International Journal of Grid Distribution Computing
Vol.8, No.1 (2015), pp.25-32
http://dx.doi.org/10.14257/ijgdc.2015.8.1.03

# A Novel Architecture Design of Large-Scale Distributed Object Storage System

Shan Ying[1] and YAO Nian-min[2]

*1 College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China; 2 College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*
*shanyingsc@163.com*

***Abstract***

*A novel architecture design of large-scale distributed object storage system (called DOSS) is proposed. Our design takes several aspects effecting on the overall performance of DOSS into consideration, including an improved model of interaction based on the traditional interactive mode of object-based storage systems, MDS(Metadata Server) management scheme and a load balancing scheme combines the definition of the maximum load for OSD(Object Storage Device), hierarchical model and re-polling in order to solve access of heat objects issues. Our experimental results show that the architecture is an effective way to promote the performance of DOSS.*

***Keywords:*** *distributed object storage system; MDS management; model of data interaction; load balancing*

## 1. Introduction

Object storage system separates data path, control path and management path in order to provide scalability, high performance, security and data sharing cross platforms of storage service effectively [1~3]. With expanding scale of DOSS and diversity of user requirements, system complexity and requirements of different levels of service, architecture design shows more and more importance for promoting the performance of DOSS.

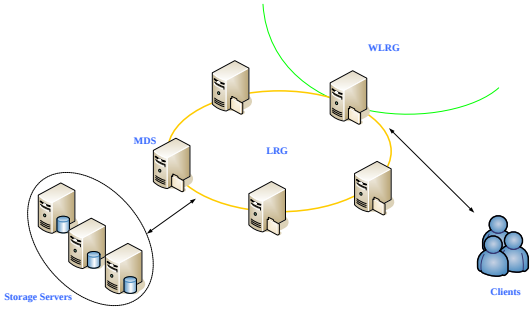
As we know, architecture design includes many aspects, while each design of the structure and level will have effect on the overall performance to some extent. Our design mainly takes the following aspects into consideration. Firstly, an improved model of interaction is proposed based on the traditional interactive mode of object-based storage systems [4, 5]. Secondly, a metadata management scheme is introduced which take full account of the master metadata selection from both the local and global performance improvement. Thirdly, a load balancing scheme proposed combines the definition of the maximum load for OSD, hierarchical model and re-polling in order to solve access of heat objects issues. Our interaction model of object-based storage and MDS management scheme can refer to the literature [6, 7].

## 2. Main Ideas and Problem Formulation

### 2.1. DOSS Architecture Model

The architecture model we proposed for distributed object storage is shown in Figure.1. While the primary parts and relation of the components and the implemental details is described as follows. In this figure, there are three main components: (1)Clients; (2) Storage servers; (3) MDSs. Local MDSs composes LRG, and multiple LRG composes WLRG. Each LRG connects to the storage servers for objects storage.

ISSN: 2005-4262 IJGDC



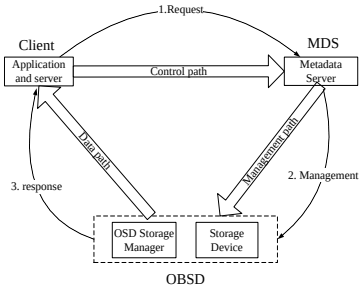




**Figure 1. Architecture Model for DOSS** **Figure 2. Interactive Model of DOSS**

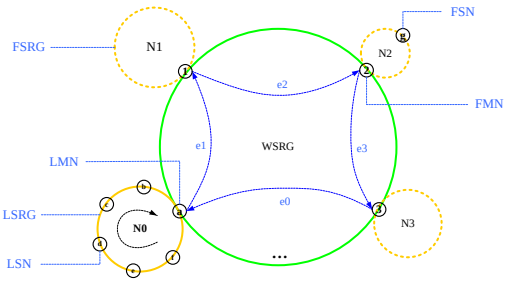
### 2.2. Interactive Mode of Data for DOSS

The interaction model of DOSS is shown in Figure.2. The interaction model has three paths: control path, data path and management path [8~13]. The three paths are separated.

The Interactive process of DOSS is described as follows:

1. Clients send request to MDS for only once, and then wait for receiving data.
2. MDS is no longer a transfer station, but a functional entity.
3. Because authorization and authentication of clients are accomplished by MDS, OSD trust MDS which means OSD will not authenticate the requests and commands from MDS.
4. OSD sends response data to client after accomplishing the OSD storage operation.
5. In the whole process, MDS has been a manager of OSD operation.

### 2.3. MDS Management Scheme



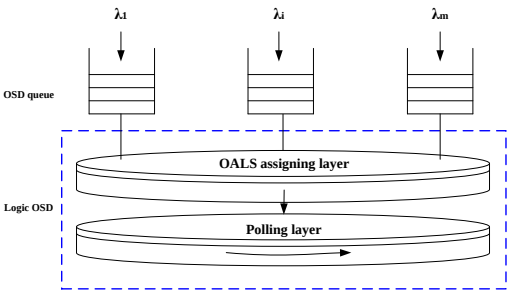




**Figure 3. The Topology Structure of Local Group and Global Group** **Figure 4. Layer Model**

In DOSS, clients from different regions send requests to the most appropriate MDS, especially in DOSS, MDS in different regions are jointly responsible for all kinds of operation of metadata, while the performance of r/w operation heavily depends on the topology structure of MDS in different geographical region.

In our scheme, MDS distributed in different regions are partitioning at first, and then MDS are divided into many groups. In order to simplify the discussion of the topology in the following section, each MDS is considered as a node. When a node in the group communicates with another node, a topology tree is constructed and the primary node of local group which is in the same group with request node is treated as the root of the tree.

Figure.3 shows the topology composed by WLRG and SRG. The SRG requested by clients is called LSRG, and other SRG in the WLRG is called FSRG. The Nodes in SRG are divided into two categories: Master Node (MN) and Secondary Node (SN). The Node requested by clients at a certain time becomes the MN, SN will substitute MN while topology is changed or some operation happened. Since SRG has two types of LSRG and FSRG, the MN and SN in LSRG is called as LMN and

LSN, and MN and SN in FSRG is called as FMN and FSN. The problem of virtual center nodes selection is treated as a similar k-center problem[14, 15].

### 2.4. Load Balancing Scheme for DOSS

In DOSS, clients from different regions send requests to popular object which may produces increasing of objects heat and load imbalance. Our load balancing scheme may be a effective way to solve the issue.

Definitions associated with our load balancing strategy are described as follows:

Definition 1: Set of objects $OB$: $\{ob_1,\cdots,ob_i,\cdots,ob_m\}$, $m$ is total number of objects, $ob_j \in OB$ and $ob_1 \cap \cdots \cap ob_i \cap \cdots \cap ob_m = \phi$ .Let set of storage nodes be $D:\{d_1,\cdots,d_j,\cdots,d_n\}$, and $n$ represents total number of storage nodes.

Definition 2: Let properties of objects be $ob_i = (oid_i, osz_i)$, $oid_i$ and $osz_i$ represent object identify and object size, respectively.

Definition 3: Disk model is $d_j = (dc_j, dt_j, dl_j, db_j)$ and $dc_j$ , $dt_j$ , $dl_j$ , $db_j$ disk represents disk capacity, transfer rate, the current load, and disk bandwidth respectively.

Definition 4: Set of requests $REQ$: $\{req_1,\cdots,req_k,\cdots,req_u\}$ , $req_k = (oid_k, rat_k, rst_k, rft_k)$, $rat_k$ , $rst_k$ , $rft_k$ represents starting time of requests, starting time and completion time of service, respectively.

Definition 5: Maximum load of each OSD is

$$loadm_j = (db_j / dbs)\sum_{i=1}^{m} h_i \ . \tag{1}$$

$db_j$ , $dbs$ , $h_i$ represents bandwidth of $d_j$, total bandwidth of disk, heat of object.

Definition 6: Average response time of requests is

$$MRT = mrt(REQ) = (1/u)\sum_{k=1}^{u} rp_k \ . \tag{2}$$

$u = |REQ|$ and $rp_k$ is response time of requests. Derivation of definition 5 and 6 is described as follows:

Let object requests arrival rate distribution obeys Poisson distribution, arrival rate is $\lambda$ , while object size obeys zipfian distribution. Requests probability of $ob_i$ is $p_i$ , by the Zipf law, $\sum_{i=1}^{m} p_i = 1$ [16,17]. Requests arrival rate of $ob_i$ is $\lambda_i = p_i \cdot \lambda$ , expected service rate is $\mu = E[\mu_i] = \sum_{o_i \in OB} p_i$ , bandwidth of $d_j$ is $db_j$ , the number of service requests at the same time $n_j = db_j / \sum_{ob_i \in OB} p_i \cdot obw_i$ , bandwidth of requests of $ob_i$ is $obw_i$ . Since queue service intensity of OSD obeys $\rho_j = \frac{\lambda_j}{n_j \mu} < 1$, then average waiting time is $ewt_j = \frac{\rho_j}{n_j \mu - \lambda_j}$ .

Since distribution of access of objects assumed is Poisson distribution, the heat of the object is $h_i = \lambda_i \cdot osrv_i$ . $osrv_i$ is fixed service time of $ob_i$ , $osrv_i = osz_i / dt_j$ . LEE [18] adopts with service time by order of the requested object to obtain the minimum difference between the disk load and service time for each disk. The formula can be seen by the service time that service time is proportional to the size of the object in the premise of differences of the storage device transfer rate, so we combine object

the size of the heat with the object and the object is arranged in order of size, in order to achieve minimum service time [19~22].

The strategy makes grading the different OSD in terms of OSD ability possible. Hierarchical model is shown in Figure.4. In this scenario, MDS sends objects to a local queue of OSD with FCFS rule. Logic OSD is logically divided into two levels, distribution layer and the polling layer.

In the distribution layer, the OSD layer is considered as a logical layer which has n logical storage device, the higher the level, the stronger the service capacity. OSD hierarchical is taken by logical classification of bandwidth, for example, if the bandwidth of $OSD_A$ is n times the bandwidth of $OSD_B$, then $OSD_A$ can be divided into n logic bandwidth, while the processing capacity of $OSD_A$ is n times $OSD_B$ [23,24]. Hierarchical level division achieved collaborative service of heterogeneous OSD and overall service performance.

Polling layer assigns the object to disk reasonably according to the characteristics of the object by way of semi-polling. Polling mode is adapted which OSD with high level has polling priority, while polling begins until level of $OSD_A$ without polling and level of $OSD_B$ are equal. Therefore, this approach assign objects of high heat to a different OSD according to service capabilities, then the object with similar size will be assigned to OSD with the same bandwidth in a balanced manner which takes different request levels and different service levels into account.

Since object waiting for allocation according to service in chronological order, the hottest document is assigned to different OSD in order to ensure load balancing.

Definition 7: Set of logic layer $L$: $\{L_1,\cdots, L_s,\cdots,L_v\}$, $v$ is number of divided logic layer.

Definition 8: The maximum load of $L_s$ is $loadl_s$, the total load of each OSD is sum of OSD load on its own layer, $load_j = \sum_{s=1}^{v} load_j^s$, $load_j^s$ is load on each logic layer $L_s$ of $OSD_j$. Allocation algorithm described as follows.

**Algorithm for the Strategy**

```
program Assignobject
Call algorithm LogicLevel to divide logic OSD;
All objects are ordered according service time of in ascending, perform the following steps for
object i ;
    Caculate loadm_j of OSD_j with fomula(2);
    If objects are not fully allocated, then call the
    function AssigneachLOSD in order to assign object
    to appropriate OSD;
end.
program AssigneachLOSD
For s = 1 and s ≤ v, polling L_s
   While k = 1 and k ≤ n, do
      If load_k^v ≤ loadl_v and load_k ≤ loadm_k
      Assigned ob_i to OSD_k ;
      load_j = load_j + load_i ;
   k = k + 1 ;
s = s + 1 ;
```

end.
program LogicLevel
Order OSD ascending according to bandwidth
$s = 1$, $v = 0$, $loadl_s = 0$
For $j = 1$ and $j \le n$, scan OSD sequentially
While $db_j - db_{j-1} \neq 0$, do
$loadl_s = db_j - db_{j-1}$, divided logical area is
layer $s$ of the logical OSD, $v = s$;
$s++$;
end.

OSD is divided into logic OSD with $v$ layers according to different bandwidth from $L_1$ to $L_v$. The bandwidth on the same layer is the same which means the maximum load of each layer on logic OSD $load_i$ is equal. Logic OSD in different levels have different functions, the higher the level, the stronger processing capability, in order to ensure the object of different heat distributing in different OSD. Meanwhile, time of polling of OSD with strong service capacity increase to ensure the balance of heat and size distribution of objects in the OSD, thereby minimizing heat problems of object accessed. Since service capacity of OSD is different, the lower OSD level has the weaker service capacity, on which some requests with little heat and long time service is suitable processed.

## 3. Performance Evaluation

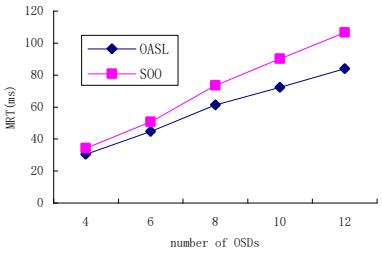
In our experiment, we employ a simulation to evaluate the proposed architecture. We first introduce experimental settings. We used NS-2 as the tool to establish overall architecture of DOSS and disksim4.0 as tool to realize the OSD nodes.

Our experiment follows several premises. Firstly, the access of large files relative to transmission time, seek and rotational latency delay can be ignored. Secondly, Poisson distribution to reach the object, object access obeys Zipf distribution. Thirdly, queuing model obeys M/G/1 model, service rules obeys FCFS rule.

As for the heterogeneous case of OSD, different properties of OSD are tested. To make test results more clearly, we use the OSD which is not identical to simulate heterogeneous OSD. We compare our strategy (OASL for short) with a simple pure polling [16, 18] the case (SOO for short) the object allocation strategy in order to evaluate two different distribution strategies for the impact on system performance over time, our test does not consider different batches of the distribution of the request.

With the increase number of requests in the OSD, performance difference of OSD will change accordingly. For this problem, we tested the changes MRT while the OSD were 4, 6, 8, 10 and 12. The result in Figure.5 shows the two strategies has little difference while number of OSD is 4. However, with the increase of the number of OSD, OASL strategies produces small impact on MRT than SOO MRT, but this gap increases with the number of OSD gradually widening. In our analysis, as taking into account the object size and capacity of two disk services, OASL strategy continue to show its better than SOO with the increasing number of OSD. From the figure, the trend curves shows that the growth rate of MRT in OASL shows slow growth than that of the MRT in SOO, with the increasing number of OSD, the increasing the number of OSD has little influence on MRT in OASL relatively. It can be seen that MRT changes are relatively stable with the increasing number of OSDs.



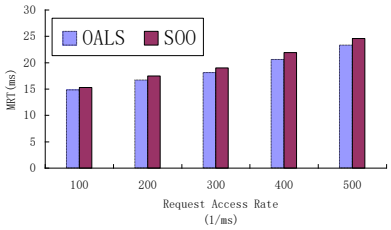




**Figure 5. MRT of Different Number OSD** **Figure 6. MRT of Different Request Access Rate**

MRT are tested with the rate of requests access rate changing in Figure.6. MRT is tested when request access is 100,200,300,400,500 (1/ms). The figure shows the impact of MRT in OASL and SOO strategy with the request arrival rate increasing, however, as the request arrival rate in the same circumstances, OASL strategy obtains less MRT than SOO. So we speculate that it is closely related the distribution of objects request on one hand and layered allocation of the OSD bandwidth on the other hand. It can also be seen from the figure, with the request arrival rate increases, OASL strategy has less influence on growth rate of MRT than SOO.

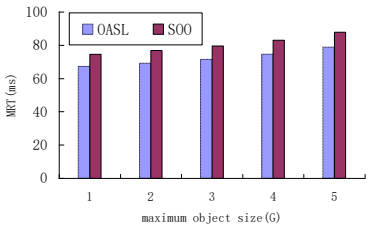



**Fig. 7. MRT of Maximum Object Size**

In the distribution of the object by way of polling, the relatively large size of the object will significantly affect the performance of load balancing strategy, so large objects are adopted to test performance of SOO and OASL strategy. Experiment results shown in Figure.7 shows MRT increased for SOO and OASL with the maximum object size increasing. However, you can see, when the largest object is the same, MRT in OASL is always less than in SOO, as the largest size of object increases, the growth rate of MRT in OASL is less than SOO. When the biggest object is of relatively large, OASL showed excellent performance.

## 4. Conclusion

In this paper, we design the architecture of DOSS. Our design takes several aspects effecting on the overall performance of DOSS into consideration, including improved model of interaction, metadata management scheme and load balancing scheme. Our experimental results show that the architecture is an effective way to promote the performance and ensure the load balancing.

In our immediate future work, we will take dynamic load balancing scheme into our consideration which is known as a significant aspect in object storage system to ensure communication reliability and distribution rationality.

## Acknowledgement

This work is supported by the Fundamental Research Funds for the Central Universities No.DL13BBX03 and Fundamental Research Funds for the Central Universities: HEUCFT1202,HEUCF100609.

## References

[1] M. Mesnier, G. R. Ganger and E. Riedel, "Object-Based Storage", Communications Magazine, vol. 41, no. 8 **(2003)**, pp. 84-90.
[2] A. Azagury, V. Dreizin, M. Factor, E. Henis, D. Naor, N. Rinetzky, O. Rodeh, J. Satran, A. Tavory and L. Yerushalmi, "Towards an object store", Proc. 20th IEEE/11th NASA Goddard Conf. Mass Storage Systems and Technologies, **(2003)**.
[3] M. M. Factor, K. Meth and D. Naor, "Object Storage: The Future Building Block for Storage Systems", Proceedings of the 2nd International IEEE Symposium on Mass Storage Systems and Technologies, **(2005)**.
[4] M. Mesnier, G. R. Ganger and E. Riedel, "Object-Based Storage pushing more functionality into storage", Communications Magazine, vol. 34, no.17, **(2008)**, pp. 88-90.
[5] D. H. C. Du, "Advancements and Future Challenges of Storage Systems", Proceedings of the IEEE, **(2008)**.
[6] S. Ying, Y. Nianmin and Z. Jianming, "Research on Architecture of Object-based Storage System", Journal of Computer Research and Development, vol. 46, **(2009)**, pp.198-202.
[7] S. Ying and Y. Nianmin, "Research on MDS Selection Scheme in Distributed Objected Storage System", Journal of Wuhan University of Technology, vol. 33, no. 7, **(2011)**, pp. 143-146.
[8] D. Feng and H. Liu, "Scheduling in Huge Object-based Storage System", Proceeding of Japan-China Joint Workshop on Frontier of Computer Science and Technology (FCST), **(2006)**.
[9] Yu Hua, Y. Zhu, H. Jiang, D. Feng and L. Tian, "Scalable and Adaptive Metadata Management in Ultra Large-scale File System", Proceedings of the 28th International Conference on Distributed Computing Systems, **(2008)**.
[10] S. A. Brandt, E. L. Miller, D. D. E. Long and L. Xue, "Efficient Metadata Management in Large Distributed Storage Systems", Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage System and Technologies, **(2003)**.
[11] W. Lin, Q. Wei and B. Veeravall, "A Weight-based Metadata Management Strategy for Petabyte-scale Object Storage Systems", Proceedings of the fourth international workshop on Storage Network Architecture and Parallel, **(2007)**.
[12] Q. Liu, D. Feng and F. Wang, "Resrarch on MetaData Server of High Reliability", Computer Engineering, vol. 34, no. 17, **(2008)**, pp. 88-90.
[13] Y. Hua, D. Feng and B. Xiao, "TBF:An Efficient Data Architcture for Metadata Server in the Object-based Storage Network", Proceedings of International conference on Networks (ICON), **(2006)**.
[14] J. Li, Xu Liu and J. Zhu, "r-Dominating Set Problem and k-Center Problem in Weighted Trees", OR Tranansaction, vol. 13, no. 2, **(2009)**, pp. 111-118.
[15] S. Khuller and Y. J. Sussmann, "The capacitated k-center problem: SIAM Journal on Discrete Mathematics", vol. 13, **(2000)**, pp. 403-418.
[16] T. Xie and Y. Sun, "A File Assignment Strategy Independent of Workload Characteristic Assumptions", ACM Transactions on Storage, vol. 5, no. 3, **(2009)**.
[17] Z. Zeng and B. Veeravalli, "On the Design of Distributed Object Placement and Load Balancing Strategies in Large-Scale Networked Multimedia Storage Systems", IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 3, **(2008)**, pp. 369-382.
[18] L. W. Lee and P. Scheuermann, "File Assignment in Parallel I/O Systems with Minimal Variance of Service Time", Transactions on Computers, vol. 49, no. 2, **(2000)**, pp. 127-140.
[19] P. Scheuermann, G. Weikum and P. Zabback, "Data partitioning and load balancing in parallel disk systems", The VLDB Journal, vol. 7, **(1998)**, pp. 48-66.
[20] Z. Gongye, L. Wei and C. Jincai, "Hotspot data balancing in OBS", Journal of Huazhong University of Science and Technology (Nature Science Edition), vol. 35, no. 12, **(2007)**, pp. 28-31.
[21] W. Fang, Z. Shunda, F. Dan and Z. Lingfang, "Hybrid object allocation policy for object storage systems", Journal of Huazhong University of Science and Technology (Nature Science Edition), vol. 35, no. 3, **(2007)**, pp. 46-48.
[22] L. J. Qin, D. Feng, L. F. Zeng and Q. Liu, "Dynamic Load Balancing Algorithm in Object-Based Storage System", Computer Science, vol. 33, no. 5, **(2006)**, pp. 88-91.
[23] Z. Shengli, "Method of Data Assignment on Heterogeneous Disk System", Mini-Micro Systems, vol. 25, no. 11, **(2004)**, pp. 1970-1974.
[24] S. A. Weil, K. T. Pollack and S. A. Brandt, "Dynamic Metadata Management for Petabyte scale File System", Proceedings of the 11th International Conference on High Performance Computing (HiPC), **(2004)**; Bangalore, India.

# Authors

**Shan Ying**. Lecturer of College of Computer Science and Technology, Harbin Engineering University. Born in 1981, received Ph.D. degree from College of Computer Science and Technology, Harbin Engineering University. She majors in network storage and cloud storage.

**YAO Nian-min.** Professor and Ph.D. supervisor of College of Computer Science and Technology, Harbin Engineering University. Born in 1974, member of information storage technology specialty committee in China Computer Federation. He majors in Wireless Sensor Networks and cloud storage system.