ibm.com

Rapid elasticity and the cloud - Cloud computing news

Edwin Schouten

7-9 minutes

September 12, 2012 I Written by:

Categorized: Storage

Share this post:

Screen Shot 2015-04-03 at 11.18.26 AM

More cloud providers nowadays are advertising their capabilities on elastic scaling, but what is this exactly and how do we benefit from this?

One of the five characteristics of cloud computing

In 2011 the National Institute of Standards and Technology (NIST) defined cloud computing in their publication, “The NIST Definition of

Cloud Computing", NIST Special Publication 800-145. Many vendors and experts since than have recognized this as the universal definition of cloud computing. To refresh your memory:

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

With their definition of cloud computing, also five essential characteristics have been defined that every true cloud computing implementation should have:

- On-demand self service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured Service

Back to elasticity

Elasticity is basically a 'rename' of scalability, which has been a known non-functional requirement in IT architectures for many years already. Scalability is the ability to add or remove capacity, mostly processing, memory, or both, to or from an IT environment when this is needed. This it typically done in two ways:

- **Horizontal** scalability: Adding or removing nodes, servers or instances to or from a pool like a cluster or farm.
- **Vertical** scalability: Adding or removing resources to an existing node, server or instance to increase the capacity of the node, server or instance.

Most implementations of scalability are implemented using the horizontal method, as this easiest to implement especially in the current web-based world we live in. A well known example is adding a load balancer in front of a farm of web servers that distributes the requests. Vertical scaling is less dynamic most of the time because this requires reboots of systems, sometimes adding physical components to servers.

Why call it elasticity?

Well, traditional IT environments do build-in scalability in their architecture, but actual scaling up or down is not done very often. This has to do with the amount of time, effort and cost accompanied with scaling up. Servers have to be bought, operations needs to screw these into server racks, install and configure them, and then the test team needs to validate the functioning and only after all this has happened you have scaled up. And you don’t buy a server for just a few months — normally this is three to five years. So it’s a long term investment you make.

Elasticity is doing the same, but more like a rubber band. You ‘stretch’ the capacity when you need it and ‘release’ it when you don’t anymore. And this is possible because of some other characteristics of cloud computing, namely “resource pooling” and “on-demand self service.” Combining these characteristics with advanced image management capabilities allows you to scale in a much more agile manner.

Three forms for scalability

Below I describe the three forms of scalability as I see them, describing what makes them different from each other.

Manual scaling

Manual scalability starts with a manual forecast of the expected workload on the cluster or farm of resources, then manually adding resources to add capacity. This is done using mostly physical servers, which are manually installed and configured. Ordering, installing and configuring physical resources take a lot of time, so the forecasting needs to be done weeks if not months in advance. Another downside to manual scalability is that removing resources mostly does not result in cost savings as the physical server has been paid for already.

Semi-automated scaling

Semi-automated scalability takes advantage of the concept of virtual servers, which are provisioned (installed) using predefined images. Either a manual forecast or automated warning of system monitoring tooling will trigger operations to expand or reduce the cluster or farm of resources.

By using predefined, tested and approved images, every new virtual server will be exactly the same as all the others (except for some minor configuration) which gives you repetitive results. This also reduced the manual labor on the systems significantly, and it is a well known fact that around 70 to 80 percent of all errors are caused by manual actions on systems.

Using virtual servers also has a huge benefit, this does allow getting cost savings once a virtual server is de-provisioned (removed). Freed resources can be directly utilized for other purposes.

Elastic scaling (fully-automated scaling)

Elasticity, or fully-automated scalability, takes advantage of the same concepts that semi-automated scalability does, but removes any manual labor needed to increase or reduce capacity. Everything is controlled by triggers of the system monitoring tooling, which gives you this “rubber band” effect. If more capacity is needed now, then it’s added now and there within minutes. No need for the additional capacity anymore? Based on the system monitoring tooling the capacity is reduced instantly.

IBM has a number of offerings today that can help a customer with this topic, and is working on expanding them similar to Amazon Elastic Beanstalk and Microsoft Azure which can be thought of as an integrated service solution providing platform as a service (PaaS) capabilities. See an online demo here.

Other IBM solutions are for instance Kaavo – an IBM partner image available in IBM SmartCloud Enterprise which offers a monitoring and management platform that supports creation of policies for scaling applications on the IBM Cloud and WebSphere Virtual Enterprise which also provides policy based scaling for J2EE applications, and can be deployed on a public and private cloud infrastructure.

Considerations

Elastic scaling is indeed a great feature, but there are some things to consider. For example, if an application defect causes CPU usage to spike, would you really want to wake up in the morning to have your footprint expanded by thousands of virtual machines? Naturally this all comes down to setting the right monitoring triggers and configuring the elastic scaling with maximum limits. Always stay in control and implement “what if” scenarios.

Another example is the software licensing impact; adding new instances of an application that uses licensed software most likely will also require you to obtain additional licenses or at least evaluate the licensing model to validate that it doesn’t. A short checklist for your consideration to implement elastic scaling:

- Under what conditions should the application scale up or down?

- What pieces should scale?
- How would the customer account for surrounding systems such as monitoring, or backup?
- Should this process be automatic or should a manual approval step be part of the solution?
- Are there other business considerations such as licensing or contracts that need to be considered?

  Similar to cloud computing itself, elastic scalability is a great technology with a potentially huge business value, but it does not automatically solve all of your problems.