: acceleration without consuming valuable PCIe lanes



Feature	Specification	
Form factor	U.2	
Host interface	PCIe	Gen3x4
FPGA	Xilinx	Kintex Ultrascale Plus KU15P
	LUT	523K
	Flops	1.0M
	DSP	1968
	BRAM	34 Mb
	URAM	36 Mb
DDR Channel0	1x SMT	4 GB, DDR3, 2400 MTS
Power		25 W
SSD		4 TB
NOR Flash	1x	256 MB, QSPI
JTAG USB	1x	
LED		
Temp Sensor	1x	

Observations and learnings from SmartSSD™ 1.0

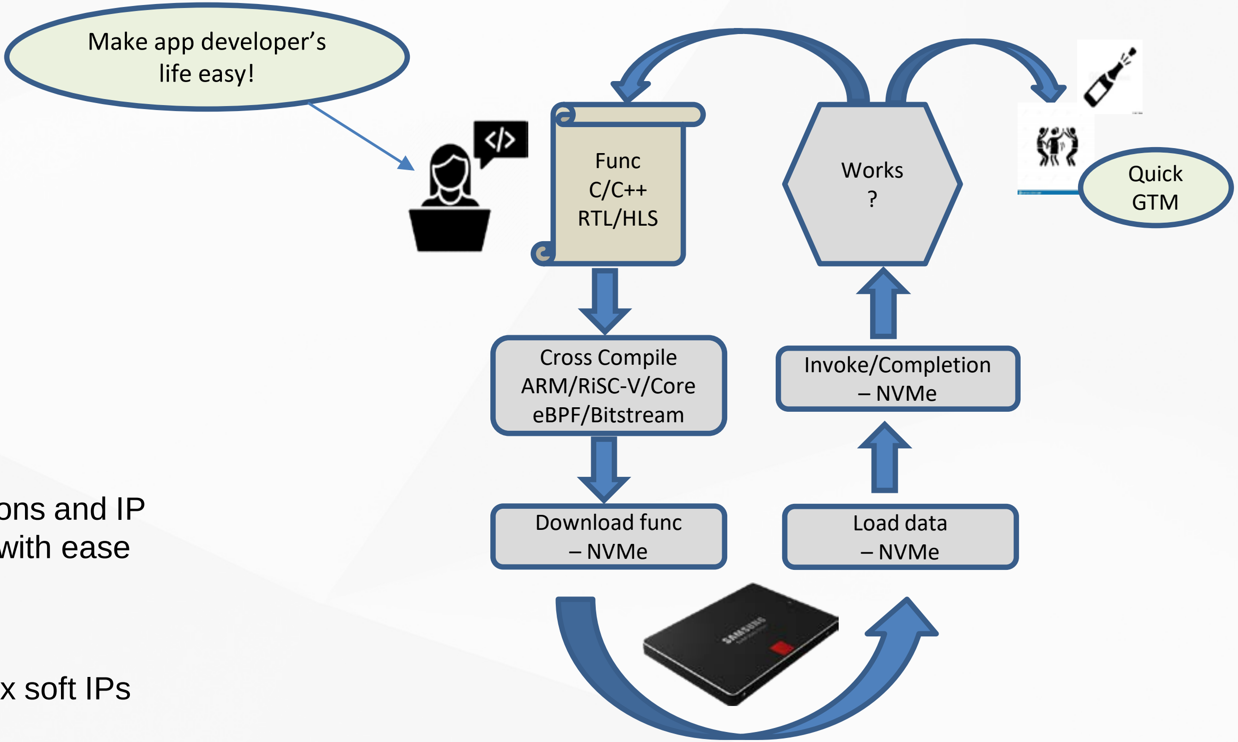
- Value in near storage compute
  - Efficient utilization of flash bandwidth
  - Reduced system resource costs
  - Lower latency experienced by applications
- One size does not fit all
  - Wide range of use cases, wide range of requirements
  - Value proposition differs for different use cases
- Cost
  - Value gained by user must be higher than the cost incurred including externalities
- Power
  - Value offered must be realizable within user power envelope
- Optimized architectures
  - Multiple architectures
  - Maximize value for different market segments

Application and user requirements categories

- Development
  - What does it take to develop a CS application?
- Runtime
  - Does it work fine, meets necessary functional and performance expectations?
- Platform
  - Can it run on current and future Datacenter infrastructure?
  - Form factors, power, thermal considerations, server compatibility, scalability, and so on ..
- Deployment
  - What are the operational, management needs?
  - Security, discovery, configuration, monitoring, different system architectures, in-field debug, upgrades and so on …..

Development requirements

- Ease of development
  - Quick and easy iteration
  - Software like flow
  - Easy debug
  - RTL, S/W, or other skills
- IP reuse
  - Users able to reuse their existing IP
- Portability
  - Users be able to move their applications and IP from one provider/vendor to another with ease
- Complex Soft IPs
  - Valuable to be able to offload complex soft IPs



Development requirements – more

- Compute resource type
  - Hammer for every problem or Swiss Army knife?
  - FPGA, GPU, TPU, SoC, Fancy Co-processor?
- Easy system stack integration
  - High level abstraction APIs
  - Standardized methods
- Ecosystem – Standards, open source
  - Users may have their own drivers
  - OR, they may like Industry Standard drivers, protocols, interfaces
  - Open source user libraries
- Quick GTM
  - Fast feature enhancements
  - Future proof

Runtime requirements

- Host in Control
  - Orchestration, DMA initiation, error/exception handling
  - DMA execution by device

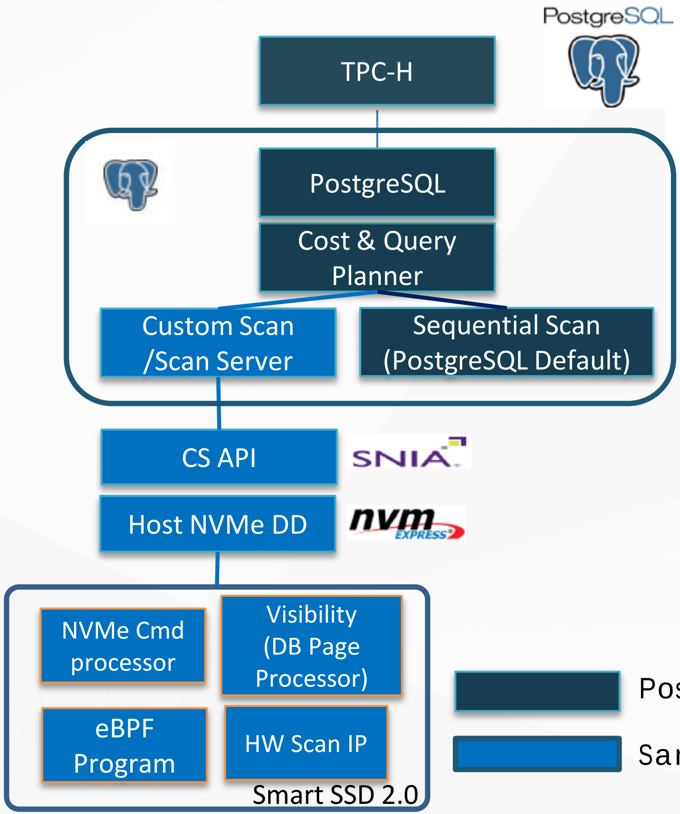
- Host orchestration efficiency
  - Low overhead of Data loading, scheduling, buffer management
  - Can reduce net value of the solution

- Predictability
  - CS device operation must be predictable

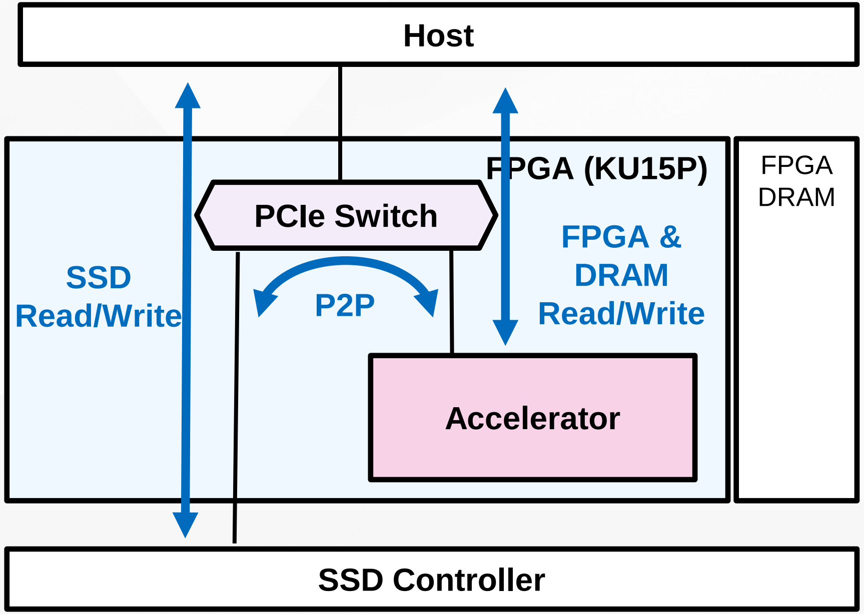
- Error and exception handling
  - Graceful handover to host
  - Limited, pre-determined fallout
  - Sufficient diagnostic data for quick debug

Gen II – SmartSSD™ 2.0

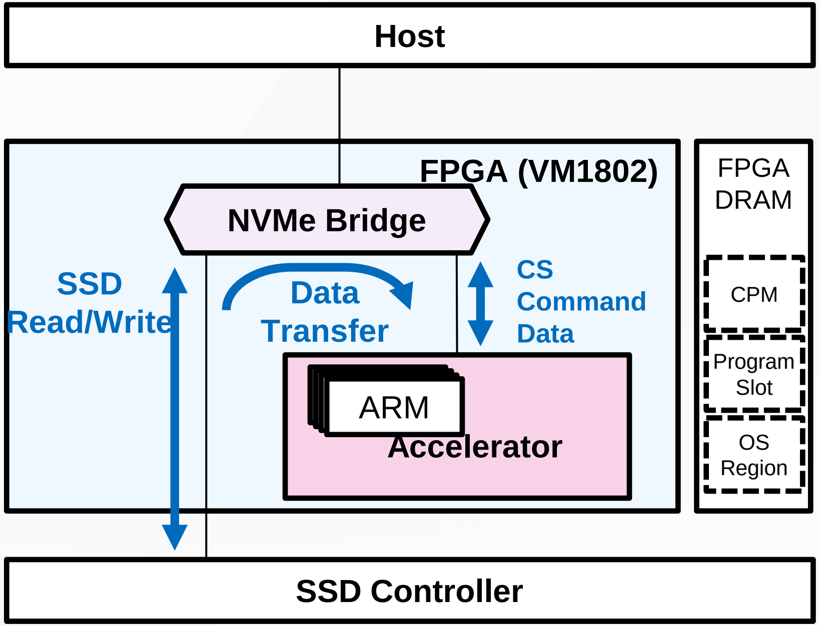
- Fixed function
  - PostgreSQL scan heavy query offload
  - Fourth most popular database
  - Easy Plug-in interface, custom-scan
- Standards based interface
  - NVMe based orchestration (TP4091)
  - eBPF support
- More FPGA resources
  - Versal VM1802 device
  - 1M LUTs
  -
- ARM as additional compute resource
  - Dual A72 + Dual R5
  -
- E3.L form factor
  - 40 watts power
  - 16 GB DRAM
  - PCIe Gen 4x4 support



SmartSSD™1.0 vs 2.0 Hardware

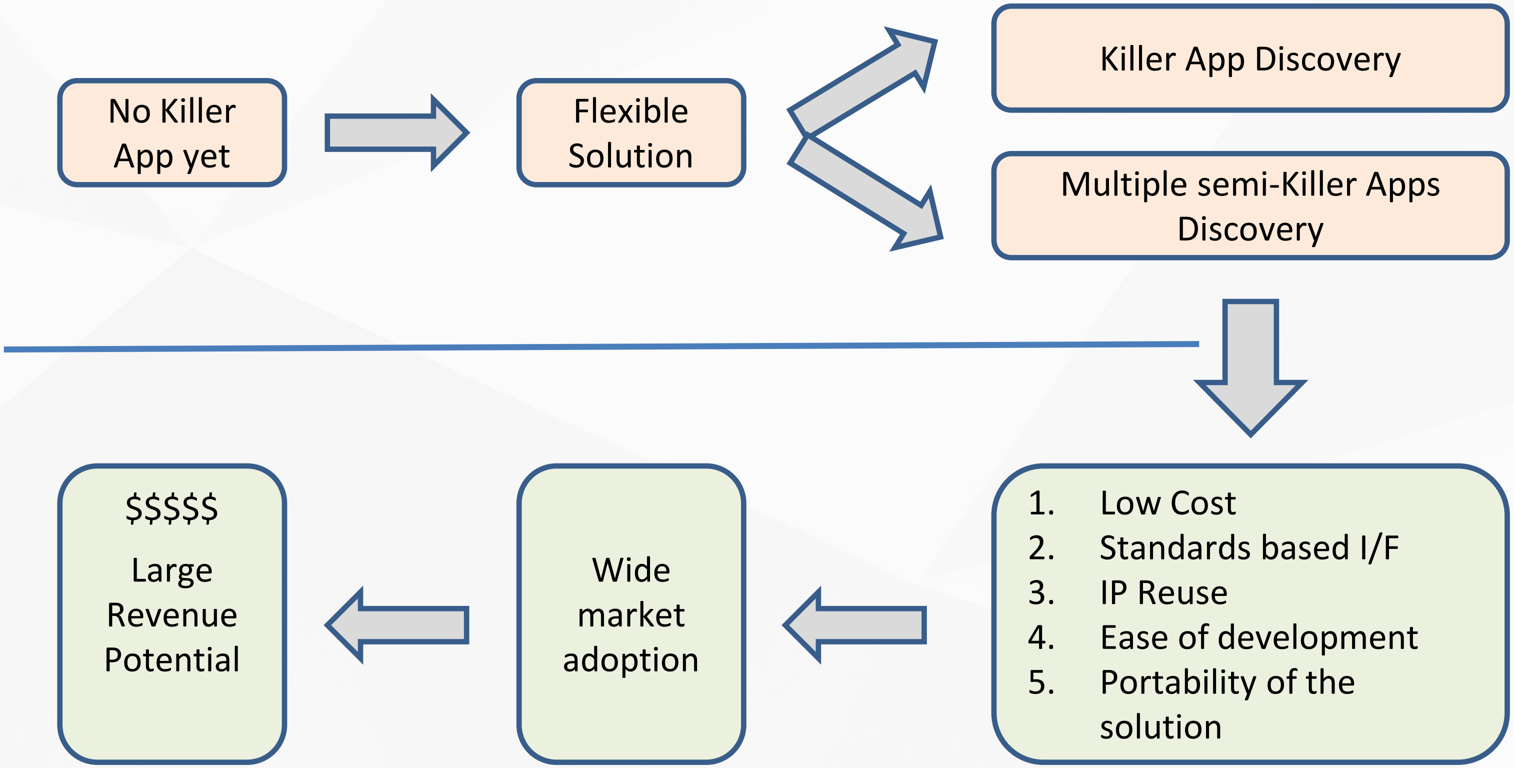


SMART SSD™ 1.0



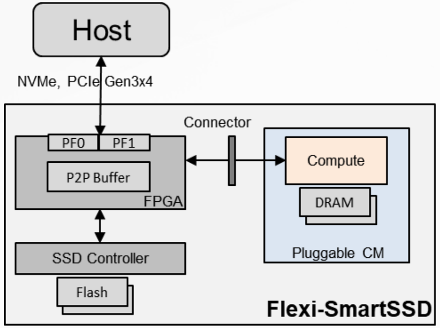
SMART SSD™ 2.0

Another Path to Market



Computational Storage Pathfinding – Flexi-SmartSSD™

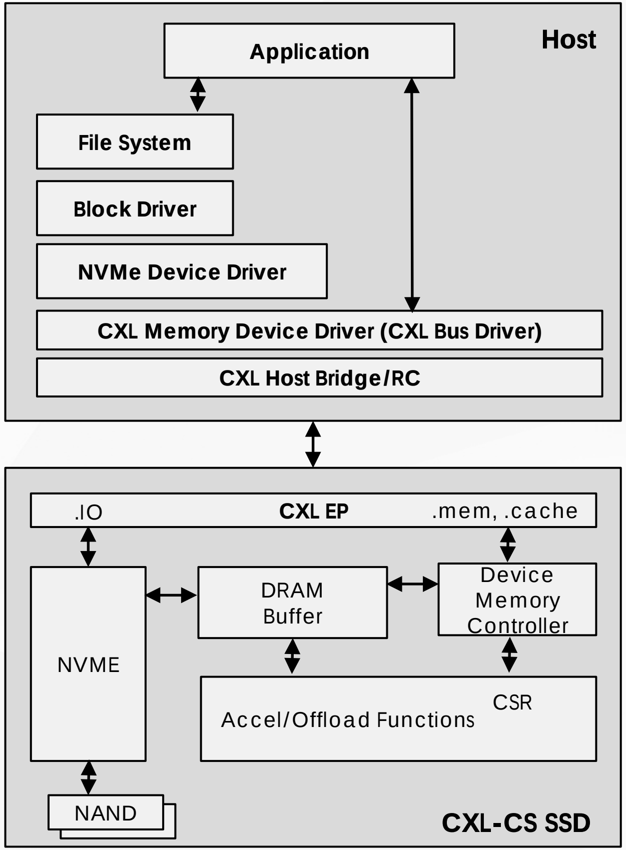
- Flexi-SmartSSD™
  - Multi-function storage device
  - Upto four PF PCIe storage device
  - PF0 NVMe device driver
  - PF1 Compute driver from user
- Base platform
  - Off the shelf SSD
  - Lost cost, low power, small sized FPGA
- Compute options
  - FPGA, GPU, TPU, NPU, SoC
  - Industry partnerships, ecosystem
- What problems it solves?
  - Wide range of application requirements
  - Complex Host software stack integration
  - Mismatch of application requirements and compute type
  - High power and cost of solution
  - Limited addressable market segment



Feature	Specification		Comments
Form factor	AIC	FHFL	
Host interface	PCIe	Gen4x8	Host interface
FPGA	Versal	XCVM1802-2MSEVSVA2197	
	LUT	899K	
	Flops	1.8M	
	DSP	1968	
	BRAM	34 Mb (967x)	
	URAM	130 Mb (463x)	
	APU Core	Dual A72	
	RPU Core	Dual R5F	
DDR Channel0	1x SMT	16 GB, DDR4, 3200 MTS	on-board
DDR Channel 1, 2	RDIMM	64 GB, upto 256GB, DDR4	per channel
Power		75 W	worst case, max configuration
Status LEDs	5x		Power good, FPGA boot, Memory Error PCIe Link
M.2 Connectors	4x	22x110, M Key, Gen4x4	For compute modules
NOR Flash	1x	256 MB, OSPI	FPGA bitstream
SD Card	1x	upto 32 GB	Boot ARM
JTAG USB			Debugger board
LED	8x		For status, debug, user control
GPIO	12x		For status, debug, user configuration
Temp Sensor	1x		board tempertaure
Heat-sink	1x		FPGA

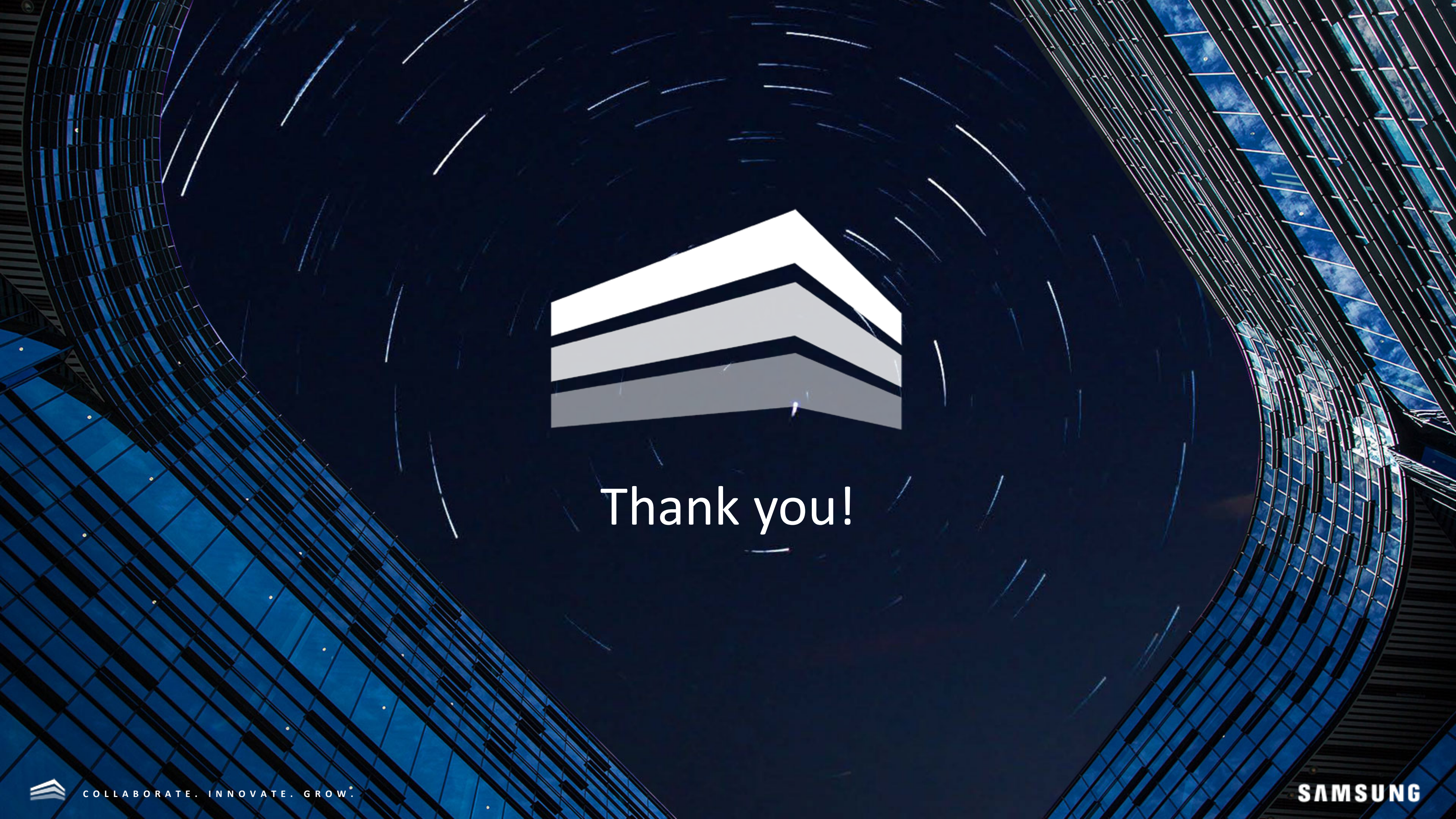
CXL based Computational Storage

- CXL interface
  - Memory interface, load/store
  - Small sized access, 64B
  - Cache coherency features
- NVMe interface
  - Block interface, file read/write
  - Fine tuned for large transfers
- Best of both the worlds
  - Use NVMe for bulk data loading for processing
  - Use CXL for acceleration/compute orchestration



Call for action

- Collaboration: End users, vendors, service providers, academia
  - Many pieces to the puzzle!
- Value propositions and validation
  - Where Computational Storage makes sense and where it does not
- System architectures
  - New possibilities to take advantage of Computational Storage technology
- Ecosystem development
  - Tools, solutions
  - Reference designs, examples
  - Standardization
  - Open source


Thank you!
COLLABORATE. INNOVATE. GROW.
SAMSUNG