2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing

Expanding the Cloud: A component-based architecture to application deployment on the Internet

Mark Wallis, Frans Henskens, Michael Hannaford
Distributed Computing Research Group
University of Newcastle
Newcastle, Australia
mark.wallis@studentmail.newcastle.edu.au

***Abstract*—Cloud Computing allows us to abstract distributed, elastic IT resources behind an interface that promotes scalability and dynamic resource allocation. The boundary of this cloud sits outside the application and the hardware that hosts it. For the end user, a web application deployed on a cloud is presented no differently to a web application deployed on a stand-alone web server. This model works well for web applications but fails to cater for distributed applications containing components that execute both locally for the user and remotely using non-local resources.**

This research proposes extending the concept of the cloud to encompass not only server-farm resources but all resources accessible by the user. This brings the resources of the home PC and personal mobile devices into the cloud and promotes the deployment of highly-distributed component based applications with fat user interfaces. This promotes the use of the Internet itself as a platform. We compare this to the standard Web 2.0 approach and show the benefits that deploying fat-client component based systems provide over classic web applications. We also describe the benefits that expanding the cloud provides to component migration and resources utilisation.

***Keywords*-cloud computing; component-based architecture; internet;**

I. Introduction

A key aspect of Cloud Computing in the web application space is that the user is abstracted away from the elastic resources and distributed communication which occurs within the cloud [1]. This abstraction allows companies to dynamically expand, contract and migrate their computation and storage tasks between various distributed nodes, without the user experiencing any disruption. With the rise of Web 2.0, fully-fledged "web applications" that replace classic fat-client PC applications are increasing in popularity [2]. This move has given rise to highly complex web browsers requiring plugins and extensions to support dynamic, interactive user interfaces. Such user interfaces have always been available to fat-client applications without this level of complexity. This increase in web browser complexity can be tied back to the rise in security incidents related to web browser functionality [3].

This research presents the idea of extending the boundary of the cloud to encompass not only server resources but all the resources available to the user. The user interface into applications deployed on the cloud no longer needs to be a web browser. Instead, a fat-client style component with full migration and persistence support provides the user interface. This component then communicates back to other components deployed in the cloud to provide application features such as data storage and computational functions.

The remainder of this paper firstly reviews the current problems in the Web 2.0 and cloud computing space in section II. A high-level concept is then presented in section III detailing extension of the boundaries of the cloud aimed at component-based application architecture, as opposed to classic Web 2.0 web deployments. Future work in this field is then detailed for further discussion in section IV.

II. Problem Description

The sheer number of reported security issues [4] that relate to the current generation of web browsers bodes the question, "Why are web browsers so insecure?". Web browsers, like the Internet itself, have evolved greatly over the past 20 years from their original roots in the academic community to global deployment for the everyday user. At its core, web browser technology is still built on a client-server architecture where often unauthenticated content is requested by the user and executed on the clients. Security has always been added on as an after-thought to the design. The Internet, and as such web browsers, were never designed to be secure from the ground up [5], hence as we try and abuse the web browser architecture by requiring it to achieve more and more tasks in a Web 2.0 world, the problems are potentially being made worse.

While browser rewrites from the ground up [6] are attempting to address some of the concerns, they fail to address the key issue that the purpose of a web browser is to allow the user to download and execute a wide range of unauthenticated content from insecure sources. As the dynamic nature of this content increases the security implications also increase.

The following proposed system moves away from the current web application design and instead suggests the deployment of component-based applications on the Internet

978-0-7695-4039-9/10 $26.00 © 2010 IEEE
DOI 10.1109/CCGRID.2010.14

using an extended version of the Cloud Computing concept to provide an execution environment.

III. Proposed System

This research proposes a new design for deploying applications on the Internet. By designing applications according to a component-based architecture we can treat the Web itself as a distributed platform for execution. Take for example a simple personal calendar application. Such an application can be composed of a user interface component and a data storage component. By extending the concept of the cloud to encompass all computing resources available to the user the data storage component can execute either within a classic cloud environment (i.e. a remote server farm) or on a local server managed by the user themselves. An abstracted execution environment allows this data storage component to migrate to a remote server farm at a later date if the user wished to offload the storage management. Inversely, the user-interface component can migrate between resources managed by the user. With the appropriate runtime extraction the user-interface is able to migrate from the user's desktop to their mobile device, and back again, without losing state. The ability to store data either locally or remotely in a transparent fashion will greatly help address issues raised in our previous work in personal data storage on the Internet [7].

While such distributed systems have previously been built on middleware layers [8] and virtual machines [9], we propose to instead implement the required enhancements at the Operating System level. Using well documented APIs this leaves the host-specific implementation up to the operating system vendor. A byte-code style layer allows code execution across multiple hardware platforms, in much the same way as Java and .NET currently abstract their compilation.

Only some of these benefits can be achieved using existing systems such as Service Orientated Architectures (SOA). While SOA addresses some of the concerns presented, it was not designed to treat the Internet as an execution platform. As such, it does not offer such features as component migration made available by our proposed extensions.

IV. Future Work

Research into this field is currently being undertaken from the following aspects by the Distributed Computing Research Group at the University of Newcastle.

1) Investigation into a common byte-code layer which allows cross-platform code execution while allowing operating system vendors to implement the system hyper-calls in a hardware specific fashion
2) Development of an extended Enterprise Service Bus architecture for component communication within the expanded cloud which can cater for mobile and distributed components executing both within server farms and remotely in the user's environment
3) Security design from the ground-up that promotes digitally signing each component-to-component call to allow the authorisation of all content executed by the user
4) A dynamic component library approach to software deployment that will provide the same level-of-service currently offered by SAAS [10] solutions
5) Design patterns for building systems based on component architectures with distributed communication methods

References

[1] G. Boss, P. Malladi, D. Quan, L. Legregni, and H. Hall, "Cloud computing," IBM Corporation, Tech. Rep., October 2007. [Online]. Available: http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf

[2] T. O'Reilly, "What is web 2.0," *O'Reilly Net*, 2005. [Online]. Available: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[3] J. Ryder, "Castles built on sand: Why software is insecure," *SecurityFocus*, 2002.

[4] WebDevout. (2009, November) Web browser security statistics, http://www.webdevout.net/browser-security. [Online]. Available: http://www.webdevout.net/browser-security

[5] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the internet," *Internet Society, Histories of the Internet*, 2003. [Online]. Available: http://www.isoc.org/internet/history/brief.shtml

[6] C. Reis, A. Barth, and C. Pizano, "Browser security: lessons from google chrome," *Commun. ACM*, vol. 52, no. 8, pp. 45–49, 2009.

[7] M. Wallis, F. Henskens, and M. Hannaford, "Publish/subscribe model for personal data on the internet," in *6th International Conference on Web Information Systems and Technologies (WEBIST-2010)*. INSTICC, 2010.

[8] A. Puder, K. Ramer, and F. Pilhofer, *Distributed Systems Architecture: A Middleware Approach*. Morgan Kaufmann, 2005.

[9] E. G. Sirer, R. Grimm, A. J. Gregory, and B. N. Bershad, "Design and implementation of a distributed virtual machine for networked computers," in *SOSP '99: Proceedings of the seventeenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 1999, pp. 202–216.

[10] K. Bennett, P. Layzell, D. Budgen, P. Brereton, L. Macaulay, and M. Munro, "Service-based software: the future for flexible software," in *Seventh Asia-Pacific Software Engineering Conference (APSEC'00)*, vol. 17th, 2000, p. 214.