Front. Electr. Electron. Eng. China 2008, 3(2): 171–175
DOI 10.1007/s11460-008-0037-z

RESEARCH ARTICLE

Jiaxing SONG, Chuang LIN, Weidong LIU, Shaoyu CHEN

Replica location mechanism in data grid based on ED-Chord

Abstract A peer-to-peer hierarchical replica location mechanism (PRLM) was designed for data grids to provide better load balancing capability and scalability. Global replica indexes of the PRLM are organized based on even distributed Chord (ED-Chord) structure. The locality can optimize queries on local replica indexes of virtual organizations. ED-Chord protocol collects the node identifiers information using a distributed method and assigns optimal identifiers for new nodes to make them more uniformly distributed in the entire identifier space. Theoretical analysis and simulations show that PRLM provides good performance, scalability and load balancing capability for replica location in data grids.

Keywords grid, peer-to-peer, replica location, load balance

1 Introduction

A data grid is characterized by its tremendous data size, heterogeneous structure and wide area distributed data storage sites [1]. Therefore, data replication is usually adopted to reduce access latency, improve data locality, and increase reliability, availability and performance. In data grids, a data file usually has several replicas. How to locate the physical copies by its unique logical name of a data file is referred to as replica location problem, which is one of the hot research topics in data grids.

A centralized directory service is provided in a globus data grid to locate replicas [2], which limits the scalability and reliability. Decentralized, adaptive mechanism compresses all replica location information of the whole system and stores it on one local node of each virtual organization (VO) for replica location [3]. Although it can achieve good performance, storage and update cost is the biggest problem. Dynamic self-adaptive replica location method distributes global replica location information on multi-host nodes uniformly by dynamic mapping [4], which has good scalability and performance for replica location. In this method, since each host node must maintain contact information with others, storage and update also cost too much.

For resource sharing, peer-to-peer (P2P) [5] structure has great advantage on scalability, adaptability and load balance. Chord [6] is one of the most famous P2P structures. In this paper, we improve on Chord with better load balancing capability, which is called even distributed chord (ED-Chord), and use it in data grids for replica location. We design a P2P-based hierarchical replica location mechanism (PRLM). Theoretical analysis and simulations show that PRLM provide good performance, scalability and load balancing capability for replica location in data grids.

Translated from *Journal of Tsinghua University* (*Science and Technology*), 2007, 47(1): 123–126 [译自: 清华大学学报 (自然科学版)]

Jiaxing SONG (✉), Chuang LIN, Weidong LIU, Shaoyu CHEN
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
E-mail: jxsong@tsinghua.edu.cn

2 ED-Chord: improved load balancing DHT structure

2.1 Load balance of Chord

Chord organizes all nodes in a circle and provides fast distributed computation of a hash function mapping keys to nodes responsible for them. The consistent hash function assigns each node and key an m-bit identifier using a base hash function such as SHA-1. A node's identifier is chosen by hashing the node's IP address, while a key identifier is produced by hashing the key. Consistent hashing assigns keys to nodes as follows. Identifiers are ordered in an identifier circle modulo 2^m. The key is assigned to the first node whose identifier is equal to or follows k in the identifier space. This node is called the successor node of key k. If identifiers are represented as a circle of numbers from 0 to $2^m - 1$, then the successor node of key k is the first node clockwise from k.

The load balancing capability of Chord depends on how uniformly node identifiers cover the entire identifier space. Due to the random character of IP addresses, node identifiers cannot cover the identifier space uniformly. As

a result, keys cannot be allocated to nodes evenly, which has been proven by experiment results.

To improve the load balancing capability of Chord nodes, we present the ED-Chord structure. In ED-Chord, all node identifiers can cover the identifier space more uniformly than in Chord, so that keys can be allocated to nodes evenly.

2.2 Design principle of ED-Chord

We define relative standard deviation of the distance between neighboring Chord nodes to evaluate the uniform degree of node identifiers in the identifier space. The distance between two neighboring Chord nodes is the difference of their identifiers. If there are N nodes in Chord circle and the distance between neighboring nodes is $x_1, x_2,\ldots, x_N$, then the average is $\bar{x}=\frac{1}{N}\sum_{i=1}^{N} x_i$, the sum is $A=\sum_{i=1}^{N} x_i$, and the standard deviation is $\sigma=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2}$. According to Chord protocol, A is a fixed value for different N. In addition, a bigger N means a smaller σ. To provide comparability of distance standard deviation for different N, we define relative standard deviation as $\varphi=\sigma/\bar{x}$. When A is fixed and N is variable, relative standard deviation is able to show how uniformly node identifiers cover the whole identifier space.

While we obtain all the identifiers of nodes in Chord and the distances between neighbors, we should select an appropriate identifier for a new node to make the identifier space uniformly covered. If there are N nodes now, since relative standard deviation has been adopted to represent how evenly the identifiers are distributed, we can select a distance and divide it into two parts, and then get $N+1$ distances. The selected distance and the position to divide must guarantee that $\Delta\varphi = \varphi_{N+I} - \varphi_N$ is minimal, φ_N is the relative standard deviation of N nodes, and φ_{N+1} is the relative standard deviation of $N+1$ nodes (including the new node). If x_m is the maximal distance and s is a division factor which can divide x_k into two parts of $x_k \times s$ and $x_k \times (1-s)$, where $0<s<1$, $1 \leqslant k \leqslant N$, then it has been proven that when $x_k = x_m$ and $s = 0.5$, $\Delta\varphi$ will get minimal value. This result shows that when the new $(N+1)$th node joins, its identifier position should be just in the middle of the maximal distance.

2.3 Identifier assignation of ED-Chord

It is assumed that node identifier space is 2^m, and then we can use the following procedure to assign an identifier I_n to a new node n of the ED-Chord E.

Procedure of identifier assignation

Begin

1) From any node of E, every direct successor node is visited in turn. Thus the identifier of each node and the number of all nodes N_c can be obtained. During the process, node k identified by I_k will be found. The distance between node k and its direct successor node is maximal, which is represented by D_{max}.

2) If $N_c = 0$, then $I_n = 0$, go to end.

3) If $N_c = 1$, then $I_n = (2^{m-1}+I_1) \bmod 2^m$, go to end.

4) If $N_c > 1$, then $I_n = (I_k + D_{max}/2) \bmod 2^m$, go to end.

End

The main difference between ED-Chord and Chord is that they have different methods of assigning the node identifier. After a new node in ED-Chord gets its identifier, it will join, leave and search keys using Chord protocol. Consequently, ED-Chord has the same good properties of Chord besides better load balancing capability.

In ED-Chord, the following three conditions should be satisfied, which also limits the area where ED-Chord can be used.

1) The number of nodes is not too large, which is usually on the level of 10^1–10^3. Furthermore, the time delays between nodes should be short.

2) The nodes in ED-Chord must have good stability, they cannot join and leave ED-Chord frequently.

3) The nodes in ED-Chord require highly good load balancing capability.

3 P2P-based replica location mechanism

3.1 Architecture of PRLM

Every data file in data grids has a logical data name (LDN) as its global identifier. A data file may own some replicas which are identified by physical data names (PDN). The problem of replica location is how to find one or more PDNs of a data file by its LDN.

Figure 1 shows the architecture of P2P-based replica location mechanism (PRLM). In the figure, SNs represent data storage node, LRIs represent local replica index

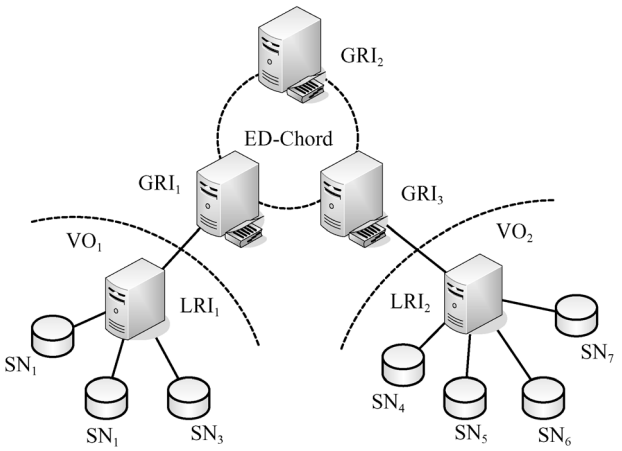

Fig. 1 Architecture of PRLM

nodes, and GRIs represent global replica index nodes. SNs, LRIs and GRIs are organized on three levels, so the architecture of PRLM has a hierarchical structure.

Data files and their replicas are distributed in SN nodes. Each VO generally has at least one LRI node. The LRI node can not only store and maintain local replica index but also provide replica registration and location service for SN nodes. Furthermore, there is a cache in each LRI node to store some replica information that has been searched recently. GRI nodes are on the top level of PRLM and organized with a P2P structure based on ED-Chord protocol. The primary task of GRI nodes is to establish the mapping from data files LDN to LRI. There are also global replica index caches in GRI nodes to store some mapping data recently.

Owing to some properties and requirements of PRLM, GRI nodes are organized using ED-Chord. First, the replica location service is very important in data grid, since it is accessed frequently, therefore it must have high performance and low time delay. Second, the replica location service has to deal with large amount of data, thus its scalability and load balancing capability should be guaranteed. Finally, the global replica index nodes are usually servers with high performance and bandwidth. The number of them cannot be very large; besides, these servers cannot join and leave with high frequency. In a word, the ED-Chord protocol is suitable for PRLM.

3.2 Working procedure of PRLM

GRI nodes in PRLM get their identifiers by ED-Chord protocol and finish other operations by Chord protocol, such as joining and leaving PRLM. LRI nodes are on the middle level of PRLM, on one hand they provide local registration and location service for SN nodes in VOs. On the other hand they also provide global registration and location service through GRI nodes on the top level. In each VO, the LRI node maintains internal replica location information. While a data replica can be found inside the virtual organization, it is unnecessary for the LRI node to send queries to GRI nodes. At the same time, cache is used in the LRI node, so some global replica location information can be accessed directly. Space locality and time locality are made use of fully.

If a SN node wants to find some replicas of a data file, they will take the following steps. First, it sends a replica location request to the service provided by the LRI node, LDN of the data file is given out in the request. Second, the LRI node searches its local replica index cache. If the desired replica information exists in the cache, they will be returned to the SN node which has sent the request. Otherwise if desired replica information does not exist, the LRI node will use SHA-1 Hash function to allocate a key to the LDN and send the key to the GRI node connected with it. Third, when the GRI node gets the key, it will begin searching mappings from LDN to LRI. If desired mappings can be found in its cache, the GRI node will return them to the source LRI node. Otherwise if there is no desired mapping in the cache, a global search for the key will be started in GRI nodes by ED-Chord protocol. The searching results will be still returned to the source LRI node. Finally, when the source LRI node gathers the information of destination LRI nodes, it will send replica location query requests to those destination LRI nodes. After searching in their replica location indexes, the destination LRI nodes will send location information about replicas to the source LRI node. At the same time, the source LRI node will store this replica location information in its cache. When the SN node, which sent the original replica location request, receives PDNs of replicas, it will be able to visit the replica directly.

3.3 Performance analysis of PRLM

1) Time delay

Assume that the average time delay from SN nodes to LRI nodes is l_1; the average time delay from LRI nodes to GRI nodes is l_2; the average time delay between neighboring GRI nodes is l_3; the request processing time in LRI nodes is p_1 and the request processing time in GRI nodes is p_2. If there are N GRI nodes, according to ED-Chord protocol, each replica location query needs O (lb$N$) hops to arrive at the destination GRI node. As a result, the whole average time delay for a global replica location is $L_Q = l_1 + p_1 + l_2 + \lambda(1\text{b}N)(p_2 + l_3)$, where λ is a constant. So the response time of global replica location has a logarithmic increase along with N. When a new GRI node joins PRLM, it will visit all the existing GRI nodes in ED-Chord circle. Thus the average time delay for GRI nodes to join PRLM is $L_J = N(p_2 + l_3)$, which has a linear increase with N. Because there are not too many GRI nodes in PRLM and these nodes will not join and leave PRLM frequently, this time delay is acceptable.

2) Load balance

Because GRI nodes are organized in an ED-Chord circle, their identifiers cover the identifier space uniformly. For any replica in SN nodes, it has a key identifier by hashing its LDN. According to ED-Chord protocol, all the key identifiers are distributed in GRI nodes evenly. As a result, every GRI node will maintain replica location information equally, so that the load balancing capability of PRLM can be guaranteed.

3) Scalability

The scalability of PRLM appears in two aspects. On one hand the average replica location time delay is scalable, because it has a logarithmic increase along with the number of GRI nodes. On the other hand, according to ED-Chord protocol, the number of routing table items in each GRI node is scalable, because it also has a logarithmic increase along with the number of GRI nodes.

4 Simulation experiment results

4.1 Results of ED-Chord

In the simulation experiment, there are N nodes in ED-Chord circle. After these nodes join and leave ED-Chord circle for thousands of times, the relative standard deviation of distance between neighboring nodes can be calculated. Five main parameters in the simulation are given below and Fig. 2 shows the experiment results.

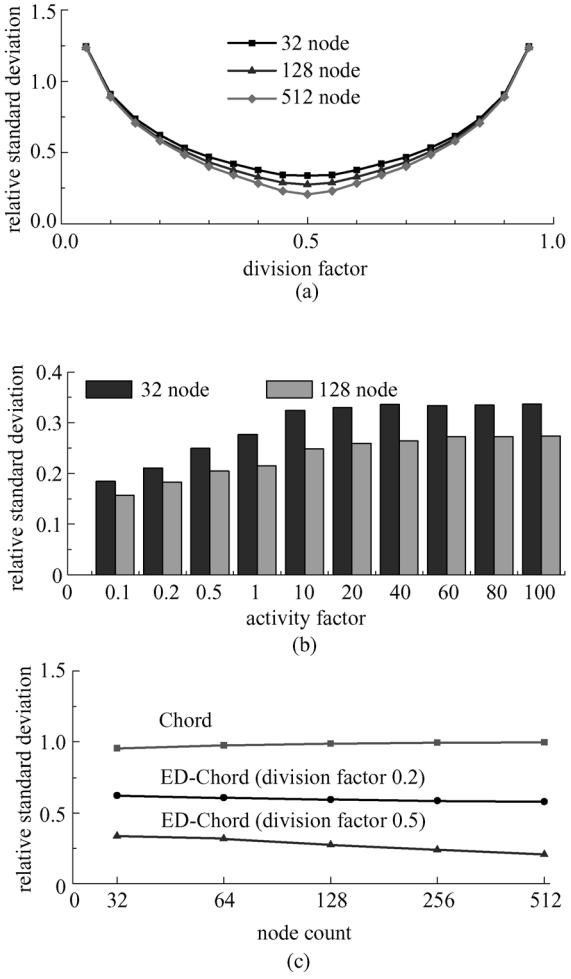

Fig. 2 Simulation results of ED-Chord. (a) Division factor and relative standard deviation; (b) activity factor and relative standard deviation; (c) nodes distribution comparison between ED-Chord and Chord

1) N: number of nodes in ED-Chord circle.

2) r: activity factor, it controls that nodes join and leave ED-Chord for $N \cdot r$ times.

3) c: range factor, it controls that the number of nodes is in the field of $N \pm Nc$.

4) s: division factor, it controls how to divide the maximal distance between neighbor nodes.

5) t: repeating times of simulation experiment, whose default value is 1000.

When r is 100 and c is 0.05, Fig. 2(a) shows the values of relative standard deviation with different s and N. It is obvious that 0.5 is the best division factor. When s is 0.5 and c is 0.05, Fig. 2(b) shows the values of relative standard deviation with different r and N. Two conclusions can be drawn from Fig. 2(b). On one hand the relative standard deviation will increase when nodes join and leave continually, but the increment is limited; on the other hand the relative standard deviation will decrease when the number of nodes becomes larger, which indicates that node identifiers distribute more uniformly. When r is 100 and c is 0.05, Fig. 2(c) shows the values of relative standard deviation with different s and N. In Fig. 2(c) the relative standard deviation of ED-Chord is obviously less than that of Chord, so ED-Chord will provide better load balancing capability.

4.2 Results of PRLM

The performance analysis of PRLM is also tested by simulation experiment. Parameters in Sect. 4.1 are the same with this simulation. When r is 1, c is 0.05, s is 0.5, and t is 100, for different N with 10000 operation of replica location, the experiment records the average response time L_Q and the average number of route table items in GRI nodes. Experiment results show that both of them have logarithmic increase along with N. Thus it is proved that GRI nodes in PRLM will provide better load balancing capability. Under the same experiment conditions as above, the relative standard deviation for the number of keys on GRI nodes is calculated. When GRI nodes are organized by ED-Chord, this relative standard deviation is less than that by Chord. This also indicates that PRLM has good load balancing capability.

5 Conclusions

In this paper, an ED-Chord structure based on the relative standard deviation concept is presented, which has better load balancing capability than Chord. At the same time, the procedure of identifier assignation for ED-Chord nodes is given. Furthermore, a PRLM of data grids is designed based on ED-Chord. Theoretical analysis and simulation results show that node identifiers in ED-Chord can cover the entire identifier space uniformly than in Chord, and PRLM can provide good performance, scalability and load balancing capability for replica location in data grids.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 90412012).

References

1. Chervenak A, Foster I, Kesselman C, et al. The data grid: towards an architecture for the distributed management and

analysis of large scientific datasets. Journal of Network and Computer Applications, 2000, 23(3): 187–200
2. Vazhkudai S, Tuecke S, Foster I. Replica selection in the globus data grid. In: Proceedings of 1st IEEE/ACM International Conference on Cluster Computing and the Grid. Brisbane, Australia: IEEE Press, 2001: 106–113
3. Ripeanu M, Foster I. A decentralized, adaptive, replica location mechanism. In: Proceedings of 11th IEEE International Symposium on High Performance Distributed Computing. Edinburgh, Scotland: IEEE Press, 2002: 24–34
4. Li D S, Xiao N, Lu X C, et al. Dynamic self-adaptive replica location method in data grids. In: Proceedings of IEEE International Conference on Cluster Computing. Hong Kong: IEEE Press, 2003: 442–446
5. Kant K, Iyer R, Tewari V. A framework for classifying peer-to-peer technologies. In: Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid. Berlin, Germany: IEEE Press, 2002: 368–375
6. Stoica I, Morris R, Karger D, et al. Chord: a scalable peer-to-peer lookup service for internet applications. In: Proceedings of ACM SIGCOMM 2001. California: ACM Press, 2001: 160–177