ITÉRATION 3

Un peu plus de fichiers...

Modalités

- Travail individuel en autonomie puis en groupe
- 1 journée en présentiel

Livrables

- Jobs adaptés et lancés sur un cluster spark

Objectifs

- Utilisation de spark sur des jeux de données plus conséquents
- Utiliser des machines distantes
- Voir quelques optimisations

Compétences

- Utiliser Spark sur des données volumineuses

3.0 — Stan est de passage

5m — Présentiel

Après votre montée en compétences et l'exécution de vos premiers jobs, dont l'alerte a réveillé Henri pendant ses congés sous la pluie (et oui comme tout administrateur système il sait TOUT ce qu'il se passe sur nos machines,...). Celui-ci s'est empressé d'ajouter un ticket dans le backlog de notre cher Stan sur Jira.

Stan déboule alors dans votre bureau avant votre daily du mercredi, quel mépris des traditions et vous demande de lui rendre des comptes. Vous lui montrez les quelques pipelines, votre équipe lui semble déjà plus confiante qu'au début de la semaine, Youpie!

Il regarde ensuite votre jeu de données et la… vous apercevez la désillusion dans son regard, il ne s'attendait pas à un livre aussi beau! Mais cela ne s'arrêtera pas là, il vous demande maintenant d'en utiliser d'autres. Des bruits de pas et des voix retentissent dans le couloir… c'est Sonia et Léon de la compta qui passent dans le couloir; Sonia explique à Léon comment elle a eu accès à plein de livre de façon légale en utilisant l'Ordinateur, vous vous dites "Banco, c'est ce que Stan veut!"

Suite à une discussion et une pause café bien méritée avec les gens de la compta, vous revenez dans votre open space réchauffé par l'équipe et vous leur présentez le lien que Sonia vous a donné contenant environ quelques milliers de livres.

Et … vous recevez toutes et tous un message d'Henri :

> *Je vais déployer vos machines pour un cluster Spark, mais pour cela, j'ai besoin de vos clés SSH (publiques !). Remplissez le fichier sur le disque partagé pour que je puisse le faire quand j'aurai 5 minutes. Rien ne sera installé sur ces machines, c'est à vous de jouer !*

RESSOURCES

- Le fichier de Sonia est sur le drive.

COMPÉTENCES ASSOCIÉES

- Écoute intra-service
- Daily buissonnier

3.1 — Ajout des fichiers de Sonia

1h — Présentiel

Maintenant vous êtes prêt à impressionner Stan, Maddie et la direction, vous vous lancez:

- Téléchargez le fichier du lien de Sonia
- Utilisez la commande `tar` pour décompresser le fichier
  (*mnémotechnique: les paramètres suivant* `tar` *sont* `xf` *comme "extract file"*
  vous pouvez rajouter l'option `v` *pour afficher plus d'informations)*
- Combien de fichiers y a t'il dans le jeu de données de sonia?

RESSOURCES

- La commande tar: https://www.geeksforgeeks.org/tar-command-linux-examples/
- Fichier de Sonia: bookcorpus.tar.bz2

3.2 — Adaptation du pipeline existant

2h — Présentiel

Les données sont prêtes, le job que vous avez créé sur des textes dans l'itération 2 va devoir être adapté pour prendre en compte plusieurs fichiers.

- Adaptez ce job pour qu'ils prennent en entrée tous les livres du fichier de Sonia.
  - Commencez par trouver les thématiques du corpus complet
  - (bonus) Essayez de découper le corpus par livre.
- Explorez les RDDs de ce job
  - Combien de partitions contiennent les RDD?
  - Quelle est la taille d'une partition?
- Lancez le job.
  - Combien de temps prend t'il pour s'exécuter?
  - En utilisant l'interface web et en découpant votre code, quelles parties sont les plus chronophages?

RESSOURCES

- Documentation spark des RDD https://spark.apache.org/docs/latest/rdd-programming-guide.html

---

3.3 — L'union fait la force (partie 1)

1h — Présentiel

Une optimisation simple à faire lorsque l'on utilise Apache Spark est d'utiliser plus de machines, pour cela il va falloir installer spark sur les différents postes et créer ce que l'on appelle un cluster de calcul.

Nous utiliserons les machines du datacenter déployées par Henri !

Pour cela, un peu de Linux et de configuration sera nécessaire:

- Télécharger la version de spark adéquate (probablement la dernière version)
- Décompressez la dans un dossier que vous nommerez comme vous le souhaitez (utilisez un nom simple, car il va falloir créer des variables d'environnement)
- Créer les variables d'environnement proposées dans le tutoriel.
- Pour vérifier l'installation rapidement, essayez de lancer le `spark-shell` (pas besoin de faire grand chose avec, sauf si vous souhaitez essayer un peu de scala...) et vérifiez que l'interface web de spark est bien lancée

RESSOURCES

- Page de téléchargement de spark https://spark.apache.org/downloads.html
- Variable d'environnement https://www.it-connect.fr/definir-des-variables-denvironnement-sous-linux%EF%BB%BF/
- Tutoriel un peu vieux d'installation de spark https://computingforgeeks.com/how-to-install-apache-spark-on-ubuntu-debian/?utm_content=cmp-true

---

3.3 — L'union fait la force (partie 2)

3h — Présentiel

Pour créer un cluster spark, vous aurez besoin de 2 types de machines:

- une ou plusieurs machine de type **driver** (ou master) auxquelles on soumet les jobs et qui seront chargées de répartir ces jobs sur le second type de machine. (dans certains cas le driver peut aussi faire des calculs)
- plusieurs machines de type **worker** (ou slave) qui sont chargées de faire les calculs, celles-ci vont contenir l'adresse du driver afin de s'y connecter.

Lorsque vous utilisez pyspark, vous êtes dans un cas dit "standalone" où le driver fait aussi office de worker.

Avant de continuer, regardez comment créer un single node cluster, c'est-à-dire d'avoir le worker et le driver sur la même machine physique.

NOTE: La machine a une adresse publique (celle à laquelle vous vous connectez via SSH) et des adresses privées accessibles depuis les autres machines du cluster (192.168.x.x). Utilisez l'adresse locale pour configurer le cluster. Sinon, n'importe qui depuis Internet pourra soumettre des tâches !

Dans cet exemple, 51.68.13.129 est l'adresse publique et 192.168.4.157 est l'adresse privée.

```
$ ip addr list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
        link/ether fa:16:3e:82:de:8d brd ff:ff:ff:ff:ff:ff
        altname enp0s3
        inet 51.68.13.129/32 metric 100 scope global dynamic ens3
        valid_lft 67647sec preferred_lft 67647sec
        inet6 2001:41d0:304:400::25cc/128 scope global
        valid_lft forever preferred_lft forever
        inet6 fe80::f816:3eff:fe82:de8d/64 scope link
        valid_lft forever preferred_lft forever
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
        link/ether fa:16:3e:39:ca:f2 brd ff:ff:ff:ff:ff:ff
        altname enp0s4
        inet 192.168.4.157/24 metric 100 brd 192.168.4.255 scope global dynamic ens4
        valid_lft 77374sec preferred_lft 77374sec
        inet6 fe80::f816:3eff:fe39:caf2/64 scope link
        valid_lft forever preferred_lft forever

$ export SPARK_LOCAL_IP="192.168.4.157"

$ python3
Python 3.10.12 (main, May 27 2025, 17:12:29) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from pyspark import SparkContext
>>> sc = SparkContext("local", "test")
>>> sc.stop()
>>> quit()
```

Maintenant passez en groupes.

- Sur une des machines, lancez le spark-driver
- Sur les autres lancez des spark-worker et liez les au driver

Pour cela vous aurez besoin d'un peu de réseau:

- Renseignez vous brièvement sur ce qu'est une adresse IP
- Faites de même pour les ports logiciels.

Pour finir essayer de changer les drivers et worker dans votre îlot!

RESSOURCES

- Un peu de doc sur le réseau: https://linuxhandbook.com/find-ip-address/
- Deux tutoriels sur l'installation de spark en cluster :
  https://medium.com/@sandeepsinh/spark-installation-on-single-node-7e4cf4514c29
  https://phoenixnap.com/kb/install-spark-on-ubuntu
- Une ressource générique orientée sur les ports:
  https://www.cloudflare.com/learning/network-layer/what-is-a-computer-port/
- Documentation de la commande ss (permet de voir les ports utilisés):
  https://phoenixnap.com/kb/ss-command

3.3 — Un peu de Machine Learning (bonus)

∞h — Présentiel

Vous êtes arrivés au bout de cette itération?

Félicitations! Maintenant vous pouvez utiliser les ressources pour découvrir la partie orientée Machine Learning de spark pour créer un modèle sur le problème de votre choix!

Vous pouvez même profiter des autres machines pour augmenter votre puissance de calcul…

RESSOURCES

- https://spark.apache.org/docs/latest/api/python/reference/pyspark.mllib.html

---

Livrables

À la fin de cette itération vous devez avoir

➔ Le job adapté sur les quelques livres

➔ Le résultat du job lancé sur le cluster
➔ Un cluster Spark fonctionnel