Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Phase space load balancing priority scheduling algorithm for cloud computing clusters

Zhou Zheng

To cite this article: Zhou Zheng (2023) Phase space load balancing priority scheduling algorithm for cloud computing clusters, Automatika, 64:4, 1215-1224, DOI: 10.1080/00051144.2023.2254981

To link to this article: https://doi.org/10.1080/00051144.2023.2254981

Published online: 11 Sep 2023.

AUTOMATIKA
2023, VOL. 64, NO. 4, 1215–1224
https://doi.org/10.1080/00051144.2023.2254981

Phase space load balancing priority scheduling algorithm for cloud computing clusters

Zhou Zheng

Information Center, Wuxi Vocational Institute of Arts & Technology, Yixing, People's Republic of China

ABSTRACT

Due to the development of new technologies such as the Internet and cloud computing, high requirements have been placed on the storage and management of big data. At the same time, new applications in the cloud computing environment also pose new requirements for cloud storage systems, such as strong scalability and high concurrency. Currently, the existing nosql database system is based on cloud computing virtual resources, supporting dynamic addition and deletion of virtual nodes. Based on the study of phase space reconstruction, the necessity of considering traffic flow as a chaotic time series is analyzed. In addition, offline data migration methods based on load balancing are also studied. Firstly, a data migration model is proposed through analysis, and the factors that affect migration performance are analyzed. Based on this, optimization objectives for migration are proposed. Then, the system design of data migration is presented, and optimization research is conducted from two aspects around the migration optimization objectives: optimizing from the data source layer, and proposing the LBS method to convert data sources into distributed data sources, ensuring the balanced distribution of data and meeting the scalability requirements of the system. This paper applies cloud computing technology and phase space reconstruction to load balancing scheduling algorithms to promote their development.

ARTICLE HISTORY
Received 3 April 2023
Accepted 29 August 2023

KEYWORDS
Cloud computing; phase space; load balance; scheduling algorithm

1. Introduction

Cloud computing is a new IT publishing model that divides cloud computing into three service types based on service types: infrastructure, platform, and software. Users can obtain virtual machine resources from cloud providers and use the user department to deploy their own application systems [1]. In addition, due to the increase in virtual resources in cloud computing, manual and manual management of large-scale clusters is inefficient. Today, there is a need for an automated cloud computing virtual resource management technology that can not only dynamically respond to user load changes, but also effectively reduce the burden on managers [2]. This is highly consistent with the storage format of NOSql databases. For example, scalability when processing massive data can be achieved through elasticity and fragmentation. Scalability refers to the ability to expand or reduce specific resources to meet certain requirements [3, 4]. This paper introduces the basic principle of SVR and combines it with phase space reconstruction data of chaotic time series to predict short-term traffic flow. Finally, in order to verify the prediction accuracy of this model, various evaluation indicators for traffic flow prediction are introduced for error analysis, and the results are compared and analyzed with BP neural network and conventional SVR models [5]. Based on the research of massive data migration technology and analysis of migration models, the research objectives of this article are as follows: consider the data migration system from two aspects of concurrency and load balancing, consider the load balancing degree of the data source cluster and the load balancing of the migration target cluster from the perspective of load balancing, and design a data migration system [6]. Then, a data migration model is established, and based on this, LBS algorithm and Astraea scheduling algorithm are proposed. A large number of simulation experiments are conducted to verify the rationality and effectiveness of the two algorithms. The research in this paper has strong scientific significance and application prospects [7]. On the one hand, most of the research on data migration focuses on stand-alone data migration technology, while the research on distributed data migration technology is still in the development stage [8]. This article aims to improve the performance of data migration from

CONTACT Zhou Zheng zhengzhou5281cc@126.com Information Center, Wuxi Vocational Institute of Arts & Technology, Wuxi, Yixing 214206, People's Republic of China

the perspective of concurrency and load balancing. In recent years, there are many methods to improve the performance of distributed data migration, including using load balancing to improve performance to achieve higher resource utilization [9]. However, most studies only focus on the unilateral equilibrium of data source clusters or job clusters, ignoring that data migration is a collaborative process between two clusters. This article studies data migration from cluster to cluster, analyzes the main factors that affect data migration performance, and comprehensively considers the load balancing between the data source cluster and the target cluster.

2. Related work

The literature has designed a cloud based Nosql cluster automatic configuration framework NACFrame, which provides a unified interface for the outside world. For most popular nosql engines, cloud platforms such as Amazon EC2, openstack, and eucalyptus can be used to automatically expand or reduce nosql cluster resources in a customized and automated manner [10]. The literature introduces that based on the NACFrame framework, by selecting a specific nosql system hbase and a specific cloud platform openstack, the nosql cluster automatic configuration system Nosman has been designed and implemented based on the cloud computing platform, which can effectively store and manage big data according to administrator or application specifications. The literature outlines that through functional and comparative performance testing of the Nosman system, it can provide big data storage and management services, and can automatically configure Nosql cluster resources based on dynamic load changes while ensuring user SLA requirements [11]. This verifies the effectiveness and practicality of the cloud computing Nosql cluster automatic management technology proposed in this article [12]. The literature proposes an automatic configuration management technology for cloud computing Nosql clusters based on reinforcement learning. By modelling the configuration process of virtual machine clusters in a cloud computing environment as a Markov decision model, Nosql cluster resources can be automatically configured according to the dynamic changes in system operation status and load [13]. The literature describes the theory and conditions of phase space reconstruction and the feasibility of its application in traffic flow prediction. On this basis, the KNN algorithm is improved and applied to phase space reconstruction [14]. Finally, the key parameters of phase space reconstruction are obtained: embedding dimension and delay time. This is the establishment and experimental verification of a short-term traffic flow prediction model based on phase space reconstruction and SVR. Firstly, the basic principle of SVR is described. Then, combining the content of phase space reconstruction, a prediction model is designed, and the model is trained and tested using the analyzed data. Thirdly, analyze the traffic flow prediction and evaluation indicators of the model [15]. Finally, the results are compared and analyzed with neural network models and conventional SVR models. The literature introduces optimization methods for data migration from the migration target cluster layer, and uses the MapReduce programming framework to provide a distributed concurrency foundation for data migration. At the same time, using the precise resource control capabilities of Hadoop yam, the resources of the migration target cluster are evenly allocated to ensure that the resources of the migration target layer are fully utilized. The proposed LBS algorithm and Astraea algorithm are experimentally verified. Firstly, the experiment analyzes the balance of LBS algorithm data on data blocks and nodes. At the same time, it is analyzed that the data source cluster using LBS algorithm can maintain its balance under the dynamic expansion of node increase and decrease and data increase and decrease. In addition, the experiment also analyzed the effectiveness of the task scheduling layer optimization algorithm Astraea under different task combinations.

3. Design of phase space construction model for cloud computing cluster

3.1. Cloud computing virtual machine cluster configuration

The state of the virtual machine automatic configuration decision model is mainly information about virtual machine resources in cloud computing. In order to configure the decision module, it is necessary to solve the optimal configuration behaviour of virtual machine resources in this state space. Because MDP requires state discretization, we use the number of virtual machine resources applied by cloud application providers at the current stage and in running state as the state space in the state configuration decision model, represented by si. The state space of the model is shown below.

$$S = \{s_1, s_2, \ldots, s_i\} \tag{1}$$

When the application system scale is relatively large, the k value can be larger; Instead, you can set smaller values. Therefore, the action space of the model is as follows:

$$A = \{\text{add}^{(\mathbf{k})}, \text{remove}^{(\mathbf{k})}, \text{no-action}\} \tag{2}$$

The state transition function is also known as the state transition probability. In the virtual machine resource automatic configuration model, we consider many uncertain factors in the actual operation of the system. Probability is introduced to consider the probability that the application system will transition from

the current state s to the next state's after executing action a. Therefore, the definition of the state transition function can be expressed as:

$$p^{2}_{(5,5)} = \begin{cases} \frac{n'}{n} & n' < n,\ n \neq 0 \\ \frac{n}{n} & n' > n,\ n \neq 0 \\ 1 & n = n' \end{cases} \quad (3)$$

The return function requires rewards or penalties for the decision-making behaviour of the decision-making module, which affects the subsequent decision-making behaviour of the decision-making module, and therefore reflects the control strategy of the system manager to a certain extent. For example, when an application system adds or deletes virtual machines, the CPU, memory, bandwidth utilization, and other system states of virtual machine resources will change accordingly. At this time, the processing capacity of the application system increases, which can meet the system processing requirements of cloud computing application providers and real-time load requirements of users, and set a larger return value; On the contrary, after the adjustment, the processing capacity of the application system has not been improved, or even worse, and cannot meet the SLA of users and application suppliers. Therefore, small return values can be set. The return function is defined as the ratio of gain to cost, as shown in the following equation:

$$R = \frac{gain}{cos\, t} \quad (4)$$

Where, gain is defined as follows:

$$gain = \frac{\sum_{i=1}^{m} g(x_i)}{m} \quad (5)$$

"m is the number of performance indicators. Here, we choose the throughput and response time of the system as performance indicators, so m = 2, g (xi) represents the benefits of each performance indicator, as shown in the following formula:"

$$g(x_i) = \begin{cases} 1 \\ e^{-p} * \left| \frac{x_i - x_i}{x_i} \right| \end{cases} \quad (6)$$

P is the control parameter and xi is the last performance parameter value. The cost is defined as follows:

$$cost = 1 + \frac{\sum_{i=1}^{z} (1 - \bar{u}_i)^{\frac{1}{k}}}{z} \quad (7)$$

Where k is the control parameter, and we select CPU, memory, and bandwidth as the utilization parameter iu, so Z = 3. The average value of the utilization parameter iu for virtual machines used by cloud computing providers is as follows:

$$\overline{u_i} = \frac{\sum_{i=1}^{n} u_i}{n} \quad (8)$$

Cloud computing applications need to respond quickly to a large number of load requests in a short time. Q-learning maintains a two-dimensional query table indexed by state-action values, also known as a Q-table. This table stores the estimated values of state-action values Q (s, a), and is an evaluation criterion for solving optimal strategies. Its recursive form is as follows:

$$Q(s, \alpha) = R(s) + \gamma \sum_{ses} P^{A}_{(s,s)} V(s) \quad (9)$$

In the actual operation of a cloud application system, the configuration decision-making module needs to continuously make configuration decisions, select optimal policies, and execute optimal actions based on user load inputs and system operation status. At the same time, the estimated value of Q (s, a) in the Q table should be continuously updated according to the following formula to make the estimated value closer to the true maximum value, so that the decision-making module can make the optimal strategy closer to the actual situation.

$$Q(s, a) = Q(s, a) + dR(s) + Q(s, a) - Q(s, a)] \quad (10)$$

3.2. Model construction of phase space reconstruction

If the chaotic attractor of a time series is restored from the chaotic system, the prediction model can be expressed as a functional relationship between adjacent or subsequent points of the prediction point, thereby obtaining the prediction data of the prediction point. However, in practical systems, describing short-term states using components is quite complex, and each node component is nonlinear.

For chaotic time series, it is actually a seemingly irregular system in which the values of variables change disorderly over time. In chaotic systems, there are a large number of trajectories formed by variables involved in motion. Regardless of the calculation of invariants, or the construction and testing of models in the system, information can be extracted from each single variable time series of these trajectories. When a component of a single variable is delayed at certain fixed points in time, the space in which the new dimensional variable observed by the variable is located is the phase space, which can restore the organizational structure and various laws of the original system. Therefore, for the original variable, the phase space that changes due to time delay must be equivalent to the original system.

Due to the generality of chaotic time series, delaying a certain variable can achieve the same effect as described above, that is, it can capture the traffic flow phase space equivalent to the original system. At the same time, it is quite necessary to conduct phase space refraction for the specific components of each variable. Because the utilization value of components is far from

satisfying the current traffic flow state, nonlinear problems can only be solved by transforming phase space. Therefore, placing variables in a new dimensional space and reconstructing the phase space become one of the prerequisites and key steps for solving chaotic time series.

The method of coordinate delay is used to reconstruct the phase space, in which the data at a certain time delay point is processed as a new dimension, thereby embedding it into a space that is "equivalent" to the original system. The Taken theorem proves that there exists an appropriate embedding dimension in a one-dimensional time series that is infinite and has no interference.

$$\begin{aligned} X_1 &= (x(1), x(1+t), \ldots, x(1+(m-1)t)) \\ X_2 &= (x(2), x(2+t), \ldots, x(2+(m-1)t)) \end{aligned} \tag{11}$$

In formula (11), m is the embedding dimension, t is the delay time, N is the total number of phase points, and N = n – (m–1) t. In a time series, not all the closest neighbours of a component of a variable at a certain time point have compactness, so setting a delay will delete the negligible closest neighbours, that is, the components at a certain time point will be selected for phase space jump mapping. The embedding dimension grasps the dimensional space of the phase space, thereby selecting the optimal attractor subspace. Due to the selection of delay time and embedding dimensions, some variables cannot enter phase space reconstruction, so only N variables are actually reconstructed, which is also the number of phase spaces.

The delay time t is one of the important parameters in the theory of phase space reconstruction, and the optimal delay time t cannot be too large or too small. When the delay time is too large, the amount of compressed information between the reconstructed variables is large, and the signal is distorted, resulting in no correlation between the variables and excessive randomness. When the delay time is too small, the differences between reconstructed variables are too small to distinguish, resulting in redundant errors. Therefore, selecting an appropriate delay time t is the key to maintaining the relationship between the original system variables.

The embedding dimension m can be regarded as a geometric constant of a chaotic attractor. When the embedding dimension is too small, the reconstructed attractors will overlap, resulting in self intersection, and therefore cannot be fully opened. When the embedding dimension is too large, the redundant dimension will introduce noise and increase unnecessary computation. Therefore, the appropriate embedding dimension m, like the delay time t, is a key parameter to maintain the relationship between the original system variables and system characteristics.

There are two different viewpoints on the selection of t and m: one is that there is no correlation between them, and their selection can be conducted independently, independent of the influence of another parameter. The methods for obtaining delay time include mutual information method or autocorrelation function method, while the methods for calculating embedding dimension include false nearest neighbour method. Another view is that they are not mutually independent and must be selected simultaneously, such as the embedded window method.

Assume that the state transition form of the time series in the m-dimensional phase space is as follows:

$$X_{i+t} = F(x) \tag{12}$$

In the above formula, xi represents the ith phase point in the phase space, which is the specific representation of a single variable in the reconstructed phase space, and t represents the delay time, so it can be expressed as follows:

$$(x_{i+1}, \ldots, x_{i+1+(m-1)t}) = F(x_i, \ldots, x_{i+(m-1)t}) \tag{13}$$

Therefore, the model input and output formulas based on phase space reconstruction and SVR are as follows:

$$\text{input} = \begin{cases} X_1 = (x_1, \ldots, x_{1+(m-1)t})^T \\ X_2 = (x_2, \ldots, x_{2+(m-1)t})^T \\ \vdots \\ X_n = (x_n, \ldots, x_{n+(m-1)t})^T \end{cases} \tag{14}$$

$$\text{output} = \begin{cases} Y_1 = x_{1+(m-1)t+1} \\ Y_2 = x_{2+(m-1)t+1} \\ \vdots \\ Y_n = x_{n+(m-1)t+1} \end{cases} \tag{15}$$

When analyzing the prediction results, the commonly used evaluation indicators in short-term prediction mainly include mean square error mse, average absolute error mae, average relative error mape, and similarity. In this article, we will use the above representative indicators to analyze the accuracy of the model's prediction results.

$$MSE = \frac{1}{N}\sqrt{\sum_t (Y(t)' - Y(t))^2} \tag{16}$$

Mean square error refers to the square root of the square sum of errors between the predicted value and the actual measured value, and then takes its average value. It can not only represent the size of the error, but also describe the discrete or concentrated distribution of the error. When the mse is large, it means that the error dispersion is high and the prediction effect is not ideal; On the contrary, it means good prediction results. In the t-fold cross validation method for selecting model

parameters in this paper, the error formula is applied to determine whether the parameters are available.

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{t}} |\mathrm{Y(t)}' - \mathrm{Y(t)}| \tag{17}$$

Average error refers to the average absolute value of the error between the predicted value and the actual measured value, which can only describe the magnitude of the error, but cannot describe its distribution. Similarly, the smaller the mae value, the higher the accuracy of the results.

Average relative error mape.

$$\mathrm{MAPE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{t}} \left| \frac{\mathrm{Y(t)}' - \mathrm{Y(t)}}{\mathrm{Y(t)}} \right| \tag{18}$$

Similarity, taking the equality coefficient ec as an example

$$\mathrm{EC} = 1 - \frac{\sqrt{\sum_{\mathrm{t}} (\mathrm{Y(t)}' - \mathrm{Y(t)})^2}}{\sqrt{\sum_{\mathrm{t}} (\mathrm{Y(t)}')^2} + \sqrt{\sum_{\mathrm{t}} (\mathrm{Y(t)}')^2}} \tag{19}$$

3.3. Simulation experiment results and analysis

This article introduces the model design idea based on phase space reconstruction and SVR, the specific method of selecting the optimal model parameters, and the evaluation indicators of the prediction model. The following is the specific implementation process of the experiment. The experimental data is the data reconstructed from the phase space of the sample data, hereinafter collectively referred to as the sample data. The experimental operation environment is shown in Table 1.

The model used in this article is mainly divided into two parts: model training and model testing. All analysis data is taken as sample data, with three quarters of the sample data as a training set and the remaining quarter as a test set. In the process of optimizing model training parameters, the k-fold cross validation method is used to solve the key parameters of the SVR model, namely, the penalty factor (c), the kernel function parameter (y), and the error ($\in$). The parameter k for cross validation in the experiment is 10, and the search space range for model parameters is shown in Table 2.

In the process of training the model, as shown in Figure 1, the learning curve of the prediction model based on phase space reconstruction and SVR shows a

Table 1. Experimental operation environment.

Environmental information	Explain
Simulation platform	Python3.7, Pycharm Profeesional
Operating system	MacOS Sierra 10.12.3
Processor	3.0 GHz Intel Core i5
Memory	12GB

Table 2. Model parameter search scope.

Parameter	Value range
c	[1,2,3 … 100]
y	[0. 00001,0. 0001,0.001,0.01,0.1,1]
ε	[0.1,0.2,0.3 … ,1]

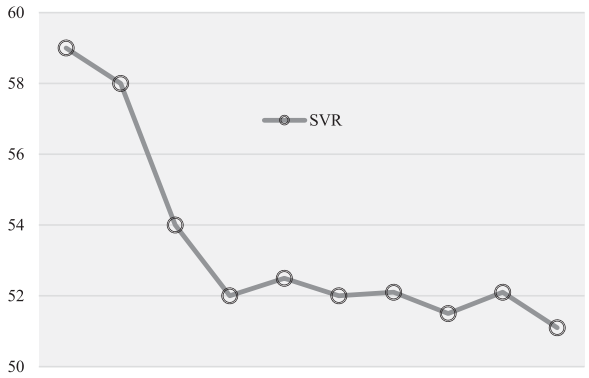

Figure 1. Learning curve based on phase space reconstruction and SVR model.

downward trend in the overall learning curve. That is, as the size of training sample data increases, the error of mse will also become smaller and smaller. Therefore, the parameters selected for this model are more accurate, that is, the optimal parameters are the penalty factor $C = 5$, and the kernel function parameters $\gamma = 0.0001$, error $\epsilon = 0.1$. At this point, the regression surface f (x) for optimal parameter allocation is the optimal predictor.

Verify whether the prediction results of the prediction model based on phase space reconstruction and SVR are accurate, that is, the test part of the model. The test data of the model is the remaining quarter of the sample data, which is placed in the optimal predictor to calculate the prediction results. The comparison between actual measured values and predicted values is shown in Figure 2.

BP neural network was proposed in the 1980s and is mainly used for pattern recognition or prediction. The method consists of a multilayer feedforward network, and is calculated using the idea of error back propagation during the solution process. The weights

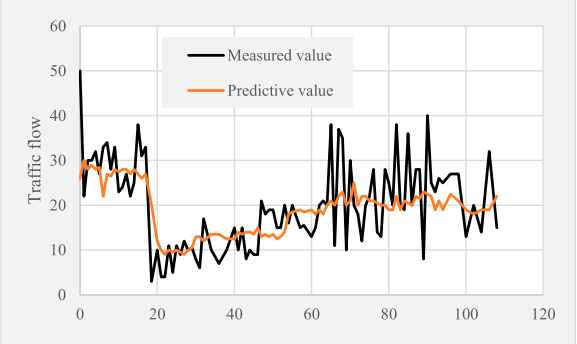

Figure 2. Display diagram of measured and predicted values.

Table 3. Comparison of indicators between BP neural network and SVR model.

Model	MSE	MAE	MAPE	EC
BP	59.16	5.67	0.30	0.82
SVR (General)	48.63	5.18	0.31	0.84
SVR (phase space)	38.90	4.65	0.30	0.86

and thresholds in the network are obtained through reverse adjustment, and the steepest descent method is used during the convergence process. This paper uses a three-layer BP neural network. In addition, to demonstrate the role of reconstructed phase space in SVR models, traditional SVR models are also introduced. The three models were evaluated using the prediction indicators listed in the application, and the results are shown in Table 3.

From Table 3, it can be seen that the model has the smallest MSE and MAE, while the largest EC, which is equivalent to the MAPE index being equal to the BP neural network. MSE, MAE, and MAPE represent the magnitude of the model error. The larger the value, the greater the error. EC represents the degree to which the predicted value of the model tends to be consistent with the actual measured value. The larger the value, the higher the consistency. Therefore, compared to the other two models, this model has better prediction performance.

4. Research on load balancing priority scheduling algorithm

4.1. Optimization method of data source layer

When the LBS method completes distributed conversion of data sources, it is said that distributed data sources achieve static load balancing. If each SHARD meets the following conditions:

$$\forall i = 1, 2, \ldots, n_3 \frac{\omega_i^1}{N_i^s} \approx \varphi \tag{20}$$

The execution process of this method is shown in Figure 3, including two main steps: data partitioning and chunk distribution. Data partitioning also includes the mapping of records to ranges and chunks in the data source; Chunk distribution describes the mapping of chunks to shards.

The first step is to use the Hash method to map a Record to a Range based on the Shard Key. The Meta Server only needs to store a few Hash functions. Secondly, through a simple slicing process (slicing can be uniform or random), a Range is mapped to several Chunks based on the amount of data covered. MetaServer needs to store the entire mapping table. Step 3: Map the Chunk to Virtual Shard using a distributed consistent hash method based on the "one-sided proximity principle". Meta Server needs to store consistent hash functions and chunk change rules (the increase or decrease in chunk size leads to changes in

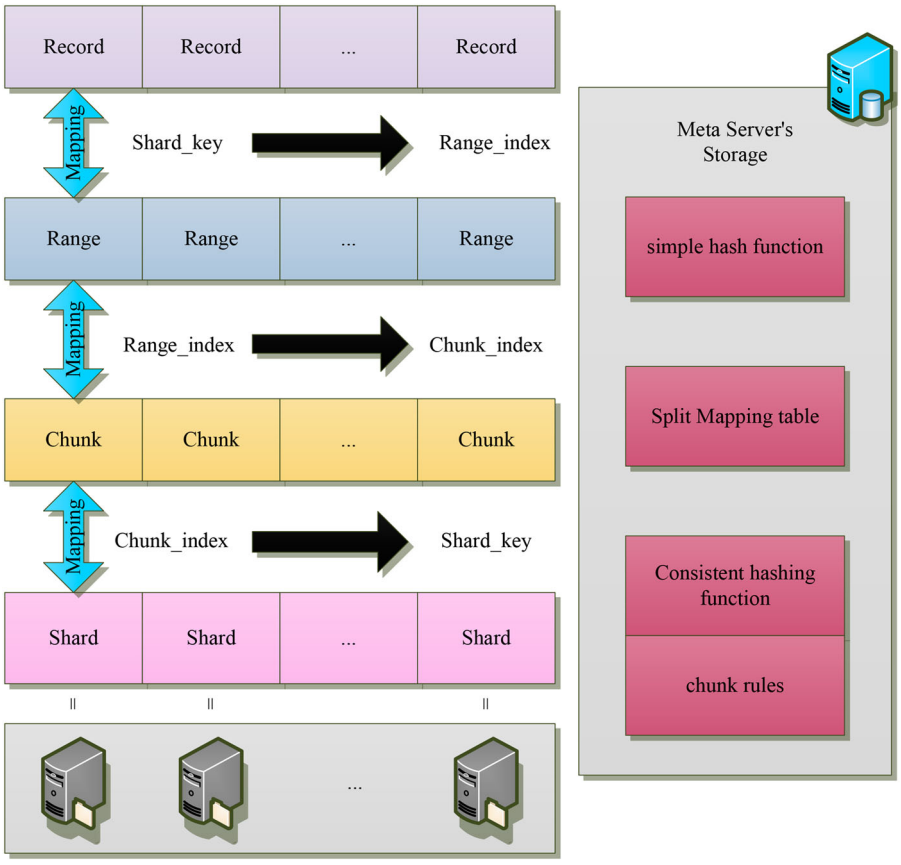

Figure 3. Execution process of LBS method.

the chunk mapping relationship). There is no need for shard changes caused by sudden increases or decreases in storage nodes, as it conforms to the mapping rules of distributed consistent hashes. Each Virtual Shard corresponds to a physical node, and all Chunks stored on the Virtual Shard will be stored on the corresponding physical node.

4.2. Optimization method of job scheduling layer

Assuming that the capacity of the migration target cluster is c, the scheduling of migration tasks needs to meet the following constraints:

$$\sum_{J_j \in JS_{Nunting}} |J_j| \leq \mathbb{C} \tag{21}$$

See the following formula for the calculation of heat hjk of data source Dj.

$$a_{ij} = \begin{cases} 1, & D_i \in J_j \\ 0, & D_i \notin J_j \end{cases} \tag{22}$$

$$b_{jk} = \begin{cases} 1, & J_j \in P_k \\ 0, & J_j \notin P_k \end{cases} \tag{23}$$

$$\forall i = 1, 2, \cdots, M; \forall k = 1, 2, \cdots, C$$

$$h_{ik} = \sum_{j=1}^{N} a_{ij} \times b_{jh} h_k = max(h_{ik}) \tag{24}$$

Finally, the optimization goal of the task scheduling layer will be equivalently translated into the following: Find the partition with the smallest number of elements in the migration job set batch process, and the partition needs to meet the following constraints:

$$\forall i = 1, 2, \cdots, M;\ j = 1, 2, \cdots, N;\ k = 1, 2, \cdots, C \tag{25}$$

The evaluation function that gives an approximate solution based on the relationship between data source access heat and runtime is shown in the following equation:

$$\forall i = 1, 2, \cdots, M; \forall k = 1, 2, \cdots, C$$

$$h_{ik} = \sum_{j=1}^{N} a_{ij} \times b_{ik} h_k = max(h_{ik})$$

$$Approxi(P\_JS) = -\sum_{k=1}^{c} h_k^{e-1} = -\sum_{k=1}^{C} max(h_{ik})^{r-1} \tag{26}$$

4.3. Simulation experiment design

The load rate calculation formula is as follows:

$$\lambda = \frac{Iosd}{time} \tag{27}$$

Where, λ Is the load rate in rea/sec, load is the number of customer visits, and time is the access time. To compare the experimental results, the standard cycle of the load was set to T = 360 min.

Create 25 virtual machines on the cloud platform to run YCSB client simulation user load programmes. The Nosman system can dynamically create virtual machines and deploy nosql clusters on the cloud platform. The resource configurations of nosql nodes and YCSB nodes are shown in Table 4.

As shown in Figure 4, the adjustment of the system with different input load amplitudes is shown. As can be seen from the figure, when the load amplitude changes slightly, the system only outputs one decision behaviour, and the number of nodes increases from 4 to 8. Because the load does not change dramatically at this time, the system will not adjust more resources. As can be seen from the figure, as the load amplitude continuously changes, the system makes multi-step decisions and continuously adjusts the number of nodes to adapt to the task requests of the load. The frequency of change in the number of nodes is basically positively correlated with the frequency of change in the load amplitude. The number of nodes increased from 4 to 10, and then stabilized to 14 in multiple steps. The load range has changed dramatically, and the system needs to configure more

Table 4. Cluster node resource configuration.

Configure	Nosql node	YCSB node
Operating system	Ubuntu linux	Ubuntu linux 12.05
CPU	8 cores	Dicaryon
Memory	12G	8G
Hard disk	100G	100G

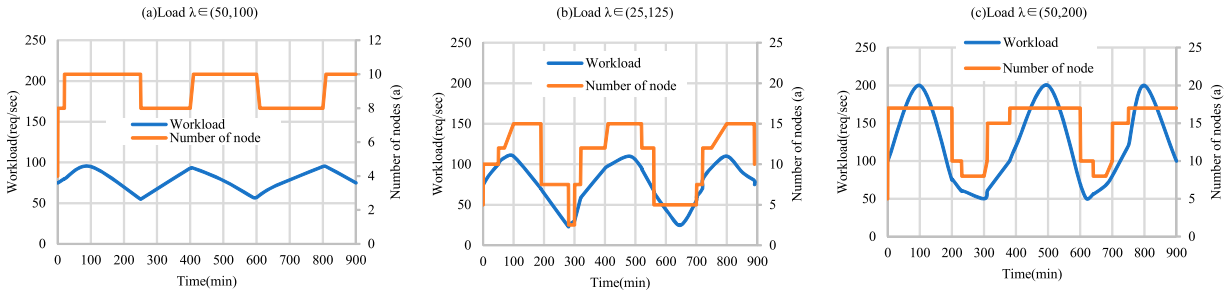

Figure 4. Adjustment of cluster resources under different amplitude loads.

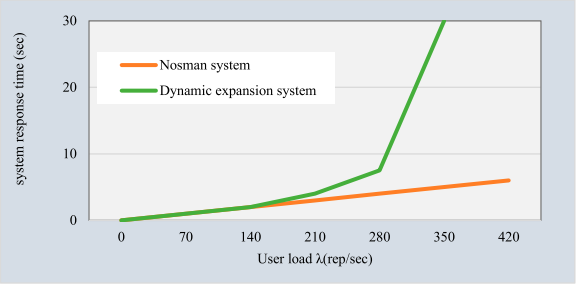

Figure 5. Change of system response time with user load.

cluster resources, with the number rising to 17. In addition, when the load range is at a trough, the system also has the behaviour of significantly deleting the number of nodes.

Through experiments to verify whether the two systems can meet the user's SLA requirements, we chose the response time of the Nosman system and the dynamic expansion system as the reference number, as shown in Figure 5. As can be seen from the figure, the solid line represents the Nosman automatic configuration system proposed in this article, and the dotted line represents the improved dynamic expansion system. As can be seen from the figure, by continuously adjusting cluster resources, we find that the response time of the NOSQL system is basically stable and slowly increasing, and the response time of the dynamically expanded system varies greatly at different stages.

The Nosql cluster automation framework proposed in this article can allow customized adjustment behaviour, so the system can allow any number of nodes to adjust the cluster behaviour, that is, the number of nodes that can be added or deleted at any one time can be selected in the user interface of the Nosman automatic configuration system. For fine-grained adjustment behaviours, such as adding one or two nodes at a time, when the load changes rapidly, more resources cannot be added in a timely manner, and thus cluster performance cannot be well optimized. On the contrary, for coarse grained adjustment behaviours, such as allowing 8 or more nodes to be added or deleted at one time, there is a risk of partial failure. For example, after the actual adjustment of cluster resources, the number of added nodes may be less than 8, but adding more nodes at one time will perform more thorough changes, which can quickly adapt to load requests and better optimize cluster performance.

As shown in Figure 6, the change in the number of nodes in the Nosman system is shown when the load is f = 1/2 T and f = 2/1, respectively. The curve in the figure clearly shows the above situation. When allowing one node to be added or deleted at a time, the cluster adjusts in a very granular manner. However, due to internal delays in the system, the system cannot fully observe changes in load. Since the automatic configuration technology in this article is based on reinforcement learning algorithms, which allow multi-step adjustment behaviour, this delay situation can be improved after the system has been running for a certain time. For faster changing loads, the

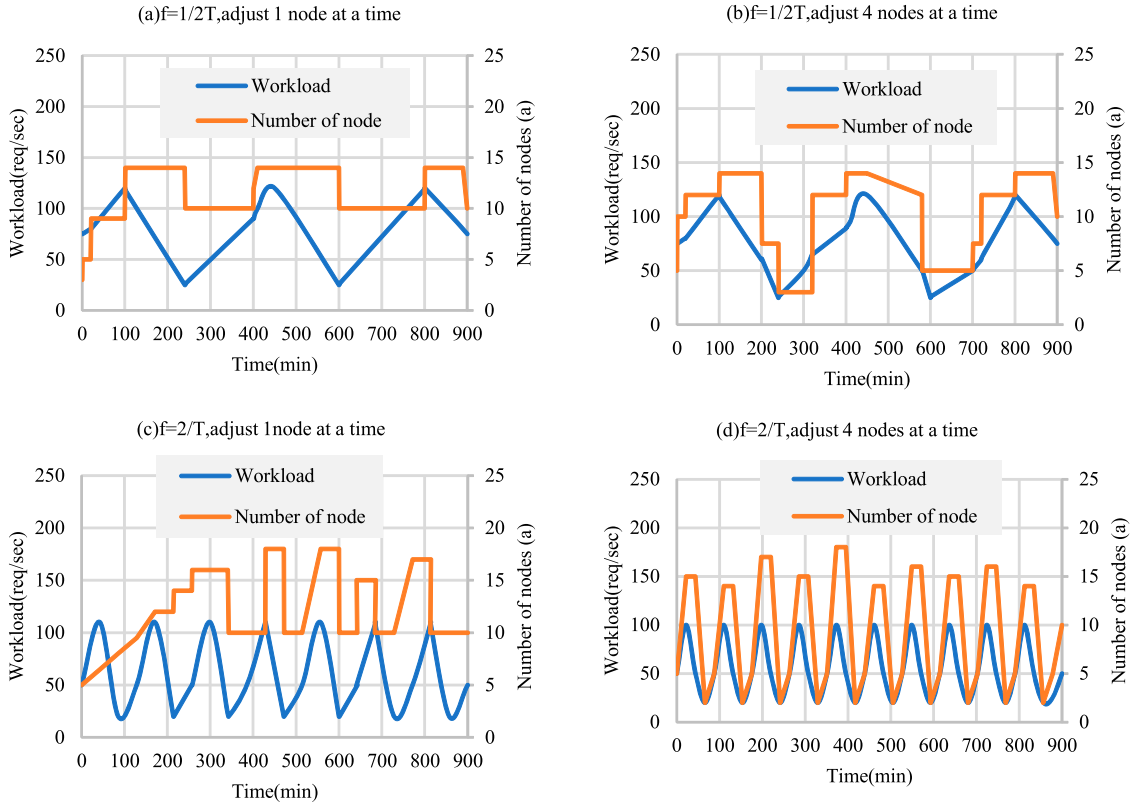

Figure 6. Adjustment of cluster resources with different load frequencies and different adjustment behaviours.

load frequency is twice the standard frequency. When adding or deleting one node at a time, load changes cannot be captured. As shown in the figure, the number of clusters varies from 11 to 14, and the adjustment behaviour of clusters varies with the periodic changes in load. However, because the number of nodes allowed to be adjusted at one time is too small, when the load is at a low ebb, the number of nosql nodes cannot be effectively reduced. As can be seen from the figure, allowing more adjustments at once can enable the cluster to better adjust cluster resources.

5. Conclusion

Due to the development and expansion of emerging technologies such as computer hardware, cloud computing, and the Internet of Things in recent years, the data owned by humans has accumulated at an unprecedented speed and continues to grow. The era of big data has quietly arrived. In the era of big data, the competitiveness of enterprises depends on whether they can utilize data more effectively and analyze knowledge from data more effectively. Especially in search business, the massive data rules mined have great value: it can improve the traffic conversion rate, thereby affecting user search satisfaction, and even guiding the direction of business development. With the increase in the value of data mining, offline data services are facing enormous opportunities and challenges. Due to data centric businesses, the inevitable transfer of large amounts of data has become a fundamental part of offline business processing: Whether large-scale data from data sources can be effectively and stably transferred to the target storage system largely determines the quality of offline business. In order to solve the problems of dynamic load changes and elastic resource requirements faced by resource managers in Nosql clusters, this article first proposes a cloud computing based automatic resource configuration management technology for Nosql clusters, and introduces reinforcement learning technology for cloud virtual resource management. In addition, the configuration management process of the NOSql virtual machine is modelled as a Markov decision model, and the addition/deletion operations of the virtual machine are automatically determined based on the dynamic changes in the execution status and input load of the system. This can respond to the real-time task requests of users, and timely complete the automatic configuration and management tasks of the NOSql virtual resources based on the dynamic changes in load to ensure the SLA requirements of cloud resource users.

Disclosure statement

The authors declare that they have no conflict of interests.

Ethical approval

This article does not contain any studies with human participants performed by any of the authors.

Data availability

Data will be made available on request.

ORCID

Zhou Zheng http://orcid.org/0009-0009-5673-9203

References

[1] Sadeeq MM, Abdulkareem NM, Zeebaree SRM, et al. Iot and cloud computing issues, challenges and opportunities: a review. Qubahan Academic J. 2021;1(2):1–7. doi:10.48161/qaj.v1n2a36

[2] Zhu W, Zhuang Y, Zhang L. A three-dimensional virtual resource scheduling method for energy saving in cloud computing. Future Gener Comput Syst. 2017;69:66–74. doi:10.1016/j.future.2016.10.034

[3] Han J, Haihong E, Le G, et al. Survey on NoSQL database. 2011 6th International Conference on Pervasive Computing and Applications; 2011; IEEE: 363–366.

[4] Tudorica BG, Bucur C. A comparison between several NoSQL databases with comments and notes. 2011 RoEduNet International Conference 10th edition: Networking in Education and Research; 2011; IEEE: 1–5.

[5] Yang H, Li X, Qiang W, et al. A network traffic forecasting method based on SA optimized ARIMA–BP neural network. Comput Netw. 2021;193:108102, doi:10.1016/j.comnet.2021.108102

[6] Du Y, Wang Z, Huang D, et al. Study of migration model based on the massive marine data hybrid cloud storage. 2012 First International Conference on Agro-Geoinformatics (Agro-Geoinformatics); 2012; IEEE: 1–4.

[7] Liu Y, Shan G, Liu Y, et al. Blockchain bridges critical national infrastructures: e-healthcare data migration perspective. IEEE Access. 2022;10:28509–28519. doi:10.1109/ACCESS.2022.3156591

[8] Teli P, Thomas MV, Chandrasekaran K. Big data migration between data centers in online cloud environment. Procedia Technol. 2016;24:1558–1565. doi:10.1016/j.protcy.2016.05.135

[9] Dongsheng W, Chuanhe H. Distributed cache memory data migration strategy based on cloud computing. Concurrency Comput: Pract Experience. 2019;31(10):e4828, doi:10.1002/cpe.4828

[10] Kassela E, Boumpouka C, Konstantinou I, et al. Automated workload-aware elasticity of NoSQL clusters in the cloud). 2014 IEEE International Conference on Big Data (Big Data); 2014; IEEE: 195–200.

[11] Truyen E, Bruzek M, Van Landuyt D, et al. Evaluation of container orchestration systems for deploying and. managing NoSQL database clusters. 2018 IEEE 11th International Conference on Cloud Computing (CLOUD); 2018; IEEE: 468–475.

[12] Farias VAE, Sousa F R C MJGR, et al. Regression based performance modeling and provisioning for NoSQL cloud databases. Future Gener Comput Syst. 2018;79:72–81. doi:10.1016/j.future.2017.08.061

[13] Makris A, Tserpes K, Andronikou V, et al. A classification of NoSQL data stores based on key design characteristics. Procedia Comput Sci. 2016;97:94–103.

[14] Zhang Y, Huang G. Traffic flow prediction model based on deep belief network and genetic algorithm. IET Intel Transport Syst. 2018;12(6):533–541. doi:10.1049/iet-its.2017.0199

[15] Ma K, Dong F. Live data migration approach from relational tables to schema-free collections with MapReduce. Int J Serv Technol Manag. 2015;21(4–6):318–335. doi:10.1504/IJSTM.2015.073942.