topicos

1. INTRODUCTION

- Num. DC a aumentar
- Replicar totalmente tem problemas
- Replicação parcial
- Queries sobre dados replicados parcialmente
  - Standard solution?
- Views materializadas replicadas totalmente
- Contribuições

2. SYSTEM OVERVIEW

- System model
  - Replicação parcial
- System API
  - "Create table"
  - "Create view"
    - CRDT não uniforme
    - put numa table $\implies$ puts nas várias views
      - consistência das views face aos dados - in sync
- System description
  - CRDT não uniforme
  - Implementação de queries?

3. IMPLEMENTATION

4. EVALUATION

5. RELATED WORK

6. CONCLUSIONS

7. SYSTEM OVERVIEW

Possiveis pontos mais detalhados?

7.1 System model

- Network assumptions
- Client-server interaction (refer key-value store interface? Maybe refer this instead in System API?)
- Server-server interaction? (is it needed? We'll already touch this in Replication.)
- System guarantees
  - CRDTs
  - Consistency level
- Replication
- Async
- Op-based
- Maintains consistency, i.e., transaction level based.
- Partial (system admin defined, each server only has a subset of the data based on topics. Potencially some data can be replicated everywhere)

7.2 System API

- Basically how can we translate a problem to sql-like operations
- Create table
- Create view
- Updates (incluir problema de consistência de views/dados)
- Queries (incluir aqui problema de os CRDTs não uniformes precisarem de mais dados? Ou na zona da view?)

7.3 System description

- Structure? Maybe that's for implementation? How much detail?
  - Internal partitioning vs external partitioning? Capaz de não ser boa ideia...
- CRDTs and non-uniform CRDTs?

8. IMPLEMENTATION

- Go
- Transactions (TM/Mat?)
- Replication (RabbitMQ and other stuff?)
- Communication (protobufs. Also worth noticing the compability with existing AntidoteDB clients)
- CRDTs (version management at least)