Lab results: Benchmarking a EVO:RAIL system

May 12, 2015 by [vMario156](#) [Leave a Comment](#)

I had the chance to play around in our solutions center at work to with a EVO:RAIL system from Fujitsu the last days.

I thought it would be interesting to benchmark the system to see how much performance you can get out of it and would like to share the results with you in this post.

Hardware & Software Setup:

Fujitsu CX250 S2

Latest Drivers & Firmware Updates

vSphere 5.5U2

4 Nodes each with:

2x Intel E5-2620v2@2,1GHz

192GB RAM

1x 400GB SSD

3x 1,2TB HDDs

2x 10GbE

Test Setup:

-Four Windows Server 2012R2 VMs (one running on each node)

-8 vCPUs & 16GB RAM for each VM

-8 additional VMDKs per VM, distributed across multiple PVSCSI controllers

-IOmeter 1.1.0 installed, controlled via one master server to ensure simultaneous start time

-Two workload profiles with reasonably real-world workload settings (4k / 60% read/ 95% random & 8k / 50% read / 80% random)

-8 outstanding IOs per target

-10 minutes runtime for each test
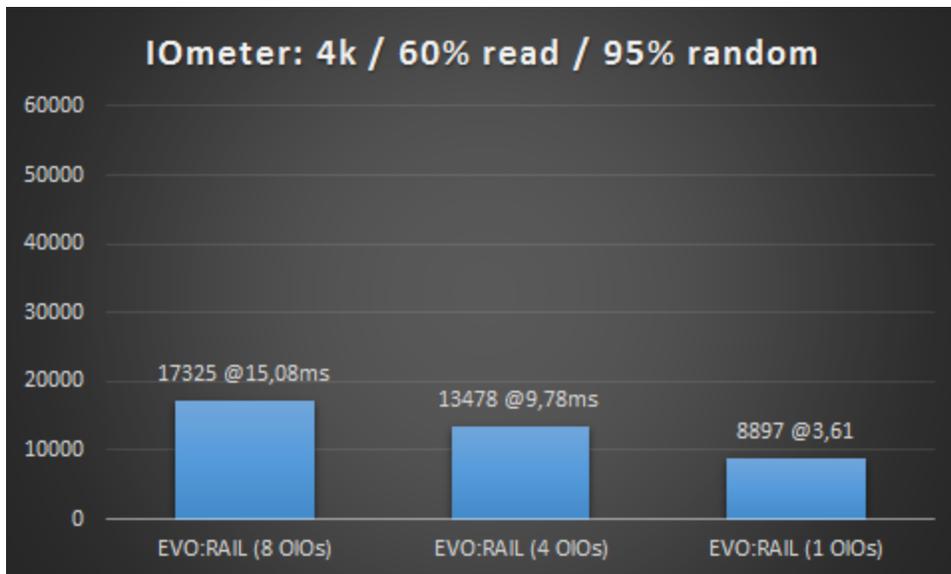
-Multiple runs for each profile

Platform specific settings:

-Stripe witdh: 3

Results:

IOmeter: 4k / 60% read / 95% random

17325 @15,08ms

EVO:RAIL

IOmeter: 8k / 50% read / 80% random

12137 @21,76ms

EVO:RAIL

It´s important to look also at the latency and not only at the amount of IOPS. With the "# of Outstanding I/Os" setting within IOmeter you can heavily influence the amount of IOPS you will get as result. In my opinion it´s important to choose a value, which keeps the latency at an acceptable level.

For the tests I used 8 outstanding I/Os like mentioned above, because I used the exact same settings recently for another platform benchmark.

As the latency for both tests with 15ms & 22ms was a bit too high for me, I also performed the first test with some other outstanding I/Os values:

IOmeter: 4k / 60% read / 95% random

	EVO:RAIL (8 OIOs)	EVO:RAIL (4 OIOs)	EVO:RAIL (1 OIOs)
	17325 @15,08ms	13478 @9,78ms	8897 @3,61

Findings (in no specific order)**:**

-There have been big differences between the results of the same workload in multiple runs. Also the single worker / threads numbers had big discrepancies, so I performed around 10 runs to get a realistic average result.

- In regards to the overhead the EVO:RAIL system used between 5-10% during the benchmarks, which are also the numbers that are official specified by VMware.