

Benchmarking AyraDB Next-Generation Database on Davinci-1 Super-Computer

Carlo Cavazzoni², Chiara Francalanci³, Paolo Giacomazzi³, Nicolò Magini², Roberto Morelli² and Paolo Ravanelli¹

¹ Cherrydata srl, Via Abano 9, Milano, 20131, Italy

² Leonardo S.p.A, Torre Finmeccanica, Via Raffaele Pieragostini, Genova, 16149, Italy

³ Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, 20133, Italy

Abstract

This paper presents the results of a 1-year project aimed at testing the performance and scalability of AyraDB, a next-generation database, on Davinci-1 super-computer. AyraDB is unique as it is *fully peer-to-peer*, with no central node to coordinate other storage nodes and *no need for caching data in-memory*, with consequent linear scalability. All competing solutions, including the most popular databases such as MongoDB and Redis, are not fully peer-to-peer, have central coordination nodes and/or cache data in memory with a bottleneck to scalability. These unique features make AyraDB particularly suitable to store and retrieve satellite data, which are notoriously challenging due to their size and layered structure, and position it as an enabler of new use cases in the space economy. In this paper, we present the results of a large-scale test aimed at verifying and measuring AyraDB's performance on Davinci-1's HPC infrastructure. An HPC infrastructure is in fact designed for data- and processing-intensive applications and can support scalability without infrastructural bottlenecks limiting software performance. Tests have been performed with a total of 500 runs on a number of 8-core servers growing from 1 to 20 and a data size growing from 10 Gbyte to 500 Gbyte. Results show how AyraDB reaches 1 million requests/s with 13 servers, with linear scalability and consistent read/write performance, beating the state of the art by 5X factor.

Keywords

Big data, big-data infrastructure, database, horizontal scalability, davinci-1, ayra db.

1. Introduction

Estimates indicate that the growth trend for big data projects is still exponential [3]. There is a clear market need for new and more efficient data management technologies to enable a variety of big data applications. For example, satellite data are notoriously big data and raise several challenges not only from the perspective of analytic applications (such as statistics or machine learning), but also to be stored and retrieved efficiently (see, for example, [6]). These challenges have led to the design a new database, called AyraDB, that can be an enabler of big-data applications, from both a technical and economic point of view [7]. The key requirements in the design of AyraDB have been: 1) horizontal scalability, to accommodate growing capacity requirements by adding nodes to the database infrastructure, 2) on-disk data storage with no caching, to work efficiently with all types of data access loads, 3) thrifty hardware consumption through greater performance, to reduce costs and, thus, increase

ITADATA2022: The 1st Italian Conference on Big Data and Data Science, September 20–21, 2022, Milan, Italy

EMAIL: carlo.cavazzoni@leonardo.com (A. 1), chiara.francalanci@polimi.it (A. 2), paolo.giacomazzi@polimi.it (A. 3), nicolo.magini@leonardo.com (A. 4), roberto.morelli.ext@leonardo.com (A. 5), paolo.ravanelli@cherry-data.com (A. 6)

ORCID: 0000-0002-0373-8065 (A. 2); 0000-0001-9584-3200 (A. 3)



© 2022 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

economic scalability. We believe that by satisfying these fundamental requirements, AyraDB can be effectively applied to the management of large datasets, providing performance and economic benefits with respect to competing databases.

The goal of the research presented in this paper is to perform a large-scale test of AyraDB to verify and measure AyraDB's performance edge. A large-scale test requires an HPC infrastructure, which is in fact designed for data- and processing-intensive applications and can support scalability without infrastructural bottlenecks limiting software performance. We have executed a large-scale test of AyraDB's performance and scalability on Davinci-1 super-computer, located in Genova and belonging to Leonardo Company. Davinci-1 is the perfect infrastructure to run AyraDB's tests. Davinci-1 is managed in house with total control of the infrastructure, enabling the design of a test with no bottlenecks and providing dependable performance benchmarks. These tests are key to demonstrate the technical and business readiness of AyraDB, since an efficient management of large datasets represents a critical success factor.

Finally, these tests have helped us estimate the energy savings enabled by AyraDB. We have estimated that AyraDB can significantly reduce hardware and, thus, energy consumption. With the tests presented in this paper, we have gained a deeper understanding of the green benefits of AyraDB and verified our estimates. In terms of long-term sustainability, we believe that the test can act as a proof of concept of the technical features of AyraDB and of its suitability for challenging applications. A number of application opportunities could open up, particularly in the space industry, enabled by the mix of performance, scalability and favourable economics of AyraDB.

The presentation is organized as follows. The next section reviews the state of the art. Section 3 presents the main technical features of AyraDB. Section 4 describes the characteristics of the Davinci-1 super-computer. Section 5 presents the results of our tests. Conclusions are finally drawn in Section 6.

2. State of the art

An extensive survey of benchmarking efforts of big data systems can be found in [8]. Most scientific benchmarking studies focus on a) comparative analyses of selected market solutions with a limited number of nodes (typically 3) and rarely test the maximum performance of databases or their scalability limits, understandably due to the cost of such large-scale tests, b) the design of fair and objective benchmarking tools executing the right set of tests to obtain dependable results. This indicates a general focus of scientific studies on methodological issues and a lack of scientific (particularly open access) studies pushing performance to a limit.

Business owners of database solutions and market service providers tend to provide more extensive benchmarks, sometimes executed by (paid) third parties. The clear conflict of interest makes these benchmarks potentially optimistic towards the marketed solution, against competitors. However, for the same reason, we would not expect the marketed solution to perform better than the corresponding benchmarks provided by the owner/service provider. In other words, we could consider these benchmarks as best case scenarios.

The focus of this paper's benchmarking effort is a key-value read/write load. With this focus, looking at the top performing solutions we have found that the highest performance obtained is approximately 300k operations/s. For example, Couchbase has published benchmarking results showing how Couchbase can reach 300k operations/s with a cluster of 10 servers, with a negligible increase when the number of servers is doubled [9]. According to this benchmarking effort, MongoDB can reach only 200k operations/s with 20 servers. Scylladb's Website provides a benchmark showing a maximum per-core throughput of Scylladb of 6.2k operations/core and an overall maximum throughput of (again) 300k operations/s. This throughput halves when write operations are performed and is reduced by a factor of 3 with a load that limits the benefits of in-memory caching. In [12], authors provide benchmarks for Riak database, showing how Riak DB has a performance that settles around 300k operations/s with an enterprise-significant number of keys in the database.

A thorough market analysis has been conducted as part of the Hippj project (SME Instrument phase 1, [2]). The resulting market position of AyraDB is shown in Figure 1. AyraDB represents the only

database that can offer both technical and economic scalability. While in-memory databases have a technical scalability that is comparable with that of AyraDB, their economics are far less favorable. On-disk databases, are less expensive, but they lack in latency, throughput or both. This indicated that AyraDB involves no trade-off between cost and performance and no limitations on the application scenarios.

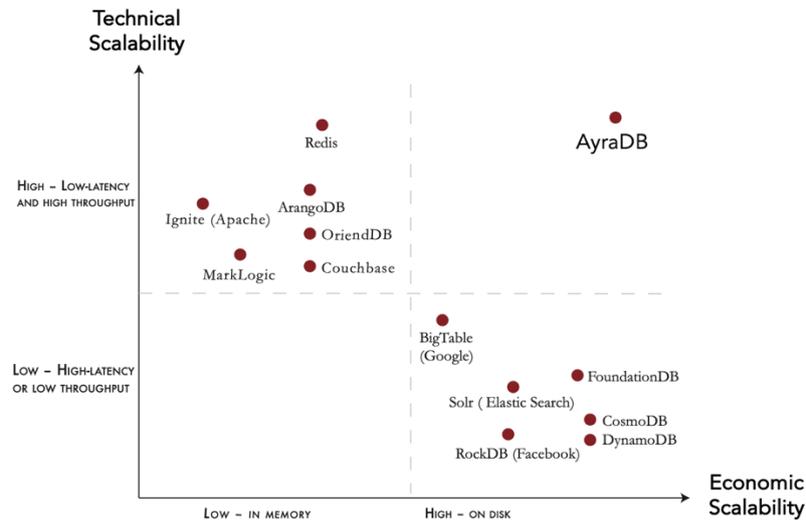


Figure 1: Market position of AyraDB

A distinctive feature of AyraDB is that data are stored on disk and do not need to be cached in memory to obtain sub-millisecond response time. A sub-millisecond response time can be reached with commodity SSD hard disks. These features position the economics of AyraDB in the high-performance range. At the same time, costs are reduced by a factor of 20, as memory is far more expensive than disks. A competing solution with a technical and economic scalability comparable to that of AyraDB could not be found in [2].

A best-in-class cost-to-performance ratio is only one dimension of AyraDB’s competitiveness. AyraDB has *linear* scalability. This means that if hardware capacity doubles, AyraDB’s performance doubles accordingly. Competing solutions do not scale linearly, as to double their performance, they require more than double capacity (see, for example, [9]). Above a certain amount of capacity, their performance stops increasing. This means that even if the cloud provider can provide additional capacity, the database cannot exploit it to scale up. In contrast, AyraDB can use additional capacity efficiently, with no upper bounds to scalability.

3. Technical features of AyraDB

The core of AyraDB is a key-value database. Data are stored in tables, on disk, where a table is a collection of records, indexed by a key. An AyraDB cluster can have one or multiple servers, and in a multi-server cluster all servers have the same role, and there is not a centralized server that manages the cluster, that is, the AyraDB cluster is fully peer-to-peer.

The records of a table are distributed among all the servers of a cluster, based on the hash of the key that indexes the record. Therefore, all servers statistically store the same number of records.

AyraDB implements three types of tables:

- *fixed_length*: this table has a column scheme, that is, the user can configure the number of fields (columns) of the record, and the labels of the columns. All records in the same *fixed_length* table have the same column structure. Moreover, in the *fixed_length* table the maximum length in bytes

of each field is explicitly defined. Fixed_length tables are particularly efficient in use cases such as time series, satellite imagery.

- *padded*: the difference between padded and fixed_length tables is that in padded tables there is no maximum limit on the length of a field.
- *nosql*: nosql tables have no column scheme. Each record can have an arbitrary number of fields, with arbitrary labels.

Each table has a configurable replication factor R . With $R=1$, a record is stored on one server only. With $R=2$, a record is stored on two servers, and so on. In a cluster with M servers, the maximum replication factor is $R=M$, meaning that each record is stored on all servers. Each table may have a different replication factor. Replication is used to make the database more reliable, because if $R=2$, the failure of a single server does not prevent us from reaching all records. In general, with replication factor equal to R , the system is fully functional even with the failure of $R-1$ servers.

AyraDB exposes a set of HTTP-REST APIs, that are normally used wrapped in higher-level APIs in various programming languages such as C, Java, Python, and NodeJS. The basic record-based APIs are:

- *READ*: to read a record (or a set of fields), given the target table and key.
- *INSERT/UPDATE*: to write/update a record (or a set of fields), given the target table and key.
- *DELETE*: to delete a record given the target table and keys. In nosql tables, the DELETE API can be used to remove a field from a record.

The table-based APIs are:

- *CREATE_TABLE*: to create a new table
- *DELETE_TABLE*: to completely remove a table
- *TRUNCATE_TABLE*: to delete all the records of a table
- *RESTRUCTURE_TABLE*: to modify some features of a table, such as the replication factor, or the column scheme (for fixed_length and padded tables).

The server-based APIs are:

- *ADD_SERVER*: to add a server to a cluster
- *REMOVE_SERVER*: to remove a server from a cluster

Replication is managed at record level. With a replication factor equal to R , each record has R copies on R different servers. If at least one of the servers storing the same record is active, a read query for that record is successful. With a replication factor of R , the AyraDB cluster can operate transparently and consistently with up to $R-1$ failed servers. Consistency is guaranteed through the synchronization operation that is performed automatically when the servers become available again. AyraDB provides automatic synchronization of data when one or multiple servers fail or are shut down for maintenance reasons. Write/delete operations that would be performed on unavailable servers are logged persistently on disk and, when the server becomes available again, logged operations are performed in the order in which they were logged.

4. Davinci-1 super-computer

The benchmarks performed in this work were run on a private cloud infrastructure provisioned through the *OpenStack*® cloud technology [4]. This framework allows easy deployment of the servers needed for the benchmarks. The computing nodes used to instantiate the servers are part of the high-performance computing cluster Davinci-1 located in Genoa and belonging to the Leonardo company. The HPC is equipped with 56 CPU nodes and 80 GPU nodes. Both InfiniBand and 10 Gbps Ethernet connections are available for node communication, but for the benchmark only Ethernet was used. A detailed schema of the infrastructure is reported in Figure 3.

The nodes used for the cloud tenant belong to the CPU family and are equipped with 2 24-core CPUs (Intel® Xeon® Platinum 8268 CPU @ 2.90GHz) and 768 GiB of RAM. Through the

OpenStack® layer, virtualization of the servers that are used for scaling of the benchmarks described in this work takes place.

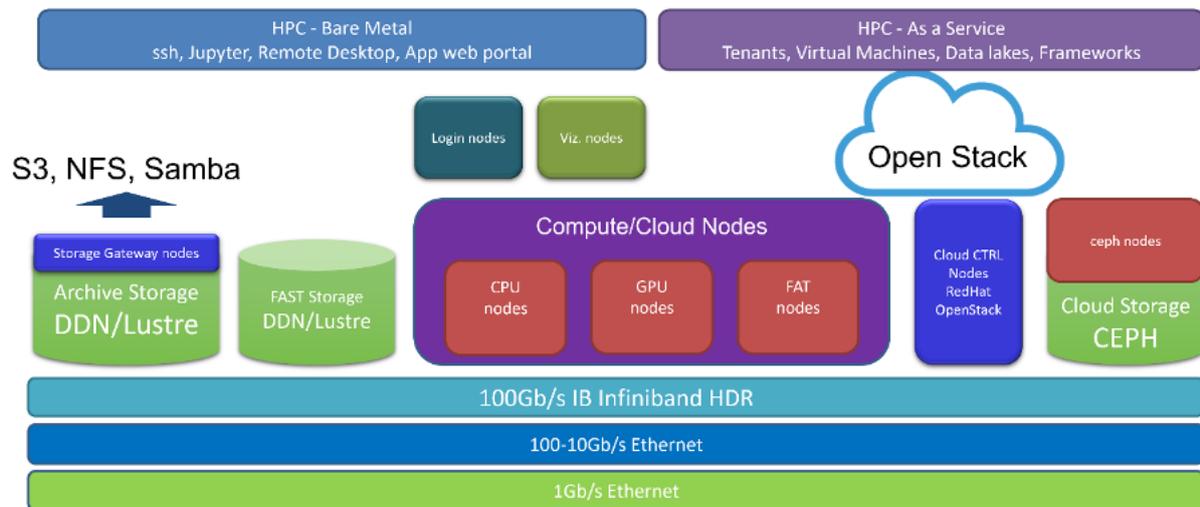


Figure 3: Davinci-1 HPC infrastructure schema.

For benchmark execution, three different kinds of virtual machines are deployed: metaclient, clients and servers. While the metaclient has no specific hardware requirements, clients and servers need some computational resources. Each of these virtual machines was deployed with 8 VCPUs, 64 GiB of RAM and 200 GiB of disk storage. The block devices used for disk storage on these VMs are provisioned on a *Ceph* [5] storage cluster equipped with a mixture of solid-state drives and rotational hard disks, configured to replicate the data 3 times across its 4 servers, and interconnected to the compute nodes through the 10 Gbps Ethernet network.

5. Empirical testing

5.1. The AyraDB Benchmarking Backend

The AyraDB Benchmarking Backend (ABB) is a system designed to enable a simple and straightforward benchmarking procedure for AyraDB clusters of arbitrary configuration and size. ABB is designed for simplicity and easy usage, as in any case a single command is needed to benchmark any configuration of AyraDB.

Figure 4 shows the architecture of the benchmarking system. It is a hierarchical three-level system, where the top level (Benchmarking backend) is the orchestrator of the benchmarking activity. The machine where the benchmarking backend runs is conventionally named metaclient. The benchmarking backend, receives as input the specifications of the requested benchmark, and implements it by sending commands to the clients, in the client subsystem. The clients, in turn, translate the commands from the metaclient into a workload for the servers of the AyraDB cluster. The system is design to work with different hardware configurations, including a multi-site distributed infrastructure. In this benchmarking effort, we have focused on a single-site infrastructure, where the network latency is minimum and negligible with respect to the database latency. This facilitates the interpretation of results, by isolating the performance of the database from other infrastructural constraints.

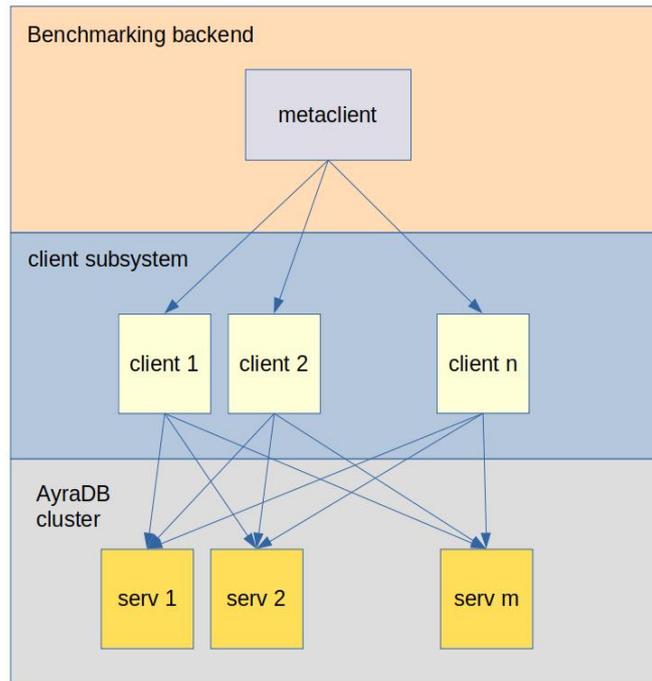


Figure 4: The architecture of the AyraDB Benchmarking Backend

5.1.1. Configuration of a benchmark run

A benchmark run is configured with the following parameters:

- M : the number of servers (in the AyraDB cluster).
- B (GByte): the size of the table, per server. With one server, the size of the table is B GByte, with 2 servers, the size of the table is $2B$ Gbyte and in general with M servers the size of the table is MB Gbyte.
- L (byte): the size of the table's records. The total number of records of the table is $MB \times 10^9 / L$.
- C : the number of clients (in the client subsystem). The number of clients is determined with the simple rule of 1 client every three servers. Therefore, with 1, 2, and 3 servers, we use 1 client. With 4, 5, and 6 servers, we use 2 clients, and so on.
- The type of operation:
 - a. read: with the read operation, a record (or a selection of fields of the record) is read from the table and sent to the requesting client. Records are selected randomly and uniformly.
 - b. update: with the update operation, a record (or a selection of fields of the record) is modified in the table, according to the values provided by the requesting client. Records are selected randomly and uniformly.
- The addressed fields of the record (both for read and update)
- N : the number of operations (read/update) per connection per round (see section 5.1.2).
- P : the pipeline size of read/update operations.
- R : replication factor.
- T : table type (fixed_length, padded, nosql)

5.1.2. Execution of a benchmark run

A benchmark run has two phases:

- Phase 1: initial table loading: in this phase the clients load the table on the servers, by creating records and loading each record on the AyraDB cluster.
- Phase 2: measurement of the maximum throughput of read/update. This phase is executed as a series of rounds, where in the i th round a total of iM connections from the clients to the AyraDB servers is set up. On each connection, N read or update operations are executed and the throughput of each connection is registered. When all the connections have executed N operations, the total throughput is calculated. As the number of connections per server increases, throughput grows, until the maximum throughput of the AyraDB cluster is reached. When the measured throughput becomes stationary, the benchmark run is stopped and the maximum throughput is stored. At round i , each server will have i connections from the clients.

The time required to each connection to complete the requested N operations can vary slightly, therefore, if the throughput of each connection is just added up to calculate the total throughput, the result may have an optimistic bias, due to the fact that there may be periods where not all the iM connections are active (some connections will have finished their job earlier than other connections) and, as a consequence, the throughput of some connections would be larger. In fact, with a smaller number of active connections, the throughput of the individual connection is larger.

In order to avoid this possible bias, each connection produces a time series of throughput measurements, formatted as a list of (t_i, ρ_i) pairs, where t is the absolute time and ρ is throughput. In particular, ρ_i is the throughput of the connection in the time interval $[t_i, t_{i+1}]$. At the end of one round, the metaclient will receive from the clients iM time series of throughput, $(t_{i,j}, \rho_{i,j})$, where j is the connection index, ranging from 0 to $iM-1$. Since the clients are synchronized with the Network Time Protocol (NTP) the time values in the time series are consistent. The metaclient determines the time interval $[t_s, t_e]$, where all the iM connections are active. The value of t_s is easily calculated as

$$t_s = \max_{j \in [0, iM]}(t_{0,j})$$

For the calculation of t_e , the last time value of each series, referred to as $t_{i,j}$ is considered:

$$t_e = \min_{j \in [0, iM]}(t_{i,j})$$

The metaclient does not consider the throughput statistics outside the time interval $[t_s, t_e]$, therefore, it is guaranteed that the throughput statistics associated to a group of iM connections are calculated on data collected when exactly iM connection are actually active.

5.2. Benchmarking results

For the benchmarking activity carried out so far we have used, for both clients and servers, the following machines:

- 8 cpus
- 32 Gbyte RAM
- 256 Gbyte SSD

The parameters of the benchmarks are:

- M : the number of servers (in the AyraDB cluster), ranging from 1 to 13.
- B (GByte): the size of the table, per server. We have used 10, 20, 30, and 40 GByte/server.
- L (byte): the size of the table's records. We have used 1500 byte.
- C : the number of clients (in the client subsystem), a function of the number of servers, as specified in section 5.1.1
- The type of operation: read/update.
- The addressed fields of the record: one field of 100 byte
- N : the number of operations (read/update) per connection per round: 1 million.
- P : pipeline size, equal to 16.
- R : replication factor, equal to 1.
- T : table type, fixed_length.

Figure 5 shows the throughput of the read operation as a function of the number of connections, for an AyraDB cluster of 1 server, with table size equal to 20 GByte. The throughput grows steadily as

the number of connections increases, until it reaches the maximum system's throughput, where it saturates.

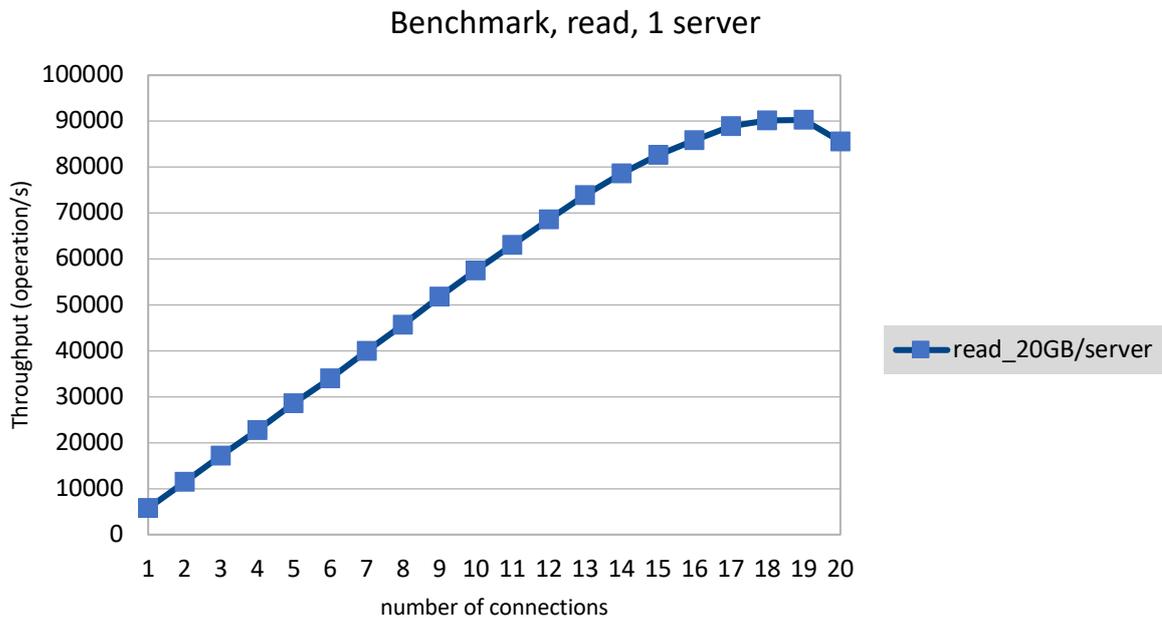


Figure 5: Measured throughput of read operation, 1 server, table size 20 GByte/server

The same behavior, on a different scale, is shown in Figure 6, reporting the measured throughput of the read operation as a function of the number of connections, for an AyraDB cluster of 13 servers. Qualitatively, the behaviors shown in the two Figures are quite similar, however, with 13 servers throughput exceeds 1 million read/s.

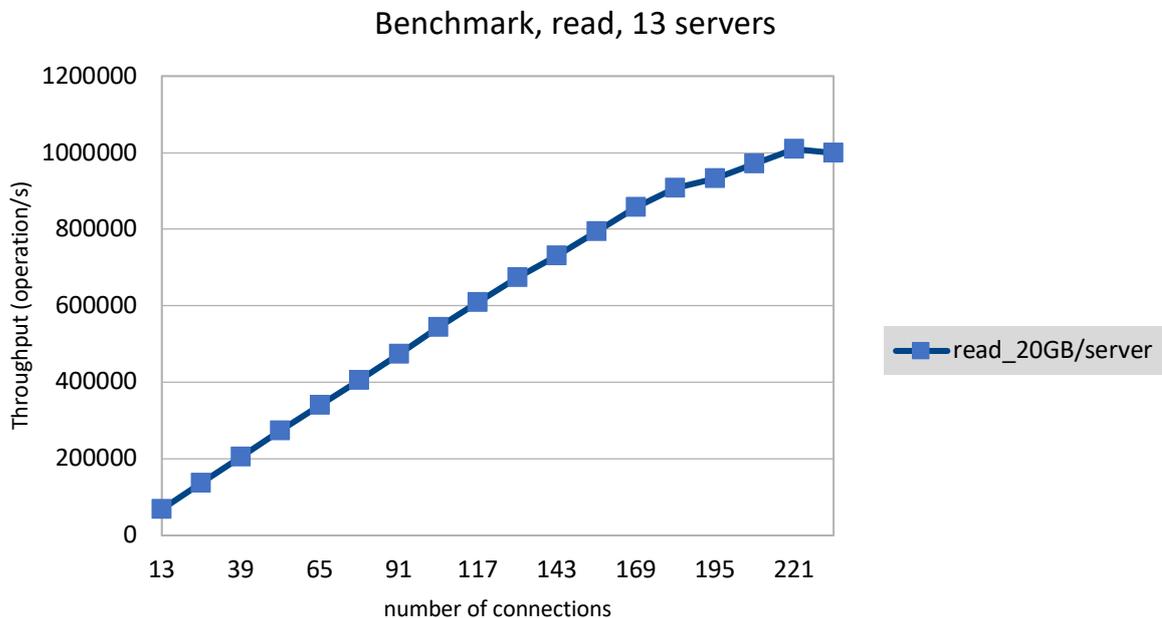


Figure 6: Measured throughput of read operation, 13 server, table size 20 GByte/server.

Figure 7 shows the throughput of the update operation as a function of the number of connections, for an AyraDB cluster of 1 server, with table size equal to 20 GByte. Similar to read operations, the throughput with update operations grows steadily as the number of connections increases, until it

reaches the maximum system's throughput, where it saturates. Similar results are obtained with 13 servers, as shown in Figure 8.

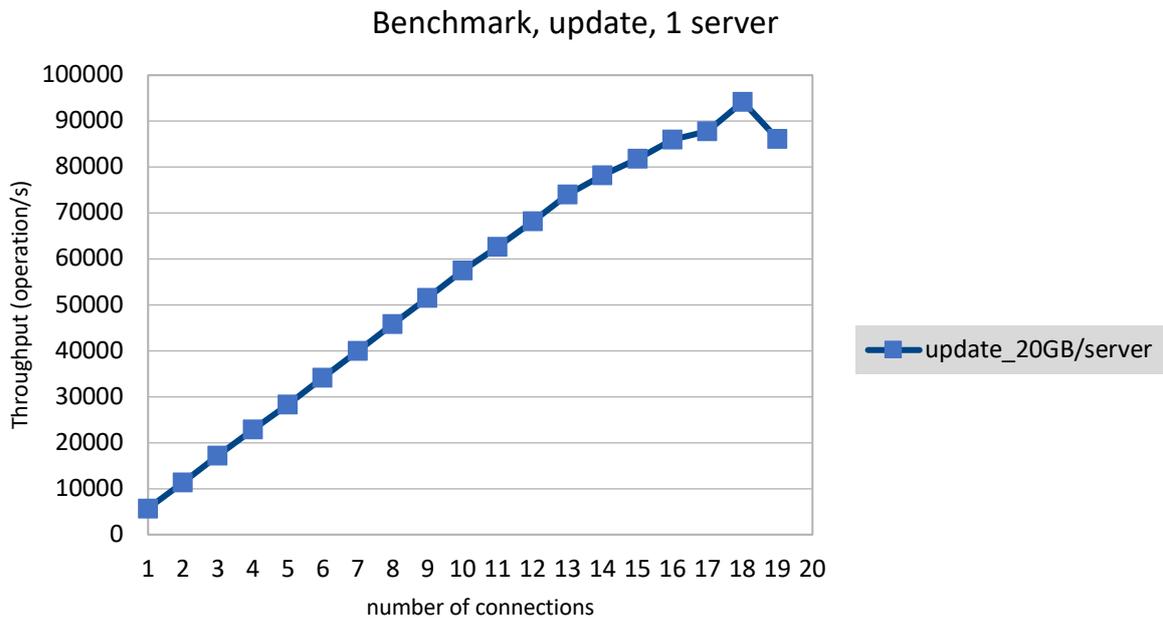


Figure 7: Measured throughput of read operation, 13 server, table size 20 GByte/server.

The scalability of throughput as a function of the cluster size is shown in Figure 9, reporting the maximum throughput as a function of the number of servers. The AyraDB cluster size ranges from 1 to 13 servers, and the table size ranges from 10 GByte/server to 40 GByte/server. Therefore, with 13 servers and 40 GByte/server the absolute table size is 520 GByte. The figure includes both read and update benchmarks, for all the considered cluster sizes and table sizes.

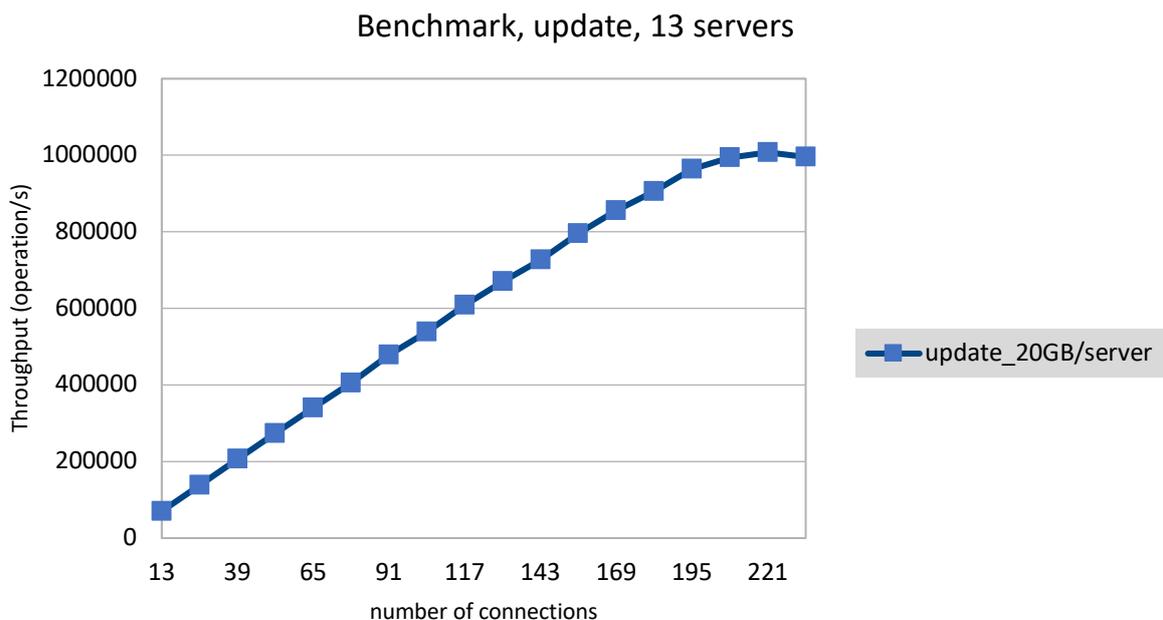


Figure 8: Measured throughput of read operation, 13 server, table size 20 GByte/server.

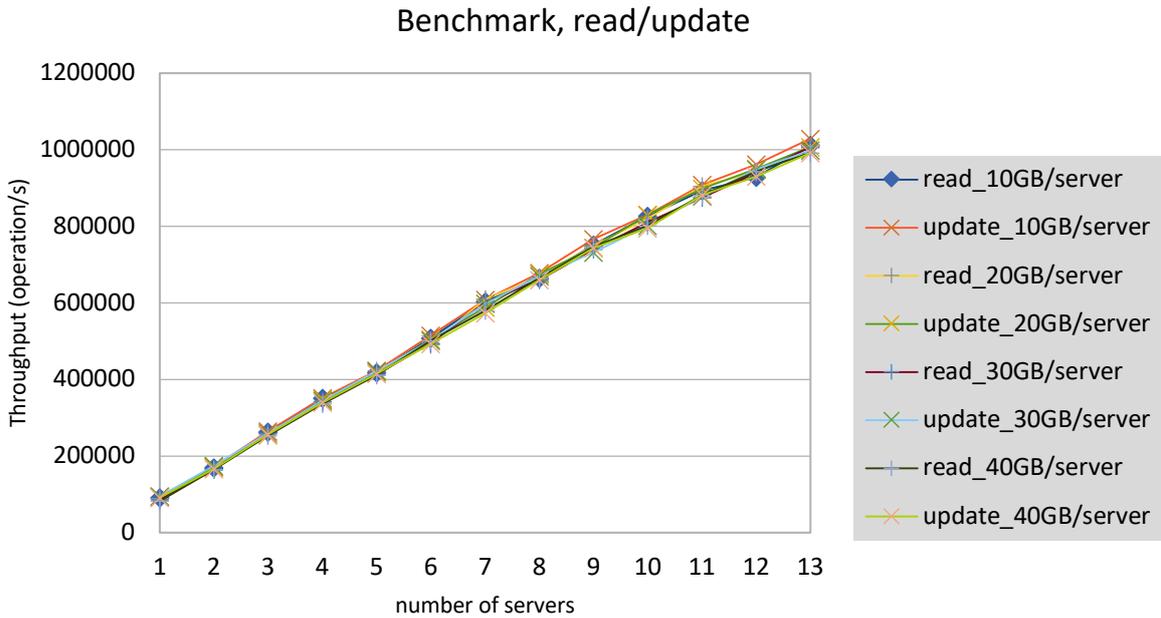


Figure 9: Maximum throughput of read/update operation, 1 to 13 servers, server, table size equal to 10 GByte/server, 20 GByte/server, 30 GByte/server, 40 GByte/server.

Figure 9 shows that AyraDB has a good linear scalability of throughput, as the cluster size grows. Moreover, the throughput of the read and update operations are quite similar. Finally, table size does not seem to impact significantly of the throughput performance.

6. Discussion and conclusions

Results show how AyraDB reaches 1 million operations/s with 13 nodes. As noted before, both InfiniBand and 10 Gbps Ethernet connections are available for node communication in Davinci-1, but for this paper’s benchmark only Ethernet was used. Future work will benchmark AyraDB with InfiniBand, hopefully breaking the 1 million operations/s limit. However, comparing the current result with the state of art suggests that AyraDB has linear scalability and per-core throughput that, to the best of our knowledge, is 5 times greater than the best available benchmark.

Obtaining scalability at a fraction of the cost unlocks business benefits in a number of use cases. It reduces the barriers to entry for SMEs in data-intensive businesses, it enables new business models based on the exploitation of large datasets, it helps ensure the success of independent players in the data market and it is environmentally sustainable. In our professional activity, we have found that most companies experience technical scalability issues with their database and struggle to accommodate big data requirements. We have explored these issues and thoroughly understood business needs. We have responded to these needs by designing AyraDB, a next-generation database that has demonstrated to be far less expensive and more scalable than market average.

There is a strong market need for efficient big data technologies. We have repeatedly observed hardware scalability issues with most of our clients (including over 20 large corporations). Scalability issues seem to be so pervasive that we have started looking for a cause of scalability issues that could be general and common to all our clients. We believe that the slow down of Moore’s law is a root cause of current scalability issues. In the past, growing storage and processing capacity demands were satisfied by continuous hardware innovation that was guaranteed by Moore’s law. In fact, Moore’s law

states that the ratio of processing capacity to cost doubles every 18 months: Moore's law has been published in 1965 and has proved accurate for several decades. Lately, the semiconductor business has slowed down its innovation pace. Admittedly, Intel stated in 2015 that the pace of advancement has slowed. They have also stated that hyperscaling will continue to guarantee scalability. Hyperscaling is the ability to seamlessly provision and add computing, memory, networking, and storage resources to a given computer or set of computers to meet growing capacity requirements. Hyperscaling is at the basis of cloud computing, that is in fact presented as scalable since computing resources can be added as needed with virtually no limits. However, there is a fundamental difference between Moore's law and hyperscaling. While the first guarantees that unit capacity costs decrease over time, the second can provide capacity by adding new resources with the same unit costs. In other words, hyperscaling can guarantee technical scalability, but does not guarantee economic scalability and environmental sustainability. With hyperscaling and, thus, cloud computing, if data grow in size, costs grow accordingly. Clearly, this is a threat for many industries, particularly for the space economy, as it creates a growing issue of economic scalability.

With this study, we have taken a first step towards building a scalable infrastructure that can take full advantage of hyperscaling, with linear scalability and no bottlenecks due to in-memory caching. Empirical results show how AyraDB's thrifty hardware consumption through greater performance translates into lower costs and, thus, increases economic scalability. Tests with a replication factor $R > 1$ are ongoing work. Further work is needed to complete the tests by:

- Using Infiniband and an increasing number of servers,
- Testing AyraDB with a growing dataset size,
- Perform a comparison with the different types of tables supported by AyraDB (fixed length or SQL, padded and noSQL),
- Perform an extensive AyraDB benchmarking on bare metal, that is, with non-virtualized architectures, in comparison with the virtualized architectures used in the benchmarks performed so far,
- Build a "data processing pipeline" for key applicability areas with stringent performance requirements, among the applications relating to the analysis of geographic and satellite data.

7. Acknowledgements

This work was funded by the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 951732. This work expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this work.

8. References

- [1] <https://scylladb.medium.com/cockroachdb-vs-scylla-benchmark-b0215e81144>.
- [2] Hippj project, <https://cordis.europa.eu/project/id/867276>
- [3] IDC "Worldwide semiannual big data and analytics spending guide", 2021, <https://www.idc.com/getdoc.jsp?containerId=US47485920>
- [4] <https://www.openstack.org/>
- [5] <https://ceph.io/>
- [6] Boudriki Semlali, BE., El Amrani, C. (2021). Satellite Big Data Ingestion for Environmentally Sustainable Development. In: Ben Ahmed, M., Mellouli, S., Braganca, L., Anouar Abdelhakim, B., Bernadetta, K.A. (eds) Emerging Trends in ICT for Sustainable Development. Advances in Science, Technology & Innovation. Springer, Cham. https://doi.org/10.1007/978-3-030-53440-0_29
- [7] www.ayradb.com

- [8] Fuad Bajaber, Sherif Sakr, Omar Batarfi, Abdulrahman Altalhi, Ahmed Barnawi, Benchmarking big data systems: A survey, Computer Communications, Volume 149, 2020, pp. 241-251, ISSN 0140-3664
- [9] Couchbase performance benchmarks, <https://www.couchbase.com/benchmarks>, 2022
- [10] Scylladb performance benchmarks, <https://www.scylladb.com/product/benchmarks/aws-i2-8xlarge-benchmark/#~:text=ScyllaDB%20on%20AWS%20i2,-8xlarge%20Benchmark&text=Benchmarking%20database%20systems%20helps%20users.of%20wide%20column%20NoSQL%20databases>.
- [11] Ahmet Ercan Topcu, Aimen Mukhtar Rmis, Analysis and evaluation of the riak cluster environment in distributed databases, Computer Standards & Interfaces, Volume 72, 2020, ISSN 0920-5489, <https://doi.org/10.1016/j.csi.2020.103452>.