Agent cloning *

Onn Shehory, Katia Sycara, Prasad Chalasani, and Somesh Jha
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
onn,katia,chal,sjha@cs.cmu.edu

Abstract

Multi-agent systems are subject to performance bottlenecks in cases where agents cannot perform tasks by themselves due to insufficient resources. Solutions to such problems include passing tasks to others or agent migration to remote hosts. We propose agent cloning as a more comprehensive approach to the problem of local agent overloads. According to our paradigm, agents may clone, pass tasks to others, die or merge.

1 Introduction

Assume a multi-agent system (MAS) that receives a stream of tasks. The agents have *capabilities* which indicate the types of tasks they can perform and *capacities* which indicate the amounts of resources that the agents can access and use for task execution. Tasks are categorized by types that can be handled by agents with appropriate capabilities. We discuss the case where the task flow to an agent overloads it. This is categorized below:
1. An agent in a MAS is overloaded, but the MAS as a whole has the required capabilities and capacities.
2. The MAS as a whole is overloaded, i.e., the agents that comprise the MAS do not have the necessary capacities (however there may be idle resources in the computational system where the agents are situated).

As a result of such overloads, the MAS will not perform all of the tasks in time, although the required resources may be available to it. Some solutions suggest themselves:
1. First case – overloaded agents should pass tasks to others which have the capabilities and capacities to perform them.
2. Second case – overloaded agents create new agents to perform excess tasks and utilize unused resources or migrate to other hosts.

We present agent cloning as a means for implementing these solutions. We consider cloning in an open environment where agents might dynamically appear or disappear and use the RETSINA domain-independent infrastructure [5] to which we apply the approach.

2 The cloning approach

Cloning is a possible response of an agent to overloads. Agent overloads are due, in general, either to the agent's limited capacity to process current tasks or to machine overloads. Other approaches to overloads include task transfer and agent migration. Task transfer, where overloaded agents locate other agents which are lightly loaded and transfer tasks to them, is similar to processor load balancing. Agent migration, which requires that overloaded agents, or agents that run on overloaded machines, migrate to less loaded ones, is similar to process migration and mobile agents [1]. Agent migration can be implemented by creating its clone on a remote machine, transferring its tasks to it and dying. Thus, agent mobility is an instance of agent cloning.

To perform cloning, an agent must reason about its own load (current and future), its host load as well as capabilities and loads of other machines and agents. Accordingly, it may decide to: create a clone; pass tasks to a clone; merge with other agents; or die. Merging of two agents, or self-extinction of underutilized agents is an important mechanism to control agent proliferation.

Since the agent's own load and the loads of other agents vary over time in a non-deterministic way, the decision of *whether and when* to clone is non-trivial. Prior work [4] has presented a model of cloning based on prediction of missed task deadlines and idle times on the agent's schedule in the RETSINA multi-agent infrastructure [5]. This research implements a stochastic model of decision making based on dynamic programming to determine the optimal timing for cloning.

*This material is based upon work supported in part by ARPA Grant #F33615-93-1-1330, by ONR Grant #N00014-96-1222, and by NSF Grant #IRI-9612131.

3 The Cloning Procedure

The cloning procedure consists of the following:

- Reasoning before cloning: includes the reasoning about the (possibly dynamic) task list with respect to time restrictions, capability and resource requirements. It results in a decision to clone or transfer tasks.

- Task splitting: includes reasoning that considers the time intervals in which overloads are expected and accordingly selects tasks to be transfered.

- Cloning: includes the creation and activation of the clone, the transfer of tasks, and the updates of connections between agents via matchmaking. The basic actions are:
  (i) Create a copy of its code. This copy, however, may have to undergo some modification.
  (ii) When cloning while performing a specific task, an agent should pass to its clone only the relevant sub-tasks and information. Otherwise, the clone may face the same overload problem as its creator. In contrast to the typical approach to agent migration [2], cloning does not require the transfer of an agent state. The only transfer necessary is of the set of tasks to be performed by the clone.

- Reasoning after cloning: collects information regarding the benefits of the cloning and environmental properties (such as task stream distribution), and statistically analyzes them, to learn for future cloning.

While the reasoning of whether to initiate cloning is performed continually (i.e., when there are changes in the task schedule or if previous attempts to clone have failed), the cloning itself is a one-shot procedure.

4 Simulation

Simulation results show that cloning increases the performance of the MAS. To simulate the RETSINA agents [3], we have measured their resource consumption. We simulated the agent system with and without cloning, with the following settings: 10 to 20 agents; 10 clones allowed; up to 1000 tasks dynamically arriving at the system; normal distribution of tasks with respect to the required capabilities and resources for execution (10% of the tasks beyond the ability of the agents to perform them within their particular deadlines); an agent can perform 20 average tasks simultaneously. The results of the simulation are depicted in figure 1. The graph shows that for small numbers of tasks (0 to 100) a system which practices cloning performs

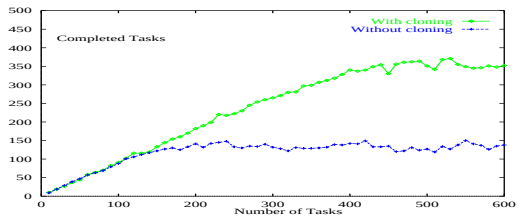

Figure 1. Task execution w/wo cloning

(almost) as well as a system with no cloning When the number of tasks increases, the cloning system performs much better. Beyond some threshold, (around 350 tasks) even cloning cannot help. Note that in the range 150 to 350 tasks cloning results in task performance which is close to the optimal (85% as compared to 90% which is optimal), where optimality refers to the case in which all of the available resources are efficiently used for task performance.

5 Conclusion

Agent cloning is the action of creating and activating a clone agent (locally or remotely) to perform some or all of an agent's tasks. Cloning is performed when an agent perceives or predicts an overload, thus increasing the ability of a MAS to perform tasks. We have presented agent cloning as a means for balancing the loads and improving the task performance of a MAS running on several remote machines. We found that for large numbers of tasks, cloning significantly increases the portion of tasks performed by a MAS. Currently we are in the process of embedding the cloning protocol into each autonomous agent in the RETSINA MAS. In future work we intend to use cloning for agent mobility. In addition, we are developing protocols for agent merging or self-extinction.

References

[1] D. Chess, B. Grosof, and C. Harrison. Itinerant agents for mobile computing. TR RC 20010, IBM Research, 1995.

[2] R. Clark, C. Grossner, and T. Radhakrishnan. CONSENSUS and COMPROMISE: Planning in cooperating expert systems. *Int. Journal of Intelligent and Cooperative Information Systems (submitted)*.

[3] K. Decker, A. Pannu, K. Sycara, and M. Williamson. Designing behaviors for information agents. In Proc. of AA-97.

[4] K. Decker and K. Sycara. Intelligent adaptive information agents. *Journal of Intelligent Information Systems*, 9:239–260, 1997.

[5] K. Sycara, K. Decker, A. Pannu, M. Williamson, and D. Zeng. Distributed intelligent agents. *IEEE Expert*, pages 36–45, December 1996.