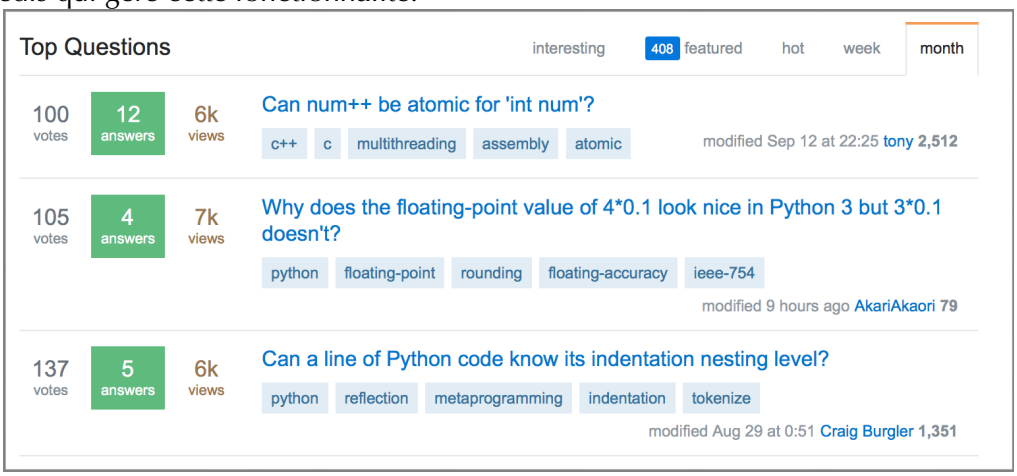
FC – Architectures Big Data

Mars 2024

1

De plus en plus de sites Web offrent une fonction de vote (par exemple, StackOverflow ci-dessous) sur les articles. En prenant en compte les votes, les articles (ou les commentaires) sont hiérarchisés et affichés en fonction de leur score. Vous allez devoir construire une base Redis qui gère cette fonctionnalité.



Pour représenter un article, la structure envisagée est la suivante:

- un ID (par exemple articles:3),
- un titre,
- un timestamp,
- un lien,
- un utilisateur,
- le nombre de votes.

Une instance d'article pourrait être:

- articles:72
- titre: Redis pour les nuls
- timestamp: 1380886601
- lien: http://www.foo.org/articles/1
- utilisateur: utilisateurs:23
- votes: 14

a. quelle structure de Redis vous paraît être la plus appropriée ? Justifiez votre réponse.

Implanter en Python une méthode qui permette de stocker les articles dans la structure choisie.
Implanter également une méthode permettant de récupérer les articles stockés.

On veut également pouvoir accéder aux articles soit de manière séquentielle (une timeline des articles), soit en fonction de leur score.

b. quelles structures de Redis vous paraissent les plus adaptées pour ces deux besoins ?

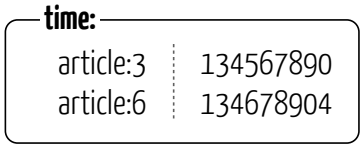
On veut aussi connaître les utilisateurs qui ont voté pour un article.

c. même question que précédemment concernant la structure.

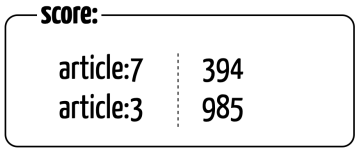
2

On vous demande de créer une méthode Python qui permette de gérer simultanément deux deux structures (vous choisirez les types les mieux adaptés) afin de stocker l'ordre de publication des articles et les scores.
Autrement dit, on veut avoir:

- selon le temps, la liste des articles



- selon le score, la liste des articles



On fera les hypothèses suivantes:

- le score est initialisé à 197 au moment où l'article est créé, puis on ajoute ensuite 197 pour chaque vote.
- pour éviter qu'un utilisateur ne vote plusieurs fois, on maintient une structure qui permet de savoir qui a voté pour un article donné. Afin de préserver la mémoire, cette structure n'est conservé qu'une semaine car on fait l'hypothèse que l'on ne peut voter pour un article que pendant une semaine.

On veut connaître, pour un article donné, la liste des utilisateurs qui ont voté pour cet article. Écrire la méthode qui permet de réaliser cette fonctionnalité.

Vous devrez également écrire les méthodes qui permettent de réaliser les fonctionnalités suivantes:

1. récupérer la liste de tous les articles,
2. écrire une méthode permettant d'augmenter le nombre de vote d'un article (rappel: un utilisateur ne peut voter qu'un seule fois pour un article donné),
3. écrire une méthode qui permet de retourner les 10 articles avec le score le plus important.

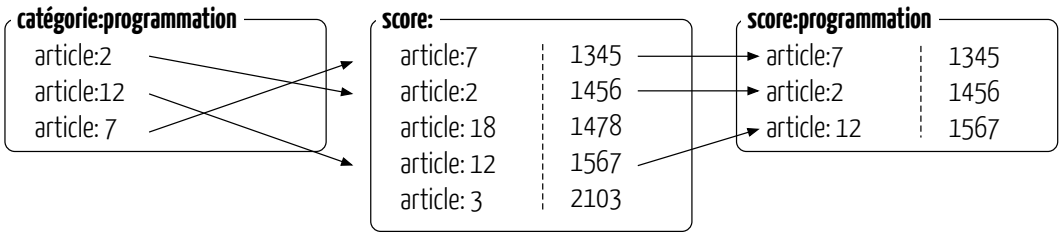
3

Question bonus: jusqu'à maintenant, on a considéré une méthode qui permet de récupérer des scores sur tous les articles, mais souvent, les articles sont regroupés dans des catégories (Java, Scala, Python, Redis,…).

La gestion des catégories implique 2 modifications:

- ajout d'une information qui permette de savoir à quelle catégorie appartient un article,
- récupérer les articles d'une catégorie donnée et leurs scores.

Écrire une méthode qui permette d'ajouter ou de supprimer une catégorie à un article, et à ajouter un article dans un groupe.
Écrire une méthode qui permette d'obtenir les scores des articles d'une catégorie donnée.



Vous rendrez votre travail en envoyant un message à olivier.perrin@loria.fr avec comme objet **[FC] nom**, et un **readme.txt** répondant aux questions, ainsi que les **sources Python** de votre projet.