

# [Tech Field Day VFD3 - Pure Storage and the all-flash revolution](#)

As we close out our first day of presentations here at [Virtualization Field Day 3](#), we are at the office of Pure Storage in Mountain View. Pure Storage is a really neat company for a number of reasons. Their all-flash array is a product that is not an evolution of an existing product which was simply being augmented with a flash tier to accelerate data storage and retrieval. In fact, they launched among our community using great events like Tech Field Day to an avid audience of storage enthusiasts.

## ✘ **What is the strategy to all-flash?**

How about a simple strategy: Let's deliver an all-flash storage array for a lower price than traditional spinning disk? Wow! That's quite an aggressive tagline, but what Pure Storage does is to work at delivering a performance and consolidation platform that lowers the per-VM cost to bring its customers 0.3-0.7ms data access with 5-10x consolidation through inline de-duplication and compression.

So how do they do this? Very good question, and it is comprised of a lot of features at the hardware and software layer. I couldn't do it justice in a quick post, so please forgive me that I won't dive into the deep technical goodies here, but I wanted to look at some of the other aspects that make Pure Storage interesting in what they do.

## **The Forever Flash promise**

✘ This is really cool! When you bring a Pure Storage product into your data center you will size it as needed and the typical experience is to acquire storage with a long lease cycle because of the high cost to acquire enterprise scale storage.

The challenge is that the same large scale storage really needs care and feeding, and with the aggressive moves happening in storage engineering, it seems counter-productive to sign on for a long lease on large storage.

With the Forever Flash program you can actually upgrade your controllers every 3 years to align with the updates that have been engineered by Pure Storage, and to top it off the rest of your storage in the chassis then has its support cycle re-aligned with the upgraded hardware. Effectively it is as if you just put the product on the floor and started your support contract again.

Incremental upgrades also give the same re-up for your support, so you can continue to grow your Pure Storage environment and stay up to date on features, hardware, software and support, all at the same time.

For info on the Forever Flash program head on over here: [Forever Flash with Pure Storage](#)

## **Is it really lower cost?**

I have to be honest that it sometimes seems like a little sleight of hand when I see all-flash options that can be deployed for a lower per-VM or per-GB than other traditional storage arrays.



Image courtesy of Pure Storage - <http://www.purestorage.com/resources/roi.html>

Looking at the diagram we can see how the overall cost is accounted for with storage and how Pure Storage is able to put storage on the floor at a customer for a per-GB cost that comes in much lower than expected because of compression, power/cooling reduction, and minimal management overhead for the administration.

That being said, there is a real cost to bringing all-flash solutions into the data center. In my opinion, this may not be appropriate for many SMB organizations with moderate workloads. There is a definite target market, and a lot of factors come into play to define where the sweet spot is for moving to an all-flash solution.

The basic deployment is a 2 controller implementation with a half-populated shelf, so there is an entry point that is attainable for many organizations.

## **The bits matter**

Literally. The way that one storage stands apart from another is the way the bits are read, written, replicated, de-duplicated and generally managed. Hardware becomes a genuine question when dealing with flash storage because of the alternatives that are available (MLC, eMLC, SLC) and the Pure Storage goal is to find the balance with performance and reliability with managing the price point to keep their solution as a cost-effective offering.

Since their first array hit production, they have replaced 5 drives altogether and one was for a firmware issue. That's a pretty good track record. We discussed a lot of deep-dive details on hardware, software, and workloads which was eye opening and encouraging.

Pure Storage also built their solution with a 512 byte block default rather than the traditional 4K block size. This has some really slick advantages in how the performance can be increased at many points. In fact, it means that block alignment is no longer an issue because the block sizing eliminates the performance issues that come into play with 4K blocks and certain application/VM features.

I'll be sure to post the videos so that you can see some of the targeted talks on content such as the thick versus thin, eager zero versus lazy zero on SSD. Very interesting info on tactically handling performance in conjunction with your hypervisor features.

## **My thoughts**

I really like the idea of what Pure Storage is doing as a company, and as product creators on both the software and hardware side. It would be great to be able to have an all-flash solution in my data center, and that time may come as my workloads are more able to take advantage of the predictable performance and speed of all-flash arrays.

## **I see OpenStack in there ☐**

There is a current Cinder driver for the Folsom release with an upcoming update to support the Havana build. With an increased customer field growing in the OpenStack space, the team has added that they will be increasing some focus on the platform to align with requirements from the

consumers.

## Is Pure Storage right for you?

This one will be something that every organization has to evaluate, but I can say that the people and the product are great here at Pure Storage and it is absolutely worth putting on the evaluation plan to see how it may fit into your data center.

You can be the judge with your particular situation, but make sure to reach out to the team at Pure Storage on Twitter ([@PureStorage](https://twitter.com/PureStorage)) and at their website <http://www.purestorage.com> for more details.

**DISCLOSURE: Travel and expenses for Tech Field Day - Virtualization Field Day 3 were provided by the Tech Field Day organization. No compensation was received for attending the event. All content provided in my posts is of my own opinion based on independent research and information gathered during the sessions.**

---

## [Getting to know Infinio - Putting RAM to work like never before](#)

✖ If you haven't already seen Infinio Systems, it is time to stop and take a look. This is a significant new vendor, with a significant product. Infinio Accelerator is doing something different from we have seen up to now.

Oh...and did I mention that the product is officially in General Availability as of right now!

### Cache is King

The advantage to host caching is that the read caching is as close to the workload as possible and you are able to maximize the benefit. With so many hardware-based flash cache products entering the market, and software products that can leverage hardware flash and SSD hardware for host-side read caching, Infinio could turn the market on its side with what they are doing.

I participated in a pre-launch demo of the Infinio Accelerator which gave me an early view of the deployment and management of the product. I was absolutely impressed. This has the potential for a significant customer market because of the ability to leverage host RAM as a distributed flash cache without the need for retrofitting your storage array with potentially expensive flash drives, or adding PCI flash hardware at the hosts.

### How does it all work?

Infinio Accelerator is a software-only NAS offload engine. It deploys using an OVA on each host and utilizes local RAM (8GB by default) for its cache storage and the virtual appliance is held on local storage with the host.



Once deployed, you can pick which of your hosts you want to accelerate, and which NFS volume and then you are off to the races. The wizard ticks away nicely in the background as it imports the OVA into your environment and loads the IP configuration that you've specified during the install.



The bonus of Infinio is that it intercepts traffic, but does not put itself in between the host and the NFS workload. In other words, if something goes wrong with your Infinio Accelerator environment, the only thing that will happen is that your vSphere environment loses the extra advantage, but no production stoppage occurs. That's comforting if you ask me!

Any workload applies too. This isn't isolated to VM guests running a specific OS or anything. Any VM, vApp or VDI workload on NFS will see the benefit. If you decide you need to try it and then back out, just de-provision the accelerator using the install wizard and it cleanly backs out the Infinio tools. No reboots. No leftovers.

It's as easy as 1-2-3!



Image source: <http://www.infinio.com/about-our-product/what-is-it>

And yes, I speak from experience. I've installed and uninstalled multiple times to be sure how the process works. No impact to my active workloads other than becoming more awesome!

## **Acceleration for the rest of us**

I think this is a great fit with organizations at every level from SMB to big Enterprise and it is great foray into accelerating workloads without having to take the steps to architect hardware SSD solutions into your datacenter. A little more RAM is a short step away for many and you will see the benefit immediately to judge if you want to put Infinio to work as a production solution.

There are a few requirements which you have to meet which are highlighted during the deployment wizard:



Standard vSwitches? Not a problem. In fact, it's a requirement. With the 1.0 release of Infinio Accelerator, Distributed vSwitches are not supported so that will have to be a consideration for your design.

The win in this case is that you don't have to be running Enterprise+ with vDS in order to use the Infinio Accelerator. That's a big differentiator for many customers. Many advanced storage I/O tools require vDS, which requires Enterprise+ licensing on the vSphere host. Not a concern in our case now.

## **Just like the Queen song: I want it now!**

There is a reason that we have been chatting up this product up to now and that's because it has

been brewing for a while in Beta and now with the GA launch you will see much more great info coming out. Head on over to <http://www.infinio.com/> and you will be able to sign up for the download.


Once you're signed up, you get a **30 day trial for free** and you can turn that into a production product without disruption.

Join the live technical Q&A at 3:00 PM Eastern time today (November 5th just in case you are reading this after the launch) by registering here: <https://www.brighttalk.com/webcast/10295/90639>

Don't let your lack of SSD be a barrier to putting performance into your vSphere and NFS environment.

---

## [Nice Flash messages in Rails 2 and Rails 3](#)

 I've been doing some Rails work recently and one of the things that I've found to be a nice aesthetic add-on is putting some nicer flash messages in.

Over at [dzone.com](http://snippets.dzone.com/posts/show/3145) there is an example which I've worked off of - <http://snippets.dzone.com/posts/show/3145>

The example is as follows. Add this section to your **application\_helper.rb**

```
def show_flash
  [:notice, :warning, :message].collect do |key|
    content_tag(:div, flash[key], :class => "flash flash_#{key}") unless flash[key].blank?
  end.join
end
```

In your CSS file you now add the settings to style your messages. Note that my examples assume some images are present so add images as you wish for within the message:

```
.flash_notice {
  BORDER-RIGHT: #090 4px solid; PADDING-RIGHT: 10px; BACKGROUND-POSITION: 5px 50%;
  BORDER-TOP: #090 4px solid; PADDING-LEFT: 10px; BACKGROUND-IMAGE:
  url(/images/icon_success_lrg.gif); PADDING-BOTTOM: 10px; MARGIN: 2px; BORDER-LEFT:
  #090 4px solid; TEXT-INDENT: 40px; PADDING-TOP: 10px; BORDER-BOTTOM: #090 4px solid;
  BACKGROUND-REPEAT: no-repeat
}

.flash_warning {
  BORDER-RIGHT: #c60 4px solid; PADDING-RIGHT: 10px; BACKGROUND-POSITION: 5px 50%;
  BORDER-TOP: #c60 4px solid; PADDING-LEFT: 10px; BACKGROUND-IMAGE:
  url(/images/icon_warning_lrg.gif); PADDING-BOTTOM: 10px; MARGIN: 2px; BORDER-LEFT:
  #c60 4px solid; TEXT-INDENT: 40px; PADDING-TOP: 10px; BORDER-BOTTOM: #c60 4px solid;
```

```
BACKGROUND-REPEAT: no-repeat
```

```
}
```

```
.flash_error {
```

```
BORDER-RIGHT: #f00 4px solid; PADDING-RIGHT: 10px; BACKGROUND-POSITION: 5px 50%;  
BORDER-TOP: #f00 4px solid; PADDING-LEFT: 10px; BACKGROUND-IMAGE:  
url(/images/icon_error_lrg.gif); PADDING-BOTTOM: 10px; MARGIN: 2px; BORDER-LEFT: #f00  
4px solid; TEXT-INDENT: 40px; PADDING-TOP: 10px; BORDER-BOTTOM: #f00 4px solid;  
BACKGROUND-REPEAT: no-repeat
```

```
}
```

In your view, which in my case is the views/layouts/application.html.erb (same for Rails 2 or 3) I have added this where I want my flash notices:

```
<%= show_flash %>
```

If you are running Rails 3 there are two additional steps you need to take. First you need to change the code on your view to display as follows:

```
<%= raw show_flash %>
```

Note that under Rails 3 the flash will render the html code as text rather than as the raw html. By adding the raw statement you fix this issue and it will render correctly.

The second change you want to make is for your scaffolds. By default the show.html.erb file already contains a message section. If you leave this in place you will end up with 2 flash messages on each render of the show action. To fix this, go into the show.html.erb under your app/view/scaffoldname/ folder and remove this line:

```
<p id="notice"><%= notice %></p>
```

Here are the 3 image files that I've used as referenced in the CSS



That's my hopefully useful Rails tip of the day!