Review of Distributed Snapshots: Determining Global States of Distributed Systems

Aditya Patel

23 September 2019

In a distributed system, a process can timestamp its own events (based on local time) or timestamp when it sends a message to other process (in different node) or receives a message from process (in different node). Determining the global state of the system using this information can help solve many problems in distributed systems, in particular, stable property detection problem (without a need of global clock). The paper presents a global-state detection algorithm to solve stable property detection problem.

An event e in a process p may change state of p itself or state of at most one channel incident on p (if a message is sent or received along the channel). Theoretically, an event can represented as a 5-tuple (p, s, s', M, c), meaning a process p can change the state of process from s to s', and might chanege the state of c if M was sent to or from p. The idea of the global state algorithm is that 1) each process record its own state and 2) two process connected by a channel will cooperate together to form a "global state between the processes." The algorithm is superimposed on existing computation (i.e. it runs concurrently with existing computation).

Global-state recording algorithm works by sending and receiving markers along a channel while following two protocols: 1) p sends one marker to q along a channel c only after p records its state but before p sends further messages. 2) q records 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. If q never recorded before (which would happen when the system with this protocol is initialized), it would store state of c as empty sequence. As long as messages are not stuck in the channel, and as long as recording the state doesn't take infinite time, this algorithm is guaranteed to terminate. In order to retrieve global state of the system, recorded process state and channel state must be retrieved. The paper presents important properties of the states recorded by the algorithm and states how to use these to solve the stability detection problem.

Potential flaws of the paper: The paper makes many assumptions which does not always hold in real distributed systems. For example, authors assume that the channel buffer can hold infinite messages and these messages are received in the order sent. While the first scenario is obviously unrealistic, second scenario may not always be guaranteed. Moreover, in case of fault along channel, some or all of the markers can be lost, making the result of algorithm incorrect. Moreover, collecting process and channel states from the nodes could be practically slow. Given this limitations, the algorithm cannot be directly implemented in real distributed systems.

Strength of paper: The paper provides a strong theoretical foundation of an algorithm that could help record the global state of the distributed systems. They provide theoretically rigorous definitions of process, channel, event, and states which helped prove theoretical guarantees of the algorithm. This has greatly benefited the research community in distributed systems by providing a strong foundation (cited 3500+). A real distributed system could be inspired from the ideas presented in the paper.