BRIDGING ECOLOGY AND CLOUD: TRANSPOSING ECOLOGICAL PERSPECTIVE TO ENABLE BETTER CLOUD AUTOSCALING

6

Tao Chen, Rami Bahsoon

School of Computer Science, University of Birmingham, UK

6.1 INTRODUCTION

In cloud environment, the quality of service (QoS) and cost/energy objectives for cloud-based services can be tuned by accessing software configurations (e.g., the number of service threads, size of connection pool, and session lifetime) and hardware resources (e.g., the CPU, memory, and Virtual Machine) that are shared, leased, and priced as utilities. Such a feature is fundamentally facilitated through autoscaling: an automatic and elastic process, typically running on a Physical Machine (PM), that adapts software configurations and hardware resources provisioning on-demand according to the changing environmental conditions (e.g., the workload).

What make the cloud autoscaling challenging are the dynamic, uncertainty and possible trade-offs on objectives (i.e., QoS and cost/energy objectives) exhibited in the process. In cloud, dynamics and uncertainty arise from the unpredictable environmental conditions and QoS interference – a scenario where the competing demands of some of the services can interfere with the QoS of others, provided that there are many cloud-based services run on a shared infrastructure. Trade-offs are associated with runtime autoscaling decisions as to the appropriate amount of scaling applied to software configurations and hardware resources. These trade-offs can be, for example, whether to choose throughput over cost or which competing cloud-based services to focus on.

To address those challenges, prior efforts in cloud autoscaling have relied on rules-based control [14], control theoretic mechanisms [22], and computational intelligence techniques [5], with particular focus on improving the optimality, elasticity, and scalability of the autoscaling system. However, the runtime stability and sustainability in the cloud, which are mainly concerned with the long-term benefit for both cloud consumers and provider, have not been explicitly tackled. Here, *sustainability* refers to the ability of cloud to endure the stress caused by dynamic and uncertain events, e.g., workload and QoS interference, with an aim to continually optimize QoS attributes of all cloud-based services while minimizing their costs and energy consumptions. The longer the time when no violation of Service Level Agreement (SLA) and budget requirements occur, the better the *stability*. Following the intuition that *computer systems can be better understood, controlled, and developed when viewed from*

the perspective of living systems [12], we argue that the perspective of ecology and natural ecosystem[1] is a new, yet neat view for computer science researchers to design a novel autoscaling system in the cloud. In particular, understanding stability and sustainability of natural ecosystems, as well as how we can better manage them spontaneously, has been the core research theme for ecologists. These will therefore provide many useful insights for researches in cloud autoscaling.

In this article, we explore the potential benefit of using ecological view when designing an autoscaling system in the cloud. From the ecological point of view, we intend to render the cloud environment as a natural ecosystem and to design autoscaling system in the cloud derived/inspired from ecological techniques. We then propose a sensible translation of ecological principles, theories, and models into cloud autoscaling analogues. Particularly, we have explored the biotic characterizations of cloud-based services and the underlying cloud primitives w.r.t. the principles of living organisms/species, nonliving components, habitats, biodiversity, disturbances, species competition, trophic web, and natural evolution. We propose an ecology-inspired self-aware architectural pattern extending on the self-aware patterns from our prior work [11]. The new pattern explicitly caters for the key levels of biotic information, which are also systematically linked to the original principles of self-awareness. Finally, we highlight the challenges and opportunities for future investigations of ecology-inspired autoscaling in the cloud.

6.2 MOTIVATION

Elastic autoscaling in the cloud has been an increasingly important research topic since the emergence of cloud computing paradigm. Efforts have been spent to deal with the dynamics, uncertainty and trade-offs exhibited in the autoscaling process [14,22,5,7]. Nevertheless, how autoscaling can improve the stability and sustainability of the cloud as a whole has not been explicitly studied in prior work. Undoubtedly, stability and sustainability are among the most desirable attributes of cloud computing. Table 6.1 illustrates the benefits of explicitly considering stability and sustainability when autoscaling in the cloud.

Natural ecosystems are considered to be robust, efficient, and scalable systems that are capable to cope with dynamics and uncertainty, possessing several properties that may be useful in cloud autoscaling systems. These properties include self-awareness, self-adaptivity, and the ability to provide solutions for complex scenarios [17], e.g., resolving trade-offs. Many of these properties can be understood via well-known ecological models [16], which provide a theoretical basis for the occurrence of self-awareness and self-adaptivity, resulting from the interactions among the individuals and their environment, leading to complex and emergent behaviors [17] (e.g., evolution driven by natural selection).

Among others, stability and sustainability are the most desirable attributes in natural ecosystem and they have been studied by the ecologists for decades. We advocate that the well-established ecological principles, theories, and models can provide rich source of inspiration to spontaneously improve the stability and sustainability of the cloud as a whole. This will allow all the cloud-based services to stay robust and generate minimal overhead when optimizing their objectives and complying their SLA/bud-

---

[1]We will discuss this in details in Section 6.3.

Table 6.1 Comparing autoscaling in the cloud with/without tackling stability and sustainability

With stability and sustainability	Without stability and sustainability
Improve QoS, cost and energy from a long-term perspective	Improve QoS, cost and energy, but might only be effective in short-terms
Resilient to extreme cases, e.g., sudden and spiked workloads	Vulnerable to extreme cases, e.g., sudden and spiked workloads
Aim for less scalings and smaller overhead	Easy to result in unnecessary scalings and larger overhead

get requirements. However, this begs the question: *How can we systematically incorporate the natural ecosystem and cloud autoscaling?*

6.3 NATURAL ECOSYSTEM

From the ecological perspective, the term *ecosystem* refers to a natural community where all the living organisms (e.g., plants and animals) and nonliving components (e.g., air, light, and water) exhibit dynamic, and uncertain interactions with each others and the environment, emerging as a system [20].

The dynamics in an ecosystem is represented by trophic web – an interaction network that models the consumer–resource relationship, for examples, predator–prey or organism–resources. The trophic web often consists of different trophic levels, each of which represents a family of functionally consistent species. The consumer–resource relationship takes place between different trophic levels. The foundation of a trophic web are the autotrophic species (e.g., most plants) who can produce complex organic compounds from nonliving components. In contrast, higher levels in the trophic web are heterotrophic species (e.g., animals) who cannot fix carbon and use organic carbon for growth. There is a special kind of mixotrophic species that use a mix of different sources of energy and carbon, e.g., the venus flytrap and oriental hornet.

An ecosystem might face disturbances caused by either natural or human induced stress, e.g., tornado and deforestation are examples of natural and human-induced stress, respectively. Sustainability often refers to the endurance of an ecosystem in the presence of disturbances. Better sustainability of an ecosystem implies better stability – an ecosystem is said stable if, when disturbances occur, it is not affected or it is able to quickly resume its prior stable state after disturbances.

According to the well-recognized insurance hypothesis [21], ecologists have acknowledged that better stability and sustainability of an ecosystem can be achieved on higher biodiversity. This is because an ecosystem with large diversity of species will be able to respond to the disturbances in different ways, and thus it is more likely to resume the previous stable state as some species can compensate for those that disappear.

Biodiversity is a corollary of evolution, which describes the ability of a species to survive and reproduce. This is attributed to the fact that a species contains different individuals whose concrete characteristics vary, e.g., their genetic code and habits. Driven by the natural selection during evolution, new species might be created and the incapable ones might disappear. Evolution might lead to the change of an ecological niche, which represents how the species responds to the changes of resources and competitors. Changing the niche could also imply conversion of the habitat (e.g., land or sea) that
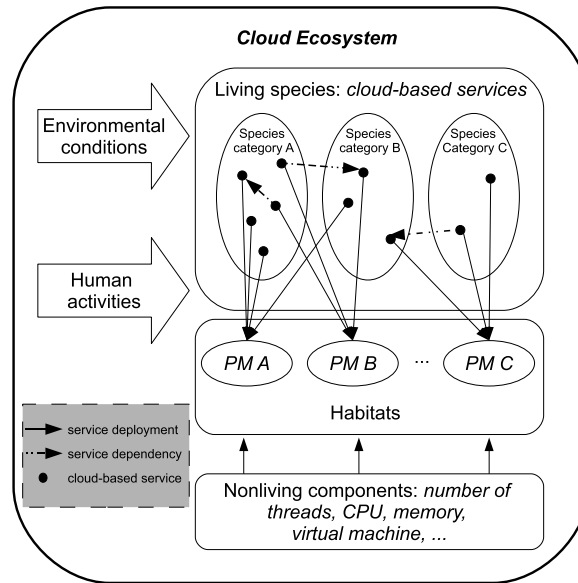
FIGURE 6.1

The architecture of cloud ecosystem

is used by the species, e.g., the ancestors of whales were living on land. According to the competitive exclusion principle [15], evolution is one of the results of competition, in which the benefit of one species can be lowered by the presence of the others – this might be due to the limited supply of certain resources. A special case, namely coevolution, occurs if one species changes in response to the changes in others.

6.4 TRANSPOSING ECOLOGICAL PRINCIPLES, THEORIES AND MODELS TO CLOUD ECOSYSTEM

Existing ecology researches have provided many insights on how we can better preserve the natural ecosystem, particularly with an aim to improve its stability and sustainability. Our hope is to learn and investigate how these insights can be used to derive better elastic autoscaling in the cloud. In fact, as we will show below, the collection of cloud-based services operates in a way that has many similarities to the natural ecosystem, and therefore emerging as a *cloud ecosystem*.

   Ecosystem. The 3-layered architecture of cloud ecosystem is illustrated in Fig. 6.1. At the top layer, the cloud-based services are regarded as living organisms, which are categorized as species within the cloud ecosystem. These species can be fundamentally classified based on the nature of the services, e.g., their functional dependency to other services and whether it is a commercial web service or a scientific service, etc. This is similar to the general classification of species in the natural

ecosystem, e.g., plants, animals and microorganisms. Such classification might be useful for us to study the characteristics of the species in a cloud ecosystem. At the intermediate layer, one or more species might coexist in a habitat, i.e., a PM that encapsulates the necessary nonliving components. Here, different PMs might be heterogeneous, leading to various forms of habitat. In such a context, the nonliving components can be the fine-grained and reusable software configurations (e.g., thread of service) and hardware resources (e.g., CPU and memory), as shown in the bottom layer. In particular, the software configurations are counterparts of infinite natural resources (e.g., light and air) while the hardware resources are the components that are subject to limited supply, e.g., soil and water in the natural ecosystem. Externally, the cloud ecosystem would be affected by disturbances, including both environmental conditions and human activities: the former refer to the factors that are controlled by neither the cloud provider nor service owners, e.g., the workload and the size of incoming jobs. The later, on the other hand, represent the activities that would influence cloud-based services, as conducted by the service owners or cloud provider.

Similar to the natural ecosystem, the cloud ecosystem reacts to the emergent disturbances and preserves stability in the same pattern: When disturbances occur (e.g., sudden changes in workload), the organisms (cloud-based services) would have to amend their demand for nonliving components (e.g., CPU, memory) or demand for the other species of organisms (when the services have functional dependency), in order to survive in the cloud ecosystem. In certain cases, organisms, or even the entire species, would need to change the habitat (VM migration), creating opportunities for multiple species in the same habitats. All these facts imply evolution that changes the biodiversity, e.g., the demand of services, their underlying VM and neighboring services are changed. In the following, we will explain the mapping between ecological principles and cloud autoscaling system in detail.

Trophic web. The notion of trophic web can also find a matching principle in a cloud ecosystem. As we can see from Fig. 6.2, the functional dependency between cloud-based services can be modeled as a predator–prey relationship, e.g., one or more business services (predator) can depend on a database service (prey). In the natural ecosystem, a larger population of species A (predator) usually means a smaller population of species B (prey), until at some point in time this A declines due to a lack of B. In contrast to the predator–prey relationship, the cloud ecosystem would not change the quantity of services, but could affect their ability in serving requests and jobs. For example, when the number of predator services increases, their prey services can gradually reach their limit in handling requests, which would be equivalent to "dying out". Consequently, the predator services would decline due to the "dying out" of their prey, unless they can evolve themselves to seek alternatives. On the other hand, the correlation between cloud-based services and the software/hardware resources is clearly an organism–resources relationship. It is obvious to see that the species in a cloud ecosystem can only be either autotrophic or mixotrophic because, on the one hand, they directly consume the nonliving components and, on the other hand, they may rely on the organic carbon from the other species.

Disturbances. Similar to natural ecosystems, the cloud ecosystem exhibits various forms of disturbances caused by the environmental conditions. These disturbances can be naturally induced stress, e.g., the changes in colocated services and cohosted Virtual Machines (VMs), changes of software configurations and hardware resources provisioning, and VMs migration/replication. Humans, i.e., service owners and cloud providers, might also create stress by changing the SLA and budgets requirements, deploying/removing cloud-based services or amending the prices for renting software and hardware resources.
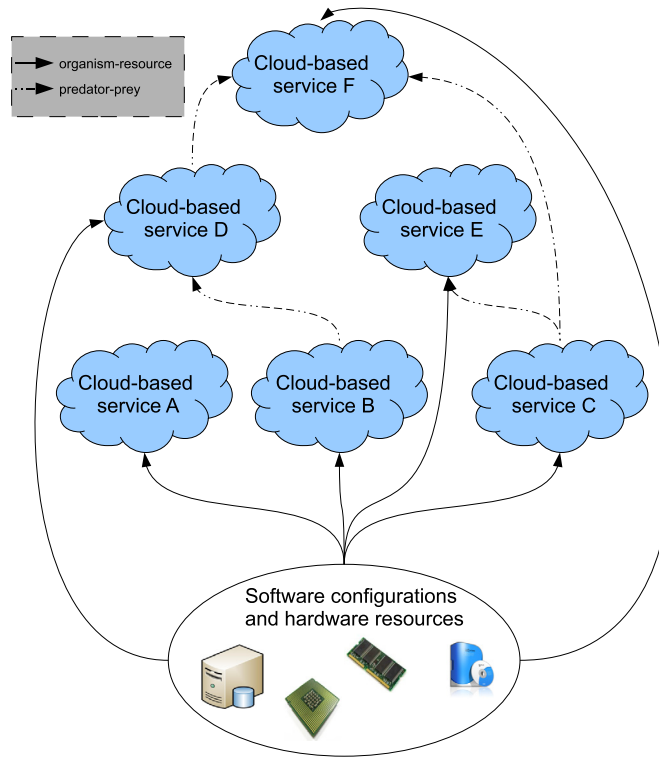
FIGURE 6.2

The trophic web of cloud ecosystem

Biodiversity and evolution. In the cloud ecosystem, the species, or cloud-based services, differ depending on their QoS sensitivity to different software configurations and hardware resources, as well as their SLA and budget requirements. In addition, there can be a large diversity of the cloud-based services running on the same PM, or habitat. These are clear evidences of biodiversity in the cloud ecosystem. Through elastic autoscaling, evolution in the cloud ecosystem refers to changing the ability of cloud-based services in accessing the software configurations and hardware resources, and possibly the deployment of services. This fact allows such evolution to directly influence the biodiversity in the cloud ecosystem. Precisely, evolution can be regarded on two levels: (i) at the microevolution level, where cloud-based services can continually evolve by changing their software configurations and hardware resource provisioning based on their demand; and (ii) at the macroevolution level, where adding/removing cloud-based services and VM migration/replication can cause changes to their habitats.

Species competition. QoS interference is a good example of species competition in the cloud ecosystem, as the cloud-based services running on the same habitat are competing for resources when the supply is limited. Consequently, they need to either evolve themselves (through autoscaling) or

die out (crash or be removed by the owners due to severe SLA violations). This phenomena in cloud also play a role of a counterpart to the coevolution principle in a natural ecosystem: to mitigate QoS interference, two or more cloud-based services might need to evolve in response to each other.

6.5 **ECOLOGY-INSPIRED SELF-AWARE PATTERN**

We now describe our preliminary research outcome that extends the self-aware patterns to incorporate ecological principles and biotic information for cloud autoscaling. As systematically documented in our handbook [11,10], self-aware patterns describe sets of capabilities (i.e., levels of awareness) to acquire knowledge for a *node* which, in the context of self-aware computing systems, can refer to a process, machine, or any conceptual part of a software system being managed. Similar to the original self-aware patterns, the ability to acquire knowledge also plays a crucial role in ecology-inspired patterns, but it particularly focuses on acquiring different levels of biotic information and knowledge. Specifically, when describing ecology-inspired pattern in cloud autoscaling, a *node* refers to an autoscaling process, which maintains biotic information and manages a group of species (i.e., cloud-based services) separated by their categories and/or habitats (i.e., PMs). As mentioned in previous sections, different groups of species and the nonliving components form the cloud ecosystem.

A possible ecology-inspired pattern and the corresponding self-aware capabilities are shown in Fig. 6.3. The key capabilities to acquire biotic information in the pattern can be discussed as follows:

- **Disturbance-Awareness.** This is the basic level of awareness in an ecology-inspired pattern. It may acquire knowledge about either natural or human induced stress. It corresponds to the stimulus-awareness in self-aware patterns.
  - *Example.* The cloud autoscaling process is able to sense different forms of disturbances, e.g., changes in workload, the neighboring services/VMs, changes in pricing/requirements and availability of software/hardware resources. For instance, with disturbance-awareness, the cloud ecosystem is able to identify the source of stimulus and react upon.
- *Trophic-web-Awareness.* Interactions within ecosystem are expressed in a trophic web and therefore awareness of such a web is essential to capture the occurrences of possible interactions. In particular, this type of awareness could acquire knowledge about either internal or external interactions. For example, the relationships between species in the corresponding group are clearly internal interactions; while those between different groups or even different ecosystems can be seen as external interactions. This awareness also permits coevolution, i.e., two or more species might need to evolve in response to each other. For instance, coevolution on two or more services implies that scaling decisions for one could have potential implications on the others. As a result, when making scaling decisions, implications on all services involved in the coevolution need to be catered for. To this end, Trophic-web-Awareness helps us to measure, understand and quantify those implications, thus enabling the evolution of one service with respect to the others, leading to better-informed decision making process. Trophic-web-Awareness corresponds to the Interaction-awareness in self-aware patterns.
  - *Example.* The cloud autoscaling process is able to recognize the relationships between cloud-based services and software/hardware resources, their functional dependency, and the topology. Specifically, when disturbances are detected, cloud-based services would need to evolve in or-
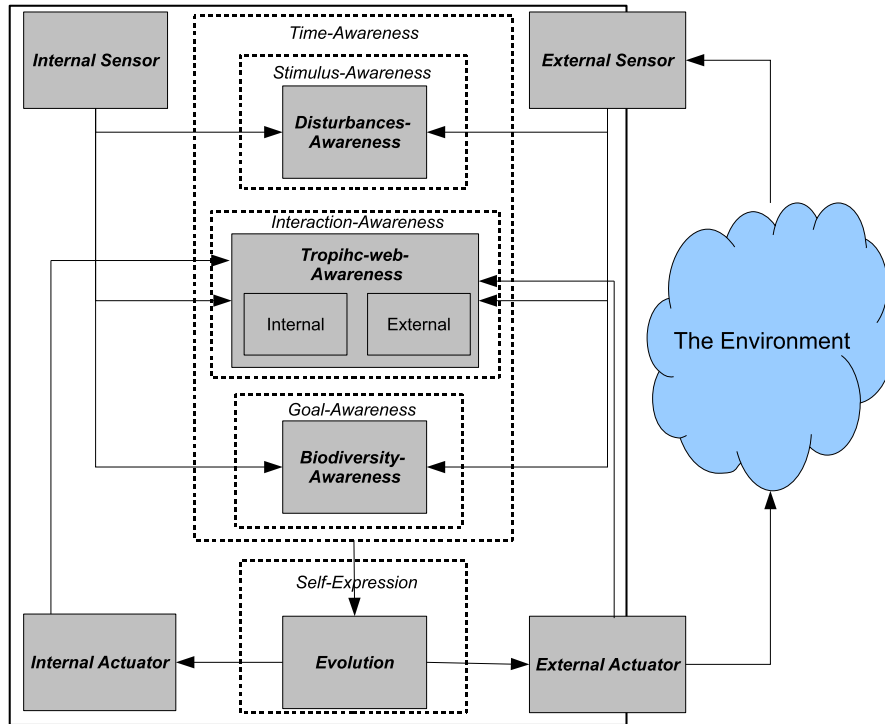
FIGURE 6.3

The ecology-inspired pattern

der to maintain stability. Knowledge of the trophic web can help us to explore the direction of evolution, e.g., assess the effects of a certain way of evolution, and comply with the constraints imposed by dependency and topology.

- *Biodiversity-Awareness.* Increasing biodiversity is the key to improving stability and sustainability of an ecosystem. Awareness of biodiversity permits the ability to reason about and acknowledge the effects of evolutions w.r.t. stability and sustainability for the entire ecosystem. Therefore, it corresponds to the Goal-awareness in self-aware patterns.
  - *Example.* The cloud autoscaling process is able to reason about how different forms of evolution can affect the biodiversity which, in turn, influences the long-term stability and sustainability related to QoS, cost and energy consumption in the cloud ecosystem. Such awareness could produce explicit answers on *when* the cloud should scale and *what* is the amount of scaling that leads to higher biodiversity. Concretely, biodiversity serves as the metric to assess the direction of evolution, thus providing guidance for the cloud ecosystem to maintain high biodiversity.
- *Evolution.* Evolution of an ecosystem can be regarded on two levels: (i) at the microevolution level, where the species can evolve themselves to adapt to the environment or the other species; and

(ii) at the macroevolution level, where the species can change their habitats. This corresponds to the Self-expression in self-aware patterns.
– *Example.* The cloud autoscaling process is able to know *how* the cloud ecosystem can evolve. This is concerned with whether vertical scaling, horizontal scaling, or both should be triggered. Here vertical scaling implies evolution at the microlevel, in which the cloud-based services can adapt their software configurations and hardware resource provisioning based on their demand. Similarly, horizontal scaling means evolution at the macrolevel, e.g., adding/removing cloud-based services and VM migration/replication, which can cause changes to their habitats.

All the levels of awareness in the cloud ecosystem can be connected to the Time-awareness from the self-aware patterns. This is because the natural ecosystem, which exists for millions of years, can often gain rich benefits from its long history. In the context of a computing system, such historical data can be relevant to disturbances, changes of trophic web, and evolution of the biodiversity. As a result, historical knowledge plays an integral role in our ecology-inspired pattern, for example, evolution of the biodiversity could gain insights from the past biodiversity levels, including their implication on the current cloud ecosystem and the entire path of evolution. This information can help us guide both the micro- and macrolevel evolution.

6.6 OPPORTUNITIES AND CHALLENGES

We have already shown that the cloud ecosystem exhibits many similarities to the natural ecosystem. Presumably, when cloud autoscaling leads to a higher level of biodiversity, global stability and sustainability of the cloud ecosystem would be expected to improve. This direction of research will create several opportunities and challenges:

- Autoscaling in the entire cloud is a complex and large-scale control problem. The notion of evolution from the natural ecosystem can provide inspiration about how to ensure high biodiversity in the cloud ecosystem, and thus improve global stability and sustainability. However, incorporating the control mechanisms of biodiversity and cloud autoscaling is a research challenge. Additionally, selecting the right measurements and form of biodiversity for cloud ecosystem is also a difficult issue. We expect to obtain similar perception as our prior work [7,9,3] when applying ecological principles to this challenge.
- Dynamics and uncertainty in the cloud significantly influence the design of an autoscaling system due to the time-varying workload, QoS interference, and the behaviors of cloud-based services. We hope that the mechanisms and models from the trophic web can better handle the dynamics and uncertainty. These mechanisms and models can provide us with insight about the interactions in different trophic levels and competition between species (i.e., cloud-based services). In addition, ecologists have applied several metrics (e.g., Shannon entropy) to quantify biodiversity in the trophic web. The challenge is how the trophic web can be used to correlate the QoS with cloud configurations and resources; and how it might influence the decision making of autoscaling.
- QoS interference and trade-offs are important issues in autoscaling decision making. Here, trade-offs do not only refer to the naturally conflicted objectives of the same cloud-based service (e.g., throughput vs. cost/energy), but also to the conflicted objectives for different cloud-based services

caused by QoS interference. While QoS interference might be tackled using computational intelligence [4,8,6,5], it is still challenging to study how the insights of species competition and coevolution can be used to resolve the trade-offs and to mitigate the effects of QoS interference in the cloud.

6.7 RELATED WORK

There have been some successful attempts in applying ecological principles, theories, and models to address issues in computer science. Examples can be found in the area of software engineering [19], collaborative adaptive systems [1,18], distributed computing [2], and grid computing [13]. However, there have been very limited efforts on adopting the ecological point of view for cloud autoscaling.

Among others, ECOS [19] is a research project that adopts ecological models to analyze the evolution of open-source software. Particularly, ecology-inspired methods are used to understand and better explain how the projects evolve, and what the factors are that drive the success of these projects. The goal is to optimize the fitness of software projects, leading to better stability of open-source software ecosystem.

Briscoe and Wilde [2] intend to apply ecological thinking to create scalable and self-organizing approaches for distributed evolutionary computing. The aim is to maintain a stable evolution of the processes in distributed environment, i.e., what processes should run independently or incorporated.

Ecology-inspired approach has been used in grid computing [13] with an aim to handle dynamics, uncertainty, diversity, and evolution in grid services. However, the authors focused on subscription, discovery, selection, and composition of grid-based services, as opposed to the cloud autoscaling of software configurations and hardware resources in our work.

The EU funded, multidisciplinary project DEVERISTY [1] is possibly the closest work to our research. DEVERISTY exploits the analogy with natural ecosystems in order to come up with better principles and mechanisms for handling the emergence of diversity in collaborative adaptive systems. The objective is to increase diversity in software systems and thus achieve better stability, as well as ability to react to unpredicted events [18]. Such an objective is consistent with our work, but we have particularly focused on the context of cloud computing and autoscaling, which exhibit some unique characteristics.

6.8 CONCLUSION

In this article, we have presented an intuitive and sensible transposition of ecological perspective to the context of cloud autoscaling. Deriving from existing self-aware software patterns, we have also proposed an architecture pattern for enabling such a transition, with respect to the different principles of natural ecosystem, including disturbance, trophic-web, biodiversity and (co-)evolution. The challenges for this direction of research were also discussed.

Cloud computing will continue to attract more and more participants for its scalable, elastic and on-demand promises. Stability and sustainability can quickly become critical quality indicators of cloud-based services, leading to several new challenges. Until recently, there has been an increasing interest in investigating the assurance of long term benefits in the cloud via autoscaling. To the

best of our knowledge, there is no known method that is explicitly developed for the long term benefit of stability and sustainability of cloud-based services. In this premise, the well-established ecological principles appear to be a neat solution for those challenges, as we have discussed in this article. This will further advance the existing view of the cloud and its autoscaling.

As future work, we aim to explicitly tackle the aforementioned research challenges, particularly focusing on the design of ecology driven mechanisms to handle dynamics and uncertainty exhibited in the cloud.

REFERENCES

[1] B. Baudry, M. Monperrus, C. Mony, F. Chauvel, F. Fleurey, S. Clarke, Diversify: ecology-inspired software evolution for diversity emergence, in: IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week, Feb. 2014, pp. 395–398.

[2] G. Briscoe, P. De Wilde, Ecosystem-oriented distributed evolutionary computing, in: 2012 12th UK Workshop on Computational Intelligence (UKCI), Sept. 2012, pp. 1–8.

[3] T. Chen, R. Bahsoon, Self-adaptive and sensitivity-aware QoS modeling for the cloud, in: Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS'13, IEEE Press, Piscataway, NJ, USA, 2013, pp. 43–52.

[4] T. Chen, R. Bahsoon, Symbiotic and sensitivity-aware architecture for globally-optimal benefit in self-adaptive cloud, in: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2014, pp. 85–94.

[5] T. Chen, R. Bahsoon, Self-adaptive trade-off decision making for autoscaling cloud-based services, IEEE Trans. Serv. Comput. (2015), http://dx.doi.org/10.1109/TSC.2015.2499770.

[6] T. Chen, R. Bahsoon, Towards a smarter cloud: self-aware autoscaling of cloud configurations and resources, Computer 48 (9) (Sept. 2015).

[7] T. Chen, R. Bahsoon, Self-adaptive and online QoS modeling for cloud-based software services, IEEE Trans. Softw. Eng. (2016), http://dx.doi.org/10.1109/TSE.2016.2608826.

[8] T. Chen, R. Bahsoon, G. Theodoropoulos, Dynamic QoS optimization architecture for cloud-based DDDAS, in: 2013 International Conference on Computational Science, Proc. Comput. Sci. 18 (2013) 1881–1890.

[9] T. Chen, R. Bahsoon, X. Yao, Online QoS modeling in the cloud: a hybrid and adaptive multi-learners approach, in: Proceedings of the IEEE/ACM 7th International Conference on Utility and Cloud Computing, 2014, pp. 327–336.

[10] T. Chen, F. Faniyi, R. Bahsoon, Design Patterns and Primitives: Introduction of Components and Patterns for SACS, Springer International Publishing, Cham, 2016, pp. 53–78.

[11] T. Chen, F. Faniyi, R. Bahsoon, P.R. Lewis, X. Yao, L.L. Minku, L. Esterle, The handbook of engineering self-aware and self-expressive systems, preprint arXiv:1409.1793, 2014.

[12] S. Forrest, J. Balthrop, M. Glickman, D. Ackley, Computation in the wild, in: Robust Design: A Repertoire of Biological, Ecological, and Engineering Case Studies, 2005, pp. 207–230.

[13] L. Gao, Y. Ding, L. Ren, A novel ecological network-based computation platform as a grid middleware system, Int. J. Intell. Syst. 19 (10) (2004) 859–884.

[14] R. Han, L. Guo, M. Ghanem, Y. Guo, Lightweight resource scaling for cloud applications, in: 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, May 2012, pp. 644–651.

[15] G. Hardin, et al., The competitive exclusion principle, Science 131 (3409) (1960) 1292–1297.

[16] S.P. Hubbell, The Unified Neutral Theory of Biodiversity and Biogeography, MPB, vol. 32, Princeton University Press, 2001.

[17] S.A. Levin, Ecosystems and the biosphere as complex adaptive systems, Ecosystems 1 (5) (1998) 431–436.

[18] D. Mendez, B. Baudry, M. Monperrus, Empirical evidence of large-scale diversity in API usage of object-oriented software, in: 2013 IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM), IEEE, 2013, pp. 43–52.

[19] T. Mens, M. Claes, P. Grosjean, Ecos: ecological studies of open source software ecosystems, in: IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering, 2014, pp. 403–406.

[20] S.T. Pickett, W. Carson, Ecology: individuals, populations and communities, Brittonia 39 (3) (1987).

[21] S. Yachi, M. Loreau, Biodiversity and ecosystem productivity in a fluctuating environment: the insurance hypothesis, Proc. Natl. Acad. Sci. 96 (4) (1999) 1463–1468.

[22] Q. Zhu, G. Agrawal, Resource provisioning with budget constraints for adaptive applications in cloud environments, IEEE Trans. Serv. Comput. 5 (4) (2012) 497–511.