cloudscaling

Converged Storage, Wishful Thinking & Reality

by Randy Bias, Co-founder & CEO Cloudscaling

Introduction

One of the things I find particularly frustrating about the current open cloud marketplace is the surfeit of folks who are relatively new to infrastructure. With both buyers and sellers, you see a certain amount of reinvention of the wheel combined with wishful thinking. This is particularly egregious in the area of distributed cloud storage, where instead of looking for practical solutions to the challenges, we see a number of folks promulgating the idea that there are simple answers to complex storage problems.

I have news for you, there aren't. You can pick the red pill, the blue pill, or even both depending on what you are trying to do. What you cannot do is pick one pill to cure all ills.

Distributed File Systems are NOT a Panacea: There is No Pill

Wishful thinking typically comes in the form of distributed storage solutions, like Ceph and GlusterFS, that promise to deliver all things to all people, conveniently ignoring the realities of the last 30 years. I don't mean that these technologies are bad; they are not. What I mean, is that how the sellers market them and how the buyers want them to behave is disconnected from reality.

Distributed storage technologies are possible and valuable, but like anything, you have to understand the problems you are solving before you deploy them. And when you deploy them correctly to solve the real storage problems that exist in today's private clouds, you will find that we are still where we were before:

> *Use case matters and no amount of technology can fix immutable laws of economics, technology, and business dynamics.*

Put more simply, you can't buy one storage solution to solve all storage problems.

Enterprises Use "Tiered Storage"

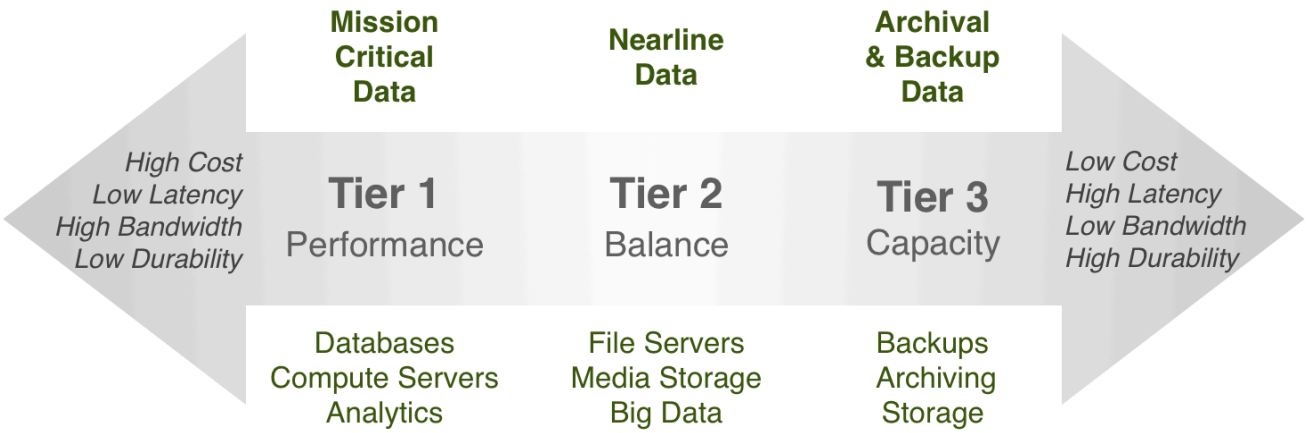
Quite a while ago, folks figured out that there was a need for storage for a variety of applications, but that these use cases fell into several key camps:

- Mission critical data (PERFORMANCE): Fast and accessible regardless of cost
- Nearline (BALANCE): Easy to access, but cost effective data
- Archival and backup (CAPACITY): Slow to access, but as cheap and reliable as possible

Storage that solved for these were typically referred to as “Tier 1”, “Tier 2”, and “Tier 3” storage by most storage sellers and buyers. If we were to look at the detailed requirements for these use cases it would look like this:

	Cost	Latency	Bandwidth	Availability	Durability	MTBF	Consistency
Tier 1 Performance	$$$$$	Low	High	High	Low	High	Strong
Tier 2 Balance	$$$	Medium	Medium	Medium	Medium	Medium	Strong or Weak
Tier 3 Capacity	$	High	Lower	Low	Extreme	Low	Eventual

For most purposes you can think of tiered storage as on a continuum upon which your use case would fall, dictating which tier makes sense:



Now, this isn’t exact as the variety of solutions you might find for tiered storage do have a tendency at times to blur the lines; however, you would find that this is how most enterprise storage engineers and enterprise storage vendors think of the world.

Tiered Storage in the Cloud Era

What about in today's cloud era? Well, the process of tiering storage looks much the same, but the actual solutions have changed significantly. It looks more like this:

	Non-Cloud	Cloud
Tier 1 Performance	SAN, some NAS →	Elastic Block Storage
Tier 2 Balance	NAS, some SAN →	Distributed File Systems
Tier 3 Capacity	Tape Backup →	Object Storage, some Tape

Why the change? Put simply, the general desire amongst mid-tier and larger enterprises is to deal with the ongoing storage sprawl now that scale for enterprises is becoming significant. Elastic Block Storage (EBS) is simply an approach to abstracting away SAN/NAS storage into a more manageable "pooled" model. Some will refer to this notion as storage virtualization, software-defined storage (SDS), or similar.

What's important is that the contract is still similar to the existing Tier 1 block storage solutions EBS is replacing. Low latency, high throughput, and high cost for mission critical applications. This replaces the SAN/NAS model by mostly being a better, more abstracted, and hopefully more "scale out" solution than the prior approach.

The next time you get a chance, ask your typical enterprise storage seller how they are dealing with all of the pairs of NetApp and EMC boxes. Mostly they will tell you they aren't. It's manual, it's painful, and it's very expensive. Enterprises want this resolved.

Enter the distributed storage myth.

The Distributed Storage Myth

The distributed storage myth promises the pooled model for storage, but everywhere, converging all three tiers of storage. Now, you, imaginary buyer, buy one storage solution and use it for all parts of your cloud storage: Tier 1 through Tier 3.

This is the false promise of distributed storage. At the very least, you will deploy the same distributed storage technology three times to reduce the size of failure domains, but more likely you will deploy three different storage technologies three separate times, once for each tier, that are designed specifically to solve the problems in that tier.

That's still a massive advance over the way things have been done in the past. You will still reduce costs moving to a cloud storage model. It's just that it won't be done by having one massive, infinitely scalable storage pool. Such a thing can never exist.

Here's why.

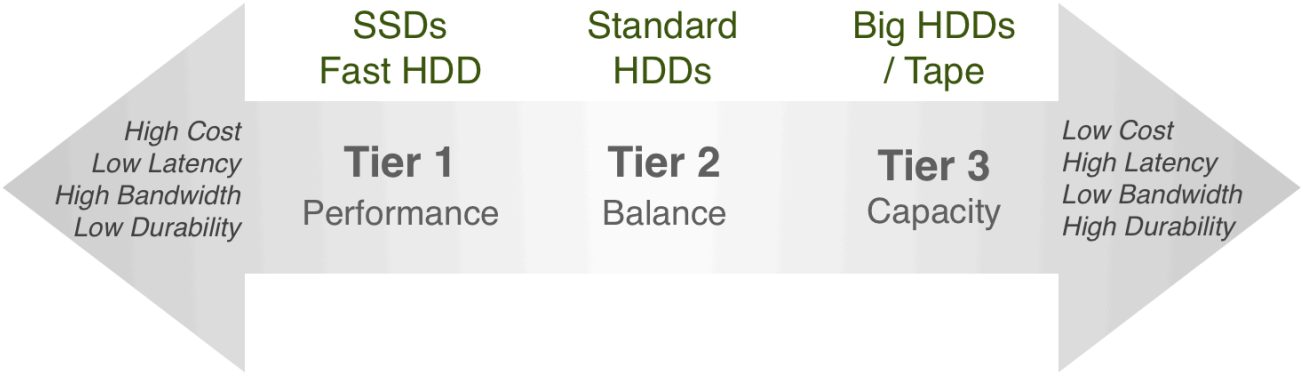
The Problem

The fundamental error in thinking should be apparent. How do you build storage that is high capacity while also low latency? Low cost, while also high bandwidth? The answer is that you can't. Tiered storage can't be converged because of:

1. Cost - The economics of disk drives, SSDs and tape
2. Technology - The wide variance of technology applications for spinning disks, SSDs and tape
3. CAP - The unremitting truth of the CAP theorem

1) The Economics

Disk drives, SSDs, and tape, come in all shapes, sizes, and flavors, but typically they fall along that spectrum of capability.



The bottom line is that technology that solves these particular problems is a best fit somewhere in this spectrum, but will not span the entire spectrum. You can't buy cheap, high capacity, SSDs that don't ever wear out. You can't buy cheap, high capacity, low latency tape drives.

2) The Storage Technologies

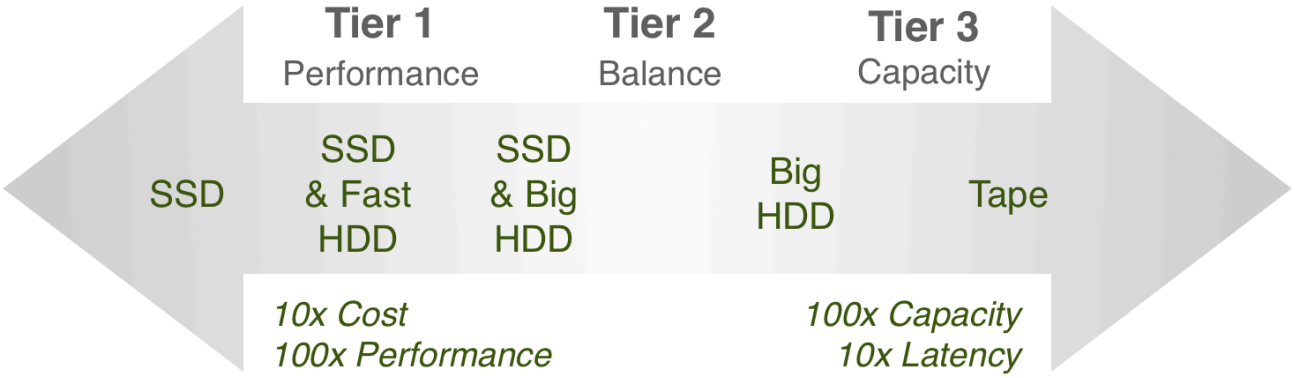
If you want to optimize for the lowest cost per IOPS, you need SSDs. If you want to optimize for the lowest cost per GB, you need high capacity tape or high capacity, low speed "green drives". If you want somewhere in the middle, you compromise appropriately.

SSDs are fast. Screaming fast. Ultra low latency and great bandwidth throughput (mostly). But they are expensive. Very expensive compared to the other storage technologies out there. SSDs are a best fit for Tier 1 storage.

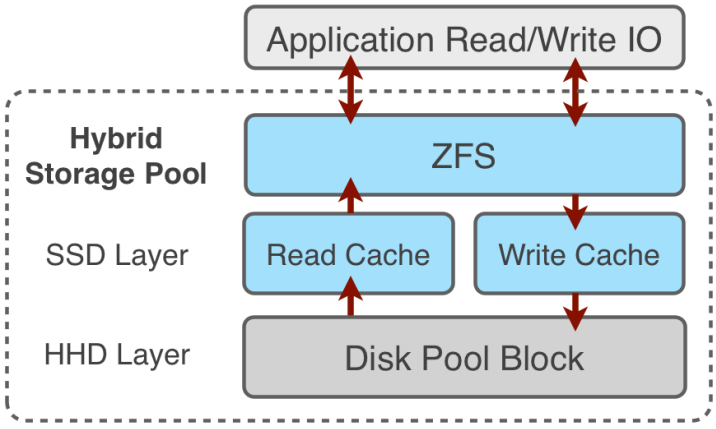
Spinning disks come in a wider variety of flavors than SSDs, but most of the 15K and 10K RPM high end of the market are being put out to pasture in favor of SSDs, really only leaving 7200 RPM disk drives and so-called "green drives", which usually operate around 5000-6000 RPM, reducing power consumption. The

tradeoff for these drives vs. 10/15K RPM drives and SSDs is that they give up speed in favor of cost and capacity. Spinning disks are a best fit for Tier 2 storage and historically the high end of this technology (10K/15K drives) were used for Tier 1. Sometimes these are used for Tier 1 when you have a hybrid storage pool (more below). Sometimes extremely inexpensive desktop SATA drives are used to allow spinning disks to be more appropriately deployed for Tier 3 applications. This is the most flexible technology option.

Tape still exists and for good reason. If Google needs tape, then you might as well. Tape is slow, but has made steady progress over the years and tape vendors have driven prices down consistently. There is some question as to whether tape disappears eventually, but we just don't know when that will happen. Tape is still fabulous for long term, durable, cheap storage. And the way that tape backup systems work means that you usually have versioned copies on multiple physical devices that are offline until you need access. That means that a software error that might damage a piece of data won't be replicated, ruining the other copies.



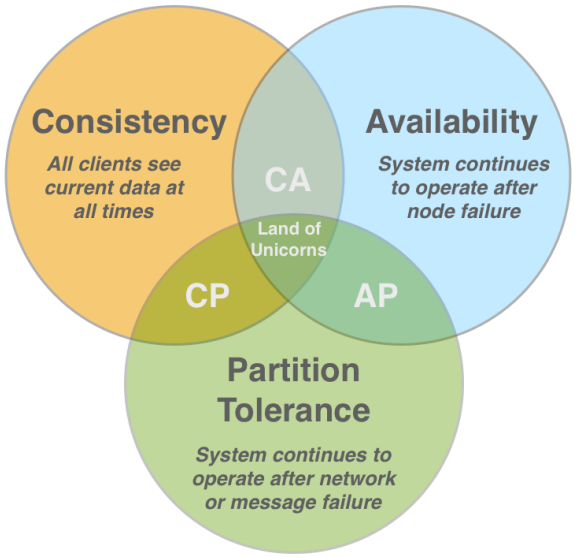
Just a quick note on hybrid storage pools and hierarchical storage management (HSM). Many of the above technologies can and have been combined to allow for interesting multi-tier configurations. In the case of HSM, software helps move unused data from the high tiers (Tier 1) to the low tiers (Tier 3) as it becomes stale. In the case of hybrid storage, SSDs are frequently used to speed up pools of spinning disk drives, optimizing and balancing between cost and latency in a sane manner. It won't work for all workloads, but it works for many or even most.



3) The Unremitting Truth of the CAP Theorem

It's beyond the scope of this document to get deep into the details of the CAP theorem and you will find people arguing about its applicability left and right all over the net. In order to make progress I have to assume you understand what it is. If not, Google will help you.

Mostly what we care about CAP theorem and its applicability to storage can be found in this fabulous blog posting titled Eventually Consistent by the inimitable Werner Vogels, CTO of Amazon [@werner]. Werner focuses on explaining the whys and wherefores of eventual consistency, but in the process he does a particularly good job of explaining the value of consistency generally for storage systems by way of explaining CAP.



I know this is a long article already and Werner's is also pretty deep, but please check out and at least skim his article before proceeding. I would like to quote a few key passages in case you choose not to read his very detailed explanation. Some editing performed for brevity. All emphasis is mine.

On CAP Theorem itself:

> *"A system that is not tolerant to network partitions can achieve data consistency and availability, and often does so by using transaction protocols. To make this work, client and storage systems are part of the same environment and they fail as a whole ... network partitions are a given and as such* ***consistency and availability cannot be achieved at the same time****. This means that one has two choices on what to drop; relaxing consistency will allow the system to remain highly available under the partitionable conditions and prioritizing consistency means that under certain conditions the system will not be available."*

On networking partitions in a distributed system:

> *"****Partitions happen when some nodes in the system cannot reach other nodes, but all can be reached by clients.*** *If you use a classical majority quorum approach, then the partition that has W nodes of the replica set can continue to take updates while the other partition becomes unavailable. The same for the read set.* ***Given that these two sets overlap, by definition the minority set becomes unavailable****. Partitions don't happen that frequently, but they do occur, between data centers as well inside data centers."*

On eventual consistency in traditional (non-cloud, non-distributed) models:

> *"Eventual consistency is not some esoteric property of extreme distributed systems. Many modern RDBMS systems that provide primary-backup reliability implement their replication techniques in both synchronous and asynchronous modes. In synchronous mode the replica update is part of the*

> *transaction, **in asynchronous mode the updates arrive at the backup in a delayed manner**, often through log shipping. **In the last mode** if the primary fails before the logs are shipped, **reading from the promoted backup will produce old, inconsistent values**. Also to support better scalable read performance RDBMS systems have started to provide reading from the backup, **which is a classical case of providing eventual consistency guarantees**, where the inconsistency windows depends on the periodicity of the log shipping."*

Let me boil it down to three key truths:

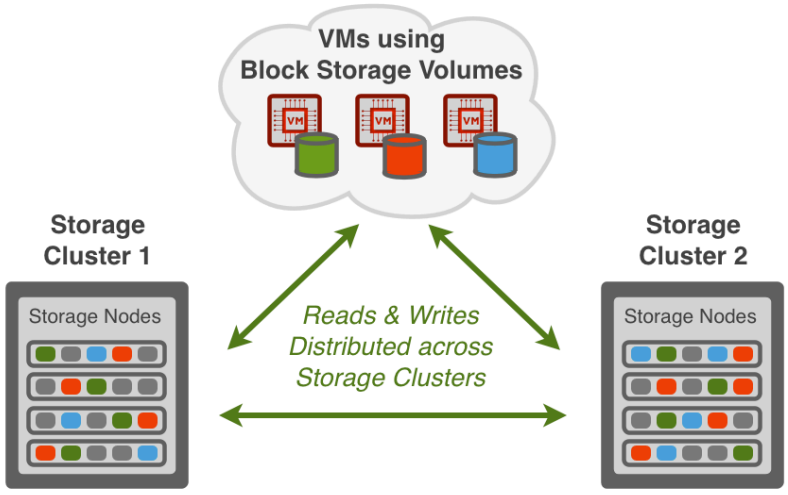
1. Consistency + availability is not achievable in a distributed system
2. Network partitions happen
3. Eventual consistency is not new

These are immutable laws or truths of computer science, distributed computing and cloud storage. Certainly open to some interpretation and we are still learning, but no one has proven that you can have all of CAP. You only get two.

CAP In Pragmatic Terms

The discussion of CAP theorem may sound like an academic exercise, but it is not. I want to show you a pragmatic and real world scenario of why this matters. Let's say you are running one of these new-fangled and frankly, somewhat experimental, distributed storage systems like Ceph, GlusterFS, or Sheepdog.

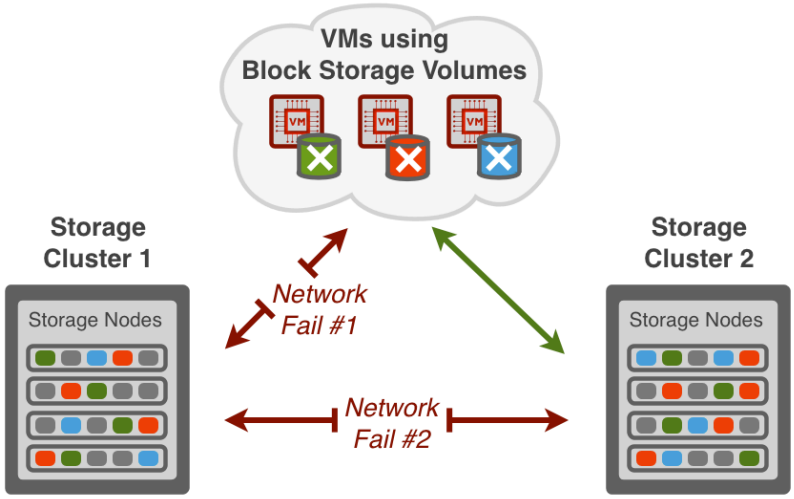
In your deployment, you are providing block storage interfaces to a large population of virtual machines. Let's say that you have two racks and are running one large distributed storage system across both racks. A separate cluster of VMs is running on hypervisors that access the cluster across the network. Here's a diagram:



Now, these various storage solutions, while being object storage-based under the hood, violate the typical use case for object storage and attempt to provide strong consistency instead of eventual consistency. They do this, because they are providing Tier 1 block storage capabilities and the consumer of these

systems is assuming that their block storage devices are always available, low latency, and strongly consistent.

All VMs are running on top of this distributed storage and their reads and writes are being dutifully spread across both racks. What happens if the VMs can't reach cluster 1? Or if both clusters can't reach other? This happens all the time and is more common than you might think. Perhaps a route is lost or an operator makes a mistake.



The answer is that strongly consistent systems will stop allowing writes and may even, under some circumstances, stop allowing reads. They do this to guarantee a block device is as consistent as possible, but the effect on your VMs and cloud is catastrophic.

Your entire cloud melts down. Poof. Gone. This kind of massive failure domain is actually **worse** than the old SAN/NAS model. At least in that model, your SAN and NAS were only supporting a subset of your cloud.

Imagine a different example like a software bug that causes data corruption, which happens to even the best of us (which I linked to previously re: the Google need for tape). This is an example of an overly large failure domain, something we want to avoid in a true cloud system.

Massive Failure Domains

Large failure domains are truly bad, with a capital B. Distributed storage systems solve the scale-out problem, but they don't solve the failure domain problem. Instead, they make the failure domain much larger. I talked about this in depth in my OpenStorage Summit Presentation in 2012. See slides 17+ in particular.

Implied Usage Contracts

It's beyond the scope of this post to go into this in too much detail, but a whole area here that I have ignored is what the "implied contracts" are with the end user of your OpenStack cloud. Implied contracts exist all throughout a system and come from assumptions that end-users make about how the system works. Frequently these assumptions are set over time by best practices.

For example, on Amazon Web Services (AWS), all of your snapshots for their Elastic Block Storage (EBS) service are stored in their Simple Storage Service (S3). What that means is that you have an implicit guarantee that in the case of an EBS node or volume failure (poof! your data is gone), you can restore from a snapshot, using the snapshot as a de facto backup.

Great, but if you converge your storage tiers, then this implicit contract with the end user is violated. Data damage to your single distributed storage system could cause loss of ALL data with no backup. This is why enterprises think in terms of data retention and backup policies and use multiple, independent, storage technologies and systems for each tier of storage.

Why Distributed Storage, As Marketed, Violates the Three Key Drivers for Tiered Storage

As a reminder, the three key drivers for tiered storage are:

1. Cost
2. Technology
3. CAP

Distributed storage solutions, as marketed, are selling the questionable notion that these drivers do not exist. Yet they do and frankly there isn't any way to ignore them.

The economics should be startlingly obvious. The spectrum of cost for storage means that you are always trading off cost versus speed. If you build one storage cloud that is all SSDs it will be prohibitively expensive for object storage and backup/archival use cases. If you build a storage cloud using ultra low speed green drives it will be too slow for mission critical, high IOPS database use cases.

Regardless of cost, the various storage technologies are simply meant for different purposes. While spinning disk drives can scale up a bit and down a bit, they can't ever be as fast as SSDs or as cost effective as tape (hello AWS Glacier!). Some are best for low latency, some for high capacity, and others for long term durability or low power use. You have to use the right tool for the job.

CAP, while seemingly an academic exercise, is, in truth an immutable reality of cloud computing that cannot be ignored. Distributed storage solutions have no basis for claiming they will make CAP disappear. More on this in a moment.

Here's the nut: you cannot have your cake and eat it too. Period.

How Distributed Storage Should Be Deployed

Distributed storage solutions are worthy of being deployed. While some of the technology is still, frankly, experimental in my book, I think there are very clear and compelling use cases for it. As long as you don't try to converge your storage tiers, which is a pipe dream, you can deploy them successfully.

Instead, as you are evaluating your options for your OpenStack cloud or similar you should be thinking in terms of the three tiers. Pick one solution for elastic block storage that is strongly consistent and preferably network partition tolerant. Pick another solution for shared files via a distributed file system approach for your tenant's VMs. And pick yet another solution for your long term archival and storage. What might that look like?

Well, a conservative approach might be:

	Cloud	Approach
Tier 1 Performance	Elastic Block Storage	OpenStack Block Storage (Cinder) + its scheduler capabilities + a tier of traditional NAS/SAN solutions with a hybrid storage pool for EBS
Tier 2 Balance	Distributed File Systems	GlusterFS for a shared file system amongst your virtual servers for shared assets like web images, user directories and the like
Tier 3 Capacity	Object Storage, some Tape	OpenStack Object Storage (Swift) for object storage for archival and backup

Of course, you can reconfigure this a number of different ways. Ceph or Sheepdog for Tier 1, accepting potential network partition issues, parallel NFS (pNFS) on multiple VMs for Tier 2, and tape for Tier 3. Do it whichever way you think makes sense. Just don't be under the illusion that the tiers are going away.

Other Topics

There is just a ton to talk about in this area and I wish I had more time, but off the top of my head this is some of what is interesting to get more in-depth on:

- Hybrid storage pools and getting the economics of spinning disk with the speed of SSDs
- Latency in distributed storage systems and why spinning disks are limited, particularly during replication storms
- Failure modes for distributed storage systems such as partially failing disks with massive latency, data corruption bugs (bit rot), and operator errors
- Forthcoming storage technologies that obviate or change the game yet again
- And more ...

Wrapping Up

I really hope this helps buyers and sellers think about cloud storage, particularly in OpenStack clouds, in a better way. We all want success for the OpenStack and private cloud ecosystems generally and snake oil helps no one.

Remember, tiered storage isn't going away due to the economics, technology and computer science (CAP Theorem) behind it. You can successfully deploy and manage even experimental distributed storage solutions by paying attention to these constraints.

Good luck.

Contact Us

To learn more about how Cloudscaling's products and services can help you deploy and manage private elastic cloud capabilities, visit us at www.cloudscaling.com.

Sales
+1-415-508-3270 sales@cloudscaling.com

Media Relations
Robert Cathey +1-865-386-6118
robert@cloudscaling.com

ABOUT CLOUDSCALING

Cloudscaling is the leader in elastic cloud infrastructure. The company's core product, Open Cloud System (OCS), is the world's most advanced OpenStack cloud infrastructure system. OCS is designed to meet the requirements of next-generation dynamic applications, delivering the agility, performance and economic benefits of leading cloud providers, but deployable in the customer's data center and under the IT team's control. Cloudscaling is backed by Trinity Ventures and headquartered in San Francisco. For more information, please visit www.cloudscaling.com.

cloudscaling

Cloudscaling
45 Belden Place
San Francisco, CA, 94104
Main: +1-877-636-8589
International: +1-415-508-3270

openstack CLOUD SOFTWARE

Cloudscaling is the trusted source for information on OpenStack and together with the community is making OpenStack more production-grade. For more information, please visit www.openstack.org.