Nimbus Elastic Scaling in the Clouds

Tim Freeman, David LaBissoniere, Paul Marshall, John Bresnahan, Kate Keahey

The need for computational power often comes in bursts: it may be triggered by real-time sensors returning data indicating some sort of anomaly like a hurricane, earthquake, or market disruption causing demand for additional processing and analysis or it may be tied to the support of periodically conducted experiments, educational activities, or even conference deadlines. The ability to elastically expand and contract the resource base as needed in such cases allows us to react to such need with a good response time. Our poster presents early experiments with providing this capability as well as the architecture and implementation of a production implementation of a highly available elastic scaling system.

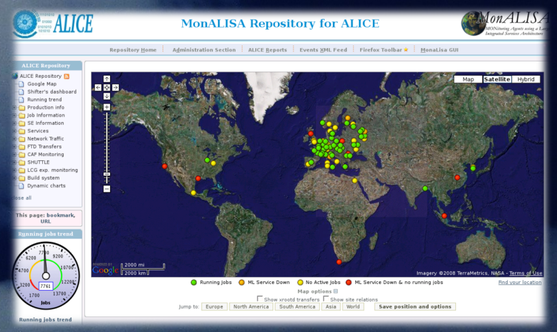
Examples: What Elastic Scaling Can Do for You

Proof-of-Concept Elastic Scaling with ALICE

The ALICE high-energy physics experiment at CERN runs its computational jobs on a collection of globally distributed resources, managed by a scheduler called AliEn. In 2008 we collaborated with ALICE scientists to elastically extend the resources available to AliEn. The idea was to monitor the length of the work queue for AliEn: if the queue length got beyond certain threshold additional virtual machines (VMs) were provisioned in the cloud.

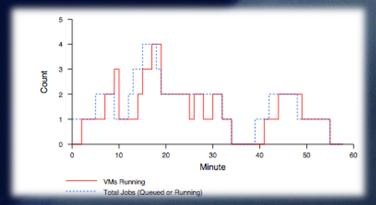
In order for the AliEN scheduler to work with the newly provisioned resources, the Nimbus Context Broker was used to securely integrate the deployed VMs into the AliEn testbed. Once the VMs were joined to the testbed, the scheduler would schedule jobs on them. The system was used to execute production workloads and proved that elastic extension of the testbed into the cloud can be achieved. This work was a collaboration with Artem Harutyunyan and Predrag Buncic at CERN.

See also:
Harutyunyan et al., "Dynamic Virtual AliEn Grid Sites on Nimbus with CernVM", CHEP 2009



MonaLISA reporting for ALICE shows the availability of additional VMs deployed on the Nimbus cloud at University of Chicago.

Elastic Site Prototype



Torque jobs and running VMs, using the on-demand policy. VMs are launched and terminated as needed.

In our early elastic computing work we developed a model of an "elastic site" that efficiently adapted services within a site such as batch schedulers, storage archives, or Web services to take advantage of elastically provisioned IaaS cloud resources. We implemented a resource manager, built on the Nimbus toolkit to dynamically and securely extend a physical Torque cluster into the cloud.

We developed and evaluated a set of policies for resource provisioning on a Nimbus-based cloud at the University of Chicago, another at Indiana University, and Amazon EC2. The policies consisted of an on-demand policy that reacted immediately to changes in demand, a steady stream policy that responded conservatively to changes in demand, and a bursts policy that attempted to launch an appropriate number of nodes based on the amount of job queued wall time.

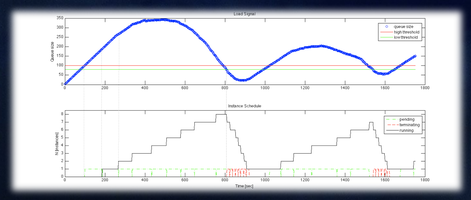
See also:
Marshall P, Keahey K, Freeman T. "Elastic Site: Using clouds to elastically extend site resources," in IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), May 2010.

Pilot Project for the Ocean Observatories Initiative

Elastic scaling was again demonstrated in the pilot project for the Ocean Observatories Initiative. We built a prototype based on reliable AMQP messaging. Worker nodes were virtual machines (VMs) running on Amazon EC2. Each node subscribed to an AMQP queue and accepted work. Meanwhile, the queue was monitored by other processes that dynamically add and remove nodes in accordance with a policy.

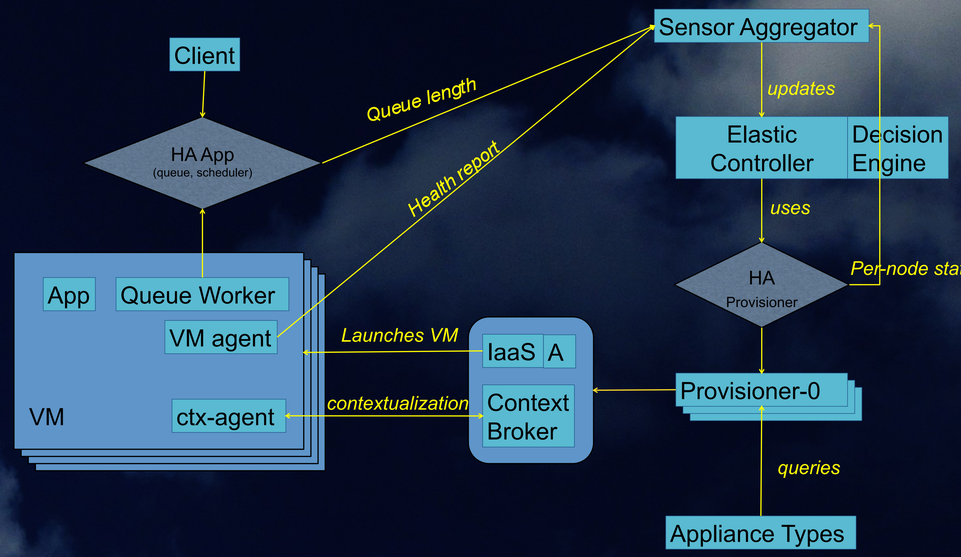
In our demo, a naïve policy was used that launched and terminated worker VMs based on the count of messages in the work queue. When the count was above a threshold, a node was added. Likewise, a node was removed whenever the count was below another threshold. As the graph demonstrates, the long bootstrap time of workers has a substantial effect on system behavior.

See also:
Arrott M, Clemesha A, Farcas C, Farcas E, Meisinger M, Raymer D, LaBissoniere D, Keahey K. "Cloud Provisioning Environment: Prototype Architecture and Technologies". Ocean Observatories Initiative Kick Off Meeting. Tabernash, CO. September 2009.



Work queue size and running VM count, over time. The policy launches and terminates VMs based on the queue size.

Highly Available Production Elastic Services



Abstraction: a workload queue
- e.g., ALiEn, PBS, AMQP,…

React to sensor information
- properties of the queue: longer, shorter…
- health information about the VM workers
- general deployment information
- potentially other information about the system

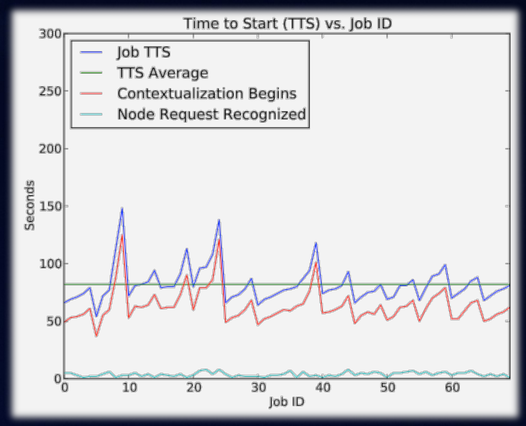
Integrate policy and resource information
- policies on how to scale
- evaluated against needs and resources

Scale to demand
- provision resources across different cloud providers
- use contextualization to integrate new resources

Highly Available Services
- Integrating the operation chain

Customizable to input, policy, provider, etc. and Scalable
- Latest tests scale to 100s of nodes on EC2, working on 1000s

Release in early 2011



This graph shows elastic processing scaling under an average load scenario: 5 jobs are submitted every 2 minutes over 28 minute period; a new VM is deployed for each job.

Acknowledgements

FutureGrid
Computation Institute
National Science Foundation
OOI Ocean Observatories Initiative