Evaluating the impact of SSDs and InfiniBand in Hadoop cluster performance and costs

Nicolas Poggi, David Carrera Barcelona Supercomputing Center (BSC)

This report evaluates the impact in terms of performance and costs of introducing Solid State Drives (SSDs) and InfiniBand networking technologies, over a typical commodity Hadoop cluster setup for Big Data processing. For the evaluations, over 1,000 benchmark runs are executed to the same Hadoop cluster, but varying the software (SW) configuration to find the best performance of a particular hardware (HW) setup. Different HW setups are simulated by executing benchmarks with distinct OS and Hadoop configurations employed to use only selected resources.

The benchmarking process and analysis is automated by using the ALOJA project open source tools. Which also makes the results available online to expand details, compare with other setups —including Cloud, and to reproduce or audit benchmarks[1]. This report intends to provide a baseline analysis of such technologies as the project explores new HW for datacenter solutions. Results show that as expected, both technologies can speedup Big Data processing. However, unlike commonly perceived, SSDs and InfiniBand can actually improve the cost-effectiveness of even small clusters. Ultimately, reducing the TCO of Big Data deployments while speeding up analysis.

1 Introduction

During the past years the exponential growth of data, its generation speed, and its expected consumption rate presents one of the most important challenges in IT. Under this scenario, storage becomes a critical component to efficiently store, process, and scale rapidly as data grows. While Big Data has traditionally been a field for economical and large storage solutions such as rotational drives e.g., SATA drives, Solid State Drive technology offers reduced latency and increased throughput over their rotational counterparts. NAND flash technology also has other benefits for improving data centers, such as increased density, higher reliability and power savings. As during the last years, the cost of SSDs have been decreasing while their capacity increases [9, 11, 12]; not only projections show that they are viable storage solution for big data processing, but they already can provide a reduced Total-Cost-of-Ownership (TCO) and speedup applications.

Traditional Hadoop architecture was designed with spinning disk in mind. In this distributed architecture, the thinking is that you can fetch data from another node faster than getting it from local slow spinning hard disk. However, as SSDs get adopted in Hadoop clusters, the faster throughput and lower latencies shift the bottleneck from storage to network. Adding more nodes to increase Hadoop scalability further increases the network traffic between the nodes. So high speed networking technologies, like 10Gig Ethernet and Infiniband become almost compulsory to keep up with flash technology. The results prove this as well.

The intent of this technical report, is to find the exact numbers of both technologies, and to use them

[1] Results and sources available at http://aloja.bsc.es

as a reference base in terms of job execution time —performance— and costs. Also, to evaluate SSDs and how they can function independently or together with traditional disks over commodity, entry level clusters for Hadoop. Along with the SSDs evaluation, we test impact of high-speed, low-latency networking such as InfiniBand, and compare it to its commodity counterpart, GigaBit Ethernet. Where we first evaluate the performance in terms of *speedups*; and second, the cost-effectiveness of each HW and SW combination. The final goal is to produce baseline numbers of expected performance of market-available consumer grade technologies. To later contrast them in our current research for future datacenter technologies. As well as to make the results publicly available for other researchers and Big Data practitioners.

This study is part of the ALOJA project, an open research initiative from the Barcelona Supercomputing Center (BSC) to increase cost-efficiency and the general understanding of Big Data systems via automation and learning.

1.1 Organization

The report is organized as follows: Section 2 presents the ALOJA project in which this study is framed. Section 3, the technical specifications of the cluster are provided. Followed by Section 4, which presents performance metrics, speedups and cost-effectiveness data covering different subsets of executions. In Section 5 we highlight the main results and observations from the benchmarks. Finally, in section 6 we make some final remarks and present the future lines for the project.

2 The ALOJA Big Data benchmarking project

The ALOJA project [10] is an open initiative from the Barcelona Supercomputing Center (BSC) to explore and automate the characterization of cost-effectiveness for Big Data deployments. BSC is a center with over 8 years of research expertise in Hadoop environments, which counts support from industrial partners in the area of Big Data technologies. ALOJA attempts to provide solutions to an every time more important problem for the Big Data community, which is the lack of understanding of what parameters, either SW or HW, determine the performance of Big Data workloads. Therefore the selected configuration determines the speed in which data is processed and returned, and most importantly, the hosting budget.

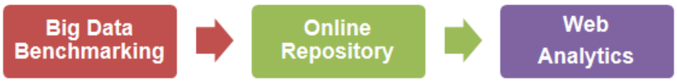

Figure 1: *Main components and workflow in the ALOJA platform*

Additionally, as open-source frameworks *i.e.*, Hadoop and Spark become more common, they can be found in a diversity of operational environments. Comprising from low-end commodity clusters, low-powered microservers, to high-end data appliances, including all types of Cloud-based solutions at scale *i.e.*, IaaS and PaaS. Where due to the large number of software configuration options —more than 100 interrelated parameters [4, 5, 6], and the increasing number of deployment types optimizing the performance of Big Data systems requires extensive manual benchmarking [5, 8]. For these reasons, as well as the lack of automation tools for Big Data had led us to build progressively, an automated benchmarking platform to deal with defining cluster setups, server deployment, defining benchmarking execution plans, orchestration of configuration, and data management of results.

The ALOJA platform is further composed of open-source tools to achieve an automated benchmarking of Big Data deployments. These tools are used by researchers testing new features and algorithms, or by practitioners needing either to privately test their own system and application, or to improve benchmark results. The 3 main components are: Big Data benchmarking scripts, that deploy servers and execute benchmarks; the online repository; and Web Analytics tools, that feed each other in a continuous loop as benchmark are executed as shown in Figure 1.

2.1 Benchmarking methodology

Due to the large number of configuration options that have an effect on Hadoop's performance, this study makes use of the ALOJA platform to manage and update cluster configuration, orchestrate benchmark execution and data collection. In order to automate the process of testing over 1,000 benchmark runs with different settings. As well as to automate the evaluation of the results through the online analytic tools [1], following the workflow previously presented in Figure 1.

Hadoop's distribution includes jobs that can be used to benchmark its performance, usually referred as *mi-*

cro benchmarks, however these type of benchmarks usually have limitation on their representativeness and variety [7]. ALOJA currently features the HiBench open-source benchmark from Intel [7], which can be more realistic and comprehensive than the supplied example jobs in Hadoop. HiBench features several ready to use benchmarks from 4 categories: micro benchmarks, Web search, Machine Learning, HDFS benchmarks. While the figures in the next sections show the results from functional benchmarks, more specifically *terasort* (using 100GB), *DFSIOE R/W*, *sort* and *wordcount*; the whole HiBench was run and results can be found in the online repository [1] detailed next.

2.2 Online repository and Platform Tools

Part of the goals for ALOJA include the creation of vendor-neutral, open public Big Data benchmark repository. This effort currently features more than 50,000 Big Data job benchmark runs, along with their performance traces and logs. As few organizations have the time or performance profiling expertise, we expect the repository will benefit the Big Data community to complement their cost-performance comparison needs, without having to repeat benchmarks. As well as to bring more transparency to product claims and reports, as results are fully disclosed and experiments can be easily repeated, especially for public Cloud results.

In total for this study, more than 1,000 executions were run on the cluster, with different combinations of storage, networking and Hadoop configuration options *i.e.*, the concurrent number of containers (mappers or reducers). The results repository keeps growing as we continue the project's benchmarking efforts. The results from this report can also be compared with online resources from different clusters including Cloud deployments.

3 Experimental setup

The system under test (SUT) is composed of 9 machines, of which 1 is used as the master node that inludes the Hadoop *NameNode* and *JobTracker* services. The rest 8 nodes, have a worker role (slaves) and each include the Hadoop *DataNode* and *TaskTracker* services. Each machine has 64GB of RAM and 12 CPU cores (2x6) with hyperthreading, for a total of 24 hardware threads. Storage-wise, each machine is equipped with 6x1 TB SATA drives, 1 for the OS, 5 for application use, and 2x240GB *SanDisk CloudSpeed ECO* SATA SSDs for application use. Finally, each node has a 4x1Gb Ethernet adapter (only one port used for the tests), and a 2xFDR 56Gb InfiniBand adapter (only one port used for the tests using *IPoIB*.

Cluster	
Processor	Intel(R) Xeon(R) E5-2630L@2.00GHz
# of nodes	8 + 1 master
Memory Capacity	64GB
Storage (1)	1x1TB SATA 3.5" 7200RPM (OS)
Storage (2)	5x1TB SATA 2.5" 5400RPM (apps)
Storage (3)	2x240GB SATA SSD (apps)
Interconnect (1)	4 x 1 Gb Ethernet
Interconnect (2)	2 x FDR 56 Gb IB
CPU	
Architecture	Sandy Bridge
# of sockets	2
Cores per socket	6
Threads per core	2
# of cores	12 (6x2)
# of hw threads	24 (2x6x2 = 12x2)
Max frequency	2.0 GHz
Memory BW	42.6 GB/s
Hadoop	
Version	Apache Hadoop v.1.0.3
OS	
Type	GNU/Linux
Version	Ubuntu Server 14.04 (Trusty Tahr)
Kernel	3.13.0-65-generic x86_64

Table 1: *Specifications for the System Under Test*

Table 1 summarizes the specific HW components and versions used in this study, as well as the Operating System (OS) version. The Hadoop version used for the tests is *1.0.3*. This particular version was chosen as it is compatible with the *Java instrumentation Suite* (JIS) [10] developed at BSC, which allows low-level performance analysis by using BSC's HPC performance tools [2]. Results with other versions including Hadoop v2 can be found in the online repository. The operating system was Ubuntu Linux 14.04 (*Trusty Tahr*) with kernel version 3.13.

3.1 Tested HW and SW configurations

With respect to storage, a JBOD (Just-a-Bunch-Of-disks) setting was selected as recommended by the Hadoop Guide [13], to increase Hadoop's disk throughput. But at the expense of application-level reliability, as opposed to commonly used hardware-level RAID

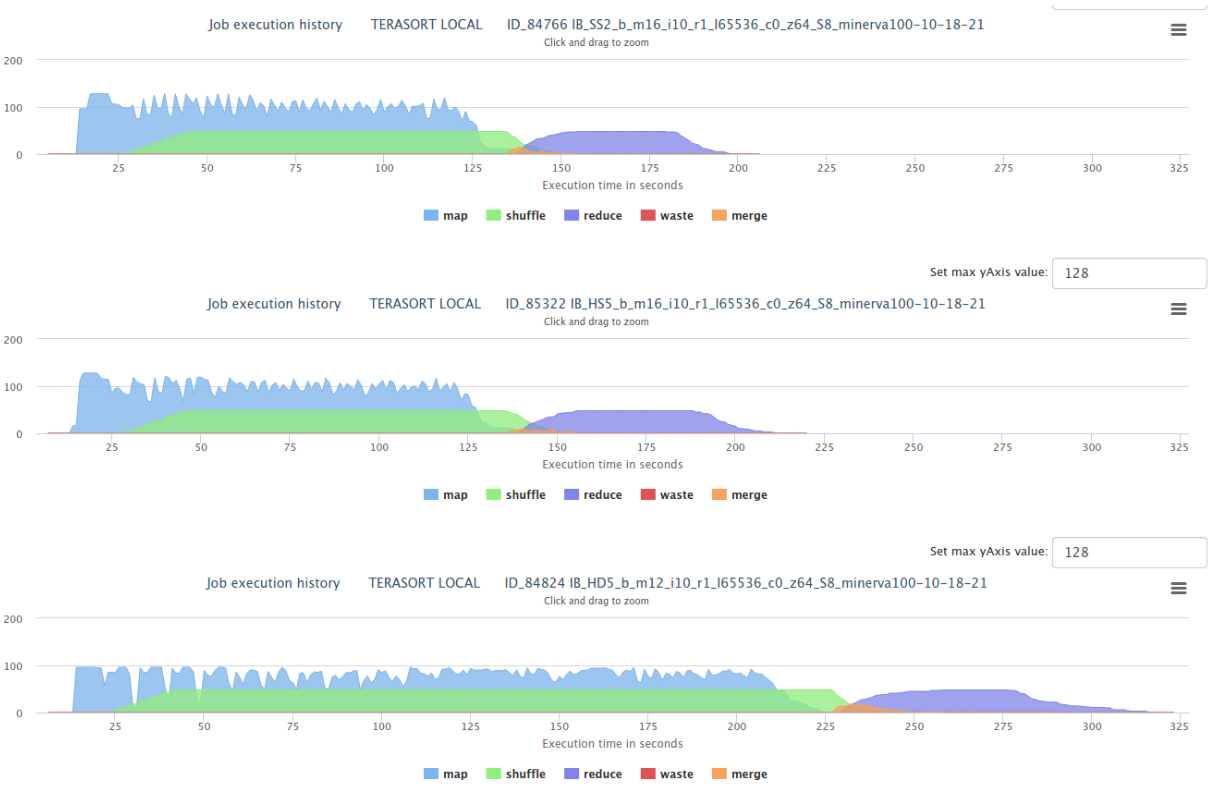

Figure 2: *Comparison of Hadoop processes and execution time for the fastest runs in different disk configuration categories*

technologies for database systems.

The following different disk combinations were tested:

- From 1 to 5 SATA drives as JBOD, as opposed to more reliable RAID technlogies as recommended by the Hadoop Guide [13], identified in test names by HDD, HD2, HD3, HD4, HD5 respectively.
- 1 and 2 SSDs as JBOD (SSD, SS2).
- 5 SATA drives as JBOD and Hadoop /tmp directory on 1 SSD (HS5).

With respect to networking, 1Gb Ethernet and InfiniBand (56Gb FDR using IPoIB) were compared. With respect to Hadoop, the number of parallel container per node was tuned *i.e.*, number of mappers/reducers, different numbers were tested *i.e.*, 12, 16 and 24. Also different block sizes, I/O buffers sizes, and compression. More details on the methodology for selecting SW configurations in [10].

4 Evaluations

This section presents the impact of different Hadoop configuration parameters, as well as hardware configurations to both performance and costs. The following experiments can also be obtained in ALOJA's online application [1], which provides further details and access to the logs.

4.1 Performance metrics

The most relevant metric to determine performance of a Hadoop application is the execution time. The time it takes a Hadoop job to complete, in this case we measure it in seconds. As an example of the different Hadoop internal processes per job, Figure 2 presents the different Hadoop phases for *terasort* of the fastest runs for 3 different disk configurations: 2 SSDs, 5 SATA disks and /tmp in 1 SSD, and 5 SATA disks respectively. In this case all using IPoIB networking, as it gave the best results over GbE for each of the different disk configurations.

The fastest configuration as expected is using 2 SSDs, 207s. However, by using a combination of 5 SATA drives and 1 SSD for temporary files, the second combination achieves a close 228s, while increasing the capacity at the cost of the extra SATA drives. The duration of the reduce phase is roughly the same for the 3 jobs in Figure 2, but when Hadoop's temporary directory resides on flash storage —first

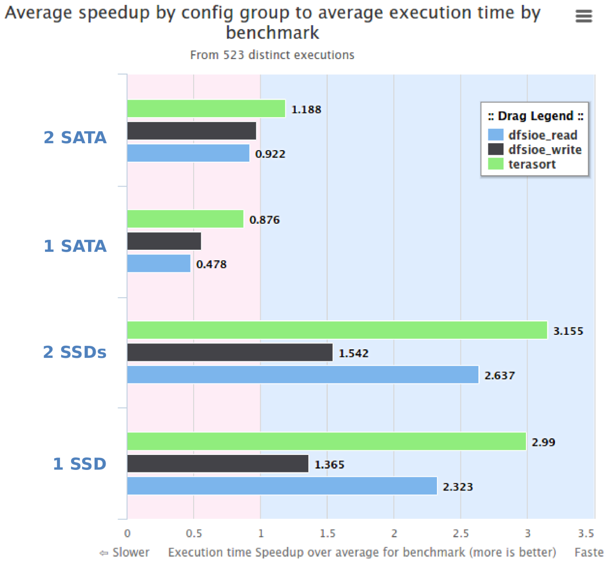

Figure 3: *Speedup by storage configuration*

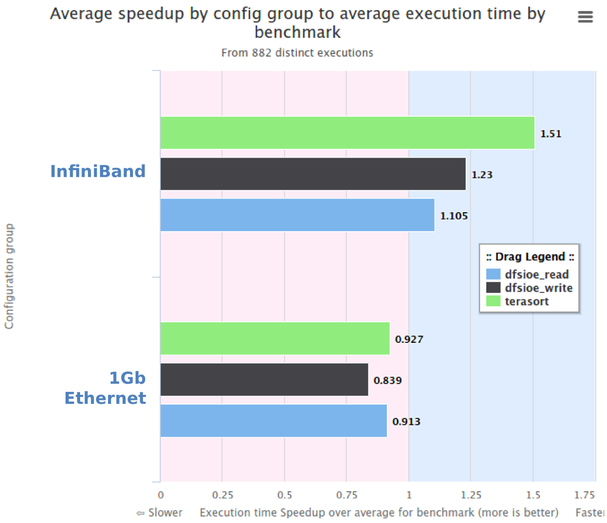

Figure 4: *Speedup by network configuration*

and second job— the duration of the shuffle phase is significantly shorter. The third configuration with 348s is the fastest among all those using only SATA disks (5 of them), but 68% slower than the 2 SSD option. The rest of the results for other benchmarks can be seen online. Briefly to summarize, *wordcount* being CPU-bound achieves similar numbers on all 3 of the configurations. *DFSIOE* does not make use of temporary storage, so the SSD only solutions is significantly faster than the combination or the SATA only solutions. *DFSIOE* results can be seen in the next sub-section that details the *speedup* of selected benchmarks.

4.2 Performance Speedups

The *speedup* is defined as the relative performance improvement, in this case, to the average execution time for the selected benchmark. Where the average execution time is at 1 on the X axis. Any value below 1 represents a *speed-down* and a value above 1 represents a *speed-up* over the average. Figure 3 shows speedups by comparing four storage configurations of 523 different runs. The aggregation groups being: 2 SATA, 1 SATA, 2 SSDs, 1 SSD respectively. Here we can see that the configuration with 2 SSDs is the best on all benchmarks, followed by the single-SSD one. The speedup is especially notable for *terasort*. We can also notice here that while scaling in SATA drives from 1 to 2, the speedup almost doubles, for SSDs the increase is limited between 15 to 30% according to the benchmark. Since SSDs scale as well as SATAs or better, this is probably due to either the benchmark or Hadoop not pushing enough data to fully utilize the SSD performance, or to some hardware or OS bottleneck.

Figure 4 shows speedups by network configuration (Ethernet vs. InfiniBand) by aggregating 882 different runs of the selected benchmarks. Where InfiniBand is faster than Ethernet as expected, but different according to the benchmark. While *DFSIOE* Read gets up to 60% increase, *terasort* 20%, both in average.

The next Figures 5 for ethernet and 6 for InfiniBand, shows the effect of varying the number of concurrent containers *i.e.*, mappers and reducers in parallel, to each disk technology. In Figure 5 (ethernet-based), we can see that the most optimal configuration for 1 and 2 SSDs is by using 16 concurrent processes per node. For 5 SATA + /tmp in SSD, 24 concurrent processes is the most efficient. While for 5 SATA, 12 concurrent processes is the maximum concurrency allowed by the HW. On Figure 6 (IB-based), we can see that for 1 SSD, also the most efficient is 16 containers, while having 2 SSDs allows stretching it to 24 concurrent processes, giving 10% more speedup. For the SATA tests, the same values as with ethernet are recommended.

Varying the number of concurrent processes per node according to the HW has the following implications: having a higher number of disks in the system, allows for running more concurrent processes, speeding up runs. This is the same for 5 SATA drives as for SSD disks. However while SATA doesn't get significant boost with InfiniBand; for SSD disks, to have higher concurrency by adding more disks, fast

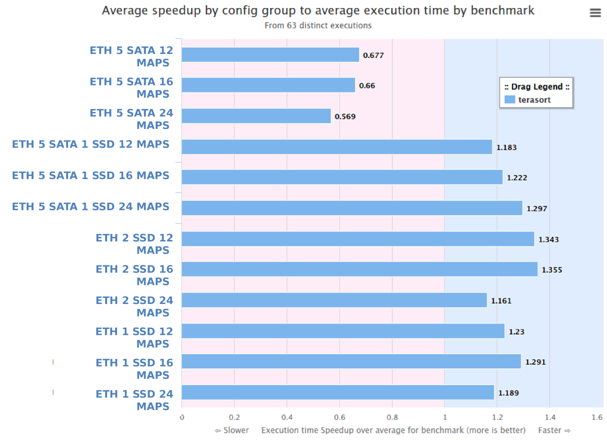

Figure 5: *Speedup by storage + network + Hadoop processes (ethernet)*

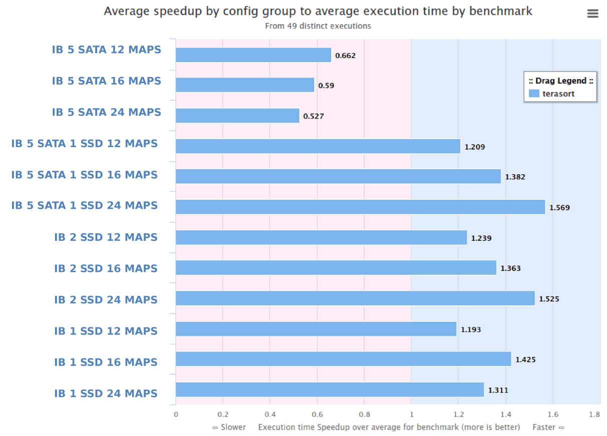

Figure 6: *Speedup by storage + network + Hadoop processes (InfiniBand)*

networking is required as shown in Figure 6 in the InfiniBand use case.

4.3 Cost-effectiveness

While the previous sub-section showed best results in absolute performance terms, here we rank executions by the execution time over the total cost of executing the job, as if it was a cloud-based resource being fully utilized at all times. The cluster costs are calculated by adding the HW costs of the particular setup. Amortizing HW over 3 years, adding service and maintenance cost over this period to estimate the cost per hour of the setup. The costs used for the SUT can be seen at the `http://aloja.bsc.es/clustercosteffectiveness` page, along with the rest of the clusters benchmarked for ALOJA. The page also allows the user to change costs, to experiment the impact of expected costs.

The cost-effectiveness diagram in Figure 7, places different evaluated HW configurations as if they were different clusters, where each bubble represent a different cluster. It places each cluster at an (X,Y) location of a bidimensional space, where the X and Y axis represent the normalized execution time and cost respectively. The figure is further divided into 4 quadrants, to represent the execution cost vs. performance result. The quadrants being: Fast-Economical, Fast-Expensive, Slow-Expensive, and Slow-Economical. With point (0,0) represents the best cost-effective execution. The bubble sizes in this case represent the number of executions that fit each HW configuration. This type of analysis, allow the user to quickly filter out *slow-expensive* HW configurations, to concentrate (run more benchmarks with different SW configurations) in the most beneficial ones, closer to point (0,0) in the chart.

To summarize numerically Figure 7, Table 2 ranks the best configurations with respect to cost-effectiveness with their absolute costs and performance obtained. The best configuration takes 207 seconds, with a cost of $ 0.33 USD per run. As best ranked configuration we have IB networking and 2 SSD disks. This higher-end setup is also the most cost-performing in this case. Of course the storage capacity for this configuration is not very large, so if higher capacity is needed, one of the configurations with 5 SATA disks would be more appropriate, although possibly a bit slower and more expensive.

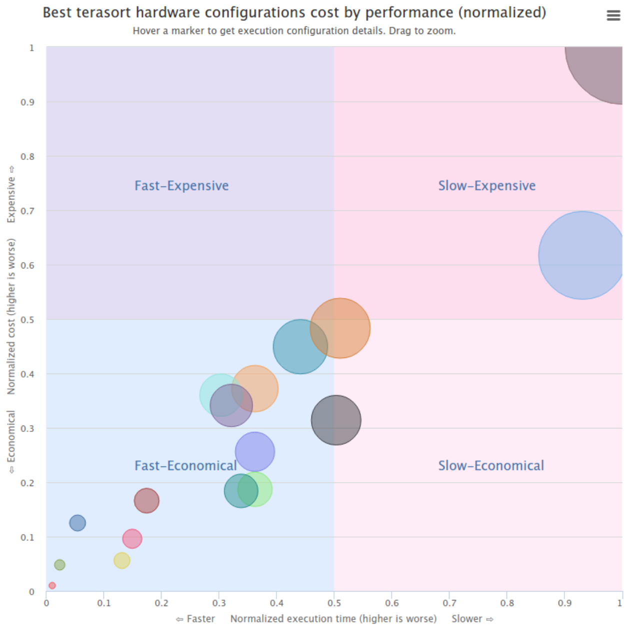

Figure 7: *Comparison of cost effectiveness*

Rank	Cluster	Exec cost	Exec time	Network	Disk
1	minerva-100-10-18-21	0.33 US$	207 s	InfiniBand	2 SSD drives
4	minerva-100-10-18-21	0.35 US$	258 s	Ethernet	2 SSD drives
5	minerva-100-10-18-21	0.37 US$	265 s	InfiniBand	1 SSD drive
3	minerva-100-10-18-21	0.38 US$	228 s	InfiniBand	5 SATA /tmp to SSD
6	minerva-100-10-18-21	0.40 US$	275 s	Ethernet	5 SATA /tmp to SSD
8	minerva-100-10-18-21	0.41 US$	348 s	Ethernet	1 SSD drive
9	minerva-100-10-18-21	0.44 US$	348 s	Ethernet	5 SATA drives
10	minerva-100-10-18-21	0.48 US$	325 s	InfiniBand	5 SATA drives

Table 2: *Summary of cost effectiveness by setting up HDFS to different disk configurations*

5 Discussion

From the speedup diagrams it can be seen that, while SSD and InfiniBand always provide better performance, the actual gain depends on the specific benchmark under consideration. For *terasort* and *DFSIOE* the speedup is considerable, but for *wordcount* it is negligible as it is CPU-bound (not shown in the paper, results can be obtained online). *terasort* benefits the most from having Hadoop's temporary directory on SSD as it uses it heavily, while DFSIOE R/W do not use it. Notable speedup results for *terasort* with respect to storage (using IB networking):

- 5 SATA disks in JBOD configuration are 80% faster than a single SATA disk.
- A single SSD disk is 300% faster than a single SATA disk and at least 100% faster than 5 SATA disks (depending on number of processes)
- Doubling the number of SSDs (2 vs 1) only gives only a 20% performance improvement, due to Hadoop or the benchmark not pushing enough data (since SSDs normally scale well). Less benefit is expected by adding a 3rd volume, especially in the ethernet case.
- 5 SATA disks in JBOD configuration + 1 SSD for Hadoop temporary directory are 300% faster than 1 SATA disk and more than 100% faster than 5 plain SATA disks (without SSD). This is the same performance as 2 SSDs, but with a much larger storage capacity.

Notable speedup results for *terasort* with respect to IB vs. Ethernet networking:

- On average, IB is 60% faster than Ethernet.
- The fastest SSD configuration gets a speedup of 13% with IB.
- Due to the higher network speed, IB stresses more the disks in terms of IOPS and throughput, so when using IB the disk is more of a bottleneck than with Ethernet.
- IB enables to increase the performance by adding more SSDs as a JBOD into the system.

Most cost-effective disk/net configurations for *terasort* (see also table 2):

- First: 2 SSDs for HDFS + IB.
- Second: 2 SSDs for HDFS + ETH.
- Third: 1 SSD for HDFS + IB.
- Fourth: 5 SATA for HDFS, 1 SSD for /tmp + IB
- Fifth: 5 SATA for HDFS, 1 SSD for /tmp + ETH

6 Conclusions

This report presented the impact of introducing both SSDs and InfiniBand to a commodity-style Hadoop cluster. A setup that is typically found in many enterprises, but with the addition of the SSDs and IB for evaluation. We have run over 1,000 benchmarks of different application domains by varying software configurations to test the distinct hardware scenarios. As an example, we have shown that for InfiniBand networks to be cost-effective, they need to be combined with SSDs or other fast disk options. If not the improvement they provide is negligible to justify the costs. Also, best Hadoop configuration options such as the number of containers (mappers and reducers) to run in parallel according to the available CPU cores and job types, or the best cost effective compression factor to use according to the different workloads was presented. Showing the best performing concurrency varies according to the chosen disk and network. Results show that as expected, both technologies can speedup Big Data processing. However, unlike commonly perceived, results show that SSDs and InfiniBand can actually improve the cost-effectiveness, even in small clusters when they are busy most of the time. Ultimately, they speedup up Big Data usage while reducing the TCO at the same time, and possibly space and energy costs.

These results are also utilized as baseline numbers for future work in the ALOJA project, to compare them with larger cluster sizes, Cloud-based deployment and newer technologies *i.e.*, PCIe NVRAM disks (*FusionIO* - now part of SanDisk). We believe that a tiered storage layer consisting of different hierarchy levels from: RAM, PCIe Disks, SSDs, and rotational; can greatly reduce operation costs, while improving significantly the availability of Big Data applications. Hadoop is already working towards this and supports tiered storage starting from version 2.3, although this hasn't been as widely adopted and publicized by the various Hadoop distros, see [3] for an example.

ALOJA provides to the Hadoop community by producing more knowledge and understanding of the underlying Hadoop runtime while it is executing. Our intent is that researchers and organizations evaluating or deploying the Hadoop solution stack will benefit from this growing database of performance results and configuration guidance.

Acknowledgements

We would like to thank SanDisk corporation for supplying the SSDs used in the experiments and the domain knowledge providing optimizing system settings. This work is partially supported by the BSC-Microsoft Research Centre, the Spanish Ministry of Education (TIN2012-34557), the MINECO Severo Ochoa Research program (SEV-2011-0067) and the Generalitat de Catalunya (2014-SGR-1051).

About BSC and the authors

The Barcelona Supercomputing Center (BSC) was established in 2005 and is the Spanish national supercomputing facility. The Center houses MareNostrum, one of the most powerful supercomputers in Europe. The mission of BSC is to research, develop and manage information technologies in order to facilitate scientific progress. BSC was a pioneer in combining HPC service provision, and R&D into both computer and computational science (life, earth and engineering sciences) under one roof. The centre fosters multidisciplinary scientific collaboration and innovation and currently has over 400 staff from 50 countries. BSC also participates in various bilateral joint research centers with companies such as IBM, Microsoft, Intel, NVIDIA and Spanish oil company Repsol. The centre has been extremely active in the EC Framework Programmes and has participated in seventy-nine projects funded by it. BSC has been a pioneer in research relating to Hadoop starting from 2008 with an initial collaboration with IBM to explore scheduling of Map/Reduce jobs. The collaboration resulting in contributions on SLA-aware scheduler to the Hadoop community. Since 2013, BSC has been working closely with both Microsoft product and research groups exploring new architectures for Hadoop as the ALOJA project. BSC maintains the largest Hadoop benchmarking repository with over 50,000 Big Data benchmark executions, freely available for Hadoop community. It also provides a development sandbox to test analytic and learning algorithms on top of the growing body of benchmarks.

Dr. Nicolas Poggi is an IT researcher with focus on performance and scalability of Data intensive applications and infrastructures. He is currently leading a research project on upcoming architectures for Big Data at BSC and Microsoft Research joint center. He received his PhD in 2014 AT BarcelonaTech-UPC, where he previously obtained his MS degree in 2007. He is part of the High Performance Computing group at DAC, and of the Data Centric Computing (DCC) research group at BSC. He has also been a Visiting Research Scholar at IBM Watson in 2012 working in Big Data and system performance topics. Nicolas combines in his work both a pragmatic approach to performance and scalability with the use of Machine Learning techniques. His aim is to improve and foster open Internet technologies, making them more reachable and most importantly, efficient to host.

Dr. David Carrera manages the Data Centric Computing (DCC) research group at BSC and has lead and been involved in several EU and industrial research projects. He is also an Associate Professor at BarcelonaTech-UPC, where he previously received his MS degree in 2002 and his PhD from the same university in 2008. His research interests are focused on the performance management of data center workloads. In 2015 he was awarded an ERC Starting Grant for the project HiEST. He received an IBM Faculty Award in 2010. He is an IEEE member.

References

[1] BSC. Aloja home page: http://aloja.bsc.es/, 2015.

[2] BSC. Performance tools research group page: http://www.bsc.es/computer-sciences/performance-tools, 2015.

[3] eBay tech blog. Hdfs storage efficiency using tiered storage: http://www.ebaytechblog.com/2015/01/12/hdfs-storage-efficiency-using-tiered-storage, 2015.

[4] D. Heger. *Hadoop Performance Tuning https://hadoop-toolkit.googlecode.com/files/White paper-HadoopPerformanceTuning.pdf.* Impetus, 2009.

[5] D. Heger. *Hadoop Performance Tuning - A Pragmatic & Iterative Approach.* DH Technologies, 2013.

[6] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu. Starfish: A self-tuning system for big data analytics. In *In CIDR*, pages 261–272, 2011.

[7] S. Huang, J. Huang, J. Dai, T. Xie, and B. Huang. The HiBench benchmark suite: Characterization of the MapReduce-based data analysis. *Data Engineering Workshops, 22nd International Conference on*, 0:41–51, 2010.

[8] K. Kambatla, A. Pathak, and H. Pucha. Towards optimizing hadoop provisioning in the cloud. In *Proceedings of the 2009 Conference on Hot Topics in Cloud Computing*, HotCloud'09, Berkeley, CA, USA, 2009. USENIX Association.

[9] J. O'Reilly. Ssd prices in a free fall: http://www.networkcomputing.com/storage/ssd-prices-in-a-free-fall/a/d-id/1320958, 2015.

[10] N. Poggi, D. Carrera, A. Call, S. Mendoza, Y. Becerra, J. Torres, E. Ayguadé, F. Gagliardi, J. Labarta, R. Reinauer, N. Vujic, D. Green, and J. Blakeley. ALOJA: A systematic study of hadoop deployment variables to enable automated characterization of cost-effectiveness. In *2014 IEEE Intl. Conf. on Big Data, Big Data 2014, Washington, DC, USA, October 27-30, 2014*, pages 905–913, 2014.

[11] PriceG2. When will ssd have same price as hdd: http://www.storagenewsletter.com/rubriques/market-reportsresearch/when-will-ssd-have-same-price-as-hdd-priceg2, 2013.

[12] K. Vermes. The price of ssds will continue to fall: http://www.digitaltrends.com/computing/as-ssd-capacity-starts-to-grow-the-prices-are-starting-to-come-down, 2015.

[13] T. White. *Hadoop: The Definitive Guide.* O'Reilly Media, Inc., 1st edition, 2009.