

Analysis of Biological Sequence Search Performance in NoSQL Database

Quezia N. Flach

Federal Institute Farroupilha
Alegrete - Brazil

Arthur F. Lorenzon

Federal University of Pampa
Alegrete - Brazil

Marcelo C. Luizelli

Federal University of Pampa
Alegrete - Brazil

Fabio D. Rossi

Federal Institute Farroupilha
Alegrete - Brazil

ABSTRACT

Data generated by research involving genomic data needs high processing power due to the large volume of data generated by these surveys. The development of computational tools softened the impacts of a large amount of data caused by such research. Still, the processing and storage of genomic data is a challenge. In this work, we intend to evaluate the performance of a distributed NoSQL database and possibly present a more feasible performance solution by analyzing the behavior of the NoSQL DynamoDB database when carrying out a genomic search. The results showed that NoSQL databases have superior scalability and performance to relational databases, and perform very closely with high-performance applications over multiprocessing environments. NoSQL databases consist of a new model that gives storage for large volumes of data and processing capacities far superior to relational databases due to the arrangement of data coming from big data and data science area. The use of NoSQL approaches over distributed data makes them more flexible concerning performance, as they can grow infinitely above resources according to the database demand. It makes them an ideal tool to perform the metagenomic search.

General Terms

Algorithm, Database.

Keywords

BLAST, NCBI, Sequence alignment, Metagenomics, Taxonomy assignment

1. INTRODUCTION

Over the years, computing has proved to be an essential tool for human development. This advancement has facilitated the study and development of several areas of knowledge, such as bioinformatics ([16]). Bioinformatics is an innovative and relatively recent science. It makes use of machines with high-performance processors that aim to understand and store research on biological material. Such advances in nucleotide sequencing known as NGS (a new generation of sequencing technology) have attached great importance to research in the field of genetics, and biology ([22]).

Metagenomics is a field of bioinformatics that studies DNA samples drawn from a given environment. The collected genetic material contains information about the genes that make it up and its contributions to the development and functioning of organisms in this environment.

The dynamics of data analysis using this field of bioinformatics is characterized by the extraction of a large volume of data. For such information to be analyzed, it does not require the storage and processing of such data. Therefore, the use of a NoSQL database ([18]) is a useful option for the processing and storage of genomic data. The term NoSQL came up to meet the new paradigms of computing like Big Data and Data Science and becoming a solution to the problems of processing, storing, and managing large volumes of unstructured data.

NoSQL databases have a distributed, fault-tolerant structure, based on splitting the data across multiple servers, reducing the risk of data loss or failure. Because they are designed to work with large amounts of data, the NoSQL Banks present a flexible framework model that does not require a fully structured model, making it easier to access stored data. Another essential feature is the ability to scale data, as there is a greater need for performance ([13]).

The study of Metagenomics, in conjunction with Bioinformatics, made it possible to genetically sequence any living organism in a short time, giving rise to a wide range of genomic sequences. Considering that in a single sequencing thousands of data pairs are extracted, the storage of these requires high processing power. That is why the area of information technology comes as a solution in the process of high-performance automation, storing, and analyzing such data ([14]).

For this reason, interest arose in developing a new approach using the NoSQL DynamoDB Database to store and query metagenomic data. The goal of this project is to analyze the performance of the NoSQL DynamoDB tool when performing a metagenomic search. This study will conduct tests on the performance of the DynamoDB tool, evaluating its performance when playing a metagenomic search, and analyze its processing time when implementing such a task.

This work hypothesizes that NoSQL databases available in cloud environments can offer very similar performance to the current tools for searching for biological sequences, without the cost of a high-performance infrastructure. It would make such

environments very attractive to future research in this area as it allows the use of virtually infinite resources.

This paper examined the performance regarding the execution time of a NoSQL database for genomic search compared to a relational database and compared it to two implementations developed in MPI. The paper is divided as follows: Section 2 presents preliminary experiments and our hypothesis; Section 3 presents some works related to this proposal; Section 4 presents the methodology for implementing the NoSQL database search; Section 5 shows a discussion about the results; Section 6 presents our conclusions and future work.

2. BACKGROUND

With the advent of technologies like data mining, several organizations have been putting a considerable effort in collecting and analyzing large amounts of data to get a commercial advantage ([5] [20]). Besides, the advances in areas like engineering and telecommunications allow the development of storage devices with higher capacity at lower costs. Moreover, discoveries in networking allow performing data manipulation through the network at higher transfer rates.

Given these questions, areas like Big Data ([6]) and Internet of Things [8] that require the processing of large amounts of data have been more and more discussed, which have been raising the demand for mechanisms to process large amounts of structured, semistructured, and unstructured data ([15] [21]).

With this constant demand for processing large amounts of data, some limitations of relational databases become even more visible. For example, to scale a relational database, users have to invest in more powerful – and expensive– computers, since this kind of database cannot be easily distributed since joining tables and partitioning data across multiple instances demand a significant effort. Besides, users also have to convert all data into a table to put it into relational databases. In this context, non-relational databases (also known as NoSQL databases) arise with the proposal of handling unstructured data such as emails and multimedia files efficiently ([9]). Unlike relational databases that are table-based, NoSQL solutions can organize data through different approaches:

- Key-Value:** it allows users to manipulate data by using keys that correspond to data that can be contained in several file formats such as Strings, JSON (JavaScript Object Notation), BLOB (Binary Large Object), etc. This type of database also employs cache mechanisms in data mapping to improve database performance.
- Document Store:** it puts data in key-value pairs and compresses it in documents that contain attribute metadata associated with stored content.
- Graph-Based:** it uses the graphical representation to address scalability issues present in relational databases. Graph-based NoSQL databases use graph structures with edges and nodes to represent and store data. Each node may have relationships (represented by edges) with other nodes. Besides, both nodes and edges have their own properties.
- Column Store:** it groups data into columns instead of rows like relational databases. Using this approach, it is possible to put heterogeneous data in the same table since one entry does not necessarily need to have the same attributes as the other entries. Besides, each row can be stored in different storage devices, which facilitates the use of parallel architectures.

NoSQL databases bring several benefits over relational databases, such as i) scalability, since we can take advantage of parallel archi-

tures which in its turn allows it to be used to store large amounts of data; ii) flexibility, since NoSQL databases enable users to handle diverse types of unstructured data such as multimedia assets and word-processing documents.

However, studies show that not all NoSQL databases perform better than relational databases ([10]). Thus, as the best database option –either SQL or NoSQL– may vary depending on the workload and the business demands. There is a concern about evaluating different types of databases under various scenarios that demand fast processing of vast amounts of data. In this sense, this paper compares the performance SQL and NoSQL database for metagenomic search, which is a required field of Bioinformatics.

3. PRELIMINARY EXPERIMENTS

Preliminary tests were conducted to evidence the differences between writing and reading over SQL and NoSQL databases. To this end, a well-known performance benchmark ¹ for databases was used. This tool allows the insertion of a large amount of randomly generated data and makes it possible to read the records based on their keys. The tests were performed using MySQL 5.1.73 ([2]) and DynamoDB 2.0.0 ([17]), on the server with the following characteristics: Linux operating system Ubuntu 14.02, Intel Core i5-3210 of 2.50GHz, 8GB of RAM.

The benchmark was configured to perform 500 insertions per query using BTree, and the search was conducted on 10000 records with 100% randomized. Although several works show that NoSQL databases always perform better than SQL databases, in Figure 1, one can see that this is not always true. DynamoDB reads records about three times faster than MySQL, but still loses performance in terms of database writing.

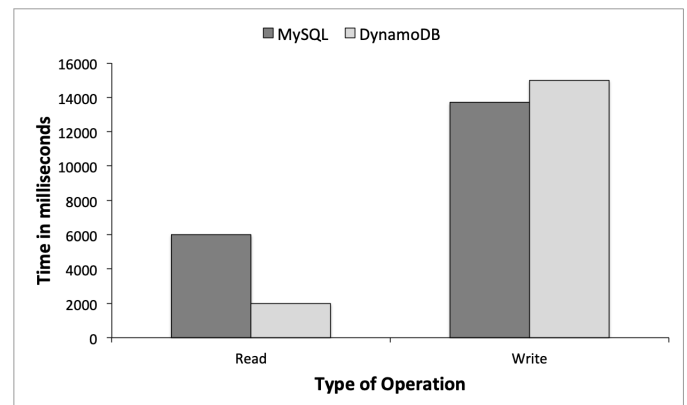


Fig. 1. Comparison of reading and writing of the same application, on different types of databases. MySQL as a relational database option, and DynamoDB as a NoSQL database option.

These results reinforce the hypothesis of this work: the search for metagenomic data using NoSQL databases is a better alternative in terms of performance than on traditional SQL databases. SQL databases are table-based, whereas NoSQL databases can be a document-based, key-value, graph, or column-oriented. It makes relational SQL databases better options for applications that require transactions returning multiple columns or legacy systems created in a relational structure. However, NoSQL databases

¹<https://github.com/STSSoft/DatabaseBenchmark>

are an excellent choice for fast-growing businesses or databases without clear schema definitions, as is often the case with mobile apps, real-time analytics, and content management systems.

4. RELATED WORK

The Basic Local Alignment Search Tool (BLAST)² ([1]) was developed to perform searches comparing primary biological sequences against a database containing a massive amount of information, returning more similar sequences and statistical significance of the search. The BLAST tool is a very used tool within bioinformatics because it is a quick tool, and there are several variations of BLAST, each with a specific objective. This set of program variations of the BLAST package can be called the BLAST family. BLAST programs are hosted at the National Center for Biotechnology Information (NCBI)³ and are linked to the NCBI database. Searches performed with BLAST through the NCBI web page will search for similar sequences in that database. The BLAST program performs alignments between DNA sequences or proteins called query against the sequences deposited in the database, and these sequences are called the subject. Most works that explore performance in genomic search use NCBI database and parallel programming interfaces such as MPI (Message Processing Interface) ([23]) or new architectures, such as the use of CUDA over GPUs ([4]). Among the implementations for a high-performance cluster, we can mention mpiBLAST and MPI-blastn.

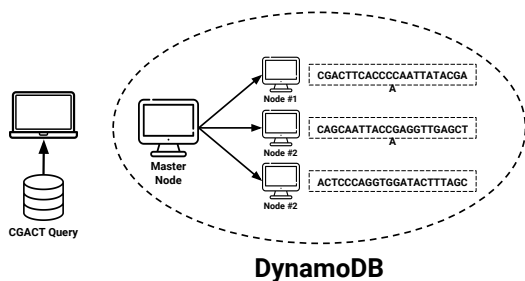


Fig. 2. Metagenomic NoSQL search architecture. The NCBI database is distributed among several processing nodes, so that the search based on fragments is carried out in parallel, optimizing the total performance of the search.

[12] developed mpiBLAST, which consists of a model that improves genomic search performance based on a computationally intensive sequence alignment process, as more machines are added to the cluster. The mpiBLAST parallelization strategy is based on partitioning the database entry into many fragments, as many as the number of computers on which the application will run. Developed by [7] MPI-blastn was designed to solve the performance problem in sequence processing. Using this tool, it was possible to reduce the search time in a 100,000 line file, which would take about 72 hours to complete, to less than 5 minutes. A significant reduction, taking into account the large volume of data that this file has.

Replication of data in Bioinformatics Workflows using NoSQL database developed by [11], presents a study on the impact of the replication of these Workflows, comparing the results of two

²<https://blast.ncbi.nlm.nih.gov/Blast.cgi>
³<https://www.ncbi.nlm.nih.gov>

NoSQL, Cassandra, and MongoDB databases. Regarding MongoDB, Cassandra provided operations to read these files in less time. The differential of this article is to evaluate the scalability and performance of a NoSQL database against SQL database and high-performance applications on multiprocessing architectures. To the best of our knowledge, no work performs this type of evaluation.

5. METHODOLOGY

In this section, the tools used to configure the database will be presented, and also the methodologies used to achieve the expected final result will be given. The objective is to allow the reader to have an initial vision regarding the concepts of each fundamental element in the development of this project.

5.1 Tools

Metagenomics is a technique that allows studying the genes of microorganisms of a specific environment, without having to realize individual cultures. The genetic material collected in a metagenomic study serves to identify functional genes or new metabolic cultures to evaluate microbial diversity and to analyze the action of the entire population in an ecological niche. Their research enables genome structures of a non-cultured organism, and using sequencing technologies, define biomarkers used to classify the behavior and type of process occurring in specific environments ([14]).

The internationally used genomic information base consists of the NCBI (National Center for Biotechnology Information) launched in the late 1980s by the US government, consisting of the first public database containing DNA sequences of the most diverse organisms. It is one of the largest repositories of biological information available, which has a series of tools and resources to aid scientific research in genetics. The NCBI has extensive genetic database fed daily with new sequences. The data generated by these sequences are stored in Fasta format, as shown in Figure 3, text-based to represent sequences of DNA, RNA, or nucleotide sequences.

```
>gi|1708|emb|X00640.1| Rabbit 18S ribosomal RNA sequence
TACCTGGTTGATCCTGCCAGTAGCATATGCTTCTCAAAGATTAAAGCATGCATGCTTAAGTACGCACGCCGGTACAG
TGAAGTGGCAATGGCTTAATAATCAGTTATGGTCTTGTGGTCCCTCCTCTCTCACTTGGATAACTGTGGTAAAT
CTAGAGCTAAATACATCCGCTGGGGCTGTCCCTTTGTGGGGATCCGCTTTATCAGATCAAAACCAACCCCGCT
CAGTTTCCCCCGCCCGCCGNNNGGTGGGGCTGGACTGCTTTGGTACTAGATAAATCGGGCCGATCGCAGNC
CCGGTGGGCAATCGAACGTATGCCCATCAACTTCGATGGAGTCCGGCTACTCATGGTGAACCGGGTGGAGC
GGATCAGGGTTCGATTCGGAGAGGGAGCTGAGAAACGGCCACACATCAAAGGAGGAGAGCGCCGCAAAATCAAC
CACTCCCGACCCGGGGAGGTAGGACGAAAACAACAACAGACTCTTTCGAGGCCCTGCATTTGGAAATGAGTACAT
TTCAAATCTTCAAGCAGGATCCATTTGGAGGGCAAGTCTGGTGGCAGCACGCCGGTAAATTCAGCTCAACAGCGACA
TCAAAGTTGCTGCAAGTAAAAGCTCGTAGTTGGATCTTGGAGGGTGGTGGGGGGTCCCGCGAGGCGGACCA
CCGCCGCTCCCGCCCTTGCCTTCGGGCCCCCTCGATGCTTCTAGCTGAGTGTCCGGGGGCCCGCAAGGTTTACT
TTGAAAAATTAGAGTGTCAAAGCAGGCCAGGGCGCTAGATACCGAGGCAGGAAATAGTGAACAGGACCCGGGTC
TATTTTGGTTTTTCGAACTGAGGCGATTAAGAGGAGACGCCGGGGCACTTCTGATTCGCCCTGATAGAGTGA
TTCTTGAGCGGGCAGAGCGGACAGAGCAAGCAATTTGCAAGAATTTTTCATTAATCAAGAACCAAGTGGAGG
TTGAAAGCAGTACAGATCCGTCGTAGTTCGACACAAACAGTCCGACTGGCGATGGCGGGGTTATTCCTATGCC
CGACCGGAACCTTCGGAAACCAAGTCTTGGGTTCCGGGGGAGTATGGTTGCAAAAGTGAACCTTAAGAAATTA
CGGAGGGCCACCAAGGATGGATCTTGGGGCCATTTTACTCAACAGGGAAACTCACCCGGCCGACACGGACA
GGATTGACAGATTGATGCTCTTCTCGATTGTGGCTGGTGGTGCATGGCTTCTAGTGTGGAGGCAATTTGTCT
GGTTAATTCGATAACGAACCTAGACTCTCGATGCTAAGTACTGACGACCCGAGCGGGTCCGCCCAACTTTT
CAGAGGCAAGTGGGTTTCAGCACAGATTTAGCAATAAGAGTCTGTGATGCCCTTAGATGTCGGTGGCAGCGCC
GCTACACTGACTGGCTCAGCGTGTGCTCACCTACCGCCGAGCGGGTAAACCCGTTGAAACCCATTCGTGATAGGATC
GGGATGCAATTTTCCATGAACGAGGAATCCAGTAAAGTGGGGTCAAGCTTGGCTGATTAAGTCCCTGCC
TTTGTACACACCGCCGCTCACTACCGATTTGATGGTTAGTGGCCCTCGGATCGCCCGCGGGGTCCGCCACGG
CTGTGGCGGAGCGCTGAGAAAGCGGTGAACTGACTATAGAGGAAGTAAAGTCTGAACAGGTTCTCGTAGTGA
CCTGGGAAGGATCATT
```

Fig. 3. NCBI file structure. For each existing organism, a line with all the associated genetic code is maintained in this file. The input fragments search for compatible sections, line by line.

This information from the Fasta file has been rearranged into a JSON ([19]) file, input format for most NoSQL databases (as can be seen in Figure 4). The chosen NoSQL database was the DynamoDB ([3]) in its cloud environment version. DynamoDB is a non-relational database that delivers reliable performance on any scale.

Figure 2 presents the metagenomic search process proposed in this work. The search occurs in the same way as the MPI implementations. The NCBI base is distributed among all processors. The user

submits a query (consisting of a genetic sequence fragment), and this query is sent to each node so that it checks for the existence of an entire sequence in which the fragment exists.

In the case of existence, the name of the living being corresponding to the sequence is displayed as one of the possible living beings related to the search. In the experiments performed, the search query counted on 30,000 and 100,000 genomic sequence fragments (lines), which should find the corresponding living beings. This entry for the search is the same as that used in the mpiBLAST and MPI-blastn experiments, allowing a comparison parameter.

5.2 Implementation

The tests were performed on a Dell PowerEdge M630 server (2 x 2.4 GHz Intel Xeon E5-2620 v3 processors (total of 12 cores) and 32 GB of RAM each). The relational database management system chosen for the experiments was MySQL. MPI applications run on a multicore bypass without the need for external network communication. The non-relational database selected was DynamoDB. The MySQL and DynamoDB databases have been populated with the NCBI (National Center for Biotechnology Information) database, which has more than 200 million genetic sequences (one per line), consisting of the genetic codes of the majority of current living beings.

For such evaluations, it is inserted into a table called Genome, and two attributes called String and Name. In the string attribute, all the genetic strings of the NCBI database were stored, and in the Name attribute, the name of the living entity related to the string was stored. A fragment of the data.json file used for the insertion of the data in DynamoDB can be seen in Figure 4, and that served to populate the attributes of the Genome table. Several items are replicated in JSON and inserted row by row by the batch-write-item parameter.

```
{
  TableName: "Genoma",
  Item: {
    "Cadeia": "GATCCCATCAACTCTACTACTACTA",
    "Nome": "Macaco",
  }
}
```

Fig. 4. Insertion JSON example. NoSQL generally uses JSON to represent data in the form of records. This allows flexibility, since it is possible to add new attributes, exclude others that are no longer used, without worrying about changing the entire database.

After all the data loaded in the Genome table, the search is performed in the same way, using the CLI. In this example, the search is conducted in the Genome table, in the String attribute, by the string located in the search.json file.

```
dynamodb query --table-name Genoma
--projection-expression "Cadeia"
--expression-attribute-values
file://busca.json
```

In this case, the searched string consists of the GTAC fragment, as shown in Figure 5.

The database returns as a response, the list of genetic strings that contain the fragment, and the respective living name corresponding to that fragment. Because the genomic search uses as input only fragments, several responses can be generated for the same search.

```
{
  "id": {"S": "GTAC"},
}
```

Fig. 5. Searching JSON Example. This is an example of a fragment that must be searched for in the NCBI database. One may notice that the search may offer several answers, given that a fragment may fit into several genetic codes.

6. DISCUSSION

The tests feature performance scalability between one and twelve processors. The results of the tests with a search entry of 30,000 metagenomic fragments can be seen in Figure 6.

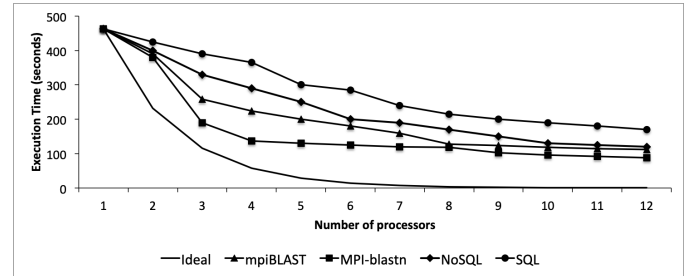


Fig. 6. Searching 30,000 metagenomic fragments against NCBI database.

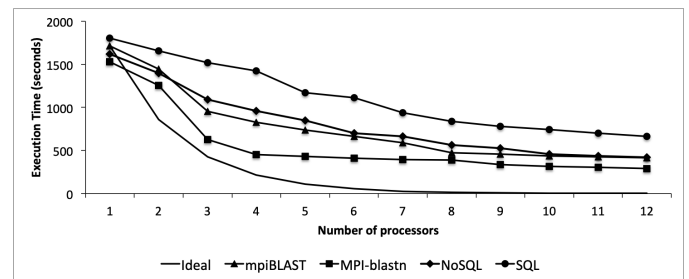


Fig. 7. Searching 100,000 metagenomic fragments against NCBI database

None of the implementations can match the optimal theoretical performance due to resource sharing. However, the two implementations in MPI show the best performance. This result is expected since MPI implementations are optimized for high-performance computing. In particular, MPIblastn provides further optimizations in data preprocessing. It is interesting to note that the behavior of the search on the NoSQL database.

Although it is better than searching in the relational database, the results show that as the number of processors increases, the results are closer to the mpiBLAST implementation. The Figure 7 shows that when the size of the search input is increased to 100,000 metagenomic fragments, the results presented by the NoSQL database are even better, reaching the same search time of mpiBLAST with the use of ten, eleven and twelve processors. Perhaps on a more powerful server, searching for the NoSQL database may perform better than the mpiBLAST implementation.

From the standpoint of executing instructions, once the SQL statement is received, whatever the database does, it performs several steps that include error checking in the syntax of the statement,

analysis of which is the best way to access the data (by index, table structure, using statistics) and other details. Thus, the execution time of an SQL statement involves several factors that include several internal aspects of each database's operation. These aspects may have influenced the positive results of the NoSQL database search.

NoSQL is much more flexible and scalable since data can be stored without a predefined schema, and development is faster, and the integration code is more reliable. This translates into quicker and quicker real-time application changes. Besides, most SQL databases have characteristics that only allow vertical scalability, since distributing those databases would require joining tables across servers, which is not a trivial task. On the other hand, NoSQL databases take advantage of high parallel infrastructures through characteristics that allow horizontal scalability, making them a suitable solution for scenarios involving the processing of large amounts of data ([9]).

7. CONCLUSION AND FUTURE WORK

SQL databases respect the relational model because they are based on the fact that all data is stored in tables. In this subject, with SQL, being the relational database that always has the ACID properties (Atomicity, Consistency, Isolation, and Durability). And these are four properties that describe the main characteristics of a transaction in SQL. NoSQL databases have very different features. They are non-relational databases, and the properties just mentioned (ACID) are no longer mandatory.

The designation NoSQL was initially used in the late 1990s and was reintroduced at the beginning of 2009. This type of database was created because the relational model has some limitations when data volume is high, and workloads are significant. Thus, unlike the previous ones, they are horizontally scalable and do not generally support joining instructions and operations. In this way, unstructured data (such as articles, photos, social data, videos, or content within a blog post) can be stored in a single document that can be easily found but is not necessarily categorized into fields such as a relational database.

Although NoSQL databases consist of a new paradigm that provides storage for large volumes of data and processing capabilities far superior to relational databases due to the format of data coming from big data and data science environments, their use over distributed environments makes them more flexible regarding performance, as they can grow infinitely over resources depending on demand on the database. It makes them an ideal tool to perform the metagenomic search. The hypothesis tested in this article, which consisted of which NoSQL databases present enough performance for metagenomic searches, proved right. This work showed that for a metagenomic search on the NCBI database, the choice of NoSQL databases is the best option when the other option is a relational database. However, with the analysis of the results obtained so far, the MPI-blastn tool presents better processing time than the NoSQL database. Even so, the NoSQL database has an adequate processing time compared to mpiBLAST, especially to the relational database.

8. REFERENCES

- [1] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.
- [2] David Axmark and Michael Widenius. Mysql introduction. *Linux J.*, 1999(67es), November 1999.
- [3] Simone Brunozzi. Big data and nosql with amazon dynamodb. In *Proceedings of the 2012 Workshop on Management of Big Data Systems*, MBDS '12, pages 41–42, New York, NY, USA, 2012. ACM.
- [4] Ian Buck. Gpu computing with nvidia cuda. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.
- [5] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4), 2012.
- [6] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mobile networks and applications*, 19(2):171–209, 2014.
- [7] R Dias, Miguel G Xavier, Fabio D Rossi, Marcelo Veiga Neves, TAP Lange, A Giongo, César Augusto Fonticieiha De Rose, and EW Triplett. Mpi-blastn and ncbi-taxcollector: improving metagenomic analysis with high performance classification and wide taxonomic attachment. *Journal of bioinformatics and computational biology*, 12(03):1450013, 2014.
- [8] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.
- [9] Neal Leavitt. Will nosql databases live up to their promise? *Computer*, 43(2):12–14, 2010.
- [10] Yishan Li and Sathiamoorthy Manoharan. A performance comparison of sql and nosql databases. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 15–19. IEEE, 2013.
- [11] Iasmini Virgínia Oliveira Lima. Replicação de dados em workflows de bioinformática usando os bancos de dados nosql.
- [12] H. Lin, X. Ma, W. Feng, and N. F. Samatova. Coordinating computation and i/o in massively parallel sequence search. *IEEE Transactions on Parallel and Distributed Systems*, 22(4):529–543, April 2011.
- [13] Bernadette Farias Lóscio, Hélio Rodrigues de OLIVEIRA, and Jonas César de Sousa PONTES. Nosql no desenvolvimento de aplicações web colaborativas. *VIII Simpósio Brasileiro de Sistemas Colaborativos*, 10(1):11, 2011.
- [14] Diana Marco. *Metagenomics: Theory, methods and applications*. Horizon Scientific Press, 2010.
- [15] ABM Moniruzzaman and Syed Akhter Hossain. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*, 2013.
- [16] Jason H. Moore. Bioinformatics. In *Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '07, pages 3435–3457, New York, NY, USA, 2007. ACM.
- [17] Ariel Ortiz. Architecting serverless microservices on the cloud with aws. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, pages 1240–1240, New York, NY, USA, 2019. ACM.
- [18] Giuseppe Paterno. Nosql tutorial: A comprehensive look at the nosql database. *Linux J.*, 1999(67es), November 1999.
- [19] Felipe Pezoa, Juan L. Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 263–273, Republic and Canton

- of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.
- [20] Jaroslav Pokorny. Nosql databases: a step to database scalability in web environment. *International Journal of Web Information Systems*, 9(1):69–82, 2013.
 - [21] Sharvari Rautmare and DM Bhalerao. Mysql and nosql database comparison for iot application. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pages 235–238. IEEE, 2016.
 - [22] Fabrício R Santos and José Miguel Ortega. Bioinformática aplicada à genômica. *Melhoramento Genômico, Minas Gerais: UFV*, pages 93–98, 2003.
 - [23] CORPORATE The MPI Forum. Mpi: A message passing interface. In *Proceedings of the 1993 ACM/IEEE Conference on Supercomputing*, Supercomputing '93, pages 878–883, New York, NY, USA, 1993. ACM.