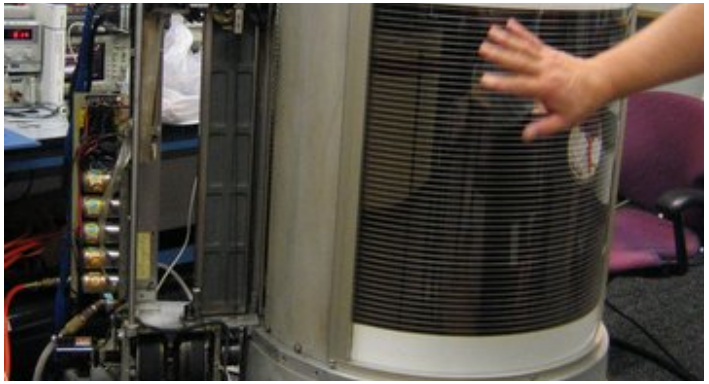
In-Situ Processing
Bringing Intelligence to Storage

Vladimir Alves, PhD

Kamyar Souri

NxGn Data

The Evolution of Storage



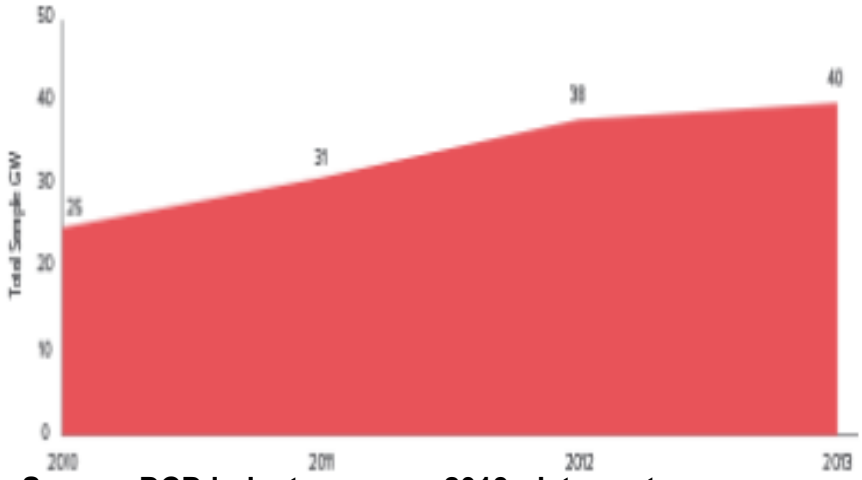
Since 1956 storage devices have been performing the same basic functions: read and write

The Evolution of Storage

Data volume growth: 7.7 zettabytes in 2017

Source: Cisco Global Cloud Index (2012-2017)

Energy consumption growth: 40GW in 2017



Source: DCD industry census 2013: data center power

Moving Computation to Data: Cheaper than Moving Data

- *A computation requested by an application is much more efficient if it is executed near the data it operates on [1]*
  - *minimizes network traffic*
  - *increases throughput & performance of the system*
  - *Example: HDFS [2] provides interfaces for applications to move computation closer to where data is located*
- Interesting anecdote: In their 2009 ground-breaking paper on Cloud Computing the authors from UC Berkeley showed that it is faster to overnight a modern hard drive than it is to transfer it's content over a network

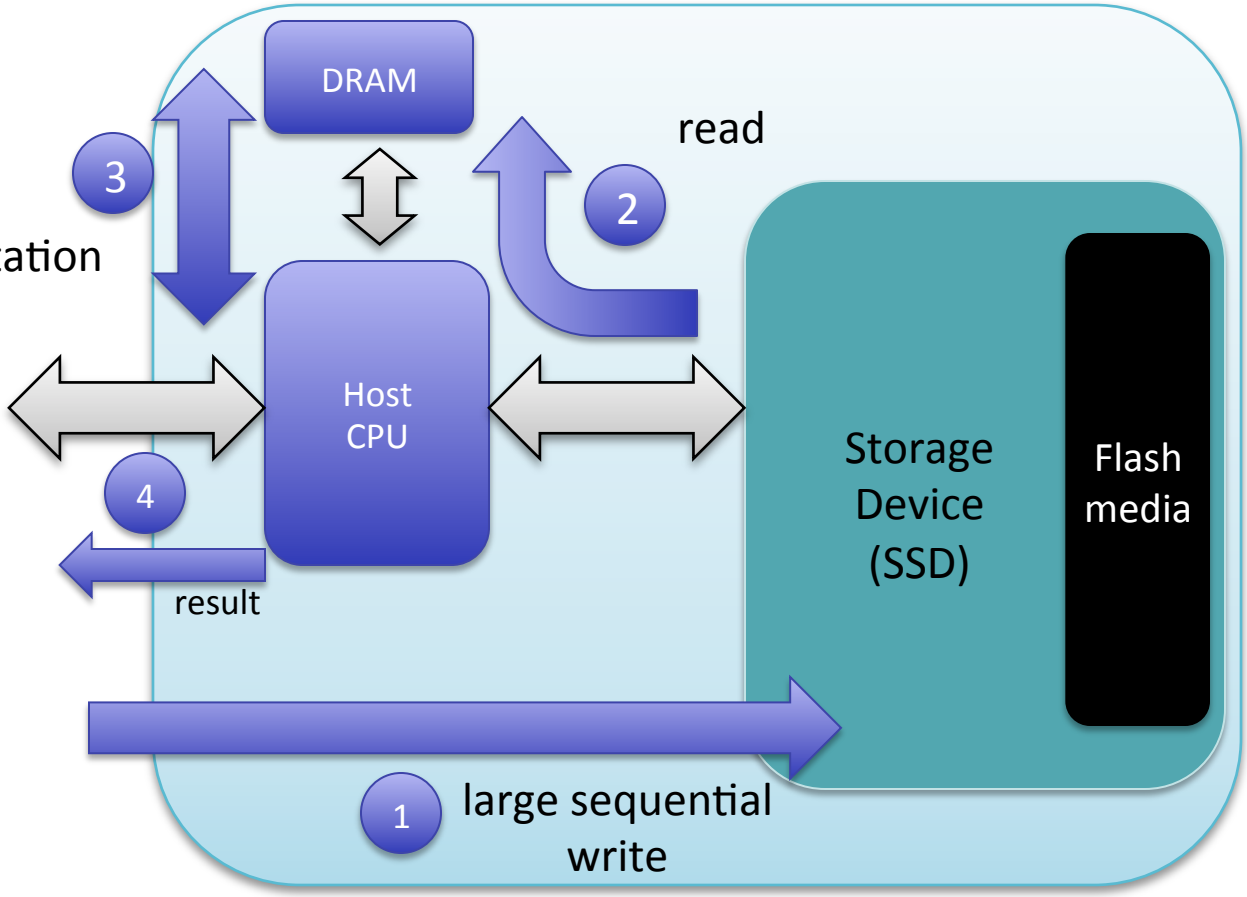
Moving Computation to Data (2)

- Moving computation to data is better than the other way around
- This is especially true for Big Data: large sets & unstructured data.
- Traditional approach: expensive and high-performance servers coupled with SAN/NAS storage
- Eventually limited by networking bottlenecks

Moving Computation to Data (3)

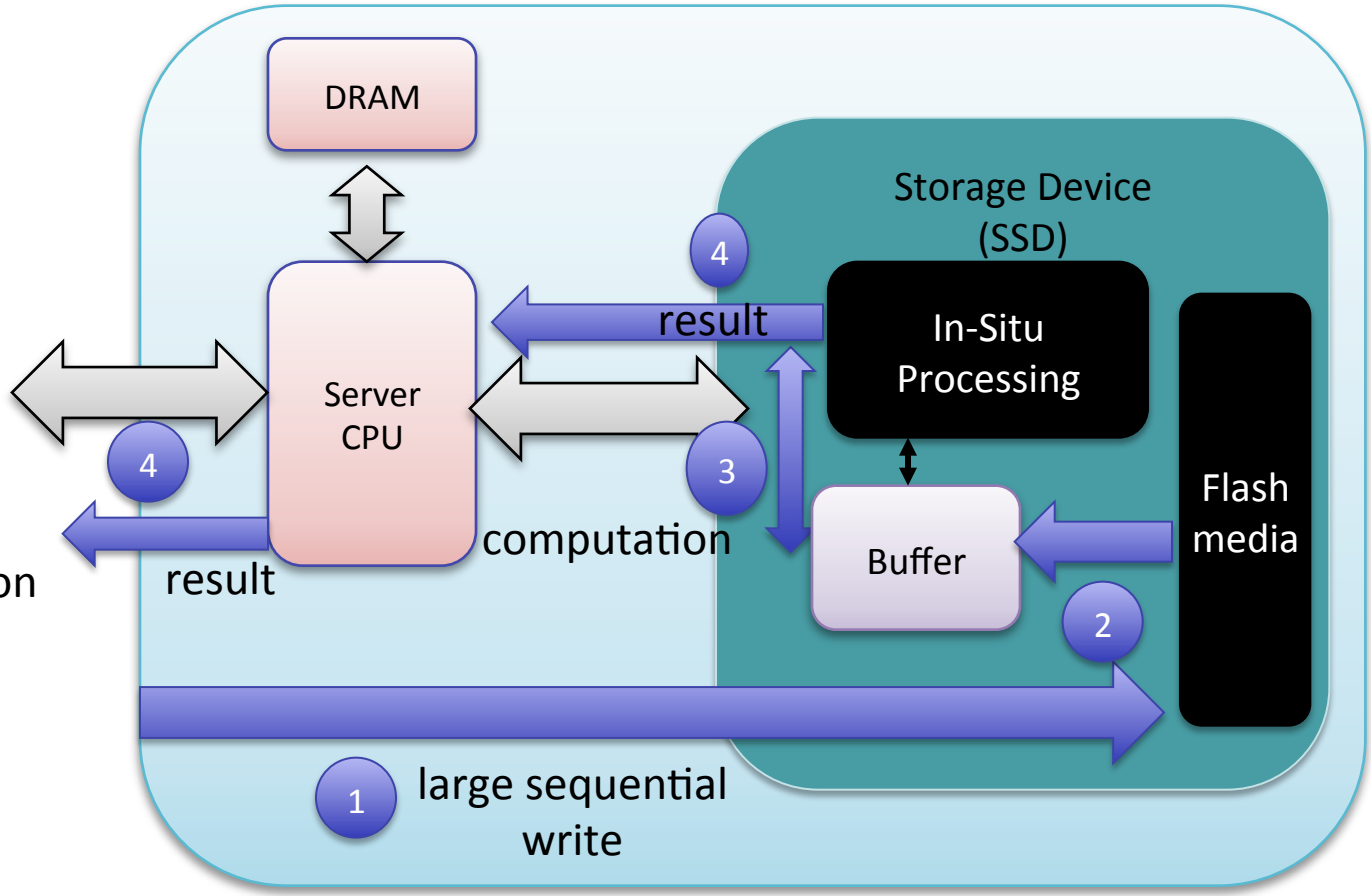
Steps 2 and 3 are repeated multiple times

Computation requires large power consumption



Moving Computation to Data (4)

Computation requires lower power consumption and can be performed more efficiently



NxGnData

Easier said than done…

The goal:

- Executing code within the storage device (in-situ)
- Seamless system level integration
- Minimum impact on application

The challenges:

- Protection against malicious code
- Fast runtime environment setup time
- Freedom in selecting programming language (C, Python, Perl…)
- Multi-tenant environments
  - Security and
  - Data isolation

Easier said than done…

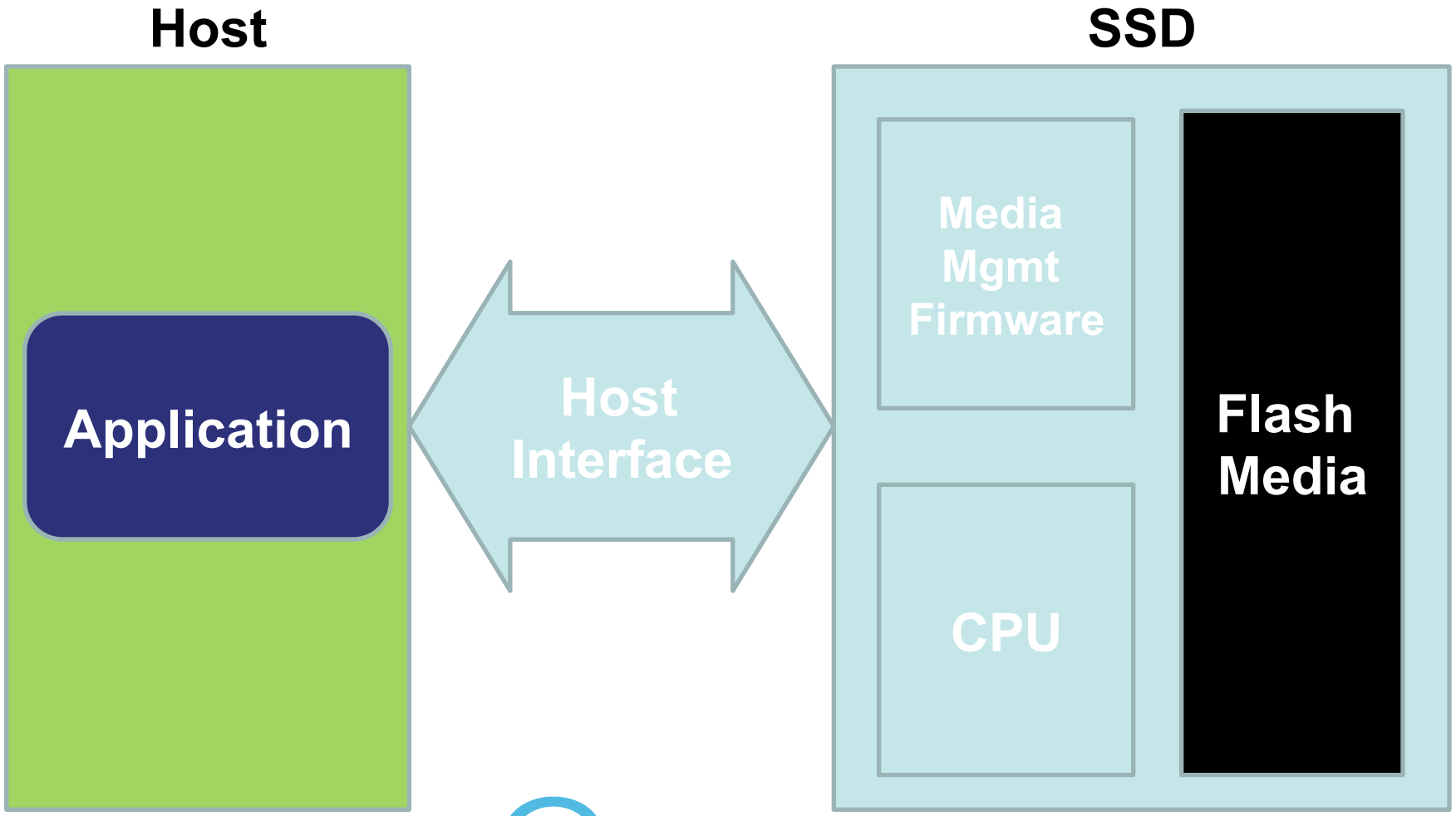
The goal:

- Executing code within the storage device (in-situ)
- Minimum impact on application
- Seamless system level integration

The challenges:

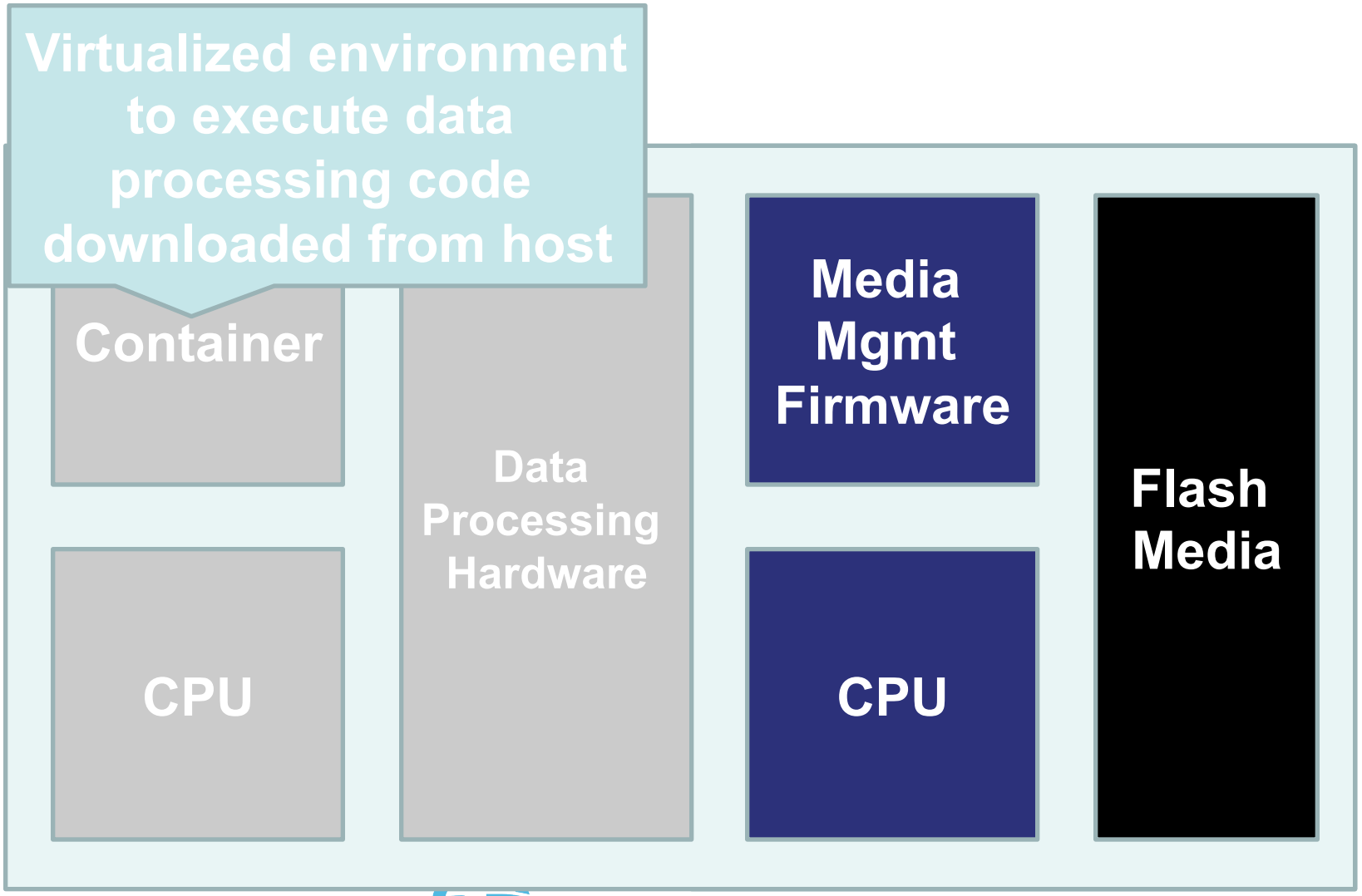
- Protection against malicious code
- Fast virtual environment set up
- Freedom in selecting programming language (C, Python, Perl…)
- Multi-tenant environments
  - Security and
  - Data isolation

Today's paradigm



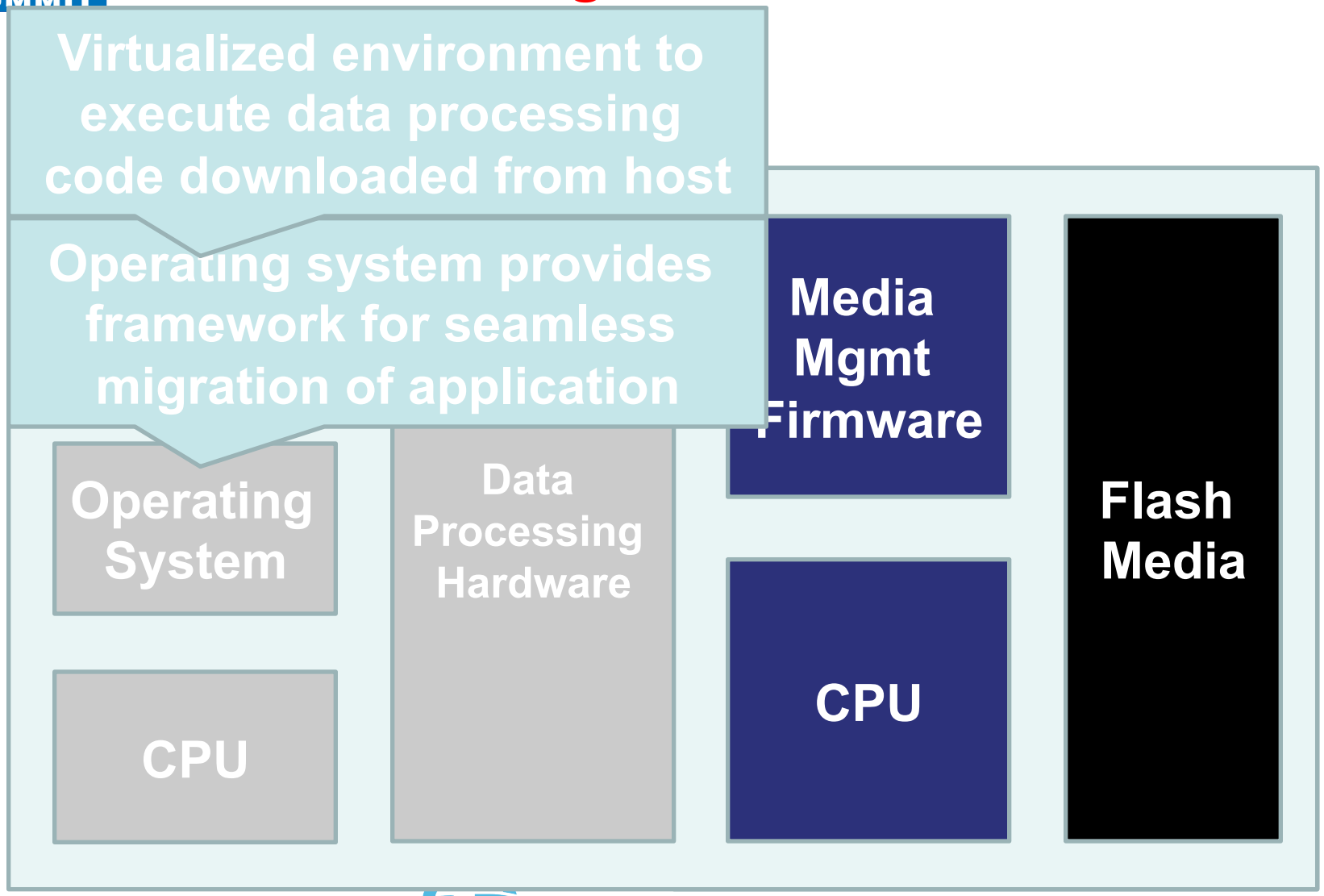
Flash Memory SUMMIT

In-Situ paradigm



NxGnData

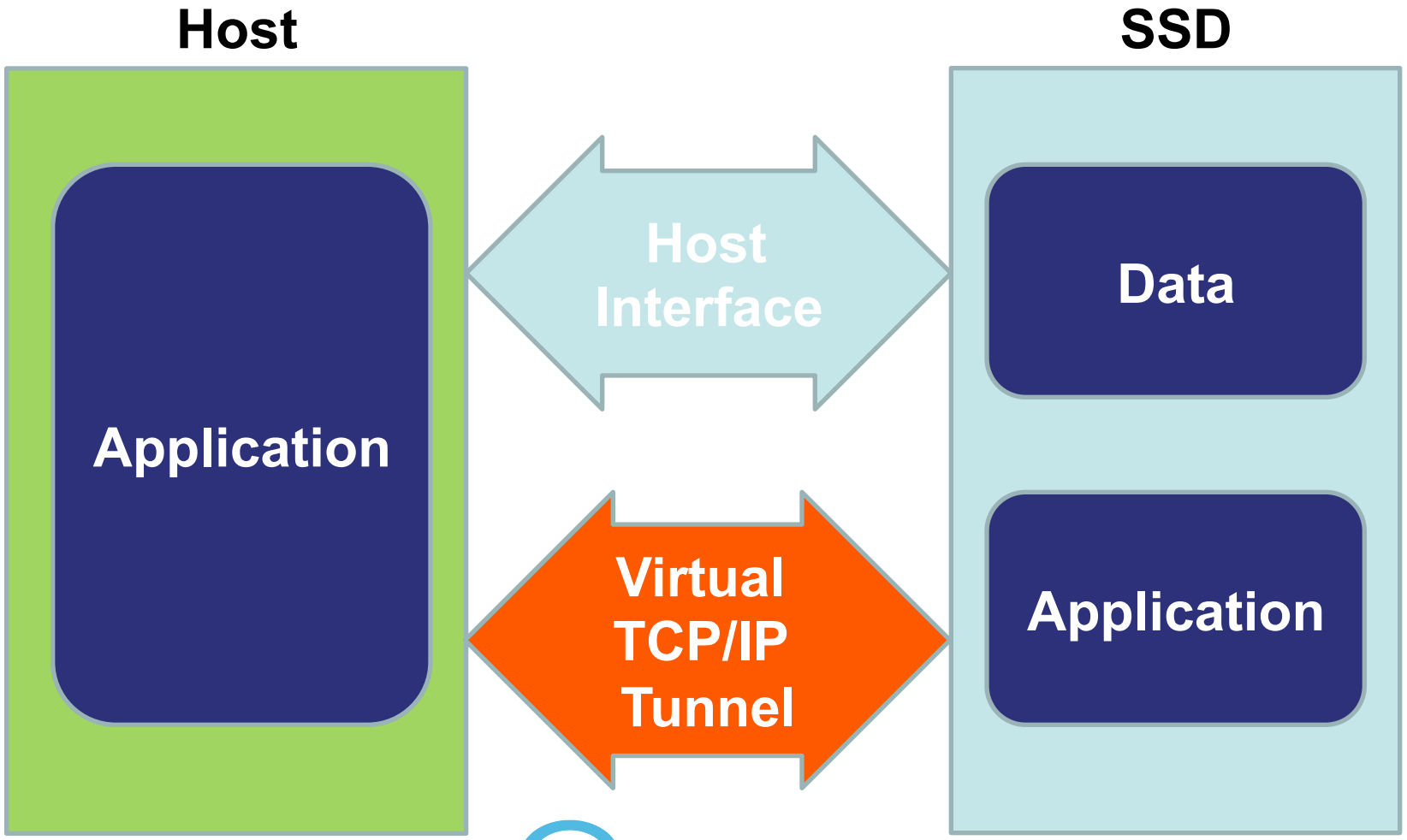
In-Situ Paradigm



Container's Role

- Creates a virtualized environment to execute data processing code
- A secure sand box which prevents the code from accessing other users code and data
- Prevents a malicious code to compromise system stability

Application communication path



Virtual TCP/IP Tunnel

- Running application on the drive requires some data/ code communication
- Many software technologies rely on TCP/IP as a communication path
- TCP/IP tunnel between host and device opens the door to many other data processing use cases

Flash Memory SUMMIT

Use Cases: what can we dream of?

Making Hadoop Fly in Clouds

Another example would be a predicate-pushdown o
fact any other data-crunching framework such as Sp
where storage cloud is physically separated from co
known is that when deployed in a cloud, Hadoop ca
screamingly inefficient compared to on-premise Ha
be entirely embedded into cloud-storage systems li
it into OpenStack Swift. Then Hadoop could detect s
making Hadoop friendlier for the cloud or should I s

Grepping S3/CloudFiles

Yet another class of killer applications would be a cl
within cloud storage is challenging to say the least. A
scan and index whole data corpus periodically. This
ZeroVM could enable a low-cost interactive search a

In-situ Cloud-enabled MapReduce

Lightweight, low-cost MapReduce framework which
about crunching your dataset instantly by renting ag
there. 5TB of data stored in S3 would be scattered usu
local scan could be significantly less than the resources re

Source: Open Stack

Grepping S3/ CloudFiles

➢One of the killer application is low-cost *search/scan* feature in cloud-storage

In-situ Cloud–enabled MapReduce

➢Lightweight, low-cost MapReduce embedded into cloud-storage

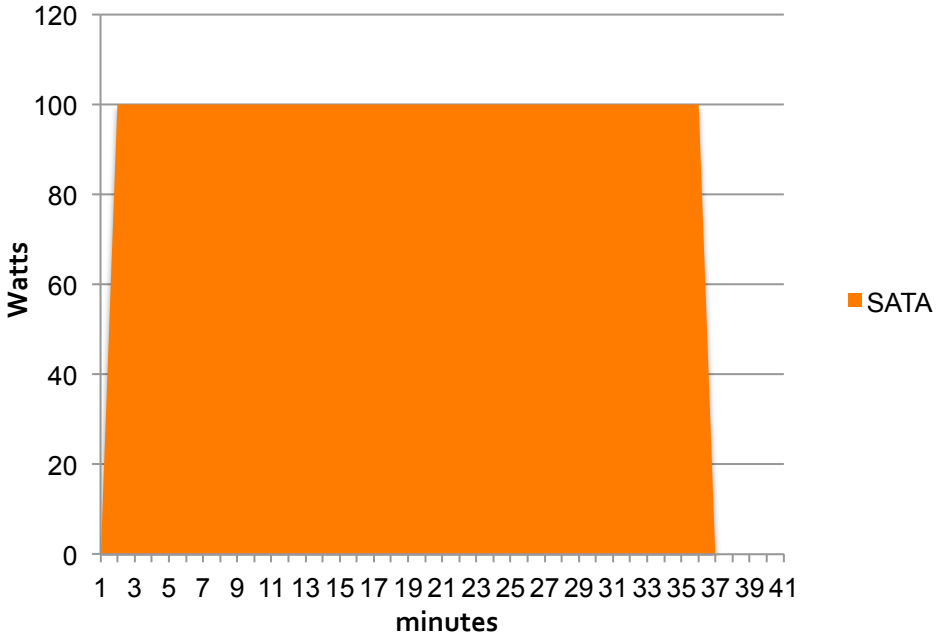
Making Hadoop Fly in Clouds

➢Hadoop could detect *smarter* cloud storage systems for performance optimization

NxGnData

Case study: energy efficiency (1)

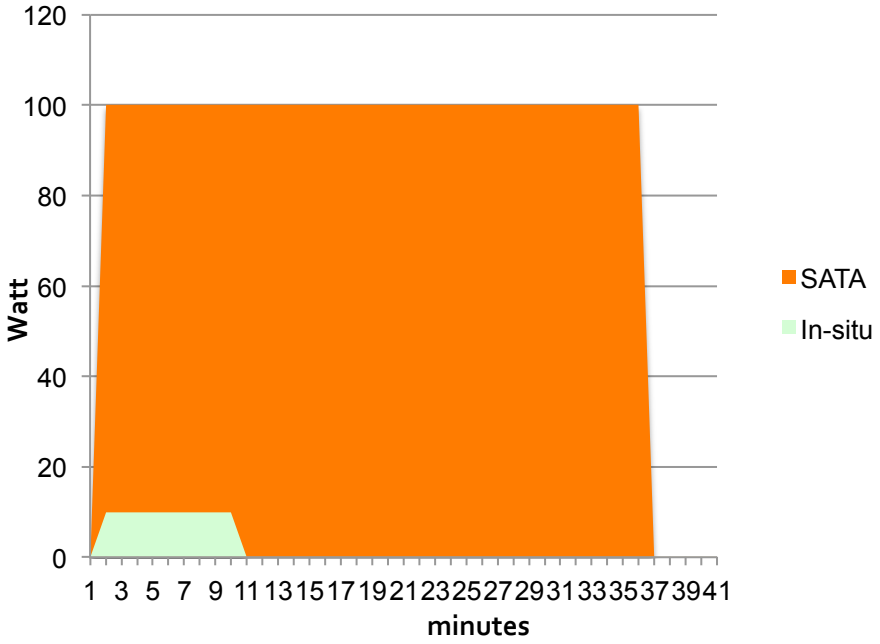
	Current Paradigm
Drives per server	1 (SATA 6G)
Capacity per drive	1TB
Host CPU	x86
Complete search	35 minutes *
Power consumption (CPU, DRAM, SSDs)	>100W
Energy consumed	100W*35min = **210kJ**



* assumes data transfer is bottleneck

Case study: energy efficiency (2)

	Current Paradigm	In-situ processing
Drives per server	SATA (6G)	PCIe NVMe
Capacity per drive	1TB	1TB
Host CPU	x86	x86
Complete search	35 minutes	9 minutes*
Power consumption (CPU, DRAM, SSDs)	>100W	<10 W
Power consumed (for entire search)	100W*35min = **210kJ**	10W*9min = **5kJ**
Energy Savings	**>98%**	
Computation time	**~ 25%**	

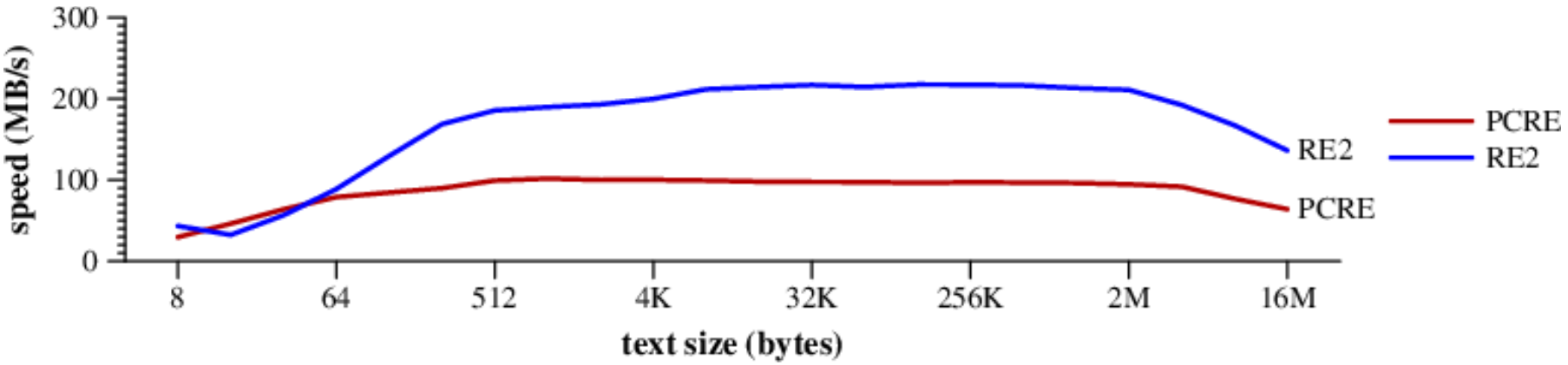


* HW acceleration engine @ 2GB/s

Case Study: text search performance

- Typical text search/scan computation performance on CPUs around hundreds of MB/s [xx]

Case Study: text search performance



This benchmark searches for .*$ in a randomly-generated input text of the given size. It gives a sense of the flat out search speed. [3]

Case Study: text search performance

- Hardware acceleration engines can push text search throughput to several GB/s [3]
- Efficient for search over extremely large sets of unstructured data
- Performance is limited by the media or the HW acceleration engine

Conclusion

- In-situ processing can significantly accelerate certain applications
- It can also greatly increase energy efficiency at the system level
- In-situ processing is a revolutionary approach that can fuel data analytics innovation
- There is a need for standardization of mechanisms necessary for the seamless use of computational resources embedded in storage devices.

NxGn Data, Inc.

- Visit us at booth 707

References

① Hadoop Fundamentals: Moving computation to data, not the other way around - http://fierydata.com/2012/02/17/hadoop-fundamentals-moving-computation-to-data-not-the-other-way-around/

② The Hadoop Distributed File System: Architecture and Design - http://hadoop.apache.org/docs/r0.18.0/hdfs_design.pdf

③ Regular Expression Matching in the Wild - http://swtch.com/~rsc/regexp/regexp3.html

④ LXC – Linux Containers - https://linuxcontainers.org/

⑤ Virtual Tunnel - http://vtun.sourceforge.net/

Reviewing Possible Use Cases

- Running grep remotely on SSD
- Searching large data sets using hardware assisted search engine on SSD
- Performing Map section of Map-Reduce on SSD
- And many more!