Consignes mail Hernandez

> Dans le premier quart de votre stage (avant le vendredi 28 avril), il vous est demandé d'envoyer par mail à votre tuteur
>
> - un résumé du sujet de stage (une douzaine de lignes) écrits avec vos propres mots. Ce résumé permet d'informer votre tuteur d'adaptations éventuelles du sujet original mais aussi d'estimer votre compréhension du sujet.
> - Et un point sur le déroulement du stage (premières impressions, fréquence de rencontre avec les personnes qui vous encadrent, résumé des tâches accomplies les semaines passées, perspectives des tâches la/les semaines à venir...).

Résumé

Copié/collé du sujet

> Ce stage vise à construire une base de données géo-référencée, la plus complète possible, sur le territoire et de développer les outils nécessaires pour son interrogation et l'extraction d'informations utiles. Les technologies utilisées seront python et SQL.

Résumé de ma compréhension du sujet de stage

Le but final est d'implémenter une base de données géo-référencées. Celle-ci permettra d'absorber les données provenant de sources diverses, dont chacune a ses propres conventions de standard de représentation et de découpage géographique. Je devrais porter une attention à l'important volume des données et privilégier les traitements en flux.

En premier lieu, je dois développer un script bash (non interactif) qui va se charger d'installer postgresql et l'extension postgis. Puis, ce script doit se charger de modifier les configurations par défaut de postgres qui ne sont pas satisfaisantes, comme les règles d'accès, les logs, ou encore la création de rôles de lecture et d'écriture desquels les utilisateurs nouvellement créés pourront hériter. Ainsi, après exécution du script, le serveur doit être prêt à accepter les connexions distantes. On peut ensuite s'y connecter avec n'importe quel logiciel client postgres ; sous réserve de se connecter avec mot de passe en tant que rôle enregistré préalablement (et manuellement) dans la base.

En deuxième lieu, je dois développer en python du code qui va aller chercher ces données en ligne, puis les importer dans la base. Cependant il sera potentiellement préférable de passer par un format intermédiaire (JSON). Ainsi la base n'aurait qu'à supporter ce format en import. Le code python se chargerait alors de lire (parser) correctement les données téléchargées en entrée. En fonction du format, il construirait l'équivalent JSON dont la structure correspondra à celle de la base. L'avantage d'adopter cette procédure serait double. D'une part obtenir des données utilisables avant même de leur import au sein de la base. D'autre part l'utilisation de bibliothèques déjà existantes de conversion d'un format de stockage de données géographiques vers du GeoJSON. Bien entendu, ce n'est qu'une possibilité c'est pourquoi nous en débattrons le moment venu, au fur et à mesure de l'avancement et des confrontations aux problématiques d'implémentation. De plus, il faudra gérer avant l'import le nettoyage des données : vérifications sur le typage, gestion des champs vides, etc.

En troisième lieu, je devrais mettre à disposition un ensemble d'accesseurs. Pour ce point, je devrais m'interroger sur la structure de données intermédiaire en mémoire à utiliser. Je pourrais utiliser l'ORM (Object Relationnal Mapping) SQLAlchemy afin de créer un ensemble de requêtes permettant de sélectionner des sous-ensembles de données en fonction de certains critères. Ces critères peuvent être sur les valeurs des attributs ou porter sur l'emprise géographique par exemple. On peut également imaginer des formats de retours en GeoJSON.

La réalisation de cette base de données va permettre :

- Un stockage massif performant de données
- De faciliter grandement le partage et l'accès aux données (avoir un unique emplacement de ressources, multi-utilisateur). En effet, le partage de données se fait actuellement via des fichiers multiples (une même donnée est répartie sur plusieurs fichiers). L'intercompatibilité de ces formats divers est hasardeuse.
- De s'épargner la recherche de liens géographiques dès qu'il s'agit de recroiser des données à traiter puisque le découpage sera déjà présent via la structure fine de la base.
- De solliciter des requêtes sur la base: elles offriront un premier filtrage basique "de masse" en amont afin de travailler avec des données plus spécifiques et de volume approprié en aval.

Ressenti et avancement

Impressions

En ce qui concerne mes premières impressions je suis très bien accueilli et je me sens très à l'aise à l'IFSTTAR. Je vois Pierre, Pascal, et Marc (en formation adulte, nous partageons la même pièce) tout au long de la journée et chacun n'hésite pas à se déplacer pour éclaircir ou préciser un point. Le midi est l'occasion de parler technique ou de dériver sur d'autres sujets. J'ai un poste de travail agréable et adapté.

Tâches

Comme je vous l'ai dit par mail plus tôt, je donnerai mon avancement chaque semaine afin que vous puissiez suivre le projet. Cependant cela ne se substitue pas à une vision à plus long terme. Bien que similaire à la partie précédente, j'ai trouvé nécessaire de remettre l'ensemble des tâches ci-dessous. En effet, le découpage séquentiel en liste présente l'intérêt de pouvoir faire correspondre les tâches à leur ordre chronologique de réalisation.

Les tâches réalisées

- m'appréhender du sujet et discuter des : aboutissements/de l'intérêt, objectifs, technos, et outils
- **[bash]** script d'installation et de configuration d'une base de données postgres+postgis
- comprendre le modèle conceptuel et physique de données de découpage géographique réalisé par Marc. en discuter, le comprendre, l'affiner si besoin est

Les tâches en cours

- **[python]** réalisation du squelette/structure du projet : comment organiser les modules, niveau de découpage du code, quelles classes créer, fonctions utilitaires (logs)

Les tâches restantes (loin d'être équivalentes en termes d'envergure, cependant chacune a son importance), présentées chronologiquement

- rédaction du rapport de stage, à commencer en parallèle des tâches ci-dessous

- **[bash]** lancement du script sur le serveur "de production" (machine virtuelle distante, en attente de disponibilité) et vérifications
- rédaction de documentations et de tests, en parallèle de l'avancement du code
- **[python]** fonction(s) pour venir implémenter le modèle conçu par Marc : requêtes de création des tables et relations
- **[python]** import de fichiers GeoJSON dans la base
- **[python]** téléchargement automatique de fichiers de sources diverses : INSEE, data.gouv.fr, open-data, IGN...). Les structures de ces données ont été explorées par Marc afin de concevoir le modèle de la base, et les sources/liens de téléchargement conservés. Mon rôle inclut le téléchargement automatisé de ces ressources : décompresser puis supprimer ce qui ne nous intéresse pas.
- comparaison des bibliothèques disponibles pour les conversions
- **[python]** fonctions de passage de types de fichiers contenant des données géo-référencées (.shape, .dbf, et de nombreux autres formats) en fichiers GeoJSON prêts à être importés. un exemple : "shape_ign_2_geojson()"
- **[python]** discussion des besoins puis création d'un ensemble de requêtes d'accès (utilisation potentielle de l'ORM SQLAlchemy)