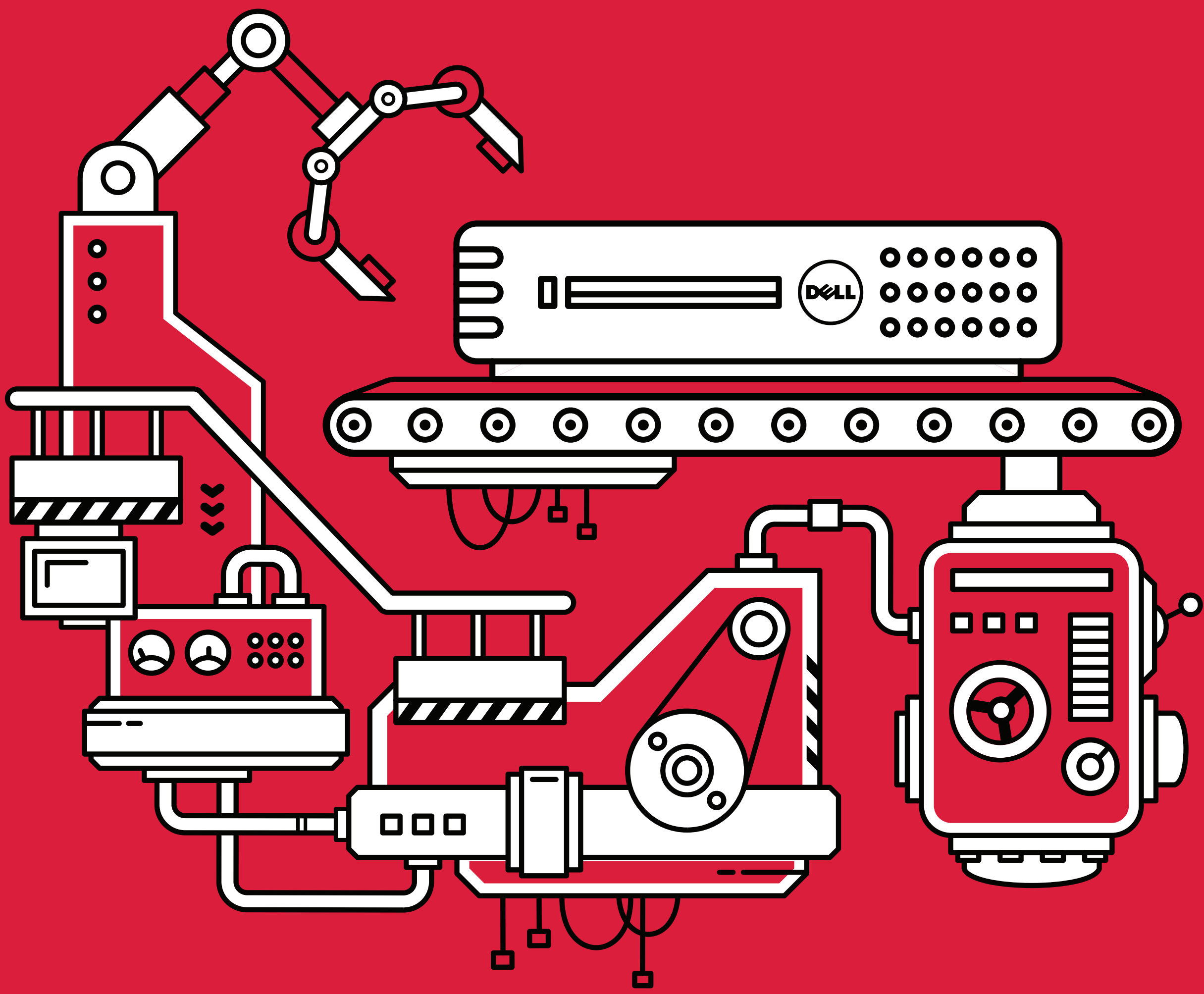
BENCHMARK

December 13, 2016

Jaafar Almusaad
Business Development Manager
Infraflex



Virtual Machines vs Bigstep Metal Cloud in Machine Learning

The Test Environment

The virtualized environment is underpinned by VMware ESXi 6.0. It is made up of three Dell PowerEdge R730xd, which is designed specifically for VMware vSAN. Each server has dual Intel Xeon E5-2650v3, 128GB of DDR4 2133, 3x 10k spindles and 1x SSD. The SSDs are not visible to the virtual machines and only used for vSAN caching.

Each Dell server has 20 Processor Cores or 40 Threads in total. All virtualization features were enabled.

The H2O cluster is made up of three nodes running CentOS 64Bit. Each is configured with 40 virtual processors and 32GB of RAM. The nodes were configured so that each one runs on a different host (Dell server) for best performance. The level of utilization from other virtual machines was relatively low. Neither CPU nor Memory resources were oversubscribed while testing.

The Tests

An example flow called “Airline Delay” was run on both the virtualized and bare-metal platforms. The tests were carried out on a publicly available dataset, which consists of 152 Million observations and 31 attributes. The dataset size is about 14.5GB. A smaller subset of the same dataset consisting of 2000 observations was also used to test the Cache Memory performance.

The way each test works is as follow. First, a copy of the dataset is placed on each node on the same path. H2O loads portion of the dataset into the memory of every node (or the whole dataset if one node is used) and parses the data into its native format. This process is very IO intensive.

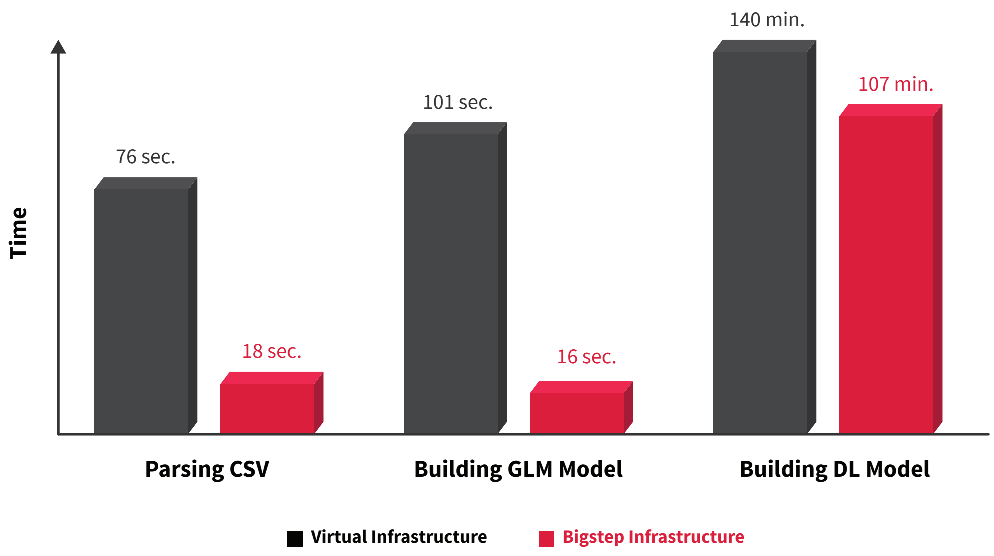
Once data is parsed in-memory, H2O runs two flows: GLM and Deep Learning. GLM is largely single-threaded and runs on one processor thread at a time. The Deep Learning model is vastly parallel. H2O’s Deep Learning implementation is heuristic by default. However, it was configured to perform the same amount of computation on each run. The slight variances are due to scoring the model, which could pick different observations each time, though the number of observations per scoring was fixed to 50k.

Each test was repeated three times and the best performance was recorded.

Test 1

3-Node Cluster (15GB file) fast_mode: true, number of epochs: 10

The virtualized environment is underpinned by VMware ESXi 6.0. It is made up of three Dell PowerEdge R730xd, which is designed specifically for VMware vSAN. Each server has dual Intel Xeon E5-2650v3, 128GB of DDR4 2133 (was 64GB during the first test), 3x 10k spindles and 1x SSD. The SSDs are not visible to the virtual machines and only used for vSAN caching.



The Deep Learning in H2O, by default, is stochastic. This makes the total execution time unsuitable for measuring performance. However, I managed to disable “early stopping”. It forced the model to perform the same amount of computation set by the number of epochs, which is 50 in this case. This is true for both “fast mode” and the normal mode. H2O took considerably longer to build the Deep Learning model, but the models on the bare-metal and virtualized clouds were almost identical.

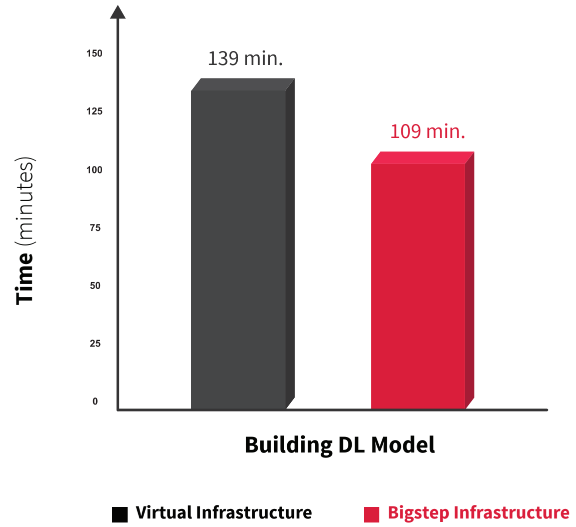
Parsing large files is IO intensive. As such, the SSD performance of the bare metal was significantly better than the 10k spindles on the virtualized servers. I observed a considerable portion of CPU time in the VM cluster wasted while staling for IO.

The new bare metal managed to gain a significant edge over the VM in GLM test
The Deep Learning (with fast mode enabled in H2O) utilizes all processors but performs less processing on each observation, making it ***memory bound***.
The bare metal cluster managed to crunch 260 thousand samples per second (sps) against 198k for the VM cluster.

Timewise, it was 1hour and 49 minutes versus 2 hours and 20 minutes and a performance increase of about 25%.

Test 2

3-Node Cluster (15GB file) fast_mode: false, number of epochs: 0.1

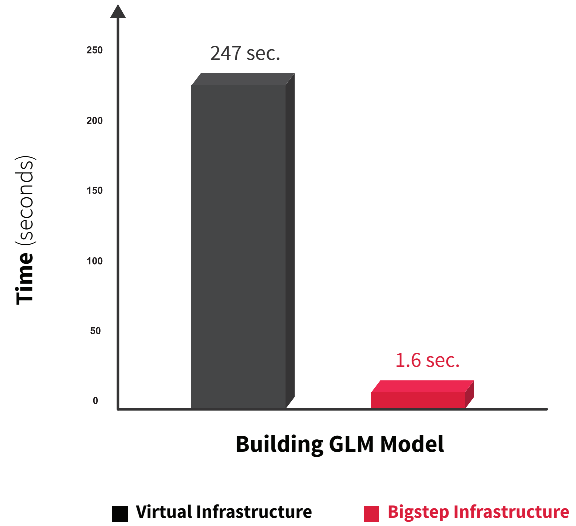


Note: For this test, the number of epochs was reduced to 0.1 only.
In Test 2, the Deep Learning test was repeated but with fast mode disabled. In this case, H2O performs more complex processing on each observation and hence it becomes CPU bound.

Here the bare metal was once again able to outperform the VM by a margin of about 15%. The virtualization features of the recent Xeon processors were able to minimize the hypervisor overhead significantly. Running the same test over older CPUs is likely to produce poorer performance.

Test 3

3-Node Cluster – 4MB file



In order to eliminate the effect of memory bandwidth, a smaller version of the dataset was used, which was about 4MB only. The small size allowed the entire dataset to reside in the cache memory. The GLM test was very interesting, and controversial.

The greater performance on the bare metal is clearly attributed to the cache performance, since the dataset was so small. On the other hand, the extremely slow performance on the VMs can be attributed to a bug or inefficiency in the hypervisor (it took less time to run the same task on a 15GB dataset).

I monitored the utilization during GLM execution on the VM cloud via Water Meter (performance monitoring utility that comes with H2O). I observed a single thread out of the 120 threads being utilized for a short period before the utilization jumps to another processor thread at random. My guess is that GLM was unable to converge due to the cache being rebuilt and then destroyed when execution is shifted to a different node, socket or core. Since this problem was not present in the bare metal cluster, it is safe to assume that the cause was clearly the hypervisor.

Analysis

The objective of this benchmark was to gain a better understanding of the impact of virtualization on machine learning applications. Having matched the hardware very closely, the tests produced comparable performance. Disabling Early Stopping feature on H2O's Deep Learning model made the test repeatable.

Over 1000%
performance gap in GLM tests with small datasets

Unsurprisingly, the bare metal was faster in all tests. The performance gap was within the range of 15 ~ 25% in CPU-bound tests, but over 1000% in others like GLM with small datasets.
IO wise, since H2O loads the entire dataset into memory, the only disk IO needed was to load the datasets (CSV files) into memory. This is a one-time task. H2O applies in-memory Map-Reduce algorithms and is optimized to reduce synchronization to minimum. The 1GB network on the VMs did not appear to have made a measureable impact. However, larger clusters can saturate a Gigabit network.

Conclusion

The tests exposed that virtualization can have a severe impact on performance in some workload patterns. Hypervisors are optimized for efficiency over performance.
The virtualized cluster in this test enjoyed dedicated hardware resources. In practice, however, most commercial virtualized clouds are oversubscribed, resulting in unpredictable and rather degraded performance.

For many tests, utilization tops 100% of the CPU resources. This fact alone defeats the purpose of virtualization in machine learning environments.