Time

Clock Synchronization
- Time is unambiguous in centralized systems: the system clock keeps time, all entities use this for time.
- Distributed systems: each node has its own system clock.
  - Crystal-based clocks are less accurate (1 part in million).
  - Problem: An event that occurred after another may be assigned an earlier time.

Physical Clocks: A Primer
- Accurate clocks are atomic oscillators (one part in $10^{13}$).
- Most clocks are less accurate (e.g., mechanical watches).
  - Computers use crystal-based clocks (one part in million), which result in clock drift.
- How do you tell time?
  - Use astronomical metrics (solar day).
  - Coordinated universal time (UTC): an international standard based on atomic time.
  - Add leap seconds to be consistent with astronomical time.
  - UTC broadcast on radio (satellite and earth).
  - Receivers accurate to 0.1 – 10 ms.
- Need to synchronize machines with a master or with one another.

Event Ordering in Make

[Diagram showing event ordering in Make]
Clock Synchronization

Each clock has a maximum drift rate $\rho$

$1 - \rho \leq \frac{dC}{dt} \leq 1 + \rho$

Two clocks may drift by $2\rho \Delta t$ in time $\Delta t$

To limit drift to $\delta$, resynchronize every $\frac{\delta}{2\rho}$ seconds

Cristian’s Algorithm

- Synchronize machines to a time server with a UTC receiver
- Machine P requests time from server every $\frac{\delta}{2\rho}$ seconds
  - Receives time $t$ from server, P sets clock to $t + t_{\text{reply}}$
  - Use $(t_{\text{req}} + t_{\text{reply}})/2$ as an estimate of $t_{\text{reply}}$
- Improve accuracy by making a series of measurements
Cristian’s Algorithm

- Each clock has a maximum drift rate $\rho$.
- $1 - \rho \leq \frac{dC}{dt} \leq 1 + \rho$.
- Two clocks may drift by $2\rho \Delta t$ in time $\Delta t$.
- To limit drift to $\delta$, resynchronize every $\delta / 2\rho$ seconds.

Berkeley Algorithm

- Used in systems without UTC receiver.
- Keep clocks synchronized with one another.
- One computer is master, others are slaves.
- Master periodically polls slaves for their times.
- Average times and return differences to slaves.
- Communication delays compensated as in Cristian’s algorithm.
- Failure of master => election of a new master.
**Network Time Protocol (NTP) Strata**

Each message bears timestamps of recent events:

All messages sent using UDP in current version of NTP

- More Stratum 2 computers
- More Stratum 1 time servers
- Directly to Stratum 0 devices
- And so on…

Stratum 3: Send requests to one or

Stratum 2: Send requests to one or

Stratum 0: atomic clocks, GPS clocks, radio clocks

Up to 256 (!) strata levels supported

---

**NTP Synchronization**

\[
\begin{align*}
o_i &= (T_{i-2} - T_{i-3} + T_{i-1} - T_i) / 2 \\
d_i &= T_{i-2} - T_{i-3} + T_i - T_{i-1}
\end{align*}
\]
Logical Clocks

Instead of synchronizing clocks, event ordering can be used

Each process

Among these there were:

In 1999 there were 175,000 hosts running NTP in

timestamps to events using the following rules:

•

Lamport’s Logical Clocks

What do we know about a and e?

3. The happened before relation is transitive

Rule 1: a

Rule 3: a

Example: L(e) < L(b) but b || e

Rule 1: a

Rule 3: a

Physical time

Physical time

Logical Clocks

Lamport Clocks

Logical Clocks

Lamport Clocks

Physical time

Physical time
Vector Clocks

- Own element in clock = (2,1,0) for event c
- Can you see a pair of parallel events?
- And now the converse is also true (L(e) < L(e') implies e happened before e')
- Note that e happened before e'
- What's the meaning of =, <=, max etc for vector timestamps?
- At p, get max ((0,0,1), (2,2,0)) = (2,2,1) and add 1 to
- How can we overcome the limitation of Lamport's logical clocks?
- Vector clocks are used in many schemes for replication of
- Algorithms (e.g. Cristian's and NTP) synchronize clocks in
- Accurate timekeeping is important for distributed systems
- Distributed processes often need to coordinate their
- Mutual exclusion that is based solely on message passing
- Often called the critical section problem
- Discusses in detail in OS courses
- In this class, we need
- Mutually exclusive activities
- Physical time
- For ordering of an arbitrary pair of events at different
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Mutual exclusion that is based solely on message passing
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or
- Compare elements pairwise
- Physical time
- The happened-before relationship between events.
- Reflects a flow of information between them
- Vector clocks are an improvement on Lamport clocks
- Lamport clocks are counters that are updated according to
- For ordering of an arbitrary pair of events at different
- We can tell whether two events are ordered by happened-before or