Towards a Dependability Control Center for Large Software Landscapes

Florian Fittkau, André van Hoorn, and Wilhelm Hasselbring

EDCC 2014 @ Newcastle upon Tyne, UK

2014-05-14

ExplorViz
Fully-Automatic Systems

Introduction

- Dependability: manual management tedious in large systems (e.g., clouds)
- Automatic techniques proposed
- Operators often mistrust fully-automatic systems
Introduction

- Dependability: manual management tedious in large systems (e.g., clouds)
- Automatic techniques proposed
- Operators often mistrust fully-automatic systems
- Vision: semi-automatic control center
Interactive approach for the live, explorable visualization of software landscapes [FWWH13]
Four perspectives:

- Symptoms
- Diagnosis
- Planning
- Execution
1. Phase: Symptoms

Envisioned Control Center

Symptoms

ExplorViz

Envisioned Control Center

- Jira (10.0.0.1)
- Workflow (10.0.0.4 - 10.0.0.7)
- PostgreSQL (10.0.0.3)
- Cache 10.0.0.8
- HyperSQL 10.0.0.9
- Neo4j 10.0.0.9
2. Phase: Diagnosis

Envisioned Control Center

![Diagram of Envisioned Control Center with Neo4j node and kernel node]
2. Phase: Diagnosis

Envisioned Control Center

Neo4j
10.0.0.9

kernel

Fittkau, van Hoorn, Hasselbring
Towards a Dependability Control Center
Envisioned Control Center

**Anomaly Score**

- Red dashed line: Alert level 1
- Orange dashed line: Alert level 0.5
- Blue solid line: Normal score

**Average Response Time**

- Green line: 30 minutes
- Grey bars: Average response times at different times of the day.
3. Phase: Planning

Envisioned Control Center

Warning
The software landscape violates its requirements for response times.

Countermeasure
It is suggested to start a new node of type 'm1.small' with the application 'Neo4J' on it.

Consequence
After the change, the response time is improved and the operating costs increase by 5 Euro per hour.

Start the instance?

Automatic change dialog
Envisioned Control Center

Node context menu

- Show details
- Restart
- Terminate
- Start new instance of same type
3. Phase: Planning (cont’d)

Envisioned Control Center

Node context menu

Application context menu
4. Phase: Execution

Envisioned Control Center

- Pushing execute button in the planning perspective
- Execution perspective is opened
- Shows **what is planned** and **what has already been conducted**
Prototype Tools

Envisioned Control Center

- Monitoring
  - Kieker [vHWH12]
- Online performance anomaly detection
  - ΘPAD [Bie12, Fro13]
- Root cause localization
  - RanCorr [MRvHH09]
- Online capacity management
  - SLAStic [vMvHH11]
Related Work

- Cloud management
  - E.g., Amazon CloudWatch, Microsoft Azure auto-scaling
  - No manual refinement of the reconfiguration plan
Related Work

- **Cloud management**
  - E.g., Amazon CloudWatch, Microsoft Azure auto-scaling
  - No manual refinement of the reconfiguration plan

- **Application performance monitoring (APM)**
  - E.g., AppDynamics, ExtraHop, or SPASS-meter [ES12]
  - Most tools only provide monitoring and reactive analysis
Related Work

- **Cloud management**
  - E.g., Amazon CloudWatch, Microsoft Azure auto-scaling
  - No manual refinement of the reconfiguration plan

- **Application performance monitoring (APM)**
  - E.g., AppDynamics, ExtraHop, or SPASS-meter [ES12]
  - Most tools only provide monitoring and reactive analysis

- **MAPE-K control loop tools**
  - E.g., Rainbow [GCH+04] or TRAP
  - Large part focuses on automatic adaptation
Summary

- Envisioned semi-automatic control center for cloud dependability basing on ExplorViz\(^1\)
- Open source tool ExplorViz available at [http://explorviz.net](http://explorviz.net)
- Visual plug-in-based integration platform for dependability management approaches

---

\(^1\)florian.fittkau@email.uni-kiel.de
Summary

- Envisioned semi-automatic control center for cloud dependability basing on ExplorViz\(^1\)
- Open source tool ExplorViz available at [http://explorviz.net](http://explorviz.net)
- Visual plug-in-based integration platform for dependability management approaches

Future Work

- Provide an implementation for our control center concept
- Develop plug-ins to integrate a number of reasonable dependability management approaches

\(^1\)florian.fittkau@email.uni-kiel.de
Tillmann Carlos Bielefeld.  
Online performance anomaly detection for large-scale software systems.  

Holger Eichelberger and Klaus Schmid.  

Tom Frotscher.  
Architecture-based multivariate anomaly detection for software systems.  
Masterarbeit, Kiel University, Oktober 2013.

Florian Fittkau, Jan Waller, Christian Wulf, and Wilhelm Hasselbring.  
Live trace visualization for comprehending large software landscapes: The ExplorViz approach.  

Rainbow: architecture-based self-adaptation with reusable infrastructure.  

Nina S. Marwede, Matthias Rohr, André van Hoorn, and Wilhelm Hasselbring.  
Automatic failure diagnosis in distributed large-scale software systems based on timing behavior anomaly correlation.

André van Hoorn, Jan Waller, and Wilhelm Hasselbring.  
Kieker: A framework for application performance monitoring and dynamic software analysis.  

Robert von Massow, André van Hoorn, and Wilhelm Hasselbring.  
Performance simulation of runtime reconfigurable component-based software architectures.  