




# Shared data types for cloud computing: Commutative Replicated Data Types

Marek Zawirski, INRIA & UPMC, France
marek.zawirski@lip6.fr

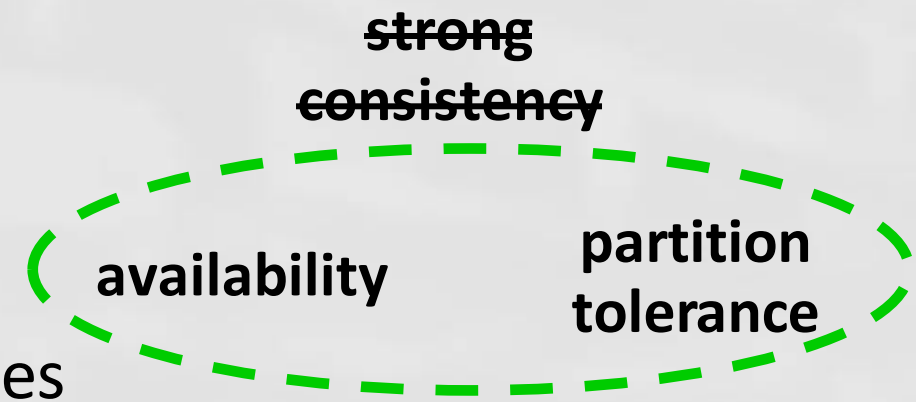
Advisor: Marc Shapiro, INRIA & LIP6, France
In collaboration with: Nuno Preguiça, UNL, Portugal

EuroSys Doctoral Workshop 2011, Salzburg

# Motivation

## Cloud apps need a scalable replicated data layer

- CAP theorem




- *Eventual Consistency:*
  - transient inconsistencies
  - *eventually* converge

| | |
|---|---|
| ✓ Available, scalable | ● Ad-hoc, error-prone |
| ✓ Fault-tolerant | ● No sound theory |
| ✓ In production! | ● Only simple data models |

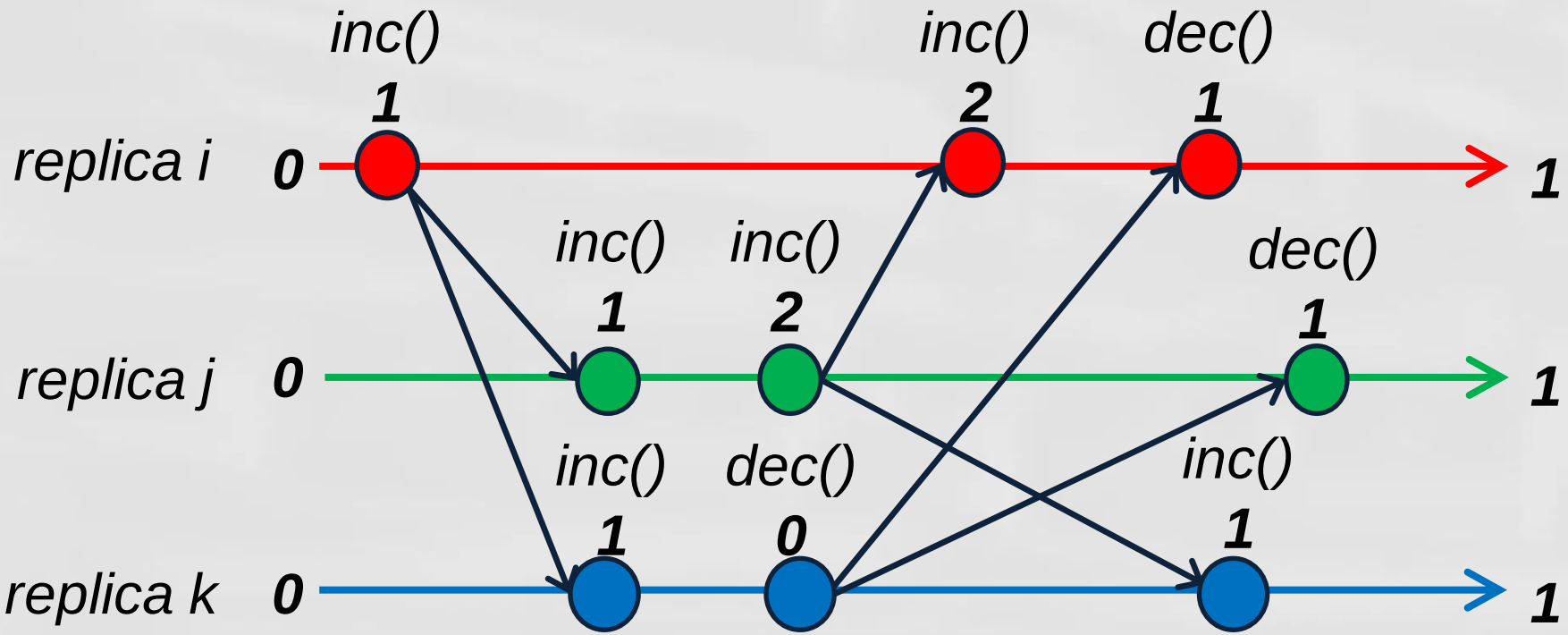
# Commutative Replicated Data Types

## A principled approach to eventual consistency

- High-level data types
- Simple theory:
  every pair of concurrent operations *commute*
- Local replica: accept operations,
  always responsive, never blocking
- Remote: propagate operations by *cbcast,*
  replay them

# CRDT example(s)

## Integer counter: inc(), dec()




More than that: registers, ~sets, graphs, seqs…

# Open problems, research directions

## 1. Building systems with CRDTs

- Goal: *proof of concept* app using CRDTs
- Challenges:
  - Drawbacks (ease of use, completeness etc.)?
  - Are CRDTs composable in practice?
  - Can data be partitioned easily?
  - Performance

# Open problems, research directions

## 2. Meta-data overhead

- Problem: meta-data accumulates (e.g. VVs, tombstones)
- Challenges:
  - How to *garbage collect* in poor network conditions?
  - Avoid the problem beforehand?

## 3. A dose of synchronization

- Gen. problem: infrequent non-commutative operation, maintaining invariants

…