Efficient Scaling of Stateful Actors

1 Thoughts

1.1 Problem Statement

Actor Frameworks: Actors provide a shared nothing abstraction which allows horizontal scaling. Scaling stateful actors is a challenge because of state reconciliation overheads. The problem has similarities with storage systems where replication is necessary to scale read/write performance. Some key differences are:

1) In a LAN setting, the number of instantiated actors can be much larger than the total number of read/write replicas – Can we show some plots based on data? 2)

Given a latency SLA and read write requirement, we need to decide the following things: 1) Whether to instantiate a new actor, and 2) Consistency level of the new actor

We can probably break down the problem. In the first instance the user can provide client SLAs with both consistency and latency similar to Pileus and we need to propose an algorithm which decides whether to instantiate an actor or not. As a next step, we can remove consistency from the SLA and see if we can decide the consistency from the latency SLA

This work is different from Pileus because it does not handle replica creation, leader election which our work needs to consider in an online manner

In "Take me to your leader" work, given a set of possible sites, it tries to figure out replica locations, roles and leader. This calculation happens periodically and the replica roles are changed. They do not consider SLAs though.

1.2 Paper Format