

Search Server Requirements and the CAP Theorem

In looking at our search server requirements, I have identified requirements which are in conflict with Brewer's famous [CAP theorem](#). The CAP theorem elucidates three elements of a distributed system which cannot all be present. These elements are consistency, availability, and partition tolerance. I have found a solution which abides by the realities of the CAP theorem while still meeting our requirements.

To begin, here are the search requirements relevant to the CAP theorem:

1. **Consistency** - Searches should always reflect the latest data in the database.
2. **Availability** - Database transactions should always succeed regardless of whether the data has been indexed to the search server.
3. **Partition Tolerance** - the search server and the database are on separate machines.

According to the CAP theorem, **you must pick only two of these**.

Which of these can we sacrifice?

Clearly, **we cannot sacrifice partition tolerance**. Since a search server is an external system with large memory requirements, it should not be run on the same machine as the database.

We also **cannot sacrifice availability**. Search indexing is not as mission critical as a database write. Also, search indexing introduces a small amount of latency. For these reasons, database transactions should be independent of search indexing. If indexing fails for any reason, the update transaction should still succeed.

This means we will have to accept some tradeoff around consistency. Fortunately, we can still guarantee consistency at the application level in certain places where it is needed. This is known as **Eventual Consistency**. We will introduce a **Global Consistency Level** into our search index which will allow us to enforce consistency when it is needed.

How can the application enforce consistency when it is needed?

1. Search indexing will be asynchronous. This will allow our database transactions to succeed whether or not indexing is complete.
2. After data has been changed in the database, the application will store the `change_date` field of the newly updated row (in the session or in a hidden form field) so that it can

enforce the desired level of consistency in subsequent searches. This value will be known as the **Global Consistency Level**.

3. A **Search Replication Listener** will listen for updated rows in MySQL. It will then index these rows, and simultaneously update the Global Consistency Level in the search server with the `change_date` column of the newly updated row.
4. Future searches will return the updated Global Consistency Level as metadata along with the search results.
5. For certain operations where a known level consistency is needed (such as a search after an update), the application will check the search results to ensure that the Global Consistency Level is at least as high as the last write.
6. If the consistency level is too low, the application may retry the search until it achieves the desired consistency level. If the consistency level is not adequate after 5 seconds, it should be considered an error and operations should be alerted.

Frequently Anticipated Questions

Eventually consistent? Does this mean the search server will lose data or become out of sync?

No! Updated data will always be indexed -- *eventually*. **It is up to the application to wait for the consistency level it needs for a given operation.** Most searches won't need to worry about the consistency level. However, when the user updates data, the application will need to check the consistency level of the next search to ensure that the search is not returning stale results. The application should retry the search periodically until the search becomes consistent.

Does this mean searching will be unreliable?

No. Having the absolute latest data searchable is generally not an issue to the user. There are certain cases, such as searching after an update, where we must guarantee that the latest data is searchable. In these cases, we will need to check the Global Consistency Level of the returned results to make sure we have the latest updates. In 99 percent of cases, the data will be up-to-date. If the search does not become consistent within a couple of seconds, there is an error with the search server, and an e-mail should be sent to operations team to alert them that the sync process is offline.

