DE LA RECHERCHE À L'INDUSTRIE

VMs and containers for a Slurm-based development cluster

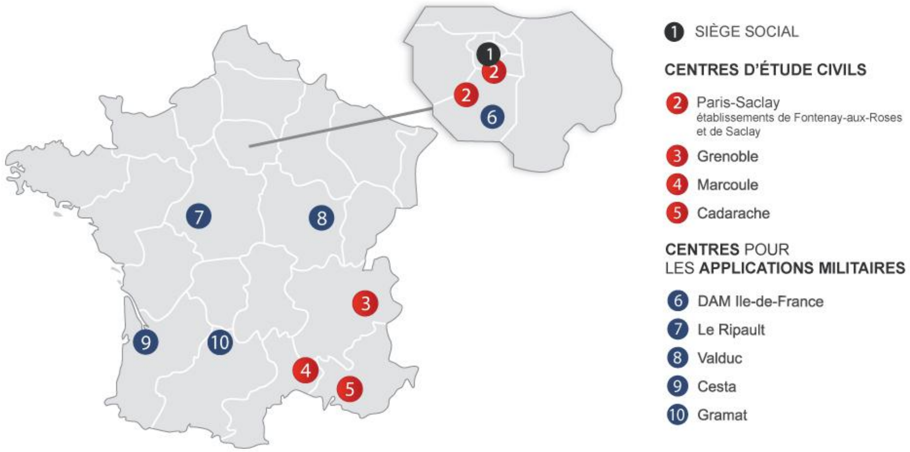
September 18, 2019

François Diakhaté

About CEA

CEA: The French Alternative Energies and Atomic Energy Commission

- **9 research centers in France**
- **Many areas of research including:**
  - Low carbon energies,
  - Information and healthcare technologies
  - Defence and security
  - Fundamental research in the physical sciences and life sciences
- **Large HPC infrastructures**
  - 2 production compute centers



CEA Compute Centers

TGCC: Très Grand Centre de Calcul du CEA

- **Hosts two main projects**
- **CCRT**
    - Shared computing center for French industrial and research partners
    - Cobalt supercomputer (installed in 2016)
        - 1422 Intel Broadwell nodes (2x14 cores, 128GB), Infiniband EDR
        - 252 Intel Skylake nodes (2x20 cores, 192GB), Infiniband EDR

cea
arianeGroup
CERFACS
EDF
ifp Énergies nouvelles
INERIS
ingeliance
IRSN
MICHELIN
ONERA
L'ORÉAL
SAFRAN
SOLEIL SYNCHROTRON
TA TechnicAtome
THALES
ThalesAlenia Space
TOTAL
Valeo

TGCC: Très Grand Centre de Calcul du CEA

- **Hosts two main projects**
- **GENCI (Grand Equipement National pour le Calcul Intensif)**
  - Part of the European PRACE project (Partnership for Advanced Computing in Europe)
  - Available for french academia, industries, and european PRACE members
  - Irène Joliot-Curie supercomputer (installed in 2017)
    - 1656 Skylake nodes (2x24 cores, 192GB), Infiniband EDR: 6,9 PF/s peak)
    - 828 KNL nodes (68 cores, 96GB), Bull eXascale Interconnect: 2,5 PF/s peak)
    - A 12 PF/s peak AMD Rome partition is currently being installed

TERA: CEA Defense computing center

- **Part of the simulation project for French nuclear deterrence**
- **Tera-1000 supercomputer (installed in 2016)**
    - 2192 Haswell nodes (32 cores, 128GB), Infiniband FDR: 2,6 PF/s peak
    - 8256 KNL nodes (68 cores, 192GB), Bull eXascale Interconnect: 23,4 PF/s peak

OCRE: A pre-production compute center for R&D

- **A testbed for new hardware**
  - From both user and sysadmin perspective
  - Many node partitions spanning a large number of technologies
- **Computing resources for HPC R&D**
  - Collaborations around new / experimental hardware
  - Development and evaluation of scientific applications and system tools
    - In a representative HPC cluster environment
- **A pre-production environment**
  - Staging for new features before deployment on production centers
    - Obtain feedback from users
  - Slurm-based cluser
  - Overall setup as close as possible to production clusters

Flexible compute clusters

HPC clusters must become more flexible

- **A trend in all our clusters**
  - More and more varied scientific communities
  - New software and frameworks not designed with HPC clusters in mind
  - Users must be able to easily deploy all kinds of software
- **Applies especially for a cluster serving developers**
  - Evaluate or develop software for all levels of the stack
    - Including system or kernel level software
  - Run continous integration tests in various environments
  - Reproduce bugs in specific setups
- **VMs and containers can help achieving that flexibility**

Private Cloud On A Compute Cluster

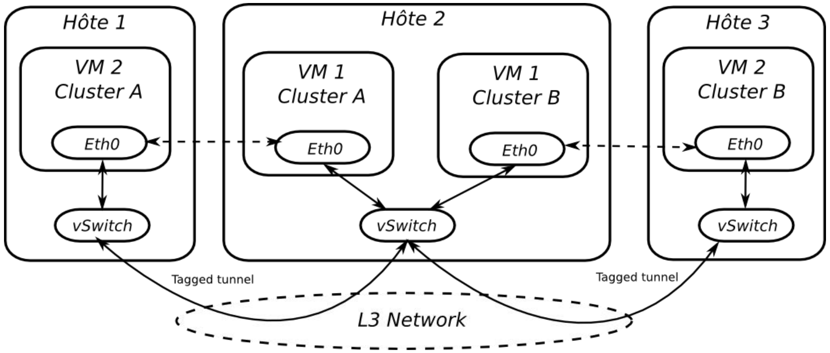
- **Bring “private cloud” features to a HPC cluster**
  - Deploy VMs and containers
  - Set up virtual networks
- **HPC oriented features**
  - Low overhead virtualization (NUMA awareness, Infiniband virtualization…)
  - Interfacing containers and HPC runtimes
- **Use the existing cluster infrastructure**
  - Slurm resource management, usual job allocation sematics
    - Number of tasks, cores and/or memory per task
    - A task can also be a VM
  - All nodes can be used for both regular jobs and hosting VMs
    - Usual scheduling, priorities, job management, accounting…
  - Images are stored in the shared parallel filesystem
    - Directories as image repositories
    - Directly accessible from compute nodes

Launching a virtual cluster

- **Example:**
  - pcocc alloc –c 8 image1:32,image2:512
  - Allocates 32 VM of one type and 512 VM of another type, each with 8 cores
- **Details:**
  - A slurm job is created to allocate the resources
    - A spank plugin setups virtualization resources during prolog
    - Virtual networks for the cluster are setup according to the VMs configurations
  - Qemu is launched as a user task
  - VM instances are created with ephemeral drives
    - Copy-on-Write using the repository image as reference
  - CPU and memory are configured according to the task allocation
    - NUMA topology is setup to match the host allocation
    - Each vCPU is bound to one of the allocated cores
    - Near native compute performance (2% overhead for HPL at 2116 cores)
  - Cancelling the job destroys VMs and cleans up nodes

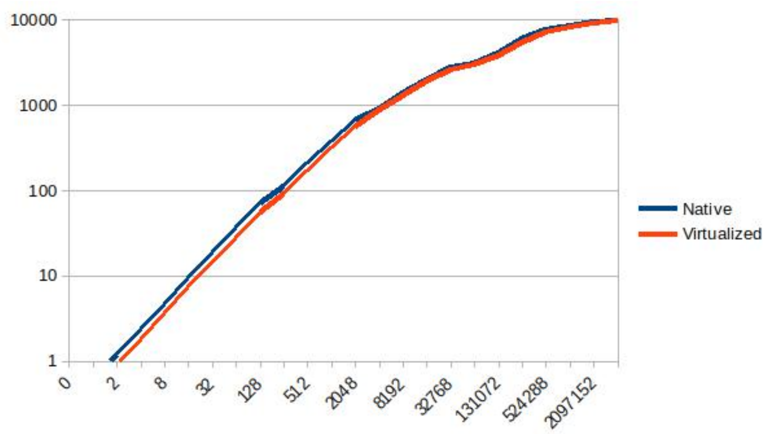
Virtual Ethernet networks

- **Create an isolated virtual Ethernet switch per virtual cluster**
  - Virtual Ethernet interfaces are attached to VMs belonging to the network
- **Implemented with Open vSwitch and the VXLAN overlay network**
  - Encapsulate Ethernet packets in UDP packets
- **Optional L3 services**
  - Automatically assigns IP to VMs
  - Provides DHCP/DNS services
  - NAT-based routing to the host cluster network
  - Reverse NAT to VM ports (SSH access)
- **Not performance optimized but good enough for most use-cases**
  - 5-10 Gbit/s between two remote VMs on our hardware



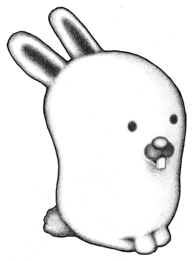
Virtual Infiniband networks

- **Create an isolated Infiniband partition per virtual cluster**
  - Makes use of Infiniband SR-IOV
    - Multiplex a physical device into multiple virtual functions (VFs)
  - VFIO is used to isolate the device from the host
    - Applies an IOMMU
  - OpenSM is reconfigured dynamically to restrict VFs to a specific partition
    - Equivalent to a VLAN
  - Infiniband VFs are attached to VMs belonging to the network
- **Near native performance**



Virtual clusters

Mounting host filesystems in VMs



- **Exported by Qemu with the 9P protocol**
  - Filesystem is accessed with privileges of the job owner
  - Fairly slow for metadata and small accesses

- **A more efficient replacement is being developped by RedHat**
  - Virtio-fs
  - Used in the Kata containers project

Docker VM allocation

- **Similar to the docker-machine tool**
  - Leverage the VM allocation support of pcocc
- **Setup a Docker environment with a single command**
  - pcocc docker alloc
    - Allocate a VM running a docker daemon
      - Hosts filesystems are available
    - Setup environment variables to redirect Docker API calls to the VM
    - Use the docker CLI as if the daemon was running locally
- **Support for Docker-based workflows**
    - Build containers with Dockerfiles
    - Use tools like docker-compose

Run unprivileged containers

- **Restriction: keep the same user id within the container**
  - Similar to other HPC-oriented container runtimes
  - pcocc run -I ctr-image [cmd]
  - Works with OCI images
    - Extracts the image once on first use
    - A cache is used for next runs
- **Namespaces**
  - Based on the bubblewrap tool
    - Used in Flatpak, a package management tool using container features
  - Uses unprivileged user namespaces if enabled
  - Can run as a setuid binary otherwise

Containers

Interface containers with host software/hardware

- **Ideally, containers should be self-contained and deployable everywhere**
- **Unfortunately, some HPC features require tight integration to the host**
  - High performance interconnect libraries
  - MPI libraries / launchers and related tuning
  - GPU runtime (CUDA)
- **Ability to define modules which inject files and/or environment variables**
  - Specified when running a container with the -M flag
  - Restricted to “sufficiently compatible” containers (glibc ...)
- **Example modules**
  - Nvidia module
    - Provides access to the host Nvidia runtime
  - OpenMPI modules
    - Injects recommended OpenMPI libraries and configurations
  - WI4MPI module
    - MPI wrapper interface
    - Translates e.g, from an IntelMPI linked application to an OpenMPI library

Planned improvements

- **Add virtual network support for containers**
  - A slurm plugin sets up a “pod” during prolog for each allocation
    - Network namespace with requested network interfaces
    - Ability to select other namespaces to unshare from the parent namespace
  - Runs user tasks using these namespaces
    - User processes setup their own mount namespaces as today

- **Allowing multiple uids in containers**
  - User namespaces with subuid/subgid
    - Maps to dedicated uid ranges on the parent namespace
  - Integration to our compute centers must be evaluated
    - Allocations of UID ranges with LDAP based accounts
    - Possible impacts on filesystems, accounting, node management etc
  - Support for full system containers, building images…

Example use cases

Large scale reproducers / debugging

- **Example: reproducers for Lustre issues**
  - Instanciate Lustre servers and client VMs
  - Reproduce issues which only happens at large scale and crash the Lustre servers
    - Issue with > 100 client nodes required to reproduce the problem
  - See Dominique Martinet presentation at Lustre Admins and Devs Workshop 2016

Example use cases

Development or evaluation of system level software

- **Easy to deploy test-beds**
  - No need for special privileges
- **Several internships every year**
  - Examples:
    - Large scale configuration deployment with Puppet
    - Development and evaluation of a diskless node management solution
    - Design of an elasticsearch based solution to process HPC logs
- **Teaching for ENSIIE engineering school**
  - HPC oriented system administration classes
  - Leverage Slurm cluster for the labs

Example use cases

Jenkins-based continous integration environment

- **Provide users with their own Jenkins instance**
  - Set up and managed by our team with the Lurch tool
  - Per-user or per-group
- **Jenkins worker accesses the HPC cluster**
  - Runs as the Jenkins instance owner and is able to submit SLURM jobs
  - Sets up custom execution environments thanks to pcocc
- **Quickly run large test suites using many parallel nodes**
  - Ex: rebuild our HPC oriented Linux distribution
- **Perform validation on representative hardware**
  - Large number of cores
  - Infiniband interconnect

Thank you !

Questions ?

Crédits photos: Jatin Sindhu, Peacock plumage