Towards THE MAGIC GREEN BROKER

un courtier d'énergie pour le Cloud

Jean-Louis Pazat
IRISA/INSA Rennes, FRANCE
MYRIADS Project Team
OUTLINE

- Clouds and SOA: A green perspective
- The Magic Green Broker
- Conclusion
CLOUDS & SOA
a green perspective
CLOUDS: An “old” idea?

GRID foundations: Computing as an "utility"

Novembre 1998
CLOUDS vs GRIDS

• GRID is about technology
  • Tools for using distributed resources in a seamless way

• CLOUD is about business
  • Pay-per-use model for data & computation

• The technical basis of Clouds:
  • Virtualization

• Clouds are today Data/computer centers
  • Federations of Clouds/Grids in Clouds/Grids in Grids are for tomorrow
CLOUDS

- Two types of usage
  - Computing
  - Data Storage

- More than one concept
  - IaaS: Infrastructure as a Service
    - you get a "virtual computer"
  - PaaS: Platform as a Service
    - includes OS and software tools to build your (web) application
  - SaaS: Software as a service
    - Software is installed, running. You can call a “service”
Green Challenges for CLOUDS

- **Should be seen in a business perspective**
  - **CLOUD users**
    - don’t care about energy used in Data centers
    - their main concern is price
  - **CLOUD providers**
    - Care about energy
      - Because of the cost
      - Because of their reputation
- **How to enforce a green perspective ?**
  - Taxes ?
SERVICES & SOA

• In the SOA model applications are hidden
  • Everything is a “service” or a “process”
• No more application view
  • A service does something for you
  • Could be a piece of software or a human
  • Described with an interface, can be dynamically discovered and invoked
  • One should not worry about “who” does the work (it may change)
  • Most services are stateless
• A Business Process describes
  • How to achieve some goal
  • Uses services
• New interest in SOA due to Cloud usage
SOA: A green perspective?

- Green IT?
  - Green workflows, green services, green SLAs

- Green Services?
  - Services can expose some QoS parameters
  - => Need for green parameters
    - Energy, CO2

- Usual difficult problem:
  - How to enforce/guarantee SLA?
    - Even more difficult with energy
Cloud computing & Services

- **Layered Architecture**
  - **Final User**
    - chooses a service provider
    - calls services
  - **Service provider**
    - chooses a platform provider
    - provides service implementation
  - **Platform provider**
    - chooses infrastructure provider
    - deploys/configure service
  - **Infrastructure provider**
    - Mainly Provides (V)Ms
Cloud Computing & Services
User vs Provider Requirements

**Users’ view:** Price (including energy) & Quality (including CO2)
- Final user (SaaS user) needs
  - guarantees on service execution (Performance, availability)
- SaaS provider (SaaS & PaaS user)
  - guarantees on single service execution (Performance, availability)
  - Easily deployment of distributed applications
- PaaS provider (IaaS user)
  - Easily deployment of distributed applications
  - Elastic resources

**Providers’ view:** Profit & Reputation
- Efficient use of resources (including energy consumption)
  - Includes « virtual resources » such as Services
  - High resource utilization
Cloud Computing & Services
a limited choice …

Final user

Service provider

Platform provider

Infrastructure provider

Amazon Web Services™ SES
Amazon Web Services™ EMR
Amazon Web Services™ EC2
Google Apps
Google App Engine
Google Compute Engine
Cloud Computing & Services
I want to choose
Cloud Computing & Services
I want to choose

- I want to choose everything
- According to « green » and many other guarantees
- I want to change dynamically my providers
- Today
  - this is not possible
  - However it would be a nightmare…
Cloud Computing & Services
A bigger Picture

- This is a business organization for tomorrow’s CLOUDs
- This is not a technical layered architecture
- Many different players
- Need for a broker to help users
- A «Magic Broker» could help…

The key player is the Service composition provider

Service Composition provider
Service provider
Platform provider
Implementation Provider
Infrastructure provider

SaaS
PaaS
IaaS
The Magic Green Broker
The Magic Green Broker

• I need a « service/cloud advisor »
  • Let’s call it a Magic Green Broker
  • Allows to choose the best providers, the best configurations
  • Is able to dynamically adapt software & resources
  • Negotiates/controls quality
  • Is the best way to get end-to-end energy control ?

• But still one of my dreams…
Building a Magic Green Broker

• Adaptation at the IaaS level: Snooze
• Multilevel adaptation: SAFDIS
• Brokering at the PaaS level: QU4DS
Adaptation at the IaaS level

SNOOZE: an Autonomous & Energy Aware VM Management System

Context & Motivation

- Huge data centers (dozen of thousands of servers)
  - Scalable management of large numbers of VM
- Energy-hungry data centers
  - Energy conservation during periods of low utilization

Approach

- Scalability
  - Distributed VM management
  - Dynamic hierarchical management system
    - Self-organizing & self-healing hierarchy
- Energy conservation
  - Idle nodes in power-saving mode
  - Holistic approach
• Scalable VM Management System
  • Hierarchical algorithm for VM placement
  • Scalable monitoring
  • Heartbeats, multicast groups and election algorithm for self-organization and self-healing
• Holistic approach for Energy conservation
  • Underload/overload mitigation
  • Dynamic consolidation
    • Modified version of the SERCON algorithm
  • Power management
• Evaluation on Grid’5000 on a 144 node cluster with realistic scientific and elastic web applications
  • Fault tolerance features do not impact application performance
  • Negligible cost is involved in performing distributed VM management
  • System remains highly scalable with increasing number of VMs and hosts
  • Significant energy savings for elastic web applications
Multilevel Adaptation: SAFDIS

- The SAFDIS experimental framework
  - Generic framework based on the MAPE-K model
  - Distributed & decentralized; OSGi implementation
  - Focus on Distributed Adaptation & Planning

![Diagram showing Monitoring, Analysis, Planning, and Execution with Service Composition and Single Service nodes connected to an Execution Platform]
Adaptation at Any Level

Adaptation Framework

- Policy
- Guide
- Decision
- Planning
- Execution
- Event Manager

Service-Based Application
- Monitor
- Probe
- Effector

Service-Oriented Platform
- Monitor
- Probe
- Effector

Infrastructure (OS & Hardware)
- Monitor
- Probe
- Effector

Framework internals
Monitoring

- **Goal**
  - To provide a dynamic view of the system
- **How**
  - Probes → Monitors → Event Manager
  - Pull & push probing
  - Composite events
Results

• Prototype
  • partly integrated in a component based environment
  • Large scale experiments still missing
  • Used in an « Internet of Things » Application

• Some lessons
  • Decision is one of the most important part
    • Early bad decisions can lead to worse configurations
    • Late decisions cannot solve problems
  • Planning is difficult
    • Efficiency ?
    • Avoid « hazardous states »
A PaaS level Broker: Qu4DS

- **Contract types**
  - deployment and service execution
  - Fixed (or negotiated) time slot
  - Multiple service calls
  - SLA templates used for negotiation
    - Created by the service provider
    - Only Labels (no values)

- **Pricing model**
  - Pay-per-use
  - Function of expenses
  - Fines
  - No auctions
Work done

- An approach for specifying and enforcing SLAs
  - Creation of SLA templates + Pricing model
- Enforcement: QoS assurance mechanisms
  - Performance & Fault tolerance
- Validation
  - Prevention of losses for distinct fine costs

Lessons learned

- PaaS as a third party?
  - Allows to define better SLAs
    - IaaS do not provide realistic SLAs for business use
    - Works like an insurance mechanism for final users
  - Allows to enforce fault tolerance
    - Could also use more than one IaaS provider
  - Could generate profits
    - By underprovisionning resources (=overbooking)
Conclusion
Conclusion

- What is so new in these designs?
  - There should be a contract between any piece of software from Application to OS
  - These contracts are a place of choice for green parameters
- We still have a lot of work to design a Magic Green Broker
  - We need to re-think OS level software design
Thanks for your attention

Questions ?