Open Computing Infrastructures for Elastic Services: Contrail Approach

Christine Morin, INRIA
Contrail Project Coordinator

contrail is co-funded by the EC 7th Framework Programme under Grant Agreement nr. 257438
Contrail Objectives

• Design, implement, validate and promote an open source software stack for cloud federations

• Develop a comprehensive Cloud platform integrating a full IaaS and PaaS offer

• Allow Cloud providers to seamlessly integrate resources from other Clouds with their own infrastructure

• Provide trusted Clouds by advanced SLA management

• Break the current customer lock-in situation by allowing application migration from one cloud to another
Contrail in a Nutshell
Federation Scenarios

- Cloud brokering
- Cloud aggregation
- Cloud bursting
Cloud Brokering

Contrail Federation Service

Cloud A

Cloud B

Cloud C

Find the best cloud to execute a given application
Cloud Aggregation

Contrail Federation Service

Provide the user with resources from different cloud providers.
Cloud Bursting

Seamless integration of resources from a public cloud in a private cloud

Private cloud

Contrail Federation Service

Public cloud
Federation Services

- User management
- Authentication, authorization (access, usage)
- SLA management
- Negotiation, monitoring
- Reputation management
- Coordination of application deployment over multiple clouds
PaaS Objectives

• A coherent set of high-level cloud services (PaaS)

• Elasticity, scalability

• Identify & implement a simple & powerful set of functionalities which address real programmers’ needs

• Define & enforce SLAs that are relevant in a PaaS environment
PaaS Services

- High level services
  - SQL / NoSQL
- Runtime environments
  - Map-reduce / Bags of tasks / Web applications
- IaaS cloud
  - Federation of IaaS clouds
• **Integration between multiple services**

  • Amazon & Azure are the only truly integrated PaaS offers

  • No good integrated open-source offer at the moment

• **SLA enforcement**

  • Very little support for SLA in current PaaS offers

  • All implemented as simple rules triggered by threshold violations
Gaps in the State of the Art (2/2)

- **Customer lock-in**

- Current PaaS platforms do not offer any easy way out

- The open source principle allows for **new business models**

- Small applications have an economic incentive to use public clouds

- When the application grows you can seamlessly migrate a combination of private and public clouds (i.e. federation)

- Migrating (parts of) an application to a private cloud allows for increased levels of **customization**
Composed Services

Web Service 1

Web Service 2

Data Store

Web Service 3
ConPaaS Service Interface

Service Consumer

Functional Interface

Management Interface

Provisioning Interface

Administrator

-
Example: a Web Application

1. Load Balancer
2. Application Servers
   - PHP
   - PHP
   - PHP
   - PHP
   - PHP
   - PHP
3. Data Services
   - Key-Value
   - NoSQL
   - SQL
   - GAFS
Service Deployment

VM

Ganglia gmetad

PHP Manager

VM

Ganglia gmond

Proxy / Load Balancer

VM

Ganglia gmond

Lighttpd

VM

Ganglia gmond

PHP process

VM

Ganglia gmond

Proxy / Load Balancer

VM

Ganglia gmond

Lighttpd

VM

Ganglia gmond

PHP process
SLA in ConPaaS

End user

SLA: average(response_time) < 500ms

PHP service
(using 1 or more machine instances)

Logging service
(using 1 or more machine instances)

SQL data service
(using 1 or more machine instances)

No particular SLAs here. However, these services are being provisioned to help the PHP service maintain its own SLA.
Integrated SLA Management

Service Consumer

Functional Interface

Management Interface Provisioning Interface

Binding

Functional Interface

Management Interface Provisioning Interface

SLA Negotiation
Bag of Tasks

BoT Service

BoT Master

Monitoring

BoT Workers

Result Repository

Binding

Data storage service

PaaS customer

Management API request

Functional API request

Control
- Hadoop instantiated on a per user basis
- Isolation
- JobTracker providing standard Hadoop interface
- Automatic adaptation of the number of worker machines
- New TaskTracker created on demand for map and reduce tasks
- Input/output data stored in GAFS
Database as a Service

- Key value store
  - Scalarix
- SQL
  - MySQL, PostgreSQL
SQL Database Service

PaaS customer

Other services

SQL Server manager

Monitoring

MySQL agent

MySQL

DB storage

SQL Server service
Components of the SQL Service
<table>
<thead>
<tr>
<th>Service</th>
<th>Status</th>
<th>Created Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front-End</td>
<td>running</td>
<td>May 20, 2010</td>
</tr>
<tr>
<td>Database</td>
<td>stopped</td>
<td>January 21, 2011</td>
</tr>
<tr>
<td>Ads</td>
<td>stopped</td>
<td>July 14, 2010</td>
</tr>
<tr>
<td>Ismail TESTING</td>
<td>running</td>
<td>March 1, 2011</td>
</tr>
<tr>
<td>service-4</td>
<td>stopped</td>
<td>March 2, 2011</td>
</tr>
</tbody>
</table>

Access:
- Front-End: [http://contrail.aws.com/blog](http://contrail.aws.com/blog)
Configure service

Service Name: Front-End
Type: PHP Web Service
Date of creation: May 20, 2010

Stage

URL: http://10.3.4.54-stage/

You may update the stage by:
- uploading archive
- checking out repository

Choose File: app.zip

Example: .zip, .tar of your source tree

+ commit new version

Versions

<table>
<thead>
<tr>
<th>Active Version</th>
<th>Created Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>code-BGvkbY</td>
<td>a few moments ago</td>
</tr>
<tr>
<td>code-sTfxBL</td>
<td>a few moments ago</td>
</tr>
</tbody>
</table>

PHP Settings

SSH: admin@10.2.45.2:/home/admin/php.ini

- Software version: 5.3
- Maximum script execution time: 30 seconds
- Memory limit: 128M
SLA Management

Cloud User Interface + API

IaaS Federation
(distr. registry, VO management, identity mgmt)

Integration, Validation and Testing

Use cases and Exploitation

Platform-as-a-Service

well known abstractions
(POSIX API, x86 ISA, IP Network)

Core Virtual Infrastructure Layer

Network
Compute
Storage
Providers

Map/Reduce
Bag-of-Tasks
Web-servers
SQL

key-value store
structured storage

Resource Selection

Security

SLA Management

Monitoring
Accounting
SLAs
LA: Objectives

• **Support for the full life cycle of SLAs**
  - Creation, instantiation and enactment of agreements at all levels of the Cloud services stack: infrastructure, IaaS federations, ConPaaS
  - Dynamic SLA negotiation, monitoring and enforcement

• **Provide monitoring and accounting for Contrail services**
  - Measure and record usage of resources per user and attribute that usage to the respective providers in the federation
  - Secure monitoring data distribution and aggregation

• **Extend SLAs to QoP**
  - Integrate QoP guarantee specification in SLAs, monitor and possibly enforce them

• **Allow cost-based QoS enforcement**
Interaction Model

- User
- Federation
- Provider
- IaaS / PaaS Resource

Negotiate

Provision

Split SLA

Provision

SLA violation

Adjustment

Enforcement action

Monitoring events

Adjustment
Contrail Approach

• Reuse SLA@SOI framework as a starting point
  • Integration with Contrail internal interfaces and components
  • Integration with domain-specific reasoning/monitoring plugins

• Extend SLA@SOI with:
  • Federation support
  • QoP support
  • Integration of external providers
  • Reputation model for providers

• Leverage accounting and economic models to do Cost-based QoS enforcement
Contrail IaaS

- Processor, network and storage virtualization
- Virtual Execution Platform (VEP)
- Virtual Infrastructure Network (VIN)
- Global Autonomous File System (GAFS)
A reliable distributed storage system for cloud customers and cloud infrastructure

- Storage as a Service
- Storage of user data
- Accessible from within the cloud and over the Internet
- Storage repository for the cloud infrastructure
- VM images
Storage Repository

Diagram:
- **Data Center A**
  - Server
  - Hypervisor
  - VM1
  - VM2
  - Loads VM image for VM 2

- **Data Center B**
  - Stores VM image for VM 2

- **Storage Repository** (cross data center)
  - Stores VM image
  - Replicates VM image

- **User Interface (web GUI...)**
  - Uploads VM image
Global Autonomous File System (GAFS)

- **Posix** file system
- **Distributed**: across different data centers, federated
- **Autonomous**: self-healing, self-tuning, zero configuration
- **Reliable**: highly available, with data safety and QoS
- **Scalable**: can be dynamically scaled up and down on demand
GAFS built from XtreemFS

- Object-based
- Distributed (WAN)
- Replicated
- SSL support
- Linux/Windows/Mac OS
Autonomous File System

- Manual maintenance/management infeasible (size, geographic distribution)
- Automatic installation/configuration of servers
- Automatic replacement of failed servers
- Admins replace failed servers from time to time
- Automatic tuning of the file system
- Replica placement and replication factor
Computing Resource Management

- High Performance Real-Time Scientific Data Analysis
- Distributed Provision of Georeferentiated Data
- High-throughput Electronic Drug Discovery
- Multimedia Processing Service Marketplace

Use cases and Exploitation

- Map/Reduce
- Bag-of-Tasks
- Web-servers
- key-value store
- structured storage
- SQL

Platform-as-a-Service

well known abstractions
(POSIX API, x86 ISA, IP Network)

Cloud User Interface + API

- Resource Selection
- Monitoring
- Accounting
- SLAs

IaaS Federation
(distr. registry, VO management, identity mgmt)

Core Virtual Infrastructure Layer

Network
Storage

Security

Providers

Integration, Validation and Testing
Application Deployment, VM & Resource Management

- Application deployment
- SLA document for resource provisioning
- OVF document for application deployment
- Document linking the SLA & OVF documents
- Provisioning based on negotiated SLA
- Resource reservation (optional)
- Elasticity
Resource Allocation

• Customers satisfaction
• SLA compliant resource allocation
• Efficient resource management in data centers
• Policies: green computing, business, utilization…
VMs on a single multi-core processor.
VMs on a single server
VMs on a single cluster
VMs in multiple data centers
VM Management in First Contrail Release

- Re-use OpenNebula
- Basic resource management policies
- Focus on application deployment based on OVF
- Use GAFS as the repository for VM images in OpenNebula
- Other IaaS technologies supported in forthcoming releases
Virtual Network Infrastructure

- High Performance Real-Time Scientific Data Analysis
- Distributed Provision of Georeferentiated Data
- High-throughput Electronic Drug Discovery
- Multimedia Processing Service Marketplace

Use cases and Exploitation

Platform-as-a-Service

- Map/Reduce
- Bag-of-Tasks
- Webservers

- key-value store
- structured storage
- SQL

well known abstractions
(POSIX API, x86 ISA, IP Network)

Cloud User Interface + API

- Federation
- Security

Core Virtual Infrastructure Layer

- Compute
- Storage

- Providers

Integration, Validation and Testing

IaaS Federation
(distr. registry, VO management, identity mgmt)
Virtual Network (VIN)

- A **cohesive**, virtual cluster **platform** with a both secure and efficient virtual network (VN).

- The **VIN** serves **within a cloud**, and extends to other infrastructures (for **cloud federation**), to the **GAFS** storage, and to external **client machines**.

- **Authentication** is based on cloud-level user accounts; all **communication is encrypted**.
Virtual Network: Limitations of Current Solutions

• Many vendor-specific solutions are available or announced

• Existing solutions neither offer cloud federation (for commercial reasons) nor SLA’s

• Open source solutions

  • **OpenNebula**: no proper VN support, allows “hook” scripts to plugin (not useful for VN solutions

  • **Nimbus**: simplistic network configuration; VN separation by Ethernet bridges in VMM's

  • **Eucalyptus**: no VN's; relies on VLAN's and firewall rules

  • **ViNe**: dedicated user-space routers
VIN Design

• Using IPsec as base technology
  – Faster (in kernel) than Open VPN (in user space)
  – Encryption can be chosen, even turned off
  – Flexible, can be used both on physical machines and within a guest VM
  – Promising platform that can integrate all envisioned platforms (virtual clusters, federations, GAFS, user machines)

• Create virtual network (VN) dynamically per user/
Example: VN Support for Web Application
Break Vendor Lock-in

- Use open standard
- OCCI for IaaS
- OVF (DMTF) for application descriptions
- CDMI for VM image storage
- WS-Agreement for SLA
- ...
Conclusions

- Bringing trust in cloud computing is challenging

- Contrail European Project (2010-2013)
  - http://www.contrail-project.eu

- Contrail technology implementation in progress

- Leverage existing open source software & rely on standards

- First release in Spring 2012 in open source

- One of the case studies to be selected to become a demonstrator
contrail is co-funded by the EC 7th Framework Programme

Funded under: FP7 (Seventh Framework Programme)
Area: Internet of Services, Software & virtualization (ICT-2009.1.2)
Project reference: 257438
Total cost: 11,29 million euro
EU contribution: 8,3 million euro
Execution: From 2010-10-01 till 2013-09-30
Duration: 36 months
Contrail Federation

contrail
open computing infrastructures for elastic services

Public/Private

Amazon

Azure
Cooperation over Clouds: VDC

- An organization with a data center
  - data center managed as a private cloud (cloud API)
  - peak load: submit requests to external clouds
- Virtual Data Center
  - user sees the same execution environment on external clouds as in the organization data center
  - accounting, bills to the organization
- Organization users can share data and services
  - all services available on the organization data center are extended to the external providers
Integrated IaaS & PaaS

Runtimes for elastic applications

High level storage services

IaaS
Architecture of the SQL Service

VM level

Clients

Federation level

ConPaaS

SQL Server manager

VM level

storage

JDBC  ODBC  REST  AMQP

MySQL agent  MySQL  MySQL agent  MySQL  MySQL agent  MySQL

MySQL  MySQL  MySQL  MySQL
Options for Storing the Database Files & Logs
Key Value Store Service

Scalarix

P2P overlay

Replication

Paxos transaction protocol

Globally distributed DHT
Challenges / SLA lifecycle & QoP

• Offer SLAs at Federation level
  • SLA splitting/composition issues, split/migrate decisions and similar trade offs
  • SLA splitting in cloud-bursting scenarios

• Enforce QoS guarantees when lower layers don't offer any

• Cost-based QoS enforcement
  • SLA offers the best possible quality at a predefined cost, monitoring is not (only) on QoS but also on cost (from accounting). Resource allocation is filtered in SLA layer and further provisioning requests are denied if their cost is not on budget

• Monitor offered QoP and enforce it

• issues are: observability, security & privacy
Challenges / Monitoring & Accounting

• Automatic monitoring setup from SLA definition
  
  • SLA specification language should be complete and precise enough (e.g. When, Which, Where, What, How)
  
  • System should offer enough data from sensors and flexible aggregation / composition at SLA level
  
• Enable pay-per-use across different providers
  
  • Different providers may have different billing/charging models
SLA manager with domain-specific plug-ins

- Negotiation
- Monitoring
- Provisioning
- Enforcement
Contrail Software Stack

- **Federation layer**
  - User portal to multiple providers
  - Negotiation with providers

- **Provider layer**
  - Business layer
  - SLA negotiation
  - Monitoring
  - SLA enforcement
  - Accounting, billing

- **Resource layer**
  - Resource allocation
  - VM deployment
Contrail in a Nutshell

- High Performance Real-Time Scientific Data Analysis
- Distributed Provision of Georeferentiated Data
- High-throughput Electronic Drug Discovery
- Multimedia Processing Service Marketplace

Use cases and Exploitation

Platform-as-a-Service

- Map/Reduce
- Bag-of-Tasks
- Web-servers
- key-value store
- structured storage
- SQL

Core Virtual Infrastructure Layer

Cloud User Interface + API

- well known abstractions
  (POSIX API, x86 ISA, IP Network)

IaaS Federation

- Providers
  (distr. registry, VO management, identity mgmt)

Integration, Validation and Testing

Security
SLA Negotiation

federation

user management/authorization

SLA negotiation

SLA lifecycle management

SLA coordination

application coordinator

monitoring

provider

user management/authorization

SLA negotiation

SLA lifecycle management

SLA enforcement

monitoring agent

resource

resource reservation

application deployment

appliance management

GAFS

VIN

appliance monitoring

appliance hosting

appliance
Application Deployment

**federation**
- user management/authorization
- SLA negotiation
- SLA lifecycle management
- SLA coordination
- application coordinator
- monitoring

**provider**
- user management/authorization
- SLA negotiation
- SLA lifecycle management
- SLA enforcement
- monitoring agent

**resource**
- resource reservation
- application deployment
- appliance management
- GAFS
- VIN
- appliance monitoring
- appliance hosting
- appliance
## Contrail Consortium

<table>
<thead>
<tr>
<th>Country/Partner</th>
<th>Research</th>
<th>SME</th>
<th>Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>INRIA</td>
<td>EBM WebSourcing</td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>ZIB</td>
<td></td>
<td>HP-IIC</td>
</tr>
<tr>
<td>Italy</td>
<td>CNR</td>
<td></td>
<td>Tiscali</td>
</tr>
<tr>
<td>the Netherlands</td>
<td>VUA</td>
<td>GENIAS</td>
<td></td>
</tr>
<tr>
<td>Slovenia</td>
<td></td>
<td></td>
<td>XLAB</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>SFTC</td>
<td></td>
<td>CONST</td>
</tr>
</tbody>
</table>