Replication Strategy in Unstructured Peer-to-Peer Systems

Guofu Feng [1, 2], Yuquan Jiang[1], Guihai Chen[2], Qing Gu[2],
Sanglu Lu[2], and Daoxu Chen[2]

[1] Nanjing Audit University
School of Information Science
Nanjing, 210029, China
{fgf, jyq}@nau.edu.cn

[2]The State Key Lab. for
Novel Software Technology,
Nanjing, 210093, China
{ghchen, guq, sanglu, cdx}@ nju.edu.cn

Abstract

The unstructured Peer-to-Peer (P2P) systems usually use a "blind search" method to find the requested data object by propagating a query to a number of peers randomly. In order to increase the success rate of blind search, replication techniques are widely used in these systems. Most P2P systems replicate the most frequently accessed data objects to improve system performance. However, existing replication strategies cannot answer the question that how many replicas of an object should be kept in the P2P system. If an object is replicated excessively, it inevitably will affect the average efficiency of a replica, which will decrease the whole search performance. This paper addresses the issue of finding the proper number of replicas for an object according to its query rate. In this paper, we firstly investigate the precise relation among success rate, the allocation of replicas and query rate. Then we propose an approach of the allocation of copies to optimize the success rate. As a benchmark, our result offers a new understanding of replication.

1. Introduction

The P2P (Peer-to-Peer) computing model offers a radically different and appealing alternative to the traditional client-server model for many large-scale applications in distributed settings. In this model, end-users share resources in a peer style, potentially acting as both client and server. The P2P approach removes single point failure and associated performance bottlenecks; it also releases the network traffic overhead by providing services locally. However, locating content or service in an efficient and scalable way is still one of the most challenging problems.

P2P systems were classified into three different categories [1]: (1) Centralized P2P systems, where a central directory server is used to manage the metadata (i.e. Napster [2]); (2) Decentralized structured P2P systems, where there is no central server and peers are structured according to some strict rules for file placement and object discovery [3,4,5]; (3) Decentralized unstructured P2P systems, where there is no coupling between topology and data location, such as Gnutella [6], Morpheus [7] and KaZaA [8]. Of all the categories, the decentralized unstructured P2P systems are most commonly used in today's Internet.

In the decentralized unstructured P2P systems, the peers are organized to form an ad hoc network. Each peer connects to a set of neighbors when it joins the network, through which user queries are propagated to retrieve data. As the P2P network topology is unrelated to the location of data, a peer has no idea about which peer has the required object, which may lead to "blind search". The peers receiving a query are unrelated to the target of the query.

Replication technique is often used to improve the performance of blind search. By proactively deploying data replicas on several other peers, the consumed messages will be decreased before hit and the hit rate will be improved when the number of the probe messages is restricted.

Considering replication, the key question is "which, where, when and how many do we distribute the copies?" A lot of previous researches address the problem of "which", "when". However, little work has been done on the issues of "where" and "how many". In the usual approaches, the data objects that are accessed more frequently are distributed with more copies, which helps to increase the hit rate of the popular objects, and thus the whole search performance can be promoted. However,

This work is partially supported by China NSF grant (60573106 and 60573131); the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z199; Jiangsu Provincial NSF grant (BK2005208 and 06KJD520090); the National Grand Fundamental Research 973 Program of China under Grant No.G2002CB312002.
1-4244-0910-1/07/$20.00 ©2007 IEEE

such approaches cannot answer the question of how many replicas of an object should be kept in the P2P system. If an object is replicated excessively, it inevitably will affect the average efficiency of its replicas and the success rate gain from the less popular, which will decrease the whole search performance. So, what is the threshold for the popular objects, and if given the limited storage space what is the appropriate distribution according to the query rate among the entire objects? On the other hand, because the place of a replica has much effect on its value, then what is the impact of the topology on the number of the necessary replicas?

The fundamental question we address in this paper is: given the query rate and the limited storage space, how to find out the optimal number of copies for each data object.

The rest of the paper is organized as follows. In Section 2, we investigate the impact of topology on the success rate and present an allocation where the factor of topology is taken into account from a centralized view to achieve the optimal success rate. In Section 3, we present some comparisons with some widely used replications, such as proportional replication, uniform replication and square root replication, to verify the validity of our proposed strategy. In Section 4 we make some feasible assumptions and propose a more practical strategy for the allocation in the fully distributed P2P. The replication strategy is also compared with the optimal result in Section 3. Finally, this paper concludes with Section 5.

2. The Optimal Allocation from a Centralized View

2.1. The Unstructured Peer-to-Peer Model

Considering a simple model of unstructured P2P system, the overlay network consists of N peers, each with capacity ρ, which is the set of replicas that a peer holds (where we assume there is no more than one copy per data object). Assume there are m distinct data objects in the system. The query rate vector q takes the form $q_1 \geq q_2 \geq q_3 \geq \ldots \geq q_m$ 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 object.

Just as described in [9], we define ***replication strategy*** as the means specifying the number of copies for each object if given the distribution of access frequency. Replication strategy sometimes is called ***allocation***. Here the main metric we consider is the success rate.

Here let r_i denote the number of copies of the ith object. Just as described in [9], a replication strategy is an *allocation* of storage rooms to the data objects according to their access frequency if given the restriction of storage space and the query rate of each object. A replication *strategy* or allocation is a mapping from the query rate q to the copy number r. The allocation is presented by the vector $r = (r_1, r_2, r_3 \ldots r_m)$.

We assume that the peers randomly issue queries, i.e., they issue queries with the same probability in each request. In the following, we will introduce how to find an allocation to optimize the search success rate when Random Walks routing mechanism is used in the P2P network.

2.2. The Topology and the Potential Energy

The unstructured and decentralized P2P systems firstly arrange the peers as a logical overlay network, in which every peer is only aware of its neighboring peers. The query messages are transmitted peer by peer to retrieve the data. Therefore, a peer's contribution to success rate lies on its capability to satisfy the queries, i.e. the capability of receiving messages and the query rate of the data objects that the peer stores.

Suppose that the overlay network is a connected and undirected graph. If a replica of object i is placed on peer u, the value of the replica lies on the times that the object i is requested and query messages are routed to peer u at the same time.

In the *TTL*-based searches, the probability that a peer receives messages is impacted by the forwarding pattern and the topology of overlay network. In flooding, only the queries issued by the peers within the range of peer u, i.e., the peers whose distance to peer u is less than *TTL*, can reach peer u. Furthermore, all the queries within the *range* of peer u inevitably can reach peer u. Therefore, in flooding, the contribution of a replica on peer u is closely related to the number of peers within the range of peer u. But in Random Walks, that a peer is within the range of peer u doesn't mean that its queries determinately reach peer u for the reason of underlying topology and forwarding pattern. For example, when the peer 4 in Figure 1 receives query, the probability that peer 2, peer 3 and peer 6 receive messages from peer 4 in Random Walks entirely is 1/3, i.e. the reciprocal of the degree of the peer 4. If every peer issues a query with *TTL*=1, the average messages received by peer 1, 2, 4 are 5/6, 5/6 and 4/3 respectively. (Here we ignore the fact that a peer will not retransmit a query back, which will lead to slight distortion to the real P2P systems. It is more acceptable when the average degree is large.)

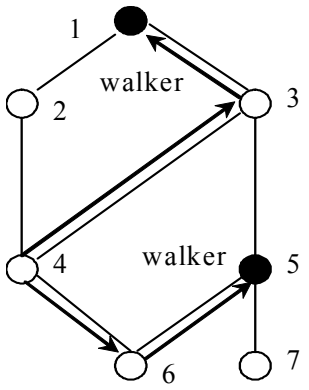

Figure 1. The simple model of overlay network and the principle of Random Walks

We define ***Potential Energy E*** as a peer's capability of receiving query messages. Then a peer's Potential Energy is closely related to the Potential Energy of its neighbors and the probability that it can receive messages from neighbors. Therefore, for Random Walks, ***E*** can be recursively defined as:

$$E_u = \sum_{v \in D} (E_v / d_v) \quad (1)$$

where ***D*** is the neighbor set of peer ***u***, d_v is the degree of peer ***v***, and E_v/d_v is the probability that peer ***u*** can receive messages from peer ***v*** through the link<*v, u*>.

Define $E = (E_i)$, where E_i is the Potential Energy of peer ***i***. Define matrix $M = (m_{ij})$. If there is a link between peer ***i*** and peer ***j***, m_{ij} and m_{ji} are $1/d_i$ and $1/d_j$ respectively, and others 0. Then according to 1),

$$E' = E \times M \quad (2)$$

where ***E'*** is an iteration of ***E***. A peer will propagate its Potential Energy to its neighbors during iteration and the Potential Energy will converge to some stable state, in which the Potential Energy in every peer roughly represents the relative capability of receiving messages.

We assign ***E*** the initial value ***1/n*** since we assume that the peers have the same probability to generate requests. Thus we can obtain the Potential Energy for each peer by the iterative computation of (2). Since the iteration stands for the transition of the Potential Energy, moreover, only the requests within a peer's range can reach it, the iteration times should be *TTL*, or less than *TTL* for the less precise results. And the final Potential Energy approximately represents the ratio of the number of messages that a peer receives to all the messages in the whole system during a period of time.

Potential Energy implies a peer's capability of receiving messages. If a replica is deployed on a peer with higher Potential Energy, it will satisfy more requests inevitably.

The Potential Energy as a transitional parameter assists us to pursue the solution in the following and has little effect on our final conclusion, so here we do not investigate it more.

2.3. The Potential Energy and the Optimal Replication Strategy

Let's firstly only consider the replicas of one object and assume that ***S*** requests have been randomly issued. If we ignore the lost messages when walkers hit the targets and terminate, all the messages generated during the process will be,

$$M = S * TTL * k \quad (3)$$

Let Δ denote ***TTL*k*** in the following. According to the definition of Potential Energy, the number of messages that peer ***u*** have received will be,

$$M_u = E_u * M = E_u * S * \Delta \quad (4)$$

That is, if a replica is placed on peer ***u***, the peer ***u*** will satisfy the requests for M_u times. Therefore, the search success rate will be

$$SR_u = M_u / S = E_u * \Delta \quad (5)$$

Because of the assumption that all the peers issue requests with the same probability, among the ***S*** requests above, the number of requests issued by each peer approximately is

$$R = S / N \quad (6)$$

A peer ***u*** receives M_u messages and the messages usually come from a subset peers within its *TTL* range, so from an overall point of view a replica on peer ***u*** can guarantee a relatively fixed number of peers, whose requests can be satisfied. The fixed number will be,

$$n_u = M_u / R = E_u * N * \Delta \quad (7)$$

Because of the same requesting probability, the search success rate can also be expressed as the ratio of the guaranteed peers to all the peers from a holistic point,

$$SR_u = n_u / N \quad (8)$$

However, if we have already placed a same replica in the system, we will get the less success rate than that given by Expression (8). This is because that the walkers will terminate when queries reach the first replica and be satisfied, and correspondingly the second peer receives fewer queries than that given by the Expression (4).

As far as the success rate concerned, when we deploy another same replica in the system, it is less effective in improving the success rate since some peers' requests have been guaranteed. In other words, the new deployed replicas are effective only for the peers that still haven't been guaranteed in terms of the search success rate.

Suppose that some replicas have been deployed in the system, and ***n*** peers have been guaranteed. Then if we also place the same replica in another peer ***v***, the number of peers whose requests can be guaranteed for the first

time (Here, the peers still haven't been guaranteed before the insertion of the new replica) is,

$$n_v = (N - n) / N * E_v * \Delta \quad (9)$$

According to (8), the increased gain in search success rate is

$$\delta_v = n_v / N = (N - n) / N^2 * E_v * \Delta \quad (10)$$

Only one data object is considered in the analysis above. Given the vector of *query rate* of data objects, if a data is seldom accessed, the contribution of its copies to success rate is little, even though the replicas can satisfy all the requests of all the peers. Therefore, the relative gain of a new replica of object ***i*** on peer ***v*** will be

$$\delta_{vi} = q_i * \delta_v = q_i * (N - n_i) / N^2 * E_v * \Delta \quad (11)$$

where n_i is the number of peers, whose requests for object ***i*** have been satisfied.

Therefore, when a unit of storage space on peer ***v*** is available, the decision of which copy is selected is mainly determined by the gain of δ_{vi}. We assume that the peers have been sorted in terms of their Potential Energy. From the largest to the least, we allocated all the storage rooms step by step. In each step, the copy, whose δ_{vi} is largest, will be firstly deployed and it can be guaranteed that the success rate gain at every step is largest. Correspondingly the number of copies of each object, i.e. the allocation, is finally determined after the deployment. And the final optimal success rate is expected to be the sum of the gain in every deployment,

$$SR = \sum_{i=1}^{N|\rho|} \delta_i \quad (12)$$

where δ_i represents the gain determined by Expression (11) in step ***i***.

Theorem 1: ***It can be proven that the final vector r is optimal in terms of success rate.***

Proof: *If vector **r** isn't the optimal allocation, in other words, the copy sequence in the deployment above isn't optimal, we must be able to find another sequence, in which the sum of success rate gain at every step is higher.*

*We assume that the **m**th copy of objects **i** and the **n**th copy of object **j** are respectively deployed on peer **u** and peer **v** at step **a** and step **b**. According to (11), at step **a**,*

$$q_i * (N - n_i^m) \geq q_k * (N - n_k), \ k \neq i \quad (13)$$

*If the deployment at step **a** isn't optimal, there must be another copy, such as the **n**th copy of object **j**, should be deployed at peer **u**. We also assume that the **m**th copy of object **i** is deployed on the peer **v** in the second sequence. If we select one from all the copies deployed after step **a** in the first sequence for peer **v** at step **b**, for the sake of Expression (13) and (11), the success rate gain of the **m**th copy of object **i** is the highest.*

Let's examine the sum of the success rate gain in both sequences.

$$\begin{aligned} &(E_u * q_i * (N - n_i) + E_v * q_j * (N - n_j)) \\ &-(E_v * q_i * (N - n_i) + E_u * q_j * (N - n_j)) \\ &= (E_u - E_v)(q_i * (N - n_i) - q_j * (N - n_j)) \\ &\geq 0 \end{aligned} \quad (14)$$

Therefore, the sum of gain in the second sequence is less than or equal to that in our first sequence.

*Therefore, the **m**th copy of object **i** must be deployed on peer **u** at step **a** in order for the maximum success rate gain. Similarly, it can be proven that our deployment sequence is optimal and the final allocation **r** is optimal.*

3. Simulations

3.1. Simulation Model

In this section, we evaluate the performance of the proposed method. The BA model [10] and the Waxman model [11] are respectively used to generate power-law and random graph topology. Each network has 10,000 peers with the average degree of 3.6. Figure 2 illustrates the distribution of the number of the peers versus the degree. There are 200 distinct objects, and no same replicas at any peer.

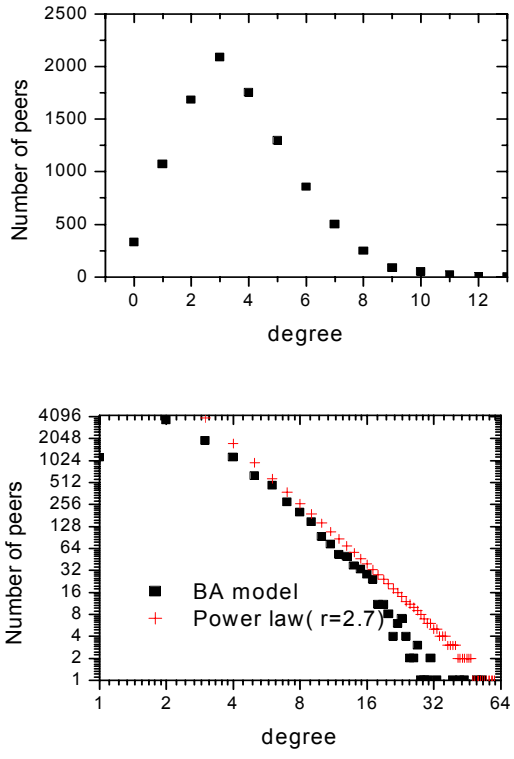

Figure 2.The distribution of degree of Waxman topology and BA topology

In each search, we randomly select a peer to issue queries, and compute the success rate of 60,000 requests. In Random Walks scenario, ***k*** is set 3 and *TTL* is set 6, which are close to the parameters in Gnutella. The search success rate is considered as the main metric to evaluate the system performance. The parameters and their default values are listed in table 1.

Table 1. The simulation parameters and its default values

Parameters	Default values
Number of Pees	10,000
Routing Mode	Random walks
Query Rate	Zipf (α =0.92)
Topology of Power law	BA model
Random Graph	Waxman model
Number of Links	18000
Number of Objects	200
Number of Walkers	3
Capability per Peer	8
TTL	6

3.2. Simulations and Analysis

Lv. et al in [9] examined the uniform replication strategy and proportional replication strategy, and finally proposed an optimal replication strategy of square-root replication to minimize the average path length. In this section we will make some comparisons with these widely used replication strategies under the topology of Power-law.

Figure 3 illustrates the tendency comparison of success rate versus the number of replicas when different replication strategies are used. It shows that the success rate with the allocation proposed in this paper is the highest at any average number of copies among all the allocations. The uniform allocation is better with more copies, while the proportional allocation is better with fewer average copies. As a compromise, the square root allocation proposed in [9], works well as displayed in Figure 3. The proportional allocation is usually used in the most replications, but it is far from optimal in terms of success rate, usually about 10% lower than the Potential Energy based allocation. Other experiments not illustrated here show that the proposed allocation in this paper works more predominantly when there are more data objects and fewer storage rooms.

Figure 4 represents the distribution of the number of copies per object. If we consider the Potential Energy based allocation optimal, the allocation of square root is relatively better, whose curve is closer to the optimal. The proportional allocation has a close relation to the query rate, so its effectiveness lies much on the access mode. We also can find that most replication strategies usually allocate excessive copies for more popular objects and too few for the less.

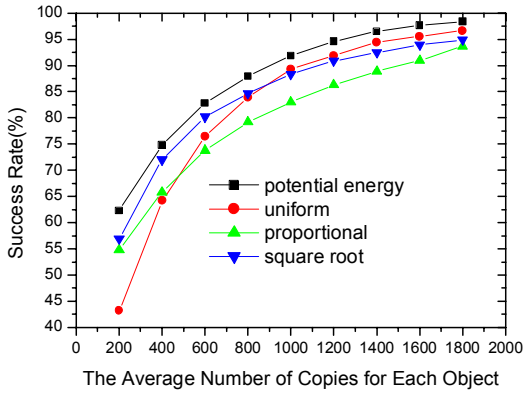

Figure 3. The success rate with different allocation strategies

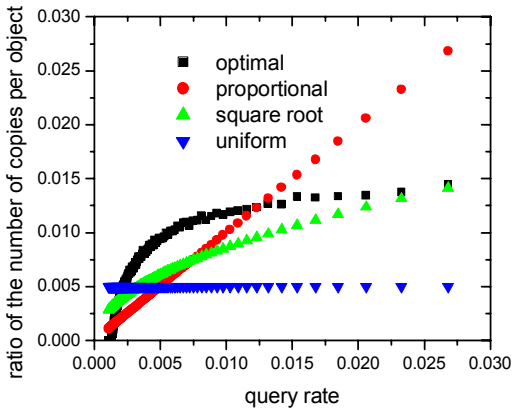

Figure 4. The distribution of the number of copies per object

4. The Practical Variation of the Optimal Allocation for Unstructured P2P

4.1. Optimal Allocation for Unstructured P2P

As an optimal allocation, the discussion in Section 3 provides an approach and benchmark to the replication strategies, but it has more value in academic than in practice, because the allocation is in a centralized manner, which is not appropriate for the fully decentralized P2P systems, i.e., it is difficult to get the Potential Energy of every peer, and it is infeasible to deploy the replicas for each storage unit one by one. In this section, we are striving for a more practical allocation based on the conclusion in Section 3.

Here we randomly select peers to deploy replicas on since it is difficult to calculate its Potential Energy. Then

we assume that the peers in overlay network have the same average Potential Energy. The assumption is reasonable especially when several copies per object are deployed, for some are deployed on the peers with higher Potential Energy and some on with lower. Therefore, all the peers have the equivalent average capability of receiving messages.

The number of peers has a close relation to the number of copies that have already been deployed. The first copy of a data can effectively cover Δ peers if we ignore the loss because of the repeated messages passing the same peers. The second copy covers fewer peers than the first one because of the peers having been guaranteed, and so forth.

Let n_i denote the number of peers. We assume that some peers' requests can be guaranteed after we place the ith copy of a data item on a randomly selected peer. Δ_i represents the increased number of peers having been guaranteed for the first time after the deployment of the ith copy. δ_i stands for the increased success rate due to the insertion of the ith copy. Therefore,

$$\begin{cases} n_1 = \Delta \\ n_2 = n_1 + \Delta - \Delta * n_1 / N \\ n_3 = n_2 + \Delta - \Delta * n_2 / N \\ \dots \\ n_i = n_{i-1} + \Delta - \Delta * n_{i-1} / N \end{cases} \quad (15)$$

By subtracting the adjacent expressions with the repeated iterations, we can obtain,

$$n_i = \Delta \sum_{k=0}^{i-1} (\frac{N-\Delta}{N})^k = N(1-(\frac{N-\Delta}{N})^i) \quad (16)$$

$$\Delta_i = \Delta * (\frac{N-\Delta}{N})^{i-1} \quad (17)$$

Expression (17) shows that the effectiveness is gradually decreasing with the increase of the copies of the objects, which is in accord with our intuition.

Due to the imbalance of query rate, the contributions of two replicas of two different objects will be different, even though the replicas are inserted with the same sequence number. To take the query rate into consideration, the gain of the ith copy of object j will be,

$$\Delta_{ji} = q_j * \Delta * (\frac{N-\Delta}{N})^{i-1} \quad (18)$$

$$\delta_{ji} = \Delta_{ji} / N = q_j * \Delta * (\frac{N-\Delta}{N})^{i-1} / N \quad (19)$$

Then the tendencies of the gain increment of all objects with the increasing of copies are a cluster of curves, displayed in Figure 5. If we draw a horizontal line in Figure 5, the intersections of the line and the curves mean that it will deploy different copies for each object if we obtain the same gain increment.

Return to the essential problem of allocation. Suppose that we have already deployed some replicas of objects in the system. If a unit of storage is available, we are always based on the current state to select a copy to make the gain increment of success rate highest according to the Expression (19). Therefore, after the deployment for all the storage space, the final gain increment of every object is approximately equal with the final allocation $r = (r_1, r_2, r_3 \dots r_m)$, because for a data object even with high access frequency, its increment gradually decreases with the increase of the number of its copies, as displayed by Expression (19). And correspondingly, Δ_i of every object i is approximately equal after the final deployment, that is, Δ_{ir_i} for every object i is approximately equal according to (19), where

$$\Delta_{ir_i} = q_i * \Delta * (\frac{N-\Delta}{N})^{r_i - 1} \quad i = 1, 2, 3 \dots m \quad (20)$$

For any two object i and j,

$$q_i * \Delta * (\frac{N-\Delta}{N})^{r_i - 1} = q_j * \Delta * (\frac{N-\Delta}{N})^{r_j - 1} \quad (21)$$

$$(\frac{N-\Delta}{N})^{r_i - r_j} = q_j / q_i \quad (22)$$

Taking the restriction of storage space into consideration,

$$\begin{cases} r_i = \log_{\frac{N-\Delta}{N}}^{q_j} - \log_{\frac{N-\Delta}{N}}^{q_i} + r_j \\ \sum_{i=1}^{m} r_i = N |\rho| \end{cases} \quad (23)$$

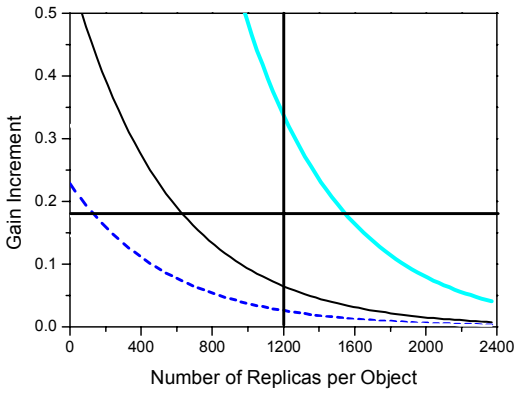

Figure 5. The tendencies of the gain increment of objects with different access frequency versus the increasing of copies, given by the Expression (19) in an overlay network with 10,000 peers

We can get the final solution to the problem of allocation,

$$\begin{cases} r_1 = (\sum_{i=1}^{m} \log_{\frac{N-\Delta}{N}}^{q_i} - m \log_{\frac{N-\Delta}{N}}^{q_1} + N|\rho|) / m \\ r_i = r_1 + \log_{\frac{N-\Delta}{N}}^{q_1} - \log_{\frac{N-\Delta}{N}}^{q_i} \quad , \quad r_i \geq 0 \end{cases} \tag{24}$$

Here we have no intention for further work on how to deploy the replicas in the real systems after the optimal allocation is given, since the problem has been discussed in [9].

4.2. Simulations

In this section, we will exhibit some simulation results to verify our ratiocinations.

Figure 6 illustrates the comparison of our theoretical results given by Expression (19) with the actual experimental result. In the simulation, a room per peer and only a data object are shared. 20 copies are deployed on the randomly selected peers in each step until 1,800 replicas in the whole system, and the success rate is calculated every 60,000 queries. As Figure 6 shows, the result of simulation is consistent with Expression (19): the relation between the success rate and the number of copies is credible. This is our base for the following work. At the same time, the result indicates that our assumption, the peers in overlay network have the equivalent Potential Energy if the peer to deploy replica on is randomly selected, is reasonable

Because BA topology and Waxman topology have equivalent peers and links and the peers are randomly selected, the sums of Potential Energy obtained in each step are approximately equal, and so the sum of δ_v is, where v represents the peers deployed in a step. Therefore, the trends of success rate versus the number of copies under both topologies are approximately equal.

Figure 7 illustrates the comparison of theoretic allocation given by Expression (24) with the experimental optimal one proposed in Section 3. It shows that a high consistency between theoretic result and the experimental results under both topologies, which implies that our allocation with practical variation is consistent to the optimal strategy in Section 3. Namely, our conclusion is also tenable in the fully distributed P2P systems. Furthermore, the allocation under BA topology fluctuates more visibly compared with the result under Waxman topology. This is because that the gaps of Potential Energy among peers in Power-law topology are more notable than those in random graph.

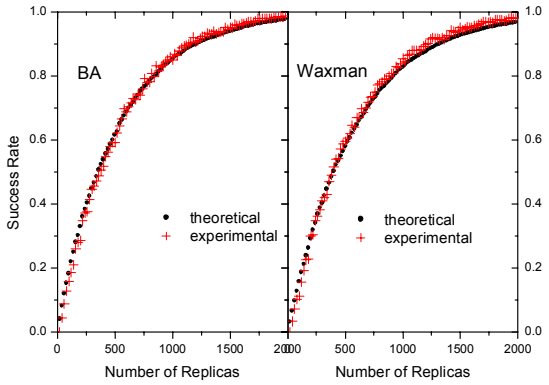

Figure 6. The success rate as a function of the number of replicas

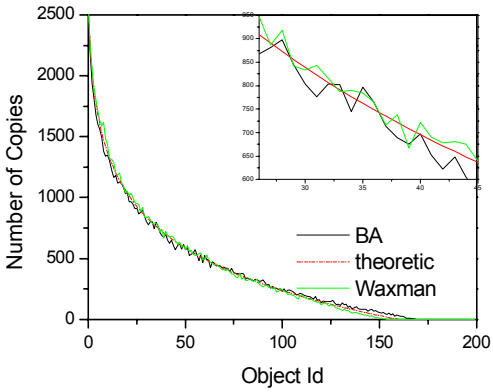

Figure 7. The comparison of allocation in the practical variation with the theoretically optimal result

5. Conclusions

This paper tries to resolve a question: given the fixed storage room and query rate, what is the optimal allocation in an unstructured P2P network? The relation between success rate and query rate is investigated and a centralized algorithm for the optimal success rate is proposed to allocate the replicas, which can be considered as a benchmark to evaluate the replication strategies. Finally a more practical variation is proposed. The simulation results verified the reasonability of our theory. Our fundamental results offer a new understanding of replication and show that the usual replication strategies are far from optimal in consideration of search success rate.

References

[1] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks", In: *Proc. of the 16th ACM Int'l Conf. on Supercomputing (ICS 2002)*. New York: ACM Press, 2002. 84-95.

[2] Napster, http://www.napster.com/, 2001.

[3] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard. Karp and Scott Shenker, "A Scalable Content-Addressable. Network", *In Proceedings of the ACM SIGCOMM 2001*,, San Diego, CA, USA, August 2001. pp.161~172.

[4] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications", In: *SIGCOMM*, San Diego, CA, USA, Aug. 2001.

[5] B. Y. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", *Technical Report UCB/CSD-01-1141*, University of California at Berkeley, Computer Science Department, 2001.

[6] S. Daswani and A. Fisk, "Gnutella UDP Extension for Scalable Searches (GUESS) v0.1 [EB/OL]", http://groups.yahoo.com/group/the_gdf/files/Proposals/GUESS/guess_01.html, 2002-08-01

[7] Morpheus. http://www.musiccity.com/, 2002.

[8] KaZaA. http://www.kazaa.com/, 2002.

[9] E.Cohen and S.Shenker, "Replication strategies in unstructured peer-to-peer networks", *In Proc. of ACM SIGCOMM'02*, San Diego, CA, USA, Aug. 2002

[10] R. Albert, H. Jeong, and A. L. Barabasi, "Error and attack tolerance of complex networks", *Nature*, 406, 378, 2000.

[11] B. Waxman, "Routing of Multipoint Connections", *IEEE J. Select. Areas Connun,* SAC-6 (9): 1617-1622, dec. 1988