Eng Int Syst (2021) 2: 129–136

Engineering
Intelligent Systems

Design of Intelligent Transportation System Based on a Genetic Algorithm and Distributed Computing

Wei Yuqing* **and Gao Xing**

School of Mathematics and Information Science & Technology, Hebei Normal University of Science & Technology, Qinhuangdao, Hebei, 066004, China

With the development of society the Internet and artificial intelligence technology have been increasingly used in human life. When people use the Internet and artificial intelligence technology, the historical data generated is saved, which not only has a high scientific research value, but also has a high commercial value. By making full use of this data, the income of the enterprise can be effectively improved, the costs of the enterprise can be reduced, the competitiveness of the enterprise in the whole field can be improved, and most importantly the development potential of the enterprise can be improved. However, the biggest difficulty is that it is difficult to extract beneficial data to the development of enterprises quickly and accurately. Firstly, as the dataset is so large, and the amount of data with practical value for enterprises is scarce, there is also no rule to follow within the distribution. Therefore, it is very difficult to quickly extract the data that is beneficial to the development of enterprises. Secondly, the development of data extraction technology is not mature, and the efficiency and accuracy of data extraction are low. Current transportation big data has the characteristics of multi-source heterogeneous data, complex data types and high requirements for real-time data. Using traditional data extraction methods to extract useful traffic information from big data has been unable to keep up with the development of the times and social needs. This project aims to develop an intelligent transportation big data monitoring platform based on Ambari technology to improve the intensity of traffic supervision and the utilization of traffic data resources.

Keywords: Genetic algorithm; Distributed computing; Intelligent transportation; System design

1. INTRODUCTION

In recent years, with the continuous improvement of the national economy and the rapid development of urbanization, the number of motor vehicles is increasing exponentially, and people are using the expressway more and more frequently [1]. However, the increasingly serious "traffic diseases", such as traffic congestion, traffic accidents, and frequent highway toll evasion, which have not been paid enough attention to in the past, are slowly becoming the focus of management [2]. Despite using traditional and conservative measures, such as increasing the number of roads and the width of road surface, increasing the number of vehicle evacuation commanders at intersections, and increasing public transport vehicles, these essential problems are not being solved. The rapid development of big data and cloud computing technology provides feasible new ideas for solving traffic problems [3]. By using emerging data processing and data mining technology, the "sleeping data" can be transformed into an opportunity to reduce traffic and avoid accidents on

*Address for correspondence: Wei Yuqing, School of Mathematics and Information Science & Technology, Hebei Normal University of Science & Technology, Qinhuangdao, Hebei, 066004, China, Email: yqw_wei@126.com

Table 1 Genetic algorithm execution process.

Genetic algorithm execution process
1. Select the population C, count the population number P, set the number of iterations T, crossover probability Pc, mutation probability Pm, iteration stop threshold ED
2. Randomly select N individuals as the initial population, and select appropriate coding rules to encode the chromosomes.
3. Calculate the fitness value of each chromosome according to the designed fitness function.
4. According to the selection strategy, select individuals with better fitness.
5. Execute the crossover operator according to the crossover probability.
6. Execute the mutation operator according to the mutation rate.
7. Determine whether the iteration stop threshold is less than ED or whether the number of iterations is greater than T. If the algorithm end condition is met, the clustering result will be output as required; otherwise, go to step (3) to execute the loop until the algorithm end condition is met.

expressways [4]. By using big data technology, resources can be shared, and public security organizations can reduce crime and improve three-dimensional information highway traffic guidance; service and logistics guidance will also become a reality. The construction of an intelligent transportation system will lead our transportation to a new stage [5].

2. THEORETICAL ANALYSIS OF A GENETIC ALGORITHM AND DISTRIBUTED COMPUTING

2.1 Basic Principle of a Genetic Algorithm

A genetic algorithm is a global optimization algorithm developed on the basis of natural biological evolution. This algorithm not only uses biological knowledge, but also uses mathematical and computer science knowledge when calculating the global optimal solution [6]. A brief introduction to genetic knowledge and evolutionary knowledge in biology is necessary to understand the genetic algorithm. In the process of biological evolution, the most basic and important component is a gene, which covers a large amount of genetic information required by biological evolution, and each gene determines a physiological characteristic of the evolutionary individual. In the process of a genetic algorithm, coding and genes play the same role. Different from biological evolution, the coding itself cannot express any information, and the information in a genetic algorithm is represented by the amount of code and the order of them [7]. The two most widely used permutation sequences in a genetic algorithm are binary and floating point code. In a binary system, there are only two types of encoding: 0 and 1. All information generated in a genetic algorithm is represented by these two codes. Another important component in biological evolution is a chromosome where a chromosome is a combination of genes. Different positions of genes on the chromosome often correspond to different gene information; this is different from the calculation information caused by the different order of zero and one in a binary system. The subject of an individual within biological evolution depends on the context of its definition [8]. If the knowledge of biology is used to explain the individual, then the individual represents a single organism in the biological population; if the knowledge of genetics is used to explain the individual, then the individual represents a type of genetic object composed of multiple pieces of genetic information [9]. The basic unit of biological evolution is population. From the perspective of biology, population represents the aggregation of all the same species in a certain range. In the process of biological evolution, crossover often occurs [10]. Crossover refers to the exchange of genes in the same position of the same type of chromatids on homologous chromosomes, which is one of the reasons for species diversity [11]. In the process of biological evolution, not only does the higher probability crossover phenomenon occur, but also the lower probability mutation phenomenon occurs. If crossover is the exchange of genes, then mutation is the spontaneous change of genes, just like the code 1 in a genetic algorithm becomes a 0. Variation is the main reason for species diversity [12].

2.2 Overview of Distributed and Parallel Technology

With the progress of the times and the development of society, the use of Internet and artificial intelligence technology in all walks of life is growing exponentially. Some experts predict that the Internet data generated in the next ten years will be 40 times as much as the Internet data produced in the history of mankind [13]. However, due to the defects of traditional clustering algorithms, the utilization rate of Internet data and artificial intelligence data will remain at a low level. In this context, the basic idea of a distributed system is introduced in this paper, which integrates distributed computing with a clustering algorithm to improve the utilization of Internet data and artificial intelligence data.

2.2.1 Overview of Distributed Computing Technology

Distributed computing mainly consists of two frameworks: a parallel computing framework and a distributed file system. A parallel computing framework is used to divide the whole task into several smaller tasks, and then process each task separately [14]. When all of the small tasks are processed, they are summarized together. A distributed file system is used to divide the collected data into each node to reduce the computing requirements of a single node.

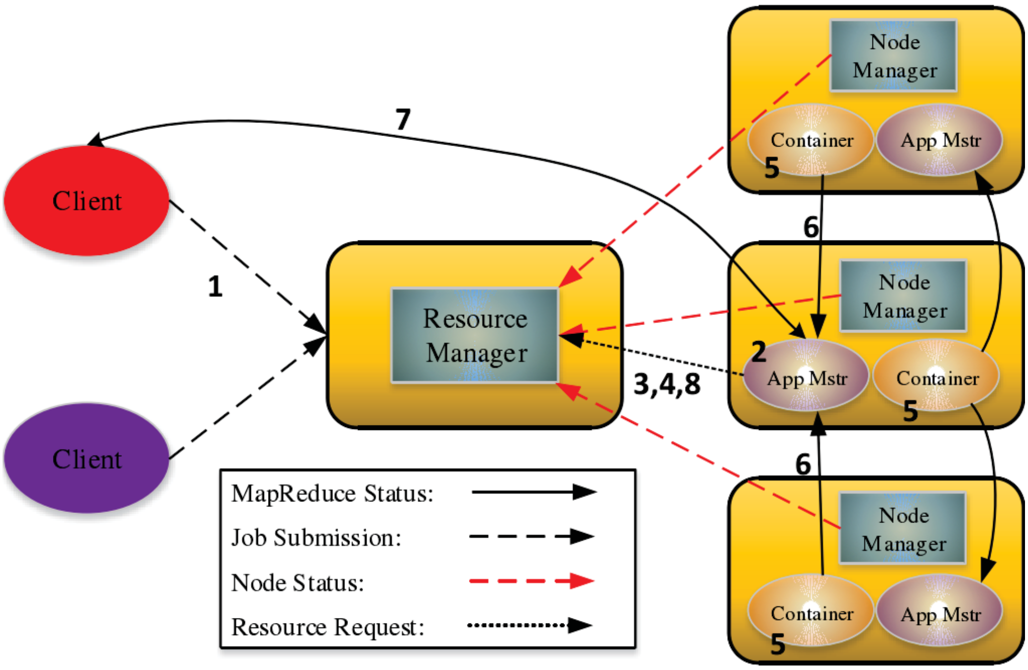

Figure 1 Brief framework of YARN.

2.2.2 Performance Evaluation Index of Parallel Computing

The traditional clustering algorithm does not require high computing power or other performance of the computer; it only requires the computer to have a high data storage capacity. This is also an important reason for the low efficiency and low accuracy of the traditional clustering algorithm [15]. The performance of sequential algorithms is evaluated using four aspects: accuracy, cost, speedup and scalability.

(1) Accuracy

Accuracy:

$$E_2 = \frac{\sum_{i=1}^{K} c_i}{n} \quad (1)$$

E_2 represents the accuracy of an experiment, n represents the number of individuals, i represents the cluster number, and Ci represents the correct number of data in the i-th cluster. The average accuracy rate is:

$$E_p = \frac{\sum_{i=1}^{N} E_{2i}}{N} \quad (2)$$

Ep represents the average accuracy of N experiments, N represents the number of experiments, i represents the number of experiments, and E_{2i} represents the accuracy of the i experiment.

(2) Parallel computing calculation cost T

$$T_t = T_n * n \quad (3)$$

(3) Parallel computing speedup ratio T

$$T_r = T_s / T_t \quad (4)$$

T indicates the running time of the algorithm in a stand-alone environment (4) Parallel computing expansion ratio E

$$E_r = T_r / n \quad (5)$$

2.3 Resource Scheduling Platform YARN

YARN is a Hadoop 2.0 version of cluster resource management system. Many processes of distributed computing require these two systems. The brief framework of YARN is shown in the following figure:

From the above figure, it is clearly shown that the Application Master and Node Manager are the main modules that make up YARN. In addition, the Container is also an important part. The specific execution process of YARN is shown in the following table:

3. INTELLIGENT TRANSPORTATION SYSTEM DESIGN AND RESULT ANALYSIS

3.1 System Framework Design

The intelligent transportation big data intelligent monitoring system is committed to providing guaranteed services for the safe operation of the traffic data platform. The monitoring platform is based on the existing mainstream open source big data components, and based on the characteristics of the traffic big data, the Ambari open source components are selected and re-developed. The independent development increases the system's maximum support for the unique characteristics of the transportation field, so as to meet the needs of traffic

Table 2 YARN execution process.

YARN execution process
1. The client sends a request to the ResourceManager for the resources needed to run the ApplicationMaster.
2. The ResourceManager applies for a container, and then runs the ApplicationMaster.
3. After the ApplicationMaster is started, it will report the node status to the ResourceManager, register in ResourceManager, and then keep in touch with the ResourceManager in heartbeat mode after registration.
4. The ApplicationMaster applies to the ResourceMaster for the containers necessary for the program to run.
5. ResourceMaster allocates a corresponding number of containers to the ApplicationMaster. The ApplicationMaster initializes the containers after obtaining the containers. After initialization, the ApplicationMaster establishes communication with the NodeManager. NodeManager starts the container, and the container is responsible for executing the specific application process. During the operation, the application and NodeMaster establish a heartbeat connection to monitor and manage each node.
6. The container uses the FPC remote communication mechanism to transfer the program running status to the ApplicationMaster.
7. When the program is running, the client can get the running status directly from the ApplicationMaster.
8. After the end of the application, the ApplicationMaster requests to release resources.

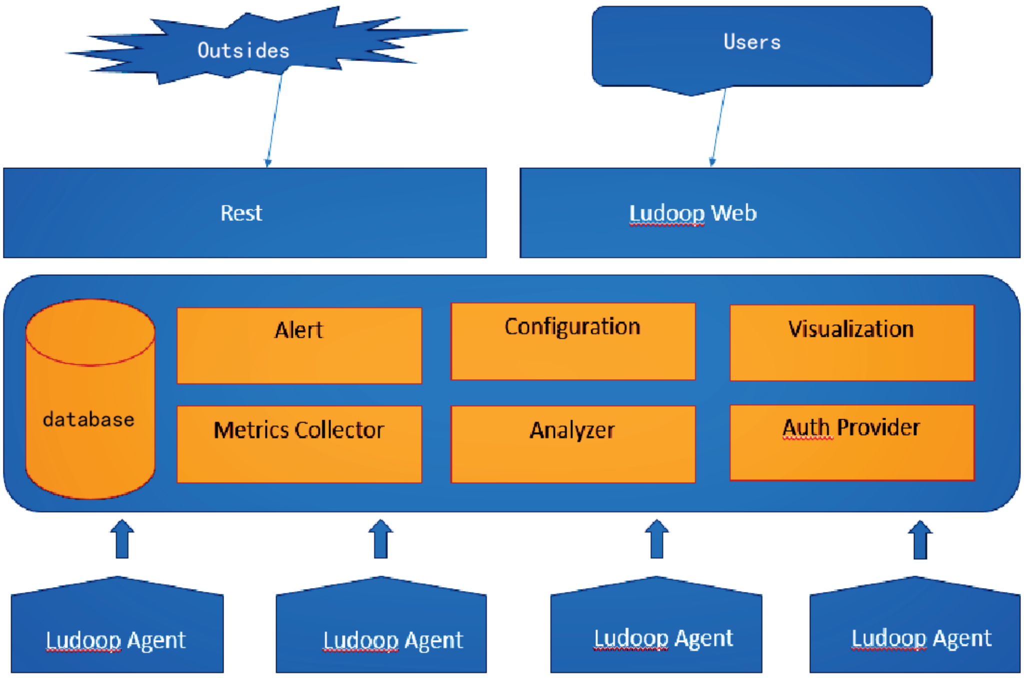

Figure 2 Logic design diagram.

big data collection, monitoring, and analysis to the greatest extent. Support deployment transactions, that is, deployment success or deployment failure, rather than a partial success or partial failure. The distributed configuration management system adopts an open-source framework with independent research and development. At the same time, it provides WEB UI management functions to view deployment logs. Based on a mature underlying framework, the service components are encapsulated in the upper layer, and the upgrade and update of the underlying framework are adapted with minimal changes to ensure the stability and high availability of the overall framework. The main design goal of this system is to provide auxiliary tools for data monitoring and adjustment for operation and maintenance personnel. These tools include cluster monitoring, service component monitoring, host information, configuration history, warning information, and administrator information modules.

The cluster monitoring module includes HDFS disk usage, number of running nodes, CPU usage, cluster load, NameNode heap usage, and ResourceManager heap usage. Operation and maintenance personnel can customize the threshold percentage by changing the color feedback usage status of the current pie chart, in order to clearly understand the running status of each task. Operation and maintenance personnel can also view real-time data every 5 minutes of the day, and can also view historical data every 5 minutes of any day; service component monitoring modules include Hadoop HDFS, MapReduce, Tez, Hive, Hbase, Pig, Sqoop, Oozie, ZoopKeeper, AmbariInfra, Ambari Metrics, Spark, ZeepelinNotebook, Slider and the detailed status of the components running on each node; the host information module includes real-time feedback and recording of the server's IP address, kernel, running memory, disk usage, average load, version, number of components and other information.

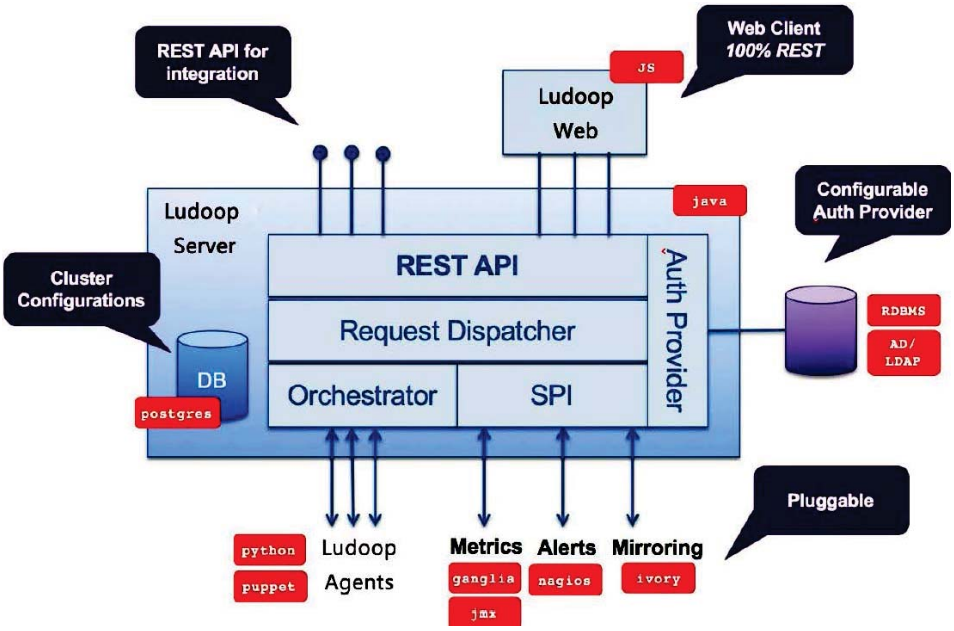

Figure 3 Overall architecture design diagram.

The operation and maintenance personnel can also add host operations according to the situation; the configuration history module can be selected by the data analyst to install service components according to the needs of the business, and the service formation provides unified management. The configuration history module can also record logs of operation and maintenance personnel in real time, and provide functions such as system logs; the administrator module can view the basic information description of all service components, including basic information such as the status. Users can filter the required information according to their needs; the warning module reports various abnormal situations found in the monitoring process in real time. When the monitoring data exceeds the threshold, the system will prompt the user with an alarm message. At the same time, the operation and maintenance personnel can resolve the warning message through simple start and stop operations.

3.2 Core Subsystem Function Module Design

Hortonworks provides a component that integrates a lot of big data for beginners, mainly used to monitor the status of the cluster. The main design is shown in Figure 3:

The core parts of the overall design are the Ludoop Server and the Ludoop Agent. The Ludoop Server provides the installation, configuration, and upgrade file preparation of the service components of the Master node. The Ludoop Agent primarily installs the Agent on the slave node for data collection and installation, the distribution, and installation, start, stop, restart and other operations of the upgraded components. The early databases of the Ludoop Server and Ludoop Agent mainly used PostgreSQL, however it now supports MySQL. MySQL was used in this case as it supports persistence. As shown in the figure below, the server side mainly maintains three types of states: the current state of the cluster, the desired node state, and the operating state.

3.3 System Function Design and Realization

The intelligent transportation big data intelligent monitoring system is an auxiliary tool for data monitoring and the adjustment of operation and maintenance personnel, including cluster monitoring, service component monitoring, host information, configuration history, warning information, and administrator information modules.

3.3.1 Login Page

Users only need to enter the assigned account number and password to log in to the traffic big data intelligent monitoring system. If the password does not match the account number, the system will refuse to log in.

3.3.2 Cluster Monitoring Summary Page

The cluster monitoring module includes HDFS disk usage, the number of running nodes, CPU usage, cluster load, Name Node heap usage, and Resource Manager Heap usage, as shown in the figure. Operation and maintenance personnel can customize the threshold percentage in order to clearly understand the running status of each task. The operation and maintenance personnel can also view the real-time data, such as CPU usage, every 5 minutes of the day, as well as the historical data every 5 minutes of any day. In addition, users can add service components according to their monitoring needs. The operation steps are as shown in the figure to add services, and then following the add service wizard to select the services to be added. The interface will provide a list of all services, service version numbers, and service function descriptions, users can then click the box in front of the service to choose according to their needs, and then click [Next] to continue adding the service. Assign masters as shown in the figure, the system provides alternative information, and assign master components to the host where the user wants to run

Table 3 Summary of hardware configuration in operating environment.

Serial number	Hardware item name	Hardware Configuration	Quantity	Remarks
1	Master node server	CPU: 4 physical cores. 2.0+GHz		
		Memory: 64GB		
		Hard Disk Space: 1TB	2	
		Operating system: CentOS 6.5+ or Redhat Enterprise 6.5+	2	
2	Slave node server	CPU: 4 physical cores, 2.0+GHz	2	
		Memory: 64GB		
		Hard Disk Space: 2TB	2	
		Operating system: CentOS 6.5+ or Redhat Enterprise 6.5 +		

Table 4 List of software environment configuration in operating environment.

Serial number	Software item name	Software version	Quantity	Remarks
1	Operating system	CentOS 6.5+ 64	6	Product operating system
2	Operating environment	JDK 1.7+ 64	6	
3	Web container	Tomcat 7.0 +	6	2 SSO, 2 interface services, 2 authorization systems
4	Database	MySQL 5.0+	2	Master-slave backup
5	Service coordination	ZooKeeper 3.3+	1	
6	Load balancing	Nginx or Apache	3	

them as needed. If the information is wrong, the [Back] button in the lower left corner can be used to return to the previous level, otherwise click [Continue] to add the service. Assign Slaves and Clients as shown in the figure, select the desired host and corresponding attributes, and click the [Return] button in the lower left corner to return to the previous level, otherwise click [Next] to continue the Add Service Wizard. Customized services are shown in Figure 4.17. All services are listed here. We can adjust the required services according to our needs. Click the [Return] button in the lower left corner to return to the previous layer, otherwise click [Next] to continue the adding service wizard. Check as shown in the figure, the user checks the verification information again, click the [Back] button in the lower left corner to return to the previous level, otherwise click [Next] to continue the adding service wizard. Install the service and start the test. The installation process and the information displayed on the installation completion interface are shown respectively. Click the [Return] button in the lower left corner to return to the previous level, otherwise click [Next] to continue the adding service wizard. Abstract As shown in the figure, the system will automatically generate some prompt information, and the user can modify it according to the prompt information. Click [Finish] to complete the operation of adding service components.

3.4 System Environment Configuration

Table 3 shows the hardware environment configuration list used in the operating environment of the intelligent transportation big data intelligent monitoring system.

The software environment configuration list used in the software product operating environment of the intelligent transportation big data intelligent monitoring system is as shown in Table 4.

3.5 System Performance Analysis

Table 5 shows the software operating performance configuration of the intelligent transportation big data intelligent monitoring system.

4. CONCLUSION

Today, we have entered the "Big Data" era. The data generated when we use the Internet for study and work will automatically be saved by the Internet system. This data includes a lot of valuable information to us. How to quickly and accurately extract this information and use it is a major problem we are facing today. Although the use of traditional data mining technology can extract the data we need from a huge database, the high amount of time and low data accuracy are unable to meet current needs, so a new type of data mining technology that can quickly extracted data with a high level of accuracy is very necessary. Since entering the new century, the number of people and the number of vehicles in all countries in the world has increased significantly. The speed of road expansion is far behind the speed of population expansion. In these years, the traffic operation has also produced a huge amount of traffic. The way in which to effectively use traffic data and apply it to road research and development and traffic control is a major problem facing the transportation field today. As today's transportation departments cannot effectively use traffic data, the deficiencies of traffic control are becoming increasingly obvious. The traditional single-center relational database technology processing method can no longer meet the needs of the current stage. The proposal of a smart transportation big data system undoubtedly increases the problem-solving efficiency. As the subsystem data monitoring system of the smart transportation big data system, through the Internet of Things, the combination of application, big data technology,

Table 5 List of running performance configuration.

Name	Detailed requirements
Concurrency	The system can accommodate 100,000 people
User navigation click-through rate	300 requests/sec per server
User navigation response time	Less than 5 seconds
Traffic View Click Rate	300 requests/sec per server
Response time for traffic viewing	Less than 2 seconds
Hard disk usage alarm	The system can automatically check the space of the disk partition. When the disk occupancy rate>90%, and the remaining disk memory <5G, it will send an alarm.
Ping police	The ping server sends 10 ping packets to each IP of the target machine. If the server does not receive a data packet response from the target machine's IP, it will send a ping unreachable alarm.
Insufficient memory warning	Swap_in and swap_out have an indicator, if the value of this indicator is > 20 and lasts for 15 minutes, an alarm will be sent. Only when the values of swap_in and swap_out are 0, can the warning prompt be cancelled.
Monitor average load response	Delay $<$ 180s
Application average visit response	Delay $<$ 180s
Alarm response	Delay $<$ 180s
Monitoring data collection	Delay $<$ 300s
Monitor API performance	Support a single point of 20 concurrent requests
Comparison of monitoring streams	Reflect the maximum, minimum, and period in a day
Monitor historical data	Supports up to 30 data that can be queried, with a storage period > 5 years
Cluster node	Supports the dynamic expansion of nodes >300
Disk storage space	Gives an alarm when the used space > 90%
CPU	Gives an alarm when the CPU usage rate has been > 95% for 30 minutes

and cloud computing system has greatly promoted the storage and processing of traffic data and the display of friendly interfaces, and provided strong support for solving this traffic problem.

ACKNOWLEDGEMENTS

The study was supported by key subjects of "Research institute of hebei university of applied technology" Province "Depend on the Internet Research and Practice on teaching reform of computer public course. Grant No. JY2019064"

REFERENCES

1. A. Hwang and J. Hoey, Smart Home, the Next Generation: Closing the Gap between Users and Technology. In: AAAI Fall Symposium on Gerontechnology, Arlington 39(5) (2012), 14–21.
2. S. Jones and L. Shao, Content-based Retrieval of Human Actions from Realistic Video Databases. Inf Sci 23(6) (2013), 56–65.
3. JA. Lowthian, et al., The Challenges of Population Ageing: Accelerating Demand for Emergency Ambulance Services by Older Patients, 1995–2015. Med J Austr 194(11) (2011), 574–578.
4. A. Maratea, A. Petrosino and M. Manzo, Generation of Description Metadata for Video Files. In: Proceedings of the 14th International Conference on Computer Systems and Technologies—CompSysTech'13, vol 76(7) (2013), 262–269.
5. J. Matejka, T. Grossman and G. Fitzmaurice, Video Lens: Rapid Playback and Exploration of Large Video Collections and Associated Metadata. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST'14, ACM, New York 82(9) (2014), 541–550.
6. M. Mazloom, et al., Querying for Video Events by Semantic Signatures From Few Examples. In: Proceedings of the 21st ACM International Conference on Multimedia 68(5) (2013), 609–612
7. S. McCloskey and P. Davalos, Activity Detection in the Wild using Video Metadata. In: Pattern recognition (ICPR) 58(7) (2012), 3140–3143.
8. S. McGinnis and J. Moore, The Impact of the Aging Population on the Health Workforce in the United States: Summary of Key Findings. In: Cahiers de Sociologie et de Démographie 28(6) (2006), 568-573.
9. R. Mehla and R. Aggarwal, Automatic Speech Recognition: A Survey. Int J Adv Res Comput Sci Electron Eng 3(1) (2014), 45–53.

10. F. Metze, et al., Beyond Audio and Video Retrieval: Topic-oriented Multimedia Summarization. Int J Multimed Inf Retriev 2(2) (2013), 131–144.
11. A. Mihailidis, et al., The COACH Prompting System to Assist Older Adults with Dementia through Handwashing: An Efficacy Study. BMC Geriatr 8(5) (2008), 28.
12. SA. O'Neill, et al., Video Reminders as Cognitive Prosthetics for People with Dementia. Ageing Int 36(2) (2010), 267–282.
13. P. Panchal, S. Merchant and N. Patel, Scene Detection and Retrieval of Video using Motion Vector and Occurrence Rate of Shot Boundaries. In: 2012 Nirma University International Conference on Engineering (NUiCONE) 65(9) (2012), 1–6.
14. DP. Papadopoulos, et al., Automatic Summarization and Annotation of Videos with Lack of Metadata Information. Expert Syst Appl 40(14) (2013), 5765–5778.
15. BV. Patel and BB. Meshram, Content Based Video Retrieval Systems. Int J UbiComp 3(2) (2012), 13–30.