Quorum Set Systems for Generalized Resource Allocation Problem

DWILYA MAKIWAN\(^1\)  ARMIN LAW\(^2\)

Abstract: Mutual exclusion (mutex) is fundamental problems in distributed systems in which a resource must be used at most one process at a time when some processes were communicating each other to compete in accessing. Many resource allocations problems have been constructed base on the mutual exclusion problem, and it can be generally formed to problems called \((n,m,k,d)\)-resource allocation and further extended to \((n,m,k,d)\)-resource allocation, i.e, the problem to synchronize accesses on \(k\) identical resources among \(n\) processes which belong to \(m\) groups and each resource can be used by at most \(d\) processes of the same group at a time. The generalized resource allocation problem can be solved using a quorum system called \((m,k,d)\)-coterie and \((m,k,d)\)-coterie that satisfies the relaxed safety property of the original mutual exclusion. If quorum based algorithm of mutual exclusion directly adopt to the quorum systems then the massage complexity is the same and the maximum degree of concurrency of the generalized resource allocation problem of \(\sum_{i=1}^{n} d_{i}\) is achieved.

Keywords: Mutual exclusion, quorum systems, resource allocation, coteries.

1. Introduction

Algorithm for the mutual exclusion problem must dealing with messages delay; therefore it needs algorithm based on set systems called quorum-based algorithm. The synchronization algorithm based on quorum system is a robust approach since it is significantly tolerated the failures of nodes and communications that can lead to network partitioning. Every two quorums must satisfy intersection property and each node has only one permission to issue. In a quorum system, if a process request and receives permission from all nodes in a selected set, then it can enter the critical section. Therefore, it guarantees the safety condition in which at most one process is on critical section at a time. It must guarantee the satisfying of safety conditions, i.e., at most one process accessing resource at the same time, and liveness properties, i.e., a process requesting a resource will eventually succeed [2,4,6,7].

There have been proposed a new solution for \(h\)-out of \(k\)-Mutual Exclusion and it gave us a new comprehension about a quorum system called \((m,h,k)\)-coterie [7], and further it has been expanded to solving \((n,m,k,d)\)-resource allocation problem [4]. Basically, this paper presents the construction of \((m,k,d)\)-coterie and \((m,k,d)\)-coterie that has been studied in [3,6,7] with some addition to make the construction more robust and satisfy the required properties of the generalized resource allocation problem which is the safety property is relaxed from the original mutual exclusion problem.

2. Related Work

Mutual exclusion aims to synchronizing all the access of processes in distributed systems into single indivisible resource [9]. One of algorithm that can solve the problem of processes communication is algorithm introduced by Lamport [5], when a process wants to access resources, it has to had permission from another process on the system by comparing smallest time stamp [9]. As follow to ensure safety, for mutual exclusion every two quorum (subsets of processes collection) must intersect, this helps increase fault tolerance because if a quorum is hit due to that some member in the quorum has failed, some other quorum may still be available to grant access of resource. A simple approach is processes compete for quorums as usual, but only one quorum can be acquired at a time. After a process has successfully acquired a quorum, it “helps” other processes of the same group to enter the critical section directly without being blocked by the quorums they are waiting for [3].

3. Set Systems for Resource Allocation Problem

To learn about quorum set systems, give a basic definition of coterie.

Definition 1 Let \(P\) be a finite set of elements where \(C \subseteq 2^P\) is nonempty set of \(P\), called coterie under \(P\) if and only if:

\[
\text{Minimality: } (\forall Q, Q' \in C) \{Q \subsetneq Q'\}, (Q, Q' \in C \text{ is quorum})
\]

\[
\text{Intersection: } (\forall Q, Q' \in C) [Q \cap Q' \neq \emptyset].
\]

Example 1

Let \(C_1 = \{\{1,2\}, \{2,3\}\}\), \(C_2 = \{\{2,4\}, \{3,4\}, \{1,4\}\}\) coterie under \(P = \{1,2,3,4\}\). So that all \(\{1,2\}, \{2,3\}, \{2,4\}, \{3,4\}\) and \(\{1,4\}\) are quorums.

For every set system is said to be solution of resource allocation problem if the following conditions hold:

Safety : There is only one process can access the resource at any time.

Liveness : a process requesting a resource will eventually succeed.

3.1 Quorum system for \((n,m,k,d)\)-resources allocation

\((n,m,k,d)\)-resources allocation problem concerns the scheduling of \(k\) identical resources among \(n\) processes which belong to \(m\) groups. Each resource can be used by at most \(d\) processes of the same time.

3.1.1 \((m,k,d)\)-coterie

Definition 2 \((m,k,d)\)-coterie A collection of sets \(M=(C_1, \ldots, C_n)\)
where $C_i$ is a $d$-coterie under $P$, $\forall C_i \in M$ is a $(m,k,d)$-coterie if following conditions hold:

**Disjoint:** $\forall \ell < m$ there exists $\exists C \in M$ such that $C$ and $C_i$ are disjoint $\forall 1 \leq i \leq \ell$

**Bicoterie:** $\forall K = (C_i', ..., C_j') \in M$ for any $\ell < m$, there exists pair of $(C_i', C_j')$ forms bicoterie $\forall 1 \leq i \leq j \leq \ell$.

### 3.1.2 Construction of $(m,k,d)$-coterie

Assume $n = 2kd^2$, $m = 2k$, partition $P$ into $m$ subsets of $P_1, ..., P_m$. Create $d$-coterie $C_i$ on each set $P_i$ by construct $d$ disjoint sets (quorums) $Q_{ij}$, $1 \leq i \leq m, 1 \leq j \leq n$.

Hence,

$$P = \bigcup P_i$ and $P_i \cap P_j = 0, \forall 1 \leq i, j \leq m$$

(1)

$$Q_{ij} = \{p_{ij}^d | p_{ij}^d \in P_i, 1 \leq d \leq n\}$$

(2)

$$Q_{(i+1)j} = \{p_{(i+1)j}^d | p_{(i+1)j}^d \in P_{i+1}, 1 \leq d \leq n\}$$

(3)

$$C_i = \{Q_{ij} \cup Q_{(i+1)j} \mod m \}$$

(4)

with, $|Q_{ij}| = 2d$ if $|Q_{ij}| = d$, $|Q_{ij}| = 2d + 1$ if $|Q_{ij}| = d + 1$, and $|Q_{ij}| = 2d + n$ if $|Q_{ij}| = d + n$.

Let $M = (C_1, ..., C_m)$ with $C_i = \{Q_{ij} | 1 \leq i \leq m\}$.

**Lemma 1.** $C_i$ is d-coterie under $(P_i \cup P_{i+1})$ for $1 \leq i \leq m$.

**Proof.** $P_1, ..., P_m \in P$ with condition of (1), $C_i \in P_i$ and $C_{i+1} \in P_{i+1}$. It will shows $C_i$ is $d$-coterie under $(P_i \cup P_{i+1})$

As we know $C_i \in P_i$ and $C_{i+1} \in P_{i+1}$, (2) and (3) represent

$$Q_{ij} = \{p_{ij}^d | p_{ij}^d \in P_i, 1 \leq d \leq n\}$$

(5)

$$Q_{(i+1)j} = \{p_{(i+1)j}^d | p_{(i+1)j}^d \in P_{i+1}, 1 \leq d \leq n\}$$

(6)

$$C_i = \{Q_{ij} \cup Q_{(i+1)j} \mod m \}$$

(7)

So that, $C_i \cap Q_{ij}$ and $C_{i+1} \cap Q_{(i+1)j}$ are disjoint sets and because from (4) $C_i = Q_{ij} \cap Q_{(i+1)j} \mod m$ we can see $C_i$ is a disjoint sets with $Q_{ij} \in P_i \cup P_{i+1}$.

**Lemma 2.** $(C_1, C_{i+1})$ is a bicoterie $\forall 1 \leq i \leq m - 1$ and $(C_i, C_{i+2})$ is a disjoint $\forall 1 \leq i \leq m - 2$.

**Proof.** By partitioning $P$ into $P_1, ..., P_m$, and using (1) appear collections $C_i \in P_i$, $C_{i+1} \in P_{i+1}$, $C_{i+2} \in P_{i+2}$. It shows $(C_i, C_{i+1})$ is a bicoterie and $(C_i, C_{i+2})$ is a disjoint.

From (2) and (3) we directly construct

$$C_i = Q_{ij} = \{p_{ij}^d | p_{ij}^d \in P_i, 1 \leq d \leq n\}$$

(8)

$$C_{i+1} = Q_{(i+1)j} = \{p_{(i+1)j}^d | p_{(i+1)j}^d \in P_{i+1}, 1 \leq d \leq n\}$$

(9)

$$C_{i+2} = Q_{(i+2)j} = \{p_{(i+2)j}^d | p_{(i+2)j}^d \in P_{i+2}, 1 \leq d \leq n\}$$

(10)

By joining these elements according to construction on (4) shows

$$C_i = Q_{ij} = Q_{ij} \cup Q_{(i+1)j} \mod m$$

(11)

$$C_{i+1} = Q_{(i+1)j} = Q_{(i+1)j} \cup Q_{(i+2)j} \mod m$$

(12)

$$C_{i+2} = Q_{(i+2)j} = Q_{(i+2)j} \cup Q_{(i+3)j} \mod m$$

(13)

**Theorem 1.** $M$ is $\bigcup \{n \mod \frac{m}{T} \}$-coterie, with size of each coterie is $d = \frac{n}{m}$ the size of each quorum is $Q_{ij} = 2 \frac{n}{m}$.

For more general with $m = tk, k = \frac{m}{t}$, and $t = 1, 2, ..., \lfloor \frac{m}{k} \rfloor$ construction can be done by a similar procedure. Partition set $P$ into $P_i$ and create $d$-coterie by create $d$ disjoint sets or quorum

$$C_i = \{Q_{ij} | 1 \leq i \leq m, 1 \leq j \leq n\}$$

(14)

$$Q_{ij} = \{p_{ij}^d | p_{ij}^d \in P_i, 1 \leq d \leq k \}$$

(15)

$$Q_{(i+1)j} = \{p_{(i+1)j}^d | p_{(i+1)j}^d \in P_{i+1}, 1 \leq d \leq k \}$$

(16)

$$C_i = \{Q_{ij} \cup Q_{(i+1)j} \mod m \}$$

(17)

Let coterie $C_i = Q_{ij} \cup Q_{(i+1)j} \mod m$, then

$$Q_{ij} = Q_{ij} \cup (U_{1 \leq j \leq d} \{p_{ij}^d \in P_{i+k}, 1 \leq d \leq k \})$$

(18)

1. $|Q_{ij}| = td$ if $|Q_{ij}| = d$, $|Q_{ij}| = td + 1$ if $|Q_{ij}| = d + 1$, and $|Q_{ij}| = td + n$ if $|Q_{ij}| = d + n$.

2. $Q_{ij} \cap Q_{(i+1)j} = \emptyset$, $1 \leq j \leq d$

3. $Q_{ij} \cap Q_{(i+1)j} = \emptyset$, $1 \leq j \leq t$ and $1 \leq i \leq j \leq d$

By the definition, $M = \{C_1, ..., C_m\}$ is coterie.

**Theorem 2.** $M$ is a $\bigcup \{n \mod \frac{m}{T} \}$-coterie with $1 \leq \frac{m}{T}$. $\bigcup \{n \mod \frac{m}{T} \}$-coterie under $P$.

**Proof.** $M = \{C_1, ..., C_m\}$ and it shows $M$ is $\bigcup \{n \mod \frac{m}{T} \}$-coterie under $P$.

Each pair $C_i$ and $C_j$ are disjoint because all element on each pair set was not same. But for pair sets of $C_i$ and $C_j$, $C_i$ and $C_j$ are bicoterie because each element was intersected with their pair sets.
3.2 Quorum system for \((n,m,k,d)\)-resource allocation

For resource allocation problems with at most \(d\) (different degree) processes at the same time, it will be solved by using quorum system called \((m,k,d)\)-coterie.

3.2.1 \((m,k,d)\)-coterie.

Definition 3 \((m,k,d)\)-coterie. A collection set \(M = (C_1, ..., C_m)\). And \(C_i\) is \(d\)-coterie under \(P\), for every \(1 \leq i \leq n\), called \((m,k,d)\)-coterie under \(P\) if disjoint and bicoterie hold.

By using this definition, we definitely can construct some steps to build \((m,k,d)\)-coterie to solve the \((n,m,k,d)\)-resource allocation.

3.2.2 Construction of \((m,k,d)\)-coterie

\((m,k,d)\)-coterie can be constructed by using the modification of \((m,d)\)-coterie by assuming \(m = tk\) and \(n = tk \sum_{i=1}^{m} d_i \mod m\). We first partition \(P\) into \(P_i\) to create coterie \(d_i\), where \(d_i \leq Q_{ij} \), \(C_i = \{Q_{ij}\}\) ................. (9)

\[Q_{ij}(\mod 1) = \{p_{ij}^{*+1} \in P_{i+1}^{*}, 1 \leq s \leq d \} \] ............. (10)

with \(Q_{ij}(\mod 1) = \{d_{ij}\}

\[C_i = Q_{ij} \cup Q_{ij+1(\mod d)} \] \mod m

\[= Q_{ij} \cup (U_{1 \leq s \leq d} \{p_{ij}^{*+1} \in P_{i+1}^{*}, 1 \leq s \leq d \}) \mod m \] ............. (11)

Finally, we have

\[Q_{ij} \cap Q_{ij+1} = \emptyset \] 

\[Q_{ij} \cap Q_{ij+1} = \emptyset \] 

\[Q_{ij} \cap Q_{ij+1} = \emptyset \]  

And \(M = (C_1, ..., C_m)\) is coterie.

Theorem 3 \(M = \left(\frac{m}{t}, ..., \frac{m}{t}\right)\)-coterie with \(t = 1,2,3, ..., \left\lfloor \frac{m}{t} \right\rfloor\).

Proof. Let \(M = (C_1, ..., C_m)\). It shows \(M = \left(\frac{m}{t}, ..., \frac{m}{t}\right)\)-coterie under \(P\). By using the construction (11)

\[C_i = Q_{ij} \cup Q_{ij+1(\mod d)} \] \mod m

\[Q_{ij} = Q_{ij} \cup (U_{1 \leq s \leq d} \{p_{ij}^{*+1} \in P_{i+1}^{*}, 1 \leq s \leq d \}) \mod m \]

by bring us into result on (4) and (5) with satisfying of disjoint for \(Q_{ij} \cap Q_{ij+1} = \emptyset\) and bicoterie for \(Q_{ij} \cap Q_{ij+1} = \emptyset\) and it directly shows \(M = \left(\frac{m}{t}, ..., \frac{m}{t}\right)\)-coterie. 

Example 3.

With \(m = 2k\), \(m = 4\), \(k = 2\) and \(d = \{2,3,3,2\}\) we will construct \((m,k,d)\)-coterie. Partition \(P\) into 10 subsets each quorum. Sets it up into 4 different coteries

\[C_1 = \{1,2,5,7,9\}, \{3,6,8,10\}\]

\[C_2 = \{5,6,11,14,17\}, \{7,8,12,15,18\}\]

\[C_3 = \{11,12,13,20,23\}, \{14,15,16,21,24\}\]

\[C_4 = \{1,3,20,21,22\}, \{2,4,23,24,25\}\]

Each pair \(C_1\) and \(C_3\) \(C_2\) and \(C_4\) are disjoint because all element on each pair was not set. But for pair sets of \(C_1\) and \(C_3\), \(C_2\) and \(C_4\) and \(C_i\) are bicoterie because each element was intersected with their pair sets.

Now let more understand about the communication using the \((m,k,d)\)-coterie by having the algorithm below.

**Trying Section**  
// When \(p_i\) wishes to access a resource \(r_i\)  
1: \(ts_i++\); // \(ts_i\) is current logical time  
2: Select a quorum \(Q\) in \(C_i\);  
3: send \(req(ts_i, p_i)\) to \(p_j\), \(v \in \emptyset\);  
4: Add \(p_j(\in Q)\) answering ack into AGREE_;  
5: if (there is a \(Q(\in C_i)\) \(\leq\) AGREE_){    
6: \(state := Critical\ Section;\)

7: else-if { // If there exists \(p_j(\in Q)\) answers wait  
8: Add \(p_j\) answering wait into DISAGREE_;  
9: Select another quorum \(Q^\prime \in C_i\), such that \(Q \cap DISAGREE_ = \emptyset \land Q = max\{|Q \cap AGREE_\}|\);  
10: if (there is no quorum satisfy) {  
11: \(state := Wait;\)

12: \(Q := (Q^\prime \cap Q)\) and go to line 3; }

**Exit Section**  
// When user \(p_i\) leaves resource \(r_i\)  
1: send exit to \(v \in\) AGREE_ \(\cup\) DISAGREE_;

When \(p_i\) receives \(req(ts_j, p_j\) message:

1: if (PERM_ = \(\emptyset\)) {  
2: send ack to \(p_j\) and add (\(ts_j, p_j\) \(\to\) PERM_);  
3: else-if { // If there exists \(ts_j, p_j\) \(\in\) PERM_  
4: Let \((ts_j, p_j)\) is the highest priority in QUEUE_;  
5: Insert req\((ts_j, p_j)\) into QUEUE_;  
6: if \((ts_j, p_j) > max(ts_r, p_r, (ts_r, p_r))\) {   
7: send reclaim to \(p_r\)\;  
8: else-if {  
9: send wait to \(p_r\); }

When \(p_i\) receives exit message from \(p_j\):

1: Remove request \(p_j\) from PERM_;  
2: if (QUEUE_ \(\neq\) \(\emptyset\)) {  
3: // Let \((ts_j, p_j)\) is the highest priority in QUEUE_;  
4: Move req\((ts_j, p_j)\) from QUEUE_ to PERM_;  
5: send ack to \(p_j\); }

When \(p_i\) receives reclaim message from \(p_j\):

1: if \((p_i\ not\ in\ critical\ section\ and\ p_i(\in\ AGREE_)\) {  
2: Move \(p_i\) from AGREE_ to DISAGREE_;  
3: send relinquish to \(p_j\); }

When \(p_i\) receives relinquish message from \(p_j\):

1: // Let \((ts_r, p_r)\) is the highest priority in QUEUE_;  
2: if \((ts_r, p_r) > (ts_j, p_j)\) {  
3: Move req\((ts_j, p_j)\) from PERM_ to QUEUE_;  
4: send ack to \(p_j\);  
5: Move req\((ts_r, p_r)\) from QUEUE_ to PERM_;}

Fig. 1. The \((m, k, d)\)-coterie based algorithm
Theorem 4. The algorithm in Fig. 1 adopting \((m, h, k)\)-coteries achieves the maximum degree of concurrency.

The \((n,m,k,d)\)-resource allocation problem can be solved using quorum system called \((m,k,d)\)-coterie since \(d\)-mutual exclusion with \(d\)-access together satisfying of disjoint and bicoterie conditions. This problem also free from deadlock and starvation because the execution of resources at the end will be succeed since it adopts Lamport algorithm directly with timestamp method \((ts, \ p)\). The same stands for \((n,m,k,d)\)-resources allocation with quorum system called \((m,k,d)\)-coterie.

4. Conclusion

A \((m,k,d)\)-coterie by partitioning universe set into some disjoint \(d\)-coterie, may be one way to resolve \((n,m,k,d)\)-resource allocation problem with each size of new quorum is \(|Q'_i| = td\) and its depend on the number of elements that inserted from previous quorum \(Q_{i+1}\). The construction of \((m,k,d)\)-coterie leads us to solution of \((n,m,k,d)\)-resource allocation with capacity of \(d\) resources is different, both of it guarantee safety and liveness by using timestamp algorithm. If mutex system straight adopted into quorum system that is \((m,k,d)\)-coterie, then the massages complexity is \(\sum_{i=1}^{d} d_i\).

References