DATA-CENTRIC COMPUTER ARCHITECTURE

Pankaj Mehra, Founder and CEO

July 3, 2017

Data-Centric Computer Architecture

1 **Elements of Infrastructure: Bits, Cores, and Fabrics**

2 **Data Sources, Data Varieties, and Data Growth**

3 **Data Lifecycle and Business Value of Information**

4 **Toward a Memory-Centric Architecture**

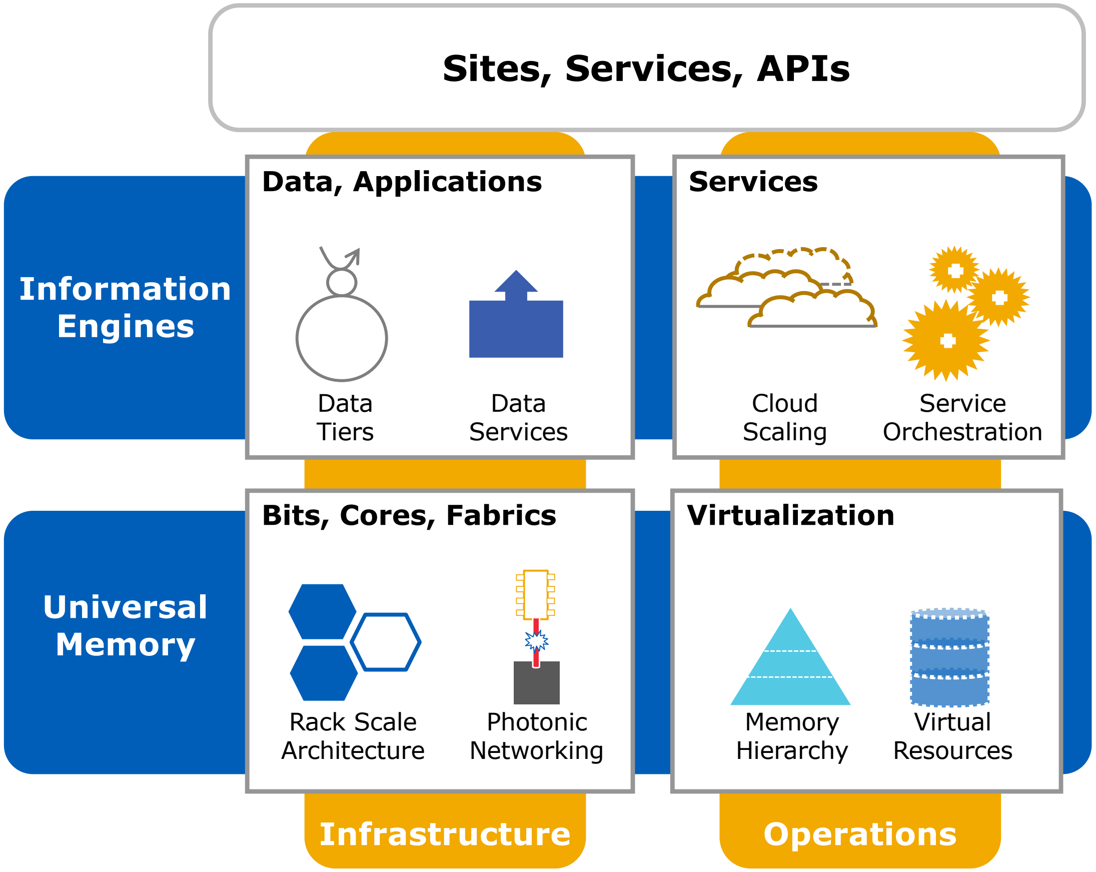
5 **iMemory Prototype**

Bits, Cores & Fabrics: the elements of infrastructure

Data Center Infrastructure in context

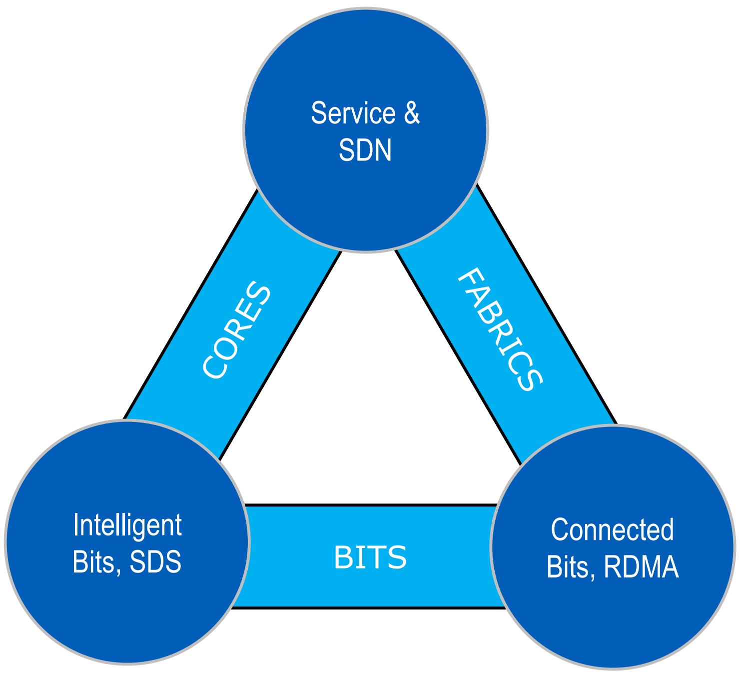
Key Themes

- Data centricity
- New memories



Bits, Cores & Fabrics

The foundation of infrastructure



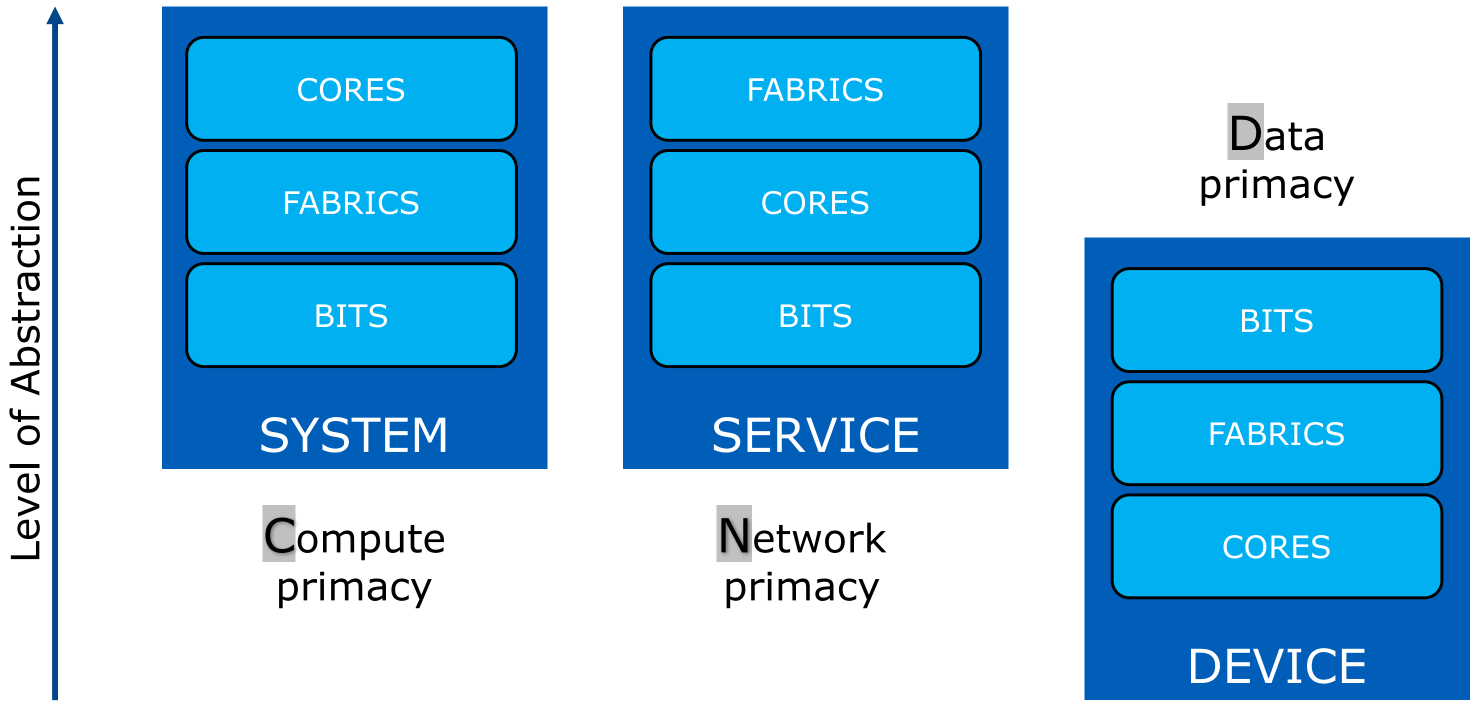
Universal System Concepts

Services	Information	Cloud
mServices	Objects	Protocols
APIs	Metadata	Topologies
Software	Data	Routes
OS	Metabits	End points
Cores	**Bits**	**Fabrics**
ALUs	Caches	I/O Ports
Firmware	Data Paths	Switches
control	state	flow

Universal Hardware Concepts

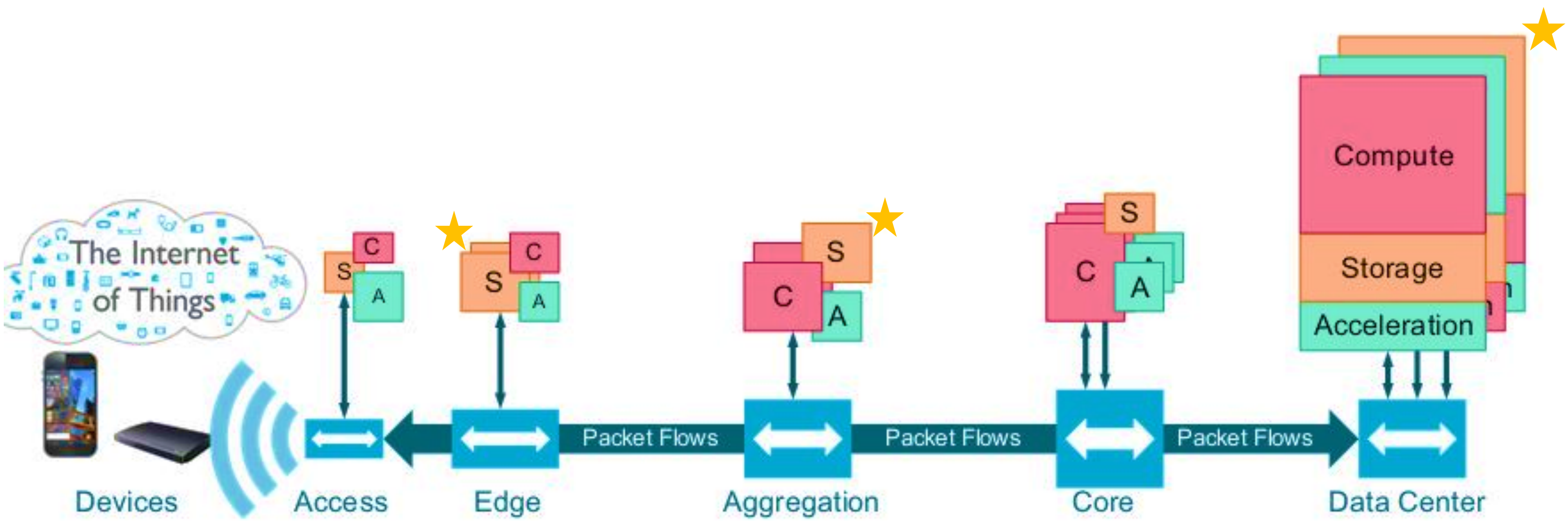
Systems, Services, Devices

Bit primacy historically at device level only



The quest for data primacy

Follow the bits



Graphic courtesy: ARM

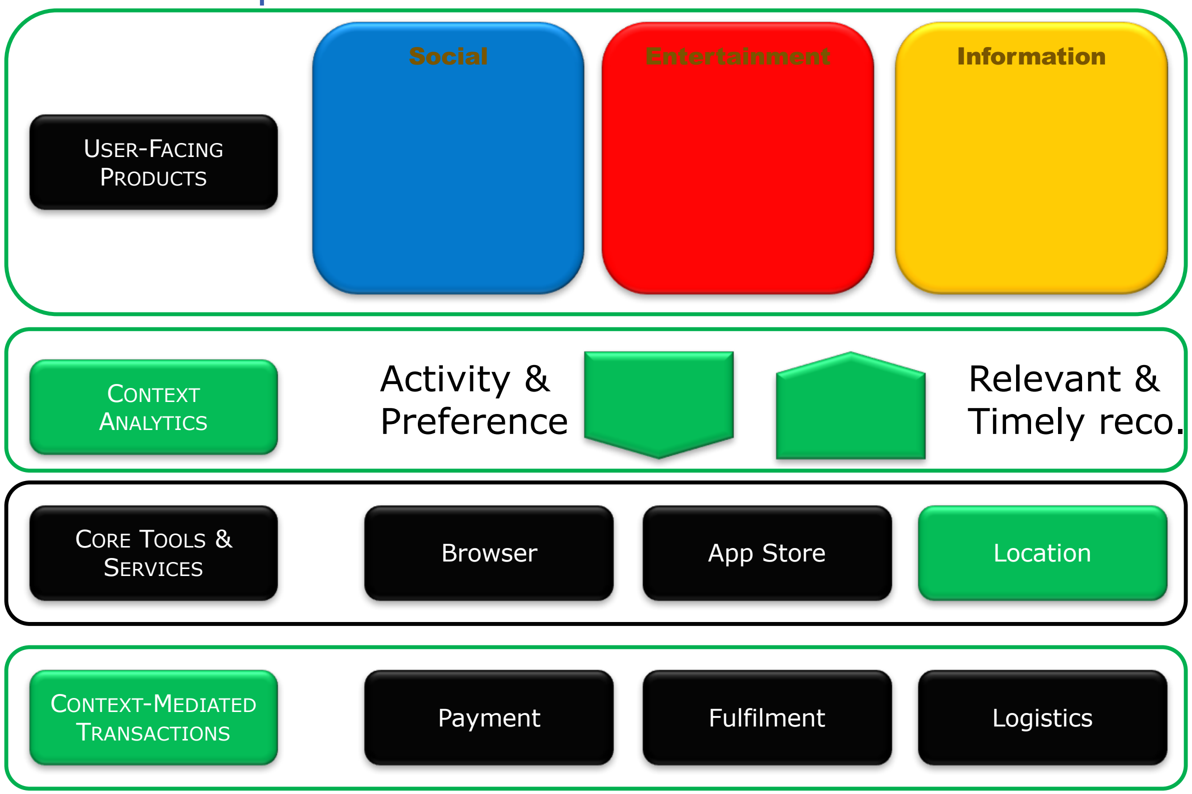
We used to call them Computer Centers!

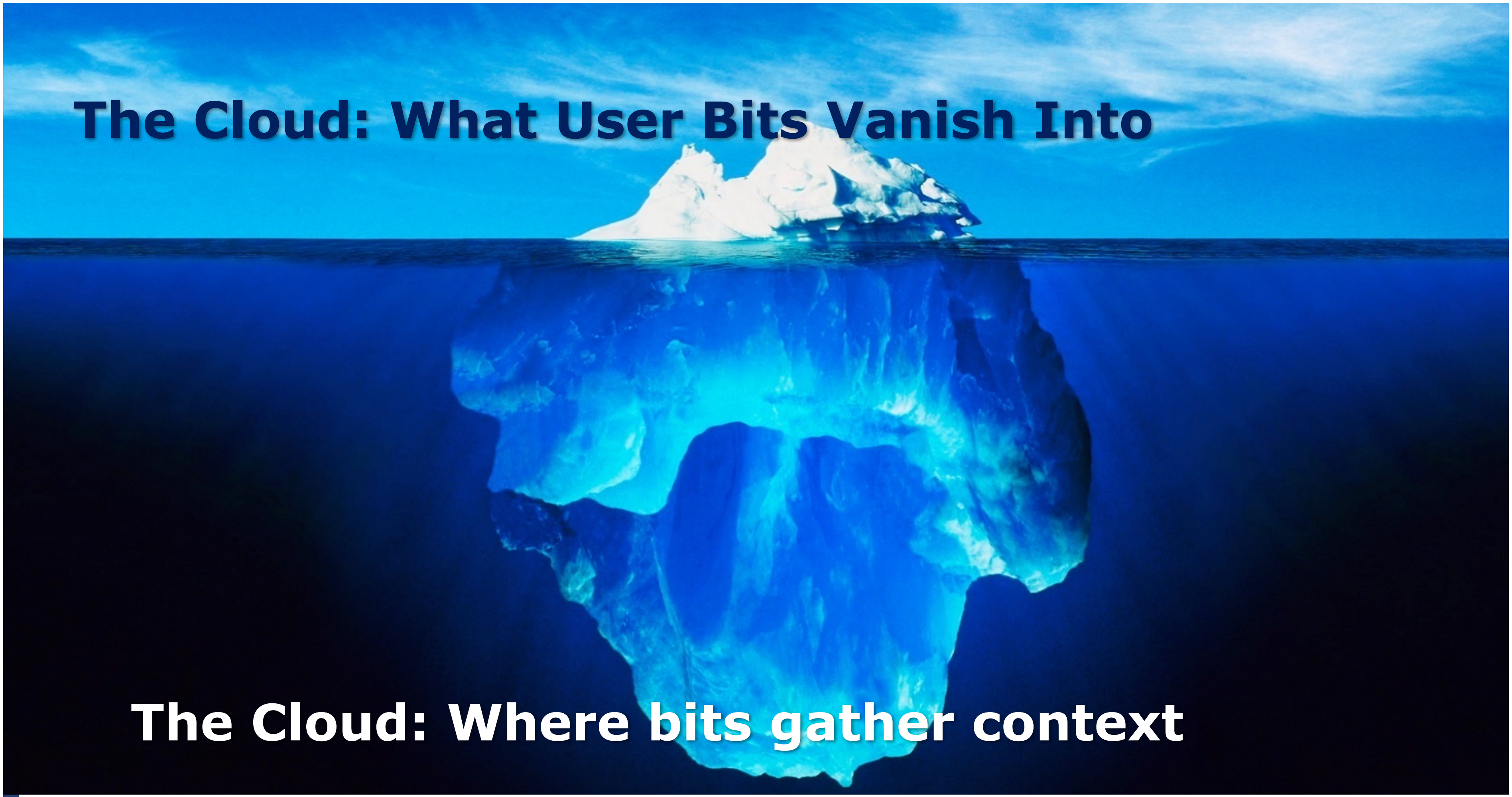
Data at the Center: Why?

Sources, Varieties, Growth

Typical One-Stop Online Portfolio

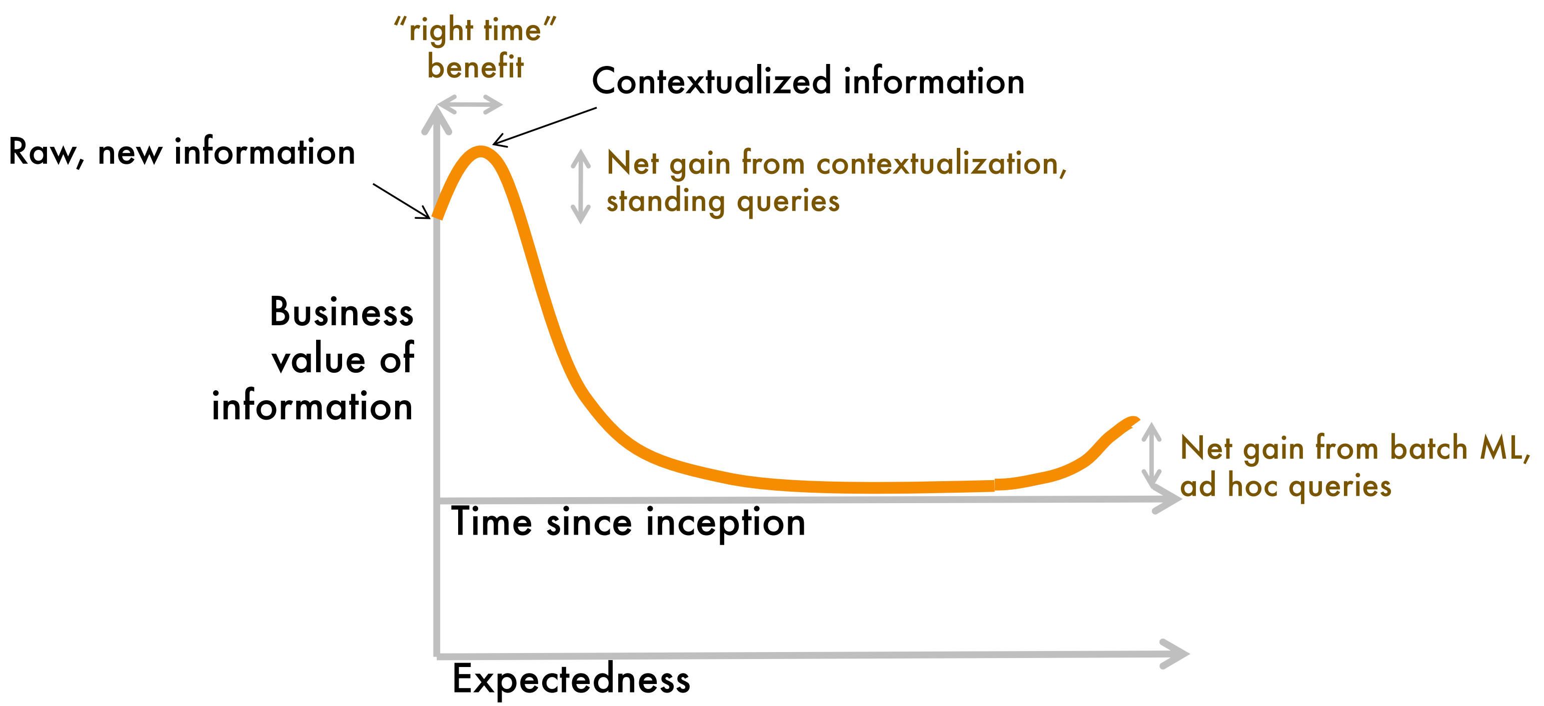
The perfect user data trap




The Cloud: What User Bits Vanish Into
The Cloud: Where bits gather context

SLA

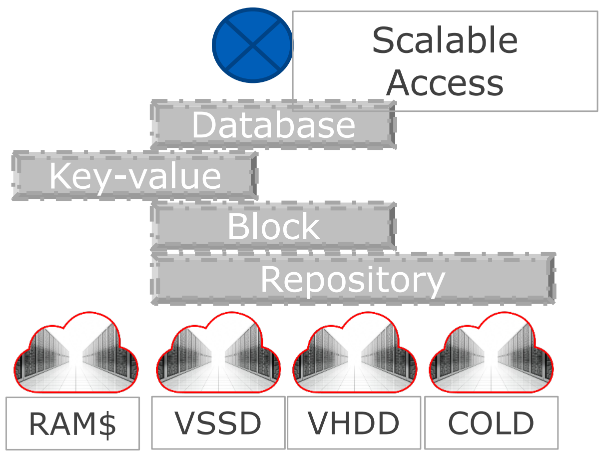
THE RIGHT INFORMATION AT THE RIGHT TIME … IN THE RIGHT CONTEXT



Typical Storage Abstraction Cake

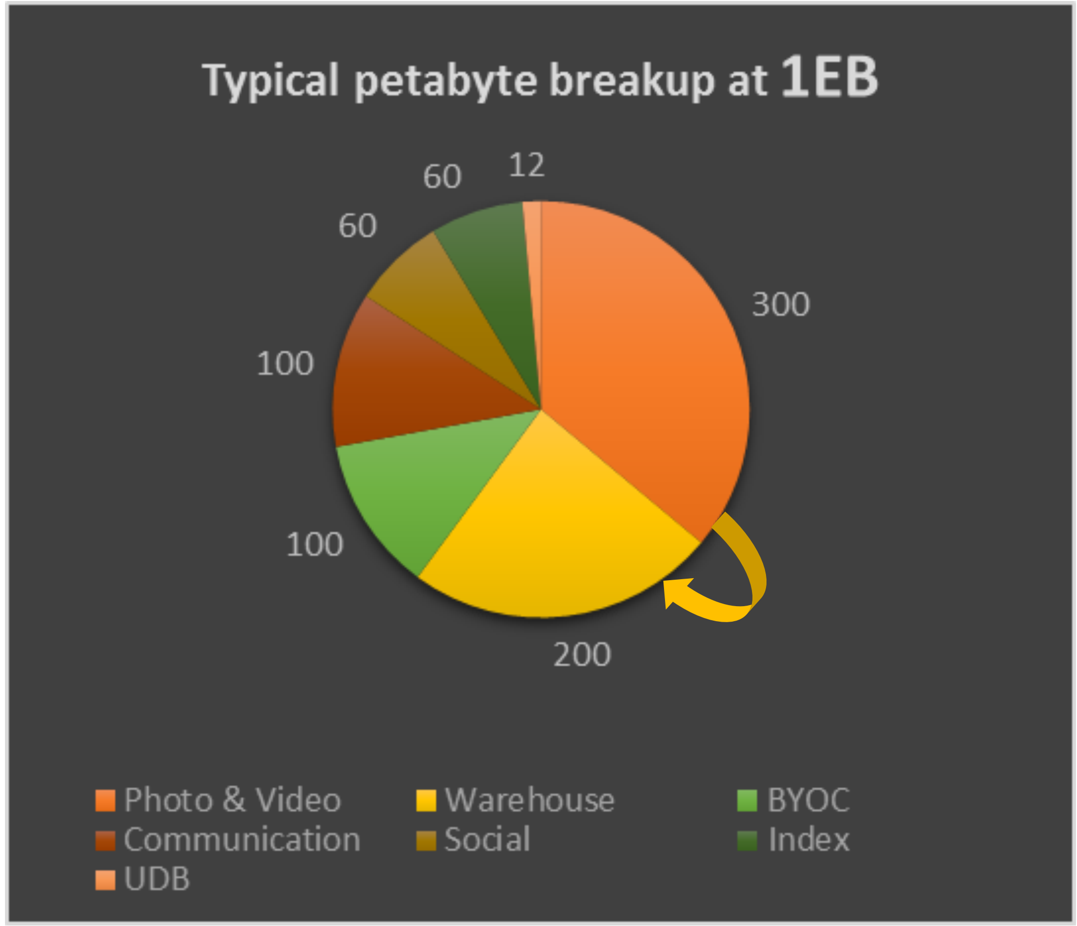
Often a shared utility owned by an Infrastructure & Ops team
for internal properties + 1000s of ecosystem partners
+ IaaS customers?

Not uncommon to find multiple EBs across 100Ks servers



User Data

- Generally,
  - Never-say-no attitude!
  - "Free & Unlimited" BYOC
  - 40+% growth in photo and video tier
    - Machine learning based information extraction
  - Users revealing each other's context in social graphs and CCOs
    - Advertising gold!



Logging, and not just transactions

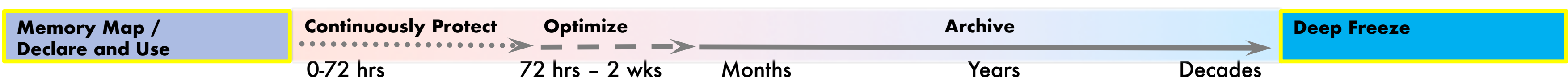
The root of all data collection

TLOG	ALOG	ELOG
TRANSACTION LOGGING	**APPLICATION LOGGING**	**LOG EVERYTHING**
Business Critical Tx in Operational Data Stores **Paid transactions ($0.10/tx)** **→ Free Transactions** ** **($0)** **Blockchain (FSI, pharma, …) for Distributed Ledger	**SIEM (ArcSight), Kissmetrics (SaaS) and Google Analytics, spur a wave of app logging** **5 EB in MSFT Cosmos!**	**The user is the product** **Every read becomes a write** **PBs/day pour in from phones, fixed cameras, cars (GM), travelers, …**

Lifecycles and Business Value of Information

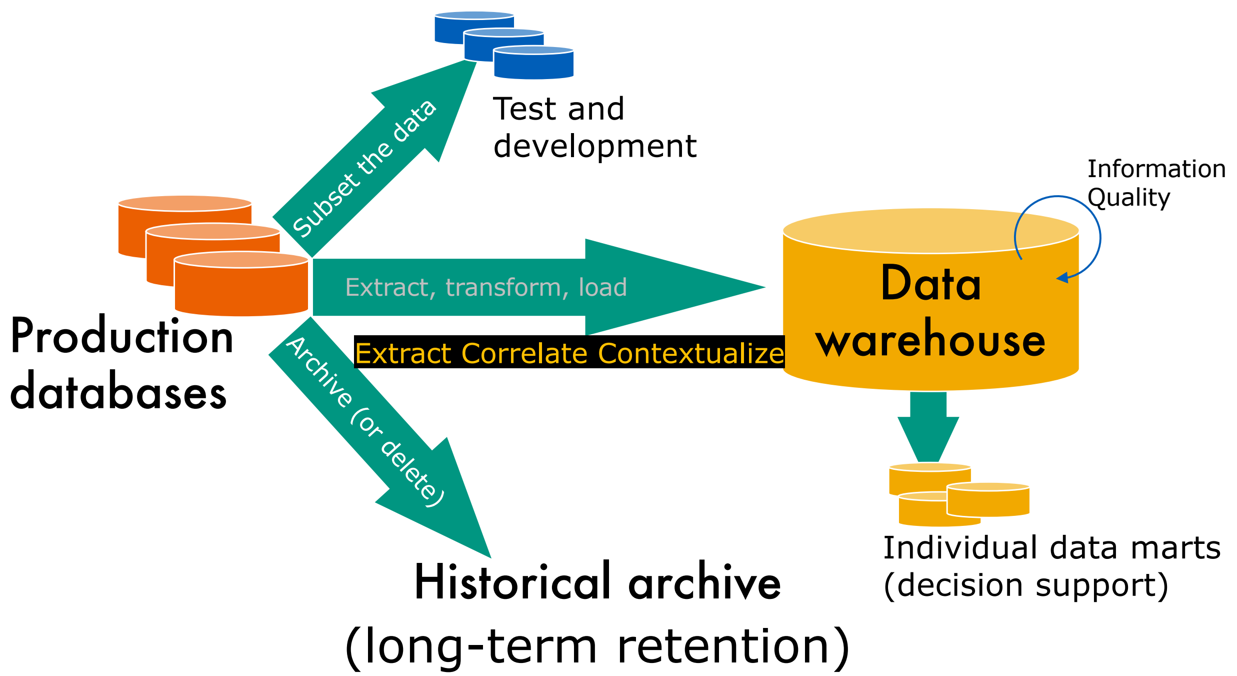
Information Lifecycle Management

Driven more by protection and retention than by cost



- Operational
  - frequently updated during 72 hours after creation
- Transitional
  - infrequently updated
  - converted to business record format
- Archival
  - static (rarely accessed)
  - subject to long-term records management

Copy Data Management



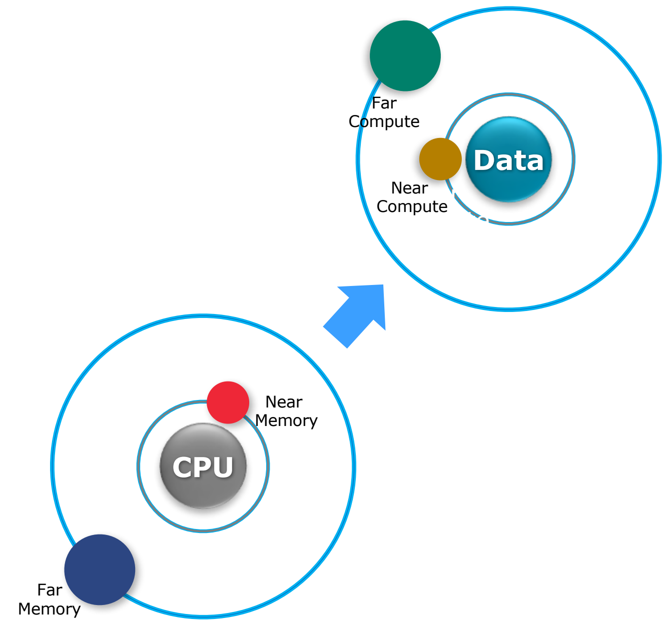
Toward MCA
Memory-centric
Computer Architecture

Shipping computation to the data

Power
Reduction in data movement count and distance

Performance
Parallelism, Bandwidth, and Latency

Cost
Low gate count embedded cores with future open ISA and tools

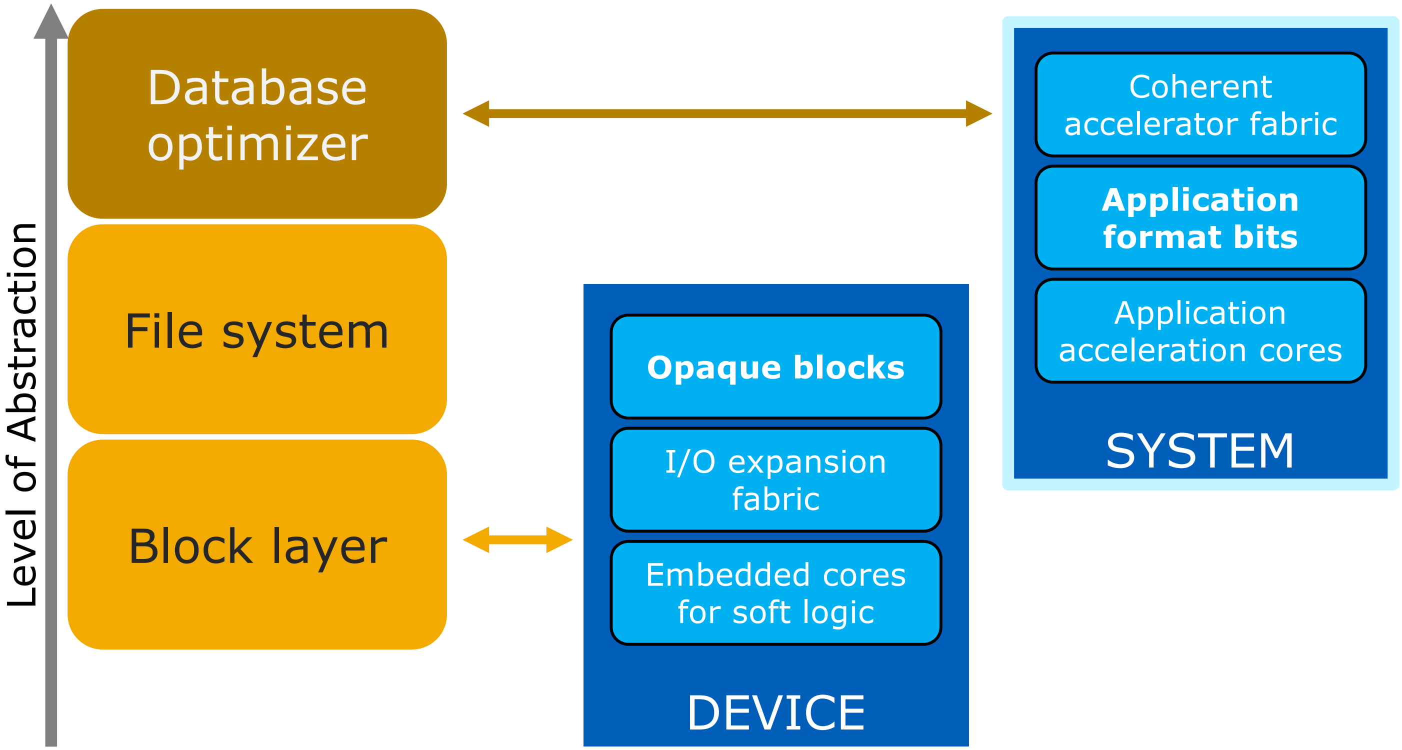


Works best when simple expressions computed against large number of data records

iMemory: Bits meet Cores

Beyond Devices: Data Primacy as the ticket to systems

Domain Specific Language optimizers are key



a new tier in the Data Center where
Data can be Big *and* Fast

Market Segments and Currently Architected Tiers

	HPC	Hyperscale Server	Enterprise Server	Enterprise Storage, Converged
Compute Tier	**Memory-storage convergence in full swing.** Several monumental shifts driven by the need to query petabytes in real time 1. Hana,a database without an I/O stack 2. Spark and ML placing analytics in focus 3. Petabytes held in DRAM by memcached and redis 4. Kafka, a pub-sub system without any storage I/O 5. pmemobj, ext4-DAX maturing			
Archive Tier	**All about highest capacity at the lowest cost.** Evolutionary shifts driven by the need to store and process exabytes at lowest cost 1. Unified scale-out filesystems for block-file-object 2. Spark and ML in Compute Tier highlight the need for bandwidth over latency in archive tier 3. Encryption, Access Control, Global deployment and wide-area optimization of data synch are key Revolutionary shifts driven by the need to retain data for 20-100 years 1. Sustained investment in optical and DNA storage to create an alternative to tape below HDD tier			

Confluence of forces driving a memory-centric tier

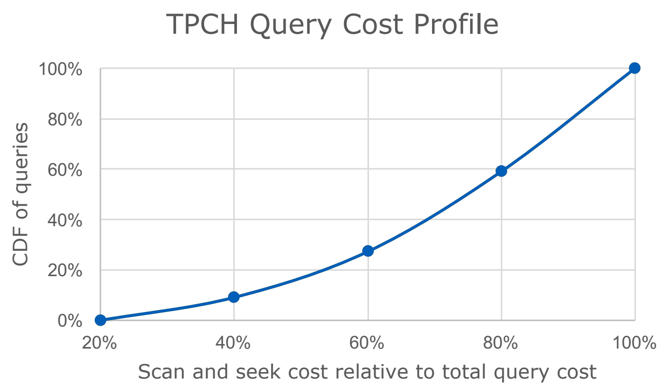
	HPC	Hyperscale Server	Enterprise Server	Enterprise Storage, Converged
Compute Tier				
Memory Tier				
Archive Tier				

Memory and storage converge
Accessing big data using ~~I/O~~ memory semantic

Memory disaggregates across fabric
Provision working memory for ~~peak~~ median usage

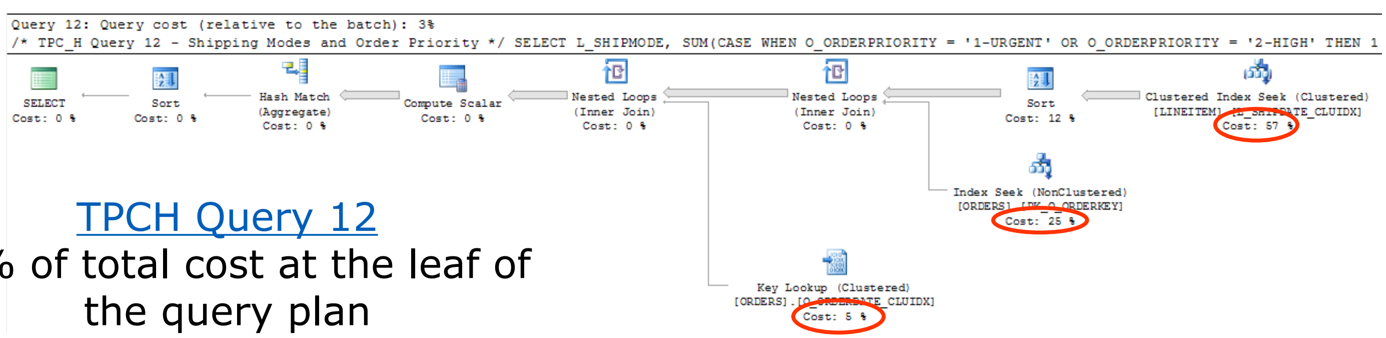
Memory-centric addressing
Bulk of processing happens near the ~~CPU~~ memory

Query execution dominated by scan bandwidth



Scan and seek cost relative to total query cost	Number of TPCH queries
<20%	0
20%-40%	2
40%-60%	4
60%-80%	7
80%-100%	9

Most queries dominated by scan and seek cost

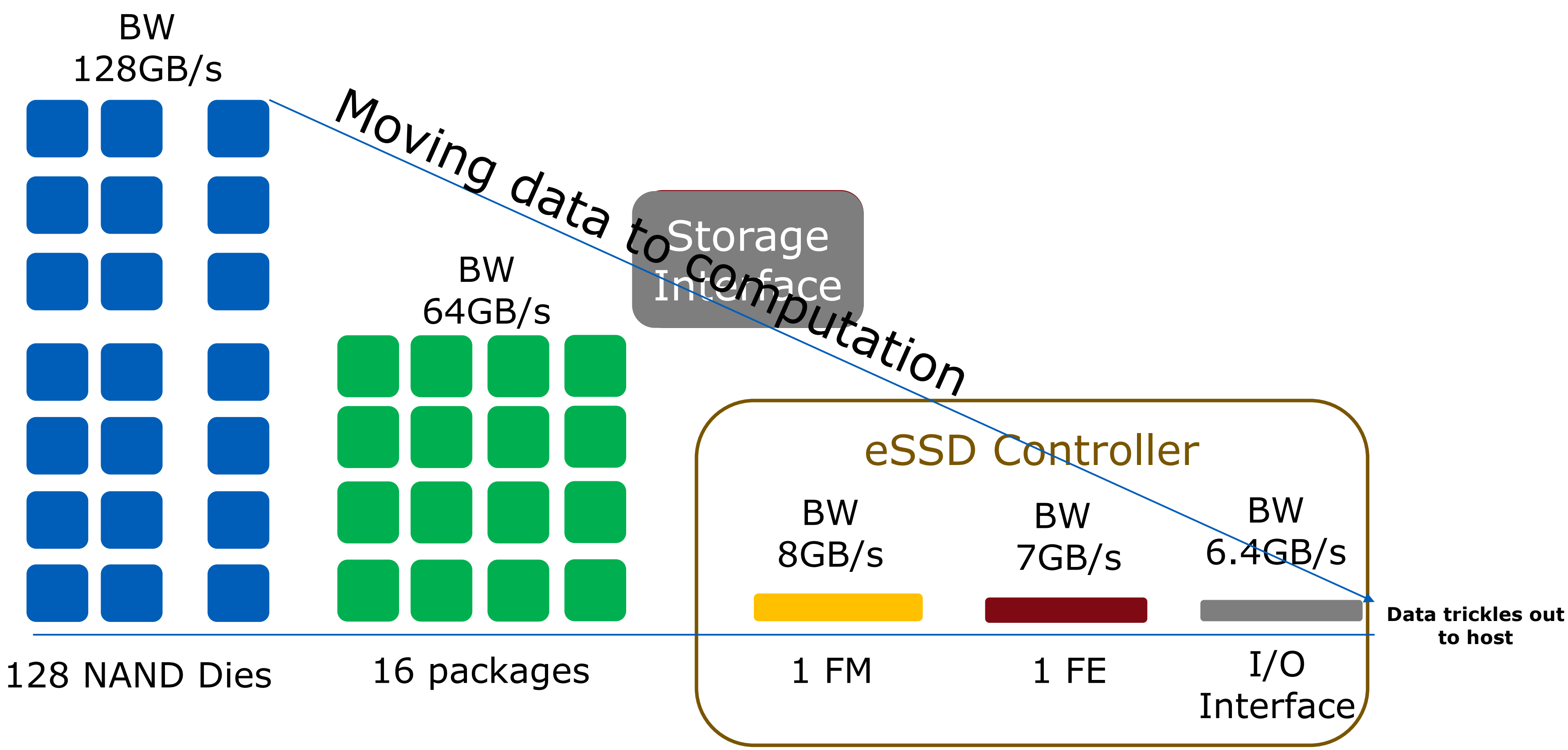


TPCH Query 12

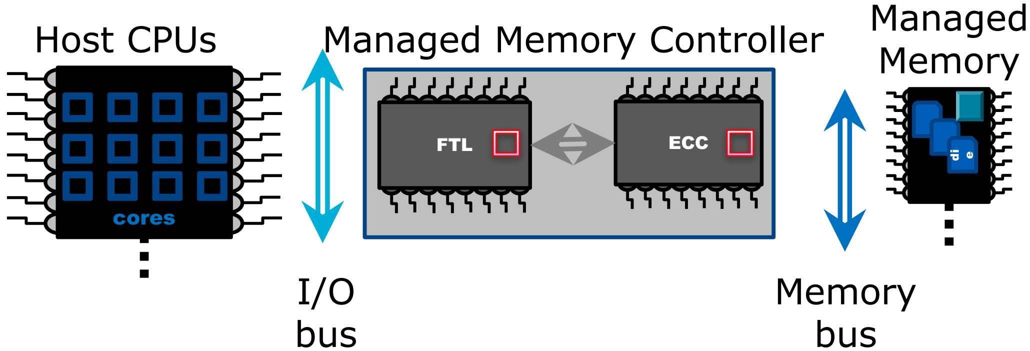
87% of total cost at the leaf of the query plan

Source: http://www.qdpma.com/tpch/TPCH100_Query_plans.html

The Bandwidth Mismatch



Possible Placements of Compute Cores in iMemory



- Conventional placement of compute cores
- Core integrated with controller
- Core integrated in die or package

Benefit

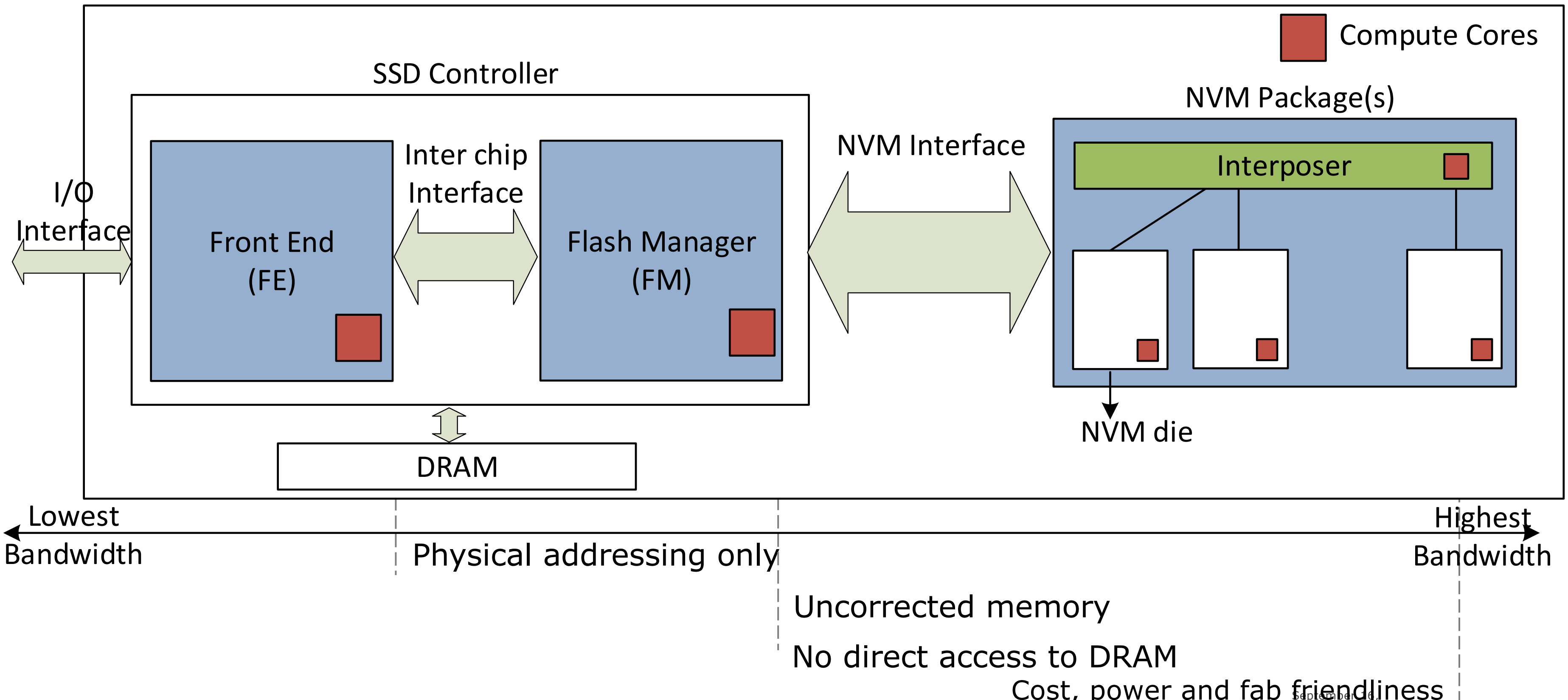
- $/perf and W/perf
- Greater bandwidth and lower latency between a computation and its data

Challenge

- Lack of ECC and possibly FTL functionality

Challenges of Core Placement in SSDs

Exploiting memory bandwidth requires rethinking memory management



Cores near memory

How many cores?

Scan bandwidth	130 GB/s		
Average record size	1000 B		
Record scan bandwidth	130 M records/s		
Computation (Instr/Record)	10	100	1000
Total processing power required (MIPS)	1300	13000	130000
Processing power per core	800 MIPS (say)		
# of cores	1.6	16.2	162.5

Another metric

MIPS/Scan bandwidth -> Processing power required per unit of available scan bandwidth
For example, in the case above, the system requires 10, 100 or 1000 MIPS per GB/s

Need low gate count, cache-less cores tuned for data-intensive workloads

iMemory Architecture

Achieving 100GB/s processing rate

- Fast Read Path:
  - Judicious core placements enable iMemory to exploit internal read bandwidth and provide order of magnitude processing bandwidth.
  - iMemory exposes cores, translations, and data placement via APIs to database optimizers.
- Auto targeting and Just-In-Time (JIT) enabled data-layer optimizers
  - Generated (not handwritten) code efficiently targets 10s-100s of DPU cores in iMemory.
  - JIT compilation improves system efficiency with optimal targeting of iMemory.
- Application aware ECC to enable high throughput decoding
  - ECC engine aware of logical and physical database schemata (record size, column count and sizes, row or column order).
  - Decoder informed on a query-by-query basis about table fields used, projected or ignored.

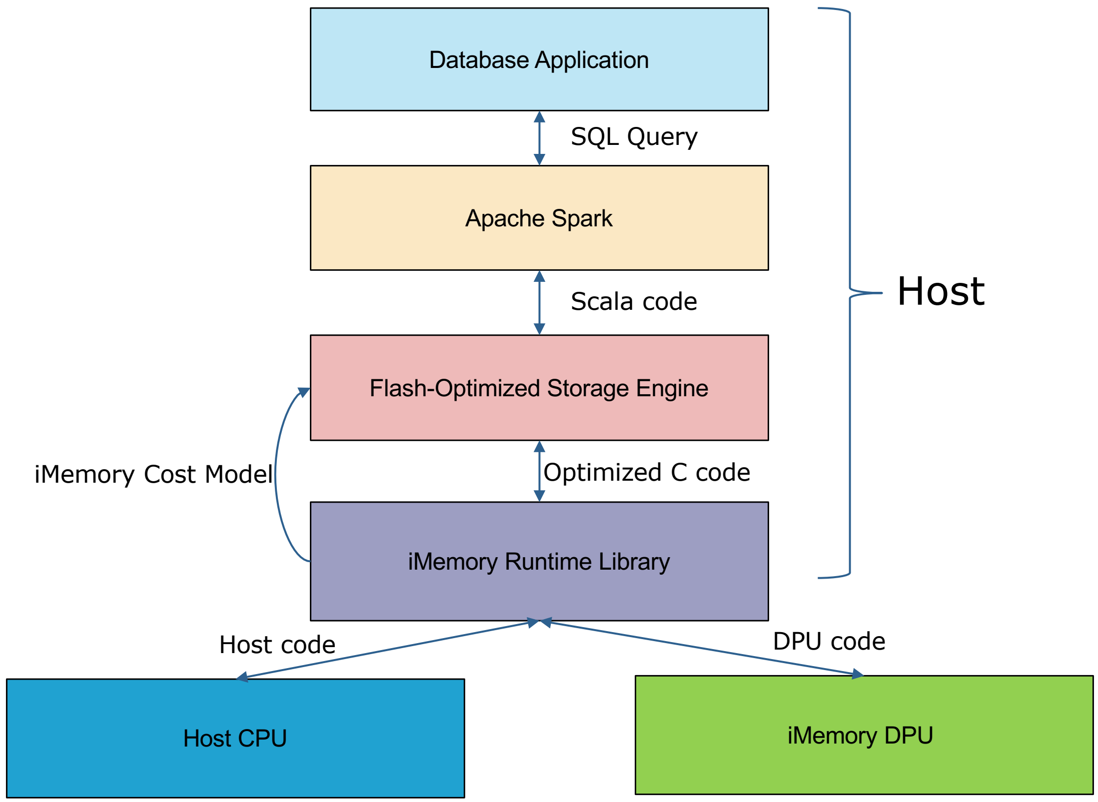
Scan Bandwidth

The road to 32.5 GB/s per TB

iMemory-1	iMemory-2	iMemory-3
Cores integrated within SSD controller	Cores integrated with memory package	Deep Integration of cores with die CuA
2019	**2020**	**2022**
4 GB/s per TB	10-12 GB/s per TB	32.5 GB/s per TB

Key Technology Enablers: Controller enhancement, Packaging, Die Enhancement

iMemory System Software Stack



Aligning with
Industry and Academic Initiatives

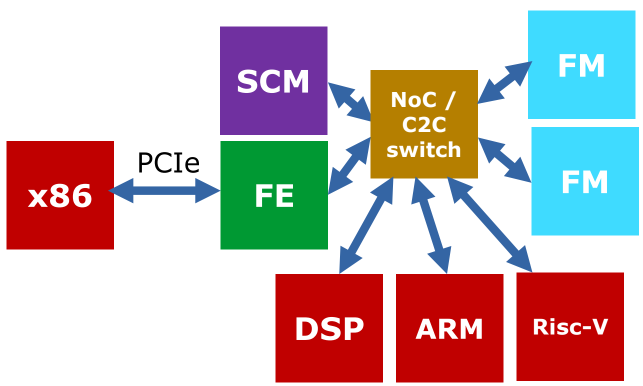
Analytics Infrastructure Scaling Trends

If it does not scale, it will fail

UP Scaling	Scaling OUT	Scaling IN
Driven by limited scale-out of Oracle RAC and even Hana for handling transactions and mixed workloads	From analytics under *add as you grow* Hadoop instances and *more iron at Spark*, now transactions in Spanner, Cockroach	*Scaling out without sprawling out.* e.g., • RackOut (SoCC'16) uses RDMA to disaggregated data shared within rack. • HPC and ML, memcache, etc. benefit 36-88% from using fewer nodes with MORE MEMORY

Scaling Down

an attractive alternative



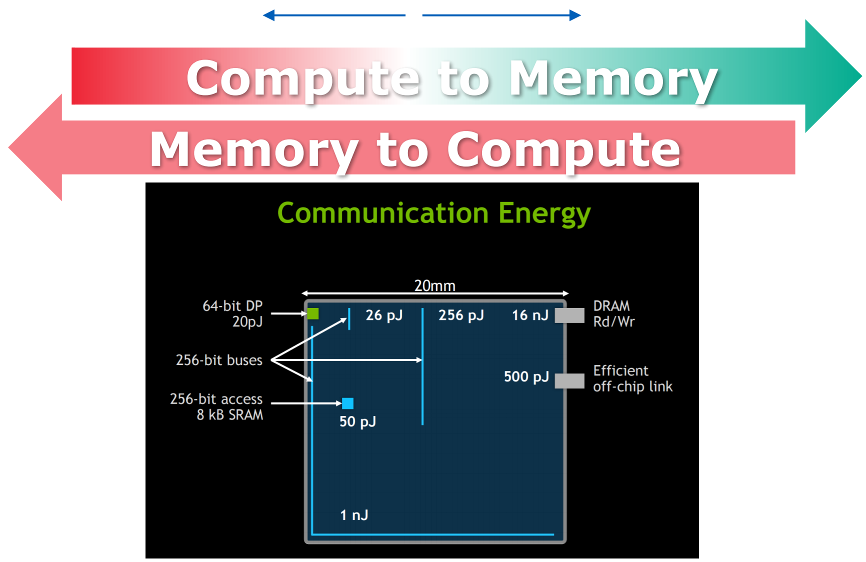
- Makes sense for lightweight compute and moderate to high bandwidths
  - Key-Value Stores, for instance!
- Delivers best cost when integrated with semiconductor memories such as flash and perhaps SCM
- Integrated with SCM, it could give GPUs, FPGAs, and von Neumann configurations with big memory a run for the money
  - HANA and IMDBs, for instance
- REQUIRES
  - Investment in optimizers
  - Low power, low cost interconnects
  - Silicon integration of cores with memory

Anthropomorphic Workloads

Hard: Logic and memory on same die
Hard: Cores routable using 3-4 metal layers
Lack of killer apps and optimization ecosystem

PIM cores

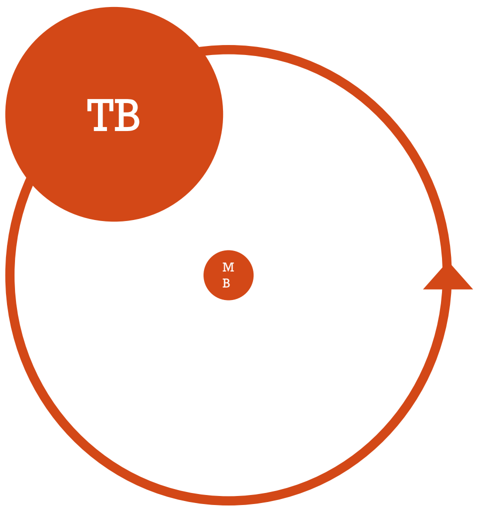
Optimizers, JIT compilers, x-compilers



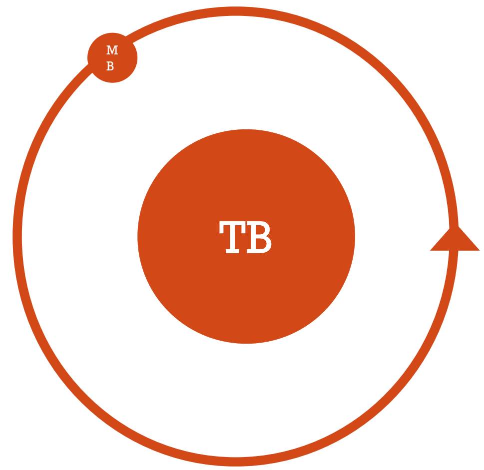
DATA GRAVITY

Also applies to data versus data

- How we process ML training



- How we should process ML training



- Feed the hungry GPUs

Optimizing Data Placement

A key optimization to develop 3-5 years out

- Pressures to get to even lower power
  - Long tail of extreme personalization + Privacy concerns/laws $\Rightarrow$ Learning in the field
  - Evolving world requires always (online) learning algorithms
- Pressures to get to even higher performance
  - Ad hoc queries against petabytes of data in real time (this talk)
  - Long standing queries (context aware computing)
  - HTAP (Ananlytics and context mediated transactions)
- Compilers and runtimes do not even recognize this as a problem yet
  - Yet, leaders in industry and academia believe this is one of the most important problems
- E.g. carefully placing matrices and vectors in such a way that dot products, matvecs, gemms, and tensor products can be computed w/o data movement
- Now, add memristive logic

THE ULTIMATE QUESTION BEFORE COMPUTER ARCHITECTS

Is this also the von Neumann vs non-von-Neumann question?