

Profiling, suivi et statistiques des jobs de calcul et des accès aux stockages distribués avec Elasticsearch et Grafana



La plateforme "Eli"

Bruno Bzezniak (GRICAD)
Oliver Henriot (GRICAD)

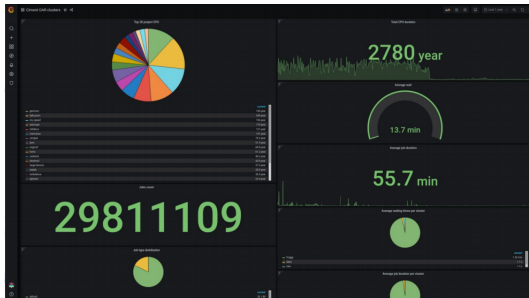
4 décembre 2020





- ▶ GRICAD fournit aux chercheurs du bassin Grenoblois des services et infrastructures pour le calcul et le traitement de données massives.
- ▶ 300 noeuds de calcul et une 30aine de noeuds de stockage
- ▶ 400 utilisateurs actifs
- ▶ Une grille de calcul best-effort: CiGri
- ▶ Parfois plus de 5000 jobs / heure
- ▶ Plateformes hétérogènes, au moins 3 systèmes de stockage différents
- ▶ Gestionnaire de ressources OAR

- ▶ Besoin de statistiques sur les jobs (nombre, fréquence, taille, heures de calcul par utilisateur, par projet,...)
- ▶ Avant 2017: stats obtenues par requetage de la base de données OAR: fastidieux et "one-shot"
- ▶ A partir de 2017: utilisation de la stack Elasticsearch ELK (Elastic, Logstash, Kibana)
- ▶ Depuis 2019, passage à ELG (Elastic, Logstash, Grafana)



Ca chauffe sur le stockage!



- ▶ Baisses de performance constatées, souvent sur les systèmes de stockage
- ▶ Chaque système propose des métriques et un monitoring propre
- ▶ Le flux d'informations important
- ▶ On n'a pas toujours accès au "passé" et lorsqu'un problème est détecté, il est souvent trop tard pour faire un diagnostic



Le besoin de supervision et de diagnostic des baisses de performance s'apparente à un problème de big-data et de fouille de données. L'idée d'utiliser un serveur Elasticsearch et d'y injecter les données de supervision de nos systèmes est apparue naturellement dès que les premières baisses de performance de notre plateforme "Bettik" ont été rencontrées en 2019.

Monitoring des volumes BeeGFS





- ▶ Colmet collecte des informations systèmes, toutes les 5 secondes pour chaque job (un client par noeud de calcul)
- ▶ Les infos sont centralisées, via ZeroMQ, par un collecteur
- ▶ Jusqu'en 2020: stockage dans des fichiers HDF5 et interrogation via python/Matplotlib: plantages fréquents dû au temps d'extraction des données
- ▶ A partir de 2020, Colmet intègre nativement un collecteur pour Elasticsearch
- ▶ <https://github.com/oar-team/colmet>



OAR, Colmet, BeeGFS... Le flux de toutes ces informations réunies peut être très important: jusqu'à 10000 métriques par seconde! En outre, nous aimerions corréler les informations issues de colmet et celles de la supervision des systèmes de stockage.

- ▶ Cela ne tourne plus sur une seule VM!
- ▶ Mise en place d'un cluster de 3 serveurs "recyclés" + une VM pour la partie Grafana
- ▶ Nous allons bientôt mettre en place 3 nouveaux serveurs dédiés (investissement de 8.5k€ / nœud)
- ▶ C'est la naissance de la plateforme "**ELI**"

Et si on indexait tous les fichiers?!



La plateforme et les compétences étant là, de nouvelles idées apparaissent



Et si on indexait tous les fichiers?!



https://github.com/bzizou/fs_walk

README.md



fswalk

An efficient multiprocessing directory walk and search tool

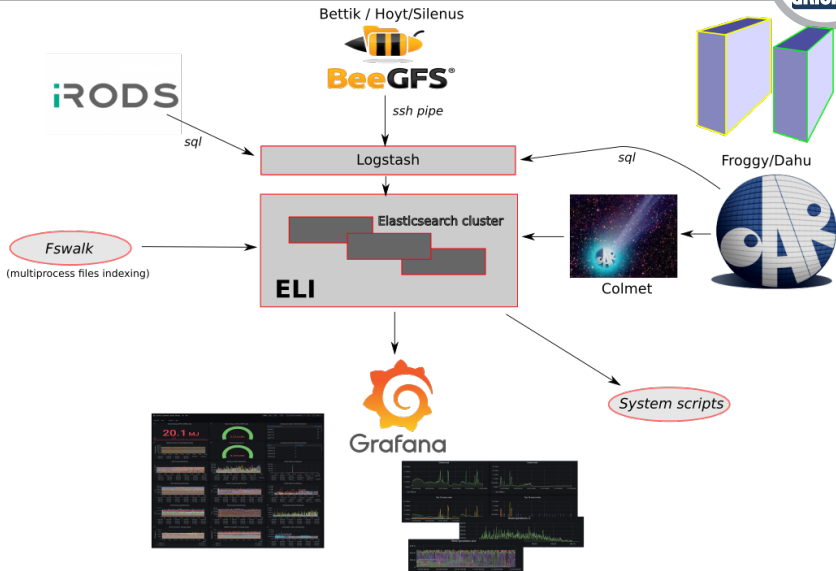
Introduction

fswalk is a simple python script that recursively walks through a filesystem directory to gather files meta-data and collect them into a **json file** or an **Elasticsearch** database. It runs several processes, each responsible of doing the list of the files contained into a subdirectory. Collected meta-data are `filename`, `path`, `uid`, `gid`, `size` and `atime`. The output is either a json file sent on the fly to stdout, or an Elastisearch indexing. A simple search option is provided to retrieve files by their owner, group or a part of the name.

The script also provides an option to do a quick analyze of the resulting output file.

warning: When the results are sent to stdout, due to multiprocessing and not to slow down the thing, the json file is printed with an extra `,` sign that might break json compatibility. The `pyjson5` python library allows such non-standard json file to be read.

Installation



Et si ça pouvait servir à nos utilisateurs?



"Ingestion d'un fichier de séquences ADN pour recherche d'intervalles (1.7 milliard de lignes)"

Sur la base d'un tel cas test, nous prévoyons de "Kubernetes-iser" notre plateforme Eli afin d'offrir des slice ELG temporaires pour ce genre de problèmes.



- ▶ Depuis 2016, l'équipe système de GRICAD se penche sur l'indexation Elasticsearch de ses données de stats et de supervision
- ▶ Courant 2020, GRICAD a commencé à mettre en place une solution unifiée pour la supervision et les statistiques de l'usage de ses plate-formes.
- ▶ D'abord destiné aux administrateurs, l'accès sera prochainement ouvert aux utilisateurs, en particulier pour ce qui concerne le profiling système des jobs.
- ▶ Les outils utilisés sont pour la plupart très classiques: Elastisearch [1] pour l'indexation des métriques, Grafana [2] pour la visualisation; Pour ce qui concerne la collecte, nous utilisons Logstash [3] souvent interfacé avec des scripts via SSH, ou OAR/colmet [4] pour la collecte des informations système associées à chaque job de calcul, ou encore des outils comme fswalk [5] pour indexer les fichiers.
- ▶ Nous envisageons de fournir l'accès à ces outils à nos utilisateurs pour des problèmes scientifiques particuliers

- [1] <https://www.elastic.co/guide/index.html>
- [2] <https://www.elastic.co/guide/en/logstash/current/index.html>
- [3] <https://grafana.com/>
- [4] <https://github.com/oar-team/colmet>
- [5] https://github.com/bzizou/fs_walk