Location Management in Mobile Computing

Case Studies

General Comments

Ajay Gupta
WMU-CS

Part taken from Chrysanthis and Pitzoura, ICDE 2000

Locating Moving Objects

† Examples of moving objects
  • mobile devices (cars, cellular phones, palmtops, etc)
  • mobile users (locate users independently of the device they are currently using)
  • mobile software (e.g., mobile agents)

† How to find their location - Two extremes
  • Search everywhere
  • Store their current location everywhere
    ▶ Searching vs. Informing

Locating Moving Objects

† What (granularity), where (availability) and when (currency) to store

Granularity

Availability

Exact location

some portion

the whole network

nowhere

Never update

Always update (at each movement)

Granularity

Availability

currency

nowhere

grain

Moving Objects

Architectures of Location DBs

† Two-tier Schemes (similar to cellular phones)
  • Home Location Register (HLR): store the location of each moving object at a pre-specified location for the object
  • Visitor Location Register (VLR): also store the location of each moving object at a register at the current region

† Hierarchical Schemes
  • Maintain multiple registries
Two-tier Location DBs

- Search
  - Check the VLR at your current location
  - If object not in, contact the object’s HLR

- Update
  - Update the old and new VLR
  - Update the HLR

Hierarchical Location DBs

Maintain a hierarchy of location registers (databases)
A location database at a higher level contains location information for all objects below it
Hierarchical vs. Two-tier

(+): No pre-assigned HLR
(+): Support Locality
(-): Increased number of operations (database operations and communication messages)
(-): Increased load and storage requirements at the higher-levels

Locating Moving Objects

Caching
- cache the callee’s location at the caller
  (large Call to Mobility Ratio)

Replication
- replicate the location of a moving object at its frequent callers
  (large CMR)

Forwarding Pointers
- do not update the VLR and the HLR, leave a forwarding pointer from the old to the new VLR (small CMR)
- When and how forwarding pointers are purged?
- Concurrency, coherency and recovery/checkpointing of location DBs

Querying Moving Objects

- Besides locating moving objects, answer more advanced queries, e.g.,
  - find the nearest service
  - send a message to all mobile objects in a specific geographical region
- Location queries: spatial, temporal or continuous
- Issues: representation, evaluation and imprecision

Most current research assumes a centralized location database
Querying Moving Objects

How to model the location of moving objects?

Dynamic attribute (its value changes with time without an explicit update) [e.g., in MOST]

For example, dynamic attribute $A$ with three sub-attributes: $A.value$, $A.updateTime$ and $A.function$ (function of a single variable $t$ that has value 0 at time $t=0$)

- The value of $A$ at $A.updateTime$ is $A.value$
- At time $A.updateTime + t0$ is $A.value + A.function(t0)$

How to represent and index moving objects?

- Spatial indexes do not work well with dynamically changing values
- Value-time representation
  - An object is mapped to a trajectory [Kolios 99]

Case Studies
Case Study: Coda

- Client-Server System with two classes of replication w.r.t. consistency
  - Disconnected vs. Weakly connected
    - Hoarding, Caching/Server callback, No Prefetching
  - During connections: Allows AFS clients (Venus) to hoard files.
    - Hierarchical, prioritized cache management → equilibrium.
    - Track dependencies, bookmarks
  - During disconnections: Venus acts as (emulates) a server
    - Generates (temp) fids, services request to hoarded files.
  - On reconnection, Venus integrates locally changed files to servers.
    - Considers only write-write conflicts - no notion of atomicity
    - User conflict resolution/ Application-aware adaptation [Odyssey]
    - Uses optimistic replication technique

Coda Client Space Management

- Space requirements - 10MB
  - Space for hoarding applications
  - Space use during Emulation (in particular logging)
  - Space for Recoverable Virtual Memory (cache directory, symbolic link, status of block etc.)
- Free disk space techniques
  - Compression of file cache and RVM (space vs. computation time)
  - Abort updates made by users (reduce log space)
  - Allow file cache and RVM to be copied to flash cards/floppy disks.

Case Study: Consistency in Bayou

- A bottom-up approach to specific design problems
  - More distributed than coda, more emphasis on "small" clients
- Key features:
  - Read-any/write-any to enhance availability
  - Anti-entropy protocol for eventual consistency
  - Dependency checks on each write
  - Application-specific resolution of update conflicts
  - Primary server to commit writes and set order
  - Session consistency guarantees
- How effective is anti-entropy?

Anti-entropy Protocol

- Server propagates write among copies.
- Eventually all copies "converge" towards the same state.
- Eventually reach identical state if no new updates.
- Pair-to-peer anti-entropy
  - Each server periodically selects another server
  - Exchange writes and agree on the performed order
  - Reach identical state after performing the same writes in the same order.
Case Study: Rover

- Rover [Joseph 97] provides an environment for the development of mobile applications.
- Applications are split into client and server part communicating with Queued RPCs.
- Application code and data are encapsulated within Relocatable Dynamic Objects (RDOs).
- Access Managers at client and server handle RDOs.
- Client’s operational log is lazily transferred to the server.
- Disconnections are supported by the local cache.
- Some support for primary copy, optimistic consistency.

Case Study: Pro-Motion

- Pro-Motion [Chrysanthis 97] is designed for the development of mobile database applications.
- It shares similar architecture as Rover with a multi-tier C-I-S model.
- Compact is the unit of caching and hoarding.
  - It encapsulates cached data, methods, consistency rules and obligations (e.g., deadlines).
- Supports both tentatively committed transactions and two-tier replication.

Case Study: Rome

- Rome [Fox 99] goal is the timely and in context delivery of information.
- Information should be received when and where it is needed.
- Its fundamental building block are the triggers:
  - pieces of data bundled with contextual information
  - Condition: (location ∈ R) ∧ (time > t) ⇒ action
- It is similar to active databases but with decentralized management.
- It provides an extensible framework and building blocks leveraging on internet service.

Mobile Access to the Web

- Three-tier Architectures: Client - Web Server - Data Server
- Web Server can act like a server-side agent.
  - Prefetching at its cache can hide some latency.
  - Scripts at the Web server can perform user-specified filtering and processing.
- Most solutions use a Web proxy to avoid any changes to the browsers and servers.
  - Pythia [Fox96]
  - Mobile Browser (MOWSER) [Joshi 96]
    - Distillation: highly lossy, real-time, datatype specific compression that preserves semantic content
  - WebExpress [Housel 97]
WebExpress

- Utilizes the C-I-S Model
- Goals: reduce traffic volume and reduce latency
- Intercept any http request and perform four optimizations:
  - Caching at both CSA & SSA of graphics and html objects
  - Differencing: only changes are communicated
  - Long-live TCP/IP Connection: CSA & SSA use a single TCP connection
  - Header reduction: SSA includes the required browser capabilities. They are not sent by the CSA.
- While disconnected (off-line mode) uses CSA cache

Advances in Mobile Web Servers

- W4 for Wireless WWW [bartlett 94]: Mosaic on PDA
- Dynamic Documents: Tcl / Xml scripts that execute within the mobile browser to customize the html documents
- Dynamic URLs [Mobisaiac 94]: They support mobile web servers and work with active pages.
- IPIC [Shrinivasan 99]: A match head sized web server

Mobility Middleware in the Market

- Most middleware market are based on TCP/IP and socket-oriented connections
- Wireless-friendly TCP versions have been proposed but no major products have adopted it
- Microsoft's Remote Access supports cellular communication by integrating Shiva's PPP suite
- Shiva's PPP (Point-to-Point protocol) suit provide a remote access client to either wired or mobile servers
  - E.g., mobile clients can access Tuxedo transaction services
- MobileWare Office Server: An agent-based middleware that supports Lotus Notes, Web access, database replication, etc.
  - Connection profiles, checkpointing, compression, security

State of Mobile DB Industry

- Sybase SQL Remote (Sybase SQL AnyWhere)
  - MobiLink: Centralized model to control replication
  - Application-specific bi-directional synchronization using scripts
  - UltraLite: in-memory dbms (50KB)
- ORACLE
  - Oracle Mobile Agents middleware
  - Oracle 8 Lite: supports bi-directional replication between a client and a server (50-750KB)
  - Oracle Replication Manager: supports replication across multiple servers based on the peer-to-peer model
- MS SQLServer
  - Merge replication and conflict resolution
  - Alternative clients: Outlook and MS ACCESS
- IBM DB2 Everywhere (100KB)
Unsolved Problems

- Integration and evaluation of algorithms with applications
- Broadcast disks
  - Information update/consistency and data temporal coherence - meet time constraints of requests
  - Relation between server broadcasting and client caching.
- Multiple broadcast channels and multiple database access
- Efficient, scalable, adaptive mechanisms
  - Location handling
  - Trigger management
- Programmer Interface for Application-aware adaptation
  - Data fidelity vs. consistency
  - Semantic consistency needs metadata/requirements
- Multimedia and QoS

True?

Mobile and wireless computing attempts to deliver today's and tomorrow's applications on yesterday's hardware and communication infrastructure!