International Journal of Distributed Systems and Technologies
Volume 8 • Issue 1 • January-March 2017

Algorithms for Maintaining Consistency of Cached Data for Mobile Clients in Distributed File System

Pavel Bžoch, University of West Bohemia, Pilsen, Czech Republic
Jiří Šafařík, University of West Bohemia, Pilsen, Czech Republic

ABSTRACT

A cache stores data in order to serve future requests to those data faster. In mobile devices, the data have to be transferred from a server through the mobile network before being stored in the cache. The mobile network is prone to failure caused by users' movements and by the placement of base transceiver stations. Moreover, the mobile devices use various telecommunications technologies and therefore the speed of the network is highly variable. Using a cellular network for communication is also expensive. The cache is an intermediate component which addresses this problem. Once the data are downloaded, they can be stored in the cache for possible future reuse. When using a cache, the system designer presumes that the data will be requested again in the future. On the other hand, the original data stored on the server can be changed. Then, the cached data are in an inconsistent state. In this paper, authors present an adaptive method for maintaining the consistency of cached data which saves network traffic by reducing the number of messages needed for inconsistency detection.

KEYWORDS

Cache, Client-Side Caching, Consistency, Distributed File System, Mobile Devices

INTRODUCTION

The popularity and use of mobile devices has grown recently. The term "mobile devices" includes cell phones, personal digital assistants (PDA), smart phones, netbooks, tablets etc. Also, the performance and the storage capacity of these devices have been enhanced. Smart phones can have up to 4-core processors like personal computers. For storing users' data, microSD cards are usually used. Having this potential, the mobile devices, namely cellular phones, can be used not only for making calls and sending Short Message Services (SMSs), but also for creating and playing multimedia files, accessing the internet or reading and writing e-mails. The mobile devices can even be connected together and can be used for distributed computing (Liang, Hsieh, & Lyu, 2007).

Multimedia files, the results of distributed computing and also binaries of mobile applications demand lots of storage space. Though the capacity of commonly used microSD cards is increasing, they are still not sufficient to store all the demanded data. In this case, the data can be stored on remote storage and accessed via wireless networks. When remote storage is chosen to store users' content, the users wish to access the data as simply as if the data were stored locally. Additionally, other demands such as performance, reliability, availability or security are required from the remote storage. Use of a distributed file system can satisfy all these requirements.

DOI: 10.4018/IJDST.2017010102

The wireless connectivity is the biggest disadvantage of accessing the data remotely. Wireless technologies can provide very slow connectivity and can be expensive (e.g. cellular networks with old technologies) or can be quick and relatively cheap such as Wireless Fidelity (Wi-Fi). Nowadays, new technologies for increasing network speed are being put into service (e.g. Long Term Evolution (LTE) (Khan, 2010)). However, these technologies are not available everywhere. They are usually put into service in big cities first and only then expand to rural areas. The price of the data transfer is still high. Moreover, time division multiple access with preference for voice is used to bundle the data transfer and voice service. Thus, the speed of the data transfer is reduced when there are many phone calls (Ahamed, 2008).

For increasing the performance of mobile devices and saving network communication, accessed data can be stored in an intermediate component called a cache. The capacity of a cache is limited due to the limited capacity of memory cards; therefore, it cannot store all the demanded data. When the cache is full, the caching algorithm uses caching policy to mark the old cached content to be replaced (Aho, 1971). These policies (described e.g. in (Bžoch, Matějka, Pešička, & Šafařík, Design and Implementation of a Caching Algorithm Applicable to Mobile Clients, 2012) and (Bžoch, Matějka, Pešička, & Šafařík, Towards Caching Algorithm Applicable to Mobile Clients, 2012)) usually use local statistical information collected from the user's previous behaviour for making a replacement decision. In (Bžoch, Matějka, Pešička, & Šafařík, Design and Implementation of a Caching Algorithm Applicable to Mobile Clients, 2012) and (Bžoch, Matějka, Pešička, & Šafařík, Towards Caching Algorithm Applicable to Mobile Clients, 2012), the authors presented new caching policies called Least Frequently Used with Server Statistics (LFU-SS) and Least Recently and Frequently Used with Server Statistics (LRFU-SS) which use a combination of local and server statistics for improving the read hit and saved bytes indicators. The caching in mobile devices is depicted in Figure 1.

In comparison to other caching policies, both novel caching algorithms have better results, meaning that the cached data remain in the cache for a longer time period. Meantime, the original data may be updated more frequently. In that case, the cached data are in an inconsistent state. Accessing inconsistent data is undesirable.

In this paper, authors provide an overview of existing algorithms for maintaining the consistency of cached data. Then, novel algorithms for maintaining cache consistency are proposed. The first algorithm is derived from Network File System (NFS); the second and third algorithms are the authors' inventions.

Figure 1. Caching in mobile devices