

Consistency Intro

29 December 2009
Lecture 11

Slide Credits: Maarten van Steen

Consistency

- Introduction (what's it all about)
- Data-centric consistency

December 29, 2009

ISE 431: Distributed Information Systems

2

Performance and Scalability

Main issue: To keep replicas consistent, we generally need to ensure that all **conflicting** operations are done in the same order everywhere

Conflicting operations: From the world of transactions:

- **Read-write conflict:** a read operation and a write operation act concurrently
- **Write-write conflict:** two concurrent write operations

Guaranteeing global ordering on conflicting operations may be a costly operation, downgrading scalability

Solution: weaken consistency requirements so that hopefully global synchronization can be avoided

December 29, 2009

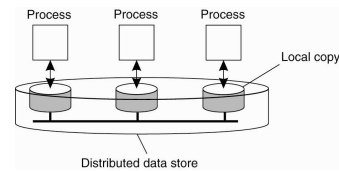
ISE 431: Distributed Information Systems

3

Data-Centric Consistency Models

Consistency model: a contract between a (distributed) data store and processes, in which the data store specifies precisely what the results of read and write operations are in the presence of concurrency.

Essence: A data store is a distributed collection of storages accessible to clients:



December 29, 2009

ISE 431: Distributed Information Systems

4

Continuous Consistency

Observation: We can actually talk about a **degree of consistency**:

- replicas may differ in their **numerical value**
 - Value may be how many updates are missing
 - Weight may be the mathematical distance
- replicas may differ in their relative **staleness**
- there may be differences with respect to (number and order) of **performed update operations**

Conit: consistency unit → specifies the **data unit** over which consistency is to be measured.

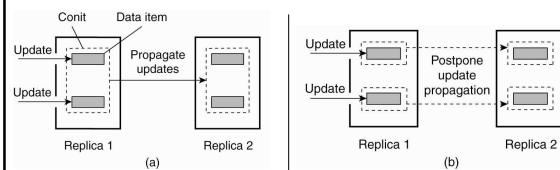
December 29, 2009

ISE 431: Distributed Information Systems

5

Conits

- The **granularity** of your conit is important
- With a given consistency policy:
 - Smaller conits mean you may defer pushing updates
 - Bigger conits means you must push earlier

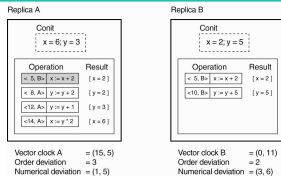


December 29, 2009

ISE 431: Distributed Information Systems

6

Example: Conit



Conit: contains the variables x and y .

- Each replica maintains a **vector clock**
- B sends A operation $[<5, B>: x := x + 2]$; A has made this operation **permanent** (cannot be rolled back)
- A has three **pending** operations \rightarrow order deviation = 3
- A has missed **one** operation from B , yielding a max diff of 5 units \rightarrow (1, 5)