UNIVERSITY OF CALGARY	Course Number: SENG 609.22	Course Name: Agent-based Software Engineering
	Session: Fall, 2003	Department: Electrical and Computer Engineering
		Document Type: Tutorial Report

Examining Performance Issues of Multiagent Systems

"Computers can figure out all kinds of problems, except the things in the world that just don't add up."
James Magary

SENG 609.22
Kendra Hamilton
December 2003

Abstract

As the implementation of multiagent systems becomes increasingly common, performance measurement starts to become increasingly of concern. This concern, of course, is not unique to multiagent systems however, the unique qualities of such systems demands further investigation into performance related issues. The characteristics of autonomousity, learning, and possible mobility of agents add challenges to performance measurement. Furthermore, these very characteristics are also some of the key agent characteristics, which need to be measured in terms of performance.

Introduction

One of the defining characteristics of a software agent is that an agent is *interactive*; the agent can communicate with other agents within its environment. A second characteristic is that an agent can *learn* and make decisions about its environment.

As agent systems and the hosts and networks in which they operate increase in size and complexity, effective performance measurement becomes increasingly critical to the success of the agent system. The attributes of interactivity and learning of agents provide both a challenge and an opportunity to the system's designer particularly in terms of performance. Ideally the agent should be designed to be aware of its own performance, the performance of the society in which it operates, the performance of the system in which it operates, and take appropriate steps related to performance in order to meet its individual goals and the goals of the society.

This paper examines performance measurement and analysis in terms of agent systems. In this paper, performance will be considered using the following definition, "The performance of [an 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 (Lee et al., 1998)." Issues such as the quality of the finished task are not discussed.

Why Measure Performance?

- Relationships between customer and service provider are now regularly governed by contracts or more informally Service Level Agreements. These documents often provide performance measurements and performance levels which must be met by the system in order for the customer's business to benefit. The agent system must be able to measure its performance in order for the service provider and customer to determine whether these service levels are being met and thus providing benefit to the customer.

- Regular performance measurements allow the system to monitor trends and to determine if any measurement is significantly off the expected level. If it is, the

system can choose to react proactively, such as notifying a system administrator, rather than waiting for the system users to notice the diminished performance.

- Of specific significance to agent systems is the ability of the agent to gather information about its surrounding environment and autonomously determine how to best adapt in order to improve performance.

- Well designed performance measurements of the agent system can give system designers and overall high-level view of the system. This view can help the designer identify performance bottlenecks and take corrective action.

- When first implemented, new systems have relatively few requests made of them. However, with time the number and frequency of requests grows. Regular performance measurement and analysis allows the system designer to determine the effects the increased demand is having on system performance and to extrapolate expected future performance.

- Performance measurements allow software agent researchers to benchmark agents implemented using different technologies against each other. The results of benchmarking can provide system designers of agent technology with valuable information regarding their product. Furthermore, benchmarking results can provide application system designers information on agent technologies under review for future software development.

Issues with Performance Measurement Gathering

The quantum Zeno effect tells us that the act of measuring a system affects the system such that the behaviour may be different than it would have been if the system was not measured [Bruce 1997]. The designer needs to be cognizant of this when designing an agent to measure its behaviour. If too many measurements are made and significant resources are spent performing the measurements and analyzing the results, the system may perform poorly. However, if too few measurements are made the agent well have insufficient information with which to learn and make decisions regarding behavior, which will have the most positive impact on performance. This is particularly important given that the nature of mobile agents who regularly traverse the Internet which continually involves and changes. Performance measurements, which were appropriate today, may no longer be appropriate tomorrow.

A mobile agent system is composed of multiple autonomous agents, which may be in an undetermined state at any given moment in time. However, it is often desirable to take performance measurements at discreet points in time. The coordination needed to synchronize performance measurements over a mobile agent system may be difficult to achieve. In order to eliminate issues associated with synchronized performance measurements, the designer may allow each agent to take measurements as the agent deems appropriate it. While this approach may reduce some of the logistical issues of coordination, it introduces problems with data analysis.

In the case of mobile agents, the host on which the agent executes and the network the agent uses to pass messages can change at any time. Moreover, the system designer will likely have no control regarding the configuration, the performance potential, or the performance demand of the host system and network. Therefore, performance measurement cannot be done as a task which is pursued for time and then forgotten; rather, performance measurement and analysis must be an ongoing task which is designed into the agent system.

While the need for multiple performance measurements has been previously discussed, the diversity of agents tasks' and goals' within the system can make it difficult to decide upon appropriate measurements. For example, network bandwidth is of more concern to an agent which is mobile than one which is not.

Metrics Design

"I hate everything that merely instructs me without augmenting or directly invigorating my activity."
Goethe

According to Helsinger, the system designer should consider a number of issues when designing performance metrics.

- Who will use the results? Is the intent that individual agents will collect and analyze the results? Are the metrics designed to help the system designer improve system performance? Perhaps the metrics are to be used to alert the system administrator of an unusual condition.
- What type of information does the metric represent? Performance can be measured at regular intervals over a long period of time. This type of measurement can be an excellent source of historical trend analysis, useful for detecting large performance patterns or for projecting future system behavior. Alternatively, performance metrics that measure the current status of a specific element can alert the system to potential problems if the current state continues.
- How will data be transmitted and analyzed? One model is to have it agents transmit the raw metrics to a central agent for analysis. This approach has the benefit of reducing the processing load on individual agents and allows the central agent to consider the whole system when recommending actions to improve performance. However, this approach uses significant network resources in transmission and has increased time lag between requests for data and receipt thereof. A second model is to have several agents spread throughout the system, which receive and analyze the metrics. This approach can reduce transmission costs and the time lag issue but does not provide for possibility of making recommendations with the whole system in mind.
- What level of performance does the metric measure? Metrics can consider low-level tasks such as CPU utilization or network utilization. Other higher-level metrics may consider total time to complete a user request. Each of these levels of abstraction must be considered when designing for different users of the

metrics as previously discussed. As well, data transmission methodology must be considered. Application level performance analysis cannot be completed unless measurement data is combined into a central location.

Possible Metrics

Below is a list of metrics in use in software agent systems as reported in current literature. It should be noted that most studies thus far deal with the category of mobile agent systems rather than the broader category of agent systems.

1. Measuring the size of software agents can provide guidance to system administrators to the resources needed to support the agent system. Changes to agent size can alert the system administrator to system growth. In mobile agents, agent size affects system performance and thus this metric can provide clues to possible system bottlenecks.
2. As described above, one of the key characteristics of agent systems is the sharing of knowledge between agents (communication performance). Therefore, network usage statistics should be collected by individual agents. In the Cougaar system (Helsinger, 2003) agents determine whether or not messages should be compressed depending on network usage. If the usage is high, the messages are compressed even though this approach takes more CPU resources. When usage is low, messages are not compressed in order to save CPU resources.
3. For mobile agent systems, possible metrics include measuring the time spent and resources needed to create the agent on the local host, to create the agent on a remote host, to travel from one host to another host, and time to issue a remote call (mobility performance, reproduction performance). These metrics are important for the agent to help it decide whether or not to move from its current host to another host when requiring the services of the other host or whether or not to simply perform a remote call.
4. In order to satisfy customer requirements, application level metrics (operation performance) must be considered as well. For example, a possible example of search agents is the number of information sources reviewed. This metric may be combined with user quality surveys to determine whether more or less information sources should be reviewed and whether the user will accept the longer wait for increased searches.
   Operation performance metrics (time taken to complete a task or achieve a goal) can be taken at the agent level or the system level.
5. One of the features of software agents is their ability to learn. Measurement of how long it takes an agent to learn (learning performance) and make adaptations to its behaviour based on the learning can help designers determine the strength of their design.

Conclusion

Clearly, establishing strong performance measurement techniques is of importance to the successful implementation of multiagent systems. While significant work in the area of performance measurement has been often been undertaken with regards to procedural

software programs and client server designs, there is a lack of such study in the area of multiagent systems. Most literature and research regarding performance and agents is limited to the aspect of mobility in mobile agent systems and utilization of network and CPU resources. Unfortunately, little research has been done on some of the unique characteristics of agent systems such as the ability to learn and reason in order to achieve the goals of the agent. Hopefully, continued research in the area of agent systems will lead to a more complete understanding of the issues of performance and methods of measuring performance. Furthermore, the Coougar system (Helsinger, 2003) has demonstrated the ability of agents to take performance measurements, learn from the measurements, and change their behaviour. The system provides an example of the exciting opportunities provided by agent systems in the area of performance measurement eventually leading to performance improvements.

Bibliography

Bruce, Colin “The Einstein Paradox”, pp 249-250, Perseus Books, Massachusetts, United States,1997.

Dumke, Reiner, et al, “Performance Engineering in Agent-based Systems: Concepts, Modeling and Examples”, http://ivs.cs.uni-madeburg.de/~wille/forschung/canada.pdf

Helsinger, Aaron, et al., “Tools and Techniques for Performance Measurement of a Large Distributed MultiAgent Systems”, International Conference on Autonomous Agents; Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, pages 843-850, 2003.

Elarde, Joseph V., Brewster, Gregory B., “Performance Analysis of Application Response Measurement (ARM) Version 2.0 Measurement Agent Software Implementations”, IEEE Performance, Computing, and Communications Conference, 2000, pages 190-198.

Lee, L.C. et al., “The Stability, Scalability and Performance of Multiagent Systems” BT Technology Journal, Vol 16 No 3, July 1998.

Plale, B., Schwan, K., “Run-time Detection in Parallel and Distributed Systems: Application to Safety-Critical System” International Conference on Distributed Computing Systems, pages 160-170, June 1998.