Scalability and Information Exchange Among Autonomous Resource Management Agents

Siri Fagernes, Alva L. Couch

Scalability and Information Exchange among Autonomous Resource Management Agents

Siri Fagernes[1] and Alva L. Couch[2]

[1]Westerdals Oslo ACT, Dep. of Technology
Oslo, Norway
siri.fagernes@westerdals.no

[2]Tufts University
Medford, MA, USA
couch@cs.tufts.edu

Abstract. We study a scenario of autonomous resource management agents, aiming for fulfilling a management goal of balancing value of service with cost. We aim for a model of management based on fully distributed knowledge, avoiding traditional challenges associated with centralized approaches. Our results indicate that lack of information about the actions of other agents can be mitigated via direct observation of each agent's environment.

Keywords: resource management, autonomous agents, cloud management, reactive approaches, decentralized knowledge

1 Introduction

We present a theoretical model of distributed resource management, which is analysed through simulations. The model involves autonomous agents that must collaborate, either directly or indirectly, to achieve a common management goal (balance cost and value). Our previous work has focused on studying the coordination of only two agents, whose primary task is to control resource usage in the system they operate, and try to estimate, based on varying information access, how to adjust their current resource level optimally.

In this paper, we study the coordination of a larger group of autonomous resource management agents, in the setting where they share a common resource pool. The main research objective is to determine whether the agents can achieve their common goal in an optimal manner without exchanging local information with each other. We see that individual observations of behaviour observed by each agent can replace information exchanged directly among the agents, which increases scalability.

2 Related Work

An automatic resource management process can be either *reactive* [1] or *predictive*, determined by how resources are automatically adjusted. The predictive

approaches are typically based on having access to a complete model of the system, which (in theory) gives the ability to provide QoS guarantees and a more detailed view of the dynamics of the system. The major challenge of such approaches is getting access to such knowledge, if it is even possible. Examples of model-based approaches are [2], [3], [4], [5], [6] and [7].

Reactive approaches are designed to make appropriate decisions when one lacks complete knowledge of the system model, and are used in complex systems for decision making. A common challenge in reactive approaches is that the learning algorithm responsible for making decisions requires a *training period* for gathering data to make appropriate action decisions. Examples of reactive approaches are presented in [1], [8], [9], [10], [11], and [12].

Most of the existing approaches are based on *centralized knowledge*. This means they have the advantage of one component having complete system knowledge, which avoids the complexity of coordination and communication overhead in distributed approaches. However, centralized approaches in larger complex systems – cloud systems – have several drawbacks, including limited scalability, single point of failure issues, and potential bottlenecks.

3 Method and Approach

Our management scenario consists of ten autonomous resource management agents $Q_i, i \in [1, 10]$, where Q_i controls a separate resource variable R_i. Each resource variable (or component in the system) contributes to delivering a service S. The main objective of management is to achieve efficient management of all the different system resource variables, with the objective to achieve a balance between cost and produced value. To determine value of the delivered service, the performance metric P represents job throughput, i.e., the reciprocal of response time. The response time will depend on how the system is able to cope with current load, which is defined as an arrival rate of requests. The system performance is hence defined as the request completion rate, and is modeled approximately as

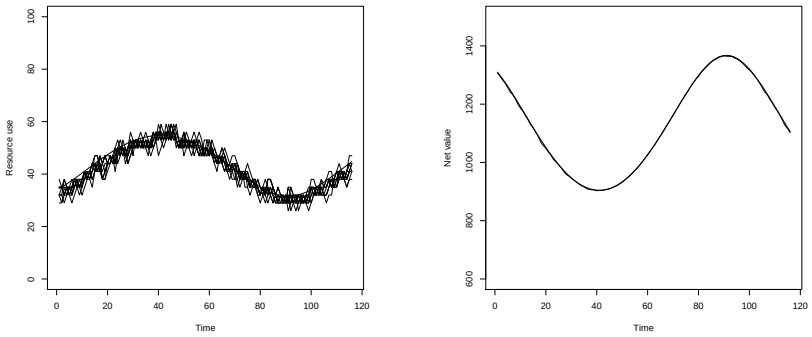
$$P = B - \frac{\gamma L}{R} \tag{1}$$

where B is the baseline performance (the performance when the system is not affected by load). γ is a constant representing resource-intensitivity, i.e. increased γ represents a service in which the service requests are more resource-intensive. Further, we define associated value of service to be proportional to throughput, so that

$$V = \alpha P = \alpha \sum_{i=1}^{10} P_i. \tag{2}$$

Similarly, cost C is proportional to resource use R, so that

$$C = \beta R. \tag{3}$$



(a) Resource use (b) Net value

Fig. 1: 10 operators, each controller has information about current load. $\gamma = 1$ for all agents.

The autonomous agents (resource controllers), which are responsible for making decisions on resource use and adjustments, do *not* have access to knowledge of this theoretical model of the system's performance. Each agent observes how system value V changes with changes in R_i, $\Delta V/\Delta R_i$, and based on the local knowledge of associated cost $C(R_i)$, the closure operator can make an estimate of how net value $N = V - C$ changes with R_i, by calculating $\Delta N/\Delta R_i$. If this value is positive, the controller will increase R_i, and if it is negative, decrease R_i. This *hill-climbing* strategy will converge to a global optimum whenever the objective function N is convex.

The agents have a perception of how system value depends on resource use. The theoretically correct global value is defined as

$$V = \alpha(B - \frac{\gamma L}{R_1} - \frac{\gamma L}{R_2} - \ldots - \frac{\gamma L}{R_{10}}) \tag{4}$$

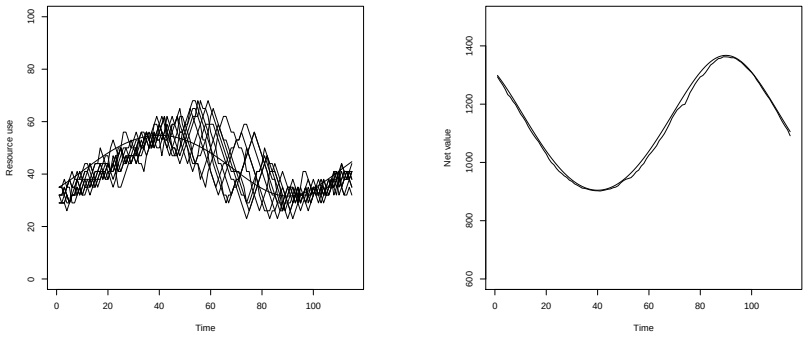
In our experiments, the agents assume that the value-resource relationship is modelled as

$$V = a\frac{L}{R} + b. \tag{5}$$

4 Results

When each operator receives individual value feedback, no external information about other operators is needed. When the operator has a semi-accurate model of the system dynamics and current system load, all operators perform very close to optimal, as seen in Figure 1.

Providing less information (no load information) reduces the precision of the results (Figure 2a), but the performance (achieved net value) is quite close to the theoretical optimum (Figure 2b).



(a) Resource use (b) Net value

Fig. 2: 10 operators, each controller lacks information about system load. $\gamma = 1$ for all agents.

5 Conclusion and Further Work

Although there has been significant research efforts aimed at achieving fully decentralized management of larger complex systems, most proposed solutions so far has been based on either pure centralization or partly centralization based on delegation of responsibility. Our work has been an attempt to achieve pure decentralized management. The goal of our approach is trying to come up with an intermediate approach between delegated management and agent based management, in which there is higher predictability and more accurate goal achievement.

This study indicates that to achieve self-optimising behaviour among autonomous agents working towards the same goal, without direct coordination, excessive information exchange or centralized knowledge, is the ability to monitor their invidual behaviour. This means that developing efficient feedback mechanisms is a crucial factor to reduce the need for global information exchange.

One particular issue that we have not studied, is how the precision of our proposed model is affected by more heavily varying system load. Also, to test the robustness of the model, this needs to be implemented in a real scenario.

References

1. Harold C Lim, Shivnath Babu, Jeffrey S Chase, and Sujay S Parekh. Automated control in cloud computing: challenges and opportunities. In *Proceedings of the 1st workshop on Automated control for datacenters and clouds*, pages 13–18. ACM, 2009.
2. Wesam Dawoud, Ibrahim Takouna, and Christoph Meinel. Elastic vm for cloud resources provisioning optimization. In *Advances in Computing and Communications*, pages 431–445. Springer, 2011.

3. Nilabja Roy, Abhishek Dubey, and Aniruddha Gokhale. Efficient autoscaling in the cloud using predictive models for workload forecasting. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 500–507. IEEE, 2011.
4. Zhenhuan Gong, Xiaohui Gu, and John Wilkes. Press: Predictive elastic resource scaling for cloud systems. In *Network and Service Management (CNSM), 2010 International Conference on*, pages 9–16. IEEE, 2010.
5. Nedeljko Vasić, Dejan Novaković, Svetozar Miučin, Dejan Kostić, and Ricardo Bianchini. Dejavu: accelerating resource allocation in virtualized environments. In *ACM SIGARCH Computer Architecture News*, volume 40, pages 423–436. ACM, 2012.
6. Zhiming Shen, Sethuraman Subbiah, Xiaohui Gu, and John Wilkes. Cloudscale: elastic resource scaling for multi-tenant cloud systems. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, page 5. ACM, 2011.
7. Upendra Sharma, Prashant Shenoy, Sambit Sahu, and Anees Shaikh. A cost-aware elasticity provisioning system for the cloud. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 559–570. IEEE, 2011.
8. Pradeep Padala, Kang G Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, and Kenneth Salem. Adaptive control of virtualized resources in utility computing environments. *ACM SIGOPS Operating Systems Review*, 41(3):289–302, 2007.
9. Azbayar Demberel, Jeff Chase, and Shivnath Babu. Reflective control for an elastic cloud application: an automated experiment workbench. In *Proceedings of the 2009 conference on Hot topics in cloud computing (HotCloud'09)*, 2009.
10. Shicong Meng, Ling Liu, and Vijayaraghavan Soundararajan. Tide: achieving self-scaling in virtualized datacenter management middleware. In *Proceedings of the 11th International Middleware Conference Industrial track*, pages 17–22. ACM, 2010.
11. Rodrigo N Calheiros, Christian Vecchiola, Dileban Karunamoorthy, and Rajkumar Buyya. The aneka platform and qos-driven resource provisioning for elastic applications on hybrid clouds. *Future Generation Computer Systems*, 28(6):861–870, 2012.
12. Jose F Martinez and Engin Ipek. Dynamic multicore resource management: A machine learning approach. *Micro, IEEE*, 29(5):8–17, 2009.