Legion Structure for Quorum-Based Location Management in Mobile Computings

Ming-Jeng Yang, Yao-Ming Yeh, and Yao-Ming Chang
Department of Information & Computer Education
National Taiwan Normal University
Abstract

- We propose a theory of *Legion structure* that can be used to construct some schemes of distributed applications, such as location management, mutual exclusion, etc. We also present a new and simple distributed quorum-based location management scheme – *LegRing*, which is developed from the theory of *Legion* structure. With small quorum size and symmetric property, the *LegRing* scheme can be extended to a fault tolerant and load balanced location management algorithm.
1. Introduction

- One mutual exclusion (1-Mutex; K-Mutex) -- Concurrent access to one (or K) shared resource of the critical section (CS) must be synchronized such that at any time only one (or K) process can access the CS.
  - Permission-Based Approach: Quorum
  - Token-Based Approach

- Data Replication — Data replicated among distributed processes should be monitored to make the data consistent
  - Primary Site Approach:
  - Majority Voting:
  - Quorum: Write-Quorum, Read-Quorum
Critical Section:
Permission-Based Approach: Quorum

- Given that one resource is shared by 5 processes
- To get the resource:
  - one should get the permissions by other 4 processes
  - Quorum-based Set System:
    \[ Q_1 = \{0,1,2\}, Q_2 = \{1,2,3\}, Q_3 = \{2,3,4\}, Q_4 = \{3,4,0\}, Q_5 = \{4,0,1\} \]
  - one need only to get the permissions by other 2 processors
Data Replication: Quorum-based

- Given that data are replicated among 9 processes
  - To Write Data: Need to write to 9 processes
  - To Read Data: Can read from any process

- Majority Voting
  - To Write Data: Need to write to 5 processes
  - To Read Data: Need to read from 5 processes

- Quorum-based
  - To Write Data: Need to write to 5 processes
  - To Read Data: Need to read from 3 processes
1. Introduction-Related Work

- In Prakash’s paper [5], a dynamic load-balanced location management scheme with an iterative and grid-based quorum construction is proposed. $N$ location servers are divided into quorums of cardinality of $0.97N^{0.63}$ and $2\sqrt{N} - 1$ respectively.

- Ihn-Han Bae in [6] proposed a distributed location management scheme using the quorum, which is based on the triangle configuration of location servers.
2 The System Model

Fig. 1. Logical view of a mobile network with distributed location database.
3.1 Quorum, Set System, and $K$-Coterie

**DEFINITION 1.** A set system [7] $C = \{Q_1, Q_2, \ldots, Q_n\}$ is a collection of nonempty subsets $Q_i \subseteq U$ of a finite universe $U$.

*Example:*

$C = \{Q_1, Q_2, \ldots, Q_n\} = \{\{1,2,3\}, \{2,3,4\}, \ldots, \{9,1,2\}\}$
DEFINITION 2. A \textit{k-coterie} \cite{8} is a nonempty set system that has the following properties:

[I] \textbf{Nonintersection property:} Given any \( h \) (\( h < k \)) elements \( Q_1, Q_2, \ldots, Q_h \in C \) such that \( Q_i \cap Q_j = \emptyset \) (\( i \neq j, 1 \leq i, j \leq h \)), there exists another element \( Q \in C \) such that \( Q \cap Q_t = \emptyset \), (\( 1 \leq t \leq h \)).

[II] \textbf{Intersection Property:} Among any \( k+1 \) elements \( Q_1, Q_2, \ldots, Q_{k+1} \in C \), there exist at least two elements \( Q_i \) and \( Q_j \) (\( i \neq j, 1 \leq i, j \leq k+1 \)), such that \( Q_i \cap Q_j \neq \emptyset \).

[III] \textbf{Minimality property:} For any pair of distinct elements \( Q_i, Q_j \in C \), there doesn’t exist \( Q_i \subset Q_j \).

Each element \( Q \) of \( C \) in definition 1 and 2 is called a \textit{quorum}. 
3.1 Quorum, Set System, and $K$-Coterie (Cont.)

$Ex: C = \{Q1, Q2, ..., Qn\} = \{\{0,1\}, \{1,2\}, \{2,3\}, \{3,4\}, \{4,5\}, \{5,6\}, \{6,0\}\} \text{ is a 3-coterie.}(k = 3)$. 
DEFINITION 3. A **Legion** $\text{Leg}(k_1,k_2,\ldots,k_m)$

$$\equiv\{C_1,C_2,\ldots,C_m\},$$
where $1\leq m$, $k_i\in\{\text{null},1,2,\ldots,n\}$, is a collection of set systems that has the following properties:

[I] $C_i \equiv \{Q_1, Q_2, \ldots, Q_n\}$ is a $k_i$-coterie, if $1\leq k_i \leq n$.

[II] $C_i \equiv \{Q_1, Q_2, \ldots, Q_n\}$ is a set system and could be a $k$-coterie ($1\leq k \leq n$) or not (i.e., don’t care), if $k_i=$null.

[III] For any pair of quorums $Q_s \in C_i$ and $Q_t \in C_j$, there is $Q_s \cap Q_t \neq \emptyset$, where $C_i$ and $C_j$ are different set systems (i.e., $i \neq j$, $1\leq i,j \leq m$).
The Structure of Legion

- Example: $Leg(k_1, k_2, \ldots, k_m)$
  $$\equiv \{C_1, C_2, \ldots, C_m\}$$
  $$\equiv \{\{Q_1, Q_2, \ldots, Q_n\}, \{Q_1, Q_2, \ldots, Q_n\}, \ldots, \{Q_1, Q_2, \ldots, Q_n\}\}$$
3.3 The Applications of Legion Structure

- \( \text{Leg}(k_1,k_2,\ldots,k_m) \equiv \{C_1,C_2,\ldots,C_m\} \)
- **Example**: \( \text{Leg}(1) \equiv \{C_1\} \) \((m=1)\)
  \[ \{\{1,2,4\},\{2,3,5\},\{3,4,6\},\{4,5,7\},\{5,6,1\},\{6,7,2\},\{7,1,3\}\} \]

- **Application**: one mutual exclusion algorithm in distributed system
3.3 The Applications of Legion Structure (Cont.)

- $\text{Leg}(k_1, k_2, \ldots, k_m) \equiv \{C_1, C_2, \ldots, C_m\}$
- **Example:** $\text{Leg}(k) \equiv \{C_1\}$ \hspace{1cm} ($m=1$)
  
  $\{\{0,1\}, \{1,2\}, \{2,3\}, \{3,4\}, \{4,5\}, \{5,6\}, \{6,0\}\}$ is a 3-coterie. ($k_i = 3$).

- **Application:** $k$ mutual exclusion algorithm in distributed system
3.3 The Applications of Legion Structure (Cont.)

- \( \text{Leg}(k_1,k_2,\ldots,k_m) \equiv \{C_1,C_2,\ldots,C_m\} \)

- **Example:** \( \text{Leg}(1,\text{null}) \equiv \{C_1,C_2\} \) (m=2)

\[
\{\{1,4,7,8,9\},\{2,5,8,9,1\},\{3,6,9,1,2\},\{4,7,1,2,3\},\{5,8,2,3,4\},\{6,9,3,4,5\},
\{7,1,4,5,6\},\{8,2,5,6,7\},\{9,3,6,7,8\}\},
\{\{1,2,3\},\{2,3,4\},\{3,4,5\},\{4,5,6\},\{5,6,7\},\{6,7,8\},\{7,8,9\},\{8,9,1\},\{9,1,2\}\}\]

- **Application:** replica control in distributed database systems.
### 3.3 The Applications of Legion Structure (Cont.)

- **Leg**\((k_1,k_2,…,k_m) ≡ \{C_1,C_2,…,C_m\}\)
- **Example:** \(\text{Leg}(\text{null},\text{null}) ≡ \{C_1,C_2\}\) (\(m=2\))
  \[
  \{\{0,1,2\},\{1,2,3\},\{2,3,4\},\{3,4,5\},\{4,5,6\},\{5,6,7\},\{6,7,8\},\{7,8,0\},\{8,0,1\}\},
  \{\{0,3,6\},\{1,4,7\},\{2,5,8\},\{3,6,0\},\{4,7,1\},\{5,8,2\},\{6,0,3\},\{7,1,4\},\{8,2,5\}\}\]
- **Application:** location management in mobile computing.
Distributed Applications in Legion Structure

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Location Management</th>
<th>Data Replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Leg(Null,Null)</td>
<td>Leg(1,Null)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Leg(1)</td>
<td>Leg(k)</td>
</tr>
</tbody>
</table>

Set System Type

Null 1 K
4 A Location Management Scheme from $\text{Leg}(\text{null}, \text{null})$

DEFINITION 4. In an $N$-LegRing system, the Update-set system ($U$-set) and Query-set system ($Q$-set) are defined as follow:

\[
U\text{-set} = \{\{n, (n+1) \mod N, (n+2) \mod N, \ldots, (n+d-1) \mod N\} | 0 \leq n \leq N-1\} \left\lfloor (N-1)/d \right\rfloor \left\lceil \sqrt{N} \right\rceil
\]

\[
Q\text{-set} = \{\{n, (n+d) \mod N, (n+2d) \mod N, \ldots, (n+kd) \mod N\} | 0 \leq n \leq N-1, k = (N-1)/d\}
\]

here $d = \sqrt{N}$ ; $n$, $k$, and $N$ are all integers.

each element of $U$-set and $Q$-set is called an $U$-quorum and a $Q$-quorum, respectively.
4.1 A Simple Scheme with Quorum Size $\sqrt{n}$

U-set = \{ 
\{0,1,2\}, \\
\{1,2,3\}, \\
\{2,3,4\}, \\
\{3,4,5\}, \\
\{4,5,6\}, \\
\{5,6,7\}, \\
\{6,7,8\}, \\
\{7,8,0\}, \\
\{8,0,1\} 
\}

Q-set = \{ 
\{0,3,6\}, \\
\{1,4,7\}, \\
\{2,5,8\}, \\
\{3,6,0\}, \\
\{4,7,1\}, \\
\{5,8,2\}, \\
\{6,0,3\}, \\
\{7,1,4\}, \\
\{8,2,5\} 
\}

Update Set
Query Set

$N = 9$, $\sqrt{N} = 3$; $N = 21$, $\sqrt{N} = 5$
Quorum-Based location service

U-set = \{\{0,1,2\},
        \{1,2,3\},
        \{2,3,4\},
        \{3,4,5\},
        \{4,5,6\},
        \{5,6,7\},
        \{6,7,8\},
        \{7,8,0\},
        \{8,0,1\}\}\,

Q-set = \{\{0,3,6\},
        \{1,4,7\},
        \{2,5,8\},
        \{3,6,0\},
        \{4,7,1\},
        \{5,8,2\},
        \{6,0,3\},
        \{7,1,4\},
        \{8,2,5\}\}\,
4.2 Properties

THEOREM 1. The U-set and Q-set of an $N$-LegRing system defined in definition 4 satisfy the properties of Leg(null, null).

THEOREM 2. The size of U-quorum defined in definition 4 is $\lceil \sqrt{N} \rceil$ and the size of Q-quorum is $\lfloor \sqrt{N} \rfloor$ or $\lceil \sqrt{N} \rceil$.

THEOREM 3. The U-set and Q-set of an $N$-LegRing system defined in definition 4 satisfy the properties of Symmetric Set System (SSS).
# 5. Comparison

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Quorum Size</th>
<th>Symmetric</th>
<th>Load Balance</th>
<th>Fault Tolerance</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle configuration [Bae6]</td>
<td>$\sqrt{2N}$</td>
<td>No</td>
<td>No</td>
<td>Yes/Partial</td>
<td>$n(n+1)/2$</td>
</tr>
<tr>
<td>Dynamic hashing + Grid based scheme [Prakash5]</td>
<td>$2\sqrt{N} - 1$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>$m*n$</td>
</tr>
<tr>
<td>Dynamic hashing + Iterative approach [Prakash5]</td>
<td>$0.97N^{0.63}$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>$n$</td>
</tr>
<tr>
<td>Random + LegRing scheme</td>
<td>$\sqrt{N}$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>$n$</td>
</tr>
</tbody>
</table>