

Slurm

Installation & Configuration

Matthieu Hautreux - matthieu.hautreux@cea.fr

Plan

- Rappels
- Sécurité dans Slurm
- Configuration et activation des composants
- Configuration avancée



Rappels

Slurm - Rappels

- **S**imple **L**inux **U**tility for **R**esource **M**anagement
 - Simple → Scalable
- Projet démarré au **LLNL** en 2002
 - Lawrence Livermore National Laboratory
 - Livermore, CA, USA
- Continué par **SchedMD** depuis 2010
 - Entreprise créé par les deux développeurs principaux de l'époque

Slurm - Rappels

- Produit **OpenSource** écrit en C
 - Licence GPLv2
- Utilisable sur la majorité des **environnements** de type **UNIX**
 - AIX, Linux, BSD, ...
- Utilisé sur une multitude de grands calculateurs à travers le monde
 - Dont certains parmi les plus grands

Slurm - Rappels

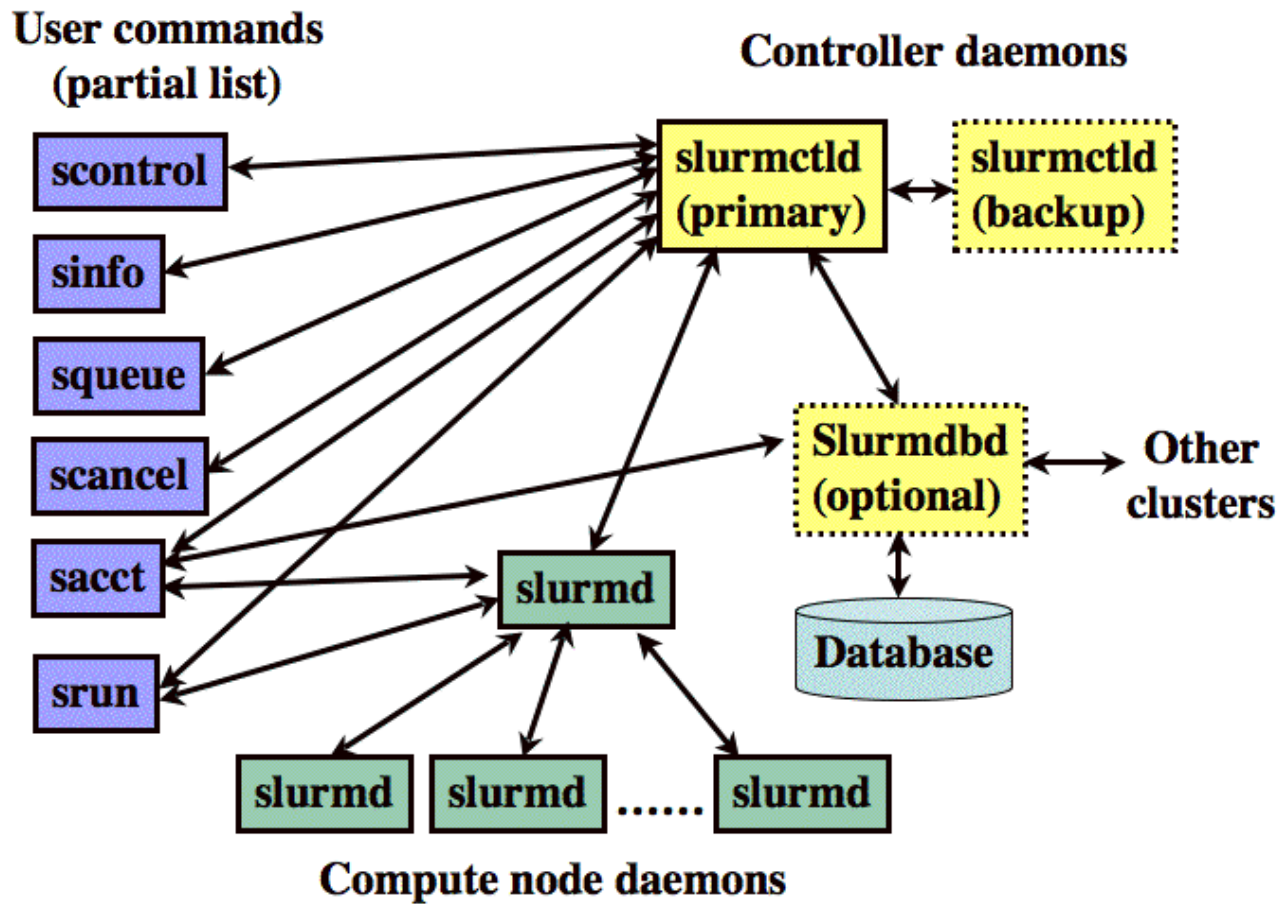
- **Scalable**

- Permet la gestion de plusieurs dizaines de milliers de nœuds
- Permet la gestion de plusieurs centaines de milliers de cœurs de calcul

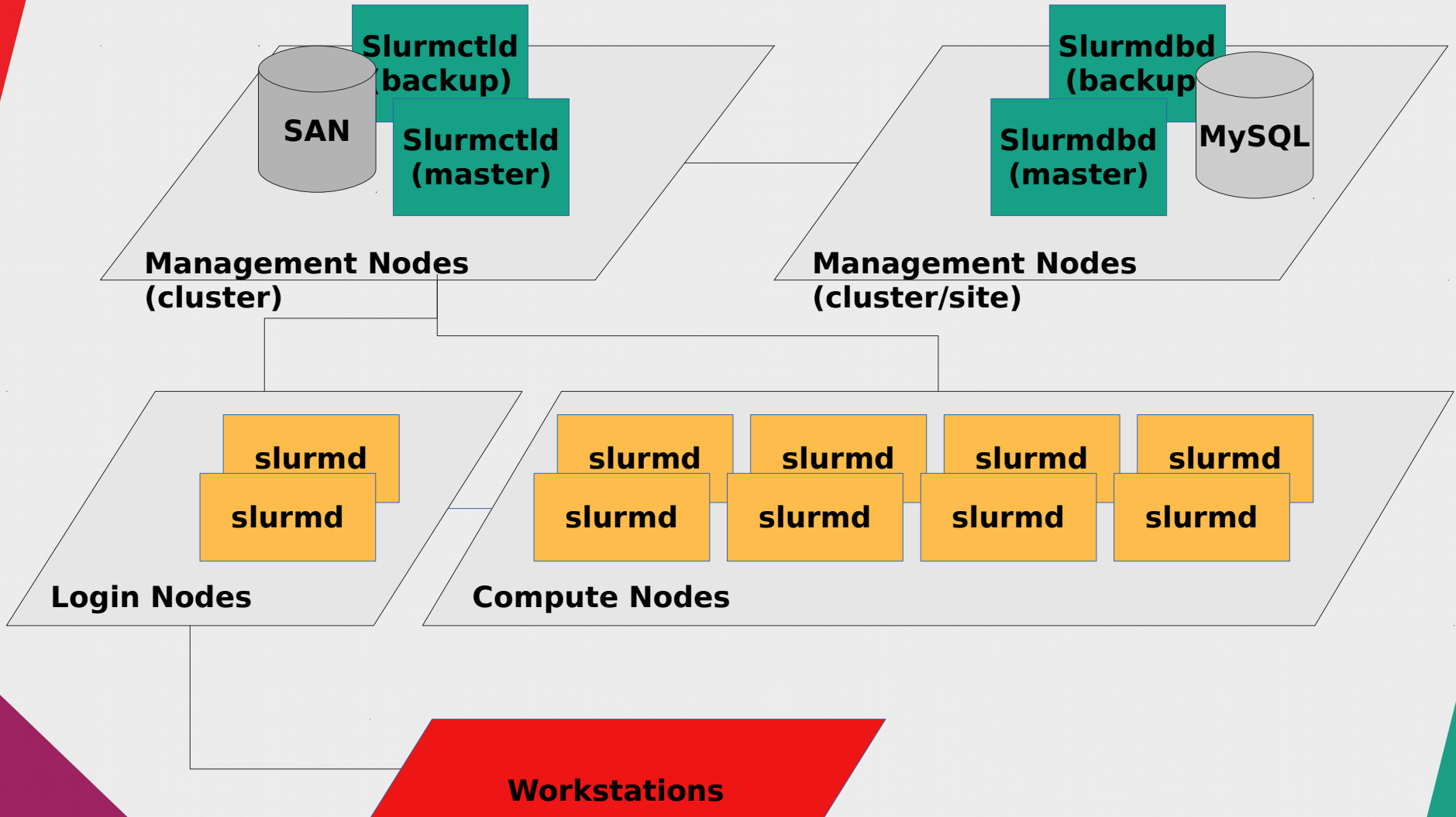
- **Modulaire**

- Basé sur la notion de **plugins** pour spécialiser différentes parties du produit en fonction des besoins

Slurm - Rappels



Slurm - Rappels





Sécurité dans Slurm

Slurm - Sécurité

- **Principaux objectifs de sécurité**
 - **Garantir un fonctionnement sécurisé en mode multi-utilisateurs**
 - **Notion de « multi-tenant » (multi-tenancy)**
 - Plusieurs utilisateurs utilisent le même service
 - **Exécution de programmes à l'identité des utilisateurs en faisant la demande**
 - Nécessité de restreindre les droits à ceux de l'utilisateur en question le temps des opérations

Slurm - Sécurité

- **Principaux objectifs de sécurité**
 - **Garantir la validité des identités des utilisateurs et entités logiques du système**
 - **Authentification des utilisateurs et des composants**
 - **Garantir l'authenticité des autorisations d'accès aux ressources accordées par le contrôleur**
 - **Pour garantir un fonctionnement sécurisé et performant en environnement distribué**

Slurm - Sécurité

- **« Security Plugins »**
 - **Auth Plugin - Authentification**
 - **Plugin permettant la fourniture d'une méthode d'authentification des messages basée sur un concept de « auth_info »**
 - Une partie du message contient les informations de validation...
 - **Chaque message contient une section « auth_info » générée par l'émetteur et validée par le récepteur qui en extrait les uid/gid de l'émetteur**

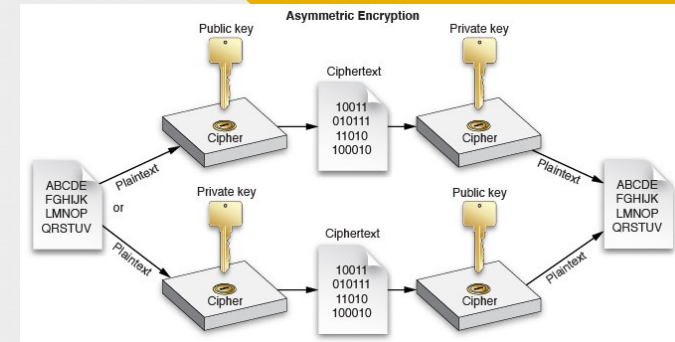
Slurm - Sécurité

- **« Security Plugins »**
 - **Crypto Plugin - Authenticité**
 - **Plugin permettant la génération et la validation d'une « signature » des « job credentials »**
 - issus de slurmctld et utilisés sur les slurmd
 - **Le « job credential » intègre sa signature**
 - L'utilisateur ne doit pouvoir la régénérer pour garantir l'authenticité du message issu du contrôleur.

Slurm - Sécurité

- **Auth Plugin - Authentification**
Implémentations
 - **Deux plugins disponibles**
 - **None : confiance totale envers les utilisateurs du système.**
 - Les uid/gid fournis par l'utilisateur sont considérés comme fiables !
 - **Munge : utilisation d'un système annexe nommé Munge**
 - Plus d'information dans les slides suivants !

Slurm - Sécurité



- **Crypto Plugin - Authenticité**
Implémentations

- **Deux plugins disponibles**

- **OpenSSL** : utilisation de cette librairie pour signer les « job credential » via une clé privée/publique
 - Slurmctld est le seul à connaître la privée...
 - Tout le monde connaît la publique
- **Munge** : utilisation d'un système annexe nommé Munge
 - Plus d'information dans les slides suivants !

Slurm - Sécurité

- **Recommandations**

- **MUNGE est le système recommandé par les développeurs de Slurm pour les deux types de plugins**
- **MUNGE (MUNGE Uid aNd GID Emporium) est un produit développé par le LLNL**
 - <https://dun.github.io/munge/>
- **MUNGE est disponible nativement sur de nombreuses distributions Linux**

Slurm - Sécurité

- **MUNGE**

- **Repose sur le partage d'un secret partagé par l'ensemble des nœuds intégrés dans un même espace de confiance (REALM).**
- **Repose sur la possibilité d'obtenir de façon fiable l'UID et le GID d'origine d'une connexion de type Socket BSD sur les systèmes UNIX.**
- **Fourni une API C et des commandes pour une utilisation scriptée**

Slurm - Sécurité

- **MUNGE**

- Permet de générer un « munge credential » chiffré intégrant :

- L'UID et le GID de l'utilisateur à l'initiative de la requête
 - Un contenu à transmettre de façon sécurisé
 - La date d'émission et le TTL (time-to-live)
 - Une restriction optionnelle sur l'UID/GID autorisé à déchiffrer le « credential »

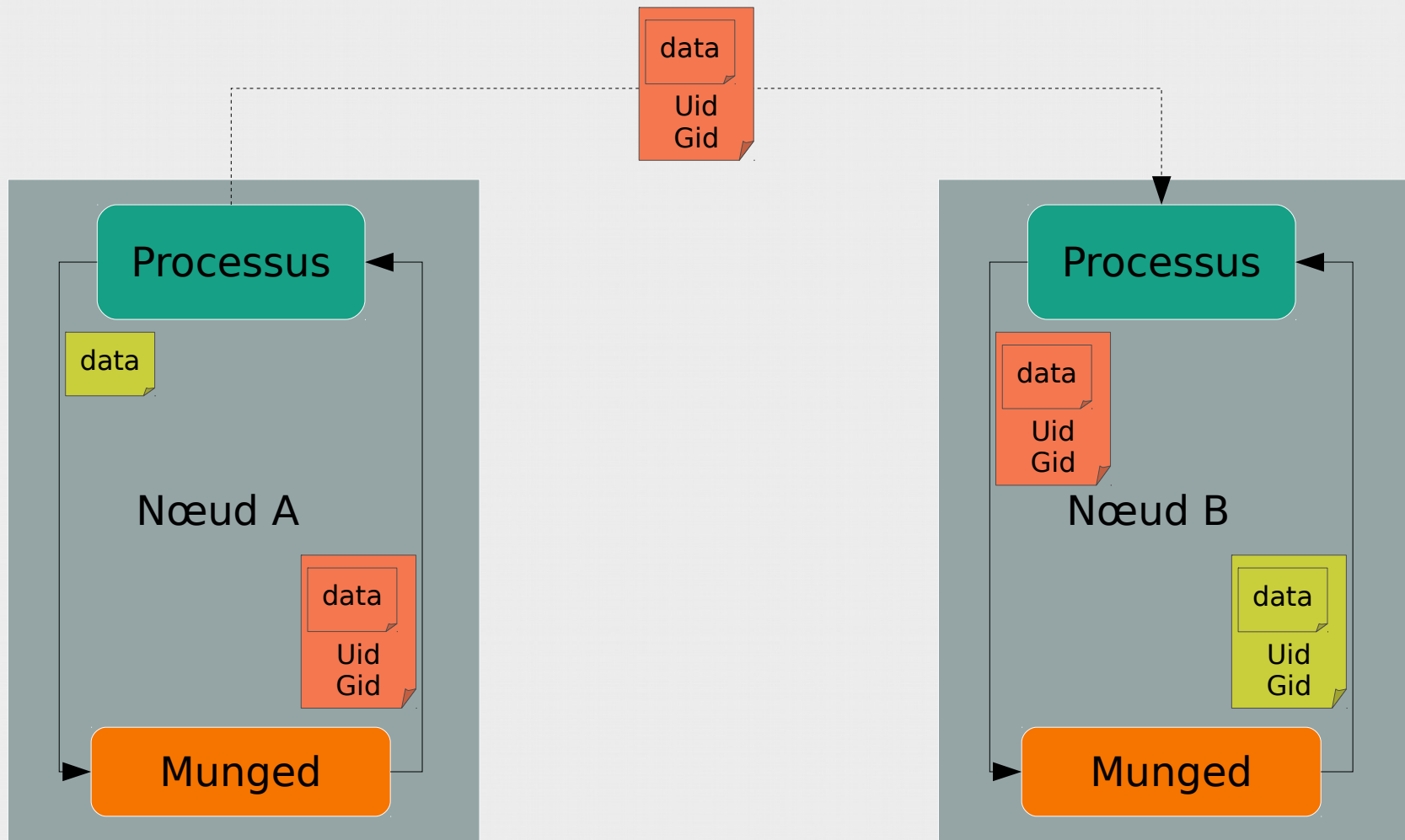
Slurm - Sécurité

- **MUNGE**

- **Permet de déchiffrer un « munge credential »**

- **Afin de vérifier qu'il n'est pas expiré**
- **Afin de vérifier qu'il n'a pas été « rejoué »**
 - **Le même credential a déjà été traité préalablement**
- **Afin d'obtenir le contenu déchiffré**
- **Afin d'obtenir l'UID/GID de l'émetteur**
- **À condition que l'UID/GID à l'initiative soit autorisé à effectuer l'opération**

Slurm - Sécurité



Slurm - Sécurité

- **Auth Plugin - MUNGE**
 - **Chaque utilisateur/démon**
 - **génère des « credentials » associés à son UID/GID et les ajoute à ses messages**
 - **valide les « credentials » des messages qu'il reçoit et valide que les UID/GID associés sont licites pour le service requis**

Slurm - Sécurité

- **Crypto Plugin - MUNGE**

- **Slurmctld**

- **génère des « job credentials » contenant les spécificités des ressources accordées**
 - **génère un « munge credential » du contenu du « job credential »**
 - **associé à son UID/GID**
 - **restreint en ouverture à l'UID/GID des slurmd**

Slurm - Sécurité

- **Crypto Plugin - MUNGE**

- **Slurmd**

- **Extrait le « munge credential » du « job credential »**
 - **Valide que l'UID/GID associé au « munge credential » est l'UID/GID de slurmd ou root**
 - **Valide que le contenu chiffré est identique au « job credential »**

Slurm - Sécurité

- **MUNGE - limitation**

- **Restreint à un ensemble de machines de « confiance »**

- **Tout accès root à une machine permet d'obtenir la clé partagée et de forger ses propres credentials**
 - **Tout accès root à une machine permet d'usurper l'identité de tout utilisateur sur l'ensemble des machines du REALM**

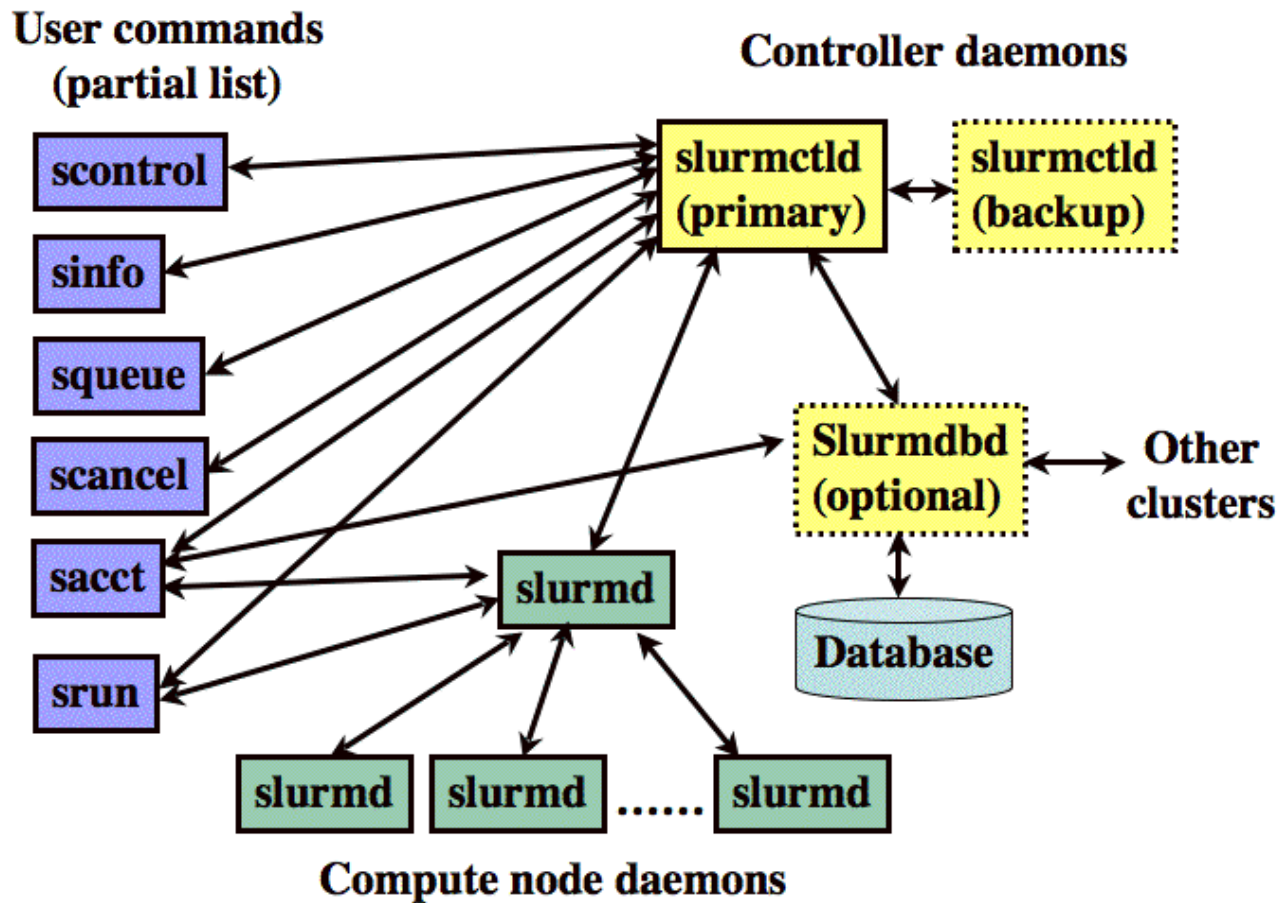
→ C'est normalement le cas d'un calculateur

- **Pas d'accès physique, sécurité OS**



Configuration et activation des composants

Slurm - Architecture



Slurm - Configuration

- **Fichiers de configuration**
 - **Spécificités du format**
 - **# commentaires**
 - **key=value**
 - **key=val1,val2,...**
 - **key=val attr1=a1 attr2=a2 ...**
 - **key=nodeset[1-10,12-20] attr=ns[1-10,12-20] ...**
 - **Include other_file_path**

Slurm - Configuration

- **Fichiers de configuration**
 - **Fichiers principaux**
 - **/etc/slurm/slurm.conf**
 - Utilisé par slurmctld
 - Utilisé par slurmd
 - Utilisé par les commandes lignes
 - srun, sbatch, squeue...
 - **/etc/slurm/slurmdbd.conf**
 - Utilisé par slurmdbd uniquement

Slurm - Activation

- **Services principaux (RedHat style)**
 - **slurmctld**
 - **systemctl start/status/stop slurmctld**
 - **slurmdbd**
 - **systemctl start/status/stop slurmdbd**
 - **slurmd**
 - **systemctl start/status/stop slurmd**

Slurm - Configuration

- **Slurmdbd**

- **Nécessite la mise en place préalable d'une base de données MySQL**
 - **Mariadb**
- **Nécessite la création d'une « database » dédiée à Slurm**
 - **« slurm_acct_db » par défaut**
- **Nécessite la fourniture des droits d'accès pour la manipulation depuis slurmdbd**
 - **Le démon peut être local ou distant**

Slurm - Configuration

- **Slurmdbd**

- **Peut être utilisé pour gérer différents clusters Slurm**

- **Afin d'éviter le déploiement d'une base et d'une instance par cluster**
 - **Afin de centraliser la définition des accounts / users**
 - **Afin de centraliser les statistiques et leurs traitements automatisés**
 - **Outil « sreport »**

Slurm - Configuration

- **Slurmdbd**
 - **Est accédé par slurmctld pour**
 - **l'interrogation des accounts / users / qos / associations / ...**
 - **l'écriture des états du cluster**
 - **l'écriture des informations sur les jobs enregistrés et leurs statistiques**
 - **...**

Slurm - Configuration

- **Slurmdbd**

- **Est accédé par les utilisateurs pour**
 - **Obtenir des informations sur leurs jobs en cours ou passés**
- **Est accédé par les administrateurs pour**
 - **Obtenir des informations sur leurs jobs en cours ou passés**
 - **Générer des rapports d'utilisation du cluster**
 - **Modifier les accounts / users / qos / associations / ...**

Slurm - Configuration

- **Slurmdbd**

- **Permet une purge régulière des données en base si nécessaire**
 - **Pour limiter la volumétrie et simplifier les conversions pendant les changements de versions**
- **Permet un archivage au moment des purges pour conservation externe des données purgées**
 - **Configurable par type de données**
 - Ex : job, job steps, ...
 - **Script d'archivage à fournir**

Slurm - Configuration

- **slurmdbd.conf**

- **Éléments de configuration principaux**

- **DbdHost**

- le nom de la machine hébergeant le service
ex : localhost, vm0, ...

- **DbdAddr**

- l'adresse IP (ou un alias) utilisée pour les connexions si différent de ANY
ex : 127.0.0.1, localhost, vm0-eth0, ...

- **DbdPort**

- Le port TCP ouvert en écoute, 7031 par défaut
ex : 7031, 12345, ...

Slurm - Configuration

- **slurmdbd.conf**
 - **Éléments de configuration principaux**
 - **AuthType**
 - La méthode d'authentification souhaitée
ex : `AuthType=auth/munge`
 - **SlurmUser**
 - Nom de l'utilisateur local qui sera utilisé pour exécuter l'application
ex : `slurm, root, ...`
 - **PrivateData**
 - Permet de limiter la visibilité des informations en fonction des utilisateurs si nécessaire
ex : `jobs, usage, users, events, ...`

Slurm - Configuration

- **slurmdbd.conf**

- **Éléments de configuration principaux**

- **StorageType / StorageLoc**

- Type de DB utilisé (plugin associé)

- ex : `accounting_storage/mysql`

- Nom de la database utilisée sur le backend

- ex : `slurm_acct_db`

- **StorageHost / StoragePort**

- Alias/IP pour l'accès à la DB

- Port TCP pour l'accès à la DB

- **StorageUser / StoragePass**

- Utilisateur dans la DB pour accéder à la database Slurm

- Mot de passe pour l'accès à la DB

Slurm - Configuration

- **slurmdbd.conf**
 - **Éléments de configuration principaux**
 - **LogFile**
 - Le fichier de log utilisé par le démon
 - **DebugLevel**
 - Le niveau de verbosité de la log
 - Utile de l'augmenter pour rechercher l'origine d'un problème
 - ex : 0,1, ... , 9 | quiet, fatal, error, info, debug, ...
 - **DebugFlags**
 - Utile pour améliorer la verbosité de certaines sous-parties
 - ex : DB_JOB, DB_QOS, DB_ASSOC, ...

Slurm - Configuration

- **slurmdbd.conf**
 - **Éléments de configuration principaux**
 - **Encore bien d'autres à découvrir**
 - « `man slurmdbd.conf` »
 - <https://slurm.schedmd.com/slurmdbd.conf.html>

Slurm - Configuration

- **Slurmctld / Slurmd**
 - **Peuvent partager un même fichier de configuration**
 - **Doivent même le faire par défaut, sans quoi un avertissement est inscrit dans la log pour le faire savoir pour chaque nœud en infraction (configurable!)**
 - **Un hash de la conf est transmis par chaque nœud pour validation auprès du contrôleur**
 - **Partagent le même fichier que les commandes lignes**
 - **Qui doivent connaître les entités du cluster et les modes de communication**

Slurm - Configuration

- **Slurmctld / Slurmd**
 - **Doivent partager la même définition des nœuds intégrés au calculateur**
 - Pour permettre des communications arborescentes (scalabilité)
 - **Disposent de pragmas communs et de leurs propres pragmas**
 - Simplifie la factorisation des configurations
 - Simplifie la colocalisation slurmctld/slurmd

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration principaux**
 - **ClusterName**
 - Le nom du cluster utilisé.
ex : ensiie, hpc, ...
 - **SlurmUser**
 - Le nom de l'utilisateur exécutant slurmctld
ex : slurm, root, ...
 - Slurmd s'exécute généralement en tant que root car doit pouvoir devenir n'importe quel utilisateur au moment des jobs

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration principaux**
 - **ControlMachine**
 - Le nom de la machine contrôleur.
Ex : vm0, ...
 - **ControlAddr / SlurmctlPort**
 - L'alias/IP à utiliser pour le contacter si différent de ControlMachine
 - Le port TCP à utiliser (default : 6817)
 - **SlurmctlTimeout**
 - Le timeout de connexion en secondes vers le contrôleur
Ex : 300

Slurm - Configuration

- **slurm.conf**

- **Éléments de configuration principaux**

- **StateSaveLocation**

- L'emplacement de sauvegarde des fichiers d'état du contrôleur. Utilisés pendant les redémarrage ou les bascules HA (si backup).

- Ex : /var/slurm/slurmctld, /tmp/slurmctld, ...

- **SlurmctldDebug**

- Niveau de verbosité du contrôleur

- Ex : 0, ... , 9 | quiet, fatal, error, info, debug, ...

- **SlurmctldLogFile**

- Fichier de log à utiliser (absence → syslog)

- Ex : /var/log/slurmctld.log

Slurm - Configuration

- **slurm.conf**

- **Éléments de configuration principaux**

- **AccountingStorageType**

- Le type de backend d'accounting

- Ex : `accounting_storage/slurmdbd`

- **AccountingStorageHost**

- La machine à contacter pour ce service

- Ex : `vm0`

- **AccountingStorageEnforce**

- Le type de contrôle effectué par le contrôleur dans la gestion des jobs à partir des infos du backend

- Ex : « `qos` » , « `qos,limits` » , ...

Slurm - Configuration

- **slurm.conf**

- **Éléments de configuration principaux**

- **SelectType**

- Le type d'algorithme de sélection de ressources utilisé par le contrôleur.

- Deux principaux choix :

- Sélection de nœuds complets (linear)

- Sélection de cpus individuels dans les nœuds (cons_res)

- Ex : `SelectType=select/cons_res`

- **SelectTypeParameters**

- Paramètres additionnels pour l'algo précédent

- EX : `CR_Core,CR_CORE_DEFAULT_DIST_BLOCK,CR_ONE_TASK_PER_CORE`

Slurm - Configuration

- **slurm.conf**

- **Éléments de configuration principaux**

- **SchedulerType**

- Le type d'optimisation de scheduling souhaité.

- Deux choix possibles

- Highest Priority First (builtin)

- Backfilling logic (backfill)

- Ex : SchedulerType=sched/backfill

- **SchedulerParameters**

- Paramètres additionnels pour l'algo précédent

- Ex : bf_window=1440,bf_interval=60,defer ...

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration principaux**
 - **PriorityType**
 - Type d'algorithme de priorisation
 - **Basic** : priorité basée sur l'ordre d'arrivée
 - **Multifactor** : combinaison de différents factors (cf cours 1)
Ex : `PriorityType=priority/multifactor`
 - **PriorityWeight{QOS,Fairshare,...}**
 - Les poids des différents factors (cf cours 1)

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration principaux**
 - **SlurmdPort**
 - Le port TCP utilisé par les démons « slurmd »
Default : 6818
 - **SlurmdSpoolDir**
 - L'emplacement des fichiers d'état de slurmd
Ex : /var/slurm/slurmd, /tmp/slurmd, ...
 - **SlurmdDebug**
 - Niveau de verbosité de slurmd
Ex : 0, ... , 9 | quiet, fatal, error, info, debug, ...
 - **SlurmdLogFile**
 - Fichier de log utilisé par slurmd (absence → syslog)
Ex : /var/log/slurmctld.log

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration des nœuds**
 - **nodeName**
 - Noms des nœuds configurés par cette ligne de conf
Ex : `nodeName=vm[0-3]`
 - **Sockets**
 - Nombre de socket (processeurs physiques)
Ex : `Socket=1`
 - **CoresPerSocket**
 - Nombre de « core » par socket (default : 1)
Ex : `CoresPerSocket=1`

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration des nœuds**
 - **ThreadsPerCore**
 - Nombre d'hyperthread par core (default : 1)
 - Ex : `ThreadsPerCore=1`
 - **RealMemory**
 - Quantité de mémoire disponible pour le calcul en Mo
 - Ex : `RealMemory=8000`
 - **State**
 - État de base du nœud au démarrage du contrôleur
 - Ex : `UNKNOWN, DOWN, DRAIN, ...`

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration des nœuds**
 - **Exemple complet**

```
NodeName=vm[0-3] Sockets=2 State=UNKNOWN RealMemory=8000
```

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration des partitions**
 - **PartitionName**
 - Nom de la partition
Ex : vm, haswell, skylake, ...
 - **Default**
 - Yes/No, pour indiquer que la partition est la partition des jobs n'indiquant aucune partition spécifique
 - **State**
 - L'état de la partition au premier démarrage de Slurm
Ex : Up, Down, ...

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration des partitions**
 - **MaxTime**
 - Durée maximale des jobs dans cette partition en minutes
Ex : 90, INFINITE, ...
 - **MaxNodes**
 - Nombre maximum de nœuds allouables par job dans cette partition
 - **Nodes**
 - Liste des nœuds associés à la partition
Ex : Nodes=vm[0-3]

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration des partitions**
 - **Exemple complet**

`PartitionName=vm Nodes=vm[0-3] Default=YES MaxTime=INFINITE State=UP`

Slurm - Configuration

- **slurm.conf**
 - **Éléments de configuration principaux**
 - **Encore bien d'autres à découvrir**
 - « `man slurm.conf` »
 - <https://slurm.schedmd.com/slurm.conf.html>

Slurm - Activation

- **Activation via `systemctl`**
- **`systemctl start slurmdbd`**
- **`systemctl start slurmctld`**
- **`systemctl start slurmd`**
- *`systemctl start/status/stop ...`*
- *`systemctl enable/disable ...`*



Configuration avancée

Slurm - Configuration avancée

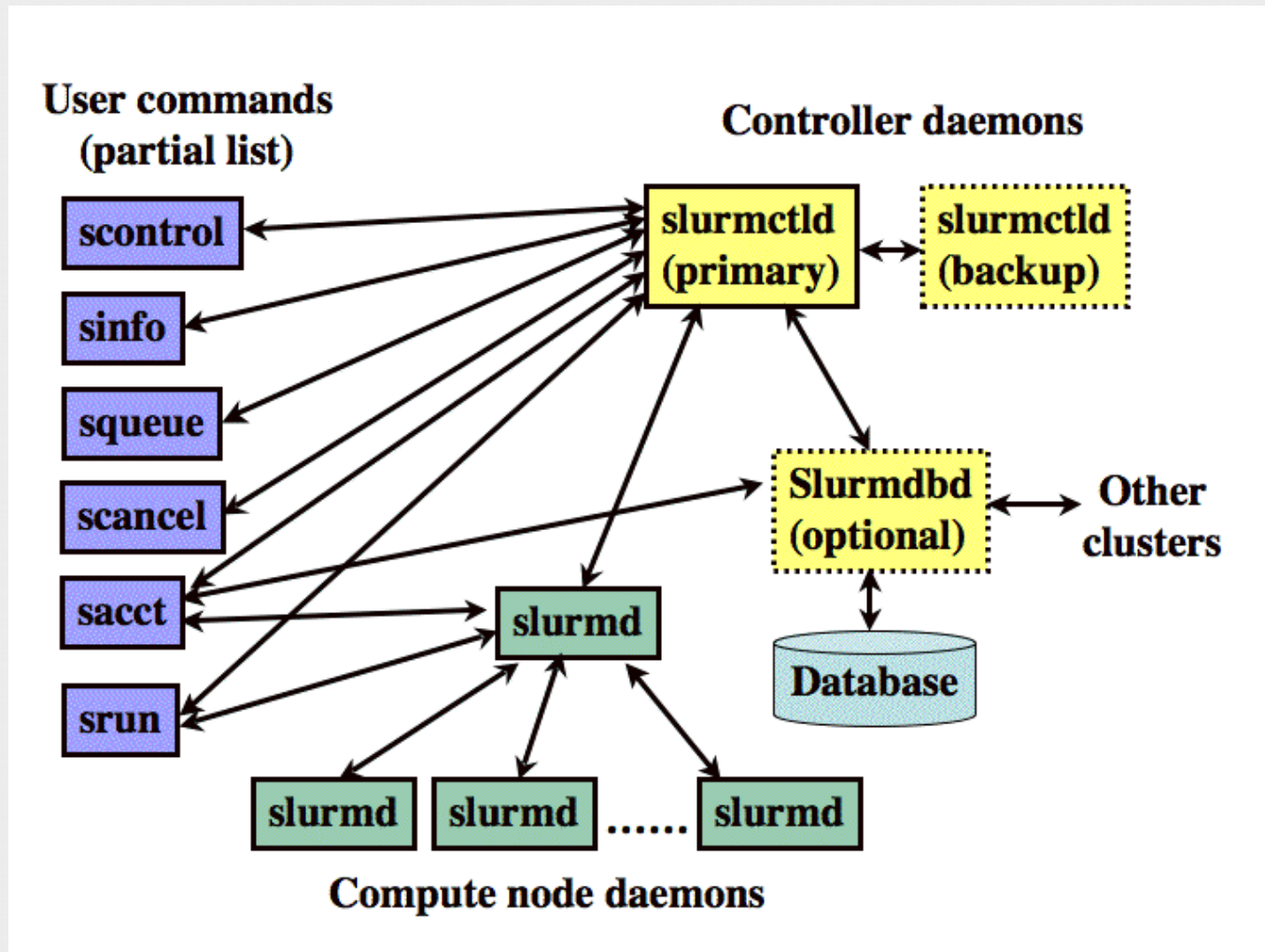
- **Beaucoup de paramètres sont disponibles....**
 - **Mais il est préférable de commencer simple et d'optimiser ensuite**
 - Les man pages et les docs sont alors indispensables
 - Ainsi qu'un peu de patience et beaucoup de pratique / tests
 - **Il existe un générateur de conf !**
 - Il reste toutefois assez simpliste lui aussi
<https://slurm.schedmd.com/configurator.html>

Slurm - Configuration avancée

- **Optimisations discutées ici**
 - **Communications arborescentes**
 - **Sélection topologique des ressources**
 - **Confinement et binding des processus**

Slurm - Configuration avancée

- **Communications arborescentes**



Slurm - Configuration avancée

- **Communications arborescentes**
 - **Objectif**
 - Améliorer la performance des communications entre slurmctld et les slurmd
 - Améliorer l'agrégation des retours d'information des jobs vers le contrôleur
 - **Moyen**
 - Hiérarchiser & Paralléliser les communications entre les éléments

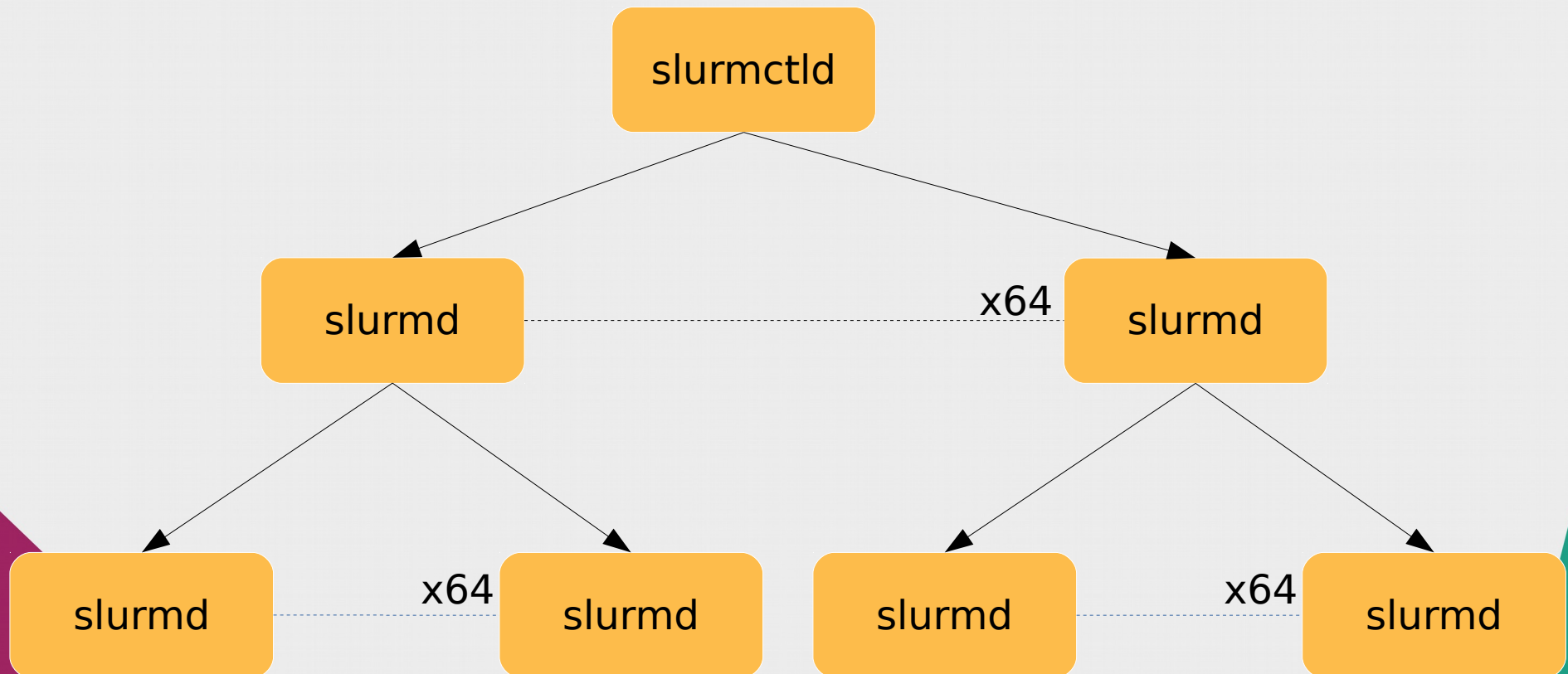
Slurm - Configuration avancée

- **Communications arborescentes**
 - **Notion de « fan-out » défini à l'aide du paramètre `TreeWidth` de `slurm.conf`**
 - **Correspond au nombre maximum de destinataires contactés en parallèle par chaque participant**
 - **Default=50**
Ex : `TreeWidth=64`
 - **Conseil `slurm.conf`**
 - **Racine carrée pour $N < 2500$**
 - **Racine cubique pour $N > 2500$**

Slurm - Configuration avancée

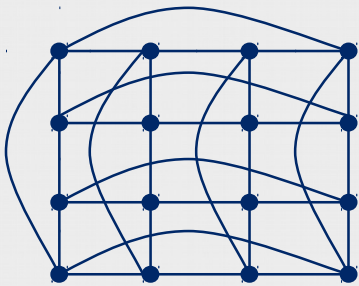
- **Communications arborescentes**

TreeWidth=64

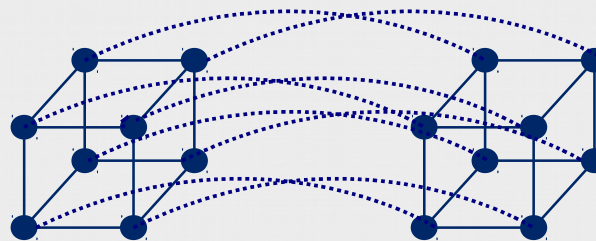


Slurm - Configuration avancée

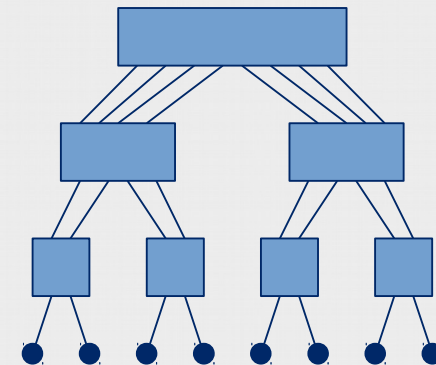
- **Sélection topologique des ressources**
 - **Différentes topologies réseau pour différentes machines**
 - Mesh, Tore, Hypercube, Fat-Tree, ...



Tore 2D (3D, ...)



Hypercube



Arbre (Fat Tree)

Slurm - Configuration avancée

- **Sélection topologique des ressources**
 - **Plugin « topology » de Slurm**
 - **Topology/tree permet de définir une hiérarchie de switchs d'interconnexion dans le cas d'un fat tree.**
 - Et d'optimiser le remplissage des switchs lors de la sélection des ressources pour chaque job
 - C'est la famille de topologie la plus courante sur les clusters de quelques dizaines/centaines de nœuds
 - **TopologyPlugin=topology/tree dans slurm.conf**
 - Fichier de conf topology.conf à renseigner

Slurm - Configuration avancée

- **Sélection topologique des ressources**
 - **Exemple de topology.conf**

```
SwitchName=l1 Nodes=vm[0,2]
```

```
SwitchName=l2 Nodes=vm[1,3]
```

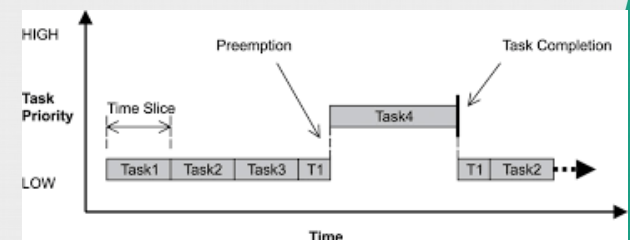
```
SwitchName=t0 Switches=l[1-2]
```

Slurm - Configuration avancée

- **Confinement et binding des processus**
 - **Par défaut, l'allocation des ressources est « laxiste »**
 - **L'utilisateur peut potentiellement accéder à plus de ressources qu'allouées**
 - **Par défaut, l'allocation est globale pour les processus d'un même nœud**
 - **Chaque processus accède à l'ensemble des ressources et se déplace d'une unité de calcul (cpu) à une autre au gré de l'OS**

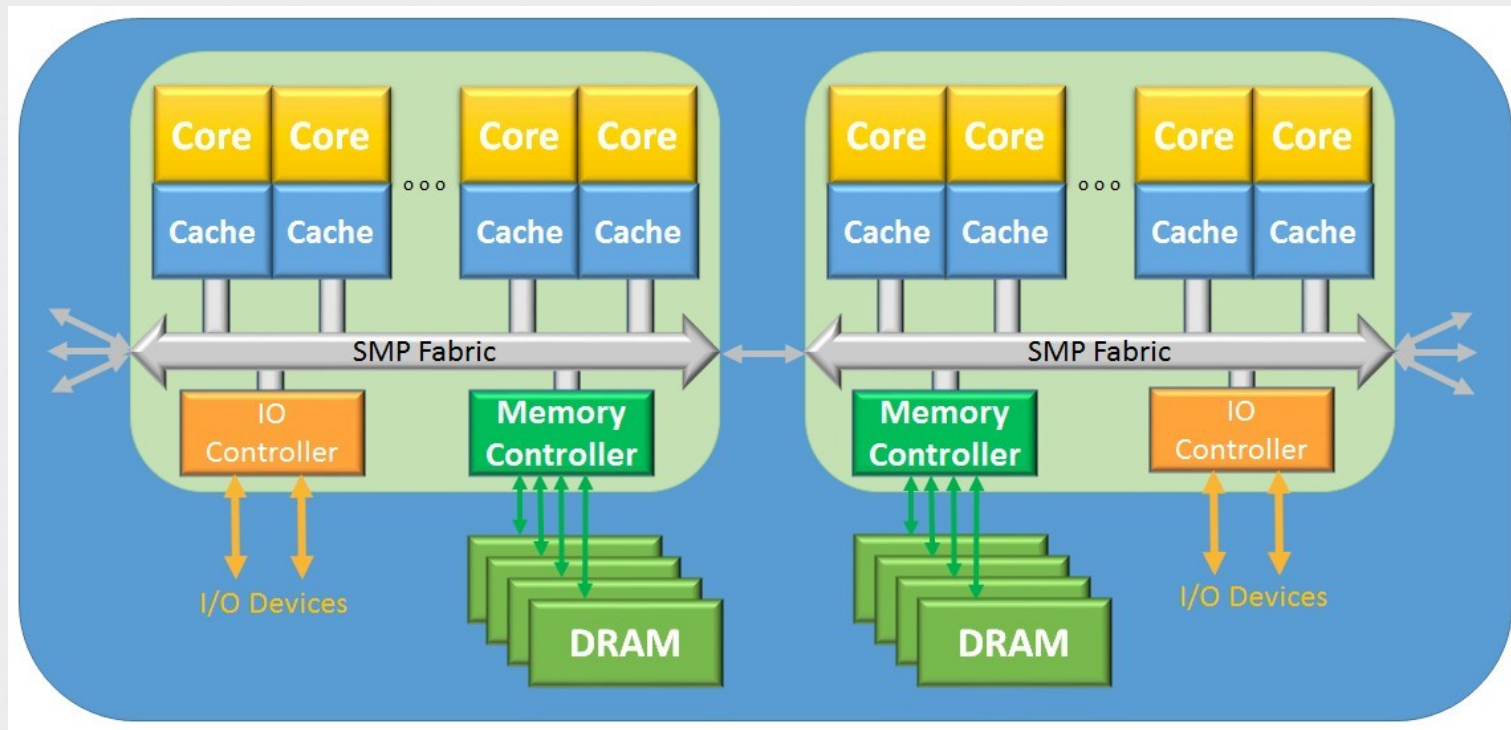
Slurm - Configuration avancée

- **Confinement et binding des processus**
 - Le fonctionnement par défaut n'est donc pas optimal lorsque :
 - Le nœud de calcul est partagé entre plusieurs jobs
 - possible avec `select/cons_res`
 - Les processus ont besoin de rester sur les mêmes unités de calcul pendant leur exécution pour bénéficier des effets de cache
 - Souvent le cas en HPC !



Slurm - Configuration avancée

- **Confinement et binding des processus**

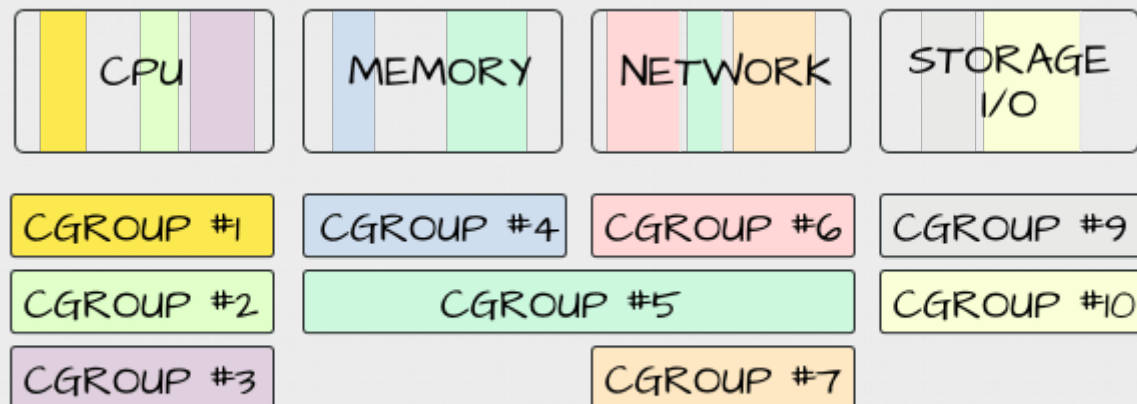


Slurm - Configuration avancée

- **Confinement et binding des processus**
 - **TaskPlugin permet de mettre en place des mécaniques de confinement et de binding pour résoudre ces problématiques**
 - **TaskPlugin=task/cgroup est une des méthodes permettant cela**
 - **A définir dans slurm.conf**
 - **Fichier de conf cgroup.conf à renseigner**

Slurm - Configuration avancée

- **Confinement des processus**
 - **Les « cgroup »**
 - Fournissent des primitives de regroupement de processus avec inclusion automatique des processus fils
 - Fournissent différents sous-systèmes pour traiter le partage de différentes ressources



Slurm - Configuration avancée

- **Confinement des processus**
 - Le « **cgroup** » « **cpuset** » permet
 - de créer des groupes de processus associés aux jobs et jobsteps
 - d'associer à ces groupes les cores/cpus alloués
 - Idem pour la mémoire (« **memcg** »), les GPUs (« **devices** »), ...

Slurm - Configuration avancée

- **Binding des processus**
 - **Linux/glibc fournit des primitives permettant de limiter l'exécution de processus sur certains cpus du système uniquement (cf « man set_schedaffinity »)**
 - **Cette logique est utilisée dans task/cgroup pour découper les cores alloués entre les processus et garantir leur immobilité pendant l'exécution**
 - **Améliorant les effets de cache**

Slurm - Configuration avancée

- **Confinement et binding des processus**

- **Exemple de cgroup.conf**

- `CgroupMountpoint=/dev/cgroup`

- `CgroupAutomount=yes`

- `CgroupReleaseAgentDir="/etc/slurm/cgroup"`

- `ConstrainCores=yes`

- `TaskAffinity=yes`

- `ConstrainRAMSpace=yes`

Il n'y a plus qu'à mettre cela en place par
vous même...

Ce sera l'objet du TP 2