Présentation de l'équipe ERODS

Efficient and RObust Distributed Systems

Vivien Quéma

Réunion PIMLIG

14 mars 2013

(basé sur une présentation de N. De Palma)

Cloud Computing

* Characteristics
  * Large scale virtualized infrastructure
  * On demand resource provisionning
  * Auto-scalability
* Benefits
  * Cost saving (pay as you use)
    * Service for a given period of time
    * Avoid full IT cost
  * Better availability and scalability
    * Basic support for replication, load-balancing and partitioning

Cloud Computing

* 3 delivery models
  * IaaS (e.g Amazon)
    * Register, provision, start, stop, destroy VM
    * Install servers though package manager
  * PaaS (e.g Heroku, PhpFog)
    * Provision, start, stop, destroy servers
    * Install application through server-level managers
  * SaaS (e.g SaleForces)
    * Ready to use application
    * End-user account and right through app-level manager

Cloud Computing

* 4 deployment models
  * Public
    * Everybody can provision VMs
  * Private
    * Own datacenter and cloud stack
    * Own purpose
  * Hybrid
    * Own datacenter and cloud stack
    * Extend the IT capacity with public cloud
  * Community
    * Service specific
    * Highly distributed
    * Classical aggregated ressource

Challenges

* Automation
  * To much management operations are performed by hand
  * Lack of observability and control
* Performance
  * Instable performances and SLA
* Robustness
  * Very dynamic, subject to reconfiguration and failure

Performance challenge

* Complex SLO and SLA
  * Multiple service level objectives (SLOs)
    * performance, availability, security, energy consumption, etc.
  * Trade-off antagonist SLOs
    * E.g. “at least 99% of client requests are admitted and processed within 1s”
* Different workloads, difficult to predict
* Need to map SLOs to resource allocation and configuration

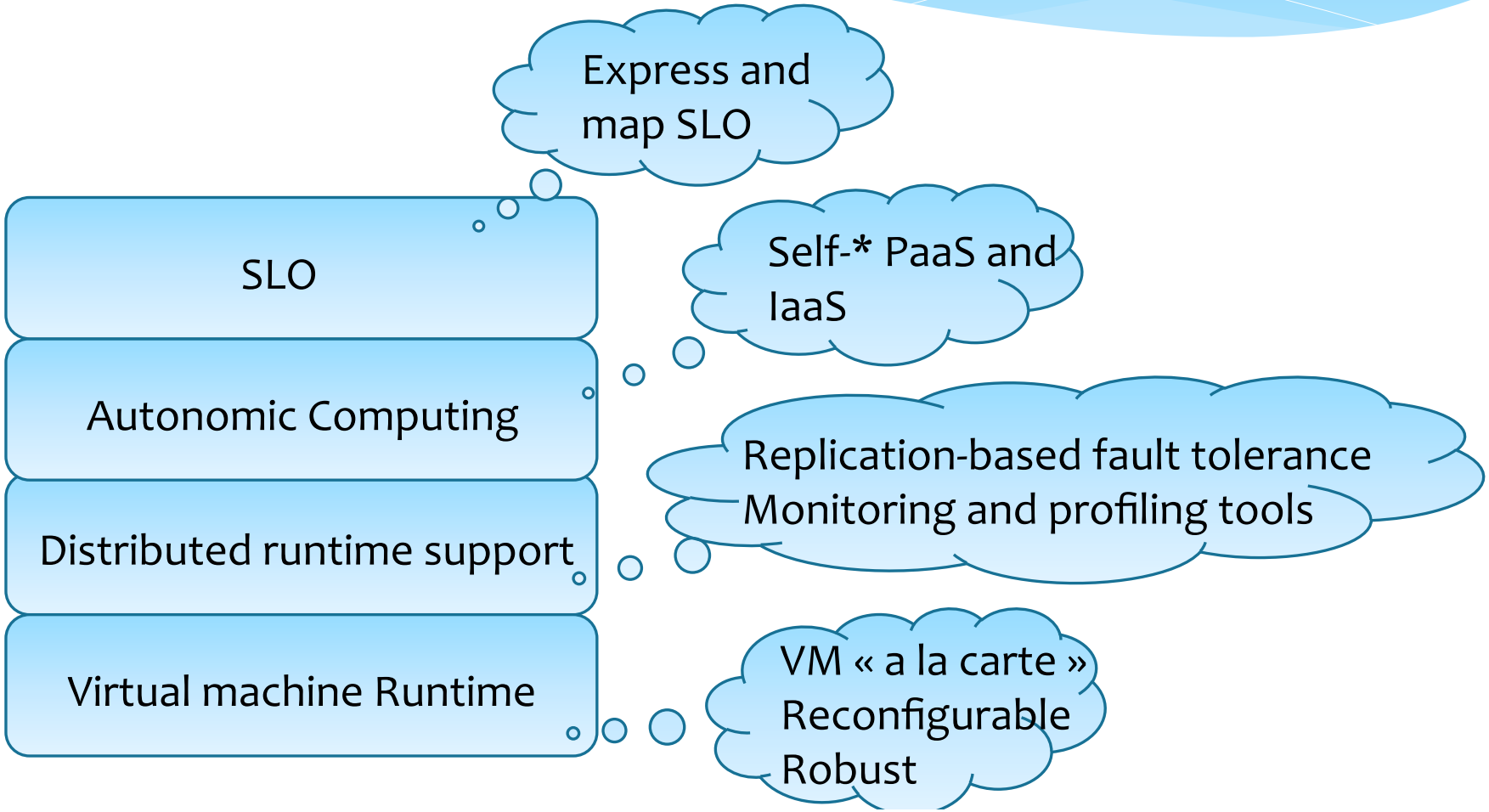
Automation challenge

* Complex distributed system and infrastructure
  * Multiple layers of management (IaaS/PaaS/SaaS)
  * The management system is also a distributed system
* Manual management
  * Configuration error
  * Low reactivity
  * Ressource overbooking
* Need to automate all common management tasks (e.g configuration, start-up, auto-scalability, update)

Robustness challenge

* Classical software stack not suited to resist failures and to face evolution
* Basic support for robustness
  * VM snapshooting and checkpointing
  * Basic replication protocols
    * Difficult to use in cloud (WAN, NAT...)
  * REST model with state in DB
    * SQL/NoSQL
* Need for reliable and reconfigurable application, container, VM

Our view: A reconfigurable, robust and efficient cloudware



Autonomic Computing

* Design Self-* properties for the cloud
  * Self-Configuration
    * Self configure the whole software stack
  * Self-Optimization
    * When scale up/down ?
    * How scale up/down ?
    * What scaling up/down ?
  * Self-Repair
    * IaaS-level repair
    * PaaS-level repair
    * Application-level repair
  * Coordination of Control loops

Autonomic Computing

* Leverage knowledge about the architecture of the application
* Decentralized self-management
  * Scalable, reliable, asynchronous distributed messaging system
  * Basic ability to add/remove self-configurable VM
* Control Theoric approach (with N. Marchand)
  * Nonlinear continuous-time model using fluid approximations
  * Model to scale up/down
* Discrete controler synthesis (with E. Rutten)
  * Synchronous programming
  * Invariant of control loops

Distributed runtime support

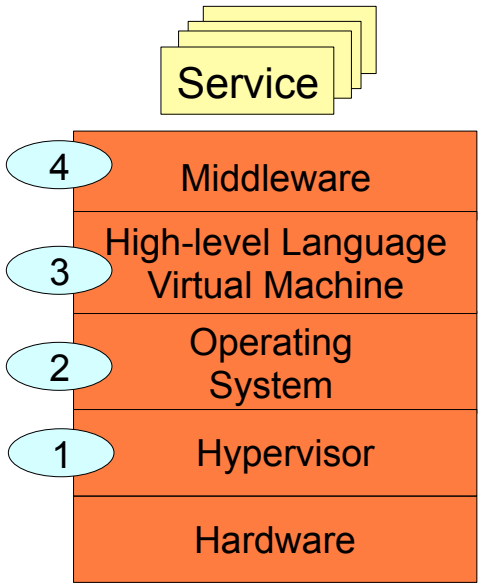
* Large-scale systems & replication-based fault-tolerance
  * How to design efficient replication protocols for WAN?
  * How to design robust replication protocols?
  * How to design multi-facet replication protocols?
* Runtime support for efficient distributed systems
  * How to understand the performance of networked (multicore) systems?

Distributed runtime support

* Large-scale systems & replication-based fault-tolerance
  * Both theoretical aspects:
    * Design accountable P2P systems and P2P supervision
    * Design leader-free consensus protocol
  * And practical aspects:
    * Design NAT-resilient P2P systems
* Runtime support for efficient distributed systems
  * Profiling tools (remote memory accesses and high-level correlation)

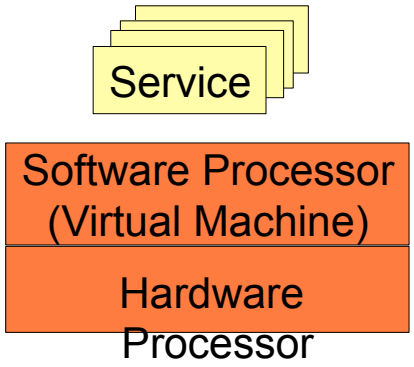
Virtualization in the cloud

* At different levels, with different goals
  * Enable hardware sharing and remote management
  * Enable software reuse
    * Existing operating systems
    * Existing high-level language virtual machines (JVM or CLR)
    * Middleware frameworks
* Enable new programming models
  * Service-oriented architecture
  * Quality and elasticity of services
* Our virtualization challenges
  * Green performance
  * Reconfigurable software
  * Robust execution



Virtualization in the Cloud

* Design a virtual machine
  * Collapse virtualization layers for a small and efficient runtime
  * Specifically designed for distributed service-oriented programming
* Define a "software processor"
  * Providing high-level, safe, and portable instruction set
  * With core mechanisms enabling modular and reconfigurable software
  * Include fault-tolerance and distribution from day one in the design
* Expected priorities amongst scientific challenges
  * End-to-end performance, with a reduced energy consumption
  * Mastering software reconfigurability
  * Overall robustness, helping tolerate partial failures
  * Helping with distributed programming



People

* Sara Bouchenak, MdC UJF
* Fabienne Boyer, MdC UJF
* Noel De Palma, Pr UJF
* Didier Donsez, Pr UJF
* Olivier Gruber, Pr UJF
* Renaud Lachaize, MdC UJF
* Nicolas Palix, MdC UJF
* Vivien Quéma, Pr Grenoble INP

People

* Phd
  * Xavier Etchevers. 3A
  * Loic Letondeur. 1A
  * Ahmed El-Redanne. 1A
  * Soguy mak-kare gueye. 1A
  * Baptiste Lepers, doctorant. 2A
  * Gautier Berthou, doctorant. 2A
  * Pierre-Louis Aublin doctorant. 2A
  * Amit Sangroya, doctorant. 1A
  * Amadou Diarra (en attente de validation par le fonctionnaire défense)

Summary

* Context
  * Cloud computing
* Challenges
  * Automation
  * Performances
  * Robustness
* Interests
  * Autonomic computing
  * Distributed runtime support
  * Virtual machine technologies