

Quorum Generation Algorithm Based on the Longest-shortest Distance

<http://dx.doi.org/10.3991/ijoe.v9iS6.2922>

LI Meian, WANG Buyu and GAO Jing
Inner Mongolia Agriculture University, Huhehaote, China

Abstract—Many algorithms on quorum generation have been proposed but those algorithms with cyclic quorum usually had a common defect that the nodes would congregate in a special area of the quorum at the end of the algorithm (Called this problem as CE). For resolving this problem, a new algorithm on quorum generation based on the longest-shortest distance has been proposed in this paper. This algorithm has the same quorum length and time complexity as Li's algorithm [8], but the space complexity is $O(\sqrt{N})$, lower than Li's algorithm [8]. And this algorithm has not the problem of CE. So based on the longest-shortest distance to chose a node to insert the quorum is an executive method on resolving the problem CE.

Index Terms—distance, generation algorithm, longest-shortest distance, quorum

I. INTRODUCTION

Since the Makeawa's [1] algorithm has been proposed, researches on distributed mutual exclusion algorithm entered the competitive stage based on quorum. The applications based on quorum to decide had become more and more widely [2,3,4]. Especially the replica strategies based on quorum [5,6,7] in database gradually became one usually means. But designs of the symmetric quorum generation algorithm became more and more difficult because of the contradiction of the quorum length and the time complexity. For resolving this contradiction, LI proposed three quorum generation algorithms [8,9,10]. Firstly, he proposed a quorum generation algorithm based on cyclic coding, translating the quorum to a vector, quorums' interaction to if there are any common node. It's average quorum length is $2\sqrt{N}$, the algorithm's time complexity is $O(\sqrt{N})$ and the space complexity is $O(N^2)$. Secondly, he proposed a quorum generation algorithm base on cyclic subset set, translating the quorums' interaction to the cover of a distance vector by the distance of any two nodes in the quorum. It's average quorum length is $2\sqrt{N}$, the time complexity is $O(N)$ and the space complexity is $O(N)$. Thirdly, an algorithm has been proposed by him based on quorum properties. This algorithm has the property of backtracking. So it's quorum length is $O(\sqrt{N}+1)$, it's time complexity is $O(N)$. All these algorithms as above had a common defect that the nodes would congregate in a special area of the quorum at the end of the algorithm running (Called this problem as Congregation at the End, CE). This property made the nodes in the quorum distribute non-uniform. For resolving

this problem, a new quorum generation algorithm based on the longest-shortest distance (Called it as LSD algorithm) will be proposed in this paper. It keeps the quorum length and time complexity as same as the LI's algorithm [10] but not exist the situation of CE.

II. MODEL

Because it's difficult to search the symmetric quorum for any scale system if the quorum is not cyclic [10], we will design a quorum generation algorithm and it can generate cyclic quorum to any scale distributed system. Based on the properties of cyclic quorum proposed in LI's algorithm [10], if numbered every node in the system such as $0, 1, \dots, N-1$, the quorum of node 0 should have common nodes with the quorum of every other node in the system, and the quorum of node 1 can generate by the quorum of node 0 and so on. Use a vector to denote the quorum of node 0, such as $a_0 a_1 a_2 \dots a_{N-1}$, and a_i should be 0 or 1, 0 denotes that this node was not included in quorum and 1 denotes that this node was included in quorum. So the cyclic quorum of node i should be $a_{N-i} a_{N-i+1} a_{N-i+2} \dots a_{N-1}$ and a_i is the status of the node i if it is in the quorum. So judge the quorums of two nodes if they are interaction became judge if the inner product of the quorums is larger than 0. The inner product of $a_0 a_1 a_2 \dots a_{N-1}$ and $a_{N-i} a_{N-i+1} a_{N-i+2} \dots a_{N-1}$ is $a_0 a_{N-i} + a_1 a_{N-i+1} + a_2 a_{N-i+2} \dots a_{N-1} a_{N-i-1}$. To ensure the value of this inner product is larger than 0, there must one or more items of this inner product be 1. Because the subtract of index of a_i in every item is $N-i$ or i , and the subtract of index of a_i can be looked as the distance of two nodes in the quorum, if the distance of two nodes in the quorum can be ensured, the inner product must be larger than 0. To $i \in [1, N-1]$, if the distances of any two nodes in the quorum can cover distance i , the quorums' interaction one by one should be ensured. At the same time, because every inner product includes two distances, i and $N-i$, if the quorum of node 0 interacts the quorum of node i , the quorum of node 0 must interact the quorum of node $N-i$. So if the quorum of node 0 interacts the quorum of node $i \in [1, N/2]$, the quorum of node 0 must interact the quorum of node $i \in [N/2+1, N-1]$. So to search the cyclic quorum of node 0, only to need ensure the distances of any two nodes in the quorum can cover the area of $i \in [1, N/2]$. So the quorum generation algorithm become the problem as follow: in one N -nodes vector of 0 or 1, set some locations to 1 to ensure the distances of all the locations with 1 can cover the area of $i \in [1, N/2]$.

III. ALGORITHM IDEA

The reason of CE to the algorithms[8,9,10] is that one node included in quorum only can ensure the quorum of node 0 interacts one another quorum at the end running of the algorithm. For resolving this problem, to search a node with the longest distance to every node in the quorum and to insert it in the quorum, and this distance is the longest in all those shortest distance that the node to all those nodes in the quorum. That's to say, we should calculate the shortest distance of a node which can be included in the quorum to every nodes in the quorum firstly, then search the longest distance in all those shortest distance. This node with the longest-shortest distance should be included in the quorum. So design a quorum generation algorithm that chose a node which has the longest-shortest distance near to all those nodes in the quorum to insert in quorum become an able thing. The quorum generation algorithm in this paper should include the plans as follow:

1. Search those nodes can be included in the quorum and include them in a backup vector ;
2. Calculating the shortest distance of every node in the backup vector to every node in the quorum;
3. Search the node with the longest-shortest distance, insert it in the quorum.

IV. ALGORITHM DESCRIPTION

A. Data Structures

quorum. It's a changeable length actor, using to store the quorum nodes in the algorithm running process.

backup. It's a changeable length actor, too. Using to store the backup nodes that can be inserted in quorum.

shortest distance. It's a changeable length actor. Using to store the shortest distance of every node in the backup vector to every node in quorum.

longest shortest distance node. It's a number to use to store the node with the longest-shortest distance.

need cover distance. It's a changeable length actor, using to store those needed to cover but not be covered distance by quorum.

quorum cover distance. It's a changeable length actor, using to store those distance can be covered by quorum.

B. Algorithm Processes

generation_backup(int[]quorum,int distance,int N). Using it to generate the backup actor to quorum with the distance while the system nodes is N.

insert_node_in_quorum(int[]quorum,int[]backup,int N). Using it to insert the node with the longest-shortest distance in backup to quorum while the system nodes is N.

quorum_generation_distance(int[]quorum).Using it to generate the distance by those nodes in the quorum.

del_node(int[]a,int[]distance).Using it to delete the distance vector's distances from vector a.

C. Algorithm Description

The overflow of the algorithm proposed in this paper is as figure 1.

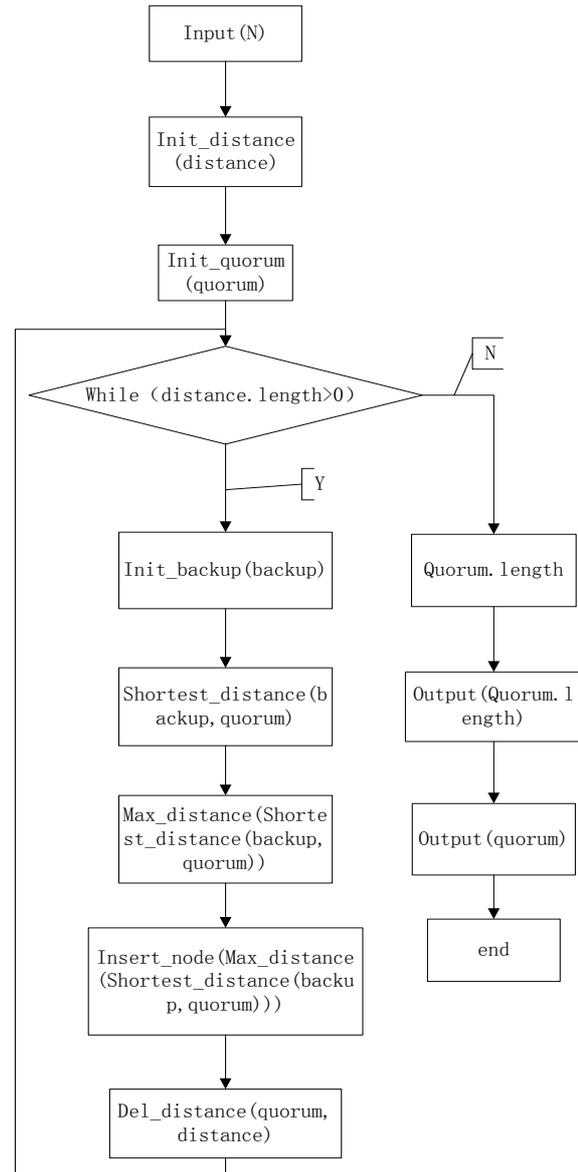


Figure 1. The overflow of the quorum generation algorithm in this paper

D. Algorithm Example

Assume N=15, the process of the algorithm running is as figure 2 to figure 5.

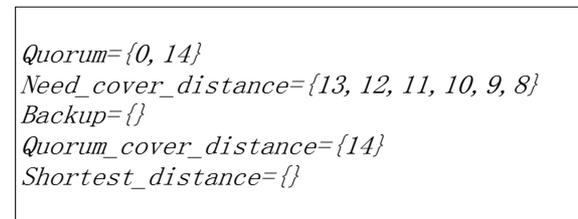


Figure 2. Algorithm initiation

```

Quorum={0, 7, 14}
Need_cover_distance={13, 12, 11, 10, 9}
Backup={6, 7, 8}
Quorum_cover_distance={7, 14}
Shortest_distance={6, 7, 6}
    
```

Figure 3. The first step

```

Quorum={0, 5, 7, 14}
Need_cover_distance={12, 11}
Backup={1, 5, 6, 8, 9, 13}
Quorum_cover_distance={2, 5, 7, 9, 14}
Shortest_distance={1, 2, 1, 1, 2, 1}
    
```

Figure 4. The second step

```

Quorum={0, 5, 7, 10, 14}
Need_cover_distance={}
Backup={1, 3, 4, 9, 10, 11}
Quorum_cover_distance={2, 3, 4, 5, 7, 9, 10, 14}
Shortest_distance={1, 2, 1, 2, 3, 3}
    
```

Figure 5. The third(ending) step

V. PERFORMANCE ANALYSIS

A. Quorum Length

When every new node was included in the quorum with n nodes, n new distances can be generated. And it can generate $n(n-1)/2$ different distances at most and $n-1$ different distances at least if there was n nodes in quorum. So the average different distances can be generated is $(n(n-1)+n)/4$ when there was n nodes in quorum. Lets the average different distances equate $N-1$, so $n=2\sqrt{N}-1$. That's to say, the average quorum length of the algorithm is $2\sqrt{N}$ in this paper. The comparison of quorum length of several algorithms is as table 1. among it the cyclic denotes the cyclic coding algorithm, \sqrt{N} and $2\sqrt{N}$ denote the algorithms with \sqrt{N} or $2\sqrt{N}$ quorum length, LSD denotes the longest-shortest distance algorithm proposed in this paper.

In table 1, LSD denotes the algorithm proposed in this paper. And from this table, one conclusion can be gotten such as fallow: the quorum length of LSD algorithm is lower when $N < 700$ and maybe is little larger than algorithm with $2\sqrt{N}$ quorum length. But in summery, the quorum length of LSD algorithm is little larger than the cyclic coding algorithm.

B. Time Complexity

Bases on the process of LSD algorithm, it needs one time query in vector `need_cover_distance`, n times plus and $n+n*n$ times subtraction, and $2n+n*n*N/2$ times comparison when there were n nodes in the quorum.

TABLE I.
 COMPARISON OF THE QUORUM LENGTH TO SEVERAL ALGORITHMS

N	7	13	21	31	43	57	73
\sqrt{N}	3	4	5	6	7	8	9
$2\sqrt{N}$	6	8	10	12	14	16	18
cyclic	3	5	6	7	9	9	13
LSD	4	5	7	9	10	13	14

N	91	111	133	157	700	1000
\sqrt{N}	10	11	12	13	27	33
$2\sqrt{N}$	20	22	24	26	54	66
cyclic	14	15	19	20	50	63
LSD	16	19	19	23	55	65

When n becomes from 2 to $2\sqrt{N}$, the times complexity is $N*N/2$, that is $O(N^2)$. This time complexity is equal to the cyclic coding algorithm[8].

C. Space Complexity

For every time running of this LSD algorithm, the space needed is several vectors with n elements and n is lower than $N/2$. So the space complexity of LSD algorithm is $O(N)$. Because the cyclic coding algorithm needs an array with $N*N$ elements, the space complexity of cyclic coding is $O(N^2)$. So the space complexity of LSD algorithm is lower than the cyclic coding algorithm.

D. Property of Congregation

In this algorithm, every node selected must be the node with the longest-shortest distance, so many nodes congregates next to next. Assume $N=22$, the quorum of LSD is 1010001000100100010001, the quorum of cyclic coding is 1101000100001000000010. From the quorum vectors, in the quorum generated by LSD algorithm, all 1 are distributed more dispersal than the quorum generated by the cyclic coding algorithm.

VI. CONCLUSION

The LSD algorithm has shorter quorum length, the average quorum length is $2\sqrt{N}$, is close to the cyclic coding algorithm's quorum length. And the time complexity of LSD algorithm is $O(N^2)$, is close to the cyclic coding algorithm's time complexity, too. But the space complexity of LSD algorithm is $O(N)$, the space complexity of cyclic coding algorithm is $O(N^2)$, LSD algorithm is more excellent than cyclic coding algorithm. At the same time, LSD can resolve the CE problem of the other cyclic coding algorithms.

REFERENCES

- [1] M. Maekawa, "A \sqrt{N} algorithm for mutual exclusion in decentralized systems", ACM Transactions Computer Systems, 1985, vol. 3, Issue 2, pp145-159. <http://dx.doi.org/10.1145/214438.214445>
- [2] Chia-Hung Tsai, Tsu-Wen Hsu, Meng-Shiuan Pan, Yu-Chee and Tseng, Cross-Layer, "Energy-Efficient Design for Supporting

SPECIAL FOCUS PAPER
QUORUM GENERATION ALGORITHM BASED ON THE LONGEST-SHORTEST DISTANCE

- Continuous Queries in Wireless Sensor Networks: A Quorum-Based Approach”, *Wireless Personal Communications*, November 2009, Vol.51, Issue 3, pp 411-426.
- [3] Gunnar F. Kaufmann, Junguk Park, Alexander V. Mayorov, Diane and M. Kubitz, “Generation of Quorum Quenching Antibodies”, *Methods in Molecular Biology*, 2011, Vol.692, pp 299-311.
- [4] Ainul Azila Che Fauzi, Ahmad Noraziah and Noriyani Mohd Zin, “A Novel Binary Vote Assignment Grid Quorum Algorithm for Distributed Database Fragmentation”, *Communications in Computer and Information Science*, 2011, Vol.194, pp 339-349. http://dx.doi.org/10.1007/978-3-642-22603-8_31
- [5] Rohaya Latip, Hamidah Ibrahim, Mohamed Othman, Md Nasir Sulaiman and Azizol Abdullah. “Quorum Based Data Replication in Grid Environment”, *Lecture Notes in Computer Science*, 2008, Vol.5009, pp 379-386.
- [6] Rohaya Latip, Aminah Hidayah and Mohamad Ariff, “Dual Quorum Replication (DQR): A Protocol for Data Replication in Data Grid”, *Communications in Computer and Information Science*, 2010, Vol.118, pp 226-233. http://dx.doi.org/10.1007/978-3-642-17622-7_22
- [7] Maurice P. Herlihy and J. D. Tygar, “How to Make Replicated Data Secure”, *Lecture Notes in Computer Science*, 1988, Vol. 293, pp 379-391.
- [8] LI Mei-An, Liu Xin-Song and Wang Zheng. “High performance distributed mutual exclusion algorithm based on cyclic coding”, *Tien Tzu Hsueh Pao/Acta Electronica Sinica*. 2005, vol.33, Issue 8, pp 1397-1402.
- [9] LI Mei-An, Liu Xin-Song and Wang Zheng, “A high performance distributed mutual exclusion algorithm based on relaxed cyclic difference set”, *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, 2007, vol.35, Issue 1, pp 58-63.
- [10] LI Mei-An, Liu Xin-Song and Wang Zheng, “A symmetric quorum generation for distributed mutual exclusion based on the properties of cyclic quorum”, *Chinese Journal of Electronics*, 2006, vol.15, Issue 1, pp 17-20.

AUTHORS

LI Meian is with College of Computer and Information Engineering, Inner Mongolia Agriculture University, Huhehaote, China (limeian1973@126.com).

WANG Buyu is with College of Computer and Information Engineering, Inner Mongolia Agriculture University, Huhehaote, China (Buyuwang2005@163.com).

GAO Jing is with College of Computer and Information Engineering, Inner Mongolia Agriculture University, Huhehaote, China (Gaojing@imau.edu.cn).

This article is an extended and modified version of a paper presented at the 2012 International Conference on Artificial Intelligence and Its Application in Industry Production (AIAIP 2012), held in Wuhan, China in December 2012. Manuscript received 09 May 2013. Published as resubmitted by the authors 26 June 2013.