Modeling Replication Strategies in Data Grid Systems with Arbitrary Clustered Demands

Jianjin Jiang, Guangwen Yang, Dingxing Wang

Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China

jiangjj02@mails.tsinghua.edu.cn, ygw@tsinghua.edu.cn, dxwang@tsinghua.edu.cn

ABSTRACT

This paper considers the relationship between request distribution and replica distribution in data grid when request exhibits arbitrary clustered demands. We first give formal model of replication strategies in data grid system. Second, we investigate what is optimal way at the objective of minimizing average access latency to replicate data when request exhibits arbitrary clustered demands. We explain why replicas should be replicated uniformly when request is uniformly distributed in a sub grid in the sense of optimal strategy. Then we investigate the relationship between different files in a sub grid. Furthermore, we analyze the case when all sub grids are equal-sized and conclude that when request is uniformly distributed in system, replicas should be uniformly distributed in system too. Finally, we give an optimal strategy when sub grids are not equal-sized and different sub grids exhibit different request clustering patterns. Compared with some popular strategies, the optimal strategy has some advantages of lower wide area network bandwidth requirement and lower average access latency. Simulation results validate the effectiveness of optimal strategy.

Keywords

Data grid, replication strategy, clustered demands, access latency, optimized distribution.

1. INTRODUCTION

Data usage behavior is a key element in data grid systems [4, 6]. According to some traces on data grid systems [8, 9] and peer to peer file sharing systems [7], request workloads exhibit clustering properties or more complex demand patterns. Replication has been proven as a useful technique in distributed system which can decrease access latency and network bandwidth requirement. The purpose of replication strategy is to decide replica distribution for a given request demand distribution [5]. In previous works about replication strategies, there is an assumption that request is uniformly distributed in system and replica is uniformly distributed in system consequently [5, 16]. The motivation of Ref. [15] is to investigate the optimal search performance in

unstructured peer to peer systems when request exhibits clustered demands. In this paper we consider download performance of data grid systems in which request can exhibit arbitrary clustered demand patterns. Furthermore, we assume that replica can be distributed in system with arbitrary distribution. We first give the model of replication strategies in data grid systems in which request can exhibit arbitrary clustered demands and arbitrary distributions.

Secondly, we investigate the relationship between request demand and replica distribution. For a request distribution, there is an optimal way to replicate data with some objectives. This paper considers the optimization problem of minimizing average access latency. In the sense of minimization of average access latency, we investigate the replica distribution when request is uniformly distributed in a sub grid. In previous works, when request is assumed to distribute uniformly in system, then a counterpart assumption is that replica is uniformly distributed in system [5, 15, 16]. In this paper we will investigate whether the latter assumption is rational in the sense of minimizing average access latency. We find that uniformly distribute replica in a sub grid will produce optimal result. Another result is that the more popular a file in a node, the more probable this node should have a replica of this file. Then we investigate the relationship between different files in the same sub grid. We consider the case when the request for a file is uniformly distributed in a sub grid. We find that the more popular a file is, the more replicas should be hold in this sub grid. Furthermore, we analyze a specific case, in which all sub grids are equal-sized. We find that when requests are uniformly distributed in all nodes of system, then replicas should be uniformly distributed in the system. However, for the case when sub grids are not equal-sized, we can not infer the uniform request imply uniform replica distribution in the sense of minimizing average access latency.

Based on the analytic results above, we investigate what is the optimal way to replicate data when requests exhibit arbitrary clustered demands, i.e. the popularity for a given file across every sub grid is not uniformly distributed. However, in a sub grid, we assume requests are uniformly distributed. In our model, sub grid size can be not equal-sized. We solve the problem by Lagrange Multiplier method and Bisection Method. Finally, we give some examples to demonstrate the performance of optimal strategy compared with other strategies. Since caching strategy is popularly used in data grid systems, we consider a classical caching strategy, LRU. We simulate optimal strategy and other strategies to compare their performance. Simulation results show that the optimal strategy can decrease average access latency and wide area network bandwidth requirement. Simulation results

verify the analytic results that the optimal strategy proposed in this paper can minimize average access latency.

The remainder of this paper is organized as follows. In next section we introduce our model and present a precise statement of the problem. In Section 3 we investigate the relationship between request distribution and replica distribution in the sense of optimized download performance. Section 4 gives the solution of minimizing average access latency when request is uniformly distributed in a sub grid but non-uniformly across sub grids. In Section 5, we give simulation results to validate our analytic results and compare the performance of different replica management strategies, including LRU. Some related works are given in Section 6. Finally, Section 7 concludes and gives future research work.

2. MODEL & PROBLEM FORMULATION

2.1 Terms and Definitions

In [10] we gave our federated data grid model. For the completeness of this paper, we sketch it briefly now. There are G sub data grids in system, every of which is a logically independent system and any two sub grids peer to each other. A sub data grid is consisted of several nodes of which one is super node and the others are normal nodes. A normal node has finite local storage space to store data replica. A super node is in charge of storing index information of those nodes that belong to this sub grid. A super node answers the request messages from normal nodes that belong to the same sub grid with it and the request messages from other sub grids. All super nodes in system cooperate to maintain the topology of system, to response the request messages, and to maintain the coherence of different replicas of data. Every node can serve the file request if it holds a replica of the requested file.

There are M nodes in system, and a node n_k belongs to a sub grid only, with $k = 1, 2, …, M$. For a sub data grid g_i, $i = 1, 2, …, G$, its size is m_i, with $m_i \geq 1$ and $\sum m_i = M$. We call the vector $m = (m_1, m_2, …, m_G)$ as grid size profile (GSP). Without loss of generality, we assume that the nodes belong to sub grid g_i are $n_{M_{i-1}+1}, n_{M_{i-1}+2}, …, n_{M_i}$, where $M_i = \sum_{h=1}^{i} m_h$, $i = 1, 2, …, G$ and $M_0 = 0$. The average sub grid size is $\overline{m} = \frac{1}{G}\sum_{i=1}^{G} m_i = M/G$. On the other hand, a node n_k belongs to sub grid g_i, $i = g(k)$, where $g(k)$ is a mapping function from the index of node n_k to the index of sub grid g_i.

There are N unique files f_j in data grid, and $j = 1, 2, …, N$. Just like some earlier works on replication strategy in peer to peer network [5, 15, 16], we assume different file has same unit size to avoid adding 0/1-knapsack-type complexities to a problem that is already combinatorial [12]. In fact, our subsequent analysis can be extended by treating the j-th file as a group of c_j files with same query rate, where c_j is the size of file f_j. Furthermore, we assume that the storage space of every node has same size and it can store K file replicas, so the system can store MK file replicas. For simplicity of presentation we will keep using homogeneous capacity setting in the sequel.

In this paper, we want to model arbitrary clustered demands and arbitrary replica distributions. More concretely, the request for a given file may be different for two nodes in system; on the other hand, the probability of a node storing a given file's replica may be different from another node. For file f_j, the normalized request rate at n_k is λ_{kj}, which is the fraction of all requests that are issued for the j-th file. In the whole system, file f_j is associated with a normalized request rate λ_j per node, that is, $M\lambda_j = \sum_{k=1}^{M} \lambda_{kj}$. The normalized cumulative request rate per node for all files is $\lambda = \sum \lambda_j$ =1.

For file f_j, there are r_j replicas distributed in the system, and we assume $r_j \geq 1$. For node n_k, there is at most one replica of f_j in its storage space. The probability of file f_j has a replica in node n_k is ρ_{kj}.

2.2 Metrics

2.2.1 Preliminary

We ignore the issue of search costs, concentrating exclusively on downloading aspects of data grid system. We suppose that if there is a replica in data grid, then it can be found definitely. When a node wants to access a file of which it does not holds a replica, it forwards this request to its super node. If the requested file is hit in the local sub grid, the origin requesting node fetches a copy from a node hosting a replica of this file. Otherwise, by means of cooperation of super nodes of all sub grids, a node storing a replica of requested file can be located. Then the origin node fetches a copy from a node hosting a replica of this file. So a file request may be served by local site, local sub grid, or any other sub grid. According to the algorithm for node n_k requesting file f_j, we consider three events and compute their probabilities respectively:

1. Event El_{kj}, which means there is a replica of file f_j in n_k, and we have

$$P(El_{kj}) = \rho_{kj} \tag{1}$$

2. Event Eg_{kj}, which means there is no replica of file f_j in n_k but file f_j hits in the other nodes of sub grid g_i, where $i = g(k)$.

3. Event EG_{-kj}, which means there is no replica of file f_j in sub grid g_i, where $i = g(k)$, and file f_j must be hit in other sub grids, so we have

$$P(EG_{-kj}) = \prod_{i=M_{g(k)-1}+1}^{M_{g(k)}} (1 - \rho_{ij}) \tag{2}$$

Since there is at least one replica of f_j in data grid, we have:

$$P(El_{kj}) + P(Eg_{kj}) + P(EG_{-kj}) = 1.$$

So:

$$P(Eg_{kj}) = 1 - \rho_{kj} - \prod_{i=M_{g(k)-1}+1}^{M_{g(k)}} (1 - \rho_{ij}) \tag{3}$$

2.2.2 Cumulative Average Hit Ratios

Having the probabilities above, we can compute the cumulative average hit ratios now. The cumulative average hit ratio of local site is $P(local)$, which means the probability of a file request can be served by local site for all file requests from any node in system. We have:

$$P(local) = \sum_{k=1}^{M}\sum_{j=1}^{N}\frac{\lambda_{kj}}{M\lambda}\rho_{kj} = \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{N}\lambda_{kj}\rho_{kj} \qquad (4)$$

The cumulative average hit ratio of local sub grid is $P(intra)$, which means the probability of a file request can be served by a site inside the same sub grid with the requesting node for all file requests from any node in system. We can know:

$$P(intra) = \sum_{k=1}^{M}\sum_{j=1}^{N}\frac{\lambda_{kj}}{M\lambda}\left[1-\rho_{kj}-\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})\right]$$

$$= \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{N}\lambda_{kj}\left[1-\rho_{kj}-\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})\right] \quad (5)$$

The cumulative average hit ratio of other sub grids is $P(inter)$, which means the probability of a file request should be served by a site located outside the sub grid the requesting node belong to for all file requests from any node in system. We have:

$$P(inter) = \sum_{k=1}^{M}\sum_{j=1}^{N}\frac{\lambda_{kj}}{M\lambda}\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})$$

$$= \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{N}\lambda_{kj}\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij}) \qquad (6)$$

If the nodes in a sub grid are connected by local area network and sub grids are connected by wide area network, then the cumulative average hit ratios can be interpreted as the ratios of different network bandwidth requirement for a file request.

2.2.3 Average Access Latency
We extend the original access model defined in [13] and [11]. In our model, accessing a file replica from a node's local storage space costs t_l, from a remote node of the same sub grid that this node belongs to costs t_g, and from a node of other sub grids costs t_G, with $t_l \le t_g \le t_G$.

What differs from the model defined in [13] and [11] is that t_G is not only the cost of accessing a file from its origin server, but also from any node in other sub grids which holds a replica of the requested file. In this paper we utilize access latency as the access cost and we assume the costs that a node accesses all the nodes in other sub grids are same.

We denote the average access latency of node n_k requesting file f_j as $t(n_k, f_j)$. Applying Eq. (1), Eq. (2) and Eq. (3), we can obtain the average access latency for a file request. The average access latency of node n_k requesting file f_j is:

$$t(n_k, f_j) = P(El_{kj})\,t_l + P(Eg_{kj})\,t_g + P(EG_{-kj})\,t_G$$

$$= \rho_{kj}t_l + [1-\rho_{kj}-\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})]t_g + \prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})t_G$$

For simplicity in the notation, let $t_g - t_l = t_{gl}$ and $t_G - t_g = t_{Gg}$, we have:

$$t(n_k, f_j) = t_g - t_{gl}\rho_{kj} + t_{Gg}\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})$$

Considering the request rate of every file and the requests originated from different nodes, we can compute the average access latency for a node requesting any file in data grid:

$$t = \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{N}\frac{\lambda_{kj}}{\lambda}[t(n_k, f_j)]$$

$$= \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{N}\lambda_{kj}\left[t_g - t_{gl}\rho_{kj} + t_{Gg}\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})\right] \quad (7)$$

For all replication strategies, the constraints below should be satisfied:

1) The number of all replicas in system is no more than the total storage space of data grid, i.e.

$$\sum_{j=1}^{N}\sum_{k=1}^{M}\rho_{kj} \le KM$$

Since storing extra replicas in blank space will not deteriorate system performance, we will have:

$$\sum_{j=1}^{N}\sum_{k=1}^{M}\rho_{kj} = KM \qquad (8a)$$

2) For every node n_k, the replicas can not exceed its storage space

$$\sum_{j=1}^{N}\rho_{kj} \le K \qquad (8b)$$

3) Since ρ_{kj} is probability, we have:

$$0 \le \rho_{kj} \le 1 \qquad (8c)$$

3. SOME INSIGHTS
Out of all replication strategies, we want to investigate what is the optimal way to minimize the average access latency. The optimization problem is:

$$\min \sum_{k=1}^{M}\sum_{j=1}^{N}\frac{\lambda_{kj}}{M}\left[t_g - t_{gl}\rho_{kj} + t_{Gg}\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})\right] \quad (9)$$

s.t.

$$\sum_{j=1}^{N}\sum_{k=1}^{M}\rho_{kj} = KM$$

$$\sum_{j=1}^{N}\rho_{kj} \le K, \quad 0 \le \rho_{kj} \le 1$$

Out of these constraints, we will utilize the first constraint for the optimization solution. We use other two constraints to determine area of feasible solutions. The classical approach to solving constrained optimization problems is the method of Lagrange Multiplier. The optimization problem in expression (9) can be solved by means of Lagrange Multiplier method. The Lagrangian of our constrained optimization problem (denoted as H) is:

$$H = \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{N}\lambda_{kj}\left[t_g - t_{gl}\rho_{kj} + t_{Gg}\prod_{i=M_{g(k)-1}+1}^{M_{g(k)}}(1-\rho_{ij})\right]$$

$$+\gamma\left(\sum_{j=1}^{N}\sum_{k=1}^{M}\rho_{kj}-KM\right)$$

Minimizing H over all ρ_{kj} for $k=1$ to M and $j=1$ to N,

$$\frac{\partial H}{\partial \rho_{kj}}=-\frac{1}{M}\lambda_{kj}\left[t_{gl}+t_{Gg}\prod_{\substack{i=M_{g(k)-1}+1\\i\neq k}}^{M_{g(k)}}(1-\rho_{ij})\right]$$

$$-\frac{1}{M}\prod_{\substack{i=M_{g(k)-1}+1\\i\neq k}}^{M_{g(k)}}(1-\rho_{ij})\sum_{\substack{i=M_{g(k)-1}+1\\i\neq k}}^{M_{g(k)}}\lambda_{ij}t_{Gg}+\gamma$$

$$=-\frac{1}{M}\left\{\lambda_{kj}t_{gl}+t_{Gg}\prod_{\substack{i=M_{g(k)-1}+1\\i\neq k}}^{M_{g(k)}}(1-\rho_{ij})\sum_{\substack{i=M_{g(k)-1}+1\\i\neq k}}^{M_{g(k)}}\lambda_{ij}\right\}+\gamma$$

$$=0$$

i.e.

$$\lambda_{kj}t_{gl}+t_{Gg}\prod_{\substack{i=M_{g(k)-1}+1\\i\neq k}}^{M_{g(k)}}(1-\rho_{ij})\sum_{i=M_{g(k)-1}+1}^{M_{g(k)}}\lambda_{ij}=M\gamma \qquad (10)$$

Analyzing Eq.(10) we can get some preliminary results about the relationship between request distribution and replica distribution. In some previous works [5, 15, 16] including ours [10], when request is assumed to be uniformly distributed in the system, then the replica is also assumed to be uniformly distributed in system too. However, this is just an intuitive assumption, and they did not explain why these two factors are reacted each other. We will investigate whether this assumption is held in the sense of minimizing average access latency in federated data grid systems.

3.1 Uniform Request vs. Uniform Replica Distribution in a Sub Grid

In this section, we will discuss relationship between request distribution and replica distribution. Are these two factors independent with each other? We analyze this problem by means of the request for a file f_j. From Eq.(10) we can know for any two request rate variables for file f_j, λ_{kj} and λ_{hj}, the equation below is held:

$$\lambda_{hj}t_{gl}+t_{Gg}\prod_{\substack{i=M_{g(k)-1}+1\\i\neq h}}^{M_{g(k)}}(1-\rho_{ij})\sum_{i=M_{g(k)-1}+1}^{M_{g(k)}}\lambda_{ij}$$

$$=\lambda_{kj}t_{gl}+t_{Gg}\prod_{\substack{i=M_{g(k)-1}+1\\i\neq h}}^{M_{g(k)}}(1-\rho_{ij})\sum_{i=M_{g(k)-1}+1}^{M_{g(k)}}\lambda_{ij}\,.$$

If $\lambda_{kj}=\lambda_{hj}$ and $g(k)=g(h)$, i.e. node n_k and node n_h belong to the same sub grid, then we have

$$\prod_{\substack{i=M_{g(k)-1}+1\\i\neq h}}^{M_{g(k)}}(1-\rho_{ij})=\prod_{\substack{i=M_{g(k)-1}+1\\i\neq k}}^{M_{g(k)}}(1-\rho_{ij})\,,$$

i.e. $(1-\rho_{kj})=(1-\rho_{hj})$, so we can get $\rho_{kj}=\rho_{hj}$, which means that when two nodes in the same sub grid have the same request rate for a given file, then in the sense of optimal replication strategy, the probability that they hold a replica of this file is same. Specifically speaking, if the request rate for a given file is uniform in a sub grid, then the replica is uniformly distributed in this sub grid in the optimal replication strategy.

Furthermore, if $\lambda_{kj}>\lambda_{hj}$, we have $(1-\rho_{kj})<(1-\rho_{hj})$, i.e. $\rho_{kj}>\rho_{hj}$, which means the more popular a file is in a node, the higher probability this node will have a replica of this file.

3.2 Relationship between Different Files in a Sub Grid

Another question is what is the relationship between different files in a sub grid or in a node? To answer this question, we consider a specific case.

Next we investigate the case when requests for a file are uniformly distributed in every sub grid but maybe varied across sub grids. This means that if $g(k)=g(h)$ and $\lambda_{kj}=\lambda_{hj}$, then $\rho_{kj}=\rho_{hj}$. So Eq.(10) can be rewritten as

$$\lambda_{kj}t_{gl}+t_{Gg}m_{g(k)}\lambda_{kj}(1-\rho_{kj})^{m_{g(k)}-1}=M\gamma\,,\text{ i.e.}$$

$$\lambda_{kj}\left[t_{gl}+t_{Gg}m_{g(k)}(1-\rho_{kj})^{m_{g(k)}-1}\right]=M\gamma \qquad (11)$$

We first consider the request for file f_j and file f_d in the same sub grid. From Eq.(11) we can get:

$$\lambda_{kj}\left[t_{gl}+t_{Gg}m_{g(k)}(1-\rho_{kj})^{m_{g(k)}-1}\right]$$

$$=\lambda_{kd}\left[t_{gl}+t_{Gg}m_{g(k)}(1-\rho_{kd})^{m_{g(k)}-1}\right]$$

If $\lambda_{kj}\geq\lambda_{kd}$, we can get $\rho_{kj}\geq\rho_{kd}$. This means that the more popular a file is in a sub grid, the more replicas will be stored in this sub grid. Furthermore, we can get if $\min\{\lambda_{kj}\}=\lambda_{km}$, then $\min\{\rho_{kj}\}=\rho_{km}$. According to constraint of $\sum_{j=1}^{N}\rho_{kj}\leq K$, we can get

$\rho_{km}\leq K/N$. This is true for every sub grid and we can use this property to limit the scope of searching the exact value of ρ_{km}.

3.3 When Sub Grids are Equal-sized

In the following, we investigate what will happen if sub grids are equal-sized. From Eq.(11) we can see that for two equal-sized sub grids where node n_k and node n_b located, which means $m_{g(k)}=m_{g(b)}$, we have:

$$\lambda_{kj}\left[t_{gl}+t_{Gg}m_{g(k)}(1-\rho_{kj})^{m_{g(k)}-1}\right]$$

$$=\lambda_{bj}\left[t_{gl}+t_{Gg}m_{g(b)}(1-\rho_{bj})^{m_{g(b)}-1}\right]\,.$$

If $\lambda_{kj}=\lambda_{bj}$, we can get $\rho_{kj}=\rho_{bj}$, which means that for two equal-sized sub grids, if their request rates for a given file is same, then the replica probability for this file is same in the sense of minimizing average access latency. Furthermore, if all sub grids are equal-sized, and the requests for a given file are uniformly distributed across all sub grids, i.e. in all nodes of system, we can know that the replicas will be uniformly distributed in the overall system. On the other hand, if $\lambda_{kj}>\lambda_{bj}$, we can get $\rho_{kj}>\rho_{bj}$. This means that for a data grid system federated by equal-sized sub

grids, if file requests are more popular in one sub grid, it will have more replicas in this sub grid too.

From the discussion above, we can infer that for equal-sized federated data grid system, when the average access latency is minimized, the trend of request distribution and the trend of replica distribution is consistent, i.e. when the request is uniformly distributed in the system, the replica is uniformly distributed in the system; when the request is clustered in some sub grids, then the replica is clustered in these sub grids too. However, we can not conclude directly the relationship for the case when sub grids are not equal-sized.

4. REPLICATION STRATEGIES WITH CLUSTERED DEMANDS

In this section, we first find the solution for a specific case of Eq.(9), then we give the definition of three popular replication strategies in distributed systems.

4.1 Solution of Optimizing Downloading Replication Strategy

Currently, we can not find the solution of Eq.(9) owing to its complexity. However, we can find the solution for some special cases. In this section, we discuss a hybrid case when the request is uniform inside sub grid but with clustered demands for different sub grids. In the following discussion, we don't require sub grids be equal-sized. To prevent confuse the notation in Section 2.2, we give the terms used in this section below.

For file f_j, it has request rate $\widetilde{\lambda}_{ij}$ per node in sub grid g_i; accordingly, the probability of file f_j has a replica in every node of sub grid g_i is $\widetilde{\rho}_{ij}$. In overall system, file f_j has

$$r_j = \sum_{i=1}^{G} m_i \widetilde{\rho}_{ij}$$ replicas totally.

Clustering ratio c_{ij} means the percent of all requests of file f_j that is originated from sub grid g_i, i.e. $c_{ij} = m_i \widetilde{\lambda}_{ij}/(M\lambda_j)$. We call c_j = max c_{ij} as maximum clustering ratio for file f_j. We call vector C_j = $(c_{1j}, c_{2j}, ..., c_{Gj})$ as clustering ratio vector for file f_j. Correspondingly, we call the vector $\Lambda_j = (\lambda_{1j}, \lambda_{2j}, ..., \lambda_{Gj})$ as request rate vector (or clustering pattern) for file f_j. We call the vector $\Lambda = (\Lambda_1, \Lambda_2, ..., \Lambda_N)$ as clustering pattern of all files.

Considering the new notations here, cumulative hit ratios Eqs.(4)-(6) listed in Section 2 can be rewritten as:

$$P(local) = \frac{1}{M}\sum_{i=1}^{G} m_i \sum_{j=1}^{N} \widetilde{\lambda}_{ij} \widetilde{\rho}_{ij} \qquad (12)$$

$$P(intra) = \frac{1}{M}\sum_{i=1}^{G} m_i \sum_{j=1}^{N} \widetilde{\lambda}_{ij} \left[1 - \widetilde{\rho}_{ij} - (1 - \widetilde{\rho}_{ij})^{m_i} \right] \qquad (13)$$

$$P(inter) = \frac{1}{M}\sum_{i=1}^{G} m_i \sum_{j=1}^{N} \widetilde{\lambda}_{ij} (1 - \widetilde{\rho}_{ij})^{m_i} \qquad (14)$$

Average access latency Eq.(7) can be rewritten as:

$$t = \frac{1}{M}\sum_{i=1}^{G} m_i \sum_{j=1}^{N} \widetilde{\lambda}_{ij} \left[t_g - t_{gl}\widetilde{\rho}_{ij} + t_{Gg}(1 - \widetilde{\rho}_{ij})^{m_i} \right] \qquad (15)$$

The constraints Eqs.(8a), (8b), and (8c) can be rewritten as:

$$\sum_{j=1}^{N}\sum_{i=1}^{G} \frac{m_i}{M}\widetilde{\rho}_{ij} = K \qquad (16a)$$

$$\sum_{j=1}^{N} \widetilde{\rho}_{ij} \leq K \qquad (16b)$$

$$0 \leq \widetilde{\rho}_{ij} \leq 1, \qquad (16c)$$

The optimization problem Eq.(9) can be rewritten as:

$$\min\left\{ \frac{1}{M}\sum_{i=1}^{G} m_i \sum_{j=1}^{N} \widetilde{\lambda}_{ij} \left[t_g - t_{gl}\widetilde{\rho}_{ij} + t_{Gg}(1 - \widetilde{\rho}_{ij})^{m_i} \right] \right\} \quad (17)$$

s.t.

$$\sum_{j=1}^{N}\sum_{i=1}^{G} \frac{m_i}{M}\widetilde{\rho}_{ij} = K$$

$$0 \leq \widetilde{\rho}_{ij} \leq 1, \quad \sum_{j=1}^{N} \widetilde{\rho}_{ij} \leq K$$

The Lagrangian of our constrained optimization problem Eq.(17) is

$$\widetilde{H} = \frac{1}{M}\sum_{i=1}^{G} m_i \sum_{j=1}^{N} \widetilde{\lambda}_{ij} \left[t_g - t_{gl}\widetilde{\rho}_{ij} + t_{Gg}(1 - \widetilde{\rho}_{ij})^{m_i} \right]$$

$$+ \widetilde{\gamma} \left(\sum_{j=1}^{N}\sum_{i=1}^{G} \frac{m_i}{M}\widetilde{\rho}_{ij} - K \right)$$

Minimizing $\widetilde{H}$ over all $\widetilde{\rho}_{ij}$ for $i = 1$ to G and $j = 1$ to N, we can get:

$$\frac{\partial \widetilde{H}}{\partial \rho_{ij}} = -\frac{1}{M} m_i \widetilde{\lambda}_{ij} \left[t_{gl} + t_{Gg}m_i(1 - \widetilde{\rho}_{ij})^{m_i-1} \right] + \widetilde{\gamma}\frac{m_i}{M} = 0 ,$$

or

$$\widetilde{\lambda}_{ij} \left[t_{gl} + t_{Gg}m_i(1 - \widetilde{\rho}_{ij})^{m_i-1} \right] = \widetilde{\gamma} \qquad (18)$$

For the first sub grid, we can find the minimum request rate for file f_x, i.e. $\min\{\widetilde{\lambda}_{1j}\} = \widetilde{\lambda}_{1x}$. Just as the discussion in Section 3.2, we can get $\min\{\widetilde{\rho}_{1j}\} = \widetilde{\rho}_{1x}$ and $\widetilde{\rho}_{1x} \leq K/N$.

According to Eq.(18) we can get the relation between $\widetilde{\lambda}_{ij}$ and $\widetilde{\lambda}_{1x}$. We have:

$$\widetilde{\lambda}_{ij} \left[t_{gl} + t_{Gg}m_i(1 - \widetilde{\rho}_{ij})^{m_i-1} \right] = \widetilde{\lambda}_{1x} \left[t_{gl} + t_{Gg}m_1(1 - \widetilde{\rho}_{1x})^{m_1-1} \right],$$

Let

$$\rho(i, j, \widetilde{\rho}_{1x}) = \frac{\widetilde{\lambda}_{1x}}{\widetilde{\lambda}_{ij}t_{Gg}m_i}\left[t_{gl} + t_{Gg}m_1(1 - \widetilde{\rho}_{1x})^{m_1-1} \right] - \frac{t_{gl}}{t_{Gg}m_i} .$$

So $\widetilde{\rho}_{ij}$ can be represented as a function of $\widetilde{\rho}_{1x}$, i.e.

$$\widetilde{\rho}_{ij} = 1 - \sqrt[m_i-1]{\rho(i, j, \widetilde{\rho}_{1x})} , i = 1 \text{ to } G, j = 1 \text{ to } N.$$

Considering $\sum_{j=1}^{N}\sum_{i=1}^{G}\frac{m_i}{M}\widetilde{\rho}_{ij}=K$, we can get the value $\widetilde{\rho}_{1x}$, and then we can get all $\widetilde{\rho}_{ij}$. Now we analyze the property of $\widetilde{\rho}_{ij}$ and their sum.

Let $f(\widetilde{\rho}_{1x})=\sum_{j=1}^{N}\sum_{i=1}^{G}m_i\widetilde{\rho}_{ij}-MK$.

Since $\widetilde{\rho}_{ij}$ can be seen as a function of $\widetilde{\rho}_{1x}$, the derivative of $\widetilde{\rho}_{ij}$ is:

$$\frac{d\widetilde{\rho}_{ij}}{d\widetilde{\rho}_{1x}}=\frac{\widetilde{\lambda}_{1x}}{\widetilde{\lambda}_{ij}m_i}m_1(1-\widetilde{\rho}_{1x})^{m_1-2}\left[\rho(i,j)\right]^{\frac{1}{m_i-1}-1}>0.$$

So $\frac{d}{d\widetilde{\rho}_{1x}}f(\widetilde{\rho}_{1x})=\sum_{j=1}^{N}\sum_{i=1}^{G}m_i\frac{d\widetilde{\rho}_{ij}}{d\widetilde{\rho}_{1x}}>0$.

Seeing the fact that $f(\widetilde{\rho}_{1x})$ is a strict monotone increasing function, so if $f(0)\leqslant 0$ and $f(K/N)\geqslant 0$, then $f(\widetilde{\rho}_{1x})=0$ has solution in its domain. We can solve $f(\widetilde{\rho}_{1x})=0$ by any linear search method. In this paper, we utilize Bisection Method to solve this equation. Bisection Method can shorten the region of search by one half in every search step.

Assuming the solution of $f(\widetilde{\rho}_{1x})=0$ is $\widetilde{\rho}_{1x}^{*}$, the solved file distribution of Eq.(17) is:

$$\begin{cases}\widetilde{\rho}_{ij}=1-\sqrt[m_i-1]{\rho(i,j,\widetilde{\rho}_{1x}^{*})}, i\neq 1 \ \& \ j\neq x \\ \widetilde{\rho}_{1x}=\widetilde{\rho}_{1x}^{*}\end{cases} \quad (19)$$

In this paper, we name the replication strategy utilizing this file distribution as Optimized Downloading Replication Strategy for Arbitrary Clustered demands (ODRS-AC).

4.2 Three Classical Strategies

In this section we briefly sketch the definition of three replication strategies popular employed in distributed systems, which are investigated by related works [5, 15, 16].

4.2.1 Uniform replication strategy (URS)

URS is a very primitive replication strategy. Whether file request follows which distribution, all files in data grid have same replica probability, i.e.

$$\widetilde{\rho}_{ij}=K/N \quad (20)$$

We utilize URS as a benchmark to compare the performance of other strategies irrespective it may be not used in real world.

4.2.2 Square-root Replication Strategy (SRRS)

In square root replication strategy [5, 15], the replica probability of a file is proportional to the square root of its normalized request ratio, i.e.

$$\widetilde{\rho}_{ij}\propto\sqrt{\widetilde{\lambda}_{ij}}, i=1, 2, …, G; \ j=1, 2, …, N.$$

Considering storage constraints, we have:

$$\widetilde{\rho}_{ij}=\frac{MK\sqrt{\widetilde{\lambda}_{ij}}}{\sum_{j=1}^{N}\sum_{i=1}^{G}m_i\sqrt{\widetilde{\lambda}_{ij}}} \quad (21)$$

4.2.3 Proportional Replication Strategy (PRS)

PRS [15, 16] is a natural replication strategy. In proportional replication strategy, the replica number of a file is proportional to its request ratio, i.e.

$$\widetilde{\rho}_{ij}\propto\widetilde{\lambda}_{ij}, i=1, 2, …, G; \ j=1, 2, …, N.$$

Considering storage constraints, we have:

$$\widetilde{\rho}_{ij}=K\widetilde{\lambda}_{ij} \quad (22)$$

5. SIMULATION RESULTS

In this section we compare the performance of our ODRS-AC with URS, SRRS and PRS through simulation. Since caching strategy is popularly employed in current data grid environments, we simulate the performance of a classical caching strategy, LRU.

For each replication strategy and LRU, we examine:

• What are cumulative average hit ratios and the respective bandwidth requirement in a system using this strategy?

• What is the average access latency in a system using this strategy?

• What is the performance difference between two demand patterns?

5.1 System Parameters

Table 1 lists system parameters used in this section. The size of a file in data grid is 1GByte, and the bandwidth between two nodes in a sub grid is 100 Mbps, so the time for transferring a file between two nodes in the same sub grid is $t_g=80$ sec. The bandwidth between different sub grids is 10 Mbps, so $t_G=800$ sec. Because the local hit does not need network bandwidth, so $t_l=0$ sec. Many network applications show that data request pattern over objects follows a zipf-like distribution [2] with a skewness parameter of zipf-exponent (α). We use zipf-like distribution as the file request distribution in our test. In this paper, we utilize the distribution when α = 0.6, and the results related to other values are skipped due to space constraint, which have similar trend as α = 0.6.

Table 1. System parameters

Parameter	Value
System size M	1000
Number of sub grid G	50
Mean sub grid size $\overline{m}$	20
Node capacity K	5
Number of file N	100

(1) Grid size profile

In our simulation test, we borrow the idea from GT-ITM [17] to generate the size of every sub grid. The result we get is a random array of sub grid size, which has a mean sub grid size and a

standard deviation. The standard deviation measures how widely spreads the values in grid size profile are. For the presentation reason, we sort the produced sub grid size in descending order. We should note that the rank in Figure 1 may be different from the sub grid index in simulation test. The standard deviation of grid size profile shown in Figure 1 is 1.79, the biggest one is 24, and the smallest one is 17. The results about other GSPs are omitted here due to space constraints.
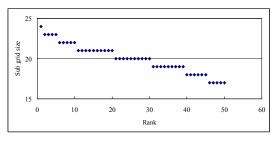

Figure 1. Grid size distribution

(2) Demand patterns

To simplify the presentation, for a file, there is a sub grid holding the maximum clustering ratio c_j, and the clustering ratio in other sub grids keep same, i.e. clustering ratio vector for file f_j is: $\boldsymbol{C_j} = (c_{1j}, c_{2j}, ..., c_{Gj})$, in which $c_{ij} = \boldsymbol{c_j}$ and $c_{hj} = (1-\boldsymbol{c_j})/(G-1)$, $h=1, 2, ..., j-1, j+1, ..., G$. For a given file f_j, all sub grids have same probability to hold the maximum clustering ratio of this file. Since the sub grids may be not equal-sized, the request rate is not uniformly distributed in the nodes of other sub grids. According to the statistics in [7], clustering tends to be more pronounced for non popular files, so we pay more attention to those non popular files. In our simulation, the requests for top 25 most popular files are uniformly distributed in every sub grid, i.e. every sub grid hold 2% of requests for these files. For the rest files, we consider 18 demand patterns, in which all files hold same maximum clustering ratio. In the first demand pattern, all sub grids produce the same proportion of request for a given file. In the last demand pattern, there is a sub grid produce 40 percent requests for a given file. The results about other complex demand patterns are omitted here due to space constraints.

5.2 Simulation Method

It is difficult to model all dynamics of an Internet-based data grid system. We don't consider the problems of node coming/going and search operation in system. While our model does not capture all aspects of reality, it helps us understand the fundamental properties of replication strategies with clustered demands.

For URS, SRRS, PRS and ODRS-AC, we can compute replica probability distribution for every request distribution and the number of replicas for every file, and then distribute these replicas to nodes of data grid according to their probability distribution. Owing to the randomness of distribution process, the result of distribution process is different for every run. We run the distribution process for 100 times for every demand pattern. To test system performance, we run 1000 simulation iterations for every distribution result. A node requests a file on average in one simulation iteration. So, in the whole simulation, every node makes 100,000 file requests on average. For LRU, we collect the

statistics over a simulation period that every node makes 100,000 file requests on average.

In the test about LRU, every node in data grid employs LRU algorithm to manage its storage space. The system only has a master replica for every file in initial state. The master replica of every file can not be deleted from system to guarantee that there is at least one replica of every file in system. When a file request misses at a node, it prefers to fetch this file's replica from a node at the same sub grid with origin requesting node. If the sub grid hasn't the replica of requested file, the requesting node should fetch the file's replica from a node outside local sub grid. If the storage space of a node is full when a new file replica arrives, the least recently used item should be evicted. With the running of simulation, the storage space of all nodes in data grid will be full at some time. We collect the statistics of simulation process since the storage space of every node becomes full. We should note that the replication strategies we simulate in this paper keep the content of every node unchanged in the simulation process, but in LRU test, the replica distribution fluctuate with every file request.

When running simulation program, we collect the statistics about local site hit number, intra sub grid hit number and inter sub grid hit number respectively for every strategy. Dividing the hit number by the total request number, we can get the cumulative hit ratios. By means of these hit ratios, we can get average access latency ultimately.

5.3 Cumulative Hit Ratios

Figure 2 shows the result about cumulative local site hit ratio. We can see that when maximum clustering ratios are less than 22%, PRS holds the highest local site hit ratio among all strategies considered here. On the other hand, when maximum clustering ratios exceed 22%, LRU holds the highest local site hit ratio. The third one is ODRS-AC, and URS holds the lowest local hit ratio. With the increasing of maximum clustering ratios of those less popular files, the local site hit ratio of LRU, PRS, SRRS and ODRS-AC increases too, which means that the more skewed a demand pattern is, the more requests will be served by local site for these strategies. However, the local site hit ratio for URS keeps unchanged for all demand patterns. An interesting phenomenon is that the curve of SRRS is similar to that of ODRS-AC.
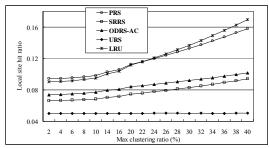

Figure 2. Comparison of cumulative local site hit ratio

Figure 3 shows the trend of inter grid hit ratio, which can be interpreted as the wide area network bandwidth requirement if sub grids are connected by wide area network. In these strategies, URS occupy the highest inter grid hit ratio. The second one is LRU, and the third one is PRS. While PRS and LRU occupy the

highest local site hit ratio for different maximum clustering ratios, this will not produce optimal performance. We can see that the curves of ODRS-AC and SRRS is below that of PRS, which means that PRS and LRU will require more wide area network bandwidth compared with ODRS-AC and SRRS. The curve of ODRS-AC is the lowest one among all strategies, which is a side effect when we minimize average access latency. With the increasing of maximum clustering ratios, the curves except URS decline, which means that the more skewed the demand patterns are, the fewer requests should be served by other sub grids, or the fewer requirements for wide area network bandwidth are.
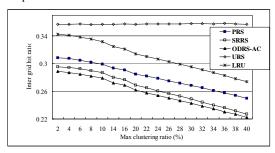

Figure 3. Trend of cumulative inter grid hit ratio

Figure 4 presents the trend of intra grid hit ratio for every strategy. SRRS occupy the highest intra grid hit ratio. The second one is ODRS-AC. We can see that the curves of SRRS and ODRS-AC differ little, which is because the curves of local site hit ratio and inter grid hit ratio for these two strategies are with similar trend. For some other system configurations, we find similar phenomenon, and we can not explain it very clearly. Since the focus of this paper is to model replication strategies with arbitrary clustered demands and give optimal replication strategy, we will investigate this problem in future works. The intra grid hit ratio for SRRS and ODRS-AC increases with the increasing of maximum clustering ratios. For URS, the intra grid hit ratios keep nearly unchanged for all demand patterns. For PRS, we can see that when maximum clustering ratio is relatively little, it increases slowly. However, when demand pattern become more skewed, the intra grid hit ratio for PRS decreases. Out of these five strategies, the intra grid hit ratio of LRU is lowest, and it exhibits similar trend as that of PRS.
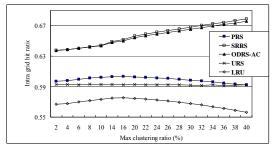

Figure 4. Trend of cumulative intra grid hit ratio

5.4 Average Access Latency

For a data grid system, average access latency is an important metric. Currently, for most systems, wide area network bandwidth is less than local area network bandwidth, so inter grid hit ratio is the dominant factor for average access latency. The curves of every strategy in Figure 5 exhibit similar trend as that of Figure 3 respectively and we omit the analysis about them due to space constraints. Figure 5 shows that ODRS-AC occupies least average access latency among five strategies, which is consistent with the analysis result. We should note that while PRS and LRU have higher local hit ratio, but their average access latency is more than ODRS-AC and SRRS. So we can say that the strategy only improving local hit ratio can not guarantee the optimal performance.
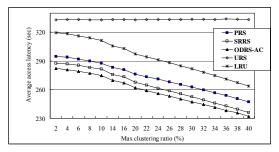

Figure 5. Trend of average access latency

6. RELATED WORKS

Current research works about replica management in data grid mainly focus on caching strategies [1, 3, 8, 14]. Their motivation is to improve hit ratio of nodes' caches and to decrease access latency. Since the decision of every node is on the basis of local information, the performance maybe not optimal at the view of overall system. Just as simulation results in Section 5 showed, we found that just improve local hit ratio does not produce optimal result.

Previously, some works on replication strategies in unstructured peer to peer networks assume that the requests and data replicas uniformly distributed in whole system [5, 16]. Proportional replication strategy and square-root replication strategy are two strategies investigated by these works. However, the measurement traces show that there are clustering existed in user requests and replica distribution. These works include measurement about data grid systems [8, 9] and peer to peer systems [7]. There are two kinds of clustering, geographical and semantic. The former measures how much overlapping between the contents of geographical nearly sites. The latter measures how much overlapping between those nodes whose interests are similar. According to this classification method, the clustering considered in this paper is mainly concerned with geographical clustering. In [15], the authors deal with the optimal search performance for unstructured peer to peer networks. The main results of [15] are that PRS can minimize average search time while SRRS can minimize query-processing load. As the results in Section 5 showed, PRS and SRRS don't produce optimal download performance in federated data grid systems.

7. CONCLUSION AND FUTURE WORKS

In this paper, we give the model of replication strategies in data grid systems when workload exhibits arbitrary clustered demands. By means of optimization average access latency, we get some relationship between demand pattern and replica distribution pattern. We find that when request for a file is uniformly

distributed in a sub grid, then the replicas of this file should be uniformly distributed in this sub grid in the sense of minimization of average access latency. Secondly, the analysis results show that the more popular a file is in a sub grid, the higher the probability there is replica in this sub grid. Thirdly, when sub grids are equal-sized and the request for a given file is uniformly distributed in system, then replicas of this file is uniformly distributed in system in the sense of minimization of average access latency. Based on the analysis results above, we solve an optimization problem about minimization average access latency when sub grids are not equal-sized and every sub grid can exhibit different clustered patterns. Simulation results show that the optimized downloading replication strategy not only minimize average access latency, but also can lower wide area network bandwidth requirement relative to some popular strategies listed in paper. In future works, we will test the performance of this optimal strategy in real systems. Furthermore, we will investigate the relationship between replica distribution and request distribution when storage space capacity is heterogonous.

9. REFERENCES
[1] Bell, W.H., Cameron, D.G., Carvajal-Schiaffino, R., Millar, A.P., Stockinger, K. and Zini, F., Evaluation of an Economy-Based File Replication Strategy for a Data Grid. In *Proceedings of the 3rd International Symposium on Cluster Computing and the Grid*, IEEE Computer Society, Washington, DC, USA, 2003, 661-668.

[2] Breslau, L., Cao, P., Fan, L., Phillips, G. and Shenker, S. Web Caching and Zipf-Like Distributions: Evidence and Implications. In *Proceedings of the 18th IEEE International Conference on Computer Communications*, IEEE Computer Society, Washington, DC, USA, 1999, 126-134.

[3] Cameron, D.G., Millar, A.P., Nicholson, C., Carvajal-Schiaffino, R., Stockinger, K. and Zini, F. Analysis of scheduling and replica optimisation strategies for data grids using OptorSim. *Journal of Grid Computing*, 2,1 (Mar. 2004), 57-69.

[4] Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S. The Data Grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications*, 23, 3 (2001), 187-200.

[5] Cohen, E. and Shenker, S. Replication strategies in unstructured peer-to-peer networks. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM Press, New York, NY, USA, 2002, 177-190.

[6] Foster, I. and Kesselman, C. (eds.). *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers Inc., 1999.

[7] Handurukande, S.B., Kermarrec, A.M., Fessant, F.L., Massouli, L. and Patarin, S. Peer sharing behaviour in the eDonkey network, and implications for the design of server-less file sharing systems. *SIGOPS Oper. Syst. Rev.*, 40, 4 (Oct. 2006), 359-371.

[8] Iamnitchi, A., Doraimani, S. and Garzoglio, G., Filecules in High-Energy Physics: Characteristics and Impact on Resource Management. In *Proceedings of the 15th IEEE International Symposium on High Performance Distributed Computing*, IEEE Computer Society, Washington, DC, USA, 2006, 69-80.

[9] Iamnitchi, A., Ripeanu, M. and Foster, I. Small-World File-Sharing Communities. In *Proceedings of the 23rd IEEE International Conference on Computer Communications*, IEEE Computer Society, Washington, DC, USA, 2004, 952-963.

[10] Jiang, J. and Yang, G. An optimal replication strategy for data grid systems. *Frontiers of Computer Science in China*, 1, 3 (Jul. 2007), 338-348.

[11] Laoutaris, N., Telelis, O., Zissimopoulos, V. and Stavrakakis, I. Distributed Selfish Replication. *IEEE Trans. Parallel Distrib. Syst.*, 17, 12 (Dec. 2006), 1401-1413.

[12] Laoutaris, N., Zervas, G., Bestavros, A. and Kollios, G. The Cache Inference Problem and its Application to Content and Request Routing. In *Proceedings of the 26th IEEE International Conference on Computer Communications*, IEEE Computer Society, Washington, DC, USA, 2007, 848-856.

[13] Leff, A., Wolf, J.L. and Yu, P.S. Replication Algorithms in a Remote Caching Architecture. *IEEE Trans. Parallel Distrib. Syst.*, 4, 11 (Nov. 1993), 1185-1204.

[14] Otoo, E., Rotem, D. and Romosan, A., Optimal File-Bundle Caching Algorithms for Data-Grids. In *Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, IEEE Computer Society, Washington, DC, USA, 2004.

[15] Tewari, S. and Kleinrock, L. Optimal Search Performance in Unstructured Peer-to-Peer Networks With Clustered Demands. *IEEE Journal on Selected Areas in Communications*, 25, 1 (Jan, 2007), 84-95.

[16] Tewari, S. and Kleinrock, L. Proportional Replication in Peer-to-Peer Network. In *Proceedings of the 25th IEEE International Conference on Computer Communications*, IEEE Computer Society, Washington, DC, USA, 2006.

[17] Zegura, E., Calvert, K. and Bhattacharjee, S. How to model an Internetwork. In *Proceedings of the 15th IEEE International Conference on Computer Communications*, IEEE Computer Society, Washington, DC, USA, 1996, 594-602.