L19: Time & Ordering

Nickolai Zeldovich

6.033 Spring 2012

Simple time sync protocol

```
sync(server):
    tsrv = getTime(server)
    local_time = tsrv
```

client

```
getTime():
    return local_time
```

server

Estimating network latency

```
sync(server):
    t_begin = local_time
    tsrv = getTime(server)
    t_end = local_time
    delay = (t_end-t_begin) / 2
    offset = (t_end-delay) - tsrv
    local_time = local_time - offset
```

Slew time

```
sync(server):
    t_begin = local_time
    tsrv = getTime(server)
    t_end = local_time
    delay = (t_end-t_begin) / 2
    offset = (t_end-delay) - tsrv

    freq = base + ε * sign(offset)
    sleep(freq * abs(offset) / ε)
    freq = base

timer_intr():    # on every oscillator tick..
    local_time = local_time + 1/freq
```

temporarily speed up / slow down local clock

Adjust local frequency estimate

```
sync_freq(server):
    tc0 = local_time
    ts0 = getTime(server)

    sleep(N)

    tc1 = local_time
    ts1 = getTime(server)

    ratio = (tc1-tc0) / (ts1-ts0)
    freq = freq * ratio
```

set local
frequency
to match
server

Summary

- NTP can synchronize time across the Internet
  - Be careful w/ assumptions, when using time

- Optimistic concurrency: concurrent changes

- Vector timestamps help detect concurrent changes