Decision Process

First, I want to know if it is possible to use the current RDBMS to query about the desired statistics. Maybe tuning properly the data would be enough, using indexes and optimized queries, if the queries are taking too much time.

After that, if it isn't enough because of the time consumption or query complexity, I need to know about the database size. Only if it is several terabytes I would consider Hadoop/Mapreduce as a good ideia. Otherwise I would consider other solutions, as NoSQL databases or even a Java code to process what I need.

Search Functionality

We can use some NPL algorithms to filter very common words before mapping it, as a list of stop words. If we use an appropriate list of stop words, our index will became more readable and useful.

Consider we mapped our entries as

WORD\tFORUM_ENTRY

where *WORD* is a given word that will be indexed and *FORUM_ENTRY* is forum entry identifier. In this approach, we can use a combiner to reduce the data transferred to the reducers. So, a combiner will receive the mapped entry and print the word and a list of forum entries. The reducer will receive entries ordered by word and it will merge all lists of forum entries as sets and ordering before printing.

Concluding, we can reduce the index size using stop words or other technique to remove very common words. Also, using a combiner, we can reduce the amount of data been transferred in the network and optimize our reducers execution if we implement it wisely.

Other Questions

We can explore more questions about tags as, i.e., what is the probability about a question been answered given a tag. Also, we can query about a question's average response time given a tag and the hour that it was added.

Also, some dataset properties weren't used, as score. Using filtering patterns we can query what are the top 10 questions by score and what are the top 10 tags given the sum of its posts score.