Design a Distributed cache (e.g. memcached)

[This problem may have been done in the class. Whatever the instructor taught you supersedes the following]

* Read up on single-server cache first

* Distributed cache needs multiple shards (server). How many shards are required depends on what is the most QPS you can handle on a single machine. For memcached, one of the most popular distributed caches, it's 50KQPS default.

* Distributed/Consistent hashing is a good mechanism to implement a distributed cache.

Important: In an interview, it is fair game to be asked to code a basic distributed hashing mechanism e.g. consistent hashing (see links below).

* It is likely, that you may be asked to "design memory cache". That doesn't mean "memcached", the popular distributed cache. "memory cache" is just a lazy way of saying - design cache on a single server and then think about distributed. Mostly distributed.

Reading pointers:

This one is AMAZING: http://codecapsule.com/2012/11/07/ikvs-implementing-a-key-value-store-table-of-contents/

https://www.quora.com/Whats-the-maximum-throughput-in-queries-per-second-for-memcached-with-small-object-sizeshttp://www.tom-e-white.com/2007/11/consistent-hashing.html

https://ivoroshilin.wordpress.com/2013/07/15/distributed-caching-under-consistent-hashing/ [Java code]

Numbers every engineer should know: https://gist.github.com/jboner/2841832