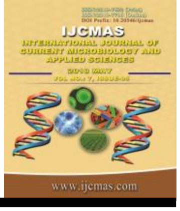
EXCELLENT PUBLISHERS

International Journal of Current Microbiology and Applied Sciences
ISSN: 2319-7706 **Volume 7 Number 05 (2018)**
Journal homepage: http://www.ijcmas.com



Review Article https://doi.org/10.20546/ijcmas.2018.705.356

Chronological Evolution of Mobile Agent: A Plausible Paradigm of Mobility

Rahul Singh Chowhan[1]* and Purva Dayya[2]

[1]Agriculture University, Jodhpur, India
[2]MPUAT, Udaipur, India

**Corresponding author*

ABSTRACT

Keywords

Distributed computing, Autonomous communication architecture, Remote objects

Article Info

Accepted:
22 April 2018
Available Online:
10 May 2018

Cognizance of computers in human existence has brought in a revolution of doing things differently, efficiently and in a more user-friendly way. Awareness of computers via education has acted as second progressive phase in which we have recognized the computers as part of day-to-day life. Multi-fold evolution of knowledge-of-techniques, collection of field skills, acquisition and availability of committed resources, increased uptime of servers and on-demand usability and serviceability has advanced the Internet in leaps-and-bounds increasing the overall productivity. Expansion of computer technology in all dimensions of science has made it possible for human to see an automated present with enormous future scope and new possibilities in research area, industrial art and scientific investigation. The management of resources is the central theme point around which all distributed network is laid and administered.

Introduction

The Computing has gone through numerous changes since the introduction of the primary registering machines. Improvements in innovation have brought about the accessibility of quick and modest processors, and advances in correspondence innovation have brought about the accessibility of lucrative and exceptionally capable systems. Among these, the central connectivity of network that binds the systems together is the one part that is shared by clients constantly (Rajesh Kumar *et al.*, 2015). This becomes the bottleneck problem in centralized schemes where single point control and failure are possible. The incorporation of various system administrations and networking unit brought forth the new worldview of communicating and interacting over the Internet called as Distributed Computing. This has completely changed the way of figuring out messages along with offering speedy and exact responses for an assortment of complex issues in various fields. These days we are completely immersed by the data age, and using spends our time in information exchange and conveying and assembling data through the Internet. The Internet is never ending and advancing along progressive frameworks to impart more unique ways for communication, storage and dissemination of information with more reliable, fast and secure networks. Throughout the years, a few strategies have developed to empower these improvements, extending from shortsighted information sharing to cutting edge frameworks supporting a huge number of administrations (Mustafa Akif Karzan and Nadia Erdoga, 2013). This paper gives a review on various facets of distributed system and processing frameworks. This includes the basic client-server mechanism, message passing mechanism and a scenario based discussion on working of new age distributed communication protocol called as mobile agent, briefing the facilities and add-ons with it.

Semantically rich distributed system

The distributed system empowers local administrators to make plans that think about nearby goals and requirements inside the limits of the general framework destinations. At the beginning of computer systems, centralized computing was the only fundamental paradigm that encompassed the allocation of numerous resources available with server-hosts connected to a small group of computers called client-hosts. This concept has kept the client-hosts simplest possible by allowing the offloading of computation on to a central workstation., Up to a greater degree of dependency, this type of system terminology relies on network resources, like servers, infrastructure for computation & processing of data, saving files to the storage, etc. In this typography, client-hosts are diskless nodes that are dependent on central terminal to load its operating system. Simply, clients act as an input/output interface to the server because they neither have their own operating system nor personalized resources (Tina Setter *et al.*, 2016). The conventional client-server architecture and technology like Remote Procedure Call (RPC) falls in this category. From programming perspective, this paradigm has also circumscribed the greatly practiced procedural/imperative programming language like C.

Now-a-days, the much extensive taxonomy used for centralized paradigm is Thin Client, that refers to a system with various dependencies on peripherals for successful execution. They are purposely built for remoting into a server allowing multiple client-hosts to share their computations with a server farm over a network. The client-hosts are highly dependent on another, especially a server, for approval of various system roles and authorizations. The predefined roles considered by servers fickle irrespective of the domain and discipline of clients connected together (Youssef M. Essa *et al.*, 2013). Nearby choices from various parts of the framework are incorporated through coordination and correspondence of various components.

Features of mobility

In computer science, remote evaluation is a term that belongs to the family of mobile code, within the field of code mobility. It is for any technology that involves the transmission of executable software code from a client hosts to a server hosts for execution to be happen at the server and the result is sent back to the client after execution for this resources of server side are used. An example for remote evaluation is grid computing: An executable task may be sent to a specific computer in the grid. After the execution has terminated, the result is sent back to the client (Gaoyun Chen *et al.*, 2010). The client in turn may have to reassemble the different results of multiple concurrently calculated subtasks into one single result.

Code-on-Demand is used for any technology that sends executable code from a server host to a client host on the request of the client's application. Code on demand is a specific use of mobile code under the field of code mobility. The internet has various kinds of platforms that communicate very frequently with each other by wired or wireless media (Schoeman *et al.,* 2003). This kind of system often requires an interoperable platform that can allow communication over different platforms. Languages like Java introduce the concept of the byte code from which the platform independencies can be achieved at some level of execution. This introduces “application scriplets” or applets, which are the lines of code that is located on server and is downloaded on browser on demand of user.

The processing in distributed systems requires complex runtime frameworks that are capable to balancing load within the system as well as handling of incoming load from outside the network. This must not affect the overall performance, low time message delivery, reliability, parallel computation and other factors. Distributed objects describe a form of gateway operation allowing the communication with encapsulated data and actions remotely. Likewise, REV provides code for execution of intended management function while CoD retrieves and caches code to execute the intended management function (Ahila *et al.,* 2014). Distributed application structure defines client-server model that does segregation of workloads between service or resource provider, called servers and service or resource requester, called clients. These two separate components, a client and a server, which communicate over a network through a TCP/IP handshake paradigm. The client requests information, while the server responds when its advertised services are accessed. This each request/response is a complete round trip on the network. The code that implements these services i.e. the know-how is hosted locally by the server, also server has processing capabilities. Client decides with some intelligence which of services offered by server it should use (Rajesh Kumar *et al.,* 2015).

Shift from centralized to cooperative paradigm

This is semantically profound paradigm which encapsulates the dynamic mobility of the objects. Introduction of cooperative paradigm has made the mobility of static objects over the network nodes connected within one cluster. This paradigm has spawned a new technology called mobile agent technology which is not just limited to features previously existed in the client-server based technologies. This has allowed it be called as paradigm shift that has completely transformed the way communication has ever happened over the network nodes. This not only adds up mobility but can monitor, investigate, cooperate, form communities, communicate with each other, compete with other, can work to achieve a goal, remote execution on remote nodes and all that on behalf of user/ owner (Rahul Singh Chowhan *et al.,* 2016).

Today, the central paradigm linking all distributed object technologies is a synchronous message-passing paradigm whereby all objects are distributed, but stationary, and interact with each other through message-passing. Earlier scheduling approaches were constrained to use of shared memory of machine within the system or essentially utilize work-stealing across and over fixed memory units for load balancing to reduce the process migration (Pandey *et al.,* 2013).

These paradigms need to be enhanced in some fashion with additional paradigms such as asynchronous message-passing, object mobility, and active objects. The revolution in web has brought in paradigms that allowed "executable content" to become a common term in web glossary. Like applets (Code-on-Demand) are essentially sets of program code that can be downloaded, instantiated and executed in web browser. This concept has made an analogy with servlets (Remote EValuation) that move business logic to server for execution and instantiation by allowing the client to upload the code to a server (Shigeki Shiokawa, 2016). As the market trend towards wide-area open networks grows in large scale, an increasing number of people are expected to use partially connected devices such as laptops, mobile phones, personal digital assistants (PDA) home and business computers etc. So they can access/do their work from anywhere and at any time. However, developing distributed applications that can function in a mobile computing environment is difficult for various reasons:

Mobile devices are not permanently connected to the network and often disconnect from it for longer period of time.

Even when connected, the connection often has low bandwidth, high latency and prone to sudden network failures.

Also, the computer may be forced to use different transmission channels depending on its physical location, the performance of its network connection can vary dramatically from one session to another.

And, every time the mobile device reconnects, the network address assigned to it may change.

That's why any distributed application that need to use internet resources effectively from a mobile platform must deal with the existing environmental changes and intolerant network conditions (Sobitha Ahila and Shunmuganathan, 2014).

Mobile Agents represents the next forward leap in the evolution of executable content on the Internet, allowing program code to be transported along with state information. Like IBM Aglets are Java objects that can move from one host to another in a heterogeneous network. That is, it can get executed on one host and can suddenly come to a halt on the current host so to be dispatched onto a remote host, and resumes its execution there. They take along the program code as well as its data supporting the ideas of autonomous execution and dynamic adaptation on its itinerary. The Mobile Agent paradigm conspicuously makes a proviso of set of basic services that the server should provide. The clients make use of the resources/services offered by the server by sending mobile agents to the server for a purpose. Then they use the resources/services served by server as desired to achieve a goal. Though, the functionality provided by mobile agents is not something that cannot be accomplished and executed with conventional client-server mechanism (Eldegwi *et al.,* 2015). However, the implementation of any new functionality becomes much easier with their help. Essentially, at the fundamental level they provide a layer of abstraction between the services provided by the server and the way they are used.

A key characteristic of the mobile agent paradigm is that any host in the network is allowed a high degree of flexibility to possess any amalgamation of know-how, resources, and processors. Its processing capabilities can be combined with local resources. Know-how (in the form of mobile agents) is not tied to a single host but available throughout the network. If we compare these three paradigms, we will see the chronological trend shifting toward greater flexibility (Zhixin Tie, 2013). The client and the server have merged with the time and technology and have become single host playing both the roles. The applet and the servlet, while serving as client and server extenders, respectively, have been combined and improved with the emergence of mobile agents. A mobile agent is not bound to the system where it begins the execution (Rahul Singh Chowhan and Rajesh Purohit, 2016). The mobile agent is free to travel among the hosts in the network. Created in one execution environment, it can transport its state and code with it to another execution environment in the network, where it resumes its execution.

Scenario discussion

Let's make it an easy cup of tea taking everybody along and understand it with help of the metaphoric scenario stated as under:

Overview

There are two friends, namely Radha and Rashi, and Radha wants to bake a chocolate cake. The requirements' analogy with the scenario is as follows:

A chocolate cake: An outcome of a specified service running on server.

The recipe: Business logic or a sequence of execution steps in form of know-how.

Specified ingredients: Code contents that are movable.

A microwave oven: Immovable resource available at server required to handle the requests.

A person/machine to blend all ingredients appropriately: A computational component responsible for execution of code.

Preparation of cake: Execution of the service, and

The location to prepare the cake: The remote site, a server, where service is executed.

Scenario 1: Client-server paradigm

Radha wanna make a chocolate cake, but

She doesn't know the recipe.

She doesn't possess needed ingredients.

She doesn't have microwave oven.

Providently, she remembers that her friend, Rashi, knows how to prepare a chocolate cake and has a fully-fledged modular kitchen. Since Rashi is usually quite happy to prepare cakes on request.

Radha makes a call to Rashi to ask*: "Will you prepare a chocolate cake for me, if you are free?"*(This resembles to the handshake process that happens in client-server mechanism).

Rashi agrees and acknowledges: *"Yes, I'm free. I will make a chocolate cake for you."*

Rashi prepares the chocolate cake and delivers it to Radha.

Scenario 2: Remote evaluation paradigm

Radha wanna prepare a chocolate cake.

She knows the recipe.

She neither has the required ingredients nor an oven (immovable resource).

Her friend Rashi has both ingredients and oven but at her own place, yet she doesn't know how to make a chocolate cake.

She knows that Rashi is happy to try new recipes

She phones Rashi asking, "*Can you make me a chocolate cake? Here is the recipe: Take 3 eggs…etc.*" (Mechanism of Lookup table used at server side in REv).

Rashi prepares the chocolate cake following Louse's recipe and delivers it back to her.

Scenario 3: Remote evaluation paradigm

Radha wants to prepare a chocolate cake.

She has at home both the required ingredients and an oven

She lacks the proper recipe.

However, Radha knows that her friend Rashi has the right recipe and she has already lent it to many friends.

Radha phones Rashi asking, "*Can you tell me you chocolate cake recipe*?".

Rashi tells her the recipe and Radha prepares chocolate cake at her home.

Scenario 4: Remote evaluation paradigm

Radha wants to prepare a chocolate cake.

She has the ingredients.

She does not know recipe and also doesn't have the oven at home.

However, she knows that her friend Rashi knows the recipe and Ram has an oven at his place.

So, Radha take the ingredients to Rashi, and then they prepare the chocolate batter and then go to Ram's home, where she learns to bake the cake.

Finally, she returns back to her home.

These scenarios clearly draw an analogy with the real time implementation of technological shift happened from Client-Server to Mobile agent Mechanism. Mobile agent(s) are becoming a popular entity with prominent modern technological scope in computer science discipline. They are in the process of evolution from being confined to research work to being a pragmatic and commercial product. The main benefit that mobile agents serve is that they can move closer to place where the business logic or information resources are located. This way they reduce the overall server load and network latencies that are high in client server mechanism. It can delegates tasks to other agents while working for accomplishment of its own goals.

Mobile agents have comforted the user at some extent of communication but security is still the major concern in wide acceptance of this technology. Many commercial and non-commercial companies are accepting it under constrained usage of mobile agent for purpose of goal achievement. We are currently accepting the mobile agent being coupled with security tactics offered by programming language or the system to which mobile agents are migrating. Though the mixed architecture are becoming more famous among the use for e-commerce and online rescheduling purposes.

References

Ahila, S. Sobitha, and K. L. Shunmuganathan. "Overview of mobile agent security issues—Solutions." In Information Communication and Embedded Systems (ICICES), 2014 International Conference on, pp. 1-6. IEEE, 2014.

Eldegwi, H.M., M. B. Badawy, and Hamdy M. Kelash., "Building a Secure Decentralized Energy System with Remote Monitoring Using Mobile Agents." In 2015 Fifth International Conference on e-Learning (econf), pp. 263-268. DOI: 10.1109/ECONF.2015.80, IEEE, 2015.

Gaoyun Chen, Jun Lu, Jian Huang, and Zexu Wu. “Saaas-the mobile agent based service for cloud computing in internet environment.” In 2010 Sixth International Conference on Natural Computation, vol. 6, pp. 2935-2939. IEEE, 2010.

Mustafa Akif Karzan, and Nadia Erdogan, “Topic Based Agent Migration Scheme via Publish/Subscribe Paradigm”, International Journal of Information and Education Technology, Vol. 3(3): 290-294, ISSN: 2010-3689, DOI: 10.7763/IJIET.2013.V3.283 (2013)

Pandey, Mr Rajesh, Mr Nidheesh Sharma, and Mr Ramratan Rathore. “Aglets (A Java Based Mobile Agent) and It’s Security Issue.” International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) 2. 4 (2013).

Rahul Singh Chowhan and Rajesh Purohit, “Study of mobile agent server architectures for homogeneous and heterogeneous distributed systems.” International Journal of Computer Applications 156, no. 4 (2016): 32-37.

Rahul Singh Chowhan, Amit Mishra, and Ajay Mathur, “Aglet and kerrighed as a tool for load balancing and scheduling in distributed environment.” In Recent Advances and Innovations in Engineering (ICRAIE), 2016 International Conference on, pp. 1-6. IEEE, 2016.

Rajesh Kumar, S Niranjanr, and Yashpal Singh, “A Review on Mobile Agent Technology and Its Perspectives.” Journal of Computer Sciences and Applications, vol. 3, no. 6 (2015): 166-171. DOI: 10.12691/jcsa-3-6-11.

Rajesh Kumar, S Niranjanr, and Yashpal Singh, “A Review on Mobile Agent Technology and Its Perspectives.” Journal of Computer Sciences and Applications, vol. 3, no. 6 (2015): 166-171. DOI: 10.12691/jcsa-3-6-11.

Schoeman, Marthie, and Elsabé Cloete. “Architectural components for the efficient design of mobile agent systems.” Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology. South African Institute for Computer Scientists and Information Technologists, 2003.

Shigeki Shiokawa, “Performance analysis for use of mobile agent in wireless multihop networks.” In 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 827-832. doi: 10.1109/icufn.2016.7537153, IEEE, 2016.

Sobitha Ahila, S., and K. L. Shunmuganathan., “Overview of mobile agent security issues—Solutions.” In Information Communication and Embedded Systems (ICICES), 2014 International Conference on, pp. 1-6. IEEE, 2014.

Tina Setter, Andrea Gasparri, and Magnus Egerstedt. “Trust-based interactions in teams of mobile agents.” In 2016 American Control Conference (ACC), pp. 6158-6163. DOI: 10.1109/ACC.2016.7526637, IEEE, 2016.

Youssef M. Essa, Gamal Attiya, and Ayman El-Sayed. "Mobile agent based new framework for improving big data analysis." In Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on, pp. 381-386. doi: 10.1109/cloudcom-asia.2013.75, IEEE, 2013.

Zhixin Tie, “A Mobile Agent-Based System for Server Resource Monitoring.” Cybernetics and Information Technologies, Volume 13, Issue 4, Pages 104–117, ISSN (Print) 1314-4081, DOI: 10.2478/cait-2013-0057, December 2013.

How to cite this article:

Rahul Singh Chowhan and Purva Dayya. 2018. Chronological Evolution of Mobile Agent: A Plausible Paradigm of Mobility. *Int.J.Curr.Microbiol.App.Sci.* 7(05): 3051-3057.
doi: https://doi.org/10.20546/ijcmas.2018.705.356