

# Distributed Snapshots: Determining Global States of Distributed Systems by K. Mani Chandy and Leslie Lamport

CSC 724 Paper review - Vaibhav Singh, vsingh7

January 13, 2019

## 1 Summary

The paper presents algorithms by which processes in a distributed system can come together to determine the global state of the system. The processes do not share common clock or memory.

## 2 Description

The paper defines a class of problems which are composed of *\*stable\** properties i.e. if a statement is true at some point of time it remains true at all later instances of time.

It defines a distributed system as a finite set of processes and finite set of channels.

Process is defined by a set of states, an initial state, and a set of events.

Event  $e$  is defined as an atomic operation which changes the state of originating process  $p$  and at most one more channel incident on/to  $p$ .

Marker message is defined as a special message having no effect on the underlying computation and states of any process.

Global state of a distributed system is defined as a set of all component processes and channel states.

Given the above definitions, The global state-detection algorithm is defined as follows:

For each channel  $c$  connected to process  $p$ ,

$p$  first records its state and then sends marker messages along  $c$ , before sending any further messages along  $c$ .

On receiving marker message along  $c$ , if recipient process  $q$  hasn't recorded  $c$ 's state,  $q$  records its own state and records state  $c$  as empty. Else  $q$  records the state of  $c$  as the sequence of messages received along  $c$  after  $q$ 's state was recorded and before  $q$  received the marker along  $c$ .

The paper then establishes proofs of global-state recording algorithm terminating in finite time, given the assumptions that markers do not remain forever in a channel and processes take a finite time in recording their states (both reasonable).

### 3 Strong Points

The paper defines the relationship among local process states, global system states, and points in a distributed computation. The algorithm defined in the paper should work for a system working within the constraints of using a reliable communication medium.

### 4 Weak Points

The paper assumes a lot of things, not all of which might be reasonable. For example, Channels are assumed to have infinite memory (buffers), to be error free, to deliver messages in the order sent.

The paper also assumes no failures occurring among processes or channels (for example a system using \*best-effort\* communication protocols like UDP).

The algorithm also works only for a class of systems which behave causally i.e. systems having stable properties.

The number of messages increases polynomially with increase in number of processes (as each process will have to take into account the states of every other process in the system). This can lead to substantial bandwidth consumption to get distributed snapshot for a large enough number of processes in the system.

### 5 Improvement

As mentioned in [1] The limitation of messages being ordered can be addressed by using techniques like *message inhibition* [2], or piggybacking of control information on computation messages to capture out-of-sequence messages.

### References

- [1] Ajay D Kshemkalyanit, Michel Raynalt and Mukesh Singhal, *An introduction to snapshot algorithms in distributed computing*.
- [2] Helary I-M, *Observing global states of asynchronous distributed applications* Proc. 3rd Int. Workshop on Distributed Algorithms, LNCS 392 (Berlin: Springer) pp 124-34