Mobile Computig: Conclusions

Evaggelia Pitoura

Computer Science Department,
University of Ioannina,
Ioannina, Greece
http://www.cs.uoi.gr/~pitoura

Summer School, Jyvaskyla, August 1998

Mobile Computing Models

- What is the best way to partition a computation as well as the functionality of a system or application between stationary and mobile elements?

Adaptivity: the functionality assigned to a mobile element must be made dependent on the resources currently available to it and on the present connectivity conditions.

Mobile Computing Models

- At which level should mobility be handled? What are the primitives, if any, that an operating system, a database management system, or an application must provide for treating mobility? Should the user be aware of any environmental changes?

Support for mobility at multiple levels, e.g., through a cooperation among the underlying systems and applications.

Need to effectively monitor: (a) location changes, and (b) changes in the availability of resources

In attaining resource and location data, such a monitoring process cooperates with low-level system routines. Since this process is continuously active, its efficient implementation is not trivial.

In addition, an event-delivery mechanism to convey information about changes in the environment to interested applications.

Finally, define and implement primitives primitives for an application to

(a) communicate information related to its semantics and resource needs (e.g., level of acceptable consistency) to the underlying system, and

(b) exercise control over the system's support for mobility (e.g., overwrite file-level prefetching)

Mobile Computing Models

- What is an appropriate model for mobile computing?

Mobile Agents

Provide constructs and primitives for moving computations

Adaptivity is handled by dynamically relocating computation on and off a mobile host.

Such constructs are provided either by the operating system (e.g., Tacoma[2]), a programming language (e.g., Telescript[4], Obliq[1]) or a programming environment (e.g, Rover[3]). These constructs can be used to design and implement software for mobile computing at multiple levels.

Agent models must provide primitives at least for

(a) moving computation and data across different sites of the network,

(b) replicating computation and data at different sites, and

(c) invoking operations on data located at remote sites.

Agents should also encapsulate information related to their consistency and currency semantics. For instance, resolution procedures to be applied in case of conflicting operations, or consistency conditions such as those in quasi copies.

Mobile Computing Models

– *Extend the client/server models with agents/proxies*

to handle weak connectivity and offload computation from the mobile client to the fixed network.

Proxies handle connectivity at various levels, e.g., at the transport level or at the file system level.

Techniques to handle weak connectivity include queuing incoming and outgoing messages, filtering information, compressing, batching together multiple requests, and reordering requests based on priorities.

Proxies can perform part of the computation in lieu of their clients.

Proxies may either be static or follow the majority of their clients

When agents are static, mobility of clients is handled by replicating servers or proxies and attaching the client to the server/proxy located in its current neighborhood.

Mobile Computing Models

Summary

Issue	Appoach
Mobility of hosts	Replicate servers or server-side static agents and attach the client to the nearest agent/server Move the server-side agent
Variability	Adaptivity: Dynamic partition of data and computation, using for instance mobile (relocatable) agents
Weak connectivity	Insert intercept agents at both sides to: - queue messages - filter/compress information - reorder/prioritize messages - perform caching, etc.
Limited resources	Insert proxy/surrogate agent at the fixed network to offload functionality from the mobile host

Disconnected Operation

Which data and functionality to hoard?

A promising approach is to relate hoarding in databases to view materialization.

How to maintain consistency?

Optimizing the size of the log is critical in terms of performance and memory capacity for database operations

How to integrate the results

By re-executing the operation log at the fixed host

Weak Connectivity

In *file systems*: (a) cache updates are deferred and transmitted in the background; (b) cache misses are queued, or selectively serviced; (c) the server sends invalidation reports massively at various degrees of granularities, e.g, files or sets of files.

Adaptive to the available bandwidth by varying: (a) the frequency of transmitting updates, (b) the number of misses serviced or queued, (c) the granularity at which cache invalidation is performed.

– Weak connectivity in *database systems*?

What part of the database functionality to assign to mobile hosts.

Database fragmentation and fragment placement.

Mobile transaction model: an open-nested transaction model.

Other System-Level Issues

Broadcast

Use broadcasting to:

- transmit cache invalidation reports to clients.

- in concurrency control or other transaction management protocols

Mobility

At the system level, mobility in replicated file systems is handled by letting a client access copies from the servers located closest to it.

Mobility also affects algorithms for placing replicas

Relocating transactions in progress involves transmitting state-related information including locks or timestamps.

System-Level Support

Summary

Issue	Approach
Disconnections	Replicate/cache data at the mobile host Move/relocate data to the mobile host
Weak connectivity	Revise cache policy to: - selectively serve or queue cache misses - asynchronously propagate cache updates to the server - asynchronously propagate cache invalidations to the client and change the cache vaildation granularity Revise coherency control for replication: - read-any/write-any copy and propagate updates asynchronously - bound the divergence among copies
Variability	Adaptivity: Make the techniques for weak connectivity adaptive to the currently available bandwidth
Mobility	Dynamic replica placement Transparently switch to the closest server
Broadcast	Propagate cache invalidation reports via broadcast Revise cache policies for broadcast-based data delivery Convey transaction management related information via broadcast

Query Processing

Location

location data (i.e., data items that hold information about the location of moving elements)

– requires cooperation with low-level (e.g., data-link or network layer) routines that maintain location information.

– fast changing

– have both a spatial and a temporal dimension

data models that support dynamic attributes, e.g., attributes whose value changes with time.

and

new techniques for indexing them.

Approximate information: query processing may involve data acquisition or provide an approximate answer.

Disconnections and Weak Weak Connectivity

that take into account the fact that some sites may be only intermittently connected to the fixed network

Query Processing

Database interfaces for posing queries and representing their results using small screens

Broadcast

Instead of posing queries, clients listen to the broadcast channel and tune in to get the information they need.

In a sense, instead of posing queries to the server, clients "filter" the broadcast.

Techniques for structuring and indexing the broadcast information

Combined with on-demand delivery, in which case clients can also directly send requests for data to servers.

Deriving optimal query execution plans than involves both accessing the server and filtering the broadcast.

Query Processing

Issue	Approach
Mobility of hosts	Location of moving users is changing fast: - store location as a dynamic attribute - derive appropriate techniques for indexing dynamic attributes - querying location data may give approximate answers - querying may involve data acquistion
Disconnections	Derive query execution plans that sustain disconnections
Weak connectivity	Derive query execution plans that minimize the deployment of the wireless link
Limited client resources	Use semantics and icons to pose queries and represent results on small screens
Limited battery power	Derive power-efficient query plans
Broadcast	Use broadcast for data delivery Issues include: - indexing the broadcast items - determining the structure of the brodcast (e.g., how frequently is an item broadcast) - handling updates Combine broadcast and on-demand delivery Issues include: - determining which data to broadcast - sharing the channel - querying

Locating Mobile Users

Various data management techniques can be used to enhance location schemes: e.g., cache or replication

Further: cache replacement policies and/or prefetching

dynamic algorithms for replica placement can be deployed to determine the sites of replication

Concurrency control and failure recovery

References

[1] L. Cardelli. A Language with Distributed Scope. *Computing Systems*, 8(1):27–59, 1995.

[2] D. Johansen, R. Renesse, and F. B. Schneider. Operating System Support for Mobile Agents. In *Proceedings of the 5th Workshop on Hot Topics in Operating Systems (HOTOS-V)*, May 1995.

[3] A. D. Joseph, J. A. Tauber, and M. F. Kaashoek. Mobile Computing with the Rover Toolkit. *IEEE Transactions on Computers*, February 1997.

[4] J. E. White. Mobile Agents. General Magic White Paper, www.genmagic.com/agents, 1996.