Arquitecturas para tratar grandes volúmenes de información

Infraestructura para Big Data

Arquitecturas para tratar grandes volúmenes de información

Índice:

1.1 Arquitecturas de referencia para Big Data

- Necesidades de los entornos de procesamiento para Big Data.
- Infraestructura para BigData: servidores físicos (On Premise) versus virtual (Cloud)
- Elementos básicos: CPUs, almacenamiento, interconexión, GPUs, coprocesadores.
- Optimización para sistemas que tratan grandes volúmenes de información.
- Nuevas tendencias de computación.
- Casos de estudios: optimizando el rendimiento.

1.2 Instalación y configuración de un cluster Big Data (Hadoop)

1.3 Supervisión y mantenimiento.

1.4 Evaluación de prestaciones y optimización con tuneado de parámetros.

1.5 Infraestructura para otros entornos Big Data: Ecosistema Spark

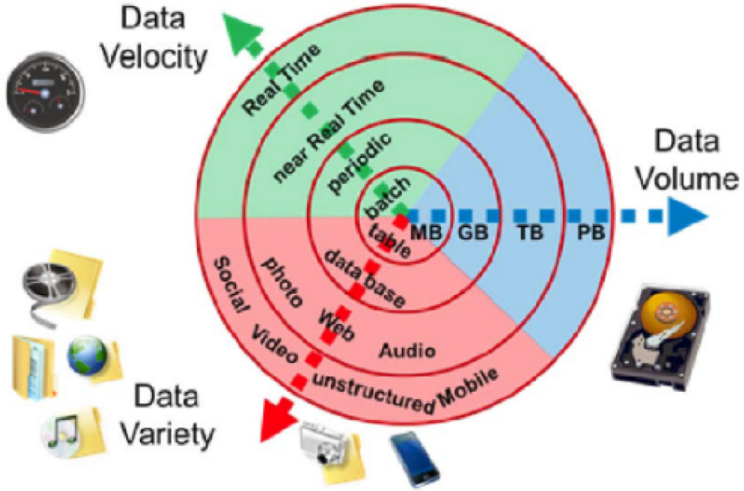
Arquitecturas para tratar grandes volúmenes de información

Índice (cont):

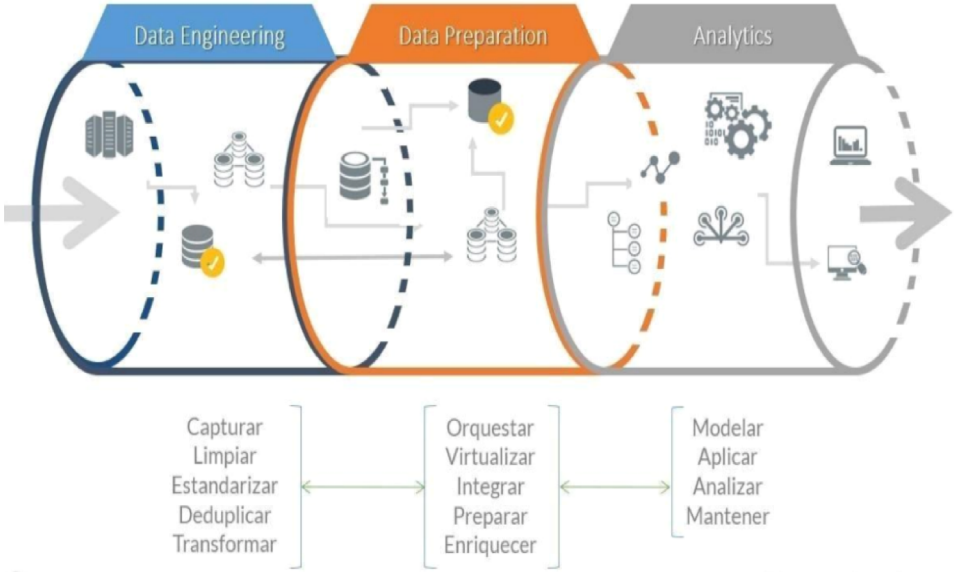
Necesidades de los sistemas para Big Data

Infraestructura que procesen:

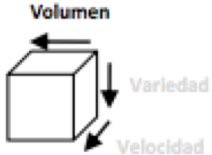
- Gran volumen de información.
- Con variedad de datos.
- Velocidad de llegada.



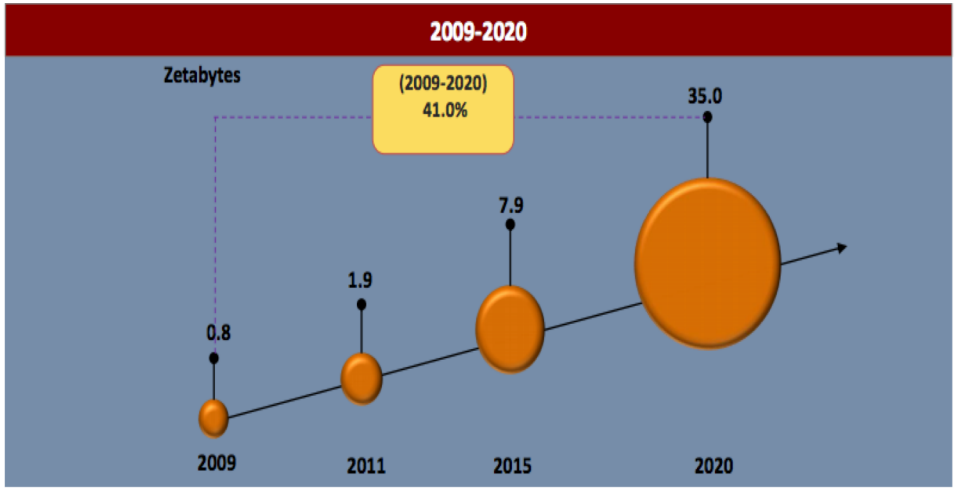
Gestionando toda la vida del dato desde su captura/preparación/enriquecimiento hasta su análisis/modelización/mantenimiento.



Necesidades para Big Data: Números de crecimiento

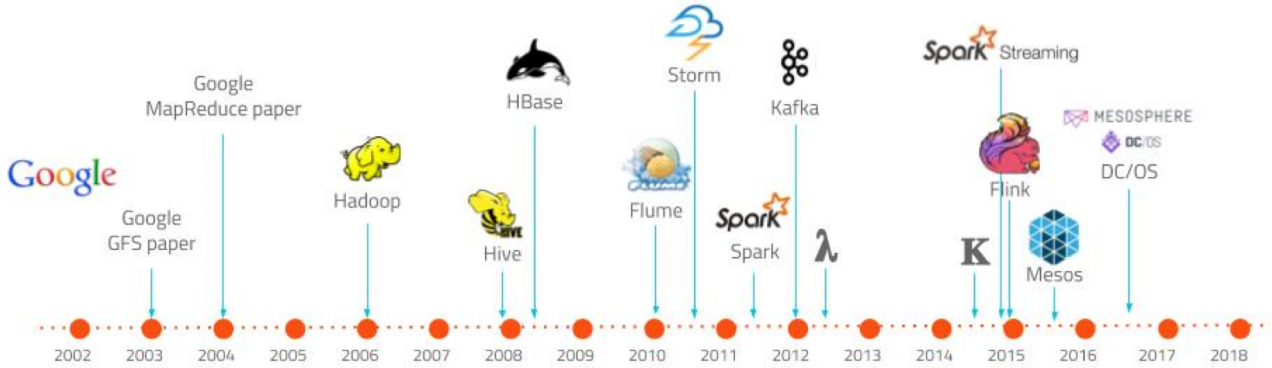


La tasa de crecimiento anual prevista es del 41%

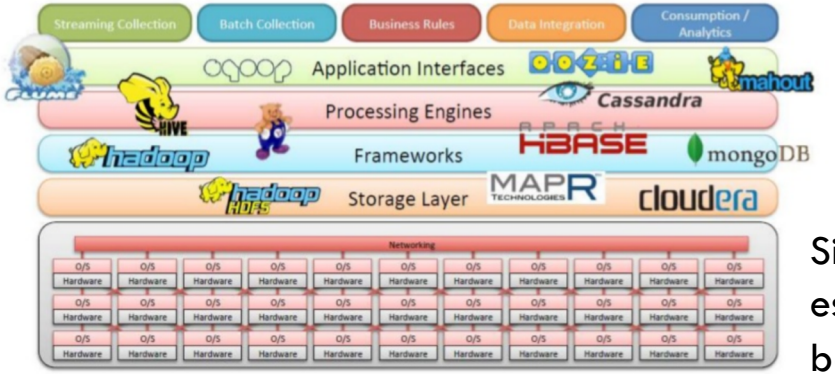


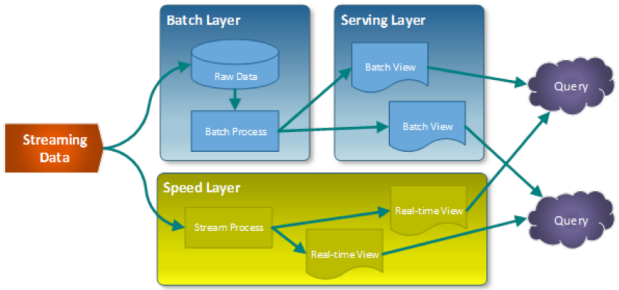
El 80% de los datos son desestructurados

Arquitectura de los sistemas para BigData



Línea temporal de las tecnologías para Big data

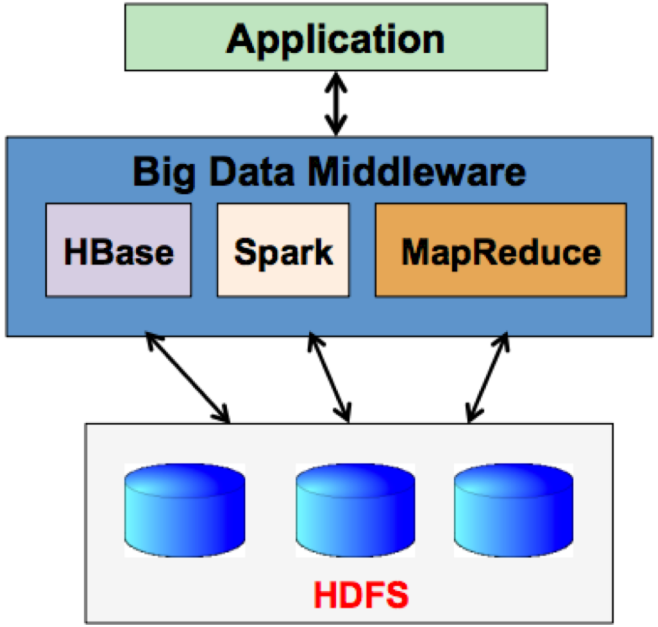




Sistema robusto tolerante a fallos, que sea linealmente escalable y que permita realizar escrituras y lecturas con baja latencia =>Arquitectura lambda

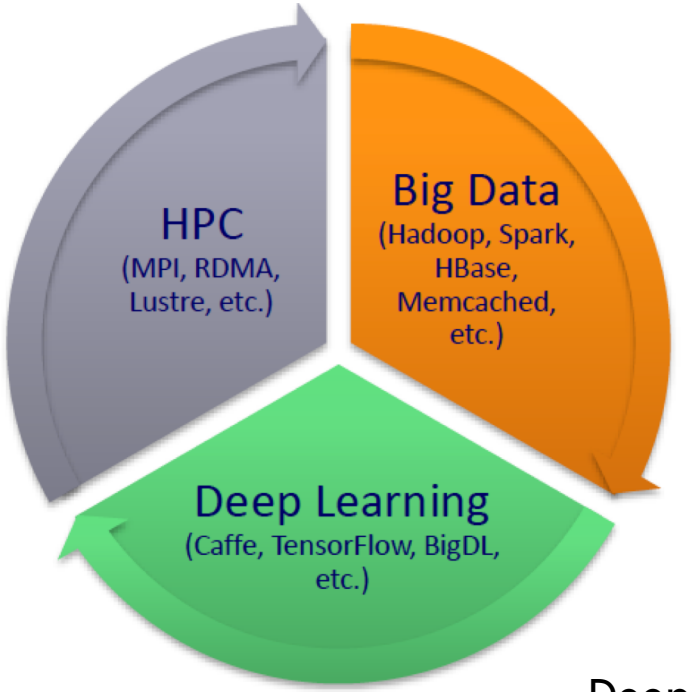
Entornos (Frameworks) más usados para procesamiento de grandes volúmenes de información

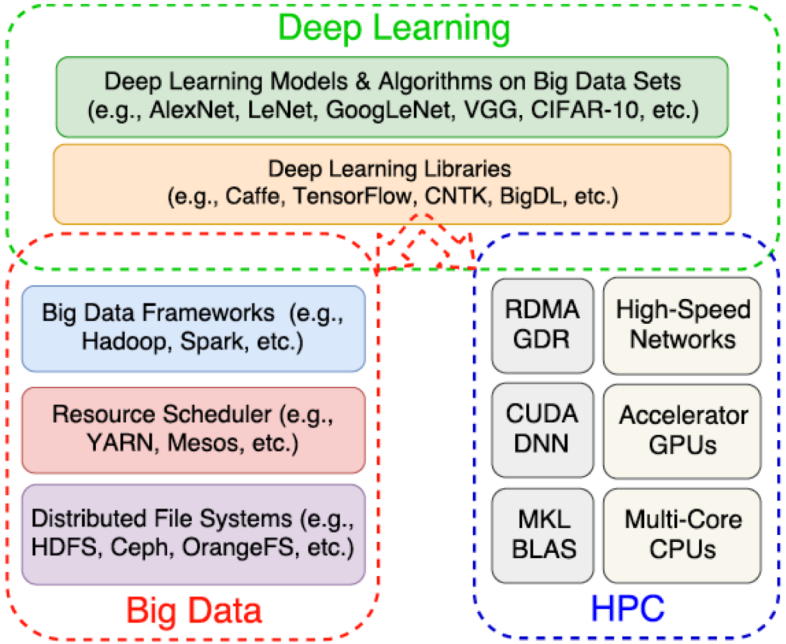
- *Frameworks* para Big Data: Hadoop MapReduce y Spark son actualmente los entornos de ejecución más populares
- Hadoop Distributed File System (HDFS) es el sistema de ficheros que está por debajo de Hadoop, Spark, y la base de datos Hbase (Hadoop database)
- Hoy en día, se utilizan a nivel de explotación en organizaciones como: Facebook, Yahoo!,…



Sistemas para BigData, HPC y Deep Leaning

Influencias entre High Performance Computing(HPC), Big Data, y Deep Learning (DL)





Deep Learning (DL) es un subconjunto de Machine Learning (ML), que está revolucionando los entornos de Big Data

Infraestructura de los sistemas para BigData:

Cluster de ordenadores:

Solución *low-cost*

Servidor especializado:

Opciones de integración de arquitecturas Big Data	
Arquitecturas físicas	Plataformas Cloud

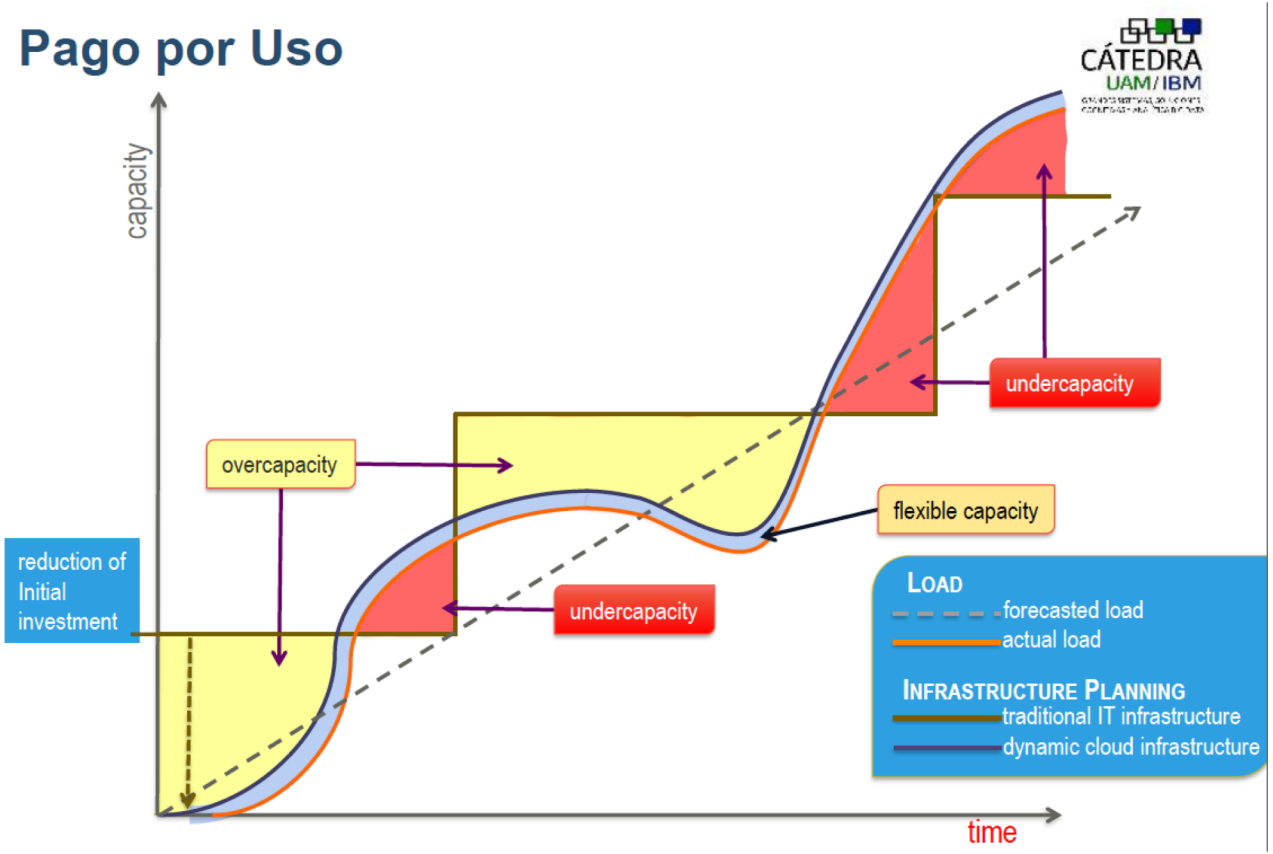
- Almacenamiento datos en la nube
- Ahorro en el hardware
- Mantenimiento por parte del proveedor
- Menor control del sistema desarrollado

Ejemplo:

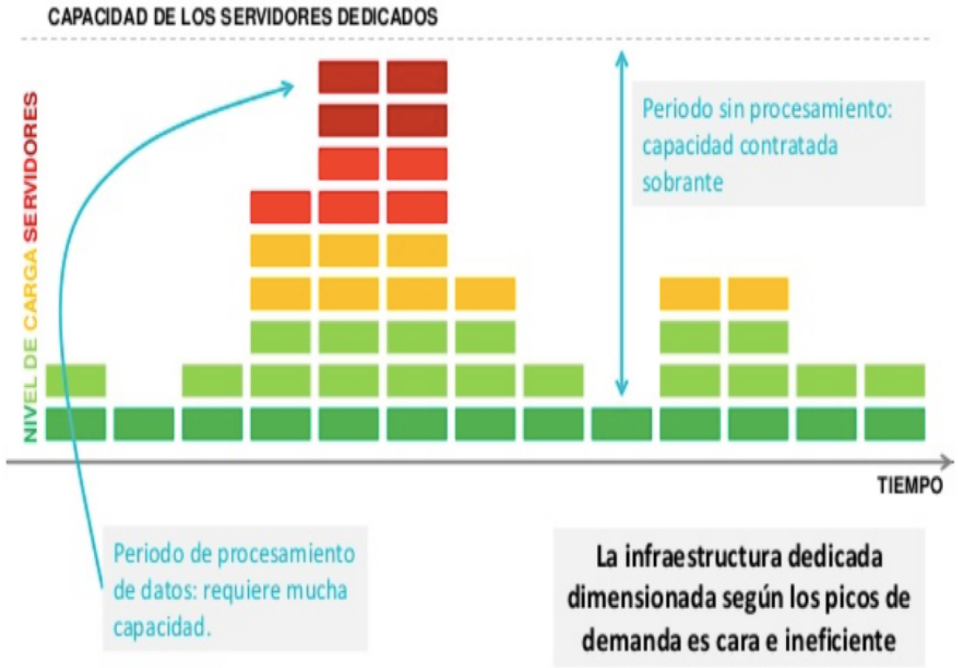
amazon web services

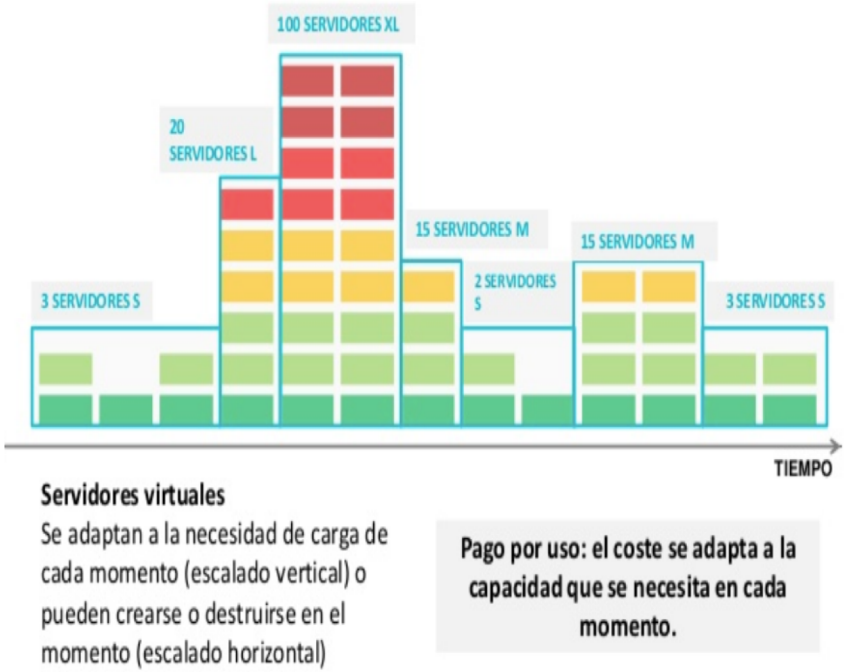
- Coste fijo + coste variable
- Soluciones adaptables
- Escalabilidad automática

Sistemas para BigData: Flexibilidad de la infraestructura

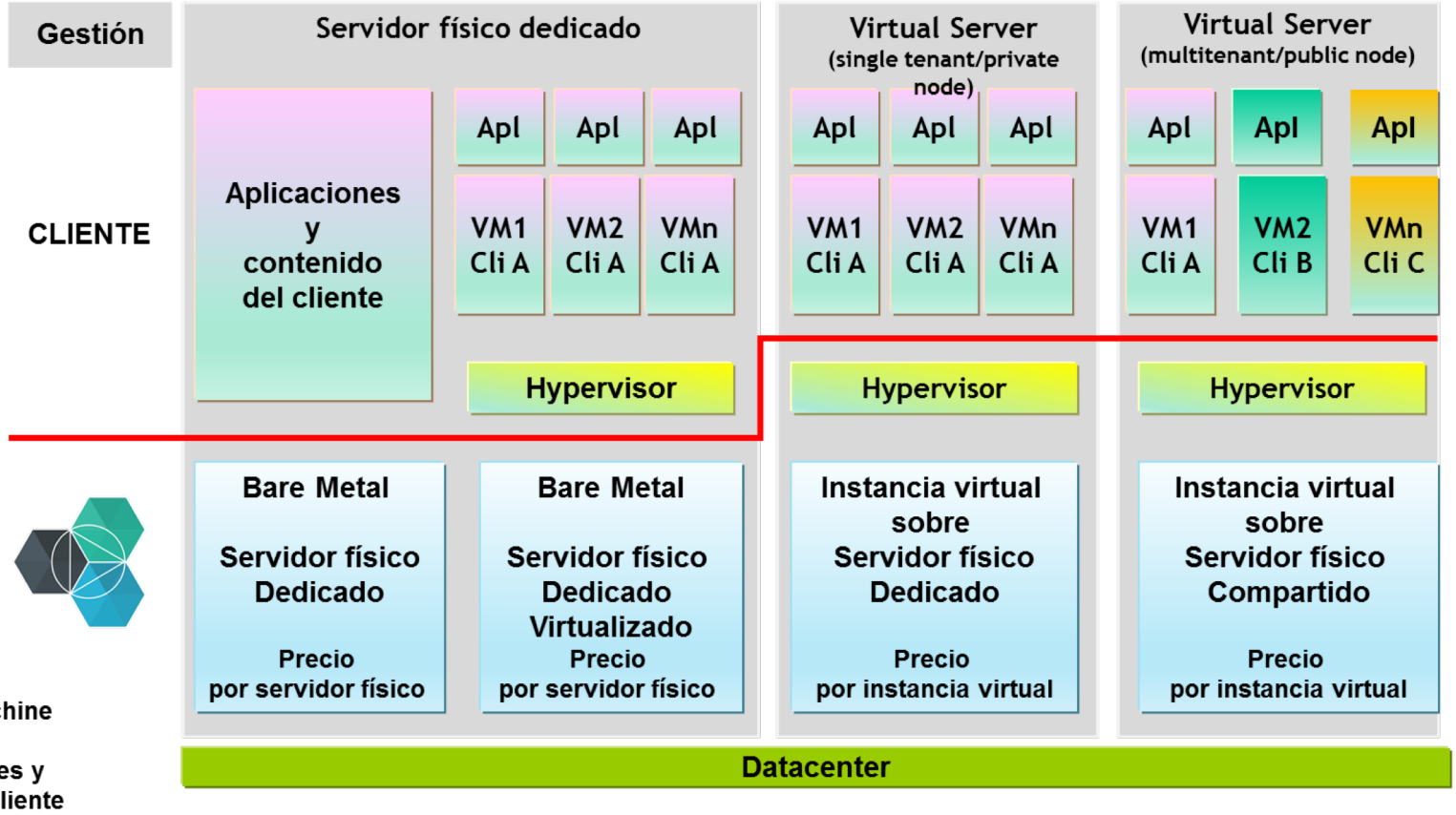


Sistemas para BigData: Flexibilidad de la infraestructura



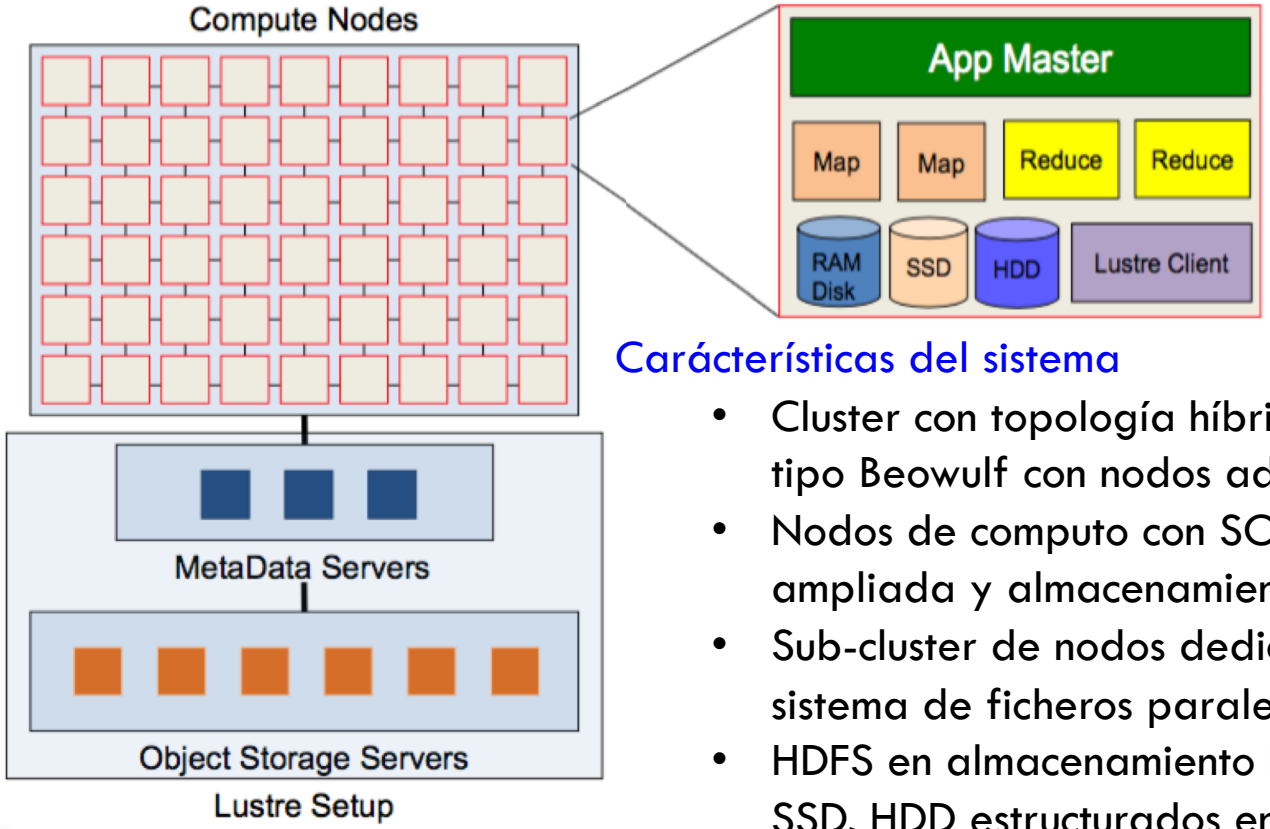


Sistemas para BigData: Servidores en "Bare Metal" o Virtuales



VM: Virtual Machine
Cli: Cliente
Apl: Aplicaciones y contenido del cliente

Sistema para Bigdata: Arquitectura de referencia



Carácterísticas del sistema

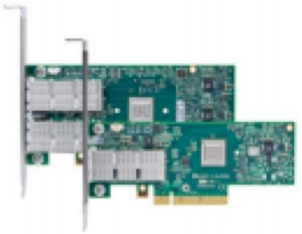
- Cluster con topología híbrida de una arquitectura tipo Beowulf con nodos adicionales para I/O.
- Nodos de computo con SO versión ligera; memoria ampliada y almacenamiento local pequeño.
- Sub-cluster de nodos dedicados para I/O con un sistema de ficheros paralelo, (en la figura Lustre)
- HDFS en almacenamiento heterogéneo: RAMDisk, SSD, HDD estructurados en RAID, JBOD,…

Infraestructura: Características de los componentes base

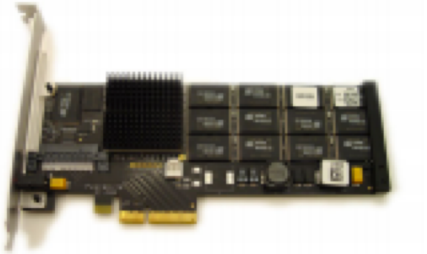
- Sistema multiprocesador/multicore con memoria compartida NUMA.
- Componentes:
  - Procesador: Multi-core/many-core con Hyperthreading.
  - Almacenamiento:
    - Memoria (DDR4 , Flash, 3D Xpoint)
    - HDDs, Solid State Disks (SSDs),
    - Non-Volatile Random-Access Memory(NVRAM), y NVMe SSD.
  - Red de Interconexión con RDMA (Remote DirectMemoryAccess) networking
    - InfiniBand y RoICE (RDMA over Converged Enhanced Ethernet)
  - Aceleradores
    - NVIDIA GPGPU,
    - IntelXeon Phi,
    - FPGA

Multi-core Processors

Accelerators / Coprocessors
high compute density, high performance/watt
>1 TFlop DP on a chip



High Performance Interconnects - InfiniBand
:1usec latency, 100Gbps Bandwidth



SSD, NVMe-SSD, NVRAM

Arquitecturas para BigData: Optimizando el procesador

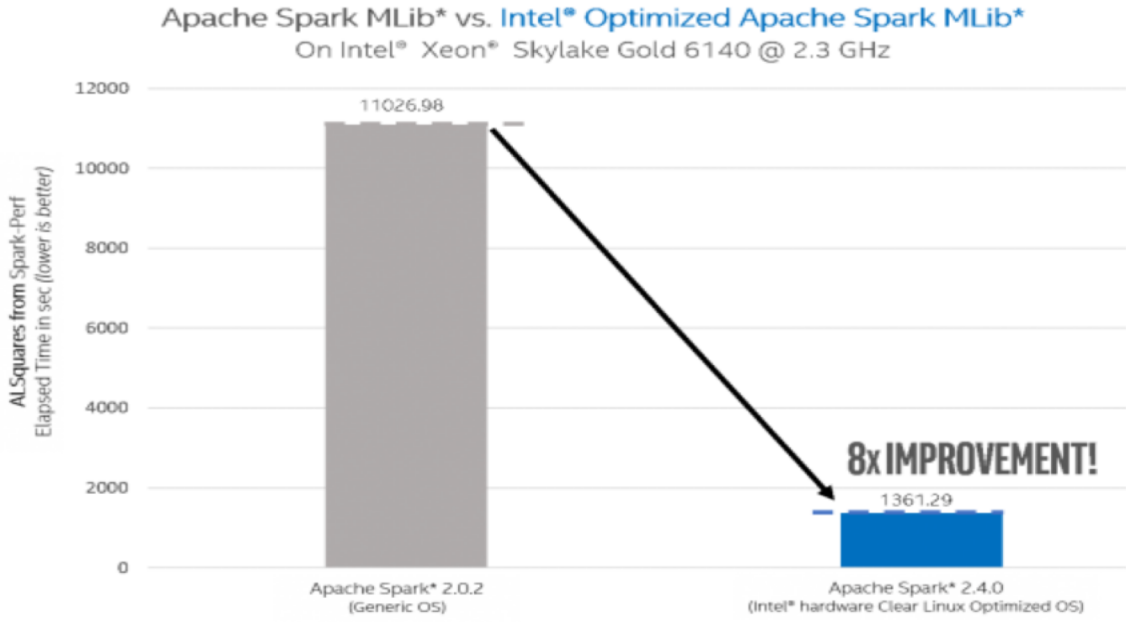
Accelerating Apache Spark machine learning with Clear Linux* OS …

https://01.org/blogs/2018/apache-spark-clear-linux

Caracteristicas:

Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
Intel® Memory Protection Extensions (Intel® MPX)
Intel® Ultra Path Interconnect (Intel® UPI)

Math LIB:
Intel MKL 2018.3.222 vs F2JBLAS



Hyper-threading (HT) technology was disabled to achieve better performance ¡!

Arquitecturas para BigData: El procesador optimizado

[1] Architectural Impact on Performance of In-memory Data Analytics: Apache Spark Case Study

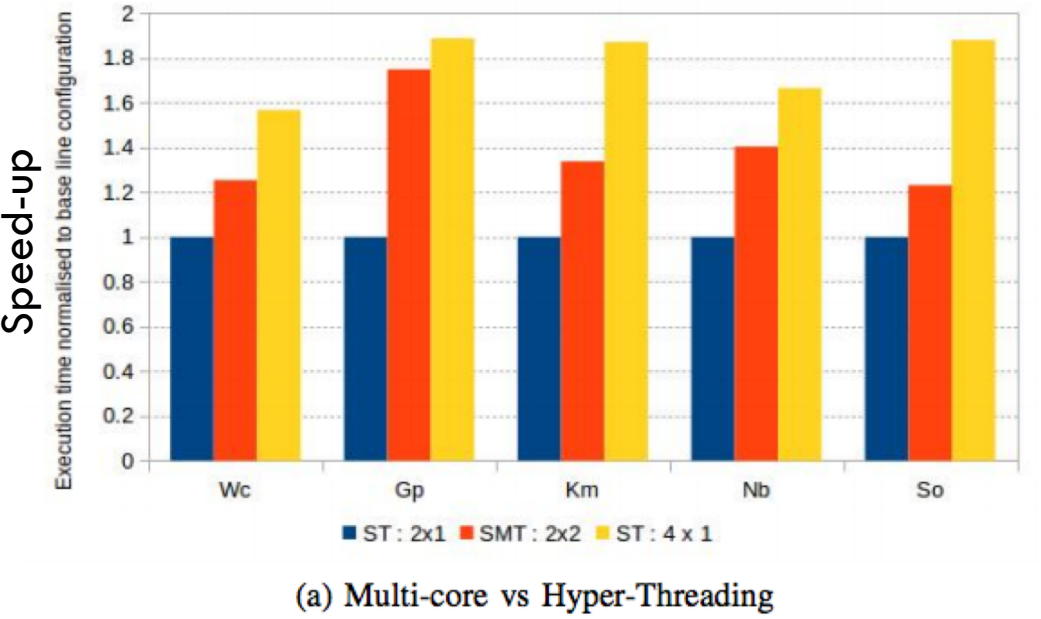
TABLE VII: Machine and Spark Configurations to evaluate Hyper Threading

		ST:2x1	SMT:2x2	ST:4x1
Hardware	No of sockets	1	1	1
	No of memory nodes	1	1	1
	No. of cores	2	2	4
	No. of threads	1	2	1
Spark	spark.driver.cores	2	4	4
	spark.default.parallelism	2	4	4
	spark.driver.memory (GB)	24	24	24

TABLE III: Machine Details.

Component	**Details**	
Processor	Intel Xeon E5-2697 V2, Ivy Bridge micro-architecture	
	Cores	12 @ 2.7GHz (Turbo up 3.5GHz)
	Threads	2 per Core (when Hyper-Threading is enabled)
	Sockets	2
	L1 Cache	32 KB for Instruction and 32 KB for Data per Core
	L2 Cache	256 KB per core
	L3 Cache (LLC)	30MB per Socket
Memory	2 x 32GB, 4 DDR3 channels, Max BW 60GB/s per Socket	
OS	Linux Kernel Version 2.6.32	
JVM	Oracle Hotspot JDK 7u71	
Spark	Version 1.5.0	

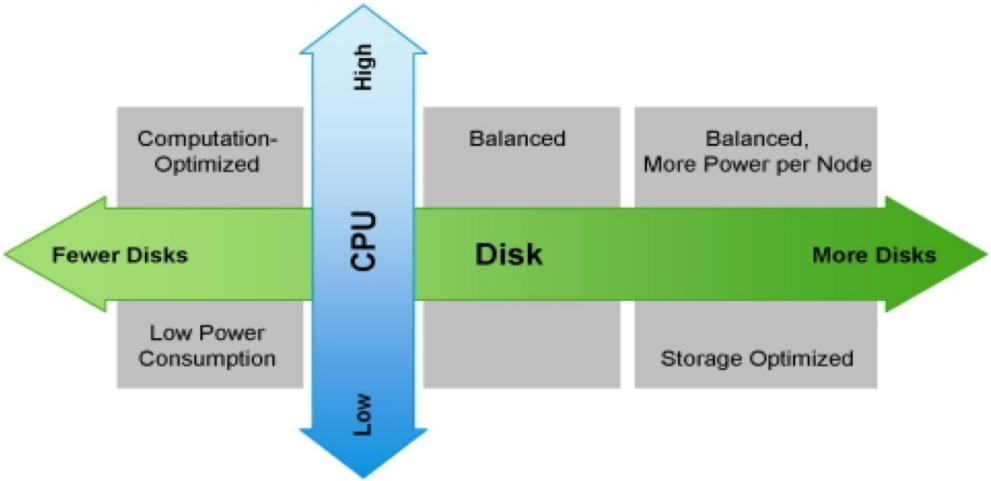
Arquitecturas para BigData: El procesador optimizado

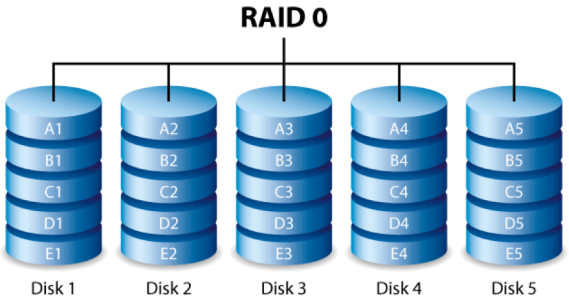


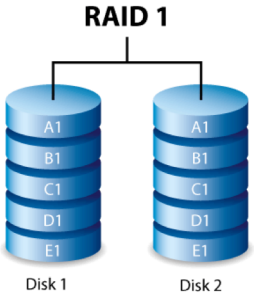
(a) Multi-core vs Hyper-Threading

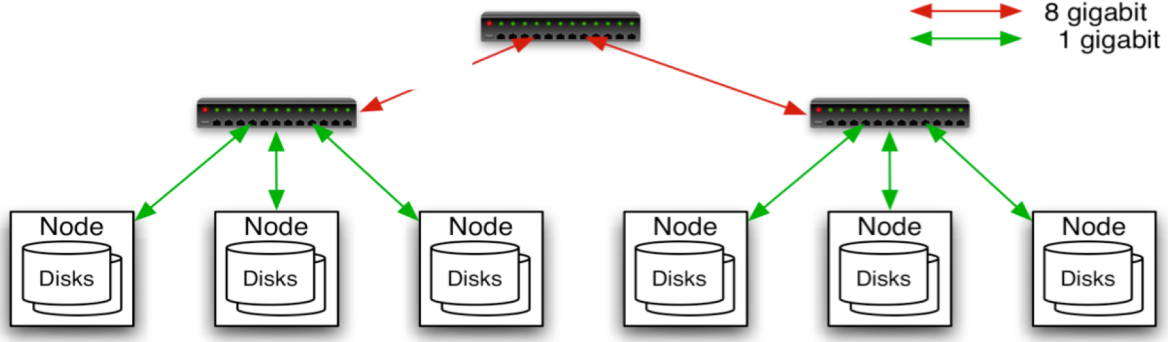
- Word Count (Wc): counts the number of occurrence of each word in a text file
- Grep (Gp): searches for the keyword The in a text file and filters out the lines with matching strings to the output file
- K-Means (Km): uses K-Means clustering algorithm from Spark Mllib. The benchmark is run for 4 iterations with 8 desired clusters
- NaiveBayes (Nb): runs sentiment classification
- Sort (So): ranks records by their key

Infraestructura para BigData: Requisitos del sistema



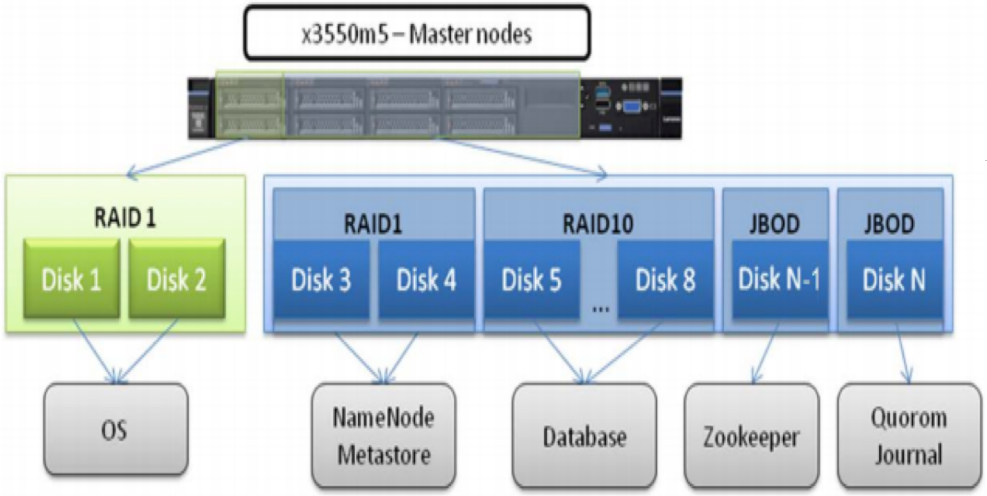






Infraestructura para BigData: Configuración de nodos

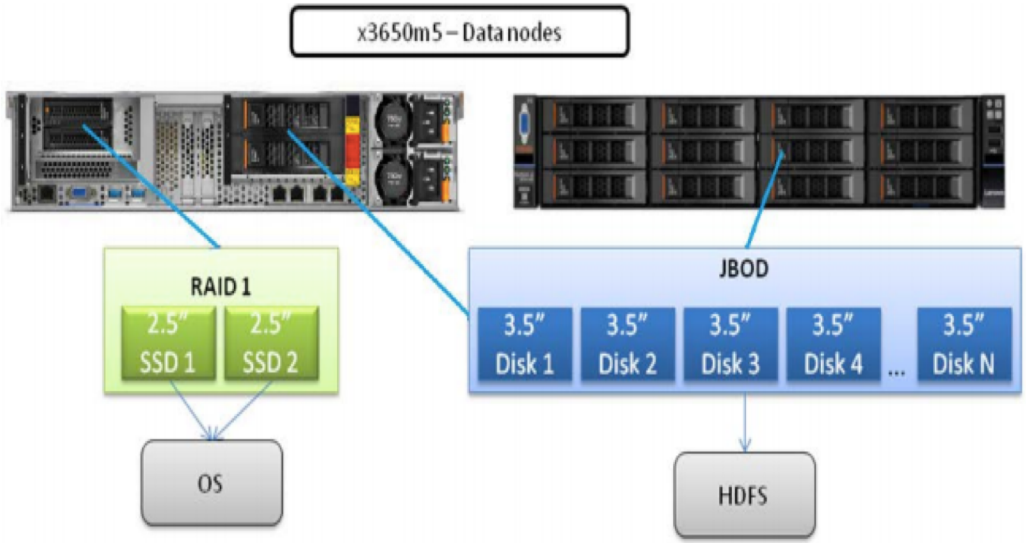
Nodo Maestro



Component	Master node configuration
System	System x3550 M5
Processor	2 x Intel Xeon processor E5-2650 v4 2.2 GHz 12-core
Memory - base	128 GB – 8 x 16 GB 2133 MHz RDIMM (minimum)
Disk (OS / local storage)	OS: 2x 2.5" HDD or SSD Data: 8 x 2TB 2.5" HDD
HDD controller	ServeRAID M5210 SAS/SATA Controller
Hardware management network adapter	Integrated 1GBaseT IMM Interface
Data network adapter	Broadcom NetXtreme Dual Port 10GbE SFP+ Adapter

Infraestructura para BigData: Configuración de nodos

Nodo de datos



Component	Data node configuration
System	System x3650 M5
Processor	2 x Intel Xeon processor E5-2680 v4 2.4GHz 14-core
Memory - base	256GB: 8x 32GB 2400MHz RDIMM
Disk (OS)	2x 2.5" HDD or SSD
Disk (data)	4 TB drives: 14x 4TB NL SATA 3.5 inch (56 TB Total) 6TB drives; 14x 6TB NL SATA 3.5 inch (84 TB total) 8 TB drives: 12x 8TB NL SATA 3.5 inch (96 TB Total)
HDD controller	OS: ServeRAID M1215 SAS/SATA Controller HDFS: N2215 SAS/SATA HBA
Hardware storage protection	OS: RAID1 HDFS:None (JBOD). By default, Hortonworks maintains a total of three copies of data stored within the cluster. The copies are distributed across data servers and racks for fault recovery.

Infraestructura para BigData: Red de comunicación

Figure 7. Lenovo RackSwitch G8272

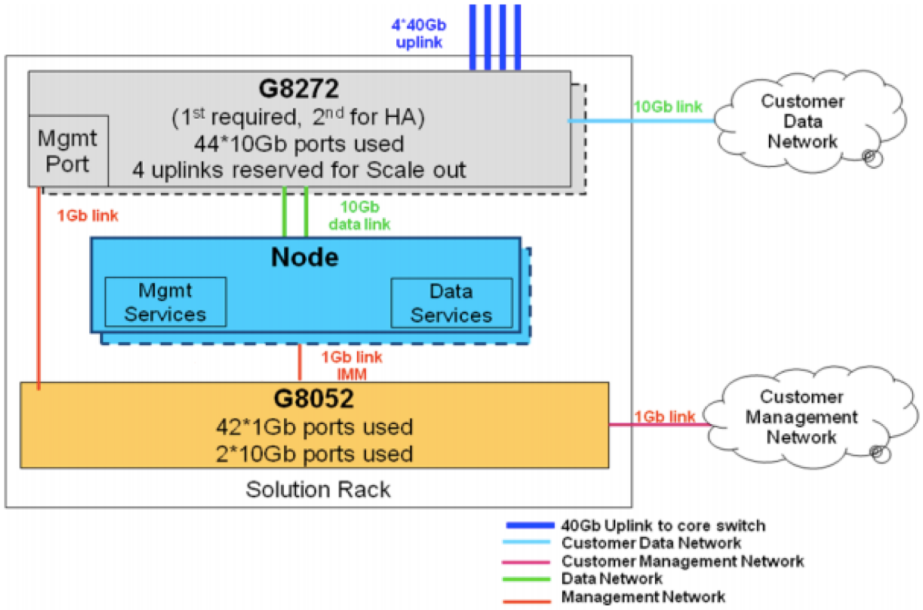
The enterprise-level Lenovo RackSwitch G8272 has the following characteristics:

- 48 x SFP+ 10GbE ports plus 6 x QSFP+ 40GbE ports
- Support up to 72 x 10Gb connections using break-out cables
- 1.44 Tbps non-blocking throughput with low latency (~ 600 ns)
- Up to 72 1Gb/10Gb SFP+ ports
- OpenFlow enabled allows for easily created user-controlled virtual networks

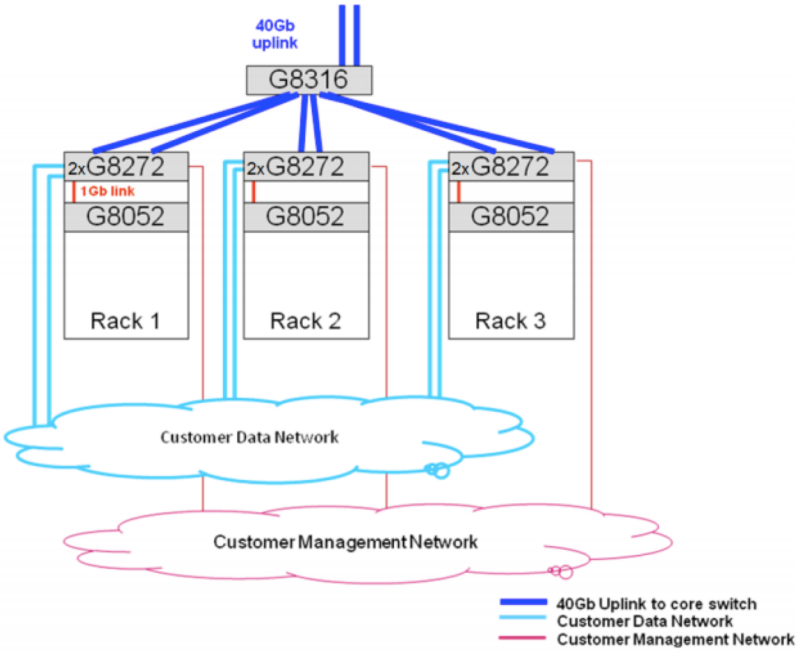
Figure 6. Lenovo RackSwitch G8052

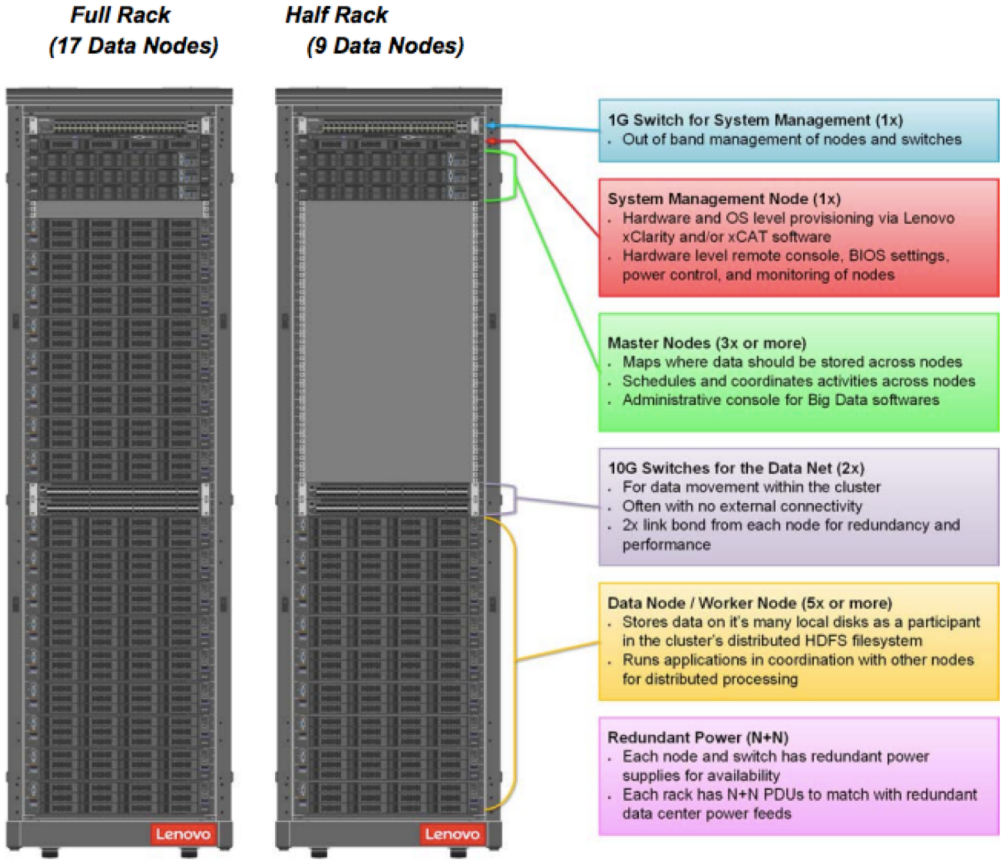
Lenovo RackSwitch G8052 has the following characteristics:

- A total of 48 1 GbE RJ45 ports
- Four standard 10 GbE SFP+ ports
- Low 130W power rating and variable speed fans to reduce power consumption

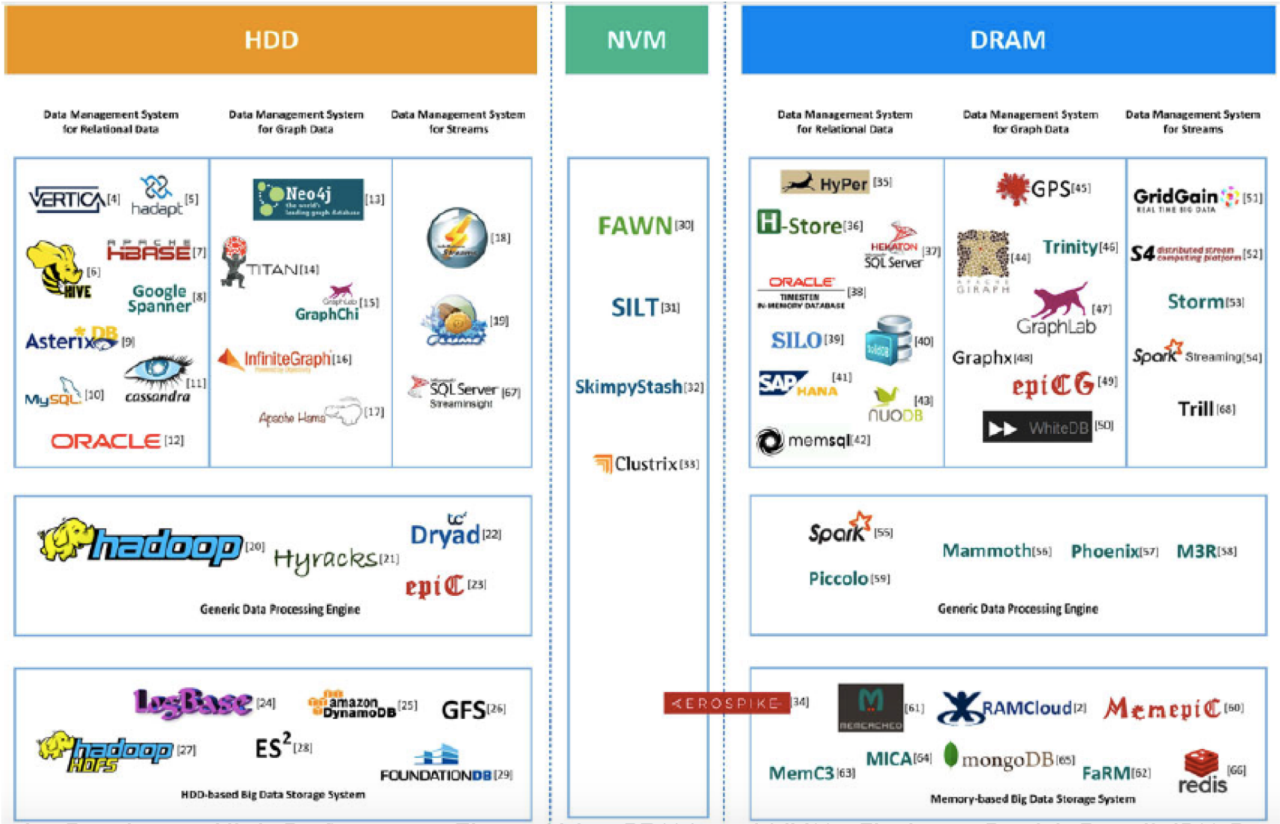


Infraestructura para BigData: Cluster





Sistemas para BigData: Disk-based vs in-memory based .



Ref: Running Apache Spark on a High-Performance Cluster Using RDMA and NVMe Flash por Patrick Stuedi, IBM Research

Infraestructura para BigData: Repaso de conceptos

Debe entender y ser capaz de responder :

- ¿Qué es?
- ¿Qué se mejora?
- ¿Cuándo tiene sentido usarlo y que implica?

Para los siguientes conceptos (ver también repaso de Fundamentos de sistemas):

- ✓ Servidor Físico, Virtual, Contenedor
- ✓ Hyperthreading (HT), SMT
- ✓ Cache, Memoria principal, NUMA
- ✓ SSD, HDD, NVMe SSD
- ✓ RAID, JBOD

Tipos de Red de Interconexión

Muchas posibilidades:

ATM, Myrinet, Gigabit Ethernet, Fast Ethernet, Infiniband

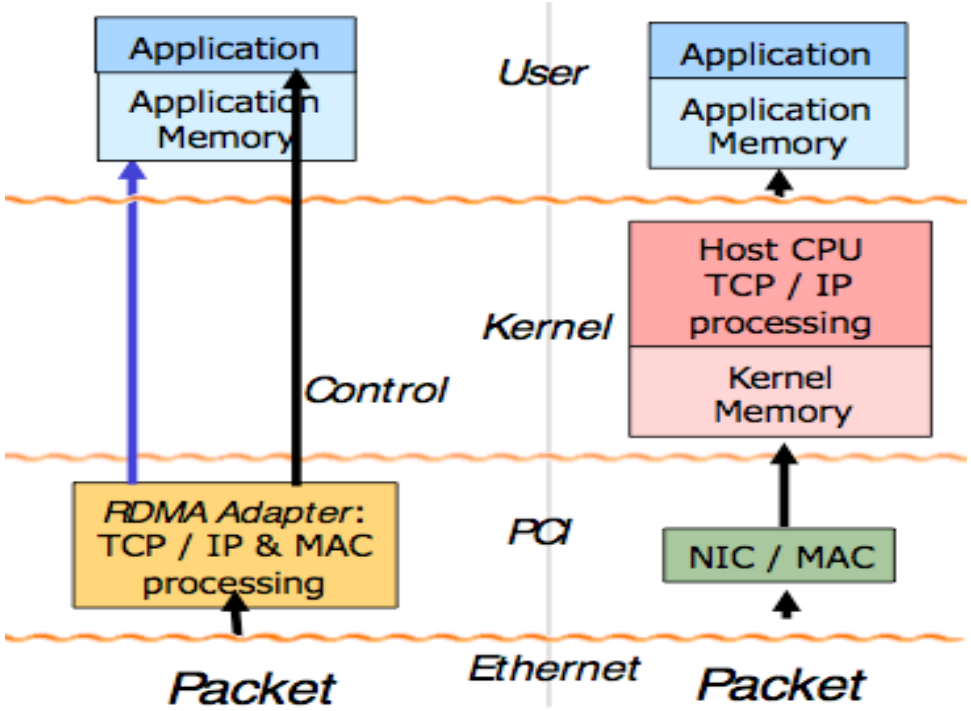
- Fast Ethernet (para gestion)
  - La red barata más rápida disponible
  - Ofrece un ancho de banda suficiente para la mayoría de situaciones.
  - Hasta 100-1000 Mbps
- Gigabit Ethernet:
  - Muy rápida (10, 40 y100 Gbps)
  - Coste decreciendo rápidamente.
- Infiniband:
  - Muy rápida
  - baja LATENCIA
  - coste mas alto

Caudal de Infiniband, bruto / eficaz

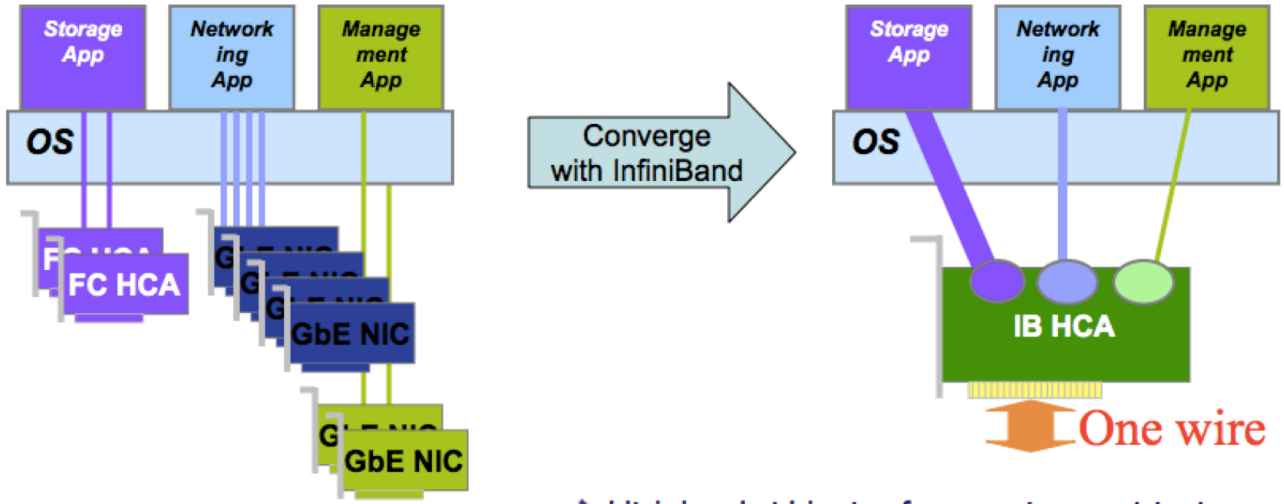
	SDR	DDR	QDR
1X	2,5 / 2 Gbps	5 / 4 Gbps	10 / 8 Gbps
4X	10 / 8 Gbps	20 / 16 Gbps	40 / 32 Gbps
12X	30 / 24 Gbps	60 / 48 Gbps	120 / 96 Gbps

Inteconexión: Redes de baja latencia (Low Latency Interconnects)

- Obtetivo: Disminuir la latencia para un paquete reduciendo el número de copias por paquete.



Infraestructura para BigData: Convergencia con Infiniband



- Slower I/O
- Different service needs – different fabrics
- No flexibility

- High bandwidth pipe for capacity provisioning
- Dedicated I/O channels enable convergence
  - For Networking, Storage, Management
  - Application compatibility
  - QoS - differentiates different traffic types
  - Partitions – logical fabrics, isolation

Remote Direct Memory Access

- **R**emote
  - –data transfers between nodes in a network
- **D**irect
  - –no Operating System Kernel involvement in transfers
  - –everything about a transfer offloaded onto Interface Card
- **M**emory
  - –transfers between user space application virtual memory
  - –no extra copying or buffering
- **A**ccess
  - –send, receive, read, write, atomic operations

Arquitecturas para BigData: RDMA

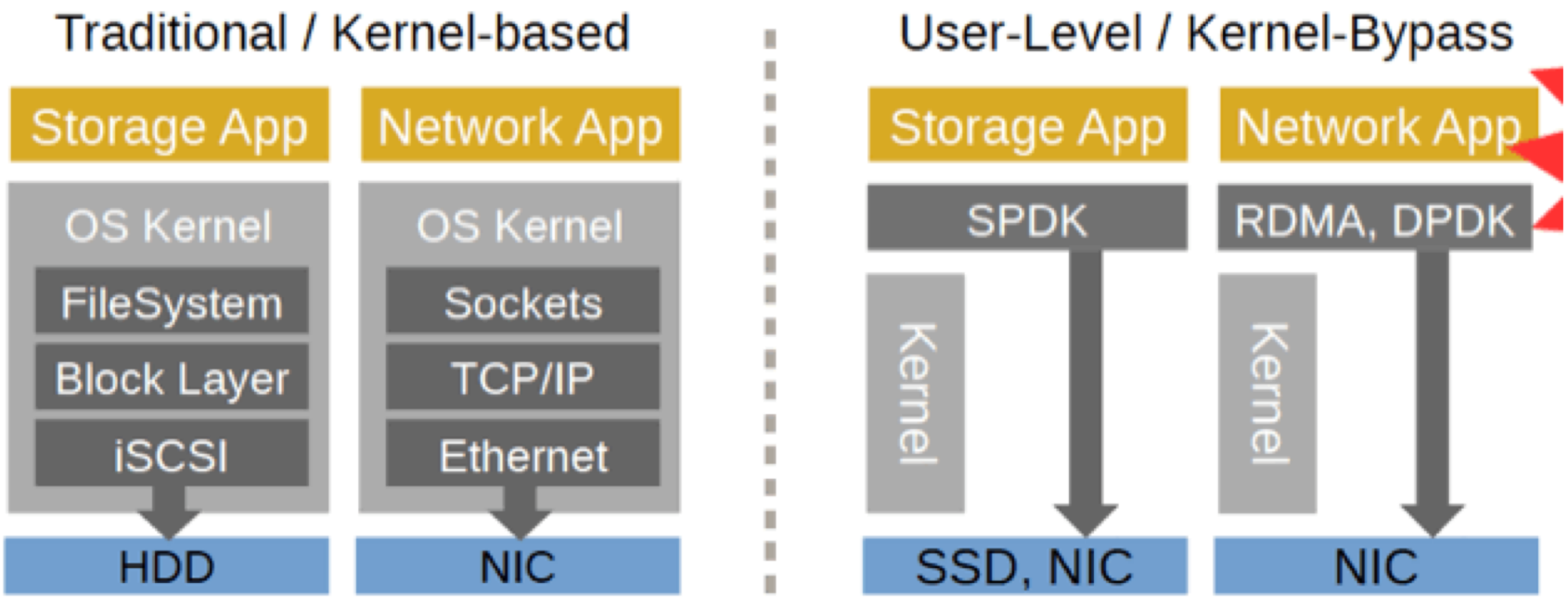
Similitudes y diferencias entre TCP y RDMA

- Ambas utilizan el modelo cliente-servidor
- Ambas requieren de una conexión para transporte fiable
- Ambas proporcionan un modo de transporte fiable
  - TCP garantiza secuencias en orden de **bytes**
  - RDMA garantiza secuencias en orden de **mensajes**

RDMA aporta :

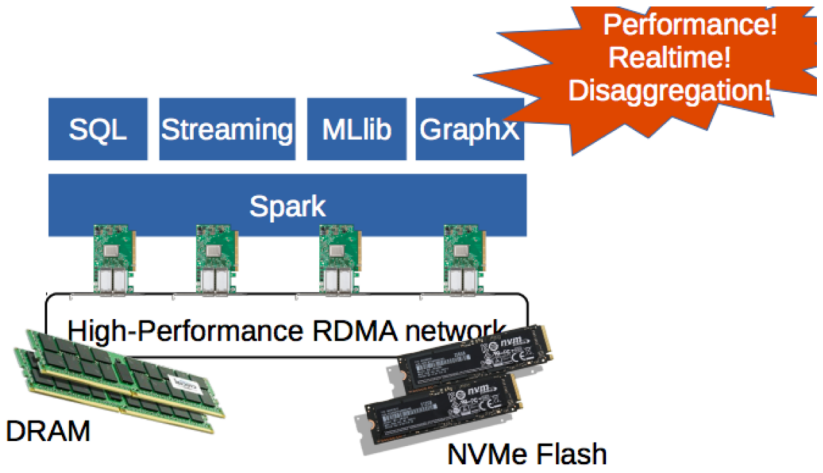
- "zero copy" – datos transferidos directamente de memoria virtual de un nodo a memoria virtual de otro nodo
- "kernel bypass" – no involucra al sistema operativo en las transferencias de datos
- Operación asíncrona – Los threads no se bloquean durante la transferencia de I/O

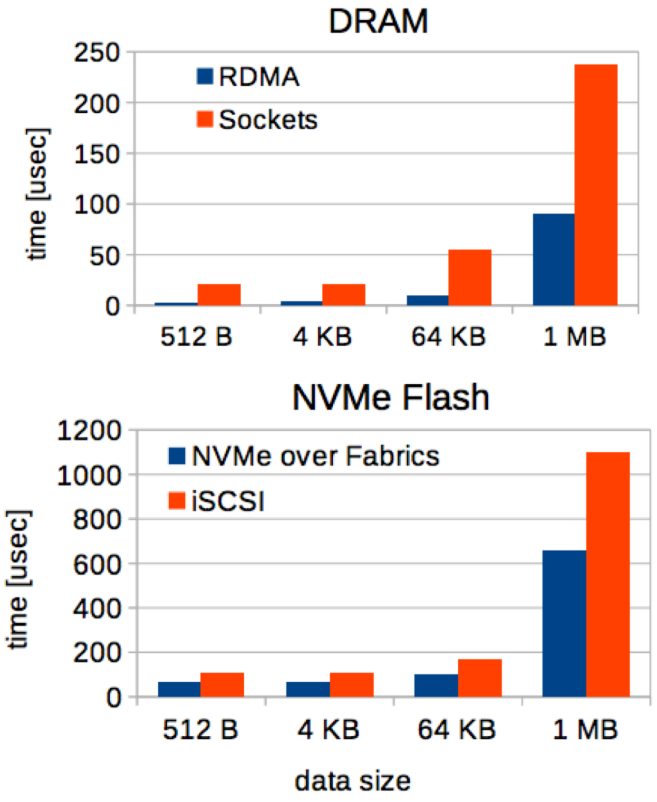
Arquitecturas para BigData: Liberar la CPU



Ref: Running Apache Spark on a High-Performance Cluster Using RDMA and NVMe Flash por Patrick Stuedi, IBM Research

Mejoras: Red con RDMA y Almacenamiento con NVMe





Ref: Running Apache Spark on a High-Performance Cluster Using RDMA and NVMe Flash por Patrick Stuedi, IBM Research

Infraestructura para BigData: Repaso de conceptos

Debe entender y ser capaz de responder :

- ¿Qué es?
- ¿Qué se mejora?
- ¿Cuándo tiene sentido usarlo y que implica?

Para los siguientes conceptos:

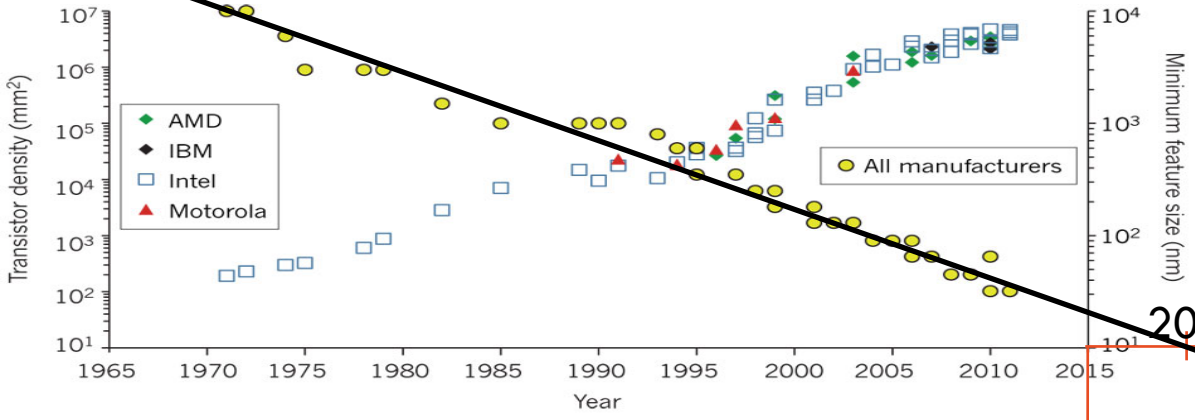
- ✓ Transferencias RDMA
- ✓ RDMA vs trasferencia TCP/IP
- ✓ Zero copy
- ✓ Latencia Infiniband

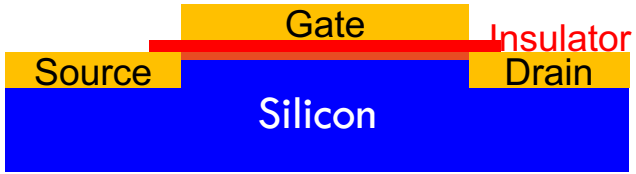
Evolución de las Tecnologías para BigData

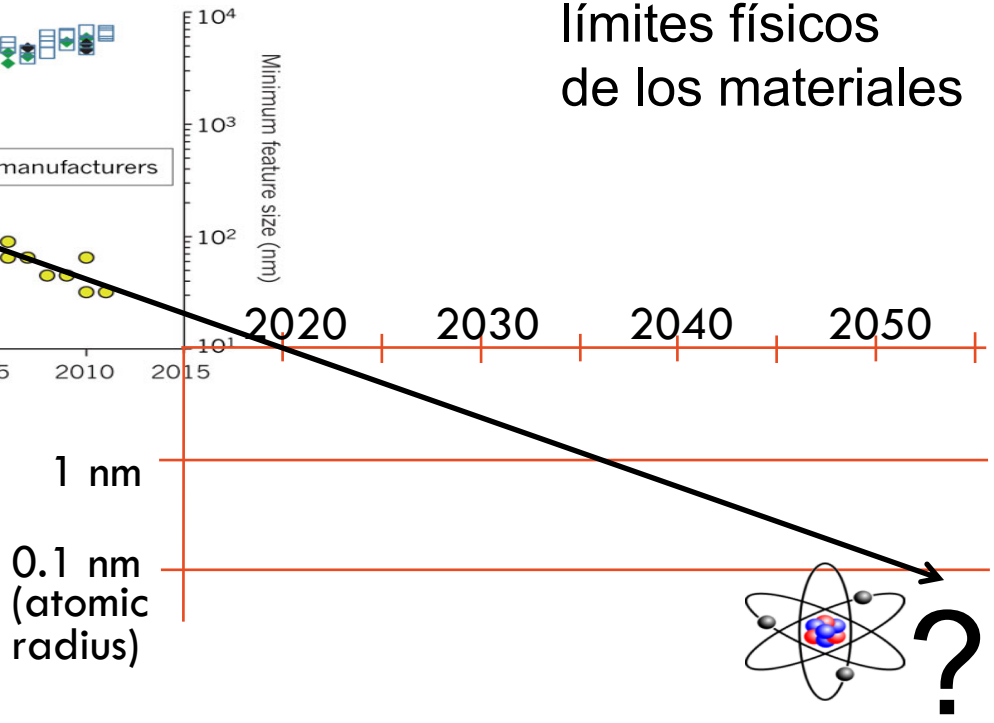
- Tecnologías actuales de computación para BigData
  - Sistemas Multicore
  - **Coprocesadores: GPUs, FPGA**
- Tecnologías disruptivas:
  - Neurocomputación
  - Computación Cuántica

Futuro de los procesadores

Ley de Moore … pronto entraremos en los límites físicos de los materiales







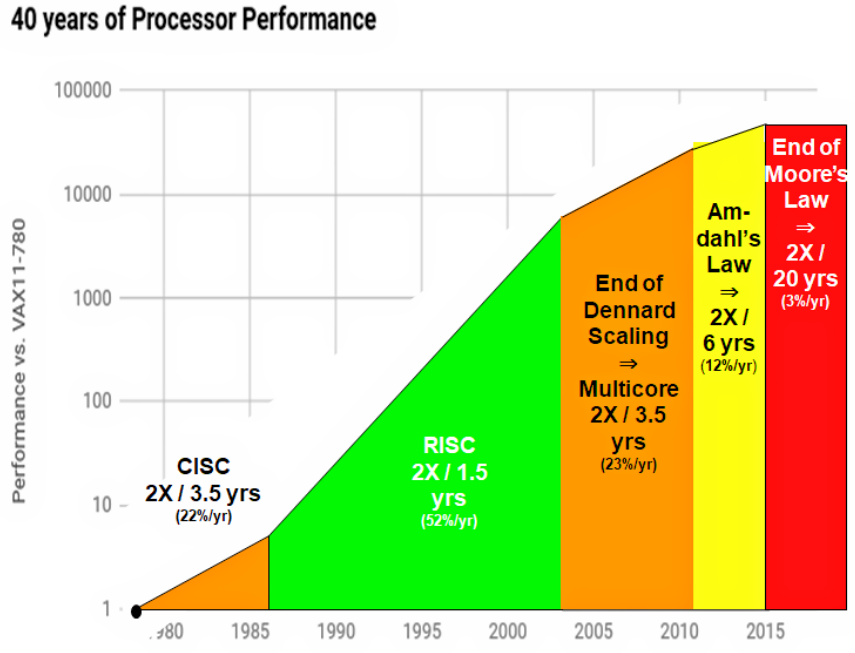
Moore, G. E. *Electronics* **8**, 114–117 (1965).
Image from: Ferain, I. *et al.*, *Nature* **479**, 310–316 (2011).

Evolución del rendimiento

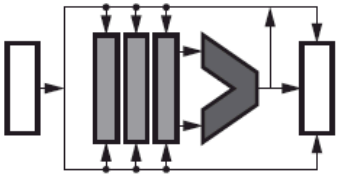
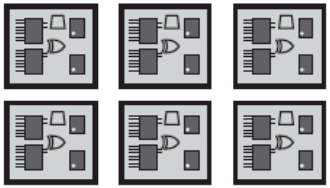
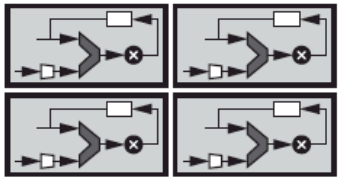
El rendimiento de los procesadores de propósito general se está estancando:

Se necesita nuevas tendencias para dar soluciones:

- Arquitecturas específicas para cada dominio.
- Computación aproximada.
- Tecnologías más disruptivas: procesadores cuánticos.



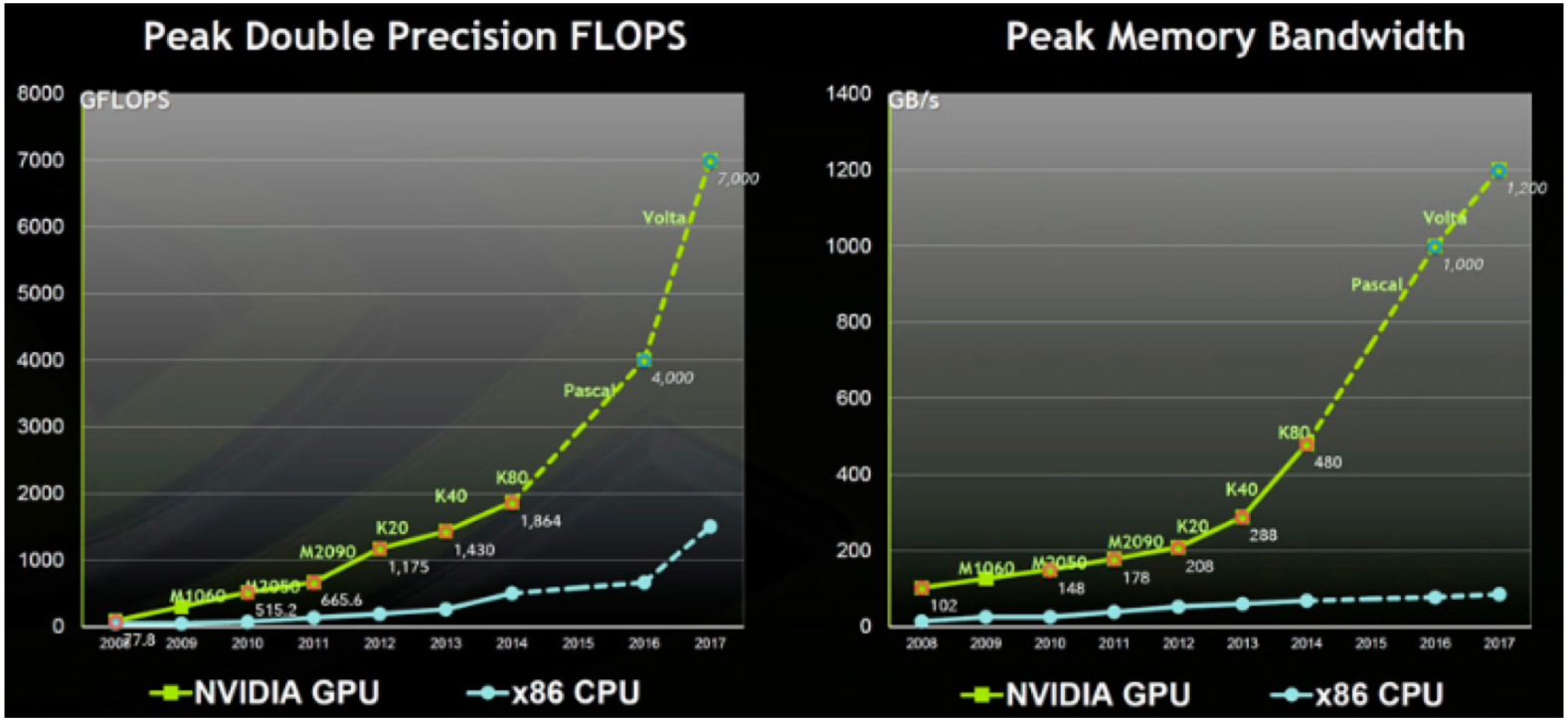
Necesidades de computación

CPU	FPGA	GPU
Scalar Processing	**Adaptable Hardware**	**Vector Processing** *(e.g., GPU, DSP)*
		
Complex Algorithms and Decision Making	Processing of Irregular Data Structures *Genomic Sequencing*	Domain-specific Parallelism
	Latency Critical Workloads *Real-Time Control*	Signal Processing *Complex Math, Convolutions*
	Sensor Fusion *Pre-processing, Programmable I/O*	Video and Image Processing

WP505_02_092918

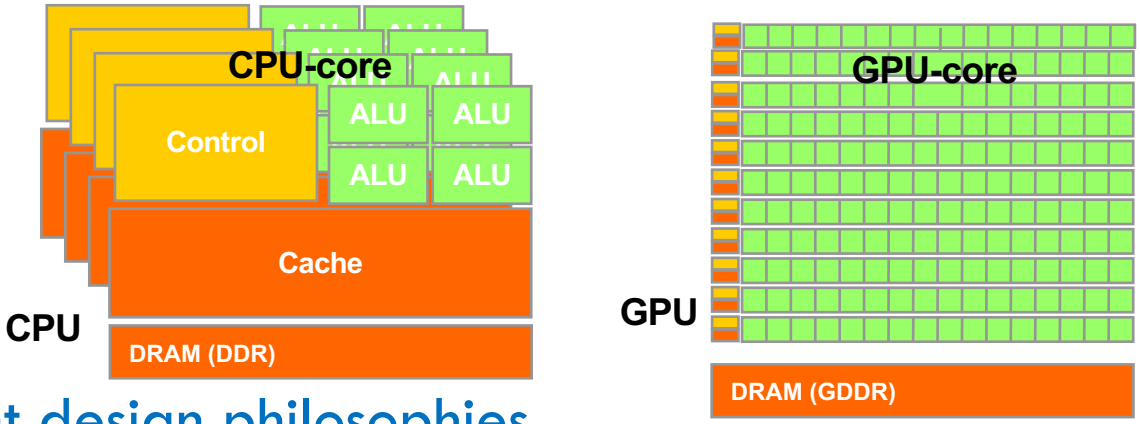
CPU vs GPUs

Performance Trends



CPU vs GPUs

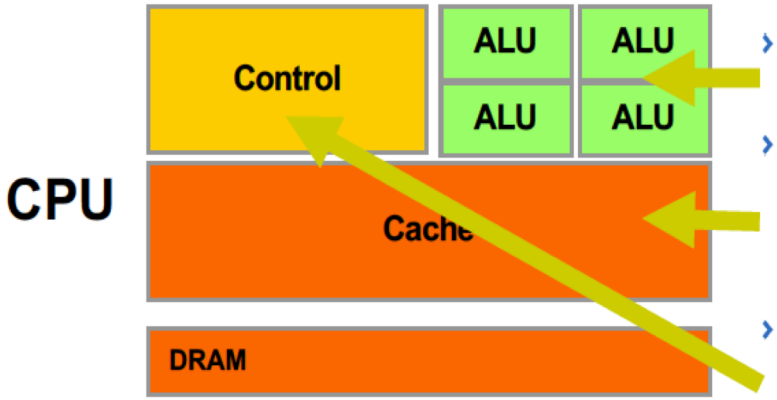
- The GPU is specialized for compute-intensive, highly data parallel computation (what graphics rendering is about)
- So, more transistors can be devoted to data processing rather than data caching and flow control



Different design philosophies

CPU: A few out-of-order cores vs GPU: Many in-order cores

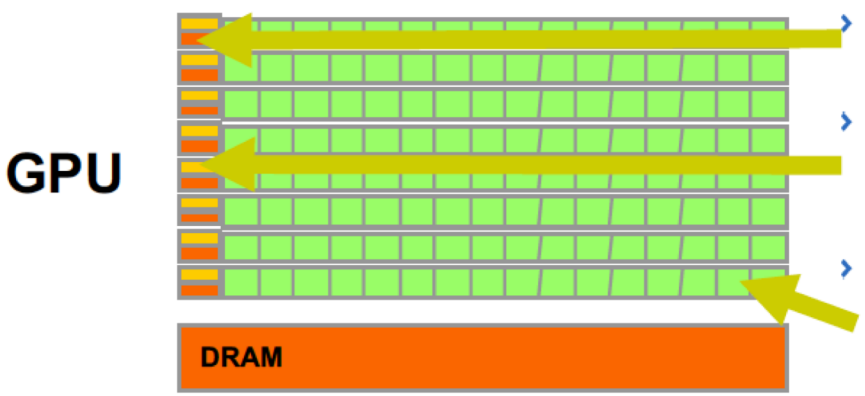
CPU: Latency Oriented Design



- Powerful ALU
  - Reduced operation latency
- Large caches
  - Convert long latency memory accesses to short latency cache accesses
- Sophisticated control
  - Branch prediction for reduced branch latency
  - Data forwarding for reduced data latency

CPUs for sequential parts where latency matters CPUs can be 10X+ faster than GPUs for sequential code

GPUs: Throughput Oriented Design



- Small caches
  - To boost memory throughput
- Simple control
  - No branch prediction
  - No data forwarding
- Energy efficient ALUs
  - Many, long latency but heavily pipelined for high throughput
- Require massive number of threads to tolerate latencies
  - Threading logic
  - Thread state

GPUs for parallel parts where throughput wins GPUs can be 10X+ faster than CPUs for parallel code

Infraestructura BigData: Aceleradores/Coprocesadores GPU

CPU **vs** **GPU**

Optimized for **low latency**

+ Large main memory
+ Fast clock rate
+ Large caches
+ Branch prediction
+ Powerful ALU
– Relatively low memory bandwidth
– Cache misses costly
– Low performance per watt

Optimized for **high throughput**

+ High bandwidth main memory
+ Latency tolerant (parallelism)
+ More compute resources
+ High performance per watt
– Limited memory capacity
– Low per-thread performance
– Extension card

Review of GPU Architecture

Streaming Multiprocessors (SM)

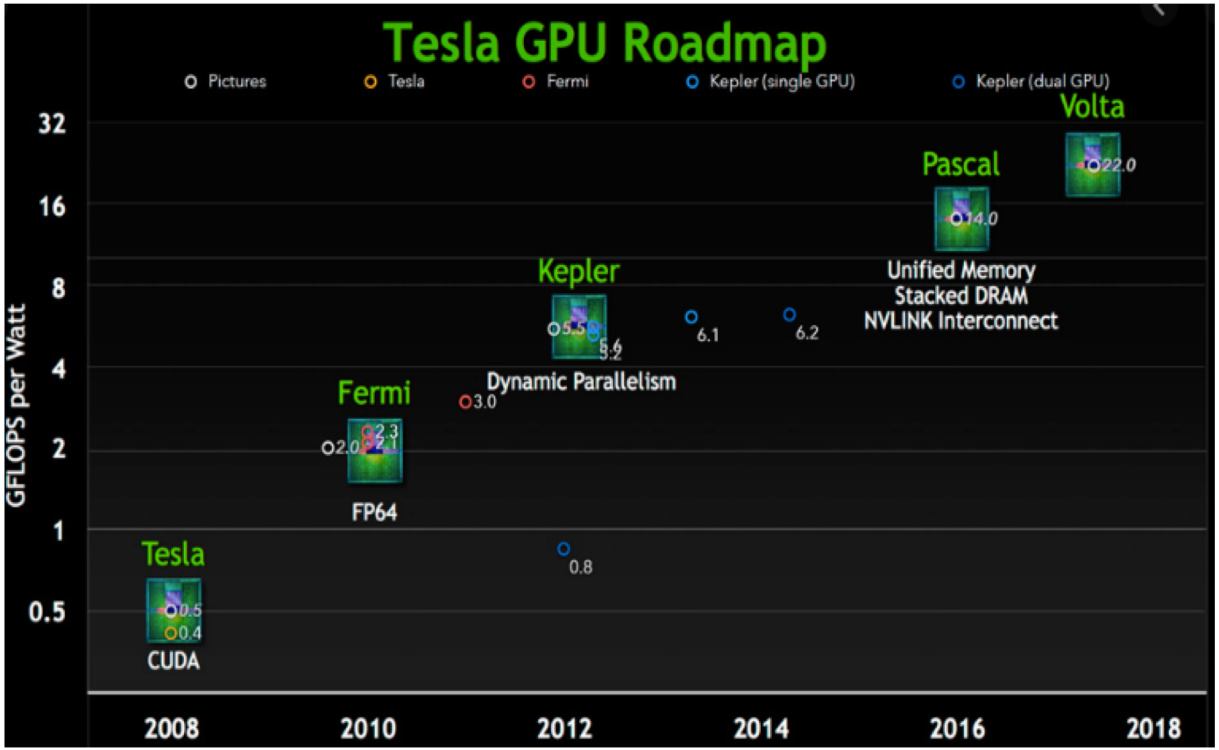
- Compute Units (CU)

Streaming Processors (SP) or CUDA cores

- Vector lanes

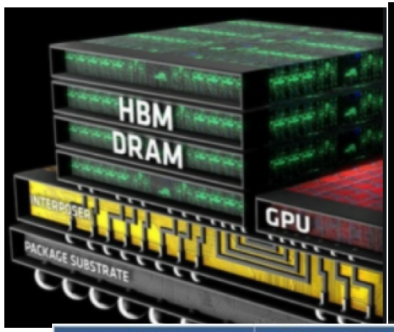
Number of SMs x SPs

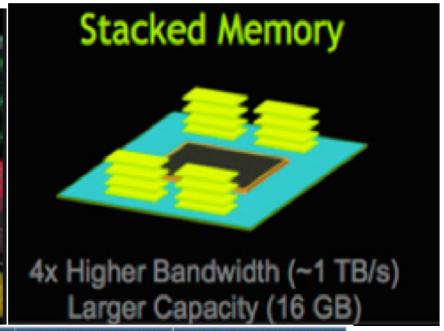
- Tesla (2007): 30 x 8
- Fermi (2010): 16 x 32
- Kepler (2012): 15 x 192
- Maxwell (2014): 24 x 128
- Pascal (2016): 56 x 64
- Volta (2018): 80 x 64



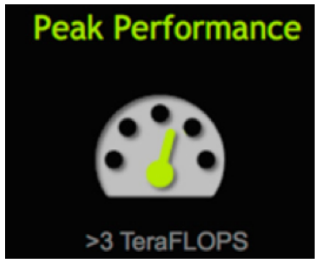
GPU: Improving Features

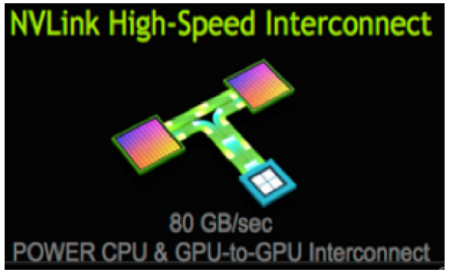
HBM **High Bandwidth Memory**



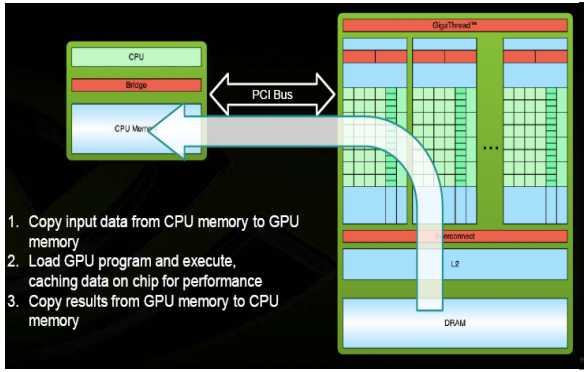


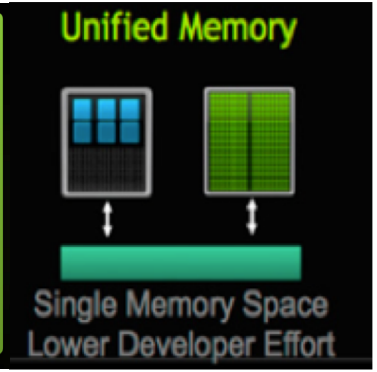
Item	DDR3 (x8)	GDDR5 (x32)	4-Hi HBM (x1024)
I/O	8	32	1024
Prefetch (Per IO)	8	8	2
Access Granularity (=I/O x Prefetch)	8Byte	32Byte	256Byte
Max. Bandwidth	2GB/s	28GB/s	128~256GB/s

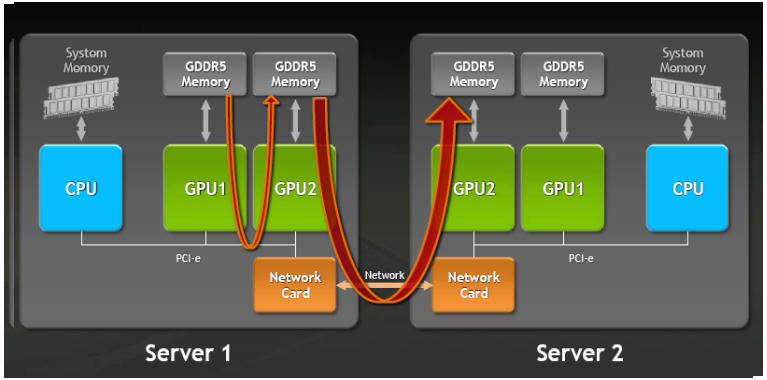




Full GPUDirect

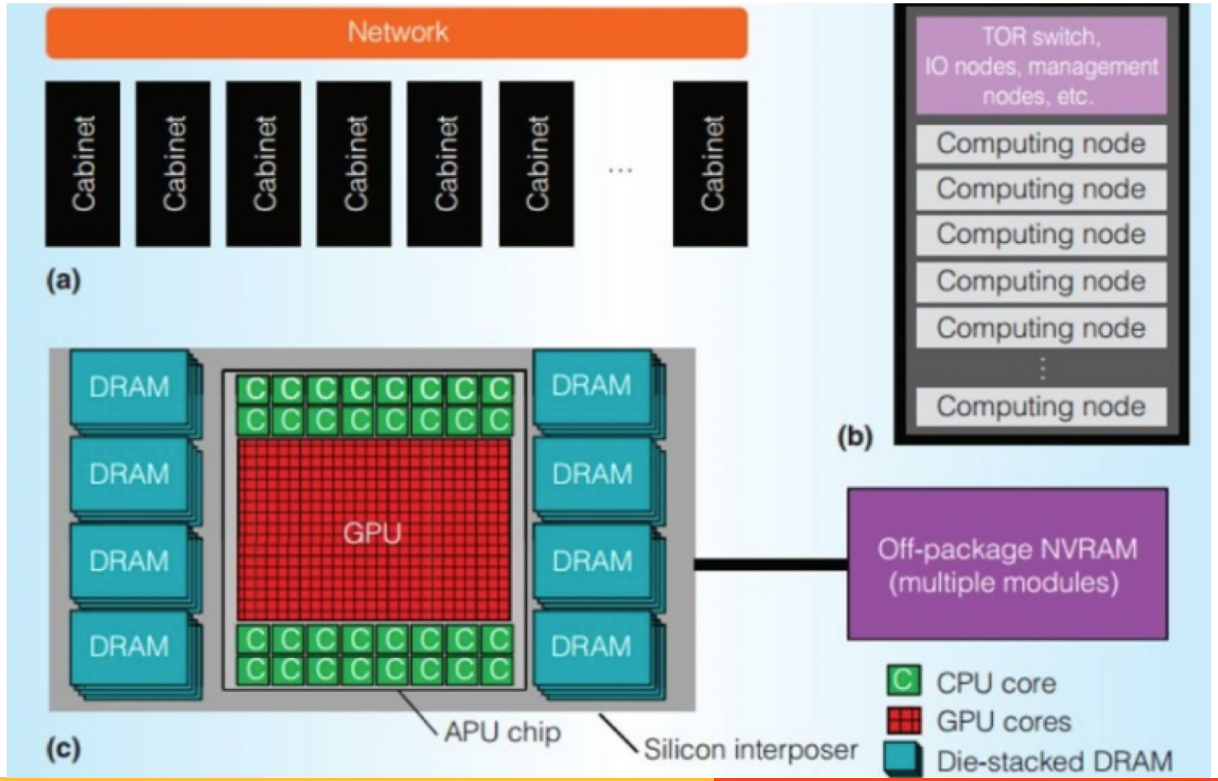






CPU AMD: Adaptar el procesador a la computación

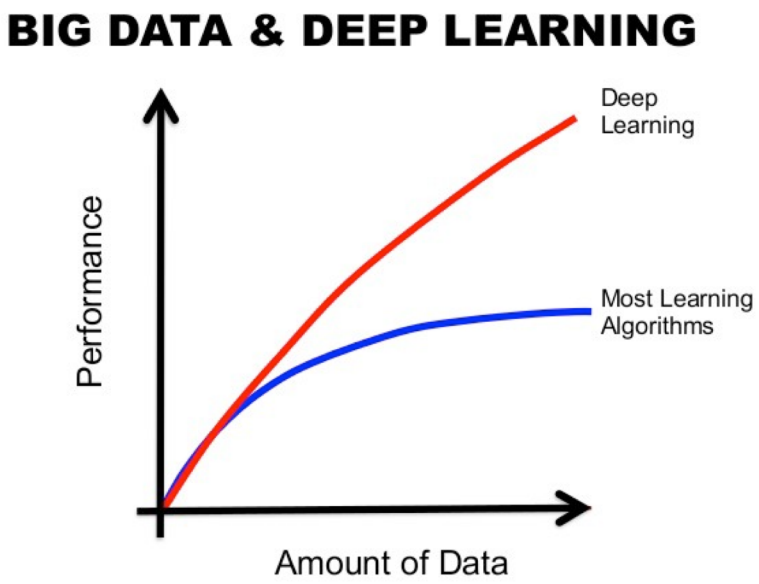
AMD Exascale Heterogeneous Processor: CPU, GPU y memoria HBM2 en una CPU



Caso de integración: Deep Learning usando GPUS con TensorFlow y Spark

Deep learning uses general learning algorithms

- The algorithms need to build the layers of an artificial neural network
  - Training data
- Processing this training data requires lots of computation
  - Convolutional NN -> Matrix multiplications



Source: https://towardsdatascience.com/7-practical-deep-learning-tips-97a9f514100e

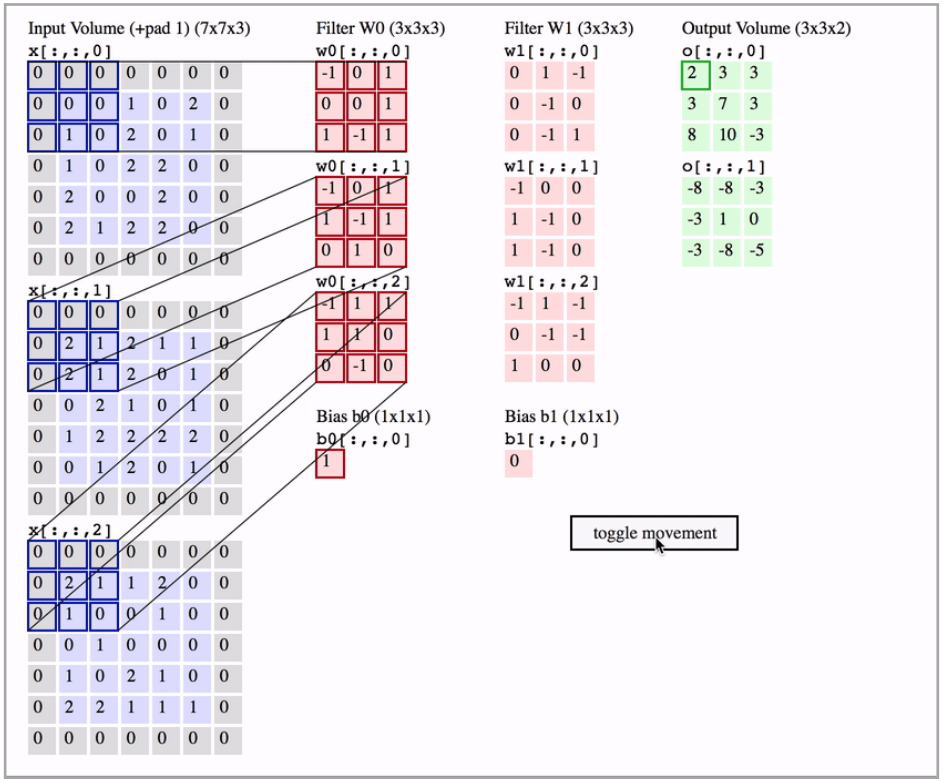
Deep Learning usando GPUs con TensorFlow y Spark

Se crea una red neuronal Convolucional

Operación basica:

$$\begin{matrix} & \vec{b}_1 & \vec{b}_2 \\ & \downarrow & \downarrow \\ \vec{a}_1 \rightarrow & & \\ \vec{a}_2 \rightarrow & & \end{matrix} \begin{bmatrix} 1 & 7 \\ 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} 3 & 3 \\ 5 & 2 \end{bmatrix} = \begin{bmatrix} \vec{a}_1 \cdot \vec{b}_1 & \vec{a}_1 \cdot \vec{b}_2 \\ \vec{a}_2 \cdot \vec{b}_1 & \vec{a}_2 \cdot \vec{b}_2 \end{bmatrix}$$

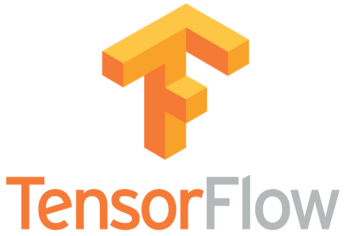
$A \qquad B \qquad C$



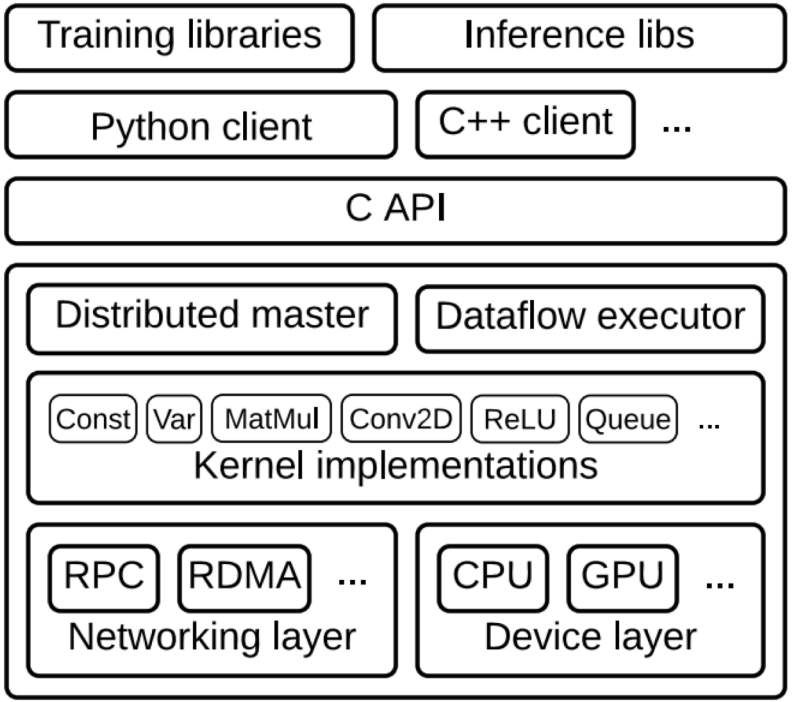
Source: https://medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207

Integración: Deep Learning usando GPUs con TensorFlow y Spark

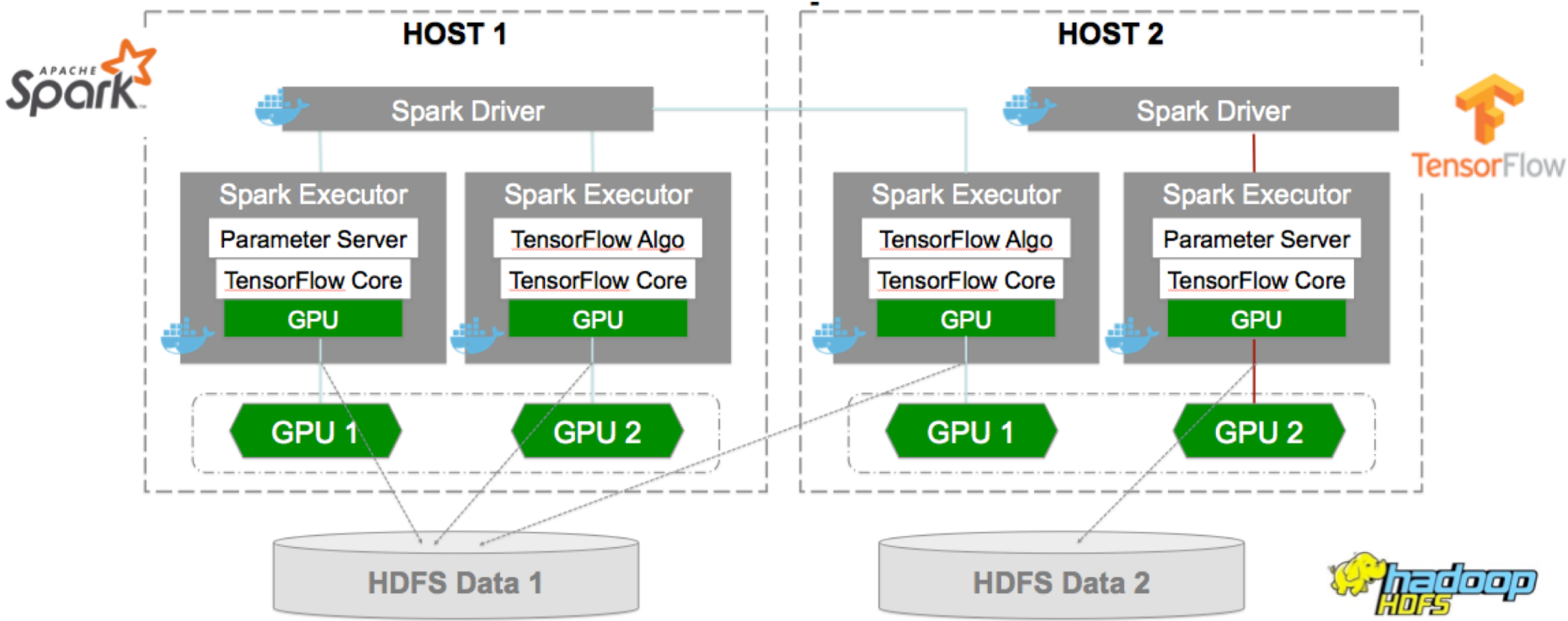
Arquitectura de TensorFlow



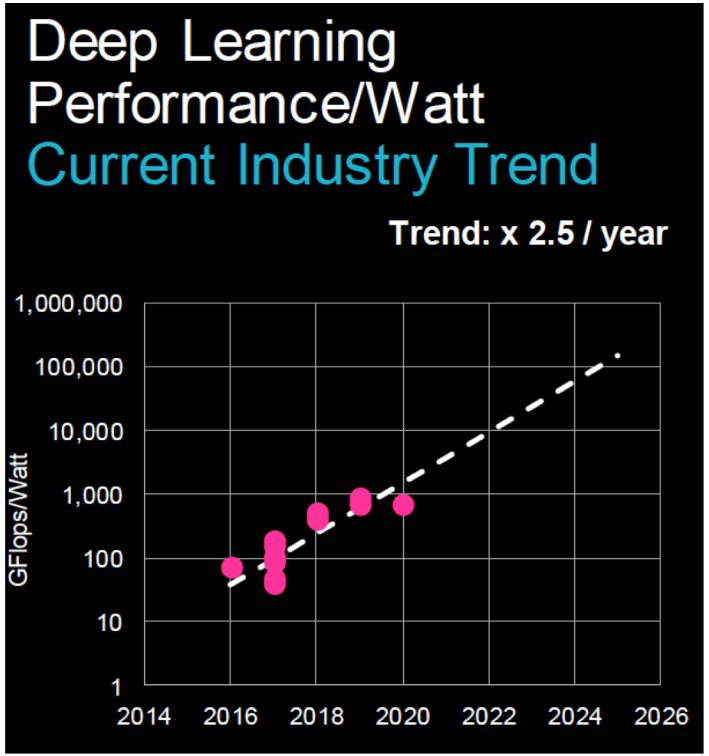
Source: www.tensorflow.org/extend/architecture



Deep Learning with TensorFlow and Spark



Deep Learning: GPU and FPGA



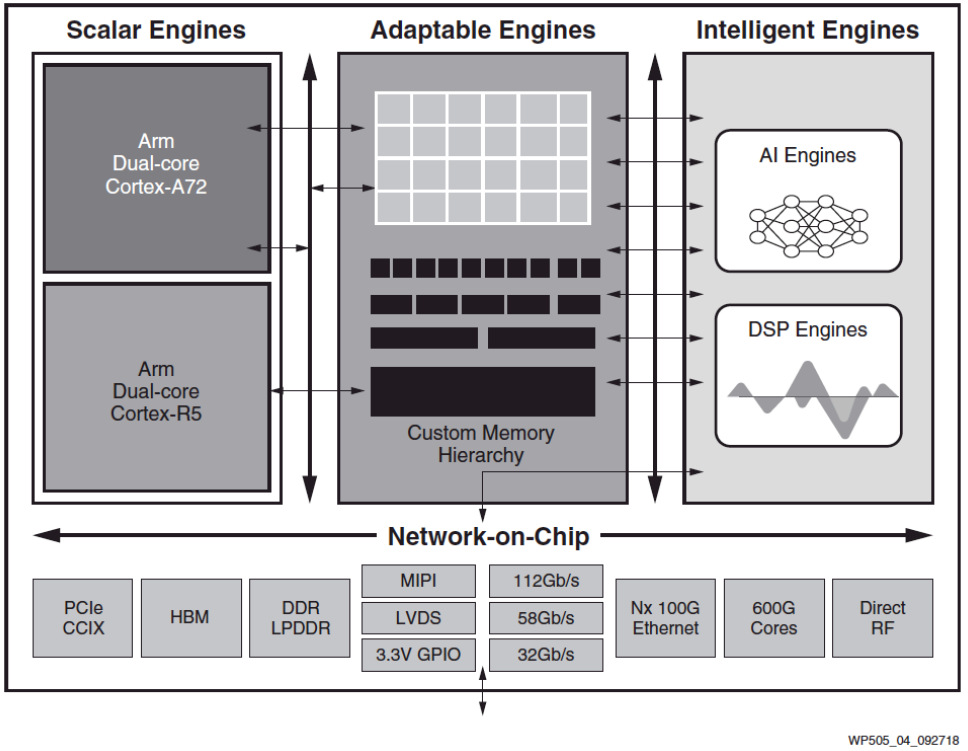
Los aceleradores actuales y los que se van utilizar a medio plaza para deep Learning están basados en tecnología CMOS

- Evolución de GPUs.
- Nuevas posibilidades con FPGAs
- Diseño de ASICs específicos.

Arquitecturas para BigData: Coprocesadores FPGA

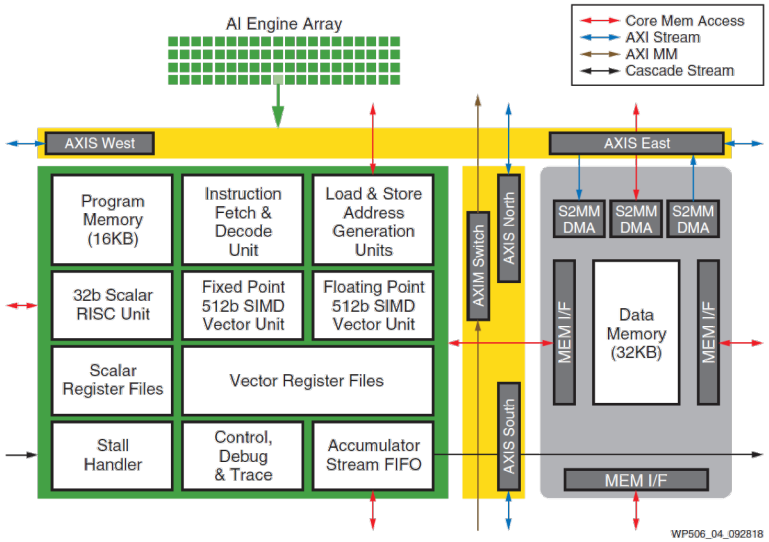
Hardware Reconfigurable
Capaz de integrar todas las necesidades de computación:

Adaptive
Compute
Acceleration
Platform (ACAP)

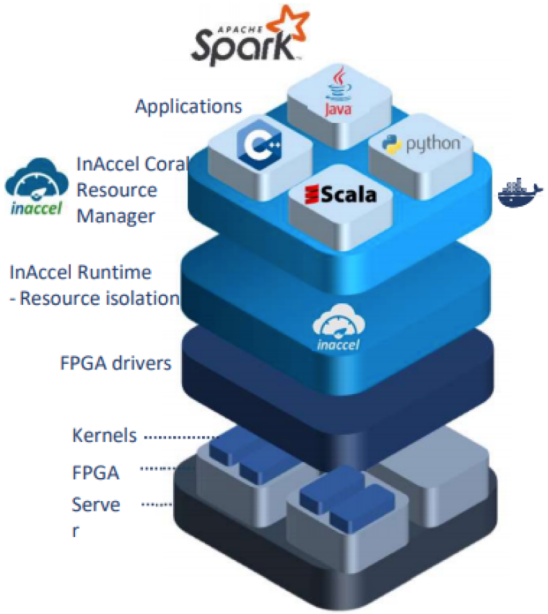


FPGA Xilinx versal: Processing System

- Dual-core ARM A72 with 2x single-threaded performance of previous generation A53's
- Dual-core ARM R5 for real-time and deterministic processing
- Adaptable resources (FPGA)
- AI Engine



Coprocesadores FPGA



InAccel Coral:

- Program against your FPGAs like it's a single pool of accelerators
- "automated deployment, scaling, and management of FPGAs"

> **Up to 15x speedup for LR ML (7.5x overall)**

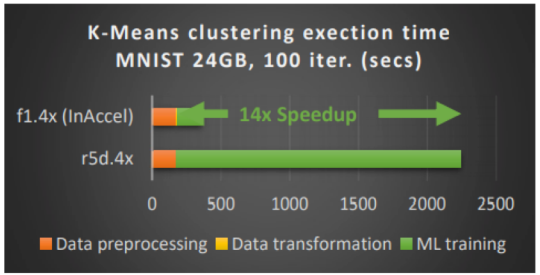
> **Up to 14x speedup for Kmeans ML (6.2x overall)**

> **Spark- GPU* (3.8x – 5.7x)**

> **F1.4x**
>> 16 cores + 2 FPGAs (InAccel)

> **R5d.4x**
>> 16 cores

*[Spark-GPU: An Accelerated In-Memory Data Processing Engine on Clusters]



Coprocesadores FPGA

The main differentiating factor is that **can be reconfigured** as opposed to the other chips:

- It allows for specifying hardware description language (HDL) that can be in turn configured in a way that matches the requirements of specific tasks or applications.
- It is known to consume less power and offer better performance.
- It can also offer a cost-effective option for prototype. It is much more flexible and is, therefore, a good choice for applications that involve customer-centric applications

Advantages

- It is highly flexible and is suited for rapidly growing and changing AI applications. For instance, with neural networks improving, it provides an architecture to undergo changes
- It shows better performance and consumption ratio
- Offers high accuracy
- FPGA shows efficiency in parallel processing. Overall it has significantly higher computer capability
- FPGAs offer lower latency than GPUs

Disadvantages

- Difficult to program and Development time is more
- Performance may not be up to the mark sometimes
- Not good for floating-point operation.

Infraestructura para BigData: Repaso de conceptos

Debe entender y ser capaz de responder :

- ¿Qué es?
- ¿Qué se mejora?
- ¿Cuándo tiene sentido usarlo y que implica?

Para los siguientes conceptos:

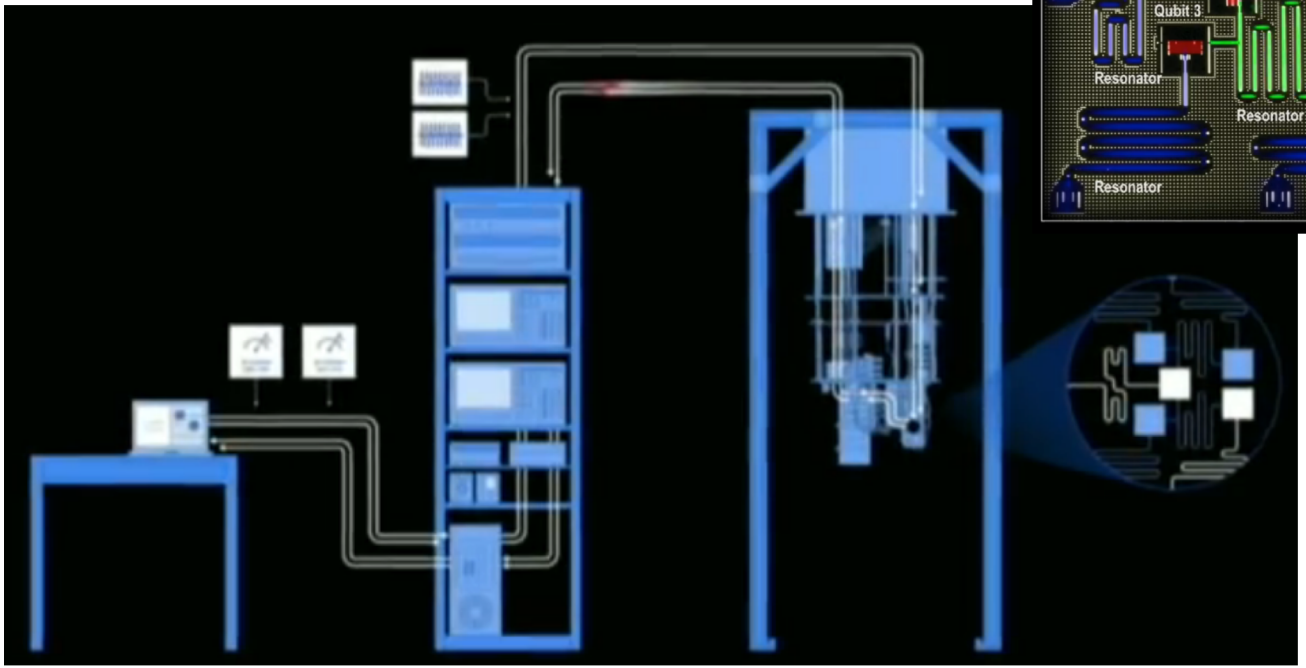
- ✓ GPUs vs CPU
- ✓ Modelo de programación GPU
- ✓ FPGA
- ✓ Casos de optimización con GPU (TensorFlow)

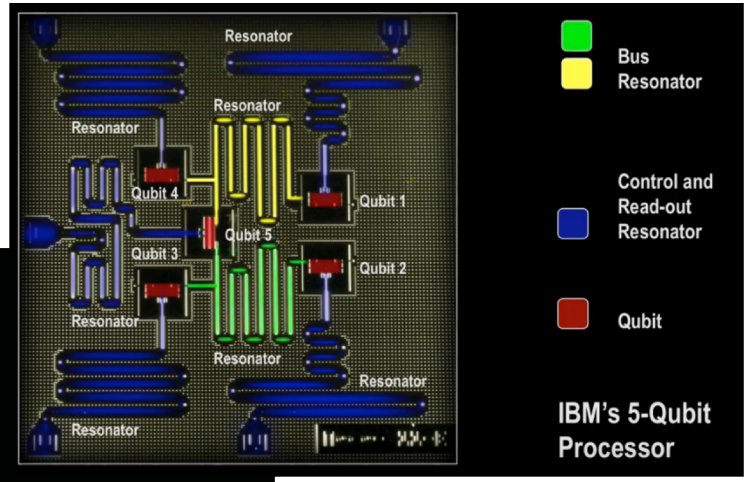
Evolución de las Tecnologías para BigData

- Tecnologías actuales de computación para BigData
  - Sistemas Multicore
  - Coprocesadores: GPUs, FPGA
- **Tecnologías disruptivas:**
  - Neurocomputación
  - **Computación Cuántica**

Computador cuántico de IBM

Join the IBM Q Experience Community
https://quantumexperience.ng.bluemix.net





Referencias

1. ACCELERATING APACHE SPARK MACHINE LEARNING WITH CLEAR LINUX* OS FOR INTEL ARCHITECTURE® AND INTEL SOFTWARE OPTIMIZATIONS.https://01.org/blogs/2018/apache-spark-clear-linux/

2.- Architectural Impact on Performance of In-memory Data Analytics: Apache Spark Case Study

3.- Ref: Running Apache Spark on a High-Performance Cluster Using RDMA and NVMe Flash por Patrick Stuedi, IBM Research