DISTRIBUTED SYSTEMS
CSCI 4963/6963

9/22/2015
Replicated Log and Dictionary Problems
(Wuu and Bernstein)

- System Model
  - N sites (processes), numbered 1, ..., N
  - 3 types of events: local, send msg, receive msg
  - Can have site crashes, lost messages, broken links, and network partitions
  - Events are atomic
  - Sites have stable storage (survives crashes)

- An execution is a set of events and a partial order (happens-before), denoted \(<E, \rightarrow>\) .
Log Definition

- Every site $i$ has a log $L_i$.
- Every site has a local counter (clock)
  - Just counts events (not a Lamport clock)

- Each entry $e$ in the log corresponds to an event.
  - Record of event $e$ is $eR$:
    - $e. \text{op}$: operation type
    - $e. \text{time}$: value of counter
    - $e. \text{node}$: node ID
  - May only be interested in logging certain types of events.

- $L(e)$ denotes log at node where $e$ took place, immediately following event $e$. 
The Log Problem

• The log problem is:

Find an algorithm to maintain the logs such that, given an execution \(<E, \rightarrow>\), if \(f \rightarrow e\) then, \(fR \in L(e)\).

• Naïve solution:
  • node \(i\) sends entire log with every message to node \(j\)
  • On receipt of message, node \(j\) updates log \(L_j = L_j \cup L_i\)

• Is it possible to reduce message overhead?
Observation

- Once site i knows that site j knows about an event, it does not need to include that event in its messages to site j.