Cloud is Not a Big Switch

Lori MacVittie, 2009-10-08

Why Carr's analogy doesn't describe today's cloud environments and how SOA can get us closer to what he describes

Back when cloud first starting drifting in to obscure the computing landscape there were a lot of parallels drawn between it and grid, and a lot of analogies used to explain the concept behind it. Cloud computing is most often analogized using Nicolas Carr's analogy of the cloud as an electrical grid; that's always bothered me at almost a visceral level. But I could never articulate ***why*** well enough and a lot of smart people told me that if I just thought about it a while I'd figure it out. So I left it alone.

As is often the case, the discordance floated around in the back of mind long enough and when the answer floated to the top I understood why it bothered me: an electrical grid delivers raw resources (electricity) and the cloud delivers complete applications – not compute resources. See, Carr's analogy was more about business models and commoditization of computing resources, not a technical description. But without any good technical description available to non-technical folks, they'll take what they can get.

Interestingly enough, though, I think we can get technically closer to Carr's analogy if we apply SOA principles to application design in the cloud.

WE DELIVER APPLICATIONS, NOT RESOURCES

The reason cloud (today, at least) doesn't technically fit Carr's electrical grid analogy is that you'd have to have ***compute resources*** delivered via the Internet in the same way electricity is delivered to your house. It's a ***resource*** that's only useful when consumed by something requiring power: like an appliance. Compute resources are RAM, disk, and CPU cycles. If ***they*** were delivered via the Internet and then consumed by something requiring them, i.e. an application, ***then*** we'd be living in Carr's analogy. But that doesn't happen. Your servers and desktops don't reach out into the great beyond and allocate more memory or CPU cycles from "the cloud" when necessary. That would require an entirely new operating system model that simply doesn't exist at the moment regardless of the haphazard slapping of "cloud" tags onto operating systems and products that is happening right now. In fact, given the physical limitations of how computers work, it ***can't*** exist and won't exist in that form unless there are some really interesting developments in the future. The best we can hope for is the distribution of discrete application logic across compute resources regardless of location. Our servers and desktops – if enabled with a real cloud operating system - could reach across the Internet and distribute application logic ***into*** the cloud but this is something that, unless technology advances, isn't going to happen either in today's cloud environments.

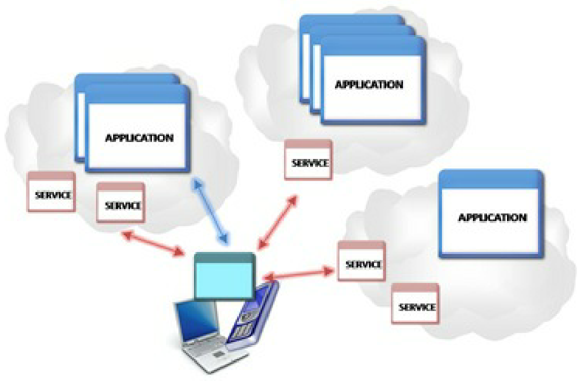
Furthermore, applications aren't even really distributed in the cloud today; not really. We clone applications in the cloud, they sit behind a load balancer or application delivery network, and the ***requests*** are distributed across those instances, but the actual applications aren't really distributed because they are fully contained within a ***single virtual container.*** The entirety of an application's logic is running on a single system, with well delineated lines of separation between their architectural tiers.

That's the cloud as we know it today, and it's nothing like an electrical grid or even close to a truly distributed system as is often described as "a grid". The next step in getting closer to grid is for application logic to be distributed across available compute resources on-demand, as necessary, automatically. Analysts and pundits might refer to this as "workload distribution" which today is impossible with monolithic application architectures because no automated system is able to parse out the discrete application and business functions that make up an application and automatically – and intelligently – distribute them on-demand to compute resources across the cloud.

But *we* can, if we're willing to re-embrace SOA and its core principle of decomposition, and thus get one step closer to making The Big Switch reality.

HOW SOA MAKES CLOUD MORE GRID-LIKE

SOA was designed to decompose, i.e. break down, applications into their composite business functions such that commonly used sub-processes or functions could be reused and scaled out more easily. The benefits of this approach were reuse, agility, a reduction in the length of the development life cycle, and most importantly consistency across all applications leveraging shared business functions.



Now if we actually used SOA to its full potential and then distributed those services in the cloud, we'd be a whole lot closer to a grid – and Carr's analogy – than we are today. That's because the disparate pieces of an application – its business functions – really would be distributed across the cloud in a more grid-like fashion. Individual business functions would be scaled as needed, and the ***application*** would be distributed instead of simply individual requests. With the right infrastructure – enabled with application-switching capabilities – every service could be distributed and be invoked on-demand according to the specific needs of the user, the application, and the conditions of the network and its environs at that very moment. In other words, contextually.

While we wouldn't be pulling compute resources to us as we do electricity, we would be utilizing compute resources on a more granular level because the application has been properly decomposed into its most elemental and basic pieces. The application might be loaded from one "cloud" and execute using the compute resources available on your desktop, but discrete pieces of logic (services) might be executing in one cloud or even another. The application executes as a set of distributed services – potentially in parallel when possible if we so desire – which is much closer to the concept of grid computing and thus closer to Carr's analogy.

SOA, properly applied, turns the growing plethora of "clouds" into The Cloud and makes it possible to ignore cloud boundaries and treat all available compute resources as compute resources. SOA needs no new standards, because SOAP and WSDL and WSIL and UDDI already form the basis for the application interoperability necessary to create such a beast. Services could be deployed not only across similar cloud models, but across disparate cloud models. There's no reason an application deployed on Microsoft Azure, for example, can't invoke services deployed in IaaS provider Blue Lock's cloud ***and*** services deployed on Amazon's cloud while simultaneously also including services from SaaS provider Salesforce or Google.

The standards for this type of intercloud composite application ***already exist***. The logistics of deployment and management and costs may make such an application financially unfeasible, but it is ***technically*** possible.

The Cloud isn't a grid itself, but if we add a dash of SOA to the mix, we can take The Cloud one step closer to truly distributed, grid-like computing.

Related articles & blogs:

- Use The Source, Luke!
- Cloud Computing Makes Servers Obsolete
- Governance: Service Catalogs and the Cloud
- SOA Announces Comeback Tour
- Have a can of Duh! It's on me
- Forklifts, Rip and Replace, and Other IT Fairy Tales
- Intercloud: The Evolution of Global Application Delivery

- Cloud Balancing, Cloud Bursting, and Intercloud
- The Infrastructure 2.0 Trifecta
- The Context-Aware Cloud

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com