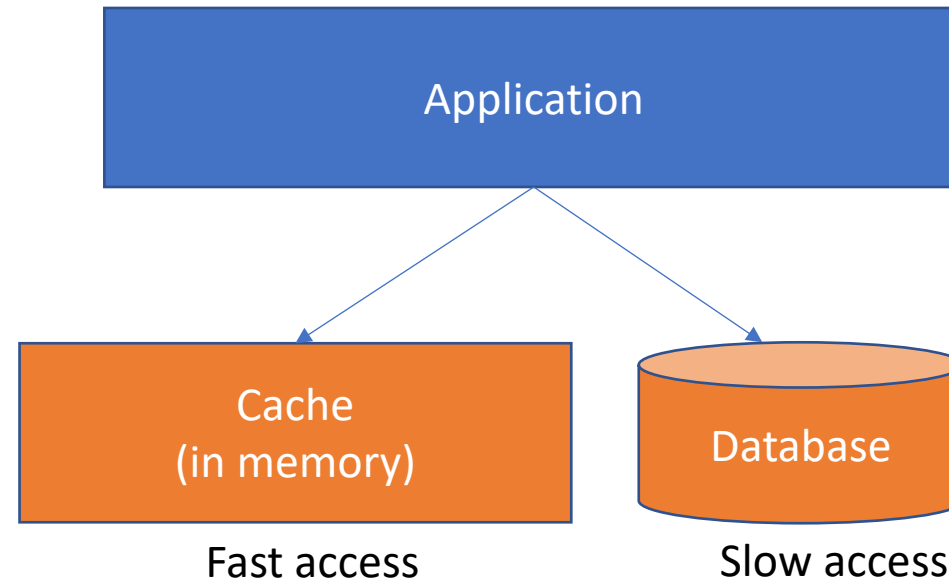


# Caching

**Read:** If object is in cache and is not stale, get it from cache.  
If not, get it from database



**Eviction policies** (when cache is full).

1. Least recently used
2. Least frequently used
3. Most recently used
4. Last/First in-first out

**Write:** Different schemes.

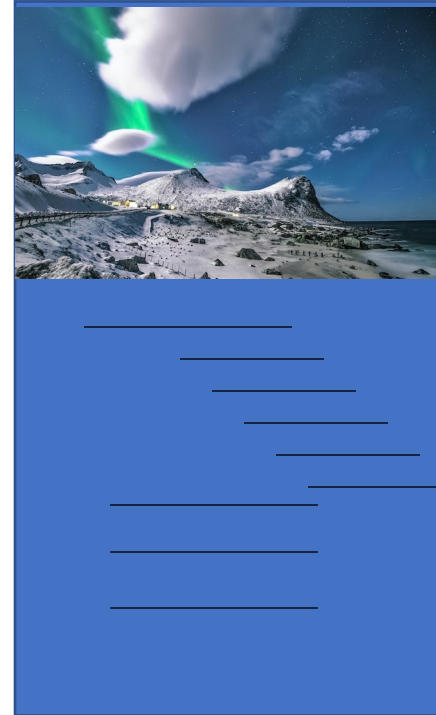
1. Flush/mark(stale) cache entry if it exists, Write to disk
2. Write to cache and disk.
3. Write to cache, write to disk later (need to track this)

- Time To Live (TTL) field specifies lifetime of entry in cache. Forces the flushing out of data when TTL expires, and events may be triggered. Separate thread may be needed to check expiry.
- Checkpointing saves periodic versions of data on disk

# Organizing Information

Assume

- Your social media app allows **public media posts** (video/image)
- Users are allowed to **comment**
- Comments can be numerous (1000s)
- Users are allowed to **reply to comments** (threads)
- Replies can be numerous too (100s)
- Users can be sent emails notifying them of
  - 'engaging' posts
  - 'engaging' threads
  - If they click on the link in email, they should be taken to the post/thread



*How will you store the data for this scenario?*