

An improvement in LVCT cache replacement policy for data grid

Abstract: Caching in data grid has great benefits because of faster and nearer data availability of data objects. Caching decreases retrieval time of data objects. One of the challenging tasks in designing a cache is designing its replacement policy. The replacement policy decides which set of files are to be evicted to accommodate the newly arrived file in the cache and also whether a newly arrived file should be cached or not. The traditional replacements policies, based on LRU and LFU algorithms are not suitable for the data grid because of large-and-varying size and retrieving cost of data objects in data grid [1]. Other class of replacement policies used in data grid is based on calculating utility function for each file which also takes care of size and retrieving cost of the file along with Locality Strength [2, 3, 4]. We can easily deduce that lesser the value of locality strength for a file, it is better to evict that file because of its low probability of re-accessing while other two parameters remaining the same. In LVCT (Least Value Based on Caching Time) policy [1], locality strength in the utility function is estimated using a term "Caching Time (CT)" where CT for a file F is defined to be sum of size of all files accessed after last reference to file F. While this policy showed better performance compared to previous policies [2, 3, 4, 5], it did not take care of the cases where CT computed in the policy is almost same but the number of files accessed after last reference to file F differs largely. For example, two cases where (i) number of files is large but sizes of these files are small and (ii) number of files is small but sizes of these files are big after last reference to a file; can result into nearly same value of CT and hence same value of locality strength. However, locality strength of first case should be low compared to second case because of large number of files accessed after its last reference having the same value of CT for both. Here, we propose that number of files accessed after last reference to file F should also be considered along with CT in estimation of locality strength so that a better decision can be made for the eviction of a file.

References:

- [1] Song Jiang and Xiaodong Zhang, "Efficient distributed disk caching in data grid management", Proceedings of IEEE International Conference on Cluster Computing, 2003.
- [2] R. Wooster and M. Abrams, "Proxy caching that estimates page load delays", Proceedings of 6th international World Wide Web Conference, April 1997
- [3] P. Lorensetti, L. Rizzo and L. Vicisano, "Replacement Policies for a Proxy Cache", <http://www.iet.unipi.it/luigi/research.html>
- [4] P. Cao and S. Irani, "Cost-aware WWW proxy caching algorithms", Proceedings of USENIX Symposium on Internet Technologies and Systems, 1997
- [5] Ekow Otoo, Frank Olken and Arie Shoshani, "Disk Cache Replacement Algorithm for Storage Resource Managers in Data Grids", Proceeding of IEEE Conference on Super-Computing, 2002

Primary author: Dr KASHYAP, Arti (LNM Institute of IT)

Co-authors: RATHORE, A. (LNMIIT); ACHARA, J. P. (LNMIIT); GUPTA, V.K. (LNMIIT)

Presenter: ACHARA, J. P. (LNMIIT)

Track Classification: Computing Technology for Physics Research