A SCALABLE DISTRIBUTED SPATIAL DATA STORAGE MODE

Zhou Yan [a, *], Zhu Qing [a]

[a] State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, China, (zhouyan, zhuqing)@lmars.whu.edu.cn

Commission VI, WG VI/4

KEY WORDS: Object-based Storage, Distributed Storage Model, Spatial Data

ABSTRACT:

With the rapid development of techniques for acquisition of spatial data, the storage and management demands of spatial data are increasing from PB to TB level. GIS generally store spatial data in the form of files or tables of database, but not the spatial entity that is described and stored on object-oriented model. This inconsistent storage model could destroy the integrity of spatial entity, and the structured relational table is not always suitable for unstructured spatial data. Aiming at this situation, based on the self-characteristics of spatial data such as large size, territorial, multi-dimension etc., this paper presents a scalable distributed spatial data storage mode by combines object-based storage (OBS) thought into GIS. And describes its promising advantages for distributed spatial data storage, which include scalability, high performance and security etc.

1. INTRODUCTION

With the rapid development of geotechnologies, new data capturing technologies such as various high-resolution sensors continuously create vast amounts of spatial data in real time or near real time. The storage and management requirements of spatial database are increasing from Gigabytes to Terabytes, even to Petabytes level. Correspondingly, these bring a new problem between the exploding increase of data volume and effective utilization of spatial data. Traditional centralized storage approach has appeared more and more difficulties for managing massive spatial data with limited storage capacity and poor scalability. Furthermore, as the massive, territorial distribution, multi-users access characteristics of spatial data, distributed approach is a more suitable alternative for massive spatial data storage in network environment.

In recent decades, distributed GIS has made rapid progress in supporting distributed spatial analysis computing, data sharing across platform, and dealing with multi-users simultaneous access etc. However, with the increasingly booming of spatial data, distributed GIS therewith open, dynamic and dependent on network characteristics has to face more challenges such as large-scale remote data access and transferring performance, dynamic increase of storage capacity and data security across heterogeneous platform etc, especially, when multi-users largely simultaneous access, huge data size request, long distance data transferring. Therefore we urgently need an appropriate spatial data storage mode, which could to make a voice to the current challenges and meet the goals of massive spatial data storage including good scalability, security, high storage and access performance and flexible data sharing etc. Aiming these, in this paper we presents a scalable distributed spatial data storage mode by combines object-based storage (OBS) thought into GIS.

The rest of the paper is organized as follows. Section 2 briefly describes current spatial data organization and storage mode. Section 3 presents a scalable spatial data storage mode based on object-based storage (OBS) and discusses its promising advantages for distributed storage of massive spatial data. Section 4 concludes the paper and discusses the future directions of this research.

2. CURRENT SPATIAL DATA STORAGE

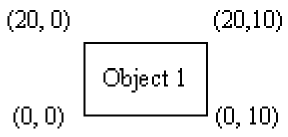
2.1 Spatial Data Storage Model

GIS spatial data has three typical storage modes (1) file storage mode; (2) relational database storage mode; and (3) object-relational database storage mode. In file storage mode, spatial data is stored as a collection of files, which are catalogued and indexed directly so that efficient access is possible. File storage mode is simple to use but low security, and has limitations for storing large-scale data sets.

Relational database storage mode stores spatial data as a collection of tables, which use database management technology to index data. Spatial data is stored as text or binary type fields in relational tables (Figure 1), however, since a spatial data object may be composed of a single point or several thousands of polygons, it is usually not possible to store collections of such spatial objects in a single relational table with a fixed tuple size. So using structured relational table to store unstructured spatial data will result in more storage space consumption and low storage performance.

Object-relational database storage mode provide geometry type for the management of spatial object, it’s more efficient but doesn’t support customized data structure. For example, oracle spatial provides an open architecture to store spatial object as Geometry type (Figure 2) within database, but it has not 3D data model at present, though also can use binary large object (BLOBs) to store undefination spatial object structures, it often suffer from a severe performance loss if many BLOBs.

* Corresponding author.



(a) Rectangle spatial object

Object ID	Attribute	Coordinate	EntitiesSEQ
1	…	20,0,20,10,0,10	1
1	…	0,0,20,0	2
…	…		…

(b) Spatial object stored as text field

Object ID	Coordinate
1	Binary block
2	…
…	…

(b) Spatial object stored as binary field

Figure 1.Relational database storage mode

Object ID	Attribute	SDO_GEOMETRY
1	…	(20,0,20,10,0,10,0,0)
2	…	…
…	…	

Figure 2. Object-relational database storage mode

Each of storage modes mentioned above has its own disadvantages so as not to store spatial data effectively. Furthermore we generally represent spatial entity as object-oriented form but store spatial data in the form of files or database tables. This inconsistent storage and representation model could destroy the integrity of spatial data. Besides these, the structured relational table is not always suitable for unstructured spatial data. Evidently, an ideal spatial data model should provide integrative storage and representation for spatial data. Here a flexible and effective spatial data model is desirable to meet the storage requirements of spatial objects. Naturally, object-oriented model could be a good selection.

2.2 Spatial Data Storage Architecture

We have discussed spatial data storage from the storage mode point of the view in section 2.1. This section will discuss the likely storage architectures of massive spatial data. To obtain a better understanding of storage architecture requirements spatial data demanded. We first need to know some basic properties of spatial data. First, spatial data have a complex structure. Second, spatial data are territorial distribution. Third, spatial data tend to be massive. Fourth, there is no standard algebra defined on spatial data. Fifth, many spatial operators are more expensive computing consumption (Gaeda et al., 1998).

Massive spatial data demands scalable storage architecture to meet the increasing data storage requirements; territorial distribution characteristic of spatial data requires a distributed storage architecture; Complex data structure and unstandard algebra definition of spatial data need a flexible object-oriented storage model. Expensive spatial data operators want a high performance storage system. Besides these, as an available storage architecture of massive spatial data, it should have high security.

There are three basic network storage architectures could be used to the distributed storage of spatial data common in use today. These are direct attached storage (DAS), network attached storage (NAS) and storage area network (SAN).

- DAS is composed of block-based storage devices connected directly to a host machine. It's high bandwidth, high security, and easy to use but have limited data sharing among servers. A good example is a small database or a file server. DAS is a common choice for applications that require high performance.

- NAS offers file-based storage to hosts connected to an IP network. It offloads the entire file system from the hosts to the file server. NAS is often chosen for applications with a need for sharing files across different platform. It is ease of use but not a very scalable system and also limited performance.

- SAN replaces the bus-based architecture of DAS with a switched fabric. It has highly scalable, high performance, high bandwidth, but not as secure, not easy to use, high costs and limited cross-platform functionality. SAN are often chosen for those applications with a need for highly scalable performance from the storage devices.

	DAS	NAS	SAN
Storage Unit	Block	File	Block
Scalability	Low	Medium	Medium
Security	High	Low	Medium
Performance	High	Low	High
Cross-platform	Low	High	Medium
Storage-scale	Small-scale	Medium-scale	Medium-scale

Table 1. The assessment of three basic storage architectures

The assessment of three basic storage architectures is listed in Table 1 (Thomas, 2002). From it we can conclude that these existing storage architectures are not completely suitable for the storage of massive spatial data because of the dissatisfied scalability, the poor security, limited cross-platform functionality or low performance respectively. It's evident we need new storage architecture for the storage of massive spatial data, which should provide high scalability, high performance and allow heterogeneous applications and systems to access storage more easily and with better security.

3. SCALABLE SPATIAL DATA STORAGE MODE

Currently object-based storage (OBS), which has been underway since 1998, has been hot spot of high-performance storage research. Nowadays, many famous universities, corporations and research institutions, such as Carnegie Mellon university (CMU), Bell Laboratory, IBM and Intel Inc. etc. have established their correlative research and projects, typically include Network Attached Secure Disk (NASD) project at CMU (Garth, 1998), Lustre system of Cluster File Systems Inc (CFS Inc, 2002), and ActiveScale Storage Cluster of Panasas Inc (Panasas Inc, 2005) etc. Technical Committee T10 and Storage Networking Industry Association (SNIA) are setting correlative standard on object-based storage technology (http://www.t10.org, http://www.snia.org).

Conceptually, the term OBS refers to a new kind of object-based storage technology, it takes an "object" as the basic storage unit of data, objects are of variable-length and can be used to store entire data structures and any data type, such as files, database tables, images, or multimedia etc (Mesnier, 2003). Compared with the traditional file/block-based storage models, OBS model adopts object interface and offload storage management function from application to storage device. Therefore, OBS has the following key features.

- Object is a logical storage unit, which contains object data, object attributes, and methods.
- Object interface provides simple method, such as create, delete, open, close, read, write and so on. These interfaces are easily extended with applications.
- OBS enables device intelligent, including device and data management, object structure and relationship interpretation, access patterns and security setting.

OBS treats storage neither as blocks nor files, but as objects, which provides the integrative object-oriented model for the storage and representation of spatial data. Furthermore related researches show, compared with other storage technologies mentioned in section 2.2, OBS has higher scalability, security and performance (Thomas, 2005; Intel Inc, 2003). Therefore OBS is very suited for large-scale data storage, it is an ideal storage model for massive geo-spatial data.

Combining OBS technology into GIS, we can partition spatial data to a collection of storage objects what we call spatial data storage object (SDSO). SDSO is a container for spatial data, attributes and methods, which can consist of any data type such as remote sensor images, DEM, vector spatial objects or multimedia data etc, and any size data, for example, an object may contain a entire image object, or just a portion of a image. SDSOs are stored on one or many object-based storage devices (OSD).

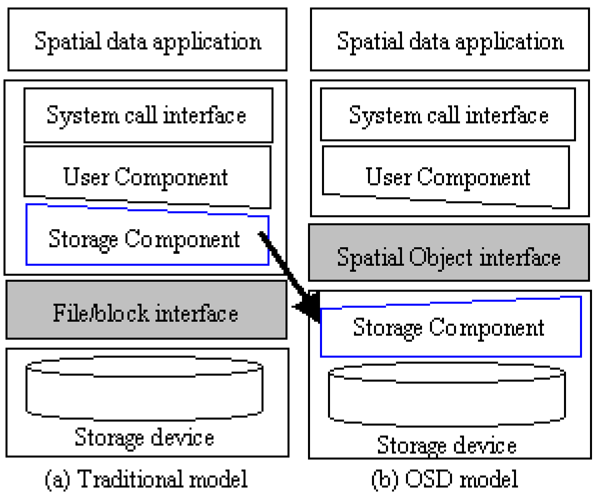


Figure 3 Spatial data storage based on OBS

As shown in Figure 3, spatial data storage based on OBS is different from the traditional file/block based storage model in which the storage management component is offloaded to the storage device and the interface of the storage device change from file/block interface to spatial object interface. The spatial data interface can directly support object-oriented spatial data storage in rock-bottom storage device and it's not necessary to map spatial object to files or database tables as traditional storage modes that often results in the destroyed integrity of spatial object (Yu et al., 2005). SDSO is the basic storage unit in OSD. OSD is responsible for managing SDSOs and providing object attribute interpretation such as object structure and relationship awareness, object content awareness, access patterns etc. SDSO may be embedded more spatial semanteme and allow to define any complicate spatial object, that is, to support customized data structure. Spatial data storage model based on OBS can support object-oriented thought directly from lower storage layer to upper application layer by defining appropriated SDSOs and using object-based storage technology. Furthermore, it provides operations associated with spatial data objects.

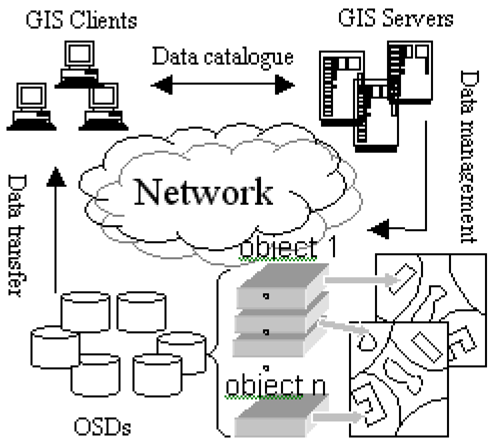


Figure 4. A scalable distributed spatial data storage mode

Figure 4 illustrates a scalable distributed spatial data storage mode based on the spatial object storage model mentioned above. It consists of GIS Clients, GIS Servers and OSDs, which be connected each other by network infrastructure and form a distributed storage systems attached network. GIS Clients send spatial data request to GIS Servers. Server is responsible for establishing and maintaining spatial metadata catalogue. OSD is a network-attached device containing the storage media, disk or tape, which is responsible for storing and managing spatial data storage objects by calling physical interfaces of OSD model provided.

A typical spatial data storage/access process could be generalized as the following steps. First, GIS Clients send a data request to GIS Servers. Second, GIS Server responds the request to search relevant spatial objects by metadata catalogue, if failure, to return failure information to GIS Client; if succeed, to return the data storage information such as location, layout, size etc to GIS Clients. And then GIS Client communicates with OSD directly to storage/obtain the required spatial objects bypass the GIS Server.

Our presented distributed spatial data storage mode combines the advantages of OBS technology and object-oriented thought into spatial data storage. It possesses the following distinct advantages.

- High performance. As noted in the introduction, traditional centralized storage mode always suffers from the slow response when simultaneous multi-users access, large data request, and long distance data transferring since delivery pathways between the server and the clients become more congested. The distributed storage mode solves these problems by transferring spatial data objects directly between destination OSD and GIS Client bypass server. This can reduce the likelihood of overloaded GIS Servers and provides system redundancy. The direct accessing approach minimizes the workload on servers and maximizes overall network bandwidth and performance.

- Security. Security is another important feature of object-based distributed storage mode that distinguishes it from file/block-based storage mode. It has typically relied on authentication of the clients and private networks to guarantee the security of the system. In the object-based security architecture, every access is authorized, and the authorization is done without communicating with a central authority that may slow the data path. It provides security at every level, such as authentication of the storage devices to the storage system, authentication of compute nodes to the storage system, authorization for compute node commands to the storage system etc (Mesnier, 2003). These levels of security can give customers confidence that they can use more cost effective, manageable and easily accessible networks, such as Ethernet for storage traffic while improving overall storage system security.

- Scalability. In this distributed storage mode, OSD is used to store spatial data. GIS Clients communicate directly with the OSD to store/access spatial data. The key benefit is that it allows to add storage processing power independently of the rest of the system. Since no need server to intermediate the transaction, thus it eliminates the potential server bottleneck, and an immediate benefit is improved scalability. Further, if spatial data storage objects distribute across many OSDs, then the aggregate I/O rate and data throughput rate will scale linearly. For example, a single OSD may be capable of delivering data at the 100 Mbps and 1000 storage I/O operations, but if spatial data is striped across 10 OSDs, then it can achieve the parallel accessing at 1000 Mbps and 10,000 storage I/O operations. It is more convenient to construct scalable massive spatial data storage system.

As mentioned above, the proposed storage mode adopts distributed storage architecture, it eliminates server bottleneck by communicating directly between storage devices and clients to get improved scalability and performance. Furthermore, it provides unified object-oriented storage model by combined object-based storage into GIS, which is suited to store unstructured spatial data. And it has higher security, scalability and performance, so it's a promising scalable object-oriented distributed storage mode.

4. CONCLUSIONS

We have introduced a new object-based storage technique and integrated it with GIS to present a scalable distributed storage mode for massive spatial data. We have illustrated spatial data storage based on OBS and the practicality of supporting scalable spatial data storage. We are in the process of establishing a prototype based on our presented mode. We plan to simulate the distributed storage environment of massive spatial data to acquire more detailed information about the trade-off in performance and storage load balance in near future.

REFERENCES

Intel Corporation, 2003. Object-based Storage: The Next Wave of Storage Technology and Devices, *Copyright © 2003, Intel Corporation,* http://turtle.ee.ncku.edu.tw/, (accessed 2 Apr. 2005).

Cluster File Systems Inc, 2002, Lustre: A Scalable High-Performance File System. pp13, http://www.lustre.org/docs/whitepaper/ (accessed 21 Mar. 2006).

Gaeda V., Gunther O.. 1998. Multidimensional Access Methods. *ACM Computing Surveys*, 30(2), pp.170-231.

Garth A.G, David F.N, Khalil A, 1998, A Cost-Effective, High-Bandwidth Storage Architecture , *Proceedings of the 8th Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, California.

Mesnier M, Gregory R.G, Riede E, 2003, Object-Based Storage. *IEEE Communications Magazine*, 41(8), pp. 84-90.

Thomas M. Ruwart, 2002. OSD: A Tutorial on Object Storage Devices, http://turtle.ee.ncku.edu.tw/ (accessed 21 Apr. 2005).

Panasas Inc, 2003, White Paper: Shared Storage Cluster Computing, pp22, http://www.panasas.com (accessed 12 Apr. 2005).

Yu Zhanwu, 2005. New Generation Storage Model for GIS Spatial Data Based on Object-Based Storage. In: *MIPPR 2005: Geospatial information, Data Mining, and Applications,* edited by Jianya Gong, Qing Zhu, Yaolin Liu, Shuliang Wang, Proc. Of SPIE Vol.6045, 60450M.

ACKNOWLEDGEMENTS

This research is supported by the National Key Basic Research and Development of China (No.2004CB318206).

* Corresponding author:

Zhou Yan, Ph.D Candidate

State Key Lab of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University
129 LuoYu Road, Wuhan,Hubei Province, P.R.China, 430079,

E-mail: swallow_anne@126.com