IT STRATEGY GUIDE

InfoWorld
GET TECHNOLOGY RIGHT

Cloud Computing: The Future of IT

See the full selection of InfoWorld IT Strategy Guides at http://www.infoworld.com/reports/y/strategy-guide .

CLOUD COMPUTING: THE FUTURE OF IT

THE CLOUD MAY STILL BE TAKING SHAPE, BUT INTERNET-BASED SERVICES ALREADY OFFER SERIOUS ALTERNATIVES TO IN-HOUSE IT SOLUTIONS

Cloud computing is the buzz phrase on the lips of every analyst and vendor. As with all new IT trends, levels of adoption are low. But adoption of what, exactly? The most interesting thing about cloud computing is that arguments over its definition continue to rage even as customers pay actual money for it.

In 2008 InfoWorld offered its own "official" definition of cloud computing. We tried to keep it simple: "Cloud computing encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities." Then we broke cloud services into seven categories – but they could easily be reduced to three: infrastructure services such as Amazon EC2; software as a service à la Salesforce; and development platforms, such as the new Windows Azure platform from Microsoft.

So where's the controversy? Well, on the one hand, some folks insist that cloud computing refers to infrastructure services alone, which is basically the utility computing model. Instead of buying new servers, you move virtual machines to a service provider over the Internet – period. At InfoWorld, we see that as an unnecessarily narrow view and prefer to include SaaS and its ilk. It's all about computing services, apps included, that you procure outside rather than inside the enterprise.

On the other hand, some big vendors go even broader and talk about the "private cloud." Yes, they say, cloud services on the Internet are cloud computing, but with the right infrastructure, IT departments can deliver their capabilities to internal customers as cloud services, too. To enable that scenario, this view aggregates other IT trends, such as SOA (service-oriented architecture), virtualization, datacenter automation, and EDA (event-driven architecture) into one harmonious private cloud.

There's a valid point here: Why shouldn't internal IT resources packaged and delivered as services qualify as cloud computing? If IT can pull it off, then the internal cloud has the same contours as the external cloud.

We haven't seen major examples of internal cloud computing initiatives yet, but in several case studies, we've certainly examined the way customers avail themselves of cloud services delivered by external providers. The InfoWorld Test Center has been busy testing cloud services, too. The majority are platform-as-a-service plays, mainly because the idea of having a new environment to play in (one instantly available without provisioning) and create new Web apps is so compelling to so many developers.

We don't expect the controversy over what cloud computing really means to end anytime soon. When you're the only exciting game in town, everyone is going to want a seat at the table, even though some players may have a slightly different game in mind. You'll see lots more cloud computing coverage from InfoWorld, as well as a lot more adoption from companies that would rather pay a little more in operating costs than invest in infrastructure to get the new capabilities they really need.

See the full selection of InfoWorld IT Strategy Guides at http://www.infoworld.com/reports/y/strategy-guide.

WHAT CLOUD COMPUTING REALLY MEANS

THE NEXT BIG TREND SOUNDS NEBULOUS, BUT IT'S NOT SO FUZZY WHEN YOU VIEW THE VALUE PROPOSITION FROM THE PERSPECTIVE OF IT PROFESSIONALS.

BY GALEN GRUMAN AND ERIC KNORR

Cloud computing is all the rage. "It's become the phrase du jour," says Gartner senior analyst Ben Pring, echoing many of his peers. The problem is that (as with Web 2.0) everyone seems to have a different definition.

As a metaphor for the Internet, "the cloud" is a familiar cliché, but when combined with "computing," the meaning gets bigger and fuzzier. Some analysts and vendors define cloud computing narrowly as an updated version of utility computing: basically virtual servers available over the Internet. Others go very broad, arguing anything you consume outside the firewall is "in the cloud," including conventional outsourcing.

Cloud computing comes into focus only when you think about what IT always needs: a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. Cloud computing encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities.

> Cloud computing encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities.

Cloud computing is at an early stage, with a motley crew of providers large and small delivering a slew of cloud-based services, from full-blown applications to storage services to spam filtering. Yes, utility-style infrastructure providers are part of the mix, but so are SaaS (software as a service) providers such as Salesforce.com. Today, for the most part, IT must plug into cloud-based services individually, but cloud computing aggregators and integrators are already emerging.

InfoWorld talked to dozens of vendors, analysts, and IT customers to tease out the various components of cloud computing. Based on those discussions, here's a rough breakdown of what cloud computing is all about:

1. SAAS

This type of cloud computing delivers a single application through the browser to thousands of customers using a multitenant architecture. On the customer side, it means no upfront investment in servers or software licensing; on the provider side, with just one app to maintain, costs are low compared to conventional hosting. Salesforce.com is by far the best-known example among enterprise applications, but SaaS is also common for HR apps and has even worked its way

See the full selection of InfoWorld IT Strategy Guides at http://www.infoworld.com/reports/y/strategy-guide.

up the food chain to ERP, with players such as Workday. And who could have predicted the sudden rise of SaaS "desktop" applications, such as Google Apps and Zoho Office?

2. UTILITY COMPUTING

The idea is not new, but this form of cloud computing is getting new life from Amazon.com, Sun, IBM, and others who now offer storage and virtual servers that IT can access on demand. Early enterprise adopters mainly use utility computing for supplemental, non-mission-critical needs, but one day, they may replace parts of the datacenter. Other providers offer solutions that help IT create virtual datacenters from commodity servers, such as 3Tera's AppLogic and Cohesive Flexible Technologies' Elastic Server on Demand. Liquid Computing's LiquidQ offers similar capabilities, enabling IT to stitch together memory, I/O, storage, and computational capacity as a virtualized resource pool available over the network.

3. WEB SERVICES IN THE CLOUD

Closely related to SaaS, Web service providers offer APIs that enable developers to exploit functionality over the Internet, rather than delivering full-blown applications. They range from providers offering discrete business services – such as Strike Iron and Xignite – to the full range of APIs offered by Google Maps, ADP payroll processing, the U.S. Postal Service, Bloomberg, and even conventional credit card processing services.

4. PLATFORM AS A SERVICE

Another SaaS variation, this form of cloud computing delivers development environments as a service. You build your own applications that run on the provider's infrastructure and are delivered to your users via the Internet from the provider's servers. Like Legos, these services are constrained by the vendor's design and capabilities, so you don't get complete freedom, but you do get predictability and pre-integration. Prime examples include Salesforce.com's Force.com, Coghead and the new Google App Engine. For extremely lightweight development, cloud-based mashup platforms abound, such as Yahoo Pipes or Dapper.net.

5. MSP (MANAGED SERVICE PROVIDERS)

One of the oldest forms of cloud computing, a managed service is basically an application exposed to IT rather than to end-users, such as a virus scanning service for e-mail or an application monitoring service (which Mercury, among others, provides). Managed security services delivered by SecureWorks, IBM, and Verizon fall into this category, as do such cloud-based anti-spam services as Postini, recently acquired by Google. Other offerings include desktop management services, such as those offered by CenterBeam or Everdream.

6. SERVICE COMMERCE PLATFORMS

A hybrid of SaaS and MSP, this cloud computing service offers a service hub that users interact with. They're most common in trading environments, such as expense management systems that allow users to order travel or secretarial services from a common platform that then coordinates the service delivery and pricing within the specifications set by the user. Think of it as an automated service bureau. Well-known examples include Rearden Commerce and Ariba.

7. INTERNET INTEGRATION

The integration of cloud-based services is in its early days. OpSource, which mainly concerns itself with serving SaaS providers, recently introduced the OpSource Services Bus, which employs in-the-cloud integration technology from a little startup called Boomi. SaaS provider Workday recently acquired another player in this space, CapeClear, an ESB (enterprise service bus) provider that was edging toward b-to-b integration. Way ahead of its time, Grand Central – which wanted to be a universal "bus in the cloud" to connect SaaS providers and provide integrated solutions to customers – flamed out in 2005.

Today, with such cloud-based interconnection seldom in evidence, cloud computing might be more accurately described as "sky computing," with many isolated clouds of services which IT customers must plug into individually. On the other hand, as virtualization and SOA permeate the enterprise, the idea of loosely coupled services running on an agile, scalable infrastructure should eventually make every enterprise a node in the cloud. It's a long-running trend with a far-out horizon. But among big metatrends, cloud computing is the hardest one to argue with in the long term.

Galen Gruman is executive editor of InfoWorld. Eric Knorr is editor in chief at InfoWorld.

See the full selection of InfoWorld IT Strategy Guides at http://www.infoworld.com/reports/y/strategy-guide.

EARLY EXPERIMENTS IN CLOUD COMPUTING

CLOUD COMPUTING HAS MANY FACES, SOME JUST BEGINNING TO TAKE SHAPE. BUT AT THE NEW YORK TIMES AND NASDAQ, FIRST STEPS INTO ON-DEMAND INFRASTRUCTURE SHOW PROMISE. BY GALEN GRUMAN

You know there's substance behind a technology buzzword when companies such as the Nasdaq OMX stock exchange and the New York Times publishing company use it for real production efforts. Cloud computing is the latest buzzword that vendors are using to spruce up the usual sales spiel, and the fever pitch is enough to make you think, "Dot-com boom, here we go again." While the skepticism is warranted, something very real is happening, and IT needs to pay attention.

So what are Nasdaq and the Times doing? In a phrase, utility computing. Both have tapped into Amazon.com's Internet-provisioned computing and storage services – Elastic Compute Cloud (EC2) and Simple Storage Service (S3) – to augment their own IT resources.

The Times processed 4TB of data through EC2 and S3, using a credit card to get the service going in a matter of minutes so that it could convert scans of 15 million news stories into PDFs for online distribution. Nasdaq uses S3 to deliver historical stock and mutual fund information, rather than add the load to its own database and computing infrastructure. Likewise, Infosolve Technologies uses Sun's Network.com grid-in-the-cloud utility to scrub customer addresses rather than stand up that infrastructure internally.

In another realm of cloud computing, companies such as medical robotics firm Intuitive Surgical and recruitment services provider Jobscience use in-the-cloud development environments to create and provision their own applications. Both companies use Salesforce.com's Force.com platform as a service, the ungainly name for this online IDE service, but other firms such as Coghead offer their own platforms.

These two forms of cloud computing – utility computing and platform as a-service – are exciting developments. Unlike SaaS (software as a service), they're aimed squarely at IT users, not at business users looking to bypass IT (or that IT is happy to let someone else take care of). But despite early promise, analysts say there's a long way to go before they're a mainstream part of your datacenter. So the question is: Do you sit back and wait for them to mature, or do you experiment so that you can get early advantage when they're enterprise-class?

COMPUTING IN THE SKY

Nasdaq OMX has lots of stock and fund data, and it wanted to make extra revenue selling historic data for those stocks and funds. But for this offering, called Market Replay, the company didn't want to worry about optimizing its databases and servers to handle the new load. So it turned to Amazon's S3 service to host the data, and created a lightweight reader app using Adobe's AIR technology that let users pull in the required data. "If I'm someone like Nasdaq, it's a cheap experiment," says Nik Simpson, a senior analyst at the Burton Group.

The traditional approach wouldn't have gotten off the ground economically, recalls Claude Courbois, an associate vice president for data products at Nasdaq: "The expenses of keeping all that data online was too high." So Nasdaq took its market data and created flat files for every entity, each holding enough data for a 10-minute replay of the stock's or fund's price changes, on a second-by-second basis. (It adds 100,000 files per day to the several million it started with, Courbois says.) The Adobe AIR app Courbois' team put together in just a couple days pulls in the flat files stored at Amazon.com and then creates the replay animations from them. The result: "We don't need a database constantly staging data on the server side. And the price is right."

The New York Times also used S3 for a data-intensive project: converting 11 million articles published from the newspaper's founding in 1851 through 1989, to make them available through its Web site search engine. The Times scanned in the stories, cut up into columns to fit in the scanners (as TIFF files), then uploaded those to S3 — taking 4TB of space — over several WAN head connections from the Times' datacenter.

The Times didn't coordinate the job with Amazon — someone in IT just signed up for the service on the Web using a credit card, then began uploading the data. "After about 3TB, we got an e-mail [from Amazon.com] to ask if this would be a perpetual load," recalls Derek Gottfrid, senior software architect at the Times.

Then, using Amazon.com's EC2 computing platform, the Times ran a PDF conversion app that converted that 4TB of TIFF data into 1.5TB of PDF files. Using 100 Linux computers, the job took about 24 hours. Then a coding error was discovered that required the job be rerun, adding a second day to the effort – and increasing the tab by just $240. "It would have taken a month at our facilities, since we only had a few spare PCs," Gottfrid says. "It was cheap experimentation, and the learning curve isn't steep.

•
Unlike SaaS, utility computing and platform-as-a-service are aimed squarely at IT users, not at business users looking to bypass IT
•

Digital Fountain, a digital-media distribution company, uses the EC2 service to deliver mobile videos over the Internet. When the company decided to launch this new offering, "we didn't want to buy our own servers and get the people to do that work," says CTO Mike Luby. So Digital Fountain now streams them from Amazon.com's EC2 servers. Because Amazon.com doesn't guarantee availability, Digital Fountain streams the video from several servers, ensuring built-in backup for its provisioning. And it can throttle the number of servers to match demand as it rises and falls, Luby notes.

Over time, Luby expects to rely on other providers in addition to Amazon.com, to ensure a geographic diversity to keep streaming times manageable, as well as to increase server density without overloading any one provider.

There's more to utility computing than Amazon.com. Sun also has its own cloud-based computing platform, Network.com. Unlike EC2, though, it's a grid, meaning it specializes in parallel processing, where a task can be broken into independent steps that a large array of processors can tackle all at once. That limits its use to applications such as rendering, data scrubbing, and image transformation. "Not everything can be thrown at the Sun grid," says Subbu Manchiraj, vice president for technology at Infosolve Technologies, a provider of data management services. But where a task can be parallelized, the benefit is huge, he said.

Infosolve has used the Sun grid for the past 18 months to scrub names and addresses, making sure they are correct (such as verifying the ZIP code and ensuring that the street addresses are properly segmented). "With Sun, we can run 2,000 processors and get the data back quickly." Plus, Infosolve is a Java shop, so its application development skills were easily tuned to the style of Sun's grid apps. That let Infosolve offer its customers a turn-key data-scrubbing service it couldn't afford to stand up itself. "It's an offsite datacenter. And we pay only for what we use," he adds.

The grid's quick scalability has meant that Infosolve doesn't need to worry about balancing customers' loads. But another factor also helps Infosolve avoid worrying about scheduling: The jobs are batched, and customers have no expectation of real-time response. Thus, if resources do run out, the customer won't ever know. Ditto if there's a failure: "We can just restart the job," Manchiraj says.

TESTING OUT ONLINE IDEs

Less mature than the cloud infrastructure plays are the app dev and app hosting platforms provisioned over the Internet. These are intended for apps that will be delivered over the Internet and through the browser anyhow, such as online commerce and services and apps delivered to mobile and remote employees. So it's no surprise that most early adopters of these online IDEs are themselves Web-based service providers.

A typical example is Jobscience, a provider of recruiting services. The company had been using Salesforce.com's customization tools, so it had gown comfortable with the underlying service availability. At the same time, the company struggled to manage its Adobe ColdFusion-based server environment, so CEO Ted Elliott began looking at using an outside hosting firm to simplify its ability to provision customers over the Internet. "But they manage to the stack, not to the app," Elliott says, and he wanted an environment that was operationally optimal, not just technically correct. So he turned to Salesforce.com's Force.com platform as a service to create and host the apps.

Elliott's biggest challenge was internal: His developers didn't want to let go the control over the app environment. But now, "they're starting to see what they can build that doesn't exist [in Force.com] while using the basics [in Force.com] such as calendaring and scheduling," he notes. So the in-house developers get to innovate differentiating apps, not build the basics that everyone else already has.

A more in-the-enterprise example is Intuitive Surgical, a surgical-robotics maker. It is involved in clinical trials of its equipment, and so needs to collect and distribute data across a range of clinical facilities, all of which are separate firms or entities. That data is inconsistent and hard to integrate to get meaningful analysis done. So Intuitive Surgical used Force.com to create a forms-based app to collect that clinical data from all trial participants. "We could build it using just their tools, so in essence, there was no programming," says Mark Burns, a clinical data specialist at the company.

But while the app is handy to collect data, Intuitive Surgical can't use it to submit trial results to the FDA. "It doesn't have the rigor that the FDA would require," he notes, around auditing of the data and tracking everything's that done to ensure the data has not been compromised or altered.

A LONG WAY TO THE CLOUD

Despite the successful examples of the first wave of infrastructure-oriented cloud computing, it's early days for these IT-oriented forms of Internet-based provisioning, and any large-scale shift to them is a good decade away, notes Ben Pring, a senior analyst at Gartner. But it is coming.

Although it can be easy to set up an S3 account using just a credit card, as the Times did, provider availability is a big factor, notes Jon Williams, CIO of Kaplan Test Prep and Admission. For example, Amazon's S3 had an outage in February, notes Burton Group's Simpson. "If half the IT infrastructure is unavailable, that's a difficult situation," he exclaims. (Another outage occurred the day this story was published.)

Some companies can't rely on Internet-provisioned infrastructure services because of regulatory compliance and security issues, adds Williams. "Many have scary compliance issues. How do you demonstrate what you are doing is in compliance when it is done outside?" says Burton Group's Simpson. He notes that the early infrastructure services aren't audited, or take the liability, for security or compliance requirements. Such issues have kept Merrill Lynch from using cloud-provisioned infrastructure. "We're very bound by regulators in terms of client data and country-of-origin issues, so it's very difficult to use the cloud," says Rupert Brown, a chief architect at the financial services firm.

To the degree that cloud computing raises risk, it will inhibit adoption in equal measure, particularly among enterprises. But every large company has noncritical areas where low cost of entry and quick deployment trump reliability, and a significant number of small businesses fall in the same category. For them, experimenting with cloud computing could put them on good footing for an agile, connected future. That's exactly what pioneers like Nasdaq and the New York Times have found.

Galen Gruman is executive editor of InfoWorld.

CLOUD COMPUTING TO THE MAX

SOME COMPANIES COULD MOVE MOST OF THEIR APPS TO THE CLOUD. HERE'S HOW ONE DID IT. **BY DAVE ROSENBERG**

Cloud services claim to provide nearly everything you need without requiring you to run your own IT infrastructure. From e-mail and Web hosting to fully managed applications to vast on-demand computing resources, the cloud is shaping up to be one of the most important technology shifts in the last few years.

Sound too good to be true? Based on my experience over the last two years, I estimate that companies can easily offload 50 to 100 percent of their needs to cloud-based services with minimal business impact and near zero risk – provided you follow the most basic best practices.

That said, not everything is easy, nor is the cloud right for everything. Certain technical requirements, such as very high performance with low latency, are challenging if not impossible. But there are a great many things that can be achieved at a lower cost and minimal risk.

> I estimate that companies can easily offload 50 to 100 percent of their needs to cloud-based services with minimal business impact and near zero risk – provided you follow the most basic best practices.

There are three basic uses of the cloud's tech resources: computing power on-demand, (such as Amazon Web Services EC2 and S3 and Google App Engine); SaaS (software as a service) applications delivered over the Internet, such as Salesforce.com and NetSuite; and PaaS (platform as a service) application development and provisioning delivered over the Internet, such as Force.com.

As cloud offerings continue to mature, I am sure we'll see multiple iterations of the offerings, as well as many permutations. For example, I don't yet know where storage as a service fits, but there are multiple offerings for that, too.

WHY I ENTRUSTED MY OWN BUSINESS TO THE CLOUD

I recently worked for an open source software company that had employees all over the world. That made us extremely dependent on technology to manage interpersonal relationships, all business functions, communications, and software development mechanics.

Having a geographically disperse team is not all it's cracked up to be. Even basic communications can be painfully difficult. If Skype were sketchy, development meetings could get set back an entire day. If e-mail were to go down, multiple business processes would require significant manual intervention. And if Salesforce.com was unavailable, we wouldn't have access to customer data for sales or support.

For the intrepid startup or IT department, the ability to outsource system operations to (theoretical) experts is very appealing, and most people are already comfortable with outsourcing certain elements such as Web servers and e-mail. And in companies with minimal IT staff, developers tend to help out with system administration.

Our goal was to not have to maintain physical machines or applications that weren't explicitly part of our development process. As a software company, we made the decision that our developers should be writing code, not performing system-administration tasks. But sadly, life isn't always that black and white. Our hosted systems and cloud services still required a modest amount of developer time, if for no other reason than the fact that they knew what they wanted the systems to do.

DEVELOPMENT SERVICES IN THE CLOUD

To keep development smooth and not have to spend a ton of money on hardware, we moved all our development applications to Contegix, a managed hosting provider that supported the range of commercial and custom products we used. Our team also had access to the boxes so that every change didn't have to go through a trouble-ticket process (unless we wanted it to).

Having our development systems hosted seemed a bit strange to some people. But most got over that when they realized most developers are working off their own code branches, so they would simply merge the local and hosted pieces of code with the trunk that sits online.

We did run into a few hiccups. The integration of our development applications (Confluence, Jira, Bamboo, and custom code) was nonexistent, so there was a fair amount of work necessary for all the systems to interact properly. And somewhere down the line, one of our guys ordered some servers that we really didn't need. We ended up virtualizing them to the max for QA and testing.

BUSINESS APPS IN THE CLOUD

Most of the cloud apps we used supported business needs, from hosting software for users to download to handling customer data. None of these were within our core competence, and we were glad to rely on someone else to manage them.

WEB SERVER AND DOWNLOADS. I remember an ugly day at work back in 1996 when someone kicked out the power-supply plug for the servers that ran all the Bell Labs and Lucent Web sites. People were running around the hallways in a panic. After that, I decided I never needed to host my own Web servers or DNS.

At my open source company, we stayed true to that conviction. We put our Web site and intranet at Rackspace because we had to deal with software downloads and needed a dedicated machine and enough bandwidth for things not to choke.

We eventually moved our product downloads to Amazon S3, where we didn't worry about administration, bandwidth, or anything else. And we spent less than $100 per month for 15,000-plus downloads. We did suffer once or twice when Amazon went down, but we were able to easily change our Web site to point to our main Web server to get the same file. If you have the luxury of occasional downtime, S3 can't be beat on price or performance.

We also started running demos of our software on EC2 that would expire based on an allotted time frame.

CRM. We tried SugarCRM first but got stuck early on by the lack of a customer-support portal module (it's since been added), so we chose Salesforce.com.

Salesforce.com is relatively easy to use – but only after the initial three or so months of painful trial and error. In addition to the base functionality, we created a very simple customer portal and used the Salesforce.com APIs to pull the data into our look and feel. This was great until we learned that we had exceeded the number of API calls and were forced to upgrade to a more expensive package. Nonetheless, in more than two years, we didn't experience any downtime of note, nor did we lose any information.

E-MAIL. We made the decision from day one that we never wanted to run our own mail server. E-mail is critical to most businesses these days, and it was critical in our case because we had a worldwide development and support team, continuous integration and build servers, forums, blogs, and so on. With all that to handle, we simply didn't want to deal with the possibility of e-mail going down. Letting someone else handle our e-mail sounded great. And it was great – except when it wasn't.

We changed e-mail providers four times in less than two years and made multiple attempts with managed IMAP, Zimbra, and Gmail before we finally got it mostly right. It all started with Rackspace's managed IMAP, which was fine – except there was no calendaring, and as we grew, so did the need for shared calendars.

We switched to Zimbra in late 2006. But it ate several people's calendars and contacts. And although the Zimbra team was fantastic in helping us with our problems, it was becoming a bit too much to deal with. So we went back to IMAP at Rackspace but kept Zimbra running.

Later on, Rackspace began offering hosted Microsoft Exchange. I had been through the hosted Exchange nightmare at another company and refused to get involved in that ordeal again. Plus, for an open source company, it's weird to depend on Microsoft.

Then, like a shining star came Google Apps for our domain. So we switched again. Our first test-drive with Google Apps was all well and good for the first few days. Everyone felt OK about using POP and the Gmail interface, and we were reasonably sure that it wouldn't eat calendars as Zimbra had. What could go wrong?

As it turned out, plenty could go wrong. This was before Gmail supported IMAP, and the POP implementation turned out to have a few very bizarre quirks, such as the fact that you couldn't POP down e-mail that you sent to yourself, including CCs. Messages would disappear into the ether. And user management was a total nightmare; we had something like 40 aliases for lists that had to be entered individually.

It took all of five minutes before our developers freaked out. So we flipped the switch back to Zimbra. Mail delivery was still way more important than calendars, and Zimbra had come out with a new version that synced better and had a slick new UI. But Gmail offered massive storage, and most of our team liked the interface, so the move back to Zimbra left some folks pining for Google. That's why when Google came out with Google Apps Premier Enterprise (GAPE), we gave that a whirl.

GAPE ended up working quite well – except for a few quirks (surprised?), such as the fact that if you use IMAP, you get this weird "All mail" folder that seems to never stop syncing on many versions of Mac Mail. But GAPE met all our requirements, including integration with Salesforce.com. Despite the occasional missing e-mail, we were sold.

I learned that e-mail and calendaring applications are the most personal apps people use, and thus the most difficult to unlearn or change. Your needs here will be driven by more than functional requirements.

> While using cloud services was not without its challenges, I can absolutely say that I would do it again.

WHY I'M DOING IT ALL AGAIN AT MY NEW COMPANY

In the end, our jump to the cloud was based on a desire to avoid expensive, cumbersome infrastructure. While using cloud services was not without its challenges, I can absolutely say that I would do it again. And in fact, I already have: I'm running my new company in a very similar way with minimal capital expenditure for hardware and reliance on a variety of trustworthy providers to manage everything.

Yes, the cloud requires you to give up some control to get benefits. But as far as I can tell, the positives far outweigh the negatives.

Dave Rosenberg is an entrepreneur and freelance writer based in San Francisco.

DIVING DEEP INTO AMAZON WEB SERVICES

FROM STORAGE TO PAYMENT, THE KING OF CLOUDS IS DANGLING AN ARRAY OF LOW-COST SERVICES. WE TAKE A CLOSE LOOK AT THE TOOLS FOR IT AND DEVELOPERS. BY RICK GREHAN

Amazon's Web Services (AWS) are based on a simple concept: Amazon has built a globe-spanning hardware and software infrastructure that supports the company's Internet business, so why not modularize components of that infrastructure and rent them? It is akin to a large construction company in the business of building interstate highways hiring out its equipment and expertise for jobs such as putting in a side road, paving a supermarket parking lot, repairing a culvert, or just digging a backyard swimming pool.

More specifically, AWS makes various chunks of Amazon's business machinery accessible and usable via REST or SOAP-based Web service calls. Those chunks can be virtual computer systems with X2GHz processors and 2GB of RAM, storage systems capable of holding terabytes of data, databases, payment management systems, order tracking systems, virtual storefront systems, combinations of all the above, and more. And when I say "usable," I really mean "rentable." You pay only for the services (and their accompanying resources) that you use.

This is a key point. You can employ an army of virtual machines, store terabytes of data, or establish an Internet-wide message queue, and you will only pay Amazon for the resources you consume. So if your business needs a cluster of CPUs and several hundred gigabytes of storage to be available, say, every Wednesday for weekly processing, you don't have to keep a room full of servers sitting idly around six days a week. You can use AWS. Therefore, AWS is particularly attractive for business systems with intermittent or transient processing needs.

Nor are the costs unreasonable. For example, storage of 100GB for a month will cost you $15 (at 15 cents per gigabyte per month), not counting 10 cents per gigabyte transferred in. (The Amazon Web Services site provides an online AWS Simple Monthly Calculator, for tallying your monthly costs of using any combination of offered services.)

As hinted above, the kinds of services range from hardware (albeit virtual) to processes. The services fall into three categories: infrastructure services, e-commerce services, and Web information services.

INVESTIGATING INFRASTRUCTURE SERVICES

The infrastructure services are composed of the Elastic Computing Cloud (EC2); Simple Storage Service (S3), a persistent storage system; the Simple Database (SimpleDB), which implements a remotely accessible database; and Amazon's Simple Queuing Service (SQS), a message queue service and the agent for binding distributed applications formed from the combination of EC2, S3, and SimpleDB.

These services provide virtually limitless compute, storage, and communication facilities. They're ideally suited for what might be called "intermittent" applications: those that require substantial compute or storage facilities on an irregular basis (for example, an application that wakes up Friday evening to process data gathered during the week). An application that requires worldwide connectivity – say, a system that processes graphics files and makes the results available to clients across the Internet – can also make good use of infrastructure services. Finally, these services act as excellent proof-of-concept laboratories for large-scale distributed applications. A development house seeking to demonstrate the feasibility of a proposed enterprise-wide application can implement a prototype using the infrastructure services, and avoid hardware costs that, if the prototype is deemed unworkable, would be a net loss.

ELASTIC COMPUTING CLOUD (EC2). Imagine a vast room filled with server systems, all networked together. Sitting at your single workstation, you create a virtual machine image that defines a 1.2GHz processor running Linux with 1.7GB of RAM and a 160GB hard disk, pre-loaded with software you have crafted specifically to number-crunch a large matrix of mined data. You deploy this image to an outside service, which manages those servers. At some future point, a boatload of matrices arrives from your data-mining operations. You instruct the service to instantiate 50 of your virtual machines, and turn each loose on one of the data matrices. Within a few seconds, 50 of those 1.2GHz processors are active and chomping on your data. They finish, deposit their results at a pre-specified storage site, and disappear.

That's EC2 in a nutshell. It's nothing less than a boundless collection of virtual computers that a user can call into existence to perform some processing task. "Boundless," however, does not mean "infinite"; rather, there is no specific upper limit – other than your wallet. Amazon's documentation states that you can commission "hundreds, or even thousands" of virtual machines simultaneously.

Because systems in EC2 are virtual, Amazon provides a range of hardware capabilities. At the low end, you can call for a 1.26GHz Opteron-class machine with 1.8GB of RAM. At the high end (at the time of this writing), you can request a 64-bit multicore system with 15GB of RAM. These specifications are approximations. Virtual machines that you instantiate are rated in EC2 Computer Units (ECUs), which Amazon defines as being equivalent to a 1.0GHz to 1.2GHz 2007 Opteron processor. (The company suggests you do your own benchmarking to determine which instance is best for your particular application.)

An Amazon Machine Image (AMI) consists of an operating system and whatever applications you want pre-loaded when the virtual machine is started. Currently, only Linux is available as an EC2 instance's OS, though this is hardly a limitation. There are quite a few distributions in Amazon's catalog of prebuilt AMIs. Perusing the list, I found ready-to-use AMIs for Ubuntu, OpenSolaris, Centos, Fedora, and many others – all told, more than 100 AMIs ready to go. You can build your own AMI using a free Amazon-provided SDK, but the process is lengthy. It is far easier to select a prebuilt AMI from the catalog, and customize it as necessary. Even so, many available AMIs include software for specific applications; you may well find one that already has much of what you need.

SIMPLE STORAGE SERVICE (S3). Amazon's Simple Storage Service (S3) is effectively a large disk drive in the ether. Strictly speaking, that's 90 percent of everything you need to know about it. It has no directories and no file names – just a big place where you can store and fetch unstructured data in gobs as small as 1 byte or as big as 5GB.

What I call a "gob," S3 calls an "object," and in place of "directory," S3 says "bucket." So when you store a 200KB JPEG on S3, you're putting a 200KB object in a bucket. A given AWS account can own up to 100 buckets. A bucket can hold an unlimited number of gobs, and it can be configured to reside either in the United States or Europe. Presumably, this provides users a comforting feeling of locality, because buckets are available anywhere on the Internet that Amazon is accessible. Cost differences between the two are tiny; a bucket in Europe will run you something like two-thousandths of a cent more per 1,000 requests than in the United States.

Digging a bit deeper, you can think of an object as a three-in-one entity: key, value, and metadata. The key is the object's name, value is its content, and metadata is an array of key/value pairs carrying information about the object. (Access permissions are also associated with an object, but are treated as separate from object storage.) An object's name can be between 3 and 255 characters, and the only constraint that Amazon places on names is that they not confuse URL parsing. Thus, an object with a name of “192.168.12.12” is a bad idea.

Whereas the architecture of S3 is effectively a flat file system, S3’s APIs permit a clever programmer to build apparent subdirectories within a bucket. The hierarchies have to be encoded in the object names, which is less than ideal; however, it’s an artifact that code could simply mask. So, if you want one directory of animals and another of vegetables, you might have object keys such as “animal-cat”, “animal-dog,” “vegetable-beet,” and “vegetable-carrot.” Using the prefix parameter of the List operation, you can restrict retrieved object keys to only animals or only vegetables. More complicated data structures should be kept in Amazon’s Simple Database.

AMAZON SIMPLE DATABASE SERVICE (SIMPLEDB). While Amazon S3 is designed for large, unstructured blocks of data, SimpleDB is built for complex, structured data. As with the other services, the name says it all. SimpleDB implements a database that sits behind a lightweight, easily mastered query language that nonetheless supports most of the database operations (searching, fetching, inserting, and deleting) you’ll likely need. In keeping SimpleDB simple, Amazon has followed the principle that the best APIs are those with minimal entry points: I count seven for SimpleDB.

A SimpleDB database is not exactly like a relational database of the Oracle or MySQL sort. (Amazon’s documentation points out that, if you do need a full-blown relational database, you are free to run a MySQL server on an AMI in the elastic compute cloud.) A SimpleDB database (a “domain” in SimpleDB parlance) is composed of items, and items are composed of attributes. An attribute is a name/value pair. At a minimum, an item must have an ItemName attribute, which serves as the item’s unique identifier. When you issue a query, the result is a collection of ItemName values – to fetch the actual content of the item (the attributes), you perform a Get operation using those values as input.

As simple as it is, SimpleDB packs surprising capabilities. A SimpleDB database can grow up to 10GB and house up to 250 million attributes. You can define up to 256 attributes for a given item, and there is no requirement that all the items in a domain have the same attributes. In addition, a given attribute can have multiple values, so a customer database could store multiple aliases in a single CustomerName attribute.

Finally, SimpleDB is designed to support “real-time” (fast turnaround) queries. To ensure quick query response, all attributes are indexed automatically as items are placed in the database. Also, Amazon’s documentation indicates that a query should take no more than 5 seconds to complete; otherwise, it will likely time out. Amazon does this to ensure that a query receives a quick response, even if a query is malformed to the degree that it would hamper the calling application.

Any computer on the Internet can add or read messages without any installed software or special firewall configurations.

AMAZON SIMPLE QUEUE SERVICE (SQS). Amazon SQS is a message queuing service in the vein of JMS or MQSeries – only simpler. SQS’s most impressive characteristic is its ubiquity. A blurb from Amazon’s documentation reads: “Any computer on the Internet can add or read messages without any installed software or special firewall configurations.” The most likely participants in SQS message transactions are, of course, instantiated AMIs in the EC2.

As with other Amazon Web services, SQS earns its name: Messages are text-only, and must be less than 8KB in length. You can build a working queue with only four functions: CreateQueue, SendMessage, ReceiveMessage, and DeleteMessage. (There are other convenience functions; ListQueues, for example, will list an account’s existing queues.)

SQS queues are designed primarily to support workflows among distributed computer systems, and as such, concurrency management and fail-over are implicit. When a client reads a message from a queue, that message is not deleted; it is simply locked in such a fashion that it becomes invisible to other clients. In that way, if the message represents a specific task to perform as part of a workflow, two clients cannot read the same message and, thereby, duplicate effort. However, if the message is not deleted before a specified timeout, the lock is released. The intent, then, is for the original reader of the message to delete it when the specified work is complete. If the original reader is unable to complete the work (perhaps on account of a system crash), the timeout expires, the message “reappears” in the queue, and a different client can read the message and undertake the specified work.

ENTERING E-COMMERCE SERVICE

The infrastructure services are composed of the Elastic Computing Cloud (EC2); Simple Storage Service (S3), a persistent storage system; the Simple Database (SimpleDB), which implements a remotely accessible database; and Amazon’s Simple Queuing Service (SQS), a message queue service and the agent for binding distributed applications formed from the combination of EC2, S3, and SimpleDB.

AMAZON FLEXIBLE PAYMENT SERVICE (FPS). Amazon’s FPS lets users tap into the company’s existing payments collection infrastructure (for a fee, of course). The idea of FPS is particularly attractive when you read that it will “take on the complexity of managing security and fraud protection” so that you don’t have to.

Two aspects of FPS are especially interesting. First, it supports micropayments, those that involve cents – or even fractional cents. This is useful when business activities involve piles and piles of transactions, each having little monetary value, but the sum of which has measurable value. Imagine selling bubblegum for 10 cents. That doesn’t seem like much – unless you’re selling, say, 100,000 pieces a month. Amazon FPS lets you aggregate micropayments into a single transaction, thus eliminating the problem of transaction costs swamping whatever profits the transactions involve.

FPS’s other interesting aspect is its support for “middleman” operations. That is, you can facilitate a transaction in which you participate neither as a sender (buyer) or recipient (seller). You can, however, take a cut of the action.

There are two ways to employ FPS in your Web application: using an Amazon-supplied “widget” (of which there are two), or hard-coding an interface. The two available widgets are Pay Now and Marketplace (both designed to be easily added to a Web site’s UI).

Amazon has automated the creation of Pay Now widgets. Connect to the online Pay Now Widgets Implementation Guide, and it walks you through the process of building a widget by prompting for various parameters (for example, the destination URL after the payment has been placed), then generates the HTML that you cut and paste into your Web site’s code. The Marketplace widget lets you act as a third party between buyer and seller. In essence, it turns you into an instant reseller. You can use a MarketPlace Widget to let sellers do business on your Web site and pay you for the privilege.

The hard-coded approach is more difficult, but more flexible, as it enables any application that can communicate with a Web service to tap into FPS. You have to express the parameters and processes for payment transactions in a specialized mini-language called Gatekeeper. Once you’ve done that, you install those instructions into the Amazon FPS, which returns a token that is essentially a handle to the Gatekeeper code. Future transactions that employ that token are shepherded by your Gatekeeper program. Details for this process can be found in the online Amazon FPS Technical Documentation.

AMAZON DEVPAY. Suppose you’ve written an amazing application that runs in Amazon EC2. You’re convinced that people would be willing to pay you to use your application. Enter the Amazon DevPay Service.

Amazon DevPay is built on the same payment management infrastructure as Amazon FPS. But DevPay – as its name attests – is designed specifically to let developers charge for the use of their EC2- or S3-based applications.

Interaction with DevPay takes place via tokens (unique identifiers). One token identifies your application; the other identifies a specific user allowed to employ your application. The first, the product token, is generated by Amazon when you register your product with DevPay. That token, combined with a user’s activation key (created when the user signs up with AWS), is implemented during product installation to generate credentials that include the second token, the user token. Your product embeds these tokens in service calls it makes to AWS, and in that way, DevPay tracks your application’s usage by a given customer.

When you register your application with DevPay, you establish how your application is priced. Users can be billed on a metered (pay for what they use) basis, they can be charged monthly, or they can pay a one-time up-front fee. Of course, you have to be careful how you structure your billing. While your clients pay you for the use of your application, you must pay Amazon for the use of its services. So, at the very least, you have to make sure that your customers pay you more than you pay Amazon. Unfortunately, Amazon does not provide a sandbox for testing your application's integration with DevPay, so you have to do your testing with real money. Fortunately, the cost of Amazon services is low enough that this is not a substantial problem.

AMAZON ASSOCIATES AND AMAZON FULFILLMENT WEB SERVICE (FWS). Anyone who has clicked through a site to order something from Amazon has used Amazon Associates: It's the service that lets you sell Amazon stuff from your Web site. You get a percentage – a referral fee – for each sale. There is not much more to be said about Amazon Associates.

A more interesting Amazon e-commerce service, however, is a remarkable kind of inverse of Amazon Associates: Amazon Fulfillment Web Service. With FWS, instead of your selling Amazon stuff, Amazon sells your stuff. Not only that, but Amazon will also warehouse, package, and ship your stuff.

FWS is actually two Web services: inbound and outbound. You use the inbound system to inform Amazon of incoming shipments bound to their warehouse. When a customer orders one of your products, you use the outbound service to inform Amazon of the sale. Based on the details of the order, Amazon packages and ships the product, and even provides tracking information that you and your customer can use to monitor the shipment's status.

Of course, there are warehousing and handling fees involved, but it's a compelling model. A small company, unable to afford warehousing and shipping costs, can "virtualize" those components with Amazon FWS, until that company is large enough to provide them for itself. And any developer interested in exploring the mechanics of the inbound and outbound services will be happy to discover that Amazon has provided "scratchpad" applications – tools that let you exercise simulations of the services.

MECHANICAL TURK. Amazon's Mechanical Turk is a peculiar service. (It is difficult to categorize; I have listed it with the other e-commerce services.) Its name comes from the famous 18th-century robotic chess player invented by Wolfgang von Kempelen. The robot, however, was no robot; inside the machine was a human chess player who operated the mechanism, unbeknownst to the human opponent. The idea of Mechanical Turk, then, is an automated front end, behind whose machinery hides a human.

Only, in this case, it's not just one human; there're lots. Whereas EC2 provides an elastic cloud of computers, Mechanical Turk provides an elastic cloud of humans. But this analogy goes only so far; the computers in EC2 are virtual, the humans of Mechanical Turk are not.

Here's how it works. Suppose you have a big pile of identical tasks that must be performed by humans. Perhaps you have a large quantity of text files that must be translated from one language to another. In the world of Mechanical Turk, you are a requester; you submit your tasks to the Mechanical Turk service, which places them on a kind of global bulletin board. Using that same service, workers log onto this bulletin board, select tasks, perform them, and post the results back to the service. Later you return to the Mechanical Turk, review the posted results, select those that are acceptable, and release funds to pay the workers. In short, the Mechanical Turk service is a middleman between employers and employees.

When I first read Mechanical Turk's description, I thought it was a great idea. It may yet be, but if my perusal of the tasks that are available is any indication, this is not a way to make any appreciable amount of money. Most of the HITS ("Human Intelligence Task," referring to a unit of work) posted paid mere pennies, and reading some of the descriptions gave me the uneasy feeling that workers would be used as human spam-bots.

It is possible that, in the future, Mechanical Turk will become a marketplace of decent work for reasonable money. For now, though, I am confident that I can make more money in less time – and do more good – by mowing the old lady's lawn next door.

> The idea of Mechanical Turk, then, is an automated front end, behind whose machinery hides a human.

WADING INTO WEB INFORMATION SERVICES

Amazon's Web Information Services are essentially query interfaces into extensive databases generated by a mixture of Web crawlers and Web traffic monitors. Data-mining organization can tap into the crawler-produced data to sift through information that is as wide-ranging as the Web itself. The utility of Web traffic data is self-evident to any company or individual interested in user visitation trends to their sites – as well as to related or competing sites.

ALEXAWEB SEARCH. Amazon's Alexa Web Search is the result of partnering between Amazon and Alexa, and it lets you query the information gathered by Alexa's Web crawler bots. The quantity of information available is difficult to gauge; Alexa has been crawling the Web for over a decade, and the Internet is in nonstop growth. Alexa's site says that, while its bots are working constantly, it takes about two months for a complete cycle through the Internet.

When Alexa adds a new Web site document to its database, it indexes about 50 attributes associated with that document. Attributes include the document's language, its Open Document Category, various parsed components of the URL, geographic location of the hosting server, and more. Also available is the document's text, the first 20KB of which is text-indexed. All this is available for searching.

Naturally, searches on such a large database can take time. The Alexa Web Search service is architected so that when you issue a search, the service returns a request ID. You use this ID to track the status of your search's progress. When the search is complete, results are stored in a (possibly gigantic) text file. The text file can be downloaded and "mined" locally.

ALEXA WEB INFORMATION SERVICE (AWIS). The Alexa Web Information Service lets you dip into traffic data gathered by various Alexa tools deployed about the Internet. You can query information for a specific URL, such as site contact information, traffic statistics (going back five years), and more. You can also discover how many links are on a given page, how many URLs are embedded in JavaScript, or the more interesting statistic of how may other sites link to the target ("inward-pointing" links). You can also use AWIS to fetch a thumbnail image of a Web page, useful for displaying pop-ups in response to a cursor hovering over links.

The accuracy of Alexa's data is unclear. The Alexa Web site states that the "traffic data are based on the set of toolbars that use Alexa's data, which may not be representative of the global internet population." Meanwhile, an Amazon Web services representative informed me that Amazon "aggregate[s] data from multiple sources to give you a better indication of Web site popularity." In any case, the ability to scour the text content of whole swaths of the Internet makes the Alexa Web service a profitable vein for Web data spelunkers.

READY FOR THE BIG TIME?

Amazon's Web Services are at once exciting and troubling. The infrastructure services adopt a sort of "mercenary" model of hardware and software horsepower; in theory, you can employ as large an army of computing power as your pocketbook can withstand. All the services offer universal availability – if your network connection can reach Amazon, it can reach AWS. These are two powerful isotopes for fueling large-scale, on-demand, software services.

On the other hand, however, some of the important components are still in beta. SimpleDB, in fact, was in limited beta and not accepting new users at the time of this writing. The description of "beta" is off-putting, as it implies an architecture whose foundation has not yet solidified. And this implication became hard reality when, in June, Amazon's S3 suffered a temporary power outage that affected such high-profile users as the New York Times, whose archives were crippled.

Furthermore, the long-term security of the entire AWS remains to be seen. We can only take Amazon's word that its systems guarantee isolation of one user's applications from another's. Put simply, AWS is only going to work if its users' trust in it is complete. A security breach of any sort would likely be a mortal wound.

Programmers and architects of distributed systems will find the infrastructure pages on the AWS site to be nothing short of a playground. You can spend hours perusing the documentation, tutorials, examples, and references to community-supplied tools and libraries.

The "cloud" services – EC2, S3, SQS, and SimpleDB – are certainly compelling. Real applications are being built atop these virtual technologies. Examples can be found at the Amazon Web Services Elastic Compute Cloud resources page.

Some of the AWS components are of questionable utility. In particular, Mechanical Turk seems to create a built-in incentive to cause tasks to be priced below what they otherwise would. However, even the Turk might be a case of a technology ahead of its time. As the ability to conduct business over the Net continues to improve, perhaps Mechanical Turk will also.

Whether the notion of Amazon's "rentable infrastructure" catches on is unknown. Its failure (should it fail) will not be for lack of information and tools. I will be eagerly prowling the AWS Web site and AWS-relevant blogs to see what creations arise from the enticing techno-tinker-toy set that AWS represents.

Rick Grehan is contributing editor of the InfoWorld Test Center.

WILL THE DOWNTURN ACCELERATE CLOUD COMPUTING?

NO ONE EXPECTS MASS ADOPTION, BUT ANALYSTS ANTICIPATE COMPANIES WILL TAKE A MUCH CLOSER LOOK AT CLOUD COMPUTING SERVICES, THANKS TO THE LOW COST OF ENTRY. **BY TOM SULLIVAN**

Facing uncertain economic times, enterprises may be more likely to turn to cloud computing services -- such as SaaS (software as a service), Amazon-style utility computing, and managed service providers -- for the lower up-front costs, the faster time to market, and the ability to add capabilities quickly without investing in new hardware.

Analysts at Gartner and TABB Group, a research and advisory firm that focuses on financial markets, agree that the current economic downturn is already sparking interest in cloud computing both on Wall Street and elsewhere. “We expect examinations of various cloud services to accelerate,” says Gartner fellow and vice president Ken McGee. “There will be a flight toward looking for lower-cost options.”

SaaS offerings in particular are becoming more attractive. Whereas enterprises “initially took advantage of SaaS collaboration, sales, and marketing applications,” explains Jeff Kaplan, managing director of ThinkStrategies, they “are now exploring SaaS solutions for back-office requirements like expense management, procurement, and supply-chain management.”

Some SaaS vendors claim to be feeling positive effects already. “We see evidence that our sales pipeline is growing because of the recession, rather than in spite of it,” says John Girard, CEO of Clickability, which provides content management as a service. “One of the great things about SaaS and cloud computing more generally is that they’re pay-as-you-go.”

Although companies won’t “immediately move resources to the cloud,” utility-style services from the likes of Amazon, IBM, GoGrid, and Google will follow the trail SaaS applications have blazed and get traction in business units, rather than as broad IT projects, says Robert Mahowold, IDC director of SaaS and on-demand research. “That’s absolutely how SaaS caught on,” Mahowald explains. “That’s what could happen as part of the economic downturn.”

GEARING UP FOR GREATER DEMAND

The biggest IT vendors and cloud providers appear to be gearing up to take advantage of enterprise interest moving forward. Amazon, IBM, Intel, Oracle, and Microsoft have all made waves recently in one way or another.

Last week IBM detailed four new cloud computing centers around the world that are part of its overall effort to help companies architect, design, and construct clouds, explains Dr. Willy Chiu, vice president of IBM high-performance on-demand solutions.

Oracle, at its OpenWorld conference, highlighted its own cloud computing intentions by announcing separate partnerships with Amazon and Intel. Oracle is making its 11g database, Fusion Middleware, and Enterprise Manager software available on Amazon’s Elastic Compute Cloud

(EC2), and the company joined forces with Intel to “accelerate enterprise readiness of cloud computing,” focusing on the key areas of software performance and power efficiency, improved virtual machine security, and standards for provisioning cloud services.

Microsoft corporate vice president of infrastructure marketing, Rob Kelly, put in a placeholder this week, essentially saying that at its Professional Developer Conference in October the company will offer more specifics about its cloud computing plans. “But you could probably guess to a large degree,” Kelly tells the IDG News Service in an interview. “We are a platform company, and we are going to offer platform elements in the cloud.”

IBM and Oracle, too, promise more to come in the near future. Oracle’s product development head Chuck Rozwat suggests the company might seek partnerships with Intel rival AMD as well as arrangements similar to its pact with Amazon for other cloud computing environments.

“You’ll see more announcements coming down with hardware and software services,” IBM’s Chiu says. “We see growing interest in efficiency. There’s also interest in deployment and testing, as well as collaboration in the cloud.”

EXPLORATION BEFORE ADOPTION

IBM customer Sogeti Group turned to the cloud for quick-deployment and collaboration capabilities when it conducted what it called “an innovation jam,” where more than 4,300 employees from 14 countries logged on to a 72-hour event hosted at IBM’s Dublin facilities.

“We started small and figured out how things work,” says René Speelman, a project leader at Sogeti Group, a $1.5 billion division of Cap Gemini. Explaining that the “innovation jam” gave them more confidence in cloud-based services than they had before, Speelman adds that “it’s all about trusting the partner you’re working with.”

Office Depot, meanwhile, turned to SaaS provider Hubspan for its integration services, which enable Office Depot to integrate its own back end with customers’ applications, such as e-procurement and ERP. Hubspan’s service allows Office Depot to “utilize cost-effective resources and make transactions happen very quickly,” according to Glenn Trommer, director of e-commerce and implementation services at Office Depot.

Trommer points out that Hubspan is the only SaaS or cloud service he’d consider for the time being because he wants to control all of Office Depot’s data in-house, which it continues to do. Also, it’s one place where Office Depot really benefits from a value-added service. “There’ve been other situations where it makes sense for us to do it ourselves,” Trommer adds. “It depends on the resources we have.”

Office Depot is by no means alone in cautiously choosing which cloud-based services to adopt, depending on existing internal resources. Thanks in large part to the turbulent economy, in fact, enterprises are refocusing IT money as they concentrate more sharply on the particular value they offer, says Robert Iati, partner and global head of consulting at the TABB Group. “Companies are going to maintain spending where they have their own value proposition,” Iati adds.

That inevitably means some enterprises will reduce reliance on external services and contractors to internally optimize their core business, Iati explains, while others will take the opposite path and look to externalize more. “To some extent, and I hate to say it, this is an opportunity to reduce head count. Those that do so will externalize. Overall, it will tend toward the external.”

Gartner’s McGee says that, as the situation stands today, it’s still too early to tell whether or not enterprise IT shops’ accelerated exploration of cloud services will translate into widespread adoption. “But the vendors will be busy filling out requests for information,” he adds.

Such cloud services are “a more attractive concept now. But it’s more challenging for IT to get the money. Any kind of investment right now will be harder to come by,” TABB Group’s Iati says. “Perhaps in next year’s budgets we’ll see more money going toward cloud services.”

Editor-at-large Tom Sullivan writes features, as well as produces the InfoWorld Daily podcast. He can be reached at tom_sullivan@infoworld.com.

IT NEEDS TO GET OVER ITS CLOUD DENIAL, OR MANAGEMENT WILL GET OVER IT

CLOUD COMPUTING COULD BRING IT DEPARTMENTS CLOSER TO OUR BUSINESS, BUT IT'S MORE LIKELY TO DRIVE US FURTHER APART. BY WILLIAM HURLEY

Cloud computing can potentially unite a user's every whim with a myriad of relevant resources. It can also help IT provide better services, expand its service offerings, and reduce cost. Cloud users make out like bandits with little risk. IT, on the other hand, has a lot more skin in the game.

IT'S ROLE IN DANGER. EMBRACING THE CLOUD CAN HELP

Wake up, IT. You're giving your executives an easy out. Why should they put up with your costly, dysfunctional ways when their fanciful dreams of a utopian infrastructure seem suddenly achievable via the cloud? When any manager with a corporate credit card can go out to the cloud and purchase new services, where does that leave an IT department that failed to satisfy its users? Out in the cold, my friends. Out in the cold.

As an industry, we've never really worked out the user-versus-IT issues that have plagued companies with more employees than can fit in a single conference room for decades. Who hasn't laughed knowingly at the uncanny accuracy of the satire on The Web Site Is Down? All these years, all these conflicts, all these ups and downs, and business users are still flooding the Internet with the same tired complaints.

But now, SMB companies especially can go to the cloud for a generous portion of their software and services needs. Users can put a bug in their bosses' ears about how much you suck and how much the latest cool Web app rocks. Then all it takes is one intrepid peon looking to make a name with a CBA, and you're 75 percent or more toast.

IT NEEDS TO SEE THE CLOUD AS ITS FRIEND, NOT ENEMY

Recently my good friend Ynema Mangum discussed cloud economy in her blog, including George Reese's post on the economics of cloud computing. I'll go into economic detail in a future column, but in brief they agree that deploying within the cloud saves an average of 29 percent over an internal IT department, with an upfront cost at or near $0.

So what's an IT department to do?

First, research. What cloud technologies are your peeps happily using at home? What stable, secure cloud computing solutions are already out there for problems you've been solving the hard way? What crusty, dusty, underperforming corners of your infrastructure have you been meaning to revamp for years? Is there a cloud way to spruce up those nooks?

Second, stop thinking that enterprise and consumer technologies are so different. That line got blurry years ago.

Third and most important, don't stay the course. Be the rock stars that self-regulate and bring glorious, cost-saving proposals before the executive gods like fragrant burnt offerings. Be the indispensable heroes in tough economic times.

Sure, every situation is unique, but all business users demand constantly available, easy-to-use technology – and unlike you they don't use technology for technology's sake. You give them what they want while saving the big bosses money, and you win.

William Hurley is the chief architect of open strategy at BMC Software, Inc.

See the full selection of InfoWorld IT Strategy Guides at http://www.infoworld.com/reports/y/strategy-guide.