

Distributed Calendar using Paxos

CS271 – Fall 2008

Introduction to Distributed Systems

Specification for Project 2

Due Date: 4th December

NO LATE TURNINS

Description

In Project 1, you had implemented Wu and Bernstein's Algorithm for maintaining a replicated log. This was an asynchronous protocol for maintaining the replicated log where communication was initiated due to causality. The advantage was lesser communication overhead. The disadvantage was that it resulted in conflicts and as a result, you had to design a conflict resolution protocol. In this project, you will implement a protocol to maintain the replicated log through consensus. In class we discussed the Paxos protocol for distributed consensus. In this project, you are required to implement the Paxos algorithm for maintaining the distributed log. The Distributed Calendar application is built on top of the distributed log similar to the way in which it was designed for Project 1.

The distributed system consists of entities or nodes which want to maintain a local calendar as well as have a view of the global calendar. Let us consider a system where nodes are numbered $1 \dots N$, and each node has an associated user (e.g., $User_i$ is associated with $Node_i$). If $User_i$ wants to set up an appointment with $User_j$ ($i \neq j$), then $User_i$ looks at the calendars of both users at $Node_i$, finds out an empty slot, and then puts the appointment in the *log*. Make note that since all the nodes have a mutually consistent view of the log, all calendars at all nodes are up-to-date. Therefore there are no conflicts in this implementation. The same idea can be extended for setting up meetings with more than 2 users.

Partial Credit for Partial Implementation

It must be noted that the Paxos protocol consists of two different major phases. The first phase corresponds to selection of a leader. The second phase consists of an atomic broadcast initiated by the leader. The complete protocol implementation should implement both these phases. We suggest starting by assuming a resilient (non failing) leader, thus avoiding the first phase of Paxos. Make sure this runs, then add the first phase for fault-tolerance. If you finish the main component, and the system, that is 75% of the grade. The remaining 25% will be given for ensuring that the leader is fault-tolerant, and the first phase of Paxos.

Implementation Details

To start with, implement this system with a system of five (5) distributed nodes. The *log* should be persistent (stored in disk to guarantee that node crashes can be tolerated). Your implementation need not have a good GUI, but a minimal UI to effectively view, insert, delete or modify events should be provided. There is no preference for language used for implementation. Your implementation should be demo-able in GSL or CSIL machines.

For starters, Java provides a good networking library for implementing communication over the network (java.net). You can rely on that library for network communication. This is just a guideline, not a mandate.

Deliverables

Due date for the project is 4th December. On 4th December, turn-in a 2 page specification of the project providing detailed design and implementation. We are planning demos on 4th and 5th, so the TURNIN deadline is 5 pm on 4th or along with your demo, whichever is earlier. If your demo is after 4th, then email the specifications to sudipto@cs.ucsb.edu (PDF files only). Along with the specification, you need to provide experimental evaluation of your implementation in terms of the amount of bandwidth consumed by this protocol for a specified workload (please refer to the workload 1 which was used with project 1. The workload had a number of operations that would have resulted in a conflict in the design for project 1, such a conflict scenario is non-existent in this project implementation. Therefore replace those conflict events with NOP). The experimental details should include the number of messages sent from each site, and the size of the messages (thereby giving an estimate of the bandwidth consumed). The bandwidth details should be generated similar to Project 1 and use workload specification 1 as directed. You should be ready to demo by 4th December. The date and times for the demo will be notified at a later date. At the time of demonstration, you should have a UI for displaying the calendar of all 5 nodes, as well as the bandwidth consumed at each node.