

Amazon EC2: Not your father's enterprise cloud

2010-10-06 09:10:04 Nate

OK, so obviously I am old enough that my father did not have clouds back in his days, well not the infrastructure clouds that are offered today. I just was trying to think of a somewhat zingy type of topic. And I understand *enterprise* can have many meanings depending on the situation, it could mean a [bank that needs high uptime](#) for example. In this case I use the term *enterprise* to signify the need for 24x7 operation.

Here I am, once again working on stuff related to “the cloud”, and it seems like everything “cloud” part of it revolves around EC2.

Even after all the work I have done recently and over the past year or two with regards to cloud proposals, I don't know why it didn't hit me until probably in the past week or so but it did (sorry if I'm late to the party).

There are a lot of problems with running traditional infrastructure in the Amazon cloud, as I'm sure many have experienced first hand. The realization that occurred to me wasn't that of course.

The realization was that there isn't a problem with the Amazon cloud itself, but there is a problem with how it is:

- Marketed
- Targeted

Which leads to people using the cloud for things it was not intended to ever be used for. In regards to Amazon, one has to look no further than their SLA on EC2 to immediately rule it out for any sort of “traditional” application which includes:

- Web servers
- Database servers
- Any sort of multi tier application
- Anything that is latency sensitive
- Anything that is sensitive to security
- Really, anything that needs to be available 24x7

Did you know that if they lose power to a rack, or even a row of racks that is not considered an outage? It's not as if they provide you with the knowledge of where your infrastructure is in their facilities, they rather you just pay them more and put things in different zones and regions.

Their [SLA](#) says in part that they can in fact lose an entire data center (“availability zone”), and that's not considered an outage. Amazon describes [this](#) as an “availability zone”

Additionally, they are physically separate, such that even extremely uncommon disasters such as fires, tornados or flooding would only affect a single Availability Zone.

And while I can't find it on their site at the moment, I swear not too long ago their SLA included a provision that said even if they lost TWO data centers it's still not an outage unless you can't spin up new systems in a THIRD. Think of how many hundreds to thousands of servers are knocked off line when an Amazon data center becomes *unavailable*. I think they may of removed the two availability zones clause because not all of their regions have more than two zones (last I checked only us-east did, but maybe more have them now).

I was talking to someone who worked at Amazon not too long ago and had in fact visited the us-east facilities, and said all of the availability zones were in the same office park, really quite close to each other. They may of had different power generators and such, but quite likely if a tornado or flooding hit, more than one zone would be impacted, likely the entire region would go out (that is

Amazon's code word for saying all availability zones are down). While I haven't experienced it first hand I know of several incidents that impacted more than one availability zone, indicating that there is more things shared between them than customers are led to believe.

Then there is the extremely variable performance & availability of the services as a whole. On more than one occasion I have seen Amazon reboot the underlying hardware w/o any notification (note they can't migrate the work loads off the machine! anything on the machine at the time is killed!). I also love how unapologetic they are when it comes to things like data loss. Basically they say you didn't replicate the data enough times, so it's your fault. Now I can certainly understand that bad things happen from time to time, that is expected, what is not expected though is how they handle it. I keep thinking back to [this](#) article I read on [The Register](#) a couple years ago, good read.

Once you're past that, there's the matter of reliability. In my experience with it, EC2 is fairly reliable, but you really need to be on your shit with data replication, because when it fails, it fails hard. My pager once went off in the middle of the night, bringing me out of an awesome dream about motorcycles, machine guns, and general ass-kickery, to tell me that one of the production machines stopped responding to ping. Seven or so hours later, I got an e-mail from Amazon that said something to the effect of:

There was a bad hardware failure. Hope you backed up your shit.

Look at it this way: at least you don't have a tapeworm.

-The Amazon EC2 Team

I'm sure I have quoted it before in some posting somewhere, but it's such an awesome and accurate description.

So go beyond the SLAs, go beyond the performance and availability issues.

Their infrastructure is "built to fail" which is a good concept at very large scale, I'm sure every big web-type company does something similar. The concept really falls apart at small scale though.

Everyone wants to get to the point where they have application level high availability and abstract the underlying hardware from both a performance and reliability standpoint. I know that, you know that. But what a lot of the less technical people don't understand is that this is HARD TO DO. It takes significant investments in time & money to pull off. And at large scale these investments do pay back big. But at small scale they can really hurt you. You spend more time building your applications and tools to handle unreliable infrastructure when you could be spending time adding the features that will actually make your customers happy.

There is a balance there, as with anything. My point is that with the Amazon cloud those concepts are really forced upon you, if you want to use their service as a more "traditional" hosting model. And the overhead associated with that is ENORMOUS.

So back to my point as to the problem isn't with Amazon itself, it's with whom it is targeted to and the expectations around it. They provide a fine service, **if you use it for what it was intended**. EC2 stands for "elastic compute", the first thing that comes to my mind when I hear that kind of term I think of HPC-type applications, data processing, back end type stuff that isn't latency sensitive, and is more geared towards infrastructure failure.

But even then, that concept falls apart if you have a need for 24x7 operations. The cost model even of Amazon, the low cost "leader" in cloud computing doesn't hold water vs doing it yourself.

Case in point, earlier in the year at another company I was directed to go on another pointless expedition comparing the Amazon cloud to doing it in house for a data intensive 24x7 application. Not even taking into account the latency introduced by S3, operational overhead with EC2, performance and availability problems. Assuming everything worked PERFECTLY, or at least as good as physical hardware – **the ROI for the project for keeping it in house was less than 7 months** (*I re-checked the numbers and revised the ROI from the original 10 months to 7 months, I was in a hurry writing this morning before work*). And this was for good quality hardware with 3 years of

NBD on site support. This wasn't scraping bottom of the barrel. To give you an idea on the savings after those 7 months it could more than pay for my **yearly** salary and benefits, and other expenses a company has for an employee for each and every **month** after that.

OK so we're passed that point now. Onto a couple of really cool slides I came up for a pending presentation, which I really thing illustrate the Amazon cloud quite well, another one of those "picture is worth fifty words" type of thing. The key point here is **capacity utilization**.

What has using virtualization over the past half decade (give or take..) taught us? What has the massive increases in server and storage capacity taught us? Well they taught me that applications no longer have the ability to exploit the capacity of the underlying hardware. There are **very** rare exceptions to this but in general over the past I would say at least 15 years of my experience applications really have never had the ability to exploit the underlying capacity of the hardware. How many systems do you see averaging under 5% cpu? Under 3%? Under 2%? How many systems do you see with disk drives that are 75% empty? 80%?

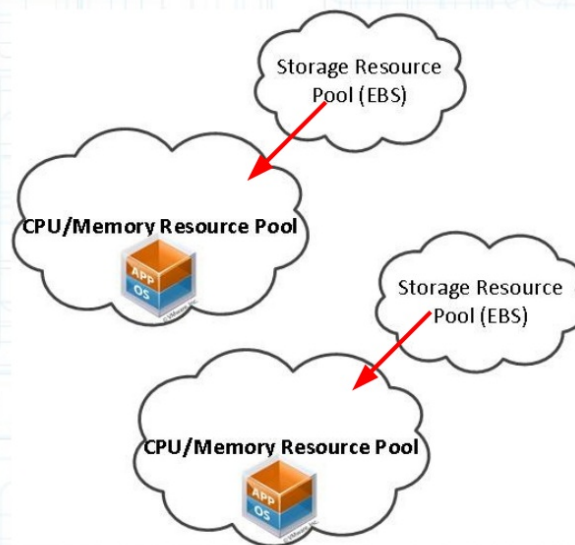
What else has virtualization given us? It's given us the opportunities to logically isolate workloads into different virtual machines, which can ease operational overhead associated with managing such workloads, both from a configuration standpoint as well as a capacity planning standpoint.

That's my point. Virtualization has given us the ability to **consolidate** these workloads onto fewer resources. I know this is a point everyone understands I'm not trying to make people look stupid, but my point here with regards to **Amazon is their model doesn't take us forward — it takes us backward**. Here are those two slides that illustrate this:

Amazon Cloud Resource Model

The Amazon resource model is very similar to managing physical servers in that each system has a dedicated set of resources, whether it is CPU, Memory or storage.

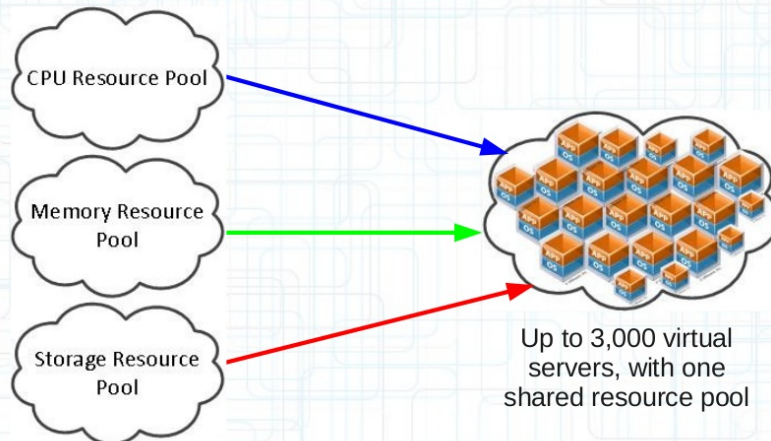
There is no ability to share resources with other virtual servers. This drives capacity utilization down, and becomes less efficient the larger you grow.



Two virtual servers, each with it's own pool of resources, not sharable *(Click image for full size)*

And the next slide

Internal Cloud Resource Model



Hosting a cloud internally drives capacity utilization significantly higher by leveraging the ability to share resources between servers, while simultaneously providing the ability to prioritize resource allocation to guarantee minimum or maximum capacity for any given application. This becomes more efficient the larger you grow.

(Click image for full size)

Not all cloud providers are created equal of course. The [Terremark Enterprise cloud](#) (not [vCloud Express](#) mind you), for example is resource pool based. I have no personal experience with their enterprise cloud (I am a [vCloud express](#) user for my personal stuff [2x1VCPU servers - including the server powering this blog!]). Though I did interact with them pretty heavily earlier in the year on a big proposal I was working on at the time. I'm not trying to tell you that Terremark is more or less cost effective, just that they don't reverse several years of innovation and progress in the infrastructure area.

I'm sure Terremark is not the only provider that can provide resources based on resource pools instead of hard per-VM allocations. I just keep bringing them up because I'm more familiar with their stuff due to several engagements with them at my last company (none of which ever resulted in that company becoming a customer). I originally became interested in Terremark because I was referred to them by 3PAR, and I'm sure by now you know I'm a fan of 3PAR, Terremark is a very heavy 3PAR user. And they are a big VMware user, and you know I like VMware by now right?

If Amazon would be more, what is the right word, honest? up front? Better at setting expectations I think their customers would be better off, mainly they would have less of them because such customers would realize what that cloud is made for. Rather than trying to fit a square peg in a round hole. If you whack it hard enough you can usually get it in, but well you know what I mean.

As this blog entry exceeds 1,900 words now I feel I should close it off. If you read this far, hopefully I made some sense to you. I'd love to share more of my presentation as I feel it's quite good but I don't want to give all of my secrets away 😊

Thanks for reading.