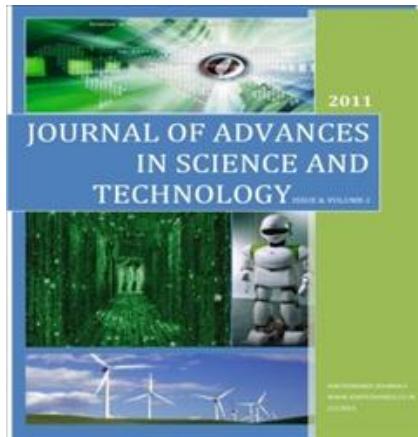


Current Cloud Architecture Technologies and Platforms Comparison



Kailash Aseri

M.Tech. (Cs) Scholar, Manav Bharti University,
Solan (H.P.)

ABSTRACT:-

GNITED MINDS

This paper presents a functional overview of popular cloud computing architectures. Background is given on the technical aspects of each architecture and parallels are drawn with the previous three chapters, whilst also discussing business concerns. As an introduction to the comparison overview, the platforms need to firstly be categorized based on their level of abstraction – whether the cloud architecture is IaaS, PaaS or a combination of both.

INTRODUCTION:

Cloud instances on hardware level (IaaS) are provided by Amazon EC2, GoGrid and Mosso Servers. Thereby only a handful of APIs are present on top for setting up the instances. This allows for running on-premises applications and common relational databases, which due to aforementioned scalability issues have only limited potential for elasticity. Alternatively, Amazon provides SimpleDB for building scalable storage programs, yet due to its proprietary nature it is not widely adopted and requires significant application reengineering.

hybrid IaaS/PaaS public clouds such as Microsoft Azure (and to an extent Sun Grid) allow for more API calls to native scalability functions. However, what goes on beneath the surface is largely abstracted (and mostly unknown, in Microsoft's case). In Microsoft Azure, programmers

are able to plan in for elasticity features in their applications (by deliberately using built-in functions where applicable) and allow for some level of automatic cloud failover/scalability optimization when running their application at Microsoft's datacenter/cloud, although 'normal' .NET applications are still possible to be executed, albeit without utilizing fully the platform's abilities (some applications may actually even run slower) cloud service PaaS such as Google AppEngine or SalesForce/Force.com are proprietary, domain-specific PaaS/SaaS (SaaS because they are delivered as a web application) that are arbitrarily programmable only to a limited extend and for specific business tasks. Google (much like Amazon) made parts of their own algorithms for highly elastic computing and storage publicly available, whereas the business development platform Force.com allows programmers using their Apex and even non-programmers using the visual GUI Visualforce to simply program customizable extensions to their main SaaS application SalesForce.com

Generally, when comparing platforms I suggest the reader to consider storage and compute capabilities differently. My research shows that most marketing materials, and cloud product descriptions do not actively force that differentiation between the compute and the storage cloud. Even some whit papers vaguely mention the topic, yet this differentiation is highly important for business considerations. Should a company's marketing or management get excited about 'moving IT to the cloud' they need to separate their storage from their computational needs in order to decide for a proper cloud architecture. Table 1 below presents some technical and functional aspects of five different cloud computing paradigms:

AMAZON WEB SERVICES. ELASTIC COMPUTE CLOUD, SIMPLE STORAGE SERVICE

Amazon Web Services is a brand of remote computing web services offered by Amazon Inc. Throughout this thesis Elastic Compute Cloud (EC2) was used to exemplify a hardware-close OS-level XEN virtualization cloud architecture that is priced by hourly usage of EC2 instance units (VMs of different hardware capabilities – a multiple of ~1,2 GHz Opteron/Xeon processors) as well data transfer fees. The largest instance is equivalent to 8 EC2-units, totals 15GB of RAM and 1690GB of hard disk space and supports 64-bit platforms. Linux, Sun OpenSolaris Microsoft Windows Server 2003 are supported, however instances running Windows cost more per hour due to licensing fees. Eucalyptus,

an open source project now supports standardized integration of Amazon's APIs into Linux distributions for the purposes of building clouds.

Amazon's Simple Storage Service (S3) is an online storage service that allows users to store unlimited amounts of data leveraging Amazon's own ecommerce infrastructure with pricing between \$0.120 and \$0.150 per GB monthly (European prices are ~20% higher). Additionally, data transfer charges (almost identical per GB/month storage pricing) and two types of HTTP requests (POST, PUT, etc. and GET separately) are billed separately. S3 leverages REST and SOAP protocols but also provides the BitTorrent P2P protocol to lower costs for high scale distribution. It has a relatively low uptime guarantee of 99.9% anchored in the S3 SLA.

Amazon SimpleDB is a distributed database for storing augmented key/blob structured data which scales automatically. A blob (Binary Large Object) is a schemaless unstructured data with varying contents. SimpleDB query execution time is limited to 5sec and operates through with a simplified SQL-like API. Still marked as "beta", SimpleDB is priced based on machine hours, data transfer and storage utilization and can

MICROSOFT AZURE SERVICES

Microsoft will offer the Azure Services runtime platform a for executing .NET applications in the cloud and will thereby sell hosting (compute) accounts and storage accounts itself. Authorization will be done via Windows Live ID. [Chappell_Az] A *fabric layer* will be present to abstract the VMs from application instances and it will assign automatically more instances/computing resources for elasticity purposes, as well as provide basic application failover/restart capabilities. The fabric layer will not allow developers to control the OS or the VMs directly thus I would categorize it in the PaaS class. VMs will run either in a *web role* or a *worker role*. A web role denotes starting an ASP.NET/IIS web application that handles network (HTTP) requests, whereas a worker role does not use the IIS (Microsoft's web server) and represents a batch background job started from a queue that can only have outgoing connections (to write results after a job is executed) and will be possible to be realized in any .NET language (C#, VB.NET, J# but also Ruby and Java using SDKs). Azure will store data in *tables* (of relational nature) and *blobs* which will be held in *containers* assigned to each customer account. SQL Data Services (a restricted view of SQL Server [Berk_ATC]) will manage the containers stored at different Microsoft datacenters and provide access to the storage data (see [Chappell_Az]).

GOOGLE APP ENGINE. BIGTABLE AND MAPREDUCE

Google App Engine, is a service offered by Google Inc to enable user-made applications to run on Google's own infrastructure via a large set of proprietary APIs. AppEngine is purely a PaaS that supports running restricted versions of Java and Python code with a 30 sec timeout and read-only file system capabilities. Application execution can only be invoked via HTTP requests. Storage is done in Datastore – a schemaless blob that is strongly consistent and that can be queried with very simple one-column WHERE clauses. AppEngine's built-in services include using the URL-fetching and mail-broadcasting services Google claims to use themselves as well as simple memory caching, scheduled tasks (cron jobs) and image manipulation functions for performing background batch jobs. Google accounts are required for setting up the applications and could also be used instead of programming user modules for user authentication. AppEngine supports limited free usage and over a certain quota (500MB storage and 5million requests per month) is priced per hourly CPU time, GB/month of storage/transfer and mails sent.

To understand how Google is able to deliver those services along with its whole set of projects I will introduce BigTable and MapReduce briefly – the technologies behind the vast amount of petabytes of data stored at Google's server and the ability to perform ultra fast searches over it. BigTable is a distributed, column-oriented, multi-dimensional sorted map that is able to run on thousands of physical machines and allow for extremely high consistency. Data is replicated on multiple machines so a hard disk failing of a given machine has no effect on bringing the whole system down, which potentially allows full consistency even if 30% of Google's servers were to fail at once (see [BigT] for details). MapReduce on the other hand is the programming model and implementation of processing BigTable's huge datasets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

Google's index of the WWW is regenerated using MapReduce and it allows for many parallel processing tasks in distributed applications. The map/reduce paradigm has since been implemented in many other projects. Please refer to [MapR] for a detailed description of it.

HADOOP AND YAHOO

Hadoop is the free and open source implementation of Google's primary infrastructure technologies – BigTable, MapReduce and Google's distributed file system (GFS). It is a top-level Apache Software

Foundation project implemented in Java that consists of a map/reduce engine, a distributed file system (HDFS), a job/task tracker and the column-oriented HBase schemaless database similar to BigTable. [http://wiki.apache.org/hadoop] Yahoo is the main contributer to the project and have recruited Doug Cutting (the original inventor) to lead Yahoo's cloud computing Hadoop efforts. [YahooH] In June 2009 Yahoo released their own distribution of Hadoop and they claim to run the largest Hadoop clusters worldwide. Hadoop is not only responsible for generating Yahoo's search index, but also employed by nearly any cloud computing stakeholder – Facebook, IBM and Rackspace (owners of Mosso Clouds) and many others. However, a closer look at Hadoop's site reveals that most of those companies use Hadoop for a limited number of background batch processing tasks – mostly analytics – log/click/ads analyses. The NY Times used Hadoop running on Amazon EC2 instances for a very large batch processing tasks (TIFF PDF conversion) that cost → under \$300 and converted all scanned articles from 1851-1922 to be made public domain. [NYT_HDP]

This clearly shows that after the platform gets developed to a mature level that enables it to be installed at any cloud provider – much like the LAMP (Linux-Apache-MySQL-PHP/Python) stack, the proprietary Microsoft/Google platform offerings could be challenged with an open system that provides similar advantages without locking the customers to a specific provider.

APPLE AND IPHONE IN THE CLOUD

Apple is not a usual follower of industry fashion, but in the context of cloud computing it employed a clouds architecture for storing user synchronization data (e-mail, contacts, calendars). MobileMe is provided as SaaS against a \$99/year subscription charge. [Maya_WrCl] However, the service experienced outages between 16-28 July 2008 that prompted users' emails to be lost and data not to be retrieved. [Newsweek08] This incident is to stress that real-time cloud architectures are apparently struggling with concurrency and consistency problems and are still highly exposed to partial outage at least and information loss and leakage at worst, and to emphasize that the results in reputation and revenue loss due to disgruntled customers are substantial. This applies in fact to all web-based services and not only the paid ones – e.g. also Gmail, Google Docs and other SaaS whose data is stored and updated in cloud architectures, spread on many servers running not yet fully researched distributed programming data manipulation techniques, are prone to problems that are not at all anticipated by the customers.

BUSINESS SOFTWARE CLOUDS – SALESFORCE.COM

SalesForce.com is a CRM software vendor founded in 1999 by Oracle executives. [Wikipedia] It delivers CRM SaaS using monthly subscription plans per user (from \$9 for the Group to \$250 for the Unlimited version). SalesForce.com, much like Amazon and Google utilizes a serious internal grid computing architecture and because new customers are allocated within this infrastructure it enables them to market their CRM as *software in the cloud*.

Co-branded with Force.com, customers are able to go beyond the delivered CRM and write their own (web) applications, that are however able to run only against Force.com's database and are written in Apex Code – SalesForce's own programming language. Apex Code's syntax is Java-like and allows for integration with web services APIs written in other languages. An isolated sandbox environment is available that serves as a test/development platform.

The Visualforce GUI application/workflow designer is provided in addition to enable code free creation of web UI using Force.com's (and Apex Code) components and mashups integration. The advantage of Force.com's applications is the ability to swiftly scale out, but the limitation of not being able to run any other applications as well as the need to SalesForce's proprietary database makes this not a real show case of cloud computing in my view. Force.com's PaaS is licensed for \$50-75 per user per month with limitations on database objects and storage space. SalesForce.com has however grown significantly and was added in Sep 2008 to the S&P500 index (after Freddie Mac and Fannie Mae).

CLOUD ARCHITECTURAL APPROACHES

Due to different device capabilities in cloud computing environment, e-Learning content adaptation and transformation need to be implemented before the content is presented to the user.

From an architectural point of view, four categories should be mentioned that represent the most significant distributed solutions for content adaptation like :

- i) *client-side* approaches,
- ii) *server-side* approaches,
- iii) *proxy-based* approaches and
- iv) *service-oriented* approaches.

CLIENT –SIDE APPROACHES

In a *client-side* approach, the transcoding process is the responsibility of the client application, as Figure-1 shows the use of client side approaches :



Figure -1 : shows the use of client side approaches

Client-side solutions can be classified into two main categories with different behaviors:

1. the clients receive multiple formats and adapt them by selecting the most appropriate one to play-out, or
2. the clients compute an optimized version from a standard one.

This approach suggests a distributed solution for managing heterogeneity, supposing that all the clients can locally decide and employ the most appropriate adaptation to them.

SERVER-SIDE APPROACHES

In a *server-side* approach, the server (that provides contents) performs the additional functional of content adaptation [30] [80] (Figure -2). In such an approach, content adaptation can be carried out in an *offline* or *on-the-fly* fashion



Figure-2 : server-side approach

In the former, content transcoding is performed whenever the resource is created (or uploaded on the server) and a human designer is usually involved to hand-tailor the contents to different specific profiles. Multiple formats of the same resources are thus stored on the server and they are dynamically selected

to match client specifications. In all the on the fly solutions, adapted contents are dynamically produced before delivering them to the clients.

PROXY-BASED APPROACH

In *proxy-based* approaches, the adaptation process is carried out by a node (i.e. the proxy) placed between the server and the client [30] (Figure-3). In essence, the proxy captures replies by the server to the clients requests and performs three main actions:

1. It decides whether performance enhancements are needed.
2. It performs content adaptations.
3. It sends the adapted contents to the client.

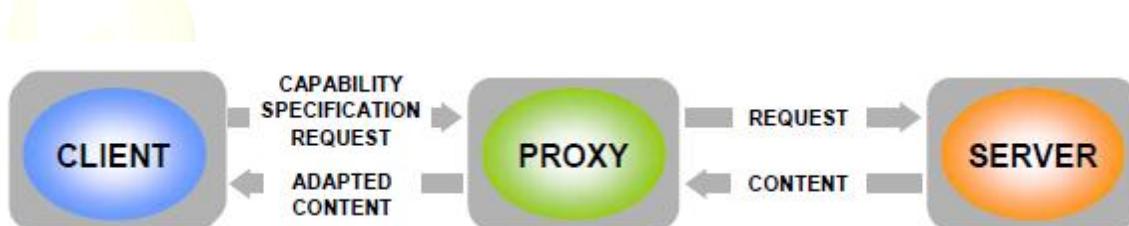


Figure -3 : Proxy based approach

To accomplish this task as a whole, the proxy must know the target device, the user capabilities (this information must be received from the client) and a “full” version of the original contents (this data must be received from the server). As a consequence, the use of network bandwidth could be intensive in the network link between the proxy and the server.

SERVICE-BASED APPROACH

The dynamic nature of adaptation mechanisms together with emerging opportunities offered by the new Web Service technologies, now provide a new approach of *service-oriented* content adaptation [30] (see Figure -4).

The philosophy at the basis of these approaches is fundamentally different from those previously discussed, since the transcoding and the adaptation activities are organized according to a service-oriented architecture. Indeed, the number of content adaptation typologies, as well as the set of multiple formats and related conversion schemes is still increasing. This dynamism is one of the reasons that

makes it difficult to develop a single adaptation system that can accommodate all the types of adaptations; therefore, third-party adaptation services are important.



Figure-4 : Service based approach

The Internet Content Adaptation Protocol (iCAP) [37] is closely related to this approach. ICAP distributes Internet-based content from the origin servers, via proxy caches (iCAP clients) to dedicated iCAP servers. For example, simple transformations of content can be performed near

the edge of the network instead of requiring an updated copy of an object from an origin server, such as a different advertisement by a content provider, every time the page is viewed. Moreover, it avoids proxy caches or origin servers performing expensive operations by shipping the work off to other (iCAP) servers. However, it only defines a method for forwarding HyperText Transfer Protocol (HTTP) messages, i.e. it has no support for other protocols and for streaming media (e.g. audio/video) and only covers the transaction semantics and not the control policy.

REFERENCES

1. AHMED, S., BURAGGA, K. & RAMANI, A. K. Year. Security issues concern for E-Learning by Saudi universities. In, 2011. IEEE, 1579-1582.
2. AL-JUMEILY, D., WILLIAMS, D., HUSSAIN, A. & GRIFFITHS, P. 2010. Can We Truly Learn from A Cloud Or Is It Just A Lot of Thunder? *2010 Developments in E-systems Engineering*, 131-139.
3. ANGEL_LEARNING. 2011. *Application Hosting Services* [Online]. Available: http://www.angellearning.com/services/application_hosting.html [Accessed July 25 2011].
4. ANGEL_LEARNING. 2011. *Standards Leadership* [Online]. Available: <http://www.angellearning.com/products/lms/standards.html> [Accessed July 25 2011].

5. ANGEL_LEARNING. 2011. *Technology and Systems Integration* [Online]. Available: http://www.angellearning.com/products/lms/tech_systems.html [Accessed July 25 2011].
6. BENEDIKTSSON, D. 1989. Hermeneutics: Dimensions toward LIS Thinking. *Library and information science research*, 11, 201-34.
7. BLACKBOARD. 2011. *About Bb* [Online]. Available: <http://www.blackboard.com/About-Bb/Company.aspx> [Accessed July 25 2011].
8. BLACKBOARD. 2011. *Association Clients* [Online]. Available: <http://www.blackboard.com/Markets/Associations/Clients.aspx> [Accessed July 25 2011].
9. BLUMBERG, B., COOPER, D. R. & SCHINDLER, P. S. 2005. *Business research methods*, McGraw-hill education.
10. BOEIJE, H. 2002. A purposeful approach to the constant comparative method in the analysis of qualitative interviews. *Quality and Quantity*, 36, 391-409.
11. BOTT, E. 2011. *Google's Blogger outage makes the case against a cloud-only strategy* [Online]. www.zdnet.com. Available: <http://www.zdnet.com/blog/bott/googles-blogger-outage-makes-the-case-against-a-cloud-only-strategy/3300> [Accessed May 13 2011].
12. CARLIN, S. & CURRAN, K. 2011. Cloud Computing Security. *International Journal of Ambient Computing and Intelligence*, 3.
13. CASQUERO, O., PORTILLO, J., OVELAR, R., ROMO, J. & BENITO, M. 2010. Strategy approach for eLearning 2.0 deployment in Universities. *Digital Education Review*, 1-8.
14. CHISNALL, P. M. 1981. *Marketing research: analysis and measurement*, McGraw-Hill London.
15. CHOW, R., GOLLE, P., JAKOBSSON, M., SHI, E., STADDON, J., MASUOKA, R. & MOLINA, J. Year. Controlling data in the cloud: outsourcing computation without outsourcing control. *In*, 2009. ACM, 85-90.
16. COOPER, D. R., SCHINDLER, P. S. & SUN, J. 1998. *Business research methods*, Irwin/McGraw-Hill Burr Ridge, IL.

17. DHILLON, G. & BACKHOUSE, J. 2000. Technical opinion: Information system security management in the new millennium. *Communications of the ACM*, 43, 125-128.
18. DOCEBO. 2011. *Docebo E-Learning solutions* [Online]. Available: http://www.docebo.com/cms/home_elearning_lms_multimedia_courses [Accessed July 25 2011].
19. DOCEBO. 2011. *DoceboLMS E-Learning Platform* [Online]. Available: http://www.docebo.com/cms/page/61/Docebo_LMS_learning_system [Accessed July 25 2011].
20. DOCEBO. 2011. *DoceboLMS Features* [Online]. Available: http://www.docebo.com/files/brochure/DoceboLMS_Features_ENG.xls [Accessed July 25 2011].
21. DOCEBO. 2011. *E-Learning solutions overview* [Online]. Available: http://www.docebo.com/cms/page/59/Elearn_and_Online_learning_solutions [Accessed July 25 2011].
22. DOCEBO. 2011. *Why choose DoceboLMS?* [Online]. Available: http://www.docebo.com/community/doceboCms/set-language_English_language-english.html [Accessed July 25 2011].
23. DOWNES, S. 2006. E-learning 2.0. *eLearning magazine: education and technology in perspective*, <http://elearnmag.org/subpage.cfm>, 29-1.
24. EAVES, M., MACLEAN, H., HEPPELL, S., PICKERING, S., POPAT, K. & BLANC, A. 2007. Virtually There: Learning Platforms. Scunthorpe: Yorkshire and Humber Grid for Learning Foundation/Chelmsford: Cleveratom.
25. EL-KHATIB, K., KORBA, L., XU, Y. & YEE, G. 2003. Privacy and security in e-learning. *International Journal of Distance Education Technologies*, 1, 1-19.
26. FOSTER, I., ZHAO, Y., RAICU, I. & LU, S. Year. Cloud computing and grid computing 360-degree compared. *In*, 2008. Ieee, 1-10.
27. GAMMACK, J., HOBBS, V. & PIGOTT, D. 2006. *The book of informatics*, Nelson Australia.
28. HOLLISTER, S. 2011. *Gmail accidentally resetting accounts, years of correspondence vanish into the cloud?* [Online]. www.zdnet.com. Available: <http://www.engadget.com/2011/02/27/gmail-accidentally-resetting-accounts-years-of-correspondence-v/> [Accessed Feb 27 2011].

29. HU, Z. & ZHANG, S. Year. Blended/hybrid course design in Active Learning Cloud at South Dakota State University. *In*, 2010. IEEE, V1-63-V1-67.

30. IWEBTOOL. 2011. *What is End User?* [Online]. Available: http://www.iwebtool.com/what_is_end_user.html [Accessed July 25 2011].

31. JAMIL, D. & ZAKI, H. 2011. CLOUD COMPUTING SECURITY. *International Journal of Engineering Science and Technology (IJEST)*.

32. JENSEN, M., SCHWENK, J., GRUSCHKA, N. & IACONO, L. L. Year. On technical security issues in cloud computing. *In*, 2009. IEEE, 109-116.

33. KUMAR, A., PAKALA, R., RAGADE, R. & WONG, J. Year. The virtual learning environment system. *In*, 1998. IEEE, 711-716 vol. 2.

34. LAISHENG, X. & ZHENGXIA, W. Year. Cloud Computing: A New Business Paradigm for E-learning. *In*, 2011. IEEE, 716-719.

35. LI, J. Year. Study on the Development of Mobile Learning Promoted by Cloud Computing. *In*, 2010. IEEE, 1-4.

36. MOODLEROOMS. 2011. *About Moodlerooms* [Online]. Available: <http://www.moodlerooms.com/company/about-us/> [Accessed July 25 2011].

37. MOODLEROOMS. 2011. *Create an Secure, Collaborative Training Environment* [Online]. Available: <http://www.moodlerooms.com/markets/government-and-nonprofit/uses/> [Accessed July 25 2011].

38. MOODLEROOMS. 2011. *Hosting - Moodlerooms on the Cloud* [Online]. Available: <http://www.moodlerooms.com/lms-solutions/services/hosting-service/> [Accessed July 25 2011].

39. MOODLEROOMS. 2011. *LMS Solutions* [Online]. Available: <http://www.moodlerooms.com/lms-solutions/> [Accessed July 25 2011].

40. NEEDHAM, R. M. & SCHROEDER, M. D. 1978. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21, 993-999.

41. NEUMAN, W. L. 2003. *Social research methods: Qualitative and quantitative approaches*, Allyn and Bacon.
42. OATES, B. J. 2006. *Researching information systems and computing*, Sage Publications Ltd.
43. PATIL, S. & SHINDE, G. Year. Transforming Indian higher education through blended learning approach. *In*, 2010. IEEE, 145-148.
44. PATTON, M. Q. 2002. *Qualitative research and evaluation methods*, Sage Publications, Inc.
45. POCATILU, P., ALECU, F. & VETRICI, M. Year. Using cloud computing for E-learning systems. *In*, 2009. World Scientific and Engineering Academy and Society (WSEAS), 54-59.
46. POCATILU, P., ALECU, F. & VETRICI, M. 2010. Measuring the efficiency of cloud computing for e-learning systems. *WSEAS Transactions on Computers*, 9, 42-51.
47. POPOVIC, K. & HOCENSKI, Z. Year. Cloud computing security issues and challenges. *In*, 2010. IEEE, 344-349.
48. RAITMAN, R., NGO, L., AUGAR, N. & ZHOU, W. 2005. Security in the online e-learning environment.
49. RAO, N. M., SASIDHAR, C. & KUMAR, V. S. 2010. Cloud Computing Through Mobile-Learning. *computing*, 1.
50. RICOEUR, P. 2004. *The conflict of interpretations: Essays in hermeneutics*, Continuum Intl Pub Group.
51. SCARFONE, K., JANSEN, W. & TRACY, M. 2008. Guide to General Server Security. *NIST Special Publication*, 800, 123.
52. STRAUSS, A. & CORBIN, J. M. 1990. *Basics of qualitative research: Grounded theory procedures and techniques*, Sage Publications, Inc.
53. SVENSSON, M. 2001. e-Learning standards and technical specifications. *Disponible on-line: <http://www.luvit.com> (Febrero 2003)*.
54. VAN HARMELEN, M. Year. Personal learning environments. *In*, 2006. Citeseer, 815-816.

55. VAQUERO, L. & RODERP-MERINO, L. January 2009. A Break in the Clouds: Towards a Cloud Definition. *ACM SIGCOMM Computer Communications Review*, Vol 39 (1), P50-55.
56. VERBEEK, P. P. 2003. Material hermeneutics. *Techne*, 6, 91-96.
57. WELSH, E. T., WANBERG, C. R., BROWN, K. G. & SIMMERING, M. J. 2003. E learning: emerging uses, empirical results and future directions. *International Journal of Training and Development*, 7, 245-258.

