Script **generated by TTT**

Title: Distributed_Applications (03.06.2013)

Date: Mon Jun 03 09:18:28 CEST 2013

Duration: 37:13 min

Pages: 10

Synchronizing physical clocks

physical clocks are used to compute the current time in order to timestamp events, such as

modification date of a file

time of an e-commerce transaction for auditing purposes

Synchronization in a synchronous system

a synchronous distributed system is one in which the following bounds are defined

the time to execute each step of a process has known lower and upper bounds.

each message transmitted over a channel is received within a known bounded time.

each process has a local clock whose drift rate from real time has a known bound

Internal synchronization in a synchronous system

One process p1 sends its local time t to process p2 in a message m

p2 could set its clock to $t + T_{trans}$ where T_{trans} is the time to transmit m

T_{trans} is unknown but

$min \leq T_{trans} \leq max$

uncertainty u = (max-min) . Set clock to

t + (max - min)/2

then skew $\leq$ u/2.

Approach

A time server S receives signals from a UTC source

Process p requests time in m1 and receives t in m2 from S.

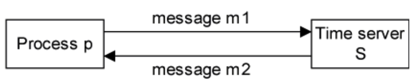
p sets its clock to: $t + T_{round}/2$

Accuracy is +/- ($T_{round}/2$ - min).

because the earliest time S puts t in message m2 is min after p sent m1: t + min

the latest time was min before m2 arrived at p: $t + T_{round}$ - min

the time by S's clock when reply message m2 arrives is in the range [t + min, $t + T_{round}$ - min]



Discussion

The approach has several problems. It is only suitable for deterministic LAN environment or Intranet.

- a single time server might fail

  ⇒ redundancy through group of servers, multicast requests

- it does not deal with faulty time servers

  how to decide if replies vary (byzantine agreement problems)

- imposter providing false clock readings

Cristian's method for an asynchronous system

Observations:

- round trip times between processes are often reasonably short in practice, yet theoretically unbounded
- practical estimate possible if round-trip times are sufficiently short in comparison to required accuracy

Approach

Berkeley algorithm

Both algorithms (Cristian and Berkeley) are not really suitable for Internet.

Cristian's method for an asynchronous system

Observations:

- round trip times between processes are often reasonably short in practice, yet theoretically unbounded
- practical estimate possible if round-trip times are sufficiently short in comparison to required accuracy

Approach

Berkeley algorithm

Both algorithms (Cristian and Berkeley) are not really suitable for Internet.

Generated by Targeteam

NTP - synchronization of servers

The synchronization subnet can reconfigure if failures occur, e.g.

- a primary that loses its UTC source can become a secondary
- a secondary that loses its primary can use another primary

Modes of synchronization

Multicast:

- A server within a high speed LAN multicasts time to others which set clocks assuming some delay (not very accurate)

Procedure call:

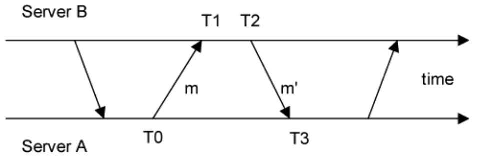
- A server accepts requests from other computers (like Cristian's algorithm). Higher accuracy. Useful if no hardware multicast.

Symmetric:

- Pairs of servers exchange messages containing time information
- Used where very high accuracies are needed (e.g. for higher levels)

Messages between a pair of NTP peers

All modes use UDP transport protocol for the message exchange



Each message bears timestamps of recent events:

- Local times of Send and Receive of previous message
- Local time of Send of current message

Recipient (Server A) notes the time of receipt T3 (we have T0, T1, T2, T3).

In symmetric mode there can be a non-negligible delay between messages

Accuracy of NTP

For each pair of messages between two servers, NTP estimates an offset o between the two clocks and a round-trip delay d_i (total transmission time for the two messages m and m', which take t and t')

$T_{i-2} = T_{i-3} + t + o$ and $T_i = T_{i-1} + t' - o$

This gives us the delay (by adding the equations)

$d_i = t + t' = T_{i-2} - T_{i-3} + T_i - T_{i-1}$

Also the offset (by subtracting the equations)

$o = o_i + (t' - t)/2$, where $o_i = (T_{i-2} - T_{i-3} + T_{i-1} - T_i)/2$

Estimate of offset

Using the fact that t, t' > 0 it can be shown that

$o_i - d_i/2 \le o \le o_i + d_i/2$.

Thus o_i is an estimate of the offset and d_i is a measure of the accuracy

NTP servers filter pairs $<o_i , d_i>$

retains the 8 most recent pairs

estimates the offset o

NTP applies peer-selection to identify peer for reliability estimate.

Accuracy

over Internet: tens of ms

over a LAN: 1 ms

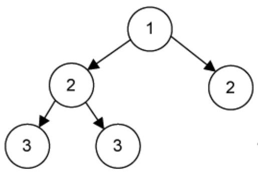
Network Time Protocol (NTP)

Cristian and Berkeley algorithm are intended for the Intranet.

NTP defines an architecture for a time service and a protocol to distribute time information over the Internet.

use of 64 bit timestamps.

NTP synchronizes clients to UTC.



Primary servers are connected to UTC sources

Secondary servers are synchronized to primary servers

Synchronization subnet - lowest level servers in users' computers

NTP - synchronization of servers

Messages between a pair of NTP peers

Accuracy of NTP

Accuracy of NTP

$T_{i-2} = T_{i-3} + t + o$ and $T_i = T_{i-1} + t' - o$

This gives us the delay (by adding the equations)

$d_i = t + t' = T_{i-2} - T_{i-3} + T_i - T_{i-1}$

Also the offset (by subtracting the equations)

$o = o_i + (t' - t)/2$, where $o_i = (T_{i-2} - T_{i-3} + T_{i-1} - T_i)/2$

Estimate of offset

Using the fact that t, t' > 0 it can be shown that

$o_i - d_i/2 \le o \le o_i + d_i/2$.

Thus o_i is an estimate of the offset and d_i is a measure of the accuracy

NTP servers filter pairs $<o_i , d_i>$

retains the 8 most recent pairs

estimates the offset o

NTP applies peer-selection to identify peer for reliability estimate.

Accuracy

over Internet: tens of ms

over a LAN: 1 ms

Generated by Targeteam