Research on the New Mode of Undergraduates' Innovation and Entrepreneurship Education Under the Background of 'Internet Plus'

Liyan Tu, Lan Wu(✉), and Xiaoqiang Wu

Inner Mongolia University for the Nationalities, Tongliao 028000, China
tlyimun@163.com, wlimun@163.com

Abstract. With the rapid development of network information technology and the popularity of mobile intelligent terminals, Internet industry has experienced rapid growth. As the main force for building socialism, college students have demonstrated unprecedented passion for innovation and entrepreneurship. In the 'Internet plus' background, under the innovation and entrepreneurship service platform for college students, resource scheduling is the key to the entire system, affecting the performance of the system. For this reason, this paper proposes a resource scheduling algorithm for college students' innovation and entrepreneurship service platform based on load balancing, and constructs a dynamic project resource allocation scheme, which improves the utilization of resources. Experimental results verify the effectiveness of the proposed resource scheduling algorithm, the match rate between the project and the university graduates has increased by 30%.

Keywords: Internet plus · College student entrepreneurship · Resource scheduling · Load balancing

1 Introduction

With the continuous development of Internet technology, human society has entered a new era of information-based social development, the Internet economy era. As a new force for innovation and entrepreneurship, college students rely on the 'Internet plus' to carry out innovation and entrepreneurship with broad prospects for development, and provide opportunities for college students to realize the success of innovation and entrepreneurship [1, 2]. 'Internet plus' is characterized by its rapid technological innovation, wide application range, and many business models as its basic characteristics, and has become an important driving force in the process of promoting China's economic and social development. Under the 'Internet plus' action plan, the Internet and traditional industries can be optimized and upgraded. In the process, more job opportunities that are consistent with the development of the times and the characteristics of college students can be derived [3, 4].

College students' innovation and entrepreneurship refers to college students who have used their own innovation and entrepreneurial abilities and possessed resource platforms to develop their own careers during or after their graduation [5]. The group of

S. Liu and G. Yang (Eds.): ADHIP 2018, LNICST 279, pp. 10–19, 2019.
https://doi.org/10.1007/978-3-030-19086-6_2

college students has a comparative advantage in 'Internet plus' innovation and entrepreneurship. For today's college students, the Internet is a part of their daily lives. They can use the Internet platform more familiarly to obtain the required information, so that college students can more easily grasp 'Internet plus' era of innovation and entrepreneurship. From this point of view, 'Internet plus' innovation and venture has become one of the patterns that college students are more inclined to choose in their innovation and entrepreneurship project activities. The 'Internet plus' era has provided favorable conditions to college students' innovation and entrepreneurial activities.

Based on the 'Internet plus' background, this paper proposes a suitable resource scheduling algorithm based on the 'Internet Plus' background, and combines the resource scheduling algorithm with various aspects of college students to match the project resources, so as to build a dynamic project resource allocation and scheduling model. The utilization of resources has been greatly improved.

2 Basic Model

2.1 Load Balancing

Load balancing has two meanings. On the one hand, a single heavy load operation is shared and distributed to multiple node devices for parallel processing. After each node device is processed, the results are summarized and returned to the user. The system processing capability is greatly improved [6]. This is what we often call a cluster technology. On the other hand, a large amount of concurrent access or data traffic is distributed to multiple node devices for processing, reducing the waiting time for users to respond. This is mainly for web applications such as Web servers, FTP servers, and enterprise critical application servers.

In order to meet different application requirements, a variety of load balancing technologies have emerged. Typical load balancing technologies include the following.

(1) Global/local load balancing

Global load balancing refers to load balancing between servers with different network structures and computers located in different locations [7]. The purpose of global load balancing is to further improve the quality of access, provide the nearest service, solve network congestion, and improve the response speed of the server.

Local load balancing refers to load balancing between servers that have the same network structure and computers located in the same location. The purpose of local load balancing is to effectively solve the problem of heavy network load. It does not need to stop existing services, change the existing network structure, but simply add a new server to the service group.

(2) Software/hardware load balancing

Software load is achieved by installing software to achieve load balancing. The advantages are low cost, flexible use, and simple configuration [8]. Its inadequacy is that the operation of the software will inevitably cause indefinite resource consumption. In general, resource consumption is directly proportional to the size of the function.

Therefore, when there is a relatively large connection request, the software itself mainly determines the success or failure of the server. Problems with the operating system itself often cause related security issues. In addition, due to the limitations of the operating system, software scalability does not show up well.

The strategy of hardware load balancing is to install special equipment to achieve load balancing, and the overall performance is well improved. In addition, due to the intelligent flow management and diversification of load strategies, the load balancing requirements can be optimized.

(3) Network-level load balancing

Starting from different levels of the network, relevant network load balancing techniques are used to solve different bottlenecks for overloading.

2.2 Resource Scheduling

The purpose of resource scheduling is to meet the system resource requirements as much as possible and to achieve the highest scheduling efficiency and minimum resource usage. The precondition for achieving this goal is to clearly grasp the resource holdings, demand, progress and other information of each subproject. The specific method is to allocate system resources based on resource availability and system requirements, using a reasonable deployment system and scientific technology methods. Resource scheduling is a necessary task for the management of the entire platform, and it is also a core issue to be resolved by the platform. A project platform often includes a large number of various types of resources.

Currently, commonly used resource scheduling algorithms include genetic algorithms, user direct assignment algorithms, shortest path algorithms, and greedy algorithms.

(1) Genetic algorithm

The complexity of genetic algorithms is relatively high. The idea of algorithm originated from the biological 'survival of the fittest.' It evolved with some evolutionary phenomena [9]. The genetic algorithm consists of four parts: coding, fitness function, genetic factor, and operating parameters. It has a wide range of applications and is mainly used to solve problems in machine learning, production scheduling, industrial operation management and other fields.

(2) User Direct Assignment Algorithm

User direct assignment algorithm is less complex. Because this algorithm does not consider the length of execution time and load conditions, it is suitable for jobs that have specific needs for resources. The idea of the algorithm is that the resource node assigned by the job is assigned by the user. The resource itself only determines the execution efficiency and completion time of the job.

(3) The shortest path algorithm

The shortest path algorithm is similar to the RIP protocol algorithm in the IP routing protocol [10]. The purpose is to complete the corresponding job with the shortest routing distance.

(4) Greedy algorithm

The greedy algorithm is also called the shortest completion time algorithm. What it is most concerned with is to complete the task in the shortest possible time. It is to schedule the task to the resource node with the shortest completion time to complete the calculation.

2.3 Web Service

Web Service provides an application to the outside world with an API that can be invoked via the Web [11]. Figure 1 shows the working principle of Web Service.

Fig. 1. Web Service working principle diagram

Web Service is a distributed, networked, modular component. It is used to perform certain specific tasks. The Web Service platform is a set of standards. In order to realize the goal of cross-platform and interoperability, Web Service is based solely on XML, XSD and other independent software vendors and platform-independent standards. It is a new platform for creating distributed applications and interoperability.

3 Load Balancing Based Resource Scheduling Algorithm

After a cluster system has been running for a period of time, the actual load on the node changes more or less compared to the amount of load recorded on the scheduler. Therefore, it is necessary to continuously query the load information and update the load information table. The purpose of the query is to make the actual load of the node match the load record. After the load query, you can see that the query data contains information that characterizes server features. However, detailed load information of a node is still difficult to characterize. Therefore, in order to obtain the load L_i of the server, information elements need to be integrated using the following formula

$$L_i = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & k_5 \end{bmatrix} \cdot \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \\ e_i \end{bmatrix} \tag{1}$$

where a is the disk usage, b is the memory usage, c is the process share, d is the bandwidth share, and e is the CPU share. The factor k_j is used to emphasize the degree of influence of each part on this type of service. The bigger k, the factor has a significant impact on the service. The k-value is determined after much experimental verification. The coefficients that do not properly reflect the load capacity of the nodes should be constantly modified to ensure that the most appropriate set of coefficients can be found in the end.

3.1 Load Time Series Preprocessing

The preprocessing of system load data is mainly to pre-process the abstracted time series so that it meets the requirements of time series analysis model input. Pretreatment methods include time series analysis models of stationary, normality, and zero mean requirement.

(1) Extract trend items

First, it is judged whether $\{x_t\}$ is a non-stationary time series and contains a trend item. After the judgment is completed, first extract the trend items d_t contained in $\{x_t\}$ as follow

$$y_t = x_t - d_t \tag{2}$$

Calculated stationary time series $\{y_t\}$. Then, the time series $\{y_t\}$ is modeled. For trend item $\{d_t\}$, multiple regression can be used to estimate the time series $\{x_t\}$, and then combine y_t and d_t to calculate the final model.

(2) Zero processing

When the time series $\{x_t\}$ is a stationary time series and the mean is not equal to zero, the sequence $\{x_t\}$ mean $\hat{\mu}_x$ needs to be calculated. Then, perform zeroing according to the following formula

$$y_t = x_t - \hat{\mu}_t \tag{3}$$

After zero processing, a new sequence for modeling $\{y_t\}$ can be generated.

(3) Standardized processing

For a given observation sequence $\{x_t\}$, if the value is too large or too small, it will affect the accuracy of the model analysis. Therefore, in order to ensure the accuracy of calculations, $\{x_t\}$ is standardized.

Assuming that the observed time series is $\left\{x_t^{\{0\}}\right\}$, the data in $\left\{x_t^{\{0\}}\right\}$ is normalized as follows

$$x_t = \frac{x_t^{\{0\}} - \hat{\mu}_x}{\hat{\sigma}_x} \tag{4}$$

where $\hat{\mu}_x$ and $\hat{\sigma}_x$ are the estimated values of the mean and variance of $x_t^{\{0\}}$, respectively.

3.2 Model Parameter Estimation

Parameter estimation based on the characteristics of the ARMA model

$$x_t = \sum_{i=1}^{n} \varphi_i x_{t-i} - \sum_{j=1}^{m} \theta_j a_{t-j} + a_t \tag{5}$$

The idea of parameter estimation is to estimate the autoregressive parameter $\varphi_i (i = 1, 2, \cdots, m)$ using autocorrelation variance function R_k. First estimate φ_i, then estimate θ_i.

3.3 Load Sequence Prediction Algorithm

The overall flow of the system load time series prediction algorithm includes six stages: observation data preprocessing, autocorrelation coefficient calculation, partial correlation function calculation, AR(p) model ordering, data prediction, and standardized data restoration. The specific algorithm is as follows:

Algorithm input: System load time series $\{sp_t\}$

Algorithm output: System load forecast value at the next moment sp_{t+1}

Algorithm flow:

Step 1: Extract the trend items of the original data and eliminate the trend items in the system load time series. Zero-mean processing is implemented after eliminating trend entries.

Step 2: Estimates φ_i and θ_i.

Step 3: Based on the AIC criterion, the order of the model is tested for suitability. The model determines the order method is

$$\mathrm{AIC}(p) = N \ln \sigma_a^2 + 2p \tag{6}$$

Among them, σ_a^2 is the residual variance, and AIC(p) is a function of the model order p.

Step 4: Predict the data value at the next moment and based on the differential intermediate data saved in the first step. Perform predictive data restores on data forecasts.

Step 5: Return the prediction result and the algorithm ends.

The algorithm flow chart is shown in Fig. 2.

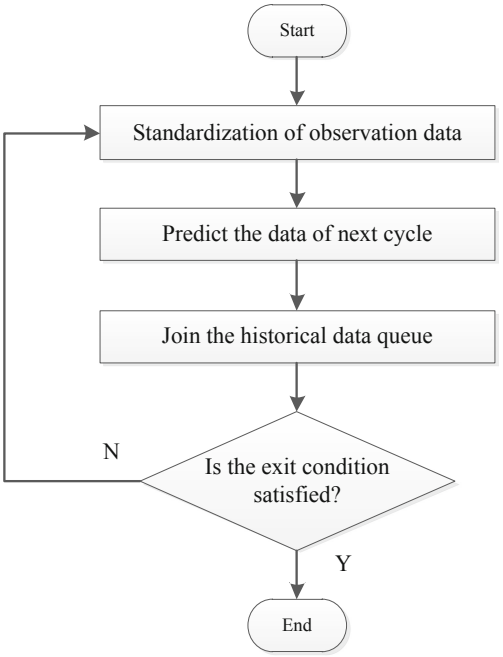

Fig. 2. Load prediction algorithm flow

4 Results and Analysis

In order to verify the accuracy and time efficiency of the scheduling algorithm, we have conducted multiple experimental studies on multiple to-do business nodes at multiple municipal offices on the youth entrepreneurship public service platform.

The analysis of college students' innovation and entrepreneurship patterns under the background of 'Internet Plus' shows that there is no adjustment to the algorithm when it comes to the allocation of projects. It takes a long time to go online and the workload of the system increases to a certain extent, which reduces the work enthusiasm for project distribution.

This section uses three servers and 70 clients as examples to verify the ARMA system load data prediction algorithm. The autocorrelation coefficient is calculated using 30 observations as a group. The initial autocorrelation coefficient curve is shown in Fig. 3. This group of coefficients tends to be stable near zero.

Figure 4 shows the prediction results of the ARMA model in this chapter. The blue curve is the observation data obtained from the acquisition system hardware resource usage. The red curve line is the ARMA predicted system load value. It can be seen from the figure that the ARMA model can accurately predict the load of the system.

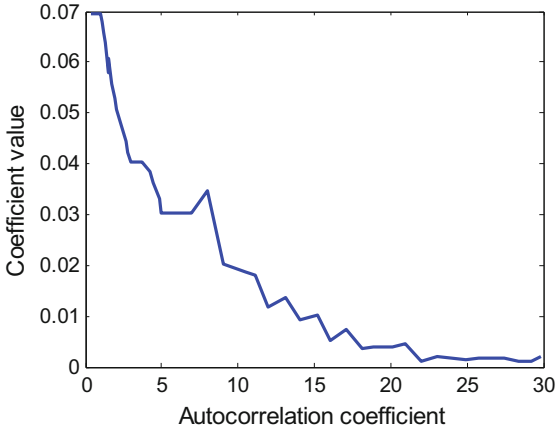

Fig. 3. Autocorrelation coefficient curve

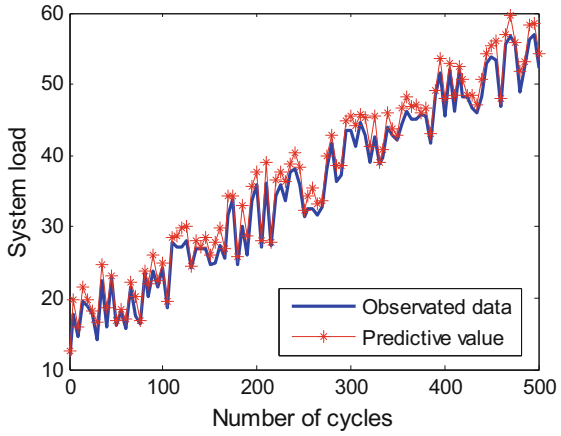

Fig. 4. Forecast result (Color figure online)

Figure 5 mainly compares the matching degree of the allocated items before and after the algorithm is used. It can be seen that before the algorithm was used, the match between the project and the youth was between 30% and 40%. That is to say, the tutors assigned by many projects did not know much about the other industries such as the entrepreneurial projects of the youth. This does not greatly help the project's growth and survival rate. However, after using the scheduling algorithm, the match between the project and youth is between 60% and 90%, and the matching degree has increased by more than 30%. Projects that are distributed in this way will have a positive effect on the growth of young people at the stages of pre-trial, actual ground-testing, and re-examination. As a result, the survival rate of entrepreneurial projects will be greatly increased.

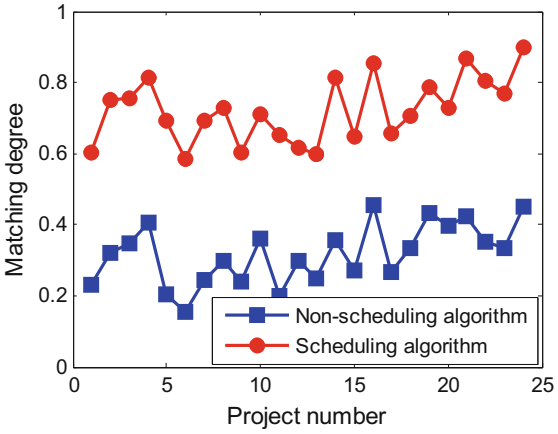

Fig. 5. Project matching degree analysis

5 Conclusion

The 'Internet plus' era is an era full of opportunities and challenges. As a fresh force for innovation and entrepreneurship, college students rely on the Internet for innovation and entrepreneurship with broad prospects for development. Based on the idea of load balancing, this paper constructs a dynamic resource allocation and scheduling model, and makes reasonable scheduling and resource allocation for each task, so that the utilization of resources can be dynamically and optimally distributed. With the development of the mobile Internet era and the popularity of smart phones, the youth entrepreneurship public service mobile platform needs further exploration.

Acknowledgements. Inner Mongolia University for Nationalities Ideological and Political Theory Teaching and Research Project (SZ2014009).

References

1. Xinchao, G.: Influence factors of the new internet technology on the development of physical education in colleges and universities. Agro Food Ind. Hi Tech **28**(1), 1508–1512 (2017)
2. Chunjiang, Y.: Development of a smart home control system based on mobile internet technology. Int. J. Smart Home **10**(3), 293–300 (2016)
3. Wu, H.: The "Internet Plus" action plan: opportunities and challenges. Frontiers **7**(1), 83–88 (2015)
4. Amadeo, M., et al.: Information-centric networking for the Internet of Things: challenges and opportunities. IEEE Netw. **30**(2), 92–100 (2016)
5. Dai, Z.C., et al.: Effects of leaf litter on inter-specific competitive ability of the invasive plant Wedelia trilobata. Ecol. Res. **31**(3), 367–374 (2016)
6. Ye, Q., et al.: User Association for Load Balancing in Heterogeneous Cellular Networks. IEEE Trans. Wireless Commun. **12**(6), 2706–2716 (2012)

7. Zhan, D., Qian, J., Cheng, Y.: Balancing global and local search in parallel efficient global optimization algorithms. J. Global Optim. **67**, 1–20 (2016)
8. Boardman, B., Harden, T., Martínez, S.: Limited range spatial load balancing in non-convex environments using sampling-based motion planners. Auton. Robots **3**, 1–18 (2018)
9. Rashid, M.A., et al.: An enhanced genetic algorithm for AB initio protein structure prediction. IEEE Trans. Evol. Comput. **20**(4), 627–644 (2016)
10. Cai, Y.D., et al.: Identification of genes associated with breast cancer metastasis to bone on a protein-protein interaction network with a shortest path algorithm. J. Proteome Res. **16**(2), 1027–1038 (2017)
11. Crasso, M., et al.: A survey of approaches to web service discovery in service-oriented architectures. J. Database Manage. **22**(1), 102–132 (2017)