BPTrends COLUMN

September 6, 2016

Process Solutions Tom Bellinson

Life in the Cloud: Part III - Scaling Up

A Brief History of Hosting

The process of building software applications and making them available to users had evolved over time. It may be helpful to take a look backwards to see how we arrived at our present state and why. Computing started in earnest in the 1960's with the advent of the mainframe computer. These air cooled behemoths housed every application. At first, other than system utilities, this meant only one or two applications (usually some sort of financial management). Gradually, these systems became better multi-taskers and could run dozens of applications simultaneously.

Next came the minicomputer. It was essentially a scaled down version of the mainframe. It's major advantage (as the name implies) was that it didn't need as much space and didn't need all the fancy plumbing to keep it cool. The concept was still the same - one computer and many applications.

In the 1980's along came the microcomputer (a.k.a. PC). At first, these devices ran one application for one person at a time. That worked out fine because a person can't really do more than one thing at a time, but some applications lend themselves to shared usage. Thus, the local area network (LAN) was born and these single-taskers were now working together with something called a server to share applications. In those early days, the server was only a repository for application files that were shared. It did not execute (run) any of them.

This started to change with the advent of the database server. This was a central application that managed structured (tabular) data on a shared server. The PCs connected to the LAN had software running on them designed to talk with the database server, which did all the work of searching and sorting through the data and passing the results back to the PC for display. This was known as a client/server application because both ends share the load. Other applications began to emerge that were client/server based in which some of the processing was done at the server and some on the client PC. Communications equipment was often added to the LAN to create a wide area network (WAN), thereby allowing remote users to gain access. Most of these early WAN connections were very slow and caused much frustration amongst users.

In the 1990's the worldwide web or Internet came into its own. Developers discovered that they could use web servers, that had been delivering static content, to do more dynamic types of data delivery by coupling them with database servers. Now, anyone with a web browser on their personal computer (which is everyone) could have access to any application being provided.

This idea of web and application servers delivering applications to any browser on any device was a game-changer. This is where we pick up the story currently in progress.

Technology Enabled Process

The realization that this new browser driven mode of operation was a game changer took time to unfold. There were technical glitches that held things back. For one, not all browsers were created equally. Second, as the notion that browsers would do more than deliver static web pages took hold the HTML language that heretofore sufficed was no longer robust enough for everything people wanted to do. Plugins and addons that provided the necessary additional functionality were hindering the universality of "the browser." Eventually, a new version of HTML was developed to rid the world of that problem.

With the way clear to deploy applications to everyone in the world, the new bottleneck became the work happening on the servers. While the browser had hindered wide application deployment, organizations had been using this new technology to improve the deployment and management of applications within their organizations. Software like financial management, customer relationship management (CRM) and human resource management (HRM) was being converted to work in the new browser driven world.

However, with the proverbial gloves off, organizations could start to expand their application base out to their customers. For some organizations, that meant delivering robust functionality to millions of users. The lowly PC was not really built for the task. This has spurred a second revolution since the commercialization of the Internet.

When corporate applications were handling dozens or even hundreds of transactions per second, they could manage things with their own data centers. Building the infrastructure to handle peak usages was easy because it was necessary to plan for some growth anyway. When the average number of transactions per second goes into the 100,000's or even millions per second, infrastructure managers face a whole new set of challenges.

Saved By PaaS

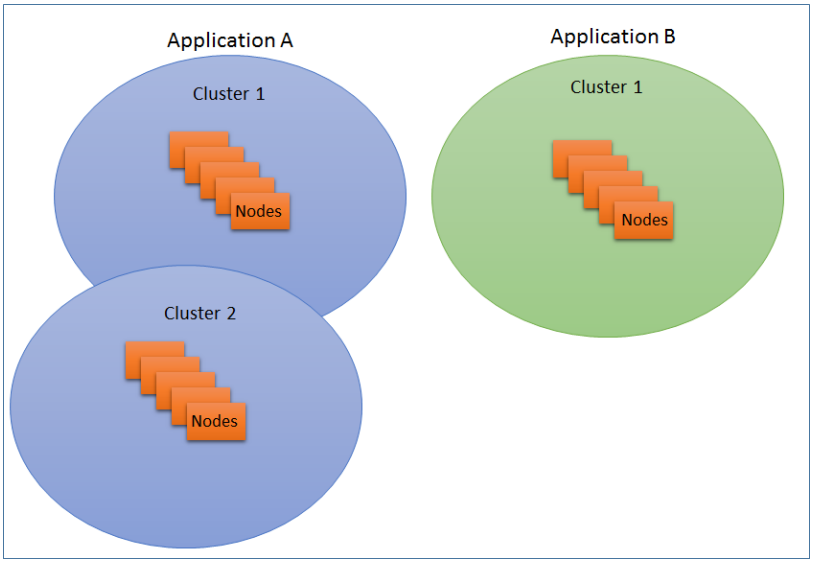
The need to maintain the infrastructure required to handle brief peaks in activity can greatly inflate the cost of deploying an application. Enter platform as a service (PaaS) providers. There are three major players in this space that truly have the capacity to handle almost any scale: Amazon (AWS), Google (Cloud Platform) and Microsoft (Azure). Amazon Web Service is easily the largest supplier of the three even though Google's infrastructure is probably larger (they use most of their capacity for their own purposes).

What these services allow organizations to do is jettison their own data centers in favor of a dynamic environment that can scale to their needs. Even here, there is an evolution taking place. In the early days of these services, organizations simply purchased compute and storage capacity. As these services have evolved, many tools have been made available to support the wide array of application environments that organizations require.

For example, in the area of storage there are needs for the traditional SQL data that is transactional and highly structured, key/value pair datastores like NoSQL and

Graph databases, and streaming storage systems designed to capture huge amounts of real-time data. Many organizations will need more than one of these datastores to handle all of their needs. The PaaS providers can deliver.

I spoke earlier of the need to scale to handle peak requirements. Here again there are tools to manage autoscaling of the computing platform on which an application runs.



The diagram above shows how applications can be organized into clusters containing individual nodes that each act like a computer with varying amount of capability (just like a computer you may buy). Of course, these are all virtual computers that are simulated with software such that multiple nodes could reside on the same physical computer. For this reason, some applications may want to utilize multiple clusters of nodes so that if one cluster fails, another will still be available to continue operations.

As you can imagine, in large scale application environments many clusters may be required to support a single application. With hundreds or even thousands of nodes in operation, managing the environment can be very challenging. One way this complexity can be mitigated is with something called "containers." Containers allow system administrators to define all of the configuration settings of each node and the cluster itself ahead so that a new cluster can be deployed at the push of a button.

Add autoscaling software to containers and the system is now capable of spawning or killing nodes as demand dictates. With computing and storage infrastructure that can scale up and down dynamically, PaaS providers offer something that organizations cannot do with their own data centers -- pay only for the infrastructure they actually use.

PaaS: Next Generation

As I mentioned, even this has begun to evolve. With nodes and clusters, thresholds must be established to determine when to add or subtract resources. These resources are adjusted according to the increment defined by the definition for a container. These increments are often costly, so adding one to gain fractional use of it is still wasteful.

Enter serverless services. We are starting to see storage and computer services that are not defined by servers -- either virtual or real. One of the pioneers in this is Amazon, which is now offering a service called Lambda, which offers the ability to upload software to their system without provisioning clusters or nodes or containers. The service simply provides a continuously scaled environment for an application and customers are billed for the time that it remains in operation at capacity levels tracked by the second.

This type of innovation is pushing the envelope for organizations that are deploying in-house developed software. By using the latest agile software development techniques, in which individual features are coded, tested and deployed into the production environment throughout each day, organizations can now make changes that can have a dramatic effect on their usage without concern for the infrastructure that supports it.

This type of capability opens up possibilities to engage customers in new and different ways with a cycle time that would have been unheard of only a few years ago. High performing software development groups can open new markets, offer new services, and evaluate feedback from changes in near real-time.

Earlier, I pointed out that shifting applications to use the browser was a game-changer. Being able to scale those applications to encompass as many people as may want to use them is yet another game-changer. These tools and techniques are truly only limited by imagination. As we begin to explore the reaches of the Internet of Things (IoT), meaning the integration of applications with all the other equipment and devices that we use, a whole new world of innovation is opening up. The possibilities for automating processes both public and private are mind-boggling. Maybe we'll explore the IoT in a future Column.

Mr. Bellinson has been working in information technology positions for 30 years. His diverse background has allowed him to gain intimate working knowledge in technical, marketing, sales and executive roles.
Most recently, Mr. Bellinson finds himself serving as President of a BPM related software start-up company called UnaPage that provides solutions based on Microsoft SharePoint. From 2008 to 2011 Bellinson worked with at risk businesses in Michigan through a State funded program which was administered by the University of Michigan.
Prior to working for the University of Michigan, Mr. Bellinson served as Vice President of an ERP software company, an independent business and IT consultant, as chief information officer of an automotive engineering services company and as founder and President of a systems integration firm that was a pioneer in Internet services marketplace.

Bellinson holds a degree in Communications with a Minor in Management from Oakland University in Rochester, MI and has a variety of technical certifications including APICS CPIM and CSCP.