Trends and Challenges in Big Data

Ion Stoica

November 14, 2016

PDSW-DISCS'16

Before starting…

Disclaimer: I know little about HPC and storage

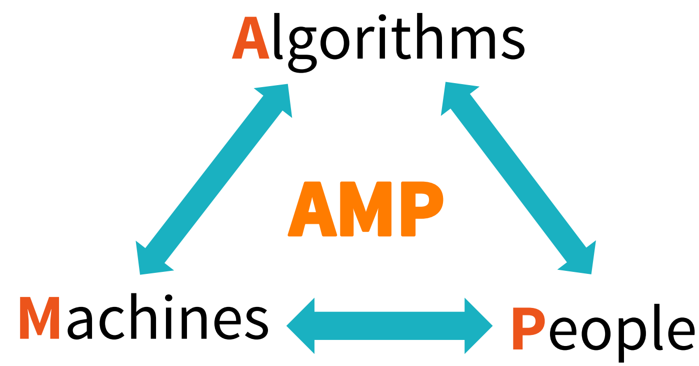
More collaboration than ever between HPC, Distributes Systems, Big Data / Machine Learning communities

Hope this talk will help a bit in bringing us even closer

Big Data Research at Berkeley

AMPLab (Jan 2011- Dec 2016)

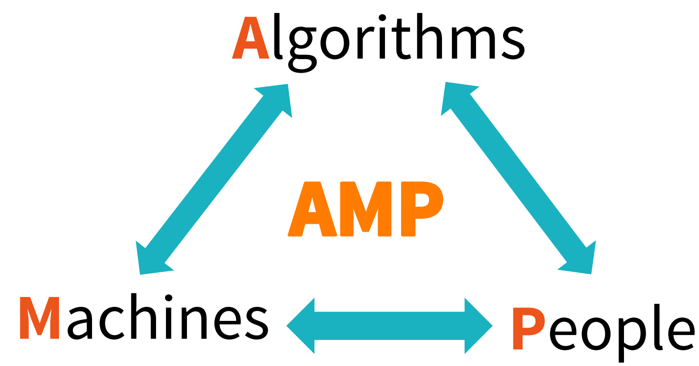
- Mission: "*Make sense of big data*"
- 8 faculty, 60+ students



Big Data Research at Berkeley

AMPLab (Jan 2011- Dec 2016)

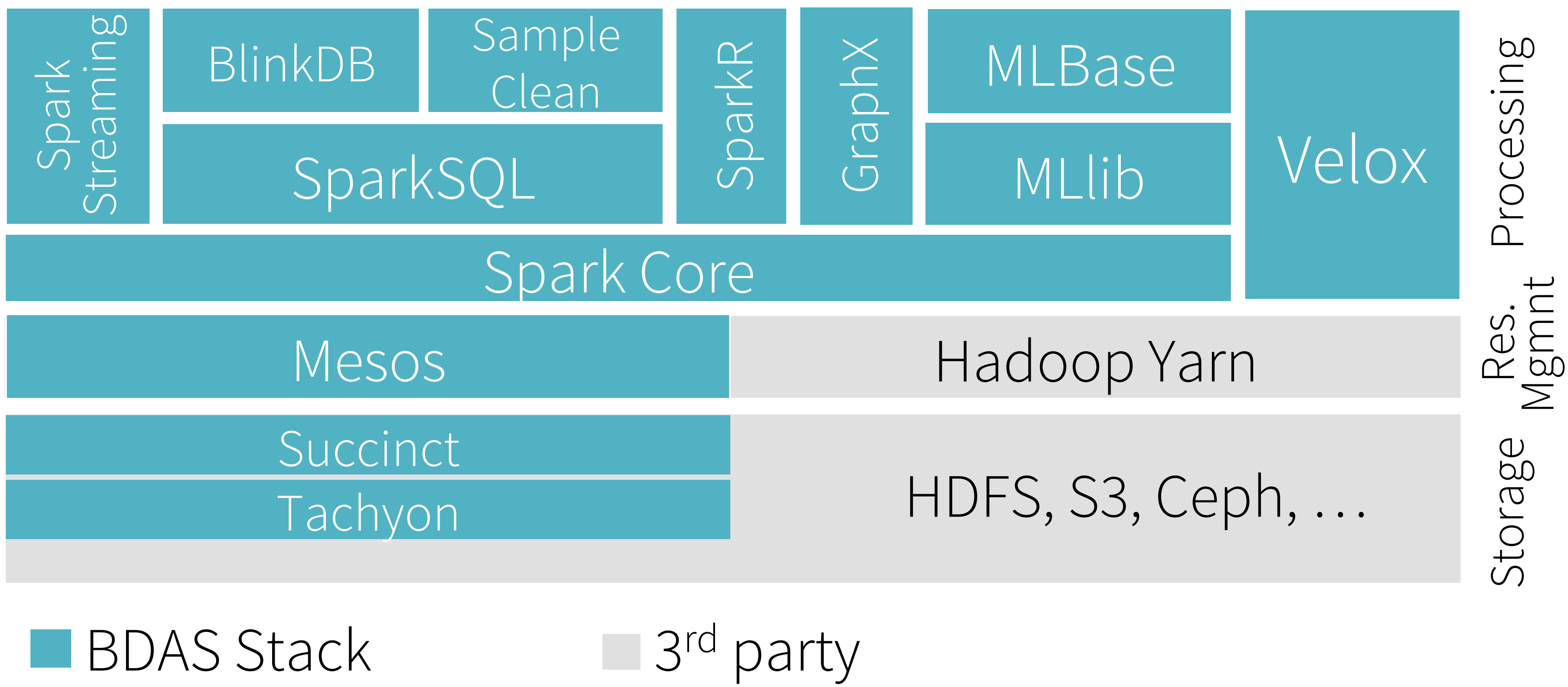
- Mission: "*Make sense of big data*"
- 8 faculty, 60+ students



Goal: Next generation of open source data analytics stack for industry & academia

Berkeley Data Analytics Stack (BDAS)

BDAS Stack


Spark Streaming
BlinkDB
Sample Clean
SparkSQL
SparkR
GraphX
MLBase
MLlib
Velox
Spark Core
Mesos
Hadoop Yarn
Succinct
Tachyon
HDFS, S3, Ceph, …
Processing
Res. Mgmnt
Storage
BDAS Stack
3rd party

Several Successful Projects

Apache Spark: most popular big data execution engine

- 1000+ contributors
- 1000+ orgs; offered by all major clouds and distributors

Apache Mesos: cluster resource manager

- Manages 10,000+ node clusters
- Used by 100+ organizations (e.g., Twitter, Verizon, GE)

Alluxio (a.k.a Tachyon): in-memory distributed store

- Used by 100+ organizations (e.g., IBM, Alibaba)

This Talk

Reflect on how

- application trends, i.e., user needs & requirements
- hardware trends

have impacted the design of our systems

How we can use these lessons to design new systems

2009

2009: State-of-the-art in Big Data

Apache Hadoop

- Large scale, flexible data processing engine
- Batch computation (e.g., 10s minutes to hours)
- Open Source

Getting rapid industry traction:

- High profile users: Facebook, Twitter, Yahoo!, …
- Distributions: Cloudera, Hortonworks
- Many companies still in austerity mode

2009: Application Trends

Iterative computations, e.g., Machine Learning

- More and more people aiming to get insights from data

Interactive computations, e.g., ad-hoc analytics

- SQL engines like Hive and Pig drove this trend

2009: Application Trends

Despite huge amounts of data, many working sets in big data clusters fit in memory

2009: Application Trends

Memory (GB)	Facebook (% jobs)	Microsoft (% jobs)	Yahoo! (% jobs)
8	69	38	66
16	74	51	81
32	**96**	**82**	**97.5**
64	97	98	99.5
128	98.8	99.4	99.8
192	99.5	100	100
256	99.6	100	100

*G Ananthanarayanan, A. Ghodsi, S. Shenker, I. Stoica, "Disk-Locality in Datacenter Computing Considered Irrelevant", HotOS 2011

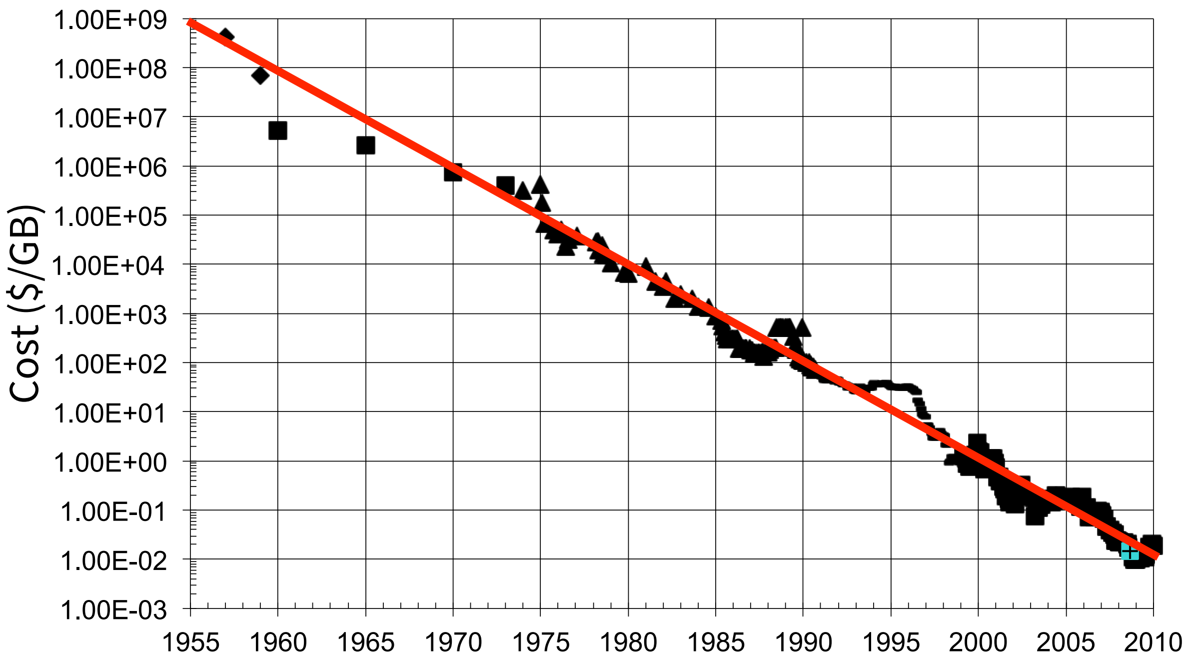
2009: Application Trends

Memory (GB)	Facebook (% jobs)	Microsoft (% jobs)	Yahoo! (% jobs)
8	69	38	66
16	74	51	81
32	96	82	97.5
64	**97**	**98**	**99.5**
128	98.8	99.4	99.8
192	99.5	100	100
256	99.6	100	100

*G Ananthanarayanan, A. Ghodsi, S. Shenker, I. Stoica, "Disk-Locality in Datacenter Computing Considered Irrelevant", HotOS 2011

2009: Hardware Trends

Memory still riding the Moore's law



http://www.jcmit.com/memoryprice.htm

2009: Hardware Trends

Memory still riding the Moore's law

I/O throughput and latency stagnant

- HDD dominating data clusters as storage of choice
- Many deployments as low as 20MB/sec per drive

Applications

Requirements:
- ad-hoc queries
- ML algos

Enabler:
- working sets fit in memory

Spark

In-memory processing

Multi-stage BSP model

Memory growing with Moore's Law

I/O performance stagnant (HDDs)

Hardware

2009

2009: Our Solution: Apache Spark

In-memory processing

- Great for ad-hoc queries

Generalizes MapReduce to multi-stage computations

- Implement BSP model

Share data between stages via memory

- Great for iterative computations, e.g., ML algorithms

2009: Technical Solutions

Low-overhead resilience mechanisms → Resilient Distributed Datasets (RDDs)

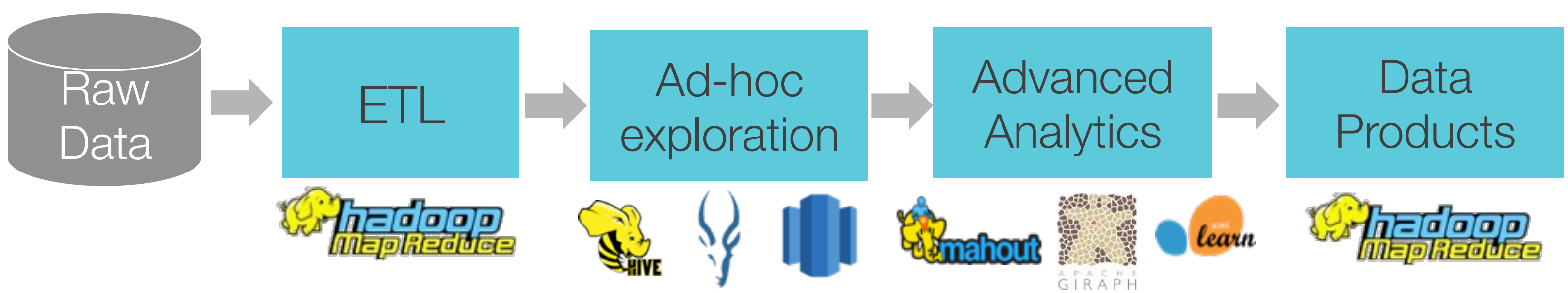
Efficiently support for ML algos → Powerful and flexible APIs

- map/reduce just two of over 80+ APIs

2012

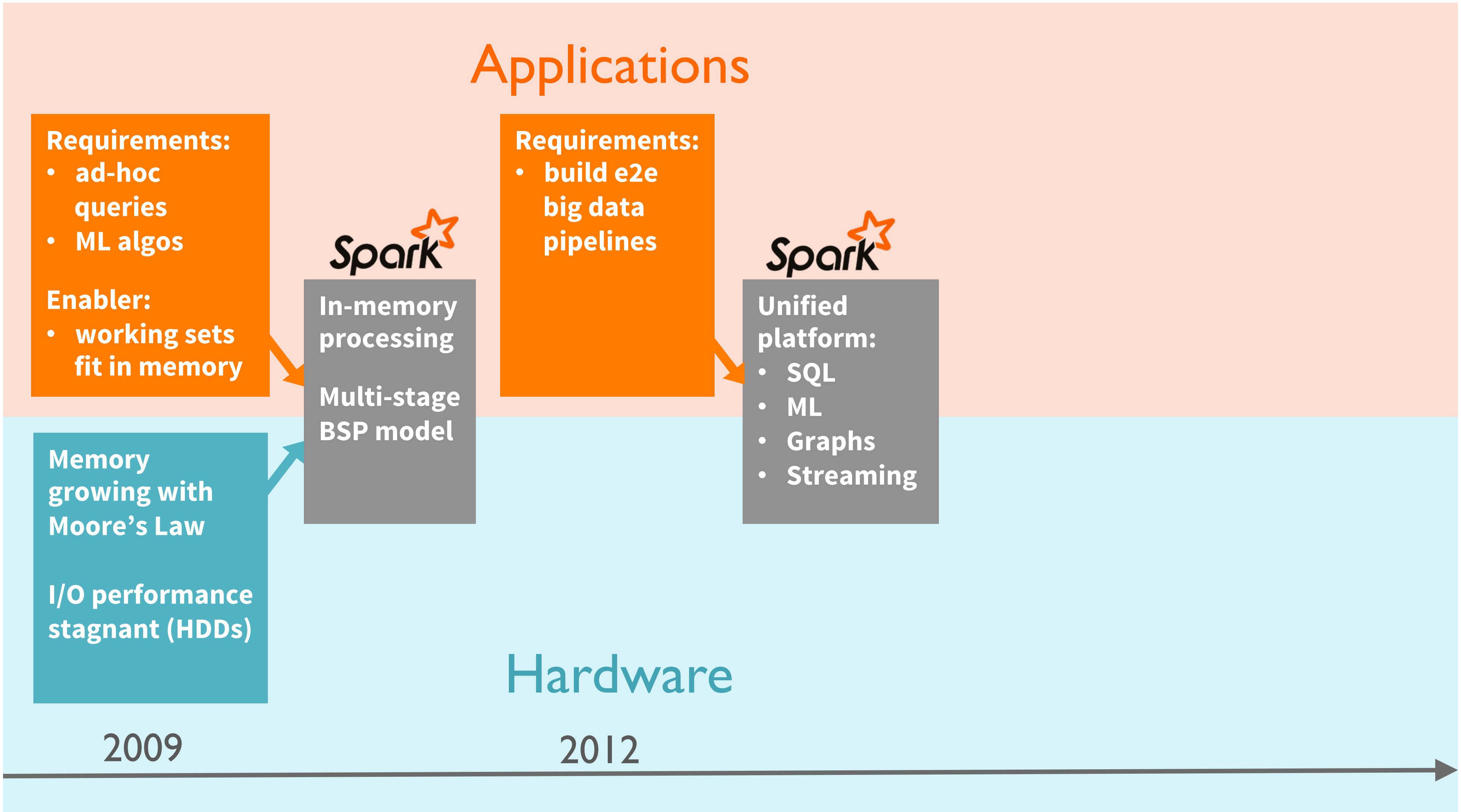
2012: Application Trends

People started to assemble e2e data analytics pipelines



Need to stitch together a hodgepodge of systems

- Difficult to manage, learn, and use

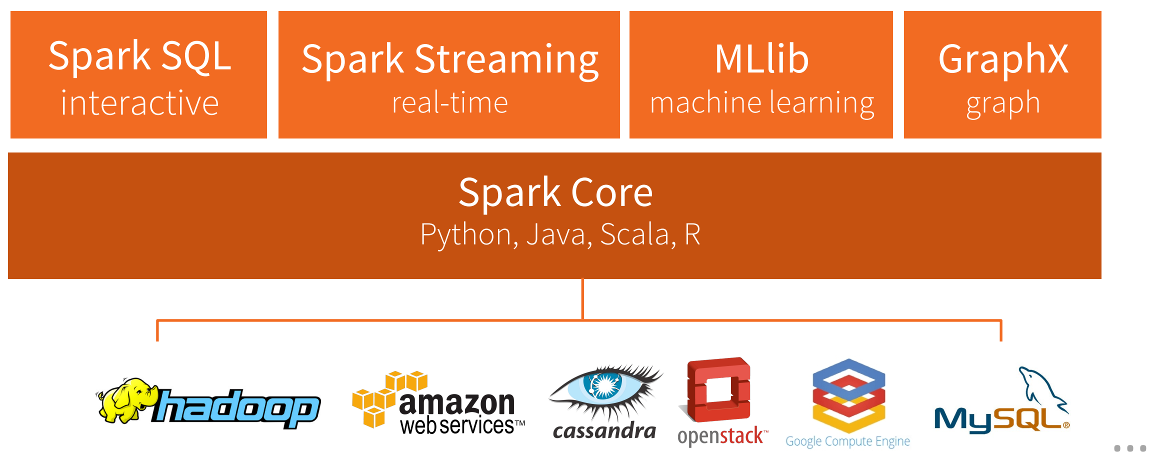

Applications
Requirements:
• ad-hoc queries
• ML algos
Enabler:
• working sets fit in memory
Spark
In-memory processing
Multi-stage BSP model
Requirements:
• build e2e big data pipelines
Spark
Unified platform:
• SQL
• ML
• Graphs
• Streaming
Memory growing with Moore's Law
I/O performance stagnant (HDDs)
Hardware
2009
2012

2012: Our Solution: Unified Platform

Support a variety of workloads

Support a variety of input sources

Provide a variety of language bindings



2014

2014: Application Trends

New users, new requirements

Spark early adopters

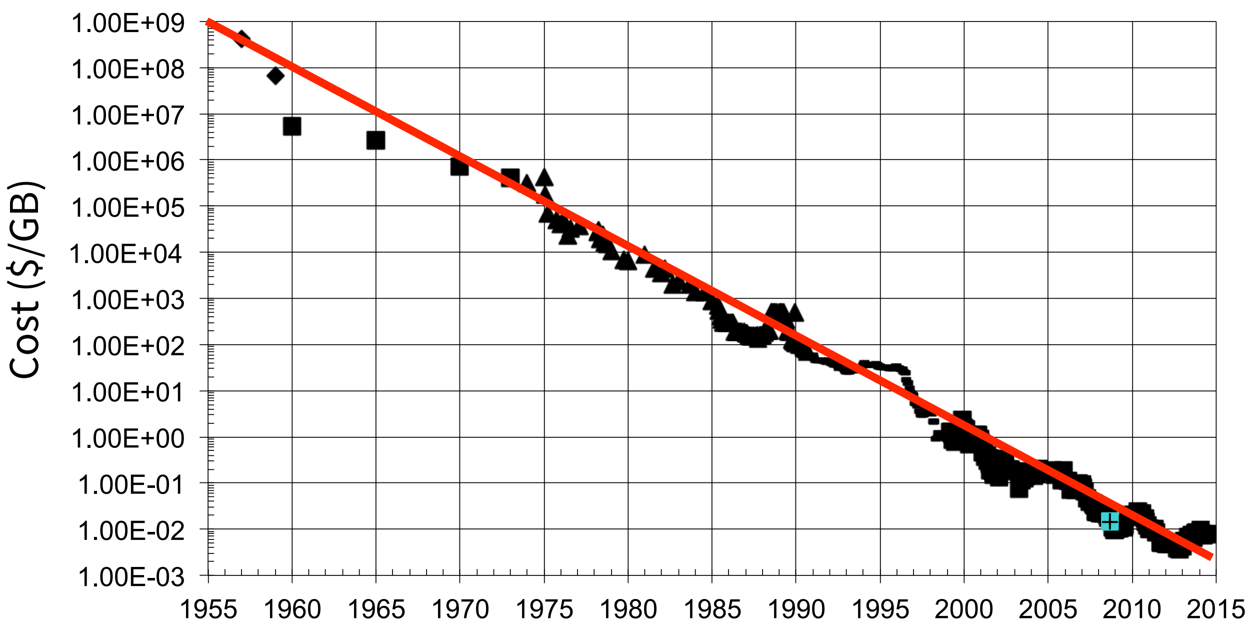
hadoop Users

Understands
MapReduce
& functional APIs

Data Engineers
Data Scientists
Statisticians
R users
PyData …

2014: Hardware Trends

Memory capacity still growing fast



http://www.jcmit.com/memoryprice.htm

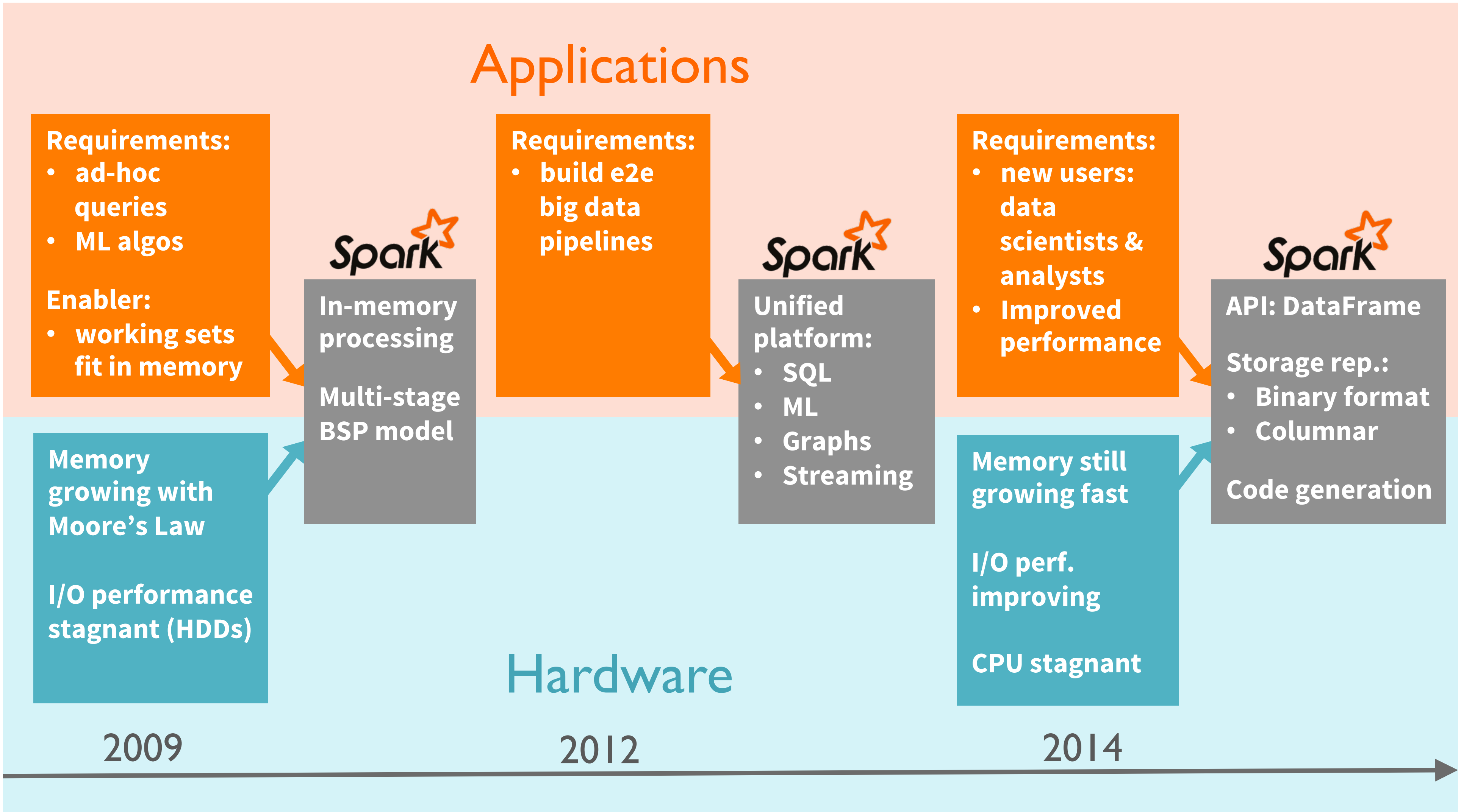
2014: Hardware Trends

Memory capacity still growing fast

Many clusters and datacenters transitioning to SSDs

- Orders of magnitude improvements in I/O and latency
- DigitalOcean: SSD only instances since 2013

CPU performance growth slowing down


Applications
Requirements:
• ad-hoc queries
• ML algos
Enabler:
• working sets fit in memory
In-memory processing
Multi-stage BSP model
Requirements:
• build e2e big data pipelines
Unified platform:
• SQL
• ML
• Graphs
• Streaming
Requirements:
• new users: data scientists & analysts
• Improved performance
API: DataFrame
Storage rep.:
• Binary format
• Columnar
Code generation
Memory growing with Moore's Law
I/O performance stagnant (HDDs)
Hardware
Memory still growing fast
I/O perf. improving
CPU stagnant
2009
2012
2014

RDD

```
pdata.map(lambda x: (x.dept, [x.age, 1])) \
     .reduceByKey(lambda x, y: [x[0] + y[0], x[1] + y[1]]) \
     .map(lambda x: [x[0], x[1][0] / x[1][1]]) \
     .collect()
```

DataFrame

```
data.groupBy(“dept”).avg(“age”)
```

DataFrame API

DataFrame logically equivalent to a relational table

Operators mostly relational with additional ones for statistical analysis, e.g., quantile, std, skew

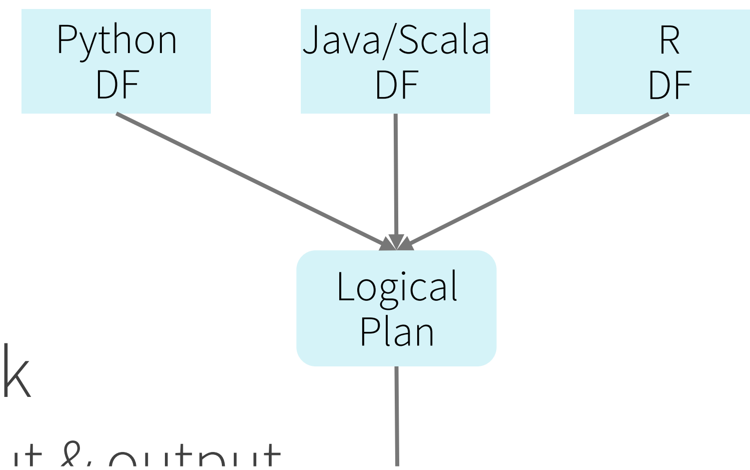
Popularized by R and Python/pandas, languages of choice for Data Scientists

DataFrames in Spark

Make DataFrame declarative, unify DataFrame and SQL

DataFrame and SQL share same

- query optimizer, and
- execution engine

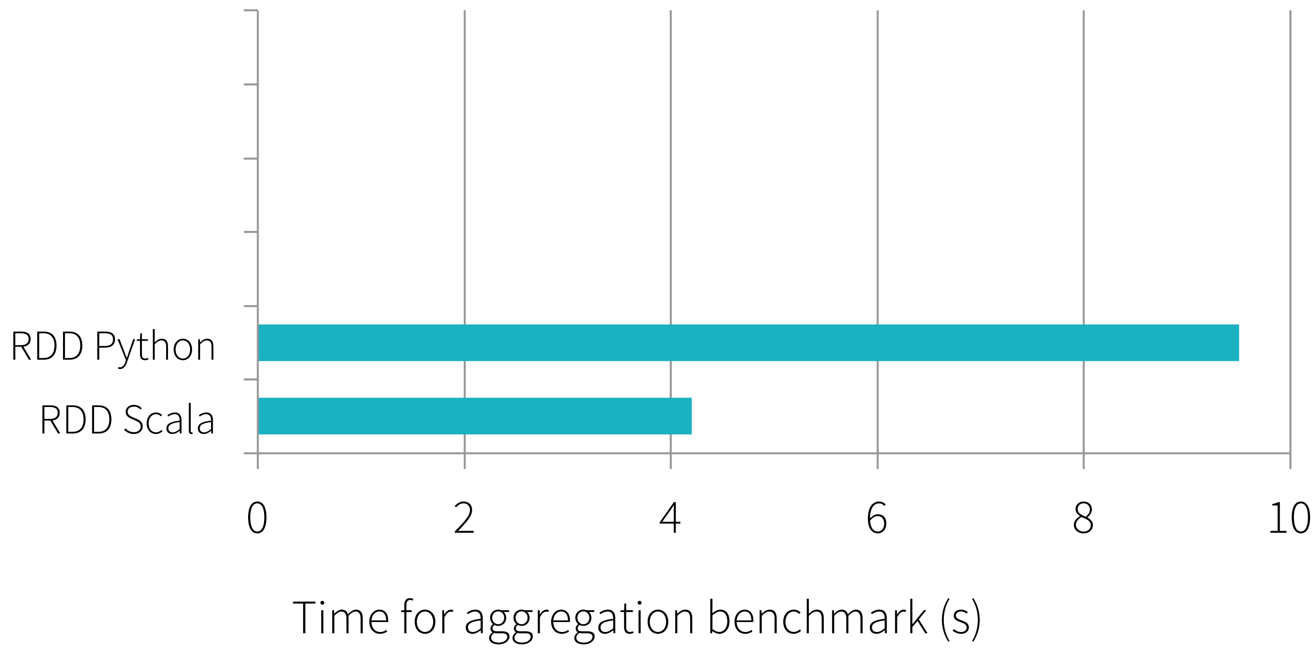


Tightly integrated with rest of Spark

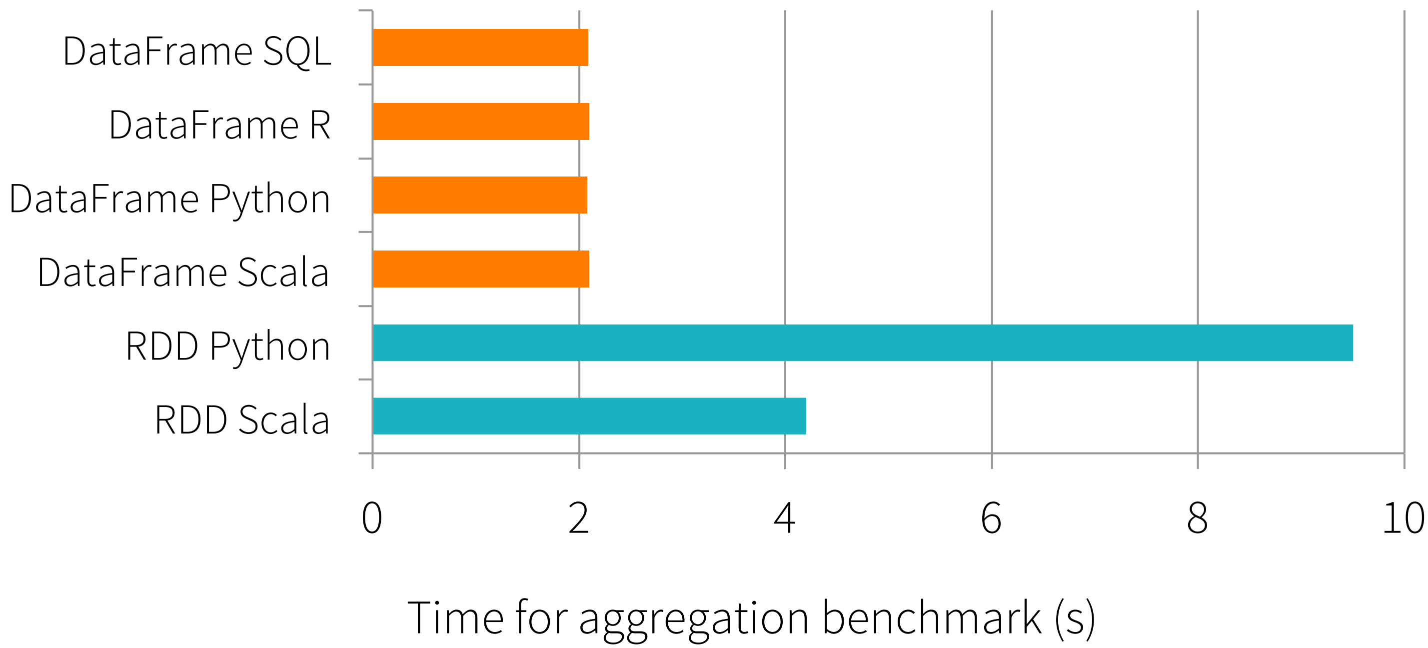
- ML library takes DataFrames as input & output
- E

Every optimizations automatically applies to SQL, and Scala, Python and R DataFrames

One Query Plan, One Execution Engine



One Query Plan, One Execution Engine



What else does DataFrame enable?

Typical DB optimizations across operators:

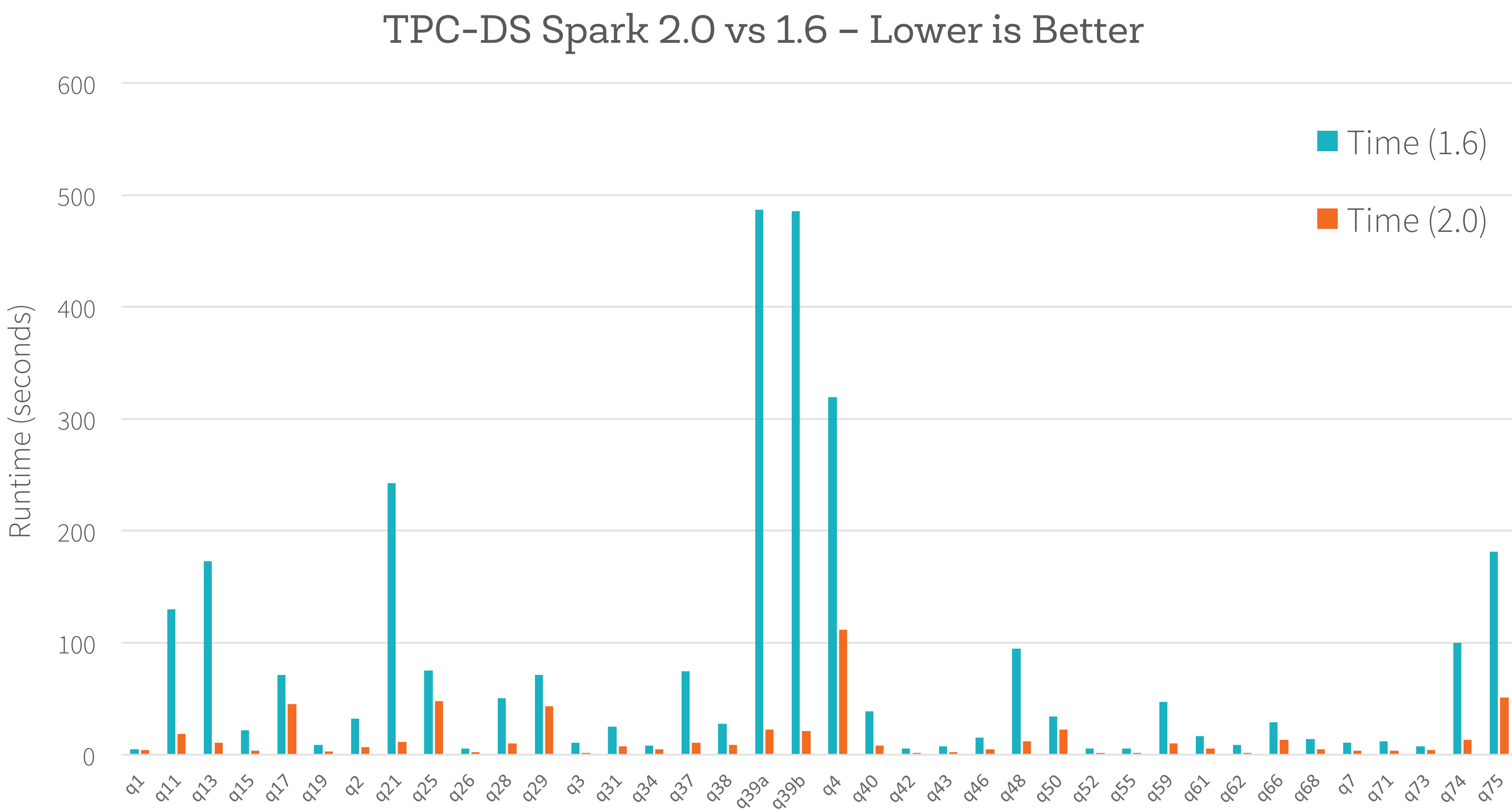
- Join reordering, pushdown, etc

Compact binary representation:

- Columnar, compressed format for caching

Whole-stage code generation:

- Remove expensive iterator calls
- Fuse across multiple operators



TPC-DS Spark 2.0 vs 1.6 – Lower is Better
Runtime (seconds)
600
500
400
300
200
100
0
Time (1.6)
Time (2.0)
q1
q11
q13
q15
q17
q19
q2
q21
q25
q26
q28
q29
q3
q31
q34
q37
q38
q39a
q39b
q4
q40
q42
q43
q46
q48
q50
q52
q55
q59
q61
q62
q66
q68
q7
q71
q73
q74
q75

2016
(What's Next?)

What's Next?

Application trends

Hardware trends

Challenges and techniques

Application Trends

Data only as valuable as the decisions and actions it enables

What does it mean?

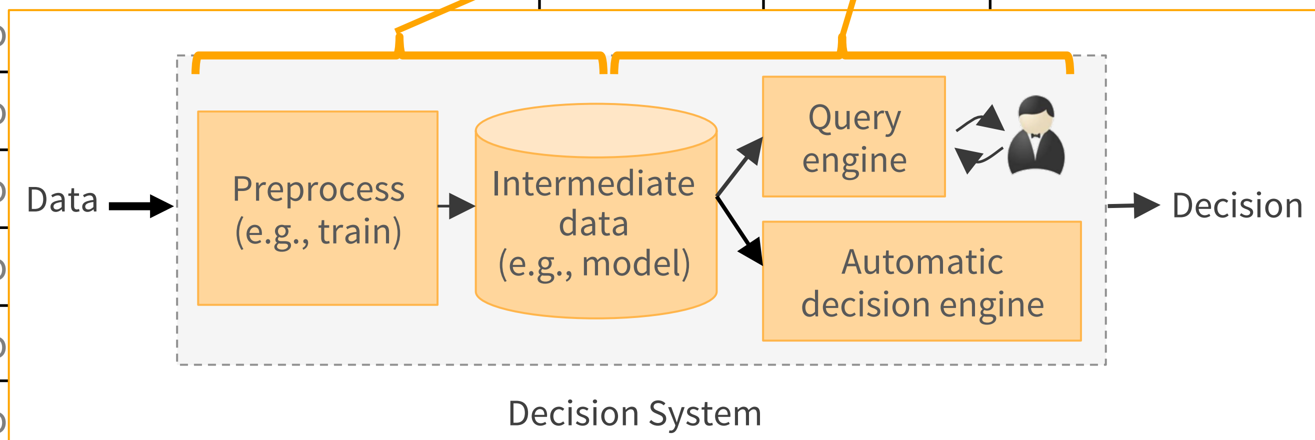
- Faster decisions better than slower decisions
- Decisions on fresh data better than on stale data
- Decisions on personal data better than on aggregate data

Application Trends

Real-time decisions	*decide in ms*
on live data	*the current state of the environment*
with strong security	*privacy, confidentiality, integrity*

Applications	Quality	Latency		Security
		Update	Decision	
Zero-time defense	sophisticated, accurate, robust	sec	sec	privacy, integrity
Parking assistant	sophisticated, robust	sec	sec	privacy
Disease discovery	sophisticated, accurate	hours	sec/min	privacy, integrity
IoT (smart buildings)	sophisticated, robust	min/hour	sec	privacy, integrity
Earthquake warning	sophisticated, accurate, robust	min	ms	integrity
Chip manufacturing	sophisticated, accurate, robust	min	sec/min	confidentiality, integrity
Fraud detection	sophisticated, accurate	min	ms	privacy, integrity
“Fleet” driving	sophisticated, accurate, robust	sec	sec	privacy, integrity
Virtual assistants	sophisticated, robust	min/hour	sec	integrity
Video QoS at scale	sophisticated	min	ms/sec	privacy, integrity

Applications	Quality	Latency		Security
		Update	Decision	
Zero-time defense	sophisticated, accurate, robust	sec	sec	privacy, integrity
Parking assistant	so			
Disease discovery	so			
IoT (smart buildings)	so			
Earthquake warning	so			
Chip manufacturing	so			tegrity
Fraud detection	so			
“Fleet” driving	sophisticated, accurate, robust	sec	sec	privacy, integrity
Virtual assistants	sophisticated, robust	min/hour	sec	integrity
Video QoS at scale	sophisticated	min	ms/sec	privacy, integrity



Applications	Quality	Latency		Security
		Update	Decision	
Zero-time defense	sophisticated, accurate, robust	sec	sec	privacy, integrity
Parking assistant	sophisticated, robust	sec	sec	privacy
Disease discovery	sophisticated, accurate	hours	sec/min	privacy, integrity
IoT (smart buildings)	sophisticated, robust	min/hour	sec	privacy, integrity
Earthquake warning	sophisticated, accurate, robust	min	ms	integrity
Chip manufacturing	sophisticated, accurate, robust	min	sec/min	confidentiality, integrity
Fraud detection	sophisticated, accurate	min	ms	privacy, integrity
"Fleet" driving	sophisticated, accurate, robust	sec	sec	privacy, integrity

Addressing these challenges, the goal of next Berkeley lab: RISE (Real-time Secure Execution) Lab

What's next?

Application trends

Hardware trends

Challenges and techniques

Moore's Law is Slowing Down

MIT Technology Review

Computing

Intel Puts the Brakes on Moore's Law

Intel will slow the pace at which it rolls out new chip-making technology, and is still searching for a successor to silicon transistors.

by Tom Simonite March 23, 2016

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive > Volume 530 > Issue 7589 > News Feature > Article

NATURE | NEWS FEATURE

عربي

The chips are down for Moore's law

The semiconductor industry will soon abandon its pursuit of Moore's law. Now things could get a lot more interesting.

E

TECHNOLOGY QUARTERLY
AFTER MOORE'S LAW

Double, double, toil and trouble

What Does It Mean?

CPUs affected most: just 20-30%/year perf. improvements

- More complex layouts → harder to scale
- Mostly by increasing number of cores → harder to take advantage

Memory: still grows at 30-40%/year

- Regular layouts, stacked technologies

Network: grows at 30-50%/year

- 100/200/400GBpE NICs at horizon
- Full-bisection bandwidth network topologies

CPUs is the bottleneck and it's getting worse!

What Does It Mean?

CPUs affected most: just 20-30%/year perf. improvements

- More complex layouts → harder to scale
- Mostly by increasing number of cores → harder to take advantage

Memory: still grows at 30-40%/year

- Regular layouts, stacked technologies

Network: grows at 30-50%/year

- 100/200/400GBpE NICs at horizon

Memory-to-core ratio is increasing

e.g., AWS: 7-8GB/vcore → 17GB/vcore (X1)

Unprecedented Hardware Innovation

From CPU to specialized chips:

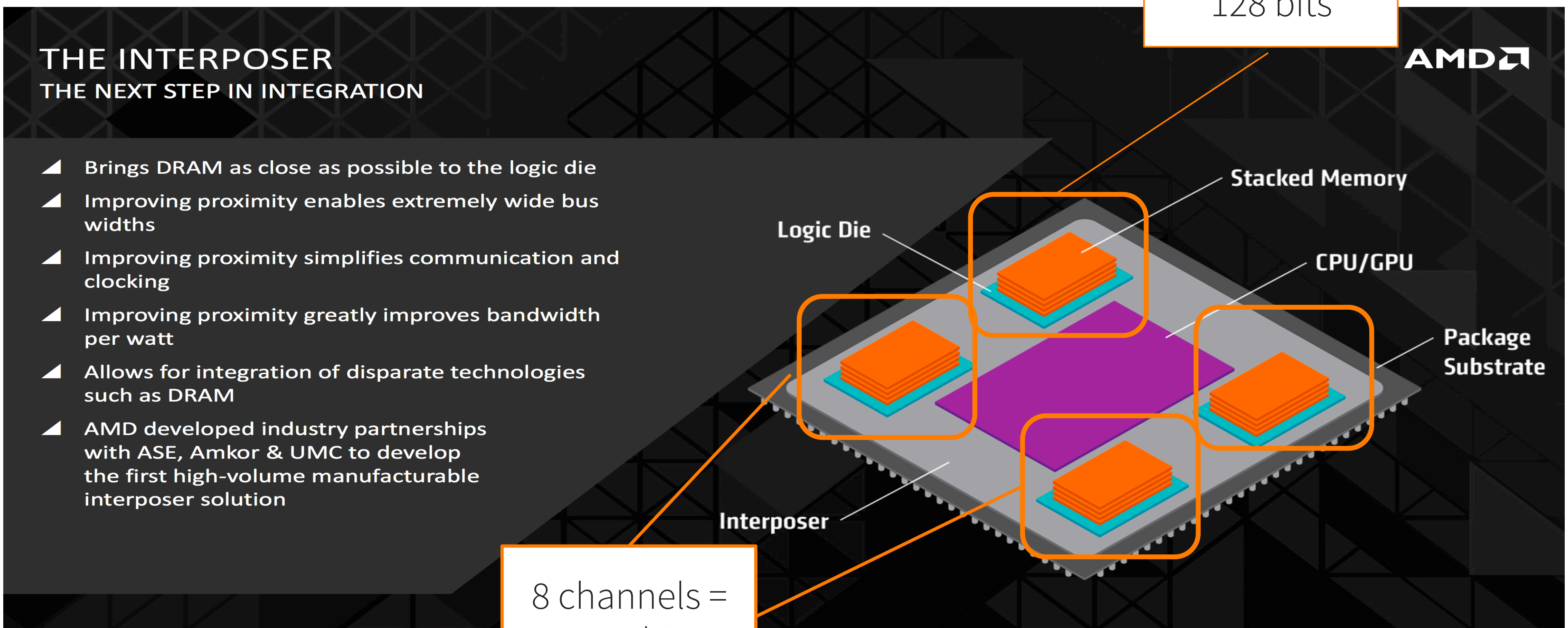
- GPUs, FPGAs, ASICs/co-processors (e.g., TPU)
- Tightly integrated, e.g., Intel's latest Xeon integrates CPU & FPGA

New, disruptive memory technologies

- HBM (High Bandwidth Memory), same package at CPU

High Bandwidth Memory (HBM)

2 channels @ 128 bits

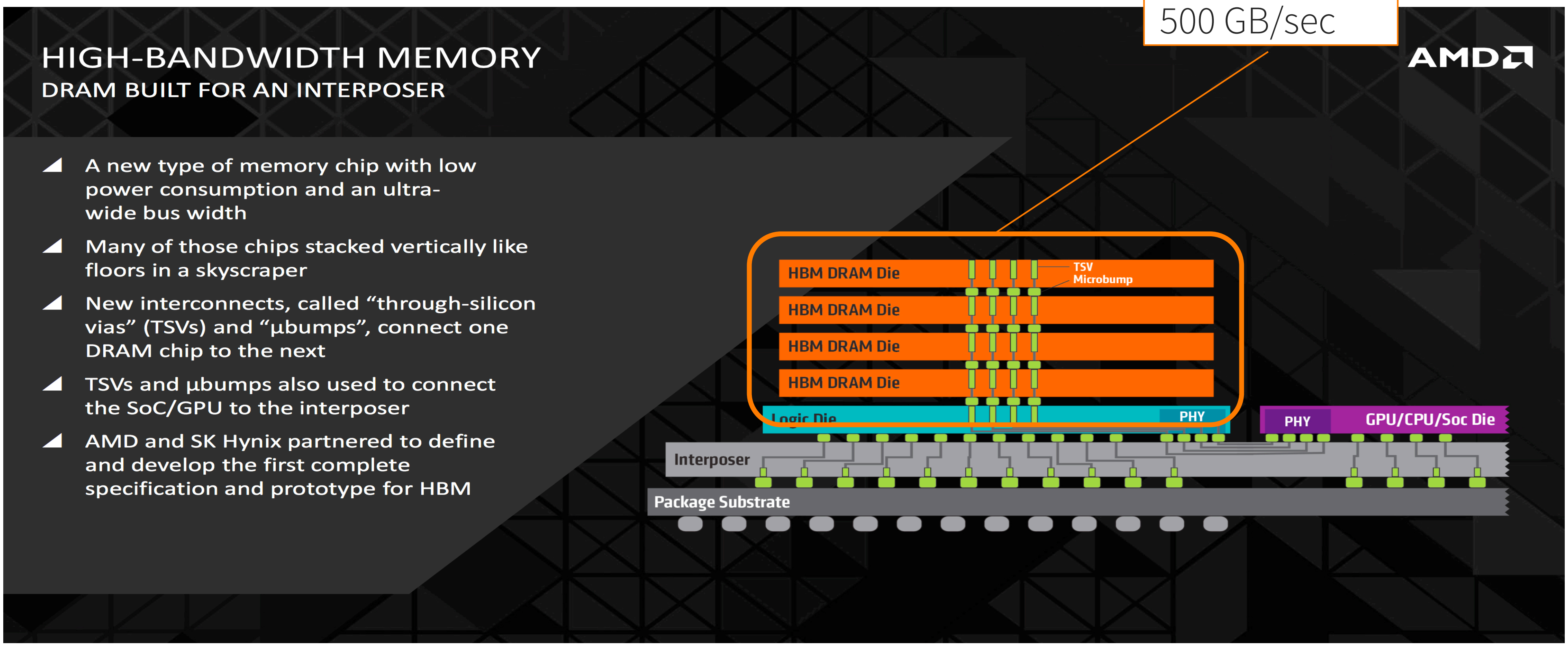


8 channels = 1024 bits

nnovations/software-technologies/hbm

High Bandwidth Memory (HBM)

8 stacks =
4096 bits →
500 GB/sec



http://www.amd.com/en-us/innovations/software-technologies/hbm

Unprecedented Hardware Innovation

From CPU to specialized chips:

- GPUs, FPGAs, ASICs/co-processors (e.g., TPU)
- Tightly integrated (e.g., Intel's latest Xeon integrates CPU & FPGA)

New, disruptive memory technologies

- HBM2: 8 DRAM chips/package → 1TB/sec

Unprecedented Hardware Innovation

From CPU to specialized chips:

- GPUs, FPGAs, ASICs/co-processors (e.g., TPU)
- Tightly integrated (e.g., Intel's latest Xeon integrates CPU & FPGA)

New, disruptive memory technologies

- HBM2: 8 DRAM chips/package → 1TB/sec
- 3D XPoint

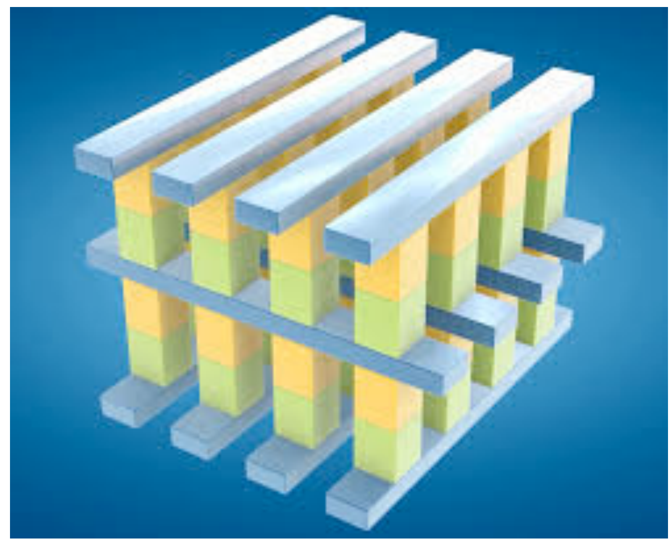
3D XPoint Technology

Developed by Intel and Micron

- Announced last year; products released this year

Characteristics:

- Non-volatile memory
- 2-5x DRAM latency!
- 8-10x density of DRAM
- 1000x more resilient than SSDs



Unprecedented Hardware Innovation

From CPU to specialized chips:

- GPUs, FPGAs, ASICs/co-processors (e.g., TPU)
- Tightly integrated (e.g., Intel's latest Xeon integrates CPU & FPGA)

New, disruptive memory technologies

- HBM2: 8 DRAM chips/package → 1TB/sec
- 3D XPoint

"Renaissance of hardware design" – David Patterson

Applications

Requirements:
- ad-hoc queries
- ML algos

Enabler:
- working sets fit in memory

Spark

In-memory processing

Multi-stage BSP model

Requirements:
- build e2e big data pipelines

Spark

Unified platform:
- SQL
- ML
- Graphs
- Streaming

Requirements:
- Real-time decisions on fresh data
- Strong security

Spark

?

Memory growing with Moore's Law

I/O performance stagnant (HDDs)

Memory rapidly evolving

Specialized processing:
- GPUs, FPGAs, ASICs, SGX, 100s core CPUs

Hardware

2009

2012

2016

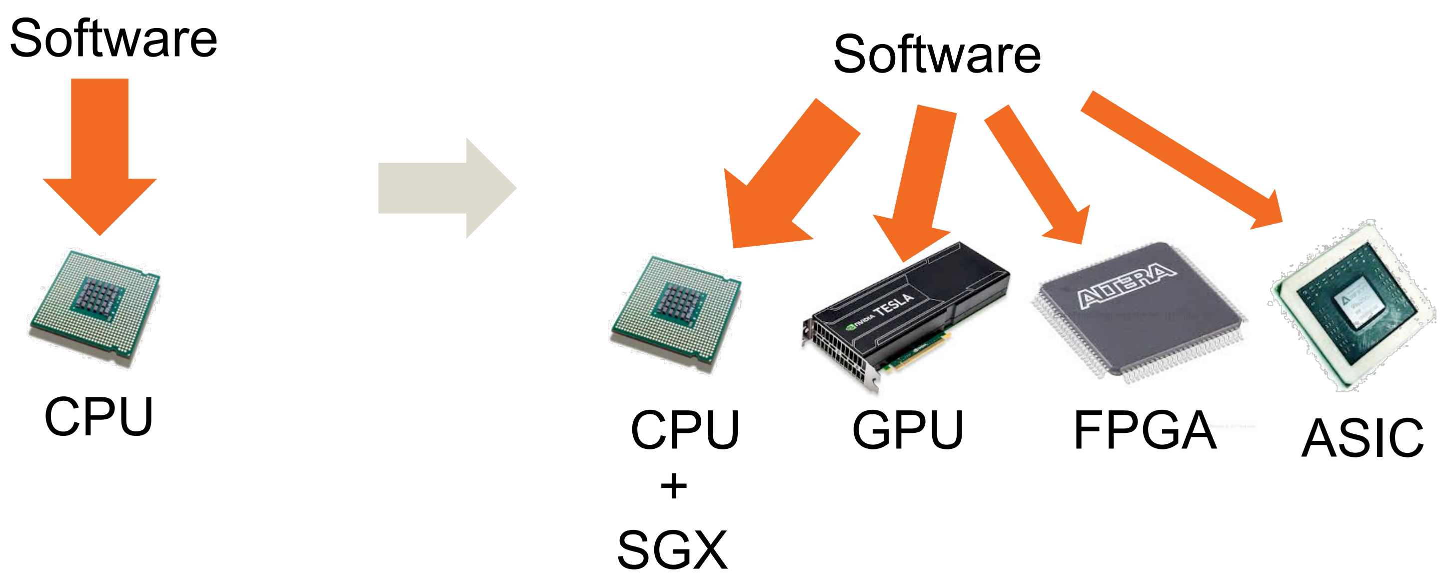
What's next?

Application trends

Hardware trends

Challenges and techniques

Complexity – Computation



Complexity – Memory

2015

L1/L2 cache	~1 ns
L3 cache	~10 ns
Main memory	~100 ns / ~80 GB/s / ~100GB
NAND SSD	~100 usec / ~10 GB/s / ~1 TB
Fast HHD	~10 msec / ~100 MB/s / ~10 TB

2020

L1/L2 cache	~1 ns
L3 cache	~10 ns
HBM	~10 ns / ~1TB/s / ~10GB
Main memory	~100 ns / ~80 GB/s / ~100GB
NVM (3D Xpoint)	~1 usec / ~10GB/s / ~1TB
NAND SSD	~100 usec / ~10 GB/s / ~10 TB
Fast HHD	~10 msec / ~100 MB/s / ~100 TB

Complexity – More and More Choices

t2.nano, t2.micro, t2.small
m4.large, m4.xlarge, m4.2xlarge,
m4.4xlarge, m3.medium,
c4.large, c4.xlarge, c4.2xlarge,
c3.large, c3.xlarge, c3.4xlarge,
r3.large, r3.xlarge, r3.4xlarge,
i2.2xlarge, i2.4xlarge, d2.xlarge
d2.2xlarge, d2.4xlarge,…

Amazon EC2

Basic tier: A0, A1, A2, A3, A4
Optimized Compute : D1, D2,
D3, D4, D11, D12, D13
D1v2, D2v2, D3v2, D11v2,…
Latest CPUs: G1, G2, G3, …
Network Optimized: A8, A9
Compute Intensive: A10, A11,…

Microsoft AZURE

n1-standard-1, ns1-standard-2,
ns1-standard-4, ns1-standard-8,
ns1-standard-16, ns1highmem-2,
ns1-highmem-4, ns1-highmem-8,
n1-highcpu-2, n1-highcpu-4, n1-
highcpu-8, n1-highcpu-16, n1-
highcpu-32, f1-micro, g1-small…

Google Cloud Engine

Complexity – More and More Constraints

Latency

Accuracy

Cost

Security

Techniques for Conquering Complexity

Use additional choices to simplify!

Expose and control tradeoffs

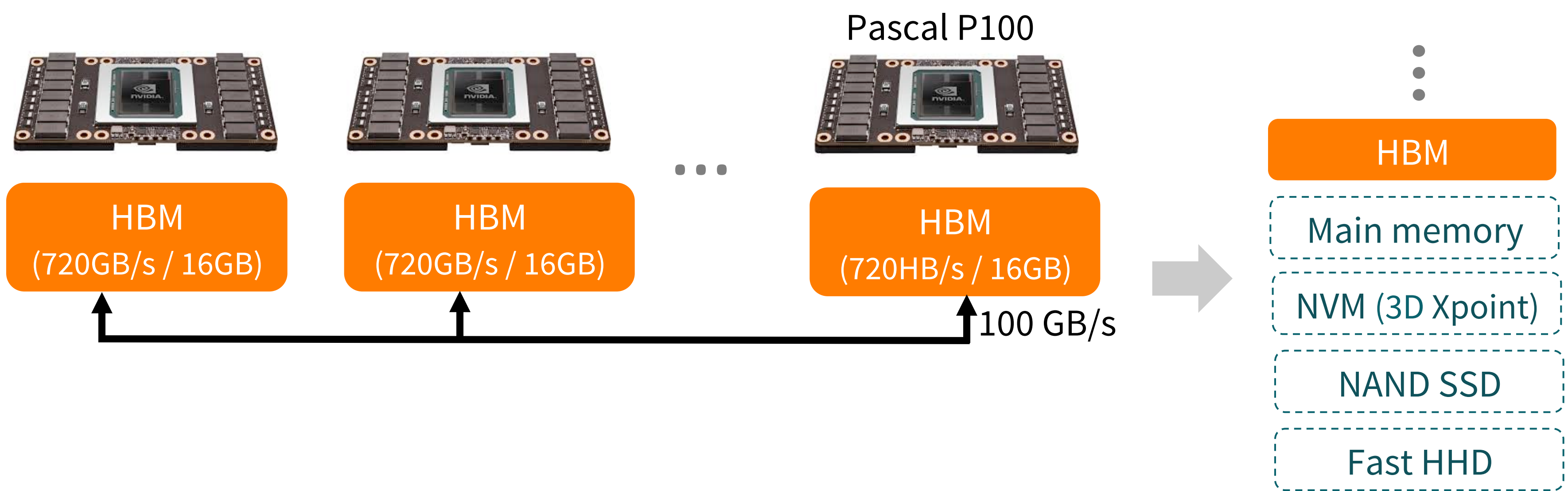
Don’t forget “tried & true” techniques

Use Choices to Simplify System Design

< 2010	2011-2014	> 2014
		
L1/L2 cache	L1/L2 cache	L1/L2 cache
L3 cache	L3 cache	L3 cache
Main memory	Main memory	Main memory
	NAND SSD	NAND SSD
Fast HHD	Fast HHD	Fast HHD

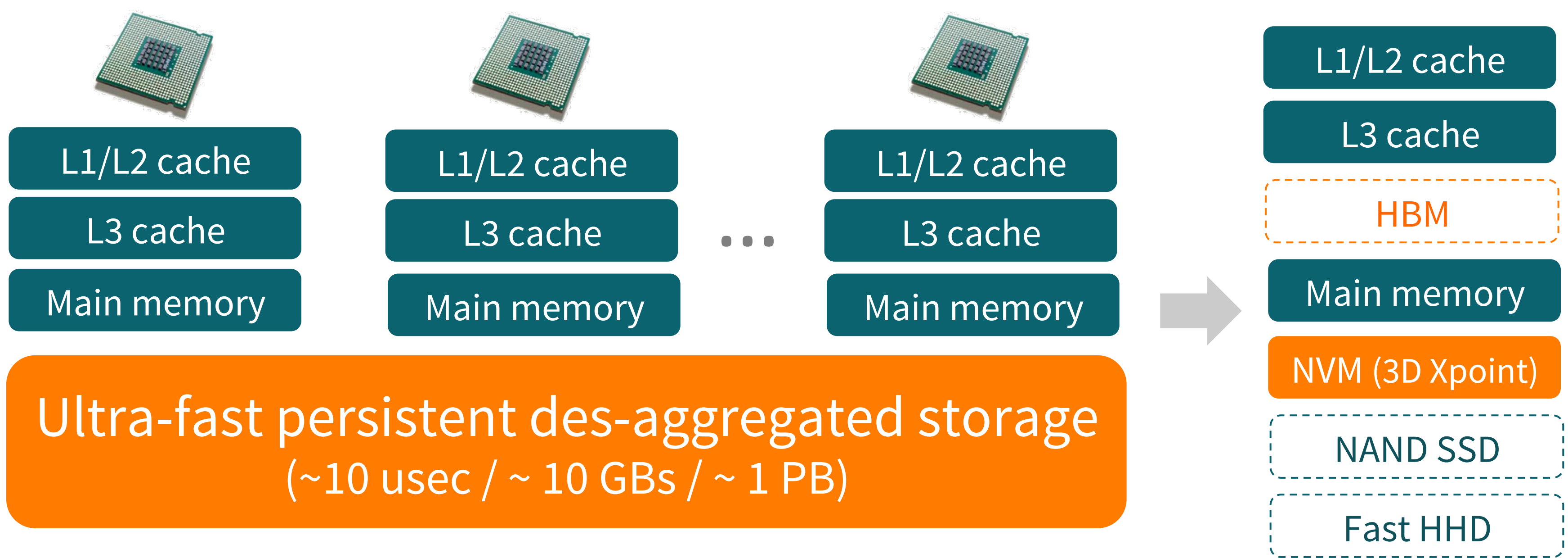
Use Choices to Simplify System Design

Example: NVIDIA DGX-1 supercomputer for Deep Learning



Use Choices to Simplify System Design

Possible datacenter architecture (e.g., FireBox, UC Berkeley)



Use Choices to Simplify App Design

Maybe no need to optimize every algorithm for every specialized processor…

… if run in cloud, just pick best instance types for your app!

Expose and Control Tradeoffs

Latency vs. accuracy

- Approximate query processing (e.g., BlinkDB)
- Ensembles and correction ML models (e.g., Clipper)

Job completion time vs. cost

- Predict response times given configuration (e.g., Earnest)

Security vs. latency vs. functionality

- E.g., CryptDB, Opaque

Expose and Control Tradeoffs

Caching vs. memory

- HBM allows to be configured either as cache or memory region

Declarative vs. procedural

- Enable users to pick specific query plans for complex declarative programs & complex environments

“Tried & True” Techniques

Sampling:

- Scheduling (e.g., Sparrow), querying (e.g., BlinkDB), storage (e.g., KMN)

Speculation:

- Replicate time-sensitive requests/jobs (e.g., Dolly)

Incremental updates:

- Storage (e.g., IndexedRDDs), and ML models (e.g., Clipper)

Cost-based optimization:

- Pick target hardware at run-time

Summary

Application and hardware trends often determine solution

We are at an inflection point both in terms of both apps and hardware trends

Many research opportunities

Be aware of “complexity”: use myriad of choices to simplify!

Thanks