Soft Computing Models for Cloud Service Optimization

G. Albeanu,
***Spiru Haret* University**
&
Fl. Popentiu-Vladicescu
UNESCO Department, University of Oradea

Abstract

- The cloud computing paradigm has already proved a new way to do business in Information Technology field.
- The optimization of services (infrastructure, platform, software) according to different criteria is a necessity nowadays.
- This paper describes the usage of evolutionary approaches to optimize cloud computing services.

Soft Computing Models for Cloud Service Optimization

- Introduction to Cloud Computing
- Dynamic Voltage and Frequency Scaling
- Problem formulation
- Solving methods
- Comments
- Conclusions
- References

Introduction to Cloud Computing

- The NIST Definition of Cloud Computing:

  "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*"

- This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Cloud computing – essential characteristics (1, 2, 3)

- **1.On-demand self-service**: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- 2.Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)
- 3.Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. Examples of resources include storage, processing, memory, and network bandwidth.

Cloud computing – essential characteristics (4, 5)

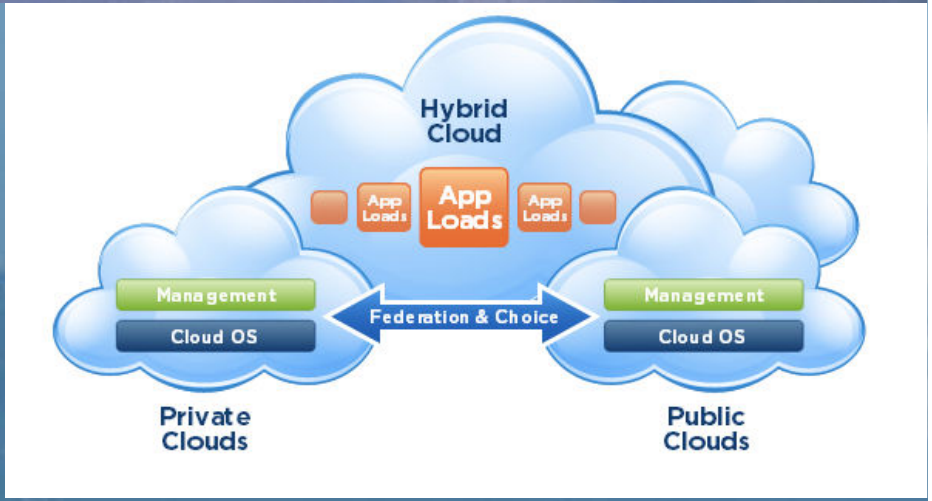
- 4. Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- 5. Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Cloud computing – service models

- Software as a Service (SaaS): The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. [Examples: *Google Apps, Microsoft Office 365, Innkeypos, Quickbooks Online, Limelight Video Platform etc*.]
- Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. [Examples: *Amazon Elastic Beanstalk, Cloud Foundry, Heroku, Google App Engine, Windows Azure Compute, Force.com etc*.]
- Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. [Examples: *Google Compute Engine, Windows Azure Virtual Machines, Amazon Cloud Formation, Rackspace Cloud, Verizon Terremark etc*.]

Cloud computing – Deployment models

- Private cloud: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units) - a cloud computing platform that is implemented within the corporate firewall, under the control of the IT department.
- Public cloud: The cloud infrastructure is provisioned for open use by the general public. It exists on the premises of the cloud provider.
- Hybrid cloud: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds)



http://www.vmware.com/files/pdf/VMware-Public-Cloud-Service-Definition.pdf

- Community cloud: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers (such as banks or heads of trading firms) from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations).

Cloud computing – SLA/Contract

- A Service Level Agreement (SLA) is a formal, negotiated document that defines (or attempts to define) in quantitative (and perhaps qualitative) terms the service being offered to a Customer. Any metrics included in a SLA should be capable of being measured on a regular basis and the SLA should record by whom.
- A Contract is a legally binding agreement between two or more parties. Contracts are subject to specific legal interpretations.

- EU Commission - SLAs can be viewed as negotiated "agreements" between different parties /entities. As "agreements", SLAs encapsulate a set of different aspects regarding the services provisioning. These refer to the agreed Quality of Service (QoS) – captured through different terms, the Service Level Objectives (SLOs), the responsibilities and obligations of the parties, as well as the penalties in cases of non-compliance to the agreed terms.
- SLA options: Best effort (Bronze) / High availability (Silver) / Fault tolerant (Gold

Cloud computing – Virtual machines

- A virtual machine (VM) is a software program or operating system that not only exhibits the behavior of a separate computer, but is also capable of performing tasks such as running applications and programs like a separate computer. A virtual machine, usually known as a guest is created within another computing environment referred as a "host." Multiple virtual machines can exist within a single host at one time. (http://www.techopedia.com/definition/4805/virtual-machine-vm)

Dynamic Energy Minimization using DVFS

- DVFS (Dynamic Voltage and Frequency Scaling) is an efficient energy saving technique for processors during program execution time.

The dynamic power consumption can be stated by the following equation.

$$P_{dyn} = N_{sw} C_L V_{dd}^2 f \quad (1)$$

In the equation, N_{sw} is the switching activity, C_L is the load capacitance, V_{dd} is the supply voltage, and f is the operating frequency. From (1), we can see that the power consumption is proportional to the product of the frequency and the square of the supply voltage. As a result, decreasing the voltage has a quadratic effect on the reduction of the power consumption.

Problem formulation: Notations

- The Cloud Platform consists of m>0 physical machines, denoted by M[1], M[2], …,M[m]. For machine M[j], its capacity c[j] (no. of instances of services) can be adapted by DVFS with energy E[j].
- E[j]=Estat(j)+Edyn(j), Edyn(j)=(e[j])(c[j])$^{\alpha}$, α in [2, 3].
- The objective is to run n services S[1], S[2], …, S[n] through a given time period given by SLA.
- To assure the necessary reliability, a number of instances (identical and independent) of every service S[i] will run concurrently on the same or different physical machine as Virtual Machine. Let A[i,j] the number of instances of S[i] running on M[j], A[j]=sum{A[i,j], i=1, 2, …, n}.

Problem formulation: Minimum Energy

Find the set of machines that are on (denoted by X, X⊆{1, 2, …, m}) and the clock frequency (f[j]) assigned to Machine M[j], represented by C[j] and an allocation A of instances (of services) to machines M[1], M[2], ..., M[m] such that

(i) for any j in X: A[j] leq C[j],

(ii) For all services S[i],

Probability (ActiveInstances of Service S[i] geq SLA[i]) geq (1 – REL[i]),

i.e. the probability that at least the number of instances of S mentioned by SLA[i], are running on alive machines after the time period is larger than the reliability requirement (i.e. 1 - REL[i]),

(iii) and the overall energy consumption Sum{E[j], j in X} is minimized.

Fault Tolerant Service Mapping on Physical Machines

Hypothesis:
All services ask for similar resources
A[1] = 1*S1+2*S2+1*S3+1*S4+1*S5+2*S6+2*S7+2*S8

C[1]=12	C[2]=8	C[3]=10	C[4]=5	C[5]=11	C[6]=9	C[7]=12
S2						S3
S7				S8		S1
S3		S4		S5		S6
S6		S7		S8	S1	S2
S8	S1	S2		S3	S4	S5
S2	S3	S4		S5	S6	S7
S4	S5	S6		S7	S8	S1
S5	S6	S7	S8	S1	S2	S3
S6	S7	S8	S1	S2	S3	S4
S7	S8	S1	S2	S3	S4	S5
S8	S1	S2	S3	S4	S5	S6
S1	S2	S3	S4	S5	S6	S7
M[1]	M[2]	M[3]	M[4]	M[5]	M[6]	M[7]

Solving Methods

- Greedy methods: First-FIT, Next-FIT, Best-FIT – **not necessary optimum solution**
- Backtracking enumeration – **too many combinations**
- Soft computing methods: a) nature-inspired methods for combinatorial optimization; b) evolutionary optimization by selection/mutation/crossover operators – **approximate solutions** but obtained in acceptable time horizon.

Evolutionary searching for X/A

- Start with an initial configuration – see the example.
- Use a Monte-Carlo strategy to generate an initial Population of ‘mutants’ of the initial configuration.
- Generate successively new populations by crossover and/or random mutations.
- Evaluate the energy required by the configuration members.
- Select only the members with power consumption below some threshold.
- Continue to generate populations until a stopping criterion is obtained (a number of iterations, an acceptable energy level).

Comments

- Technical details are related to: encode information to support the problem's data, definition of evolutionary operators like mutation and crossover in genetic algorithms, offline analysis of the final set of configurations and select those with minimum energy.
- The approach is efficient when contracts describe integer quantities of resources (time, internal memory, cloud storage etc.).

Conclusions

- Energy minimization under SLA contract based in cloud computing business is important to help clients to pay low prices.
- Smart allocation strategies of services/tasks to physical memories based on replicates running on the same machine or on other machines will increase software/hardware reliability in order to provide services over SLA mentioned by contracts.

References

- Peter Mell, Timothy Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145
- Dimosthenis Kyriazis, Cloud Computing Service Level Agreements - Exploitation of Research Results, European Commission, July 2013.
- Matthias Schmidt, Niels Fallenbeck, Matthew Smith, Bernd Freisleben, Efficient Distribution of Virtual Machines for Cloud Computing, http://www.fallenbeck.net/publications/sfsf10.pdf
- Liang W.-Y., Lai P.-T., Chiou C.W., An Energy Conservation DVFS Algorithm for the Android Operating System, http://www.ftrai.org/joc/vol1no1/Paper13-JoC_SectionE_3.pdf