Research on Association Rules Parallel Algorithm Based on FP-Growth

Ke Chen[1], Lijun Zhang[2], Sansi Li[1], and Wende Ke[1]

[1] Department of Computer Science and Technology, Guangdong University of Petrochemical Technology, Guangdong, China
[2] School of Information & Mechanical Engineering, Beijing Institute of Graphic Communication, Beijing, China
chenke2001@163.com

Abstract. In view of the FP-Growth algorithm needs to establish huge FP-Tree to take the massive memories, when is confronted with very huge database, its algorithm obviously insufficient in efficiency, This article unified FP-Growth and Parallel Algorithm, proposed one kind association rule parallel algorithm based on the FP-Growth, this algorithm in the FP-Growth algorithm foundation, with the aid of parallel algorithm's thought that carried on the database resolution as well as the FP-tree tree the division reasonable combination, In the task allocation, the load stabilization, has done the research, the duty rational distribution, the combination, has achieved the good load stabilization, raised the algorithm speed, this algorithm is suitable in the large-scale database carries on the data mining, compared with former algorithm had the remarkable enhancement in the efficiency.

Keywords: FP-Growth, Association Rules, parallel Algorithm, Load Balancing.

1 Introduction

Along with science and technology development, the people accumulated the massive data in the scientific research as well as the daily life, how from the magnanimous data to discover that the people did not know beforehand and also the latent useful information, promotes the technical development, society's progress. This article proposed one kind based on does not have the candidate item set FP-Growth parallel algorithm, raises the algorithm efficiency by this, expands the data mining the scale. The former data mining parallel algorithm usually is carries on the database division, carries on the parallel excavation by this. However carries on the primitive database the division, causes the data relevance to unearth inaccurate, because inside the database data has the incident cross-correlation, in has certain relation mutually, if divides after forcefully, will cause excavation result inaccurate [1]. In view of the fact that the above reason, this article proposed that the retention data relevance invariable data mining parallel algorithm, causes its excavation result to be more accurate, the efficiency is higher.

C. Liu, J. Chang, and A. Yang (Eds.): ICICA 2011, Part II, CCIS 244, pp. 249–256, 2011.

2 Connection Rule Parallel Algorithm

2.1 Parallel Computing Definition

The parallel computing is refers to the parallel machine, decomposes an application divide into the child duty, assigns for the different processor, between each processor coordinates mutually, parallel holds the displeasing person duty, thus achieves accelerates to solve the speed, or solution application question scale goal [2].

2.2 Basic Conditions of Parallel Computing

(1) Parallel machine. The parallel machine contains two or two above processors at least, these processors through the interconnecting network interconnection, correspond mutually.

(2) Application question must have the degree of parallelism. That is, the application may decompose into many sub-duties; these sub-duties may parallel carry out. Decomposes an application into many sub-duty processes, is called the parallel algorithm the design.

(3) Parallel programming. Provides in the parallel machine in the parallel programming environment, realizes the parallel algorithm specifically, the establishment parallel program, and moves this procedure, thus achieves the parallel solution application question the goal.

2.3 Parallel Computing Typical Algorithms

Task Allocation Algorithm. The duty most superior assignment, put n task allocation for n processor to carries out, the i processor carries out the time which the j duty needs is cij, we design one to be most superior n task allocation for n processor execution and the worst plan, causes the benefit which produces to be most superior or worst.

Proposes the data-in by the document, the document first line has positive integer n, indicated that has n duty to assign for n processor carries out, The n line in then, each line has n integer cij,1≤i≤n,1≤j≤n, expressed that the i processor carries out the benefit which the j duty needs.

Analysis: The variable x_{ij} expression the j task allocation for the i processor in the execution, the assignment problem may indicate for following linear programming question [3].

Load Stabilization Question. The different duty, when each processor were carries out , how to use the most superior method transporting to cause quantity same which assigns in n processor [4].

Analysis: Supposes each processing node duty quantity is xi, must make each processor node duty quantity to be the same, namely each processor's duty number becomes $\sum_{i=1}^{n} x_i / n$, Therefore some processor node must call in the duty, some processor node must assign out the duty, the i processor node calls in the duty to the j processor node time is min{|j-i|,n-|j-i|}. If m duty has n processor node, the ith processor node has the ai unit's duty, the jth node needs to complete bj task, the task call is balanced, that $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_i$, from the ith processor node Calls to the jth processor time is cij, the most superior plan assigns out the xij unit's duty from the ith processor to the jth processor node, may indicate is[5]:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i\,j} x_{ij} \tag{1}$$

$$\sum_{i=1}^{m} x_{i\,j} = b_j \qquad j = 1,2,...,n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = a_j \qquad i = 1,2,...,m \tag{3}$$

$$x_{i\,j} \geq 0, i = 1,2,...,m, j = 1,2,...,n \tag{4}$$

3 Association Rules Parallel Algorithm Based on FP-Tree

3.1 Algorithm Ideas

The frequent pattern tree excavation algorithm, first, scans the database to form the frequent 1 item set, and will obtain the data item according to the descending order, simultaneously deletes the support is smaller than the smallest support project, Then, the division candidate item set for many modules and uses main from the pattern parallel algorithm, the main engine allocating task to the different processor node, causes the frequent pattern storehouse to become a smaller module, through task allocation to different processor node, thus the urge treatment speed[6]. The steps:

(1) first, scans the database to form the candidate1 item set, if the candidate support is smaller than the smallest support, we delete them.

(2) Division candidate1 item set to the different processor node, forms the FP-tree tree, and maintains their related information.

(3) Uses main pattern, excavation separately the allocating task to the different processor node and carries out the FP-tree algorithm.

(4) By the main engine collection the information which completes from different processing node processing, and carries on compiles.

(5) Output.

3.2 The FP-Tree Parallel Algorithm Carries out Process

Parallel Algorithm step

(1)Task partition, put a big task divide the line into a lot of sub- task with smalls, and assigns them to the different processor point.

(2) Correspondence exchanges information mutually at the different processor node.

(3) Combination makes the task able to more effectively assign each processor.

(4) Reflect, assign task to distribute different processor node to carry on a processing.

Example :(1) Scan database to form multifarious item (minimum support is: 2)

Table 1. The first set of frequent

TID	Item set	item set after compositor
100	a, p, c, m, f, g	f, a, c, g, p, m
200	p, b, a, c, g, f	f, a, c, g, p
300	f, a, b, y	f, a, y
400	g, c, f	f, c, g
500	y, c, a, m, f	f, a, c, m, y

(2) Establish FP-tree tree

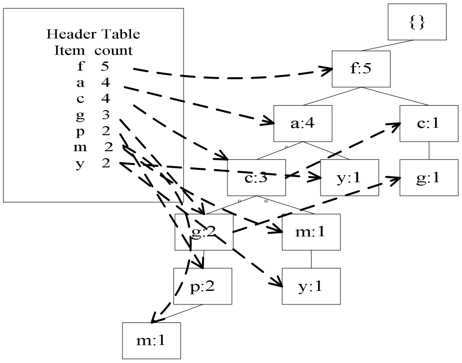

Fig. 1. The tree of frequent pattern

(3)Establish condition mode database for each node

Table 2. Conditions pattern library

Item set	Condition Mode Database
a	f:4
c	fa:3,f:1
g	fac:2,fc:1
p	facg:2
m	facgp:1,fac:1
y	facm:1,fa:1

(4) Establish the condition mode of FP-tree tree
Output result:

Table 3. For example (the minsupport is: 2)

TID	Item set	Multifarious item	Cpu
100	a, c, d, b, f, e	c, b, d, f, e, a	
200	f, c, d	c, d, f	
300	b, f, d, c, e	c, b, d, f, e	Cpu1
400	e, d, c, f, b	c, b, d, f, e	
500	c, b, a	c, b, a	
600	a, c, n, m, e	c, a, m, e, n	
700	m, b, n, c, a	c, a, m, b, n	
800	c, b	c, b	Cpu2
900	c, a, m, b, e	c, a, m, b, e	
1000	c, a, m	c, a, m	
……	……	……	Cpun

3.3 Experiment Result

Respectively with two and three processors of different arrangements start experiment, CPU and memory respectively is: 2.0GHz (dual), 1G; 1.60(dual), 2G; 1.73GHz, 0.99G. Test sources of date in: http://archive.ics.uci.edu/ml/data-sets.html. From experiment, we discover that the more the processor, the little execution time, tests result as follows:

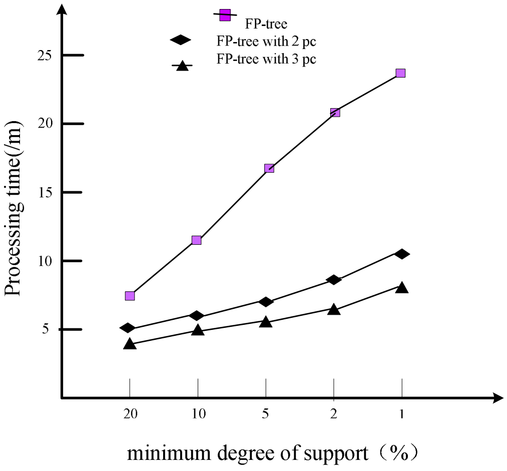

Fig. 2. The Comparison of the process time

4 FP-Growth Parallel Algorithms Based on Split

4.1 Design Thought of the Algorithm

For keeping the relativity of database data uninfluenced, put forward one kind to reserve whole database relativity don't change,

The calculate way of FP-Tree cent piece, and adopt the node of tree count method, divide equally the FP-Tree to the different node, keep a good load balance, This method can shorten the length of building up the FP-Tree, and keep the relativity of of good data,

At the same time, considered load balance and task allotment of excellent , pass to carry on FP-Tree cent piece a data of excavation, the calculate way can adapt to the data excavation of more large-scale database, can raise data excavation speed further and has been tested verify the calculate way to apply effectively[7].

4.2 Algorithm Simulation

If exists has P1, P2,…, Pn altogether N non-shared structure CPU, between them the resources are mutually independent, merely through network transmit message. But DBi(i=1,2,…, n) is the memory in Pi the Taiwan CPU business database, business has the Di strip, total business database $DB = \bigcup_{i=1}^{n} DB_i$, The general affairs several $D = \sum_{i=1}^{n} D_i$, The FP-Tree parallel computing is also works through N CPU, the processor processes own data merely, between each CPU only through network transmit message, finally result Collection on Master machine[8,9]

4.3 Experimental Results

The experiment environment and the parameter are as follows: The algorithm realizes with the C language, uses CPU is: Pentium T4200 2.00GHz; Memory: 1G; Operating system: Windows XP; Parallel environment use: mpich.nt.1.2.5, experiment data set: 40110D102K.dat and T1014D102K.dat (//http:fimi.e8.helsinki.fddata/)[10].

Table 4. 40110D102K.dat the data of set (the result of four nodes process)

Support (%)	Node 1	Node 2	Node 3	Node 4
1.5	0.41	0.43	0.43	0.45
2.0	0.22	0.23	0.24	0.20
3.5	0.03	0.03	0.035	0.05
2.5	0.10	0.10	0.12	0.15
3.0	0.03	0.03	0.02	0.01

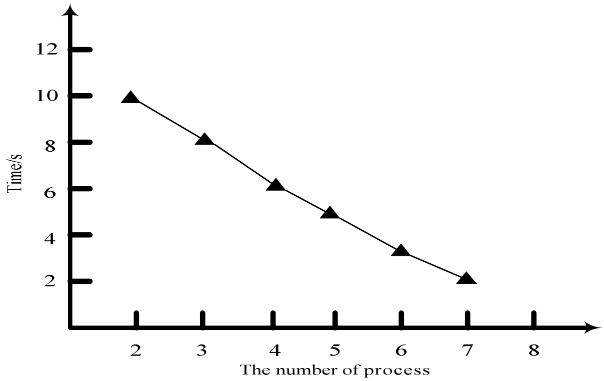

Fig. 3. The compare from the time of different nodes process

Table 5. 40110D102K. dat the data of set (the result of four nodes process)

Support (%)	Node 1	Node 2	Node 3	Node 4
9	4.90	3.66	3.69	3.95
8	6.00	5.55	6.35	6.62
7	8.82	8.33	7.25	8.63
6	10.25	9.98	9.65	12.25
5	13.05	12.00	11.66	15.05

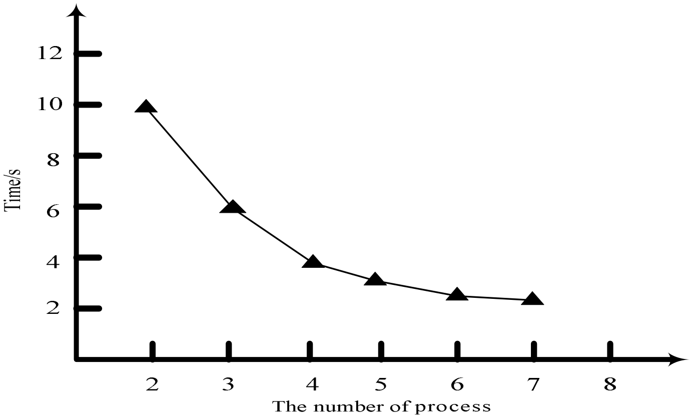

Fig. 4. The compare from the time of different nodes process

5 Summary

This article in view of the connection rule's classical algorithm FP-Growth algorithm, proposed regarding this algorithm's parallel algorithm, this algorithm does not need to scan the database many times, also does not need to have the candidate item set, This article in view of the connection rule's classical algorithm FP-Growth algorithm, proposed regarding this algorithm's parallel algorithm, this algorithm does not need to scan the database many times, also does not need to have the candidate item set, This text puts forward of reserve the calculate way of piece that the connection information carries on a cent to the FP-Tree, use this method, reserved the connection information of database. Assign task in the different processor to carry on excavation respectively, raised the efficiency of the calculate way, the calculate way also was applicable to a large-scale database to carry on data excavation.

References

1. She, C.: The data mining algorithmic analysis and the parallel pattern study, vol. 3, pp. 44–56. University of Electronic Science and Technology of China, Chengdu (2004)
2. Zhang, L.b., Chi, x.B., Mo, Z.Y.: Parallel Computing Introductory Remarks 7, 3–5 (2006)
3. Wang, G.-r., Gu, N.-j.: An Efficient Parallel Minimum Spanning Tree Algorithm on Message Passing Parallel Machine. Journal of software 11 (2006)
4. Hu, k., Cheung, D.W., Xia, S.-w.: Effect of Adaptive Interval Configuration on Parallel Mining Association Riles. Journal of Software 11 (2004)
5. Shuichi, S.: Synchronization and Pipeline Design for a Multithreaded Massively Parallel Computer (March 1992)
6. Han, J.W., Pei, J., Yin, Y.W., Mao, R.Y.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Mining and Knowledge Discovery 8, 53–87 (2004)
7. Wang, G.-r., Gu, N.-j.: An Efficient Parallel Minimum Spanning Tree Algorithm on Message Passing Parallel Machine. Journal of Software 11 (2004)
8. Liu, X.: Research and apply of the connection rule based on FP-Growth calculate way 4, 6–7 (2006)
9. Hu, Y.: Study on data mining algorithm based on the connection rule, vol. 8. Dalian Maritime affair University (2009)
10. Mao, Y.: The connection rule excavation related algorithm study, vol. (6), pp. 8–9. Southwest Jiaotong University (2009)
11. Qiu, R., Lan, R.: Highly effective FP-TREE foundation algorithm. Computer Science 3, 98–100 (2004)