On the correctness of general-purpose applications on cloud storages

Laura Bocchi
L.Bocchi@kent.ac.uk
Beyond ACID transactions

- **ACID** transactions (e.g., in DBMSs) ensure Atomicity, Consistency, Isolation and Durability
- **Long running** transactions (e.g., in SOAs) relax Isolation and Atomicity to prevent locks
- **BASE** transactions (e.g., in Clouds) relax consistency to provide availability
  - Basically Available,
  - with Soft State,
  - Eventually consistent
- Consistency, availability and partition tolerance: one must be dropped [CAP theorem]
Consider a scenario with many clients, distributed control and replicated data e.g., Big Data and a cloud.

- **Consistency**
- **Availability**

From [Kossmann et al. ’10]
Relaxing consistency

- Consider a scenario with many clients, distributed control and replicated data e.g., Big Data and a cloud

![Diagram](image)

- **X** Consistency
- **✓** Availability

from [Kossmann et al. ’10]
• Eventual consistency is **suitable sometimes**

• Broad aim: determine if the composition *application(s) + store(s)* will run *correctly*

• Communication, concurrency, SOC:
  • synchronize processes (a bit like memory fences)
  • middleware services to provide “missing” properties

• Correctness depends from the specific application e.g.,
  • YouTube: I want to see how to make a cherry pie …
  • … not necessarily the last video uploaded.

[B. & Melgratti, WS-FM13, SOCA14]