In Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002), Berlin, Germany, May 2002, pp.350-351. (workshop short paper)

Agent-based Resource Management for Grid Computing

Junwei Cao, Daniel P. Spooner, James D. Turner, Stephen A. Jarvis,
Darren J. Kerbyson*, Subhash Saini**, and Graham R. Nudd
High Performance Systems Laboratory, University of Warwick, Coventry, UK
**Modelling, Algorithms, and Informatics Group, Los Alamos National Laboratory, USA*
***NASA Ames Research Center, Moffett Field, California, USA*
Corresponding email: {junwei, dps}@dcs.warwick.ac.uk

Abstract

It is envisaged that the grid infrastructure will be a large-scale distributed software system that will provide high-end computational and storage capabilities to differentiated users. A number of distributed computing technologies are being applied to grid development work, including CORBA and Jini. In this work, we introduce an A4 (Agile Architecture and Autonomous Agents) methodology, which can be used for resource management for grid computing. An initial system implementation utilises the performance prediction techniques of the PACE toolkit to provide quantitative data regarding the performance of complex applications running on local grid resources. At the meta-level, a hierarchy of identical agents is used to provide an abstraction of the system architecture. Each agent is able to cooperate with other agents to provide service advertisement and discovery to schedule applications that need to utilise grid resources. A performance monitor and advisor (PMA) is in development to optimise the performance of agent behaviours.

1. Introduction

Software agents are accepted as a powerful high-level abstraction for modelling complex software systems. The A4 (Agile Architecture and Autonomous Agents) methodology described in this work can be used to build large-scale distributed software systems that exhibit highly dynamic behaviour. It is intended that an entire system be built of a hierarchy of identical agents with the same functionality. Agents are considered both service providers and service requestors. The implementation of system functions is abstracted to the processes of service advertisement and service discovery [3].

This methodology is integral to the system implementation of grid resource management. The system couples the performance prediction techniques of the PACE toolkit [1,5] with a scheduling system that can manage local grid resources. At the meta-level, agents cooperate to perform service advertisement and discovery, which enables the scheduling of applications over grid resources. A performance monitor and advisor (PMA) is being developed to optimise the performance of agent behaviours.

2. A4 Methodology

The A4 methodology considers the following issues:

- While many agents can be utilised as a large-scale multi-agent system, the agents are not predetermined to work together. They possess their own motivation, resource and environment.
- Autonomy is used to describe the character of an agent, with regard to its intelligence and social ability. An agent can fulfil high-level tasks directly or through cooperation with other agents.
- Architecture is used to provide a framework for interaction between agents. For example, multiple agents can be organized into a hierarchy.
- Agility is used to describe the character of the architecture, and implies quick adaptation to environmental change. While autonomy provides the system with component-level adaptability, agility provides the architecture-level adaptability of the system.

A4's emphasis is on dealing with architectural level dynamics and using simulation based analysis to provide quantitative performance evaluation and optimisation of system behaviours. The impact of the agent mobility and the choice of the service discovery strategies on the performance of the agent behaviours are studied using an A4 simulator. Experimental results for this system can be found in [2].

3. Agent-based Resource Management

Figure 1 provides an illustration of the system implementation of grid resource management, highlighting the relationship to the systems mentioned above. The agent system bridges the gap between grid users and resources in order to efficiently schedule applications that require grid resources.

PACE is used to provide prediction data concerning the performance of applications running on local high performance resources, with relevant application tools being provided to grid users. A request to execute an application is supplied with a corresponding application model developed using the tools. At the same time, PACE resource tools are embedded in each grid resource to provide a corresponding resource model, which is an important aspect of the service information of the resource. The evaluation engine is used by the multiprocessor scheduling system, which acts as a local grid resource manager.

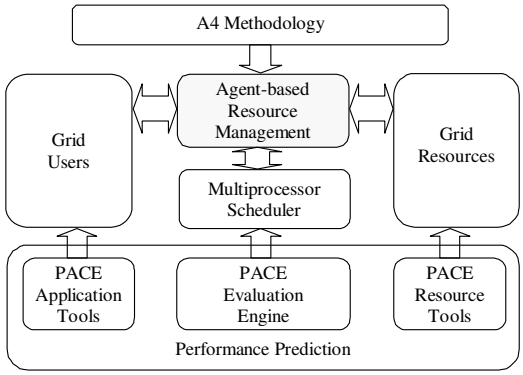

Figure 1. Agent-based Resource Management

At the meta-level, the A4 methodology is employed in the system implementation. Each agent acts as a representative for a particular scheduler at the meta-level, and considers the local grid resource as its capability to provide service. The agent structure includes a communication layer, a coordination layer and a local management layer. Agents cooperate with each other and perform service advertisement, discovery, and delivery functions to schedule applications that need to utilise available resources.

As shown in Figure 2, a performance monitor and advisor (PMA) is capable of modelling and simulating the performance of the agent system while the system is active. The PMA observes the communication traffic of the agent system and draws corresponding conclusions regarding the agents' behaviour with the intention of improving the agent performance. While the kernel of the A4 simulator is used in the PMA, including the model composer and simulation engine, the PMA has a different input and output mechanism. Statistical data is monitored from each of the agents and passed to the PMA for performance modelling. The simulation can be performed many times until a near-optimal solution is selected. The selected optimisation strategies are returned and used to reconfigure the agents accordingly.

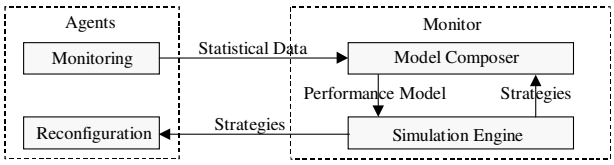

Figure 2. Performance Monitor and Advisor

4. Conclusions

An overview of an agent-based resource management for grid computing is presented in this work. A system implementation is described with performance prediction and scheduling techniques. Current work focuses on the system enhancement [4] and a transaction-based application performance modelling technology [6].

Acknowledgements

The work is sponsored in part by a grant from the NASA AMES Research Centre, and administered by USARDSG, contract no. N68171-01-C-9012.

References

[1] J. Cao, D.J. Kerbyson, E. Papaefstathiou, and G.R. Nudd, "Performance Modelling of Parallel and Distributed Computing Using PACE", in Proc. of 19th IEEE Int. Performance, Computing and Communication Conf., 485-492, 2000.

[2] J. Cao, D.J. Kerbyson, and G.R. Nudd, "Performance Evaluation of an Agent-Based Resource Management Infrastructure for Grid Computing", in Proc. of 1st IEEE Int. Symp. on Cluster Computing and the Grid, 311-318, 2001.

[3] J. Cao, D.J. Kerbyson, and G.R. Nudd, "High Performance Service Discovery in Large-scale Multi-agent and Mobile-agent Systems", Int. J. Software Engineering and Knowledge Engineering, 11(5), 621-641, 2001.

[4] J. Cao, S.A. Jarvis, S. Saini, D.J. Kerbyson, and G.R. Nudd, "ARMS: an Agent-based Resource Management System for Grid Computing", to appear in Scientific Programming, 2002.

[5] G.R. Nudd, D.J. Kerbyson, E. Papaefstathiou, S.C. Perry, J.S. Harper, and D.V. Wilcox, "PACE – A Toolset for the Performance Prediction of Parallel and Distributed Systems", Int. J. High Performance Computing Applications, 14(3), 228-251, 2000.

[6] J.D. Turner, D.P. Spooner, J. Cao, S.A. Jarvis, D.N. Dillenberger, and G.R. Nudd, "A Transaction Definition Language for Java Application Response Measurement", J. Computer Resource Management, 105, 55-65, 2002.