CS 499 – Distributed Systems  
Monday, September 21, 2009

Class Notes:

* Project 2 is due this Friday
* Project 1 Documentation is due this Friday
* Will need another “static” site. Basically another server.
  + Brian will use the ACM Server for the course jobs.
* QUIZ Wednesday

Lecture Notes:

* With an example of two Servers running. (Server A and Server B)
  + o: Tb = Ta + o

Server A

Server B

Ti-3

Ti-2

Ti-3/t

Ti-1

t’

Ti

Ti

* + Ti-2 = Ti-3 + o + t
  + Ti = Ti-1 + t’ – o
  + 🡺
  + di = a + b = t + t’
  + a = Ti-2 – Ti-3
  + b = Ti – Ti-1
  + o = a – t
  + o = t’ – b
  + 2o = a – b – t + t’
  + o = (a - b)/2 + (t’ – t)/2
  + o = oi + (t’- t)/2
    - oi 🡪 estimate for o
  + 🡺 Zero in a totally symmetric exchange
  + oi – di/2 <= 0 <= oi + di/2
    - -di/2 🡪 (-t-t’)/2
    - +di/2 🡪 (t+t’)/2
  + // What this algorithm allows for is pairs
    - <oi, di>
    - // in reality you would create a series of pairs and choose the lowest one.

Ti-2 = Ti-3 + t + o and Ti = Ti-1 + t’ – o

di = t + t’ = Ti-2 – Ti-3 + Ti – Ti-1

o = oi + (t’ – t)/2 where oi = (Ti-2 – Ti-3 + Ti-1 – Ti)/2

* Logical Clocks
  + // Physical time for us doesn’t really matter
    - The reason for this is that we are concerned with ordering and …
  + An alternative to physical clocks
  + Happened before relationship
    - order of events on a process as observed
    - Sending happens before Receiving
    - HB is transitional
      * e1 🡪 e2 & e2 🡪 e3
        + e1 🡺 e3
  + Lamport’s Logical Clock
    - // General Def: all that a logical clock is, is an integer counter that only increase
      * Software counter, only incrementing
      * Processes have their own clock
      * used to timestamp events
        + How do you do that?

you start somewhere

you use timestamps to move a time on a separate process when you are switching between processes

* + - * + What does this by us?

example:

e 🡪 e’ 🡺L(e) < L(e’)

The reverse is not true!

* + Vector Timestamps
    - What do you expect?
      * multiple elements to the timestamps
      * as many components as processes
    - What do you win?
      * overcomes the drawback (the shortcoming) of Lamport’s timestamps;
      * L(e) < L(e’) =/=> e 🡪 e’
    - How do you construct the vector timestamps
    - Vector clock Vi at process Pi 🡪
      * array of integers, dim N
      * VC1: initially Vi[j] = 0, for j = 1…N
      * VC2: before Pi timestamps event it sets Vi[i] = Vi[i]++
      * VC3: Pi piggybacks t = Vi

TODO:

* Project 2 Due FRIDAY!!
* Project 1 Documentation Due FRIDAY!