C S 5 5 0 A D V A N C E D O P E R A T I N G S Y S T E M S

PROJECT PROPOSAL

CLOCK SYNCRONIZATION IN DISTRIBUTED SYSTEM

Vinit Kumar Kamboj(A20413470)
Ritika Kumari(A20414073)
MS Computer Science

INTRODUCTION

Background

The temporal ordering of the events produced by concurrent process deals by Clock synchronization. Clock synchronization is useful synchronizing for senders and receivers' messages. The goal of clock synchronization is that multiple unrelated processes running on different machines should agree with and be able to make consistent decisions about the ordering of events in a system. To overcome the most important problem that is in a distributed system, the different nodes maintain their own time using local clocks and their time values may not be same for different nodes, that is there is no global clock within the system so that the various activities in the distributed environment should be synchronized.

There are various methods of achieving clock synchronization depending on the requirements of the situation. To behave as a single, unified computing resource, distributed systems have need for a synchronization of clocks and several algorithms have been proposed on this topic. Various clock synchronization algorithms and their advantages and disadvantages. Proposed solution for overriding the disadvantages of the clock synchronization algorithms

Motivation

With very few exceptions, almost all works on clock synchronization considered static networks. Indeed, the construction of efficient clock synchronization algorithms for dynamic networks is considered as a very important and challenging task. This paper addresses this challenge by considering highly dynamic networks in which sensors have little or no control on who they interact with. Specifically, we assume a non-adaptive scheduler adversary that dictates in advance a meeting-pattern for the sensors. However, the adversary we assume is not unlimited. Specifically, to ease the analysis, we restrict the adversary to provide independent-meeting patterns only, in which it is guaranteed that whenever a sensor views another sensor, their transitive histories are disjoint. Although they are not very good representatives of communication in static networks, independent meeting patterns fit very well with highly stochastic communication patterns as well as short-time scales, in which each sensor observes only few other arbitrary sensors. Given such an adversarial meeting-pattern, we are concerned with minimizing the deviation of each internal clock from the global time.

As our objective is to model small and simple sensors, we are interested in algorithms that employ elementary internal computations and economic use of communication. We use competitive analysis to evaluate the performances of algorithms, comparing them to the best possible algorithm that knows the whole meeting pattern in advance and operates under the most liberal version of the model that allows for unrestricted resources in terms of memory and communication capacities, and individual computational ability.

Project Goals

This project aims to achieve the following:

1) Lacking a global reference time: it's hard to know the state of a process during a distributed computation, However, it's important for processes to share a common time notion
2) The hardware clock of a set of computers (system nodes) may differ because they count time with different frequencies

Load balancing and resource sharing arise the two main objectives of distributed systems. To achieve these objectives nodes should communicate with each other. In such an environment, it is necessary that the different nodes in the system should have a common time based on which they can order the events. The clocks of the communicating nodes should agree upon a common time value. If the system is working on real time applications like aviation traffic control and position reporting, radio and TV programming launch and monitoring, multimedia synchronization for real-time teleconferencing etc, then the clocks should match with Coordinated Universal Time.

PROJECT EXECUTION PLAN

The figure below shows the basic Gantt chart which serves as a rough project plan.

ID	Task Name	Start	Finish	Duration	Oct 2018					Nov 2018			
					9/27	10/6	10/13	10/20	10/27	11/3	11/10	11/17	11/25
1.	Read papers and gather information	10/2/2018	10/22/2018	3w									
2.	Compile relevant information	10/15/2018	10/26/2018	2w									
3.	Write report	10/20/2018	11/18/2018	4.5 w									
4.	Perform review and verification against goals	10/25/2018	11/20/2018	4 w									
5.	Prepare of presentation	11/22/2018	11/23/2018	.4w									

Figure 1 Project plan chart (please zoom document for clarity)

The above figure is only a rough draft. As the project progresses from conception to realization, external factors such as examinations might influence the schedule to a certain extent.

However, constant reviews will be done to ensure that the project proceeds as planned, and the goals set out in this document are achieved to a satisfactory extent.

REFERENCES

The following list of references represents an intended reading list. Also, additional sources may be chanced upon during the research process.

[1] Elson, J., Girod, L., Estrin, D.: Fine-Grained Network Time Synchronization Using Reference Broadcasts. ACM SIGOPS Operating Systems Review 36, 147–163 (2002).

[2] Ben-Or, M., Dolev, D., Hoch, E.N.: Fast Self-Stabilizing Byzantine Tolerant Digital Clock Synchronization. In: Proc. 27th Annual ACM Symposium on Principles of Distributed Computing (PODC), pp. 385–394 (2008).

[3] Lenzen, C., Locher, T., Wattenhofer, R.: Clock Synchronization with Bounded Global and Local Skew. In: Proc. 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 500–510 (2008).