"Managing Update Conflicts in Bayou, a Weekly Connected Replicated Storage System"

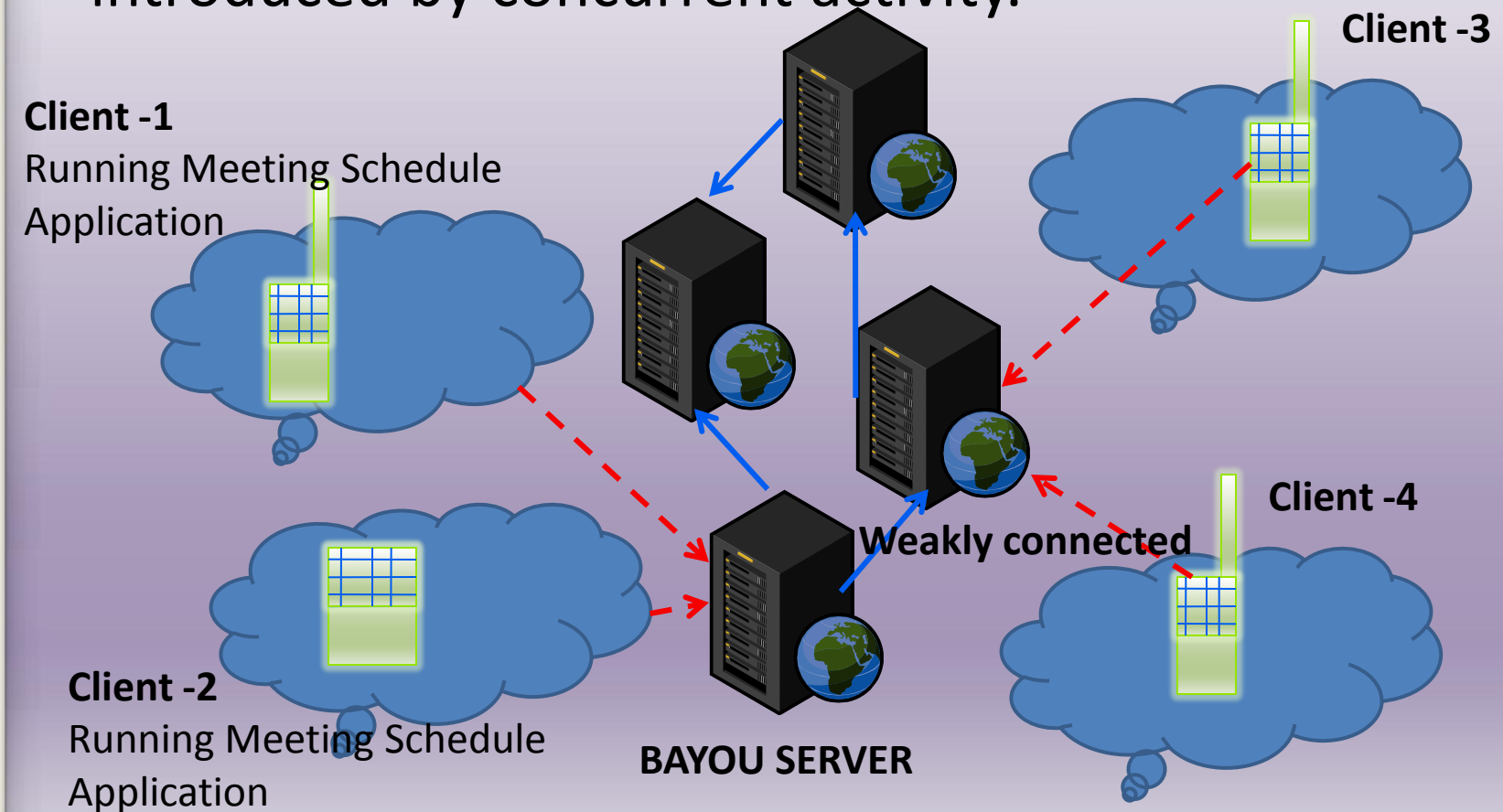
Presented by – **SANKALP KOHLI**

OVERVIEW

- **INTRODUCTION**
  - **GOALS**
  - **LIMITATIONS**
  - **SYSTEM PROPERTIES**
  - **WHY AND HOW?**
- **COLLABORATIVE APPLICATIONS**
- **DESIGN**
  - ALGORITHMS USED (Epidemic Algorithms)
  - BAYOU BASIC SYSTEM MODEL
  - IMPLEMENTATION
    - CONFLICT DETECTION AND RESOLUTION
    - MECHANISM
  - DESIGN OVERVIEW
- **EVALUATION**
  - SETUP
  - EVALUATION RESULTS
- **ADVANTAGES AND DISADVANTES**
- **DISCUSSTION (Questioners)**

INTRODUCTION

- Bayou Storage System provides an infrastructure for collaborative applications that manages the conflicts introduced by concurrent activity.



GOALS

- **Supporting Disconnected work group:**
  - Architecture does not include the notion of **“disconnected”** mode of operation.

- **System Should be Highly Available .**

LIMITATIONS

- Bayou was designed to support Few real time Collaborative applications.

- Bayou targets machines with
  - Expensive Connection Time
    - Mobile Handsets,
    - PDA's
  - Frequent of occasional disconnections.
    - Cellular telephony

PROPERTIES

- Replicated , Weakly Consistent Storage system.
- Designed for Mobile Computing Environment.
- High availability
- Novel Methods for Conflict Detection
- Defines protocol that stabilizes Update Conflicts
- In fracture for collaborative applications
- Weakly Connected

CHALLENGES

- **Concurrent and conflicting updates**
  - Consequence of high availability
- **Dependency between operations**
- **Merge Procedures**
  - need to be defined flexible merge procedures to support wide range of applications
- **Write Propagation delays**
  - Asynchronous – no bounds on time
  - cannot be enforced strict bounds as it depends on Network Connectivity.

COLLABORATIVE APPLICATIONS

- Bayou replicated storage system was designed to support variety of Real Time Collaborative applications such as :
  - **Meeting Room Scheduler**
  - **Mail and Bibliographic Data Bases**
  - Shared Calendars
  - Program Development
  - Document Editing

Collaborative Applications: Meeting Room Scheduler

- Allows Users to reserve a room.
- At most one person can reserve the room at any given point of time.
- Users interact with a Graphical Interface.
- Scheduler periodically re-reads room schedule and refreshes the users display.

- **Problem:**
  - Users reservation might be out of date wrt. confirmed reservation.

Collaborative Applications: Meeting Room Scheduler

- User can select several acceptable meetings.
- Only one requested time will eventually be reserved.
- Users reservation will not be confirmed immediately.
  - Initially it will be Tentative (Grayed out)
- Users though disconnected from the rest can immediately see the others tentative room reservation.

Collaborative Applications: Bibliographic Data Base

- Allows users to add entries to a Data Base.
- Can read and write any copy of the Data Base.

- **Approach:**
  - Each Entry has a unique key.
  - Entry key is tentatively assigned when entry is added.
  - Users must be aware of the concurrent updates and should wait till the key is confirmed.

- **Problem:**
  - Same Bibliographic entry may be added by the Different user with different Key.

- **Solution:**
  - System Detects Duplicates and merges into a single entry with a single key.

EPIDEMIC ALGORITHMS

- Randomized algorithms for distributing updates and driving the replicas towards consistency.
- Ensures that the effect of every update is eventually reflected in all replicas.
- Efficient and Robust and Scale gracefully.
- Factors considered in designing an efficient algorithm:
  - Time
  - Network traffic.
- Bayou Design Uses Epidemic Algorithms for Conflict detection and Resolution.

EPIDEMIC ALGORITHMS….. STRATEGIES USED

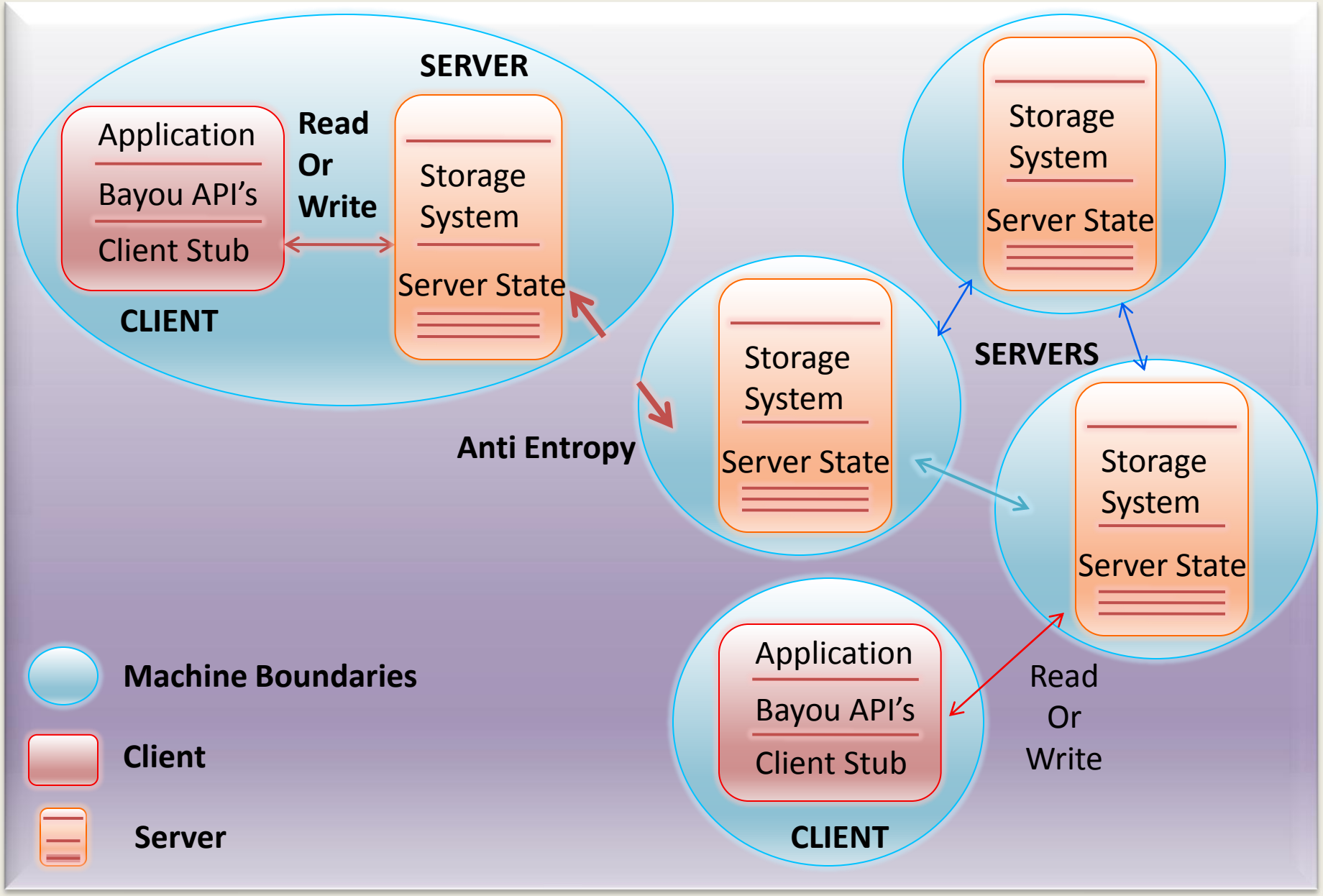
- **Different Strategies can be Uses for spreading updates:**
  - **Direct Mail:**
    - Each New Update is immediately mailed from entry site to all other sites.
    - Timely and reasonably efficient.
  - **Anti Entropy:**
    - Every Site regularly chooses another site at random.
    - Extremely reliable:
    - Propagation of Updates are Slow

EPIDEMIC ALGORITHMS….. ANTI ENTROPY cont….

- Each site executes the Anti Entropy periodically.
- Anti Entropy Algorithm is very **expensive**,
  - It involves comparison of 2 complete copies of the data base.
- Alternate Approach: Maintain the **checksum** of its database.
- Checksums at different sites likely to disagree if:

  time required for an update to sent to all sites > Expected time between new Updates.

BAYOU BASIC SYSTEM MODEL



BASIC SYSTEM MODEL Cont…(2/4)

- Each Data Base is replicated in Full at a number of servers.
- Client Application interact with the servers through the Bayou's API's.
- API's and Underlying Client Server RPC protocol supports 2 basic Operations:
  1. **Read**
  2. **Write (Insert/ Modify/ Delete)**
- Access to one server is sufficient for the client.
- Read Any/ Write Any.
- Bayou provides Session Guarantees.
- Bayou write carries information that lets each server receiving the write decide if there is a conflict and if so how to fix it.`

Basic System Model Cont…(3/4)

- Global Unique Write ID
- Storage system consists of Ordered log of writes + Data
- Each Server performs
  - Conflict Detection and Resolution.
  - write locally.
- Bayou server propagates writes among themselves during a pair wise contacts called **Anti-entropy session.**
- **Theory of Epidemic Algorithm:** As long as the set of servers are not permanently portioned, each write will eventually reach all servers.

Basic System Model Cont… (4/4)

- In the absence of new writes from the client, all Servers will hold the same data.

- The rate at which servers reach convergence depends on several factors:
  - Network Connectivity
  - Frequency of Anti Entropy
  - Policies by which servers select Anti-Entropy partners.

Basic Implementation

- Bayou System includes two mechanisms for automatic Conflict detection and Resolution.
- Mechanisms will support Arbitrary applications.
- Dependency check and Merge Procedures are Application specific.
- There are 2 States of an Update:
  - Committed
  - Tentative

CONFLICT DETECTION AND RESOLUTION

- **What is a conflict?**
  - Its Application Specific.
  - Different Application has different notion for what it means for two applications to conflict.

- **How to Identify Conflicts?**
  - It cannot be identified by simply observing conventional reads and writes.
  - Application should specify its notion of conflict.

- **How to Resolve Conflicts?**
  - Application should specify the policy for resolving the conflicts.
  - Depending on the specified Conflict policy storage system will specify the mechanism for reliably detecting conflicts and automatically resolve

Conflict detection and Resolution E.g. Application Specific Conflict

- **Meeting Room scheduling conflict**
  - Two users have concurrently updated two replicas of the meting room calendar and scheduling for the same room.

- **Bibliographic Data Base:**
  - Two users of Application describe different publications but have been assigned the same key by their submissions.
  - Two users describe the same publication but have been assigned the different keys.

CONFLICT DETECTION AND RESOLUTION…Cont…(3/3)

- Conflict Detection varies according to schematics of the application.
- An application must specify its notion of conflict along with its policy for resolving the conflicts.
- Depending on the specified conflict policy, storage systems specify the mechanisms for reliably detecting conflicts and automatically resolving conflicts.

MECHANISMS

- The Bayou system indicates two mechanisms for automatic conflict detection and Resolution.
  - **DEPENDENCY CHECK**
  - **MERGE PROCEDURES**

- Mechanism permits clients to indicate the following for each individual write operation:
  - How the system detects the conflicts involving the write
  - What steps should be taken to resolve the conflicts based on the schematics of the application.

DEPENDENCY CHECK

- Dependency check is the pre-condition for performing the update that is included in the write operation.
- Application Specific conflict detection is accomplished.
- Each write operation Includes a dependency check on Application supplied query and expected result.
- If the dependency check fails then the requested update is not performed.

Bayou Write Operation

```
Bayou Write (update,  dependency Check,  merge-procedure)
{
    if (DB_Eval(dependencyCheck.query) ==
            (dependencyCheck.result)  )
    {
            resolved_Update = Interpret and Execute merge-procedure;
    }
    else
    {
            resolved_Update =  Update the Data Base;
    }
    DB_Apply (resolved_update);
}
```

Sample Bayou – Write Operation

```
Bayou_write(
    update = {Insert, TableName, ScheduleTime,  "Comment"},
    dependency check =
{ query = "Select key from Meetings Where DAY = 12/18/09  AND START <2:30 AND END >
    1:30 PM;
            Expected Result = EMPTY },
    Merge procedure =
        { Alternatives = {{12/18/2009,3:00pm},{12/19/2009, 9:00pm}
          newUpdate = { },
         FOREACH a in alternatives
                {# check for conflict
                 if(NOTEMPTY (Select key from Meetings Where
                        DAY = 12/18/09 AND start < a.time +60 AND  END > a.time))
                        CONTINUE;
        # NO CONFLICT, CAN SCHEDULE MEETING AT THAT TIME SO INSERT IT TO newUpdate
        new Update = {insert, Meetings, a.data, a.time, 60 mins, "Budget Meeting"}
        }
        IF(newUpdate == EMPTY)
                newUpdate = {INSERT ERROR LOG};
    RETURN  newUpdate;
}
}
```

DESIGN OVERVIEW

❑ Merge Procedures:

- If a Conflict is detected then Merge procedure is run by Bayou Server to resolve it.
- Written in High Level Interpreted Language by application programmer in the form of template.
- Can read Current State of Servers Replica.
- Produces a revised Update.

DESIGN OVERVIEW Cont…

- **What if Automatic Conflict resolution is not possible?**
  - Server will log the conflict, which will be used later by the user to resolve the conflict.
  - Using Interactive Merge tool the conflicting Updates will be presented to the User
  - Bayou will not lock the file or file volume.
- **Replica Consistency:**
  - Bayou Guarantees that all servers eventually receive all writes via pair anti-entropy process.
  - Writes are performed in the same well defined order at all servers.
  - Conflict Detection and Merge Procedures are deterministic.

DESIGN OVERVIEW- Cont….(Write State)

1. **Tentative State :**
   - ❑ Initially when the write is accepted by the Bayou Server.
   - ❑ Tentative Writes are ordered according to the time stamps assigned to them by accepting system.

2. **Committed State:**
   - ❑ Eventually Each write is committed.
   - ❑ Committed writes are ordered according to the times at which they commit and before tentative writes.
   - ❑ Timestamps are monotonically increased at each server.
   - ❑ Pair of<Time Stamp, ID of the server that assigned it> produce a total order of write operation.

DATA BASE ORGANIZATION

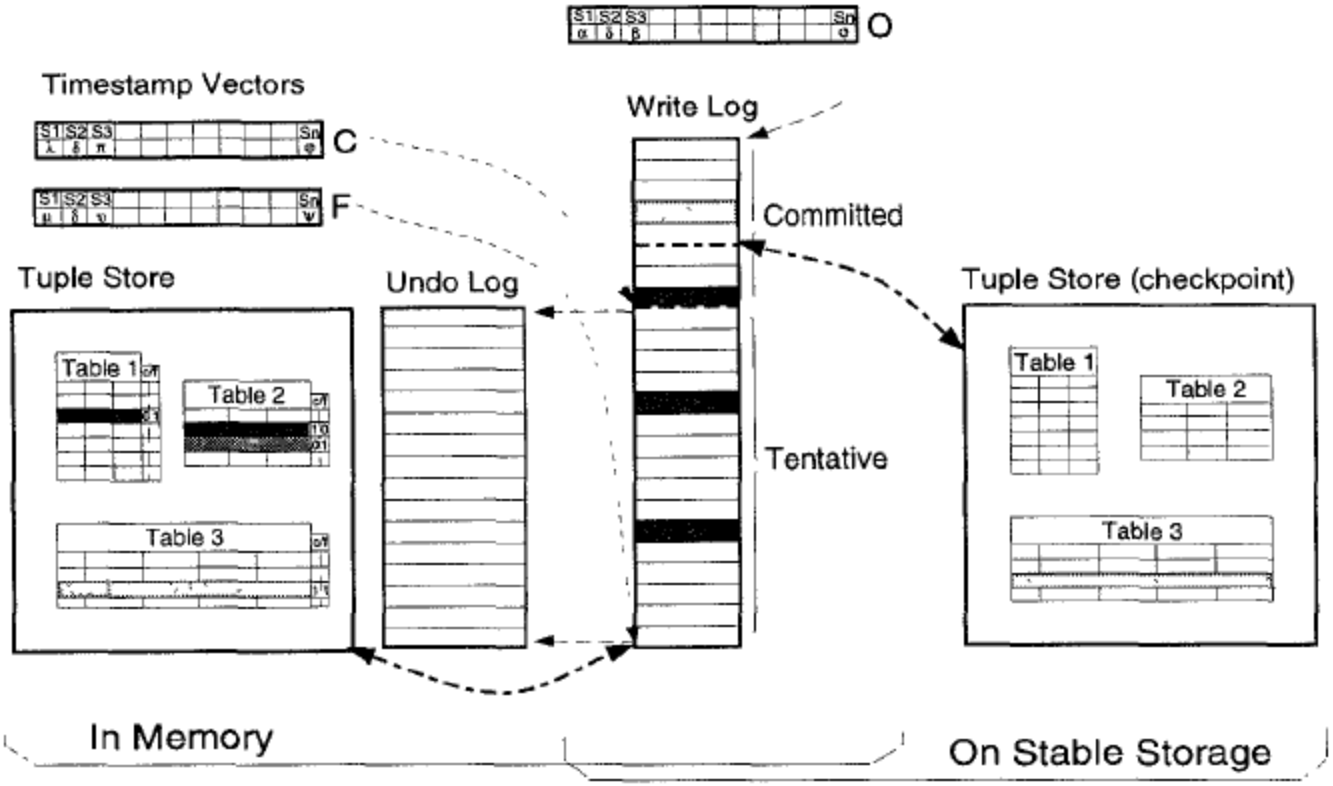


Figure 4. Bayou Database Organization

DATA BASE ORGANIZATION VECTORS

- **O Vector**

  Each server maintains this timestamp vector to indicate in a compact way the omitted prefix of the committed writes.

- **C Vector**

  Characterizes the state if the Tuple Store after executing the last committed Write in the Write Log

- **F Vector**

  Characterizes the state after executing the last tentative Write in the Write Log

- *Enable server pairs to identify the sets of writes that need to be executed*

- *These timestamp vectors are not used for conflict detection*

DATA BASE ORGANIZATION Cont…

- Data Base should support :
  1. Efficient Write Logging.
  2. Efficient Undo/ Redo Write Operations.
  3. Separate View of Committed and Tentative Data.
  4. Support for Server to Server Anti-Entropy.
  5. The Undo log should facilitate the rolling back of tentative writes that has been applied to the store.
  6. Two distinct views of Bayous data base (Committed and Full (C+T))
- Server can discard a write from write log once it becomes stable.
- Each server maintains a time stamp vector.
- The running state of each server includes two time stamp vectors that represent committed and full view.

ACCESS CONTROL

- A user may be granted:
  R/W privileges to data collection.
  Server privileges to maintain a replica.

- Mutual authentication and access control is based on public key cryptography.

- Every user possesses a public/private key pair and a set of digitally signed access control certificates.
- Types of certificates to grant, delegate and revoke access:

- AC[PU,P,D] certificate that grants privilege P on data D to user whose public key is PU
- D[PU,C,PY] certificate signed by users whose public key is PY to delegate his privileges encoded in certificate C to another user whose public key is PU
- R[C,PY] certificate signed by users whose public key is PY to revoke some user's privileges encoded in certificate C

EVALUATION - SETUP

- Server and Client for a bibliographic data base .
- Data Base contains a single table of 1550 tuples.
- Each tuple was inserted into DB with a single Write operation.
- Tested on 2 different Servers:
  - **1. Running on Sun SPARC/20**
  - **2. Gateway Liberty Laptop with Linux.**
- Language independent RPC package developed at Xerox PARC is used for communication between Bayou clients and servers.
- Results were presented for 5 different configurations of the DB characterized by the number of tentative writes.

EVALUATION RESULTS

Table 1: Size of Bayou Storage System for the Bibliographic Database with 1550 Entries
(sizes in Kilobytes)

Number of Tentative Writes	0 (none)	50	100	500	1550 (all)
Write Log	9	129	259	1302	4028
Tuple Store Ckpt	396	384	371	269	1
Total	**405**	**513**	**630**	**1571**	**4029**
Factor to 368K bibtex source	1.1	1.39	1.71	4.27	10.95

Table 2: Performance of the Bayou Storage System for Operations on Tentative Writes in the Write Log
(times in milliseconds with standard deviations in parentheses)

Tentative Writes	0	50		100		500		1550	
	Server running on a Sun SPARC/20 with Sunos								
Undo all	0	31	(6)	70	(20)	330	(155)	866	(195)
(avg. per Write)		.62		.7		.66		.56	
Redo all	0	237	(85)	611	(302)	2796	(830)	7838	(1094)
(avg. per Write)		4.74		6.11		5.59		5.05	
	Server running on a Gateway Liberty Laptop with Linux								
Undo all	0	47	(3)	104	(7)	482	(15)	1288	(62)
(avg. per Write)		.94		1.04		.96		.83	
Redo all	0	302	(91)	705	(134)	3504	(264)	9920	(294)
(avg. per Write)		6.04		7.05		7.01		6.4	

Table 3: Performance of the Bayou Client Operations
(times in milliseconds with standard deviations in parentheses)

Server Client	Sun SPARC/20 same as server		Gateway Liberty same as server		Sun SPARC/20 Gateway Liberty	
Read: 1 tuple	27	(19)	38	(5)	23	(4)
100 tuples	206	(20)	358	(28)	244	(10)
Write: no conflict	159	(32)	212	(29)	177	(22)
with conflict	207	(37)	372	(17)	223	(40)

Advantages

- Client can read or write to any replica without explicit coordination with other replicas.
- **Highly available:** Bayou will not mark conflict Data or System Unavailable.
- Bayou also provides support for clients that may choose to access only stable data.
- Dependency and Merge Procedures are more general than previous techniques.

Disadvantages

- No transparent, replicated data support for existing file systems and Data Base applications.

- Applications should :
  - Be aware that they may read weekly consistent data.
  - Be aware that their write operations may conflict with other applications.
  - Involve in detection of resolution of conflicts.
  - Should exploit domain specific knowledge

- **Expensive Merge Procedure**

Discussions

- Too Abstract!!
- Why only pair-wise communication?
- Global Strategies not possible during merging.
- Eventual Consistency!!
- When should we commit?

QUESTIONS

.

Thank You ☺