Towards the Realization of Multi-dimensional Elasticity for Distributed Cloud Systems

Hong-Linh Truong, Schahram Dustdar, Frank Leymann

Distributed Systems Group, TU Wien
&
IAAS, University of Stuttgart

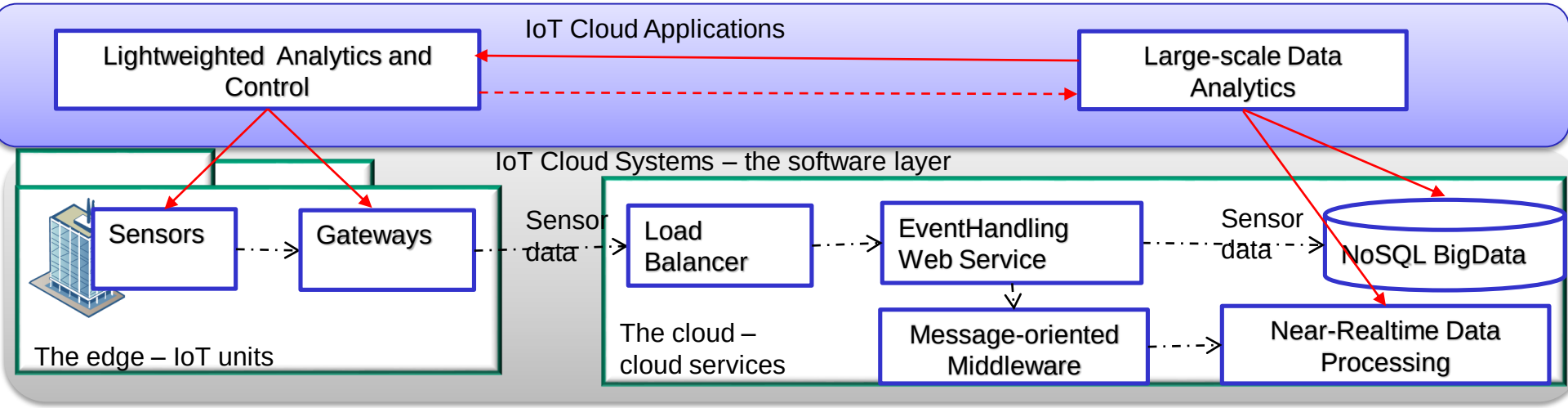
truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/staff/truong

Outline

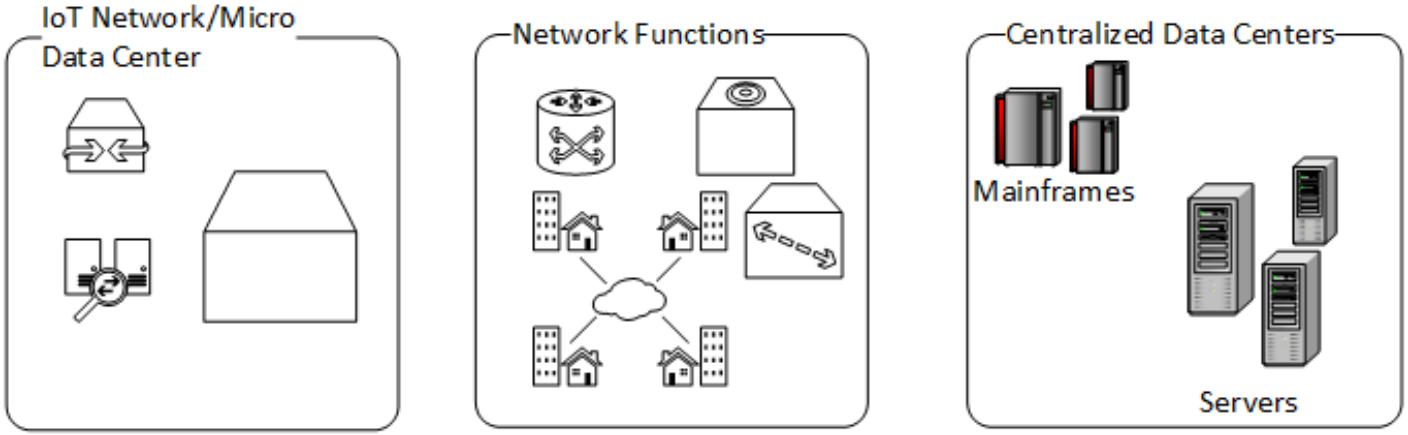
- Toward “Complete Cloud Computing”
- Multi-dimensional elasticity – key concepts
- Current effort in some EU projects
- Realizing multi-dimensional elasticity
- Conclusions and future work

Toward „Complete Computing“

Application example

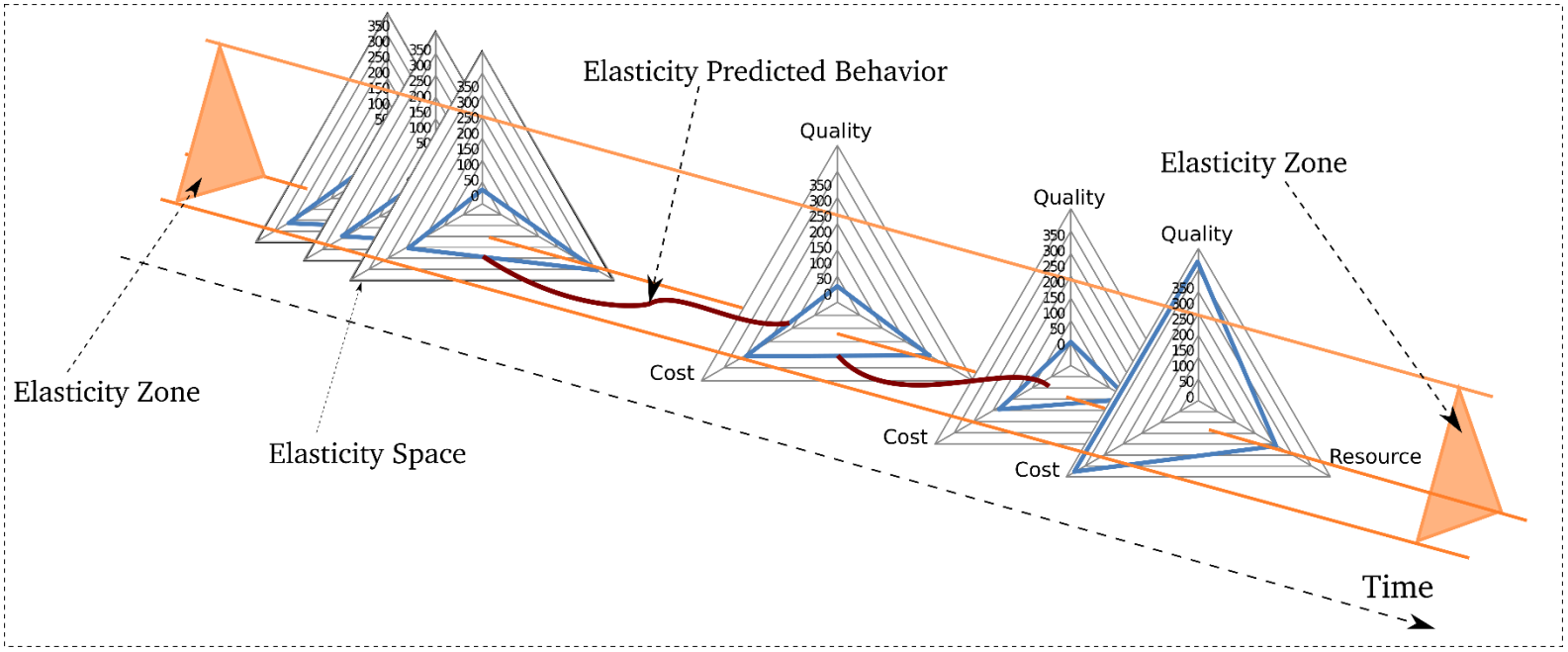


„Complete cloud system example“



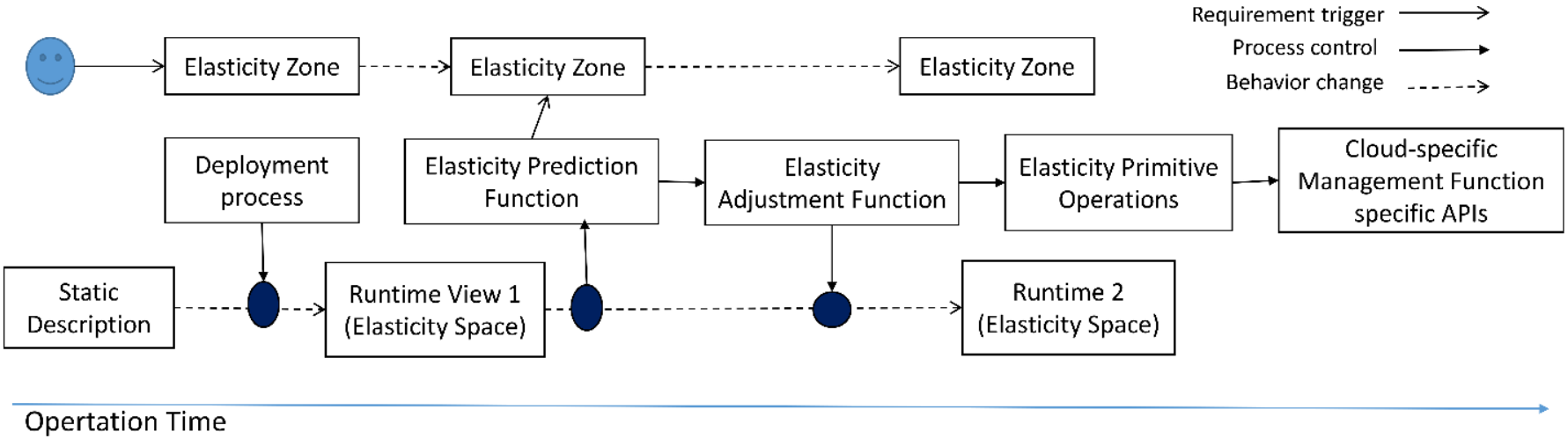
What is multi-dimensional elasticity? Key concepts

Not just auto-scaling computing resources (VMs or containers)!



Why are they important?

„High-level but complete view“



- Enable formal models, methods and tools for elasticity management, coordination and interoperability
  - Few works in multi-cloud environments
  - So far we have not entered into the edge and IoT systems

Current effort in some (finished) EU projects

Project	Multi-cloud elasticity	Edge and Cloud Elasticity	Elasticity Zone	Elasticity Space	Elasticity Prediction	Elasticity Adjustment
CELAR	Partially	Partially	Partially	Partially	No	Yes
HARNESS	No	No	Partially	No	Partially	Yes
MODAClouds	Yes	No	Partially		Partially	Yes
PaaSage	Yes	No	Partially		No	Yes

Elasticity constraints surveyed by others

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

Realizing Multi-dimensional Elasticity – Elasticity zones

Limitations

- Lack a “formal” model/specification
- Using rules fails to capture dynamics (e.g., time)
- No edge resources
- Elasticity Zones mainly defined by humans

Approach:

- Formal model: n-dimensional manifold
- More than just computing resources and prices
  - Data sources, quality of data, uncertainties
- Dynamic Elasticity Zones: event-dependent
- Triggers: Human-in-the-loop, prediction functions, pre-defined events, etc.

Realizing Multi-dimensional Elasticity – Elasticity space

Limitations

- Do not support changes of space dimensions
  - Not enough monitoring data or too much data
- Lack of correlation among various layers: e.g., either applications or VM/containers
- Lack monitoring data from edge systems and NFV

Approach:

- Algorithmic models for Elasticity Space functions
  - Generic functions for determining spaces (start, stop and why) and for different layers/topologies
- Operators on Elasticity Space, e.g., merging spaces for a composite topology of components

Realizing Multi-dimensional Elasticity – Analysis and prediction

Limitations

- Quite traditional performance analysis (for the cloud)
  - E.g., a lot of uncertainties and data quality metrics have not been considered
- Prediction is mainly for single dimensions

Approach:

- Elasticity dependency analysis
- Algorithms for elasticity prediction
- Understand which part of code cannot be elastilized and when the system might be “plastic”
- Prediction for common cloud patterns/basic building blocks

Realizing Multi-dimensional Elasticity – Patterns

Limitations

- Mainly on scalability best practices and patterns
- No real elasticity patterns
- Lack of unified way to define and model “management function” (enabling elasticity)

Approach:

- Elasticity primitives: common notations for abstracting low-level APIs for elasticity management
- Common models for elasticity primitives for network functions and IoT/edge systems
- New primitives for data-aware elasticity
- Deduce elasticity patterns

Realizing Multi-dimensional Elasticity – Adjustment functions

Limitations

- Several existing functions but no theoretical models to combine them to support multi-dimensional elasticity adjustment
- Focused mainly on centralized clouds

Approach:

- Deal with different level of abstractions and coordinated adjustment
- Fundamental steps
  - Elasticity pattern selection
  - Primitive operations selection
  - Selecting/generating/configuring elasticity adjustment functions
- Elasticity operation management (e.g. incident management)

How far we are?

- Elasticity for IoT Cloud systems
  - http://tuwiendsg.github.io/iCOMOT/
- Elasticity partially considers uncertainties
  - CloudCom 2015
- Data elasticity
  - ICSOC 2015
- Coordination-aware elasticity
  - UCC 2015 and the work in progress

Conclusions and future work

- **Multi-dimensional elasticity**
  - Key concepts for “complete computing” atop IoT, edge systems and clouds
- **Our work**
  - Analyzed current limitations of elasticity engineering in the clouds
  - Proposed key concepts and suitable approaches: Elasticity Zone, Space, Prediction and Adjustment
- **Future work**
  - Implementation is on progress for IoT, Network function virtualization, and clouds
  - Check iCOMOT http://tuwiendsg.github.io/iCOMOT/ and SINC http://sincconcept.github.io

Thanks for your attention!

Hong-Linh Truong

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at/staff/truong