Computers and Geosciences 112 (2018) 1–8

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

A dynamic replication management strategy in distributed GIS

Shaoming Pan [a], Lian Xiong [b], Zhengquan Xu [a,*], Yanwen Chong [a], Qingxiang Meng [c]

[a] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, Hubei, China
[b] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
[c] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, China

ARTICLE INFO

Keywords:
Data replication
Replication management
Geospatial data
Data popularity
Distributed GIS

ABSTRACT

Replication strategy is one of effective solutions to meet the requirement of service response time by preparing data in advance to avoid the delay of reading data from disks. This paper presents a brand-new method to create copies considering the selection of replicas set, the number of copies for each replica and the placement strategy of all copies. First, the popularities of all data are computed considering both the historical access records and the timeliness of the records. Then, replica set can be selected based on their recent popularities. Also, an enhanced Q-value scheme is proposed to assign the number of copies for each replica. Finally, a reasonable copies placement strategy is designed to meet the requirement of load balance. In addition, we present several experiments that compare the proposed method with techniques that use other replication management strategies. The results show that the proposed model has better performance than other algorithms in all respects. Moreover, the experiments based on different parameters also demonstrated the effectiveness and adaptability of the proposed algorithm.

1. Introduction

Distributed geographic information system (GIS) can store large amounts of geospatial data (Pan et al., 2017) by using more storage nodes and also, can deal with more users by dispensing requests to different servers. Moreover, distributed GIS can improve the service response time by accessing data from local storage node through local server and also, distributed GIS can reduce the risk of a single point failure. Scheduling, organizing and managing all those data in distributed environment is the key way to achieve the above benefits (Rui et al., 2017) and most of such applications are adopting this management approach (Boulos, 2005; Bell et al., 2007).

The performance of distributed system is decided by two aspects: the load balance of whole system and the service response time. One of primary method to realize load balance in distributed GIS is data placement strategy. The Access Pattern-based Storage Algorithm (APSA) (Pan et al., 2015) mines the relationship of all data and then separately stores the related data into different storage nodes so as to get a higher whole parallel access probability and avoid to access data from remote nodes. Also, Dynamic Computation Correlation based Placement strategyaaa (DCCP) (Wang et al., 2016) places the dataset considering both the I/O load and the capacity load according to their computation correlations.

Since the relationship among all data will change continuously with the running of system (Rui et al., 2012), mining their correlations in advance based on their historical relationships cannot dynamically track their characteristics. Furthermore, there are huge amount of data which will be accessed and stored, dynamically adjusting data among all storage nodes will lead to a storm of data transferring and reduce the availability of network bandwidth (Chervenak et al., 2009).

To address this problem, data replication is an important optimization strategy to keep some hotspot data as replicas in high-speed cache so as to avoid getting data from remote storage nodes repeatedly, and which can significantly improve system performance (Rui et al., 2017). In fact, some researches show that data replication strategy is the only way to improve the performance of load balance (Amjad et al., 2012) and service response time (Xiong et al., 2016). Obviously, not all geospatial data need to be used to create copies due to the massive dataset in GIS and the minimal access frequency of part of geospatial data. Thus, in order to clearly describe our method, we define that *replica* is a sub-dataset of all geospatial data and each replica has one or more *copies* which are stored into different places (Manifestly, all geospatial data will be split into two sub datasets which are the replica and the others, where only the data belongs to replica sub dataset can create copies so as to limit the total amount of copies to save storage space).

* Corresponding author.
E-mail address: xuzq@whu.edu.cn (Z. Xu).

https://doi.org/10.1016/j.cageo.2017.11.017
Received 14 March 2017; Received in revised form 9 October 2017; Accepted 22 November 2017
Available online 2 December 2017

For centralized system, there are only one cache buffer and each replica has only one copy. Examples of such strategies include the Popularity model, Markov Chain model and the Passive model. The popularity model (Shi et al., 2005) calculates the popularity of all data based on their history access records and then selects those data with higher probabilities as replicas. Markov Chain model uses a basic Markov Chain model (Li et al., 2010) or some improved Markov Chain model to cache optimum data as replicas (Rui et al., 2012). The passive model just keeps the data being accessed as replicas, such as Least Frequently Used (LFU) and Least Recently Used (LRU).

Unfortunately, unlike centralized system, data replication in distributed system must consider two more key problems: how many copies should be produced for each replica and where all copies will be placed to meet the requirements of load balance? Furthermore, dynamically tracking the characteristics of all data with the running of system is also the key of the solutions.

2. Related work

Bandwidth Hierarchy based Replication (BHR) (Park et al., 2004) and the modified BHR (Sashi and Thanamani, 2011) are two typical methods which use dynamic replication strategy to reduce data access latency and to increase resource utilization rate. The main idea of BHR is to keep the required data in the same region as much as possible to reduce the external-schedule time. BHR selects the best site to keep replicas based on all nodes' bandwidth distribution.

Hierarchical Replication Strategy (HRS) (Chang et al., 2007) is similar to BHR and is also used in the data grids. HRS aims to get an enough higher hit ratio of required data within a region so as to reduce the internal-schedule time. Rather than BHR which computes the popularities of all replicas based on the cluster level, HRS computes the popularities of all replicas at site level and can more accurately define the real characteristics of system.

Due to the special requirements and characteristics of GIS, Global User-Driven Caching method (GUDC) (Pan et al., 2017) is one of the typical data caching modes for GIS. GUDC computes all data relationship based on their historical access records and then compares the conditional prefetching probability from uncached datasets so as to select some data as replicas and then, store them into cache buffer. Meanwhile, GUDC uses the same method to delete the replicas from cache buffer to save space.

Collaboration Model for Replicas in Distributed Caching (CMRDC) (Rui et al., 2017) designed three aspects of data replication strategy for distributed GIS which are replica dataset selection, the number of copies for each replicas and replicas placement. CMRDC uses a steady-state cache hit ratio to estimate the size of replica dataset and then computes the popularities of each data and selects a given number of the higher popularity data as replica datasets.

Obviously, those methods mainly designed for GIS are very effective and provide some far better strategies to improve system performance in distributed GIS. But their effectiveness depends on two preconditions: the enough large of historical access records and the enough stability of their relationships.

Unfortunately, as a typical dynamic system, the relationships or popularities of all data are changed continuously (Rui et al., 2012) and thus mining the patterns based on historical access records to guide the replication strategy may not always work. Furthermore, we can not obtain a large enough historical access records when the system has just started.

To address above mentioned problems, the proposed method in this article, i.e., Access-related Dynamic Data Replication strategy (ADDR), also considers the three aspects of data replication strategy for distributed GIS. ADDR only uses the recent limited records to mine data relationships. Moreover, each record has a different weight based on their freshness (Chang and Chang, 2008) and thus, we can track the changes of relationships closely. Then, an enhanced Q-value scheme (Burghes et al., 1982) which considers both the fairness and capabilities of all nodes is implemented to get the number of copies for each replica and to guide the replicas placement among all nodes. Also, the replicas will be adjusted dynamically.

3. Replication management strategies

3.1. System architecture

Fig. 1 gives a typical cooperation model for replication in distributed GIS, where raster data are split into different resolution ratio image files due to pyramid model and each of them are separately stored as a single geospatial data. All geospatial data are distributed stored in M clusters and each cluster has storage node and server. Each server holds one high speed cache buffer. The server and storage node in the same cluster are connected by LAN and clusters are connected through internet. In distributed GIS, users' access requests are distributed by load distributor to different servers based on the length of access queue and the cached data set.

Since there exist massive geospatial data stored in distributed GIS and part of them only have lower access frequency (Pan et al., 2017), only part of geospatial data need to be selected as a female parent to create copies so as to save storage space. Thus, based on the above analysis, ADDR must solve three problems: selecting the appropriate sub-geospatial dataset as replicas set, allotting storage space for each replica (i.e., deciding the number of copies for each replica) and placing all copies into suitable locations.

In this paper, we proposed a complete solution to meet the above three requirements and we explained these three mechanisms in detail in the following.

3.2. Replica set selection algorithm

Firstly, let $D = \{d_1, d_2, \cdots, d_N\}$ be the set of all data stored in distributed GIS, where N is the size of dataset. Each element in D is labeled with a natural number coding scheme and the size of data d_m is s_m ($m \in [1\ \ N]$). Assuming the historical access records will be chronologically recorded and which can be denoted as $H = (\{a_1, t_1\}, \{a_2, t_2\}, \cdots, \{a_K, t_K\})$, where K is the total number of accessing to all data, $a_j (j \in [1\ \ K])$ denotes the label of j-th accessed data and $t_j (j \in [1\ \ K])$ denotes the time of j-th accessing.

Dividing historical access records into several sub-parts $H = (H_1, H_2, \cdots, H_L)$ based on a given time interval τ, where L is the total number of sub-parts and $H_i = (\{a_{1_i}, t_{1_i}\}, \{a_{2_i}, t_{2_i}\}, \cdots, \{a_{K_i}, t_{K_i}\})(i \in [1\ \ L])$ satisfy the conditions: $t_{K_i} - t_{1_i} \le \tau$ and $t_{1_{(i+1)}} - t_{1_i} > \tau$, where K_i is the total number of the accessing to all data in i-th sub-part. For $\forall H_i \in H$, let $\lambda_{m,i}$ denotes the total access count of data $d_m (m \in [1\ \ N])$ during the time of i-th sub-part $[t_{1_i}\ \ t_{K_i}](i \in [1\ \ L])$, then the popularity of data d_m can be stated as follows:

$$\xi_m = \sum_{i=1}^{L} \frac{\lambda_{m,i} \times f(L-i)}{K_i} \tag{1}$$

where $f(x)$ is a decay function. Therefore,

$$\overline{\xi} = \sum_{m=1}^{N} \xi_m \Big/ N \tag{2}$$

indicates the average popularity and then, the data which the popularity is higher than the average popularity can be selected as the member of replica set. Furthermore, let $C = \{c_1, c_2, \cdots, c_M\}$ be the set of all server in distributed GIS and M is the total number of servers. The bandwidth of server c_i is b_i and the high-speed cache buffer size is μ_i ($i \in [1\ \ M]$) and so, τ can be estimated: