CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

SNAPSHOTS

Lecture D

SAFETY AND LIVENESS

"Correctness" in Distributed Systems

- Can be seen in two ways
- Liveness and Safety
- Often confused – it's important to distinguish from each other

Liveness

- **Liveness = guarantee that something good will happen, eventually**
  - Eventually == does not imply a time bound, but if you let the system run long enough, then …

LIVENESS: EXAMPLES

- **Liveness = guarantee that something good will happen, eventually**
  - Eventually == does not imply a time bound, but if you let the system run long enough, then …
- **Examples in Real World**
  - Guarantee that "at least one of the atheletes in the 100m final will win gold" is liveness
  - A criminal will eventually be jailed
- **Examples in a Distributed System**
  - Distributed computation: Guarantee that it will terminate
  - "Completeness" in failure detectors: every failure is eventually detected by some non-faulty process
  - In Consensus: All processes eventually decide on a value

Safety

- Safety = guarantee that something bad will never happen

SAFETY: EXAMPLES

- **Safety = guarantee that something bad will never happen**
- **Examples in Real World**
  - A peace treaty between two nations provides safety
    - War will never happen
  - An innocent person will never be jailed
- **Examples in a Distributed System**
  - There is no deadlock in a distributed transaction system
  - No object is orphaned in a distributed object system
  - "Accuracy" in failure detectors
  - In Consensus: No two processes decide on different values

Can't we Guarantee both?

- **Can be difficult to satisfy both liveness and safety in an asynchronous distributed system!**
  - Failure Detector: Completeness (Liveness) and Accuracy (Safety) cannot both be guaranteed by a failure detector in an asynchronous distributed system
  - Consensus: Decisions (Liveness) and correct decisions (Safety) cannot both be guaranteed by any consensus protocol in an asynchronous distributed system
  - Very difficult for legal systems (anywhere in the world) to guaranteed that all criminals are jailed (Liveness) and no innocents are jailed (Safety)

In the language of Global States

- **Recall that a distributed system moves from one global state to another global state, via causal steps**
- **Liveness w.r.t. a property Pr in a given state S means**
  - S satisfies Pr, or there is some causal path of global states from S to S' where S' satisfies Pr
- **Safety w.r.t. a property Pr in a given state S means**

  S satisfies Pr, and all global states S' reachable from S also satisfy Pr

Using Global Snapshot Algorithm

- **Chandy-Lamport algorithm can be used to detect global properties that are stable**
  - Stable = once true, stays true forever afterwards
- **Stable Liveness examples**
  - Computation has terminated
- **Stable Non-Safety examples**
  - There is a deadlock
  - An object is orphaned (no pointers point to it)
- **All stable global properties can be detected using the Chandy-Lamport algorithm**
  - **Due to its causal correctness**

Summary

- The ability to calculate global snapshots in a distributed system is very important
- But don’t want to interrupt running distributed application
- Chandy-Lamport algorithm calculates global snapshot
- Obeys causality (creates a consistent cut)
- Can be used to detect stable global properties
- Safety vs. Liveness