FORMATION LINUX / HPC

Antoine Huette – BECHTLE

UPEC / LISA

PLAN DE LA FORMATION

- Vendredi 23/03 : Formation Linux
  - Architecture du système
  - Gestion du stockage
  - Commandes usuelles
  - Réseau, sécurité, authentification
  - Paquets / mises à jour
  - Services courants : DHCP, DNS, Web, NFS
  - Logs / dépannage

- Lundi 26 et mardi 27/03 : Formation HPC
  - Concepts HPC
  - Installation maître
  - xCAT
  - Torque / Maui
  - Supervision
  - Librairies
  - Variables d'environnement
  - Compilateurs
  - Modules
  - Programmation parallèle
  - Python
  - GRIB

FORMATION LINUX

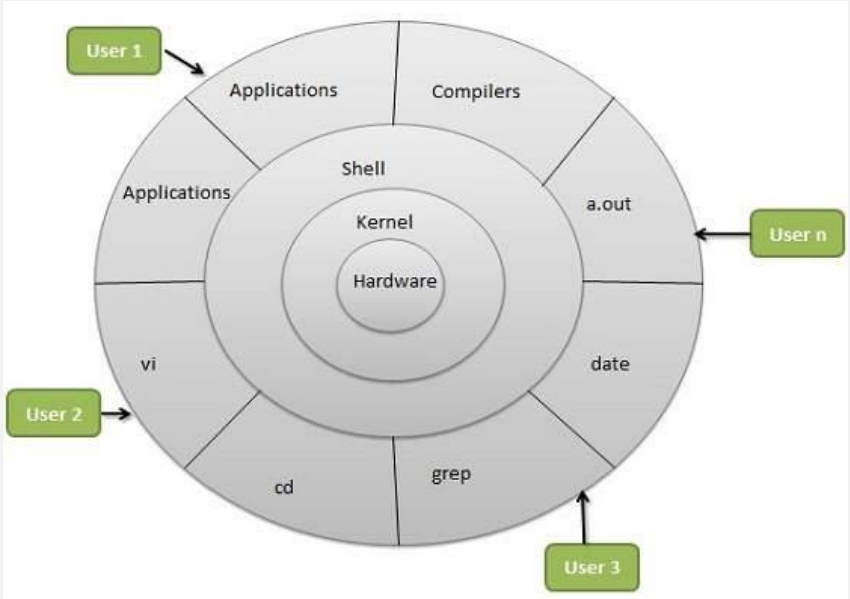
- Architecture du système
- Gestion du stockage
- Commandes usuelles
- Réseau, sécurité, authentification
- Paquets / mises à jour
- Services courants : DHCP, DNS, Web, NFS
- Logs / dépannage

ARCHITECTURE DU SYSTÈME
DISTRIBUTIONS LINUX

- Linux = plusieurs distributions
- Distribution = noyau + ensemble de pilotes périphériques, bibliothèques, utilitaires d'installation, et de logiciels
- Quelques exemples de distributions :
  - Slackware : la 1ère à avoir popularisé Linux
  - RedHat : la 1ère à avoir développé le paquetage, distribution commerciale
  - CentOS : paquets compilés depuis RedHat, distribution non commerciale, destinée aux serveurs
  - Debian : développement calqué sur celui du noyau Linux
  - Ubuntu : basée sur Debian, adaptée au grand public par Canonical
  - Suse : paquets RedHat, outils d'installation spécifiques

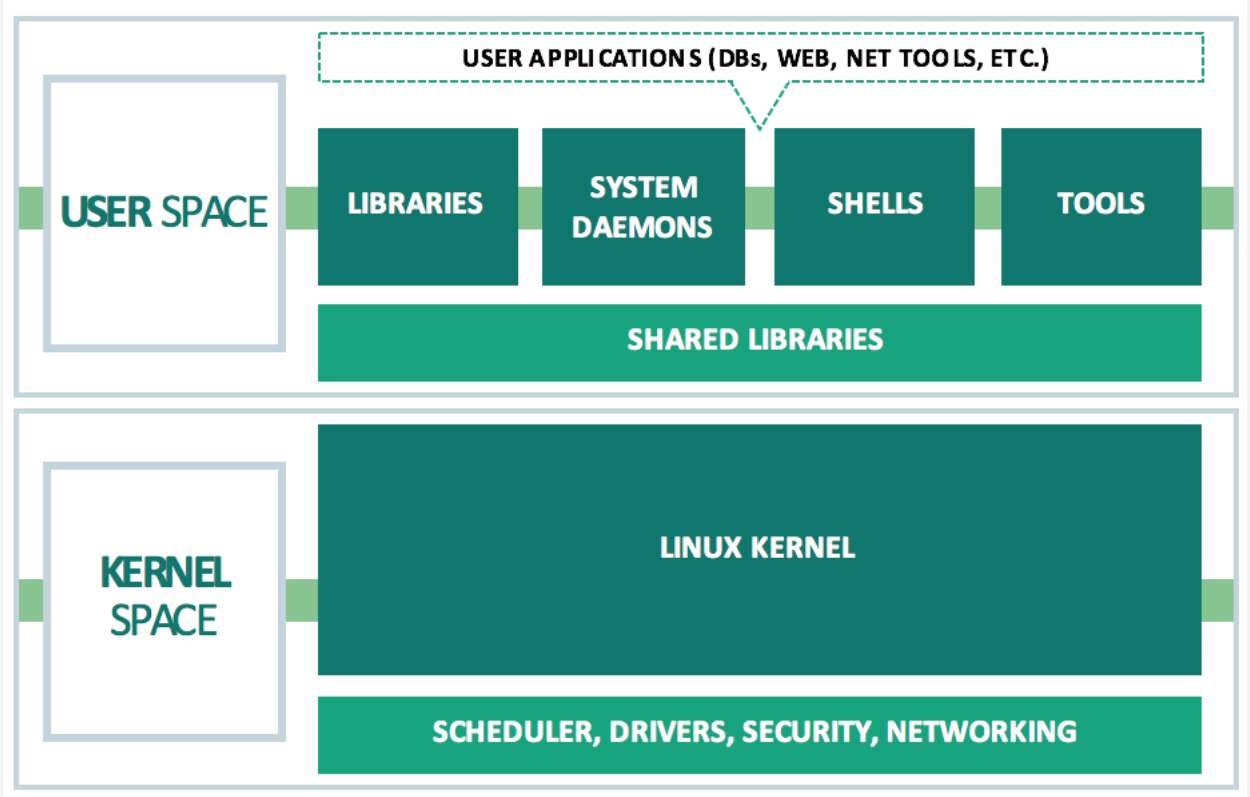
ARCHITECTURE DU SYSTÈME
COUCHES DU SYSTÈME

- Linux est un système composé de couches :
  - Couche matérielle : regroupe composants et périphériques physiques de l'ordinateur
  - Noyau (kernel) : cœur du système, contrôle le matériel, gère les ressources. Il est constitué de modules et est chargé en 1er quand la machine démarre. Les parties critiques du noyau sont chargées dans des zones sécurisées de la mémoire.
  - Shell (console) : interface du kernel qui permet d'exécuter des commandes et de les transformer en fonctions
  - Utilitaires : programmes qui fournissent les fonctionnalités nécessaires au système
  - Librairies du système : Fonctions ou programmes spéciaux utilisés par les applications pour accéder aux fonctionnalités du kernel. Ne nécessitent pas de droits d'accès au code du kernel.



ARCHITECTURE DU SYSTÈME
MODES D'EXÉCUTION

- 2 modes d'exécution :
  - Mode kernel : Le code du kernel s'exécute ici. Ce mode sécurisé a accès à toutes les ressources de la machine. Le kernel est un simple processus qui exécute chaque autre processus et leur permet d'accéder aux services et composants du système.
  - Mode user : Tous les autres programmes qui n'ont pas besoin de s'exécuter dans le mode kernel fonctionnent dans le mode user. Il n'a aucun accès au hardware et au code du kernel. Ces programmes utilisent les librairies du système pour accéder aux fonctions du kernel.

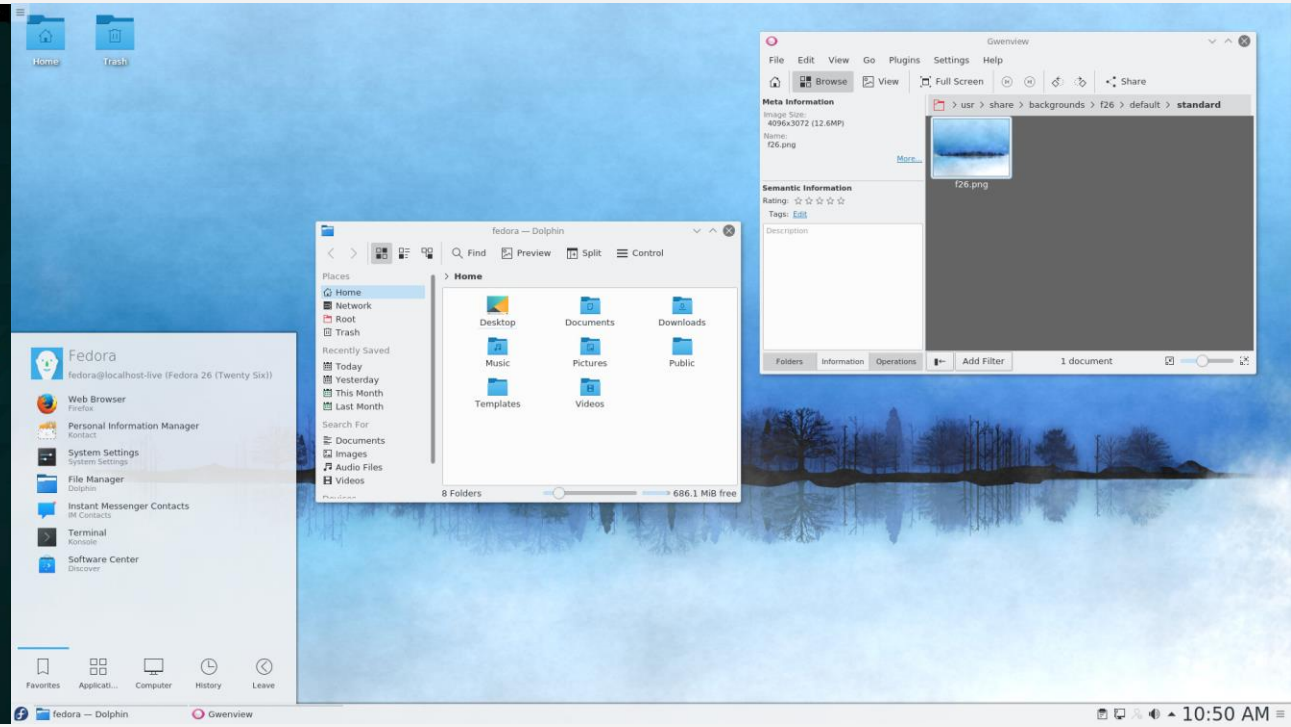


ARCHITECTURE DU SYSTÈME
BRIQUES ET ENVIRONNEMENT GRAPHIQUE

- Linux est un système composé de « briques », à l'inverse de Windows qui est très monolithique.
- L'environnement graphique est une de ces briques, le système Linux peut fonctionner sans, et on peut aisément le modifier.
- Interface graphique : basée sur le serveur X Window. Celui-ci fonctionne sous un mode client/serveur, où le client est l'application X et le serveur est le terminal X. Ce serveur X gère aussi l'écran, la souris et le clavier.
- Standard graphique : X11
- Clients graphiques = bureaux ou environnements de travail
- Exemples de bureaux : Gnome, KDE

ARCHITECTURE DU SYSTÈME BUREAUX

- Exemples de bureaux : Gnome, KDE



ARCHITECTURE DU SYSTÈME SPÉCIFICITÉS

- Spécificités de Linux :
  - Portabilité = peut fonctionner sur un très large panel de matériels
  - Open source = code disponible, développement des différentes briques organisé par des « communautés »
  - Multi-user = plusieurs utilisateurs peuvent être connectés et accéder aux ressources en simultané sur la même machine
  - Système de fichiers hiérarchique = tous les fichiers sont organisés depuis la racine « / » quelque soit leur emplacement
  - Shell = programme qui interprète et exécute des commandes

ARCHITECTURE DU SYSTÈME NIVEAUX D'EXÉCUTION

- Runlevel : détermine les services qui sont démarrés ou stoppés lors du démarrage du système. Il est défini dans le fichier /etc/inittab
- Niveaux de runlevel :
  - 0 = arrêt
  - 1 = single user text mode
  - 2 = non utilisé (peut-être customisé)
  - 3 = Multi-user text mode
  - 4 = non utilisé (peut être customisé)
  - 5 = multi-user graphical mode
  - 6 = redémarrage

Les niveaux les plus utilisés sont 3 (mode texte) et 5 (mode graphique).

Le niveau 1 est utile pour le dépannage du système.

ARCHITECTURE DU SYSTÈME SERVICES

- Linux regroupe de nombreux services
- Ces services représentent chacun une fonction ou application du système
- Les services sont localisés dans
  - le répertoire /etc/init.d sous CentOS 6
  - Les répertoires /etc/init.d/ et /etc/systemd/system/ sous CentOS 7
- Les commandes possibles sur les services sont : start, stop, restart, status
- Pour gérer un service :
  - Service <service> <commande> sous CentOS 6 ou systemctl <commande> <service> sous CentOS 7
- Pour configurer le démarrage automatique d'un service au démarrage du système :
  - Chkconfig <service> <on/off> sous CentOS 6 ou systemctl <enable/disable> <service >

ARCHITECTURE DU SYSTÈME SUPERUTILISATEUR

- Root = superutilisateur
- Il s'agit d'un utilisateur spécial disposant des permissions maximales
- Il peut par exemple modifier la configuration du système
- On peut donner des droits root à un utilisateur en le spécifiant dans le fichier /etc/sudoers
- Un utilisateur « devient » root en effectuant la commande « su », ou peut disposer de ses droits pour la commande actuelle en la précédant par « sudo »

GESTION DU STOCKAGE
PÉRIPHÉRIQUES ET PARTITIONS

- Périphériques de stockage :
  - Les périphériques de stockage sont représentés par des fichiers /dev/sdX, où X est une lettre
  - Un périphérique de stockage peut être physique (disque entier) ou virtuel (volume RAID)
  - Utilitaires pour gérer les disques : fdisk ou parted

- Partitions :
  - Chaque disque, pour être utilisable, a besoin d'une table de partitions. Celle-ci décrit l'agencement des partitions d'un disque.
  - Une partition est une unité de stockage numérotée qui divise un disque. Par exemple pour le disque sda, on peut avoir sda1, sda2 et sda3 comme partitions.
  - Il existe 3 types de partitions :
    - Primaire : 4 par disque maximum
    - Si on veut plus que 4 partitions, il faut créer une partition étendue (1 seule possible par disque). Cette partition peut contenir « en dessous » un nombre quasiment illimité de partitions
    - Ces partitions sont dites logiques

GESTION DU STOCKAGE
PARTITIONS

- Chaque partition, pour être utilisable, doit contenir un système de fichiers. Mais un système de fichiers peut être créé sur le disque, sans partition.
- Une partition est attachée au système par un point de montage
- Exemple de partitions :
  - /dev/sda1 monté sur la racine « / »
  - /dev/sda2 monté sur /boot
  - /dev/sda3 monté sur /home
  - /dev/sda4 monté en tant que swap
- Sous Linux, les systèmes de fichiers sont souvent journalisés (ext3, ext4). Les modifications apportées aux fichiers sont écrites dans un journal afin de limiter les pertes de données lors d'un crash.
- Les systèmes de fichiers sont définis et montés dans le fichier /etc/fstab

GESTION DU STOCKAGE
RÉPERTOIRES COURANTS

- Les répertoires courants d'un système Linux sont :
  - /bin = programmes utilisateurs
  - /boot = fichiers de démarrage
  - /dev = périphériques
  - /etc = fichiers de configuration
  - /home = répertoires personnels
  - /lib et /lib64 = librairies utilisées par les programmes
  - /media et /mnt = périphériques amovibles
  - /proc = informations système
  - /root = dossier personnel de l'utilisateur root
  - /sbin = programmes importants du système
  - /tmp = dossier temporaire
  - /usr = dossier d'installation de certains programmes
  - /var = données variables, logs

GESTION DU STOCKAGE
SYSTÈME DE FICHIERS

- Sous Linux, tout est fichier
- 2 types de fichiers : normaux, et spéciaux (lecteur CD, disque dur…)

- La racine « / » est le point d'origine du système de fichiers Linux. Tout est organisé dans la même arborescence, contrairement à Windows où chaque disque est représenté par une lettre et une arborescence différente.

COMMANDES USUELLES

- Commandes utiles relatives aux fichiers :
  - Pwd = afficher le chemin du répertoire courant
  - Ls = lister les fichiers du répertoire courant
  - Cd = changer de répertoire
  - Cp/mv = copier/déplacer un fichier
  - Mkdir/rmdir = créer/supprimer un répertoire
  - Rm = supprimer un fichier
  - Grep = rechercher un mot dans un fichier
  - Df = afficher les partitions
  - Mount = afficher/monter des partitions
  - Du = afficher la taille des fichiers
  - Cat = afficher un fichier dans le terminal
  - Less/more = afficher un fichier page par page
  - Touch = créer un fichier
  - Ln = créer un lien entre 2 fichiers
  - Tar = gérer les archives
  - Rsync = synchroniser 2 dossiers entre eux

COMMANDES USUELLES

- Gestion des utilisateurs
  - Sudo = passer en mode super-user
  - Adduser/deluser = créer/supprimer utilisateur
  - Passwd = créer ou modifier le mot de passe d'un utilisateur
  - Addgroup/delgroup = créer/supprimer groupe
  - Who = afficher la liste des utilisateurs connectés

- Gestion des processus et ressources
  - Ps aux = afficher processus en cours d'exécution
  - Top = afficher l'utilisation du système en temps réel
  - Free = affichage de l'utilisation RAM
  - Kill = tuer un processus

COMMANDES USUELLES

- Gestion du matériel
  - Lspci = afficher périphériques PCI
  - Dmidecode = afficher informations du BIOS
  - Dmesg = afficher messages du noyau

- Date/heure
  - Date = afficher date + heure
  - Uptime = afficher la durée d'exécution du système

COMMANDES USUELLES

- Commandes relatives au réseau
  - Ip a = afficher les interfaces réseau et leur configuration
  - Route = afficher les routes réseaux
  - Scp = copier des fichiers à travers le réseau vers une autre machine

- Commandes diverses
  - Shutdown = éteindre/redémarrer l'ordinateur
  - Hostname = afficher/configurer le nom d'hôte
  - Man = afficher le manuel d'une commande

- Redirection sortie standard vers fichier : « <commande> > <nom_fichier> »
- Envoyer résultat commande vers une 2e commande : « <commande1> | <commande2> »

RÉSEAU, SÉCURITÉ, AUTHENTIFICATION
CONFIGURATION RÉSEAU

- Fichiers de configuration réseau :
  - /etc/sysconfig/network-scripts/ifcfg-eth = configuration réseau pour chaque interface
  - /etc/sysconfig/network = nom d'hôte
  - /etc/hosts = liste de machines connues (utiles si pas de DNS)
  - /etc/nsswitch = méthodes de recherche des noms
  - /etc/resolv.conf = serveurs DNS

RÉSEAU, SÉCURITÉ, AUTHENTIFICATION PERMISSIONS

- Permissions
  - Chaque fichier ou répertoire appartient à un utilisateur et un groupe
  - 3 niveaux d'identification : propriétaire, groupe, autres
  - 3 types de permission : read, write, execute (x)
- Commandes pour modifier les permissions :
  - Chmod = modifier les permissions rwx d'un fichier pour le propriétaire, le groupe et les autres
  - Chown = modifier le propriétaire et le group d'un fichier
- Permissions spéciales :
  - Sticky bit = permet de garder en mémoire un exécutable pour un lancement plus rapide
  - Setuid = indique qu'un exécutable doit être lancé avec les droits du propriétaire (au lieu de celui qui a lancé l'exécutable)
  - Setgid = indique qu'un exécutable doit être lancé avec les droits du groupe qui l'appartient

RÉSEAU, SÉCURITÉ, AUTHENTIFICATION AUTHENTIFICATION

- Il existe de nombreuses manières de gérer l'authentification sur un système Linux
- Authentification locale :
  - Fichier /etc/passwd = contient les utilisateurs
  - Fichier /etc/group = contient les groupes d'utilisateurs
  - Fichier /etc/shadow = contient les mots de passe cryptés
- PAM (Pluggable Authentication Modules)
  - Suite de librairies qui décrit comment les applications authentifient les utilisateurs
  - Configuration dans /etc/pam.d
  - Généralement la configuration par défaut d'une application bénéficiant de PAM suffit
- LDAP (Lightweight Directory Access Protocol)
  - Service d'annuaire stocké dans une base de données
  - Configuration dans /etc/openldap

RÉSEAU, SÉCURITÉ, AUTHENTIFICATION
ACLS ET SELINUX

- ACL = Access Control List
- Permet de gérer les permissions de manière plus fine
- Commandes :
  - Getfacl = permet d'afficher les ACLs d'un fichier
  - Setfacl = permet de configurer les ACLs d'un fichier

- SELinux : définition des droits d'accès et de transition des utilisateurs, applications, processus et fichiers
- Assez contraignant, il est généralement désactivé (fichier /etc/syconfig/selinux)

RÉSEAU, SÉCURITÉ, AUTHENTIFICATION AUTHENTIFICATION

- NIS (Network Information Service)
  - Service qui permet de distribuer des noms d'utilisateur, des mots de passe et autres données sensibles aux ordinateurs qui sont dans son même domaine
  - Fonctionne sur un modèle client-serveur
  - Les clients et le serveur ont besoin du paquet rpcbind. Le paquet pour le serveur NIS est ypserv, celui pour le client est ypbind.
  - Le serveur se configure dans les fichiers /etc/sysconfig/network, /var/yp/securenets et /etc/sysconfig/yppasswdd
  - Le client se configure dans /etc/sysconfig/network, /etc/yp.conf et /etc/nsswitch.conf
  - Méthode d'authentification pas très recommandée car manque de sécurité

PAQUETS / MISES À JOUR

- 2 manières de gérer les paquets sous CentOS : RPM et yum
- RPM : RedHat Package Manager
- Commandes :
  - rpm –i = installation de paquet
  - Rpm –U = mise à jour
  - Rpm –e = désinstallation de paquet
  - Rpm –qa = liste des paquets installés
- Yum : Yellowdog Updater Modified
- Couche supérieure à RPM pour installer/mettre à jour plus facilement
- Commandes :
  - Yum list installed = liste des paquets installés
  - Yum search = rechercher un paquet
  - Yum install/remove = installation/désinstallation paquet
  - Yum update = mise à jour

PAQUETS / MISES À JOUR

- Parfois on veut installer un logiciel uniquement disponible sous forme d'archive source tar.gz
- Instructions à suivre :
  - Extraction de l'archive -> tar –xvzf <archive>
  - Configuration -> ./configure (à l'intérieur du dossier extrait)
  - Compilation -> make (certaines informations et/ou paramètres peuvent être demandés)
  - Installation -> make install

SERVICES COURANTS

- NFS : Network File System
- Permet d'accéder à des fichiers stockés sur une machine distante à travers le réseau
- Nécessite le paquet nfs-utils sur le serveur et les clients
- Configuration serveur dans fichiers /etc/exports et /var/nfs (pour les permissions), puis commande « exportfs –a » pour actualiser les exports NFS
- Configuration client dans /etc/fstab, puis commande « mount –a » pour effectuer les montages

SERVICES COURANTS

- DHCP
- Attribution d'adresse IP à des machines clientes
- Paquet nécessaire pour serveur : dhcp
- Configuration serveur dans fichier /etc/dhcpd.conf :
  - Ajouter subnet avec plage d'adresses IP souhaitée
  - Ou définir les hôtes de manière statique en renseignant adresse MAC et IP
- Configuration client dans fichier /etc/sysconfig/network-scripts/ifcfg-ethX : indiquer BOOTPROTO=dhcp

SERVICES COURANTS

- DNS
- Permet à une machine de communiquer avec des noms d'hôtes plutôt que des adresses IP
- Nécessite les paquets suivants sur le serveur : bind, bind-utils
- Configuration du serveur :
  - Fichier /etc/named.conf -> pour gérer paramètres généraux et déclarer des zones de domaines
  - Dossier /var/named/ -> fichiers des zones DNS
- Nom du service DNS : named

SERVICES COURANTS

- APACHE 2
- Serveur Web
- Configuration du serveur dans /etc/httpd/
- Nécessaire pour d'autres services tels que Nagios, Ganglia, TFTP

LOGS/DÉPANNAGE

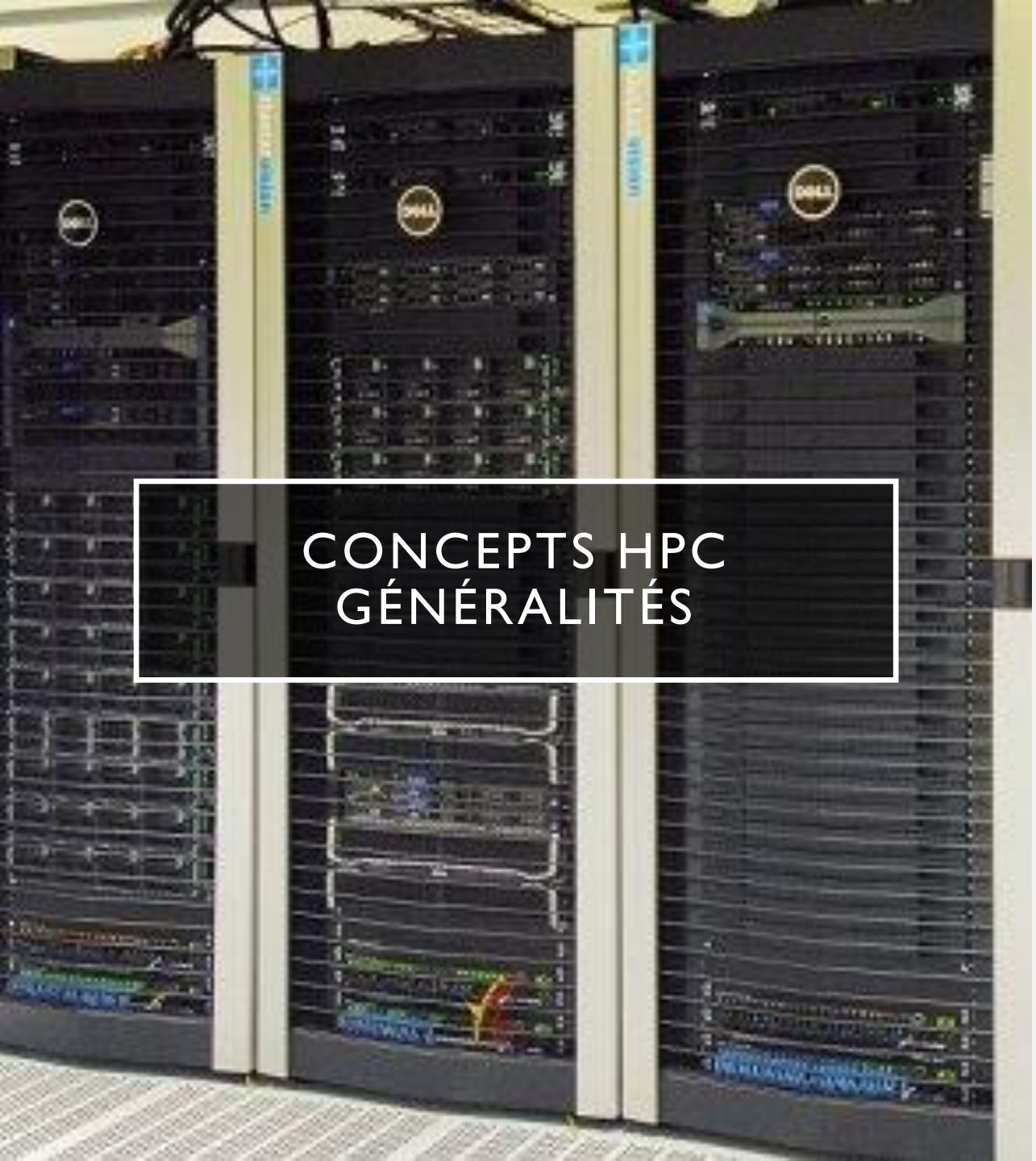
- Logs du système dans répertoire /var/log/
- Par défaut les logs sont archivés chaque semaine et 4 semaines sont gardées en mémoire
- Logs généraux du système : /var/log/messages
- Certaines applications ont un répertoire dédié dans ce chemin
- Utilitaire sous ligne de commande : journalctl (CentOS 7)

LOGS/DÉPANNAGE

- Commandes utiles pour le dépannage :
  - ps aux = afficher les processus système
  - top = afficher les processus de manière interactive
  - free = afficher utilisation mémoire vive
  - df –h = afficher utilisation espace disque
  - du = afficher espace utilisé par fichiers d'un répertoire
  - lspci = afficher périphériques PCI

FORMATION HPC

- Lundi 26 :
  - Concepts HPC
  - Installation maître
  - xCAT
  - Torque / Maui
  - Supervision
- Mardi 27 :
  - Librairies
  - Variables d'environnement
  - Compilateurs
  - Modules
  - Programmation parallèle
  - Python
  - GRIB



CONCEPTS HPC GÉNÉRALITÉS

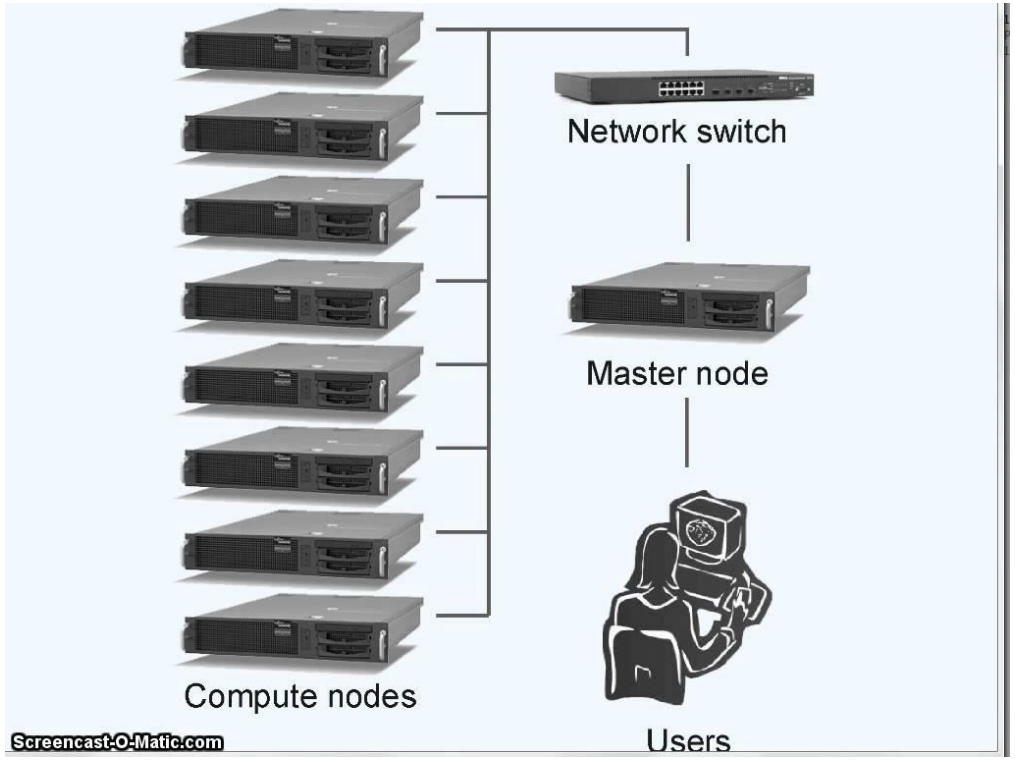
- **High Performance Computing = calcul haute performance** sur un « super ordinateur » ou « cluster », à la pointe de la technologie en termes de puissance de calcul (CPU, RAM, GPU)
- **Cluster = grappe d'ordinateurs** connectés entre eux qui travaillent ensemble de manière à ne former qu'une seule machine sur certains aspects
- Les composants d'un cluster sont généralement connectés entre eux par un **réseau rapide**. Chaque nœud dispose d'un système d'exploitation.
- Cluster plus efficace en termes de rapport performance/prix qu'un seul serveur de performance équivalente
- Clusters jouent un rôle très important dans les sciences et sont utilisés pour de nombreuses applications intensives telles que la mécanique quantique, les prévisions météo, les recherches sur le climat, l'exploitation de gaz et de pétrole, la modélisation moléculaire, ou les simulations physiques
- Matériels adaptés au HPC : serveurs bi/quadri lames (2/4 nœuds dans un même châssis) -> rapport volume/puissance (densité)

CONCEPTS HPC
GÉNÉRALITÉS

- Pour être efficace et pertinent, le code exécuté sur le cluster doit être **prévu et optimisé pour un fonctionnement massivement parallèle**.
- Quasiment tous les clusters sont installés sous un **OS Linux** (CentOS/RedHat, Debian). Les systèmes d'exploitation Linux disposent d'outils de compilation gratuits et libres pour chaque code de programmation (C/C++, Fortran, Python, Perl…)
- Dans un cluster, **l'homogénéité est vivement conseillée** : au niveau hardware (même marque/modèle pour les nœuds) et au niveau software (même version d'OS sur toutes les machines). Un cluster hétérogène est possible mais plus compliqué à gérer.
- Un cluster comporte un **gestionnaire de jobs**, responsable de la gestion des ressources (nœuds, CPUs, mémoire, GPUs…). Il permet aux utilisateurs d'exécuter des travaux (jobs) sur les nœuds de calcul de manière optimisée et planifiée. Exemples de gestionnaires de job : Torque/Maui, Slurm, PBSPro, SGE, LSF/OpenLava

CONCEPTS HPC
RÔLES DU MAÎTRE

- Dans un cluster, il existe un nœud spécial appelé « **maître** » ou frontal depuis lequel les utilisateurs et administrateurs se connectent. C'est depuis cette machine que les tâches d'administration s'effectuent et que les calculs sont lancés. Le nœud maître est également responsable de l'installation, la surveillance et la gestion des nœuds de calcul.
- **Le maître est l'élément central du cluster**. Il dispose de répertoires qui sont exportés vers les nœuds de calcul, ces répertoires peuvent donc accueillir certains éléments centralisés qui n'ont pas besoin d'être installés/configurés sur chaque nœud.
- Sur certains clusters, il peut y avoir plusieurs maîtres (redondance, répartition des charges). Ou un serveur frontal en plus du maître.



CONCEPTS HPC
RÔLES DU MAÎTRE

- Les rôles du maître sont :
  - Installation et configuration des librairies/compilateurs
  - Installation et configuration du gestionnaire de jobs
  - Installation et configuration du système de déploiement des nœuds
  - Installation et configuration de certains logiciels additionnels
  - Configuration de la surveillance et de la supervision (avec Nagios et Ganglia par exemple)
  - Serveur NTP -> le maître se synchronise sur un serveur global, puis les nœuds se synchronisent sur le master
  - Serveur Apache -> le maître en a besoin pour d'autres services (Nagios, Ganglia, TFTP)
  - Serveur DHCP -> utilisé lors de l'installation d'un nœud pour lui attribuer une adresse IP et lui envoyer les fichiers d'installation
  - Serveur NFS -> utilisé pour exporter les répertoires centraux du cluster
  - Serveur DNS -> pour la résolution de noms, optionnel car on peut utiliser juste le fichier /etc/hosts
- Attention aux mises à jour ! Vivement déconseillées pour ne pas « casser » le cluster !

CONCEPTS HPC
STOCKAGE

- Sur le maître : espace disque local pour le système et les répertoires partagés du cluster. Aussi éventuellement les répertoires utilisateur.
- Sur les nœuds : espace disque local pour le système et le volume « scratch » (/tmp) qui peut contenir des données de calcul (stockage non permanent)
- Certains clusters disposent de stockage partagé sous la forme de NAS, SAN ou système de stockage par attachement direct. Ces espaces de stockage permettent aux utilisateurs de stocker les données de compilation, avant, durant et après le calcul en lui-même. Ceux-ci sont conçus pour être à la fois rapides et sécurisés (redondance par raid, contrôleurs multiples,…).

- Les répertoires partagés du maître sont généralement les suivants :
  - /home pour les données utilisateur
  - /usr/local pour les logiciels et librairies communes
  - /opt pour certains outils et compilateurs
  - /tftpboot dans certains cas pour l'installation des nœuds

CONCEPTS HPC
RÉSEAUX

- Les différents réseaux que l'on trouve typiquement dans un cluster sont :
  - Le **réseau pour l'accès extérieur**, défini sur une seule interface du serveur maître
  - Le **réseau pour l'administration du cluster et l'installation des nœuds**, dont les besoins en bande passante sont relativement faibles
  - Le **réseau dédié au calcul parallèle**, exclusivement utilisé par les applications réalisant des opérations de communication entre plusieurs nœuds. Les performances de ce réseau influent sur les performances des calculs, c'est pourquoi on privilégie des réseaux à faible latence et haut débit de type Infiniband ou 10 Gb. Le serveur maître est souvent exclu de ce réseau.
  - Le **réseau pour l'accès IPMI** des serveurs (peut être le même que le réseau administration)
  - Le **réseau de partage de fichiers**, pour l'accès aux espaces de stockage partagés. Parfois en commun avec le réseau administration.

CONCEPTS HPC
GESTION DES PAQUETS

- Généralement dans un cluster, les nœuds de calcul n'ont pas d'accès à Internet
- Seul le maître est capable de récupérer des paquets ou mises à jour sur le Net
- Dans ce cas il est nécessaire de configurer un repository (dépôt) sur le maître afin que les nœuds puissent « taper » dedans au lieu d'aller chercher les paquets sur Internet
- Paquet nécessaire : createrepo

- Création d'un dépôt local personnalisé :
  - Créer un répertoire
  - Ajout des RPMs souhaités dans le répertoire
  - Initialisation du dépôt : createrepo <répertoire_rpm>

- Mise à jour du dépôt :
  - Ajouter/mettre à jour les fichiers RPMs
  - Mettre à jour la base de données du dépôt : createrepo –update <répertoire_rpm>

CONCEPTS HPC
GESTION DES PAQUETS

- Création d'un dépôt des paquets de l'OS (depuis le DVD d'installation) :
  - Monter l'ISO du DVD : mount –o loop <fichier_iso> <point_montage>
  - Copier les fichiers RPMs (dans dossier packages) dans un répertoire dédié
  - Créer le dépôt : createrepo <répertoire_rpm>

- Création d'un dépôt des paquets de l'OS (depuis un dépôt en ligne – versions mises à jour) :
  - Trouver un miroir adapté sur http://centos.org/downloads/mirrors/ (site proche si possible) et récupérer l'adresse rsync
  - Télécharger le dépôt entier : rsync –rz –progress rsync://[…] <répertoire> (attention à la taille nécessaire)
  - Créer le dépôt : createrepo <répertoire>

- Configuration du dépôt sur les nœuds :
  - Dans répertoire /etc/yum.repos.d/
  - Modifier ou créer fichier pour pointer vers baseurl=http://<serveur>/<dossier_rpm> ou baseurl=file:///<serveur>/<dossier_rpm>
  - Désactiver le gpgcheck

CONCEPTS HPC
NTP

- NTP = Network Time Protocol
- Nécessaire dans un cluster pour la synchronisation des horloges
- Configuration :
  - Paquet ntp
  - Fichier de configuration /etc/ntp.conf

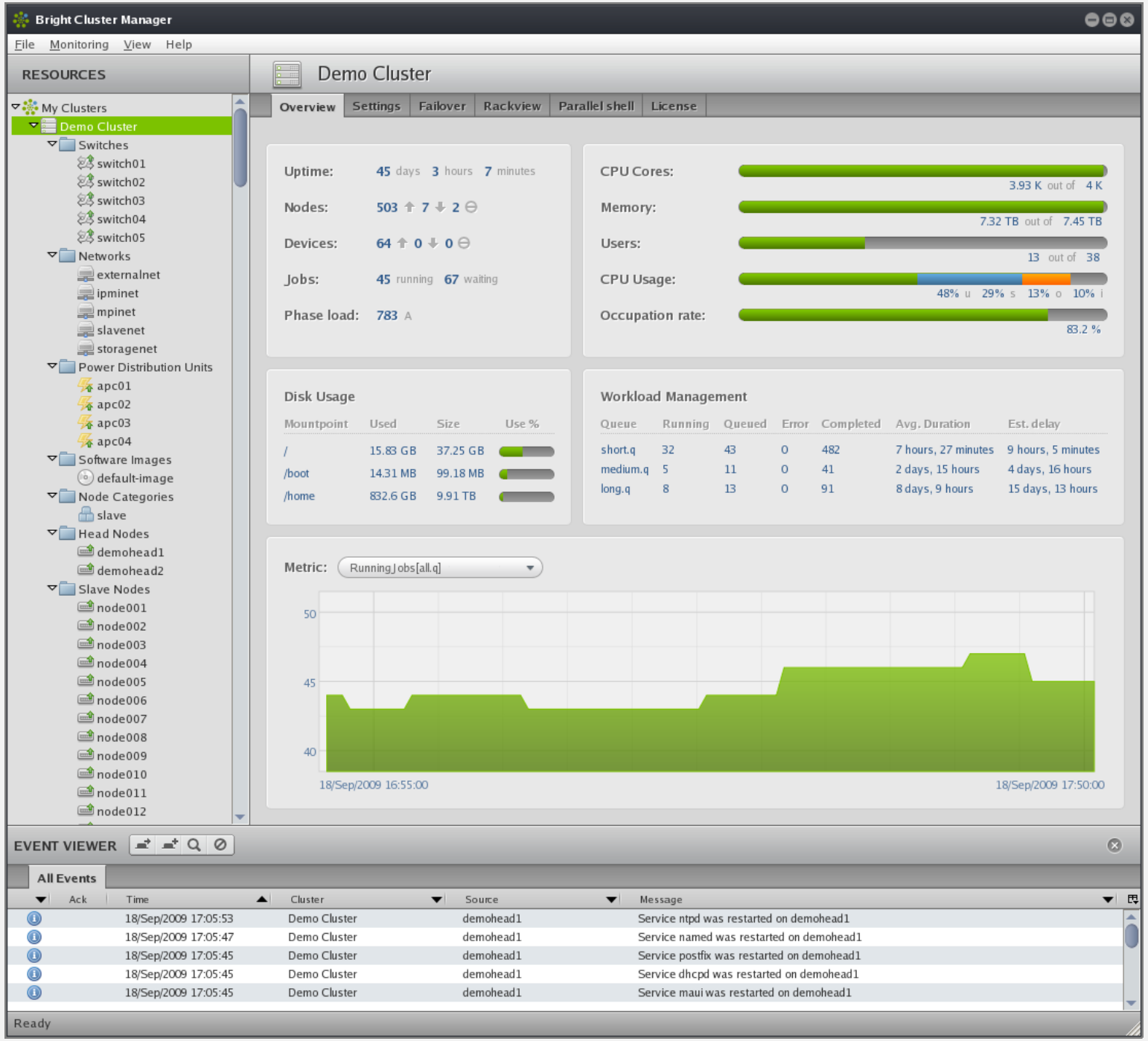
- Serveur maître :
  - Restrict <network_range> mask <netmask> nomodify notrap
  - Server <serveur_externe> iburst

- Nœuds de calcul :
  - Server <ip_master> iburst

- Redémarrer le service ntpd en cas de modification du fichier de configuration
- Pour vérifier : ntpq -p

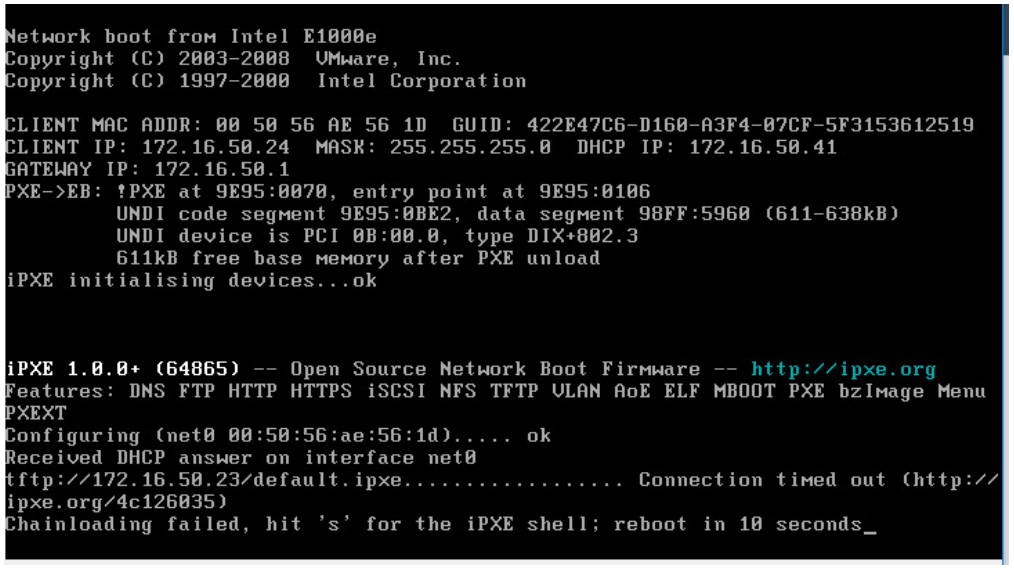
CONCEPTS HPC ADMINISTRATION

- Pour administrer le cluster, le compte root est activé sur toutes les machines
- Pour + de facilité, on peut déployer la clé SSH du compte root du maître sur les nœuds afin que la connexion sans mot de passe puisse s'effectuer :
  - Commande « ssk-keygen » sur le maître, qui génère /root/.ssh/id_rsa.pub
  - Copie du fichier id_rsa.pub dans /root/.ssh/authorized_keys sur nœuds
- Il existe des solutions de gestion de cluster, par exemple : xCAT, Bright Cluster Manager, Rocks, Cobbler, OpenHPC… Ces solutions facilitent grandement la gestion.



CONCEPTS HPC
INSTALLATION D'UN NŒUD

- Le processus général d'installation d'un nœud de calcul est :
  - Le nœud démarre en mode PXE (sur le réseau)
  - Le serveur DHCP attribue une adresse au nœud
  - Il charge certains fichiers pour démarrer un noyau minimal (initrd et vmlinuz)
  - Il reçoit un fichier de configuration décrivant l'installation souhaitée
  - Il reçoit les paquets du système et des applications puis les installe
  - Il redémarre puis (optionnel) peut recevoir des scripts de post-installation



CONCEPTS HPC
PROCÉDURES MARCHE/ARRÊT

- La hiérarchie des services du cluster induit des interdépendances qui conditionnent l'ordre d'arrêt/démarrage du cluster.
- Le non-respect de ces procédures peut provoquer le dysfonctionnement temporaire de certains services mais également des délais importants dans leur exécution.
- Procédure d'arrêt complet :
  - Arrêter les serveurs de stockage
  - Arrêter tous les nœuds de calcul
  - Arrêter le serveur maître
- Procédure de démarrage :
  - Démarrer les serveurs de stockage
  - Démarrer le serveur maître
  - Démarrer les nœuds de calcul

INSTALLATION MAÎTRE CLUSTER

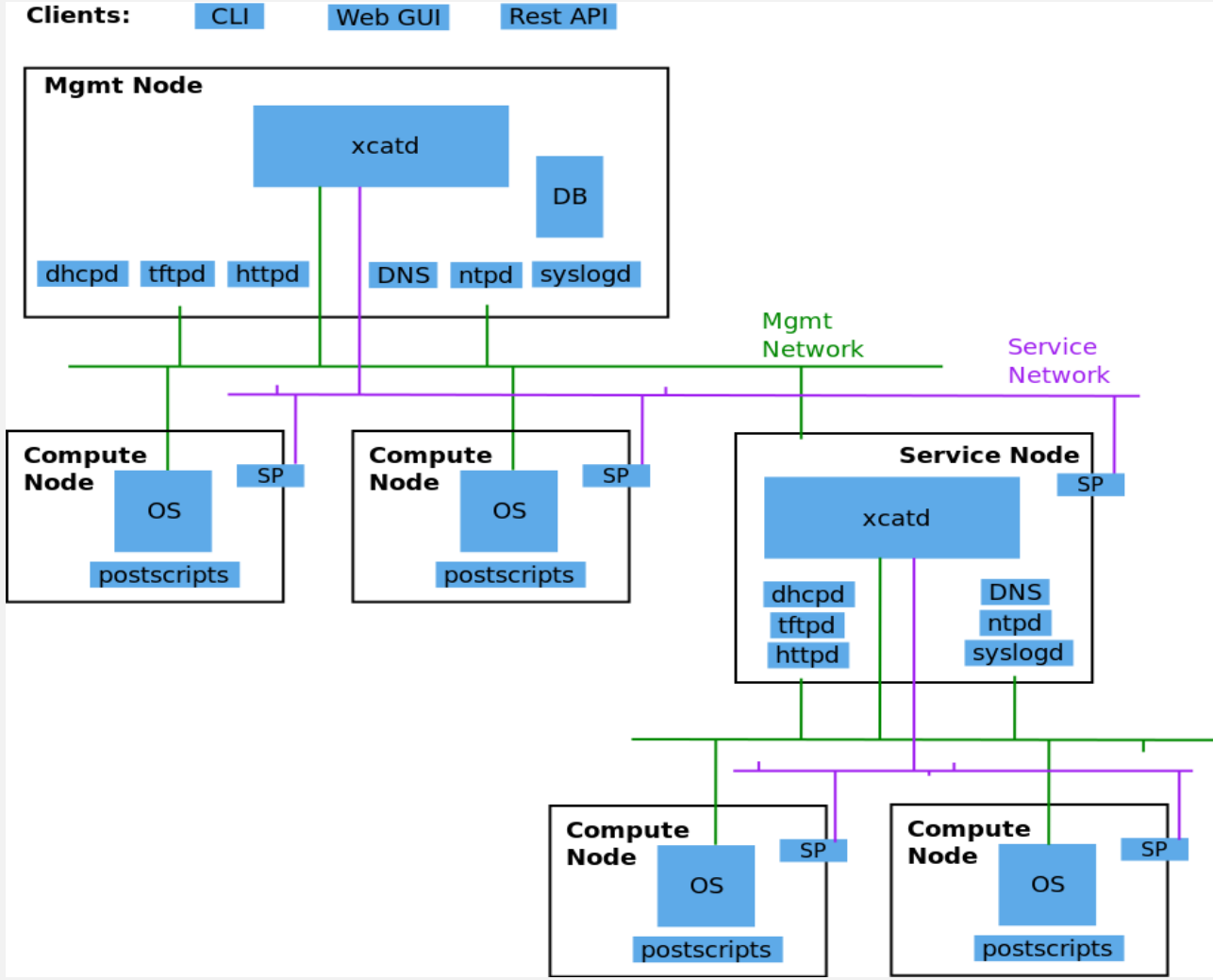
- La procédure générale pour installer et configurer le maître d'un cluster est la suivante :
  - Installation du système d'exploitation à partir du DVD
    - Attention au partitionnement : séparer au minimum le /, le /boot et le /home
  - Désactivation de SELinux dans /etc/sysconfig/selinux
  - Configuration des repositories de la distribution pour le déploiement et la mise à jour des nœuds de calcul
  - Configuration de l'authentification des utilisateurs
  - Configuration des serveurs NTP, DHCP, NFS, DNS (optionnel)
  - Configuration de la surveillance
  - Configuration du gestionnaire de jobs

XCAT
PRÉSENTATION

- xCAT = Extreme Cluster Administration Toolkit (version actuelle : 2.13)
- xCAT permet de:
  - déployer des clusters
  - installer des nœuds de manière diskful ou diskless
  - contrôler à distance les machines
  - installer et configurer les applications utilisateur
- Documentation xCAT : http://xcat-docs.readthedocs.io/en/stable/

XCAT ARCHITECTURE

- **Mgmt Node** : Serveur maître sur lequel xCAT est installé.
- **Service node** : En cas de très gros cluster, permettent d'assister le nœud maître.
- **Compute node** : Nœuds de calcul, gérés par xCAT.
- **SP = Service Processor** : Module pour gestion hardware à distance (IPMI)
- **Mgmt network** : Réseau administration pour installer et gérer les nœuds.
- **Service network** : Réseau pour l'administration IPMI
- **Application network** : Réseau utilisé pour les communications des applications (Infiniband, Ethernet 10 Gb)



XCAT
CONCEPTS DE BASE

- **Objets et base de données** : Les unités que xCAT peut gérer sont les objets. Ils permettent de représenter les entités physiques et logiques du cluster. Chaque objet possède des attributs qui sont stockés dans une base de données, constituée de tables et de champs. Les administrateurs de xCAT peuvent lire et modifier ces informations.

- **Configuration globale** : Il existe des configurations globales qui contrôlent le comportement de xCAT. Certains objets doivent être obligatoirement renseignés avant d'utiliser xCAT.

- **Réseau** : Le but de xCAT est de gérer et configurer les nœuds du cluster à partir du nœud maître. Toute la gestion matérielle, la configuration et le déploiement du système, et l'installation/configuration des applications sont faites à travers le réseau. Il faut donc bien définir cet aspect pour configurer le cluster.

XCAT
OBJETS

- Objets importants :
  - **Node** : Objet le plus important. Toute machine physique, virtuelle ou IPMI est définie comme objet node. Contient des attributs comme le groupe, l'adresse MAC, le réseau de déploiement, l'image OS.
    - Création node : mkdef –t node <nodename> <attributes>
  - **Group** : Cet objet regroupe des nodes. Pratique pour réaliser des opérations de groupe ou configurer plusieurs nœuds en même temps. Les propriétés des nœuds d'un groupe sont partagées.
    - Groupes automatiquement créés quand attributs group d'un nœud sont définis
  - **Osimage** : Représente l'OS qui sera installé par xCAT. Contient des attributs comme l'architecture système, la distribution, la liste de paquets à installer, ou des fichiers à synchroniser dans l'image
    - Plusieurs sont générées par la commande copycds (pour créer les dépôts de paquets à partir du DVD d'installation)

      Lier une osimage à un nœud : nodeset <nodename> osimage=<osimage>

- Commandes pour manipuler les objets :
  - Mkdef pour créer un objet
  - Chdef pour modifier un objet
  - Lsdef pour lister les objets
  - Rmdef pour supprimer un objet

XCAT
BASE DE DONNÉES

- Tables importantes de la base de données :
  - Site : paramètres globaux du cluster
  - Policy : configure les permissions d'accès à xCAT
  - Passwd : contient les noms d'utilisateurs et mots de passe afin que xCAT puisse accéder aux composants du cluster.
  - Networks : contient les définitions des réseaux du cluster.

- Commandes pour manipuler les tables :
  - Tabdump pour afficher une table
  - Tabedit pour éditer une table

XCAT
CONFIGURATION GLOBALE

- La configuration globale de xCAT est stockée dans la table site.
- Les attributs importants sont les suivants :
  - Dhcpinterfaces -> Réseaux sur lesquels le DHCP doit écouter.
  - Managedaddressmode -> Mode de configuration réseau durant l'installation d'un nœud. Peut être statique ou DHCP (par défaut).
  - Domain -> Nom de domaine DNS utilisé par le cluster.
  - Forwarders -> Serveurs DNS à l'extérieur du cluster.
  - Master -> Nom d'hôte du serveur maître.
  - Nameservers -> Liste de serveurs DNS que le cluster doit utiliser (souvent le serveur maître).
  - Dnsinterfaces -> Réseaux sur lesquels le DNS doit écouter.
  - Installdir -> Répertoire utilisé pour stocker les paquets de déploiement des nœuds

XCAT
RÉSEAUX

- Réseau d'administration : Utilisé par le serveur maître pour installer et gérer l'OS des nœuds. Les services suivants sont nécessaires :
  - DNS = offre des services de nom de domaine pour le cluster
  - HTTP = sert de serveur de téléchargement pour les fichiers kernel et initrd, la configuration de l'installateur et les repositories
  - DHCP = offre des services de DHCP pour le cluster
  - TFTP = sert de serveur de téléchargement pour le bootloader et les fichiers kernel et initrd
  - NFS = permet le partage de système de fichiers entre le maître et les nœuds de calcul
  - NTP = permet la synchronisation du temps pour le cluster
- Réseau de service : Utilisé par le maître pour contrôler les nœuds par IPMI (peut-être le même que réseau admin si IPMI en mode partagé)
- Réseau application : Utilisé par les applications sur les nœuds de calcul, généralement réseau rapide
- Réseau public : Utilisé pour accéder au maître depuis l'extérieur du cluster

XCAT
TYPES D'INSTALLATION DES NOEUDS

- **Stateful/diskful**
  - Cluster traditionnel avec OS installé sur chaque disque local des nœuds
  - Avantage : Approche familière pour les administrateurs
  - Inconvénients : Gérer chaque copie de l'OS, devoir faire face aux pannes de disque dur
- **Stateless/diskless**
  - Les nœuds démarrent depuis une image RAMdisk téléchargée depuis le maître
  - Avantage : Gestion centralisée de l'image OS
  - Inconvénient : Limité par l'utilisation de la RAM pour stocker l'image
  - Chaque nœud peut avoir un disque « scratch » (pour /swap, /tmp, /var, /log, …) lorsque certaines données à écrire sont très grandes ou si on veut en garder après un redémarrage

XCAT

- xCAT peut également découvrir et configurer les éléments suivants :
  - PDUs (Power Distribution Units), via SNMP -> pour surveiller utilisation électrique et contrôler alimentation des machines
  - Switchs Ethernet et Infiniband -> pour modifier identifiants, créer/gérer les VLANs
  - Drivers Infiniband -> pour installer les cartes IB sur les nœuds du cluster, la mise à jour des firmwares
  - Nvidia GPUs -> pour les outils CUDA, la mise à jour des pilotes

XCAT
I : INSTALLATION DU MAÎTRE

- Préparation du serveur maître :
  - Installation de l'OS sur le nœud maître
  - Copie de l'ISO du DVD d'installation et montage sur le maître
  - Création d'un fichier repo yum qui pointe vers le montage de l'ISO
  - Paramétrer le nom d'hôte et l'IP statique
  - Paramétrer les domaines de recherche et les DNS

XCAT
I : INSTALLATION DU MAÎTRE

- Installation de xCAT : 2 méthodes
  - Installation automatique
    - Récupération de l'outil go-xcat qui permet d'installer ou mettre à jour xCAT
    - Exécution de l'outil
  - Installation manuelle
    - Récupération des fichiers d'installation depuis http://xcat.org/download.html
    - xCAT-core.repo et xCAT-dep.repo doivent être copiés dans /etc/yum.repos.d/
    - Installation de xCAT avec la commande « yum install xcat »
    - Ajout des commandes xCAT au path : source /etc/profile.d/xcat.sh
    - Pour vérifier initialisation de la base de données : tabdump site
    - Service : xcatd

- Mise à jour de xCAT : « yum update '*xCAT*' » et « yum update '*xcat*' »

XCAT
2 : CONFIGURATION XCAT

- Saisir les champs suivants dans la table site : domain, forwarders, master, nameservers (chdef –t site <champ>=<valeur>)
- Initialiser les services DNS :
  - Makedns –n
- Définir la table networks
  - Un réseau par défaut est déjà créé, avec le même netmask et la même passerelle
  - Pour définir un réseau additionnel :
    - Mkdef –t network –o <network_name> <attributes>
    - Ou tabedit networks
    - Ou commande « makenetworks>
- Initialisation des services DHCP :
  - chdef –t site dhcpinterfaces=« ethX »
  - Makedhcp -n

XCAT
2 : CONFIGURATION XCAT

- Configuration des mots de passe :
  - Pour le compte root : chtab key=system passwd.usrname=root passwd.password=<password>
  - Pour crypter la clé : chtab key=system passwd.username=root passwd.password=`openssl passwd -1 <passwd>`
  - Pour l'IPMI : chtab key=ipmi passwd.username=ADMIN passwd.password=ADMIN

XCAT
3 : DÉCOUVERTE DES NOEUDS

- Les machines doivent être définies dans la table nodes. 2 possibilités :
  - Créer les nodes un par un en utilisant la commande mkdef
  - Utiliser les fonctionnalités de découverte automatique de xCAT
- Types de découverte automatique :
  - Par switch : les nœuds sont découverts selon le port au switch auquel ils sont connectés
  - Séquentielle : les nœuds sont prédéfinis et xCAT attribue un nom et une adresse IP quand il détecte un nœud

XCAT
INSTALLATION DISKFUL

- Création initiale de l'image : copycds <fichier .iso>
  - Crée une liste de osimages qu'on peut lister avec lsdef –t osimage ou voir ses attributs en spécifiant son nom à la fin de la commande
- Nom d'une osimage générée par copycds :
  - <os>-<arch>-install-compute : osimage par défaut pour une installation diskful
  - <os>-<arch>-netboot-compute : osimage par défaut pour une installation diskless
  - <os>-<arch>-install-service : osimage par défaut pour l'installation d'un nœud de service
- Modifier nom osimage : chdef –t osimage <ancien_nom> -n <nouveau_nom>
- Initialisation du nœud pour le déploiement : nodeset <nodename> osimage=<osimage>
- Démarrer l'installation de l'OS :
  - Rsetboot <node> net
  - Rpower <node> reset

XCAT
INSTALLATION DISKLESS

- Création initiale de l'image : copycds <fichier .iso>
  - Crée une liste de osimages qu'on peut lister avec lsdef –t osimage (ou tabdump osimage) ou voir ses attributs en spécifiant son nom à la fin de la commande
- Nom d'une osimage générée par copycds :
  - <os>-<arch>-install-compute : osimage par défaut pour une installation diskful
  - <os>-<arch>-netboot-compute : osimage par défaut pour une installation diskless
  - <os>-<arch>-install-service : osimage par défaut pour l'installation d'un nœud de service
- Modifier nom osimage : chdef –t osimage <ancien_nom> -n <nouveau_nom>
- Génération de l'image : genimage <osimage>
- Modification de l'image : Modifier les fichiers dans dossier rootimg de l'image
- Archivage de l'image : packimage <osimage>
- Initialisation du nœud pour le déploiement :
  - Chdef <noderange> provmethod=<osimage>
  - Nodeset <noderange> osimage=<osimage>
- Démarrage de l'installation de l'OS :
  - Rsetboot <node> net
  - Rpower <node> reset

XCAT
UPDATENODE

- Commande updatenode : Pour mettre à jour/modifier les nœuds après installation initiale
  - Ajouter software additionnel :
    - Dans fichier .pkglist (pour paquets de l'OS) et fichier .otherpkgs.pkglist (pour paquets autres)
    - Puis commande updatenode <noderange> -S
  - Re-exécuter ou ajouter des prescripts/postscripts (tabdump postsripts) :
    - Commande updatenode <noderange> -P pour re-exécuter les postscripts
  - Synchroniser des fichiers de configuration :
    - Fichier synclist
    - Commande updatenode <noderange> -F
  - Mettre à jour les clés SSH :
    - Commande updatenode <noderange> -K

XCAT
COMMANDES PARALLÈLES

- Pping : Ping parallèle
- Ppping : Ping parallèle entre nœuds
- Prsync : rsync parallèle
- Pscp : scp parallèle
- Psh : sh parallèle
- Xdcp : copie de fichiers parallèle (source et destination)
- Xdsh : exécution de commandes parallèles

TORQUE/MAUI
PRÉSENTATION TORQUE

- Torque : Gestionnaire de ressources -> permet de commencer, mettre en pause, annuler, surveiller les jobs
- Documentation Torque : http://www.adaptivecomputing.com/support/documentation-index/torque-resource-manager-documentation/
- Composants du cluster gérés par Torque :
  - Nœud maître -> machine sur laquelle s'exécute le service pbs_server
  - Nœuds de soumission/nœuds interactifs -> permettent un point d'entrée au cluster pour les utilisateurs
  - Nœuds de calcul -> machines sur lesquelles les jobs s'exécutent. Le service pbs_mom permet de démarrer, arrêter et gérer les jobs soumis. Il communique avec psb_server sur le maître. Le nœud maître n'est généralement pas un nœud de calcul
  - Ressources -> optionnelles, il s'agit de réseaux, systèmes de stockage, gestionnaires de licences, …qui peuvent être gérés par Torque
- Cycle de vie d'un job :
  - Création -> un script de soumission contient les instructions et paramètres du job (durée d'exécution, ressources nécessaires, commandes à exécuter)
  - Soumission -> le job est soumis avec la commande qsub. Une fois soumis, les stratégies définies par les administrateurs du cluster indiquent la priorité du job (quand il peut s'exécuter, quelles ressources il peut utiliser…)
  - Exécution -> la plus grosse partie du cycle, durant laquelle le job s'exécute. Son statut peut-être visionné avec qstat.
  - Finalisation -> une fois le job terminé, les fichiers stdout et stderr sont copiés dans le répertoire depuis lequel a job a été soumis.

TORQUE/MAUI
INSTALLATION TORQUE

- Installation des dépendances requises
- Téléchargement de la dernière version sur site Adaptive Computing
- Configuration, compilation et installation
- Torque est installé dans /var/spool/torque
- Le service trqauthd doit être lancé, il permet aux utilitaires clients d'authentifier les connexions des utilisateurs au serveur maître
- Le fichier torque/server_name doit contenir le nom du serveur maître
- Initialisation de la base de données serverdb en exécutant le script torque.setup
- Ajout des nœuds de calcul au fichier torque/server_priv/nodes
- Démarrage des services pbs_server et pbs_mom

TORQUE/MAUI
COMMANDES TORQUE

- Soumission de job :
  - Commande qsub -> prend un fichier en paramètre
  - Le script est exécuté depuis le dossier home de l'utilisateur
  - Les variables d'environnement par défaut sont utilisées sauf si l'option –v ou –V est utilisée pour personnaliser l'environnement
  - Les directives PBS doivent être déclarées en premier dans le script, par exemple :
    - #PBS –S /bin/bash
    - #PBS –m abe
    - #PBS –M email@company.com
- Surveillance des jobs : commande qstat
  - Si utilisateur sans droits admin -> seuls ses jobs sont affichés
  - Pour afficher les queues : qstat –f -Q
- Annulation de job : qdel

TORQUE/MAUI
FIN D'UN JOB TORQUE

- Une fois qu'un job est terminé, l'attribut exit_status indique le résultat du script. Il peut être vu avec la commande qstat –f.
- Les valeurs possibles sont :
  - 0 = JOB_EXEC_OK -> Succès
  - -1 = JOB_EXEC_FAIL1 -> Job a echoué avant les fichiers, pas de nouvel essai
  - -2 = JOB_EXEC_FAIL2 -> Job a echoué après les fichiers, pas de nouvel essai
  - -3 = JOB_EXEC_RETRY -> Job a échoué, nouvel essai
  - -4 = JOB_EXEC_INITABT -> Job annulé lors de l'initialisation de MOM
  - -5 = JOB_EXEC_INITRST -> Job annulé lors de l'initialisation de MOM, pas de migration
  - -6 = JOB_EXEC_INITRMG -> Job annulé lors de l'initialisation de MOM, migration
  - -7 = JOB_EXEC_BADRESRT -> Redémarrage du job a échoué
  - -8 = JOB_EXEC_CMDFAIL -> Exécution de la commande a échoué
  - -9 = JOB_EXEC_STDOUTFAIL -> Les fichiers stdout stderr n'ont pas pu être créés/ouverts
  - -10 = JOB_EXEC_OVERLIMIT_MEM -> Limite mémoire dépassée
  - -11 = JOB_EXEC_OVERLIMIT_WT -> Limite walltime dépassée
  - -12 = JOB_EXEC_OVERLIMIT_CPUT -> Limite temps CPU dépassée

TORQUE/MAUI
GESTION DES NŒUDS ET QUEUES TORQUE

- Pour afficher la liste des nœuds : pbsnodes
- Pour ajouter des nœuds, 2 possibilités :
  - En temps réel avec la commande qmgr
  - En éditant le fichier torque/server_priv/nodes, il faut redémarrer le service pbs_server après
- Torque peut associer des propriétés aux nœuds -> pratique si cluster hétérogène
- Pour empêcher les jobs de s'exécuter sur un nœud en particulier, on peut le mettre hors ligne avec la commande
  - Pbsnodes –o <nodename>
- Pour le remettre en ligne :
  - Pbsnodes –c <nodename>

- Pour créer une queue : commande qmgr

TORQUE/MAUI SUPPORT MPI

- MPI : Message Passing Interface -> Librairie utilisée par les jobs parallèles pour améliorer les communications entre les tâches distribuées à travers le cluster
- MPICH supportée par Torque -> Mpiexec remplace mpirun
- OpenMPI aussi supportée par Torque -> doit être configurée avec l'option –with-tm

TORQUE/MAUI
RESSOURCES TORQUE

- Torque gère aussi des ressources divisées en 3 catégories :
  - Configurations
    - Architecture -> système d'exploitation du nœud
    - Node Features -> propriétés du nœud
    - Local disk -> espace disque local
    - Memory -> RAM locale
    - Processors -> processeurs physiques/virtuels
    - Swap -> mémoire swap virtuelle
  - Utilisation
    - Disk -> espace disque disponible
    - Memory -> mémoire vive disponible
    - Network -> charge réseau
    - Processor utilisation -> charge CPU moyenne sur 1 minute
  - Etat du nœud
    - Idle time -> temps d'inactivité
    - State -> état du noeud

TORQUE/MAUI
ETATS DES NŒUDS

- Un nœud de calcul peut être dans un ou plusieurs de ces états :
  - Busy -> le nœud est occupé et n'accepte plus de job additionnel
  - Down -> le nœud n'est pas actif, il est inutilisable
  - Free -> le nœud est libre, prêt à être utilisé
  - Job-exclusive -> tous les processeurs disponibles sont assignés à des jobs
  - Job-sharing -> plusieurs jobs partagés s'exécutent sur le nœud qui restera dans cet été jusqu'à leur complétion
  - Offline -> le nœud a été mis hors ligne
  - Reserve -> le nœud a été réservé par le maître
  - Time-shared -> le nœud autorise des jobs à s'exécuter en parallèle
  - Unknown -> l'état du nœud n'a pas été détecté

TORQUE/MAUI
PRÉSENTATION MAUI

- Maui : Ordonnanceur conçu pour les clusters -> Permet de gérer les réservations des ressources
- Fonctionne avec n'importe quel gestionnaire de ressources dont Torque
- Un ordonnanceur détermine quand, où et comment exécuter les jobs pour optimiser l'utilisation du cluster. Ses buts sont :
  - Contrôle du trafic -> Empêcher des jobs d'interférer, et ainsi de pouvoir ne pas s'exécuter par manque de ressources
  - Stratégies -> Permettre de transformer les objectifs du cluster en stratégies d'ordonnancement (qui ou quoi est autorisé à utiliser le cluster)
  - Optimisations -> Un ordonnancement intelligent facilite un volume de jobs conséquent et leur permet de se terminer plus vite.
- Différents types de jobs :
  - Batch -> Fichier contenant des commandes à exécuter ainsi que les ressources et l'environnement souhaités. Le job est soumis dans une queue et est exécuté une fois que les ressources sont disponibles
  - Interactif -> Quand la personne qui soumet le job souhaite une réponse immédiate. Généralement en utilisant une interface Web ou un terminal graphique.
  - Calendaire -> Job exécuté à une heure donnée et parfois de manière régulière. Maui garantit alors la disponibilité des ressources à cet instant.
  - Service -> Maui peut programmer et gérer des services persistants ou à longue durée.

TORQUE/MAUI
INSTALLATION DE MAUI

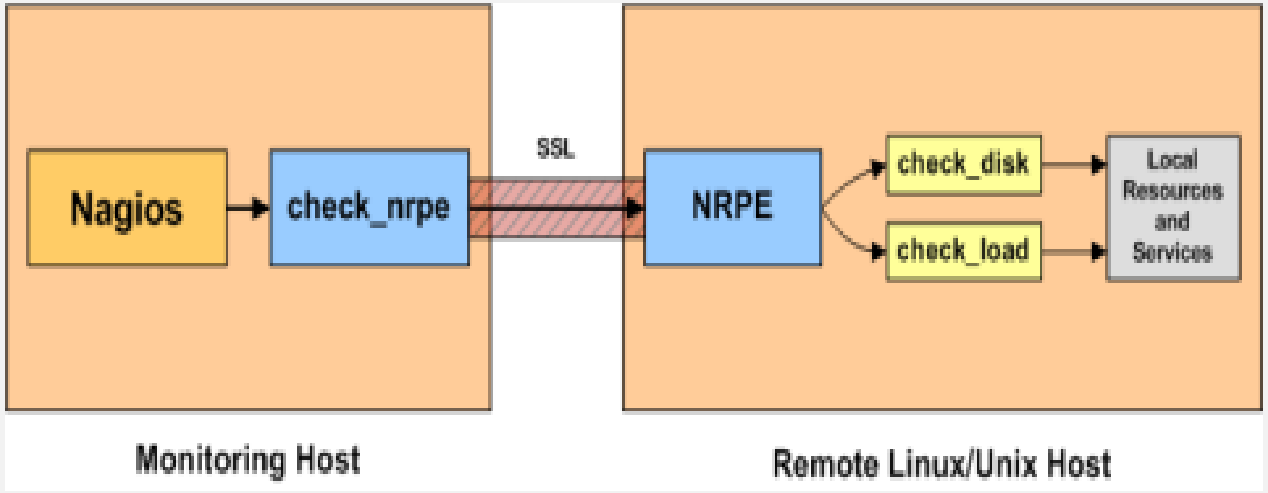
- Téléchargement archive Maui et extraction
- Configuration avec l'option –with-pbs=/opt/torque/
- Compilation et installation
- Démarrage du service Maui
- Sur cluster chimatmos : Maui installé dans /usr/local/maui

TORQUE/MAUI
CONFIGURATION ET COMMANDES MAUI

- S'assurer qu'une entrée correspondant à pbsserver existe dans le fichier /etc/hosts
- Commandes Maui :
  - Schedctl -> pour gérer le service et le comportement de l'ordonnanceur
  - Checkjob -> pour afficher état des jobs
  - Checknode -> pour afficher état d'un nœud
  - Diagnose -> afficher infos résumées sur job/nœud/réservation
  - Showgrid -> afficher une matrice durée du job/taille du job
  - Showq -> afficher différentes vues des jobs en queue, en attente et non éligibles
  - Showstat -> afficher l'utilisation du cluster actuelle et historique
  - Runjob/canceljob -> démarrer/annuler un job
  - Sethold/setqos/setspri -> configurer la pause, le QoS et la priorité d'un job
  - Releaseres -> retirer les réservations en cours
  - Showconfig -> afficher les paramètres

SUPERVISION
PRÉSENTATION NAGIOS

- Nagios : outil permettant de surveiller l'état de divers composants et services du cluster
- Installé sur le serveur maître et récupère des informations provenant l'ensemble des nœuds, switchs, PDUs…
- Fonctionnement de Nagios :



- Le service nagios s'exécute sur le maître et effectue check_nrpe à intervalles réguliers. Ceci envoie des requêtes aux services nrpe qui sont exécutés sur les nœuds. Les services nrpe exécutent des commandes pour vérifier les éléments surveillés (procs, disques, RAM, SSH, ping, …). Les informations récoltées prennent le chemin inverse et sont récupérées par le serveur nagios, qui les affichent sur une interface Web.

SUPERVISION
INSTALLATION ET CONFIGURATION NAGIOS

- Nagios est disponible dans le dépôt EPEL
- Paquets pour le serveur maître : nagios, nagios-plugins-all, nagios-plugins-nrpe, nrpe
- Paquets pour les nœuds de calcul : nagios-plugins-all, nagios-plugins-nrpe, nrpe

- Fichiers de configuration Nagios dans /etc/nagios :
  - Nagios.cfg -> Configuration générale
  - Répertoire objects : Contient les définitions des machines, commandes, services et contacts
  - Nrpe.cfg -> Configuration des commandes à exécuter pour vérifier les éléments souhaités

- Accès à l'interface Web par http://<adresse_serveur>/nagios
- Nom utilisateur et mot de passe = nagiosadmin

- On peut configurer Nagios de manière à envoyer des mails d'alerte lorsqu'un élément surveillé passe en état « warning » ou « critical ».

SUPERVISION
PRÉSENTATION GANGLIA

- Ganglia : outil qui permet de présenter des informations sur l'utilisation du cluster sous forme de graphique
- Comporte une vue synthétique d'utilisation du cluster (réseau, RAM, disques, processeurs…) ainsi qu'une vue détaillée pour chaque métrique surveillée.
- Ganglia se compose de 3 entités :
  - Gmetad -> service pour la réception des données, présent sur serveur maître uniquement
  - Ganglia-web : service pour afficher la page Web, présent sur serveur maître uniquement
  - Gmond -> service pour l'envoi des données, présent sur toutes les machines

SUPERVISION
INSTALLATION/CONFIGURATION GANGLIA

- Ganglia est disponible dans le repository EPEL

- Paquets nécessaires sur serveur maître : ganglia, ganglia-gmond, ganglia-gmetad
- Paquets nécessaires sur nœud de calcul : ganglia, ganglia-gmond

- Configuration de Ganglia dans /etc/ganglia/

- Interface Web disponible à http://<adresse_maître>/ganglia

LIBRAIRIES LINUX PRÉSENTATION

- Librairie = ensemble de fonctions, classes et méthodes regroupées dans un seul fichier
- Fonctions, classes et méthodes qui sont partagées par plusieurs applications
- Elles peuvent être appelées dans le code en faisant appel à la librairie entière

- Avantages :
  - Réutilisation des composants -> une seule librairie à mettre à jour, moins d'espace disque requis
  - Gestion des versions -> plusieurs versions différentes d'une même librairie peuvent cohabiter
  - Spécialisation des composants : les développeurs spécialisés peuvent se concentrer sur leurs compétences clés sur une seule librairie, les tests et vérifications sont simplifiés

LIBRAIRIES LINUX
TYPES DE LIBRAIRIES

- **Librairies statiques**
- Fichiers .a
- Le code de la librairie est dans son fichier, et est directement lié au programme lors de l'exécution. Le programme fait des copies du code de la librairie, qui devient partie du programme.
- Avantages : Exécution et chargement de la librairie plus rapides
- Inconvénients : Programmes plus lourds, nécessité de recompiler lors d'un changement de librairie

- **Librairies dynamiques**
- Fichiers .so
- Le code de la librairie est aussi dans son fichier, mais il est référencé par les programmes lors de leur exécution. Le code ne fait pas partie du programme.
- Avantages : Code moins lourd, librairies indépendantes du code (pratique pour mise à jour ou plusieurs versions)
- Inconvénient : Exécution et chargement des fonctions de la librairie plus lents

LIBRAIRIES LINUX IMPLÉMENTATIONS

- Les implémentations libres sont disponibles dans les répertoires /usr/local et /usr/lib64

- **ATLAS : Automatically Tuned Linear Algebra Software**
- Outil qui optimise principalement les librairies BLAS (Basic Linear Algebra Subprogram)
- Installation : yum install atlas atlas-devel
- Documentation complète : http://www.netlib.org/blas

- **LAPACK : Linear Algebra Package**
- Ensemble de sous-programmes Fortran permettant de résoudre des problèmes d'algèbre linéaire
- Installation : yum install lapack lapack-devel
- Documentation complète : http://www.netlib.org/lapack
- Sur cluster chimatmos : installé dans /opt/software/lapack

LIBRAIRIES LINUX IMPLÉMENTATIONS

- **ScaLAPACK** : Ensemble de sous-programmes Fortran permettant de résoudre des problèmes d'algèbre linéaire via MPI
- Installation : archive tgz depuis site internet
- Documentation : http://www.netlib.org/scalapack/index.html

- **FFTW** : Ensemble de sous-programmes C permettant le calcul de transformées de Fourier discrètes
- Installation : yum install fftw fftw-devel
- Pas de support MPI
- Documentation : http://www.fftw.org
- Sur cluster chimatmos : installé dans /opt/software/fftw

- Intel propose ses propres solutions optimisées nommées **Intel MKL**
- Documentation et téléchargement : https://software.intel.com/en-us/mkl
- Sur cluster chimatmos : installé dans /opt/software/intel

LIBRAIRIES LINUX IMPLÉMENTATIONS

- **HDF** : modèle pour stocker et gérer des données
- Téléchargement et documentation : https://support.hdfgroup.org/
- Type de téléchargement : Fichier source (bzip ou gzip)
- Sur cluster chimatmos : installé dans /opt/software/hdf

- **(P)netcdf = (Parallel) netCDF**
- Librairie d'entrée-sortie haute performance pour accéder à des fichiers compatibles avec le format NetCDF
- Téléchargement : http://cucis.ece.northwestern.edu/projects/PnetCDF/download.html
- Type de téléchargement : Fichier source tar.gz
- Sur cluster chimatmos : installé dans /opt/software/(p)netcdf

VARIABLES D'ENVIRONNEMENT

- Possibilité de personnaliser l'environnement utilisateur
- Fichiers à modifier dans répertoire utilisateur :
  - .bash_profile
  - .bashrc
- Les variables permettent de faire communiquer des informations entre différentes applications. Elles sont fournies par l'OS.
- Exemples de variables :
  - $TMP = chemin dossier temporaire
  - $HOME = chemin dossier personnel
  - $USER = nom utilisateur
  - $PATH = chemins de recherche pour commandes externes
- Pour définir une variable : « export <VARIABLE> »

COMPILATEURS
COMPILATEURS GNU

- Compilateurs libres multi-plateformes fournis par le système d'exploitation
- Offrent en général de bonnes performances et sont standards aux applications Linux
- Le système Linux lui-même est compilé à partir de ces compilateurs
- Compilateurs proposés : C, C++, Fortran

- Noms des paquets : gcc, gcc-c++, gcc-gfortran,

- Commandes pour invoquer les compilateurs :
  - Fortran -> gfortran [options] –o <programme.exe> <programme.f>
  - C -> gcc [options] –o <programme.exe> <programme.c>
  - C++ -> g++ [options] –o <programme.exe> <programme.c>
  - Débogueur GNU : gdb [options] <programme>

COMPILATEURS
COMPILATEURS INTEL

- Compilateurs propriétaires, optimisés pour les plateformes utilisant les processeurs du fabricant
- Utilisation à privilégier sur clusters à base Intel
- Offrent de nombreuses optimisations avancées
- Supportent la vectorisation, l'auto-parallélisation, la parallélisation via OpenMP,…

- Commandes pour invoquer les compilateurs Intel :
  - Fortran -> ifort [options] –o <programme.exe> <programme.f>
  - C -> icc [options] –o <programme.exe> <programme.c>
  - C++ -> icpc [options] –o <programme.exe> <programme.c>
  - Débogueur Intel : idb [options] <programme>

MODULES D'ENVIRONNEMENT

- Les modules d'environnement permettent de centraliser la gestion des environnements nécessaires à l'utilisation de logiciels ou librairies sans manipuler directement les variables d'environnement
- Ces modules sont écrits directement lors de l'installation et permettent d'éviter des erreurs (définition erronée par exemple) mais aussi de passer simplement d'une version à une autre sans conflit
- Les modules d'environnement peuvent altérer les variables d'environnement comme $PATH, $LD_LIBRARY_LOAD, …
- Ils peuvent être incorporés dans les scripts des gestionnaires de ressources (commandes #PBS par exemple)

- Commandes usuelles :
  - Module avail -> liste les différents modules disponibles
  - Module load <module> -> charge le module cible
  - Module list -> liste les modules chargés
  - Module unload <module> -> décharge le module cible
  - Module purge -> décharge tous les modules chargés
  - Module –V -> informations de configuration

MODULES D'ENVIRONNEMENT
MODULE FILES

- Les module files sont des fichiers qui indiquent les changements apportés à l'environnement lorsque le module est chargé
- Les répertoires des module files sont contenus dans la variable $MODULEPATH
- Un template de modulefile « use.own » est disponible pour créer des nouveaux fichiers module

PROGRAMMATION PARALLÈLE PRINCIPES

- Programmation parallèle = conception d'un code de calcul capable de résoudre un problème donné en utilisant plusieurs processeurs
- 2 solutions matérielles :
  - Mémoire partagée -> processeurs physiquement sur un même serveur, communications internes entre les processeurs
  - Mémoire distribuée -> cluster de calcul, communications réseaux sollicitées

- OpenMP : interface de programmation dédiée à la conception de codes sur systèmes à mémoire partagée
- Simple car le code généré reste assez proche du code séquentiel

- MPI = Message Passing Interface
- Bibliothèque offrant des routines de communication pour la parallélisation de code sur des systèmes à mémoire distribuée
- Code plus complexe car la parallélisation nécessite souvent de réécrire complètement le code

PROGRAMMATION PARALLÈLE PRINCIPES

- Possibilité d'utiliser la parallélisation sans réécrire le code si celui-ci a besoin de librairies de calcul standards
- Dans ce cas il suffit de compiler le code en utilisant des versions parallèles de ces librairies (GNU ou Intel MKL)
- Le code sera parallélisé à l'exécution lors des appels à ces librairies

- Sur un cluster, chaque nœud est un système à mémoire partagée -> possible d'utiliser OpenMP si tous les processeurs utilisés sont sur un même nœud
- Si on a besoin de processeurs se situant sur plusieurs nœuds -> la solution MPI s'impose

- Possibilité d'écrire des codes hybrides qui utilisent MPI entre les nœuds et OpenMP au sein même d'un nœud
- Solutions performantes mais qui sont complexes à mettre en œuvre

PROGRAMMATION PARALLÈLE OPENMP

- Parallélisation OpenMP réalisable depuis un code séquentiel
- Pour compiler avec le support OpenMP, il faut ajouter l'option –openmp (exemple : ifort –openmp exe.f90 –O exe)
- Le nombre de processus est défini dans une variable d'environnement externe : OMP_NUM_THREADS

PROGRAMMATION PARALLÈLE
MPI : PRÉSENTATION

- MPI = bibliothèque de communication qui permet la programmation sur des systèmes à mémoire distribuée (cluster de calcul)
- Disponible pour les langages C, C++ et Fortran
- Les fonctions proposées par MPI permettent de traiter :
  - Les communications entre les processus
  - La définition et le traitement des variables
  - La synchronisation des processus

PROGRAMMATION PARALLÈLE
MPI : LIBRAIRIES

- Les principales implémentations de MPI sont les suivantes :
  - **MPICH**
    - Implémentation libre du standard MPI1
    - Documentation : http://www.unix.mcs.anl.gov/mpi/mpich1)
  - **OpenMPI**
    - Implémentation du standard MPI2
    - Documentation : http://www.open-mpi.org
  - **MPICH3**
    - Implémentation du standard MPI2
    - Documentation : http://www.mcs.anl.gov/research/projects/mpich2
  - **Intel MPI**
    - Implémentation propriétaire du standard MPI2

PYTHON

- Version installée par défaut sur CentOS 6 : Python 2.6.6
- Cette version doit impérativement rester installée car yum par exemple en dépend
- Si on veut une version plus récente il faut garder la 2.6.6 à côté
- La nouvelle version de Python doit aller dans /usr/local/

PYTHON INSTALLATION

- 1e étape : télécharger les prérequis
  - yum install -y zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gdbm-devel db4-devel libpcap-devel xz-devel expat-devel
- 2e étape : télécharger, compiler et installer python
  - wget http://python.org/ftp/python[...]
  - tar xf Python-[…].tar.xz && cd Python-[...]
  - ./configure --prefix=/usr/local --enable-unicode=ucs4 --enable-shared LDFLAGS="-Wl,-rpath /usr/local/lib”
  - make && make altinstall
- 3e étape : installer pip, setuptools et wheel pour les interpréteurs Python
  - wget https://bootstrap.pypa.io/get-pip.py
  - python2.7 get-pip.py

- Python est alors installé dans /usr/local/bin/python2.7
- La version système de Python 2.6.6 reste dans /usr/bin/python, /usr/bin/python2 et /usr/bin/python2.6

GRIB

- GRIB = fichier de données météorologiques dans un format spécifique
- Documentation et téléchargement : https://software.ecmwf.int/wiki/display/GRIB/
- Type de téléchargement : fichier source tar.gz
- Installation :
  - .configure
  - .make
  - .make install
- Sur cluster chimatmos : installé dans /opt/software/grib