

Teratec :

“Au dela des approches traditionnelles”
Cloud, Big Data & HPC convergence

29-06-2016

HPC

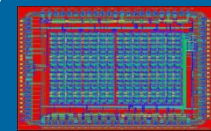
Specialized
Coprocesing

Low power,
efficient , scalable

Performant, flexible, reliable
compute and data flows

Performant, flexible
compute and data flows

Efficient caching
object storage flexibility



Accelerators
FPGA, PIM,
GPU



**Multiple
CPU type**



Interconnects



**Smart Memory
Hierarchy
Management**



**Storage
Objects
NVMe Devices**

Big Data

Machine learning

Low power
and embedded

Fast data
realtime streaming support

Insitu processing

big data

HPC

New generation of scientific applications

- with complex workflow (omics)
- New heterogeneous Resources (NVME, Copro...)
- user driven environment definition

Big Data

Fast data ,
streaming analytics capabilities
real time capabilities
Data management

Scalability

Flexible, elastic ressource scheduling

**ALGO distributed and parallel
(map/reduce, ML, Graphs,)**

Heterogeneous execution

Environnement user driven

Orchestration and management

Efficiency for Real time,

1

HPC and virtualization news expectations

Virtualisations for HPC

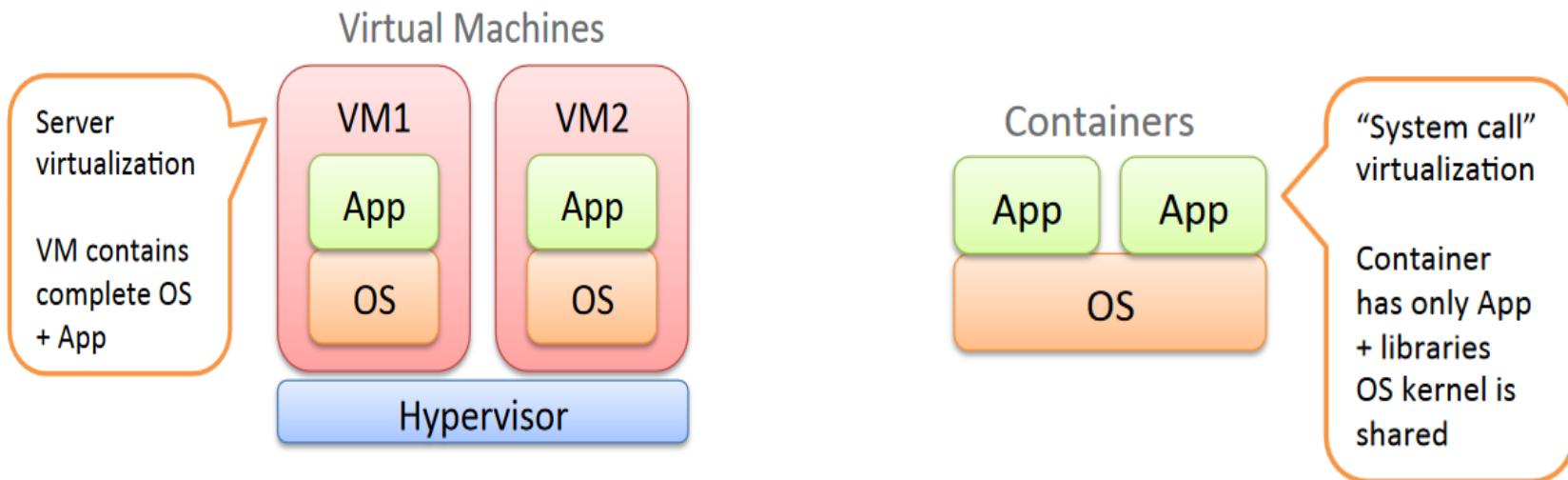
- ▶ Main Objective in an HPC context
 - control and security of application execution context
 - tasks isolation, ressources contrôles
 - Support of the diversity of application execution environnement
 - Manage diversity of the computation platform environnement
- ▶ Constraints pre-requisites:
 - Limited performance impact.
 - Strong Management of the environnement. What is executed and where?
 - optional not a pre-requisite
 - Integrated to ressources scheduler (SLURM)

Virtualisation VS containers

▶ Full system virtualized

VS

▶ System resources isolated



Sharing Node **Physical** Ressource

VS

System Ressources

Container principles: container VS Virtual Machines

	VM	Container
Hw support extensions usage(VT,Iommu, SRIOV..)	yes	NA
Software stack	Complete (from OS to app)	all except OS
Security	Complete isolation	Depending on implementations
Ressources consumption and sharing	Direct resources allocation (GPU) or sharing through virtualizations of physical resources	System ressources ,at OS level (Cgroups and namespace)
Heterogeneity/node	OS heterogeneity on same node	Lib and distrib package
Performances	Overhead for compression, IO and Network latency Boot system	Light

Containers control: cgroups

- ▶ containers integrated in linux since 2006

Control through cgroups

- Device Access
- Resource limiting
- Prioritization
- Accounting
- Control
- Injection



Containers: isolation

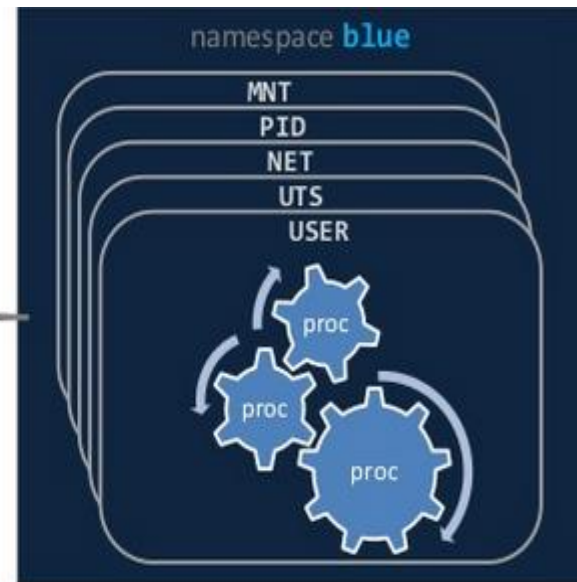
cgroups

cgroup examples

Controller	Description
blkio	Access to block devices
cpu	CPU time
cpuset	CPU cores
devices	Device access
memory	Memory usage
net_cls	Packet classification
net_prio	Packet priority

Virtualization View
Isolation namespaces

- **MNT**; mount points, files systems, etc.
- **PID**; processes
- **NET**; NICs, routing, etc.
- **IPC**; System V IPC
- **UTS**; host and domain name
- **USER**; UID and GID



Namespaces

Different namespaces = Different “Views” of the kernel

Linux 2.4.19 - 3 Aug 2002	Mount namespace	Mount Points
Linux 2.6.19 - 29 Nov 2006	UTS namespace	Hostname
	IPC namespace	Interprocess communication
Linux 2.6.24 - 24 Jan 2008	PID namespace	Processes in different PID namespace can have the same PID
	Network namespace	Network devices, IP addresses, routing tables, iptables entries
Linux 3.8 - 18 Feb 2013	User namespace	Root privileges for operations inside a user namespace, but unprivileged outside the namespace. Number of Linux filesystems are not yet user-namespace aware.

Futur: systemd cgroup unified

kernel > 3.8

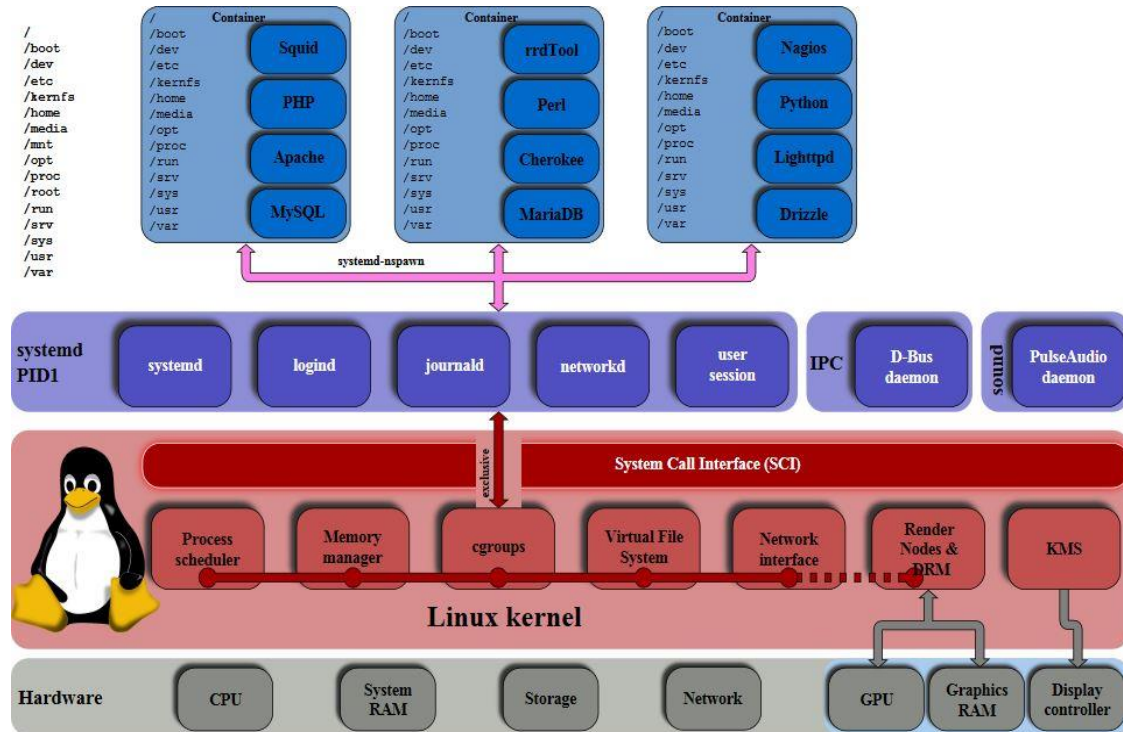
- launching container in non root user
- mapping UID GID
- Systemd-nspawn supports docker images

Linux > 4.5

- redesign Cgroups
- new API

In progress

futur micro OS and containers?



New system resource : RDMA cgroups

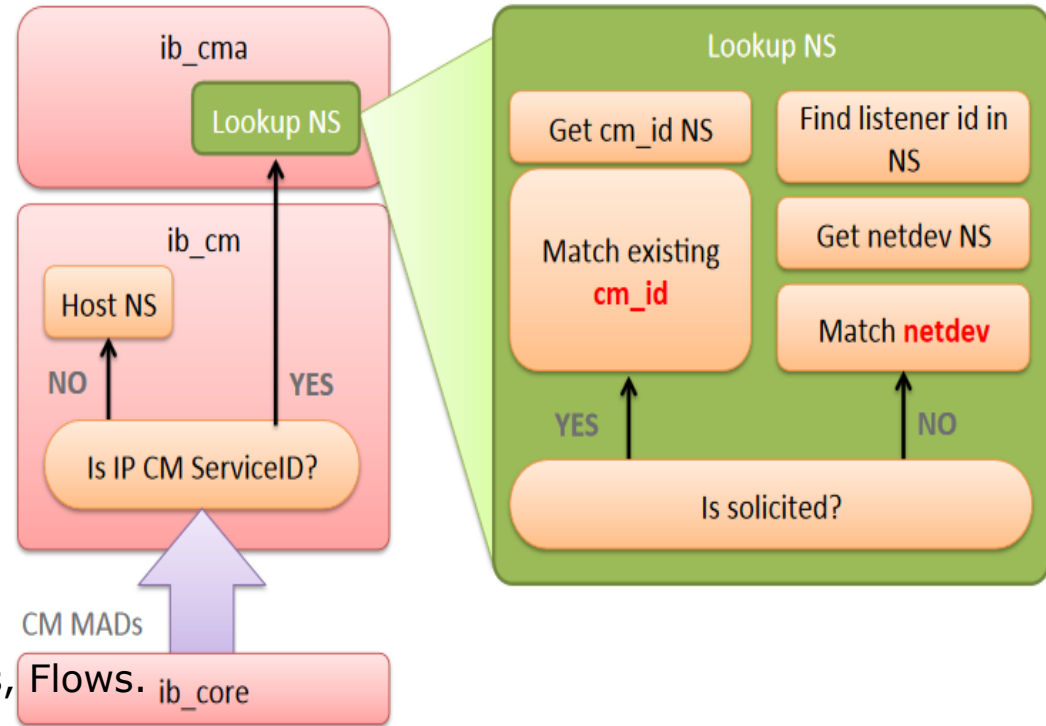
- ▶ how to manage HW resources like RDMA devices?
- ▶ > add new resources

Linux 4.4,
namespaces proposal from OFA :

- IB peripherals
- physical port
- GID
- P_Key

Cgroups

- HCA open contexts
- AHs, CQs, PDs, QPs, SRQs, MRs, MWs, Flows.



2

virtualizations
management tools

Containers use in docker



docker

To complete containers environments definition

- Storage allocation
- network configuration
- security rules

Linux Kernel

Storage

Device Mapper

Btrfs

Aufs

Namespaces

PID

MNT

IPC

UTS

NET

Networking

veth

bridge

iptables

Cgroups

cpu

cpuset

memory

device

Security

Capability

SELinux

seccomp

Virtualization management tools

▶ Docker

- tested by NERSC (MyDock)
- Limitations on
 - Security access
 - Heavy use of local HDD (not compatible with diskless)

▶ Shifter

- Developed by NERSC in collaboration with Cray
- Compatible with docker image
- Integrated with Slurm
- early version not mature

Virtualization management tools

▶ Singularity

- Purpose built
- Includes both network and file system access
- Easy move from Docker images to Singularity images
- The container can emulate a single program and can be executed directly
- Proper integration of MPI through adapted architecture with PMI

▶ PCCOC with RunC

- PCCOC is developed by CEA
- Provides the ability to an HPC user to launch a private virtual cluster through SLURM
- Today dedicated to Virtual Machines
- containers can be added using runc

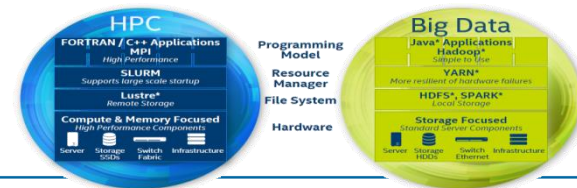
Integration test with resources manager (slurm)

	Shifter	Pcocc	Docker upon Slurm	Pcocc with RunC	Singularity
State	Production v15.12.0	Beta	POC	Concept	Production v2.0
Source	Free and open-source	Free not open-source (yet)	POC	Concept	Free and open-source
License	BSD	GPLv3	Apache v2	GPLv3	BSD
Virtualization	chroot	KVM	container	container	container
RJMS integration	SLURM	SLURM	none	any	any
Docker integration	volumes, images	none	volumes, images, network	volumes, images	volumes, images
SDN	no	yes	yes	yes	no
Checkpoint/restore	no	yes	partial	yes using CRIU	Possible
Native CPU/memory performance	yes	no	yes	yes	yes
Native network performance	yes	partial w/ SRIOV	yes (without SDN)	yes (without SDN)	yes
IO performance	? (Loop over Lustre)	VM Bad latencies Bad IOPS	Backend copy on write	Native	Native
MPI execution	Partial/complicated	?	possible	possible	Supported (optimized OpenMPI)

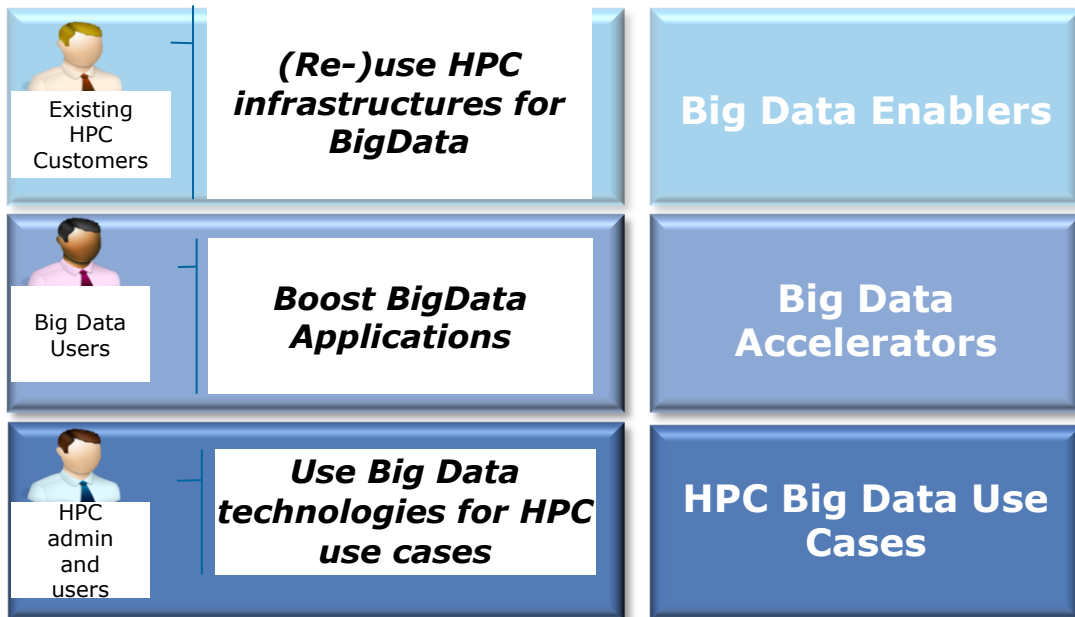
3

Synergy with big data and
cloud

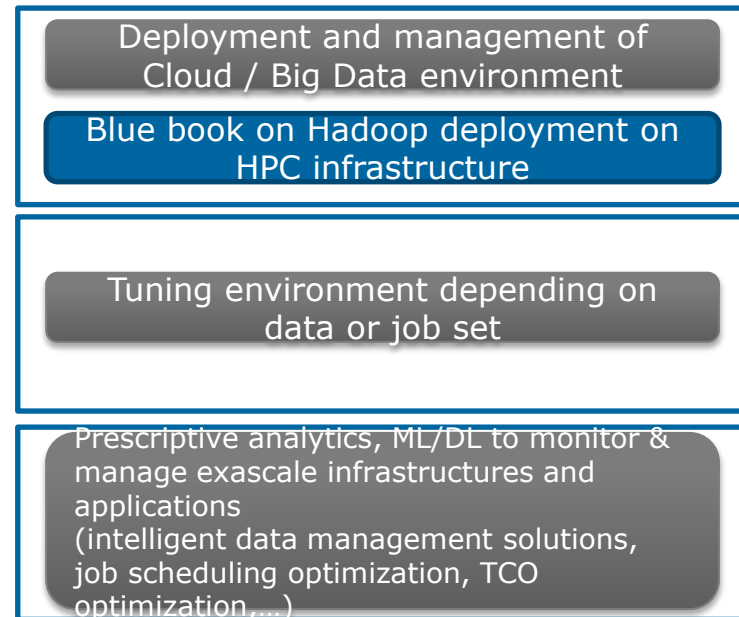
Synergies between HPC, Big Data and Cloud



Addressed customer Needs

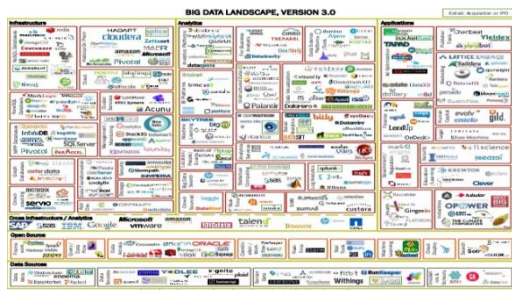


Atos Products and Research Areas

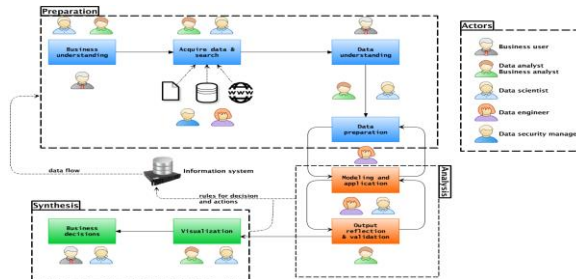


BigData context

- Dynamic ecosystem (frameworks, algos)

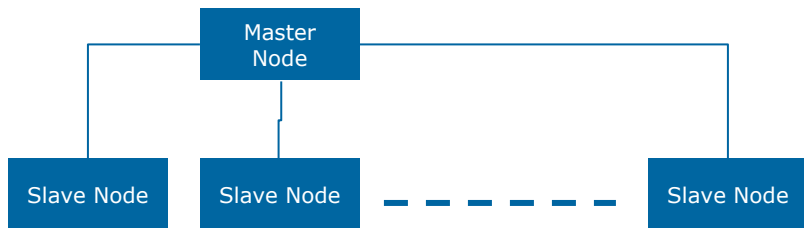


- Complex analytics workflow



- Hadoop / spark

- Distributed architecture

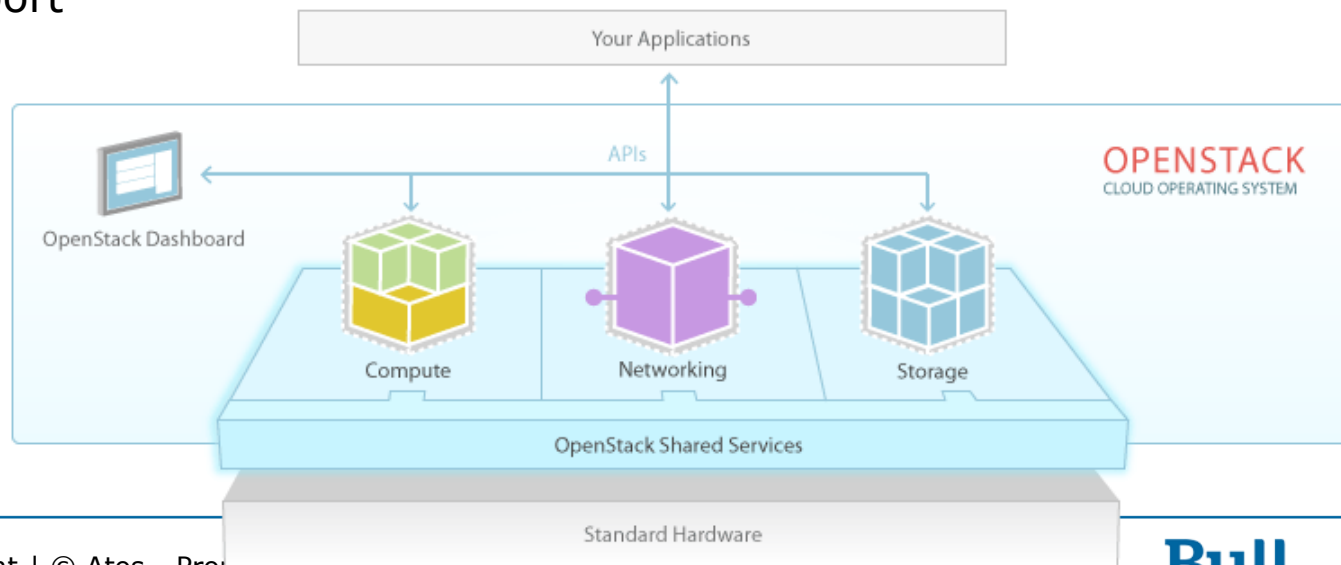


MapAdmin/Metrics	HUE				RStudio	Drill UI	
	Pig	Hive	Mahout	Impala	Spark	R	Storm
	MapReduce/YARN				TEZ	Sqoop	Flume
	HBase				NFS		
HDFS/MapR-FS							

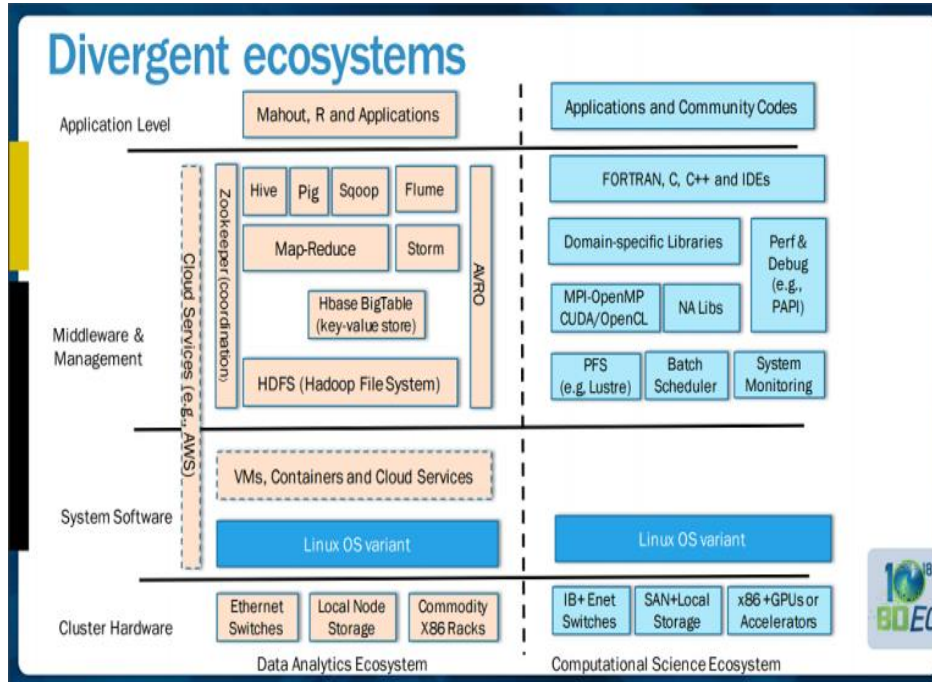


Cloud context with OpenStack

- ▶ Self-service, on-demand resource provisioning
 - virtualization: compute, network, storage
 - elasticity
 - Multitenancy support



Divergent ecosystem



HPC	BigData , Cloud
Focus on performance	Focus on productivity designed for comm infra
Adaption & optimization in application	Runtime/Platform with high level of abstraction
Data is private	Data might be shared
Batch processing	Stream & Batch processing
Non interactive	Controlled response time
Static resource management	Dynamic resource management with automatic scaling self-healing capabilities

New requirements in HPC environment

- ▶ On-demand, dynamic and fast environment provisioning (HW+SW)
 - Agile process, start small and grow by increment
- ▶ Elasticity & cloud bursting
 - Absorb peak load without overbooking, wasting resources
- ▶ User-defined software stack
 - Import new BigData tools
- ▶ Endless job support & external access through gateways
 - Stream processing, IoT use cases
- ▶ Multi-tenancy support
 - Data isolation between users
- ▶ Ease of access with abstraction tools
 - Data analysts are not familiar with HPC and needs easy to use tooling

HPC & OpenStack

- ▶ Integration to extend HPC boundaries
 - Add elasticity with cloud bursting capability
 - Add support of hybrid application spanning HPC & Cloud resources

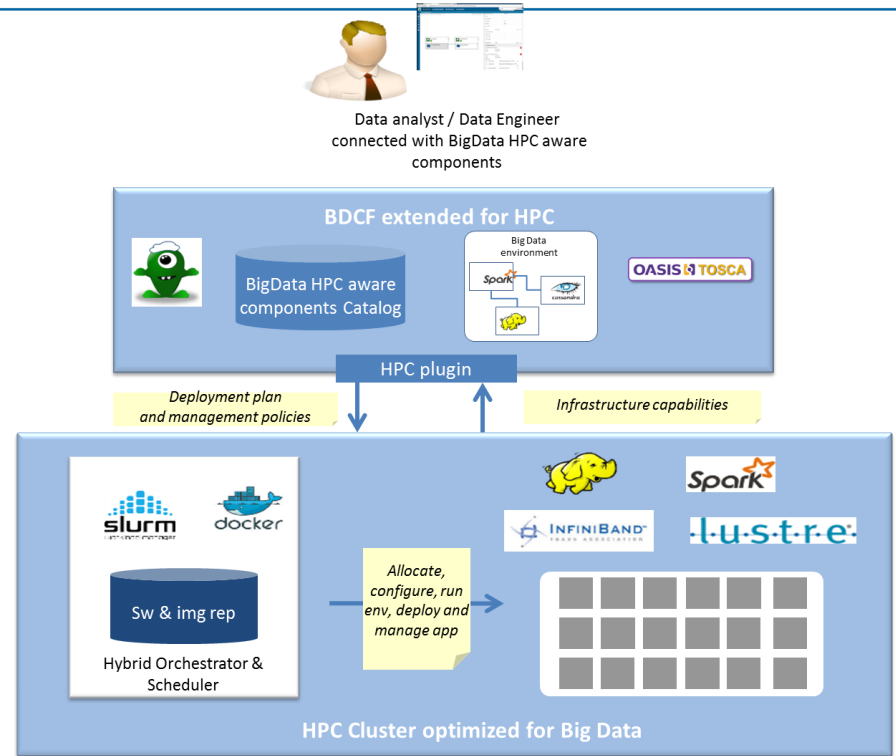
4

Towards Convergence

HPC-aware Big Data Enablement Platform

Towards a platform to abstract the overall complexity to run Big Data use cases on HPC, using the best technical features available.

- Software defined management for automatic provisioning
- User-defined software stack
- Dynamic cluster partitioning with data aware placement
- Virtualisation & Containers support



Studies for scheduling convergence

Batch submission

Some terms:

▶ Resources and Jobs Management System (**RJMS**)

– The HPC workload management tool

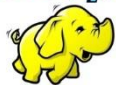
– Composed of:

- a master node that takes the decisions
- a daemon on each nodes to run and control the jobs

Examples: SLURM, OAR, PBS, ...



hadoop



▶ Big Data Analytics Framework (**BDAF**)

- Big Data Analytics management tool
- Similar to RJMS but for Big Data
- Examples: Hadoop, Spark, Flink, ...



▶ Batch submission: Submit one job to the RJMS

1. Install and configure the BDAF
2. Install and configure the distributed filesystem
3. **Stage in** the input data into the distributed filesystem
4. Submit to the BDAF the user application
5. **Stage out** the results data from the distributed filesystem

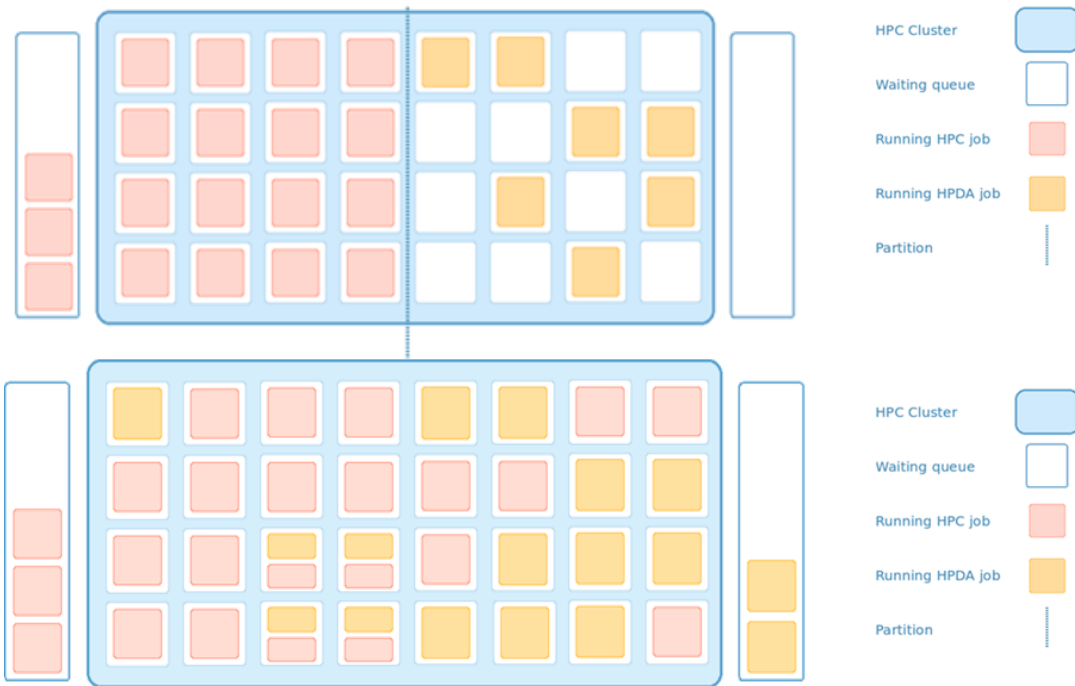
▶ **Not efficient** because of data movement

- Stage in and out can take a really long time

▶ Possible solutions:

- Use the parallel filesystem instead of the distributed filesystem
- Include staging into the scheduling policy



Advanced scheduling



▶ TODAY: Static cluster partitioning

- No node sharing
- When one queue is filling up, Available nodes on the other partition are not used

▶ FUTURE: Dynamic cluster partitioning

- Full cluster utilization
- Container Support  
- Data aware
- Mixed workload on the same resource. Make sense for example if:
 - one application IO bound
 - one application is CPU bound

Studies on data management convergence

▶ Data workflow

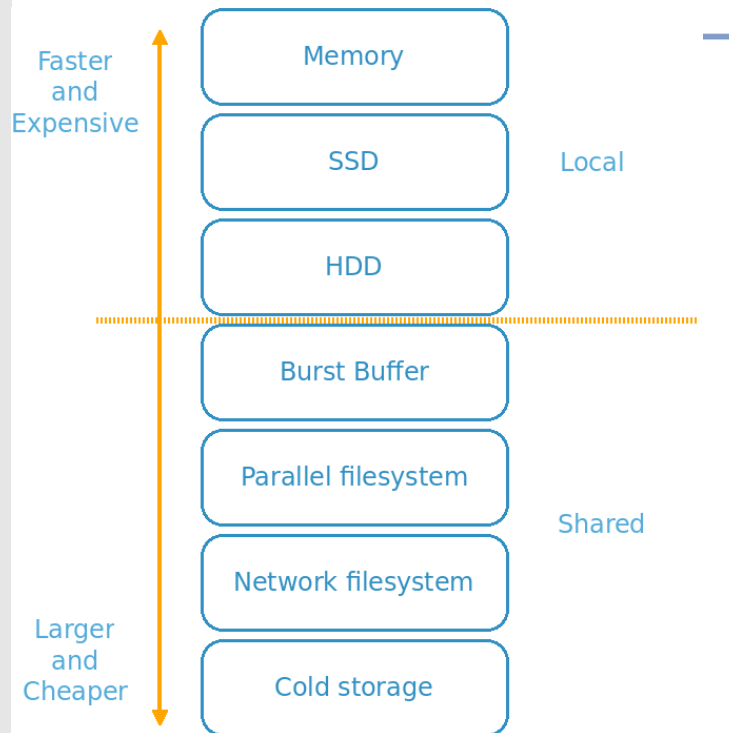
1. Ingest new data
2. Compute results
3. Query previous results
4. Use previous results to compute new results

⇒ We need **data persistence**

▶ Solutions:

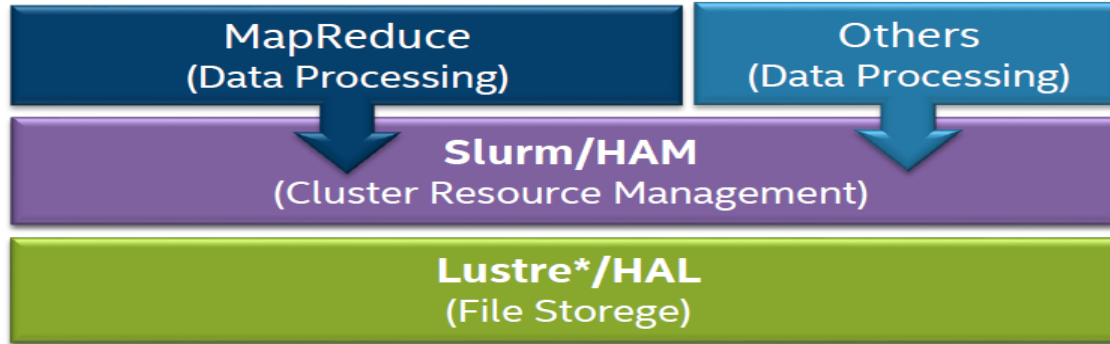
- use the parallel instead of distributed filesystem
 - Example: LUSTRE instead of HDFS
 - From local to shared: Possible congestion problem
- use local storage for better performance
 - Need data persistence for local storage
 - Use staging can be better sometimes

⇒ **RJMS need to manage data as resources**



Deploying Hadoop on Lustre.

HAM & HAL



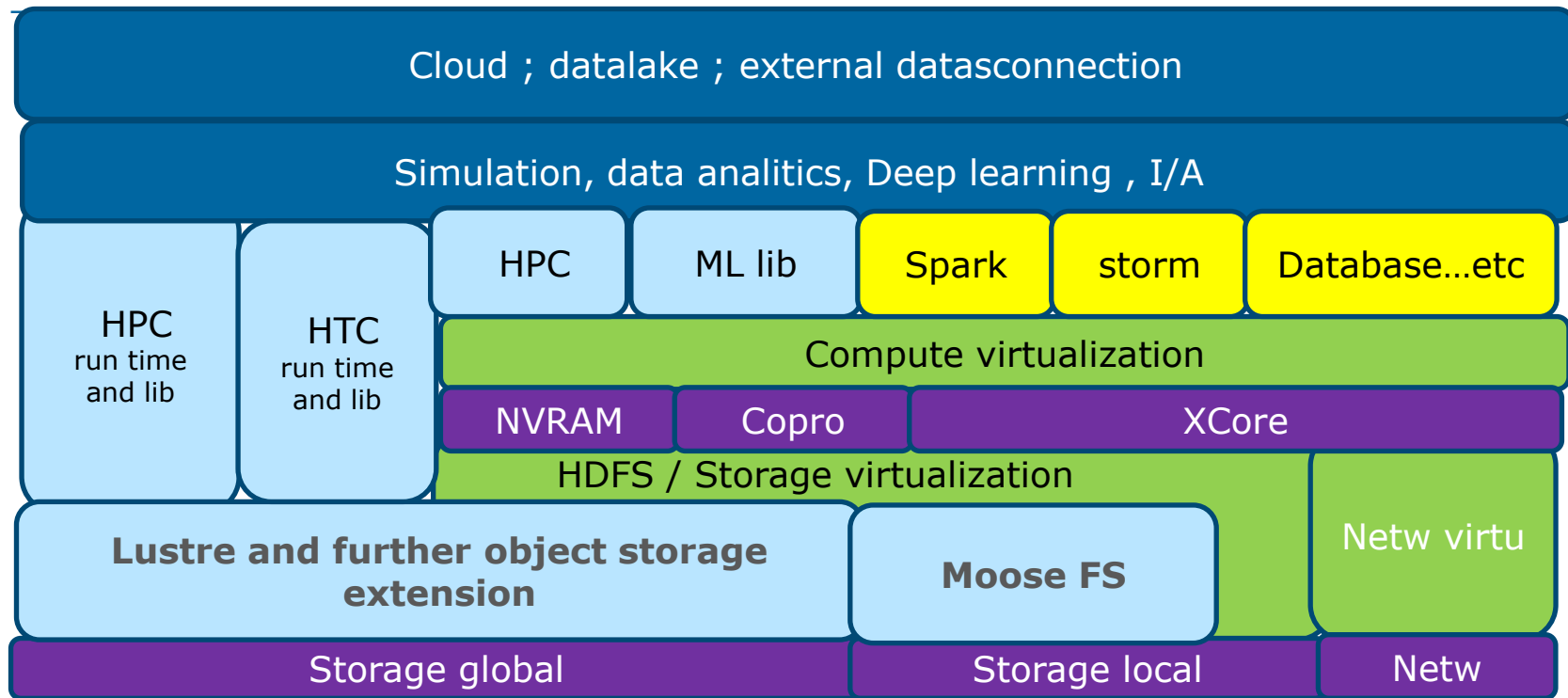
HPC Adapter for Mapreduce/Yarn

- Replace YARN Job scheduler with Slurm
- Plugin for Apache Hadoop 2.3 and CDH5
- No changes to applications needed
- Allow Hadoop environments to migrate to a more sophisticated scheduler

Hadoop* Adapter with Lustre*

- Replace HDFS with Lustre
- Plugin for Apache Hadoop 2.3 and CDH5
- No changes to Lustre needed
- Allow Hadoop environments to migrate to a general purpose file system

The target : an converged environnement



Thanks

For more information please contact:
Pascale.Rosse_laurent@atos.net

Atos, the Atos logo, Atos Codex, Atos Consulting, Atos Worldgrid, Worldline, BlueKiwi, Bull, Canopy the Open Cloud Company, Unify, Yunano, Zero Email, Zero Email Certified and The Zero Email Company are registered trademarks of the Atos group. June 2016. © 2016 Atos. Confidential information owned by Atos, to be used by the recipient only. This document, or any part of it, may not be reproduced, copied, circulated and/or distributed nor quoted without prior written approval from Atos.

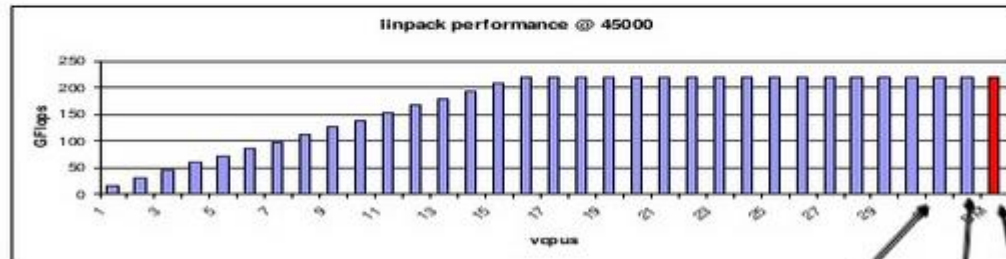
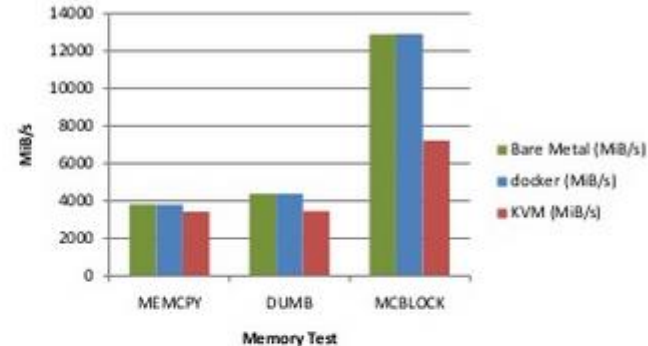
Bull
atos technologies

Performances

- ▶ Pour info,
- ▶ date de 2014

- Typical docker LXC performance near par with bare metal

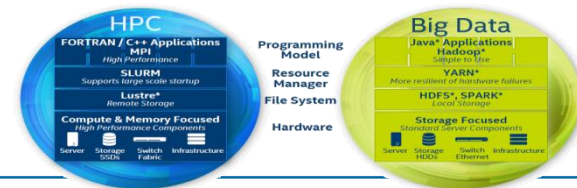
Memory Benchmark Performance



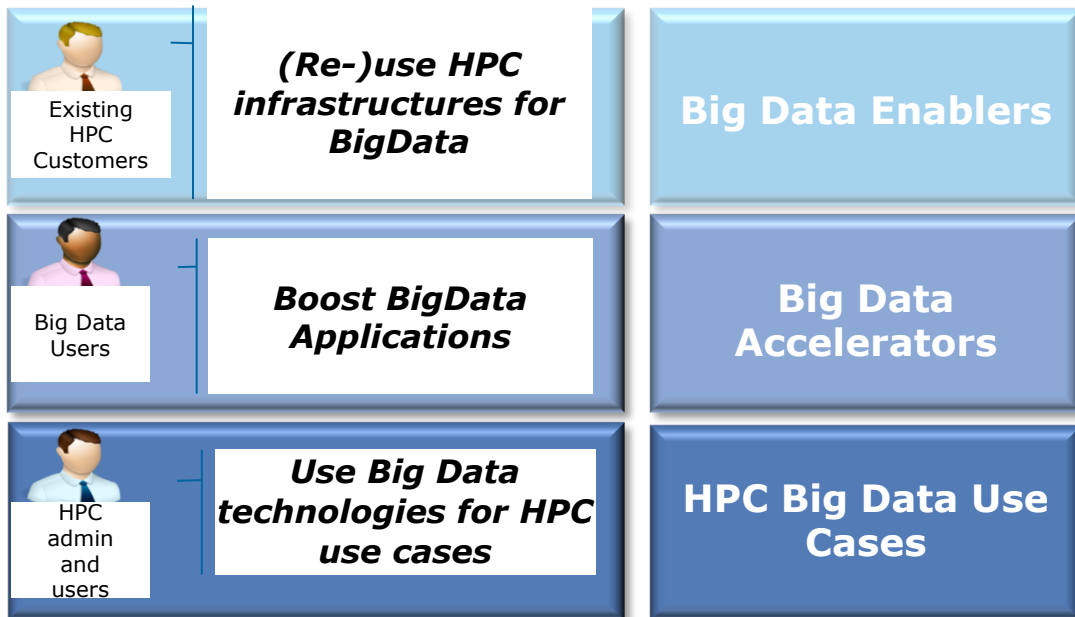
5/14/2014

220.9 @ 31 vcpu
220.5 @ 32 vcpu
220.77 Bare metal 41

Synergies between HPC, Big Data and Cloud



Addressed customer Needs



Atos Products and Research Areas

