

[Tech Notes are short articles discussing library-related technology.]

[Tech Note:] The Clouds Around Us: Cloud Computing

You may have heard about “computing in the cloud.” Or you may not have, but you probably should learn a bit about it. Cloud computing is about moving applications, or pieces of them, out of an organization (such as a library) to some external place. The place is usually a group of servers (a server farm) somewhere on the Internet. So instead of an investment in the infrastructure and maintenance of local servers, the organization pays for raw computing power.

Computing in the cloud is one of the latest hip-and-happening trends in the business world. Because of the relative novelty of cloud computing some caution is justified. It’s never a good idea to hop aboard the train in an early phase of trend development.

How does cloud computing work? The concept is fairly simple. First, consider the traditional means of running large applications (e.g., ILSes): an application appears to run on a dumb terminal or, more likely these days, your PC. In actuality, this is only the “front-end” of the application. Your computer is connected to a server that actually runs the program and returns information to your personal computer. The server constitutes the “backend”. The dedicated server may be located in the same building as you are or not. If your library shares a catalog and other components of an ILS with other libraries in a system or other consortium the server will probably be located at your system’s headquarters or the resource library.

With cloud computing, the application runs somewhere within the cloud. Ideally, the user need not be aware of the underlying technology or the physical location of the application’s computer, being concerned only with the applications that are available. Her desktop is connected via the Internet to a server farm, a collection of remote servers that runs many, many applications at once. Which server or servers an application runs on is determined by the programs already running on the machines—there is an attempt to balance the load so that all of the programs run optimally.

An example of use of server farms and load balancing is the Google search engine. When you connect to Google, your search runs on a computer in a server farm somewhere within the “Googleverse”. You neither know nor care where that computer is located, you just want your search results. Think about all of the other people who are searching at the same time, or running other Google applications. These are all spread across identical computers with masterful load balancing.

Ironically, computing in the cloud brings things full circle in terms of computing resources. In the beginning computers were managed by the priesthood of programmers and were not accessible to mere laypeople. Then came timesharing, when people had big, clunky terminals on their desktop that looked like television sets with keyboards. Staff throughout a company had to wait for information technologists to write software

for their application needs, or in some cases to even run the programs. Then the advent of personal computers allowed people to run applications on their own computers, over which they had complete control. Eventually, networking products emerged that allowed PCs to communicate with each other, transferring data back and forth, and giving us email, as well as running some shared applications. Additionally, the big mainframes of yesteryear evolved into servers within companies (and libraries); these were powerful, low-cost computers. Finally, with cloud computing, the full circle: many personal computers accessing big servers in a kind of timesharing.

There is another wrinkle to cloud computing: often what runs in the cloud is not a monolithic application, but rather a number of “web services”, mini-applications that respond with data after receiving a simple request from a person or locally-run application. Web services can be cobbled together into a full-fledged application originating at the desktop or a locally-based server. Web services are one aspect of what is called “Web 2.0”.

There are a number of companies that offer cloud computing server farms. Amazon (yes, the book people) offers something called [Amazon Elastic Compute Cloud \(EC2\)](#). There are many startups and established Internet services companies that rent space and time on these servers. Processing can cost as little as 10 cents an hour and disk space goes for 15 cents per gigabyte per month [1]. Startups (and others) love the low prices because they enable them to set up a web presence without buying or renting any of their own servers. They only pay for the computing power they need, so if their business tanks they haven't spent a lot of money. Conversely, if their service succeeds in a big way the cloud can be scaled up appropriately without much effort. So far, some 400,000 developers use EC2, with 10,000 more signing up monthly.

Google recently weighed in with its own cloud computing offering, [Google App Engine](#). A big attraction of the App Engine is that it's free, within some quite broad limits: “every Google App Engine application can use up to 500MB of persistent storage and enough bandwidth and CPU for 5 million monthly page views” [2]. It is not as flexible as EC2 because it currently will run only applications written in a language called python, but support for other languages is promised.

One of the things that these two endeavors share is that they were originally developed for internal consumption. Amazon's servers were creaking along until they were more efficiently designed to make Amazon's own website run faster. Google's huge grid of computing power has been written about extensively. Essentially, they have thousands of identical small computers around the world, and any of their wide range of applications can run on any of them. So freely offering the App Engine was not a big stretch for them. They will eventually offer higher-capacity attachment to the App Engine for a fee, probably based on processing needs and disk space as EC2 is.

What's in this for libraries? Eventually more and more pieces of applications, including ILSes, will migrate to the cloud. [WorldCat Grid Services](#) have many of the characteristics of cloud computing and offer access to much bibliographic and other information (although those services that use WorldCat Local won't be accessible until sometime after the next ALA conference in June, 2008). Other services will eventually provide more pieces of the OPAC and ILS pie, including rich search capabilities and integrated access to online materials and pointers to offline materials.

Takeaways From This Article

- Cloud computing is moving an application or pieces of an application to a remote, Internet-connected array of servers.
- Cloud computing may consist of complete applications or, more often, various *web services*, mini-applications that respond to applications' requests for pieces of information.
- In cloud computing, users will not be aware of underlying hardware that runs applications and services.
- Cloud computing is scalable, so all users are accommodated with minimal delays.
- Cloud computing can be a possible alternative to investing in local server computers, freeing capital for other purposes.
- A few web services of interest to libraries and ILS developers already inhabit the cloud, and more are coming.
- Eventually some ILSes or their modules will be based in the cloud.

[1] "Planet Amazon," *Wired*, May 2008, by Spencer Reiss, pp. 88-95.

[2] "Google App Engine," <http://code.google.com/appengine/> .

—Tom Zillner (tzillner@wils.wisc.edu)