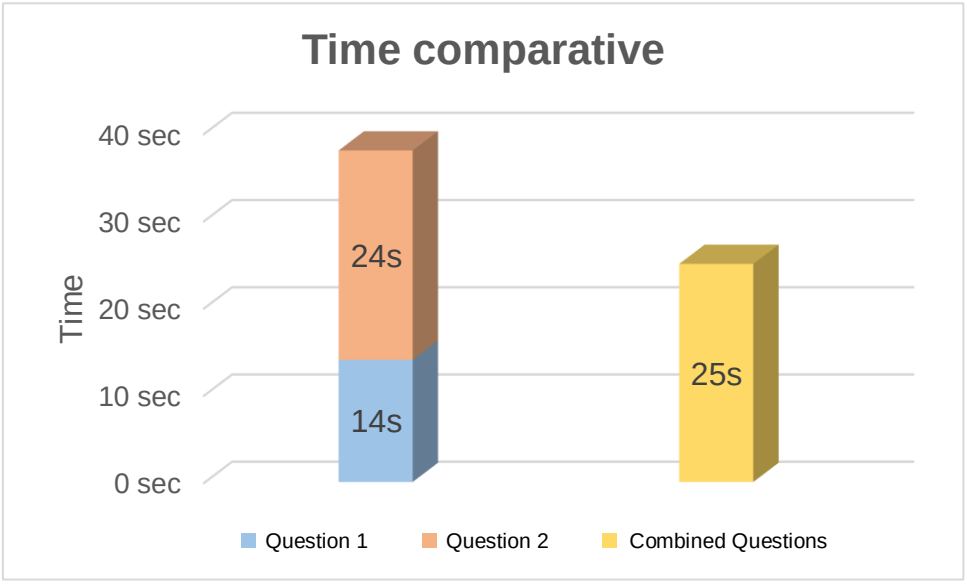
Report

In order to carry out this exercise, it is necessary to assume that the fields required to answer the two questions posed are *card, section and amount*. Specifically, for the first question it is only necessary to use *card* and *amount*, being *card* the key, because it is the field that identifies the *amount*. With respect to the second one, it would be necessary to use all three fields, having to choose the *card* as the key, since the reducer receives it in alphabetical order and can focus on locating the *section* with the largest *amount* based on the corresponding *card*. On the other hand, if the *section* is the key, you should go through all the tuples before knowing which one has the highest *amount* based on the *card*. This would mean keeping track of the *amount* in each *card* per *section*, instead of only the *amount* per *section* of the current *card*.

Based on this assumption, two types of implementations have been made, a mapper/reducer per question and a joint mapper/reducer (possible due to the fact that they share the same key that is *card*), where the time comparison will allow to see if the calculation is compensated separately or not. The expected output should be the joint implementation as it saves duplication of data routing and sorting operations. The following is a small graphic where you can see the time comparison.



Note: The start and end times provided by hadoop at the time the program was launched have been taken as the time.

The results obtained mean that performing the implementations separately saves 15 seconds of execution time compared to the joint program. As it has been assumed before, the accomplishment of shared tasks, like for example the ordering, suppose a great cost of time and it is observed that when making the joint implementation that the time only increases a second with respect to the individual execution of the second question.