Advanced Topics in Service-Oriented Computing and Cloud Computing, Winter 2017

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

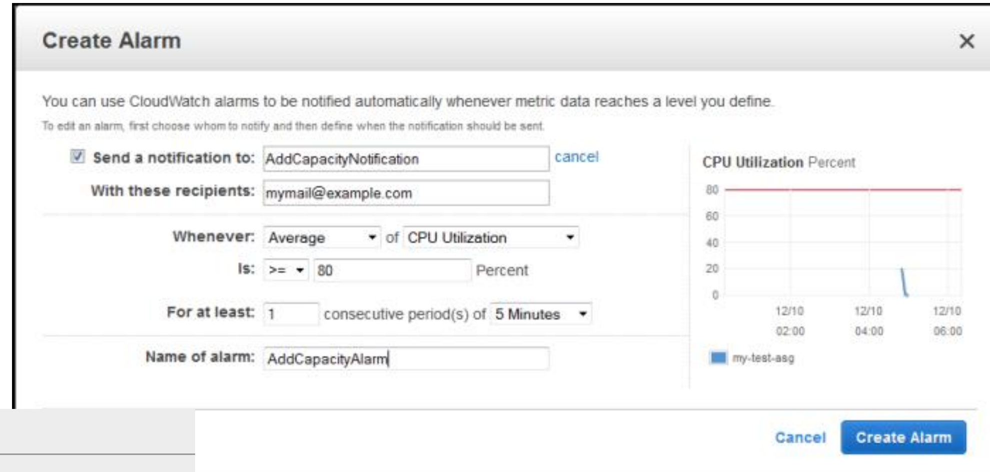
Microsoft Azure Elasticity Rules

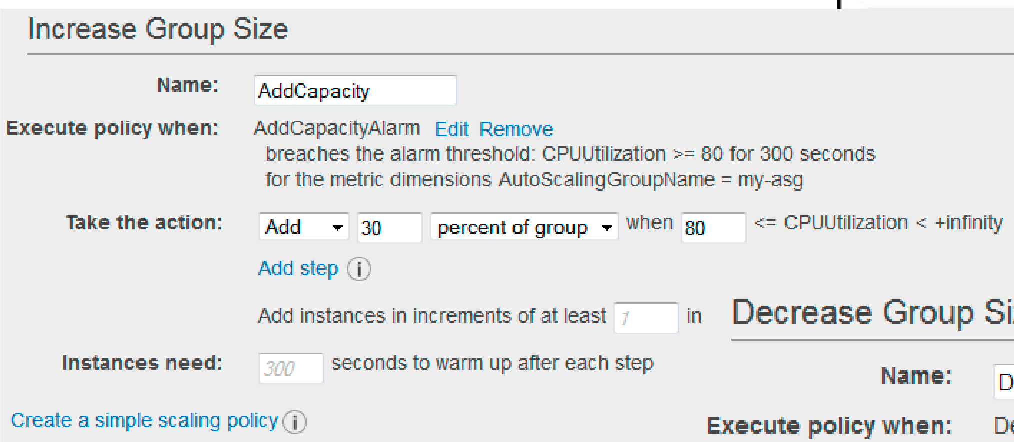
Source: https://msdn.microsoft.com/en-us/library/hh680881%28v=pandp.50%29.aspx

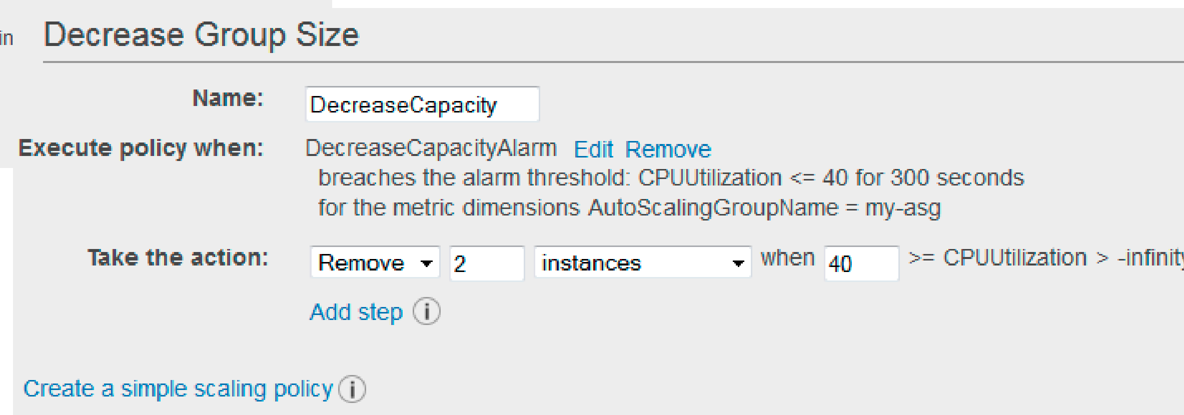
XML

```xml
<rules
  xmlns=http://schemas.microsoft.com/practices/2011/entlib/autoscaling/rules
  enabled="true">
  <constraintRules>
    <rule name="Default" description="Always active"
          enabled="true" rank="1">
      <actions>
        <range min="2" max="5" target="RoleA"/>
      </actions>
    </rule>

    <rule name="Peak" description="Active at peak times"
          enabled="true" rank="100">
      <actions>
        <range min="4" max="6" target="RoleA"/>
      </actions>
      <timetable startTime="08:00:00" duration="02:00:00">
        <daily/>
      </timetable>
    </rule>
  </constraintRules>

  <reactiveRules>
    <rule name="ScaleUp" description="Increases instance count"
          enabled="true" rank="10">
      <when>
        <greater operand="Avg_CPU_RoleA" than="80"/>
      </when>
      <actions>
        <scale target="RoleA" by="1"/>
      </actions>
    </rule>
    <rule name="ScaleDown" description="Decreases instance count"
          enabled="true" rank="10">
      <when>
        <less operand="Avg_CPU_RoleA" than="20"/>
      </when>
      <actions>
        <scale target="RoleA" by="-1"/>
      </actions>
    </rule>
  </reactiveRules>

  <operands>
    <performanceCounter alias="Avg_CPU_RoleA"
      performanceCounterName="\Processor(_Total)\% Processor Time"
      aggregate="Average" source="RoleA" timespan="00:45:00"/>
  </operands>
</rules>
```

Auto-scaling Examples from Amazon services



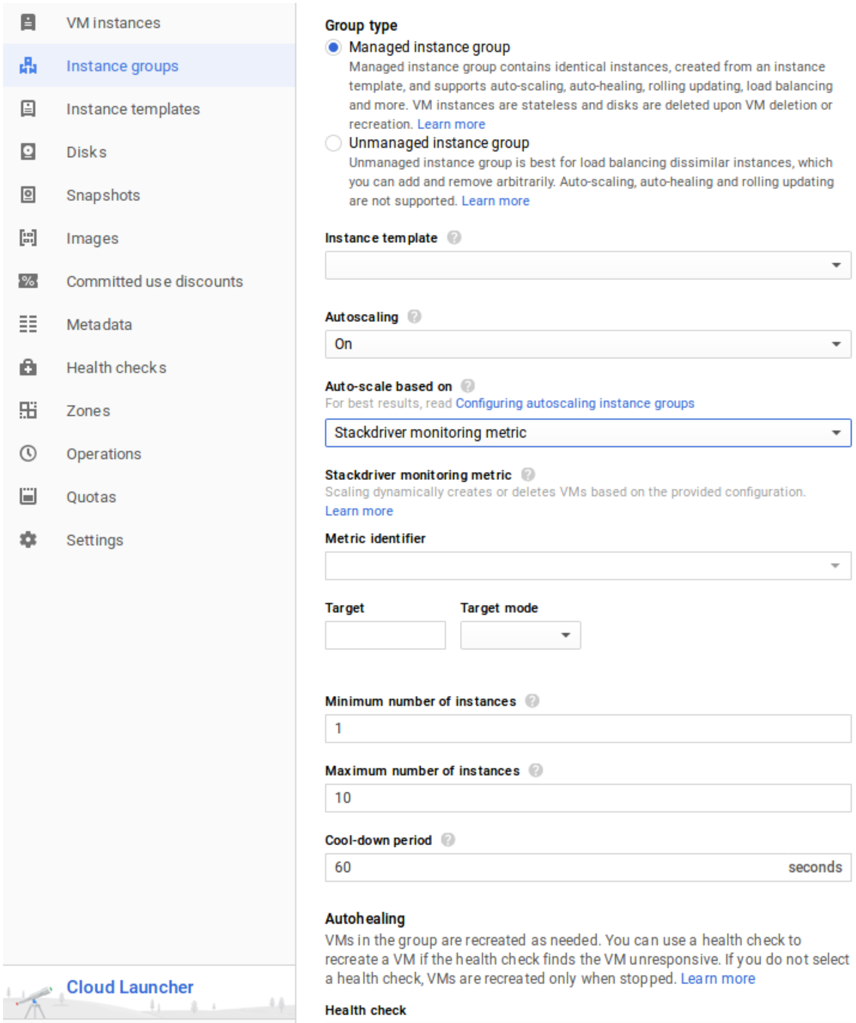





```
aws autoscaling attach-load-balancers --auto-scaling-group-name my-asg --load-balancer-names my-lb
```

Sources: http://docs.aws.amazon.com/autoscaling/latest/userguide/policy_creating.html
http://docs.aws.amazon.com/autoscaling/latest/userguide/attach-load-balancer-asg.html

Google Cloud



Understand metrics and rules for elasticity

Table 1 Summary of the reviewed literature about threshold-based rules

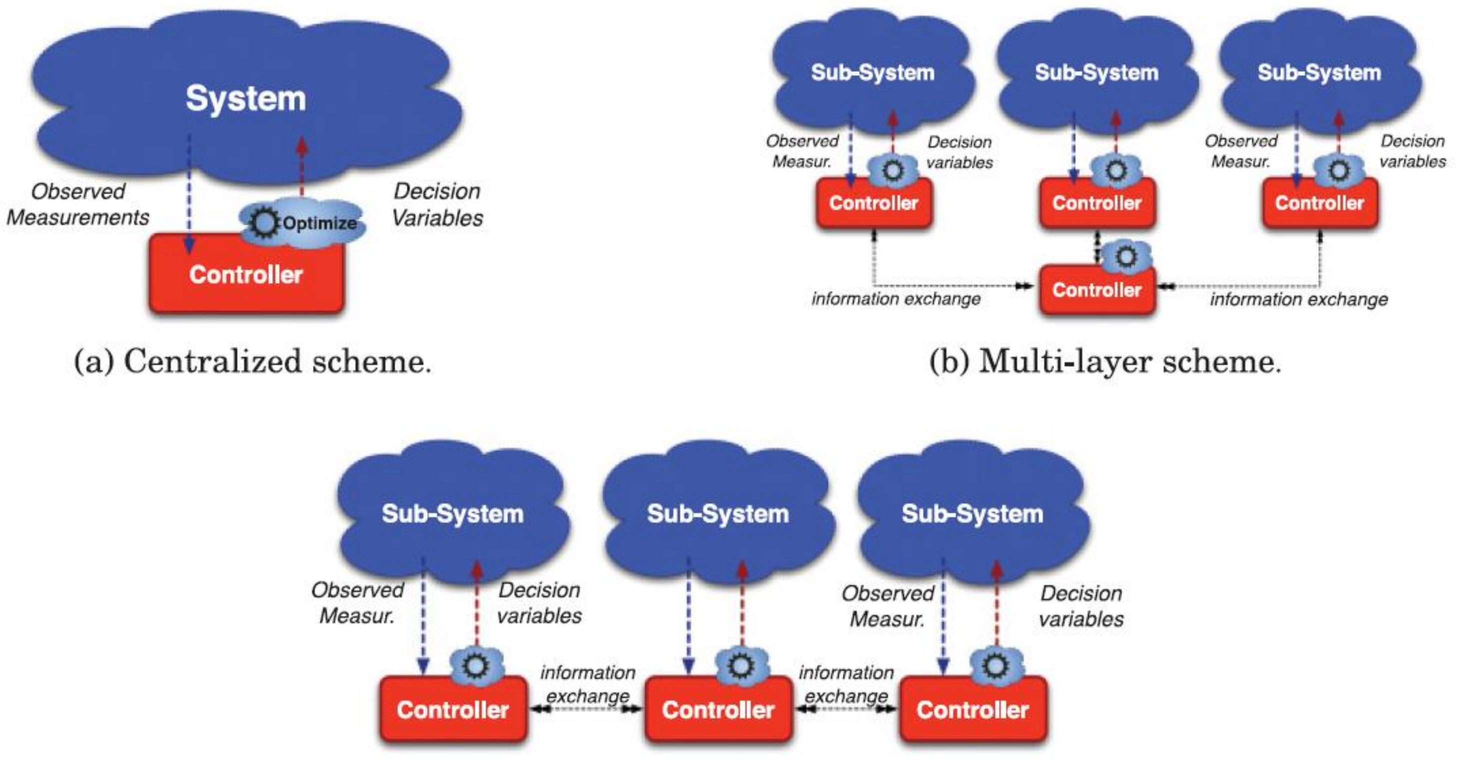
Ref	Auto-scaling Techniques	H/V	R/P	Metric	Monitoring	SLA	Workloads	Experimental Platform
[63]	Rules	Both	R	CPU, memory, I/O	Custom tool. 1 minute	Response time	Synthetic. Browsing and ordering behavior of customers.	Custom testbed (called IC Cloud) + TPC
[72]	Rules	H	R	Average waiting time in queue, CPU load	Custom tool.	–	Synthetic	Public cloud. FutureGrid, Eucalyptus India cluster
[64]	Rules	Both	R	CPU load, response time, network link load, jitter and delay.	–	–	Only algorithm is described, no experimentation is carried out.	
[48]	Rules + QT	H	P	Request rate	Amazon Cloud-Watch. 1–5 minutes	Response time	Real. Wikipedia traces	Real provider. Amazon EC2 + Httperf + MediaWiki
[52]	RightScale + MA to performance metric	H	R	Number of active sessions	Custom tool	–	Synthetic. Different number of HTTP clients	Custom testbed. Xen + custom collaborative web application
[73]	RightScale + TS: LR and AR(1)	H	R/P	Request rate, CPU load	Simulated.	–	Synthetic. Three traffic patterns: weekly oscillation, large spike and random	Custom simulator, tuned after some real experiments.
[59]	RightScale	H	R	CPU load	Amazon CloudWatch	–	Real. World Cup 98	Real provider. Amazon EC2 + RightScale (PaaS) + a simple web application
[96]	RightScale + Strategy-tree	H	R	Number of sessions, CPU idle	Custom tool. 4 minutes.	–	Real. World Cup 98	Real provider. Amazon EC2 + RightScale (PaaS) + a simple web application.
[81]	Rules	V	R	CPU load, memory, bandwidth, storage	Simulated.	–	Synthetic	Custom simulator, plus Java rule engine Drools
[77]	Rules	V	R	CPU load	Simulated. 1 minute	Response time	Real. ClarkNet	Custom simulator

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

Source: A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments, Tania Lorido-Botran , Jose Miguel-Alonso, Jose A. Lozano, http://link.springer.com/article/10.1007%2Fs10723-014-9314-7

Types of controls in distributed systems

Which models are for elasticity controls?



(a) Centralized scheme.

(b) Multi-layer scheme.

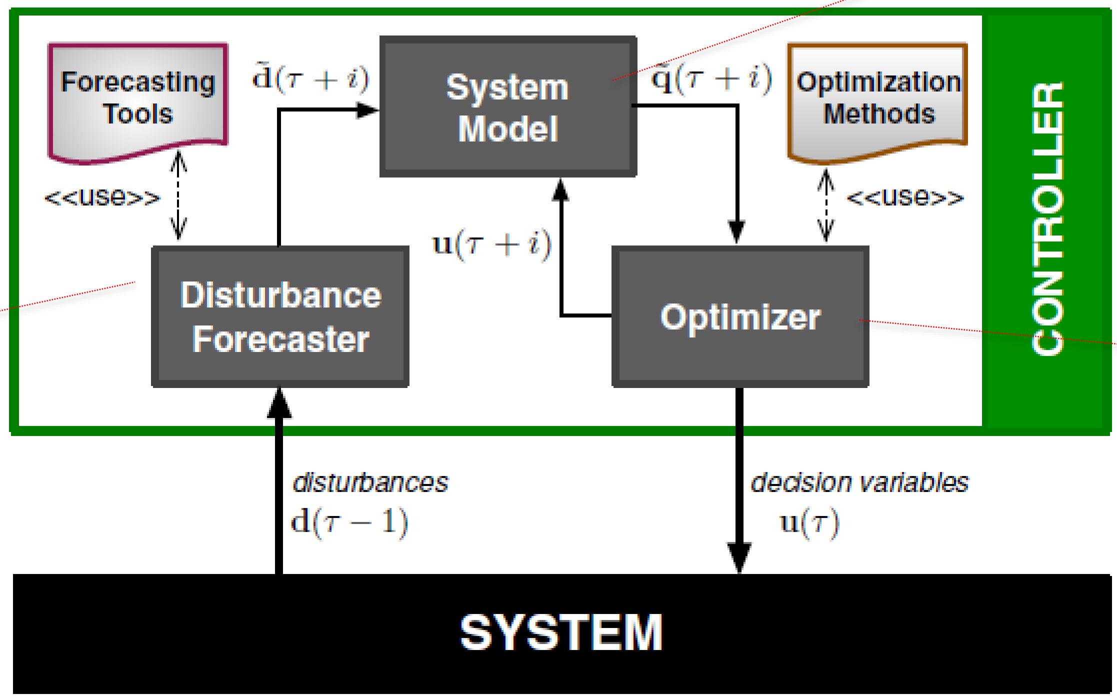
(c) Single-layer scheme.

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

Configurations & Metrics relationships

Arrival rate, processing time, network throughput, etc.



Control/reconfiguration actions

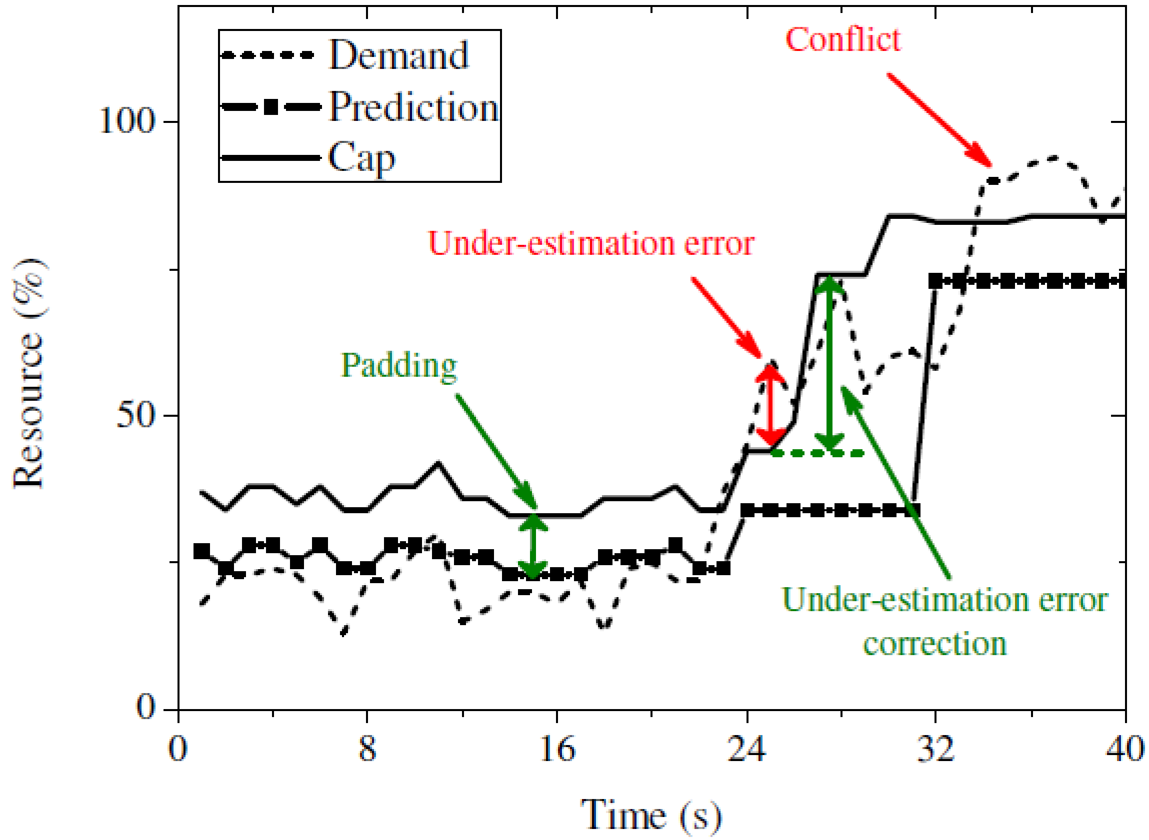
Figure source: Tiziano De Matteis and Gabriele Mencagli. 2017. Proactive elasticity and energy awareness in data stream processing. J. Syst. Softw. 127, C (May 2017), 302-319. DOI: https://doi.org/10.1016/j.jss.2016.08.037

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



Source: Zhiming Shen, Sethuraman Subbiah, Xiaohui Gu, and John Wilkes. 2011. *CloudScale: elastic resource scaling for multi-tenant cloud systems.* In Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC '11). ACM, New York, NY, USA, Article 5, 14 pages. DOI: https://doi.org/10.1145/2038916.2038921

Elasticity for Compute Resources

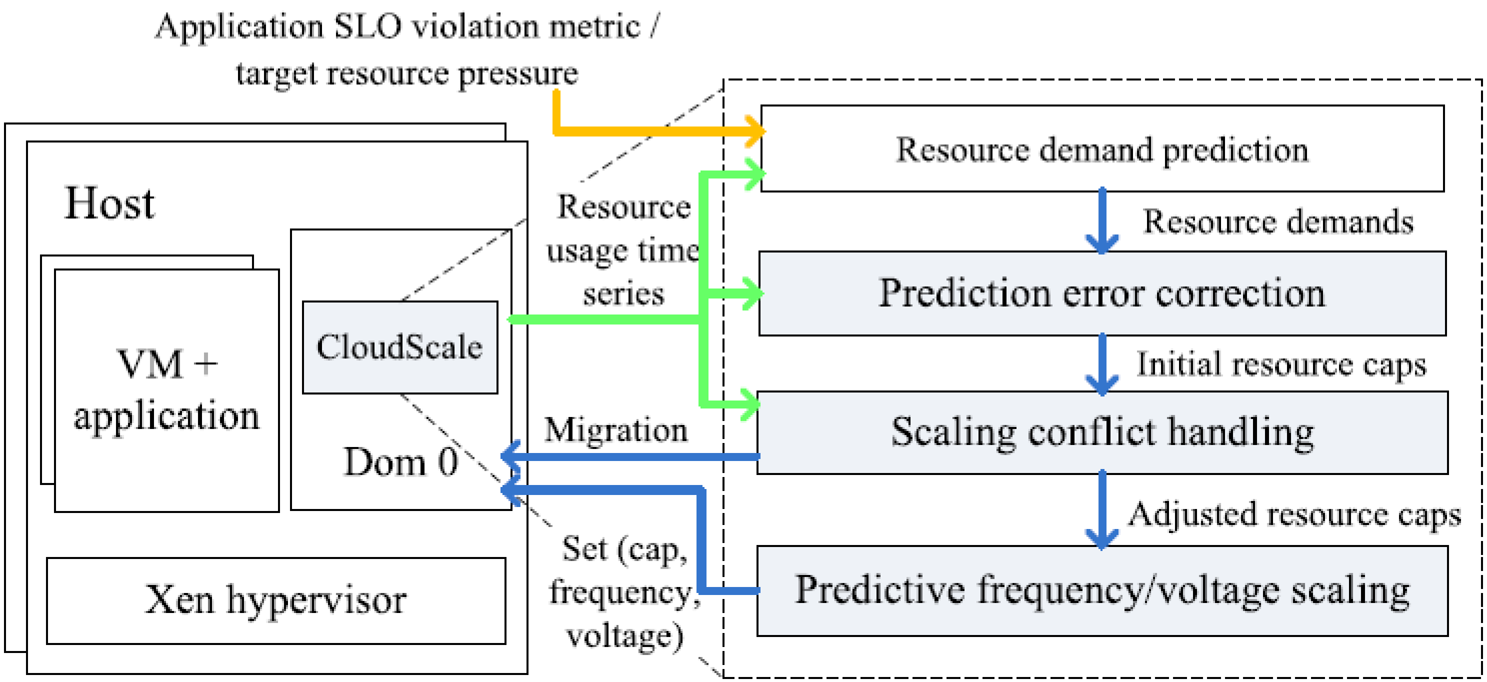
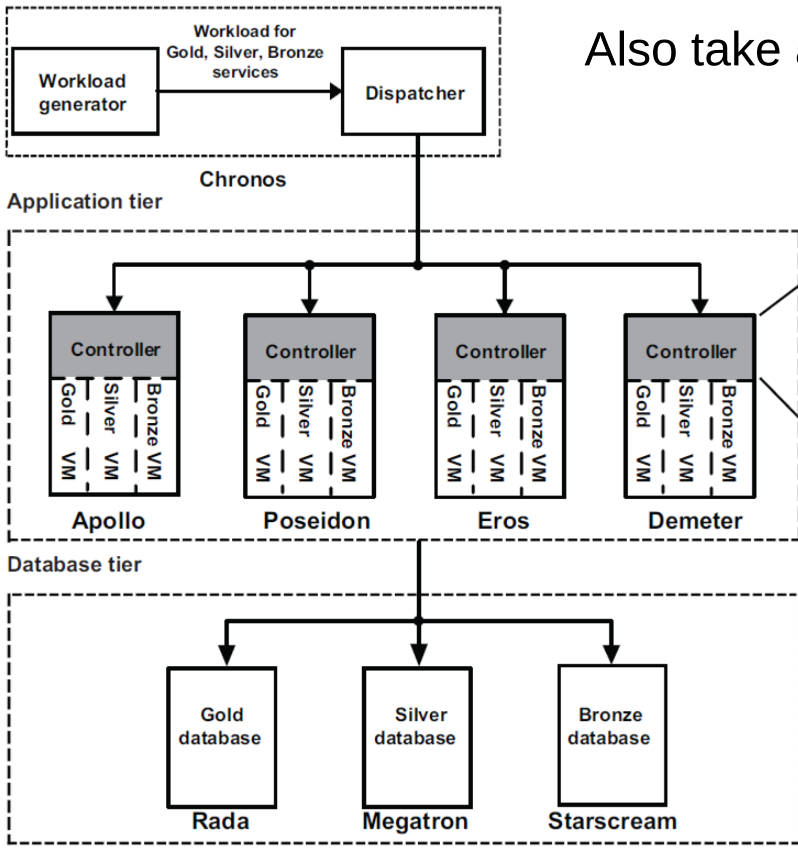


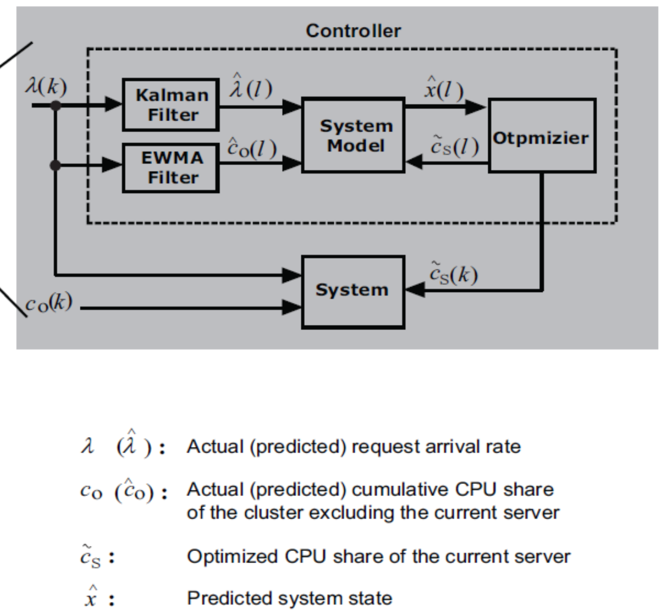
Figure 2: The CloudScale system achitecture.

Source: Zhiming Shen, Sethuraman Subbiah, Xiaohui Gu, and John Wilkes. 2011. *CloudScale: elastic resource scaling for multi-tenant cloud systems.* In Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC '11). ACM, New York, NY, USA, Article 5, 14 pages. DOI: https://doi.org/10.1145/2038916.2038921

Elasticity from computing resources

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

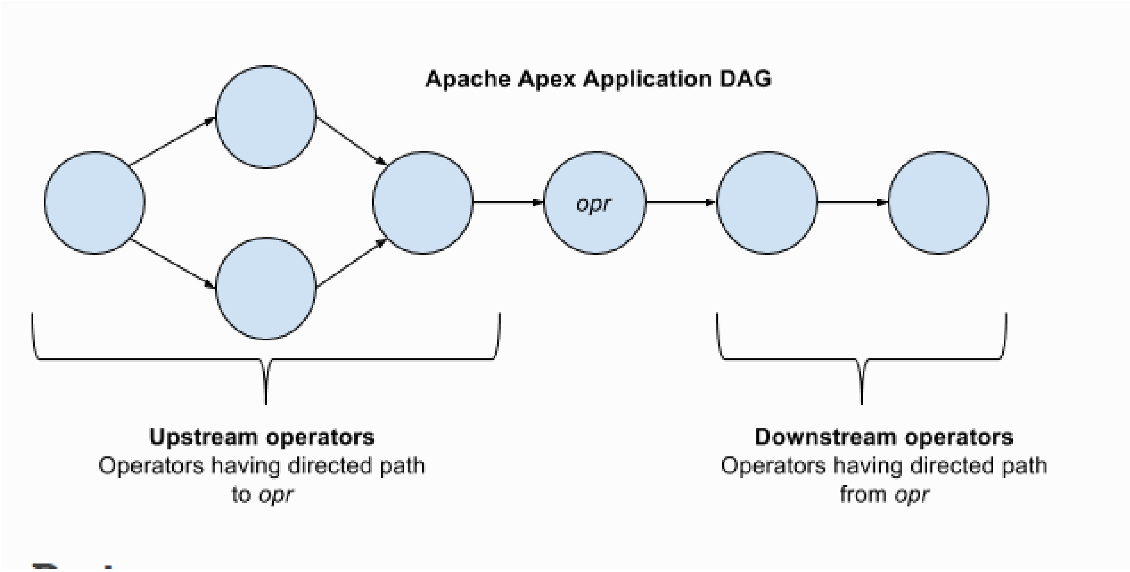




Source: Rui Wang, Dara Marie Kusic, and Nagarajan Kandasamy. 2010. *A distributed control framework for performance management of virtualized computing environments*. In Proceedings of the 7th international conference on Autonomic computing (ICAC '10). ACM, New York, NY, USA, 89-98. DOI=http://dx.doi.org/10.1145/1809049.1809066

Elasticity in streaming data processing

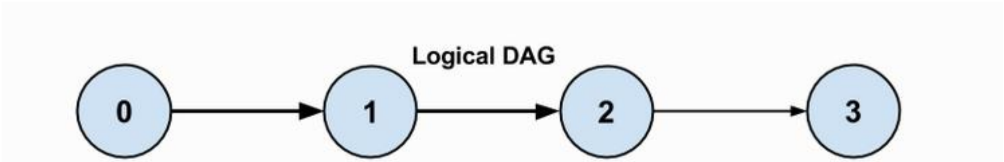
- Streaming data processing
  - What are key constructs and operators?



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

Elasticity: When, where, what, how?

Example in Apache Apex

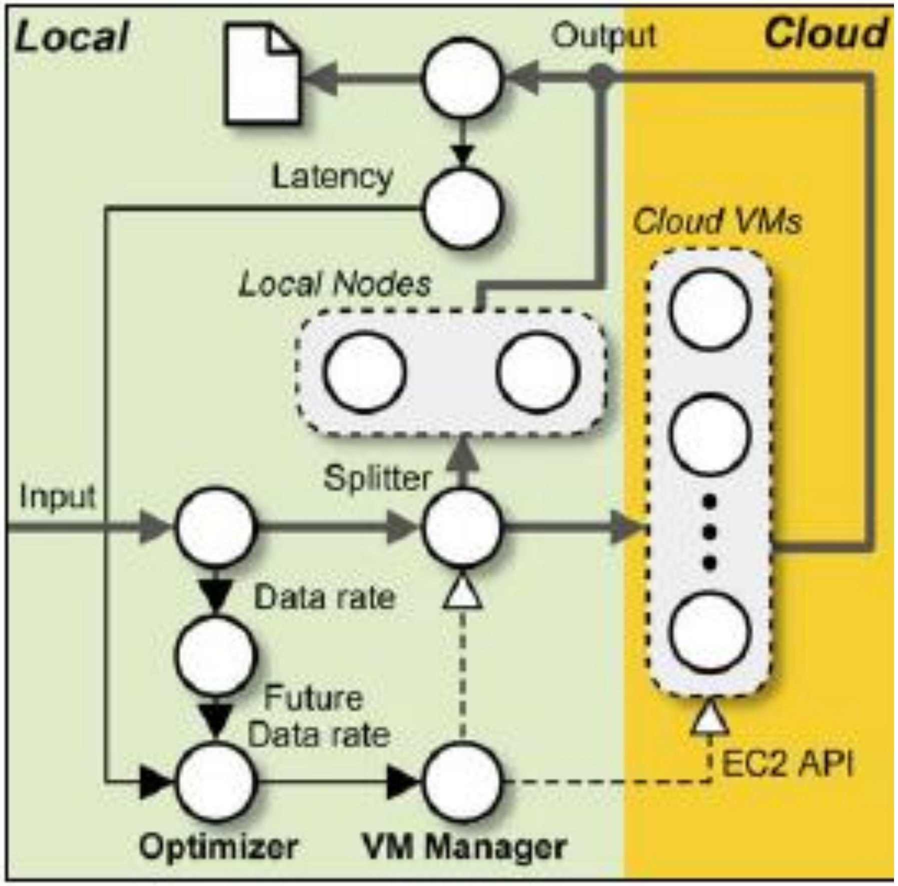


- Dynamic Partition
  - Partition operators
  - Dynamic: specifying when a partition should be done
  - Unifiers for combining results (reduce)
- StreamCodec
  - For deciding which tuples go to which partitions
  - Using hashcode and masking mechanism

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

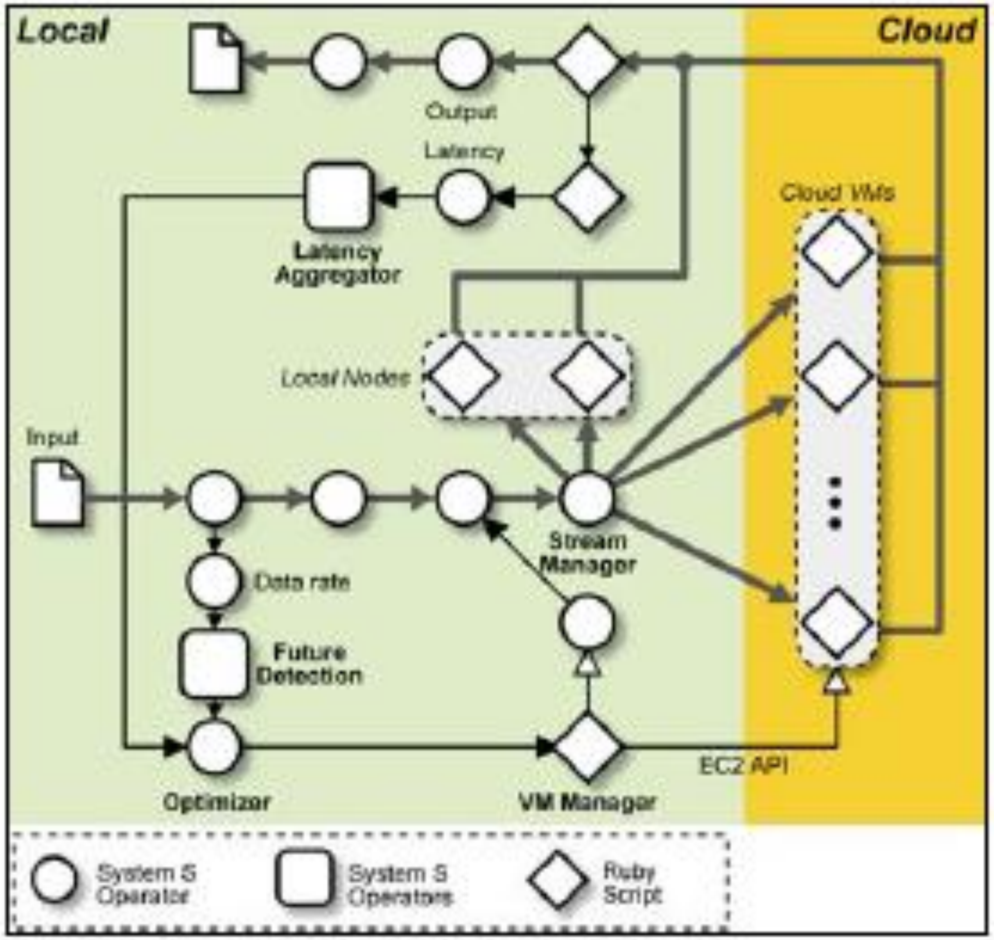
Example with ElasticStream

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



Source: A. Ishii and T. Suzumura, "Elastic Stream Computing with Clouds," 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, 2011, pp. 195-202. doi: 10.1109/CLOUD.2011.11

ElasticStream Solution



Source: A. Ishii and T. Suzumura, "Elastic Stream Computing with Clouds," 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, 2011, pp. 195-202. doi: 10.1109/CLOUD.2011.11

Other works

- Bugra Gedik, Scott Schneider, Martin Hirzel, and Kun-Lung Wu. 2014. Elastic Scaling for Data Stream Processing. IEEE Trans. Parallel Distrib. Syst. 25, 6 (June 2014), 1447-1463. DOI: http://dx.doi.org/10.1109/TPDS.2013.295

- Vincenzo Gulisano, Ricardo Jimenez-Peris, Marta Patino-Martinez, Claudio Soriente, and Patrick Valduriez. 2012. StreamCloud: An Elastic and Scalable Data Streaming System. IEEE Trans. Parallel Distrib. Syst. 23, 12 (December 2012), 2351-2365. DOI=http://dx.doi.org/10.1109/TPDS.2012.24

Example in database as a service

Elasticity: When, where, what and how?

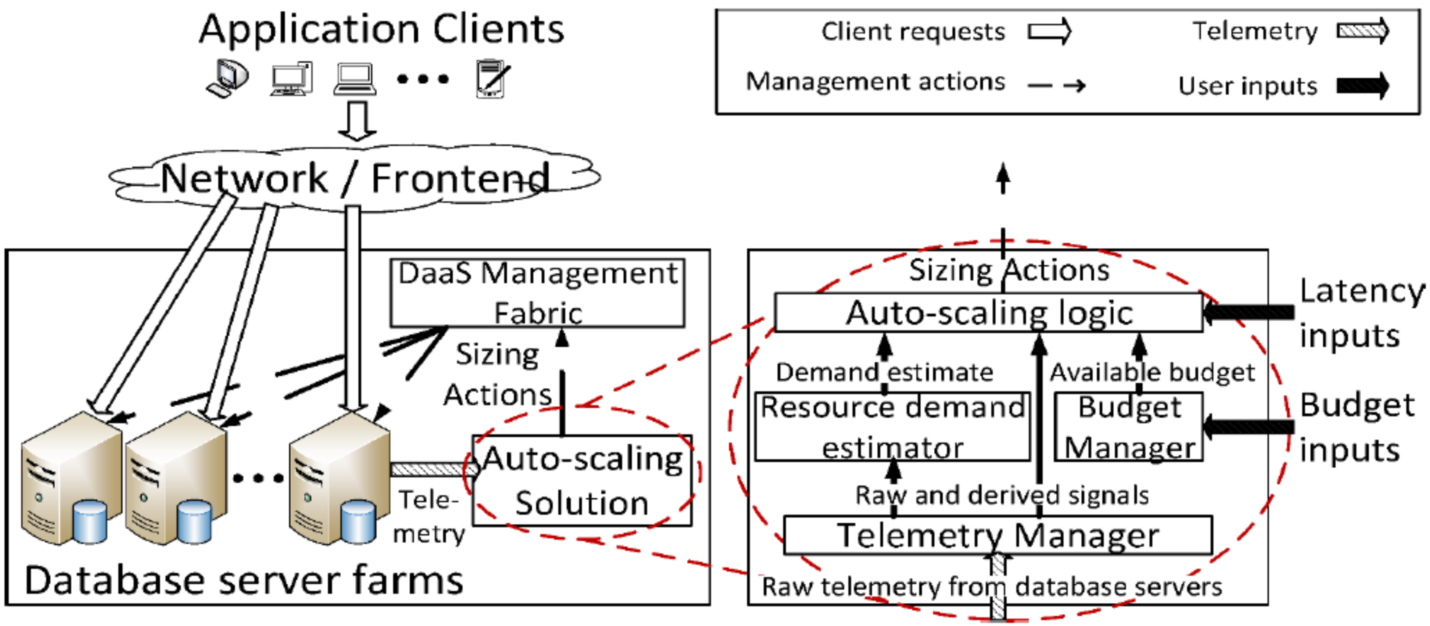
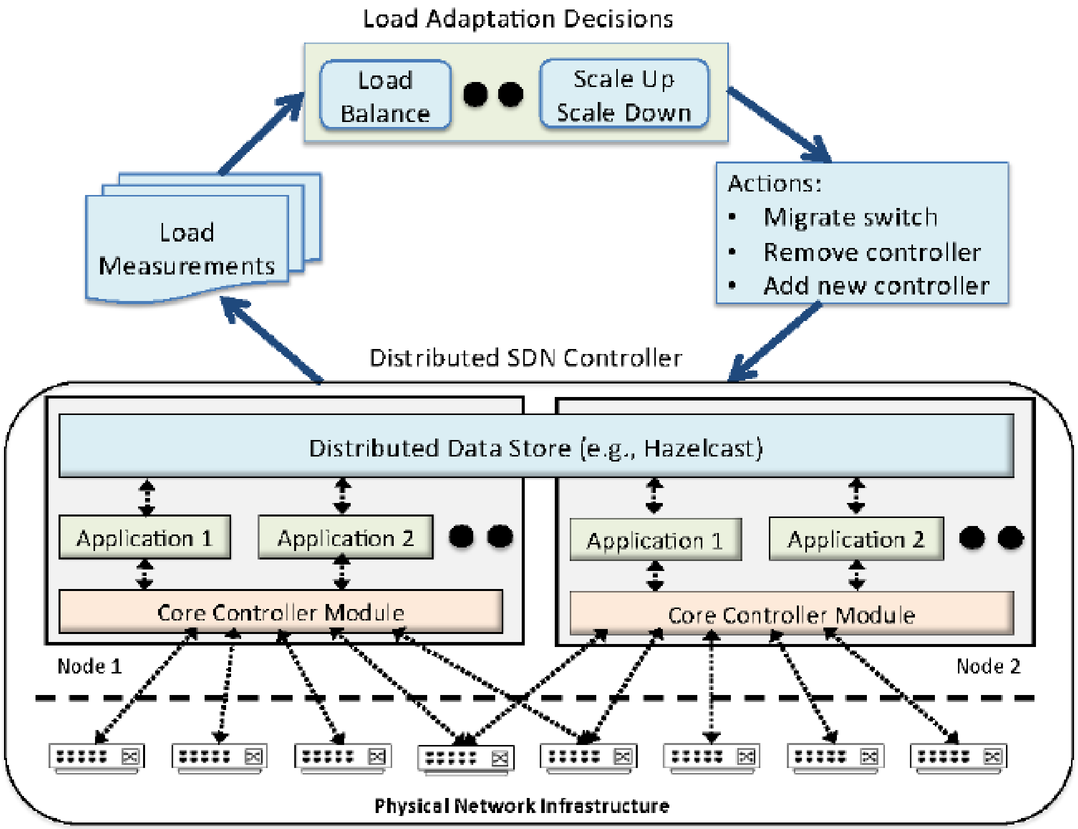


Figure source: Sudipto Das, Feng Li, Vivek R. Narasayya, and Arnd Christian König. 2016. *Automated Demand-driven Resource Scaling in Relational Database-as-a-Service*. In Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16). ACM, New York, NY, USA, 1923-1934. DOI: https://doi.org/10.1145/2882903.2903733

Also read: Harold C. Lim, Shivnath Babu, and Jeffrey S. Chase. 2010. Automated control for elastic storage. In Proceedings of the 7th international conference on Autonomic computing (ICAC '10). ACM, New York, NY, USA, 1-10. DOI=http://dx.doi.org/10.1145/1809049.1809051

Example in network layers

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



Source: Advait Dixit, Fang Hao, Sarit Mukherjee, T.V. Lakshman, and Ramana Kompella. 2013. *Towards an elastic distributed SDN controller.* SIGCOMM Comput. Commun. Rev. 43, 4 (August 2013), 7-12. DOI: http://dx.doi.org/10.1145/2534169.2491193

Distributed Coordination

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

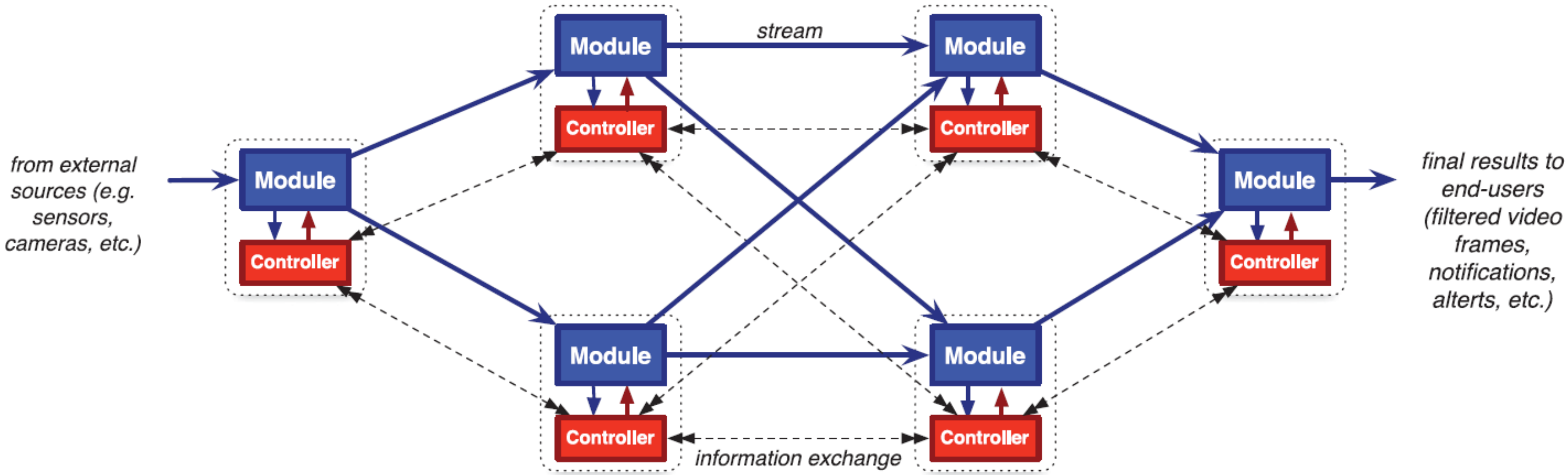


Figure Source: Gabriele Mencagli. 2016. *A Game-Theoretic Approach for Elastic Distributed Data Stream Processing*. ACM Trans. Auton. Adapt. Syst. 11, 2, Article 13 (June 2016), 34 pages. DOI: https://doi.org/10.1145/2903146

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