Performance Measurement Analysis for Multi-Agent Systems

Naresh Kumar Nagwani,
Department of Computer Science & Engineering,
National Institute of Technology, Raipur
Email: nknagwani.cs@nitrr.ac.in

***Abstract*- A multi agent system (MAS) consists of number of software agents working together for an application. Agent communication, co-ordination and agent management are the important modules of any MAS, which makes MAS a very complex system. These complex systems require lot of analysis for keeping up the good performance. In this paper number of parameters associated with a multi agent system is discussed for performance analysis. Various categories of MAS communication models are also discussed. A profiling technique is also suggested for a multi agent system.**

Keywords*: *Multi Agent Systems, Performance Analysis, Performance Parameters, Agent communication models.

I INTRODUCTION

1.1 Agents

An agent is an autonomous software unit that can exist independently of other similar units in the software system. An agent performs some functions for other agents or external actors. Agents communicate with each other via messages via an agent communication language. As per the Foundation for Intelligent Physical Agents (FIPA), the agent definition is: "An agent is the fundamental actor in a domain. It combines one or more service capabilities into a unified and integrated execution model which can include access to external software, human users and communication facilities". [2]

1.2 MAS

Multi-agent systems (MASs) are systems consisting of more than one autonomous agent that are able to interact with one another. These agents may have a global goal to solve together, or they may individually have their own goals to pursue. The particular characteristic here is that in order to achieve their goal(s), these agents must coordinate their actions [1]

Multi-agent systems, which also are referred to as Distributed Artificial Intelligence (DAI) are naturally expected to act and collaborate in a distributed environment and across different hosts [4].

1.3 Performance of MAS

Just as throughput (in terms of transactions per minute (tpm)), response time, and the number of concurrent users and tasks are important performance indicators in distributed systems, they are equally vital in multi-agent systems. However, there are other important factors introduced by the agent level which have costs, and therefore affect performance. They include the following.

- Co-ordination — as in co-ordination protocols Employed
- The agent's knowledge model
- The agent's rationality model

The performance of a multi-agent system is a measure using a set of statistical indicators of the system's major outputs and its consumption of resources, where typical indicators include throughput, response time, number of concurrent agents/tasks, computational time and communications overhead[1].

For a multi-agent model, the variables that affect performance include, but not exclusively, the numbers of agents, the number of concurrent tasks/goals the agents are carrying out, the organization of the agents and the type of co-ordination protocols employed.

From the beginning agent platforms have focused on supporting communication between agents across networks. The dominating part focuses on message transport and agent communication. These specifications concern different levels of the communication, from the low-level message transport and communication protocols to the higher level abstract speech act theories for the message content. An increasing number of agent platforms try to comply with these specifications, and some of the most popular agent platforms are found, like Java Agent Development Environment (JADE) and Foundation for Intelligent Physical Agents – Operating System (FIPA-OS), in this category, given that they aim to support all kind of multi-agent systems [4].

There are various kinds of parameters involved in the performance of a multi agent system (MAS). Study of these parameters is very essential in order to improve the performance of MAS. By modifying the values of these parameters performance can be improved. For example making just a few assumptions the performance can be

978-1-4244-4711-4/09/$25.00 ©2009 IEEE

boosted of messages handling between agents by eliminating the overhead off network communication. Many existing multi-agent platforms try to avoid this overhead by grouping agents running on the same machine, but the abstract and general message envelopes still impact the scalability of the platform, when the communication increase.

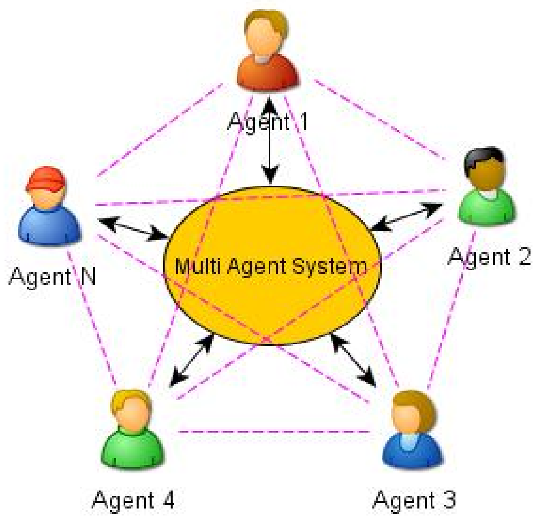

Figure 1. A Typical Multi-Agent System

It is assumed that the MAS is given a task that requires all the agents to work together in order to achieve it. The task, therefore, can be imagined divisible into *n* sub-tasks in a MAS consisting of *n* agents. The MAS uses a simple contract net protocol to co-ordinate the sub-task allocation problem.

1.4 MAS TYPES

Multi-Agent Systems (MAS's) can be classified based on the two parameters. First parameter is type of agents involved in MAS and second parameter is communication mechanism between the agents in MAS. There are 2 types of MAS based on the first parameter- Homogeneous MAS and Heterogeneous MAS. In Homogeneous MAS – All agents perform the similar tasks, whereas in Heterogeneous MAS – All agents perform the different tasks and implements different set of functionalities.

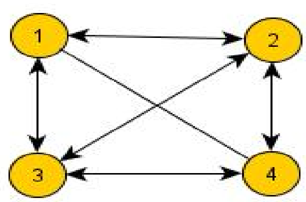

Figure 2. An example of Mesh Communication Model in MAS (With 4 Agents)

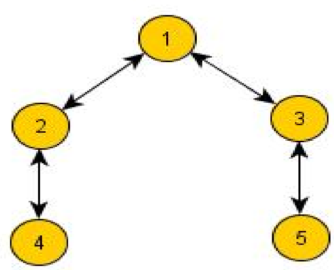

Figure 3. An example of Hierarchical Communication Model in MAS (With 5 Agents)

Using second parameter which is communication mechanism between the agents in MAS, A MAS can be classified as agents organized in either a mesh or a hierarchical structure, as shown in Fig 2 and Fig 3. In a mesh structure shown in Fig 2, every agent is connected to all other agents in the system; while agents in a hierarchical structure depicted in Fig 3, agents are connected only to their parent and children. The organizational structure defines the communication topology of the MAS, i.e. an agent can only interact with those agents to which it is connected [1].

II PERFORMANCE PARAMETERS

In this section numbers of parameters associated with the performance of MAS are mentioned. These parameters are directly and indirectly associated with performance of a MAS. These parameters are identified from the general architecture and implementation of MAS [3, 4].

A. *Number of Agents*: Number of agents in MAS is the primary parameter which indicates how large a MAS is. This parameter is directly proportional to the complexity of any MAS. If the value of this parameter is high, then complexity of the MAS will also be high.

B. *Computational Time for an individual Agent MAS*: This is the time at which any particular agent is in the Active state. This time is actually the total logical functionality execution time. Whenever an agent executes its assigned responsibilities, it enters into the active state. The total amount of time at which a particular agent is in the active it is called the computational time for an agent.

C. *Memory used by an individual agent in MAS*: The primary memory requirements for an individual agent are also a critical performance parameter. This memory is the sum of all the memory which is acquired by all the objects created by an agent.

D. *Agents Status* – This parameter indicates the transition of an individual agent in various states like Active, Wait etc. If the numbers of switch between the states are more then the performance will be poor.

E. *Agent Co-ordination in MAS* – This parameter is important for homogeneous MAS. In homogeneous MAS a complex task is divided into smaller activities are tasks among the agents hence co-ordination is an important thing here. Coordination is also different for Mesh and Hierarchical communication models.

F. *Communication* - communication between the individual agents and communication between coordinator and agents. In Mesh communication model all agents can communicate with each other, whereas in the Hierarchical communication model, agent communication happens through a proper communication hierarchy. Average message length, average number of messages to and from an individual agents and average message communication delay are the major parameters related to the communications of agents in multi agent systems.

G. *Agent Management* – Agent management deals with keeping track of agent's state, maintaining list of active agents and status of tasks allocated to the individual agent in a MAS. Agent manager keep track of how many agents have completed the assigned task and keep track of work completed and work remaining for a multi agent system. % of agents in active state and % of agents waiting for the invocation are some of the important parameters of agent management system.

H. *Dependencies between the agents* – identifying various types of dependencies between the agents. For example execution of second agent will start once first agent has finishes its tasks. Mostly it is related with the data segment where a particular agent is executing, so RAW (Read After Write), WAW (Write After Write) type of dependencies may exist in a MAS. For calculating the dependencies in MAS, dependency graphs can be created and dependencies can be calculated from these graphs.

III PROFILING OF MAS

Based on the given performance measurement parameters profiling can be done for a MAS to improve the performance. There are number of open source profiling tools available for this purpose, which can be used to find the values of performance parameters.

The following are some of the important tasks involved in profiling:

1. Memory Profiling: Memory profiling captures the memory allocated for a specific task during execution. When an agent will enter in the active state it, for performing the logical operations whatever objects the agent will be creating, the total memory allocated to these objects are captured by memory profiling techniques.
2. CPU Profiling: Amount of CPU utilization while executing a particular job is captured by CPU profiling. It is actually the CPU time spent for performing the logical tasks.
3. Remote Profiling: Profiling of a remotely placed application. This is applicable when the multi agent system is placed on a distributed environment. Agent can be placed in the different sites or nodes. In this case to generate and fetch the profile data remote profiling can be used.
4. Communication / Network Profiling: Network profiling deals with capturing the communication delay across a network for a network based application.

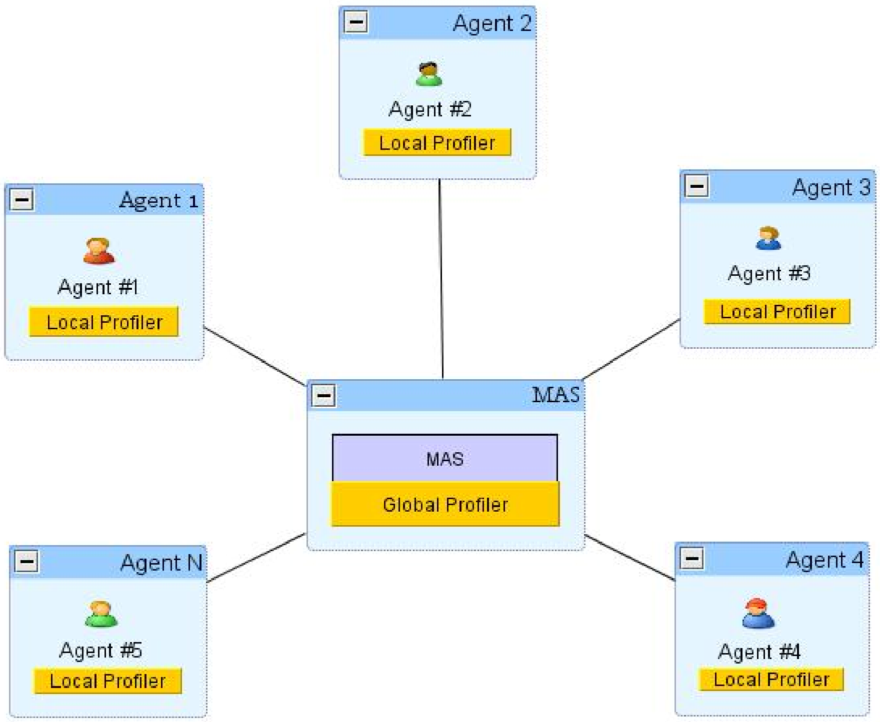

Figure 4. A Global Profiler is connected with all the local profilers in MAS.

Based on mentioned performance measurement analysis, following is the proposed model of MAS profiler, shown if fig. 4. Each agent has one local profiler and all the local profilers are connected with the global profiler. Here two terminologies are discussed which are important for MAS profiling.

- Local Profiler: The local profiler is associated with an individual agent, which is participated in MAS. Each local profiler captures memory consumed by an individual agent, CPU time taken by agent to perform the task etc. Each local profiler also keeps track of agent context switch between the various states.

- Global Profiler Data: The global profiler deals with profiling all the data captured by local profilers. It also deals with agent communication and management profiling. Mainly global profiler keeps track of there activities – Agent management, agent co-ordination and agent communication. Global profiler has the information about the communication model of multi agent system.

So a global profiler contains the values of all the performance parameters for MAS. By doing some variations and changes on these performance parameters based on certain number of assumptions performance of MAS can be achieved.

IV CONCLUSION & FUTURE DIRECTIONS

In this paper various parameters associated with the performance of multi agent systems are highlighted. Various models of MAS are discussed and impacts of performance parameters on MAS are mentioned. Profiling technique for MAS is also suggested to keep track of performance related data. The future work related to the paper could be creating some framework based on mentioned parameters and profiling technique to evaluate the performance of a multi agent system.

REFERENCES

[1] L C Lee, H S Nwana, D T Ndumu and P De Wilde: The stability, scalability and performance of multi-agent systems, BT Technol J Vol 16 No 3 July 1998.

[2] Margus Oja, Boris Tamm and Kuldar Taveter: Agent-Based Software Design, Proc. Estonian Acad. Sci. Eng., 7, 1, 5–21, 2001.

[3] Hillol Kargupta, Ilker Hamzaoglu, Brian Stafford: Scalable, Distributed Data Mining Using an Agent Based Architecture, Proceedings of High Performance Computing, 1997.

[4] Kasper Hallenborg: Core Design Pattern For Efficient Multi-Agent Architecture, International Book Series "Information Science and Computing", pp 29-36, 2004.