Department of Computing

**Hong Kong Polytechnic University**

**Comp 5527 Mobile Computing and Data Management**

**Tutorial Five Sample Solutions**

Q1: Sample Solutions:

a)

In anti-entropy protocol, every site chooses another server at random and by exchanging database records with it resolves any differences between the two servers. Anti-entropy is extremely reliable but requires examining the contents of the database and so cannot be used too frequently.

Given there is a set *S* of *n* servers and each stores a copy of a database. Each record in the database has the format of *r*(*k, v, t*) where *k* is the key, *v* is the value and *t* is the timestamp.

For anti-entropy protocol, the following algorithm periodically executed at each site *s*:

FOR Some s' ∈ *S DO*

ResolveDifference(*s, s'*)

ENDFOR

Depending on its design, its effect may be expressed in one of three ways, called *push*, *pull* and *push-pull*:

ResolveDifference (*s, s'*)= { -- *push*

FOREACH record *r* and *r’* in *s* and *s’* respectively with the same *k*

IFr.t > r'.t THEN

r'r

ENDIF

ENDFOR

}

ResolveDifference (*s, s'*)= { -- *pull*

FOREACH record *r* and *r’* in *s* and *s’* respectively with the same *k*

IFr.t < r'.t THEN

rr'

ENDIF

ENDFOR

}

ResolveDifference (*s, s'*)= { -- *push-pull*

FOREACH record *r* and *r’* in *s* and *s’* respectively with the same *k*

SELECT TRUE FROM

r.t > r'.t  r'r;

r.t < r'.t  rr';

ENDFOR

}

b)

In direct mail: each new update is immediately mailed from its local server to all other servers.

The following codes are executed at a site *s* where an update occurs:

FOREACH *s'* ∈ *S*

PostMail[to: *s',* msg : ("Update", *r*)]

ENDFOR

Upon receiving the message ("Update", *r),* site *s* executes

Find the record *r’* such that *r.k=r’.k*

IF *r’.t <r.t* THEN

*r'r;*

ENDIF

c)

* Direct mail is timely and reasonably efficient, but not entirely reliable since individual sites do not always know about all other sites and since mail is sometimes lost.
* Anti-entropy progagates updates much more slowly than direct mail.
* Anti-entropy is more reliable. Anti-entropy can be structured as an incremental process so that even servers with very intermittent connections can eventually bring their databases into a mutually consistent state.

Q2: Sample Solutions:

A pessimistic approach requires a client to acquire shared or exclusive control of a cached object prior to disconnection, and to retain such control until reconnection. Possession of exclusive control by a disconnected client would preclude reading or writing at all other replicas. Possession of shared control would allow reading at other replicas, but writes would still be forbidden everywhere.

Acquiring control prior to voluntary disconnection is relatively simple. It is more difficult when disconnection is involuntary, because the system may have to arbitrate among multiple requesters. Unfortunately, the information needed to make a wise decision is not readily available. For example, the system cannot predict which requesters would actually use the object, when they would release control, or what the relative costs of denying them access would be.

Retaining control until reconnection is acceptable in the case of brief disconnections. But it is unacceptable in the case of extended disconnections. A disconnected client with shared control of an object would force the rest of the system to defer all updates until it reconnected. With exclusive control, it would even prevent other users from making a copy of the object. An entire user community may wait for a single client for an unbounded amount of time.

Placing a time bound on exclusive or shared control, as done in the case of leases, avoids this problem but introduces others. Once a lease expires, a disconnected client loses the ability to access a cached object, even if no one else in the system is interested in it. This, in turn, defeats the purpose of disconnected operation which is to provide high availability. Worse, updates already made while disconnected have to be discarded.

An optimistic approach allows reading and writing to cached objects if the client is disconnected. The updates will be propagated to the servers when the client reconnects to the network. This supports high availability.

An optimistic approach also has its own disadvantages. An update made at one disconnected client may conflict with an update at another disconnected or connected client. There needs to be machinery in the system for detecting conflicts, for automating resolution when possible, and for confining damage and preserving evidence for manual repair. Having to repair conflicts manually violates transparency, is an annoyance to users, and reduces the usability of the system.

Coda is designed for an environment consisting of a large collection of untrusted clients and a much smaller number of trusted file servers. It is specifically not intended for applications that exhibit highly concurrent, fine granularity data access. Optimistic approach is chose because it is suitable for this situation.

Q3. Sample Solutions

The system architecture consists of the following entities:

* + the location-based application provider,
  + the mobile positioning service provider,
  + the location control center, and
  + the system components involved with implementing a particular positioning technology (wireless network infrastructure, BSs, MSs)

The location based application provider will subscribe to the mobile positioning service provider, which is often the wireless network operator.

Upon a request from a subscriber for location information about an MS, the mobile positioning service provider will contact a location control center, querying it for the location of the MS.

The location control center will gather information required to compute the MS’s location. A set of BSs could be used to page the MS, and directly or indirectly obtain the location of the location parameters (such as TOA of signals).

If a handset-based method is used, the MS determines its own position and then transmit this information to the location center (maybe through one of the BSs).

Once this information is collected, the location control center can determine the location of the MS with certain accuracy and convey this information to the service provider.

The service provider will then return this information or use this information to visually display the MS’s location to the subscriber.

The subscriber will use this information in the location based application to provide service to the mobile user.