International Journal of Computer Applications and Technology
Volume 1- Issue 2, 2012, 44-48

International Journal of Computer Applications and Technology
journal homepage: www.ijcat.com

Agent based Task Scheduling in Grid

Ashish Chandak
Department of Computer Science and Engineering, National Institute of Technology, Rourkela
Rourkela, Odisha, India

Bibhudatta Sahoo
Department of Computer Science and Engineering National Institute of Technology, Rourkela
Rourkela, Odisha, India

Ashok Kumar Turuk
Department of Computer Science and Engineering National Institute of Technology, Rourkela
Rourkela, Odisha, India

Abstract: Grid computing is considered to be wide area distributed computing which provides sharing, selection and aggregation of distributed resources. Agent paradigm has been widely used in large number of research area and now a days it is widely used in grid computing. In this paper, we proposed agent based strategy that uses knowledge base reasoning framework for task scheduling in grid which minimizes makespan.

Keywords: Grid Computing; Task Scheduling; Agent; Knowledge Base.

1. INTRODUCTION

Autonomous agents are intelligent entities that can operate on behalf of the human users autonomously to solve the problems, negotiate with other agents (peers), learn from the past and predict upcoming events. Agents are used in various application domains such as industrial applications viz. process control [1], commercial applications viz. electronic commerce [2], business process management [3], medical applications viz. patient monitoring [4], health care [5]. They get the problem from the users or other agents, discover needed resources, consult with other agents (negotiation) and offer a proper solution. They also learn from the past, update their knowledge and predict the future events [6]. Agent technology is new in distributed system and can be used in computing network. The main difference between agent and scheduler is coordination, cooperation and learning [6]. Agents work together, use the resources located on each other optimally and work as a team to solve a problem. Agents are flexible entities and are capable to adapt themselves to new environments. This means agents are well suited in dynamic environment. Even though agents are independent, they always communicate with others to discover needed resources. Following are some properties of agent:-

- Autonomous: - Agents are proactive, goal directed and is capable of acting without direct external intervention.
- Interactive: - Communicate with the environment and other agents.
- Adaptive: - Agents dynamically adapt to and learn about their environment. They are adaptive to uncertainty and change.
- Cooperative: - Able to coordinate with other agents to achieve common purpose.
- Social: - They work together.
- Coordinative: - Able to perform some activity in a shared environment with other agents, via plans, workflows, or some other process mechanism.

The agent paradigm has been successfully used in a large number of research areas. An agent-based methodology is developed for building large-scale distributed systems with highly dynamic behaviors [7,8]. Authors in [9] use ordinal sharing learning (OSL) method based on multi-agent reinforcement learning to solve the task scheduling problem in grid. A combination of intelligent agents and multi-agent approaches is applied to both local grid resource scheduling and global grid load balancing is used in [11]. Authors in [10] applied use of economic agent in grid computing. What distinguishes our work from other is that we applied knowledge base reasoning in agent for task scheduling. This chapter describes the conceptual agent framework which provides the management of various resources in grid environment. The design goals of our model focus on combining distributed resources and knowledge base reasoning agent technology to manage various resources on the grid. The evaluation of our proposal shows the advantage of using agents in grid computing.

2. AGENT DESCRIPTION

Agent is an autonomous entity that can interact with its environment. In other words, it is anything that can be viewed as perceiving its environment through sensors, and acting upon on environment with effectors. Agent processing overview is shown in Fig. 1.
An agent system is represented as:
Agent System=< Agents, Coupling >
An agent is defined in a formal way as: <$Agent_{Id}$, Role>
$Agent_{Id}$: Agent identification, which is used in order to identify every agent of the system.

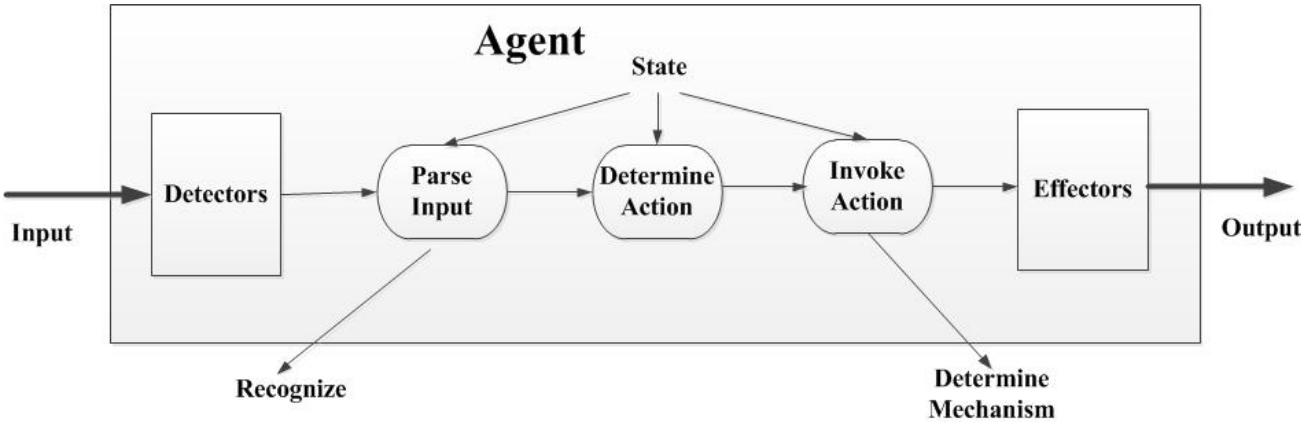

Fig. 1 Agent Processing Overview

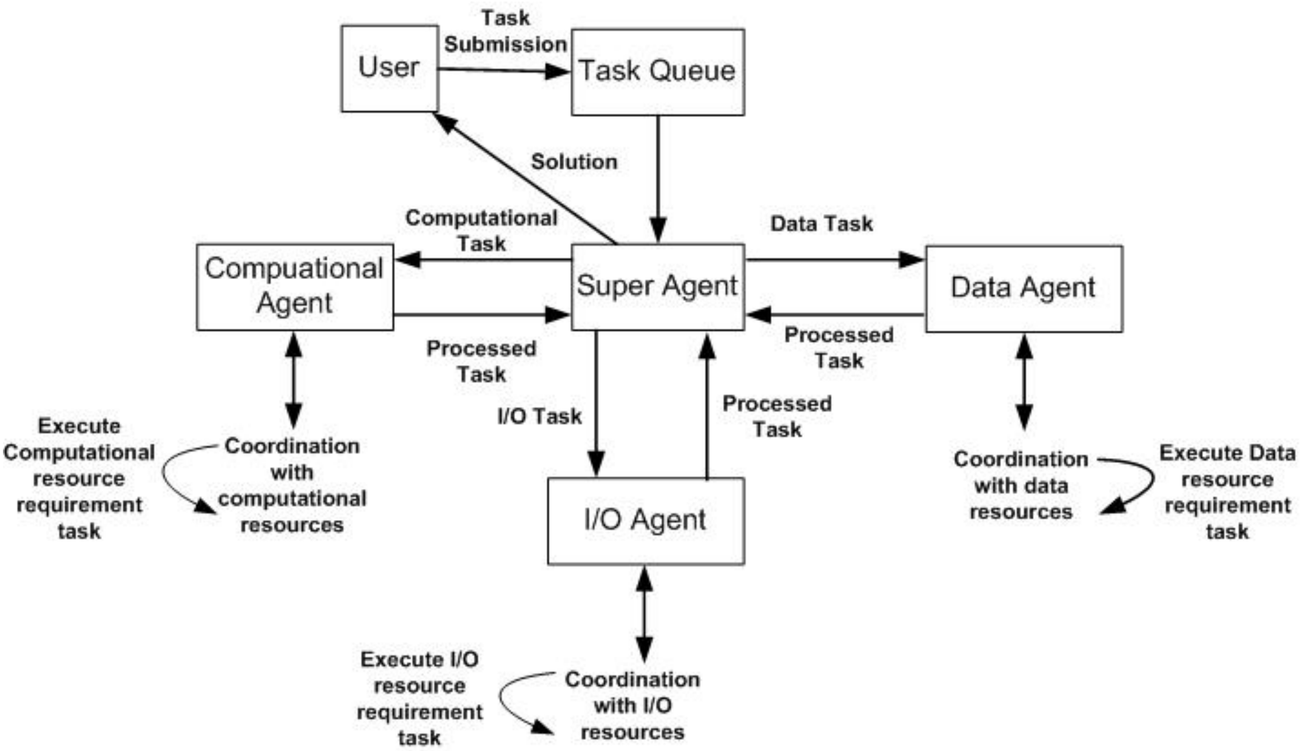

Fig. 2 Task Execution in Agent System

Role: This field represents the kind of agent i.e. computational, I/O or data.
Coupling is a mapping of an input and output from/ to with other agent.

3. TASK EXECUTION MODEL

Fig. 2 shows how a solution can be found for a specific problem using cooperation of the super-agent and other agent. Super-agent, after receiving a task is processed internally and each task is sent to the responsible agent for further processing. Results are sent back to the super-agent and a comprehensive solution is offered by agent. In our model, agent has

- the ability to link resources with/into the grid via some kind of interconnection mechanism.
- the ability to use grid resources to perform some task.
- the ability to compose grid resources to form new combined resources that can be used in the same way as the individual resources.

Internal architecture of agent is shown in Fig. 3. Agent can add new information obtained from the result of inquiries to expand their current knowledge. This causes agent to learn about what they received from other agent and to extend their current information about problems and solutions. It also prevents redundant inquires and searching. Agent can offer solution, a set of results created by inquires, for the users's problems that consist of group of low-level tasks. A problem can solve by many agents with different knowledge. The agent consists of several internal modules. Modules are responsible for performing the internal activities of the agent.

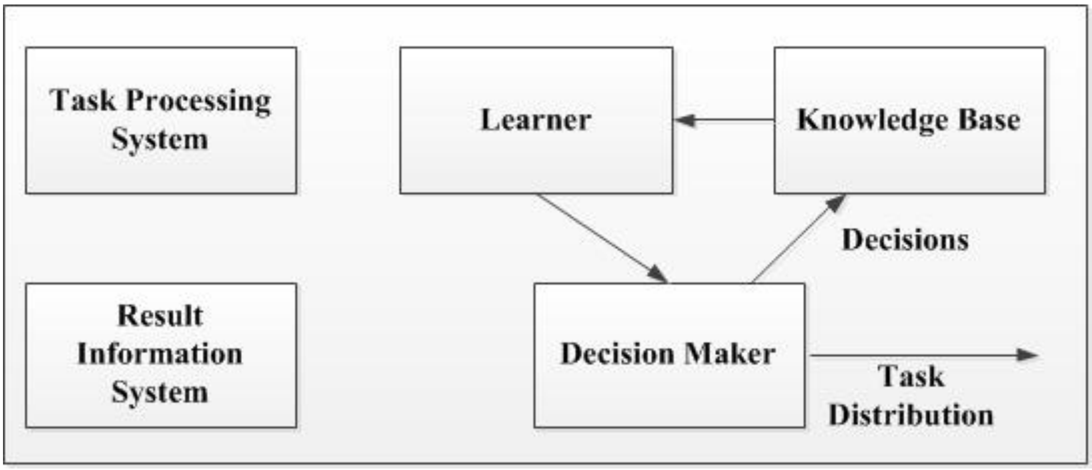

Fig. 3 Internal Architecture of Agent

Knowledge Base No. of attempt for 'X' resource	Node A	Node B	Node C
Search No. 1	Failure	Success	Success
Search No. 2	Success	Success	Failure
Search No. 3	Failure	Success	Success
Success Ratio	1/3	2/3	3/3
Ranking	3	1	2

Table 1: Inquiry Table maintained by Knowledge Base

Cooperation of the modules enables the agent to perform the required tasks. Following are components of agent.

- Result Management System (RMS):- This module collect and return results to super-agent
- Knowledge Base: - This module is responsible for storing and retrieving in formation from the result management system and learner module, which carry resource related data and task execution results respectively. Knowledge base includes computing capability and status of every node of each site. The table 1 shows inquiry table maintain by knowledge base. Ranking helps to know which node is best match for task 'X'. Ranking shows sequences of the destinations for the next search. Thus before sending inquires agent will decide which site should be targeted first. The inquiry table is updated every time when an inquiry is issued. We can see that node B should be first destination for sending next inquiry.
- Learner: - Learner takes the input from knowledge base regarding resources. Learner always looking for newer solutions and updates it to knowledge base according to latest results obtained from inquires and overwrites the old results. For example, if computational agent after refereeing its internal knowledge realizes that if task came with x demand then it is forwarded to x resource without any further inquiry. This technique prevents iterative inquires for resources.
- Decision maker: - Decision maker is central part of an agent and center of operations. Decision maker takes the resource selection decision and forward task to appropriate nodes. Decisions taken by this module are updated into knowledge base.
- Task processing system: - It process tasks.

4. SIMULATION EXPERIMENT

We developed a simulation application in MATLAB to carry out the experiments. Each simulation experiment ends when 1000 tasks executions gets completed. The arrival of tasks is modeled as Poisson random process. To evaluate performance we have considered following three types of tasks: a) I/O intensive tasks b) Data intensive tasks c) Computational intensive tasks. We consider nine nodes for computation. In without agent based scheduling, task are randomly assigned to any of the available nodes regardless of task type but in agent based scheduling, first of all incoming task type is determined and allocated to appropriate node i.e. I/O task is assigned to I/O specific node, data task is assigned to data specific node, and computational resource requirement is assigned to computation specific node since agent has knowledge of all resource in the system. We dedicate three nodes to each of task type. In agent based scheduling, if same resource requirement task came then it is directly forwarded to node where same type of task is executed previously since knowledge base keeps all the information about previous tasks execution. We evaluate performance of without agent and with agent based system. Tasks are schedule using Max- Min, Min-Min and FCFS heuristics.

4.1 Makespan Results

Makespan results are shown in Fig. 4, 5 and 6. These results clearly indicate that with agent based scheduling gives less makespan as compared to without agent based scheduling of tasks. The performance of non-agent based scheduling is turned out to be poor as compared to agent based task scheduling strategy.

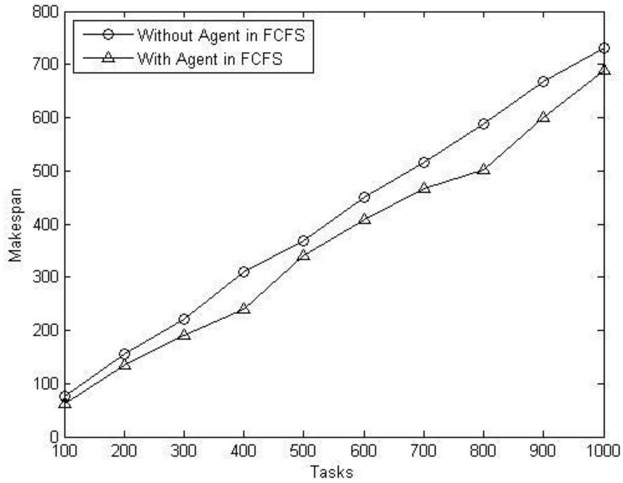

Fig. 4 Makespan Comparison using FCFS

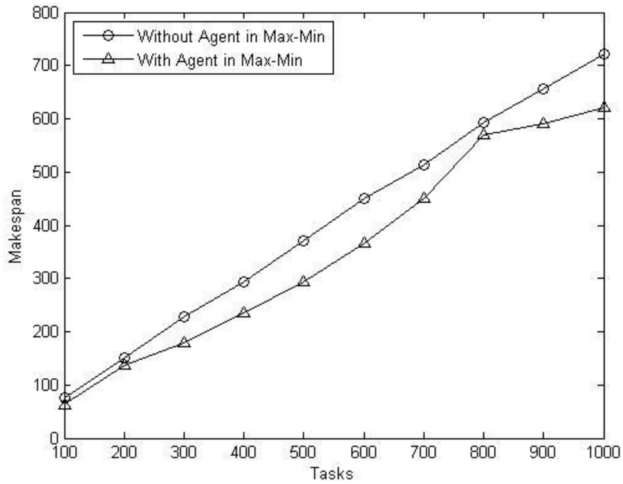

Fig. 5 Makespan Comparison using Max-Min

5. CONCLUSION

In this chapter, an agent based strategy for task scheduling in grid is presented. The proposed model consists of two different types of agent: super-agent and task specific agent. Super-agent determines task type and forwards it to task specific agent while task specific agent coordinate with specific resources and return results to super-agent. A performance evaluation of with agent based scheduling and non-agent based task scheduling is conducted. The experiment shows that our proposed framework shows optimal makespan.

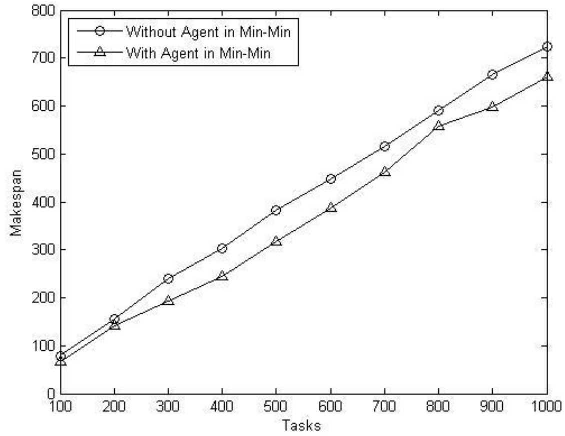

Fig. 6 Makespan Comparison using Min-Min

6. REFERENCES

[1] Corera J. M. Laresgoiti Jennings, N. R. Developing Industrial Multi-Agent Systems. In In: Proceedings of the First International Conference on Multiagent Systems, (ICMAS-95), pp. 423-430, 1995.

[2] Maes P. Chavez, A. Kasbah: An Agent Marketplace for Buying and Selling Goods. In Proceedings of First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Systems, London, UK, 1996.

[3] Faratin P. Johnson M. J. Norman T. J. O'Brien P. Wiegand M. E. Jennings, N. R. Agent-based business process management. International Journal of Cooperative Information Systems, 5:pp. 105 - 130, 1996.

[4] Hewett M. Waashington R. Hewett R. Seiver A Hayes-Roth, B. Distributing intelligence within an individual, volume 2. Morgan Kaufmann, second edition, 1995.

[5] Jennings N. R. Fox J. Huang, J. An agent-based approach to health care management. International Journal of Applied Artificial Intelligence, 9, pp. 401-420, 1995.

[6] Wang F. Areibi S. Homayounfar, H. Advanced P2P Architecture Using Autonomous Agents. International Journal of Computers, Systems and Signals, 4:pp. 115 - 118, 2002.

[7] G.R. Nudd J. Cao, D.J. Kerbyson. Dynamic Application Integration Using Agent-based Operational Administration. In: Proceedings of the PAAM00, pp. 393 - 396, 2000.

[8] G.R. Nudd J. Cao, D.J. Kerbyson. High Performance Service Discovery in Large-Scale Multi-Agent and Mobile-Agent Systems. International Journal of Software Engineering and Knowledge Engineering, 11(5):pp.621 - 641, 2001.

[9] Jun Wu, Xin Xu, Pengcheng Zhang, and Chunming Liu. A Novel Multi-Agent Reinforcement Learning Approach for Job Scheduling in Grid Computing. Future Generation Computer Systems, 27(5):pp. 430 - 439, 2011.

[10] Junwei Cao, Daniel P. Spooner, Stephen A. Jarvis, and Graham R. Nudd. Grid Load Balancing using Intelligent Agents. Future Generation Computer Systems, 21(1):pp. 135 - 149, 2005.

[11] Li Chunlin and Li Layuan. The Use of Economic Agents Under Price Driven Mechanism in Grid Resource Management. Journal of Systems Architecture, 50(9):521 - 535, 2004.