Elasticity in the Cloud

By David Chiu

Take a second to consider all the essential services and utilities we consume and pay for on a usage basis: water, gas, electricity. In the distant past, some people have suggested that computing be treated under the same model as most other utility providers.

The case could certainly be made. For instance, a company that supports its own computing infrastructure may suffer from the costs of equipment, labor, maintenance, and mounting energy bills. It would be more cost-effective if the company paid some third-party provider for its storage and processing requirements based on time and usage.

While it made perfect sense from the client's perspective, the overhead of becoming a computing-as-a-utility provider was prohibitive until recently. Through advancements in virtualization and the ability to leverage existing supercomputing capacities, utility computing is finally becoming realized.

Known to most as cloud computing, leaders, such as Amazon Elastic Compute Cloud (EC2), Azure, Cloudera, and Google's App Engine, have already begun offering utility computing to the mainstream.

A simple, but interesting property in utility models is elasticity, that is, the ability to stretch and contract services directly according to the consumer's needs.

Elasticity has become an essential expectation of all utility providers. When's the last time you plugged in a toaster oven and worried about it not working because the power company might have run out of power? Sure, it's one more device that sucks up power, but you're willing to eat the cost. Likewise, if you switch to using a more efficient refrigerator, you would expect the provider to charge you less on your next billing cycle.

What elasticity means to cloud users is that they should design their applications to scale their resource requirements up and down whenever possible. However, this is not as easy as plugging or unplugging a toaster oven.

A Departure from Fixed Provisioning

Consider an imaginary application provided by my university, Ohio State. Over the period of a day, this application requires 100 servers during peak time, but only a small fraction of that during down time. Without elasticity, Ohio State has two options: either provision a fixed amount of 100 servers, or less than 100 servers.

While the former case, known as over-provisioning, is capable of handling peak loads, it also wastes servers during down time. The latter case of under-provisioning might address, to some extent, the presence of idle machines. However, its inability to handle peak loads may cause users to leave its service.

By designing our applications to scale servers accordingly to the load, the cloud offers a departure from the fixed provisioning scheme.

To provide an elastic model of computing, providers must be able to support the sense of having an unlimited number of resources. Because computing resources are unequivocally finite, is elasticity a reality?

Sharing Resources

In the past several years, we have experienced a new trend in processor development. CPUs are now being shipped with multi- and many-cores on each chip in an effort to continue the speed-up, as predicted by Moore's Law. However, the superfluous cores (even a single core) are underutilized or left completely idle.

System engineers, as a result, turn to statistical multiplexing for maximizing the utilization of today's CPUs. Informally, statistical multiplexing allows a single resource to be shared by splitting it into variable chunks and allocating each to a consumer. In the meantime, virtualization technology, which allows several instances of operating systems to be run on a single host machine, has matured to a point of production. Virtualization has since become the de-facto means toward enabling CPU multiplexing, which allows cloud providers to not only maximize the usage of their own physical resources, but also multiplex their resources among multiple users. From the consumers' perspective, they are afforded a way to allocate on-demand, independent, and more important, fully-controllable systems.

> **"What elasticity means to cloud users is that they should design their applications to scale their resource requirements up and down whenever possible."**

But even with virtualization, the question persists: What if the physical resources run out? If that ever occurred, the provider would simply have to refuse service, which is not what users want to hear.

Currently, for most users, EC2 only allows 20 simultaneous machine instances to be allocated at any time. Another option might be to preempt currently running processes. Although both are unpopular choices, they certainly leave room for the provider to offer flexible pricing options. For instance, a provider can charge a normal price for low-grade users, who might be fine with having their service interrupted very infrequently. High-grade users, on the other hand, can pay a surplus for having the privilege to preempt services and also to prevent from being preempted.

Looking Forward

With the realization of cloud computing, many stakeholders are afforded on-demand access to utilize any amount of computing power to satisfy their relative needs. The elastic paradigm brings with it exciting new development in the computing community. Certainly, scaling applications to handle peak loads has been a long-studied issue.

While downscaling has received far less attention in the past, the cloud invokes a novel incentive for applications to contract, which offers a new dimension for cost optimization problems. As clouds gain pace in industry and academia, they identify new opportunities and may potentially transform computing, as we know it.

Biography

David Chiu is a student at The Ohio State University and an editor for Crossroads.