CS425

Spring 2015

MP3

Maekawa's Algorithm

Hao Jin

haojin2

1 Introduction

Maekawa's Algorithm is a kind of algorithm for mutual-exclusion without any leader or central server control. Each node in the network has voting set, from which they ask permission to enter the critical section.

2 Voting Set Construction

1	2	3
4	5	6
7	8	9

The above table denotes a matrix whose elements are node identifiers in the network. Every node's voting set includes it self and all the elements on the row that it lies on and all the elements on the column that it lies on. As a result:

node id	voting set
node 1	$\{1, 2, 3, 4, 7\}$
node 2	$\{2, 1, 3, 5, 8\}$
node 3	$\{3, 1, 2, 6, 9\}$
node 4	$\{4, 1, 5, 6, 7\}$
node 5	$\{5, 2, 4, 6, 8\}$
node 6	$\{6, 3, 4, 5, 9\}$
node 7	$\{7, 1, 4, 8, 9\}$
node 8	$\{8, 2, 5, 7, 9\}$
node 9	$\{9, 3, 6, 7, 8\}$

3 Handling Deadlocks

The problem is that the original Maekawa's Algorithm is deadlock-prune. One mechanism to totally eliminate the possibility of deadlock is to maintain the total ordering of request messages so that no pair of nodes who both want to enter critical section wait for permission from each other.

I maintained the total ordering through this way: The communication between nodes is by shared memory. Each node has a FIFO queue from the std library, which simulates the communication channel pointed to each one of them. All queues are protected by a mutex lock, to modify any of the queue one thread has to grab the single lock. Since all queues are locked at the same time, say we have two nodes who both want to multicast REQUEST message, one of the node grabs the lock first, then it pushes message onto all the channels of its voting set, then the other grabs the lock and pushes message onto all the channels of its voting set, for the channels of the overlap between these two voting sets, the messages are totally ordered, thus the deadlock is totally avoided through totally ordered multicast.

4 Compilation and Execution

1. unzip and move all source files into one folder, including the Makefile

2. type "make" then push enter in the terminal

3. type "./mutex ⟨cs_int⟩ ⟨next_req⟩ ⟨total_exec_time⟩ ⟨option⟩" and hit enter to run!