Getting the most out of Impala
Best practices for infrastructure optimization

Author: Alex Bordei – Product Manager at Bigstep

Executive summary

We started testing Cloudera Impala in an effort to understand what hardware setup would provide the best performance/price for it. We didn't want to see it perform in extreme cases, but in regular situations that most users would encounter. We aimed to provide a quick practical guide for choosing the infrastructure to run Impala on.
With this in mind, we looked at a medium sized deployment of 4 Full metal Compute Instances, that we think would serve over 80% of user needs and we scaled the hardware from single CPU, low RAM capacity to dual CPU, high RAM capacity.
We didn't start out aiming to prove any particular assumption. The purpose of the project was to explore and understand how Impala works with hardware. However, we did have some underlying assumptions that we considered safe. For instance, we assumed that more RAM would be better than too little and that dual CPU would outperform single CPU instances.
Our findings, however, were quite surprising.

Testing methodology

To test the database's behaviour we used the version of the TPC-DS benchmark adapted for Impala. TPC-DS provides a standard database structure that mimics the sale records of a major retailer. The benchmark kit also provides a synthetic data generator and a standard set of queries.
We performed the tests using CDH 5 and Impala 1.3.0.

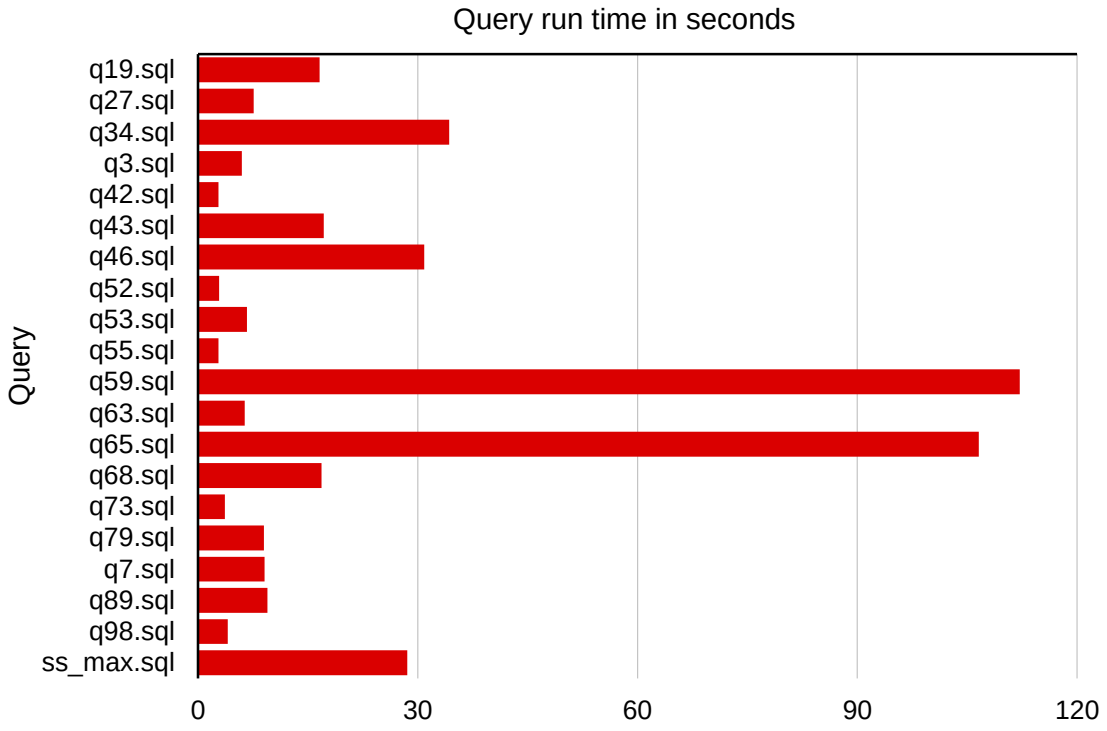
Hardware configurations

Instance type	CPUs	CPU model	Core no.	Core freq.	RAM	Local disks	Network
FMCI 6.32	Single	Intel Xeon E5-2430	6	2.2 GHz	32 GB	8 x 1 TB 7200 RPM	4 x 10 Gbps
FMCI 12.32	Dual	Intel Xeon E5-2430	12	2.2 GHz	32 GB	8 x 1 TB 7200 RPM	4 x 10 Gbps
FMCI 12.64	Dual	Intel Xeon E5-2430	12	2.2 GHz	64 GB	8 x 1 TB 7200 RPM	4 x 10 Gbps
FMCI 12.80	Dual	Intel Xeon E5-2430	12	2.2 GHz	80 GB	8 x 1 TB 7200 RPM	4 x 10 Gbps
FMCI 8.80	Single	Intel Xeon E5-2690	8	2.9 GHz	80 GB	8 x 1 TB 7200 RPM	4 x 10 Gbps
FMCI 16.80	Dual	Intel Xeon E5-2690	16	2.9 GHz	80 GB	8 x 1 TB 7200 RPM	4 x 10 Gbps
FMCI 16.128	Dual	Intel Xeon E5-2690	16	2.9 GHz	128 GB	8 x 1 TB 7200 RPM	4 x 10 Gbps
FMCI 16.180	Dual	Intel Xeon	16	2.9 GHz	180 GB	8 x 1 TB	4 x 10 Gbps

We used 1 NameNode and 3 DataNodes, each with 8 locally attached 1 TB disk drives.
We ran Ubuntu 12 TLS from our all-SSD distributed Full Metal Solid Storage system. All the instances are connected with 4 x 10 Gbps links.

A standard run

The benchmark provides a set of 20 queries that, when executed, take a certain amount of time.

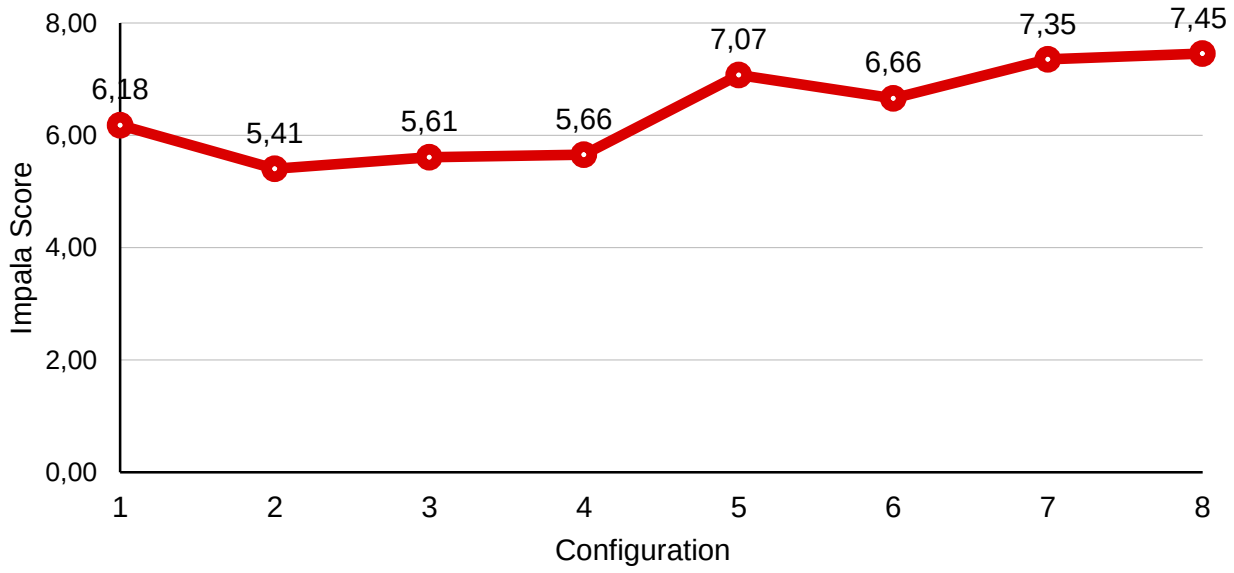


Results

We ran the set of queries on the same data set ten times on each hardware configuration, with SF=12000. Then we normalised the results, summed them up and the result was then inverted and multiplied by 100.

$$Score = \frac{1}{\sum_{i}^{20} NormAvg_i} * 100$$

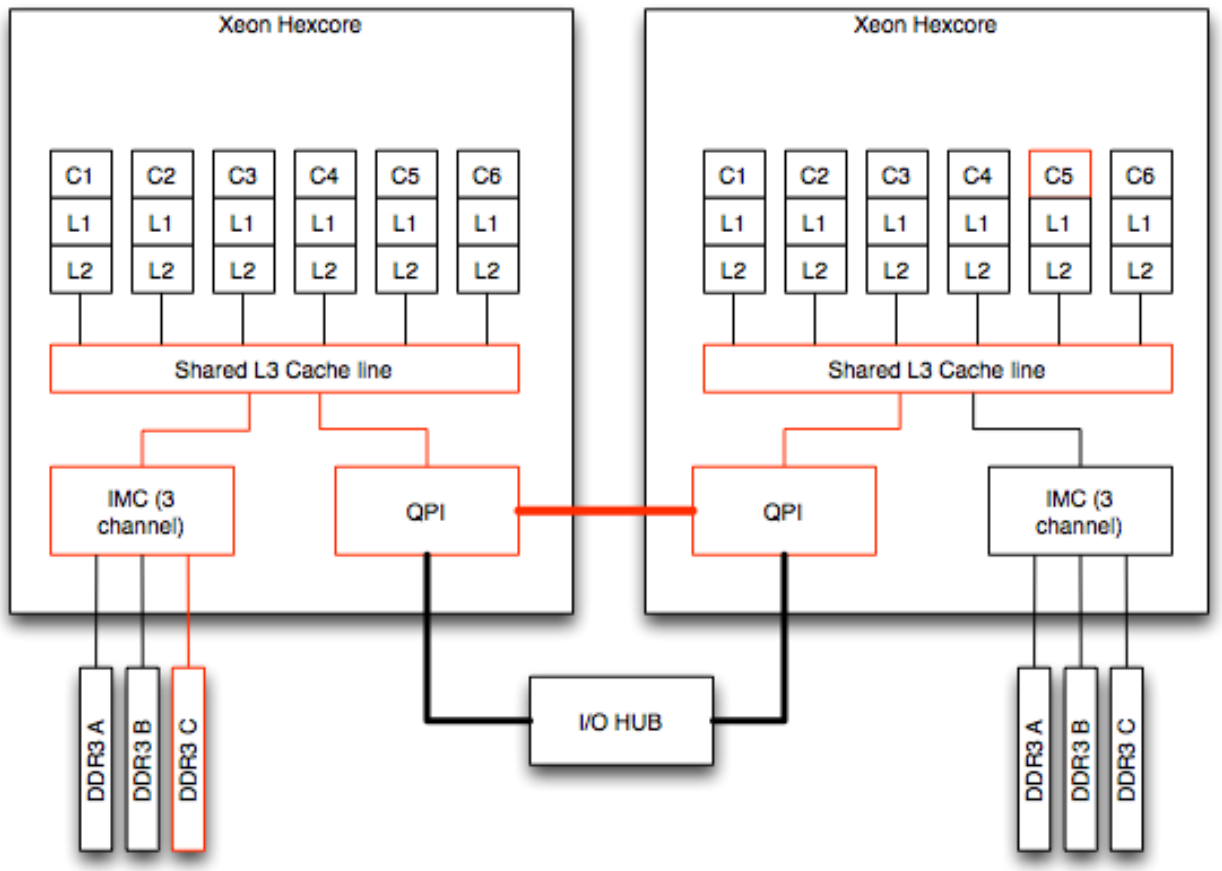
No.	Cluster Size	CPUs	RAM	Impala Score	Price/H	**Performance /Price Score**
1	1 + 3 FMCI 6.32	1 x hex-core 2.2 GHz	32 GB	6.18	£0.44	**14.040**
2	1 + 3 FMCI 12.32	2 x hex-core 2.2 GHz	32 GB	5.41	£0.60	**9.008**
3	1 + 3 FMCI 12.64	2 x hex-core 2.2 GHz	64 GB	5.61	£0.87	**6.450**
4	1 + 3 FMCI 12.80	2 x hex-core 2.2 GHz	80 GB	5.66	£0.90	**6.285**
5	1 + 3 FMCI 8.80	1 x octa-core 2.9 GHz	80 GB	7.07	£1.50	**4.717**
6	1 + 3 FMCI 16.80	2 x octa-core 2.9 GHz	80 GB	6.66	£1.80	**3.699**
7	1 + 3 FMCI 16.128	2 x octa-core 2.9 GHz	128 GB	7.35	£1.94	**3.790**
8	1 + 3 FMCI 16.180	2 x octa-core 2.9 GHz	180 GB	7.45	£2.14	**3.483**



Single vs. Dual CPU instances

The first thing that surprised us once we ran the tests was that configurations FMCI 6.32 and FMCI 8.80, that only had a single CPU, registered a higher score than the corresponding dual CPU configurations (FMCI 12.32 and FMCI 16.80). We weren't expecting the overall score to grow dramatically once we added the second CPU, but we did not expect it to drop either. After closely considering possible causes and eliminating all suspicion of erroneus testing, we concluded that the loss in performance is due to the fact that Impala is very dependent on memory access speeds and adding the second CPU slows these down.

This is not specific to Impala – any database that doesn't use procesor affinity to store data in memory will be subject to this issue as a consequence of the NUMA (Non-Uniform Memory Access) architecture limitations.
Within a NUMA architecture, memory access can be categorised as RMA (Remote Memory Access) or LMA (Local Memory Access). RMA occurs in dual CPU scenarios - when one of the CPUs accesses memory stored in a DIMM that is physically associated with the other CPU. In Intel architectures, this communication process has three steps: requests first go through the controller of the first CPU, then pass the Quick Path Interconnect (QPI) link between the controllers, then reach the controller of the second CPU and from there are sent to the destination DIMM. Not only is the route longer, QPI links are also slower than direct memory



access channels from the CPU to the DIMMs physically associated with it, which further increases latency. And all the while, the CPU core issuing the requests simply waits for the data.

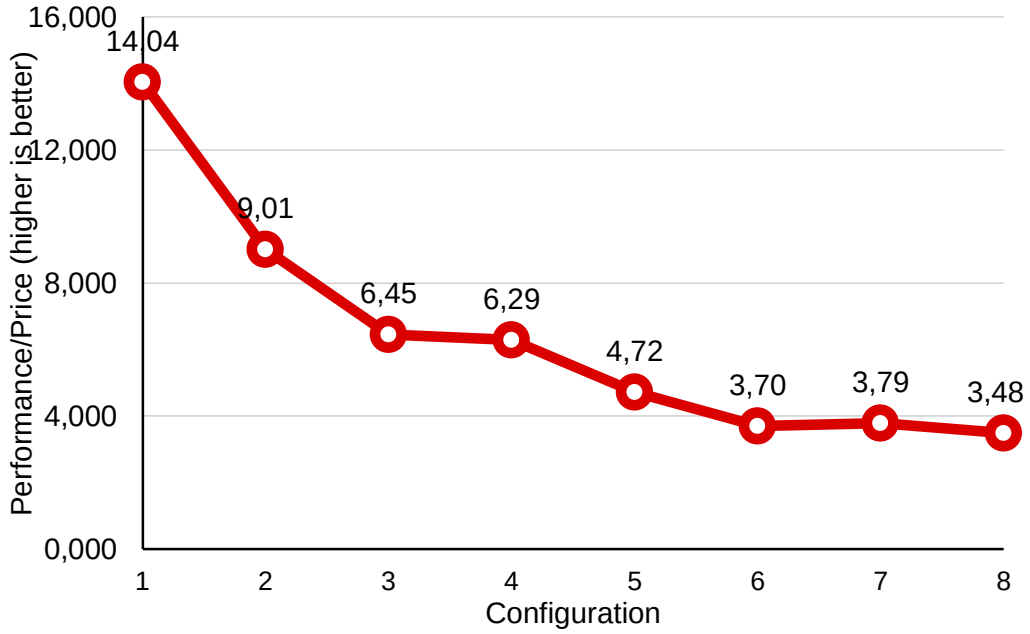
To confirm this, we have performed a SysBench test that measures the time it takes to read and write memory on a single vs. dual CPU configuration. The test further proved that memory access times are much shorter on single CPU instances, even if memory frequency is higher on the dual CPU instances.

Action	FMCI 4.8	FMCI 12.96
MEMR	52	185
MEMW	52	193

FMCI 4.8 is a single CPU instance – 1 quad-core Intel Xeon E3-1230v2 at 3.3 GHz with 8 GB RAM running at 1600 MHz.
FMCI 12.96 is the dual CPU instance – 2 hex-core E5-2430 at 2.2 GHz with 96 GB of RAM running at 1866 MHz.

Performance/Price Score

Using our standard cost structure, we added the price per hour for each instance and paired it with the Impala performance score – in order to get a Performance/Price Score and create a Price/Performance Index.



It's quite easy to notice, looking at the graph, that the highest Price/Performance Score is actually achieved by the instances with the lowest specs. It seems that the higher the specs of the instance, the lower the Price/Performance Score. This is partly to be expected because costs for specs tend to rise exponentially.

Conclusions

- Impala is sensitive to memory access time and the shortest memory access time is provided by single CPU instances. Adding a second CPU increases cost and decreases performance so it should be avoided, whenever possible.
- The higher the hardware specs of the instance, the lower the Price/Performance Score.
- The findings in this paper are not only applicable to Impala, but would very likely be reflected in the use of other memory bound databases. We always recommend benchmarking your database against a number of different scenarios, before making significant investment in infrastructure. With pay-per-hour billing available from most providers, benchmarking will have little cost but great benefits.