Research on Decentralized Group Replication Strategy Based on Correlated Patterns Mining in Data Grids

Danyang Qin[1(✉)], Ruixue Liu[2], Jiaqi Zhen[1], Songxiang Yang[1], and Erfu Wang[1]

[1] Key Laboratory of Electronics Engineering, Heilongjiang University, Harbin 150080, People's Republic of China
{qindanyang, zhenjiaqi, yangsongxiang, wangerfu}@hlju.edu.cn

[2] Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, People's Republic of China
liuruixue@hlju.edu.cn

Abstract. Aiming at the problem that most of the existing data mining based replication strategies cannot extract correlations between files effectively, a new decentralized replication strategy based on maximal frequent correlated patterns mining, called RSMFCP, is proposed. By translating the files access history to the binary access history, applying maximal frequent correlated patterns mining and performing replication, RSMFCP can extremely eliminate redundancy and optimize the replication performance. Data analysis and simulation results show that, comparing with other strategies like no replication, PRA, DR2 and PDDRA, RSMFCP can extract correlations more effectively and gain lower mean job execute time under different access patterns, which will provide a new option to reduce transmission delay in data grid.

Keywords: Data mining · Correlated patterns · Data replication · Distributed groups

1 Introduction

Data grid is a kind of integrated architecture to manage plenty of distributed data generated in some scientific, financial and medical fields [1]. In data grids, using data replication strategies can greatly reduce the bandwidth cost, improve the response time and maintain the reliability of the system. However, only single files are considered as the replicating object in most of the existing replication strategies, and the relationships between files are neglected. Because of the fact that nowadays many intensive

D. Qin—This work is supported by the National Natural Science Foundation of China (61302074, 61501176, 61571181), Natural Science Foundation of Heilongjiang Province (QC2013C061), Modern Sensor Technology Research and Innovation Team Foundation of Heilongjiang Province (2012TD007), and Postdoctoral Research Foundation of Heilongjiang Province (LBH-Q15121).

X.-L. Huang (Ed.): MLICOM 2016, LNICST 183, pp. 293–302, 2017.
DOI: 10.1007/978-3-319-52730-7_30

applications need to discover the relationships between files, it is important to extract file correlations more effectively in the related research fields. Data mining can help extract valuable information from the large data sets. Using data mining in data grids can effectively find hidden correlations between files, thus achieve the goal to optimize the replicas management module.

Two measures can be concluded to discover the hidden correlations between files, which are frequent sequence mining and correlated patterns mining. Typical strategies like PRA [2] and PDDRA [3] are mainly based on frequent sequence mining. In order to predict future requested files, when execute the strategies aforementioned, the process of frequent sequence mining will be constantly running, which will increase the number of replicas and greatly impact the value of response time and occupied storage percentage in data grids. As one of the traditional correlated patterns mining based strategies, Apriori [4] can identify the frequent item sets from the large-scale data sets and produce strong correlated patterns. Apriori is a kind of sophisticated data mining algorithm, whose optimized and derived mining measures can be applied in many different industries and fields [5–7]. However, most of the common correlated patterns mining based strategies are redundant and cannot reflect the true relationships between files to some extent [8, 9]. Therefore, based on the previous research, define the groups of associated files distributed in different sites as the distributed groups and propose a Replication Strategy based on Maximal Frequent Correlated Patterns (RSMFCP). By optimizing period parameter and designing a Maximal Frequent Correlated Patterns Miner (MFCPM), RSMFCP can be periodically invoked in the real data grids, which will help realize the goals to reduce the network delay and quickly access the valuable remote files.

2 Maximal Frequent Correlated Patterns Mining

2.1 Basic Definitions

Item is a kind of binary attribute, using logical value 0 or 1 to indicate whether the given job can access the corresponding files or not. Suppose $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ as a set of n items and the transaction associated with a unique identifier as a subset of $\mathcal{I}$. In this paper, items are defined as the accessed target files, the transaction is defined as a set of files accessed by the given job and the pattern is defined as an item set. With regard to an item set $X \subseteq \mathcal{I}$, $Supp(X)$ represents the support of X, which is calculated as the ratio of the number of transactions including X to the number of all the transactions. In fact, $Supp(X)$ is the probability of the emergence of the transactions including X. If the support of the pattern is not less than the minimum support threshold specified by users, then the pattern is frequent. Moreover, if the correlation measure of the pattern, which is denoted as $Corr(X)$ is not less than the minimum correlation measure threshold, then the pattern is correlated.

Definition 1 All-Confidence. All-confidence is a kind of correlation measure used to estimate the correlated degree for the patterns. The all-confidence of the item set $X \subseteq \mathcal{I}$ can be calculated by Eq. (1).

$$all - confidence(X) = \frac{Supp(\wedge X)}{\max\{Supp(\wedge i)|i \in X\}} \quad (1)$$

where $\max\{Supp(\wedge i)|i \in X\}$ represents the maximum support of the items in X, $Supp(\wedge X)$ represents the support of X, and i represents an item in X. All-confidence simultaneously possesses the anti-monotone, cross-support and null-invariant properties:

(1) Anti-monotone property. For any item set $I \subseteq \mathcal{I}$, $I_1 \subseteq I$, if the fact that I satisfies the constraint Q can infer that I_1 also satisfies Q, then the constraint Q is considered to be anti-monotone.
(2) Cross-support property. Given the threshold $t \in [0, 1]$ and the item set $I \subseteq \mathcal{I}$ that contains item x and y, if $(Supp(\wedge x)/Supp(\wedge y)) < t$, then I is considered to be cross-support with respect to the threshold t.
(3) Null-invariant property. When it comes to the correlation of the pattern, the null-invariant property can make sure that only the transactions including the specific pattern are analyzed [10]. For the pattern $I \subseteq \mathcal{I}$, the transactions that do not contain I is deemed as the null transactions. It makes no sense to deduce the correlation of I according to the number of null transactions, which will also help avoid the bad influence of the null transactions.

Definition 2 Frequent correlated pattern. Support and all-confidence are the measures respectively corresponding to the frequency and correlation of the pattern. Given the minimum support threshold *minsupp* and the minimum correlation measure threshold *mincorr*, if $Supp(X) \geq minsupp$ and $Corr(X) \geq mincorr$, then the pattern X is considered to be a frequent correlated pattern.

Definition 3 Maximal frequent correlated pattern. If X is a frequent correlated pattern, and the superset of X is definitely not a correlated frequent pattern, then X is deemed to be a maximal frequent correlated pattern. The definition of the maximal frequent correlated pattern can contribute to extremely decrease the number of distributed groups to replicate, decrease the occupied storage in data grids and optimize the replicating process.

2.2 Maximal Frequent Correlated Patterns Miner

In order to mine the distributed groups in data grids, it is necessary to extract the maximal frequent correlated pattern defined before. In this section, a maximal frequent correlated patterns miner, called MFCPM, is designed by Algorithm 1. The notations used in MFCPM are defined in Table 1.

Table 1. Notations used in MFCPM.

Notation	Meaning	Notation	Meaning
X_k	A pattern X of k items	$\mathcal{FCP}_k$	A frequent correlated pattern of k items
C_k	A candidate set of k items	$\mathcal{MFCP}$	A set of maximal frequent correlated patterns

Algorithm 1. MFCPM algorithm.

```
Input: A binary access history, minsupp , min-all-confidence
Output: MFCP
Begin
    k := 1 ;
    FCP_1 := {i ∈ I | Supp(i) ≥ minsupp} ;  %determine the fcp
    MFCP := FCP_1 ;
   While FCP_k ≠ ϕ do
        FCP_{k+1} := GENERATE_NEXT_FCP(FCP_k, minsupp, min-all-confidence) ;
        Foreach (X_{k+1} ∈ FCP_{k+1}) do
            If (∃X_k ⊂ X_{k+1} | (X_k ∈ MFCP)) then
                remove X_k from MFCP
      MFCP := MFCP ∪ FCP_{k+1} ; %make sure no rp exist
      k := k + 1 ;
  Return MFCP
End
```

3 Distributed Groups Replication Strategy Based on MFCPM

3.1 The Procedures of RSMFCP

Based on the MFCPM module proposed in the last section, the RSMFCP strategy is proposed aiming at P2P data grid topology and consists of 4 phases.

Extract the files access history. To require local and remote files, the current site should locally record the files access history during every executing period and the job access order is determined by the access patterns.

Translate the files access history to a binary access history. The binary access history is essentially a logical table consists of the accessed object files and jobs.

Generate the $\mathcal{MFCP}$ pattern. Design the MFCPM module to find the hidden correlations between the distributed groups and simplify the later replication process.

Replicate and replace. Choose $\mathcal{MFCP}$ as the input of this phase, and select to retain or replace the files primarily by calculating the average weight of the files to replicate and delete.

3.2 The Translation of the Binary Access History

Each site should maintain its files access history. The files access history of the site S_i is defined as a matrix $\mathbf{A}$ of $n \times m$, while n represents the total number of jobs running in the given period, m represents the sum of accessed files and $\mathbf{A}_{j,k} = \#request(F_j, J_k)$ represents the number of times that the job J_k accesses the file F_j. Before data mining, it is necessary to translate the files access history into the binary access history including logical value 0 or 1. In order to translate more quickly, the popularity of the file is introduced. If the jobs executed in S_i frequently access F_j, then F_j is considered to be popular within the scope of S_i. The average file accessed times $AvgAccess(F_j)$ is introduced to make it more convenient to evaluate the popularity of F_j in S_i. $AvgAccess(F_j)$ is calculated by Eq. (2).

$$AvgAccess(F_j) = \frac{\sum_{k=1}^{n} \#request(F_j, J_k)}{n_j} \tag{2}$$

where n_j represents the total number of jobs that access F_j.

3.3 The Replication Process of RSMFCP

In this section, the replication process of the RSMFCP strategy will be elaborated. In order to replicate the distributed groups, choose the $\mathcal{MFCP}$ pattern as the input of the replication process. Suppose $\mathcal{MFCP} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ and any element $\alpha_i \in \mathcal{MFCP}$ is the set of the files frequently accessed by the jobs. The specific steps of the replication process of RSMFCP are as follow: (1) For each $\alpha_i \in \mathcal{MFCP}$, sort the elements in $\mathcal{MFCP}$ according to a descending order of the number of the patterns contained in α_i. (2) For each $\alpha_i \in \mathcal{MFCP}$, if the storage space in S_i is enough to store all the files in α_i, then replicate all the files in α_i to S_i. (3) Otherwise, select candidate files to delete by calculating the weight of F_j in S_i according to Eq. (3).

$$FileWeight(F_j) = \frac{size(F_j) \times \#request(F_j, S_i)}{Bandwidth(S_i, S_r)} \tag{3}$$

where $size(F_j)$ represents the size of F_j and $Bandwidth(S_i, S_r)$ represents the bandwidth between the site S_i and the site S_r that contain the best replica of F_j. (4) Calculate the average weight of the files which will be replicated and deleted respectively according to Eqs. (4) and (5).

$$AvgGroupRepWeight = \frac{1}{|ToReplicate|} \times \sum_{f \in ToReplicate} FileWeight(f) \tag{4}$$

where $AvgGroupRepWeight$ represents the average weight of the files which will be replicated, the last item represents the total weight of the files which will be replicated and $|ToReplicate|$ represents the total number of the files which will be replicated.

$$AvgCandidateDelWeight = \frac{1}{|CandidateDel|} \times \sum_{f \in CandidateDel} FileWeight(f) \tag{5}$$

where *AvgCandidateDelWeight* represents the average weight of the file which will be deleted, the last item represents the total weight of the candidate file which will be deleted and $|CandidateDel|$ represents the total number of the candidate files to delete. (5) Compare the two types of average weight values aforementioned. The candidate files will be replaced to delete with the files to replicate, or give up replicating if there is $AvgGroupRepWeight > AvgCandidateDelWeight$.

4 Performance Analysis and Simulation Evaluation

4.1 Simulation Environment

In this paper, the OptorSim [11, 12] simulator is used to test the job scheduling and replicating strategies and simulate the actual data grid topology. OptorSim is a simulation package wrote by Java, which consists of the users, resource agent and many sites. Each site consists of the Computing Element (CE), Replica Management (RM) and Storage Element (SE). The simulation environment in this paper is CMS testbed grid. The CMS testbed grid consists of 20 imitative sites in Europe and America. Except for the sites in CERN and FNAL own the storage of 100 Gb, the other sites all own the storage of 50 Gb and a CE. At the beginning, the initial size of the files in the distributed groups is 1 Gb, the total number of the files is 97, the total number of the jobs is 1000 and all stored in the SEs. In addition, the sequential access pattern is selected to access the files, and the current and queued jobs access cost scheduling algorithm is applied to schedule the jobs.

4.2 Impact of the Executing Period on Strategy Performance

Considering that the given number of jobs is 1000, *minsupp* and *min-all-confidence* are both fixed, analyze the impact of different executing periods on the mean job execute time of RSMFCP. Mean job execute time is defined as the total individual executing time of every job divided by the total number of jobs executed. The mean job execute time is shorter, the performance of RSMFCP is better. The simulation result is shown in Fig. 1. When 1000 jobs are executed, it is not hard to deduce that, invoking RSMFCP

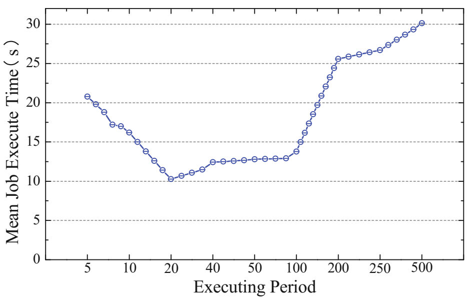

Fig. 1. Mean job execute time of RSMFCP for different periods.

after executing every 20 jobs (2%) can obtain the minimum mean job execute time. Whether the period is shorter or longer will lead to frequently accessing the remote files, which will cause the mean job execute time increase and the replicating efficiency decrease.

4.3 Impact of the Threshold to Strategy Performance

Given that the number of jobs is 1000, the executing period is 2%, *min-all-confidence* and *minsupp* respectively equal to 0.2, 0.4 and 0.6, analyze the impact of the related *minsupp* and *min-all-confidence* on mean job execute time of RFMFCP. The simulation results are shown in Fig. 2.

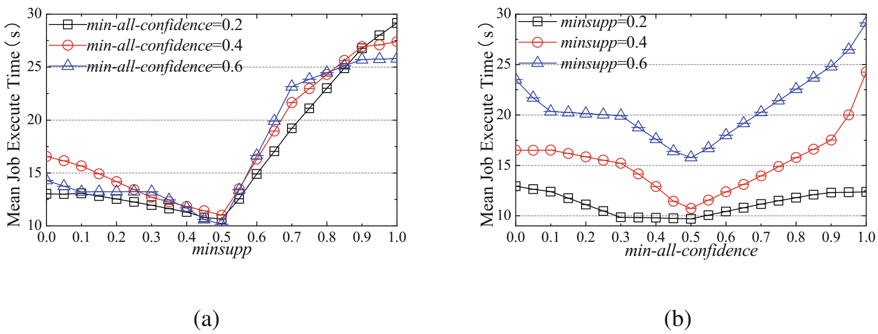

Fig. 2. Mean job execute time of RSMFCP for different thresholds (a) Mean job execute time of RSMFCP for different *minsupp* thresholds; (b) Mean job execute time of RSMFCP for different *min-all-confidence* thresholds.

It can be inferred from Fig. 2 that the mean job execute time will slowly decay when the threshold is between 0 and 0.5. In addition, the mean job execute time will rapidly grow up when the threshold exceeds 0.5, which means that the strategy performance begins to deteriorate. The simulation results show that the increase of the threshold value can result in the deterioration of the strategy. So it can be concluded that when *minsupp* and *min-all-confidence* both equal to 0.5, the mean job execute time is the smallest and the strategy performance is optimal.

4.4 Impact of the Access Patterns to Strategy Performance

Given that the number of jobs is 1000, the period is 2% and *minsupp* and *min-all-confidence* both equal to 0.5, compare the performance of the proposed RSMFCP strategy with the other four replication strategies under different access patterns. The four strategies are no replication. The five different access patterns are random access pattern, sequential access pattern, random Zipf access pattern and random walk Gaussian access pattern. Each comparison process repeats at least 10 times, after which calculate the mean values.

Mean job execute time and Effective Network Usage (ENU). The mean job execute time and ENU of the five strategies under different access patterns are shown in Fig. 3. ENU ranging from 0 to 1 is a specific ratio of the transformed files to the accessed files, which is calculated by Eq. (6). Apparently, ENU is lower, and the strategy performance is better.

$$ENU = \frac{N_{remotefileaccesses} + N_{filereplications}}{N_{remotefileaccesses} + N_{localfileaccesses}} \tag{6}$$

where $N_{remotefileaccesses}$ its the number of the accessed remote files, $N_{filereplications}$ is the number of the replicas and $N_{localfileaccesses}$ is the number of the accessed local files. The simulation results show that comparing with no replication, DR2, PRA and PDDRA strategy, the mean job execute time of RSMFCP can respectively decrease 80%, 60%, 20% and 15% at most for different access patterns. One of the main goals of the research is to minimize the bandwidth cost and decrease the network traffic, to achieve that the performance of RSMFCP is better compared with other strategies.

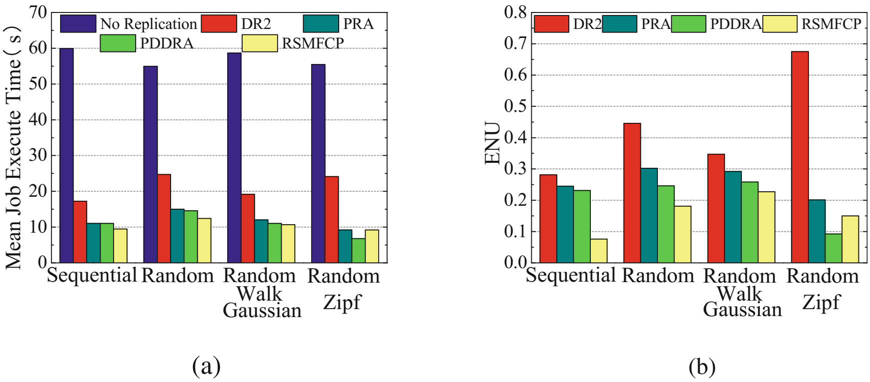

Fig. 3. Mean job execute time and ENU with different access patterns. (a) Mean job execute time with different access patterns; (b) ENU with different access patterns.

Amount of replications and occupied storage percentage. This is the number of the replicating times. Obviously, when the amount of replications is big, it indicates that most of the files required are stored in the remote sites. Besides, the occupied storage percentage is the average usage of the SEs in the grid sites. The usage of SE is the ratio of the storage resource used by files to the SE capacity. The amount of replications and occupied storage percentage of the five strategies are shown in Fig. 4. It is easy to deduce from the simulation results that the amount of replications of RSMFCP can decrease apparently with different access patterns, but it can still guarantee the availability of the files in the data grid. The amount of replications is bigger, which implies that the number of transferred files is also bigger. Therefore, the strategies of the same kind only consume the reasonable network bandwidth.

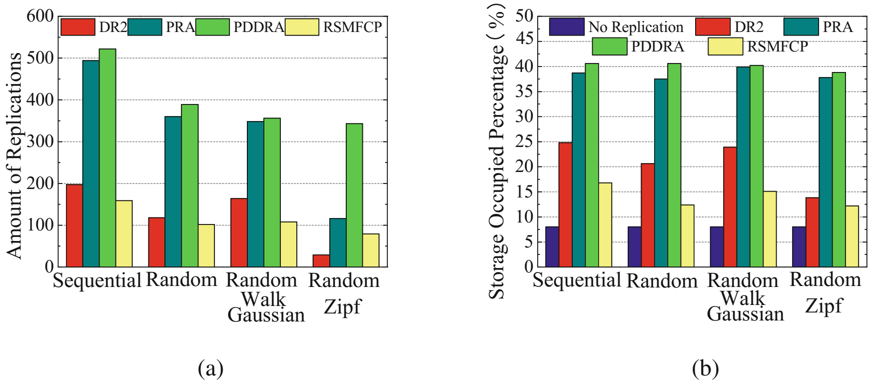

Fig. 4. Amount of replications and occupied storage percentage with different access patterns (a) Amount of replications with different access patterns; (b) Occupied storage percentage with different access patterns.

Hit Ratio (HR). HR is the ratio of the total number of times accessing the local files to the total number of times accessing all the files. HR can be calculated by Eq. (7) and the HR of the five strategies with different access patterns are shown in Fig. 5.

$$HR = \frac{N_{localfileaccess}}{N_{remotefileaccess} + N_{replications} + N_{localfileaccesses}}. \tag{7}$$

The simulation results show that compared with the same kind DR2, PRA and PDDRA strategy, the HR of RSMFCP can respectively increase 65%, 20% and 15% at most with all the access patterns.

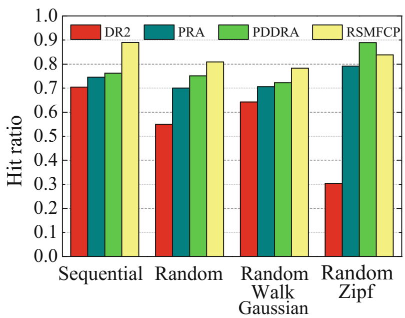

Fig. 5. Hit ratio with different access patterns.

5 Conclusion

Nowadays, the number of data generated in scientific and engineering fields gradually grows faster and faster, so the demand of computing and storing in each field is increasing. Therefore, data grid is generated as a reasonable solution. In this paper, taking the distributed groups of the sites in data grid as the mining object, MFCPM was added on traditional replication strategies and the RSMFCP strategy was proposed. Compared with the same kind of strategies, the mean job execute time and ENU of RSMFCP can decrease 80% at most, meanwhile the HR of RSMFCP can increase 65% at most. The simulation results showed that RSMFCP takes the distributed groups as the object of the research, which can reduce the number of files to replicate. Thus, RSMFCP can improve the grid performance and have some certain superiority and better application prospect. The future work will aim at optimizing the files access history in each site and applying multidimensional dynamic data mining technologies, in order to further improve the replication process and make the strategy more suitable for the realistic data grid environment.

References

1. Amornsinlaphachai, P.: Efficiency of data mining models to predict academic performance and a cooperative learning model. In: 8th International Conference on Knowledge and Smart Technology (KST), pp. 66–71 (2016)
2. Lee, M.C., Leu, F.Y., Chen, Y.P.: PFRF: An adaptive data replication algorithm based on star-topology data grids. Future Gener. Comput. Syst. **28**(7), 1045–1057 (2012)
3. Saadat, N., Rahmani, A.M.: PDDRA: a new pre-fetching based dynamic data replication algorithm in data grids. Future Gener. J. Comput. Syst. **28**(4), 666–681 (2012)
4. Agrawal, R., Imielinski, T., Swami, A.: A mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD Conference, pp. 207–216 (1993)
5. Taheri, J., Zomaya, A.Z., Bouvry, P., Khan, S.U.: Hopfield neural network for simultaneous job scheduling and data replication in grids. Future Gener. Comput. Syst. **29**(8), 1885–1900 (2013)
6. Wei, H.: Correlation mining of multi-dimensional large data sets. Mod. Comput. **9**(1), 3–8 (2012)
7. Shorfuzzaman, M., Graham, P.: Adaptive popularity-driven replica placement in hierarchical data grids. J. Supercomput. **51**(3), 374–392 (2010)
8. Bellodi, E., Riguzzi, F., Lamma, E.: Statistical relational learning for workflow mining. Intell. Data Anal. **20**(3), 515–541 (2016)
9. Jian, L., Wang, C., Liu, Y., Liang, S., Yi, W.: Parallel data mining techniques on graphics processing unit with compute unified device architecture (CUDA). J. Supercomput. **64**(3), 942–967 (2013)
10. Wu, T., Chen, Y., Han, J.: Re-examination of interestingness measures in pattern Mining: a Unified framework. Data Min. Knowl. Discov. **21**(3), 371–397 (2010)
11. Grace, R.K., Manimegalai, R.: Dynamic replica placement and selection strategies in data grids – a comprehensive survey. J. Parall. Distrib. Comput. **74**(2), 2099–2108 (2014)
12. Ma, J., Liu, W., Glatard, T.: A classification of file placement and replication methods on grids. Future Gener. Comput. Syst. **29**(6), 1395–1406 (2013)