Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2021.020303
ISSN 2790-0916 Vol. 2, Issue 3: 22-31

Distributed System Based on Correlation Sequencing Algorithm

Naseren Jaber[*]

Vytautas Magnus University, Lithuania

[*] *corresponding author*

Keywords: Correlation Analysis, Correlation Sequencing Algorithms, Distributed Systems, Distributed Architectures

Abstract: Due to the huge amount of data processed in the distributed storage system, and the transformation of tasks processed in the traditional stand-alone environment into a distributed processing mode, which increases the cost of network and scheduling, operations in traditional stand-alone databases are performed in distributed storage systems appear extra expensive in the system. With the development of high-throughput sequencing technologies, data analysis has increasingly introduced distributed technologies to address the challenges posed by the explosion of data. The main purpose of this paper is to design and study distributed systems based on correlation sequencing algorithms. In order to alleviate the relatively lagging development of this field in China, this paper builds a distributed computing and analysis platform for omics data analysis, and proposes an optimization method based on the retrieval and computing problems in omics data analysis. Experiments show that the MD5 value of the result file obtained after the error is recovered and the result file is compared with the result file without error is the same. It shows that the result obtained by the fault-tolerant method in this paper is correct. Finally, the effectiveness of the optimization is verified through specific test data.

1. Introduction

In a distributed environment, the connection operation needs to transmit multiple relationships involved in the connection between hosts, which requires a lot of network costs. In this environment, distributed connection optimization technologies continue to emerge. The current connection optimization technology mainly focuses on the optimization in the following ways. First, how to reduce the network cost of connection operations in a distributed environment, secondly, how to improve the parallelism of connection operations, thereby reducing the computation time of connection operations, and finally, how to decide the optimal execution plan for connection operations [1-2].

Copyright: © 2021 by the authors. This is an Open Access article distributed under the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (https://creativecommons.org/licenses/by/4.0/).

In related research, Abidi et al. used an intelligent hybrid learning algorithm with meta-heuristic improvement to realize flexible manufacturing system (FMS) scheduling [3]. After collecting the benchmark dataset for FMS, feature extraction was performed using t-distributed stochastic neighborhood embedding, linear discriminant analysis, linear squared regression, and higher-order statistical features. An optimal weighted feature extraction method is developed to select optimal features with less correlation using a modified lion algorithm (LA), called modified nomadic-based LA (MN-LA). Elwin introduced a distributed finite element algorithm that allows swarms of mobile robots to continuously monitor environmental quantities [4]. The algorithm ensures that each robot's estimate forms part of a global estimate that spans the entire domain, fuses measurements from the entire population, and takes into account spatial correlations between measured and estimated positions. By incorporating spatial correlations without the need to transmit measurements or measure locations, the algorithm decouples its communication needs from the spatial statistics of the environment and enables robots with fixed capabilities to monitor environments with varying spatial correlation lengths.

In this paper, based on the correlation sequencing algorithm, the design research of distributed system is carried out. This paper first designs a retrieval optimization model to solve the omics data retrieval problem. In order to break through the performance bottleneck of traditional omics analysis applications, this paper analyzes and discusses the application of correlation sequencing algorithm design in data analysis. Through the implementation and comparison of different parallelization of classical sequence alignment algorithms, the rationality of the optimal design should follow the principle of balance is proposed. Thirdly, the data are compared experimentally to verify the effectiveness of the method. Finally, combined with the proposed data analysis and optimization method, related systems and applications are designed and implemented.

2. Design Research

2.1. Characteristics of Task Scheduling Problems

The task scheduling problem in the distributed multiprocessor system environment has the following characteristics:

First, the heterogeneity of communication and computing resources. In traditional on-chip multiprocessor systems, processing nodes and network types are usually isomorphic, so the task scheduling strategy is relatively simple, only the task information and the status of processing nodes need to be considered, and even the problem of communication competition does not need to be considered [5-6]. However, in the distributed multiprocessor system, the scalability is strong, and it can also be connected to a larger system as a subsystem, and it may be heterogeneous whether it is a processing node or a network. The heterogeneity of processing nodes may be caused by different types of processors in the processing nodes. For example, there are functional differences between the CPU and GPU. It may also be caused by different frequencies of the processors, different instruction sets, and different number of cores. The difference in computing power reflected between generations of CPUs. The network type may also be a combination of mesh, star, bus and other structures, and there may not be a point-to-point communication link between some processing nodes. There may be a problem of contention between processing nodes for the communication link [7-8].

Second, the relevance of tasks. There are different dependencies between multitasking and large tasks. There are two types of dependencies between tasks: internal dependencies and indirect dependencies. The subtasks are typical internal correlations. Generally, such tasks can be represented by a DAG graph (Directed Acyclic Graph), so such tasks can also be called DAG tasks [9-10]. A large task can also be transformed into a task set with internal correlation composed of several sub-tasks by the method of task decomposition. Indirect dependency means that there is no dependency between tasks, but there is competition for resources (processors, memory, communication lines, peripherals, etc.) [11-12]. The dependencies of tasks can closely affect the performance of task scheduling [13-14]. Tasks can also be divided into divisible tasks (such as workflow) and indivisible tasks according to whether they can be divided. Indivisible tasks have the "atomicity" feature and are scheduled in the system as a whole.

Third, sharing. In the scheduling process of a distributed multiprocessor system, in order to improve resource utilization, multiple tasks need to share a set of resources. The structure and characteristics of this task can be diverse, and their requirements for computing resources are also diverse. This will result in the competition of multiple tasks for shared resources (including processing nodes, communication links, peripherals, etc.) [15-16]. In addition, the time when different tasks are submitted to the system for scheduling may also be different, which also increases the The complexity and difficulty of scheduling, and at the same time, it will cause problems such as the minimization of the time span (Makespan) and the fairness of scheduling [17-18].

2.2. System Implementation Requirements

The data analysis platform plans to further utilize the data that has been stably collected so far, and establish a regional energy consumption data analysis system on the current display and analysis platform [19-20]. This storage system mainly has the following implementation requirements:

(1) The system needs to provide concurrent query functions for multiple underlying databases. Concurrent query can concurrently acquire the data of multiple underlying data tables with the same structure or similar structure, and combine the sub-results and return them to the requester.

(2) The system needs to provide cross-database connection query function for multiple underlying data nodes. Cross-database connection query needs to connect multiple data tables distributed in two or even more databases, and return the connected result to the requester.

(3) The system needs to provide global aggregate function support. For data analysis systems, common date functions and aggregation functions such as sum and count are essential.

(4) The design of the storage system should control the development and change costs as much as possible to ensure the feasibility of the system implementation.

(5) The system does not need to provide a distributed update function. For the upper-level energy consumption data analysis system, the underlying storage system only needs to provide efficient reading and data merging functions, and the update function can be updated regularly by other components or even manually.

(6) The system does not need to provide distributed transaction support.

2.3. Correlation Analysis Calculation

There are different methods for measuring the correlation. The purpose of this study is to calculate the correlation between two RSRP sequences. It is necessary to determine the degree of the correlation. Therefore, the correlation coefficient is selected to describe the correlation between the sequences. The correlation coefficient between the online acquisition sequence segment and the matching segment in the motion sequence database is calculated by formula (1), and the calculation result is stored in the matrix ρ.

In addition, the average Euclidean distance between the signals is further calculated for the matched segments in the l matrix, as shown in the following formula:

$$d = \frac{\sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}}{n} \tag{1}$$

where d represents the average Euclidean distance between signals, X and Y represent the matching segment, and n represents the length of the matching segment.

So far, the maximum matching value, correlation coefficient, and average Euclidean distance of the matching segments have been obtained, and the following will use the three to perform adaptive voting to select the optimal matching segment. Adaptive voting means that the total weight of the three is assumed to be 1, and the total weight is divided into three parts and assigned to the maximum matching value, correlation coefficient and average Euclidean distance, as shown in formula (2). Then sum the values of the three after assigning the weights to obtain the maximum value. In this way, the maximum value is obtained for all matching sequences, and then the segment corresponding to the maximum value among the maximum values is selected as the final positioning. Fragment.

$$v = \arg\max_{i} | a_0 \times X_i + a_1 \times Y_i + a_2 \times Z_i | \tag{2}$$

Among them, a0, a1, a2 are three weights, and a0+a1+a2=1.

3. Experimental Study

3.1. Distributed Storage Architecture

Compared with traditional single-database databases, the biggest difference between distributed databases is the physical distribution. A distributed database divides data according to vertical or horizontal rules and distributes it on multiple physical machines. Although the integration of distributed database will bring a certain degree of complexity to the system, due to the common data explosion in most systems, it is urgent to rely on distributed database technology to solve the performance bottleneck of reading and writing in the storage layer. Distributed database technology has been Widely used in industrial production. On the other hand, since major manufacturers are keen to seize the market in the distributed database field, and a large number of distributed database open source projects have emerged, it can be said that a hundred flowers are blooming.

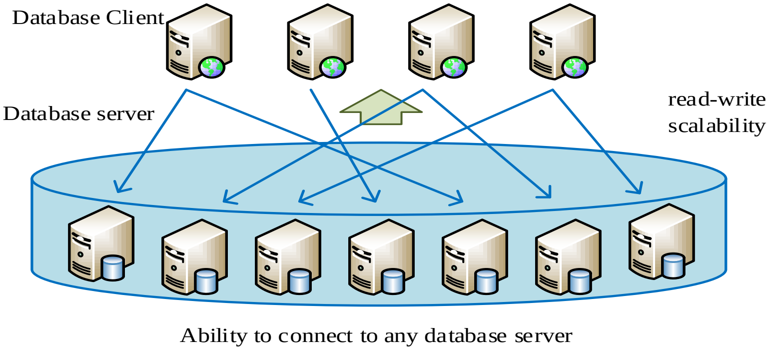

Figure 1. Distributed database deployment architecture diagram

At present, distributed databases are mainly divided into two types, one is the RAC architecture led by Oracle, and the other is the middleware-based architecture that is currently popular among open source teams and small application vendors. On the one hand, a perfect architecture requires more servers to carry, and the deployment cost of the storage system increases greatly. On the other hand, the complex deployment architecture also increases maintenance costs. When maintaining the entire storage system, the maintenance work involves many sub-nodes and components under the system. Finally, due to the complexity of distributed database technology, there are certain additional knowledge requirements for end users, application developers, and even database engineers.

3.2. Distributed Multiprocessor System Model

Figure 2 presents a simple example model of a distributed multiprocessor system, which consists of three elements: processing nodes, switching nodes, and communication links. The processing node is responsible for the execution of tasks, the switching node is used for the routing and transfer of tasks and data, and the communication link is responsible for the transmission of tasks and data between nodes.

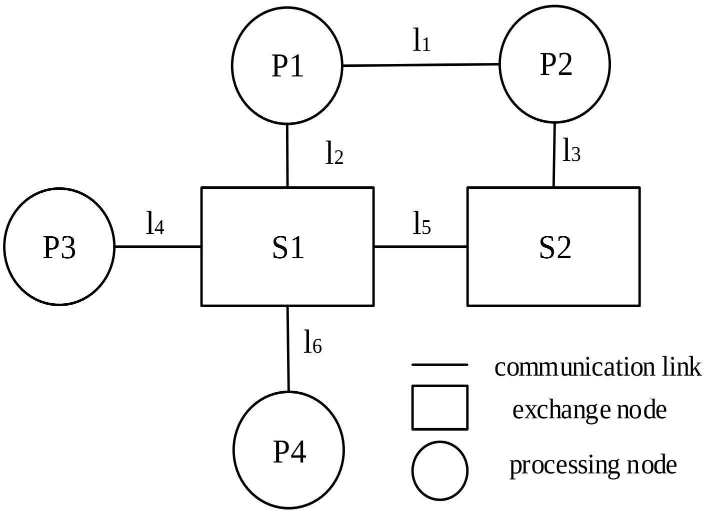

Figure 2. Example topology of a distributed multiprocessor system

The distributed multiprocessor system structure can be represented by an undirected graph DHMPS=(P,S,L), where P is the set of processing nodes in the system, represented by P={p1,p2,...,pm}, | P|=m is the number of processing nodes, {p1, p2, p3, p4} in Figure 2 is the processing node. S is the set of switching nodes in the system, which is used for routing selection, represented by S={s1,s2,...,sn}, |S|=n is the number of switching nodes, {s1 in Figure 2,s2} is the exchange node. L is the set of communication links in the system, which is the physical transmission medium between nodes. It is represented by L={l1,l2,...,lk}, and |L|=k is the number of communication links. {l1,l2,l3,l4,l5,l6} is the communication link.

3.3. Test Platform

The test platform mainly includes RSRP data acquisition platform, MEMS data acquisition platform, and then analyzes the positioning performance of the test data through MATLAB, and finally reproduces the user's motion trajectory on the server. The software and hardware platforms for this test are shown in the table below:

Table 1. List of hardware platforms for localization testing of correlation sequencing algorithms

Name	Quantity	Describe
Cell phone	1 part	Model: gt-s7568, system: android4.1, with sensor
Wireless data terminal	1	Model: huawei-e5375-2678
Sim card	1	Carrier: mobile, for wireless data terminals
Notebook	1 set	Cpu: i7-4710mq, memory: 4g, win7 system
Desktop computer	1 set	Cpu: i3-3240cpu@3.40ghz, memory: 8gb, win7 system

Table 2. List of software platforms for localization testing of correlation sequencing algorithms

Name	Description
Wifi Location	Mobile phone application software, collect MEMS data and save it as a file in.txt format
Probe	Laptop software, collect RSRP data, parse RSRP data
Hua	Laptop software to connect wireless data terminal to computer
Matlab2015b	Desktop computer software, data processing and analysis

4. Experiment Analysis

Sequence alignment was performed using different amounts of data, all data were tested 5 times, and the test results were averaged for 5 times. The required time is shown in the table and figure:

Table 3. Processing time for different data volumes

Data Volume(GB)	4	8	12	16	20
DNBLI running time(min)	21.2	44.7	72.3	95.1	112.3
Flex Run Time(min)	21.3	40.2	66.1	87.7	107.8
D-Mapping running time(min)	18.1	36.7	58.5	79.9	100.3

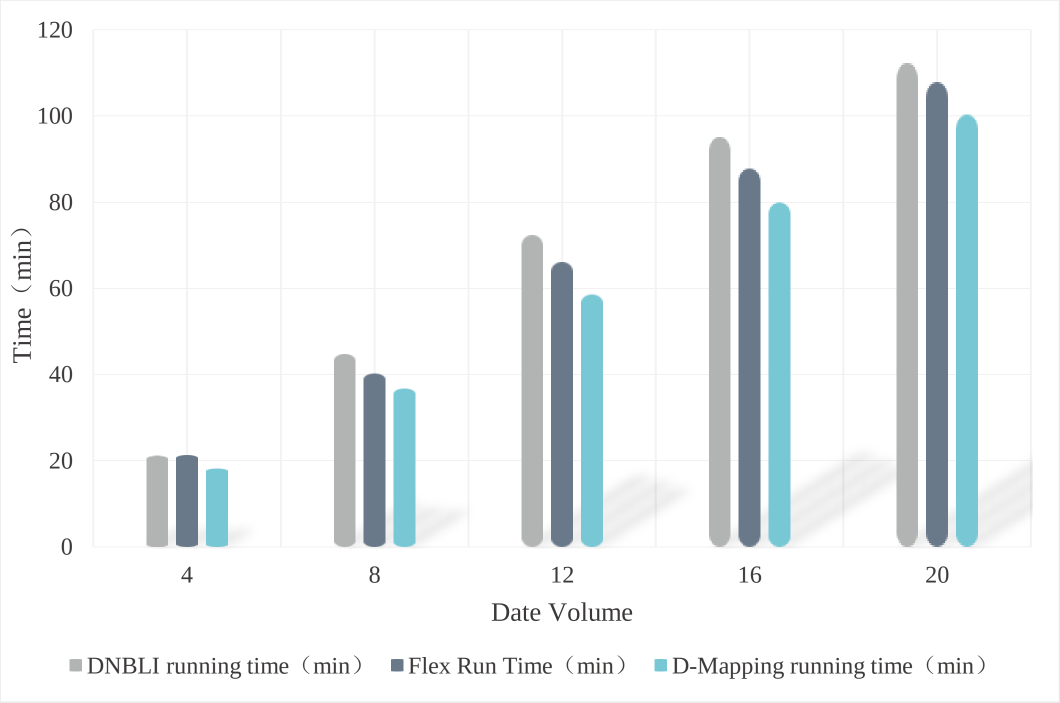

Figure 3. Analysis of processing time for different data volumes

It can be seen that when the data size is small, the performance difference of the three algorithms is not large. But when the amount of data increases gradually, the dynamic load balancing algorithm proposed in this paper has better performance than other algorithms and saves computing time.

There are two main performance indicators to measure the fault tolerance of a distributed system: checkpoint time and rollback time. Among them, checkpoint time refers to the overhead required to complete the checkpoint; rollback time refers to the rollback of using checkpoints to restore the computing state of the system time required for the process. This paper uses the system and data to test the fault tolerance performance of these two indicators. The test data is a 20G sequence extracted from SRX000600. The test content is to make node 2 fail at different time points, and test the rollback time and recovery of the system. Time, all data were tested 5 times, and the test results were averaged for 5 times. The results are shown in the table.

Table 4. Checkpoint time and rollback time at different error times

System error time point(min)	Checkpoint Time(s)	Rollback time(s)
15	4.76	0.62
30	9.71	0.76
45	14.36	0.69
60	19.85	0.72
75	25.91	0.66

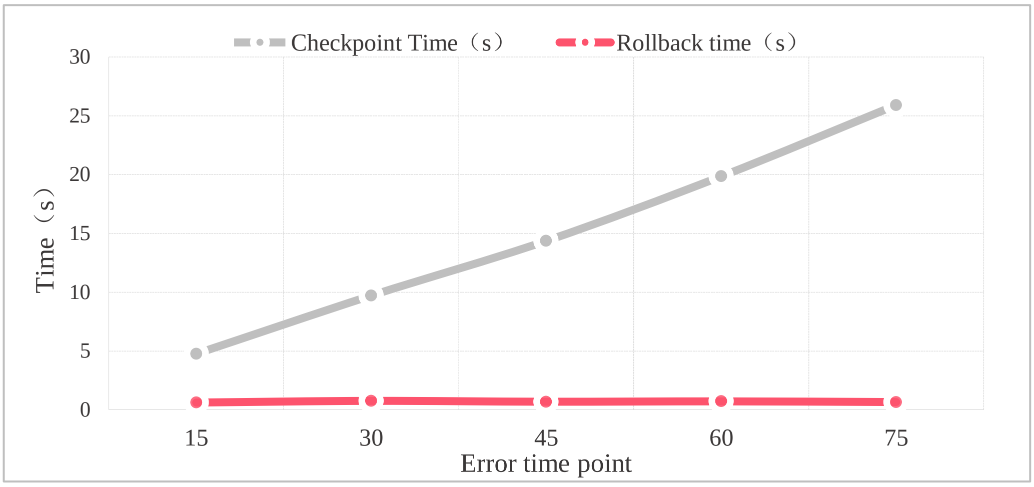

Figure 4. Analysis of checkpoint time and rollback time at different error times

The ratio of the sum of the system checkpoint time and rollback recovery time to the total system running time should be controlled within 10%. The experimental data shows that the total processing time of D-Mapping for 20G data in this system environment is about 101 minutes. It can be seen from the checkpoint time and rollback recovery time in the table that the fault-tolerant method in this paper meets this requirement. The difference between the rollback times of different data amounts is not large because the method in this paper is a user-controlled checkpoint method for the special application scenario of sequence alignment, which is highly coupled with the D-Mapping system. In the computing environment, each node only receives the unmatched sequence numbers. There is basically no difference for data of different amounts of data. The difference in time is only the error caused by the network transmission delay and the different operating states of the nodes. At the same time, by comparing the MD5 value of the result file obtained after the error is recovered and the result file is compared with the result file without error, it is found that the MD5 value is the same, indicating that the result obtained by the fault-tolerant method in this paper is correct.

5. Conclusion

After completing the preliminary system architecture design and middleware selection, a new problem emerged: in the data analysis system, it is necessary to frequently connect multiple tables across databases. With a certain amount of data, the existing middleware cannot meet the requirements of connection performance, and a query optimizer needs to be developed for the middleware to meet the performance requirements of the system for database queries. This paper analyzes and studies the database connection mechanism and characteristics in a distributed environment, and designs a connection sequence optimization scheme for the current project based on the existing resources and data model of the project. After completing the scheme design, implement the optimization scheme into the distributed middleware Presto, and finally verify the effectiveness of the optimization through specific test data.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Hennen J A, Dierking M P. Cross-correlation registration algorithm enhancements for multipixel inverse synthetic aperture LiDAR. Applied optics, 2020, 59(1):217-225. https://doi.org/10.1364/AO.59.000217

[2] Szustak L, Wyrzykowski R, Olas T, et al. Correlation of Performance Optimizations and Energy Consumption for Stencil-Based Application on Intel Xeon Scalable Processors. IEEE Transactions on Parallel and Distributed Systems, 2020, PP(99):1-1.

[3] Abidi M H, Alkhalefah H, Mohammed M K, et al. Optimal Scheduling of Flexible Manufacturing System Using Improved Lion-Based Hybrid Machine Learning Approach. IEEE Access, 2020, PP(99):1-1.

[4] Elwin M L, Freeman R A, Lynch K M. Distributed Environmental Monitoring With Finite Element Robots. IEEE Transactions on Robotics, 2020, 36(2):380-398.

[5] Williamson C, Sherer M, Zamarin D, et al. Immunotherapy and radiation therapy sequencing: State of the data on timing, efficacy, and safety.. References, 2021, 127(10):1553-1567. https://doi.org/10.1002/cncr.33424

[6] Fabreti-Oliveira R, Vale E, Oliveira C, et al. Characterization of 15 novel HLA alleles by next generation sequencing in Brazilian individuals.. REFERENCES, 2021, 97(1):60-62. https://doi.org/10.1111/tan.14109

*[7] Cheranev V, Loginova M, Jankevic T, et al. HLA-A*11:382N, a novel HLA-A null allele identified by next-generation sequencing.. REFERENCES, 2021, 97(5):448-449. https://doi.org/10.1111/tan.14185*

[8] Stratton C, Schutzbank T, Tang Y. Use of Metagenomic Next-Generation Sequencing in the Clinical Microbiology Laboratory: A Step Forward, but Not an End-All.. The Journal of molecular diagnostics : JMD, 2021, 23(11):1415-1421.

[9] Corrigan A, Taylor R, Rallis K. Next-generation sequencing of cancer genomes: lessons learned.. Future oncology (London, England), 2021, 17(31):4041-4044.

*[10] Loginova M, Paramonov I, Verevkina L, et al. A novel HLA-B*08 allele, HLA-B*08:253, was identified by next generation sequencing in two Russian individuals.. REFERENCES, 2020, 96(3):341-342. https://doi.org/10.1111/tan.13878*

*[11] Glenn M, Sherrill J, Cereb N, et al. Identification of the novel allele, HLA-DPB1*906:01, in a Caucasian individual by next-generation sequencing.. REFERENCES, 2020, 95(6):592-594. https://doi.org/10.1111/tan.13816*

[12] Soares J, Mate M A, Arumi E G, et al. Sat0531 Next Generation Sequencing And Autoinflammatory Syndromes: Clinical And Genetical Correlation On An Adult Population From A Referral Third-Care Centre. Annals of the Rheumatic Diseases, 2020, 79(Suppl 1):1223-1223.

[13] Andreev D, Smirnova V, Shatsky I. Modifications of Ribosome Profiling that Provide New

Data on the Translation Regulation.. Biochemistry. Biokhimiia, 2021, 86(9):1095-1106.

[14] Marchand V, Motorin Y. [Detection and functions of viral RNA modifications: perspectives in biology and medicine].. Virologie (Montrouge, France), 2021, 25(2):93-110. https://doi.org/10.1684/vir.2021.0899

[15] Kellner J, Ricketson L, Demczuk W, et al. Whole-Genome Analysis of Streptococcus pneumoniae Serotype 4 Causing Outbreak of Invasive Pneumococcal Disease, Alberta, Canada.. Emerging infectious diseases, 2021, 27(7):1867-1875.

[16] Andreace F, Pizzi C, Comin M. MetaProb 2: Metagenomic Reads Binning Based on Assembly Using Minimizers and K-Mers Statistics.. Journal of computational biology : a journal of computational molecular cell biology, 2021, 28(11):1052-1062. https://doi.org/10.1089/cmb.2021.0270

[17] Jac A, Ce B, Gm C. Mathematical programming modeling for joint order batching, sequencing and picker routing problems in manual order picking systems - ScienceDirect. Journal of King Saud University - Engineering Sciences, 2020, 32(3):219-228. https://doi.org/10.1016/j.jksues.2019.02.004

[18] Lecce V D, Amato A, Iaeng A, et al. Bigraph Theory for Distributed and Autonomous Cyber-Physical System Design. IAENG Internaitonal journal of computer science, 2020, 47(1):37-46.

[19] Makowski N, Campean A, Lambrecht J M, et al. Design and Testing of Stimulation and Myoelectric Recording Modules in an Implanted Distributed Neuroprosthetic System. IEEE Transactions on Biomedical Circuits and Systems, 2021, PP(99):1-1.

[20] Shaheen H I. Control system design of a virtual microgrid for Tishreen University Hospital using a distributed control strategy based on fuzzy logic controller. Tishreen University Journal for Research and Scientific Studies, 2020, 42(4):1-26.