SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2012 Vol. 55 No. 3: 714–722
doi: 10.1007/s11432-011-4447-9

The optimization of replica distribution in the unstructured overlays

FENG GuoFu[1,2*], LI WenZhong[2], LU SangLu[2],
CHEN DaoXu[2] & BUYYA Rajkumar[3]

[1]*School of Information Science, Nanjing Audit University, Nanjing* 210029*, China;*
[2]*State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing* 210093*, China;*
[3]*Department of Computer Science and Software Engineering, The University of Melbourne, Victoria* 3010*, Australia*

Received February 28, 2010; accepted April 15, 2011; published online January 2, 2012

Abstract Replication is a widely used technique in unstructured overlays to improve content availability or system performance. A fundamental question often addressed by previous work focused on: how many replicas ought to be allocated for each data item given the fixed query rates and limited storage capability? In this paper, we have put forth two optimal replica distributions to achieve the highest success rate and the lowest message consumption. Especially, we have investigated the influence of item size on replica distribution. Our results show that Square-Root Replication, which is traditionally considered to be optimal, is not always the best choice. Our study offers a new deep understanding of resource managment in self-organized unstructured overlays.

Keywords unstructured overlay, item size distribution, query rate, success rate, expected search size

Citation Feng G F, Li W Z, Lu S L, et al. The optimization of replica distribution in the unstructured overlays. Sci China Inf Sci, 2012, 55: 714–722, doi: 10.1007/s11432-011-4447-9

1 Introduction

Purely decentralized unstructured overlay is commonly used in distributed systems, such as P2P (Peer-to-Peer), web service and WSN (Wireless Sensor Network). In unstructured overlays, the nodes are organized to form an ad hoc network. When a node joins the network, it connects to a set of neighbors, through which the queries are propagated among the nodes to locate the requested data items. In traditional overlays, the structure is unrelated to the data content; thus a node has no idea about which node can better resolve the queries.

Replication is a widely used technique in unstructured overlays to improve the system performance. A frequently addressed problem in related work on replication is how many replicas the system ought to keep for each data item. Usually, we allocate the limited storage to the items carefully based on the system heterogeneity, e.g., the query rates (popularity) and item size distribution. Due to the different system

*Corresponding author (email: njufgf@gmail.com)

objectives, the final optimal replica distributions are different and sometimes they are incompatible. For example, success rate may be very low when the system achieves the lowest message consumption.

One of the most elegant and widely recognized methods is the Square-Root Replication (SRR) proposed by Cohen and Shenker [1]. Cohen and Shenker define search size as the number of probed nodes before the query is resolved. They argue that for a fixed network and a set of data items, in an optimal solution for minimizing the expected search size, the number of replicas of an item ought to be proportional to the square root of its global query rate and be proportional to its item size.

However, our study shows that there are two drawbacks in Square-Root Replication. Firstly, the viewpoint that the number of replicas ought to be proportional to the item size is a bit unrealistic. Intuitively, the number of replicas has an inverse relationship with the item size in order to utilize the storage effectively. Secondly, the objective to minimize the message consumption is not sufficient for a high system performance because it may decrease the content availability. Then both success rate and search size should be taken into consideration in the design of overlays.

In this paper, we have put forth two optimal replica distributions to achieve the highest success rate and the lowest message consumption. Especially, we have studied the influence of item size on the replica distribution. Contributions of this paper include:

(1) We put forward an optimal replica distribution in terms of expected search size.

(2) We put forth a replica distribution to achieve the highest success rate.

(3) Item size is taken into account in the optimizations. When the items have the equivalent size, our conclusion is in accord with the traditionally optimal distribution SRR.

(4) We synthesize these two different objectives and finally offer a compromised advice on the overlay design.

(5) Our analysis and simulation results show that the distribution of SRR, which is traditionally considered to be optimal, is not always the best choice in the self-organized unstructured overlays.

This paper is organized as follows. Section 2 provides some related work on optimization of replication in unstructured overlays. Section 3 presents the model and problem definition. Two replica distributions in terms of expected search size and success rate are put forth in Section 4 and Section 5 respectively. In Section 6, we have carried out simulations to validate our analysis. Finally, Section 7 concludes our paper.

2 Related work

Replication is often used in unstructured overlays to improve the system performance. Among all the prior work on replication, the problem of replica distribution is often addressed. If we take the heterogeneity factors in item size and request rate into account and allocate a proper number of replicas (i.e., the storage space) for each item, the system performance, such as success rate and expected search size, is expected to be improved when the system storage capability is fixed [2–4]. The related work shows that the replica distribution has different effects on the system performance from many aspects and consequently the optimal replica distributions are different when we set different objectives to optimize.

Among related replica distributions, Uniform Replication, Proportional Replication, Square-Root Replication and Logarithmic Replication are usually used [5, 6]. In Uniform Replication, all the items have the same replicas, while in Proportional Replication the number of replicas of an item is proportional to its request rate. In Square-Root Replication and Logarithmic Replication, the number of replicas of an item is proportional to the square-root of request rate and the logarithm of request rate respectively.

The expected search size means the network bandwidth consumption during the searches. Since the statistics about Gnutella shows that the search consumes 40% bandwidth of the whole application, it is of great necessity to minimize the expected search size by the optimization of replica distribution[1)]. Cohen and Shenker [1] have examined Uniform Replication and Proportional Replication. The results show that these two replications have the same effect in minimizing the expected search size. Further,

1) Goldman A. Building a better P2P delivery system, 2003, available: http://www.isp-planet.com

Square-Root Replication, which can bring the minimum expected search size, is put forward. Moreover, Cohen and Shenker argue that the number of replicas of an item ought to be proportional to its size to minimize the expected search size, but they did not demonstrate its validity.

However, we have found that Square-Root Replication is not always the best choice. The viewpoint that the number of replicas ought to be proportional to the item size is unrealistic. Intuitively, the number of replicas has an inverse relationship to the item size in order to utilize the storage effectively because all the searches have the equal efficacy in terms of satisfying the users' queries despite of the item size.

Simultaneously, there are some researchers paying much attention to the improvement of success rate through the proper replica distribution. Feng et al. [7] studied the influence of overlay structure on the content availability in the unstructured overlay with the routing method of Random Walks, and verified that the success rate of an item is a function of the number of randomly deployed copies. And finally ref. [7] proposes a logarithmic distribution achieving the highest success rate, where the number of replicas of an item is a logarithmic function of its query rate. But this Logarithmic Replication ignores the influence of item size on the replica distribution.

However, it must be pointed out that replica distribution has different effects on the system performance from many aspects. Neither the objective to minimize the expected search size nor the objective to maximize the success rate is enough for a high performance system. For example, Square-Root Replication can bring us the lowest expected search size but perhaps it decreases the content availability; Logarithmic Replication can lead to the highest success rate but it may result in the waste of network bandwidth. Then both factors of success rate and expected search size should be taken into consideration in the optimizations. In this paper, two replica distributions are put forth to achieve these two different objectives respectively, and then a compromise between them is made for the practicable applications.

Moreover, many researchers work on distributed replication, mainly elaborating the replication problems of when, where and how many according to different objectives and different application environments [8–12]. Shmoys et al. [10] propose a proportional replication, in which the number of replicas of an item is proportional to its request rate. This distribution is optimal in terms of the consumed network bandwidth per download, and simulations show that local storage management algorithms like LRU automatically achieve near-proportional replication. The research in [8] shows that for an overlay with a random graph topology where item replicas are uniformly distributed, the hop distance to a replica of an item is logarithmic in the number of replicas. This conclusion educes that flooding-based search time is optimized when the number of replicas is proportional to request rates. However, the focus in this paper is unstructured overlay with routing algorithm of Random Walks, but not flooding.

3 The model of unstructured overlay

We assume that the overlay is composed of N nodes. All the nodes belong to a single strongly connected random gragh topology with bidirectional interconnections. Since it is infeasible for a node to store all the addresses of other nodes in a large network, a node only stores a subset and the remaining nodes are reached via the neighbor nodes. When a node requests an item, a search for the item is triggered and other neighbor nodes relay the query if the item is not available locally. Every node shares a uniform storage capacity c to improve the system performance.

There are m distinct items with the size $v(v_1, v_2, v_3, \ldots, v_m)$ in the overlay. Any item has more than one copy in the system. They are randomly distributed among the overlay. Each item has a query rate associated with it, reflecting users' interest. The query rate vector $q(q_1, q_2, q_3, \ldots, q_m)$ takes the form $q_1 \geqslant q_2 \geqslant q_3 \cdots \geqslant q_m$ with $\sum_{i=1}^{m} q_i = 1$. The query rate q_i is the fraction of all queries that are issued for the ith item. r_i denotes the number of copies of the ith item (including the original object). The distribution is represented by a vector $r(r_1, r_2, r_3, \ldots, r_m)$. A replication strategy is a mapping from the query rate distribution q and item size distribution v to the replica distribution r. The problem addressed in this paper is how to find a proper mapping from the query rate distribution q and item size distribution v to the replica distribution r to minimize the expected search size or to maximize the success rate.

We assume that the nodes randomly issue queries, i.e., every item has the same chance to be issued in

each search. We adopt the routing algorithm of Random Walks [5, 13] to propagate the queries among the overlay. Each walker continues to go forward until the requested item is found or path length reaches TTL. In this paper, L represents TTL and the number of walkers is 1 since every walker is independent of others. Then the expected search size A can be expressed as

$$A = ls + L(1 - s), \tag{1}$$

where l is the expected search size of the successful searches, and s denotes the search success rate.

4 Minimizing expected search size

Theorem 4.1. In the optimal replica distribution in terms of expected search size, the number of replicas of an item is proportional to the square root of q_i, and inversely proportional to the square root of v_i.

Since the data items are randomly distributed among the overlay, the average number of probed peers before a copy of item i is found is given by

$$h_i = N/r_i. \tag{2}$$

Then our optimization problem can be defined as

$$\text{Objective}: \ \text{Min} \sum_{i=1}^{m} q_i N/r_i$$

$$\text{s.t.} \sum_{i=1}^{m} r_i v_i = Nc. \tag{3}$$

We construct the Lagrange function,

$$L(r_1, r_2, \ldots, r_m) = \sum_{i=1}^{m} q_i N/r_i + \lambda\left(Nc - \sum_{i=1}^{m} r_i v_i\right), \tag{4}$$

where λ is a constant.

Letting $dL/dr_i = 0$, we obtain

$$r_i = \sqrt{Nq_i/\lambda v_i}, \quad i = 1, 2, \ldots, m. \tag{5}$$

By substituting Eq. (5) in Eq. (3), we obtain

$$\sqrt{N/\lambda}\sum_{j=1}^{m} \sqrt{q_j v_j} = Nc, \tag{6}$$

$$\lambda = \left(\sum_{j=1}^{m} \sqrt{q_j v_j}\right)^2 / Nc^2. \tag{7}$$

Substituting Eq. (7) in Eq. (5), we obtain

$$r_i = Nc\sqrt{q_i/v_i} / \sum_{j=1}^{m} \sqrt{q_j v_j}. \tag{8}$$

Eq. (8) is the optimal distribution under the heterogeneous environment with different item sizes and different query rates. It shows that r_i is inversely proportional to the square root of item size, which agrees well with our intuition. In the following, we denote this optimal replica distribution in terms of search size by SRRR (square root reciprocal replication).

However, Cohen and Shenker argue that

$$r_i = Ncv_i\sqrt{q_i} / \sum_{j=1}^{m} v_j \sqrt{q_j}, \quad i = 1, 2, \ldots, m. \tag{9}$$

More interestingly, when the items have the equivalent size, these two solutions are of the same distribution.

5 Maximizing success rate

Theorem 5.1. In the optimal replica distribution in terms of success rate, the number of replicas of an item r_i is a logarithmic function of v_i/q_i.

Firstly we assume that there is only one item in the system. Here we define n_i as the average number of covered peers. Namely n_i means the average number of peers whose requests are definitely satisfied after i copies are randomly deployed in the overlay. According to the study in [7],

$$n_i = N(1 - T^i), \tag{10}$$

where T stands for $(N - L)/N$ and L denotes TTL.

For the reason that every peer has the same probability to issue a query in each search, the success rate s from i copies is the ratio of covered peers to all the peers:

$$s = n_i/N = 1 - T^i. \tag{11}$$

Since there are r_i copies of item i in the system, the success rate of item i is

$$s_i = 1 - T^{r_i}. \tag{12}$$

Accordingly the success rate gain from item i is

$$s'_i = q_i s_i = q_i(1 - T^{r_i}). \tag{13}$$

Since the success rate of an item is independent of others, the overall success rate S is the gain sum from all the items:

$$S = \sum s'_i = \sum_{i=1}^{m} q_i(1 - T^{r_i}). \tag{14}$$

Then our question can be transformed into the optimization:

$$\text{Objective}: \ \max \sum_{i=1}^{m} q_i(1 - T^{r_i})$$

$$\text{s.t.} \ \sum v_i r_i = cN. \tag{15}$$

We construct the Lagrange function L,

$$L(r_1, r_2, \ldots, r_m) = \sum_{i=1}^{m} q_i(1 - T^{r_i}) + \lambda\left(cN - \sum_{i=1}^{m} v_i r_i\right), \tag{16}$$

where $r_1, r_2, \ldots, r_m$ are variables and λ is a constant.

Let $dL/dr_i = 0$. Then

$$\begin{cases} q_1 T^{r_1} \ln T - \lambda v_1 = 0, \\ q_2 T^{r_2} \ln T - \lambda v_2 = 0, \\ \qquad \vdots \\ q_m T^{r_m} \ln T - \lambda v_m = 0, \end{cases} \tag{17}$$

$$r_i = \log_T^{\lambda} - \log_T^{\ln T} + \log_T^{(v_i/q_i)}, \tag{18}$$

$$\log_T^{\lambda} = r_i + \log_T^{\ln T} - \log_T^{(v_i/q_i)}, \tag{19}$$

where $i = 1, 2, 3, \ldots, m$.

$$v_i \log_T^{\lambda} = v_i(r_i + \log_T^{\ln T} - \log_T^{(q_i/v_i)}). \tag{20}$$

Calculating the sum of both sides of Eq. (20) for $i = 1, 2, \ldots, m$. We have

$$\log_T^{\lambda} \sum_{i=1}^{m} v_i = \sum_{i=1}^{m} (v_i (r_i + \log_T^{\ln T} - \log_T^{(v_i/q_i)})), \tag{21}$$

$$\log_T^{\lambda} = cN / \sum v_i + \log_T^{\ln T} - \sum_{i=1}^{m} (v_i \log_T^{(v_i/q_i)}) / \sum v_i. \tag{22}$$

Substituting Eq. (22) in Eq. (19), we can get the final answer to the optimization:

$$r_i = cN / \sum v_j - \sum_{j=1}^{m} (v_j \log_T^{(v_j/q_j)}) / \sum v_j + \log_T^{(v_i/q_i)}, \tag{23}$$[2)]

where i equals to $1, 2 \ldots, m$.

The solution shows that in the optimal replica distribution in terms of success rate, r_i is a logarithmic function of v_i/q_i. In the following, we denote this optimal replica distribution in terms of success rate by Log (logarithmic replication).

6 Simulations

6.1 Simulation settings

Our simulator is implemented using C++. The Waxman model [14] is used to generate the initial random graph topology. The network has 10k peers with the average degree of 8 and during the network lifetime the average degree is fixed. There are 240 distinct items, and no same replica at any peer. The item size is a random number from 0.1 to 1.9 with the average value of 1.0. Their query rates come from the snapshot of PPStream (Comprehensive Channel, August 4th, 2007)[3)]. Here the query rate of an item is defined as the ratio of its viewers to all the online viewers. In each search, we randomly select a peer to issue queries. In Random Walks, the number of walkers w is set to 1, since the walkers are independent of others without message exchange. The parameters and their default values are listed in Table 1.

In this paper the expected search size and success rate are considered to be our main metrics to evaluate the system performance.

6.2 Simulation results

Figure 1 is the curve of expected search size versus TTL. Both expected search size of $SRRR$ and SRR grow with the increase of TTL. This is because the originally failed walkers continue to look for the requested items after TTL rises while the originally successful searches consume the same messages despite the rise of TTL. Figure 1 shows that SRR is worse than $SRRR$, which means that SRR spends more network bandwidth on searches than $SRRR$. SRR consumes 15.8% messages more than $SRRR$ when TTL equals 100 and c equals 5.

Figure 2 is the curve of expected search size versus storage capability. When the system augments the storage capability and every item has more copies in the system, the walkers can find the requested items and finish the searches in advance. Therefore, as displayed in Figure 2, the expected search size in both SRR and $SRRR$ drops as the storage capability of the system rises.

There is no surprise that $SRRR$ is better than SRR in terms of expected size, as displayed in Figure 1 and Figure 2. $SRRR$ has been proven to be optimal in Section 4 as far as the search size is concerned. It can use the storage capability effectively to decrease the expected search size.

Moreover, intuitively, we do not think SRR is optimal. To obtain the same expected search size, the items with smaller size evidently consume less storage. Therefore, considering the limited storage capabi-

2) It is necessary to note that the copy number r_i of some unpopular items may be negative when cN is relatively small. Then we can set the copy number of these items to be zero, and recalculate the distribution ignoring items with negative copies.

3) http://www.ppstream.com

Table 1 Simulation parameters and default values

Parameter	Default value
Number of peers	10000
Routing algorithm	Random Walks
Random graph	Waxman model
Number of connections	40000
Number of items	240
Storage capability per peer c	5
TTL	50
Item size	randomly from 0.1 to 1.9

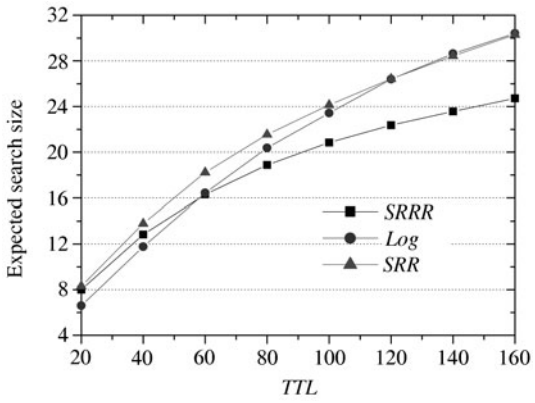

Figure 1 Expected search size comparison of different replica distributions under different TTL, where $c = 5$.

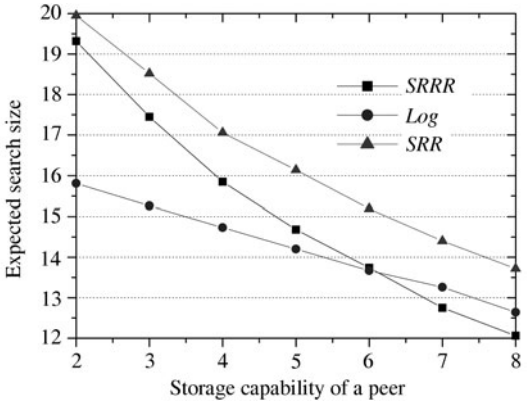

Figure 2 Expected search size comparison of different replica distributions under different storage capability, where TTL=50.

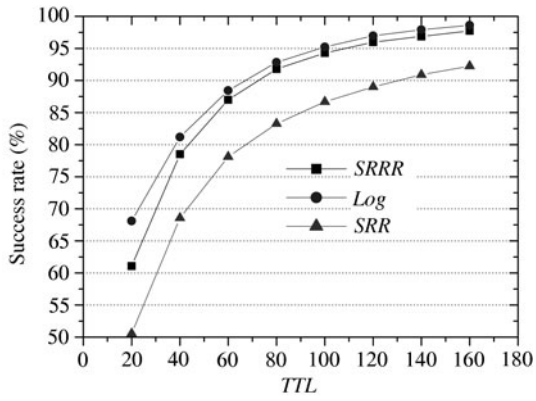

Figure 3 Success rate comparison of different replica distributions under different TTL, where $c = 5$.

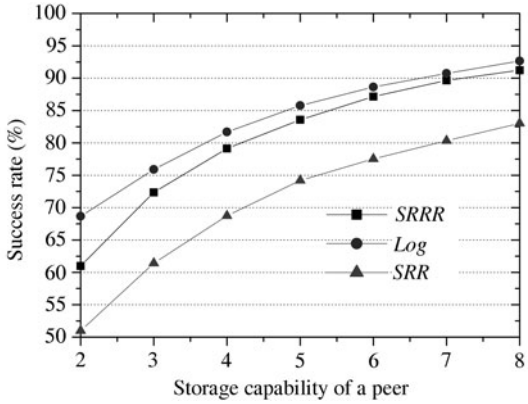

Figure 4 Success rate comparison of different replica distributions under different storage capability, where TTL=50.

lity, we should allocate more storage to the items with small size to decrease the search size effectively. Accordingly, the items with larger size should have fewer copies. This viewpoint agrees well with our solution, in which the number of replicas of an item is inversely proportional to the square root of the item size. Nevertheless SRR in [1] has not provided the related proof at this point.

Figures 3 and 4 are the comparison of success rate with different TTL and storage capability among three solutions. Both figures show that *Log* is better than the other two in terms of success rate. The result is not difficult to understand because *Log* is specially designed for the highest success rate and proven to be optimal in Section 5. On the other hand, since $SRRR$ and SRR are designed to minimize the expected search size, they have the lower performance in success rate. The success rate of *Log* is 10.31% higher than that of SRR when TTL is 60 and c is 5. The success rate of *Log* is 12.95% higher than that of SRR when every peer shares a storage capability of 4 items and TTL equals 50.

6.3 Result analysis and tradeoff

Figures 1 and 2 show us that the expected search size of $SRRR$ is larger than that of Log when TTL is small or the storage capability is tiny. This result contradicts with our previous conclusion that $SRRR$ is optimal. The reason behind the collision lies in the fact that the assumption of Eq. (2) does not hold completely when success rate is relatively low. Whereas when TTL is big or the storage is large, $SRRR$ outperforms Log.

Further, as displayed in Figure 3 and Figure 4, success rate curve of $SRRR$ is close to the Log curve when success rate is relatively high. Whereas when success rate is relatively low, Log achieves a better success rate.

Therefore, we can draw a conclusion supporting the realistic overlay design:

(1) If the system has a rigid requirement on success rate, Log is more suitable.

(2) If the system has a rigid requirement on the expected search size, we should adopt $SRRR$.

Otherwise,

(3) Log is more suitable when the success rate is relatively low.

(4) When success rate is relatively high, we should adopt $SRRR$.

7 Conclusions

This paper tries to resolve a question: given query rates, file size distribution and the fixed storage capability, what is the optimal allocation of the storage to the items? We usually have different answers to this question in terms of different objectives. This paper puts forth two allocations to achieve the lowest search size and the highest success rate respectively, and our simulations verify their validity. Finally, we make a tradeoff between them and draw a practicable conclusion to guide the overlay design: the allocation of Log is more suitable when the success rate is relatively low, while when success rate is relatively high we should adopt $SRRR$.

Simultaneously, our simulation results show that Square-Root allocation, which is traditionally considered to be optimal, is not always the best choice. Our fundamental results offer a new understanding of the resource management under the fully distributed systems. Moreover, although our conclusions are drawn under the background of P2P, they are also applicable to those fully distributed systems, whose resources are managed by means of the unstructured application-level overlays.

Acknowledgements

This work was supported by National Basic Research Program of China (Grant No. 2009CB320705), National Natural Science Foundation of China (Grant Nos. 60803111, 61073028, 61021062), and Jiangsu Natural Science Foundation (Grant Nos. BK2009396, BK2009100). The first author would like to thank Jiangsu Provincial Government and the CLOUD Laboratory for supporting and hosting his visit to University of Melbourne, Australia.

References

1 Cohen E, Shenker S. Replication strategies in unstructured peer-to-peer networks. In: Proceedings of ACM SIGCOMM'02. New York: ACM, 2002. 177–190

2 Leontiadis E, Dimakopoulos V V, Pitoura E. Creating and maintaining replicas in unstructured peer-to-peer systems. In: Proceedings of Euro-Par'06. Berlin: Springer-Verlag, 2006. 1015–1025

3 Saito Y, Shapiro M. Optimistic replication. ACM Comput Surv, 2005, 37: 42–81

4 Thampi S M, Sekaran K C. Survey of search and replication schemes in unstructured P2P networks. Network Protocols Algorithms, 2010, 2: 93–131

5 Lv Q, Cao P, Cohen E, et al. Search and replication in unstructured peer-to-peer networks. In: Proceedings of ICS'02. New York: ACM, 2002. 84–95

6 Goel S, Buyya R. Data replication strategies in wide area distributed systems. In: Qiu R, ed. Enterprise Service Computing: From Concept to Deployment. Hershey, Pennsylvania: Idea Group Inc, 2006. 211–241

7 Feng G F, Jiang Y Q, Chen G H, et al. Replication strategy in unstructured peer-to-peer systems. In: Proceedings of IPDPS'07. Washington DC: IEEE Computer Society, 2007. 1–8
8 Baev I D, Rajaraman R. Approximation algorithms for data placement in arbitrary networks. In: Proceedings of SODA'01. New York: ACM, 2001. 661–670
9 Chekuri C, Kumar A. Maximum coverage problem with group budget constraints and applications. In: Proceedings of APPROX'04. Berlin: Springer, 2004. 72–83
10 Shmoys D B, Swamy C, Levi R. Facility location with service installation costs. In: Proceedings of SODA'04. New York: ACM, 2004. 1088–1097
11 Sozio M, Neumann T, Weikum G. Near-optimal dynamic replication in unstructured peer-to-peer networks. In: Proceedings of SIGMOD/PODS'08. New York: ACM, 2008. 281–290
12 Venugopal S, Buyya R, Ramamohanarao K. A taxonomy of data grids for distributed data sharing, management and processing. ACM Comput Surv, 2006, 38: 1–53
13 Gkantsidis C, Mihail M, Saberi A. Random walks in peer-to-peer networks: Algorithms and evaluation. Perform Evaluat, 2006, 63: 241–263
14 Waxman B M. Routing of multipoint connections. IEEE J Select Areas Commun, 1988, 6: 1617–1622