Habilitationskolloquium

On Quality Issues in Complex Service-oriented Systems

Hong-Linh Truong
Distributed Systems Group, Institute of Information Systems
Vienna University of Technology

truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/Staff/truong
Outline

- Complex service-oriented systems - research challenges

- Selected works
  - Monitoring and analysis of performance metrics for workflows
  - Monitoring and analysis of data concerns and data contracts
  - Composable cost/performance evaluation for cloud applications
  - Programming human-based services in the cloud

- Conclusions and future work
Complex Service-oriented Systems

The complexity and diversity of computational, data and human resources as services, and rich service provisioning and utilization models lead to complex quality issues.
Example of complex service-oriented systems

- Multiple types of resources
- Diverse types of interactions and roles
- Multiple perspectives in quality monitoring, analysis and utilization
Multiple types of resources

Habilitationskolloquium, 17 April. 2013
Multiple types of quality metrics

- N separated workflows for M consumers: common activities but different influence factors/expected quality

Output 1: results, accuracy, performance etc.

Output n: results, accuracy, performance etc.
Multiple perspectives of elasticity in computing

1. Elastic demands from consumers
2. Output elasticity with different price and quality
3. Elastic data inputs, e.g., deal with opportunistic data
4. Elastic pricing and quality models associated resources
Multiple elasticity dimensions

Resource elasticity:
software/human-based computing elements, multiple clouds

Quality elasticity:
performance, quality of data, service availability, human trust, ...

Cost/benefit elasticity:
Pricing/Rewarding/Incentive models

- Scaling software, services, and people in the same system
- But elasticity is not just “resource elasticity”: cost/benefit and quality are also important

The Vienna Elastic Computing Model

- Multi-dimensional elasticity
- Service computing models, and
- Cloud provisioning models


Habilitationskolloquium, 17 April. 2013
Focused research challenges

quality issues in elastic complex service-oriented systems

- How to characterize and evaluate
  - performance metrics, human interactions, quality of data associated with software, humans and data in complex systems
  - Quality of service, data concerns, and service contracts associated with different types of services
- What are the interdependencies among different types of quality metrics associated with different types of resources?
- How to utilize these metrics for service engineering and optimization to support elasticity?
Contribution area: monitoring and analysis of Grid workflow performance
Characterizing and evaluating quality for Grid workflows

Monitoring and analysis at which levels of abstraction?

Which performance metrics should be analyzed?

How to measure, analyze and provide analysis results?

How to integrate with different consumers?
Contribution: performance metrics for Grid workflows

Abstracting workflows for quality monitoring, measurement and evaluation

Characterizing performance metrics

Multiple types of consumers

Measurement and analysis

Habilitationskolloquium, 17 April. 2013
Contribution: performance metrics for Grid workflows

```java
mProject1Service.java
public void mProject1() {
    A();
    while () {
        ...
    }
}
```

Examples of metrics

Table 2
Performance metrics at code region level

<table>
<thead>
<tr>
<th>Category</th>
<th>Metric name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4
Performance metrics at activity level

<table>
<thead>
<tr>
<th>Category</th>
<th>Metric name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Habilitationskolloquium, 15
17 April. 2013
Contribution: distributed and online performance monitoring and analysis

Performance overhead analysis and search for performance problems

PDQS, WIRL, and SIRWF for instrumentation and monitoring

PDQS and WARL for performance service integration

Online workflow tracing
Observing performance metrics and problems
Summary of contributions

Contributions

- A comprehensive study of performance metrics associated with multiple levels of Grid workflow abstraction
- Novel techniques and a framework for online and distributed monitoring and analysis of Grid workflows
- Online demos/movies and software: http://www.dps.uibk.ac.at/projects/kwfgrid

Related work

- Most tools focus on activity levels and analysis at the workflow engine level, lack of comprehensive studies of performance metrics and target mainly to end users

Selected publications

- Michael Reiter, Uwe Breitenbicher, Schahram Dustdar, Dimka Karastoyanova, Frank Leymann, Hong-Linh Truong: A Novel Framework for Monitoring and Analyzing Quality of Data in Simulation Workflows. eScience 2011: 105-112
Contribution area: monitoring and analysis of data concerns and data/service contracts
Quality issues in data-as-a-service

- Multiple stakeholders, the rise of data provisioning, cloud models: → data-as-a-service
- Provide data capabilities rather than provide computation or other software capabilities
DaaS design & implementation – not just „functional“ aspects

Data Assessment /Improvement

Profiling
Cleansing
Enrichment
Integration

Data concerns

Quality of data?
Privacy problem?
Located in US?
Service quality?
price?
redistribution?

APIs, Querying, Data Management, etc.

DaaS

data assets

data

Contribution: analysis and conceptual model for DaaS concerns

Hong Linh Truong, Schahram Dustdar On analyzing and specifying concerns for data as a service. APSCC 2009: 87-94

Habilitationskolloquium, 17 April, 2013
## Contribution: techniques for evaluating data concerns in DaaS (2)

### Evaluation Scope
- **At which level the evaluation is performed?**
  - Enable fine-grained evaluation with three scopes: data resources, DaaS operations, and DaaS

### Evaluation Modes
- **When the evaluation is done?**
  - Offline and online are suitable for different types of data

### Integration Model
- **How the evaluation tool is invoked?**
  - Push and pull data concerns Pass-by-value versus pass-by-reference data

---

Hong Linh Truong, Schahram Dustdar: On Evaluating and Publishing Data Concerns for Data as a Service. APSCC 2010: 363-370

Habilitationskolloquium, 17 April. 2013
Contribution: modeling complex properties of data services utilized

A description model for DaaS

Quang Hieu Vu, Tran Vu Pham, Hong Linh Truong, Schahram Dustdar, Rasool Asal: DEMODS: A Description Model for Data-as-a-Service. AINA 2012: 605-612

Habilitationskolloquium, 17 April. 2013
Contribution: data contracts

Data contract development framework

Data contract specifications and evaluation algorithms


Habilitationskolloquium, 17 April. 2013
Contribution: reconciliation of service/data contracts

Composite service-based applications

Multiple concerns at multiple levels

Multiple language specifications

WSLA
WSOL
ODRL-S

Habilitationskolloquium, 17 April. 2013


Summary of contributions

- Contributions:
  - A data concern specification for DaaS, a framework for data concern evaluation and publishing, a data contract model, algorithms for data/service contract compatibilities, a description model for DaaS
  - Main contributions for the WWTF SODI project
  - Prototypes: [http://www.infosys.tuwien.ac.at/prototyp/SOD1](http://www.infosys.tuwien.ac.at/prototyp/SOD1)

- Related work
  - Service communities focus on QoS, database communities focus on the data level

- Selected publications:
  - Hong-Linh Truong, Schahram Dustdar: On analyzing and specifying concerns for data as a service. APSCC 2009:87-94
  - Hong-Linh Truong, Schahram Dustdar: On Evaluating and Publishing Data Concerns for Data as a Service. APSCC 2010:363-370
  - Hong-Linh Truong, Marco Comerio, Andrea Maurino, Schahram Dustdar, Flavio De Paoli, Luca Panziera: On Identifying and Reducing Irrelevant Information in Service Composition and Execution. WISE 2010: 52-66
  - Quang Hieu Vu, Tran Vu Pham, Hong-Linh Truong, Schahram Dustdar, Rasool Asal, DEMODS: A Description Model for Data-as-a-Service, (c)IEEE Computer Society, The 26th IEEE International Conference on Advanced Information Networking and Applications (AINA-2012), Fukuoka, Japan, March 26-29, 2012
  - Schahram Dustdar, Reinhard Pichler, Vadim Savenkov, Hong-Linh Truong, "Quality-aware Service-Oriented Data Integration: Requirements, State of the Art and Open Challenges", SIGMOD Record, Vol. 41, Number 1, March 2012
  - Hong-Linh Truong, G.R. Gangadharan, Marco Comerio, Vincenzo D' Andrea, Flavio De Paoli and Schahram Dustdar, "Reconciliation of Contractual Concerns of Web Services." In Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions
Contribution area: composable quality evaluation for cloud applications
Performance together other issues for complex service systems

Complex composable applications/workflows

Which models should be used?

Is the data good enough to be stored and shared?

Should the data be in the cloud for improving performance?

Data in

Executed on

Data out

Where should I run the code?
What about cost and performance?
Dealing with performance and cost of complex applications in clouds

- **Application complexity**
  - Elastic high performance applications on multiple clouds: libraries, software services, virtual machines, etc.
  - Cost and performance are needed for determining which parts of the application should be executed in the clouds and when

- **Cost/performance model complexity**
  - Coarse- and fine-grained cost models of clouds at different layers:
    - Too coarse-grained (networks, storages, machines) or too fine-grained (IO calls)
    - Software-, data-, human-specific cost/performance models
    - Cost models for individual parts (workflow, MPI, OpenMP, etc.)


Contribution: composable cost evaluation

Elastic high performance applications on multiple clouds: libraries, software services, virtual machines, etc.

Utilize different performance and dependencies models for sequential, parallel, workflows, etc.

Runtime: Elastic processes

Cost/performance model i

Cost/performance model j

Cost/performance model k
Examples of fine-grained composable cost models

<table>
<thead>
<tr>
<th>Model</th>
<th>Activities</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{ds}$</td>
<td>Data storage</td>
<td>$\text{size(total)} \times t_{\text{sub}} \times \text{cost(storage)}$ where $t_{\text{sub}}$ is the subscription time</td>
</tr>
<tr>
<td>$M_{cm}$</td>
<td>Computational machine</td>
<td>$\text{cost(machine)}$</td>
</tr>
<tr>
<td>$M_{dfi}$</td>
<td>Data transfer into the cloud</td>
<td>$\text{cost(transfer}_{\text{in}})$</td>
</tr>
<tr>
<td>$M_{dio}$</td>
<td>Data transfer out to the cloud</td>
<td>$\text{cost(transfer}_{\text{out}})$</td>
</tr>
<tr>
<td>$M_{sid}$</td>
<td>Single data transfer without the cost for machines performing the transfer</td>
<td>$\text{size(in)} \times M_{dfi} + \text{size(out)} \times M_{dio}$</td>
</tr>
<tr>
<td>$M_{sim}$</td>
<td>Sequential/multi-threaded program or single data transfer with the cost for machines performing the transfer (cost monitoring)</td>
<td>$t_c \times M_{cm} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dfi}$</td>
</tr>
<tr>
<td>$M_{sei}$</td>
<td>Sequential or multi-threaded program (cost estimation)</td>
<td>$f_{pi} \times M_{cm} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dfi}$ where $f_{pi}$ is an estimated performance improvement function when $n$ expected threads to be used. $f_{pi}$ can be provided by performance prediction tools or scientists. In our case, currently, we use an ideal parallel performance improvement $f_{pi} = \frac{p}{n} \times t_c(p)$ where $p$ is the number of threads used to obtain $t_c(p)$. $p$ and $t_c(p)$ are known knowledge.</td>
</tr>
<tr>
<td>$M_{pm}$</td>
<td>Parallel/mpi programs on multiple machines (cost monitoring)</td>
<td>$n \times M_{cm} \times t_c + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dfi}$</td>
</tr>
<tr>
<td>$M_{pe}$</td>
<td>Parallel/mpi programs on multiple machines (cost estimation)</td>
<td>$n \times M_{cm} \times f_{pi} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dfi}$ where $f_{pi}$ is an estimated performance improvement function when $n$ processes are used.</td>
</tr>
<tr>
<td>$M_{wm}$</td>
<td>Workflows (cost monitoring)</td>
<td>$\sum_{i=1}^{k} (\text{size(in)}<em>i \times M</em>{dfi}) + \sum_{i=1}^{l} (\text{size(out)}<em>i \times M</em>{dio}) + \sum_{j=1}^{n} (M_{cm} \times t_c(\text{machine}_j))$</td>
</tr>
<tr>
<td>$M_{we}$</td>
<td>Workflows (cost estimation)</td>
<td>$\sum_{i=1}^{k} \text{cost}(\text{wr}_i)$. For a workflow region $\text{wr}<em>i$, $\text{cost}(\text{wr}<em>i) = \sum</em>{j=1}^{q} (\text{cost(activity)}<em>j)$ where $\text{cost(activity)}<em>j$ is determined based on $M</em>{mp}$, $M</em>{sm}$, and $M</em>{sid}$, when the activity $\text{activity}_j$ is a parallel activity, sequential activity, or a data transfer activity, respectively.</td>
</tr>
</tbody>
</table>

Habilitationskolloquium, 17 April. 2013
Online cost/performance evaluation


Application deployment

Cloud Systems
- Application Trace Extraction Tool
- Application Monitoring Sensors

Cloud Cost Estimation and Monitoring Service
- Event Matching
- Online Cost Evaluating
- Cost Estimation

Online Cost Analysis Tool
- Service Dependency Modeling Tool

Application dependency model
Examples of composable cost evaluation

Next generation sequencing analysis workflow (GSA)

Online analysis

Cloud provider bill

<table>
<thead>
<tr>
<th>Operation Name</th>
<th>Usage Type</th>
<th>Usage Value</th>
<th>Cost (EUR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RunInstances</td>
<td>EU-BoxUsage:m1.large</td>
<td>13 hours</td>
<td>4.94</td>
</tr>
<tr>
<td>RunInstances</td>
<td>EU-DataTransfer-In-Bytes</td>
<td>1,043 GB</td>
<td>0.104</td>
</tr>
<tr>
<td>RunInstances</td>
<td>EU-DataTransfer-Out-Bytes</td>
<td>4.48 GB</td>
<td>0.672</td>
</tr>
<tr>
<td>EBS:IO-Write</td>
<td>EU-EBS:VolumeIOUsage</td>
<td>2083828 I/O calls</td>
<td>0.229</td>
</tr>
</tbody>
</table>

Table 4: Operation name, usage types and usage values extracted from the Amazon billing report during the execution of the GSA workflow. We determined costs by using the following prices: 0.38 EUR per large instance (m1.large) instance-hour, 0.1 EUR per GB data transfer in, 0.150 EUR per GB data transfer out, 0.11 EUR per 1 million I/O requests.
Summary of contributions

Contributions

- A composable cost evaluation framework
- Composition techniques for elastic high performance applications
- Composable cost evaluation techniques play an important role for elasticity monitoring and controls in the ongoing FP7 CELAR (http://www.celarcloud.eu/)

Related work

- Simple cost model for particular types of clouds, focus on infrastructure not complex applications

Scientific papers

- Tran Vu Pham, Hong-Linh Truong, Schahram Dustdar: Elastic High Performance Applications - A Composition Framework. APSCC 2011: 416-423
Contribution area: programming human-based services in the cloud
Contributions: incorporate humans into a programming paradigm

<table>
<thead>
<tr>
<th>Programming languages</th>
<th>Multiple programming models</th>
<th>Execution environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Abstracting human compute units as program elements</td>
<td>• Shared memory (e.g., human –software – human), message passing (human-to-human), artifact-centric, etc., via APIs working atop the compute unit abstraction layer</td>
<td>• Computing capability /profile management: human computing power, reputation and incentive models</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Monitoring and enforcing incentives/rewards, quality of results, availability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Communication between human-middleware, among Individual Compute Units (ICU)/Social Compute Units (SCU) for exchanging artifacts and comprehending tasks</td>
</tr>
</tbody>
</table>
Our approach -- incorporate humans into a programming paradigm

Program elements → Programing languages and Programing models

Provisioning/Negotiation/Execution Cloud APIs

Service unit → Service unit → Common constructs → Hybrid units

Abstraction of HBS/SBS units and hybrid units

Service-based middleware

Runtime Monitoring and Enforcement
Communication
Service Life-Cycle Management
Capability/Profile Management

People
Things
Software
Programming constructs/elements

Similarity
Composition

Dependency
DataDependency
LocationDependency

Relationship

Elasticity
SocialRelation

Brokering

Forwarding
Delegation

Unit

Function

Cost
Benefit
Quality
Capability

Capability

HCU

HBS

SBS

Skill
- level: int

SCU

ICU

IaaS
NaaS
DaaS
PaaS
Contributions: programming hybrid services in the cloud

A cloud of hybrid services includes software-based services and human-based services that can be provisioned, deployed and utilized on-demand based on different pricing and incentive models.

HPU Definition: *HPU is a value describing the computing power of an HBS measured in an abstract unit. A cloud of HBS has a pre-defined basic power unit, hpu_θ, corresponding to the baseline skill bs_θ of the cloud.*

HPU can be defined for different „archetypes“

An „archetype“ characterizes the problem domain (e.g., bones simulation) that the ICU/SCU can solve (the type of tasks).

---

Programming a combination of HBS and SBS

e.g., preparing/managing inputs/outputs for HBS using SBS

```java
// using JClouds APIs to store log file of web application server
BlobStoreContext context =
    new BlobStoreContextFactory().createContext("aws-s3","REMOVED ","REMOVED");
BlobStore blobStore = context.getBlobStore();
// .... and add file into Amazon S3
Blob blob = blobStore.blobBuilder("hbstest").build();
blob.setPayload(new File("was.log"));
blobStore.putBlob("hbstest", blob);
String uri = blob.getMetadata().getPublicUri().toString();
VieCOMHBS vieCOMHBS = new VieCOMHBSImpl();
// assume that WM6 is the HBS that can analyze the Web Middleware problem
vieCOMHBS.startHBS("WM6");
HBSRequest request = new HBSRequest();
request.setDescription("Find possible problems from " + uri);
vieCOMHBS.runRequestOnHBS("WM6", request);
```
Summary of contributions

- **Contributions:**
  - a novel model for clouds of HBS and hybrid services provisioning
  - a framework for solving complex problems using clouds of hybrid services
  - programming primitives for hybrid services

- **Related work**
  - Crowdsourcing techniques focus on simple models of human capabilities utilization, existing cloud techniques do not consider human-based services

- **Selected publications**
  - Hong Linh Truong, Schahram Dustdar, Kamal Bhattacharya: Programming Hybrid Services in the Cloud. ICSOC 2012: 96-110. Best paper award
Conclusions (1)

- The evolution of underlying systems and the utilization of different types of resources under different models for elasticity lead to complex quality issues.
- We address several quality issues for software, data and people in an integrated manner for different perspectives.
- Our contributions:
  - Online performance monitoring and analysis for workflows.
  - Data concerns analysis and evaluation, data contracts, data and service contract compatibility for DaaS.
  - Composable cost evaluation for cloud applications.
  - Programming hybrid services in the cloud.
- These contributions support the development of multi-dimensional elasticity principles in complex service oriented systems.
Conclusions (2)

- **Research methodology:**
  - Real-world systems with measurable metrics
  - Characterizing systems through the definition, measurement, and analysis of metrics using their dependencies and system structure dependencies
  - Composable evaluation techniques are a key to metrics evaluation for understanding complex systems
  - Validation in real projects with software prototypes

- **Research collaboration**
  - Complex service-oriented systems require intensive collaborations with other scientists in other disciplines
Future work

- Hybrid compute units
  - Modeling and provisioning things, software, and people under the same service models
- Programming hybrid compute units for elastic processes
- Service engineering analytics of elastic systems
  - Elasticity specifications and reasoning techniques
  - Elasticity spaces analytics
- Application domains
  - „Social computer“ and smart cities
  - Computational science and engineering
Thank you!

Acknowledgements

Kamal Bhattacharya, Peter Brunner, Marco Comerio, Vincenzo D'Andrea, Hoa Dam, Schahram Dustdar, Thomas Fahringer, G.R.Gangadharan, Yike Guo, Lam-Son Le, Frank Leymann, Andrea Maurino, Michael Mrissa, Vlad Nae, Flavio De Paoli, Tran-Vu Pham, Reinhard Pichler, Michael Reiter, Robert Samboski, Benjamin Satzger, Vadim Savenkov, Martin Treiber, and Quang-Hieu Vu