Architecture d'un cluster

Cluster: architecture, stratégie d'organisation
Cluster: installation

Christophe PERA,Emmanuel QUEMENER, Lois TAULEL, Hervé Gilquin et Dan CALIGARU

Fédération Lyonnaise de Modélisation et Sciences Numériques
Formation Cluster 2016
christophe.pera(AT)univ-lyon1.fr

19/01/2016

Sommaire.

Cluster
Architecture et exploitation.

Quelques infos.

Qui suis je ?	Un ingénieur qui essai de survivre aux milieu des utilisateurs hyperactifs du monde de la simulation numérique.
Ma fonction ?	Responsable d'un des centres de calcul d'une Structure Fédérative de Recherche, la Fédération Lyonnaise de Modélisation et Sciences Numériques (École Centrale Lyon, École Nationale Supérieure Lyon, Université Lyon 1).
Nos Objectifs	Définir ce qu'est un cluster de calcul (HPC ?), son architecture, ses principaux composants et analyser les étapes théoriques et pratiques de sa construction.

L'histoire d'une démocratisation.

Dans les années 90s, apparition d'un environnement riche pour l'émergence du cluster :

- A l'origine, des Unix propriétaires, supercalculateurs !
- 1977 Premier cluster produit par DataNet.
- 1980 VAXcluster produit par DEC.
- Début des années 1990, la FSF créé un certain nombre d'outils de programmation libres, le PC se diversifie.
- 1991, Linus Torvald, linux et une licence Copyleft !
- V1.0 du système ouvert PVM en 1989.
- 1993, premier cluster "HIGH PERFORMANCE"
  Donald Becker and Thomas Sterling began sketching the outline of a commodity-based cluster system designed as a cost-effective alternative to large supercomputers. In early 1994, working at CESDIS under the sponsorship of the HPCC/ESS project, the Beowulf Project was started.
  (http ://www.beowulf.org/overview/history.html)
- V1.0 du standard MPI en 1994.
- V1.0 du stanard OpenMP en 1997/1998.

Un environnement propice

Plusieurs facteurs au succès de l'architecture "Cluster" Beowolf :

1. Une concurrence acharnée :
   - La production massive des matériels de calcul pour les besoins personnels, le jeu et l'animation abaissent le cout de production de ces composants.
   - Une fabrication de composants riches pré-assemblés et standardisés (processeurs/carte mére/mémoire/disque dur/..)
   - Une baisse des prix et une meilleur fiabilité.
2. Les logiciels "OpenSource" (Linux operating system, GNU compilers and programming tools and MPI and PVM message passing libraries -> accessibilité, standard de fait).
3. Développement des algorithmes parallèles en simulation numérique.
4. Une constatation : Difficile d'obtenir des performances, même avec du matériel propriétaire sans effort et travail des intervenants (administrateur système/réseau, utilisateurs/développeur).
5. l'explosion des besoins de calcul.

Définitions et remarques

Cluster (Wikipedia) : Cluster may refer to "computing".
A Computer cluster is a group of loosely coupled computers that work together closely.

Quelques précisions :

- Des matériels de calcul et des réseaux homogènes.
- Des opérations communes.
- Différent des "Grilles de calcul" et du CLOUD qui n'imposent pas une interconnexion entre matériel de calcul !
- Différents clusters en informatique : HA, Load-balancing, Computing/HPC.

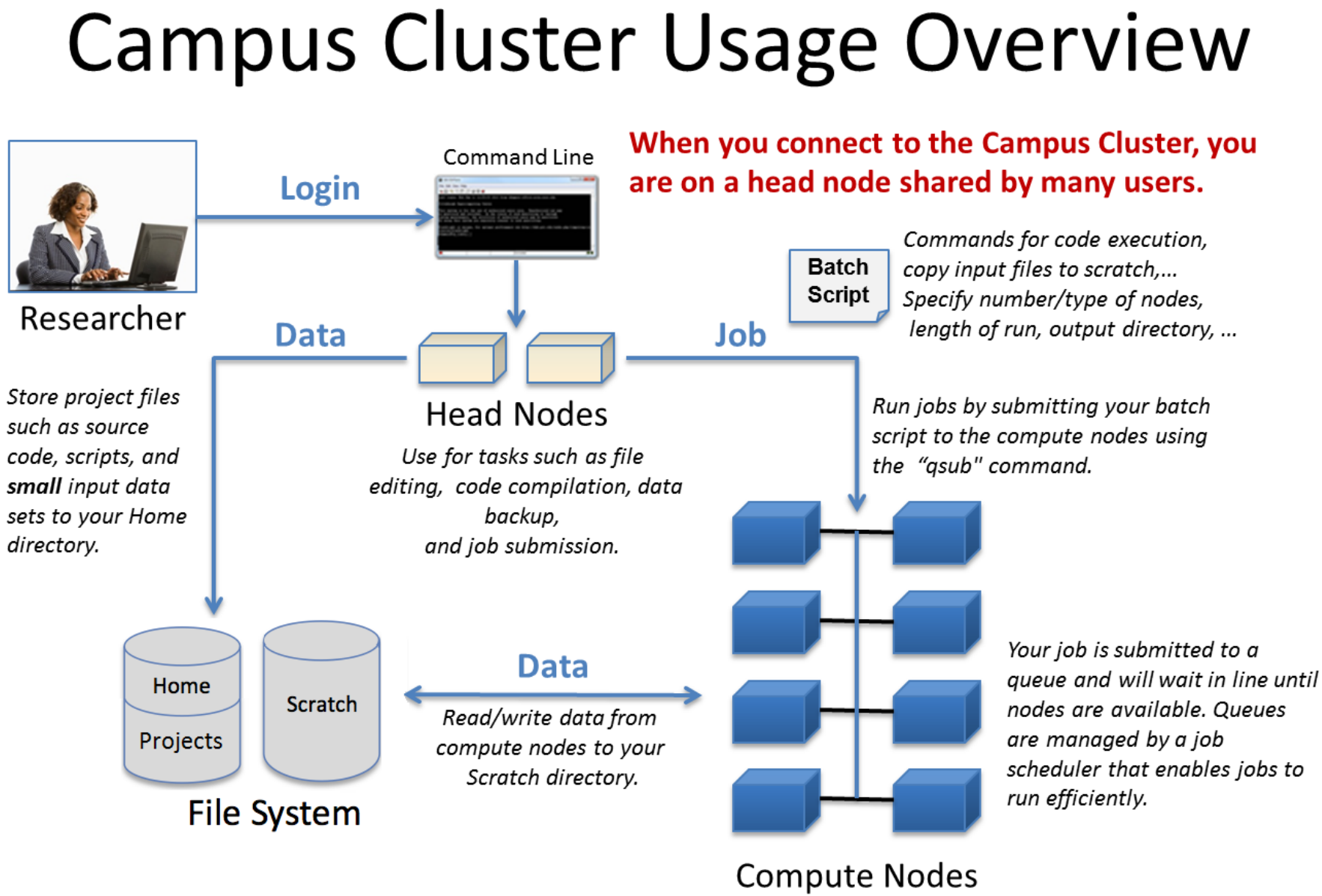
Notre domaine d'étude : Le cluster est un ensemble de moyens de calcul interconnectés qui pourra réaliser des opérations communes (parallèles) en utilisant des logiciels standard et si possible OpenSource.

Sa raison d'etre : Dépasser les limites du matériel existant en mutualisant des ressources (exécution parallélisée d'instructions, agrégation de capacité mémoire et de disque pour distribuer des modèles de donnée massifs).

Quels services pour quels utilisateurs ?

1. Une utilisation simple.
   => L'utilisateur doit pouvoir utiliser le cluster le plus facilement possible,
   => l'administrateur système et réseau aussi ....
2. Une efficacité maximale.
3. Des traitements plus rapides et mieux réalisés.
4. Des unités de traitement que l'on mutualise.

Le cas d'école.



Cas d'utilisations : utilisateur

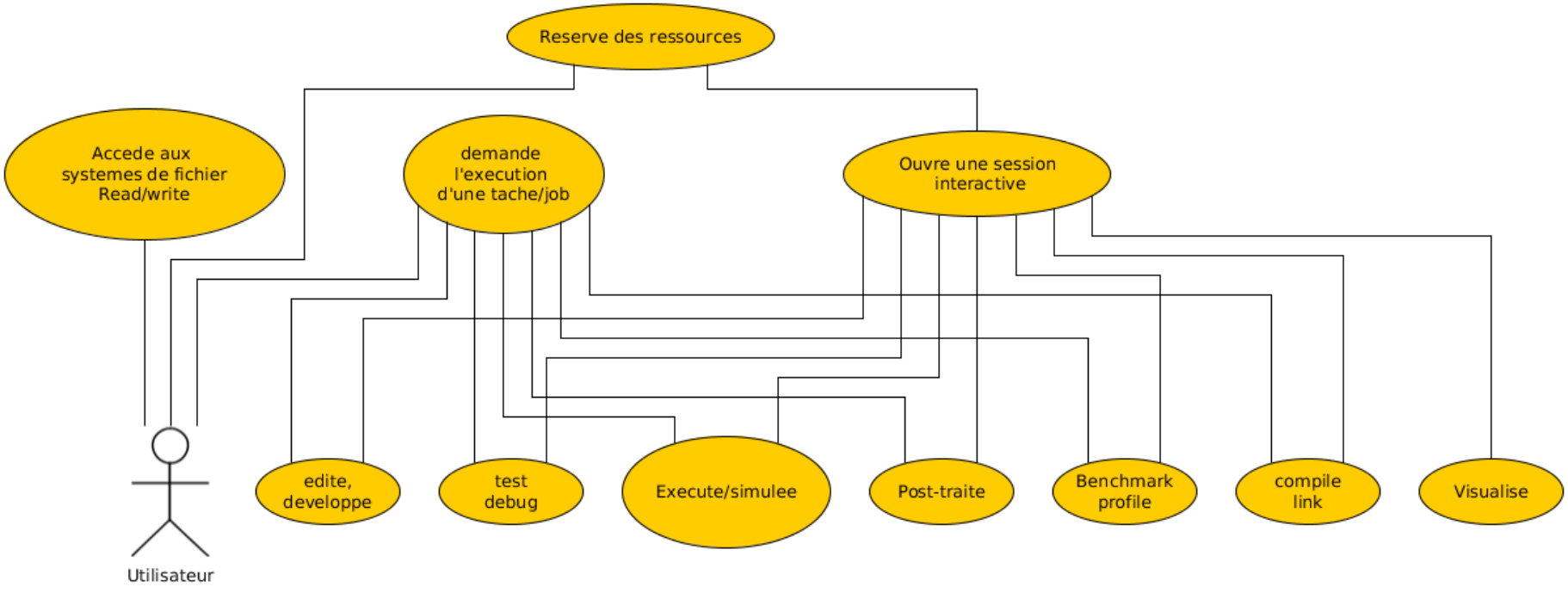


Figure: Use case utilisateur

Cas d'utilisations : administrateur.

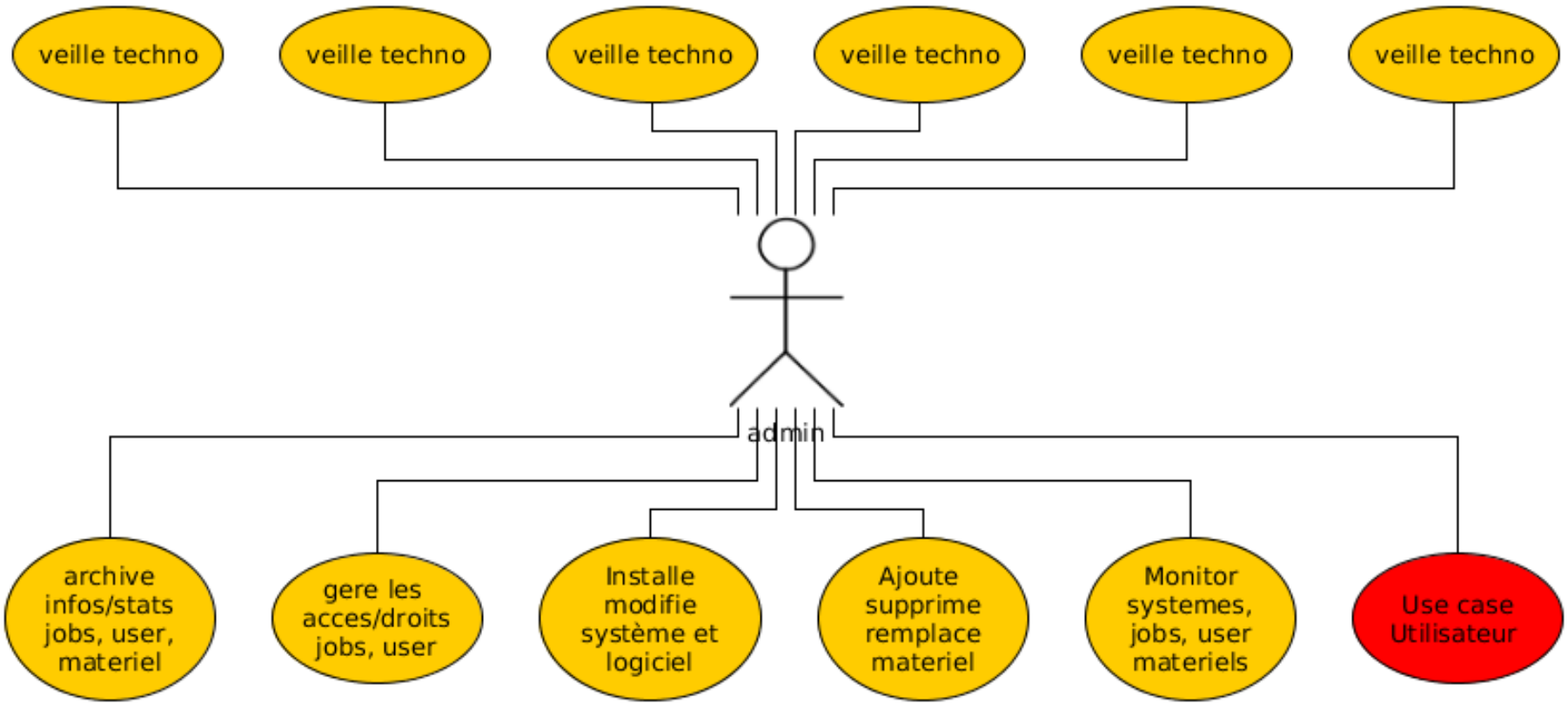


Figure: Use case Administrateur

L'approche cluster pour le calcul numérique

Les composants indispensables :

- Des systèmes de fichier partagés.
- Une gestion des taches/jobs/processus utilisateurs.
- Un environnement de travail cohérent.
- Des réseaux d'interconnexion.

Types de Nœuds/serveurs.

Les éléments physiques :

- Nœuds de login : Accès/login/transfert de fichier/passerelle X11/édition de fichier, réservation de ressources et soumission de taches/jobs.
- Nœuds de calcul : Compilation/debub/dev , exécution des traitements en fonction du gestionnaire de tache (batch manager, ressource manager).
- Nœuds data : Systèmes de fichiers réseau. Pas d'accès utilisateur.
- Nœuds de support/management : Dédiés à la gestion du cluster(batch manager, scheduling, accounting, monitoring, authentification, logging, ...). Pas d'accès utilisateur.

Besoins standards pour réseau de serveurs.

- Système d'exploitation gérant correctement le matériel !
- Services réseaux : NTP, DNS, DHCP.
- Systèmes de fichiers réseau/partagé : NFS, Lustre, PGFS, ...
- Service de logs : centralisation des logs, analyse.
- Service de monitoring des ressources.
- Authentification, gestion des comptes utilisateurs : LDAP, AD, NIS, fichiers.
- Base d'administration pour la gestion des données/informations utilisateurs, calculs et ressources.
- Services de gestion du cluster : "Boot/Provisioning", PXE, suites logiciels d'installation d'OS.

Besoins spécifiques aux clusters de calcul.

- Un réseau d'interconnexion entre nœud de calcul, donnée/fichiers. (idéalement de faible latence et de bande passante élevée)
- Un "ressource/job manager" pour la gestion des ressources de calcul partagée et des taches/calcul des utilisateurs.
- Un système de fichier parallèle (haute performance IOPs).
- Un service d'accès à distance et d'affichage déporté.

Le système d'exploitation.

A priori, une distribution Linux répondant à vos multiples critères :

- Gérant correctement le matériel ciblé (instruction processeur, mémoire, disques, réseau, carte extension calcul comme mic/GPGPU ou visualisation GPU).
- Compatibles avec les contraintes d'utilisation des applications (version de librairie, contrainte support/licence).
- Maintenue/supportée sur la durée de votre projet.
- Compatible avec les outils/logiciels/services utilisés en exploitation. Toutes les distributions modernes proposent une suite complète de développement et un environnement riche en calcul scientifique.
- ADAPTER A VOTRE EXPERTISE ! (RHEL, Centos, Debian, Ubuntu, OpenSuse, etc)

Le système d'exploitation sera installé sur chacun des nœuds avec les services. L'installation peut être :

- Locale. Écriture des fichiers sur l'espace disque local.
- Réseau (DiskLess/StateLess). Fichiers systèmes chargés en mémoire après un boot réseau (PXE/GPXE).

Pile Logicielle HPC.

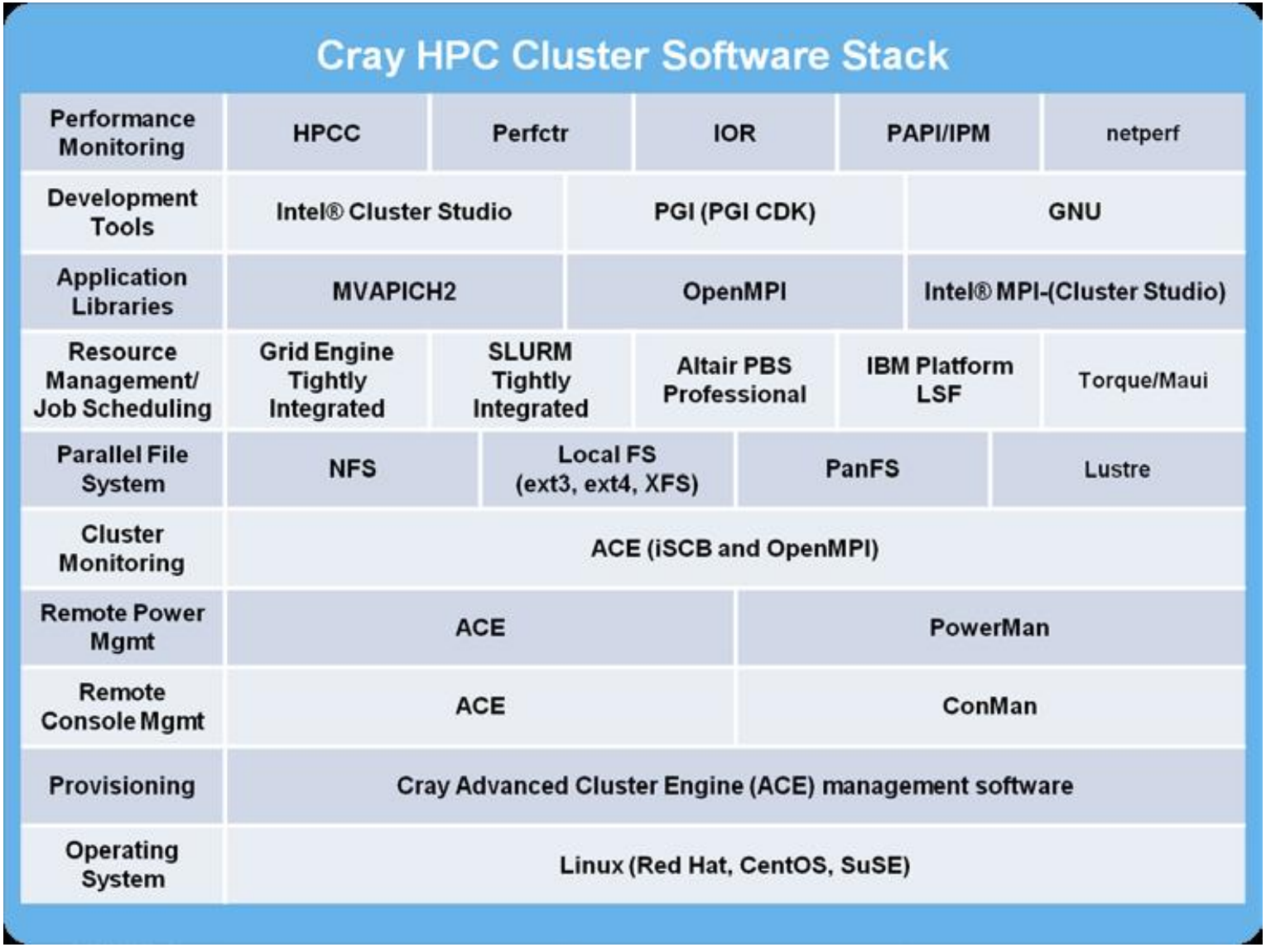


Figure: Cray HPC stack

logiciel pour le calcul numérique

"A dwarf is a pattern of computation and communication. Dwarfs are well defined targets from algorithmic, software, and architecture standpoints."

Application Area	Structured Grids	Unstructured Grids	FFT	Dense Linear Algebra	Sparse Linear Algebra	N-Body	Monte Carlo
Molecular Physics			X	X		X	X
Nanoscale Science	X			X		X	X
Climate	X	X	X		X	X	X
Environment	X	X			X	X	X
Combustion	X			X		X	
Fusion	X	X	X	X	X	X	X
Nuclear Energy		X		X	X		
Astrophysics	X	X		X	X	X	
Nuclear Physics				X			
Accelerator Physics		X			X		
QCD	X						X
Aerodynamics	X	X		X	X		

CENTER FOR COMPUTATIONAL RESEARCH
University at Buffalo The State University of New York

Figure: Phillip Colella's Seven Dwarfs

Nœud de login

- NTP :

Nœud de développement/compilation/debug/calcul/simulation

- NTP

Nœud de visualisation

- NTP :

Les services réseaux

- NTP : Network Time Protocol.
  Synchronise les horloges des nœuds. Indispensable pour les nœuds, les systèmes de fichier et tous les services distribués.
- DHCP : Dynamic Host Configuration Protocol.
  Centralise la configuration réseau des nœuds, ainsi que les informations d'installation par réseau (PXE). Peut être remplacer sur de petite configuration par les fichiers systèmes ()/etc/sysconfig/network-script/ifcfg-xxx - attention au mise à jour).
- DNS : Domaine Name service.
  Centralise la configuration des noms de machines et les domaines réseau. Peut être remplacer sur de petite config par les fichiers systèmes (/etc/hosts, /etc/resolv.conf - attention au mise à jour).
- Système de fichier distribué standard pour le partage de l'espace donnée sur tous les nœuds (user,softs, scratchs). Il peut y en avoir plusieurs en fonction des besoins (Volumétrie, robustesse, vitesse/perf, HA, cout).

Logging et monitoring

- Syslog/systemd : logs systems.
- Rsyslog, logstach, eleasticSearch : centralisation sur le réseau des logs
- Monitoring : opération active pour récupérer les informations logs que le système ne gerent pas.
  Par exemple les informations issues du ressource manager (Nagios/cacti/zabbix/ganglia), des metriques materiels.
- IDS : outil de détection d'intrusion pour traiter les mauvais comportements ou les attaques sur le cluster.

Autres services de base.

- Gestionnaire de licence : l'offre commercial HPC se developpe (FlexNet, FlexLM).
- Base d'information : utile pour les besoins d'administration, la gestion de configuration et nécessaire pour gérer les utilisateurs, les jobs, les matériels, les statistiques, ...
- Installation/Boot/Provisioning : Élément indispensable des outils de gestion de cluster (ex : PXE/Cobbler, FAI, suite logicielle de gestion cluster comme Rocks, XCAT, SSI, onesys, ...)

Authentification.

- Fichiers systèmes (passwd, groups, shadow).
- NIS : accès réseau à un service gérant les fichiers spécifiant les comptes utilisateurs.
- LDAP : structure arborescente dynamique représentant les comptes et les informations des utilisateurs, des groupes, etc (BD berkleys).

L'environnement réseau.

- Réseaux pour la gestion du Hardware - INDISPENSABLE pour plus d'une dizaine de serveur (ILOM, BMC, IPMI, IDRAC, ...).
- Réseaux externe : interface publique pour l'accès au cluster.
- Réseaux interne/privée : connexion pour les échanges entre TOUS les nœuds du cluster.
- Réseaux de stockage : accès aux systèmes de fichier.
- Réseaux Interconnect : Haute Bande Passante, latence basse pour les échanges entre les nœuds de calcul (calcul parallèle MPI).

Ces réseaux peuvent partager les mêmes "medium" en fonction des budgets et besoins.

L'interconnect.

Quelques statistiques (TOP 500 - www.top500.org) : 237 cluster Infiniband, 119 cluster 10G, 62 cluster 1G, 74 cluster "custom" .

- Fujitsu TOFU inteconnect 2 - topologie tore, latence 0,71μs et bande passante 100G/s.
- Cray gemini - topologie tore.
- Intel Omnipath (2017 ?) latence 0,2μs et bande passante 100G/s.
- Ethernet : latence de 050-125 μs (GbE), 5-50 μs (10GbE), 5 μs RoCEE.
- Infiniband : latence 1,3 μs et 40G/s de bande passante (QDR), latence 0,7 μs et 50G/s de bande passante (FDR/FDR-10), latence 0,5 μs et 100G/s de bande passante (FDR/FDR-10).

Topologie réseau et performance 1.

La nature du réseaux et sa topologie ont un impact important sur les performances des calcul parallèle (MPI), son l'évolutivité et l'homogénéité.

- Réseau en étoile/maille, tree, fat Tree (ex : switchs ethernet 1G/10G).
- Réseau de clos (ex Infiniband).
- Tore 3D et HyperCube (ex Infiniband).

Un réseau de clos permet d'interconnecter des réseaux en étoile en contrôlant la perte des performances des caractéristiques des communications point à point (http ://clusterdesign.org/cgi-bin/network/network).

Topologie réseau et performance 2.

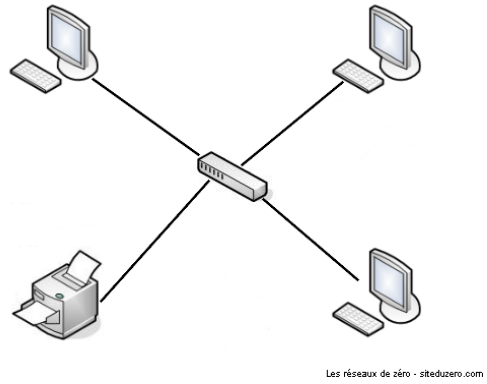


Figure: Réseau en étoile

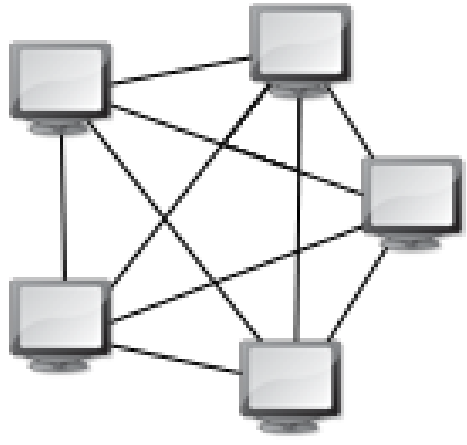


Figure: Réseau maillé

Topologie réseau et performance 3.

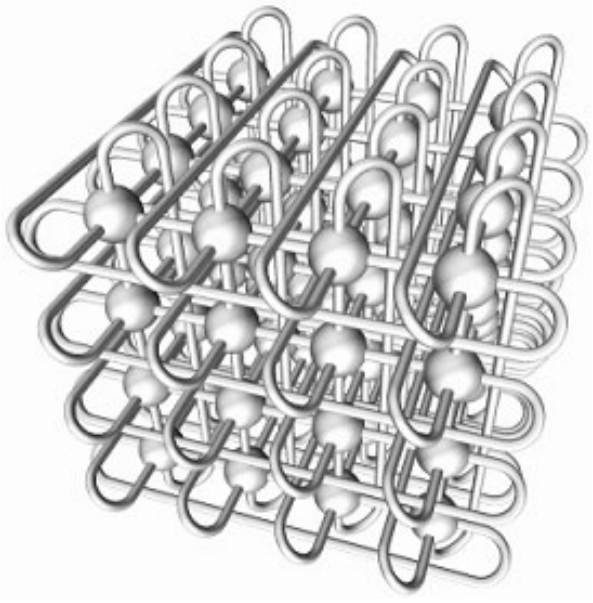


Figure: Tore 3D Fujitsu

Figure: Reseau de CLOS

Topologie réseau et performance 4.

Pour un réseau Infiniband (switch 36 port) :

- Réseau de 60 nœuds rapport 1 :1 -> 132 câbles, switchs 2/core, 4/edges, ports 18/ups, 18/nodes.
- Réseau de 60 nœuds rapport 2 :1 -> 96 câbles, switchs 1/core, 3/edges, ports 12/ups, 24/nodes.
- Réseau de 60 nœuds rapport 4 :1 -> 84 câbles, switchs 1/core, 3/edges, ports 8/ups, 28/nodes.
- Réseau de 400 nœuds, évolution possible à 500, rapport 2 :1 -> 600 cables, switchs 12/core, ports 17/edges, ports 12/ups, 24/nodes.
- Réseau de 400 nœuds, évolution possible à 500, rapport 4 :1 -> 520 câbles, switchs 4/core, ports 15/edges, 8/ups, 28/nodes.

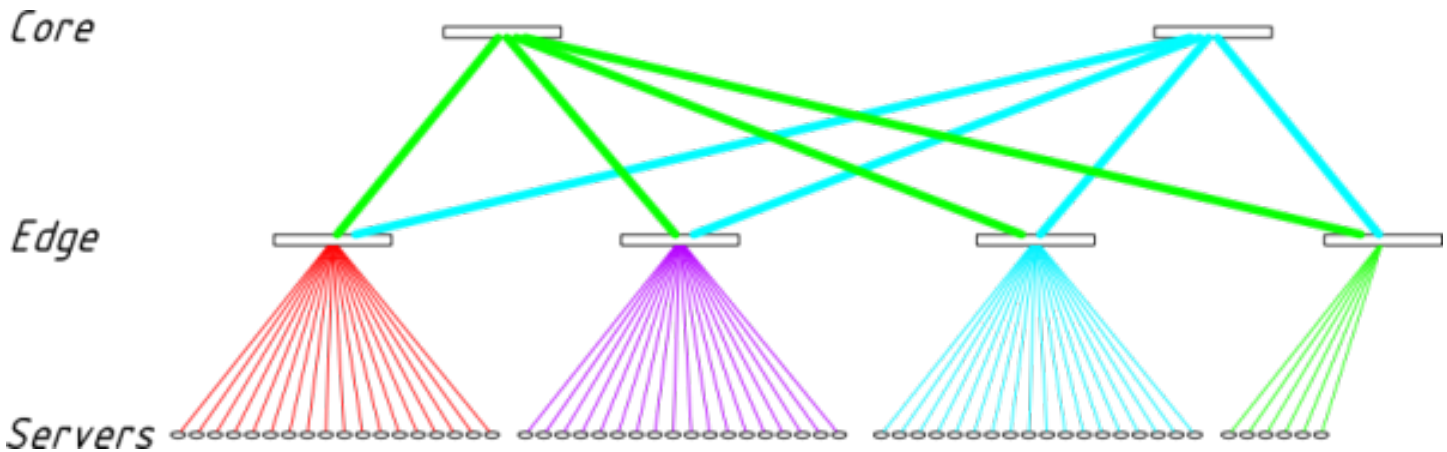


Figure: reseau clos 60 noeuds - clusterDesign.org

Système de fichier parallèle.

Système de fichier parallel Il distribue les écritures/lectures simultanées sur des disques/serveurs à travers le réseau. Les métadonnées du système de fichier peuvent être séparées des fichiers (nœuds dédiés pour les meta-données). Le réseau utilisé est de préférence un Interconnect très rapide.

- Lustre : performant, très complexe, non intégré au noyau Linux, maintenance difficile coté serveur.
- PanFS : "Appliance", bonne performance, cout très élevé, évolutif.
- GPFS : performant, très complexe, non intégré au noyau Linux, maintenance difficile, cout très élevé.
- RozoFS : performant sur les petits fichiers, non intégré au noyau Linux, mature ?
- GlusterFS : performant, complexe, intégré au noyau Linux.
- Ceph : robuste, intégré au noyau Linux.

Exemple d'infrastructure.

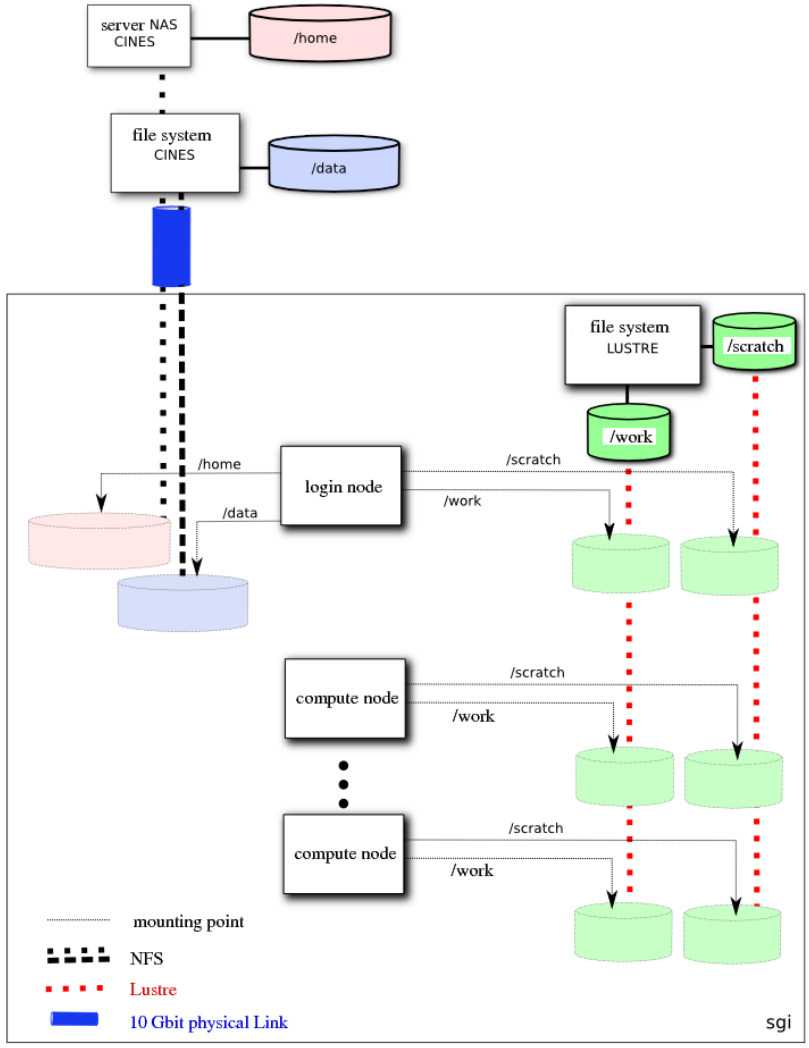


Figure: Lustre au CINES

Administration/gestion du cluster

L'administration d'un Cluster est proche de la gestion d'un groupe de Pc/serveur (Linux !)

- Automatiser le construction d'un cluster, d'un nœud.
- Maintenir la cohérence des systèmes/configurations machines.
- Automatiser les taches de maintenance.
- Monitorer l'état des nœuds/cluster et leur performance.

Solution de Gestion de cluster.

Il existe des environnements/suites logicielles qui prennent en charge l'administration d'un cluster. Elles sont liées aux distributions Linux. Le "ressource manager", l'"ordonnanceur" et l'outil de "provisioning" forme la base de gestion du cluster !

Le ressource manager connait l'état de toutes les ressources du cluster, gère le cycle des applications/processus exécutés par les utilisateurs et maintient une liste des applications qui demandent/utilisent celles-ci.

L'ordonnanceur utilise les informations du ressource manager et sélectionne les applications en liste d'attente pour exécution.

Exemple de solutions de "cluster managment" :

- Rocks
- Cobbler, xCAT (dépendances matériels)
  xCAT 2.11 dec 2015 RHEL 7.2 Ubuntu 14/15.
- Plateform HPC, Bright Cluster Management.
- OpenStack
- warewulf
- "Do-it-yourself" ?

Cas particulier d'un distribution Cluster - Rocks - 1

- Une distribution "cluster" avec un processus d'installation/gestion automatisé.
- Basé sur une RHEL.
- Une Image ISO et des dépôts dédiés.
- Packaging de tous les softs !
- Compatible RHEL/Centos.
- Gestion des packages standard (distribution), logiciels propres à ROCKS, logiciels supplémentaires provenant de la communauté ROCKS.
- Gestion de la configuration du système et des services du cluster.
- Regroupement "ROLLS" des logiciels distribués/installés par services/fonctions.

Cas particulier d'un distribution Cluster - Rocks - 2

Parmi les "Rolls" optionnels :

- Condor.
- Grid .
- Intel(compilers).
- Java.
- SCE.
- Sun Grid Engine.
- PBS.
- Area51 (security monitoring tools).
- hpc.
- ganglia.
- kvm.
- python.
- slurm.

Cas particulier d'un distribution Cluster - Rocks - 3

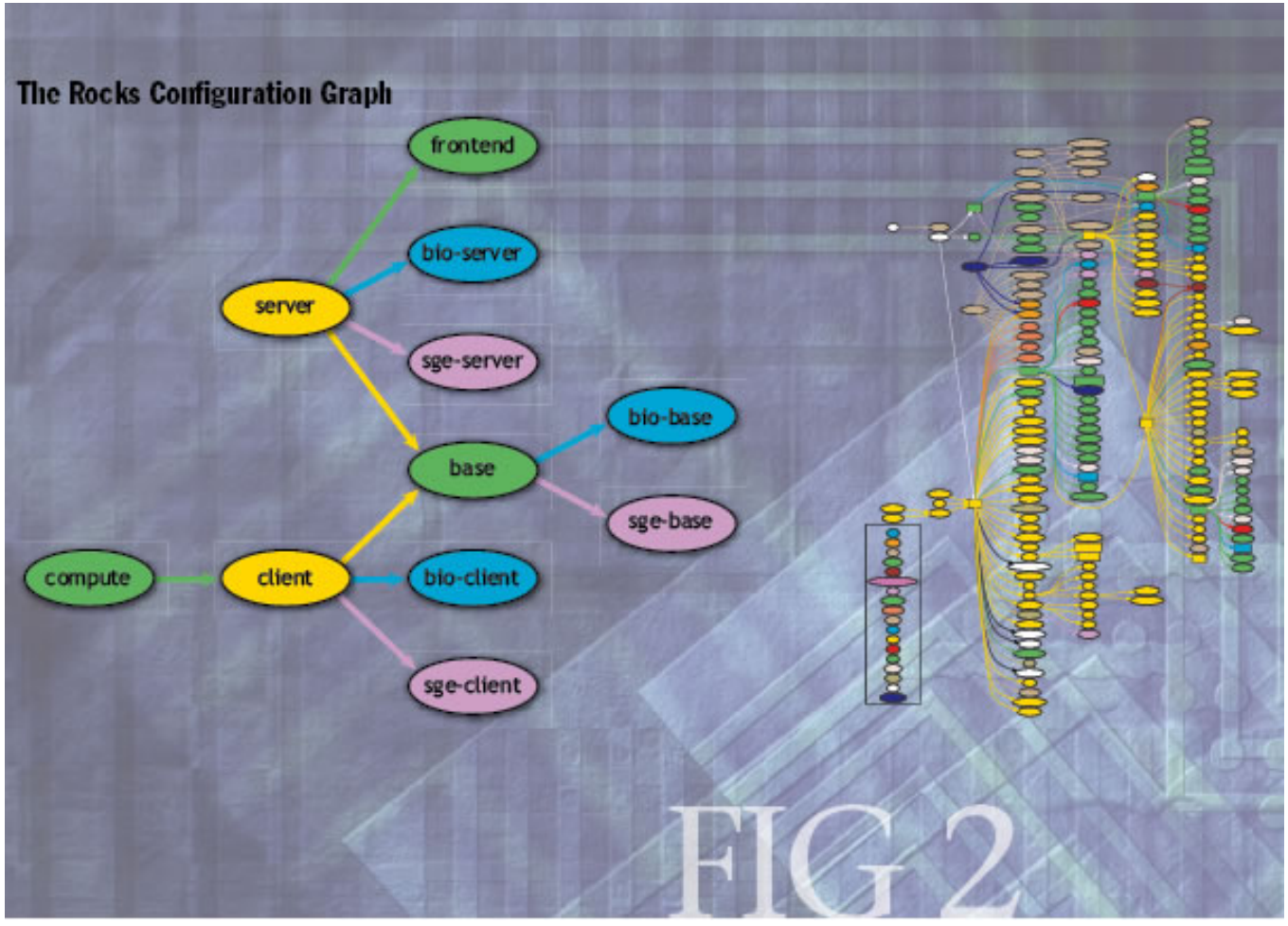


Figure: Rolls graphe

Cas particulier d'un distribution Cluster - Rocks - 3

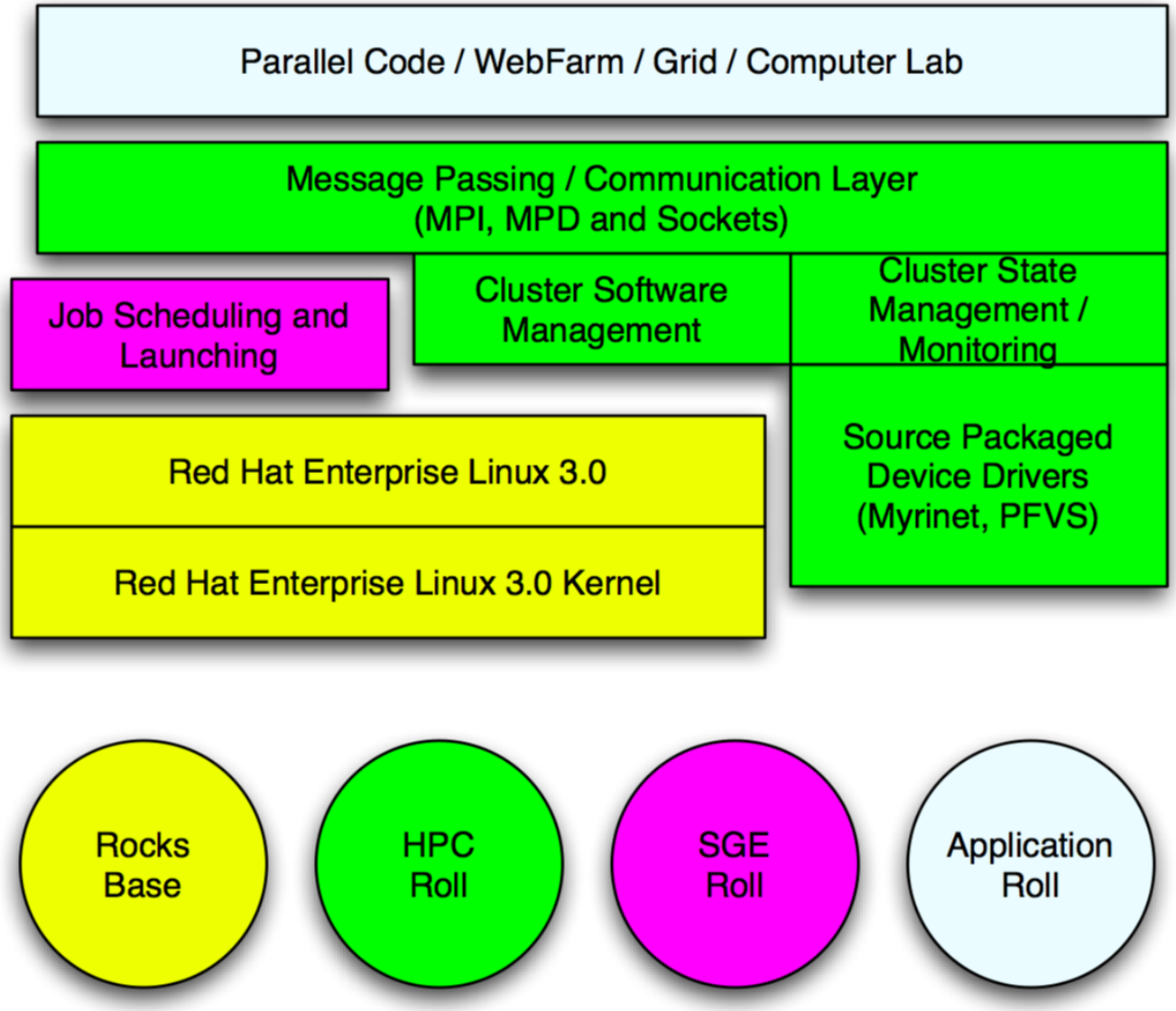


Figure: Pile/Rolls ROCKS

Le ressource/batch manager 1.

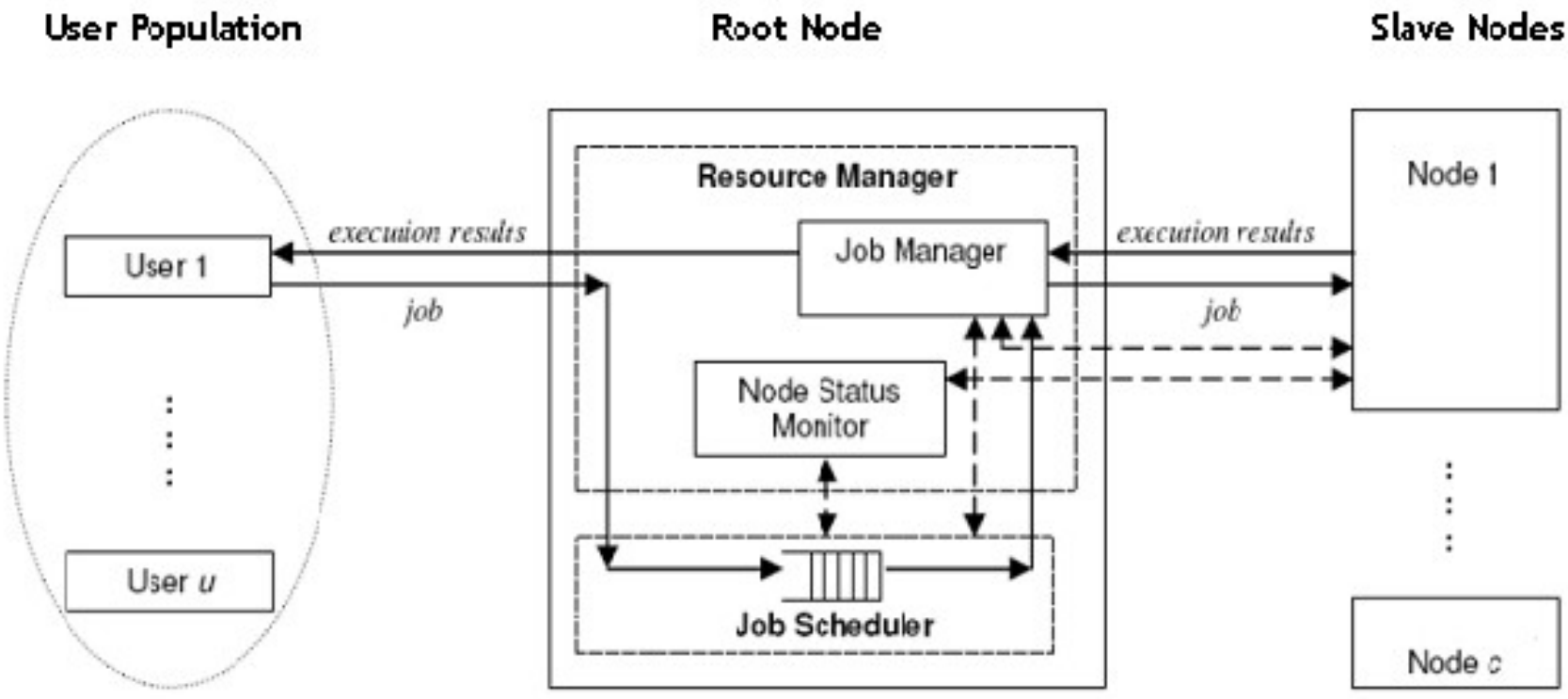


Figure:

Le ressource/batch manager 2.

Comment fonctionne le ressource/batch manager.

- En attente de soumissions de job ou de réservation de ressources de la part des utilisateurs.
- Alloue les serveurs/ressources aux utilisateurs demandeurs et démarre les applications sur les nœuds de calcul libres.
- Ordonnance les applications en attente d'exécution dans des files d'attente en fonction de divers algorithmes (backfill, faireshair, FIFO, priority).

Plusieurs solutions :

- Sun Grid Engine.
- LSF/Openlava.
- PBS (PBSpro/Torque).
- SLURM.
- OAR.

Le ressource/batch manager 2.

Comment fonctionne le ressource/batch manager.

- Tâche (soumission) Interactive (shell) / Batch.
- Tâche séquentielle et parallèle.
- Walltime (temps limite - important pour l'ordonnancement).
- Accès exéclusif / non-exclusif aux ressources.
- Appariement de ressources.
- Scripts Epilogue/Prologue (exécuter avant/après les tâches).
- Suivi (monitoring des tâches (consommation des ressources).
- Dépendance entre tâches (workflow).
- Logging et accounting.
- Suspension/reprise des tâches.

Le batch manager - gestion de jobs MPI.

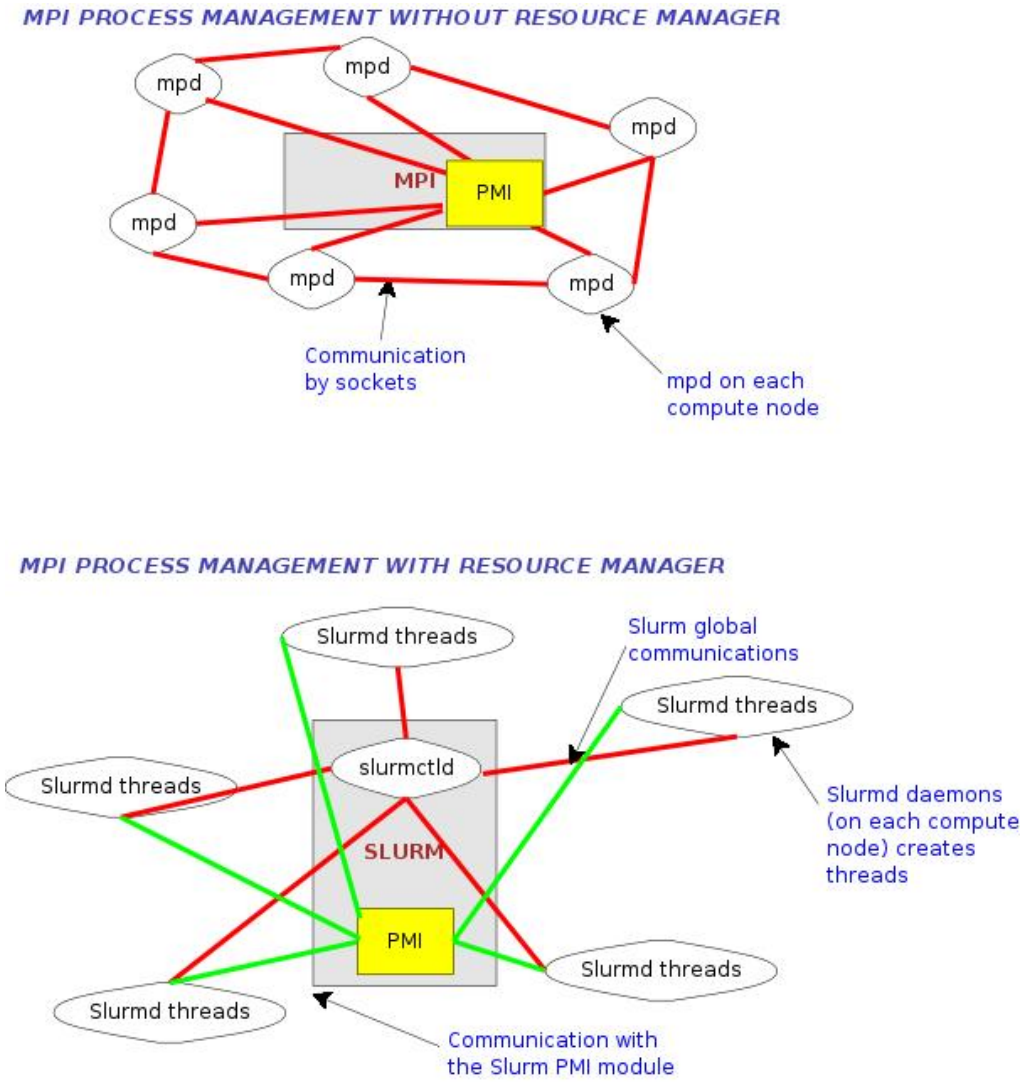


Figure: Exemple de gestion des processus MPI avec et sans batch manager.

Le batch manager - Exemple complet de soumission d'une tache Matlab.

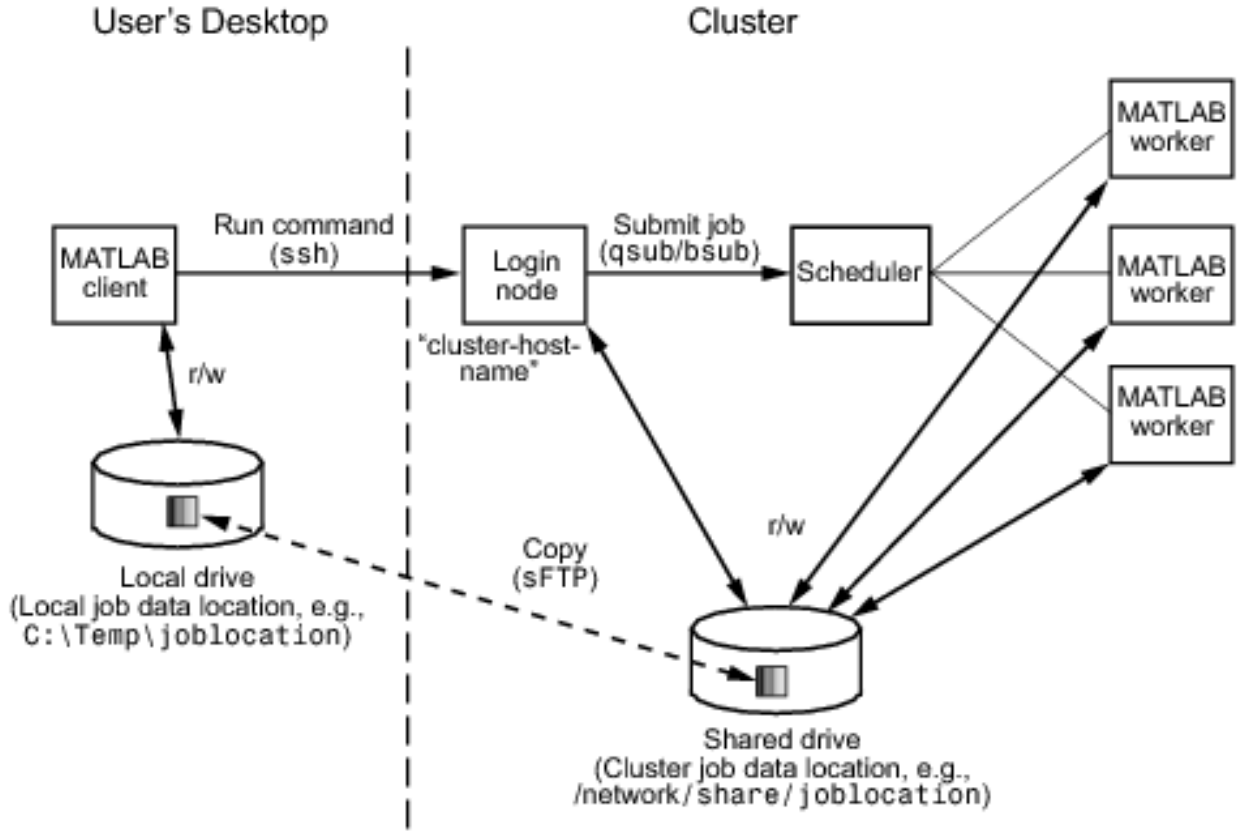
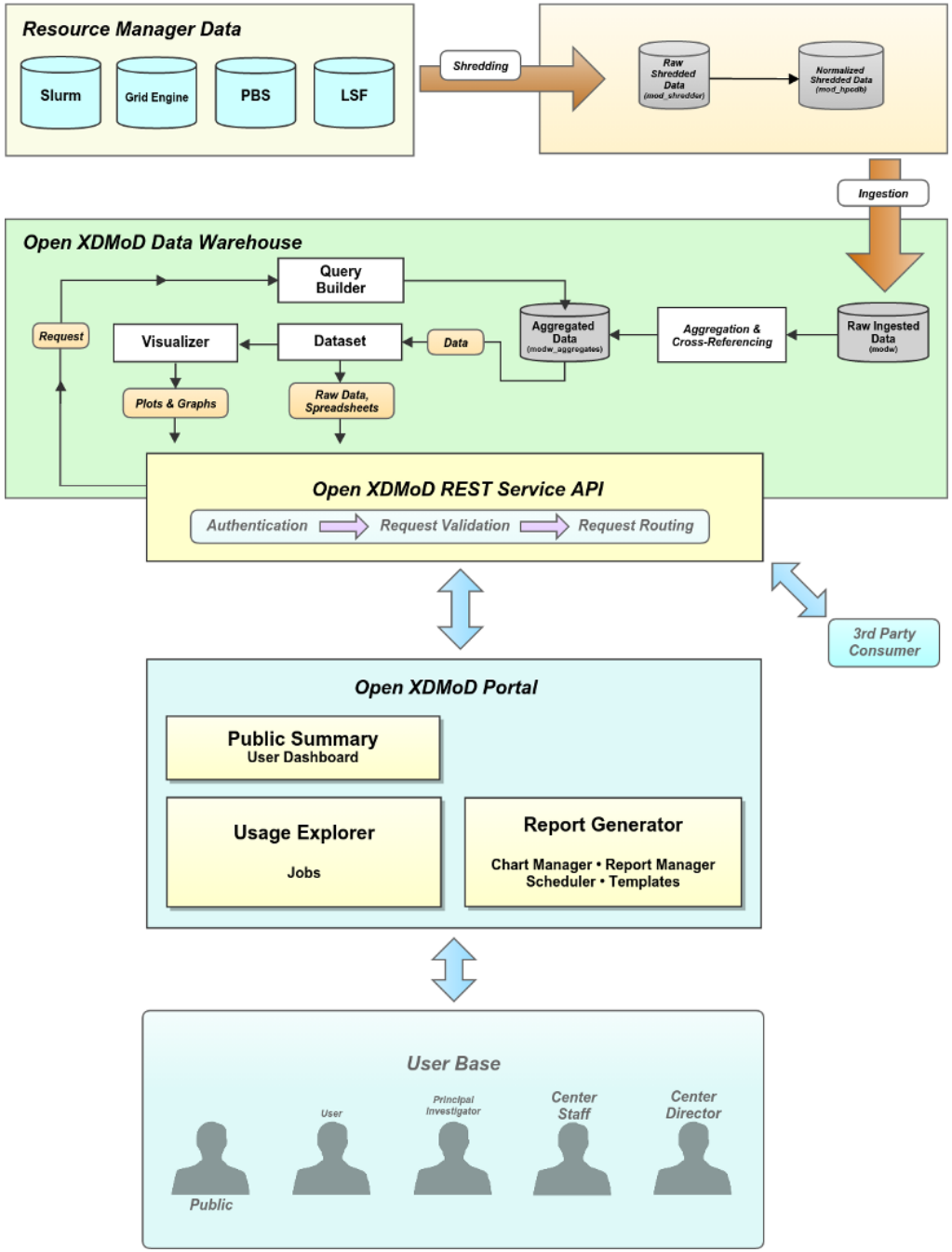


Figure: Exemple de gestion d'une soumission d'un job Matlab.

Accounting.

- Gestion des statistiques d'allocations et d'utilisation des ressources.
- Propose des metrics pour mesurer l'utilisation des ressources.

Accounting exemple SlurmDBD/XDMoD



Accounting exemple SlurmDBD/XDMoD

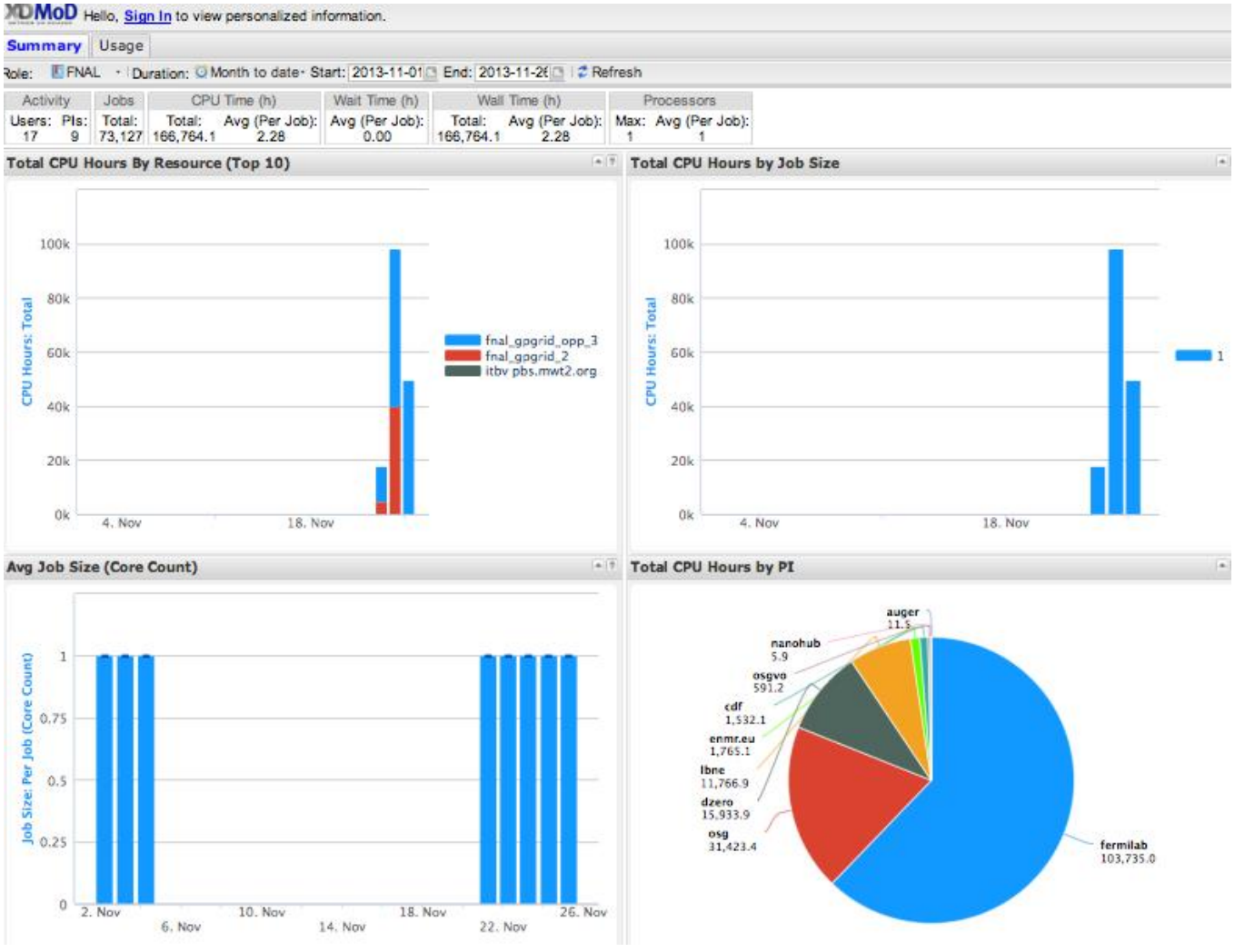


Figure: Outil d'accounting - XDMoD

Gestion de configuration.

La gestion de configuration permet la maintenance des nœuds en gardant leur cohérence. Même si l'ensemble des nœuds sont "diskless", il reste toujours des configurations et "datas" persistantes/dynamiques à gérer sur différents types de nœuds. Il existe plusieurs solutions libres efficaces.

- ssh, pdsh, clustershell, func.
- Rsync.
- Cfengine.
- Chef.
- Puppet.
- Salt.
- Rudder.
- Ansible.

Exemple ANSIBLE - 1

L'objectif de ce logiciel est de maintenir une configuration identique sur différents serveurs/machines virtuelles. Concrètement, Ansible permet :

- Installation des paquets.
- Installation des fichiers de configuration.
- Configuration des services.
- Réalisation de toutes les tâches d'administration possible par ssh.
- Couplage avec un outil de versioning pour, par exemple, suivre les évolutions du cluster.
- Notion de playbooks qui définit une ensemble de règles/scripts à appliquer/valider.

Exemple ANSIBLE - 2.

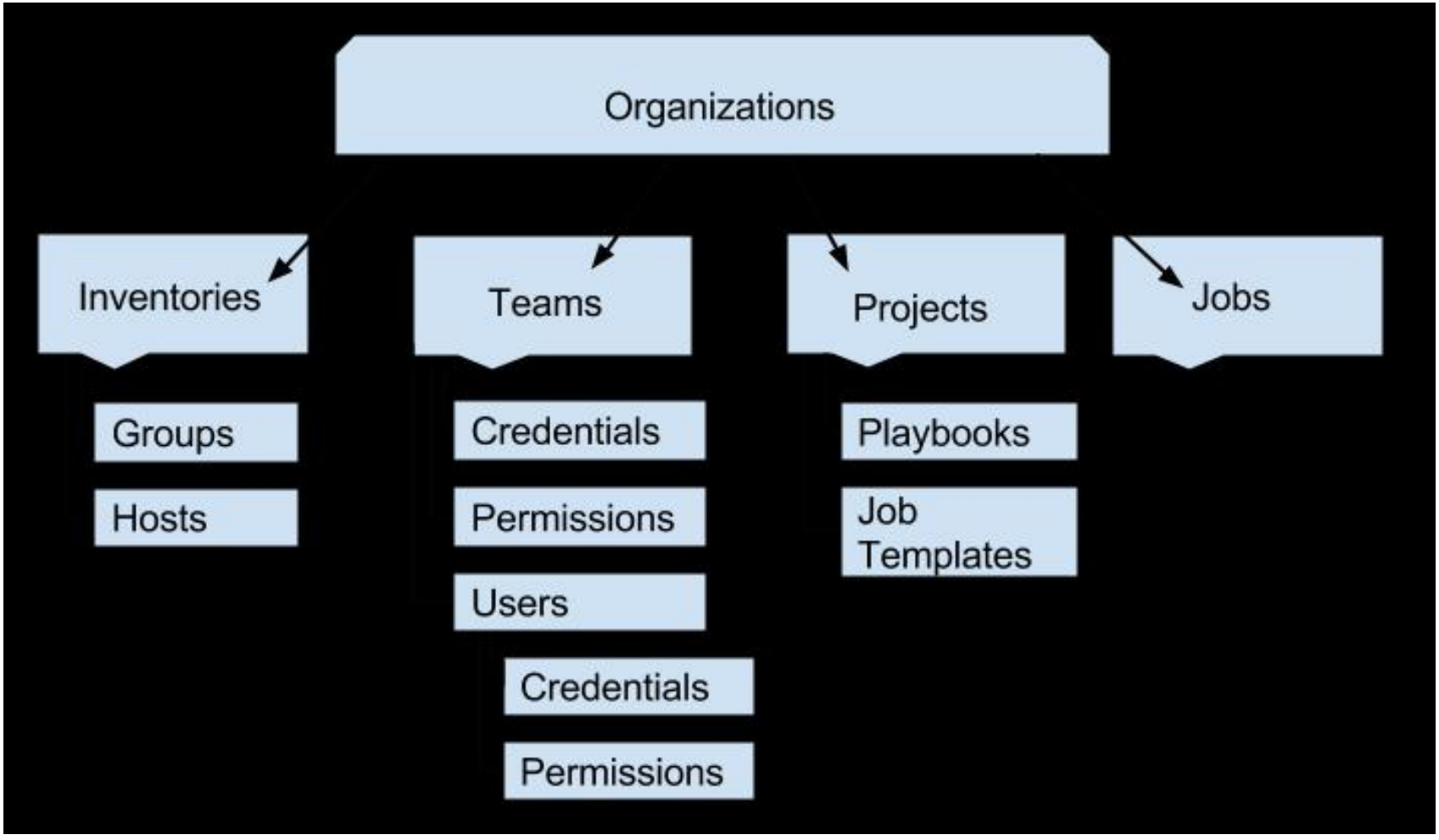


Figure: Structure des données - Ansible.

Exemple ANSIBLE - 3.

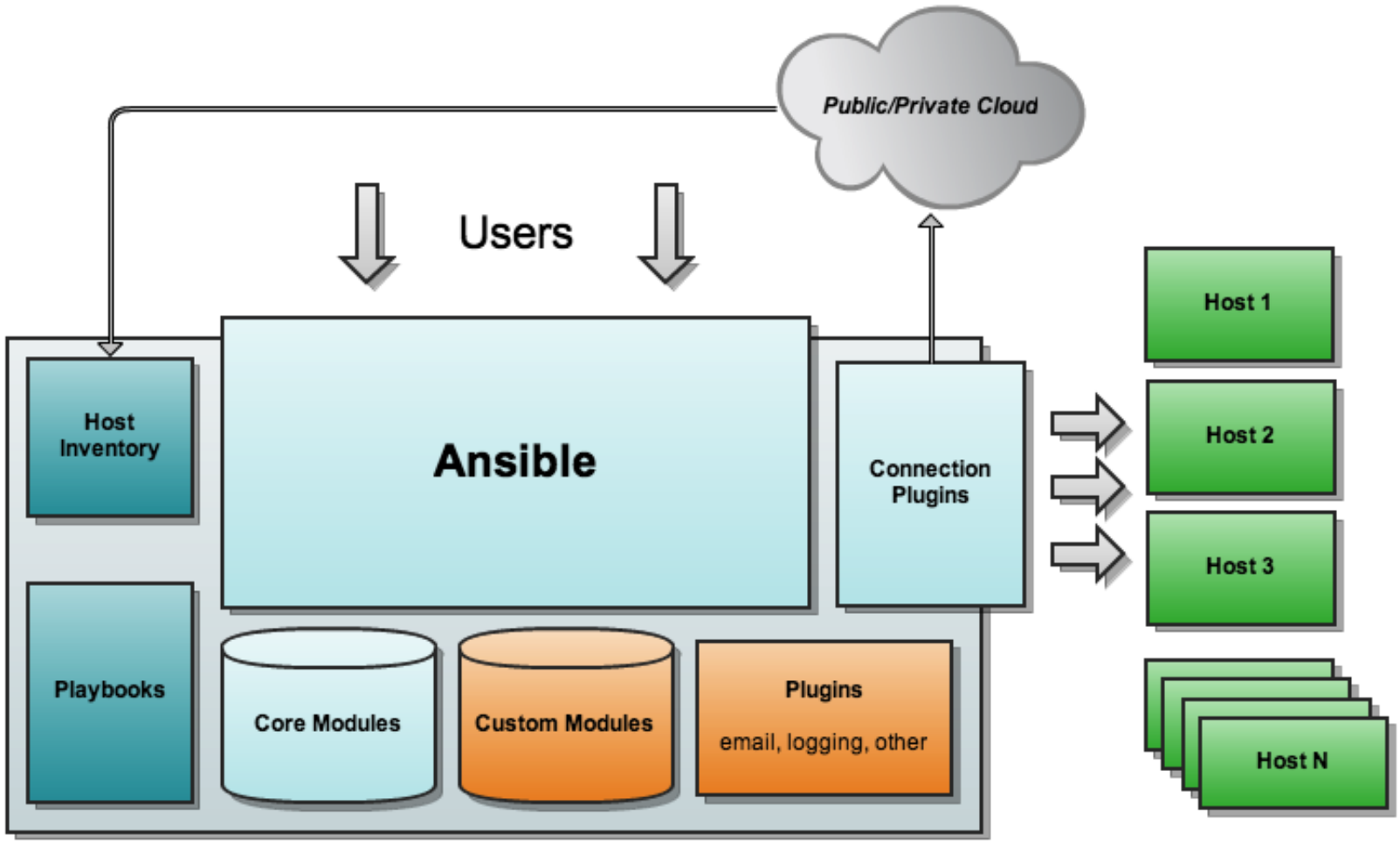


Figure: Fonctionnement - Ansible.

Installation et gestion des logiciels - 1.

Chaque distribution Linux propose ces outils de gestion "softwares".

- RedHat-like rpm, yum et dnf.
- Debian-like apt et dpkg

Avantages et inconvénients de ces solutions :

- Version, emplacement d'installation et codage binaire(optimisation -> AVX/SSE ...) imposé/unique.
- Test, Validation et maintenance automatique.
- Vaste offre logiciel mais peut etre insuffisante dans les environnements de recherche.

Une alternative : Do-it-Yoourself / sources.

- Installation sur un système de fichier réseau des applications recompilées (partage NFS ?).
- Limite la taille du système au minimum.
- Autorise la cohabitation de multiples versions optimisés.

Installation et gestion des logiciels - 2.

Pour gérer facilement les multiples versions logiciels dans l'environnement utilisateur, il existe plusieurs outils :

- SoftsEnv pour gérer les environnement statique associé aux packages standard
  (http ://www.lcrc.anl.gov/info/Software/Softenv)
- Modules pour dynamiquement changer l'environnement
  (http ://modules.sourceforge.net/)
  => solution adopté par tous les
- easybuild/easyconfig/easyblock pour gérer les sources, leur compilation et leur environnement d'installation.
  (https ://hpcugent.github.io/easybuild/).
  => projet prometteur.

Sécurité et bonne pratique.

Quelques usages pour garantir le bon fonctionnement et la sécurité d'un cluster :

- Limiter les risques.
- La dissuasion.
- La Prévention.
- La détection.
- La récupération.

Eviter les risques.

- Offrir uniquement les services minimums pour les différents nœuds.
  => nettoyage des services initd/systemd inutiles ! => Ex : pas de serveur X, de démon xinetd, sur un nœud de calcul.
- Offrir les accès/droits minimums.
- Apprenez à aimer les containers !
  => cpuset, memset, cgroups, namespace. =! docker, LXC, virtualisation.
- Installer uniquement les logiciels nécessaires.

L'objectif est de minimiser les vecteur d'attaques.

Dissuasion et prévention.

- Limiter la visibilité du cluster (ports/firewall/docs).
- Définir des règles d'usage et sensibiliser.
- Corriger les failles connus.
- Configuration minimal des services systèmes.
- Restreindre les accès et droits des utilisateurs.
- Documenter, archiver et versionner les interventions/modifications sur les systèmes (DevOPS ?).

Détection et récupération.

- Monitorer le cluster.
- Avoir un contact étroit avec les utilisateurs et encourager les retours après utilisations/problèmes.
- Automatiser les remontées d'alerte et traitement/intervention.
- Sauvegarder (incréments ?).
- Documenter la reprise d'activer après sinistre, interruption, etc.
- Estimer le niveau de perte acceptable (identifier les éléments/données critiques) et sécuriser.

Retour d'expérience.

- Attaque Brute force sur port SSH.
- Service "abort" + segFault application utilisant Lustre = freeze kernel.
- DoS/Flood fréquents.
- Vol d'identité pour demande de création de compte.
- Logiciels commerciaux "intrusif" avec des comportement illégaux ...
- Faille de sécurité dans ILO/BMC et son protocole réseau sécurité. (garder son réseau de management PRIVEE ! ! ! dédié ? ?).
- RAID matériel : /Scratch RAID 6 HS après reconstruction automatique.
- Ralentissement FS XFS après saturation - 97
- Serveur NFS avec un ralentissement ponctuel : Flush mémoire (190 Go) - mauvaise configuration du noyau.
- Temps d'exécution aléatoire MPI : topologie et "bad pining" des processus.
- Package standard MPI sans "binding RDMA".
- Zombie/Deadlock de processus utilisateur Lustre (problème fopen/fortran/timeout lock file ...).
- Compilation trop optimisé -> "instruction illégale sur certains noeuds.

Étapes essentielles, concevoir l'architecture du cluster.

1. Définir les fonctions/services que rendra le cluster.
   - Quelles applications et donc quel parallélisme (thread ? MPI ? CUDA/OPENCL ? MIC), quel profil de mémoire (quantité de mémoire, bisocket, quadrisocet), type d'IO (FS parallel, NFS, scratch local SAS/SSD, ...), type de réseau.
   - Du stockages ? de l'archivage ? des BDs ? intensité IOPs ?
   - Visualisation d'affichage simple, de modèle 2D, 3D complexe ?
   - Criticité des accès, disponibilité (HA ?, redondance, backup incrémental ?).
2. Fixer l'architecture générale du cluster.
3. Choisir le système d'exploitation, les logiciels de gestion cluster et l'environnement numérique de calcul.
4. Évaluer (test/benchmark ?) le matériel et faire son choix !

A vous de jouez !

Étudions et concevons un cluster ensemble. Trois cas d'étude qui devront tenter de respecter le cahier des charges :

- Un cluster simplifié.
- Un cluster complet et pouvant évoluant (matériel/logiciel).
- Un cluster complet.

Un cluster élémentaire.

Notre cahier des charges :

- 4 utilisateurs maximum/5 ans.
- 3 applications potentielles (matlab/python, code maison C/fortran, Logiciel Fluent/ANSYS).
- Codes peu "scalable" - performances intéressantes pour 20/30 processeurs/cœurs.
- Archivage des données indispensables. Pas d'IO intensive.

Un cluster complet et très évolutif.

Notre cahier des charges :

- 200 utilisateurs changeant régulièrement.
- Système, noyau Linux et bibliothèque doivent être "upToDate", optimisé et proche des dernières versions stables !
- Applications multiples (matlab/python, codes maison C/fortran, Paraview, Visit, Lattice-Boltzman sur GPGPU, codes commerciaux).
- IO intensives. Données critiques.
- Transferts de fichier régulier et massif (10To/mois).
- certains codes instables => zombis, lock des IOs ouvert, locks sur des secteurs mémoires.
- certains codes uniquement multithread et nécessitant de la mémoire (> 2To).
- Profil des utilisateurs très varié (expert/débutant).
- Budget annuel pour accroitre de 50% la capacité de calcul.

Une cluster complet et simple à déployer.

- Nombres d'utilisateur modérés.
- Multiples applications ayant des besoins/Dépendances standards.
- Indisponibilité minimum 24h !
- Pas de contrainte sur le système de fichier autres qu'une volumétrie > 200 To, un backup (ancienneté des sauvegardes < 24h).

Cluster : Installation.

Prerequis - Avant d'installer/deployer.

1. Définitions des applications/users/besoins.
2. Étude de marché des matériels existant - veille technologique.
3. Benchmarks.
4. Une salle correctement dimensionnée (refroidissement, puissance électrique, charge au sol)
5. Des serveurs (un appel d'offre réussi ?)
6. Réception du matériel.
7. Installation physique et raccordement des serveurs.
8. Finalement, on câble, on met sous tension .... => c'est beau, mais tout reste à faire !
9. Penser à la mise à jour des BIOS/UEFI !
   Possible via un boot réseau + micro image.

Vocabulaire.

- Provisioning : Au sens large, le provisioning est l'affectation plus ou moins automatisée de ressources à un utilisateur.
  Pour un cluster, affectation réseau et installation/configuration d'un nœud du cluster.
- Baremetal : serveur nu, matériel sans système/logiciel installé.
- Bootstrap : en système, un bootstrap est un petit programme d'amorçage qui permet d'en lancer un plus gros (par exemple, Grub/OS).

Provisioning : Méthode et scénarios.

Des méthodes d'installation :

- Clone : ghost, dd, clonezilla (problème : une nouvelle image pour chaque materiel)
- Install : jumstart, kickstart, preseed, installateur natif OS/distrib.
- Clone+Install

Des scénarios différents :

- diskfull : HD/SSD, iSCI/SAN disk.
- diskless : RAM disk.

Provisioning : technologie clé.

- PXE : "Pre-boot eXecution Environment" permet à une station de travail de démarrer depuis le réseau en récupérant une image de système d'exploitation qui se trouve sur un serveur.
- NBP : "Network Bootstrap Program" est le programme de boot telechargé par réseaux equivalent à Grub/Lilo.
- IPMI : "Intelligent Platform Management" Interface est une interface système standardisée pour gérer le matériel.
- DHCP, TFTP, iSCSI.

Cycle d'installation Kickstart.

1. Boot et chargement initrd (Image Système d'installation).
2. Exécution de l'installateur Anaconda.
3. Démarrage du wizard "graphique ou exécution du script "kickstart" décrivant l'installation.

Fichier kickstart.

```
1  ##################################################################
   # This is an installation not an upgrade
3  install
   # The location of the RPM files
5  url --url http://emstools2b.cisco.com/pub/rhel/server
   key 9a09007d99b6cd00
7  lang en_US
   # Use text mode install
9  text
   keyboard us
11 xconfig --defaultdesktop kde --resolution 640x480 --depth 8
   network --device eth0 --bootproto dhcp --onboot=on
```

Fichier kickstart.

```
rootpw --iscrypted $1$tihTg7ne$hohhkj87hGGddg9B4WkXV1
authconfig --useshadow --enablemd5
selinux --disabled
timezone America/New_York
firewall --disabled
firstboot --disable
# Reboot after installation
reboot
bootloader --location=mbr --append="console=ttyS0,9600n8"
clearpart --all --initlabel
# define partitions
part /boot --fstype ext3 --size=512
part / --fstype ext3 --size=10000 --grow
part /tmp --fstype ext3 --size=7500
part /var --fstype ext3 --size=7500
part /home --fstype ext3 --size=2500
part swap --size=2048

##################################################################
```

Fichier kickstart.

```
1  ##################################################################
   %packages
3  @engineering-and-scientific
   @mysql
5  @development-libs
   @editors
7  @system-tools
   @gnome-software-development
9  @text-internet
   @gnome-desktop
11 @core
   @base
13 @ftp-server
   @network-server
15 .....
   @admin-tools
17 @development-tools
   @graphical-internet
19 -sysreport
   mc
21 festival
   -compiz-kde
23 -knetworkmanager
   -amarok
25 ##################################################################
```

Fichier kickstart.

```
1  ##################################################################
   %post
3  # Install the yum repository configuration files
   cd /tmp
5  wget http://emstools2b.cisco.com/pub/local/lab-repos.tar
   cd /
7  tar -xvf /tmp/lab-repos.tar
   # Set an ID to be used for other scripts
9  touch /LINUX_RHEL_MINIMAL_INSTALL
   # Install Kshell as a preference of some developers.
11 yum -y install ksh
   # Configure some local NFS mount points
13 service portmap start
   mount  emsnfs:/export/linux/post   /mnt
15 cat /mnt/auto_localnfs >> /etc/auto.misc
   cat /mnt/auto_misc >> /etc/auto.misc
17 # Get the command to create the motd and create it for the first time.
   cp /mnt/createMOTDLinux /etc/init.d/create_motd
19 mv /etc/motd /etc/motd.orig
   /etc/init.d/create_motd > /etc/motd
21 umount /mnt
   mkdir /localnfs
23 lln -s /misc/tftpboot   /localnfs/tftpboot
   mkdir /opt/scratch
25 ln -s /opt/scratch /scratch
   # Create ssh authorized keys
27 # Make the directory
   mkdir /root/.ssh
29 cat  << xxEOFxx >> /root/.ssh/authorized_keys
   wwrt4EFeqnFpF3RXFhPY1eiZNAI33GopEGVTiLTO4ZW9mYC8EI7e28= root@emstools
```

Principe du Boot Réseau.

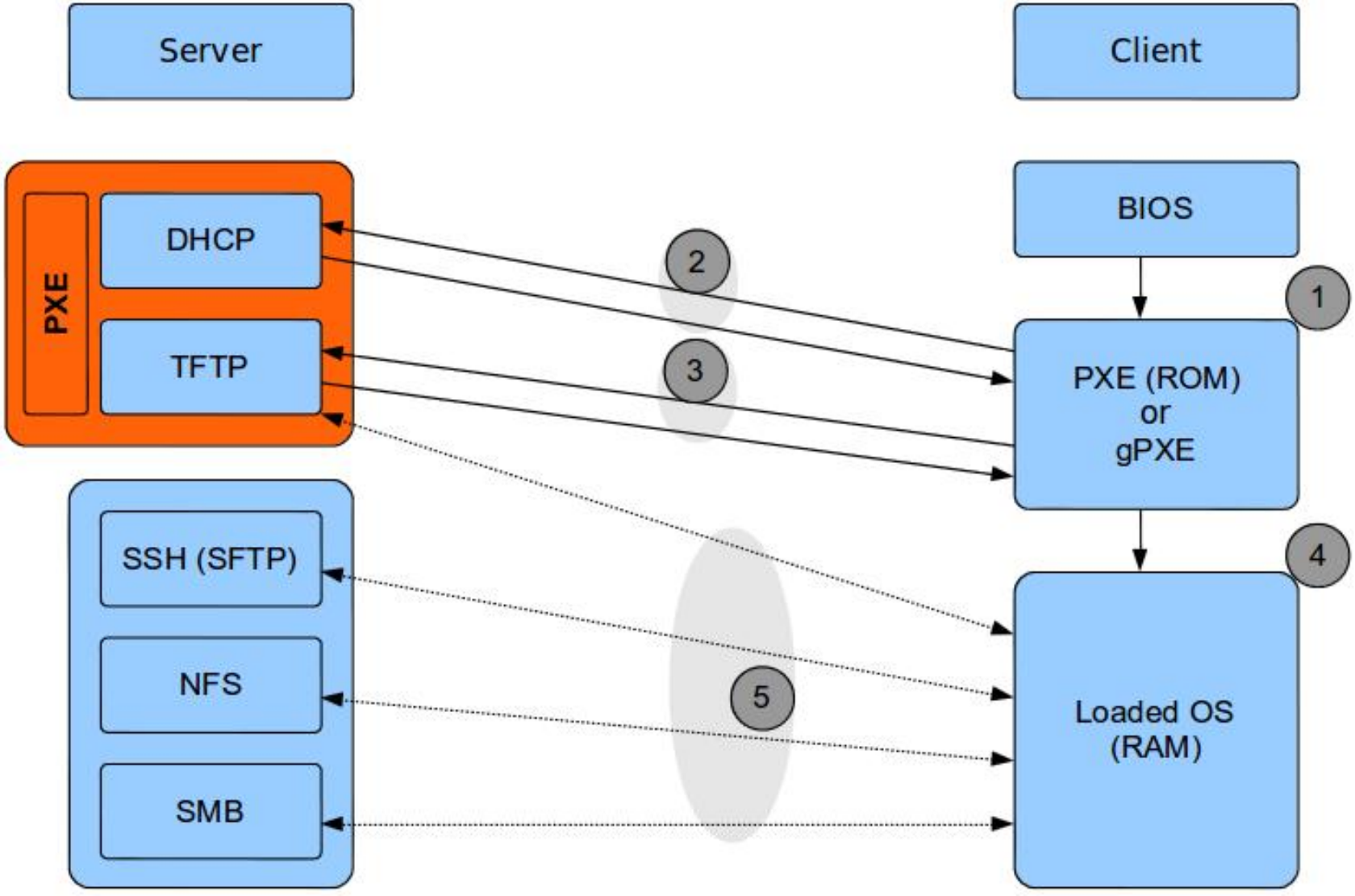
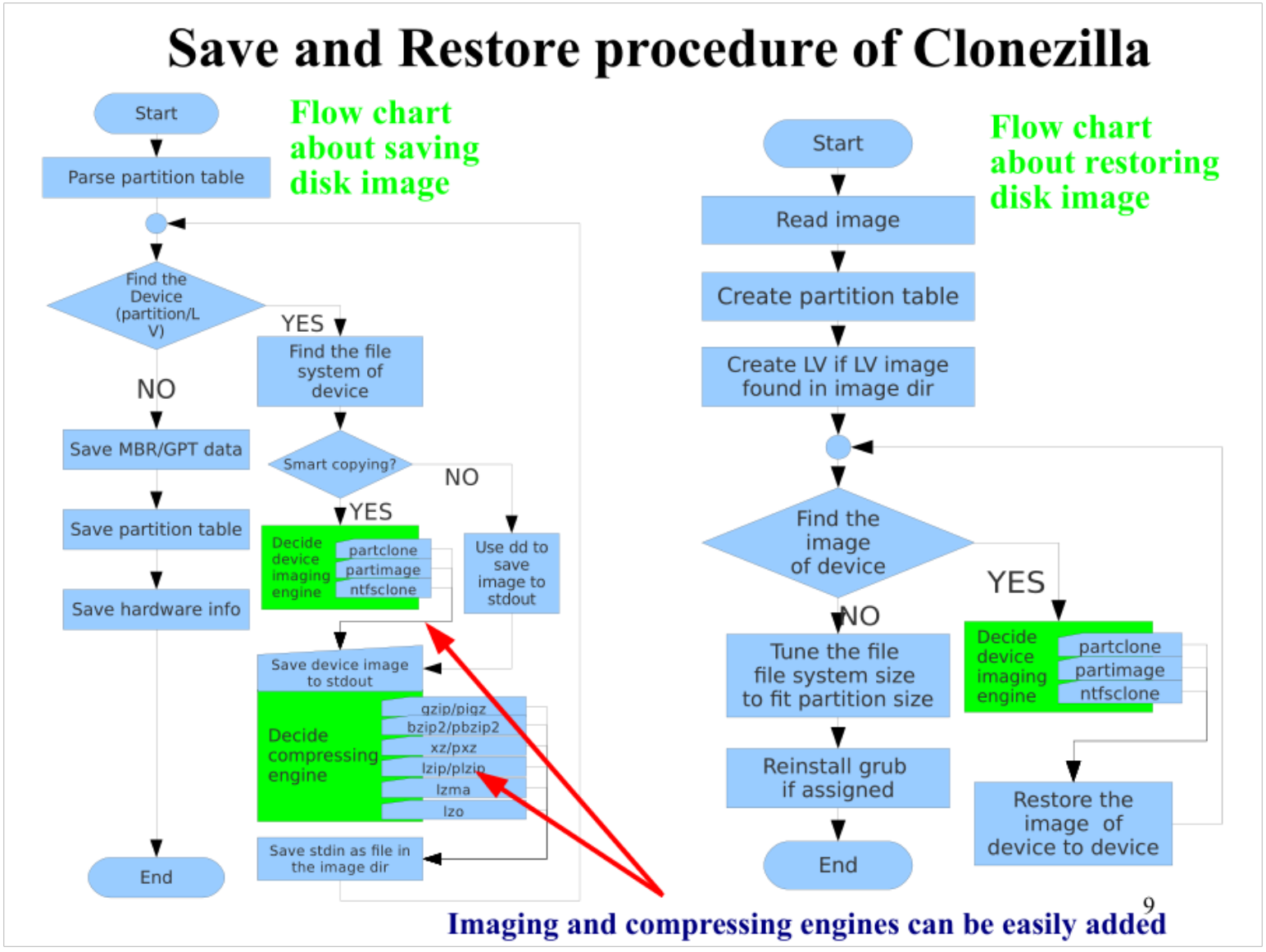


Figure:

Solution de duplication d'image - Clonezilla

- Free (GPL) Software
- FS supporté : Ext2/3/4, ReiserFS, Reiser4, XFS, JFS, HFS+, BrtFS, UFS, Minix, VMFS, FAT and NTFS,
- Supports LVM2
- Support de certaines puces RAID (Driver noyau Linux)
- Copie intelligente pour les systèmes de fichier supportés
  Pour les autres, copie secteur/secteur - dd.
- Boot loader : syslinux, grub 1/2 ; MBR and hidden data (if exist)
- Serial console
- restauration d'une image sur de multiple disques locaux.
- Mode serveur d'image avec multicast pour le transfert des images.
- Le format d'image est transparent et ouvert.

Solution Duplication Image - Clonezilla.



Solution install Cobbler/Kickstart.

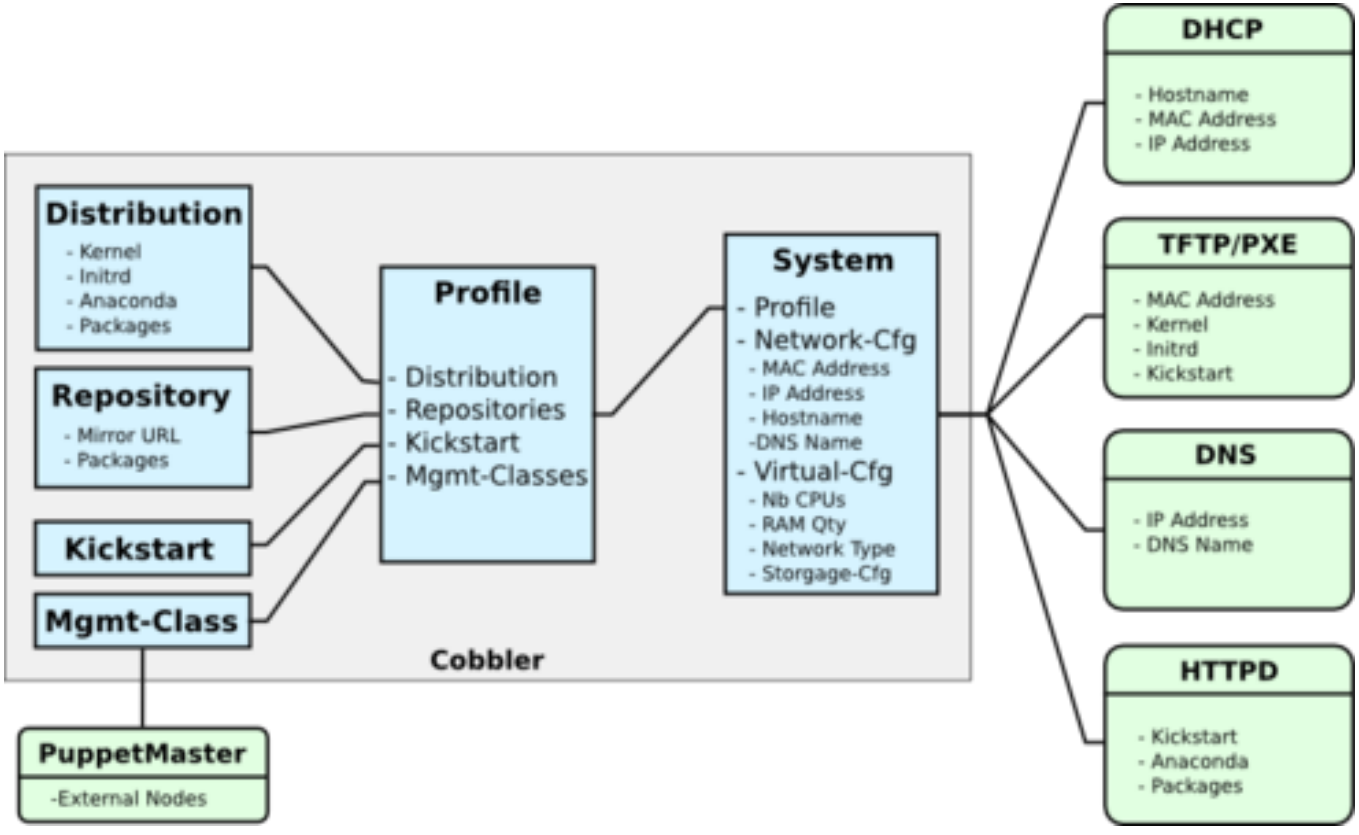


Figure: Realtion entre objet dans Cobbler.

Solution install Cobbler/Kickstart.

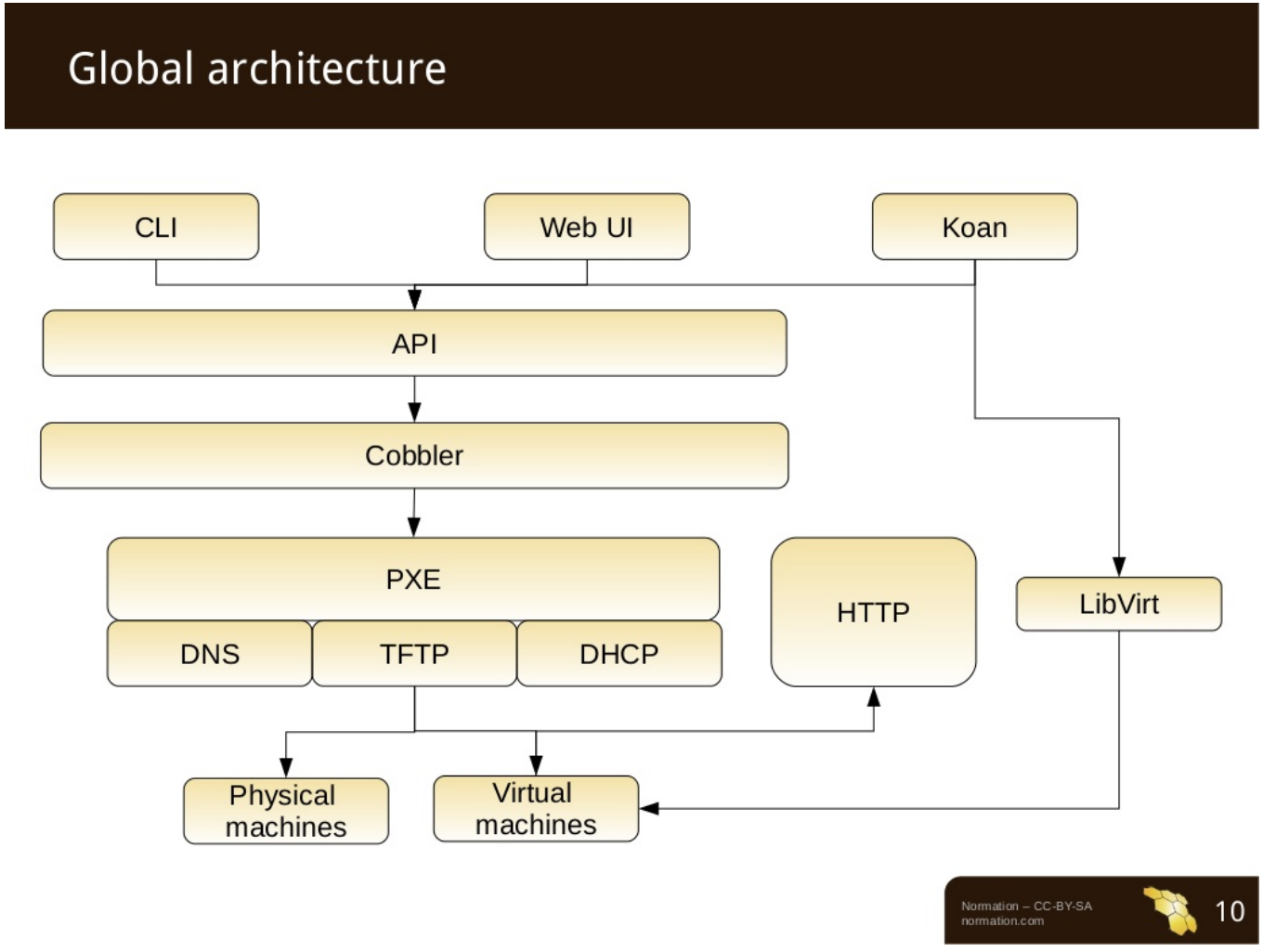
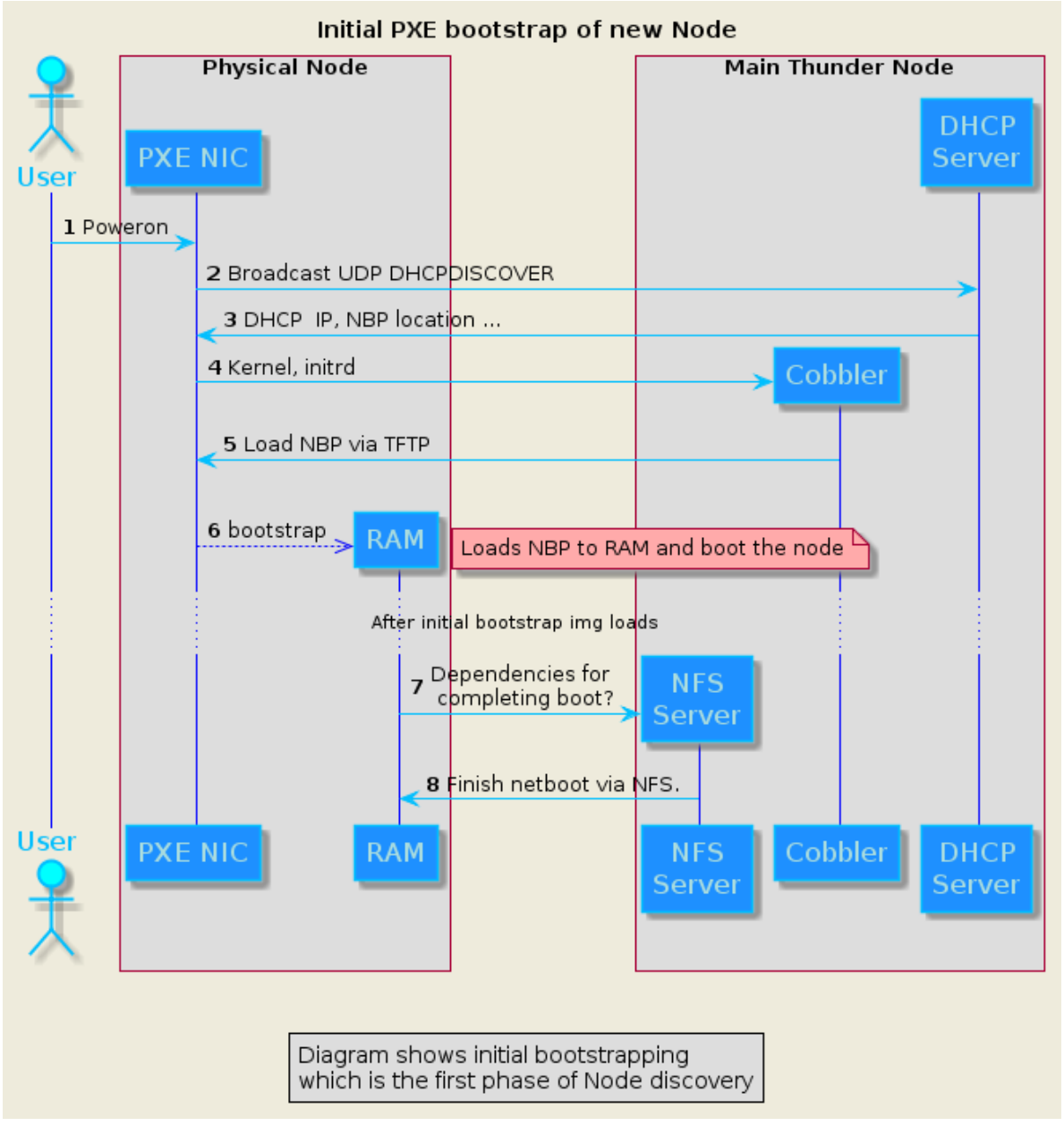


Figure: Architecture Cobbler.

Solution install Cobbler/Kickstart.



Diskless - SIDUS - Emmanuel QUEMENER.

SIDUS Single Instance Distributing Universal System, une instance unique distribuant un système d'exploitation.

Le principe Une image chargé en mémoire par boot réseau.

Les composants PXE ()utilisation d'un démarrage en réseau), AUFS(superposition de systèmes en lecture seule et lecture/écriture), TFTP (fourniture d'un noyau et d'un système de démarrage), NFSROOT(système racine unique partagé par tous les clients).

- Unicité du système : tous les clients démarrent exactement le même système (au bit près).
- Usage des ressources locales : les processeurs et mémoire vive exploités sont ceux des clients.

Pre requis pour une installation SIDUS :

- Un réseau suffisamment performant pour approcher les caractéristiques d'un disque.
- Coté serveur (nœud de management), les services DHCP, DNS, TFTP et NFS. Les deux derniers vont « porter » SIDUS.
- Côté client (nœud à démarrer/provisionner), la possibilité d'un démarrage PXE opérationnel (par la carte réseau, GPXE sur CDROM ou clé USB).

Diskless - SIDUS - Les 8 étapes d'une installation.

1. Préparation du système.
2. Installation de base (socle Debian, debootstrap).
3. Installation des paquets complémentaires (TOUT Debian-Science).
4. Purge des paquets non désirés.
5. Adaptation du système à l'environnement local.
6. Pointage du système vers les serveurs tiers :authentification et partages utilisateurs.
7. Création de la séquence de démarrage.
8. Détachement SIDUS du système hôte.

Rppel distribution - ROCKS.

- Distribution basée sur RedHat
- Base de données
- Fichiers de configuration en XML
- Directif (couche d'abstraction cluster)
- À base de ROLL (comme des modules applicatifs)
- Outil de diffusion à base de kickstart
- Peu manquer de souplesse dans certaines conditions

Une installation ROCKS.

1. Installation du serveur de login/management.
2. Insertion du DVD/Téléchargement des images ISO.
3. Exécution de l'outil d'installation "Graphique" (Une dizaine de formulaire de configuration).
4. Un café pendant l'installation ?
5. Installation des nœuds de calcul :
   - Login sur la frontale/nœud de management.
   - Execute "insert-ethers" => "Provisioning" des nœuds de calcul, enregistrement dans la BD SI.
   - Boot des nœuds de calcul.
   - Configuration automatique des réseaux et services pour ceux-ci
   - Reboot et installation des nœuds en fonction des images/rolls.
   - ...
6. Ajout des utilisateurs.
7. Début des tests avant exploitation.

Etude de cas : "baremetal" & Noeud Openstack - 1

Pre requis pour une installation reseau de noeud "baremetal".

- Configuration sur des "compute nodes" des services : tftp-server, ipmi, syslinux etc (pour le provisioning de matériel).
- Des bootstrap reseau ("Flavor") enregistrés dans Nova (gestionnaire boot/image).
- Des images systèmes adaptées stockées dans Glance. (bm-deploy-kernel, bm-deploy-ramdisk, user-image, user-image-vmlinuz, user-image-initrd).
- Le service Ironic qui gére le provisioning (détection et enregistrement des matériels).

Etude de cas : "baremetal" & Noeud Openstack - 2

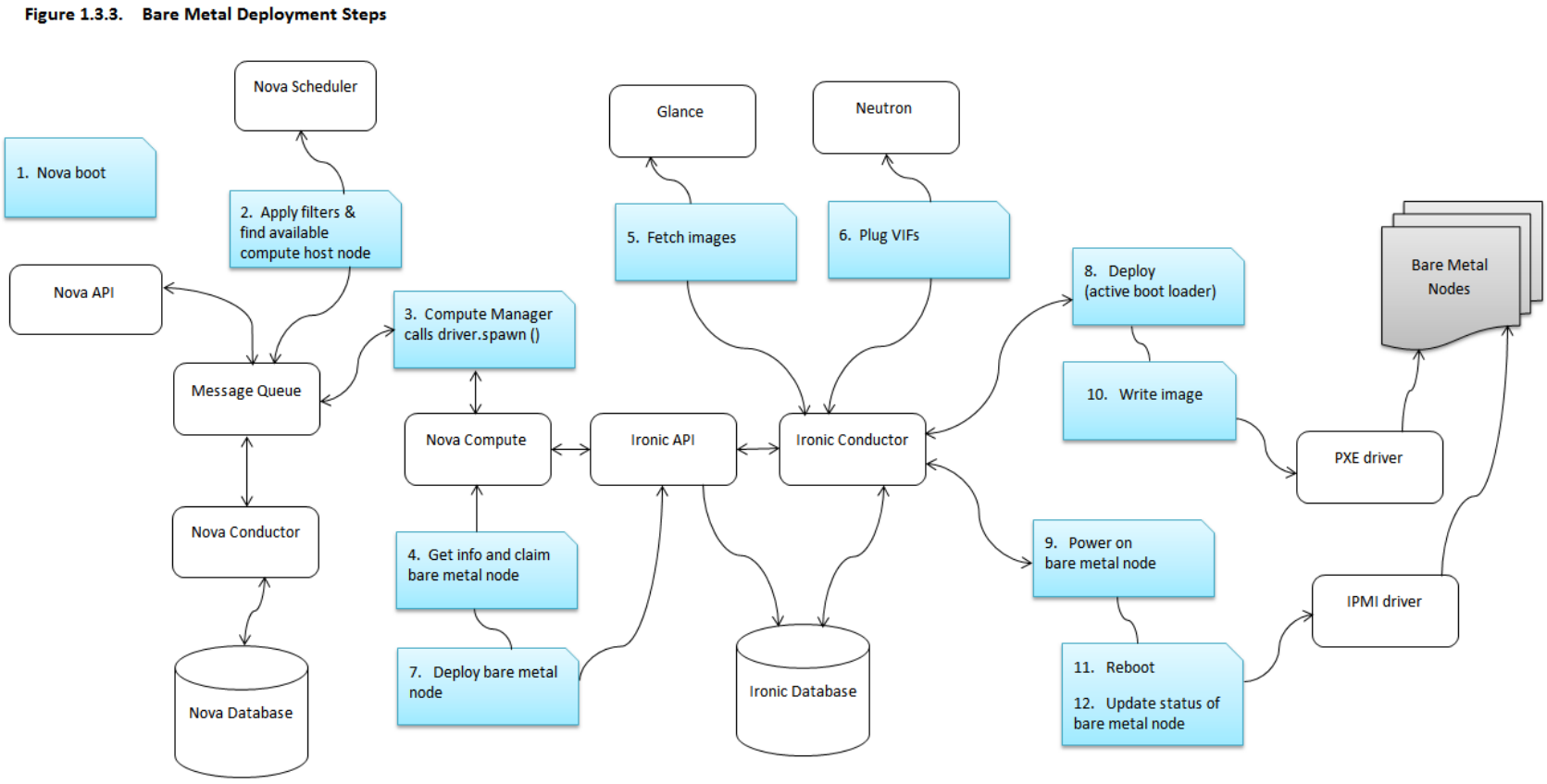


Figure: OpenStack provisioning.

Processus simplifié d'une install d'un Noeud Openstack - 1

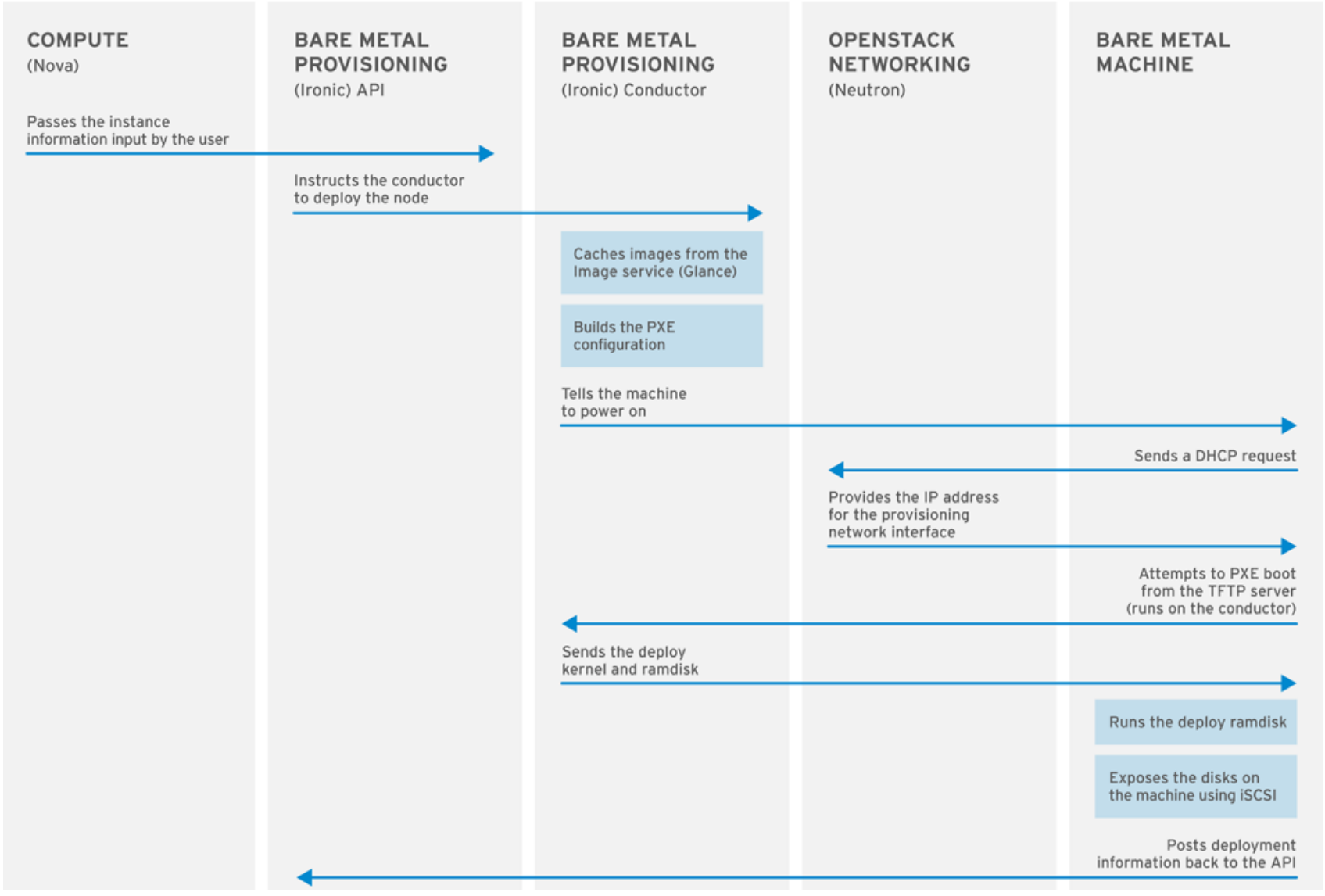
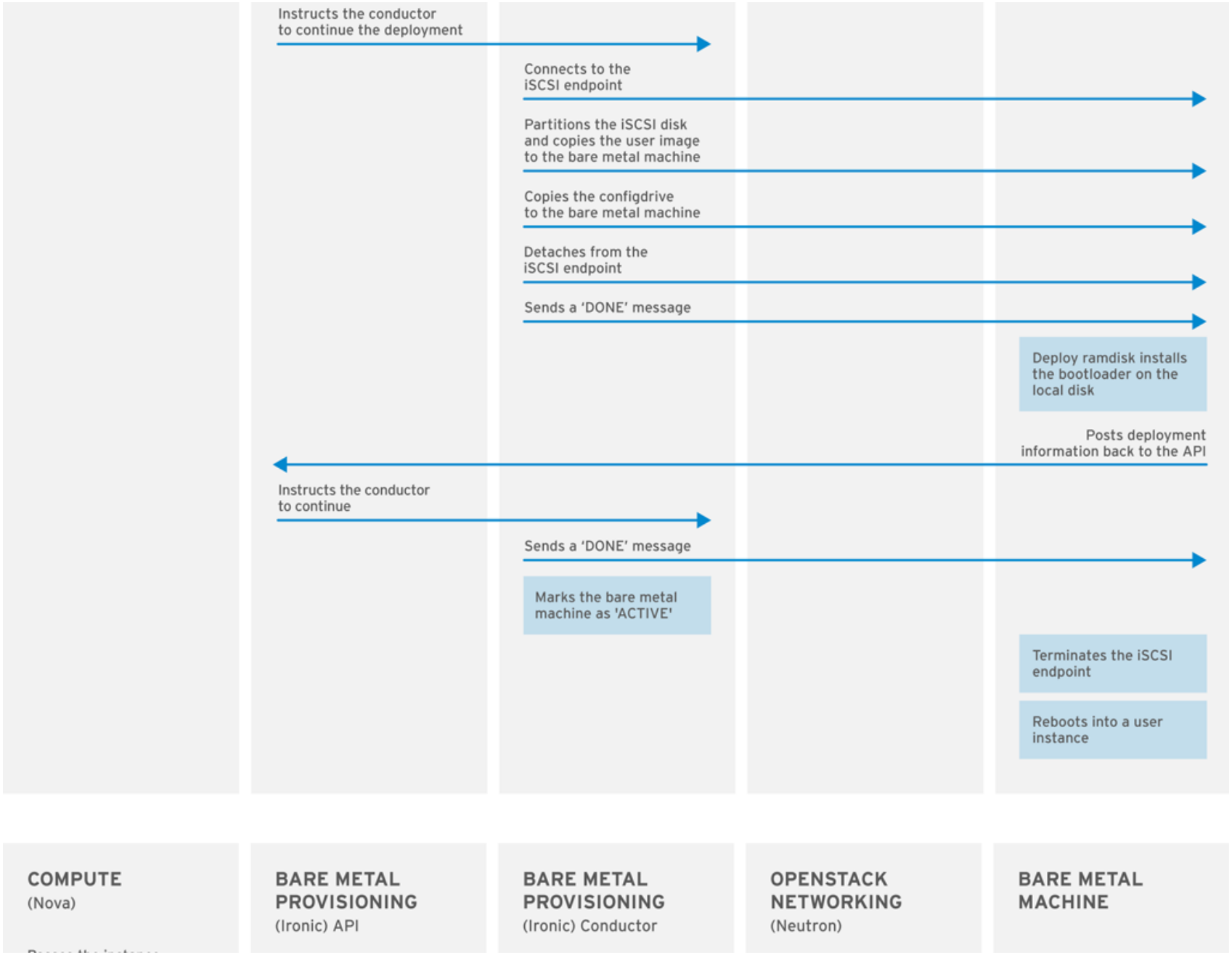


Figure: OpenStack provisioning.

Processus simplifié d'une install d'un Noeud Openstack - 2



Processus simplifié d'une install d'un Noeud Openstack - 3

- Composants riches (fonctionnalités étendues).
- Complexité du processus.
- Communauté hyperactive et concurrentiel.
- Mouvement DEVOPS -> un environnement en constante évolution.

Références.

Sites généraliste sur les clusters et le HPC :

- Beowulf http ://www.beowulf.org/overview/history.html
- ClusterMonkey
- hpc computing ()
- linux cluster institute (http ://www.lci.com)
- Initiative openHPC openhpc.org
- http ://clusterdesign.org/
- http ://www.hpcwire.com/
- http ://insidehpc.com
- http ://www.prace-ri.eu
- http ://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html

Références.

Sites Distributions linux.

- oscar/rocks/brice/platform/...
- ROCKS (http ://www.rocksclusters.org/wordpress/)
- Bright Cluster Manager
  (http ://www.brightcomputing.com/Bright-Cluster-Manager)

Références.

Outils clusters.

- forge.cbp.ens-lyon.fr/redmine/projects/sidus
- xCAT (Extreme Cluster/Cloud Administration Toolkit) (http ://sourceforge.net/p/xcat/wiki/Main_Page)
- cobbler (http ://www.cobbler.org)
- fai
- foreman
- kickstart
- Ansible : http ://docs.ansible.com/ansible/index.html
- XDMoD : https ://sourceforge.net/projects/xdmod/