Vector Clocks

COS 418 + 518: (Advanced) Distributed Systems
Lecture 5
Mike Freedman & Wyatt Lloyd

Lamport Clocks Review

Q: \( a \rightarrow b \) \( \Rightarrow \) \( \text{LC}(a) < \text{LC}(b) \)

Q: \( \text{LC}(a) < \text{LC}(b) \) \( \Rightarrow \) \( b \rightarrow a \) ( \( a \rightarrow b \) or \( a \parallel b \) )

Q: \( a \parallel b \) \( \Rightarrow \) nothing

Lamport Clocks and Causality

- Lamport clock timestamps do not capture causality
- Given two timestamps \( C(a) \) and \( C(z) \), want to know whether there's a chain of events linking them:

\[
a \rightarrow b \rightarrow ... \rightarrow y \rightarrow z
\]

Vector clock: Introduction

- One integer can’t order events in more than one process
- So, a Vector Clock (VC) is a vector of integers, one entry for each process in the entire distributed system

- Label event \( e \) with \( \text{VC}(e) = [c_1, c_2, ..., c_n] \)
  - Each entry \( c_k \) is a count of events in process \( k \) that causally precede \( e \)
Vector clock: Update rules

- Initially, all vectors are \([0, 0, \ldots, 0]\)
- Two update rules:
  1. For each local event on process \(i\), increment local entry \(c_i\)
  2. If process \(j\) receives message with vector \([d_1, d_2, \ldots, d_n]\):
     - Set each local entry \(c_k = \text{max}(c_k, d_k)\)
     - Increment local entry \(c_j\)

Vector clock: Example

- All processes' VCs start at \([0, 0, 0]\)
- Applying local update rule
- Applying message rule
  - Local vector clock piggybacks on inter-process messages

Comparing vector timestamps

- Rule for comparing vector timestamps:
  - \(V(a) = V(b)\) when \(a_k = b_k\) for all \(k\)
  - \(V(a) < V(b)\) when \(a_k \leq b_k\) for all \(k\) and \(V(a) \neq V(b)\)
- Concurrency:
  - \(V(a) \parallel V(b)\) if \(a_i < b_i\) and \(a_j > b_j\), some \(i, j\)

Vector clocks capture causality

- \(V(w) < V(z)\) then there is a chain of events linked by Happens-Before (\(\rightarrow\)) between \(a\) and \(z\)
- \(V(a) \parallel V(w)\) then there is no such chain of events between \(a\) and \(w\)
Comparing vector timestamps

• Rule for comparing vector timestamps:
  • $V(a) = V(b)$ when $a_k = b_k$ for all $k$
  • They are the same event
  • $V(a) < V(b)$ when $a_k \leq b_k$ for all $k$ and $V(a) \neq V(b)$
  • $a \rightarrow b$

• Concurrency:
  • $V(a) \parallel V(b)$ if $a_i < b_i$ and $a_j > b_j$, some $i, j$
  • $a \parallel b$

Two events $a, z$

Lamport clocks: $C(a) < C(z)$
  Conclusion: $z \rightarrow a$, i.e., either $a \rightarrow z$ or $a \parallel z$

Vector clocks: $V(a) < V(z)$
  Conclusion: $a \rightarrow z$

Vector clock timestamps precisely capture happens-before relation (potential causality)