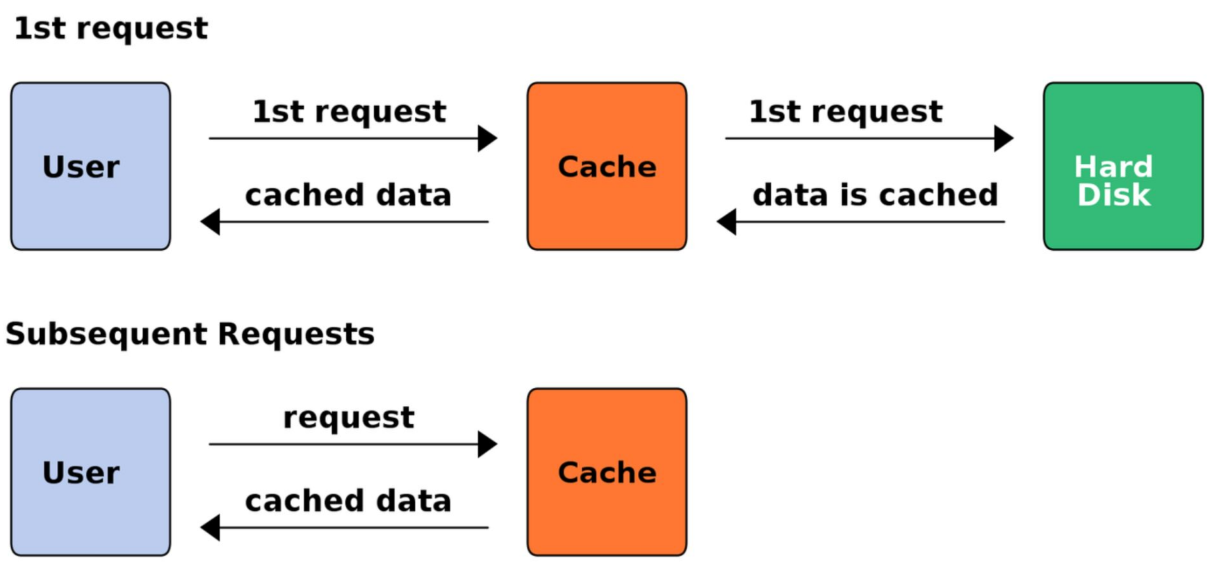
Caching

→ A cache is a high-speed data storage layer, which stores a subset of data so that future requests for that data are served up faster than is possible by accessing the data's primary storage location.

→ Caching allows to efficiently reuse previously retrieved or computed data.



→ A cache's primary purpose is to increase data retrieval performance by reducing the need to access the underlying slower storage layer.

→ Cache Operations are based on the principle of locality of reference.

Locality of Reference principle : Recently requested data is likely to be requested again

Santosh kumar Mishra

CACHE INVALIDATION

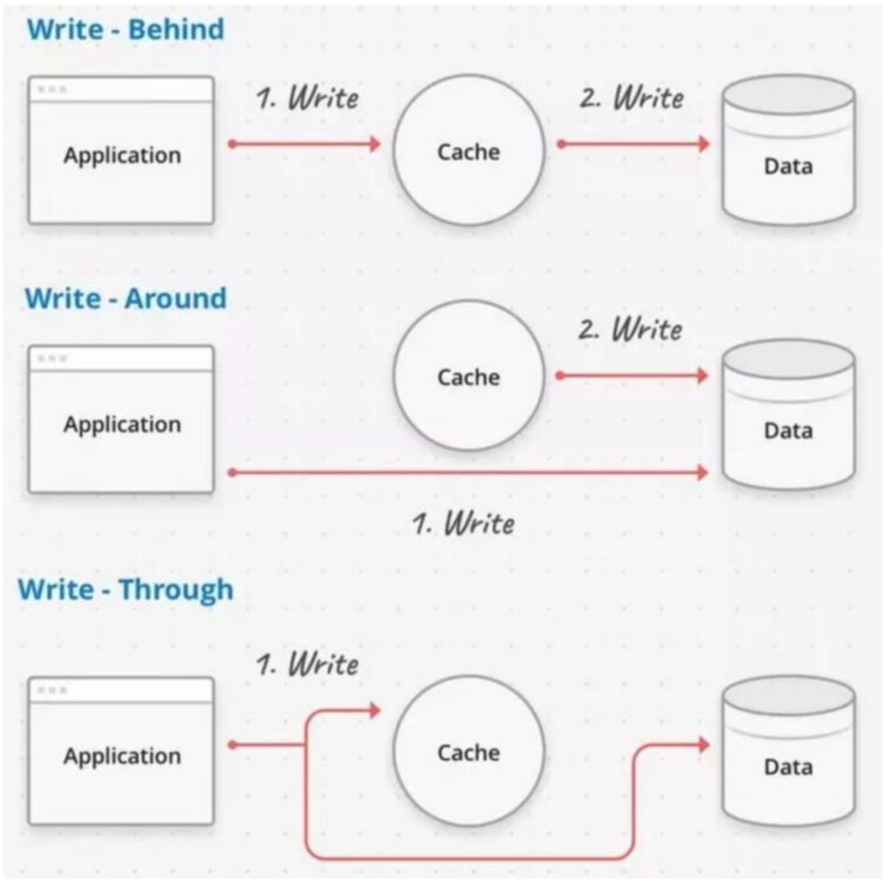
→ Caching does require maintenance for keeping cache coherent with the source of data(e.g database).

→ Every read operation should receive the most recent write.

→ Hence if data is modified in the database then it should be invalidated in cache to keep it consistent.

This process is known as cache invalidation.

The three cache invalidation schemes :



Santosh kumar Mishra

→ Write-through cache: In this scheme,the cache and database are updated at the same time.

This provides complete data consistency and minimizes the risk of data loss.

However, every write operation should be updated in cache as well as db resulting in higher latency.

→ Write-around cache: In this scheme, data is written directly to permanent storage, bypassing the cache.

This reduces the cache being flooded with write operations.

However, the subsequent read requests will face "cache-miss" and must read from slower back-end storage.

→ Write-back cache: In this scheme, data is written to cache alone and the permanent storage is updated at specified intervals.

This results in low latency and high throughput.

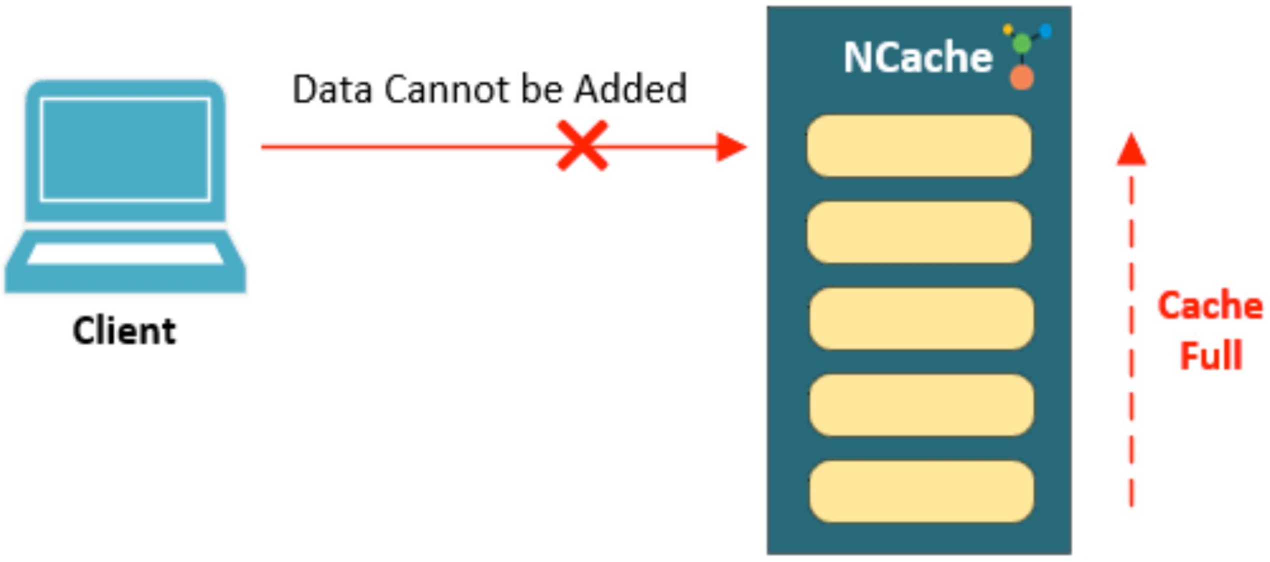
However,there is high risk of data loss in case of crash or intermittent losses since the only copy of data was in cache.

CACHE EVICTION:

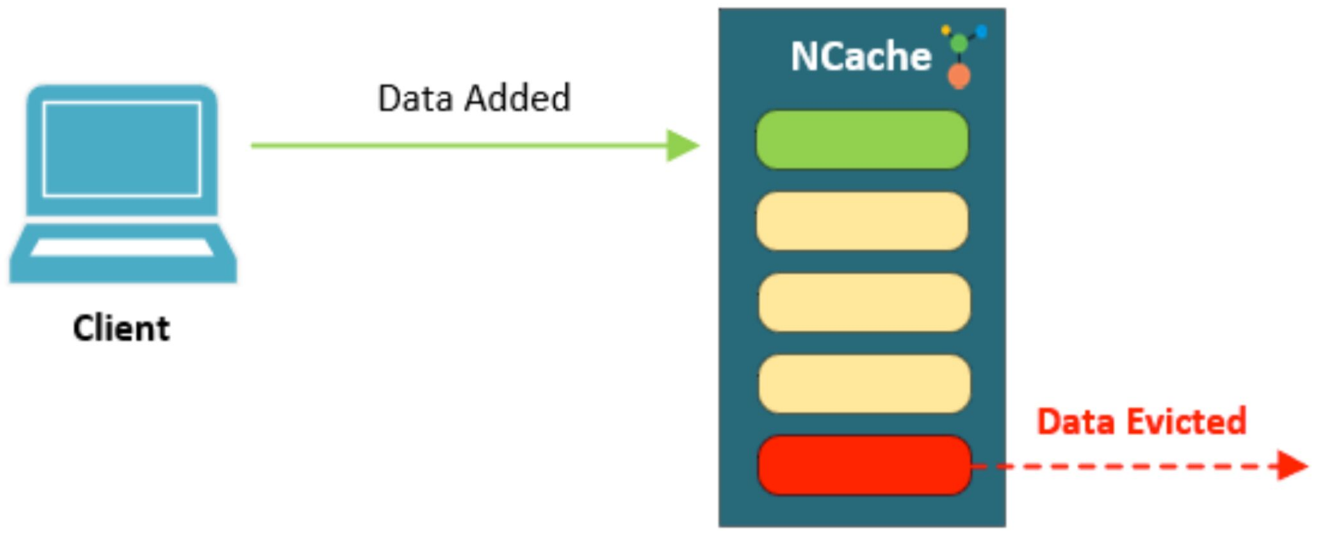
→ The pupose of cache is to reduce the average time to access data by avoiding hitting the low speed backend data sources.

→ Cache provides high speed data access but is LIMITED, which is why it is important to carefully maintain and manage the information stored in it.

→ The process by which old, relatively unused, or excessively voluminous data can be dropped from the cache to make space for the new and relatively most frequently used data is known as cache eviction.



Santosh kumar Mishra



Cache Eviction Policies:

→ First In First Out(FIFO) : Cache evicts the first block accessed.

→ Last In First Out(LIFO) : Cache evicts the block accessed most recently.

These above strategies are not the best since they hinder the whole idea of cache by disregarding how often an item is used.

in Santosh kumar Mishra

→ Least Recently Used(LRU) : Discards the item which is least recently used.

It identifies items which are relatively unused and evict that item to make space for the new items.
LRU is one of the most popular caching strategies used.

→ Least Frequently Used(LFU) : Discards the item which is least frequently used.

In this we maintain the count of the number of times an item is accessed.An item which have the least count is evicted.

Santosh Kumar Mishra
SDE @Microsoft

Follow for more

iamsantoshmishra

iamsantoshmishra

Interview Cafe

Interview Cafe Notes