

Cloud Computing: Does Nirvana Hide behind the Nebula?

Hakan Erdogmus

The buzzword-laden realm of cloud computing is upon us. Actually, according to *The Economist* (Oct. 2008), the hype surrounding cloud computing has already peaked. Regardless of the cumulus aura surrounding the topic, we're still better off not ignoring it. The trend might well represent a tipping point in a progression of repeated and insistent attempts to push, tweak, and reincarnate in new disguises an important core idea. Perhaps that idea's time has come. It looks like it's finally catching on.



At the core of cloud computing is a simple concept: software as a service, or SaaS. Whether the underlying software is an application, application component, platform, framework, environment, or some other soft infrastructure

for composing applications to be delivered as a service on the Web, it's all software in the end. But the simplicity ends there. Just a step away from that core, a complex concoction of paradigms, concepts, and technologies envelops cloud computing.

A Cloud of Definitions

For many, cloud computing represents everything new, cool, and trendy on the Web. But I need something a bit more specific. According to Wikipedia, cloud computing refers to the use of scalable, real-time, Internet-based IT services and resources, incorporating beyond SaaS many key technology trends of the 2000s. These trends include—and here comes the buzzword shower—service-oriented architecture, application service provision, Web 2.0, Web services, mash-ups, utility computing, autonomic computing, grid computing, on-demand computing, and so on. The Wikipedia entry stresses reliance on

the Internet for satisfying a broad spectrum of users' computing needs in a way that is independent of the soft and hard infrastructure supporting those needs. "The cloud element of cloud computing derives from a metaphor used for the Internet, from the way it is often depicted in computer network diagrams, and is an abstraction for the complex infrastructure it conceals," the entry adds.

Matthieu Hug's alternative definition on InfoQ (www.infoq.com/articles/will-meis-replace-extranets) is slightly more pointed in mentioning the underlying physical infrastructure: "an emerging computing paradigm where data and services reside in massively scalable data centers and can be ubiquitously accessed from any connected devices over the Internet." A more elaborate definition by Sam Johnston (<http://samj.net/2008/07/cloud-and-cloud-computing-consensus.html>) is cited in some blog entries and in a thick think-tank report on the future of software.

This plethora of definitions prompted me to go straight to a cloud-computing pundit for the real story: Marin Litoiu of York University. Litoiu was formerly with IBM Toronto Labs, where his interest in cloud computing started. At IBM he founded the Centre of Excellence for Research in Adaptive Systems to investigate key challenges residing in the cloud. According to Litoiu, cloud computing is an emerging computational model in which applications, data, and IT resources are provided as services to users over the Web (the so-called "cloud"). The main drivers of this model are economics and simplification of software delivery and operation. Litoiu echoes many others that cloud computing represents a natural evolution of the Web, the convergence of a long list of technology and research trends beginning with SaaS. So it's not surprising that the definition might make people invoke something they've already seen. "More and more types

Mission Statement:

To be the best source of reliable, useful, peer-reviewed information for leading software practitioners—the developers and managers who want to keep up with rapid technology change.

of applications are moving to the Web. Even the traditional desktop applications, like office software or software development tools, are now offered over the Web and accessed through a browser or through a remote desktop protocol," Litoiu says. He predicts that soon not a lot of software will be left on our desktops.

The Physics and Chemistry of the Cloud

At the top of the long list of technologies underlying the cloud are vast data centers and server farms, with virtual applications running on top of that physical infrastructure.

The Economist ("Down on the Server Farm," May 2008) reported that Internet companies are prepping their hardware resources by acquiring ever greater amounts of computing power. The report estimates that Google has more than a million servers in over 30 data centers across its global network. Microsoft, it says, is investing billions to grow its own physical infrastructure at a whopping rate of 20,000 servers a month. Thus the espoused massive decentralization at the periphery of the cloud is driving massive centralization at its deep center to take advantage of economies of scale in computing power, energy consumption, cooling, and administration.

As for the chemistry, the offerings—the virtual applications served out to users—have been made viable by recent advances in virtualization technologies. Litoiu defines virtualization as the users' insulation from actual resources in the cloud and from the complex, distributed infrastructure supporting those resources. Such insulation gives users the illusion, comfort, and convenience of swimming in much-more-familiar local waters, their own desktop environment. "We can think of the cloud as being a huge Internet data center, in which hardware resources are virtualized, offering a variety of services to the end users." Litoiu guesses the latest advances in virtualization technologies, combined with the increasingly acute realization of the economic burden of maintaining proprietary IT infrastructures, will push cloud computing over the mainstream adoption hump.

Software as a Service

Let's go back to the core, the SaaS concept, for a minute. In its full generality, SaaS refers not only to software delivered as a service

over the Internet but also to bootstrapping that very idea for achieving its goals. Here we go meta, then meta-squared. If an application can be a service, why not the platform on which that application is deployed? And if the platform can be a service, then why not the environment for creating the application on that platform? And why not your whole desktop? And why not designing your desktop environment? And so on ad infinitum. Hence the terms platform as a service, infrastructure as a service, desktop as a service, and others yet to be invented.

Pure SaaS, simply application delivery as an on-demand service, is no longer novel. The popularity of canned application services provided, for example, by Google Apps and Zoho suggests the concept is here to stay. I talked about the advantages and caveats of SaaS in my column "On-Demand Enterprise Services: Where's the Catch?" in the July/August 2007 issue of *IEEE Software*. Those trade-offs are widely known and aren't worth elaborating further. The benefits of scalability, reliability, security, ease of deployment, and ease of management for customers, traded off against worries of trust, privacy, availability, performance, ownership, and supplier persistence, still stand. Points listed as the benefits for the customer are now problems for somebody else—suppliers—who are better positioned to leverage their core technical competencies in those areas. The same arguments pretty much apply to the metalevels as well—to platform as a service, desktop as a service, and so on. Litoiu believes the cited concerns and weaknesses, although warranted for an uncontrollable public cloud, can be mitigated and become strengths for private clouds controlled by individual organizations or federations of organizations for the exclusive pleasure of their own user bases. What's unclear is how the scale and scope economies would play out and where the turning points happen as the SaaS and its metaparadigms scale down.

Implications for Software Development

How will cloud computing affect software development? This shifts the focus from pure SaaS to platform as a service and software infrastructure as a service. In my 2007 column on on-demand services, I was cynical about the claims that pushing application development, or application

EDITOR IN CHIEF

Hakan Erdogmus

hakan.erdogmus@computer.org

EDITOR IN CHIEF EMERITUS:

Warren Harrison, Portland State University

ASSOCIATE EDITORS IN CHIEF

Computing Now: Maurizio Morisio, Politecnico di Torino; maurizio.morisio@polito.it

Design/Architecture: Uwe Zdun, Vienna Univ. of Technology; zdun@infosys.tuwien.ac.at

Development Infrastructures: Martin Robillard, McGill University; martin@cs.mcgill.ca

Distributed and Enterprise Software:

John Grundy, University of Auckland; john-g@cs.auckland.ac.nz

Empirical Results: Forrest Shull, Fraunhofer Center for Experimental Software Engineering, Maryland; fshull@fc-md.umd.edu

Human and Social Aspects: Helen Sharp, The Open University, London; h.c.sharp@open.ac.uk

Management: John Favaro, INTECS; john@favaro.net

Processes and Practices: Frank Maurer, University of Calgary; maurer@cpsc.ucalgary.ca

Programming Languages and Paradigms: Laurence Tratt, Bournemouth University; laurie@tratt.net

Quality: Annie Combelles, DNV/Q-Labs; annie.combelles@dnv.com

Requirements: Ann Hickey, University of Colorado at Colorado Springs; ahickey@uccs.edu

DEPARTMENT EDITORS

On Architecture: Grady Booch, IBM

Architecture Practice: Frank Buschmann, Siemens

Bookshelf: Art Sedighi, SoftModule

Career Development: Philippe Kruchten, University of British Columbia

Design: Rebecca J. Wirfs-Brock, Wirfs-Brock Associates

Loyal Opposition: Robert Glass, Computing Trends

Requirements: Neil Maiden, City University London

Software Technology: Christof Ebert, Vector

Tools of the Trade: Diomidis Spinellis, Athens Univ. of Economics and Business

User Centric: Jeff Patton, consultant

Voice of Evidence: Forrest Shull, Fraunhofer Center for Experimental Software Engineering

ADVISORY BOARD

Frances Paulisch, Siemens (Chair)
Pekka Abrahamsson,
VTT Technical Research Centre of Finland
Jennitta Andrea, ClearStream Consulting
Elisa Baniassad, Chinese University of Hong Kong
J. David Blaine, consultant
Kaoru Hayashi, SRA
Simon Helsen, IBM Rational
Juliana Herbert, Herbert Consulting
Gregor Hohpe, Google
Gargi Keeni, Tata Consultancy Services
Karen Mackey, Cisco Systems
Steve McConnell, Construx Software
Erik Meijer, Microsoft
Grigori Melnik, Microsoft
Bret Michael, Naval Postgraduate School
Ann Miller, University of Missouri, Rolla
Deependra Moitra, consultant
Linda Rising, consultant
Wolfgang Strigel, consultant
Dave Thomas, Bedarra Research Labs
Markus Völter, consultant

composition in the new lingo, to end users will make it significantly less complex. I stand my ground. I don't yet believe the apocalyptic scenarios foretelling the end of the software developer. Somebody still has to create the platforms, software infrastructure, and components that users need, whoever those users might be. Whether delivered as a service or as a traditional product, these pieces are still software as we know it. Even if end users (whether lay people, IT specialists, or domain experts) might focus on customization and coarse-grain composition, software developers will continue to work on creating and maintaining the nuts and bolts that end users need. And that task is no easier than creating traditional products.

There's good reason to be wary about non-software-developer end users' ability to churn out serious, robust, and significantly customized business or technical applications on the fly. If the diversity of such applications is not to resemble that of the 10 model homes of a 20,000-unit suburb, I don't see the sort of customization foretold in company white papers, blogs, and IT editorials happening easily or anytime soon.

A Simple Litmus Test

Such myths are easy to dispel. I checked out the offering of an innovative start-up company in the cloud computing space. The company was mentioned as a shining example in one of the articles of *The Economist's* October 2008 technology supplement. Its impressive offering targets end-user creation and deployment of on-demand business applications as a service. The promotional video promised "application creation without a single line of code" and with "no need for testing." But read the fine print in the tutorials posted on the company Web site, and you'll see how such hopes are quickly quashed, except perhaps for the simplest of applications. It turns out data computations are achieved through scripting, and that means good old programming. But of course. Making the point doesn't require invoking extreme cases. Just imagine a modestly complex application with a few dozen workflows and an equal amount of business rules requiring simple computations that depend on each other. Now imagine deploying it without any testing. Good luck!

Opportunities for Developers

This example is not intended to discredit the paradigm, just the exaggerated and premature claims of end-user empowerment. On the contrary, even if lay end users won't be able to whip up a serious enterprise application in a matter of days, cloud computing opens up exciting new possibilities based on a mix of old and new technologies for the next generation of software developers. Some of these possibilities are fast turning into reality through cloud-computing metasevices in the platform and infrastructure arenas offered by established players such as Amazon (Elastic Computing Cloud), Google (App Engine), Yahoo (Y!OS), and Salesforce.com (Force.com), as well as emerging ones such as Zoho.

A compelling example is in Abel Avram's announcement on InfoQ (www.infoq.com/news/2008/12/Zoho-GAE): it's becoming possible to deploy applications created by Zoho's online service on Google's App Engine platform. Such decoupling of complementary metasevices might let smaller players tap into a larger user base that prefers the comfort of operating under the wings of a major player with lots of resources.

The Dream of Platform Independency

Platforms such as App Engine promise application deployment on large, robust server bases for instant scalability, security, and performance, but that's only an ideal for now. A video I watched about a cool social-networking application deployed on App Engine suggests otherwise. Toward the end, the video talks about platform idiosyncrasies that require special attention for good performance and high scalability. However, these are the very things that are supposed to be automatically inherited from the cloud platform, not engineered into the application. Even if total insulation from platform worries isn't a reality yet, those worries will hopefully decrease, if not entirely disappear, over time as underlying technologies mature. Despite its remaining caveats, cloud computing still represents a more practical alternative to model-driven approaches in reducing overwhelming platform concerns for application developers.

Challenges Ahead

Litoiu says demonstrating promised economic returns is the main challenge cloud

computing faces. Will customers and users realize the anticipated cost savings without giving up the convenience and comfort of local deployment, control, and operation?

On the supplier side, sustainability—the service providers' ability to meet ever-growing capacity and operational management needs—presents another possible roadblock. Coordinated, smooth evolution of the resulting systems—a tangled web of users, applications, platforms, and infrastructure—is difficult, but essential. Sustainability and evolvability may well depend on the level of automation achieved at the periphery and deep inside the cloud so that resources and services self-adapt to changes in demand and in the environment with minimal intervention.

Big Words

According to *The Economist* ("Let it Rise," October 2008),

the rise of the cloud is more than just another platform shift that gets geeks excited. It will undoubtedly transform the IT industry, but it will also profoundly change the way people work and companies operate. It will allow digital technology to penetrate every nook and cranny of the economy and of society, creating some tricky political problems along the way.

Substitute "Internet" for "cloud," and you'll find that this prediction sounds a tad too familiar. Except, hasn't it already come true? It seems to me that the next transformation begs a different shape and quality.

After all is said and done, the future of cloud computing remains uncertain, at least at a scale predicted by the *Economist* article. A telltale sign reflecting doubts about mass adoption is the still-cautious attitude of the major PC manufacturers toward slim, portable network computers ("Netbooks Confuse Some PC Makers," *USA Today*, 20 Jan. 2008). These appliances are perfectly positioned to unleash the benefits of SaaS at low cost to users. If the future of cloud computing were so certain, PC manufacturers would probably feel more gung-ho about netbooks despite the grim economic climate and thin profit margins. ☞