Comparative Study of Single and Multiple Mobile Agents visiting Network Nodes

S.A. Arekete
Redeemer's University
Redemption City, Ogun State
Nigeria

Abstract

Mobile agent technology has been found as a distributed system that is very efficient in monitoring activities of users in a network environment. There are two possibilities in launching agents to visit nodes. One option is to empower a single mobile agent to visit all the nodes in the network and collect data on users' activities. In this case, the mobile agent visits the nodes in the itinerary in turn until the last node is visited and it then returns to the home node. The alternative is to use a number of multiple agents to visit a segment of the network. In this paper, we study the effects of using multiple mobile agents as opposed to a single mobile agent to visit all the nodes of a network; we attempt to understand the gains, if any, of using a of multi-agent systems over single agent systems.

Keywords: Mobile agent, multi-agent, autonomy, itinerary, partitioning.

Corresponding Author: S.A. Arekete

1. Introduction

The agency theory has engaged the minds of researchers for a number of decades. An agent is described as an atomic software entity operating through autonomous actions on behalf of users without intervention [1].An agent is able to perceive and influence its environment, for example, the operating system, applications, and other agents by communication with these instances. The agent may follow a set of rules predefined by the user and then applies them. The key aspects of agents are their autonomy and abilities to reason and act in their environment, as well as to interact and communicate with other agents to solve complex problems [2],[3]. A static agent has the capacity to provide resources and services at a node. While every agent does not have to move, it is of use that some agents can travel across a network to perform actions on behalf of their owners. These are termed mobile agents. Rothermel and Popescu-Zeletin[4]describe mobile agents as self-contained and identifiablecomputer programs that are capable of moving within the network and can act on behalf of the user or some other entity. A mobile agent refers to a software unit that travels between network nodes following either a predefined or a context-dependent itinerary [5]. After its creation, a mobile carries its persistent state and code to another node, where its execution can be restarted or resumed, as the case may be. Through interaction with a node, a mobile agent can perform complex processing and filtering operations on retrieved data, directly control equipment and dynamically deploy software to the nodes.Schlegel*et al.*, [6] remarks that "mobile agents have been introduced as a paradigm for distributed systems, in particular to reduce network traffic by moving code close to the place where the data is. A mobile agent is a program that can migrate from one host to another host in a network of heterogeneous computer systems and fulfil a user-given task. It can autonomously work and communicate with other agents and host systems." The design of a mobile agent for monitoring the activities of users in a network environment has been presented in [7].

A mobile agent can visit all the nodes in a network, going from one node to another in turn. After the last node has been visited, it then returns to the home node with results. When the number of nodes to be visited is very large, the time taken to visit all the nodes by a single mobile agent may be large. We can therefore consider breaking the nodes into segments and use multiple mobile agents to visit each segment. This may lead to time savings (Figure 2.1).

The rest of the paper is organised as follows. In Section 2, we investigate the itinerary partitioning and formulate the equations for itinerary and sub-itineraries. In Section 3, we look at the possible gains of using multiple mobile agents over single mobile agent visiting all the nodes in the network while in Section 3, we demonstrate in concrete terms, the effect of using multiple agents. We make concluding remarks in Section 4.

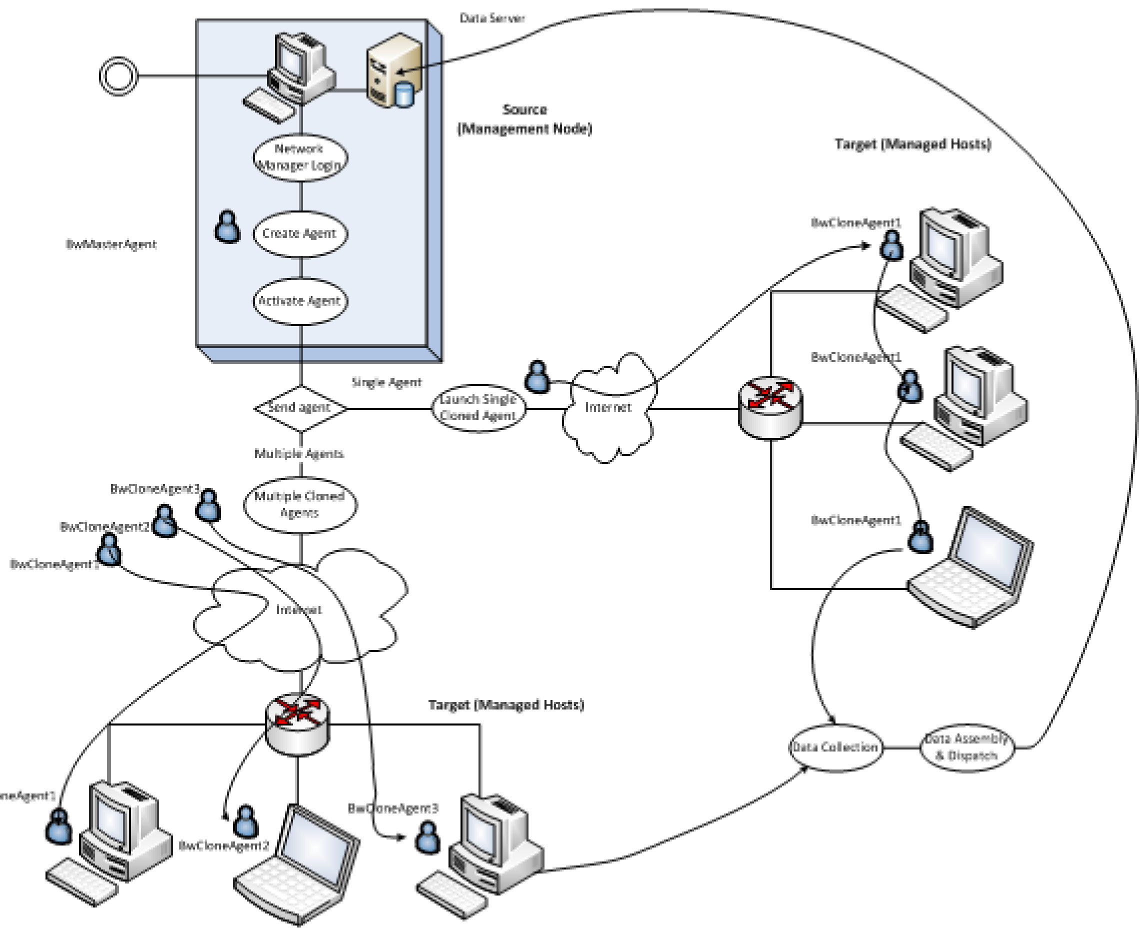

Figure 2.1 Mobile Agent Framework

2. Partitioning the Itinerary

We consider an ordered set of target nodes to be visited by the mobile agent, defined as:

$$T = \{t_1, t_2, t_3, \ \ldots, \ t_m\} \qquad \ldots\ldots\ldots\ldots\ldots\ldots \ (1)$$

If the mobile agent starts from a home node denoted by S_h, the itinerary of the mobile agent can be represented as:

$$I = \{S_h, t_1, t_2, \dots, t_m\} \quad \dots\dots\dots\dots\dots\dots \quad (2)$$

where S_h is the home node and $t_i (i \geq 1)$ is the *i*th node to be visited in the network. This itinerary is adequate for a network of a few nodes. When the number of nodes to be visited (m) is very large, the response time for a single mobile agent visiting all the nodes sequentially could become very high, resulting in bloated state problem [8],when the accumulated data from node to node would have been too much.

Rather than using a single mobile agent, we may consider using multiple agents by making clones of the mobile agent. To determine the number of mobile agents we need, we definea partition factor p to control the itinerary partition. The partition factor $p(1 \leq p \leq m)$ is the maximum number of nodes that a mobile agent can visit. Then, the number of mobile agents needed to visit the network, which is also the number of sub-itineraries, N, is given by:

$$N = {}^{m}/_{p} \quad \dots\dots\dots\dots\dots\dots\dots\dots \quad (3)$$

The sub-itineraries therefore are given by the following equation:

$$I_1 = \{t_1, t_2, \dots, t_p\},\ I_2 = \{t_{p+1}, t_{p+2}, \dots, t_{2p}\}, I_3 = \{t_{2p+1}, t_{2p+2}, \dots, t_{3p}\}, \dots \quad (4)$$

3. Single versus Multiple Agents

Consider visiting m nodes and if it takes time t to visit each node and perform certain operation, then when a single mobile agent is launched, the time to visit m nodes is given by:

$$t_m = mt \quad \dots\dots\dots\dots\dots\dots\dots\dots \quad (5)$$

Now, if we use partition factor p, then,

$$t_m = t_p = \frac{mt}{I_j} = \frac{mt}{{}^{m}/_{p}} = mt * \frac{p}{m} = t * p \quad \dots\dots\dots\dots\dots\dots\dots\dots \quad (6)$$

Hence, the time for multiple mobile agents to visit all the nodes has reduced to the time to visit one segment (or sub-itinerary) of the network since all the mobile agents would run in parallel. We look at this in different scenarios while we assume that the time to visit the next node is 1sec.

3.1 Scenario One: Varying the node size, keeping the number of agents constant

We consider the gains we may have if we release a number of mobile agents to visit growing number of nodes in a network. In Table 3.1, this scenario is depicted. Column 1 (m) is the total number of nodes to be visited. Column 2 is the number of agents cloned to visit the nodes in a multi-agent scenario. Column 3(p) is the number of sub-itineraries that would be created in a multi-agent scenario. Column 4 (t_m) is the time required to visit all the nodes (equation 4) while column 5 (t_p) is the time required to visit all the nodes when multiple agents are used and finally, column 6 is the time gained by using multiple mobile agents. It is observed that when a constant number of mobile agents are used (10 in this case), substantial time is gained. For instance, when number of nodes visited is 40, single agent requires 40 seconds to visit the 40 nodes while using 10 agents requires only 4 seconds, resulting in a gain of 36 seconds. When m = 300, a single agent visiting would require 300 seconds but 10 agents would require only 30 seconds, leading to a gain of 270 seconds. It is observed that using 2 mobile agents results in 50% time gain, using, 5 mobile agents results in 80% gain, while using 10 and 20 mobile agents result in 90% and 95% gains respectively.

Table 3.1: Varying node sizes, constant number of mobile agents

m	# of Agents	p	t_m(s)	t_p (s)	Time Gained	% Gain
40	10	4	40	4	36	90%
60	10	6	60	6	54	90%
80	10	8	80	8	72	90%
100	10	10	100	10	90	90%
120	10	12	120	12	108	90%
140	10	14	140	14	126	90%
160	10	16	160	16	144	90%
180	10	18	180	18	162	90%
200	10	20	200	20	180	90%
220	10	22	220	22	198	90%
240	10	24	240	24	216	90%
260	10	26	260	26	234	90%
280	10	28	280	28	252	90%
300	10	30	300	30	270	90%
320	10	32	320	32	288	90%
340	10	34	340	34	306	90%
360	10	36	360	36	324	90%
380	10	38	380	38	342	90%
400	10	40	400	40	360	90%

3.2 Scenario Two: Varying the node size and varying the number of agents

In this case, we vary the number of number agents as well as the size of the nodes. The result is depicted in Table 3.2. We attempt to vary the number of mobile agents from 5 to 23. We record a gain in time of between 80% to 95.65%. In other words, we have more gains in time as we increase the number of mobile agents launched.

3.3 Scenario 3: Constant number of nodes, varying number of mobile agents

In this scenario, we look at a node size of 400 and we vary the number of mobile agents from 5 to 23. The time required for the multiple agents to visit the nodes is inversely proportional to the number of mobile agents launched (Table 3.3). The percentage gain in time also increases with the number of mobile agents launched.

Table 3.2: Varying node sizes, varying number of mobile agents

m	# of Agents	p	t_m(s)	t_p (s)	Time Gained	% Gain
40	5	8	40	8	32	80.00%
60	6	10	60	10	50	83.33%
80	7	11	80	11	69	85.71%
100	8	13	100	13	88	87.50%
120	9	13	120	13	107	88.89%
140	10	14	140	14	126	90.00%
160	11	15	160	15	145	90.91%
180	12	15	180	15	165	91.67%
200	13	15	200	15	185	92.31%
220	14	16	220	16	204	92.86%

m	# of Agents	p	t_m(s)	t_p (s)	Time Gained	% Gain
240	15	16	240	16	224	93.33%
260	16	16	260	16	244	93.75%
280	17	16	280	16	264	94.12%
300	18	17	300	17	283	94.44%
320	19	17	320	17	303	94.74%
340	20	17	340	17	323	95.00%
360	21	17	360	17	343	95.24%
380	22	17	380	17	363	95.45%
400	23	17	400	17	383	95.65%

Table 3.3: Constant node size, varying number of mobile agents

m	# of Agents	p	t_m(s)	t_p (s)	Time Gained	% Gain
400	5	80	400	80	320	80.00%
400	6	67	400	67	333	83.33%
400	7	57	400	57	343	85.71%
400	8	50	400	50	350	87.50%
400	9	44	400	44	356	88.89%
400	10	40	400	40	360	90.00%
400	11	36	400	36	364	90.91%
400	12	33	400	33	367	91.67%
400	13	31	400	31	369	92.31%
400	14	29	400	29	371	92.86%
400	15	27	400	27	373	93.33%
400	16	25	400	25	375	93.75%
400	17	24	400	24	376	94.12%
400	18	22	400	22	378	94.44%
400	19	21	400	21	379	94.74%
400	20	20	400	20	380	95.00%
400	21	19	400	19	381	95.24%
400	22	18	400	18	382	95.45%
400	23	17	400	17	383	95.65%

4. Conclusion

In this paper, we present a comparative study of using single mobile agent to navigate a set of network nodes as opposed to using multiple mobile agents to visit segments of the nodes. From the simulations in section 3, we can clearly see the advantages of using multi-agent system when dealing with a network of large number of nodes. Theoretically, a gain of up to 50% is possible by introducing an additional clone of the mobile agent. When more clones are introduced, the gains rise significantly. This can enhance the activities of the network administrator when monitoring user activities in the network. However, this study did not investigate what could be the optimum number of mobile agents that can be released into the system. It is also assumed that there is no time lag in creating and dispatching the clones of the mobile agent. In most cases, there is a time factor in sprawling out the mobile agents. However, these may be negligible. Future works can investigate these areas.

REFENCES

[1] Shi, Z, and N. Zheng., Progress and Challenge of Artificial Intelligence,*Journal of Computer Science and Technology*Vol. 21 pp. 4-6, 2006.

[2] Jain, A. K., M. Aparico, and M. P. Singh. "Agents for process coherence in virtual enterprises." *Communications of the ACM* Vol. 42(3) pp. 62–69, 1999.

[3] Papazoglou, M. P., Agent-oriented technology in support of e-business, *Communications of the ACM* Vol. 44(4), pp. 71–77, 2001.

[4] Rothermel, K. and M. Schwehm., Mobile agents, In *Encyclopedia for Computer Science and Technology*, by A. Kent and J. G. Williams, edited by A. Kent and J. G. Williams. New York: M. Dekker Inc., 1998.

[5] Fuggetta, A., G.P. Picco, and G. Vigna. Understanding Code Mobility, *IEEE Transactions on Software Engineering*Vol. 24(5)pp. 342–361, 1998.

[6] Schlegel, T, P. Braun, and R.Kowalczyk, Towards Autonomous Mobile Agents with Emergent,Migration Behaviour AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan, *ACM*, 2006.

[7] Arekete, S.A., O.C. Akinyokun, O. Olabode, and B.K. Alese, Design of a Mobile Agent for Monitoring Activities of Users, *Computer Engineering and Intelligent Systems*, Vol. 4(2) pp. 33-44, 2013.

[8] Verma, V.K., R.C. Joshi, B.Xie, and D.P. Agrawal, Combating the bloated state problem in mobile agents based network monitoring applications,*Computer Networks (Elsevier B.V.)*Vol. 52, pp. 3218-3228, 2008.