SymbioticSphere: Towards an Autonomic Grid Network System

Paskorn Champrasert, Chonho Lee and Junichi Suzuki
Department of Computer Science
University of Massachusetts, Boston
{paskorn, chonho, jxs}@cs.umb.edu

Abstract

This paper describes SymbioticSphere, a novel biologically-inspired architecture that allows grid systems (application services and middleware platforms) to be scalable and adaptive to dynamic network environments. In SymbioticSphere, each service and platform is designed as a biological entity, analogous to an individual bee in a bee colony. Services and platforms implement biological concepts and mechanisms such as decentralization, emergence, energy exchange, migration, replication and death. Like in biological systems, desirable systems characteristics such as scalability and adaptability emerge from collective actions and interactions of services and platforms.

1. Introduction

Grid systems are expected to autonomously scale to enormous demand placed upon them and adapt to dynamic network environments in order to improve user experience, expand system's operational longevity and reduce maintenance cost. Based on the observation that various biological systems have already achieved these requirements (i.e. autonomy, scalability and adaptability), the proposed network architecture, called SymbioticSphere, applies biological concepts and mechanisms to design grid systems (application services and middleware platforms). In SymbioticSphere, each application service and platform is designed as a biological entity, analogous to an individual bee in a bee colony. An application service is implemented as an autonomous and distributed software agent. Each agent implements a functional service and follows biological behaviors such as migration and energy exchange. A platform is a middleware that runs on a network host and operates agents. Each platform implements runtime services that agents use to perform their services, and follows biological behaviors such as replication, death and energy exchange.

Similar to biological entities, agents and platforms in SymbioticSphere store and expend *energy* for living. Each agent gains energy in exchange for performing its service to other agents or human users, and expends energy to use network and computing resources. Each platform gains energy in exchange for providing resources to agents, and continuously evaporates energy. SymbioticSphere models agents and platforms as different species, and follows several concepts in ecological food chain to determine how much energy agents/platforms expend and how often they expend energy. The abundance and scarcity of stored energy affects behaviors of an agent/platform. For example, an abundance of stored energy indicates higher demand for the agent/platform; thus the agent/platform may replicate itself in response to higher energy level. A scarcity of stored energy (an indication of lack of demand) may cause death of the agent/platform.

Similar to biological systems, SymbioticSphere exhibits emergence of desirable system characteristics such as scalability and adaptability. These characteristics emerge from collective interactions and behaviors of agents and platforms, rather than they are not present in any single agent/platform.

2. SymbioticSphere

In SymbioticSphere, agents run on platforms, which in turn run on network hosts. Agents and platforms are designed based on the following three principles.

(1) Decentralization: Agents and platforms are decentralized. There are no central entities to control and coordinate agents/platforms (i.e. no directory servers and no resource managers). Decentralization allows agents/platforms to be scalable and simple by avoiding a single point of performance bottleneck and any central coordination in deploying agents/platforms.

(2) Autonomy: Agents and platforms are autonomous. They monitor their local network environments, and based on the monitored conditions, they autonomously behave and interact without any intervention from/to other agents, platforms and human users.

(3) Adaptability: Agents and platforms are adaptive to changing environment conditions (e.g. user demands, user locations and resource availability). Adaptation is achieved by designing agent/platform behavior policies to consider local environment conditions. For example, agents may implement a migration behavior of moving towards platforms that forward a large number of user requests for their services. This results in the adaptation of agent locations, and agents concentrate around the users who request their services. Also, platforms may invoke replication and death behaviors when their energy levels become over and below thresholds. This results in the adaptation of resource availability on platforms, and platforms adjust their population against the demands for resources.

Each agent consists of *attributes*, *body and behaviors*. *Attributes* carry descriptive information regarding the agent, such as agent ID, energy level and description of a service it provides. *Body* implements a service the agent provides. For instance, an agent may implement a genetic algorithm for an optimization problem, while another one may implement a physical model for scientific simulations. *Behaviors* implement non-service related actions that are inherent to all agents. Currently, agents support migration, replication, death, energy exchange and environment sensing.

Each platform runs on a network host and operates agents. It abstracts low-level operating and networking details, and aids developing and deploying agents. A platform consists of *attributes*, *behaviors* and *runtime services*. *Attributes* carry descriptive information regarding the platform, such as platform ID, energy level and healthy level. *Healthy level* is defined as a function of the age of and resource availability on a network host that the platform resides on. Healthy level affects behaviors of a platform/agent. For example, higher healthy level indicates higher stability of a network host that the platform resides on; thus the platform may replicate itself on a healthier neighboring host. This results in the adaptation of platform locations. Platforms seek and work on stable hosts. *Behaviors* are the actions inherent to all platforms. Currently, platforms support replication, death, energy exchange and environment sensing.

Agents and platforms have policies for each of their behaviors. A behavior policy defines when to and how to invoke a particular behavior. Each behavior policy consists of one or more *factors*, which evaluate environment conditions or agent/platform status (e.g. energy level and healthy level). Each factor is given a certain *weight* relative to its importance. Each behavior is invoked if the weighted sum of its factor values exceeds a threshold. For example, the factors in agent migration behavior include:

- *Energy Level*: encourages agents whose energy level is higher to migrate.
- *Service Request Ratio*: (# of service requests on a remote platform)/(# of service requests on a local platform), which encourages agents to move towards users.
- *Healthy Level Ratio*: (healthy level of a remote host)/(healthy level on a local host), which encourages agents to move towards healthier hosts.
- *Migration interval*: the interval from the time of a previous migration, which discourages agents to migrate too often.

The factors in platform replication behavior include:

- *Energy Level*: encourages platforms whose energy level is higher to migrate.
- *Healthy Level Ratio*: (healthy level of a remote host)/(healthy level on a local host), which encourages platforms to replicate themselves on healthier hosts.

3. Simulation Results

A series of simulations were carried out to evaluate how the biologically-inspired mechanisms in SymbioticSphere impact on scalability and adaptability of grid systems[1]. 64 network hosts are deployed in a 8x8 grid topology. At the beginning of each simulation, a single agent starts running on a platform.

Simulation results show that agents and platforms scale to the demands for services and resources by adjusting the number of them (i.e. with replication and death behaviors).

Agents and platforms also adapt to dynamic changes in the network (e.g. user location, network traffic and resource availability). For example, agents migrate towards users, contributing to reduce response time to users. In this process, the average distance between platforms and users (in a hop count) gradually decreases, although platform replication policy does not consider user location. This is an example of symbiotic emergence. If replicated platforms are placed on hosts that agents want to migrate to (i.e. hosts closer to a user), the platforms would survive. Otherwise, they would die because agents do not migrate onto them and do not transfer energy to them. Thus, in a sense, agents indirectly suggest platforms where to replicate themselves. This results in mutual benefits for both agents and platforms. Agents can work closer to users and gain more energy from the users. Platforms can gain more energy from agents.

References

[1] P. Dini, W. Gentzsch, M. Potts, A. Clemm, M. Yousif and A. Polze, "Internet, Grid, Self-adaptability and Beyond: Are We Ready?," In *Proc. of the IEEE International Workshop on Self-Adaptable and Autonomic Computing Systems*, August 2004.

[1] Simulations were carried out with the SymbioticSphere simulator, which contains 14,100 lines of Java code. It can run arbitrary number of agents, platforms, users and hosts on simulated networks. This simulator is available at dssg.cs.umb.edu/symbiosis/ for researchers who investigate autonomic grids.