La grille de calcul du LHC - survol

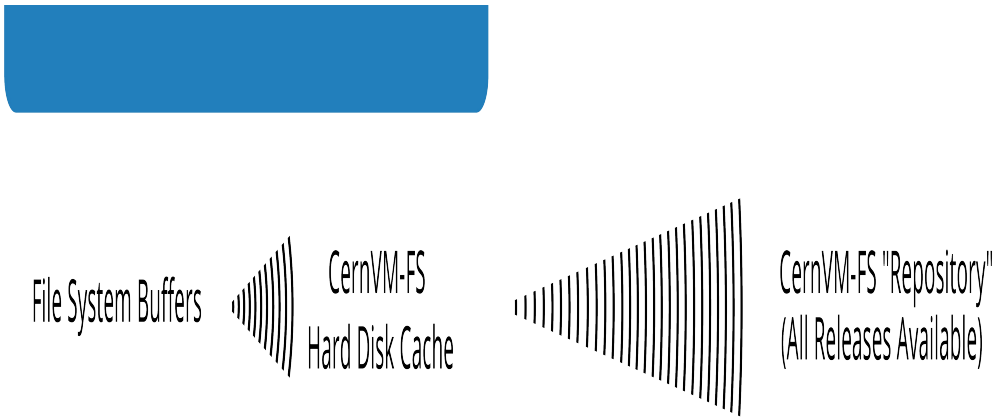
Il serait fastidieux de décrire toute l'architecture de la grille de calcul du LHC tant dans son infrastructure commune (grille européenne EGI, américaine OSG, ..) que dans l'utilisation qu'en font les expériences du LHC. Ici nous nous concentrerons sur les parties qui seraient en interaction directe avec un centre HPC ou un service intermédiaire. Nous décrirons d'abord les outils utilisés dans la situation d'un site grille « lambda » et nous préciserons ensuite quelles adaptations ont été faites pour faire face à la diversité des contraintes imposées par les centres HPC dans le monde.

Accès au software

Sur un site grille générique, le software général et le software spécifique des expériences est distribué grâce au filesystem readonly distribué appelé *CVMFS*[1] : le software est publié à un endroit (typiquement le CERN dans le cas des expériences LHC) sur CVMFS et alors disponible sur tous les sites via fuse-mount. Pour garder des performances acceptables, les sites installent des caches http, le plus souvent utilisant *squid*[2]. Le fait que les fichiers soient non modifiables permet que les fichiers en cache soient toujours valides et tels que mis en ligne centralement (sur une machine où le système n'est pas compromis).

Sur un cluster de grille typique, CVMFS est disponible sur le worker node et a un cache local disque sur la machine. Si CVMFS n'est pas disponible mais que les user namespace sont autorisés par le système alors cvmfsexec[3] permet d'accéder à CVMFS.

Si le code devant s'exécuter contient des modifications ou est développé sur la base des librairies distribuées par CVMFS, une archive (tar) du code compilé est attaché au job soumis.



[1] https://cvmfs.readthedocs.io/en/stable/
[2] https://www.squid-cache.org/
[3] https://github.com/cvmfs/cvmfsexec

Accès aux bases de données

Une partie des bases de données est quasi statique et est distribuée également par CVMFS. Une autre partie est dans des bases de données relationnelles au CERN qui sont accédées via des système de cache (*FRONTIER*[4]*, squid, varnish*[5]*).* Les accès base de données se font dans le code s'exécutant sur le worker node qui ont accès au WAN sur un site de grille typique.

Storage Element

À part les bases de données, tous les calculs sur la grille reposent sur des fichiers que l'on accède et/ou produit. Le Storage Element (SE) est un « middleware » qui forme une couche au dessus d'un système de stockage de fichiers (collection de serveurs disques, CEPH, lustre, gpfs, etc...) et présente une interface connue côté grille pour les opérations de base (get, put, rm,...). Les fichiers eux-même ne sont pas modifiables et leur intégrité est assurée par un checksum.

Les utilisateurs côté grille sont projetés sur des comptes utilisateurs locaux et leur droits sur les droits Unix de ces comptes.

Les protocoles présentés à l'extérieur étaient initialement assez spécifiques à la grille (*gridftp*, maintenant éteint) ou HEP (*xroot*) mais des protocoles plus largement utilisés comme WebDAV sont maintenant supportés.

Les deux technologies de Storage Element utilisé en France sont *EOS*[6] (GRIF, site distribué en île de France) et *dCache*[7] (CC-IN2P3 - Lyon, CPPM - Marseille, IPHC - Strasbourg, LAPP - Annecy, LPCA - Clermont-Ferrant).

Compute Element

Un Compute Element (CE) présente une interface unique[8] à des ressources de calcul, typiquement des fermes batch.

Côté grille les jobs sont soumis avec un Job Description Language décrivant les besoins (nombre de cœurs, durée, mémoire, dépendances en fichiers, software....).

---

[4] https://frontier.cern.ch/
[5] https://www.varnish-software.com/fr-fr/produits/varnish-cache/
[6] https://eos-web.web.cern.ch/eos-web/
[7] https://dcache.org/
[8] mais non standardisée, i.e. les différents CE ont une interface de soumission propre

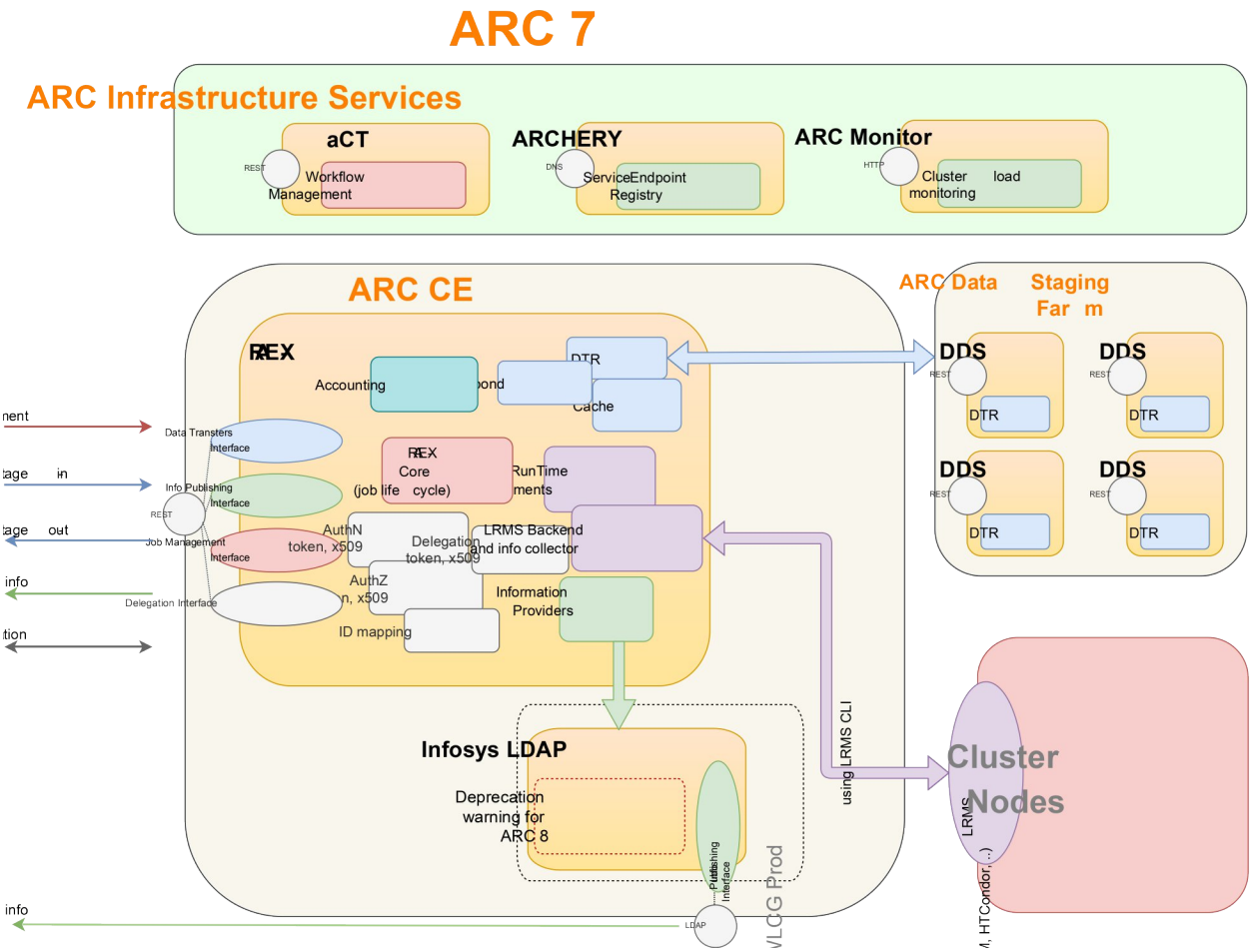
Côté ferme de calcul, le job est transformé en un job sur le système de batch cible (*SGE, LSF, PBS, Slurm, HTCondor*,...). Le Compute Element fait aussi le suivi et la comptabilité des jobs et la traçabilité des utilisateurs.

Pour les sites grille typiques, les expériences lancent des jobs pilotes qui vérifient l'environnement avant d'exécuter le code (mode push) voire dans certains cas qui demandent centralement un job à exécuter en fonction des caractéristiques du batch slot (mode pull).

Les Compute Element utilisés sur les site français sont HTCondor-CE[9] et ARC-CE[10].

Une grande partie des sites de grille sont « monolithiques » : le CE est co-localisé avec la ferme de calcul et avec le stockage. Ce n'était pas le cas de la « fédération nordique » composée de divers centres de calcul HPC et HTC dans différents pays, dont seulement certains avaient un stockage pérenne exposé à la grille avec un SE. Ils ont développé le ARC-CE qui comprend aussi une partie transfert de fichiers : l'ARC Control Tower examine le job pilote venant de l'expérience et soumet des jobs pleinement définis au ARC-CE qui détermine quels fichiers sont nécessaires. Ces fichiers sont écrits si besoin dans un espace cache sur un site avant que le job ne soit soumis au système de batch de ce site. Ainsi le job ne s'exécute que lorsque les fichiers sont déjà présents, cela évite que ce soit le job pilote qui copie des données. Cette possibilité a beaucoup été utilisée pour interfacer des sites HPC à la grille.

[9] https://htcondor.org/htcondor-ce/documentation/
[10] https://www.nordugrid.org/arc/about-arc.html



zimbCas d'utilisation de machines HPC

Les expériences ont deux grandes classes de jobs :

- Les jobs d'analyse au sens large tournent du code écrit directement par les physiciens et le plus génériquement possible. Il est difficile de savoir à l'avance si le code est adapté à un centre HPC donné. Ils ne sont pas l'objectif principal de l'utilisation de machines HPC par les expériences.
- Les jobs dit de production tournent le software officiel de l'expérience. Il existe divers workflows pour lesquels les besoins en I/O, connectivité et capacités d'utilisation en multicœur sont différents.

Une grande partie des jobs (e.g. par exemple les jobs de production et la majorité des jobs d'analyse de données pour l'expérience ATLAS) utilisent des containers singularity.

De nombreuses machines HPC en Europe et aux USA sont de nos jours utilisées par les expériences du LHC. Leur configuration hardware ne permet pas toujours d'utiliser la totalité des workflows des expériences (par ex limitation en I/O local, réseau, RAM, ...). Quasiment

chaque machine nécessite un effort particulier d'intégration, en effet les machines HPC sont très diverses dans leur configurations et leurs restrictions. Certaines ont CVMFS installé et le WAN accessible depuis les worker nodes, dans ce cas elle peuvent être accédées comme un site de grille. À l'opposé, d'autres n'ont pas installé CVMFS, n'autorisent pas les user namespaces et n'ont pas de connectivité au WAN des worker nodes....

Voici quelques exemple de manière d'utiliser des machines HPC de EuroHPC par une des expérience du LHC, ATLAS :

- Les machines HPC de la fédération nordique et la machine VEGA (Slovénie) ont été configurées pour être utilisables par la grille : ATLAS utilise les ARC-CE avec transfert de données ; CVMFS est disponible ; tout type d'exécution est a priori possible.
- KAROLINA et BARBORA (Tchéquie) n'autorisent que d'accès ssh au batch et au stockage. Les fichiers sont donc échangés avec un stockage grille via sshfs. Seule l'allocation de nœuds entiers est possible. CVMFS n'est pas installé mais les user namespaces sont autorisé et donc cvmfsexec est utilisé. Elles sont utilisées pour des jobs de production multicœur.
- LUMI (Finlande) n'autorise que des accès ssh au batch et au stockage. Les fichiers sont donc échangés avec un stockage grille via sshfs. CVMFS n'est pas installé et les user namespace non autorisés. ATLAS l'utilise avec des « gros » containers qui ont le code et les données nécessaires.
- MARENOSTRUM5 (Espagne) et SuperMUC (Allemagne) n'offrent pas de connectivité vers l'extérieure. Les fichiers sont donc échangés avec un stockage grille via scp. CVMFS n'est pas installé et les user namespace non autorisés. ATLAS l'utilise avec des « gros » containers pour le workflow de simulation qui n'a pas besoin d'accéder aux bases de données.

Plusieurs machines HPC aux USA sont aussi utilisées par ATLAS.

La situation pour les autres expériences est similaire, CMS étant aussi avancé qu'ATLAS.

Traçabilité et sécurité

Les utilisateurs ne soumettent jamais directement des jobs au CE : cela se passe toujours à travers des interfaces propres aux expériences. Cela permet de tracer les jobs et leur code à exécuter indépendamment de l'utilisateur. Il y a donc trace de ce que l'utilisateur exécute. Les CE n'ont pas besoin d'autoriser tous les utilisateurs à soumettre des jobs : seuls les comptes des opérations des expériences sont nécessaires. Les CE gardent trace des jobs exécutés. Il est donc parfaitement possible de tracer qui serait responsable de l'utilisateur de code malicieux.

Il existe une organisation de sécurité de la grille, chaque site ayant un (des) responsable(s) sécurité, plus une centralisation des alertes au niveau WLCG et EGI (OSG) qui opèrent un

MISP[11] et ont des officiers de sécurité en charge de la réponse à incident. Les administrateurs de site et les expériences ont la possibilité de suspendre ou bannir des utilisateurs. Des « challenges » sont régulièrement déclenchés pour tester la rapidité de réponse des sites à une alerte. À intervalles réguliers, des exercices de sécurité sont organisés dans lesquels un code malicieux (mais sans danger pour le site, comme du bitcoin mining ou du contact vers des sites pirates connus) est exécuté pour tester les capacités de détection et la rapidité de réponse.

[11] https://www.misp-project.org/