Galvin's All Things Enterprise

The State of the Cloud, Part 2

PETER BAER GALVIN

Peter Baer Galvin is the CTO for Corporate Technologies, a premier systems integrator and VAR (www.cptech.com). Before that, Peter was the systems manager for Brown University's Computer Science Department. He has written articles and columns for many publications and is co-author of the *Operating Systems Concepts* textbooks. As a consultant and trainer, Peter teaches tutorials and gives talks on security and system administration worldwide. Peter is also a Lecturer at Boston University and Senior Contributor to *BYTE*. Peter blogs at http://www.galvin.info and tweets as "PeterGalvin". pbg@cptech.com

In the previous edition of Galvin's All Things Enterprise, cloud was the center of attention. In spite of the over-hyping and frequent under-delivery of cloud, it's still an important, new, and evolving area of computing and, therefore, worth some discussion and analysis. The first task was defining cloud computing, and the next was exploring "why cloud"—what does cloud computing bring to the table and why should you care?

This column continues the analysis by giving examples of projects that have successfully used cloud computing. Why did they succeed when others have failed, and what did they gain by using cloud technologies? Of course there are reasons to avoid cloud computing, and those are included as well. The column finishes with a comprehensive list of cloud considerations—what you should consider when determining if a given project should be based on cloud technologies, and whether it would be best implemented in a public cloud, a private cloud, or a hybrid cloud, or built using non-cloud technologies.

Who Is in the Clouds

It seems to me that cloud computing started in the midst of Web 2.0. What at first blush was simple co-location—running an application or an entire business in someone else's datacenter—evolved to running the same on someone else's gear. That hosting model then further evolved into one of running multiple companies' applications within the same infrastructure. Such multi-use required better management tools, monitoring, alerting, and billing. Those became a set of cloud computing tools. Along the way, many Web 2.0 companies did very well by not running their own datacenters.

Take SmugMug as exhibit number one [1]. Their use of Amazon's S3 storage cloud is a case study in how a company can reduce overhead, costs, and complexity [2]. This photo sharing and management site stores the photos via the APIs provided by Amazon, while retaining the customer information and metadata within their own computers. Certainly that's an example of a company that should use the cloud and an application that was ready-made for cloud integration. Necessary cloud attributes (that it's elastic, metered [pay as you grow], shared, and Internet-based) are all present in this case. Of course, SmugMug is one of thousands of Web 2.0 companies that base their computing or their storage on the cloud.

Back during the dot-com boom, companies needed a lot of venture capital, as well as a lot of IT knowledge (either internal or for hire), to move their idea from paper to full-scale execution. Now, the idea still needs IT knowledge, but requires less funding and less investment. Fortunately for companies looking to be in the cloud, there are many providers trying to add to their client list and take their money. The leaders include both tried-and-true companies such as Amazon, Microsoft, IBM, Google, and VMware, and newer companies such as Rackspace, Salesforce.com, Joyent, NetSuite, 3Tera, Terremark, and GoGrid. These companies vary in their offerings, pricing models, and abilities, but all provide IT resources via a pay-as-you-go model.

My Cloud or Your Cloud?

As good as the public cloud model is for some companies, it leaves many other companies wanting. Questions about reliability, security, and performance, as well as regulatory requirements and corporate policies, prevent many companies from utilizing the public cloud products. Given the recent, very public cloud failures [3], companies that depend on their data and computing resources to be available or under their control are choosing not to use the public cloud, or at least not to use it for large swaths of their computing needs.

Such a choice does not mean these companies cannot have public cloud-like features for their projects. Companies still desire the elasticity, manageability, rapid deployment, and even chargeback (or the no-payment-required version known as viewback). What are such companies to do? The solution for them is to use cloud technologies within their own datacenters—private cloud, in the parlance of our times. Sometimes companies want to use private cloud, as well as using public cloud facilities where applicable. This configuration is called hybrid cloud.

Private clouds can look a whole lot like what we used to call "infrastructure," but there are some implementation choices and technologies that can give them cloud-like aspects. Consider one of my clients that had the following problem. Client X had a small, full datacenter. It was traditional in that there were dedicated servers for each application, a small SAN for all important data, a tape library for backups, and a 1 Gb network for interconnection. X was growing, needed to move to a larger datacenter, needed a DR plan beyond just shipping tapes off-site, and needed to move quickly to respond to new business-driven IT initiatives. They chose to use a co-location facility to provide ping, power, and pipe for their racks of equipment. Other improvements included moving to VMware ESX to layer applications across a pool of servers, and using NAS storage to hold their production data as well as the virtual machines. The NAS array also provided them with replication of the data to a second NAS array at a second co-location facility for DR. Moving to a 10 Gb networking interconnect gave them better performance and more room to grow without running out of throughput. The project also involved deploying tools to enable release management, configuration management, capacity management, and change management based on the virtualized environment. Should this project rightly be called a next-generation infrastructure or a private cloud? Both are correct, but because X now has infrastructure-as-a-service (IAAS) and service management for their application deployment, as well as elasticity, I believe it is a private cloud.

As another example, consider client Y. They had an existing business continuance (BC) plan, but that plan failed when it was needed the most—during a disaster. They could not gain access to their normal offices, so declared a disaster and switched over to the disaster recovery (DR) site. Workers started arriving there, and all was well until the number of workers increased. The plan had been tested, but not at the scale of the entire company. The DR infrastructure fell over and work could not proceed. After sorting through the various options, Y decided to upgrade their BC plan and facilities. Rather than have workers go to the BC site, the workers would work remotely, across encrypted tunnels, using a virtual desktop infrastructure (VDI) facility. The applications run within their BC site, but the workers get remote views of their virtual desktops from anywhere that Internet is available. Because of underlying virtualization, production applications are replicated to the BC site, so all apps are kept up-to-date. Internet technologies allow remote access, and by adding more CPU and memory to the BC farm, they can easily scale the facility as needed. Again, this could be labeled with various names, but private cloud is certainly one of them.

Cloud Candidates

Are there certain application and IT initiative aspects that predispose them to be best deployed in a public cloud, private cloud, or left as is on traditional infrastructure? There certainly are trends and success (and failure) stories that show that some projects are better matches for cloud than others. While there are not any absolute rules, a project involving these aspects is probably a good fit for a public cloud:

- Software as a service
- Audio/video/Web conferencing
- Sales/CRM automation
- BC/DR
- Training/demonstration services
- Collaboration
- Email
- Development/test facilities

Other aspects show a tendency to be best left in a private cloud:

- Large data movement
- Sensitive data
- Regulated data
- Complex processes/complex transactions
- Low latency requirements
- Non-x86 applications

Yet other aspects may reveal projects that should be left on existing infrastructure:

- Legacy applications
- Industry-specific applications
- Real-time applications
- Very large (CPU, memory, data) applications

As with cars, your (project's) mileage will vary. Every site is complex, with many decision points, criteria, and experiences. All that will provide guidance on what to place in cloud infrastructure and what to leave as is.

Cloud Considerations

In my experience, it is possible to codify at least some of those "what to run where" decision criteria. The following set of guiding factors can be useful in applying logic to the task of determining how best to run a given application or given facility.

For each of the following technology areas, you should decide whether the area is a factor or not. If an area is a factor, then document why. For example, the operating system might not be a factor because your application can run on any OS, but networking might be a factor because it's 1 Gb and you need to move to 10 Gb for the throughput your application needs. The list includes: operating systems, applications, servers, storage, networking, Internet technologies, virtualization, logging/reporting/analytics, mobile access, seasonal resource use, elasticity/scalability, and any other technology criteria that might be important to your site.

Next is a set of design requirements that could steer the project toward one type of infrastructure or another. This list includes large data movement, non-virtualizable software, low latency, and high customization requirements.

On the financial front, the following areas could be rated in terms of importance, from not important through very important: reducing OpEx, reducing CapEx, licensing cost reduction, ROI requirements, and chargeback/viewback requirements.

Another area to consider is the line of business that the application or facility is destined to support. The LOB again might have importance ratings in areas such as keeping the infrastructure separate from others, required SLA strength, capacity or performance guarantees, the need to control recovery from problems, automation of workflows, and self-service abilities.

In the risk and regulations area, some factors you should consider are the inclusion of validated systems, regulated data, sensitive/proprietary data, regulated systems, HIPAA/SOX or other regulation compliance, corporate security policy requirements, and whether there are strong security needs.

In the final area of project execution, you should think about whether staff members have the skills to design and implement the project within the facility selected, whether they can do so within any time constraints, and whether the team has the knowledge and tools for ongoing monitoring, maintenance, and management of the facility.

Beyond these considerations, don't forget any site-specific, project-specific, or staff-specific requirements or limits on the broad issue of where to run the facility. Experienced IT managers know that beyond those broad decisions, a project succeeds or fails based on the myriad of details it encompasses. Cloud is not a panacea that removes the need for planning and execution. In fact, cloud computing can place more emphasis on management, teamwork, decision-making, and debugging than more standard projects do.

One final note: cloud computing is important and is changing how infrastructure is built and used and how much it costs. That does not mean that cloud computing can solve all problems or is right for all environments or all projects. Sometimes the internal structures of a company or the ways in which roles and responsibilities are divvied up can mean the difference between success and failure of a cloud-centric project. Many companies are finding that between politics and those old structures, much internal change is needed in order for the company to embrace cloud computing.

Tidbits

If you are interested in performance analysis and debugging, especially based on DTrace, you should have a look at the new *DTrace: Dynamic Tracing in Oracle Solaris, Mac OS X, and FreeBSD* by Brendan Gregg and Jim Mauro. It's everything you could want in a DTrace book. See my full review in the Book Reviews section of this issue.

If your interests lie more in the direction of ZFS, then you might want to check out my first video publication. This one is based on the Solaris tutorials I've taught many times for USENIX and elsewhere. The official name is "Solaris 10 Administration Workshop LiveLessons (Video Training): File Systems," but it's 90% ZFS, including both theory and hands-on examples of configuring and using it [4].

On another front, I'm pleased to be part of the relaunch of *BYTE*. As a young lad, I spent many an hour poring over the pages of the venerable magazine, delving deeply into technology details of many aspects of computing. *BYTE* is back, and I'm one of the Senior Contributors there. Have a look at http://byte.com and let me know what you think.

References

[1] SmugMug: http://www.smugmug.com.

[2] SmugMug case study: http://aws.amazon.com/solutions/case-studies/smugmug/.

[3] http://www.crn.com/news/cloud/index/cloud-outages-cloud-services-downtime.htm.

[4] Available at http://www.informit.com/store/product.aspx?isbn=0321753003#Lessons and http://my.safaribooksonline.com/video/-/9780321718372.