BY AARON WEISS

COMPUTING IN THE CLOUDS

Powerful services and applications are being integrated and packaged on the Web in what the industry now calls "cloud computing"

Quick, go outside and look up at the clouds in the sky. What shapes do you see? A ball of cotton? A bunny rabbit? A massively parallel distributed data center? If the latter sounds spot on, you might already be computing in the clouds.

At least, that's the latest catchphrase buzzing around the industry. "Cloud computing," as it's being called by everyone from IBM to Google to Amazon to Microsoft, is supposedly the next big thing. But like the clouds themselves, "cloud computing" can take on different shapes depending on the viewer, and often seems a little fuzzy at the edges.

Illustration by James O'Brien

To some, the cloud looks like Web-based applications, a revival of the thin-client. To others, the cloud looks like utility computing, a grid that charges metered rates for processing time. Then again, the cloud could be distributed or parallel computing, designed to scale complex processes for improved efficiency. Maybe everyone is right. There are many shapes in the clouds.

Both Google and IBM have a vested interest in encouraging cloud computing: THEY NEED PEOPLE TO HIRE.

Cloud Shapes: The Data Center

It is not news that today's major Internet companies have built massive data centers to power their online businesses. Decades ago, computing power was concentrated in mainframes tucked away behind the scenes because there was no alternative—only a hulking room-sized box that could contain any significant amount of computational power. The idea that this power could be distributed rather than centralized seemed like such folly that in 1943, IBM Chairman Thomas Watson said famously (or infamously) that "I think there's a world market for maybe five computers."

The era of the personal computer that has flourished since the 1970s directly contradicted Watson's prediction, of course. Desktop (and laptop) computing power has been on an accelerated tear for 30 years. But the networked era, and the data centers that power it, are starting to make the IT industry—and its investors—take a fresh look at Watson's perspective on centralized computing, if not his specific enumeration.

But data centers aren't new, either. In the dot-com boom of the mid-'90s, many a startup invested venture capital into traditional enterprise solutions like Sun SPARC servers. Is this the cloud?

Yes and no. Google is often credited with innovating search on the Web and, more recently, advertising. But to many, Google's architecture behind the scenes has spawned just as significant a revolution.

The data centers of the early dot-com era were, in some respects, direct descendents of Watson's mainframes. Physically smaller, perhaps, but Sun servers and their ilk continued to represent an exclusive kind of computing—concentrated power designed, and priced, for exclusive customers—namely, enterprise.

But Google turned the data center model on its head. Rather than power a network with a small number of high-powered and very expensive servers, why not deploy cheap, commodity hardware in large numbers?

Today, Google runs an estimated half-million servers clustered into a dozen or so physical locations. By creating a network that is spread thin and wide rather than narrow and deep, Google created a new kind of concentrated power—derived more from scale of the whole than any one constituent part. This, some say, describes the cloud.

As computational and networking architecture, the cloud is very robust. Sometimes described as "self-healing," a thin, wide network can recover gracefully from the most common ailments, such as connection and hardware failures, because there are so many more drones available to take on the work.

But a cloud can consume a lot of power to run. Aside from the power needed to drive thousands, or hundreds of thousands, of processors and peripherals—hard drives, cooling fans—all these whirring machines generate lots of heat. It is estimated that 50 percent of energy costs in running a large data center are derived from cooling needs alone.

Worse still for the cloud, the world is immersed in a global energy crunch, as both demand and speculation has driven up pricing for most conventional energies toward record levels.

Reducing the operating costs of a Google-inspired data center cloud involves both physical and virtual solutions. Physically, data centers are like plants arcing toward the sunlight, migrating toward cheap energy. Google is building a major data center at a derelict factory site on the banks of the Columbia River in the Pacific Northwest, located both near cheap hydroelectric energy and a major trans-Pacific Internet node.

Likewise, Microsoft, IBM, and others are following suit, scouting sites both in the Pacific Northwest and Canada where hydroelectric power is cheaper (and greener) than the coal-derived power used throughout much of the U.S. The eyes of investors are also turning toward China, where new power plants are being rapidly built (and, some are quick to caution, without the "costly" burdens of environmental controls).

Besides cheaper power, data centers are making heavy use of virtualization to squeeze the most out of the watts they're consuming. With major vendors like VMWare and Citrix, which recently acquired the Xen virtualization platform, increasingly targeting the data center, virtualization allows a single server to run multiple operating instances simultaneously. By sandboxing each OS inside artificial boundaries, not only can each instance run independently of the others, but CPU idle time is minimized.

Just what distinguishes a "cloud" from "a bunch of machines" can be a little fuzzy. But the next evolution that may illuminate the fog is the so-called "data center OS"—or, in the spirit of the buzz, the CloudOS. In the fall of this year, VMWare and Cisco announced a joint venture to develop such a "fabric" (to use their word in spite of the mixed metaphor).

In a data center like that employed

by Google and the many other enterprises inspired by their model, each server is fundamentally an independent machine running its own copy of an operating system. For servers to work together on common tasks requires an abstraction layer of software—often, highly specialized, custom software—that intelligently divvies up jobs. But a more efficient and resource-friendly solution would be a single operating system, which intrinsically utilizes the resources of many machines.

Essentially, an operating system is designed to manage resources—hard drive space, memory, and so on. A true data center, or cloud, OS will treat every processing unit available as just another resource, relying on networking channels to replicate the kinds of intra-server channels that now coordinate events within a single physical machine. Under the command of a single "omniscient" operating system, the cloud becomes a more cohesive entity.

Cloud Shapes: Distributed Computing

Whether "the cloud" represents a data center at a single physical location or dozens, hundreds, or thousands of data centers spread around the world, its speed and efficiency is limited by how intelligently it delegates responsibility.

Completing any general computing task—say, retrieving the results of a search with relevant contextual advertisements—requires a long series of smaller jobs to be completed. Database queries, parsing of results, construction of result sets, and formatting of result pages, to name the most common. Even these tasks can be further broken down into sub-tasks. Those sub-tasks can be broken into even smaller tasks, and so on, until you're nearly down to "bare metal" as they say and dealing with disk and memory access.

Ideally, if tasks are broken into their smallest constituent jobs, and each job could be completed simultaneously using available processing resources somewhere in the cloud, you could achieve an optimally efficient architecture: the most optimistic definition of distributed computing.

In reality, some jobs are dependent on the results of other jobs. Furthermore, designing algorithms to most effectively divide jobs and distribute them throughout the cloud in real time is complex, to understate the case.

But distributed computing, like the data center itself, is not inherently new to the era of the cloud. If we consider the entire Internet a cloud, one need only look at popular projects like SETI@home and Folding@home to see public examples of distributed computing at work. In these projects, individuals run software on their PC which connects them to a server that divides large jobs among small clients to crunch numbers toward a particular goal—in these examples, searching for alien life among radio waves and computing protein folding simulations to aid medical research. You could even consider botnets—software that maliciously infects unwitting client machines to send out parcels of spam—a form of nefarious distributed computing.

The open source project Hadoop provides a general-purpose framework for developers to rapidly employ distributed computing in a wide variety of projects. Actively under development within the auspices of the Apache Foundation, Hadoop liberates software developers from creating specialized custom software for taking advantage of distributed computing in a cloud. Capable of distributing petabytes of data-processing across thousands of computing nodes, the Hadoop platform is itself comprised of several technologies to ensure efficiency and data integrity as data swirls around the cloud. Some of these subprojects, like MapReduce which divides jobs into component tasks, and HDFS, the Hadoop Distributed Filesystem, can be employed individually or together by developers in their applications.

WHY NOT JUST MOVE ALL PROCESSING POWER TO THE CLOUD, and walk around with an ultralight imput device with a screen?

Cloud Shapes: A Utility Grid

It seems like every shape one sees in the cloud is inspired by a computing model from the past. Back in the days of mainframes and fancy supercomputers housed at research universities, valuable processing time was essentially for sale. Researchers would submit jobs to these number-crunching powerhouses (in their day), and be billed for the cost of cycles used. Processing time was delivered like electricity—you paid for what you used.

Today, most medium to large-sized organizations invest in their own data centers and use them at will. Although processing cycles are not metered like a utility, there are significant costs to building a data center—real estate, hardware, power, cooling, and ongoing maintenance.

What's worse for the balance sheet is that organizations need to plan for worst-case scenarios. In the case of a data center, not only can this include the costs of backup and redundancy, but in overpowered servers capable of handling loads which can peak high but occur infrequently. Businesses may find themselves using 99 percent of their computing capacity only 10 percent of the time, leaving expensive equipment often idle and angering the accountants.

In the Web space there is a thriving market of hosting providers who invest in their own data centers and sell usage to customers, be they individuals or businesses. Although hosting providers can go some way toward minimizing underutilization of in-house servers by attaching surcharges to peak usage, their capacity to stretch can be limited. Providers typically assign clients to a whole or partial physical server, and do not truly replicate the kinds of cloud features, like distributed computing, that a dedicated data center can offer.

Software, in a cloud, becomes a service. To a company like Microsoft, whose substantial fortunes are built on software as a purchasable, local application, THE THREAT IS NOT FALLING ON DEAF EARS.

Meanwhile, massive companies like Amazon, Google, and IBM have invested in, innovated, and become expert at housing their own large-scale data centers. Invoking the idea of the cloud, all three also sense opportunity: Why not scale up their data centers—grow the cloud—and create business models to support third-party use?

In fact, Internet retail giant Amazon is the first out of the gate to commercialize their cloud. After a period of limited access testing, Amazon opened their Elastic Compute Cloud, or EC2, to fee-based public use in October 2007.

To use Amazon's EC2, customers first create a virtual image of their complete software environment using provided tools. The image is then used to create an "instance" of a machine in Amazon's cloud. Using virtualization, Amazon presents the machine image to the end user as if it were a dedicated server, with the same degree of access an administrator would have to their own server.

Customers can choose configuration templates for their machine instance—for example, a server with 1.7GB of memory, one processing core, and 160GB of storage space. Additional configurations support more memory, more cores, and more storage. But the real magic in EC2 is that customers can create and destroy machine instances at will. As a result, software can scale itself to exactly the amount of computing power it requires.

Consider a Web-based application running in Amazon's cloud. Suppose there is a sudden surge in visitors, perhaps thanks to media coverage. Today, many Web applications fail under the load of big traffic spikes. But in the cloud, assuming that the Web application has been designed intelligently, additional machine instances can be launched on demand. The application dynamically, and gracefully, scales up. When traffic slows down, the app can scale down, terminating the extra instances.

Amazon then charges customers based on their cumulative "instance hours," a way of metering usage, currently set at $0.10 per instance hour for the most basic instance type. To provide consistent and predictable performance, machine instance types are rated by what Amazon calls "compute units," which deliver a specific, known amount of performance, regardless of the underlying hardware Amazon is using inside their cloud. A second charge applies to data moved in and out of Amazon's network, ranging from $0.10 to $0.18 per gigabyte.

In contrast to Amazon's commercial effort, Google and IBM recently announced a partnership to apply a similar utility cloud model to computer science education. The two companies have put together a dedicated data center mixing both commodity and enterprise hardware. Running open source software including Linux, Xen virtualization, and Apache Hadoop, the Google/IBM cloud presents a distributed computing canvas for educational access. Both Google and IBM have a vested interest in encouraging cloud computing: They need people to hire.

A considerable concern among major enterprise is that today's computer science students lack access to the distributed computing environments that make up the cloud. Companies like Google will need new hires experienced in writing code designed to run in a cloud of perhaps thousands of processors. But educational institutions don't have these kinds of massive, sandboxed data centers dedicated for student use, and many lack instructors with leading-edge experience.

To some industry analysts, commodified cloud computing like Amazon's EC2 will change the face of enterprise computing. It may pave the way to where businesses no longer invest anything into data centers of their own. Out goes the hardware, out goes the power bill, and out goes all the processing power that rarely gets used. Instead, they buy cloud computing time from commercial providers, who can specialize in building massive data centers at sites selected to minimize operating costs.

Cloud Shapes: Software as a Service

As radical a shift as cloud computing may represent for the enterprise—who needs servers?—some think it will radically change personal computing even more. Who needs computers at all? At least, the kind that eat up battery life and contain churning hard drives. Why not just move all processing power to the cloud, and walk around with an ultralight input device with a screen?

Some say it's already begun. Burgeoning Web applications have been the rage for several years. Fueled by technological evolutions like AJAX, which allows browser-based code to behave more like a local application, and people's desire for mobility and data ubiquity, we increasingly use the Web for application functionality. E-mail was both the first "killer app" for the Internet, and later, on the Web. For many today, Web-based e-mail is the only kind they use.

THE CLOUD DEMANDS A HIGH DEGREE OF TRUST. Significant amounts of data which were previously stored only in individual offices and homes would now reside in data centers controlled by third-parties.

Taken further, Web-based applications like Google Docs threaten core productivity applications on the desktop. While the features of Google Docs are a slim shadow of a major, and majorly profitable, application like Microsoft Word, it provides a taste of a cloud-based future. Your data is reachable anywhere, and the only software you need to access it is a browser. Which, in turn, means that your data and your applications are available in one form another from your desktop PC, your laptop, and your handheld.

Software, in the cloud, becomes a service. To a company like Microsoft, whose substantial fortunes are built on software as a purchasable, local application, the threat is not falling on deaf ears. Windows Live, Microsoft's "cloud," may be today's take on Windows 1.0—a look into the future.

Even Adobe has begun to launch stripped-down versions of its computationally-heavy marquee titles as Web-based services, including Photoshop and the video editing suite Premiere.

Pundits and futurists like Nicholas Carr love this stuff because the cloud represents a paradigm shift and a disruptive force and a whole new way. Carr predicts that Google and Apple will partner to push the cloud computing/software-as-a-service model even further. He foresees a lightweight mobile device crafted by Apple that will tap into Google's cloud, bringing together the two masters of the front end and the back end.

In the near-term, a ubiquitous cloud faces obstacles. Critics argue that visions like Carr's are simply revivals of failed thin-client dreams of the past. Thin clients like those touted by Oracle-founder Larry Ellison in the '90s have not managed to become cost-effective. With prices plummeting and performance skyrocketing for full-featured desktop and laptop computers, it has been difficult to produce a relatively powerless thin client at a low enough cost to attract buyers.

Advocates for thin client cloud computing argue that full-fledged machines, however powerful, are also a hassle with negative productivity. They have many parts that can fail and their software needs constant care and feeding in a world riddled with software updates, viruses, and spyware. Centralizing processing power in the cloud liberates users to choose efficient, uncomplicated access machines. This argument, and its potential economic feasibility, gains weight in today's environment, where laptop sales are outpacing the desktop, and even more mobile devices are becoming commonplace.

For cloud computing to move front and center, the networks that tie everything together need to be extremely robust. After all, the cloud represents a significant inversion from personal computing in the 1970s and 1980s, when machines stood alone and derived all of their utility from their own capabilities. Under the cloud, client machines become nearly useless on their own. Are our networks ready to handle the load?

Many would say no, especially those living in the large U.S. market, where broadband quality and penetration fares poorly relative to many smaller industrialized nations. In contrast to local computing power, which continues to boom year after year, broadband advances here have been slow and may create at least a short-term bottleneck holding back a major shift toward mainstream cloud computing.

Network bandwidth aside, the cloud raises concerns among privacy advocates. Most significantly, the cloud demands a high degree of trust. Significant amounts of data which were previously stored only in individual offices and homes would now reside in data centers controlled by third-parties. Do we have adequate privacy laws? How should encryption play a role?

Software as a service in the cloud can revive fears of vendor lock-in, a significant consideration in the mainframe era. Supposing a cloud operator and a thin-client vendor partner together, it is possible that each half will require the other. Services from the cloud may be inaccessible to those without an access device from a single brand. Some fear that the cloud could encourage the growth of walled-gardens, a potential step back compared to the relatively open Internet of today.

Partly Cloudy?

There are those who scoff that "cloud computing" is just the latest branding of some old, familiar computing models—which is partly true. It's a buzzword almost designed to be vague, but cloud computing is more than just a lot of fog.

The cloud concept draws on many existing technologies and architectures. Centralizing computing power is not new and, if anything, is a return to the computer's roots. Nor is utility computing new, or distributed computing, or even software as a service.

But the cloud is new in that it integrates all of the above computing models. This integration requires computing's center of power to shift from the processing unit to the network. Inside the cloud, processors become commodities. It is the network that holds the cloud together, and connects the clouds to each other and the sky to the ground. ~

Aaron Weiss *is a technology writer and Web developer shivering in upstate New York, as well as human proprieter of livenudecats.com*