THE ANALYSIS AND VARIANTS OF SOLVING THE PROBLEMS OF OPERATIVE CACHING WITH THE HELP OF MEMORY CACHE TECHNOLOGY

© 2009 P.V. Nazin*

Keywords: caching, RAM, MemcachDB, Berkeley BD, clustering.

The article considers the process of organizing caching of the dynamically formed content on the side of web-server. Special attention is paid to the problem of server functioning safety, several optimization variants are suggested.

Today almost any web-project is dynamic; it means that pages requested by the user are formed "just-in-time" with the specific features of the user. During the process of page formation it is often necessary to obtain the data from rather slow sources (for example, a database, a file server, an external resource). To generate the response to a difficult request the quantity of such references can be estimated in tens, and the time of each reference processing can vary from ten milliseconds to one minute. Even with the parallel processing of difficult requests we will receive unsatisfactory time of the response.

The solution to this problem is cashing: we place the result of the calculations in some storage which has excellent characteristics in terms of time of access to the information. Now, instead of forming long requests, we address to a fast cache.

As far as web-projects are concerned the success of caching is defined by the fact that there are most popular pages on any site, some data is used on all or almost all the pages that is there are some samples which appear to be requested much more often than others. Therefore there is a possibility to replace some references to the backend with the references to the cache.

Memory cached APIs provide a giant hash table distributed across multiple machines. When the table is full, subsequent inserts cause older data to be put into the least recently used (LRU) order. Applications using memory cache typically layer cached requests and additions into the core before falling back on a slower backing store, such as a database.

In our research 4 most common problems connected with the performance of cache servers were revealed.

1. The choice of cache keys: cache system imposes some rules limiting the length of the key line and its uniqueness for each request. We offer to use the md5 algorithm of encrypting.

2. The loss of keys: cache is not a reliable storage as the key can be deleted before it expires. The architecture of the project must react quickly to the loss of keys. There are 3 main reasons for losing keys:

1. The key was deleted before it expired because the memory was used for storing other keys.

2. The key was deleted as it expired. Strictly speaking, this situation is not a loss, as we limited the expiry time of this key ourselves. However, the situation won't be satisfactory for frontend as the sorting will be formed again.

3. The most unpleasant situation is the crash of the cached process or the server where it is placed. Then we lose all the keys that are stored in the cache.

3. Clustering and the choice of the algorithm of key distribution: to start the distribution and to achieve a fail-safe system a cluster of cached servers is used instead of one server. The servers forming the cluster can possess different memory volumes; meanwhile the total cache volume will be equal to the sum of all the volumes of the caches of the cached servers forming the cluster.

4. The problem of simultaneous re-formation of caches: It appears that when several requests to re-format a particular cache are made the block system is used not to allow the process of cache reformatting to be launched in parallel.

http://memcachedb.org.

http://ru.wikipedia.org/w/index.php?title.ru.wikipedia.org/wiki/Hash.

http://weblogs.java.net/blog/tomwhite/archive/2007/11/consistent_hash.html.

http://www.lastfm.ru/user/RJ/journal/2007/04/10/rz_libketama_-_a_consistent_hashing_algo_for_memcache_clients.

http://korchasa.blogspot.com/2008/04/dogpile.html.

http://www.oracle.com/technology/products/berkeley-db.

Received for publication on 16.10.2009

* Pavel V. Nazin, post-graduate student of Volga State University of Telecommunication and Information. E-mail: vestnik@sseu.ru.