Elasticity Engineering

Real-world & Academic Implementation

Hong-Linh Truong
Faculty of Informatics, TU Wien

hong-linh.truong@tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/truong
@linhsolar
Still remember?

What is elasticity?

What is elastic computing?

Tasks in elasticity engineering?
Points of discussion in elasticity support

- **When**
  - When should we perform elasticity controls?

- **Where**
  - Where should we apply elasticity controls?

- **What**
  - What kind of elasticity we will control?

- **How**
  - How do we perform the elasticity controls?
Points of discussion in elasticity support

- Metrics for deciding elasticity
- Software and infrastructure stacks
  - Applications, middleware, compute resources or networks?
- Proactive versus reactive
- Centralized versus decentralized controls
- Reactive or predictive elasticity controls
- Synchronous or asynchronous lockstep
Microsoft Azure Elasticity Rules

Source: https://msdn.microsoft.com/en-us/library/hh680881%28v=pandp.50%29.aspx
Auto-scaling Examples from Amazon services

Sources: http://docs.aws.amazon.com/autoscaling/latest/userguide/policy_creating.html
Understand metrics and rules for elasticity

<table>
<thead>
<tr>
<th>Reference</th>
<th>Auto-scaling Techniques</th>
<th>H/V</th>
<th>R/P</th>
<th>Metric</th>
<th>Monitoring</th>
<th>SLA</th>
<th>Workloads</th>
<th>Experimental Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>[65]</td>
<td>Rules</td>
<td>Both</td>
<td>R</td>
<td>CPU, memory, I/O</td>
<td>Custom tool. 1 minute</td>
<td>Response time</td>
<td>Synthetic, Browsing and ordering behavior of customers.</td>
<td>Custom testbed (called UC Cloud) + TPC</td>
</tr>
<tr>
<td>[72]</td>
<td>Rules</td>
<td>H</td>
<td>R</td>
<td>Average waiting time in queue, CPU load</td>
<td>Custom tool.</td>
<td>—</td>
<td>Synthetic</td>
<td>Public cloud, FutureGrid, Eucalyptus India cluster</td>
</tr>
<tr>
<td>[64]</td>
<td>Rules</td>
<td>Both</td>
<td>R</td>
<td>CPU load, response time, network link load, jitter and delay</td>
<td>—</td>
<td>—</td>
<td>Only algorithm is described, no experimentation is carried out.</td>
<td></td>
</tr>
<tr>
<td>[48]</td>
<td>Rules + QT</td>
<td>H</td>
<td>P</td>
<td>Request rate</td>
<td>Amazon CloudWatch, 1-3 minutes</td>
<td>Response time</td>
<td>Real, Wikipedia traces</td>
<td>Real provider, Amazon EC2 + Hyperf + MediaWiki</td>
</tr>
<tr>
<td>[52]</td>
<td>RightScale + MA to performance metric</td>
<td>H</td>
<td>R</td>
<td>Number of active sessions</td>
<td>Custom tool</td>
<td>—</td>
<td>Synthetic, Different number of HTTP clients</td>
<td>Custom testbed, Xen + custom collaborative web application</td>
</tr>
<tr>
<td>[73]</td>
<td>RightScale + TS: LR and AR(1)</td>
<td>H</td>
<td>R/P</td>
<td>Request rate, CPU load</td>
<td>Simulated</td>
<td>—</td>
<td>Synthetic, Three traffic patterns: weekly oscillation, large spike and random</td>
<td>Custom simulator, tuned after some real experiments</td>
</tr>
<tr>
<td>[59]</td>
<td>RightScale</td>
<td>H</td>
<td>R</td>
<td>CPU load</td>
<td>Amazon CloudWatch</td>
<td>—</td>
<td>Real, World Cup 98</td>
<td>Real provider, Amazon EC2 + RightScale (PaaS) + a simple web application</td>
</tr>
<tr>
<td>[96]</td>
<td>RightScale + Strategy-tree</td>
<td>H</td>
<td>R</td>
<td>Number of sessions, CPU idle</td>
<td>Custom tool, 4 minutes</td>
<td>—</td>
<td>Real, World Cup 98</td>
<td>Real provider, Amazon EC2 + RightScale (PaaS) + a simple web application</td>
</tr>
<tr>
<td>[81]</td>
<td>Rules</td>
<td>V</td>
<td>R</td>
<td>CPU load, memory, bandwidth, storage</td>
<td>Simulated</td>
<td>—</td>
<td>Synthetic</td>
<td>Custom simulator, plus Java rule engine Drools</td>
</tr>
<tr>
<td>[77]</td>
<td>Rules</td>
<td>V</td>
<td>R</td>
<td>CPU load</td>
<td>Simulated, 1 minute</td>
<td>Response time</td>
<td>Real, ClarkNet</td>
<td>Custom simulator</td>
</tr>
</tbody>
</table>

Table rows are as follow, (1) The reference to the reviewed paper, (2) A short description of the proposed technique, (3) The type of auto-scaling: horizontal (H) or vertical (V), (4) The reactive (R) and/or proactive (P) nature of the proposal, (5) The performance metric or metrics driving auto-scaling, (6) The monitoring tool used to gather the metrics, The remaining three fields are related to the environment in which the technique is tested, (7) The metric used to verify SLA compliance, (8) The workload applied to the application managed by the auto-scaler, (9) The platform on which the technique is tested.

Types of controls in distributed systems

Which models are for elasticity controls?

(a) Centralized scheme.

(b) Multi-layer scheme.

(c) Single-layer scheme.

Figure Source: Gabriele Mencagli. 2016. *A Game-Theoretic Approach for Elastic Distributed Data Stream Processing*. ACM Trans. Auton. Adapt. Syst. 11, 2, Article 13 (June 2016), 34 pages. DOI: https://doi.org/10.1145/2903146
Predictive Model Control

Figure source: Tiziano De Matteis and Gabriele Mencaglì. 2017. Proactive elasticity and energy awareness in data stream processing. J. Syst. Softw. 127, C (May 2017), 302-319. DOI: https://doi.org/10.1016/j.jss.2016.08.037

Arrival rate, processing time, network throughput, etc.

Configurations & Metrics relationships

Control/reconfiguration actions
WARNING: You need to read papers to see the details!

SOME SELECTED ISSUES
Elasticity for Compute Resources

- Online adaptive padding
- Reactive error correction
- Deal with conflict

Elasticity for Compute Resources

Figure 2: The CloudScale system architecture.

Elasticity from computing resources

Also take a look at https://mesos.github.io/chronos

Elasticity in streaming data processing

- Streaming data processing
  - What are key constructs and operators?

Source: https://apex.apache.org/docs/apex-3.6/operator_development/

Elasticity: When, where, what, how?
Example in Apache Apex

- Dynamic Partition
  - Partition operators
  - Dynamic: specifying when a partition should be done
  - Unifiers for combining results (reduce)

- StreamCodec
  - For deciding which tuples go to which partitions
  - Using hashcode and masking mechanism

Source: https://apex.apache.org/docs/apex/application_development/#partitioning
Example with ElasticStream

- Elasticity:
  - Where?
  - When?
  - What?
  - How?

Other works


Example in database as a service

Elasticity: When, where, what and how?


Example in network layers

- Elasticity: Where and When
- What are important constraints during the elasticity control
- How do we do elasticity?

Distributed Coordination

- Follow the generic “distributed coordination”
- Cooperative versus non-cooperative models

Figure Source: Gabriele Mencagli. 2016. *A Game-Theoretic Approach for Elastic Distributed Data Stream Processing*. ACM Trans. Auton. Adapt. Syst. 11, 2, Article 13 (June 2016), 34 pages. DOI: https://doi.org/10.1145/2903146
Summary

- Multi-dimensional elasticity
  - Most work are just about resources
  - Performance metrics
- Elasticity engineering across platforms
  - Not really: some work across data centers but with the same software stack
- End-to-end elasticity toolsets
  - Usually they are not generic for different systems
  - But they follow generic models for components and engineering steps
Topics for you

- Software and infrastructure stacks
  - Elasticity in streaming processing, computing resources (VM or containers), databases, or in network controls
  - Vertical or horizontal elasticity

- Controls
  - Centralize or decentralized, Metrics, Algorithms

- Theoretical work or practical work
  - Theoretical: read selected papers & show your understanding/design on how to apply controls to your familiar systems (In assignment 1)
  - Practical: read selected papers & implement some (simple) controls with your familiar systems (In assignment 1)
Thanks for your attention

Hong-Linh Truong
Faculty of Informatics, TU Wien

hong-linh.truong@tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/truong