

ioFABRIC Vicinity Review - by Trevor Pott

What should datacenter storage look and behave like? This is a question that has preoccupied a significant number of the IT industry's best minds for the past decade. The storage wars have transformed the IT landscape and offered seemingly unlimited options. Despite this, the question of "should" still remains. Having reviewed ioFABRIC Vicinity for two months, I am prepared to argue that datacenter storage should look like Vicinity.

What is ioFABRIC Vicinity?

ioFABRIC is a young startup, but one with a lot of experienced folks who've managed to build a very feature rich product in an impressively short amount of time. The product needs some TLC to make it easier to use, but its ambitious scope does deliver.

Vicinity consists of two node types: a control node and one or more data nodes. Vicinity can currently be installed in two ways: as a bare metal install onto a Linux host or as a virtual appliance for VMware, Hyper-V or KVM. Data nodes will scan for raw storage and networking resources presented to them, claim those resources and make them available to the control node. Storage can be RAM, traditional magnetic media, SSDs or LUNs presented from existing storage solutions to a data node for consumption.

The real magic is in the QoS settings. You can pick from amongst the pre-existing QoS profiles, or define your own. QoS profiles define things like the number of replicas and the maximum storage latency that volumes using that profile should get. Administrators don't have to worry about where the storage lives, or assigning individual devices. Vicinity's algorithmic voodoo takes care of it all. Administrators can take and manage snapshots of volumes. They can manually add resources to volumes. There is even an API so the DevOps crowd can automate the whole thing.

All of the above works. It does what it says on the tin. It gobbles up storage presented to it, casts a spell upon it and presents it to workloads that wish to consume it. There isn't a lot of room for complaint in the basic functionality.

Vicinity was created by people who understand Vicinity and all of the thought conventions, restrictions, assumptions and so forth that evolved during its creation. During my review I ran across several puzzling limitations, including installation and UI choices that need to be smoothed out or better documented.

One example of such a "rough edge" is that if a data node is set up on a system it only creates a 256MB RAM disk. There is no option in the GUI to change this. Considering the importance of RAM in accelerating storage for various Vicinity QoS targets, that's kind of odd, and more than a little frustrating when I have servers with half a terabyte of RAM to be gobbled up, waiting to be used.

To be fair, an administrator can go to the command line on that data node and manually change the size of the RAM disk, but that's more than a little frustrating. What's worth noting, is that ioFABRIC has responded well to this (and many other) critiques. As of the time of this review, the small RAM drives were a frustration. By the time you read this review, ioFABRIC has promised to implement a means to set RAM disk size in the management UI.

About the Author

I am both a systems administrator and a technology writer. I have been fixing computers, networks and associated bits that go "ping" for a little over twenty years.



As a systems administrator, I have focused primarily on small business and the midmarket. This is the kind of tech work that doesn't allow for specialization. It forces sysadmins to be generalists, and it allows us to see a little bit of everything in use at some point. As a writer, I have spent much of the past five years investigating, testing and writing about every kind of storage you can imagine.



What is ioFABRIC Vicinity? (Continued)

Nevertheless, ioFABRIC understands my criticism in this regard and have committed to a solution. To be perfectly honest, this is a little jarring. I've worked with tech companies of all sizes – from startups to tech titans – for years. Very few have been so eager to actually engage with me as a reviewer to solve the issues I turn up. It's a very good sign; the companies that listen to customers tend to be the ones that make it in the long run.

Performance

Benchmarking Vicinity is something of a fool's errand. Vicinity is as fast as you make it. If you want it to go faster, add more RAM, SSDs and/or networking. If you need more archival space, add some spinning rust or (eventually) object storage.

There's overhead to ioFABRIC's approach to storage. Basic computer science says there has to be. Qualifying that overhead is the hard part. I know the servers, storage and networking in my lab pretty well and as such I can make some educated guesses. At current, it looks like there is maximum of about a 10% IOPS overhead imposed by Vicinity, when deployed as a hyperconverged solution. All things considered, that isn't all that bad, and is in line with what I see from other solutions: hypervisors impose a penalty on IOPS even when using pass-through disks and there is no getting around that.

Vicinity does seem to go faster with only bare metal data nodes. The full speed of any attached storage becomes available, as there is no hypervisor IOPS overhead. CPU overhead is between 2% and 3% on a typical metal server using SATA flash disks, but can rise when top-of-the-line NVMe flash is used. Again, this is normal; when storage ceases to be the bottleneck (as is the case with NVMe) the rest of the system becomes more fully utilized. Our bare metal Vicinity nodes with Micron 9100 NVMe drives are truly something to behold.

Vicinity connects different storage devices across multiple nodes together into one great big object store. It then makes that storage available via a shim to requesting servers. That shim is never going to be perfect, and the fact that the highest IOPS penalty I've seen, even in complicated mixed deployment models was about 10% is impressive. 10% is about as efficient as VSAs tend to get, and I've seen a heck of a lot worse than 10% overhead from other products.

Conclusion

What ioFABRIC has built in Vicinity is nothing short of astonishing, and they have reached that point where they're ready to start being put into very real use. Vicinity is absolutely ready to handle test and dev use cases. It's ready to take all your aging storage and connect it together into a cold data or archive tier. It's ready to start simulated workload trials and be used for non-persistent production workloads.

I even believe that Vicinity will pass any trials to which it is subjected and prove worthy of being primary storage.

ioFABRIC's Vicinity has great potential to solve all my storage issues. I have a list of features I'd like to see added and use cases I'd like to see covered as long as my arm. When talking to storage companies, I always do. ioFABRIC is one of the few companies I've run across that I think can deliver. In the fullness of time I expect them to be able to deliver everything I could ask them, and quite a few things I never thought to as well.

When dealing with any storage solution, caution is warranted. If the storage fails, everything fails. For today, ioFABRIC inspires optimism. If I have one parting recommendation I would hope readers retain, it is to reject your innate cynicism that Vicinity must be too good to be true. Give Vicinity a try and you'll remember why we all got so excited about software-defined storage in the first place.

