Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2021.020203
ISSN 2790-0916 Vol. 2, Issue 2: 18-25

Campus Distributed System Platform Considering Support Vector Machine Algorithm

Malavolta Ivano*

University of Antwerp, Belgium

**corresponding author*

Keywords: Distributed System, Support Vector Machine, Campus Business, Data Acquisition

Abstract: With the increase of enrollment scale, the traditional management model of the school has brought huge pressure, and various business processes are flooded with all aspects of education and teaching management. In order to solve these serious problems, the introduction of information technology and the use of network technology, Web technology, to standardize various business processes has become an imperative task. Distributed processing technology decouples and splits the business content in the original single software system. According to business requirements, these split subtasks are distributed to processing nodes with different business processing capabilities for processing, realizing the task processing. Efficiency, perfect for handling campus business. Therefore, this paper designs a distributed system(DS) for the business management of the campus, builds a network topology map through the campus network, and then classifies the campus data according to the characteristics of the support vector machine algorithm classification to achieve a unified and orderly information management. In this paper, the data collection efficiency of the DS is tested under different nodes, and the results show that with the multiplication of collection nodes, the average number of web pages per second is multiplied, and the total time consumption is multiplied.

1. Introduction

The scale of campus data is huge, and ordinary stand-alone computers cannot meet the data management requirements at all. At the same time, the distributed structure is extremely dependent on hardware resources, and the use of DSs to manage campus files needs to be classified for each grade and each class. Therefore, this paper proposes a DS based on support vector machines, which is convenient for administrators to process school information manage.

Currently, many schools use DS platforms to manage data. For example, the distributed data acquisition system of a school is a distributed structure composed of a control terminal and multiple

acquisition nodes. The control terminal is responsible for task scheduling, and the acquisition nodes are responsible for their respective acquisition tasks. The system queries the IP address of the link through DNS resolution, and then can request resources from that IP address. However, each DNS resolution is a time-consuming process, which contains a large number of requests and greatly depends on the campus network bandwidth [1]. In order to realize the functional requirements of a distributed platform in a school, after the client of the platform issues an instruction to start executing tasks, each collection node requests Redis for a set of tasks to be fetched, and obtains the task link from it. The platform ensures the security of school data. Only after authenticating with the teacher can the user remain online, and the user can open the platform business application [2]. However, looking at the domestic colleges and universities at all levels or their colleges, although the application of information technology, the development of various business systems for practical work, and the improvement of their management information level, there are still serious problems.

This paper first expounds the role of the DS, then introduces the concept and algorithm model of SVM, and then builds a DS for the campus data management business. Then, for the deployment scheme of the system, this paper compares the cost of server procurement and the deployment of virtualization. Finally, the concurrent performance and single-node performance tests were carried out, and the system's crawling efficiency of web pages and data query efficiency were analyzed.

2. DSs and Support Vector Machines

2.1. DS

In simple terms, a DS is to divide tasks into multiple task modules for partition management or storage and other operation commands. The advantages of this platform are reflected in its inherent powerful cross-platform features and its natively provided high efficiency. on the ability of distributed application development [3-4]. All result data obtained after data preprocessing, data analysis intermediate processing result data, and final result data will be saved to the HDFS file system. In the HDFS storage architecture, the traditional single master node architecture of the Hadoop version is abandoned, and the high availability storage architecture of the dual active and standby NameNode is turned to enhance the high scalability and stability of the DS cluster [5].

2.2. Support Vector Machine Algorithm

Support vector machines are a key data classification method. In the process of data classification, the support vector machine will generate an optimal hyperplane to better separate the two types of data. The selection of the segmentation plane between the two types of data is critical, and the optimal hyperplane distance is close. The vectors are called support vectors [6]. The SVM algorithm can handle both linear and nonlinear problems.

Let x be the eigenvector composed of the independent variables of a sample, and y be the response variable, which is a factor variable, taking the value of -1 or 1. The support vector machine finds a hyperplane such that the two types of points in the data set are exactly separated on both sides, that is, the samples with y value of 1 are on one side, and the samples with a value of -1 are on the other side. Since there are many such hyperplanes, in order to ensure the uniqueness of the solution, it is stipulated that the hyperplane must also have the largest geometric interval [7].

For the geometric interval r from sample i to the hyperplane, this classification hyperplane is:

$$ax+b=0 \tag{1}$$

In formula (1), a is the coefficient of x, and b is the intercept. To make the sample point with a value of 1 on the right side of the plane, and the point with a value of -1 on the left side of the plane, it is equivalent to that for any sample point, yi(axi+b)⩾0 is established. At the same time, it is hoped to maximize the minimum interval among the geometric intervals of all sample points. Then, the problem is transformed into the solution of the following convex quadratic programming.

$$\begin{aligned} &\max_{a,b} r \\ &st \quad yi(\frac{a}{\|a\|}\cdot xi+\frac{b}{\|a\|})\geq r, i=1,2,...,N \end{aligned} \tag{2}$$

If the geometric interval r in Eq. (2) is replaced by a functional interval, the solution of the convex quadratic programming remains unchanged.

3. Design of Campus DS Based on SVM

3.1. System Network Topology

The system hardware uses 2 portal servers, multiple physical servers, and a gigabit router. Database software mainly includes Voltdb database, Mysql database. Among them, Voltdb is used to store the massive original text data of the campus network, and Mysql is used to store the result data after data analysis [8]. Since the purchased physical server is placed in the computer room of the school data center, and the school network egress bandwidth is gigabit, a gigabit switch is used when selecting the switch. The network topology of the campus DS is shown in Figure 1.

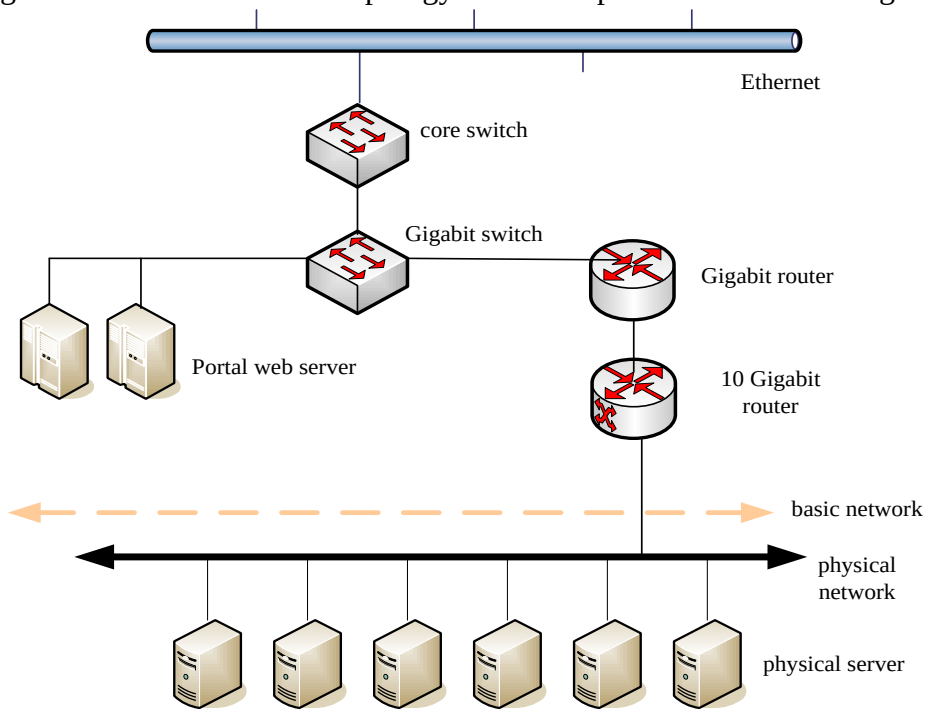

Figure 1. Campus DS network topology

Due to the limitation of the number of school public network IPs, the network environment in which the system cluster is built is a local area network. Therefore, if the system cluster is accessed through the external network, it is necessary to construct an L2TP-VPN method [9]. After setting up a VPN, both system administrators and programmers can use it as easily as they use it locally. In addition, all data on the VPN is encrypted and transmitted, which ensures the security of the entire cluster system and data [10]; after entering the big data platform cluster, the API provided by HDFS and Voltdb can be called respectively to realize the operation logic of the application, which is simple and efficient. There is no need to consider too many physical server deployment details, saving developers development and deployment time [11-12].

3.2. Distributed Database Interface Design

The distributed database interface provides users with a good interactive interface, and provides users with a complex underlying structure, allowing users to operate data through a simple visual interface, users can add, delete, modify, query and access data in persistent databases [13-14].

After the interface is started, the local data service list and the running client agent list will be downloaded through the configuration management of the metadata server. When the distributed database interface wants to view the data in the memory library, if the memory library is not started at this time, it is necessary to download the required fragmentation configuration information and memory library initialization information from the metadata server, and start the memory library [15-16]].

3.3. General Data Storage Service Based on SVM

In traditional network service programs, various school grade and class file data uploaded by end users are stored in the application program directory. In the process of application program migration or update, this method is prone to file loss or excessive data volume. large, making application migration difficult [17]. The purpose of the general data storage service is to allow applications to classify non-database data such as pictures, text, and compressed packages uploaded by users through SVM, and then store them through this service, so that these contents can be managed uniformly and reduce campus file data loss. The danger and migration difficulty [18-19].

4. Campus DS Implementation and Testing

4.1. Implementation of Platform Deployment Scheme

In order to meet the operation requirements of the campus DS, meet the design requirements of the platform and further reduce the number of hardware resources required for the operation of the platform, and reduce the procurement cost and maintenance cost, this paper proposes a hardware deployment scheme based on hardware virtualization, and Combined with the special circumstances of specific running services, some service independent hardware deployment is supplemented by the platform deployment implementation plan. Practice has proved that the hardware implementation scheme basically meets the operation requirements of the current DS platform, and in the future development, horizontal expansion of the current deployment scheme can also continue to meet future operation requirements. It actually reduces the overall hardware procurement cost, and also reduces various maintenance costs during the operation of the platform.

Table 1. Comparison of server procurement costs

Deployment method	Server type	Purchase unit price (10,000 yuan)	Purchase quantity	Total (ten thousand yuan)
Virtualization	High profile	4.5	2	30.6
	Low profile	1.8	8	
Tradition	-	1.8	21	37.8

As can be seen from Table 1, there are two types of servers using the virtualization deployment scheme, namely high-profile servers and low-profile servers. At this time, it is assumed that the unit price of high-profile servers on the market is 45,000 yuan each, and each low-profile server is 1.8 yuan. 10,000 yuan, then according to the virtualization deployment method, 2 high-profile servers and 8 low-profile servers need to be purchased, and the total procurement cost is 306,000 yuan; using the traditional deployment method, the server purchase price is 18,000 yuan per unit, and 21 The total procurement cost is 378,000 yuan. The procurement cost of servers only used to deploy main services is reduced by 19.05% compared to traditional deployment methods. It can be seen that the adoption of virtualization deployment method can effectively reduce the procurement cost of hardware equipment. At the same time, since the number of physical equipment is significantly reduced, the cost of school computer room construction, operation and maintenance, especially power consumption, will also be significantly reduced.

4.2. Performance Test

(1) Concurrency test

Test the collection performance of the DS, and judge the collection status of the entire system under different collection node configurations. Students need to obtain a lot of information in the process of learning, so DSs are used to collect different types of news content to enrich students' extracurricular life. The object of this collection is the daily news section of the news website. The total number of rolling news is 15,627, with 35 pieces of data per page, and a total of 546,945 pieces of news content, including columns such as sports, current affairs, technology, military, entertainment, and finance. The collected page content includes news text, time, source, title and other parts. Two types of tests are carried out for this data source: one is to fix the number of concurrent threads of each acquisition node to test the relationship between the acquisition speed of the system and the scale of the acquisition nodes; the other is to test the relationship between the number of concurrent threads of a single node and the acquisition speed for a single node relation.

For this data source, we use single collection node, dual collection node, and four collection node collection to count the collection efficiency. As can be seen in Figure 2 below, with the multiplication of the collection nodes, the collection effect of the collection system is also nearly doubled, which is reflected in the increase in the average number of web pages crawled per second and the reduction in the total time consumption.

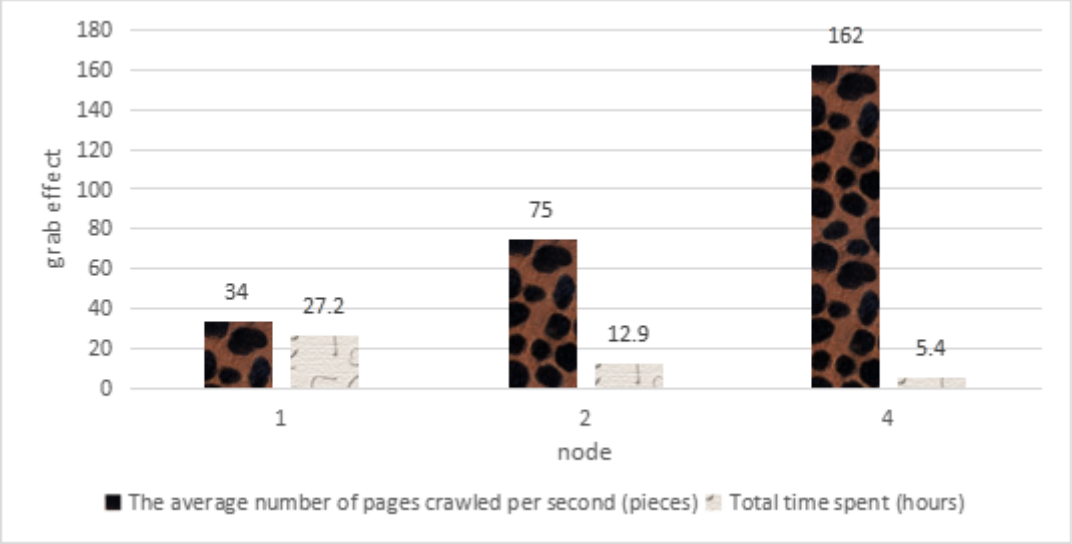

Figure 2. Statistics of web crawling rate under different collection nodes

Table 2 shows the relationship between the collection speed of the number of detection threads in the single-node mode for this data source. It can be seen that with the increase of the number of threads, the average acquisition rate does not increase linearly. Because in the process of web crawling, the interactive switching of multi-threads involves a certain time overhead. At the same time, as the number of threads increases, the bottleneck of the entire system will appear in network bandwidth, disk and memory access speed, etc. At the same time, when creating the number of threads for collection tasks, the larger the better, the same - the anti-crawling measures of the website will detect the excessively fast access speed, which will cause it to refuse to provide services.

Table 2. Statistics of web scraping rate with different number of concurrent threads

Threads	Number of pages crawled per second
10	27
20	35
40	38
80	46

(2) Single node performance test

Analyze whether the processing performance of a single node can meet the requirements when faced with different amounts of data, that is, whether its computing efficiency can keep increasing when the data processed increases. The main test is, the data access performance of Voltdb and Mysql database, test the independent read and write QPS (query rate per second QPS), and test the independent query QPS.

Adjust the two variables of the number of threads and the number of packets processed by the database at the same time for testing. The number of threads is 10, and the Mysql and Voltdb memory libraries are accessed and updated when the amount of data changes. As shown in Table 3, the execution time of the Mysql database is longer than that of the Voltdb database, but the query rate per second of Voltdb is faster than that of Mysql.

Table 3. Different data volume QPS between Voltdb and Mysql

The amount of data	1000	5000	8000	10000	20000	50000
Mysql execution time(ms)	136	1442	2173	2784	5924	11465
Voltdb execution time(ms)	22	78	165	193	586	947
Mysql_QPS	3754	3526	3214	2940	2275	1439
Voltdb_QPS	58756	51738	48762	47735	42512	39681

5. Conclusion

This paper uses the classification characteristics of SVM to design a DS for the campus, which relies on the campus network to build a network topology map. The system deploys multiple virtual servers on a single physical server, thereby improving the utilization rate of existing physical equipment, reducing the deployment of physical equipment, and achieving a double harvest of procurement costs and maintenance and operation costs. When the system is applied to school data management, the DS utilizes data processing nodes distributed in different locations to give full play to the system's functions of data classification and data storage, and cooperates to complete tasks, achieving efficient management efficiency.

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] Sontayasara T, Jariyapongpaiboon S, Promjun A, et al. Twitter Sentiment Analysis of Bangkok Tourism During COVID-19 Pandemic Using Support Vector Machine Algorithm. Journal of Disaster Research, 2021, 16(1):24-30. https://doi.org/10.20965/jdr.2021.p0024

[2] Eladl A A, Saeed M A, Sedhom B E, et al. IoT Technology-Based Protection Scheme for MT-HVDC Transmission Grids With Restoration Algorithm Using Support Vector Machine. IEEE Access, 2021, PP(99):1-1. https://doi.org/10.1109/ACCESS.2021.3085705

[3] Janani B, Vijayarani M S. Artificial bee colony algorithm for feature selection and improved support vector machine for text classification. Interlending & document supply, 2019, 47(3):154-170. https://doi.org/10.1108/IDD-09-2018-0045

[4] Bader O, Haddad D, Kallel A Y, et al. Identification of Communication Cables Based on Scattering Parameters and a Support Vector Machine Algorithm. IEEE Sensors Letters, 2021, PP(99):1-1. https://doi.org/10.1109/LSENS.2021.3087539

[5] Machiraju J, Rao S N. Effect Of K-Fold Cross Validation on Mri Brain Images Using Support Vector Machine Algorithm. International Journal of Recent Technology and Engineering, 2021, 7(6s4):301-307.

[6] Gaye B, Zhang D Wulamu A. Improvement of Support Vector Machine Algorithm in Big Data Background. Mathematical Problems in Engineering, 2021, 2021(1):1-9. https://doi.org/10.1155/2021/5594899

[7] Aridarma D, Sadikin R, Prakoso B S, et al. Ustadz Abdul Somad Lecture Sentiment Analysis Using Support Vector Machine Algorithm Comparison Of Comparative Features Selection. Jurnal Pilar Nusa Mandiri, 2020, 16(1):111-116.

[8] Yadav S, Mohan R, Yadav P K. Task Allocation Model for Optimal System Cost Using Fuzzy C-Means Clustering Technique in DS. Ingénierie des Systèmes D Information, 2020, 25(1):59-68. https://doi.org/10.18280/isi.250108

[9] Naderian S, Salemnia A. An implementation of S-transform and type-2 fuzzy kernel based support vector machine algorithm for power quality events classification. Journal of Intelligent & Fuzzy Systems, 2019, 36(6):5115-5124. https://doi.org/10.3233/JIFS-152560

[10] Kim M, Kim J. Extending the coverage area of regional ionosphere maps using a support vector machine algorithm. Annales Geophysicae, 2019, 37(1):77-87.

[11] Zakaria H, Abdullah R A, Ismail A R, et al. Predicting uniaxial compressive strength using Support Vector Machine algorithm. Warta Geologi, 2019, 45(1):13-16. https://doi.org/10.7186/WG451201903

[12] Muqeet H, Ahmad A. Optimal Scheduling for Campus Prosumer Microgrid Considering Price Based Demand Response. IEEE Access, 2020, PP(99):1-1.

[13] Klaina H, Picallo I, Lopez-Iturri P, et al. Implementation of an Interactive Environment with Multilevel Wireless Links for Distributed Botanical Garden in University Campus. IEEE Access, 2020, PP(99):1-1.

[14] Miklush V A, Tatarnikova T M, Palkin I I. Solving the problem of environmental monitoring of a port water area using a DS of sensors. Izvestiâ vysših učebnyh zavedenij Priborostroenie, 2021, 64(5):404-411.

[15] Sprenger C, Klenze T, Eilers M, et al. Igloo: Soundly Linking Compositional Refinement and Separation Logic for DS Verification. Proceedings of the ACM on Programming Languages, 2020, 4(OOPSLA):1-31. https://doi.org/10.1145/3428220

[16] Yadav S, Mohan R, Yadav P K. Fuzzy based task allocation technique in distributed computing system. International Journal of Information Technology, 2019, 11(1):13-20.

[17] Akimoto T. Developing a parallel distributed memory system of stories: A preliminary report. Procedia Computer Science, 2021, 190(2):23-30. https://doi.org/10.1016/j.procs.2021.06.002

[18] Kubiuk Y, Kharchenko K. Design and implementation of the DS using an orchestrator based on the data flow paradigm. Technology Audit and Production Reserves, 2020, 3(2(53)):38-41.

[19] Chaturvedi A, Tiwari A, Binkley D, et al. Service Evolution Analytics: Change and Evolution Mining of a DS. IEEE Transactions on Engineering Management, 2020, PP(99):1-12. https://doi.org/10.1109/TEM.2020.2987641