A Comparison of Migration Scheme of Mobile Agent in Aglet

Nripesh Kumar
Deptt. of Computer Sc. & App.
GEHU, Bhimtal

Purushottam Das
Deptt. of Computer Sc. & App.
GEHU, Bhimtal

Shambhu Prasad Sah
Deptt. of Computer Sc. & App
GEHU, Bhimtal

***Abstract*-Mobile agents (MA) are autonomous and proactive software entities which act on behalf on an owner and have the ability to migrate through a heterogeneous network of computer. In mobile agent migration, migration strategy is responsible for movement of agent from one host to another, an efficient migration strategy provide the low computation power, low bandwidth utilization. Currently there are several agent platforms which provide different approach of migration. In this paper we compare push and pull oriented approach of migration and analysis the performance with cache mechanism on aglet mobile agent platform.**

Keywords: Mobile Agent, Agency, Push, cache, migration

I. INTRODUCTION

Mobile Agents is programs which represents a user in network, and have a ability to migrate from one host to another host [1]. A mobile agent has authority to make migration related decision. After migration to the desired host the mobile agent can resume its previous state. Therefore, there is no need to maintain a continuous connection between client and server. This will save the unnecessary load. When mobile agent migrates from one host to another, It is the responsibility of the agency to migrate the mobile agent from one host to another host. For migration, A agency can use several approach of migration. A best migration strategy can be select based on the several parameters like QoS, Pos, and SOS etc. QoS parameter means quality of services which is provided by a migration strategy during migration of a mobile agent. It is varied from 0 to 1. The QoS 1 indicate that the quality of service provided by a migration strategy is good and QoS 0 indicate that no service is provided.Pos parameter means price of service or cost of migration of mobile agent from one host to another.

II. MIGRATION APPROACH

Migration strategy of mobile agent can be classified in to two category push and pull[4]. These migration strategies have a great impact on the performance of MA. In this paper we compare push approach of MA migration.

A. Push –All-to-Next

It transfers entire code to the next host, while sending the state of the mobile agent. This approach is virtually dependent on the size of the code and does not impose a continuous load.

B. Push-All-to-All

It transfers entire code of agent to all the host of a network. In this scheme mobile agent should know the entire destination before transmitting the code. When the MA arrives on the destination platform, MA's execution can start immediately without any further code transmission.

III. PERFORMANCE ANALYSIS OF MIGRATION SCHEME

Aglet is a open source agent development platform, which is developed by IBM Japan. It is widely used for developing mobile agent, It provide complete support for java language. The basic architecture of aglet consists of two layer and sub-component Aglet Runtime Layer provides following functionality.

1) Serialization and de- serialization of Aglets
2) Class loading and transfer
3) Reference management and garbage collection.

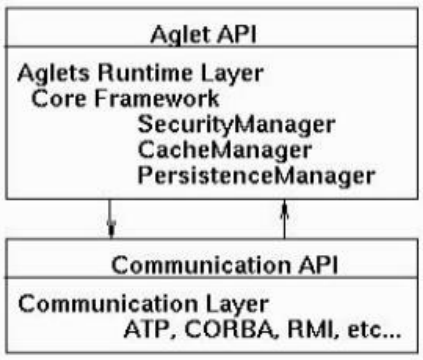

Fig. 1 [1]

A. Cache Manager

In aglet cache-manager is responsible to maintain unnecessary transfer of code during the migration. When mobile-agent maintain a connection then the local agency check which program code are cached , Cache mechanisms avoid an unnecessary transfer of code from a remote area by using data cached in the local area. At the first connection in mobile agent migration, the local area has to check which program codes are cached or not in the remote area, because Caching all MAs in the platform results in waste of storage

space. Therefore, the platform is required to choose MAs whose code should be cached, while migrating to another platform according to the policies provided in the MAS . Agent can be cached based on the mean number of visits of an MA to a platform during a given interval. Therefore, it is necessary to compute the mean number of visits of the MA to the platform from the time of MA's creation to the time of last visit of the destination. If the MA frequently revisits the platform, it is necessary for the platform to cache the MA code, in order to improve system performance. In addition, the execution time of the MA can be reduced by partially eliminating transmissions of the MA code.

IV. EXPERIMENTAL SETUP

The experiment are made with aglets2.5-alpha on three different machine, The first was equipped with 1.90GHz , 2GB memory with Ubuntu operating system ,the second was equipped with 2.53 GHz I3 CPU with 4GB memory , Ubuntu and JSDK1.7,and the third was equipped with a 2.40 GHz I5 CPU with 4 GB memory ,Ubuntu and JSDK1.7. This experiment was carried out to measure the MA migration cost of the two existing approach push all to next and push all to all with code caching mechanism. In our experiment MA has circulated the 3 nodes.

A. Performance Comparision

Table1.Push –all-to-next without cache vs. using caching mechanism

Size of MA Code (byte)	Migration Cost without cache(ms)	Migration cost with cache(ms)	Performance ratio (%)
7.5K	65	55	118
64K	140	50	280
160K	275	47	585

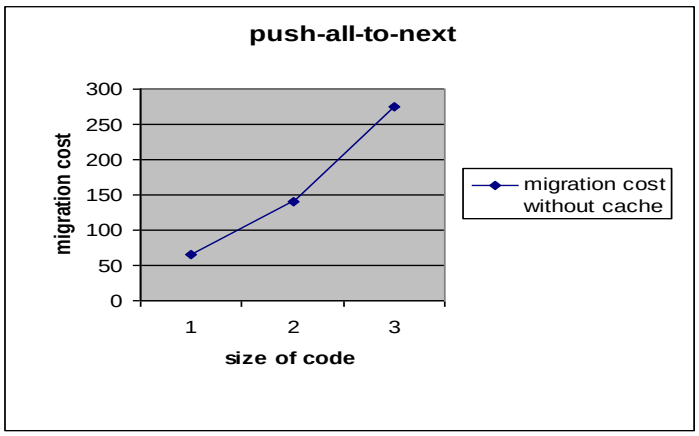

Fig. 2 (a)

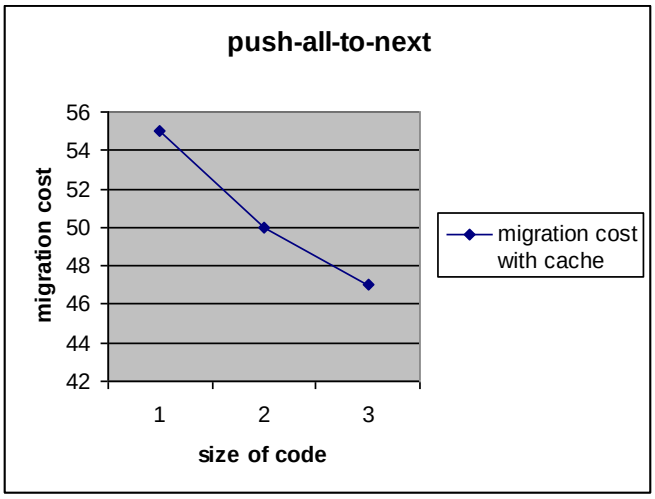

Fig. 2 (b)

Table 2. Push –all-to-all without cache vs. using caching mechanism

Size of MA Code (byte)	Migration Cost without cache(ms)	Migration cost with cache(ms)	Performance ratio (%)
7.5K	170	55	309
64K	430	70	614
160K	840	104	807

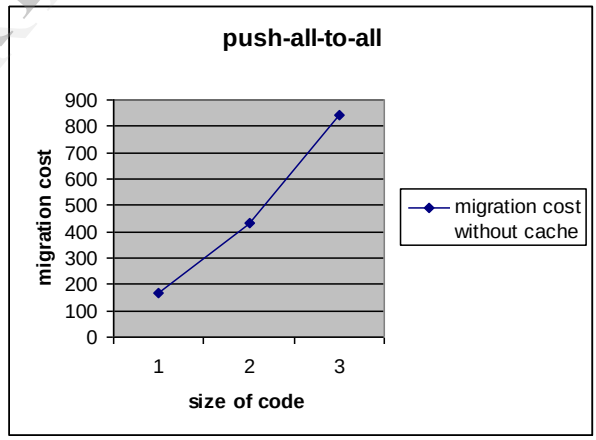

Fig. 3 (a)

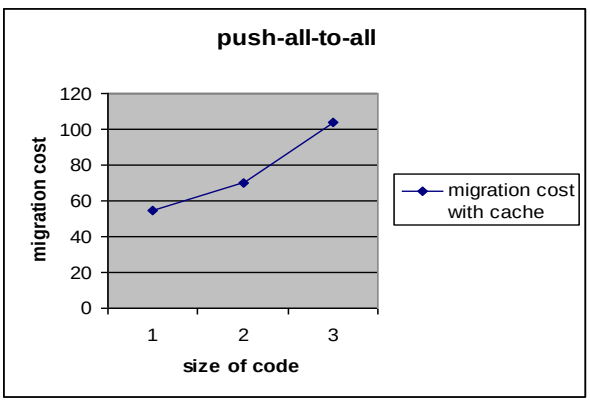

Fig. 3 (b)

The migration cost of mobile agent is calculated by adding the freezing time of all the nodes or sites. From above table when the size of MA code is varied from 7.5K to 160K, the freezing time of caching-added scheme is lower than the freezing time of without caching scheme. As the size of MA code increases, the performance of the scheme is improved from 118% to 807%.

V. CONCLUSION

This paper compares the two migration scheme push-all-to-next and push-all-to-all with the cache mechanism provided by the aglet platform. From the comparison it is clear that the cache manager available in aglet runtime layer provide the best performance in compare to general scheme.

VI. REFERENCES

1. Higashino, M.,"Mobile Agent Migration Based on Code Caching", (WAINA), 2012,pp**:**651 – 656
2. Arif Hidayat , "A Review on the Communication Mechanism of Mobile Agent ",IJVIPNS-IJENS Vol: 11 ,2011,No: 01
3. Ahmed, K.E.U.,"Strong thread migration in heterogeneous environment", ICCET '09 Vol:**1**,pp**(s):**205 – 209
4. Sakurai,"A Code Transformation Method For Strong Migration Mobile Agent",PacRim 2007,pp(s):485 – 488
5. Seungsang Sun Sungkyunkwan Univ., Suwon"An Efficient Migration Scheme for Mobile Agents in Ubiquitous Environments", ISITC 2007,pp**(s):**110 - 114
6. M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, Linear Programming and Network Flows, 2nd ed. New York: Wiley, 1990.
7. F. Raji ,B. Tork Ladani"Anonymity and security for autonomous mobile agents", IET . Secur., 2010, Vol. 4,