waters

Building the Next-Generation Data Center

Former Wachovia technologist Tony Bishop lays out the nine steps for a real-time infrastructure.

I am going to take you through a nine-step process that Wachovia went through over a three-and-a-half year period where we were able to achieve more than $100 million in annualized cost savings in our data center infrastructure. We were able to accelerate the time to provision capacity by five times and rapidly introduce new technology.

One of the things we were most proud of was the fact that we brought in 15 new technologies into our infrastructure that we were able to put into production and repeat multiple times.

For the first step, we created an economic model. This puts it into the language that the business speaks: Demand and supply. The business creates demand that consists of applications and information that eat up the supply of infrastructure. We broke this down and analyzed the users and the services and applications they need. What are they doing? They are running some sort of task or executing some sort of transaction.

Then we asked, what are the factors they care about on the demand side? Those factors include business priority, as in, does this product, service or transaction have a higher margin or am I more worried about volume? Is this an important client? What are the response times? This is how we broke down the demand side.

On the supply side, we looked at three things: execution, capacity, and allocation. For rule-driven execution, do you do that today? Is it based on rules of what's being asked or how things get executed? Or do you just pray and spray?

Then we looked at the economics, as in what the business was willing to pay. Let's break down this allocation charge and have the dialogue. Let's look at the business side in the trading world: They do transaction cost analysis. Let's do that in the infrastructure world related to the applications in the environment that you are running.

The second step was to conduct demand mapping. Here, we used multiple tools. We looked at them literally as a paper-based exercise to start. We said we would pull from the best of industry practices out there, and created a paper-based service contract. We sat down with the business operations people who support the business and certain key users within each of those business groups. We talked about a day in the life of that business. Where do they currently see service level breaches? Where do they see growth in their business? Where are their time performance issues? Where are their process bottlenecks?

We used tools like Tideway Systems that allowed us to discover our applications and their associated dependencies. We used another tool called Evident from Evident Software that allowed us to show usage of consumption behavior. We also created our own little mathematical tool to examine excess CPU memory, input/output (I/O), and bandwidth at different times.

Step three was to implement supply consumption management. By leveraging the Evident tool, we were able to see, based on location, department or organization, and based on the various application services, who was using how much and so on. It did this by talking to Cisco, BMC, and Tivoli. It talks to all the various tools that we already had in place but it pulls the data and helps us correlate it. It looks at the trends like who is using what for how long and when. I started to create a baseline and a benchmark to go after these things.

In step four, we had to translate demand and map it with the optimized supply footprint. This is probably where people either loved us or hated us. With the objective data, we used a tool that gave us the user experience. Instead of synthetic simulations of transactions, we used the tool to show, in real time, the browser to the Web server to the application server to another application server to the messaging environment to the database and back. It showed how things were executing, how long it was taking, the elapse time and the end-user experience. We did it both from a baselining and a triage view and then we did it by using the tool to say, "I am delivering the service level that we signed up for together." When we did that, we were able to look at our demand inputs.

Step five was to build a data center in a box. This is what we called our pod in a box. Instead of having storage on one part of the data center floor, and then compute over here while my file servers are sitting over there, we started creating a pod in a box where everything was self-contained. We utilized Cisco Systems switches; actually we were one of the first to test out the 7700 switches that they had just introduced. We used blade solutions vendor Verari's vertically cooled racks. We did this because we could put 66 blades in the rack by simply removing the floor tile and put two times more into a rack compared to my Hewlett-Packards and my IBMs. No offense to my IBM friends-I love your stuff-but the rack allowed me to put more into it.

This is real stuff. We hadn't quite got there but search giant Google and retailer Amazon rent out storage, and Microsoft is trying to figure out what it is going to do in this space. We went and toured data centers, and you see these containers literally sitting here. That is a data center in the box. It literally has 1,400 servers and 15 petabytes of storage. That is a shipping container you can stick on an 18-wheeler. It's available today.

In step six, we had to implement virtual execution management. The problem with grid computing that I found was everyone still uses the high-performance computing (HPC) definition only. This is problematic when you consider that there is more general purpose or larger sets of infrastructure outside of the HPC space that need dynamic control. It needs dynamic control of the session, task or input and it needs it to be able to be brokered to the most optimal platform.

We were able to create a runtime execution framework using Data Synapse's fabric server product. But the only one that is close with a fabric server product is IBM and its product WebSphere XD, but it is predominately within a cluster. What we found was that it has the ability to take in the Web servers, application servers and our Microsoft SharePoint systems, our document systems and our FileNets-all these environments that eat up a lot of infrastructure. We were able to use this mechanism to look at session queues, arrival rates in the session queues, authentication and entitlement associated with what is coming in the session queues. We then determined the rules we set from an infrastructure standpoint to say, "These things need to run on this commodity disk or this commodity platform or this specialized appliance." Sixty percent of the applications we migrated into our global utility-which was 15,000 nodes with hundreds of applications and hundreds of

services end to end-run like a utility with allocations on a temporary reserve, supplemental basis. We were able to control things like this output of information and this file content should go on this commodity storage environment. That, by the way, doesn't need to be an expensive RAID EMC system or an expensive application, depending on what your storage strategy is.

For step seven, you must build in orchestration and lifecycle management of how you build, patch, provision, change, update, repurpose and so on. If you do your demand and supply mapping, you can start to orchestrate the daytime shift and nighttime shift. Haven't we been experiencing the same problems for 20 years? Well, they did this on the mainframe. Now tools are finally catching up to what's out there for the distributed environment. There are a lot of automation opportunities with orchestration and lifecycle management around that infrastructure. I would pose that you take a look to simplify and make it more dynamic.

In step eight, run the data center like a utility. In running a utility, if you can create that ability in an automated way to provide that tooling, it's a way to run the data center like a utility. Then you can also do the charge-backs, allocations, and transparency into who is using what, where, when and how. You can determine the service levels being met, and so on. This is how you can start to transform and build that bridge between infrastructure and applications in the business side that doesn't exist in most places.

And finally, step nine is to manage it like a portfolio. There are different tools out there, such as CA, Compuware, Tivoli and so on. You have to bring it together and say there are classes of assets that can be types of appliances, types of infrastructure, performance, cost factors, and more. You start to manage it more like a portfolio where you also have infrastructure services.

When you start thinking of it as a portfolio, you can create infrastructure as a service and software as a service, and you can build your service catalogs and workflows. The opportunity is there and it makes it a lot easier to control what you are doing in your infrastructure footprint. As long as the service is delivered and you start to build credibility and a following, it makes it a lot easier to say, "Now we are going to shrink or optimize this, or change this."

Those are the nine steps that we went through that I wanted to pose to you today. Thank you.

Tony Bishop is the former senior vice president, chief architect for Wachovia.