Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2016, 3(3):669-679**

Review Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Intelligent Adaptation of Client Server Computing in Mobile Environment

Shashi Shekhar Mishra[1], Kushal Johari[2], Prasoon Johari[3]

[1]Department of Computer Application, Ch. Charan Singh University Campus, Meerut 250005, India
[2]Department of Computer Application, Invertis University, Bareilly, India
[3]Department of Computer Science, Future Group of Institution, Bareilly, India

Abstract Recent advances in wireless data networking and portable information appliances have engendered a new paradigm of computing, known as mobile computing; in which users carrying portable devices have access to data and information services regardless of their physical location or movement behaviour. As Mobile Computing becomes more dominant, system and applications must deal with a growing disparity in resources types and availability of the user interface device. Although the challenges of effectively managing these resources are partially addressed by the existing system, their solutions are hindered by the system's lack of knowledge about the application. In the recent time, research addressing information access in mobile environments has proliferated. In this survey, we provide a concrete framework and categorization of the different ways to support mobile client-server computing for information access. We examine characteristics of mobility that distinguish mobile client-server computing from its traditional computing.

Keywords Mainframe, Client/Server, Application Adaptation, Mobile Computing

Introduction

The mainframe persists as a highly capable platform for corporate computing. Its performance characteristics and cost-of-ownership advantages over alternative computing platforms are increasingly understood and appreciated. From the standpoint of cost, compliance, continuity and carbon footprint (utility power consumption), the value case for the mainframe has reclaimed its former mindshare, especially in light of current economic realities and the prevailing regulatory climate [1]. Yet, in many companies, plans to leverage this technology over the long term are challenged by three trends:

A. New workloads are being added to the traditional mainframe platform, requiring changes to the underlying technology itself. This has led to the proliferation of product or component-specific software tools for managing new hardware and software elements. The requirement to master multiple management tools is consuming an inordinate amount of the daily work schedules of seasoned mainframers and reducing the time they have available for strategic work.

B. Due to retirements and downsizing, core mainframe staffers are required to shoulder responsibilities individually that were once spread over several domain experts. In the process, traditional job descriptions have blurred, leading to the potential erosion of operational efficiency and managerial discipline that have been the hallmark of mainframe computing.

C. The average age of a mainframer is 53 years. This fact, combined with low enrolments in technology-oriented college degree programs generally, portends a shortage in qualified staff to replace retiring IT practitioners over the next decade including mainframe staff. Addressing this skills shortage will require increased attention to on-the-job training and mentoring.

Client Server Computing

According to MIS terminology, Client/Server computing is new technology that yields solutions to many data management problems faced by modern organizations. The term Client/Server is used to describe a computing model for the development of computerized systems. This model is based on distribution of functions between two types of independent and autonomous processes: Server and Client [2]. A Client is any process that requests specific services from the server process. A Server is a process that provides requested services for the Client [2]. Client and Server processes can reside in same computer or in different computers linked by a network. When Client and Server processes reside on two or more independent computers on a network, the Server can provide services for more than one Client. In addition, a client can request services from several servers on the network without regard to the location or the physical characteristics of the computer in which the Server process resides. The network ties the server and client together, providing the medium through which the clients and the server communicate.

Mobile Computing

Mobile computing is the discipline for creating an information management platform, which is free from spatial and temporal constraints. The freedom from these constraints allows its users to access and process desired information from anywhere in the space. The state of the user, static or mobile, does not affect the information management capability of the mobile platform. A user can continue to access and manipulate desired data while travelling on plane, in car, on ship, etc. Thus, the discipline creates an illusion that the desired data and sufficient processing power are available on the spot, where as in reality they may be located far away [3].

The discipline of mobile computing has its origin in Personal Communications Services (PCS). PCS refers to a wide variety of wireless access and personal mobility services provided through a small terminal (e.g., cell phone), with the goal of enabling communications at any time, at any place and in any form. PCS are connected to Public Switched Telephone Network (PSTN) to provide access to wired telephones. PCS include high-tier digital cellular systems for widespread vehicular and pedestrian services and low-tier telecommunication system standards for residential, business and public cordless access applications [3].

- High-tier digital cellular systems include:

1. Global System for Mobile Communications (GSM)
2. IS-136 TDMA based Digital Advanced Mobile Phone Services (DAMPS)
3. Personal Digital Cellular (PDC)
4. IS-95 CDMA-based CDMA One System

- Low-tier telecommunication systems include:

1. Cordless Telephone 2 (CT2)
2. Digital Enhanced Cordless Telephone (DECT)
3. Personal Access Communication Systems (PACS)
4. Personal Handy Phone Systems (PHS)

Several wideband wireless systems and special data systems have been developed to accommodate internet and multimedia services.

In mobile computing platform information between processing units flows through wireless channels. The processing units (client in client/server paradigm) are free from temporal and spatial constraints. That is a processing unit (client) is free to move about in the space while being connected to the server. This temporal and spatial freedom provides a powerful facility allowing users to reach the data site (site where the desired data is stored) and the processing site (the geographical location where a processing must be performed) from anywhere. This capability allows organizations to set their offices at any location.

It is interesting to learn some of the historical milestones achieved in mobile systems which appeared some time ago. The events can be listed as follows [4]:

April 7, 1928: The first mobile radio system went into operation in Detroit. It was used by the Detroit Police Department.

Year 1935: Frequency modulation has been developed and tested.
Year 1943: AT&T developed and introduced the Improved Mobile Telephone Service (IMTS). It consisted of a broadcast system with a higher-power transmitter. This system was followed shortly with limited cellular networks and the implementation of the first mobile radio system to connect with a fixed telephone number.
Year 1950s: Paging systems began to appear. During this period, Bell Labs continued to test the cellular techniques.
Year 1970: Federal Communication Commission (FCC) allocated spectrum space for cellular systems. At this time, AT&T proposed the cellular system that is now known as the Advanced Mobile Phone System (AMPS).
Year 1983: The cellular service was commercially implemented in Chicago and Baltimore.

Mobile Connectivity

The mobile connectivity between two nodes exists if they are continuously connected through wireless channel, and can utilize the channel without being subjected to spatial and temporal constraints.

Figure 1 illustrates the concept of a fully connected information space created through mobile connectivity mode where every unit can communicate to any other unit through wireless channel. The power of mobile connectivity has been recognized by the research community and also by the market. To fully exploit its inherent power the following challenges must be met [5]:

1. Revising the technical architecture- Mobile users are demanding. They are important to the business world. To provide complete connectivity among users the current communication technology must be revised to incorporate mobile connectivity. Additionally, application and data architectures must also be revised to support the demands put upon them by the mobile connectivity.
2. Reliability, coverage, capacity and cost- At present wireless network is less reliable, have less geographic coverage and reduced bandwidth, are slower, and cost more than the wired-line network services. It is important to find ways to use this new resource more efficiently by designing innovative applications.
3. Integration with legacy mainframe and emerging client/server applications- Application development paradigms are changing. As a result of the IT industry's original focus on mainframes, a huge inventory of applications using communications interfaces that are basically incompatible with mobile connectivity have been accumulated. Still the application development trend is geared towards wired network platform and little thought has been given to applications necessary for mobile platform. This practice must change for successful integration of mobile connectivity.
4. End-To-End design and performance- Since mobile computing involves multiple networks (including wired) and multiple application server platforms, end-to-end technical compatibility, server capacity design, and network response time estimates are difficult to achieve.
5. Security- Wireless networks have relatively more security requirements than wired network. A number of approaches have been suggested and also the use of encryption is has been proposed.

In addition to these technical challenges, mobile computing also faces business challenges. This is due to the lack of trained professionals to bring the mobile technology to the general people and development of pilot projects for testing its capabilities [5].

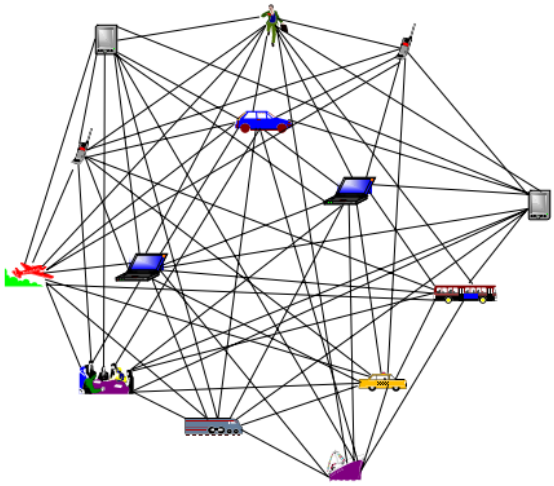

Figure 1: A nearly fully connected information space

To facilitate the data management activities, users can carry Personal Digital Assistant (PDA), laptop, cell phones, etc. At present the current technology only provides limited transaction processing capabilities but soon such facilities will be available on all mobile devices such as cell phones, laptops, palmtops, etc. Figure 1 illustrates the concept of a fully connected information space where every unit is fully connected to other units in the information space.

Architectures

Client Server Two Tier Architecture

This architecture is also called Client-Server architecture because of the two components: The client that runs the application and the server that handles the database back-end. When the client starts it establishes a connection to the server and communicates as needed with the server while running the client. The client computer usually can't see the database directly and can only access the data by starting the client. This means that the data on the server is much more secure. Now users are unable to change or delete data unless they have specific user rights to do so.

The client-server solution also allows multiple users to access the database at the same time as long as they are accessing data in different parts of the database. One other huge benefit is that the server is processing data that allows the client to work on the presentation and business logic only. This mean that the client and the server is sharing the workload and by scaling the server to be more powerful than the client, you are usually able to load many clients to the server allowing more users to work on the system at the same time [6].

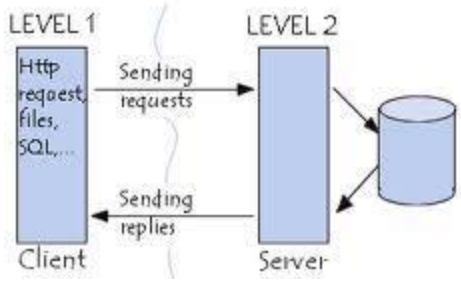

Figure 2: Two-Tier Client Server Architecture

Client Server Three Tier Architecture

This involves one more layer called the business logic tier, service tier or middle tier (layer). In the client-server solution the client was handling the business logic and that makes the client "thick". A thick client means that it

requires heavy traffic with the server, thus making it difficult to use over slower network connections like Internet and Wireless (LTE, 3G, or Wi-Fi). By introducing the middle layer, the client is only handling presentation logic. This means that only little communication is needed between the client and the middle tier making the client "thin" or "thinner". An example of a thin client is an Internet browser that allows you to see and provide information fast and almost with no delay. As more users access the system a three-tier solution is more scalable than the other solutions because you can add as many middle tiers (running on each own server) as needed to ensure good performance (N-tier or multiple-tier). Security is also the best in the three-tier architecture because the middle layer protects the database tier. Three-tier architecture is typically composed of a presentation tier, a business or data access tier, and a data tier [7].

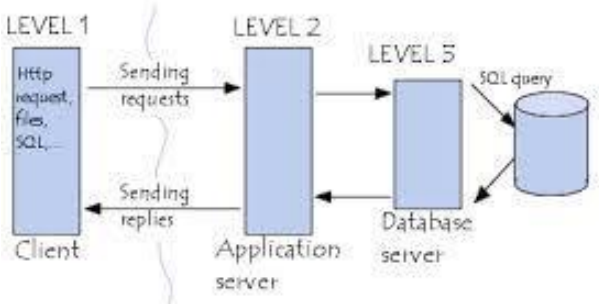

Figure 3: Three-Tier Client Server Architecture

Client Server N-Tier Architecture

Multi-tier architecture (often referred to as N-tier architecture) is a client–server architecture in which presentation, application processing and data management functions are logically separated. For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture. The most widespread use of multi-tier architecture is the three-tier architecture.

N-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application. There is one major drawback to the N-tier architecture and that is the additional tiers increase the complexity and cost of the installation.

Mobile Computing Architecture

This Architecture consists of mobility service architecture, describing the way we implement our mobility services in a computer system and mobility. Environment architecture describing how data are transmitted between Computers in mobile environments and what tasks the different stations fulfil our architecture.

Mobility Services Architecture

Mobility services can be classified into three groups. First there are services design to overcome common restrictions of Mobile computing, which arises mainly from the slowness, insecurity and instability of wireless or analogous connection lines utilized by the mobile user. These services are called common mobility services (CMS). Examples are connection management, caching or encryption services. The second group of services handles the management and administration of mobile users moving around and connecting their portables to networks at different places. These mobility management services (MMS) include tasks such as the authentication of users, accounting and billing issues or profiling of the users habits. The task necessary to adapt certain existing applications to mobile usage are implemented by high level services which are called special mobility services (SMS). Special mobility services adapt existing services to the mobile conditions [8]. For example to allow remote database access over a wireless connection line one has to take special care of possible frequent connection losses especially in the context of the state of the database.

Extended Client-Server Model

THIN Client

The prevailing models of mobile computing are predominantly based on the classical client/server architecture. This architecture is a logical model and not a physical implementation. It consists of three entities: the client, the

server and a communication channel, in most cases the network (a client can be invoked on a local server). A client is a process running in a machine (part of a fixed network) that requests service of the server which is another process running in the same or different fixed network. All the communication between these two parties is done through message passing which is the most important aspect of client-server computing.

While the most important aspect of client-server computing. The communication channel used in message passing is of two types: connection-oriented and connection-less. While the reliability is greater in the former, the overhead in establishing and maintaining connection is considerably high. In the latter, the reliability is low and might entail the parties of communication to retransmit lost messages. The TCP/IP and UDP protocols are examples of connection-oriented and connection-less communication models respectively. The client-server architecture requires that-

1. The client is always connected to the server

2. The server is highly reliable

3. The Network is fast and reliable.

FAT Client

Mobile clients must be able to use networks with rather unpleasant characteristics-intermittence, low bandwidth, high latency or high expense. The connectivity with one or more of these properties is referred to as weak connectivity. In the extreme case, mobile clients will be forced to work under the disconnected mode. The ability to operate in disconnected mode can be useful even when connectivity is available. For example, disconnected operations can extend battery life by avoiding wireless transmission and reception. It can reduce network charges, an important feature when charge rates are high. It allows radio silence to be maintained, a vital capability in military applications. Finally, it is a viable fallback position when network characteristics degrade beyond usability.

Full client architecture can be used to effectively support the disconnected or weakly connected clients. Compared to thin client architecture, the full client architecture is at the other extreme of the range of extended client server model. The full client architecture supports the emulation of functions of servers at the client host so that applications can be executed without fully connecting to remote servers. The emulation can be supported through a proxy or a "lightweight" server residing on client hosts. For example, a proxy can emulate some database operations to allow mobile users to work in a disconnected mode. Systems, enabling the full client architecture, include CODA and Web Express.

Flexible Client Server Model

Flexible client-server architecture generalizes both thin client and full client architectures in that the roles of clients and servers and application logic can be dynamically relocated and performed on mobile and stationary hosts. In the flexible architecture, the distinction between clients and servers may be temporarily blurred for purposes of performance and availability. Furthermore, the connection between clients and servers can be dynamically established during the execution of applications (e.g., for the service handoffs).

Constraints of Mobility-

Mobile computing is characterized by four constraints-

(1) Mobile elements are resource-poor relative to static elements:

For a given cost and level of technology, considerations of weight, power, size and ergonomics will exact a penalty in computational resources such as processor speed, memory size, and disk capacity. While mobile elements will improve in absolute ability, they will always be resource-poor relative to static elements.

(2) Mobility is inherently hazardous:

A Wall Street stockbroker is more likely to be mugged on the streets of Manhattan and have his laptop stolen than to have his workstation in a locked office be physically subverted. In addition to security concerns, portable computers are more vulnerable to loss or damage.

(3)Mobile connectivity is highly variable in performance and reliability:

Some buildings may offer reliable, high-bandwidth wireless connectivity while others may only offer low-bandwidth connectivity. Outdoors, a mobile client may have to rely on a low-bandwidth wireless network with gaps in coverage.

(4)Mobile elements rely on a finite energy source:

While battery technology will undoubtedly improve over time, the need to be sensitive to power consumption will not diminish. Concern for power consumption must span many levels of hardware and software to be fully effective.

These constraints are not artifacts of current technology, but are intrinsic to mobility. Together, they complicate the design of mobile information systems and require us to rethink traditional approaches to information access.

Mobility Environment Architecture

To overcome restrictions in mobile computing the above architecture was designed; the architecture consists of the following parts:-the network environment consists of mobile hosts fixed hosts and certain access points. The fixed hosts are all connected to a backbone (i.e. the internet).Mobile hosts easily didn't contact them directly but use physically closer located hosts as access points to the backbone for means of minimizing the distance which has to be bridged by a mobile connection line. In addition to the users carrying a portable computer with them, also mobile users travelling between fixed hosts are considered in our system [9].

To distinguish from conventional client-server and network tunneling systems we choose the notions front end from a mobile host, back end for a fixed host that is servicing mobile nodes and relay for an access point to the backbone. The system could be modeled using the familiar client server notion when full connectivity is guaranteed. But when dealing with weak and sometimes fully broken lines, traditional client server terms are not sufficient any more to model the system. E.g. In case of full disconnection the front end will simulate the connection to the server using cached data. Similarly a relay in our system provides more functionality then a convenient gateway does. It offers important services for mobile hosts, especially an elaborate authentication and authorization service which is of special importance for a secure mobile system. ODBC defines a standardized method for accessing databases and is based on a client/server model. This ODBC service driver was enhanced by us to fit the needs of mobile computing [10].

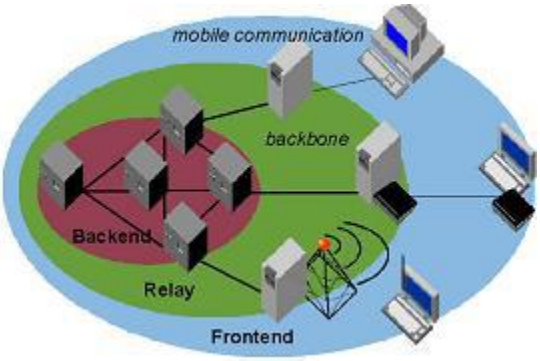

Figure 4: Mobile Environment Architecture

The Future of Mobile Applications

Much of the potential of mobile devices has yet to be exploited. Current mobile applications tend to be simple. Comparing the desktop version and PDA version of a word processor, for instance, we can see that a lot of the functionalities are not available on the PDA. Besides the small form factor which makes manoeuvring difficult and hence some of functionalities not appropriate, there is no reason why feature-rich software has to be out of reach for PDAs in the future. We believe that even the most complex actions could be carried out on mobile devices by improving input/output modality and redesigning the software engineering approach. So what can the thin-lean-mean device do for us in the future? In the not so distant future, it will probably play the role of a full-function computing but lesser device than the PC.

Software or software components can be installed on the fly from a nearby server wirelessly upon request, which can be discarded after use. For this to be feasible, software must themselves be suitably "lean" so that downloading and installing them on demand will be efficient and that they will not take up more resources than

what is commensurate with the user's needs. And lean software will naturally be more affordable – "mean". In the longer term, as the wireless infrastructures around us become sufficiently powerful and stable, it can be envisioned that mobile devices can offload more or even all of their computations to the servers, and the devices will degenerate into very thin wireless remote display terminals. Users then will be free of all the trouble of managing a computing system at home or in the pocket.

Mobile Object

Mobile Objects are programmable entities that can freely roam the network and allow clients to download the server code for execution. These objects can maintain state information that enables them to suspend execution at any arbitrary point and migrate to another location and resume execution. This information along with awareness of the mobile environment gives the object ability to make intelligent decisions regarding which host to migrate to (especially during network partitions) and what functions to perform at each host. The intention of this approach is to overcome the problems of weak connection and frequent disconnection. The Rover Toolkit, which employs Relocatable Dynamic Objects (RDO), is an example of such a model.

Application-Specific Proxy:

The application-specific proxy acts as an intermediary between clients and servers to perform computation-intensive and storage-intensive tasks on behalf of clients. An application-specific proxy on a stationary host supports proxy agents (which can be fixed or mobile) to dynamically "transcode" or "distill" application data to reduce the bandwidth consumption between the proxy and the mobile client. This proxy allows legacy and other nonstandard (including thin) clients to inter operate with existing servers. From the client's perspective, the proxy is simply a server that gets the data from someplace else.

Mobile-Aware Adaptation

Mobile clients could face wide variations and rapid changes in network conditions and local resource availability when accessing remote data. In order to enable applications and systems to continue to operate in such dynamic environments, the mobile client-server system must react by dynamically adjusting the functionality of computation between the mobile and stationary hosts. In other words, the computation of clients and servers has to be adaptive in response to the changes in mobile environments.

Application transparent adaptations are aimed at making applications work in the mobile environment without modifications. The system is entrusted with the responsibility to take required measures to counteract environmental changes like location and connection. A typical adaptation of this kind is the introduction of a third party that adds mobility awareness between the client and the server. This entity, called a proxy, can be viewed as a client to the server and as a mobility aware server to the client. The designer of this model should decide whether to place the proxy in the server, the client, both the client and the server, or the network. A couple of applications that are based on this model are Coda and Web Express.

Web Express

Web proxies are employed to enhance web browsing over wireless links without compelling server or browser changes. Web proxy can be used to pre-fetch and cache. Web pages to the mobile client's machine, compress and transform image pages for transmission over low-bandwidth links, and support disconnected and asynchronous browsing operations. Web-Express is a system based on this approach. Web-Express controls the wireless communication and optimizes the protocol in order to reduce traffic volume and latency respectively. It achieves this by employing two processes: one is the Client Side Intercept (CSI) that runs on the same mobile device as the client and the other called the Server Side Intercept (SSI) that runs within the fixed network. The idea is that the CSI would intercept HTTP requests, perform optimizations with the SSI, and reduce data transmissions. The whole process of handling an HTTP request is done as follows. The browser communicates with the local Web proxy, the CSI, over a local TCP connection using the HTTP protocol (the browser has the address of the CSI). The CSI, using a TCP connection, communicates with the SSI using a reduced version of the HTTP. The SSI reconstitutes the HTML data stream and sends it to the CSI, which then forwards it to the Web browser through the local connection. Some of the optimization methods used by Web-Express involves caching graphics and HTML objects by the SSI and the CSI, protocol reduction in terms of decreased

connection establishments, and header reductions with respect to maintaining client information at the SSI throughout the connection. Since Web-Express provides transparency to both the browser and the server, it can be used with any browser. The CSI/SSI protocols facilitate highly effective data reduction and protocol optimization without limiting any of the Web browser functionality or interoperability [11].

Coda

Coda is a file system for a large-scale distributed computing environment composed of unix workstations. It provides resiliency to server and network failures through the use of two distinct but complementary mechanisms, namely server and network failures through the use of two distinct but complementary mechanisms [12], namely server replication and disconnected operation. Server replication involves storing copies of files at multiple servers – the replication is usually aggregates of files called volumes. Disconnected operation is a mode of execution in which a caching site temporarily assumes the role of a replication site. The client a cache built based on the user profile. A file system proxy called Venus emulates the file system services in order to hide mobile issues from the mobile computer. It pre-fetches the files when it senses disconnection (data hoarding), logs the client's requests during disconnected mode (server emulation), and synchronizes the local cache with server (reintegration).In the process of hoarding a file, the proxy inquires all the available servers about the file and receives the most recent version. After modification, the file is stored at all server replication sites that are currently accessible. To achieve good performance, Coda exploits the parallelism in network protocols; for instance, parallel RPC mechanisms are used to simultaneously transmit files to multiple sites. Consistency, availability and performance tend to be mutually contradictory goals in a distributed system. Two principles guide the design of consistency mechanisms in Coda. First, the most recently updated copy of a file that is physically accessible is always used. Second, inconsistency must be rare and always detected by the system (even though it is tolerable). Coda uses atomic transactions at servers to ensure that the version vector and data of files are mutually consistent at all times. In order to overcome the data inconsistency problem resulting from partitioned sharing, Coda employs isolation-only transactions (IOT). IOT is a consistency model that uses serializability constraints to automatically detect read/write conflicts. It provides a set of options for automatic and manual conflict resolution, and to incorporate application specific knowledge to detect and resolve conflicts. The IOT mechanism is provided as an optional file system to preserve upward unix compatibility; it allows any Unix application to be executed as an IOT with its flexible interfaces being an added advantage. Unlike traditional transactions, it does not guarantee failure atomicity and guarantees permanence only conditionally.

Mobile Data Access

Mobile data access enables the delivery of server data and the maintenance of client-server data consistency in a mobile and wireless environment. Efficient and consistent data access in mobile environments is a challenge research area because of weak connectivity and resource constraints [13]. The data access strategies in a mobile information system can be characterized by delivery modes, data organizations, and consistency requirements, etc. The mode for server data delivery can be server-push, client-pull, or hybrid. The server-push delivery is initiated by server functions that push data from the server to the clients. The client-pull delivery is initiated by client functions which send requests to a server and "pull" data from the server in order to provide data to locally running applications. The hybrid delivery uses both server-push and client- pull delivery. The data organizations include mobility-specific data organizations like mobile database fragments in the server storage and data multiplexing and indexing in the broadcast disk. The consistency requirements range from weak consistency to strong consistency [14]. In this section, we will examine various proposed approaches that offer the new paradigm of data access in mobile client-server information systems.

Server Data Dissemination

In many applications (e.g., Web access), the downstream data volume from servers to clients is much greater than the upstream data volume from clients back to servers. The unbalanced communications are referred to as asymmetrical communications between clients and servers.

Application examples of asymmetrical communications in wireless environments include Hughes's Direct PC (www.hns.com) and CAI's Wireless Internet Access, where clients at mobile hosts usually have a lower bandwidth cellular or PSTN link while servers at fixed hosts may have relatively high bandwidth broadcast capability.

A challenge problem in supporting applications with asymmetrical communications is how to deliver server data and information to a large number of clients. To address this scalability problem, a new information system architecture that exploits broadcast-based dissemination capability of communications has been proposed. The central idea is that the servers exploit the downstream communication capacity in bandwidth by broadcasting data to multiple clients. This arranges arrangement is called a push-based architecture where data is pushed from the server to the clients. In contrast, most traditional client-server information systems use pull based data delivery to provide data to locally running applications.

Broadcast Disks

When a server continuously and repeatedly broadcasts data to a client community, the broadcast channel becomes a "disk" from which clients can retrieve data as it goes by. The broadcasting data can be organized as multiple disks of different sizes and speeds on the broadcast medium. The broadcast is created by multiplexing chunks of data from different disks onto the same broadcast channel. The chunks of each disk are evenly interspersed with each other. The chunks of the fast disks are repeated more often than the chunks of the slow disks. The relative speeds of these disks can be adjusted as a parameter to the configuration of the broadcast. This use of the channel effectively puts the fast disks closer to the client while at the same time pushing the slower disks further away. This technique presents an opportunity to more closely match the broadcast to the workload at the client. Assuming that the server has an indication of the client access patterns (either by watching their previous activity or from a description of intended future use from each client), then hot pages or pages that are more likely to be of interest to a larger part of the client community can be brought closer while cold pages can be pushed further away [15]. This, in effect, creates an arbitrarily fine-grained memory hierarchy, as the expected delay in obtaining an item depends upon how often that item is broadcast. The broadcast disk technique, therefore, provides improved performance for none uniformly accessed data. In the simplest scenario, the server can broadcast different items at the same frequency. With the "flat" broadcast, the expected delay required prior to obtaining an item is the same for all items broadcast (namely, half a broadcast period) regardless of their relative importance to the clients. This "flat" approach has been adopted in earlier work on broadcast-based information system such as Data cycle and the work [5, 6]. By comparison, the server can broadcast different items with differing frequency; important items can be broadcast more often than others.

Client Cache Management

Caching of frequently-accessed data items is an important technique that reduces contention and improves query response times on narrow bandwidth wireless links. The cached data can also support disconnected or intermitted connected operations. However, cache pre-fetching and consistency strategies can be greatly affected by the disconnection or weak connectivity of mobile clients. The weak connectivity makes cache coherence expensive due to communication latency and intermittent failures. Pre-fetching (or *hoarding*) data into the client cache prior to disconnection is a difficult challenge in mobile client-server computing.

Conclusion

In this paper, we have study about client server in mobile Computing. In other words, we can say that how to implement Client server in mobile computing. With advances in wireless data communication and portable computers, nomadic users will soon enjoy virtually limited access to information and services anytime and anywhere. There are however, obstacles and limitations inherent in the wireless environment. The unpredictable mobility of the user adds a great deal of complexity to the problem. Such obstacles render the traditional approach to designing and implementing client-server applications insufficient in meeting nomadic users' expectations. Mobile computing (including mobile client-server information access) is a rapidly changing research field that depends on a rapidly evolving set of technologies.

The advent of the so-called third generation wireless communication systems, such as UMTS, is an indication of the importance of research in this area. Researchers, however, should monitor these advances closely and should

adapt and direct their research based on the parameters of the latest Technology. This is important because some of the strong assumptions regarding the limitations of the wireless network or the mobile computer are being relaxed or nullified by newer technological developments. Many problems still remain to be understood and solved. At this time, it is intricately difficult to quantitatively evaluate and compare the various proposed models and techniques because of the lack of available application experiences and samples. Experimentation with these models should be the critical next step in mobile computing research. We hope that this comprehensive review will help enhance the understanding of the opportunities and limitations of mobile client-server computing. We also hope that the review substantiates the importance of recognizing and understanding emerging technologies in shaping the future directions of research in this area.

References

[1]. Yau, S. S., Karim, F., Wang, Y., Wang, B., & Gupta, S. K. (2002). Reconfigurable context-sensitive middleware for pervasive computing. *IEEE Pervasive Computing*, *1*(3), 33-40.

[2]. Weiser, M., Gold, R., & Brown, J. S. (1999). The origins of ubiquitous computing research at PARC in the late 1980s. *IBM systems journal*, *38*(4), 693.

[3]. Banikazemi, M., Olshefski, D., Shaikh, A., Tracey, J., & Wang, G. (2013). Meridian: an SDN platform for cloud network services. *IEEE Communications Magazine*, *51*(2), 120-127.

[4]. Prasad, M. R., Gyani, J., & Murti, P. R. K. (2012). Mobile cloud computing: Implications and challenges. *Journal of Information Engineering and Applications*, *2*(7), 7-15.

[5]. Kaufman, C., Perlman, R., & Speciner, M. (2002). *Network security: private communication in a public world*. Prentice Hall Press

[6]. Ryu, K. K., Shin, E., & Mooney, V. J. (2001). A comparison of five different multiprocessor SoC bus architectures. In *Digital Systems Design, 2001. Proceedings. Euromicro Symposium on* (pp. 202-209). IEEE.

[7]. Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M., & Tantawi, A. (2005, June). An analytical model for multi-tier internet services and its applications. In *ACM SIGMETRICS Performance Evaluation Review* (Vol. 33, No. 1, pp. 291-302). ACM.

[8]. www.123seminarsonly.com

[9]. Kumar, L. N., & Douligeris, C. (1998). The dynamic three-tier protocol: an access remedial scheme for DQDB Metropolitan Area Networks. *Computer Communications*, *21*(7), 624-643.

[10]. Chen, H. S., Chen, M. S., Huang, S. L., & Song, D. (1998). *U.S. Patent No. 5,822,524*. Washington, DC: U.S. Patent and Trademark Office.

[11]. Lu, Q., & Satyanarayanan, M. (1995, May). Improving data consistency in mobile computing using isolation-only transactions. In *Hot Topics in Operating Systems, 1995.(HotOS-V), Proceedings., Fifth Workshop on* (pp. 124-128). IEEE.

[12]. Satyanarayanan, M., Kistler, J. J., Kumar, P., Okasaki, M. E., Siegel, E. H., & Steere, D. C. (1990). Coda: A highly available file system for a distributed workstation environment. *IEEE Transactions on computers*, *39*(4), 447-459.

[13]. Jing, Y. (2004). *Space-time code design and its applications in wireless networks* (Doctoral dissertation, California Institute of Technology).

[14]. Satyanarayanan, M. (1996). Accessing information on demand at any location. mobile information access. *IEEE Personal Communications*, *3*(1), 26-33.

[15]. Satria Mandala, M., Ngadi, A., Abdullah, A. H., Baragada, S. R., Ramakrishna, S., Rao, M. S. & Othman, A. K. (2008). A Survey on MANET Intrusion Detection.