

Component-based system for management of multilevel virtualization of networking resources

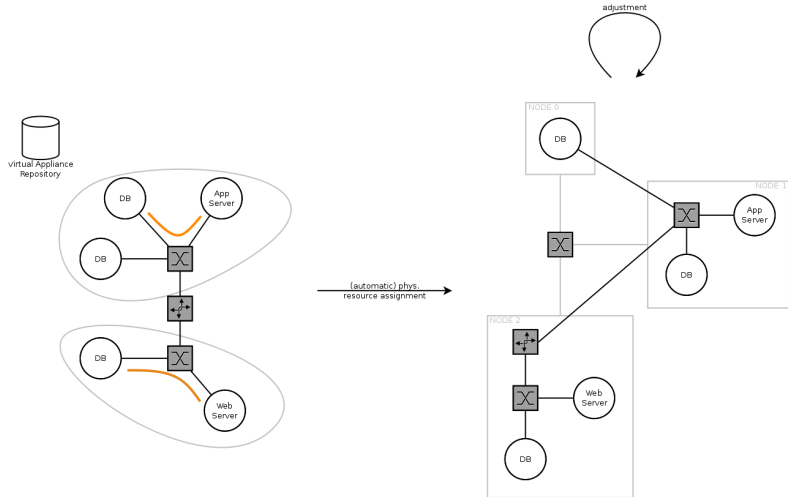
System komponentowy wspomagający wielopoziomową
wirtualizację zasobów sieciowych

Robert Boczek Dawid Ciepliński

prof. dr hab. inż Krzysztof Zieliński

29.03.2011

Multilevel network virtualization



Introduction

1. Introduction

2. Context

2.1 QoS-aware networking

2.2 Resource virtualization approaches

2.3 Multilevel network virtualization

2.3.1 Virtual network resources

2.3.2 Fine-grained QoS control

2.3.3 Virtual appliances

2.3.4 „Network in a box” concept

2.4 Applications and benefits of virtual infrastructures

2.4.1 Testing and simulations

2.4.2 Improving server-side infrastructure scalability

2.4.3 Infrastructure as a service

2.4.4 The role of resource virtualization in the SOA stack

Introduction

1. Introduction

2. Context

2.1 QoS-aware networking

2.2 Resource virtualization approaches

2.3 Multilevel network virtualization

2.3.1 Virtual network resources

2.3.2 Fine-grained QoS control

2.3.3 Virtual appliances

2.3.4 „Network in a box” concept

2.4 Applications and benefits of virtual infrastructures

2.4.1 Testing and simulations

2.4.2 Improving server-side infrastructure scalability

2.4.3 Infrastructure as a service

2.4.4 The role of resource virtualization in the SOA stack

Introduction

1. Introduction

2. Context

2.1 QoS-aware networking

2.2 Resource virtualization approaches

2.3 Multilevel network virtualization

2.3.1 Virtual network resources

2.3.2 Fine-grained QoS control

2.3.3 Virtual appliances

2.3.4 „Network in a box” concept

2.4 Applications and benefits of virtual infrastructures

2.4.1 Testing and simulations

2.4.2 Improving server-side infrastructure scalability

2.4.3 Infrastructure as a service

2.4.4 The role of resource virtualization in the SOA stack

Introduction

1. Introduction

2. Context

2.1 QoS-aware networking

2.2 Resource virtualization approaches

2.3 Multilevel network virtualization

2.3.1 Virtual network resources

2.3.2 Fine-grained QoS control

2.3.3 Virtual appliances

2.3.4 „Network in a box” concept

2.4 Applications and benefits of virtual infrastructures

2.4.1 Testing and simulations

2.4.2 Improving server-side infrastructure scalability

2.4.3 Infrastructure as a service

2.4.4 The role of resource virtualization in the SOA stack

Introduction

1. Introduction
2. Context
 - 2.1 QoS-aware networking
 - 2.2 Resource virtualization approaches
 - 2.3 Multilevel network virtualization
 - 2.3.1 Virtual network resources
 - 2.3.2 Fine-grained QoS control
 - 2.3.3 Virtual appliances
 - 2.3.4 „Network in a box” concept
 - 2.4 Applications and benefits of virtual infrastructures
 - 2.4.1 Testing and simulations
 - 2.4.2 Improving server-side infrastructure scalability
 - 2.4.3 Infrastructure as a service
 - 2.4.4 The role of resource virtualization in the SOA stack

Requirements analysis

3. Requirements analysis

3.1 Functional requirements

3.1.1 Instantiation

3.1.2 Discovery

3.1.3 Accounting

3.2 Non-functional requirements

3.3 Underlying environment characteristics

3.4 General approach and problems it imposes

3.4.1 Load balancing / Deployment

3.4.2 Infrastructure isolation

3.4.3 Broadcast domain preservation

3.4.4 Constraints

Requirements analysis

3. Requirements analysis

3.1 Functional requirements

3.1.1 Instantiation

3.1.2 Discovery

3.1.3 Accounting

3.2 Non-functional requirements

3.3 Underlying environment characteristics

3.4 General approach and problems it imposes

3.4.1 Load balancing / Deployment

3.4.2 Infrastructure isolation

3.4.3 Broadcast domain preservation

3.4.4 Constraints

Requirements analysis

3. Requirements analysis

3.1 Functional requirements

3.1.1 Instantiation

3.1.2 Discovery

3.1.3 Accounting

3.2 Non-functional requirements

3.3 Underlying environment characteristics

3.4 General approach and problems it imposes

3.4.1 Load balancing / Deployment

3.4.2 Infrastructure isolation

3.4.3 Broadcast domain preservation

3.4.4 Constraints

Requirements analysis

3. Requirements analysis

3.1 Functional requirements

3.1.1 Instantiation

3.1.2 Discovery

3.1.3 Accounting

3.2 Non-functional requirements

3.3 Underlying environment characteristics

3.4 General approach and problems it imposes

3.4.1 Load balancing / Deployment

3.4.2 Infrastructure isolation

3.4.3 Broadcast domain preservation

3.4.4 Constraints

Solaris OS as a resource virtualization environment

4. Solaris OS as a resource virtualization environment

4.1 General information

4.2 Lightweight OS-level virtualization with Solaris Containers

4.3 Crossbow - network virtualization technology

4.4 Resource access control

Solaris OS as a resource virtualization environment

4. Solaris OS as a resource virtualization environment

4.1 General information

4.2 Lightweight OS-level virtualization with Solaris Containers

4.3 Crossbow - network virtualization technology

4.4 Resource access control

Solaris OS as a resource virtualization environment

4. Solaris OS as a resource virtualization environment

4.1 General information

4.2 Lightweight OS-level virtualization with Solaris Containers

4.3 Crossbow - network virtualization technology

4.4 Resource access control

Solaris OS as a resource virtualization environment

4. Solaris OS as a resource virtualization environment

- 4.1 General information
- 4.2 Lightweight OS-level virtualization with Solaris Containers
- 4.3 Crossbow - network virtualization technology
- 4.4 Resource access control

The system architecture

5. The system architecture

5.1 High-level design

5.2 System components and their responsibilities

5.2.1 Assigner

5.2.2 Supervisor

5.2.3 Worker

5.3 Crossbow resources instrumentation

5.4 Domain model and data flows

The system architecture

5. The system architecture

5.1 High-level design

5.2 System components and their responsibilities

5.2.1 Assigner

5.2.2 Supervisor

5.2.3 Worker

5.3 Crossbow resources instrumentation

5.4 Domain model and data flows

The system architecture

5. The system architecture

5.1 High-level design

5.2 System components and their responsibilities

5.2.1 Assigner

5.2.2 Supervisor

5.2.3 Worker

5.3 Crossbow resources instrumentation

5.4 Domain model and data flows

The system architecture

5. The system architecture

5.1 High-level design

5.2 System components and their responsibilities

5.2.1 Assigner

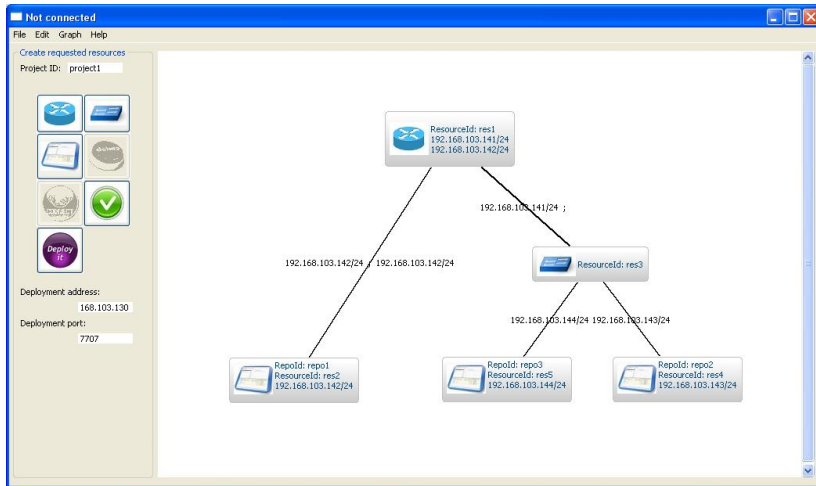
5.2.2 Supervisor

5.2.3 Worker

5.3 Crossbow resources instrumentation

5.4 Domain model and data flows

Gui console



Implementation

6. Implementation

6.1 Implementation environment

6.2 Domain model transformation details

6.3 Low-level functions access

6.4 Building and running the platform

Implementation

6. Implementation

6.1 Implementation environment

6.2 Domain model transformation details

6.3 Low-level functions access

6.4 Building and running the platform

Implementation

6. Implementation

- 6.1 Implementation environment
- 6.2 Domain model transformation details
- 6.3 Low-level functions access
- 6.4 Building and running the platform

Implementation

6. Implementation

- 6.1 Implementation environment
- 6.2 Domain model transformation details
- 6.3 Low-level functions access
- 6.4 Building and running the platform

Proposals of Case Studies

7. Case Study

7.1 Clustered GlassFish

7.2 Multimedia server

Proposals of Case Studies

7. Case Study

7.1 Clustered GlassFish

7.2 Multimedia server

Summary

8. Summary

8.1 Conclusions

8.2 Achieved goals

8.3 Further work

Summary

8. Summary

8.1 Conclusions

8.2 Achieved goals

8.3 Further work

Summary

8. Summary

8.1 Conclusions

8.2 Achieved goals

8.3 Further work