Software Group Rejuvenation Based on Matrix Completion and Cerebellar Model Articulation Controller

Li Su<sup>1,2</sup>, Yong Qi<sup>1</sup>

ABSTRACT
This paper aims to accurately evaluate the aging state of nodes in large-scale networks and identify the optimal rejuvenation plan for these nodes. To this end, the aging phenomenon in distributed systems was described as a random low-rank matrix. The CMAC network was introduced to collect the data of network nodes and evaluate their aging state and rejuvenation plan. Based on the aging state and plan applicability, the node relationship was integrated with matrix completion, aiming to improve the efficiency of aging evaluation. Compared to the traditional methods, our method significantly improved aging evaluation and reduced hardware cost, and offered suitable rejuvenation plans for aging nodes. The improvement is partially attributable to the incorporation of node relationship. The research findings shed new light on software aging and group rejuvenation.

Key Words: Software Aging, Cerebellar Model Articulation Controller (CMAC), Group Rejuvenation, Matrix Completion

DOI Number: 10.14704/nq.2018.16.5.1354

Introduction
For ultra-large-scale data centres, performance degradation is bound to occur after prolonged operation, leading to an increase in error rate and sudden hangs/crashes. This phenomenon poses serious threats to the reliability and availability of the software system. To cope with the threats, it is necessary to identify the exact aging nodes and rejuvenate the software in a timely manner. The precision of identification relies on the accurate learning and fitting of real-time state data, while the rejuvenation strategy also needs to be optimized against the real-time state data. In light of these, cerebellar model articulation controller (CMAC) and matrix completion are two possible to realize desirable software rejuvenation.

The CMAC is a neural network model inspired by the structure of the cerebellum. In the model, the neural network stores data locally and learns information rapidly through function approximation. Such a local learning mechanism is fast, light in weight and insensitive to the sequence of learning data. It also supports continuous (analogue) input and output. All these features make the model suitable for online learning and easy-to-implement in hardware/software. In addition, the CMAC works well in state data evaluation, and time-varying nonlinear system control, as it has no fixed model for status information.

Matrix completion is an ideal way to recover the general picture from some elements, a task that sounds impossible. In fact, the elements of most matrices can be recovered accurately by nuclear norm minimization, as long as the singular values are sparse and the

Corresponding author: Li Su
Address: School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; School of Mathematics and Statistics, Hainan National University, Haikou 571158, China
e-mail andy.suran@sina.com.cn
Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
Received: 28 March 2018; Accepted: 19 April 2018
sampling amount is proper (Candes and Tao, 2009). The accuracy of matrix completion depends on the existing data. Hence, it is reasonable to apply matrix completion to the rejuvenation of largescale distributed systems. In these systems, the information of all nodes must be deduced efficiently from a small amount of information, because of the high cost incurred in the real-time collection and analysis of the performance information of all nodes.

Through the above analysis, this paper creates an automatic online performance diagnosis and rejuvenation framework. On this basis, the service dependency graph was automatically constructed at the service level. Then, the CMAC was introduced for the aging analysis of the nodes, laying the basis for mapping random nodes to rejuvenation plans. After that, the missing node-plan relationships were restored based on matrix completion and node relationship. The node set with the highest correlation coefficient was selected from each time slot, and the target system was rejuvenated following the optimal plan.

**Methods**

**CMAC-based aging evaluation**

There are many similarities between a distributed system and the cerebellar neural network. In a distributed system, the same request is always responded with the same service, although it may carry different time and hardware/software attributes, depending on the requester. Similarly, the same input into the cerebellum will always yield the same output, despite the variations in the intermediate links. For instance, the neural circuit for visual information processing works in different states, but it never mistakes the visual information of the same object for that of another object. The most prominent features of cerebellar neural network are the ability of small sample learning and shape preference.

![Figure 1. The profile of cerebellum](image1)

![Figure 2. The cerebellum structure](image2)

![Figure 3. The CMAC structure](image3)

The CMAC (Figure 3) has a similar structure with the cerebellum (Figure 2). As shown in Figure 3, each input \( x \) has two dimensions \( x_1 \) and \( x_2 \). The values of \( x_1 \) and \( x_2 \) must be normalized. Through principal component analysis, the four most significant performance parameters were selected as the inputs of the aging analysis in this research. Let \( m \) and \( nb \) be the number of tiers and blocks of the input in each dimension \( x_i \). The number of one-dimensional uniform slices depends on \( m \) and \( nb \), i.e. \( m(nb-1) + 1 \). The values of \( m \) and \( nb \) were obtained by the Akaike information criterion (AIC). In Figure 3,
there are four tiers in each dimension, and two blocks in each tier. For instance, the first tier Tier 1 contains two blocks A and B. The blocks activated on the same tier were multiplied to generate the corresponding weight.

It can be seen from Figure 3 that the current input $X=(3.5, 3.5)$; the blocks activated in $x_1$ are denoted as $b$, $d$, $f$ and $g$, and those in $x_2$ are denoted as $B$, $D$, $F$ and $G$. Thus, the weight of the input in each tier is $Bd$, $Dd$, $Ff$, and $Gg$, respectively. Then, the four weights were added up to form the output. Thus, the number of weights is always the same as the number of tiers. Moreover, there are no such weights as $AB$, $AC$, $Ad$, and so on, for only the blocks in the same tier and different dimensions could be multiplied. Thus, the number of possible weights can be expressed as $m^*nb*n$, with $n$ being the number of input dimensions.

**Figure 4. Architecture of the CMAC network**

The architecture of the proposed CMAC network is presented in Figure 4. The network has four tiers: Tier 1 is the input layer; Tier 2 is the virtual address space for the $Aa$, $Ab$, $Ba$ and $Bb$ in Figure 3; Tier 3 is the physical storage space, which searches for the weight corresponding to the index in Tier 2; Tier 4 is the output layer.

Since the number of possible weights equals $m^*nb*n$, the number of weights will increase exponentially if the input is high-dimensional. In this case, some weights may never be activated and only a few weights are available. To reduce the useless space consumption, the weights were stored in the hash table, and only the activated weights were updated by the equation: $w_t+1 = w_t + \frac{\alpha}{m} \cdot e$, where $\alpha$ is the learning rate, $m$ is the number of tiers, $e$ is the error between the actual and predicted values.

The above CMAC network was trained with the optimal rejuvenation strategy of one node, and then used to evaluate aging state of a few nodes in the distributed system. The aging state was rated against a 10-point scale. The result is positively correlated with the applicability of the rejuvenation strategy and the aging state of nodes. If the score is greater than 5, it means the corresponding node is aging and should be treated with the rejuvenation strategy; If the score is below 5, it means the corresponding node is normal, or the rejuvenation plan is not suitable for the node.

**Matrix completion with side information**

For matrix completion, the author set up an aging node table, which can be viewed as a matrix of $m$ rows (nodes) and $n$ columns (rejuvenation plans). In the table, there are three types of entries, namely, evaluated entries (Type I), uncertain entries (Type II) and missing entries (Type III).

Type I entries are the scores obtained through CMAC evaluation of aging state and rejuvenation plan. If the matrix is empty, it means no plan has been inputted. Otherwise, the positive integers and zero in the matrix reflect the overall evaluation results of the rejuvenation plan and aging state. The value of zero indicates that the rejuvenation plan fails to change the aging state. In this case, either the rejuvenation plan is not suitable or the node is not aging. Any Type III data below 5 is uncertain. The uncertainty may come from that the inapplicability of the rejuvenation or the unclearness of the aging state.

Based on service dependency, the node relationship was constructed by default once the aging state and rejuvenation plan had been evaluated by the CMAC. The next task is to accurately restore the missing entries based on the evaluated entries and node relationship. Since the nodes were randomly yet uniformly selected from a few time slots for aging state analysis, the matrix is a low-rank one that satisfies the conditions of matrix completion. In other words, it is possible to deduce the relationship between other nodes and their rejuvenation plans based on the information of the evaluated entries.

First, these nodes and rejuvenation plans were normalized to the matrix $M=[M_{ij}]_{m \times n}$. Let $X$ be the target matrix with the rank of $r (r<< \min (m,n))$, and $E (|E| \ll m \times n)$ be the subset of $M$'s entries falling into Types I and II. In a standard matrix completion problem, the goals of matrix completion are to make the most accurate estimation possible for the table values of Types II and III entries, and remove the noises induced by uncertain measurements in Type I entries through re-estimation.
Since the target matrix is a low-rank one, each node i can be embedded into the r-dimensional space as vector $u_i$ and each rejuvenation plan j can be embedded into the r-dimensional space as vector $v_j$. Then, the interaction $a_{ij}$ between node i and rejuvenation plan j can be expressed as $u_i^T \Sigma_{r \times r} v_j$, with $\Sigma_{r \times r}$ being diagonal matrices. The vectors $u_i$ and vectors $v_j$ can be respectively aggregated into matrix $U_{m \times r}$ and matrix $V_{n \times r}$. Thus, the low-rank matrix completion problem was converted into the search for matrix $U$ with dimensions $m \times r$, matrix $V$ with dimensions $n \times r$, and a diagonal matrix $\Sigma$ with dimensions $r \times r$:

$$
M = U_{m \times r} \Sigma_{r \times r} (V_{n \times r})^T
$$

(1)

Here, the adaptive gradient decent (AGD), which applies to low-rank matrices, is adopted for the matrix completion. Let $M^0$ ($X^0$) denote the finite matrix of $M(X)$ on $E$. Then, $X$ can be estimated with or without Type II data.

Model 1: Without Type II data

$$
\begin{align*}
\text{min} \quad & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij}^0 - M_{ij}^0)^2 + \lambda_1 G(X) \\
& + \lambda_2 \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} S_{ij} \|X_i - X_j\|^2 \\
\text{s.t.} \quad & X = U_{m \times n} \sum_{r \times r} (V_{n \times r})^T
\end{align*}
$$

where $S_{ij}$ is the relationship between nodes $i$ and $j$; $X_i$ is the i-th row of $X$; $\lambda_1$ and $\lambda_2$ are the contributions of matrix completion and node relationship to the recovery of missing entries, respectively.

The rationale behind the third term is as follows: the relational closeness between two nodes is positively correlated with the applicability of a certain rejuvenation plan to them. Here, it is assumed that $S_{ij} = S_{ij}^{ALL} + \omega S_{ij}^{AD}$, where $S_{ij}^{ALL}$ is the similarity between nodes $i$ and $j$, $S_{ij}^{AD}$ is the correlation between nodes $i$ and $j$, $\omega$ is the weight controller of relationship determinants $\lambda_1$, $\lambda_2$, and $\omega$ were tuned by a 10-fold cross validation process.

Model 2: With Type II data.

$$
\begin{align*}
\text{min} \quad & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij}^0 - M_{ij}^E)^2 I(X_{ij}^0 \geq \theta_{ij}) + \lambda_1 G(X) \\
& + \lambda_2 \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} S_{ij} \|X_i - X_j\|^2 \\
\text{s.t.} \quad & X = U_{m \times n} \sum_{r \times r} (V_{n \times r})^T
\end{align*}
$$

Model 1 can be solved by the AGD through the following steps:

1. Set all greater-than-$\frac{2|E|}{m}$ columns and greater-than-$\frac{2|E|}{n}$ rows in $M^0$ to zero.
2. Replace all missing entries in $M$ with 0 and denote the resulting matrix as $M^{(0)}$. Let $M^{(0)} = U_{m \times r} \Sigma_{r \times r} V_{n \times r}$. Suppose $U^{(0)} = U_0 \times \sqrt{m}$ and $V^{(0)} = V_0 \times \sqrt{n}$, where $U_0$ and $V_0$ are composed of the first $r$ columns of $U$ and $V$, respectively.
3. Calculate the matrix $\sum_{r \times r}$ with the minimal squared error $\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} (M_{ij}^E - X_{ij}^E)^2$ under constant $U_i$ and $V_j$, forming a least squares regression problem with respect to $\sum_{r \times r}$.
4. Update $U_{j+1}(V_{j+1})$ using gradient descent: the size of each step should be proportional to its negative gradient relative to the objective function, i.e., $U_{j+1} = U_j + t \cdot \nabla U_j$ and $V_{j+1} = V_j + t \cdot \nabla V_j$, where $t$ is the step size optimizable by line search.
5. Repeat the first two steps until reaching convergence or the maximum number of iterations.

The gradients of $U$ and $V$ can be expressed as:

$$
\begin{align*}
\nabla U = & \frac{1}{m} U \left( \left( U \Sigma V^T \right)^T - M^E \right) \Sigma + U Q_u \\
& + \lambda_1 f(U, 2e^{(U_1^0 - Q_u) \frac{1}{2} \|Q_u\|})
\end{align*}
$$

$$
\begin{align*}
\nabla V = & \frac{1}{n} V \left( \left( U \Sigma V^T \right)^T - M^E \right) U \Sigma \\
& + V Q_v + \lambda_2 f(V, 2e^{(V_1^0 - Q_v) \frac{1}{2} \|Q_v\|})
\end{align*}
$$
\[ Q_{ui} = \frac{\sum_{j=1}^{u} u_{ij}^2 + \sum_{j=1}^{u} v_{ij}^2}{2\mu_0 r}, \quad Q_{v1} = \frac{\sum_{j=1}^{v} u_{mj}^2 + \sum_{j=1}^{v} v_{nj}^2}{2\mu_0 r} \]

\[ f(X_{mxr}, Y_{mx1}) = Z_{mxr} Z_{ij} = \begin{cases} \frac{X_{ij} + Y(i, 1)}{u_0 r} & \text{if } Y(i, 1) > 0 \\ 0 & \text{otherwise} \end{cases} \]

Note that the trimming process enhances the accuracy of matrix completion based on noisy entries, and the gradient calculation is detailed in Supplementary Information. The solution to Model 2 is the same with that to Model 1, other than replacing \( \sum \sum (X_{ij}^E - M_{ij}^E)^2 \) with \( \sum \sum (X_{ij}^E - M_{ij}^E)^2 - \theta(i,j) \) and \((\sum \sum V_j^E - M^E) \cdot I\), where \( I \) is an indication matrix or empty matrix and \( \cdot \) is the multiplication between the matrix and relevant elements. Model 2 was adopted for the subsequent analysis.

**Performance evaluation**

The performance of the rejuvenation plans was evaluated by root-mean-square error (RMSE). Let \( \mu = (\mu_1, \mu_2, \cdots, \mu_k) \) be the measured value and \( \xi = (\xi_1, \xi_2, \cdots, \xi_k) \) be the corresponding estimated value. Then, the RMSE between \( \mu \) and \( \xi \) can be expressed as:

\[ \text{error} = \sqrt{\frac{1}{k} \sum_{i=1}^{k} (\mu_i - \xi_i)^2} \]

If a rejuvenation plan has a small RMSE, then the estimated values are close to the true values. Then, the measured values of the optimal rejuvenation plan for the nodes were divided into 10 equal segments. Nine of these segments were selected and allocated to the training set, while the remaining 1 into the evaluation set. The selection and allocation were conducted 10 times, so that each segment was allocated to the evaluation set once. Then, the mean RMSE between the measured and evaluated values of ten runs were used to tune parameters \( \lambda_1, \lambda_2 \) and \( \omega \), and compare different plans.

**Experimental Analysis**

**Generation of Type I data**

In the CMAC system, the performance of several nodes were collected by the data collection module, while the aging state of these nodes and the applicability of the rejuvenation plan were determined by the decision module. Based on these prior data, the normal state of a node was modelled by the CMAC. Then, the deviation between the evaluated and collected runtimes was calculated, the thresholds were generated by the sequential probability ratio test (SPRT), and the two-threshold correlation (TTC) of the node was predicted in light of the data on normal and aging states. Previous research (Su et al., 2012) has shown that the SPRT can generate valid TTC in complex network using uncertain aging information.

The group rejuvenation in distributed systems requires a time slot. Assuming node \( x \) is aging at time \( t_1 \), the analysis module relies on the CMAC to calculate the rejuvenation plan \( P_a(x) \) of the node, judge if \( x \) needs to be rejuvenated by the plan and determine the applicability of the plan. Meanwhile, the computation module traverses the entire matrix, filling up all vacancies, and finds all the nodes \( Y \) \( (y_1, y_2, \cdots, y_n) \) in need of rejuvenation and the optimal rejuvenation plan.

Let \( C (c_1, c_2, \cdots, c_n) \) be the set of aging nodes with no need for rejuvenation, as their aging state is below the threshold. The nodes in set \( C \) are interdependent with the nodes in set \( Y \), the set of nodes in need of rejuvenation. After the rejuvenation of nodes in set \( Y \), the performance of nodes in set \( C \) will rebound within the time slot \( t_1 \). Note that the threshold differs from slot to slot. To determine the threshold, all the nodes should be ranked by rejuvenation needs after matrix completion. The time required for one-off group rejuvenation of the set of nodes with the greatest need for rejuvenation should be considered as the time slot threshold.

Whereas various rejuvenation plans differ greatly in effect and cost, the decision module must make a comprehensive evaluation of rejuvenation effect and cost of each plan. The evaluation was carried out in the following steps. First, collect the monitoring data, including the data on each measurement parameter. Let \( A (a_1, a_2, \cdots, a_n) \) be the set of parameter data, with \( n \) being the parameter dimension. Then, collect the parameter data after the rejuvenation of various plans. Let \( B (b_1, b_2, \cdots, b_n) \) be the set of post-rejuvenation parameter data, with \( n \) being the parameter dimension. Denote the various plans as constants \( P.A, B, \) with \( P \) being the input of the
CMAC training set. After that, evaluate the P-value of the post-rejuvenation parameter data by the trained CMAC model. 

The results of the comprehensive evaluation can be normalized and taken as the Type I data in the matrix. The node relationship matrix was pre-processed through the following normalization procedure: change each Type I measured value \( m_{ij} \) into \( \log_2(\max(H_{ij}) - \frac{\max(H_{ij})}{H_{ij}}) \), where \( \max(H_{ij}) \) is the largest measured node relationship and \( \max(H_{ij}) \) is the largest node relationship of plan \( j \); then, change each Type II value into \( \frac{\max(H_{ij})}{\text{Threshold}} \); finally, replace the missing values with 0s. 

**Formation of side information**

Since a service may utilize multiple ports, each service was expressed as a two-tuple \((ip, service name)\) \((Su et al., 2016)\), where \(ip\) is the unique host of the distributed system and \(service name\) is the unique service in the host. The service dependency in Orion system was introduced to our analysis: if service \(A\) requires service \(B\) to satisfy certain requests from its clients, then \(A \rightarrow B\).

Focusing on the popular client-server applications, the author constructed a service dependency graph based on the connection information. The dependency direction was determined by the lag correlation of the sent traffic between two services, because the packets from the server often change with those from the client in a client-server structure. To acquire the sent traffic of each service, the number of packets transmitted in a specific process was counted by probing the function `netdev`. Transmit triggered when the network device wants to transmit a buffer. Let \(X\) and \(Y\) be the sent traffic of services \(A\) and \(B\), respectively. Then, the lag correlation between \(X\) and \(Y\) can be defined as:

\[
\rho_{XY}(k) = \frac{\sum_{t=0}^{N-1} Y(t)X(t-k)}{\sqrt{\sum_{t=0}^{N-1} X^2(t) \sum_{t=0}^{N-1} Y^2(t)}} \quad k \in Z
\]

where \(k\) is the lag value. The value could be positive and negative. Here, the absolute value of \(k\) is set to 30, which can capture almost all traffic delays. Then, the goal is to find the optimal \(k\) that maximizes \(\rho_{XY}(k)\):

\[
k^* = \{\max(\rho_{XY}(k), k \in [-30,30]\}
\]

Where * is the dependency direction. If \(k^*>0\), \(A \rightarrow B\); otherwise, \(B \rightarrow A\).

**Restoration of the missing values**

In the CMAC network, the generalization parameter \(C\) and the number of actual tier neurons, i.e. the compression ratio of the virtual tier to the actual tier, directly bear on the convergence and operation accuracy of the network. In this research, the value of \(C\) is estimated based on the number of input dimensions, quantization series, and the number of actual tier neurons. Suppose the rank of the matrix varies from 3 to 13. Through a 10-fold cross validation, it is learned that the optimal parameters are: \(\lambda_1\) of 10~4, \(\lambda_2\) of 10~6 and \(\omega\) of 1,000. According to the 10-fold cross-validation errors for different ranks (Table), the lowest error was 0.5903 of rank 10.

To verify the effect of the CMAC, the nonlinear autoregressive exogenous model (NARX) was adopted for the same matrix completion and group rejuvenation, using a small number of monitoring nodes (i.e. the Type I data of the matrix). The aging state and optimal plan thus obtained were compared with those outputted by the CMAC. The results show that the RMSE of the NARX was 1.33, and that of the CMAC was 0.937. This means the accuracy of matrix completion mainly depends on the accuracy of the current data, and the CMAC derives a more precise aging state than the NARX. The CMAC also outperformed group rejuvenation in RMSE (0.937 vs. 1.26), indicating that node relationship is a good compensation for low-rank matrix completion. Moreover, the results on \(\lambda_1\), \(\lambda_2\) and \(\omega\) reveal that our method relies heavily on rank.

Next, the experimental platform in Reference (Su et al., 2013) was taken as a sub-network. In light of its topology, three similar sub-networks were created to form a network to be deployed streaming media application server, which simulates the streaming media files on client access server (CAS). All the nodes in one of the sub-networks were selected as monitoring nodes for aging state evaluation. During the 168h experiment, a total of 40,320 data samples were collected at an interval of 15s, using the collection parameters in Reference (Su et al., 2013). These parameters fully illustrate the usage of basic resources like the CPU, memory, I/O operation and VOD server network bandwidth.
The three-subnetwork structure was expanded into 20 servers in MATLAB, with 50 nodes per network. Then, the collected data were adopted for aging analysis and analogue network operation. The threshold value and number of rejuvenation nodes are recorded in Table 2. It can be seen that the first group rejuvenation appeared late, and the time slot of group rejuvenation exhibited a gradual decline. Initially, the system was running in a normal condition, and aging occurred after a long period of continuous operation. The population rejuvenation cycle was reduced with the continuous running of the system. The number of rejuvenation nodes was generally on the rise. The number decreased when the rejuvenation cycle was relatively short.

### Table 1. 10-fold cross-validation errors for different ranks

<table>
<thead>
<tr>
<th></th>
<th>r 3</th>
<th>r 4</th>
<th>r 5</th>
<th>r 6</th>
<th>r 7</th>
<th>r 8</th>
<th>r 9</th>
<th>r 10</th>
<th>r 11</th>
<th>r 12</th>
<th>r 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3432</td>
<td>1.2402</td>
<td>1.0376</td>
<td>0.9857</td>
<td>0.9702</td>
<td>0.8635</td>
<td>0.6746</td>
<td>0.5903</td>
<td>0.8411</td>
<td>0.8795</td>
<td>0.9906</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2. The threshold value and number of rejuvenation nodes

<table>
<thead>
<tr>
<th>Serial number</th>
<th>Slot number</th>
<th>Number of regenerated nodes</th>
<th>Regeneration threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10427</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>24705</td>
<td>12</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>27863</td>
<td>13</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>31402</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>36129</td>
<td>16</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>37288</td>
<td>14</td>
<td>23</td>
</tr>
<tr>
<td>7</td>
<td>39062</td>
<td>15</td>
<td>24</td>
</tr>
</tbody>
</table>

Then, an experiment was conducted at the centre of the network, and the data were recorded for 120h. The time slot was set to 10min. The number of the nodes at the centre of the network was 173. During the experiment, 27 nodes were monitoring as a group. There were 18 rejuvenation nodes at the centre and 25 in the group. After each rejuvenation, the performance of the monitored nodes increased by an average of 15% and 23%, respectively.

### Conclusions

In this research, the aging phenomenon in distributed systems was described as a random low-rank matrix. The CMAC network was introduced to collect the data of network nodes and evaluate their aging state and rejuvenation plan. Based on the aging state and plan applicability, the node relationship was integrated with matrix completion, aiming to improve the efficiency of aging evaluation. Compared to the traditional methods, our method significantly improved aging evaluation and reduced hardware cost, and offered suitable rejuvenation plans for aging nodes. The improvement is partially attributable to the incorporation of node relationship.

The future research will investigate the current findings from two aspects. First, information other than node relationship will be introduced to our method. Second, the proposed method will be modified to evaluate the aging state of small scale network nodes.

### Acknowledgments

This work was supported by the Hainan Province Natural Science Fund (No.20166221).

### References


Ding R, Fu Q, Lou JQ, Lin QW, Zhang DM, Xie T. Mining historical issue repositories to heal large-scale online service systems. Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) 2014; 311-22.


