G-Lab DEEP
Federation Prototype Demo

Konrad Campowsky, Fabian Eichhorn, Nils Lüdicke, Carsten Schmoll, Dr. Sebastian Wahle

G-Lab Meeting Berlin
14.-15. Februar 2012

sebastian.wahle@fokus.fraunhofer.de
Why Federation?

\[
\text{Cost} = f(\text{complexity, resource, environmental conditions})
\]

As the opposite of the abstraction level

- **Formal model**
- **Simulation**
- **Emulation**
- **Real systems**
- **Real Federated System**
  - Real systems and apps
  - Real conditions
  - Distributed resources

- **Homogeneous Federation**
- **Heterogeneous Federation**

**Models**
- Sys, Apps, Platforms, Conditions

**Models for key OS mechanisms**
- Algorithms and kernel apps
- Abstracted platforms
- Synthetic conditions

**Real OS**
- Real applications
- “In-lab” platforms
- Synthetic conditions

**Loss of real conditions**
- Loss of experimental conditions, reproducibility, repeatability, etc.

Resource Federation as a Research Field

- Federation scenarios
- Identity
- Controlled federation
- Resource description
- Policy
- Federation organization
- Resource sharing
- Resource orchestration
- Resource relationships
- Automated provisioning
- Features
- Connectivity
- Areas
- Heterogenous
- Homogenous
- Types
- Generic
- Closed
- Club good
- Open
- Commercial
- Public good
- Private good
- Time savings
- User
- Operation
- Re-use instead of re-build
- Benefits
- Access to scarce resources
- Cost reduction
- Re-use instead of re-build
- Cloud Computing
- Grid Computing
- Service Oriented Architecture
- Federation scenarios
- Recursive
- Centralized
- Stakeholders
- Resource provider
- User
- Time savings
- Cost reduction
G-Lab Ressource Federation Demo

- Combines 3 prototypes
  - ToMaTo (mostly G-Lab phase 1)
  - Teagle (extended from OS prototype as part of G-Lab Deep)
  - Multi-Hop Packet Tracking
ToMaTo Overview

- **Origin**
  - ToMaTo is being developed as part of the experimental facility of G-Lab
  - Open-Source since version 2.0

- **What it is**
  - A topology-oriented testbed software
  - An easy-to-use feature-rich tool for networking experiments

- **What it is NOT**
  - A management tool for physical network topologies — Topologies are virtual
  - A tool to manage virtualized production networks — ToMaTo is still under development

Source: Dennis Schwerdel „ToMaTo - Topology Management Tool“
http://dswd.github.com/ToMaTo/presentations/general.html
ToMaTo Overview (cont.)

- Topologies may contain
  - Devices
  - Connectors

- Devices
  - Active components
  - E.g. computers
  - Produce/Consume data

- Connectors
  - Networking components
  - E.g. switches, routers
  - Transport/Manipulate data

Source: Dennis Schwerdel „ToMaTo - Topology Management Tool“
http://dswd.github.com/ToMaTo/presentations/general.html
ToMaTo Overview (cont.)

- **ToMaTo contains 3 parts**
  - **Host part**
    - Based on PROXMOX VE
    - Offers virtualization
  - **Backend**
    - Controls hosts via SSH
    - Centralized logic, resource management, user accounts
    - Offers XML RPC interface
  - **Frontend(s)**
    - Offer a GUI to users
    - Web-based interface (default)
    - Command line interface (for scripting)

Source: Dennis Schwerdel „ToMaTo - Topology Management Tool“
http://dswd.github.com/ToMaTo/presentations/general.html
Teagle Overview

- Heterogeneous resources are offered by several domains
- Domains engage in federation and share resources
- Federation Organization represents the federation to 3rd parties
- A federation control tool called TEAGLE executes management operations via a control framework
- Re-use resources across the boundaries of domains and communities instead of re-building infrastructure
Teagle Federation Model

Virtual Resource Grouping

Domain

SET

M

REG

GW

R
- Domain Managers dispose of Resource Adaptors
- Allow to control heterogeneous resources
- Interface T1 is standardized
- Interface T2 is proprietary
- Usually T1 is CRUD
  - CREATE
  - READ
  - UPDATE
  - DELETE
- SET might be rather complex itself
Teagle Resource Relationships

- Different types of resource relationships
- $MR_A$ set of resources provided by domain $A$
- Ressource hierarchies
  - $p : MR_A \rightarrow MR_A, x \rightarrow y$
  - $p(B) = A$
- Configuration references
  - $r_{\text{intra}} : MR_A \rightarrow MR_A, x \rightarrow y$
  - $r_{\text{inter}} : MR_A \rightarrow MR_B, x \rightarrow y$
  - $r_{\text{intra}}(B) = E$ (if $B,E \in MR_A$)
- Allows the definition of virtual resource groupings
  - $VG = \{x \mid x \in MR_A \lor x \in MR_B\}$
Teagle Orchestration

- Orchestration of resources across domains
- 2 rules:
  - If w is parent of x then w must be provisioned before x
  - If z is referenced by y then z must be provisioned before y
- Rules allow sorted sequences of provisioning requests
- e.g. \{A, E, D, B, C\} and \{E, D, A, B, C\}
Design of a Virtual Resource Grouping
Why Packet Tracking?

- Nowadays an increasing number of sophisticated network applications (such as multimedia and gaming) demand a high level of quality of service (QoS) from the network, for example small latencies and low packet loss.

- In order to supervise the fulfilment of such demands, especially in mobile networks with a higher variability in network characteristics, a deeper understanding of the network and what happens within it is required.

- Packet Tracking offers a distributed network surveillance solution for tracking the paths of data packets through networks.

- The Packet Tracking software package allows to “track” single packets in the network and answer enquiries related to routes followed by any selected packet across multiple hops through the network, on topics such as packet delays and losses between nodes, service chains used by some data traffic, cross traffic influence, and network QoS.
Packet Tracking Network Observations

- Basis for experimental research
  - Observe metrics of interest
  - Monitor environment conditions

- But also: a scientific challenge
  - Accuracy vs. overhead
  - Privacy vs. information demand
  - Heterogeneity vs. comparability of results

- Coordinated passive network observation
- Enables hop-by-hop path and quality of packet delivery analysis

Map showing a packet tracking scenario involving different nodes from testbeds such as VINI, G-Lab, PlanetLab and others.
Technology

**Collector:** Calculate Path, Delay,…

- **Parameter adjustment**
- **ID generation**
- **Hash-based selection**
- **Timestamping**

**IPFIX** (id, timestamp, sample rate,..)

**Measurement Process**

- **IPFIX** (path, delay,..)
The Demo

- Use the Teagle GUI to acquire resources from ToMaTo
- Teagle domain managers connect to the ToMaTo API via XML RPC
- Once a topology has been obtained from ToMaTo, it is available to the user
- Teagle can access the topology and deploy additional resources
- Via the GUI, the user can request Teagle to install and configure the Packet Tracking measurement probes
- An example experiment will be started “sin(x)2asciiX”
- Visualize network traffic using Packet Tracking collector and netview
Thank You

Questions?

Y

Answer

known?

N

Thank
Audience

Y

State that time
has run out

N

Leave