A DYNAMIC LOAD BALANCING POLICY FOR AGENT-BASED DGIS*

LUO Yingwei, WANG Xiaolin and XU Zhuoqun

Dept. of Computer Science and Technology, Peking University, Beijing, 100871

ABSTRACT

Geo-Agents is an agent-based distributed Geographic Information System (GIS), which is a multi-agent system and consists of four types of GIS agent: GuServer, Facilitator, interface agent and GIS function agent. For a multi-agent system, there contains many distributed agents in different environments, which form a complex distributed system. In order to improve the performance, load control should be a keystone in a multi-agent system. Firstly, according to the features of distributed GIS, the load problems in Geo-Agents are analyzed; and then, aiming to host load and network load, two load control strategies adopted in Geo-Agents – agent scheduling strategy and peer host strategy, are discussed in detail.

KEY WORDS: Geo-Agents; Load Control; Host Load; Network Load; Peer Host

1 Geo-Agents: AN INTRODUCTION

Geo-Agents is an agent-based distributed GIS developed by AI lab, dept. of computer science and technology in Peking University (see Figure 1)[1][2], which is a multi-agent system and consists of four types of GIS agent: GuServer, Facilitator, Interface agent and GIS function agent.

Facilitator is the manager of Geo-Agents, which cooperates with other Facilitators to control and coordinate every GIS agent to run correctly, and holds the distributed controls of the system.

GIS function agent encapsulates a specific spatial querying, spatial processing or spatial analyzing task. According to the features of GIS, GIS function agent is classified into two types: basic function agent and domain-oriented function agent. Basic function agent completes basic services in GIS, including spatial data accessing, network analyzing, overlay analyzing, buffer analyzing and so on. Domain-oriented function agent is responsible for modeled tasks in various domains, which can be constructed by models and

[*] Corresponding author: LUO Yingwei, lyw@pku.edu.cn.

can be used generally in that domain.

Interface agent provides interfaces for users or applications to hand task. Just like Structure Query Language (SQL) of database, Geo-Agents provides GeoScript language to describe GIS task [2]. When solving a practicable problem, users or applications can simply use GeoScript statements to describe the task and hand the statements to interface agent, then interface agent will disassemble the task to subtasks autonomously, and recruit GIS function agents to complete the task concurrently.

GuServer is in charge of spatial information accessing services, which manages spatial information and spatial metadata in spatial databases.

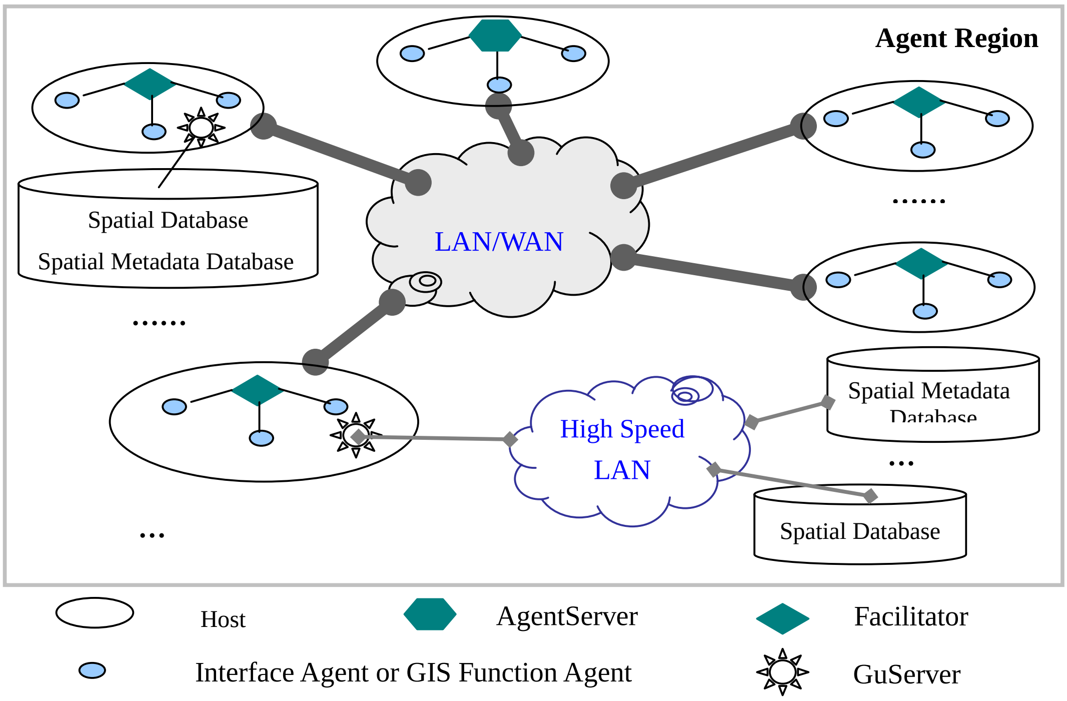

Figure 1. Geo-Agents System

In Geo-Agents, every GIS agent consists of five units: control subsystem, functional subsystem, communication subsystem, human-computer interface and data resource. It is a reactive agent, which is not only able to carry out its own task independently, but also communicate with other GIS agents, exchange information and cooperate with others.

"Agent Region" mode is adopted to manage agents distributed on network in Geo-Agents. An "Agent Region" consists of one or more hosts, which must be installed with Facilitator (and or other GIS agents). In an "Agent Region", there is one and only one Facilitator that will be configured as AgentServer. Facilitator is used to manage and coordinate GIS agents in "Agent Region", and AgentServer is the bridge between two "Agent Regions". That is to say, an agent in one "Agent Region" can communicate with agents in another "Agent Region" only through AgentServer. Of course, an authorization from target "Agent Region" is required.

A multi-agent system contains many distributed agents in different environments, which form a complex distributed system. In order to improve performance, load control should be a keystone in a multi-agent system. The existing researches imply that load control can improve performance of distributed system very well [3-6]. In multi-agent system, agent is the minimum execution unit, and the load control is to dispatch all subtasks for a user's task to different agents in different hosts averagely [7]. The load control strategy in Geo-Agents is introduced here. Firstly, the load problem in Geo-Agents is analyzed, and then two load control strategies, agent scheduling strategy and peer host strategy, are adopted to improve the performance of Geo-Agents.

2 THE LOAD PROBLEM IN Geo-Agents

Spatial data is often involved a large spatial scope, and the content is diversity. So people store spatial data in different places according spatial scope and content. Distributedness is an intrinsic feature of spatial data. But in another hand, massive related spatial data for a same spatial scope is always stored in a same spatial database, so spatial data has another feature of centralization.

The distributedness of spatial data brings many inconveniences because a spatial task always uses many kinds of spatial data from different spatial databases. In this situation, massive spatial data will be transferred from one host to another host frequently in network.

Because of the centralization, the needed data resource for many GIS tasks may be in one host. In this situation, those tasks will be completed in one host, so as the host becomes overloaded.

Generally, the load problem that a distributed system faces may involve two aspects: host load and network load [3]. In Geo-Agents, host load indicates the number of agents resided on one host, and network load contains two parts: the communication among agents and the transmission of spatial data. For any distributed GIS system, the transmission of spatial data is the primary network load. Spatial data is not transmitted directly in a message in Geo-Agents. Spatial data is encapsulated as a Geo-spatial object, and the transmitted data in a message is the reference of the Geo-spatial object. Only when spatial data is really needed, the transmission of spatial data takes place. The transmission of spatial data is through the reference of Geo-spatial object, not through the message among agents [2].

In order to achieve a high performance, it is necessary to control the load for a distributed system. Generally, load control can be accomplished from two aspects: reducing load and balancing load [4][5].

In Geo-Agents, the number of agent for completing a special task is fixed, it is impossible to reduce host load. But we can balance the host load through controlling the number of agent in different hosts.

Network load is a serious problem in Geo-Agents, which is because of that: 1) the collaboration among agents will bring a lot of communication cost; 2) massive spatial data is transmitted in network. The communication mechanism in Geo-Agents is point-to-point direct communication, so it is hard to balance network load among different hosts. But we can reduce the network load through some other effective measures.

Reducing network load can be obtained from two aspects: 1) reducing the quantity of each communication, which must be considered when designing system; 2) reducing communication times, especially massive spatial data communication times[5][6]. In Geo-Agents, the distributedness of spatial data and the collaboration among GIS agents make communication absolutely necessary. The only way is to let massive spatial data communication take place in a same host.

In those ways, it is easy to find that a conflict happens between reducing network load and balancing host load. In Geo-Agents, because of the hugeness of spatial data, reducing network load is principal. Balancing host load will be considered only after network load is accordant.

3 LOAD CONTROL IN Geo-Agents

In Geo-Agents, load is mainly controlled when GIS agent is created. There are two control strategies: agent scheduling strategy and peer host strategy.

3.1 Agent Scheduling Strategy

3.1.1 The Principles to Create an Agent

GIS Agent is designed as reusable GIS agent and ordinary GIS agent [2]. Reusable GIS agent means its result can be used again.

When creating a reusable GIS agent, if its type and task are same with an existing active reusable GIS agent (in Geo-Agents, there are some determinate reusable GIS agent types, and the task of each type is comparable [2]), it is unnecessary to create a new reusable GIS agent. Otherwise, the requirement to create a reusable GIS agent is same with creating an ordinary GIS agent.

When creating an ordinary GIS agent, there are some different requirements because of considering network load:

(1) An GIS agent must be created in an appointed host.

(2) An GIS agent must be created in local host by Facilitator.

(3) An GIS agent will be created in appropriate host(for example, the host where spatial data locates).

(4) An GIS agent can be created in any host.

In the first two cases, the host load they bring is inevitable, and there is no proper method to reduce or balance relative host load. But in the last two cases, it is necessary to control the creation of agent because of the randomicity, otherwise, there will have too many agents in one host.

During the creation of GIS agent, the network speed must be considered. In Geo-Agents, some assumptions are given: accessing an agent in a same “Agent Region” is faster than in other “Agent Region”; accessing data resource in a same “Agent Region” is faster than in other “Agent Region”; the priority of accessing data resource is higher than accessing agent. For example, supposing there are two “Agent Regions”, the data resource is in “Agent Region1”, and a user in “Agent Region2” wants to create an agent to process the data resource in “Agent Region1”. At this situation, the agent should be created in “Agent Region1”.

According to above requirements, the following principles must be obeyed when creating an agent:

(1) an agent should be created in the host where data resource it processes locates if possible.

(2) if an agent needs to process several data resources, the agent should be created in one of the hosts where data resources locate if possible.

(3) if an agent couldn’t be created in the host where data resource locates, it should be created in one of the “Agent Regions” where data resources locates if possible.

(4) an agent should be created in the same “Agent Region” with its requestor if possible.

(5) in a “Agent Region”, an agent should be created in the host where its load is light.

3.1.2 Agent Scheduling Algorithm

Scheduling agents when creating agents is a powerful means to control host load and network load. The goal of scheduling agents is: 1) making the most of CPU of all hosts, in another word, eliminating the situation that some hosts are too busy and some other hosts are idle at a same time; 2) Shorting the average response time of all tasks and reducing the network transmission as greatly as possible.

When designing an agent scheduling algorithm, three problems must be considered firstly[5]:

(1) Determinate or heuristic algorithm: determinate algorithm is adopted only when the actions(such as computing requirements, data requirements, communication requirements and so on) of all agents are foreseeable. But if the load of a system is unpredictable, a heuristic algorithm is needed. Heuristic algorithm means that the host allocation is instructed by experiential rules and heuristic information during the working progress.

(2) Distributed or centralized algorithm: centralized algorithm must gather all global information to give a decision, but distributed algorithm can give a decision only by some partial information. Gathering all global information in centralized algorithm brings heavy burden to the decision-making host, which will become the bottleneck of a system.

(3) Optimized or hypo-optimized algorithm: centralized algorithm can obtain optimized result, but the cost is higher than hypo-optimized result, because more information and complete processing are needed.

According to the agent creating principles and the agent scheduling goals, a heuristic, distributed and hypo-optimized agent scheduling algorithm is designed:

In Geo-Agents, agent scheduling is completed by AgentServer. When creating an agent, a request is sent to AgentServer to ask it find an appropriate agent type. So AgentServer maintains a host-table which records the number of created agents in each host. If an appropriate agent type can be found only in one host, the agent should be created in that host, and the number of created agents in that host increments by 1; if an appropriate agent type can be found in several hosts, after considering the agent creating principles, AgentServer will select the host where the number of created agents is minimum to create the agent, and the number of created agents in that host increments by 1.

We call this scheduling algorithm a heuristic, distributed and hypo-optimized algorithm because that: 1) it is incapable to predict that how many agents will be created in the system and how many agents are executing at a same time; 2) the scheduling criterion is the number of created agents, not the number of executing agents in one host; and 3) the scheduling criterion is obtained at the volley and it doesn’t bring additional burden for AgentServer.

3.2 Peer Host Strategy

When it is foreseeable that the host load in one host is too heavy, the peer host strategy is adopted to reduce the host load: connecting the host with some other hosts via high speed LAN, which have same software and hardware configuration with the host and can complete same tasks with the host. The host is called as peer main host, and other hosts are called as peer secondary hosts. All main and secondary hosts form a peer host group(see Figure 2). When creating an agent in main host, if the host load of main host is too heavy, the agent can be created in anyone of the secondary hosts. Of course, in order to balancing the host load of all peer hosts, the host where the host load is lightest is selected.

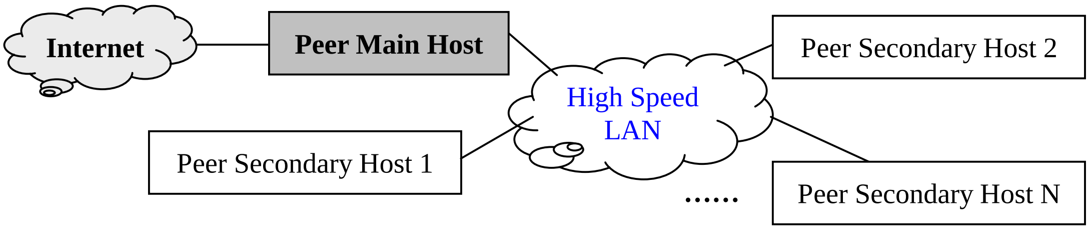

Figure 2. Peer Host Group

In Geo-Agents, the host crying for “peer host” is the hosts where GuServer locates. GuServer maintains one or more spatial databases and spatial metadata databases [2], and is responsible for managing and accessing spatial data and spatial metadata. Spatial information is centralized in the host where GuServer locates. According to the agent scheduling strategy, all agents who need GuServer to manage and access spatial information will be created in the host where GuServer locates. So the host load of the host where GuServer locates will be too heavy, and the performance of the system will be serious low. Of course, enhancing the software and hardware configuration of the host where GuServer locates can improve performance of the system, but cannot reduce the host load of that host. “Peer host” will play an important role here(see Figure 3).

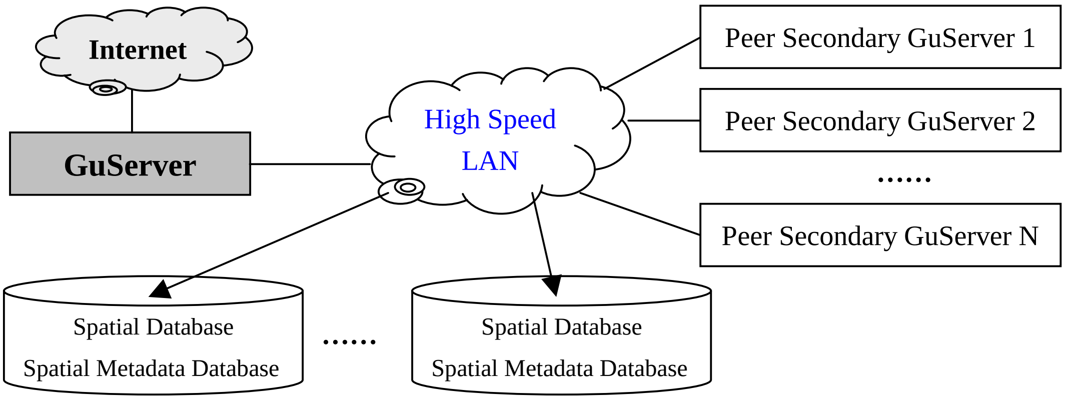
Internet
GuServer
High Speed
LAN
Peer Secondary GuServer 1
Peer Secondary GuServer 2
......
Peer Secondary GuServer N
Spatial Database
Spatial Metadata Database
......
Spatial Database
Spatial Metadata Database

Figure 3.The Peer Host Group of GuServer

Here, the host installed GuServer is main host, and the hosts installed peer secondary GuServer are secondary hosts. All GuServers manage and access same spatial data bases and spatial metadata databases. When creating agents who will use spatial information managed by the GuServer, they can be created averagely in the peer host group of GuServer.

4 CONCLUSION

The performance is an important factor to evaluate a distributed system, and load control is a powerful means to improve performance. In Geo-Agents, Agent scheduling is a random load controlling strategy used to balancing the load of whole system. Agent scheduling is an intrinsic feature of Geo-Agents. During the running time of Geo-Agents, Agent scheduling strategy automatically controls the network load and host load among Geo-Agents according to its all-around running information. Peer host is a load controlling strategy through man-made configuration. The goal of peer host strategy is to reduce host load of a certain host. When constructing a distributed GIS application using Geo-Agents, if the heavy host load of a certain host can be predicted, peer host strategy will be adopted for this host. The test results of Geo-Agents prototype system and its applications show that, the performance of Geo-Agents is improved very well by these two strategies [2].

ACKNOWLEDGEMENTS

Supported by the National Natural Science Foundation of China under Grant Nos. 60030005, 60073016, 60203002; the National High Technology Development 863 Program of China under Grant Nos. 2001AA113151, 2002AA135330, 2002AA134030; the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB312000; the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20020001015; the Natural Science Foundation of Beijing of China under Grant No. 4012007.

REFERENCES

[1] Luo Yingwei, Wang Xiaolin and Xu Zhuoqun (2002). "The Model of Distributed GIS-oriented Multi-agent System", *Acta Scientiarum Naturalium Universitatis Pekinensis* (in Chinese), 2002.5, Vol.38, No.3: 375-383.

[2] Luo Yingwei (1999). "A Study on Agent-based Distributed GIS" [PhD Dissertation](in Chinese), Beijing: Peking University, 1999.6.

[3] Zhou S.(1988). "A Trace-Driven Simulation Study of Dynamic Load Balancing", *IEEE Trans. on Software Engineering*, 1988, 14(9): 1327-1341.

[4] Shivararri G, *et al* (1992). "Load Distributing for Locally Distributed Systems", *IEEE Computer*, 1992, 25(Dec): 33-44.

[5] Lin Hwa-chu, Raghavendra C S. (1992). "A Dynamic Load Balancing Policy with a Central Job Dispatch", *IEEE Trans. on Software Engineering*, 1992, 18(2): 148-157.

[6] Luo Zhigang, *et al* (2000). "Study on CORBA Load Balancing", *Computer Science* (in Chinese), 2000, 27(10): 31-35.

[7] Myung-Hee Jo, Yun-Won Jo, Jeong-Soo Oh and Si-Young Lee (2001). "The Design and Implementation of Dynamic Load Balancing for Web-based GIS Services", *The proceedings of 22nd Asian Conference on Remote Sensing*, 5-9 Nov., 2001, Singapore.