Tree-Based Consistency Approach for Cloud Databases

Md. Ashfakul Islam

Susan V. Vrbsky

Department of Computer Science

University of Alabama

What is a cloud?

- Definition [Abadi 2009]
  - shift of computer processing, storage, and software delivery away from the desktop and local servers
  - across the network and into next generation data centers
  - hosted by large infrastructure companies, such as Amazon, Google, Yahoo, Microsoft, or Sun

Rebirth of Cloud Computing [Hayes 2008]

- Service bureaus and time-sharing systems of 1960s
- On-demand access to computing machinery for users
- Hub-and-spoke configuration
- User communication over telephone

Types of Cloud Service

- According to architectural structure [Sun 2009]
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
  - Software as a Service (SaaS)

- Database solution
  - Database as a Service (DaaS)

Database as a Service

- Very attractive solution for small startup companies
- Centralized or distributed ?
- Distributed solution suited more for TB data size [Abadi 2009]
- Uninterrupted service requires replication
- Consistency becomes a complicated issue
- Conventional time consuming way degrade the performance greatly

Issues in Databases

- ACID properties
  - **A**tomicity, **C**onsistency, **I**solation, **D**urability
- Will focus on consistency
  - If multiple copies of the same data, must all have same consistent state
  - In cloud computing replication is heavily utilized
    - To maintain consistency must communicate over network
    - Difficult if unreliable network

Related Work

- 1st consistency model was given in 1970s by Bruce Lindsay [Lindsay et al. 1979]
- Achieve consistency by maintaining distribution transparency [Vogels 2009]
- Eric Brewer's CAP theorem [Brewer 2000]
  - Of **C**onsistency, **A**vailability and network **P**artition only 2 can be achieved at a time

Related Work

- Werner Vogels presents eventual consistency model [Vogels 2009]
- Propose different types of consistency
  - strong consistency
  - weak consistency
  - eventual consistency

Maintaining Consistency in Clouds

- Updates to data require notifying all replicas of update
  - send messages to all replicas
- If network unreliable and not all replicas respond to update, all replicas must wait
- Results in unsuccessful transactions and performance degradation
- Propose tree-based system to maintain consistency with less performance degradation

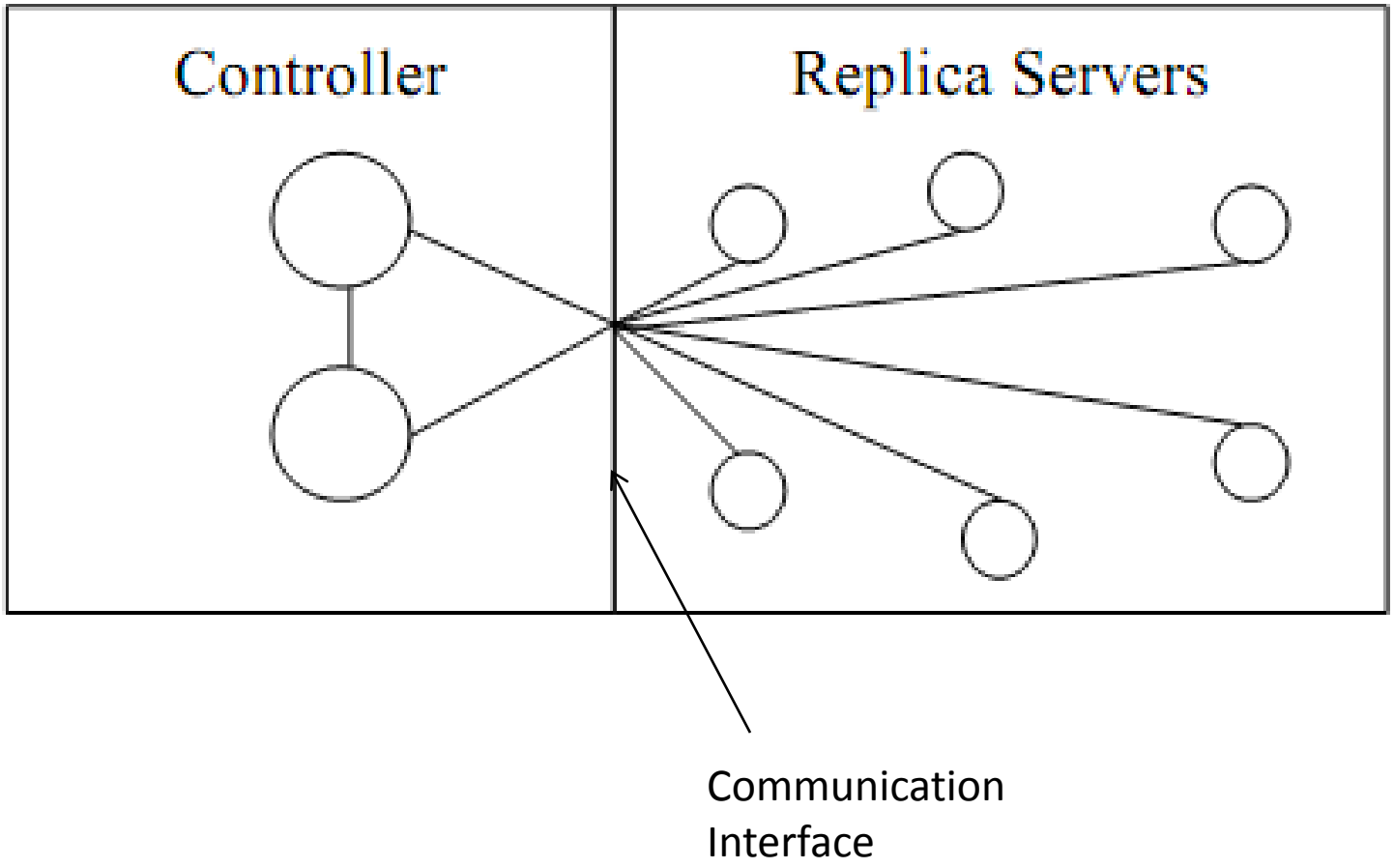
Maintaining Consistency in Clouds - Proposed Solution

- Propose tree-based system to maintain consistency with less interdependency
  - considers reliability of replicas
  - creates tree based on reliability
  - updates sent to replicas based on tree
  - tree is dynamic

Proposed System Description

- Assume following components:
  - Controller
    - two or more controllers
    - build consistency tree
    - handles failure
    - maintain failure log
  - Database Replicas
    - maintain database
    - interconnected
    - primary replica interacts with users

System Communication



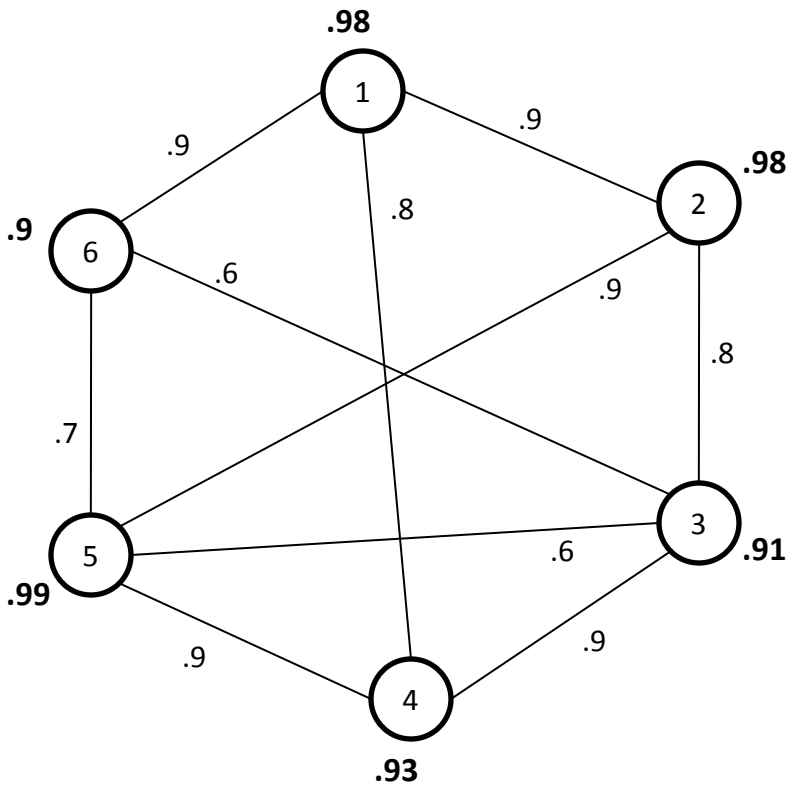
System Communication

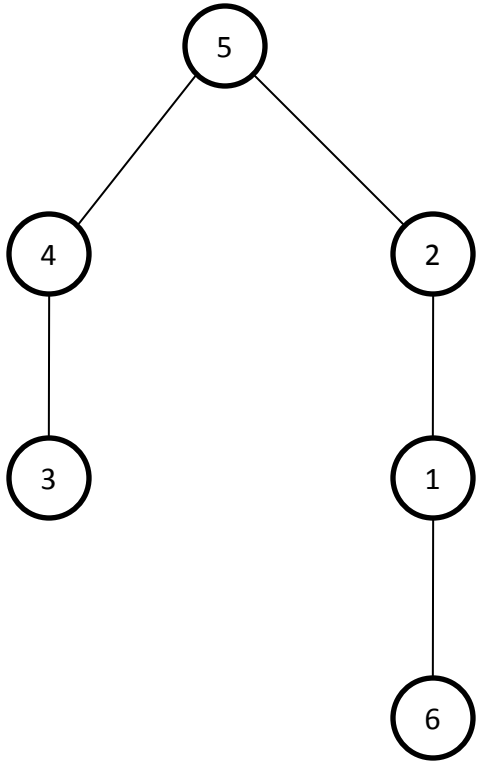
- Controller periodically
  - probes replicas
  - recalculates consistency tree
  - keeps newly added or failure recovered replicas up to date

Building Consistency Tree

- Prepare the connection graph G(V,E)
- Select the root of the tree
- Prepare the consistency tree using Dijkstra's single source shortest path algorithm with slight modification[Cormen *et. al.* Introduction to Algorithms]

Example





Update Operation

- Set Partially Consistent Flag:
  - root receives update request
  - send update request to immediate descendants
  - receives acknowledgement
  - update itself
  - sets operation sequence number as Partially consistent flag

Update Operation

- Set Fully Consistent Flag
  - leaf found empty descendants list
  - set fully consistent flag as operation sequence number
  - informs immediate ancestor
  - ancestor set fully consistent flag after getting confirmation from all descendants

Failure Recovery

- Primary server failure
  - controller finds most updated servers with help of consistency flag
  - finds max reliable server from them
  - rebuild consistency tree
  - initiate synchronization

Failure Recovery

- Other server or communication path down
  - controller is reported unresponsive behavior
  - checks server down or communication down
  - rebuild tree without down server
  - finds alternate path
  - reconfigure tree

Current and Future Work

- Maintaining consistency and high throughput among replica servers is an issue in cloud databases
  - Proposed tree-based approach to address this
  - Work-in-progress – implement simulation of proposed approach, compare to existing strategy
  - Future work – implement approach on private cloud (fluffy at UA)

References

[1] Slashdot. Multiple Experts Try Defining Clouds
Computing.http://tech.slashdot.org/article.pl?sid=08/07/17/2117221.

[2] Brian Hayes. Cloud Computing. communications of the acm, July 2008.

[3] Introduction to Cloud Computing Architecture. Sun Microsystems, white paper. June 2009.

[4] Daniel J. Abadi. Data Management in the Cloud: Limitations and Opportunities, IEEE 2009.

[5] B. G. Lindsay P. G. Selinger C. Galtieri J. N. Gray R. A. Lorie T. G.
Price F. Putzolu B. W. Wade. Notes on Distributed Databases. July
1979.

[6] Eric Brewer. Towards Robust. Distributed Systems. Annual ACM
Symposium on Principles of Distributed Computing. July 2000.

[7] Werner Vogels. Eventually Consistent. communications of the acm,
Jan 2009.

[8] Introduction to Algorithms, Third Edition, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Ex-24.3-4.