Live Visualization of Large Software Landscapes for Ensuring Architecture Conformance

Florian Fittkau, Phil Stelzer, and Wilhelm Hasselbring

SESoS 2014 @ Vienna, Austria

2014-08-26

ExplorViz
Motivation

Introduction

- Large software landscapes are complex Systems-of-Systems
- Long-lived systems often tend to **architectural erosion**
- Continuous monitoring - but huge generated data
Large software landscapes are complex Systems-of-Systems
Long-lived systems often tend to architectural erosion
Continuous monitoring - but huge generated data
⇒ Scalable, live visualization for checking landscape architecture conformance

ExplorViz
Interactive approach for the live, explorable visualization of software landscapes [FWWH13]
ExplorViz Approach
Application Perspective

ExplorViz
Evaluation Motivation

Evaluation of Applicability

- **Low overhead and scalability to large systems** for our monitoring solution are **preconditions** of our architecture conformance checking approach.

- We use Cloud infrastructures (elastic, external analysis of monitoring data).

- **Here**: **Analysis of our monitoring solution** (overhead and scalability) using Cloud infrastructures.

- **Future work**: Analysis of the scalability of our visualization.
**Evaluation of Applicability**

- **SNoMonitoring**: Generation of workload with JMeter on JPetStore instances **without monitoring**
- **SWithMonitoring**: Same as **SNoMonitoring** but **with monitoring enabled**
Figure 1: Workload curve

- Private cloud (OpenStack)$^1$

$^1$ 8 servers [2x Intel Xeon (2.8GHz, 8 cores), 128 GB RAM]
Evaluation of Applicability

Figure 2: Deployment sketch in SNoMonitoring
Figure 3: Deployment sketch in *SWithMonitoring*
Results and Discussion

Evaluation of Applicability

Figure 4: Node allocation of JPetStore instances in SNoMonitoring and SWithMonitoring
Figure 5: Node allocation of JPetStore instances and analysis worker instances in 
SWithMonitoring
Threats to Validity

Evaluation of Applicability

- Only on our private cloud
- For external validity: other environments and applications
- Duration of our scenarios may be insufficient
- Longer studies have to be conducted
Related Work

- Architecture Conformance Checking
  - Dependency-structure matrices, source code query languages, and reflexion models [PTV+10]
  ⇒ Often **no dynamic analysis** and often only target **one system**
Related Work

- **Architecture Conformance Checking**
  - Dependency-structure matrices, source code query languages, and reflexion models [PTV+10]
  - Often no dynamic analysis and often only target one system

- **Software Visualization**
  - Application performance monitoring (APM) tools like AppDynamics
  - Most tools are limited to visualizing nodes and applications
  - Single application: comparison in [FWWH13]
Related Work

- **Architecture Conformance Checking**
  - Dependency-structure matrices, source code query languages, and reflexion models \([PTV+10]\)
  - Often **no dynamic analysis** and often only target **one system**

- **Software Visualization**
  - Application performance monitoring (APM) tools like AppDynamics
  - Most tools are **limited to visualizing nodes and applications**
  - Single application: comparison in \([FWWH13]\)

- **Monitoring Solutions**
  - Monitoring-as-a-Service solution \([MLS10]\) for whole cloud topology
  - Parallel analysis infrastructure \([BN03]\) using shared storage
Summary

- **ExplorViz** provides **abstractions** intended for **visual scalability** to check landscape architecture conformance checking
- If you are interested: Open source\(^2\) and contact\(^3\)

\(^2\)http://explorviz.net
\(^3\)florian.fittkau@email.uni-kiel.de
Summary and Future Work

Conclusions

Summary

- ExplorViz provides **abstractions** intended for **visual scalability** to check landscape architecture conformance checking
- If you are interested: Open source\(^2\) and contact\(^3\)

Future Work

- **Controlled experiment** for scalability evaluation of visualization
- Provide modeling capabilities of conceptual architecture

\(^2\)http://explorviz.net
\(^3\)florian.fittkau@email.uni-kiel.de
Holger Brunst and Wolfgang E. Nagel.  

Florian Fittkau, Jan Waller, Christian Wulf, and Wilhelm Hasselbring.  
Live trace visualization for comprehending large software landscapes: The ExplorViz approach.  
In Proc. of the 1st Int. Working Conf. on Software Visualization (VISSOFT 2013), September 2013.

Shicong Meng, Ling Liu, and Vijayaraghavan Soundararajan.  
Tide: Achieving self-scaling in virtualized datacenter management middleware.  

Static architecture-conformance checking: An illustrative overview.  