EpTO: An Epidemic Total Order Algorithm for Large-Scale Distributed Systems

Jocelyn Thode*, Ehsan Farhadi†
Université de Neuchâtel
Neuchâtel, Switzerland
Email: *jocelyn.thode@unifr.ch, †ehsan.farhadi@unine.ch

***Abstract*—One of the fundamental problems of distributed computing, is the ordering of events through all peers. From all the available orderings, total ordering is of particular interest as it provides a powerful abstraction for building reliable distributed applications. unfortunately, existing algorithms can not provide reliability, scalability, resiliency and total ordering in one package. EpTO is a total order algorithm with probabilistic agreement that scales both in the number of processes and events. EpTO provides deterministic safety and probabilistic liveness: integrity, total order and validity are always preserved, while agreement is achieved with arbitrarily high probability. We are going to implement EpTO using NeEM library and show EpTO is well-suited for large-scale dynamic distributed systems, and afterwards we will evaluate this algorithm by comparing it with currently being used ordering algorithms.**

I. Introduction

The ordering of events is one of the most fundamental problems in distributed systems, and it has been studied over past few decades. A lot of researchers have been working on an algorithm with different guarantees and tradeoffs such as synchronization, agreement or state machine replication. But because these properties are strong guarantees, the algorithms that implements them do not scale very well. Existing probabilistic protocols are highly scalable and resilient, but they can not provide total ordering in large scale distributed systems, and existing deterministic protocols which provide total ordering, are not resilient and scalable.

The problem with existing deterministic total ordering protocols, is that they need some sort of agreement between all peers in the system. An agreement on the order of messages, which cause a massive amount of network traffic and overhead on the system. Plus, an agreement feature for an asynchronous system requires to explicitly maintain a group and have access to a failure detector. Due to the faults and churn in large-scale distributed systems, failure detector turns into bottleneck of the system and thus, limits scalability of the algorithm.

A. Contribution

EpTO guarantees that processes eventually agree on the set of received events with high probability and deliver these events in total order to the application. The intuition behind EpTO is that events are available quickly at all nodes with high probability, Once events are thought to be available in every peer, each peer deterministically order them by timestamp of each event and breaking ties with the id of the broadcaster peer, and deliver them to the application accordingly.

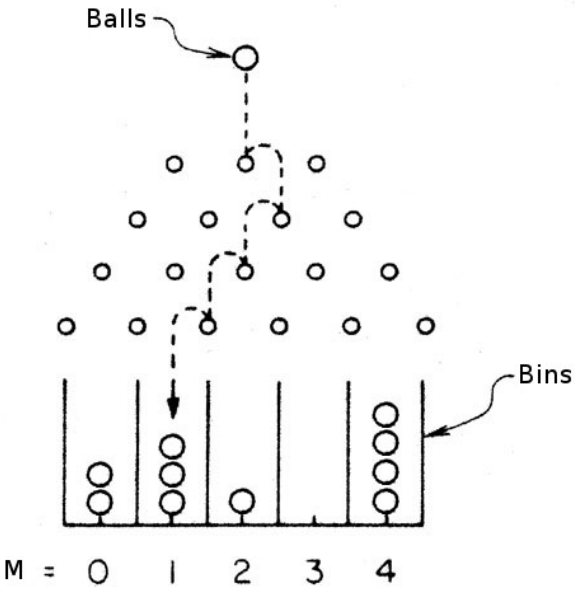

Figure 1. balls-and-bins[1].

The main insight behind EpTO dissemination protocol is a *balls-and-bins* approach. *Balls-and-bins* is a basic probabilistic problem: consider n balls and m bins where we consequently throw balls into a bin, completely random and independent from other balls.

A balls-and-bins approach abstract peers as bins and messages (events) as balls, and studies how many balls need to be *thrown* such that each bin contains at least one ball with arbitrarily high probability. using this approach The number of messages transmitted per process per round is logarithmic in the number of processes, and the total number of messages transmitted in the network before an event is delivered is low and uniform over all peers.

II. EPTO ALGORITHM DESCRIPTION

1) EpTO Dissemination Component:

2) EpTO Ordering Component:

3) EpTO Stability Oracle:

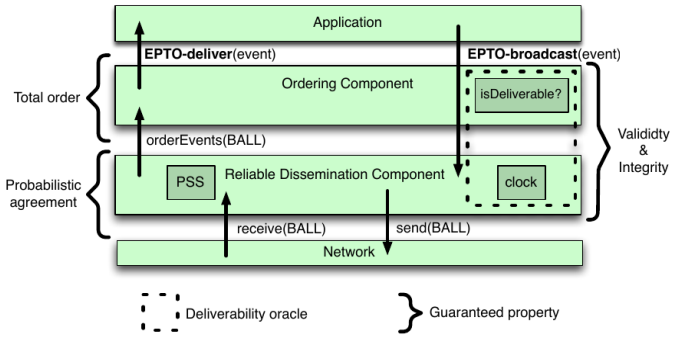

Figure 2. test^2.

III. TODO

IV. Work Plan

V. Conclusion

Acknowledgment