



Dynamic Ranking of Cloud Providers

Paweł Czarnul

Dept. of Computer Architecture Faculty of Electronics, Telecommunications and Informatics Gdansk University of Technology

G. Narutowicza, 11/12, 80-233, Gdańsk, Poland WWW: http://pczarnul.eti.pg.gda.pl e-mail: pczarnul@eti.pg.gda.pl, phone + 48 58 347 12 88, fax. +48 58 347 28 63

Bled, Slovenia, 25th October 2012





Problem and Solution

- Problem: how to rank cloud offers (laaS, PaaS and SaaS) dynamically and prevent from the vendor lock-in issue?
- Solution:
 - build a quality model along with definition of quality measurement procedures
 - Develop a dynamic evaluation engine for IaaS, PaaS, SaaS
 - adopt several techniques known from already existing price comparison engines
 - filter measured data, in particular to avoid vendor lock-in issues
 - adopt full-text search mechanism





Available Cloud Offers

- laaS Infrastructure as a Service making an infrastructure (computing, storage, operating system) with a given configuration available to a client, examples: Google Compute Engine, Amazon Elastic Compute Cloud (EC22), RackSpace Cloud Servers
- PaaS Platform as a Service offering a complete platform with particular software required by users; examples: Aneka, Google AppEngine, Windows Azure, RedHat Openshift, RackSpace Cloud Sites
- SaaS Software as a Service particular software that is managed by its provider and accessed by users from any location. Examples include Google Apps and Salesforce



Quality Assessment of Cloud Offers

- The following aspects to be addressed:
 - quality model/ontology that defines metrics to be measured
 - quality measurement procedures e.g. how frequently the metrics should be measured, this may be different for various metrics such as availability and price
 - filters applied on top of the measured values:
 - preventing from short-term peaks in measured values to affect output
 - preventing from one or few providers to take top places all the time by offering too good to be true conditions
 - considering or not sudden changes in the history of the provider which may affect user decisions who might be afraid of similar changes in the future



Quality Ontology for Clouds

The ontology will incorporate the following:

- accessibility_location_x may characterize the network between the client in location_x and the service, several entries of this type could be inserted
- availability characterizes the availability of the service itself. It could be measured by e.g. checking its availability vs availability of other services/servers in a similar geographical/provider location
- cost-effectiveness evaluated by clients,
- reconfiguration ability applicable to laaS and PaaS,
- access how easy it is to access the infrastructure and upload/download/execute applications.





Filtering and Vendor Lock-in

- Application of the following techniques:
 - filters applied on top of measured values:
 - apply a low pass filter on the resulting measured metrics.
 For instance, a one time peak in measurements of a certain value might not change the overall score of the given metric.
 - It may depend on the client whether to rely more just on recent measurements or the history
 - to avoid vendor lock-in, consider a certain number of best offers and rotation on the first ranking places, provided that results returned for the services are closer to each other than a predefined threshold



Evaluation Engine for IaaS, PaaS and SaaS

- The ontology will incorporate the following:
 - categorization of features desired by the client (laaS, PaaS):
 - memory size,
 - processor/core capabilities, GPU capability,
 - Storage,
 - operating system, particular software,
 - access interface.
 - full text search that allows formulation of desired functions in the form of human readable text (SaaS, IaaS and PaaS)
 - → creation of runtime registries of particular laaS, PaaS and SaaS offers
 - \rightarrow distinguish SaaS by categories e.g. tags



Evaluation Engine for IaaS, PaaS and SaaS

- The solution will allow:
 - On the engine side:
 - Ensure that no single 'best' offer is returned at all times
 - Will consider the client requirements when searching for offers
 - On the client side the following requirements should be considered in the search:
 - Already selected (in the past) cloud providers
 - Consideration (or not) variations in performance of providers







THANK YOU