EDITOR **SAN MURUGESAN**
BRITE Professional Services;
san@computer.org

Toward a Smarter Cloud: Self-Aware Autoscaling of Cloud Configurations and Resources

Tao Chen and Rami Bahsoon, University of Birmingham

Promoting self-aware autoscaling to intelligently handle the dynamics and uncertainty of changing workloads, configurations, and demands on resources at runtime can facilitate more scalable, elastic, and dependable cloud-based services.

Today's IT service providers are increasingly leveraging cloud computing to offer flexible, scalable, and cost-effective services. Typically, organizations can tune the quality of service (QoS) for cloud-based services by accessing hardware resources (such as the CPU and memory) and software configurations (such as the number of service threads) that are shared, leased, and priced as utilities. Autoscaling—an automatic, elastic process that adjusts software configurations and hardware resources to meet changing workloads' needs—facilitates the necessary adaptive configuring and resource provisioning in the cloud.

However, cloud-based services exhibit dynamics and uncertainty related to changing workloads, configurations, and demands on resources, which are hard to cope with. In addition, because many cloud services run on a shared infrastructure, the competing demands of some of the services can interfere with the QoS of others. This *QoS interference* has made autoscaling even more difficult to manage. We believe self-awareness can solve these problems.

LIMITATIONS AND OPPORTUNITIES

Current autoscaling systems have limited adaptivity. Today's cloud QoS models rely on static offline analysis.[1,2] This nondynamic approach doesn't accurately model QoS sensitivity to changing configurations, resources, and workloads or to QoS interference. Thus, it limits autoscaling quality and the cloud's promised benefit as an on-demand, shared, and elastic resource.

Within a shared cloud infrastructure, the granularity of autoscaling control can affect both benefits and overhead. Currently, organizations typically don't change the level and granularity of control on an entire cloud, physical machine, virtual machine (VM), or service.

Controlling an entire cloud[1] might achieve global benefits but would also greatly increase overhead. On the other hand, a more local level of control—such as on a single service—would yield low overhead but would be inadequate if global benefits are required.[2] The challenge lies in dynamically determining the level of control granularity needed.

> Self-awareness represents a promising avenue for improving self-adaptivity and autoscaling.

The process of deciding the proper autoscaling level for hardware resources and software configurations is far from trivial in the presence of QoS interference or of multiple, and possibly conflicting, nonfunctional goals such as maximizing throughput or minimizing costs.

One state-of-the art solution is carefully specifying if-condition-then-action mapping.[1] For example, the mapping could state that if a service's throughput drops below 100 requests per minute, the autoscaling system should increase the VM's memory by 200 Mbytes and the number of service threads by 20. The difficulty is identifying the best autoscaling decisions from among the many possible combinations of configurations and resources, and mapping them to the potential conditions in a way that best achieves the desired goals. Self-awareness can address this.

ENGINEERING SELF-AWARE AUTOSCALING

A system that is aware of its current state and environment can better reason about its adaptive behaviors. For example, a self-aware autoscaling process could obtain knowledge about its own or other participants' impact on a managed service's QoS models, the desired granularity of autoscaling control, and the quality of various runtime decisions about tradeoffs. These tradeoffs include whether to choose throughput over cost or which competing cloud-based services to focus on.

Through self-awareness in the autoscaling process, we propose bidirectional adaptations between the autoscaling logic and the managed services or VMs. Currently, the adaptation in the autoscaling process is often one-way. Self-awareness provides the possibility of two-way adaptation so that the autoscaling process can effectively scale and adapt its managed services and VMs.

In one direction, the autoscaling logic adapts the software configurations and hardware resources for a system's managed services and VMs. Adaptations in the other direction capture and expand knowledge about the time-varying state of its managed services and VMs, as well as the environment. It uses this knowledge to build more accurate QoS models, identify the best levels of control granularity, and yield better tradeoff decisions.

Our self-aware autoscaling approach, shown in Figure 1, is consistent with the EU's Engineering Proprioception in Computing Systems (EPiCS) project,[3] a multinational, multidisciplinary, self-aware-computing research program. Our approach acts as decentralized middleware, in which each middleware instance contributes to self-awareness.

Our conceptual architecture includes modules based on various types of self-awareness[4] (Figure 1, left) so that the system can be effectively enriched with self-awareness capabilities important to autoscaling (Figure 1, right). These are implemented in processes run by our QoS modeler, region controller, and decision maker.

QoS modeler

This element captures QoS sensitivity to QoS interference and changing configurations, resources, and workloads.

The QoS modeling approach is enriched by self-awareness[5,6] to dynamically self-adapt the selection and expression of QoS models at runtime. This self-awareness helps the system determine how its modeling can be affected by:

- the features of and changes to the managed cloud-based services that affect QoS, such as workload (*stimulus awareness*);
- interactions entailing, for example, QoS interference and resource contention (*interaction awareness*);
- historical modeling errors and data trends (*time awareness*);
- changes in nonfunctional goals (*goal awareness*); and
- the suitability of the QoS modeling algorithms being used (*meta-self-awareness*).

By leveraging improvements in machine-learning algorithms, self-awareness eliminates the need for heavy human analysis and prior design-time knowledge in the QoS modeling. However, because there is no single best algorithm for all situations,[5] selecting the right one can be challenging. Meta-self-awareness can address this issue by allowing the modeling process to identify the algorithm that will best enable other self-awareness capabilities.

The self-aware QoS modeling approach has been validated by prior research conducted in real cloud settings.[5,6] The results show that this approach offers several advantages—including lower overhead, better accuracy for workload spikes, and better stability when cloud-based services' requirements change—over state-of-the-art modeling approaches.

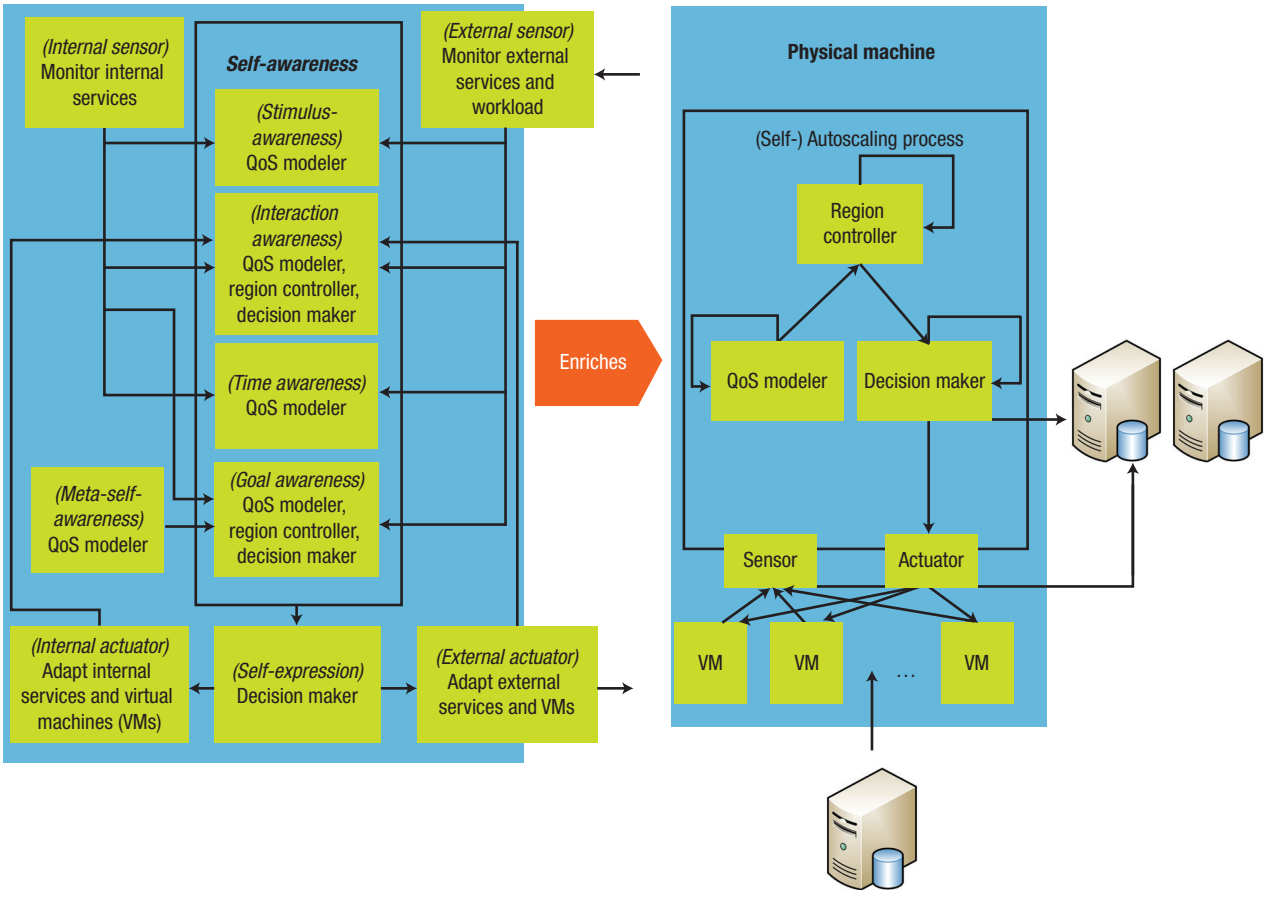

Figure 1. Self-aware autoscaling system for the cloud. On the left is the mapping between architectural components and the desired computational self-awareness capabilities. On the right is the architecture enriched by those capabilities. QoS: quality of service.

Region controller

This element determines the best level of autoscaling control granularity—yielding the best combination of benefit and overhead—under a given set of circumstances.

Self-awareness helps with this by clustering nonfunctional goals into different regions,[7] representing the logical groups for storing goals affected by the same software-configuration and hardware-resource parameters. The clustering helps separate the goals into different groups that can be considered in separate decision-making processes.

In this case, self-awareness is concerned with knowing how the region-controlling process can be affected by:

- factors such as QoS sensitivity and cost models (*goal awareness*), and
- conflicting or nonconflicting goals (*interaction awareness*).

With such awareness, the controller can self-adapt the regions and their content to changes in conditions. In particular, the controller could identify regions of goals that could be separated into different decision-making processes. This would enable dynamic implementation of autoscaling control granularity, which helps improve results and reduce overhead.

Decision maker

This element extensively reasons about and searches for the best decision as to the amount of scaling that should be applied to software configurations and hardware resources.

Self-awareness can help improve tradeoff-related decision making at runtime, particularly in the presence of changing circumstances. Here, self-awareness is concerned with knowing how decision making can be affected by:

- factors such as desired QoS levels, budget constraints, QoS sensitivity, and cost models (*goal awareness*); and
- interactions among regions of goals, which help determine which goals should be considered in the same decision-making process and which

could be considered in isolation (*interaction awareness*).

Self-awareness that leverages advanced search-based algorithms helps decision making via extensive reasoning about the effects of various autoscaling decisions and tradeoffs on goal achievement. This is particularly helpful in situations with many possible software configurations and levels of hardware resources. More importantly, self-awareness provides knowledge that helps the decision-making process self-adapt and improve.

To unlock the full potential of self-aware autoscaling in the cloud, providers and developers should design their services' configurable parameters so that the services are accessible and can be seamlessly adapted at runtime. Instead of using fixed bundles of VMs for scaling, providers should offer flexible software configurations and levels of hardware resources. In addition, they should support both vertical and horizontal scaling.

For researchers, applying self-aware autoscaling within a cloud federation is still in its infancy. The difficulty lies in how self-awareness capabilities could be incorporated with multiple, heterogeneous, possibly competing cloud providers that, in some cases, work with different standards.

As cloud computing evolves, autoscaling will require novel principles and approaches to seamlessly managing cloud-based services. Self-awareness represents a promising avenue for improving self-adaptivity and autoscaling.

REFERENCES

1. I. Brandic et al., "LAYSI: A Layered Approach for SLA-Violation Propagation in Self-Manageable Cloud Infrastructures," *Proc. IEEE 34th Ann. Conf. Computer Software and Applications* (COMPSAC 10), 2010, pp. 365–370.
2. H. Fernandez, G. Pierre, and T. Kielmann, "Autoscaling Web Applications in Heterogeneous Cloud Infrastructures," *Proc. IEEE Int'l Conf. Cloud Eng.* (IC2E 14), 2014, pp. 195–204.
3. T. Becker et al., "EPiCS: Engineering Proprioception in Computing Systems," *Proc. 15th IEEE Int'l Conf. Computational Science and Eng.* (CSE 12), 2012, pp. 353–360.
4. T. Chen et al., *The Handbook of Engineering Self-Aware and Self-Expressive Systems*, tech. report; http://arxiv.org/abs/1409.1793, 2014.
5. T. Chen and R. Bahsoon, "Self-Adaptive and Sensitivity-Aware QoS Modeling for the Cloud," *Proc. 8th Int'l Symp. Software Eng. Adaptive and Self-Managing Systems* (SEAMS 13), 2013, pp. 43–52.
6. T. Chen, R. Bahsoon, and X. Yao, "Online QoS Modeling in the Cloud: A Hybrid and Adaptive Multi-Learners Approach," *Proc. 7th IEEE/ACM Int'l Conf. Utility and Cloud Computing* (UCC 14), 2014, pp. 327–336.
7. T. Chen and R. Bahsoon, "Symbiotic and Sensitivity-Aware Architecture for Globally-Optimal Benefit in Self-Adaptive Cloud," *Proc. 9th Int'l Symp. Software Eng. Adaptive and Self-Managing Systems* (SEAMS 14), 2014, pp. 85–94.

TAO CHEN is a doctoral researcher at the School of Computer Science, University of Birmingham, UK. Contact him at txc919@cs.bham.ac.uk.

RAMI BAHSOON is a senior lecturer in software engineering at the School of Computer Science, University of Birmingham. Contact him at r.bahsoon@cs.bham.ac.uk.