빅데이터 분석을 위한 가상화와 클라우드 기술

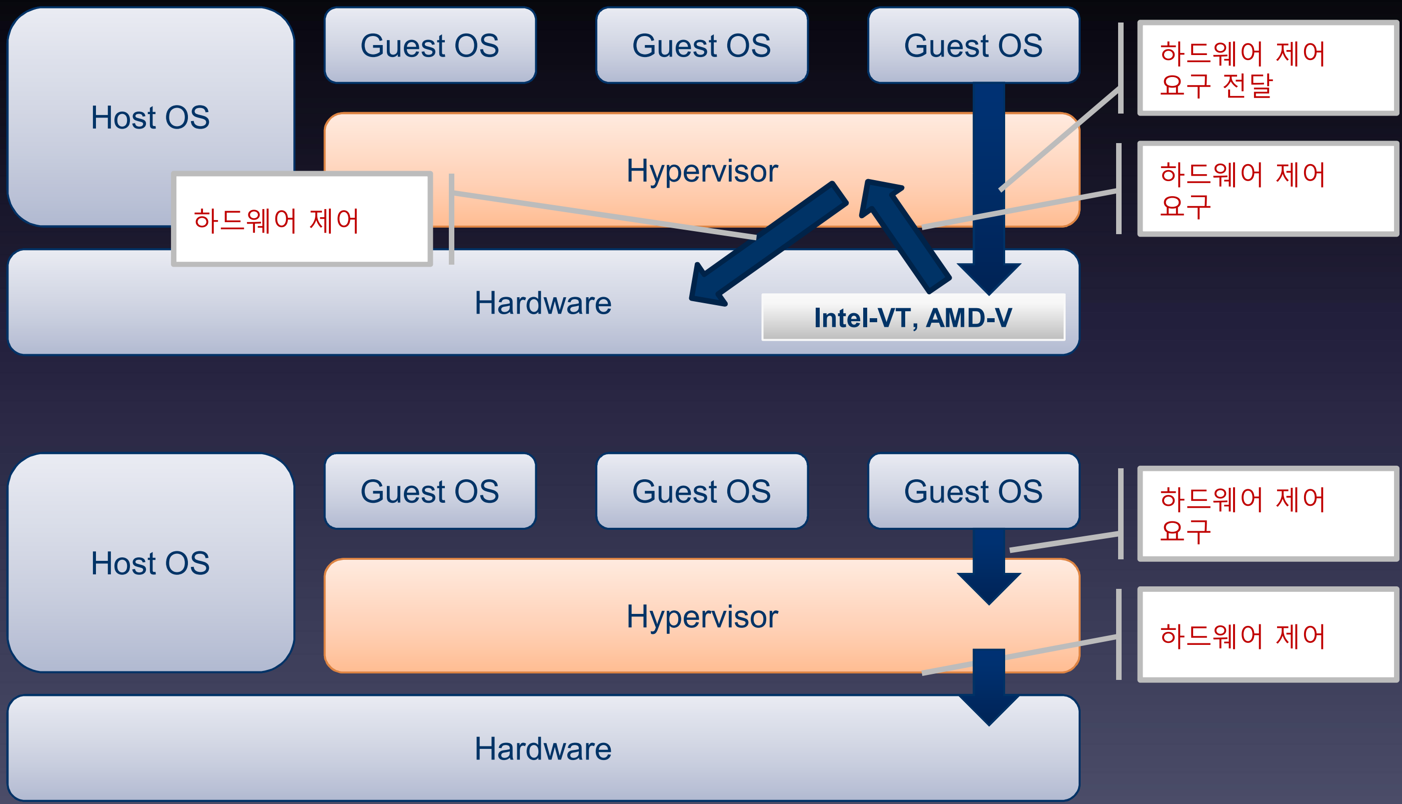
Big Data와 클라우드 컴퓨팅의 만남.

Cloud & Virtualization

- Virtualization? Abstraction?
- Virtualization Category
  – Server Virtualization (VMWare, Xen, KVM…)
  – Storage Virtualization (iscsi, scalable NAS)
  – Network Virtualization.

Server Virtualization

- OLD Term "Full" vs "Para-Virtualization"



Server Virtualization Performance

- System Setup
- Eucalyptus and Xen based private cloud infrastructure
  - Eucalyptus version 1.4 and Xen version 3.0.3
  - Deployed on 16 nodes (2 Quad Core, 32 GB of memory)
  - All nodes are connected via a 1GB connections
- Bare-metal and VMs use exactly the same software environments
  - Red Hat Enterprise Linux Server release operating system. OpenMPI version 1.3.2 with gcc version 4.1.2.

MPI Applications

Application	Matrix multiplication	Kmeans Clustering	Concurrent Wave Equation
Description	Implements Cannon's Algorithm Assume a rectangular process grid (Figure 1-left)	Implements Kmeans Clustering Algorithm Fixed number of iterations are performed for each test	A vibrating string is decomposed (split) into points, and each MPI process is responsible for updating the amplitude of a number of points over time.
Grain size (n)	Number of points in a matrix block handled by each MPI process	Number of data points handled by a single MPI process	Number of points handled by each MPI process
Communication Pattern	Each MPI process communicates with its neighbors in both row wise and column wise.	All MPI processes send partial clusters to one MPI process (rank 0). Rank 0 distribute the new cluster centers to all the nodes	In each iteration, each MPI process exchanges boundary points with its nearest neighbors.
Computation per MPI process	$O((\sqrt{n})^3)$	$O(n)$	$O(n)$
Communication per MPI process	$O((\sqrt{n})^2)$	$O(1)$	$O(1)$
C/C	$O\left(\frac{1}{\sqrt{n}}\right)$	$O\left(\frac{1}{n}\right)$	$O\left(\frac{1}{n}\right)$
Message Size	$(\sqrt{n})^2 = n$	*D – Where D is the number of cluster centers.* $D \ll n$	Each message contains a double value
Communication routines used	*MPI_Sendrecv_replace()*	*MPI_Reduce()* *MPI_Bcast()*	*MPI_Sendrecv()*

Different Hardware/VM configurations

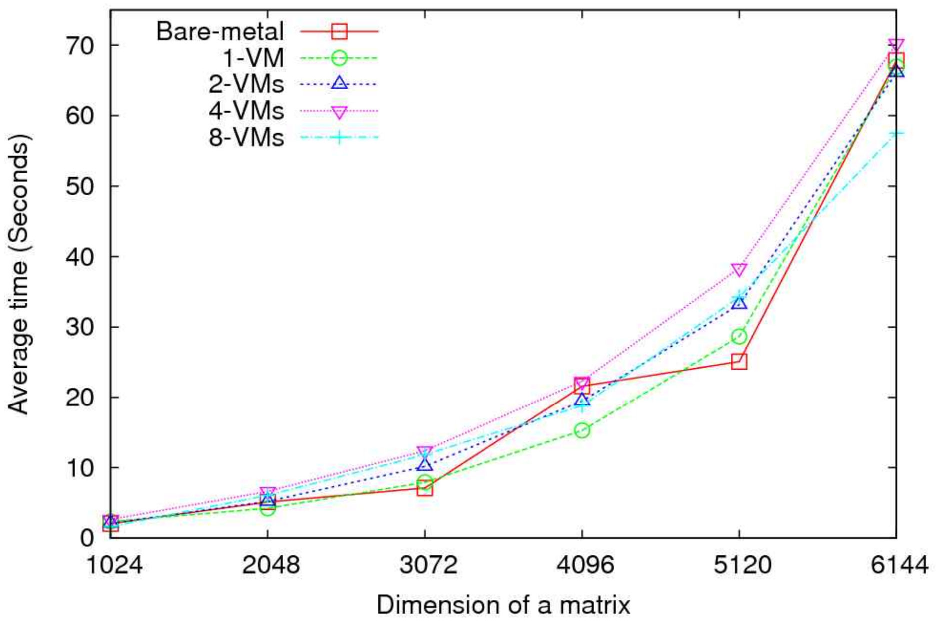
Ref	Description	Number of CPU cores accessible to the virtual or bare-metal node	Amount of memory (GB) accessible to the virtual or bare-metal node	Number of virtual or bare-metal nodes deployed
BM	Bare-metal node	8	32	16
1-VM-8-core	1 VM instance per bare-metal node	8	30 (2GB is reserved for Domo)	16
2-VM-4-core	2 VM instances per bare-metal node	4	15	32
4-VM-2-core	4 VM instances per bare-metal node	2	7.5	64
8-VM-1-core	8 VM instances per bare-metal node	1	3.75	128

- Invariant used in selecting the number of MPI processes

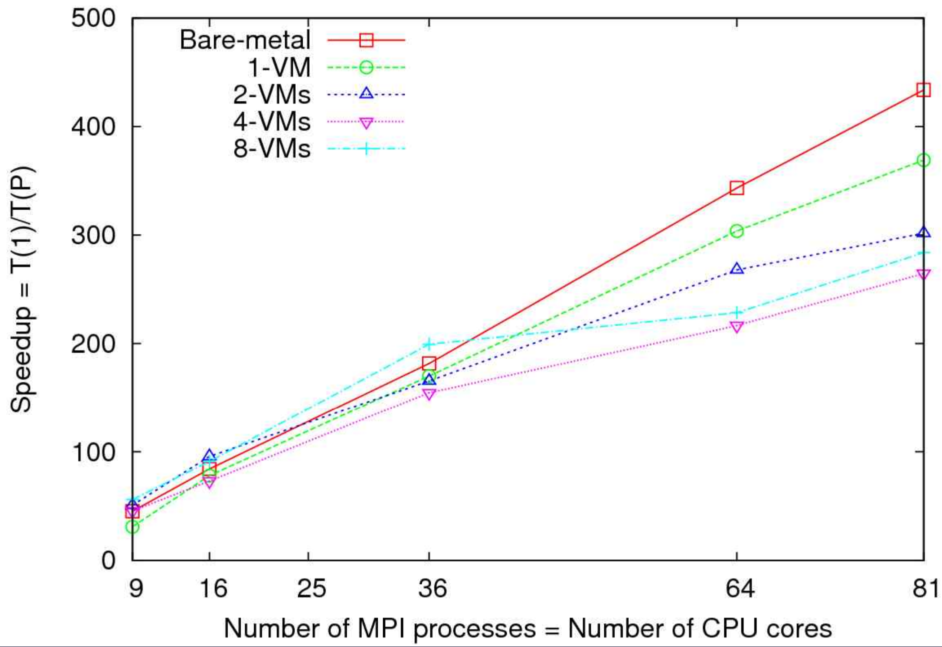
Number of MPI processes = Number of CPU cores used

Matrix Multiplication

Performance - 64 CPU cores



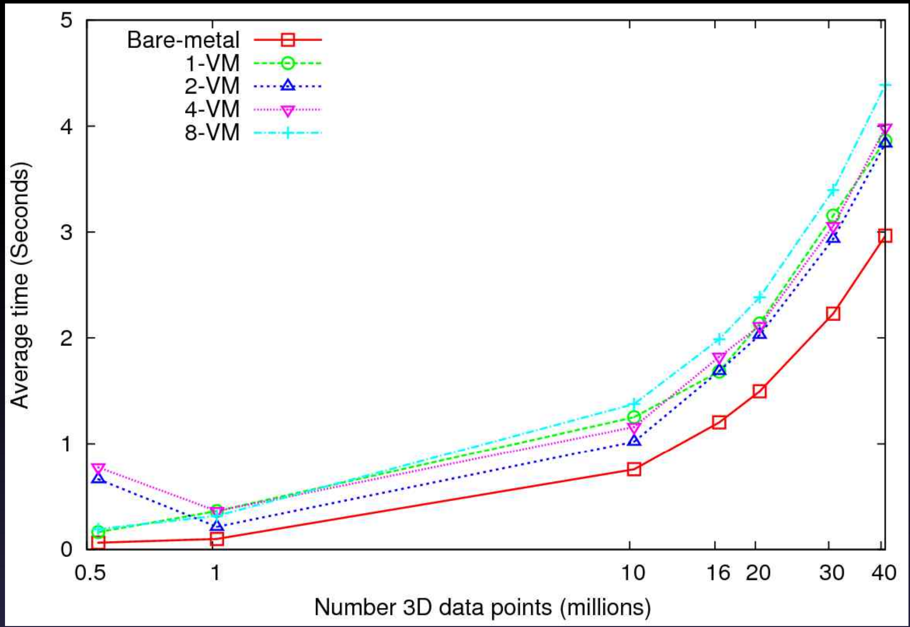
Speedup – Fixed matrix size (5184x5184)



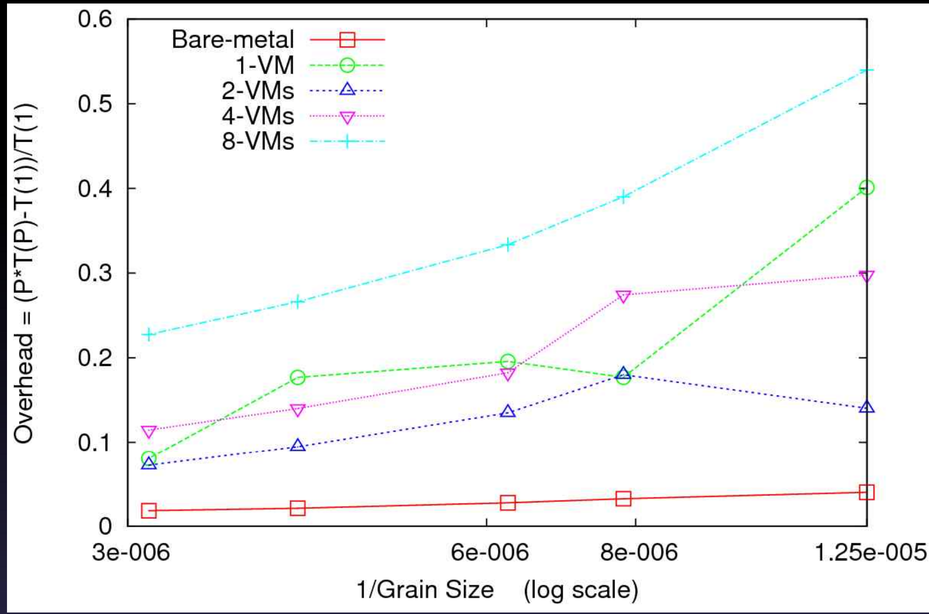
- Implements Cannon's Algorithm
- Exchange large messages
- More susceptible to bandwidth than latency
- At 81 MPI processes, at least 14% reduction in speedup is noticeable

Kmeans Clustering

Performance – 128 CPU cores



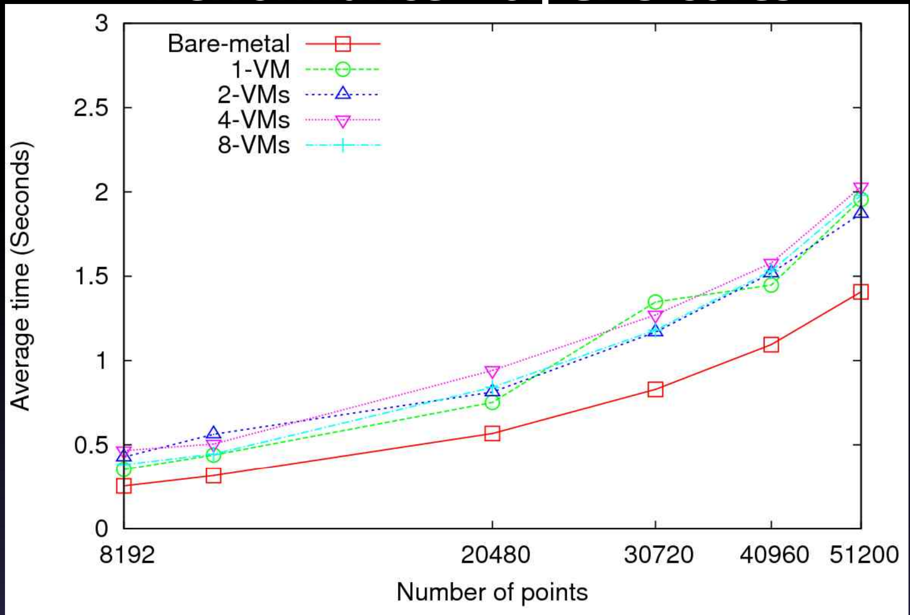
Overhead



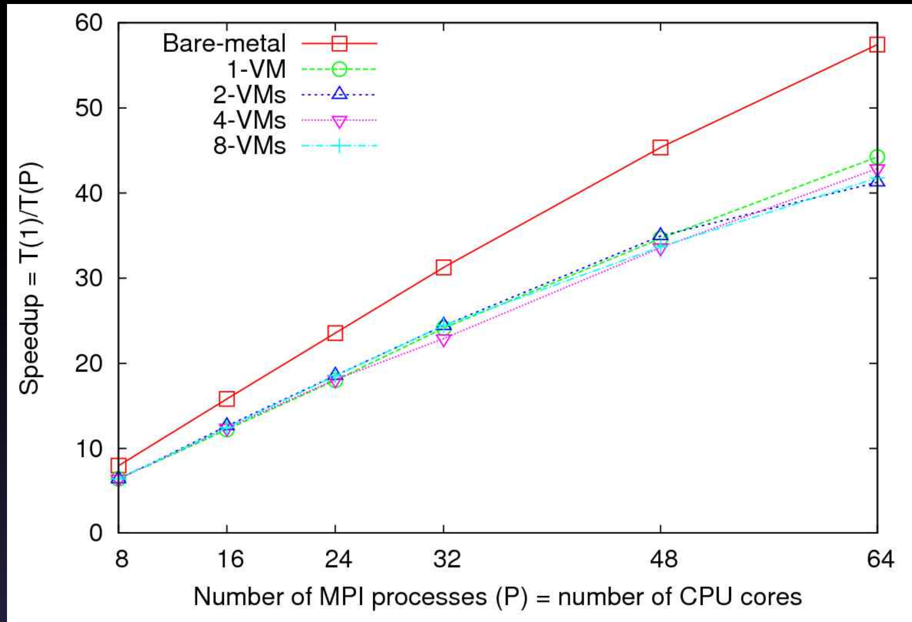
- Perform Kmeans clustering for up to 40 million 3D data points
- Amount of communication depends only on the number of cluster centers
- Amount of communication << Computation and the amount of data processed
- At the highest granularity VMs show at least 3.5 times overhead compared to bare-metal
- Extremely large overheads for smaller grain sizes

Concurrent Wave Equation Solver

Performance - 64 CPU cores

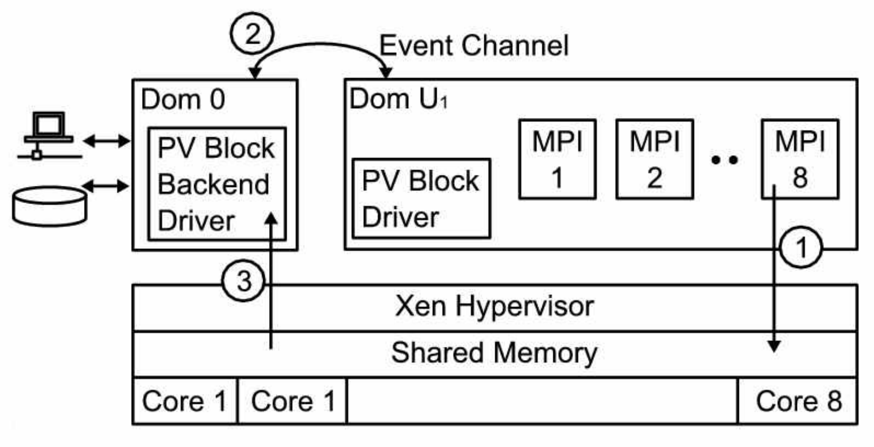


Total Speedup – 30720 data points

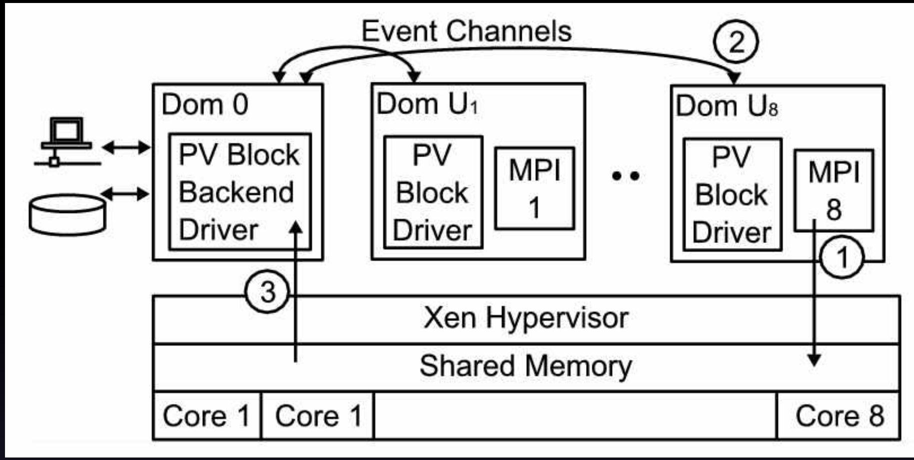


- Clear difference in performance and speedups between VMs and bare-metal
- Very small messages (the message size in each *MPI_Sendrecv()* call is only 8 bytes)
- More susceptible to latency
- At 51200 data points, at least 40% decrease in performance is observed in VMs

Higher latencies -1



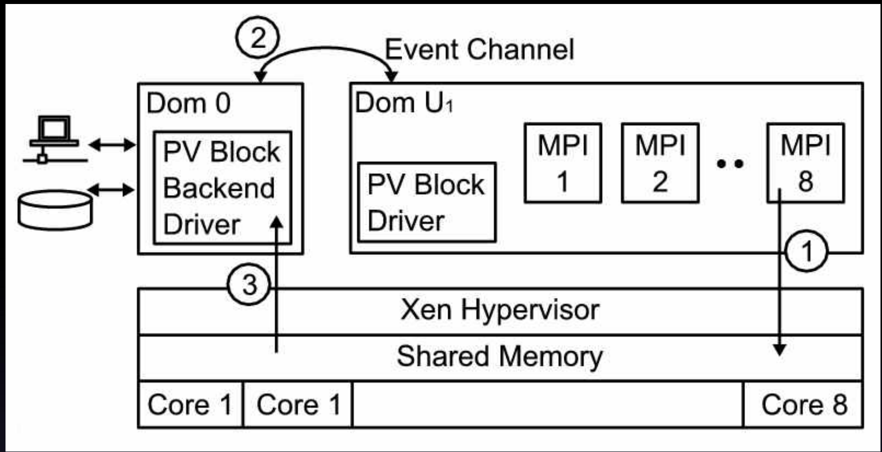
Xen configuration for 1-VM per node
8 MPI processes inside the VM



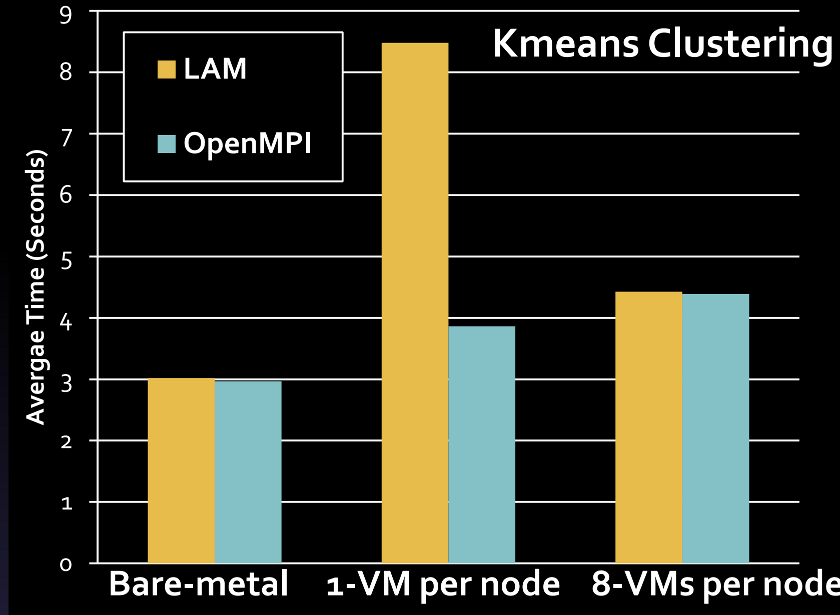
Xen configuration for 8-VMs per node
1 MPI process inside each VM

- *domUs* (VMs that run on top of Xen para-virtualization) are not capable of performing I/O operations
- *domo* (privileged OS) schedules and executes I/O operations on behalf of *domUs*
- More VMs per node => more scheduling => higher latencies

Higher latencies -2

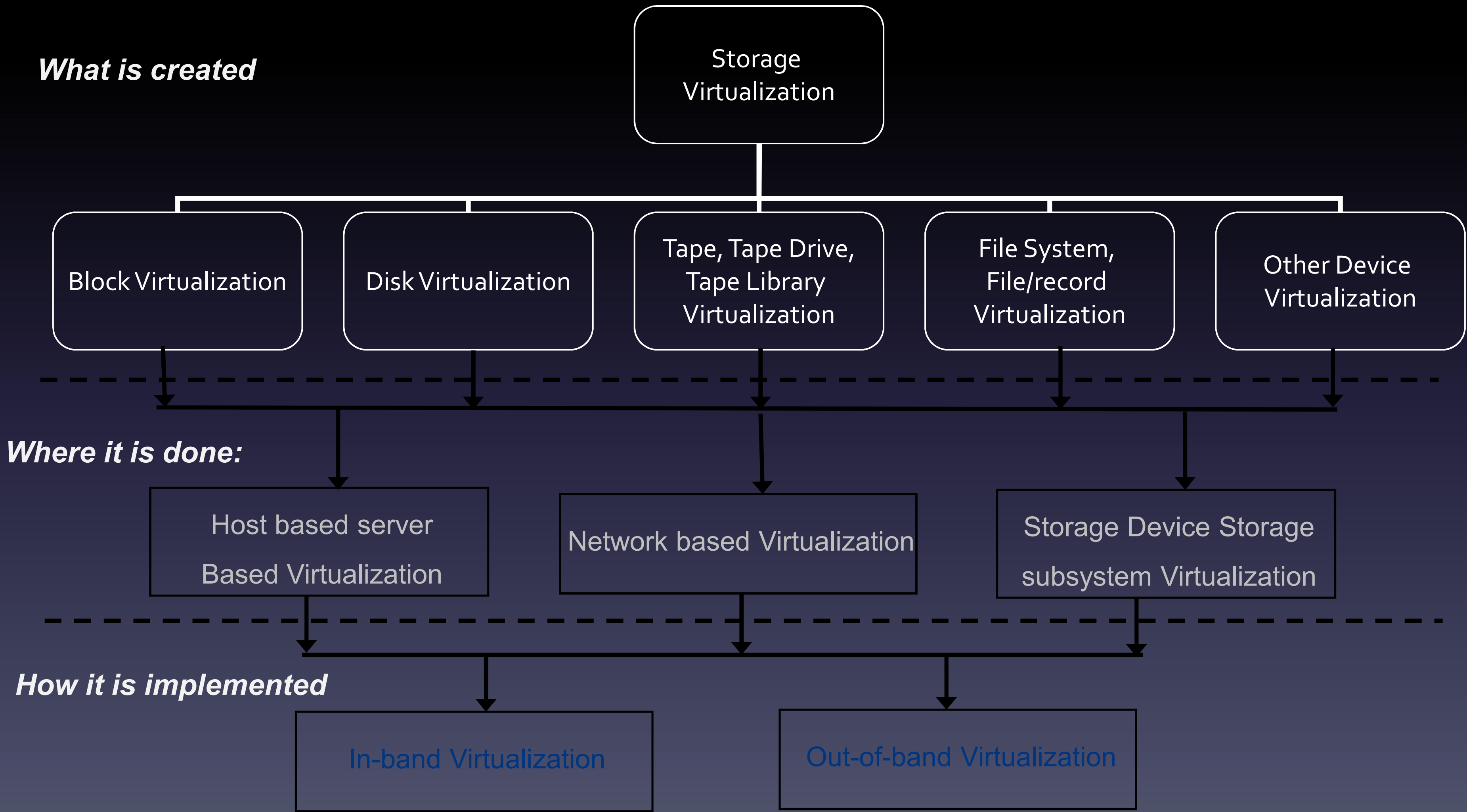


Xen configuration for 1-VM per node
8 MPI processes inside the VM



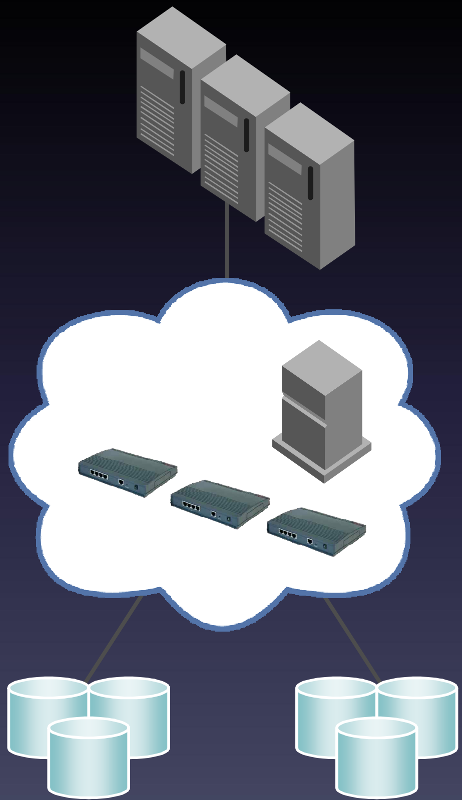
- Lack of support for in-node communication => “Sequentializing” parallel communication
- Better support for in-node communication in OpenMPI resulted better performance than LAM-MPI for 1-VM per node configuration
- In 8-VMs per node, 1 MPI process per VM configuration, both OpenMPI and LAM-MPI perform equally well

Storage Virtualization



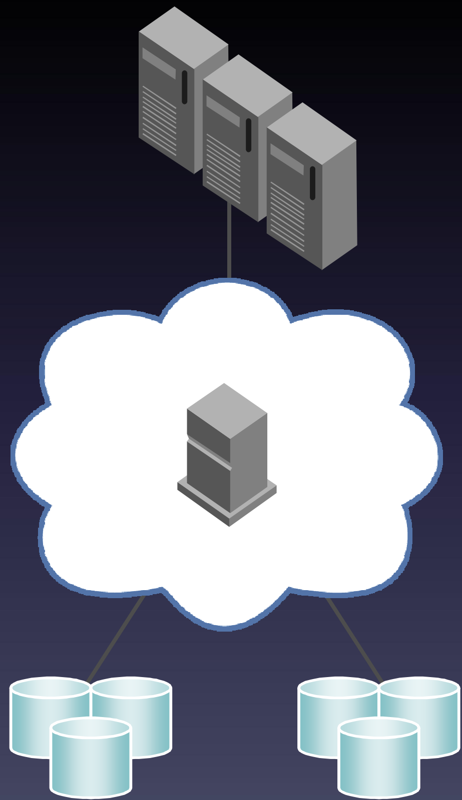
Comparison of Virtualization Architectures

Out-of-Band



- No state / no cache
- I/O at wire speed
- Full-fabric bandwidth
- High availability
- High scalability
- Value-add functionality

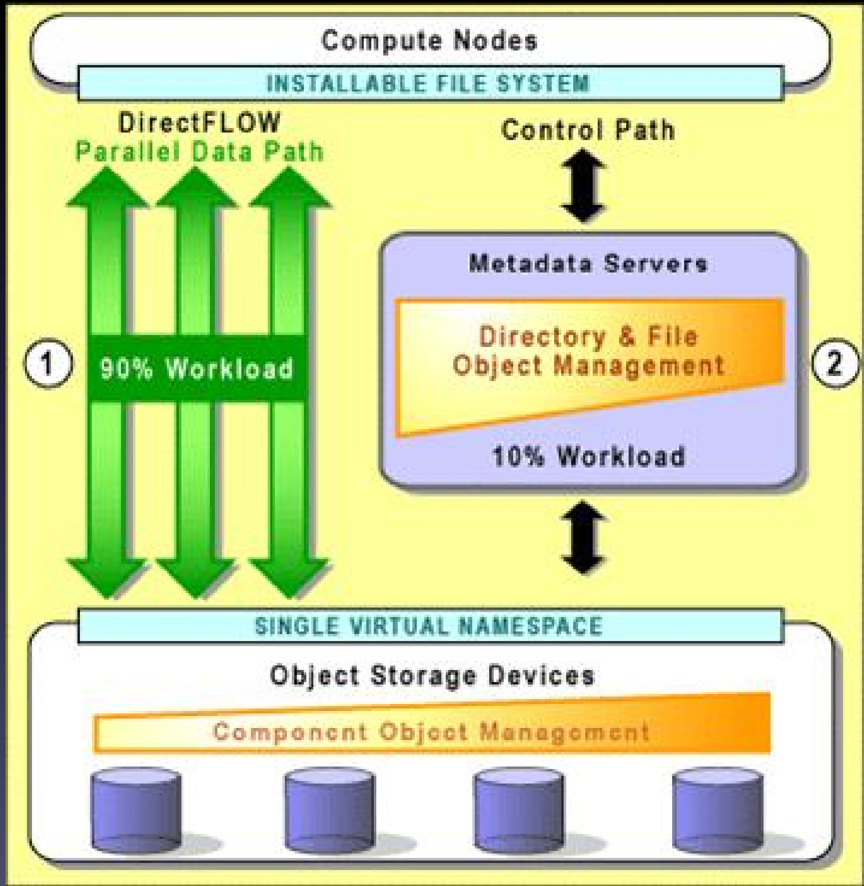
In-Band



- State / cache
- I/O latency
- Limited fabric ports
- More suited for static environments or environments with less growth
- Value-replace functionality

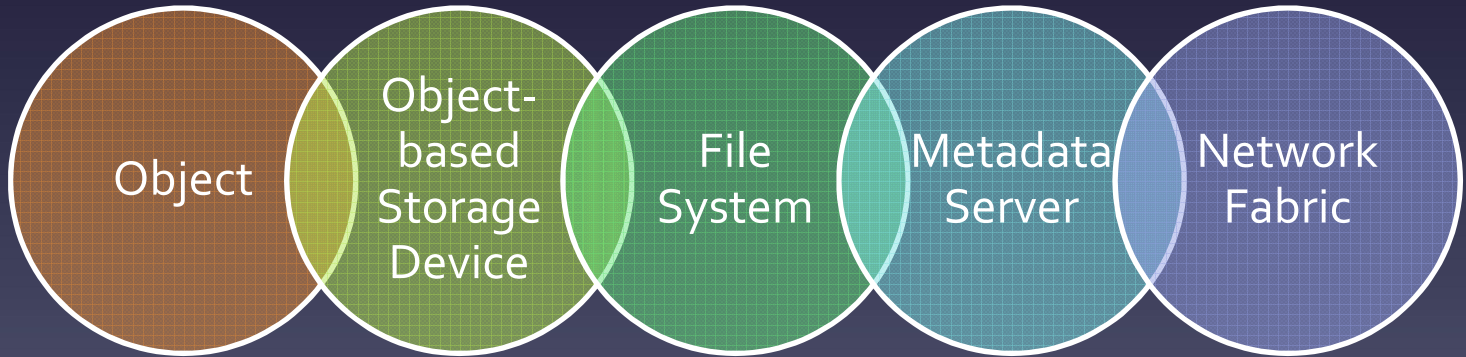
File Based Storage virtualization Object-Based Storage Arch.

- Provide Method for allowing compute nodes to access storage devices directly in parallel
  - Object Stroage Device: network-attached device containing media, disk/tapes and intelligence
- Distributes the system metadata allowing shared file access without a central bottle neck.
  - Divides logical view of the stored data from phisical view

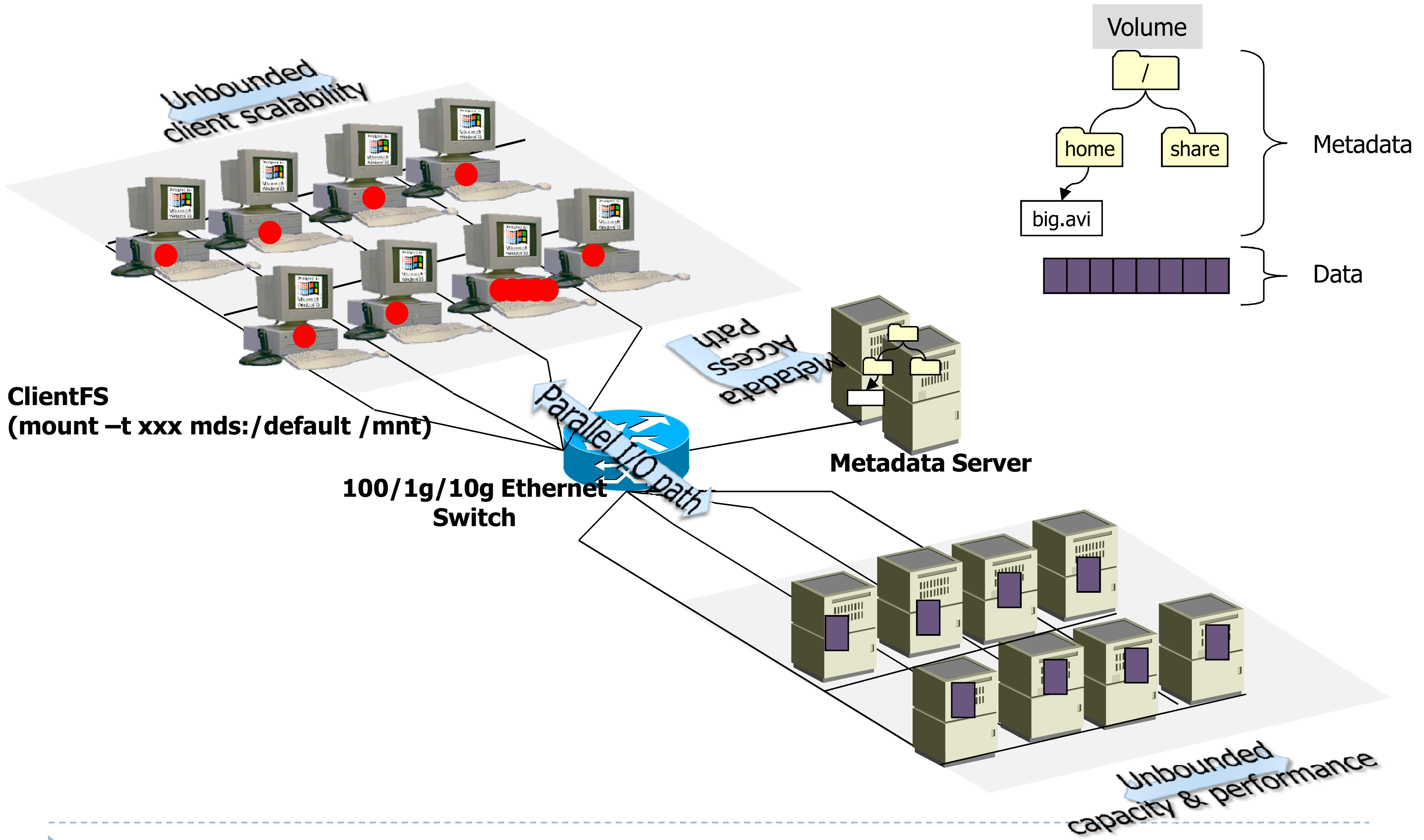


Object Storage Components

- Object
- Object-based Storage Device
- Installable File System.
- Metadata Server
- Network Fabric

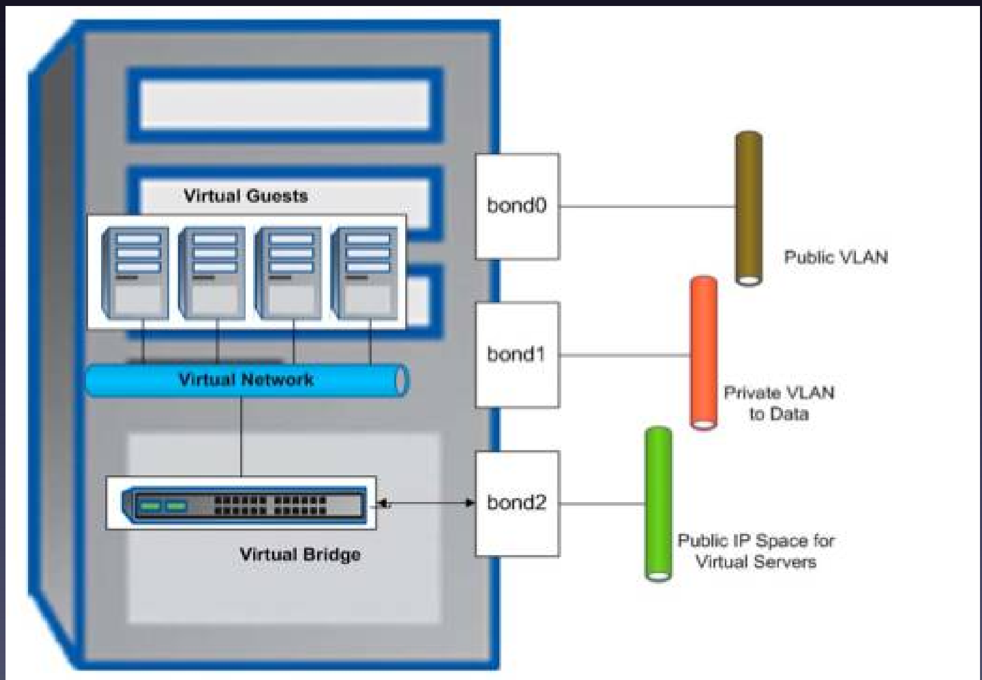


Glory File System.



Network Virtualization

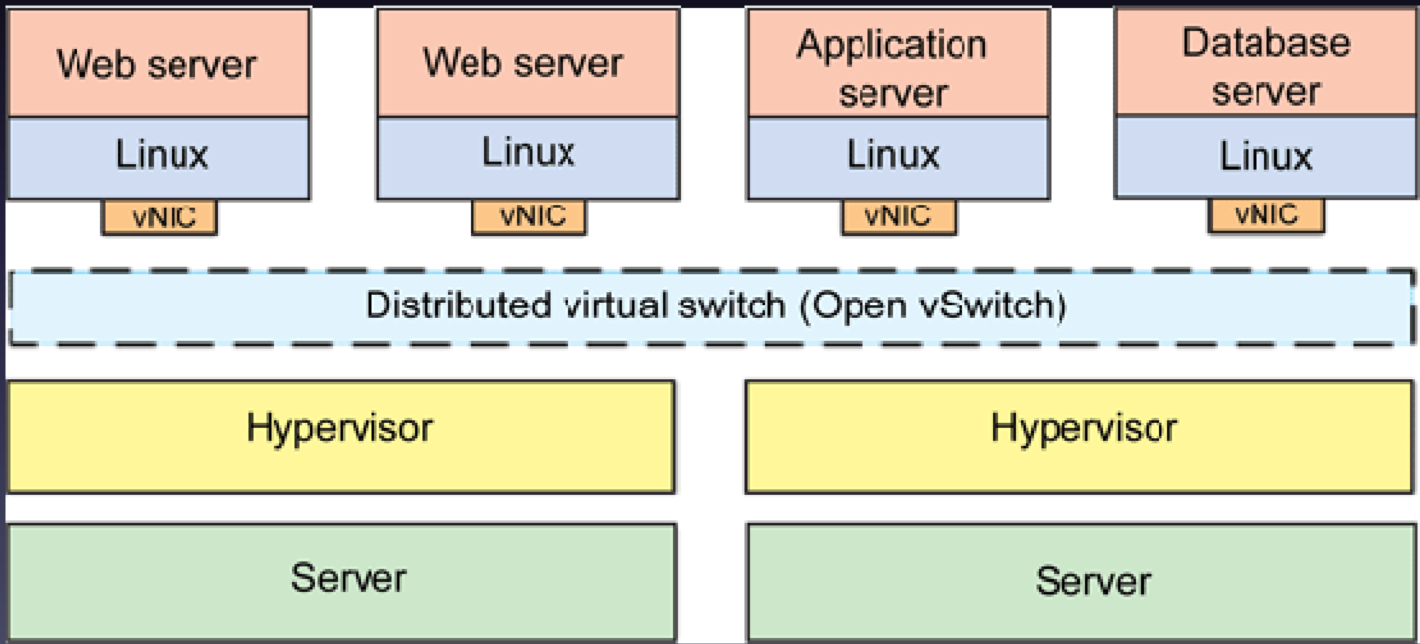
- Network Virtualization
  - 1st generation: Using Linux Kernel bridge



source: VMware

Network Virtualization

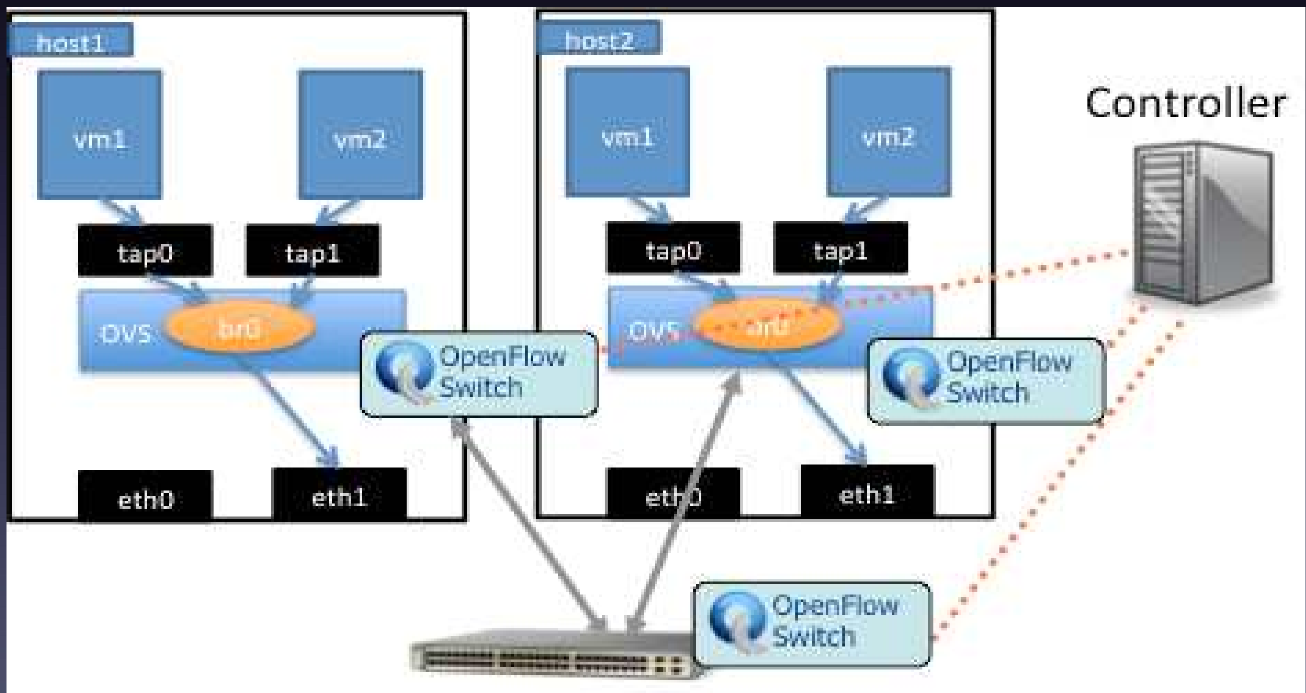
- Network Virtualization
  - 2nd generation: Using Virtual switch



source: IBM

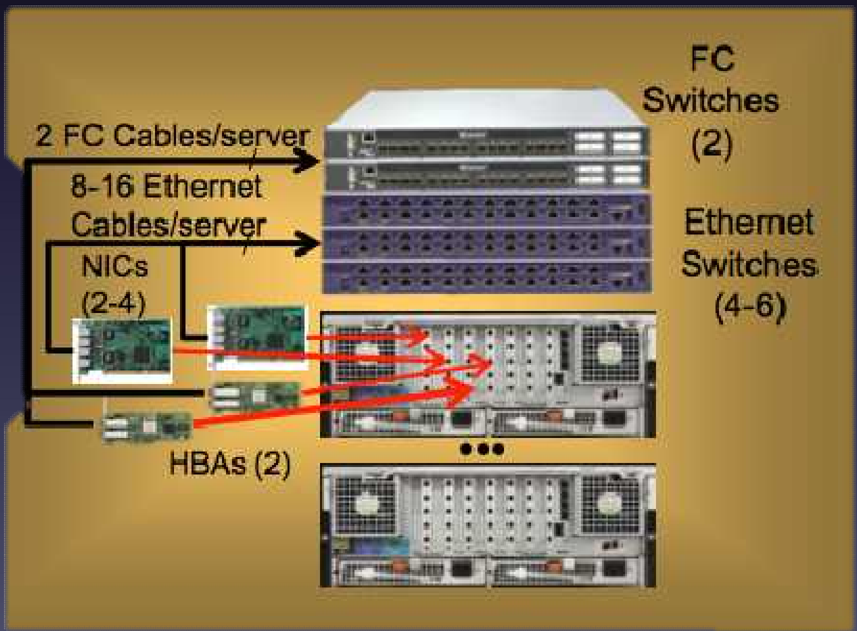
Network Virtualization

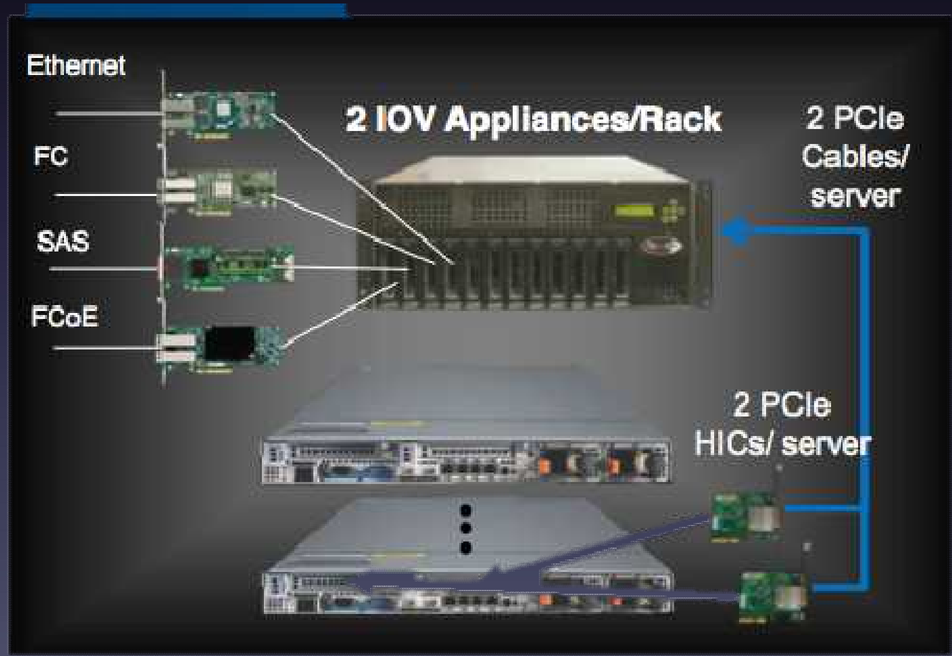
- Network Virtualization
  - Next generation: Virtual switch + OpenFlow



I/O Virtualization

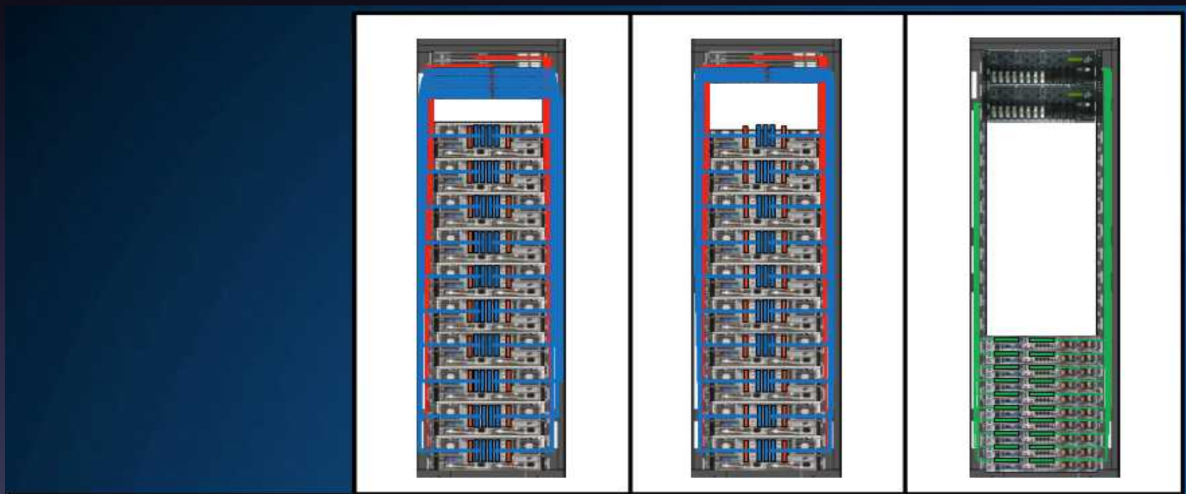
- I/O device of physical server can be virtualized.





I/O Virtualization

- I/O device of physical server can be virtualized.



Solution	FC + GbE	FC + 10GbE	vNET
TOR Switches	8 (2xFC, 4xGbE)	4 (2xFC, 2x10GbE)	2 (vNETs)
Server Cards	60 (20xFC, 40xQuad GbE)	50 (20xFC, 30x10GbE)	20 (Passive PCIe HICs)
Cables	180 (20xFC, 160xCAT5)	50 (20xFC, 30x10GbE)	20 (PCIe)
Total Rack Space	48U	46U	28U