CHEMICAL ENGINEERING TRANSACTIONS

VOL. 46, 2015

A publication of
AIDIC
The Italian Association
of Chemical Engineering
Online at www.aidic.it/cet

Guest Editors: Peiyu Ren, Yancang Li, Huiping Song
Copyright © 2015, AIDIC Servizi S.r.l.,
ISBN 978-88-95608-37-2; **ISSN** 2283-9216

DOI: 10.3303/CET1546022

A Novel Normal Cloud-Based Model for Distributed Transmission and Aggregation of the Information

Ernuan Wang*, Liang Chen

College of Information Science and Engineering, Henan University of Technology, China
wangernuan@163.com

How to efficiently transmit and aggregate the data from multiple computer nodes is a common problem in the distributed system. A novel normal cloud-based model for distributed transmission and aggregation of the information is proposed in this paper, which performs qualitative-to-quantitative conversion on the network load using the normal cloud generator. The virtual cloud algorithms are used to obtain the cloud model for data transmission and aggregation. It also acquires the intended location of the transmitted and aggregated data from analysis of the cloud drops' distribution. Experiments show that the proposed model is reliable and practically helpful and has the ability to achieve dynamic balance of network traffic.

1. Introduction

With rapid advances in the network technology, the computer system becomes distributed and networked, bringing about changes of the way the data is stored, transmitted, distributed and collected.

In the wide-area heterogeneous distributed environment, how to acquire the information from different computers, aggregate and analyze the acquired information effectively is an issue that needs to be solved urgently.

Let U={u_1, u_2, …, u_n} denote the predefined set of indicators, where $u_i > 0$. In the distributed environment, each computer node stores the data of several indicators (a subset of U). Now there is a need to aggregate and analyze all of the data stored in all of the computer nodes based on different indicators. That is, the data of the indicator u_1 stored in all computer nodes need to be aggregated into a central computer and then analyzed based on certain rules. The data of the indicator u_2 stored in all computer nodes need to be aggregated into a central computer and then analyzed based on certain rules as well. The data of other indicators is processed similarly (Li et al. (2007)).

The traditional approaches to this problem is as follows:

All computer nodes should send the collected data of indicators to the central server across the network. The data will be analyzed at the central server based on existing rules, as shown in Fig. 1.

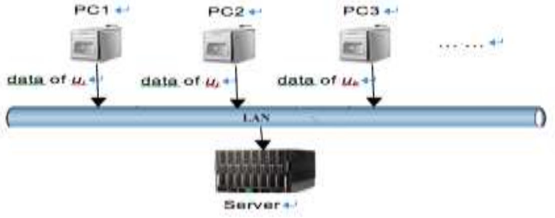

Figure 1: The traditional approaches: Only server analyzed data

Defects of the method:

1) When all computer nodes send all data of indicators to the same central server, many network resources will be consumed, and the local network traffic is large, resulting in imbalanced network traffic. 2) The dependence on the central server is very heavy and goes against the networking rule of avoiding the occurrence of centers. If the central server is down, the entire system will malfunction (Wang (2014)).

Please cite this article as: Wang E.N., Chen L., 2015, A novel normal cloud-based model for distributed transmission and aggregation of the information, Chemical Engineering Transactions, 46, 127-132 DOI:10.3303/CET1546022

2. Scenario studied in this paper

Due to enormous efforts that carriers have devoted to the 3G service, 2G and 3G mobile phones coexist. The 2G and 3G mobile communication signals almost covet every rural and urban corner of the country. The carriers need to perform detailed matching and analysis of the 2G and 3G signals in the TD-SCDMA network, including LAC configuration consistency analysis for GSM external communities in TD, analysis of redundant G external communities, the number of G neighbors more or fewer than communities (macro stations) in TD.

To match and analyze the 2G and 3G signals carefully, the carriers establish computer nodes at different places to collect local 2G and 3G signals and store them into different Excel files (the collected 2G signal is stored into the 2G file, and the collected 3G signal is stored into the 3G file).

Currently, the computer nodes need to send the 2G and 3G Excel files that store collected data to a central server across the network. In the central server, the data is processed then item after item based in data matching and analyzing rules specified by the carriers.

Obviously, the efficiency of this method is low, and much disadvantages in it, which can be known from the front page.

So, to address the problem of this method, the cloud model is introduced in this paper to propose a novel data aggregating model, which is proved to be very effective.

3. Cloud model

The cloud model is a model for qualitative-to-quantitative conversion of a particular structural algorithm. In addition to representing uncertainty in the natural language, the cloud model represents the correlation between randomness and fuzziness, and provides a mutual mapping between qualitativeness and quantitativeness. The cloud model can be used to study the universal laws about uncertainty in the human society by fully integrating fuzziness with randomness. The cloud model relies on random mathematics and fuzzy mathematics, providing an approach to depicting randomness and fuzziness of the language value and the relation between them. Furthermore, the cloud model has the ability to conduct uncertainly conversion between qualitative concept and quantitative description of the language value.

The cloud model offers a new method for investigating the uncertain artificial intelligence, and has been used effectively for knowledge discovery, system evaluation, data mining, decision making analysis and support, intelligent control and network security (Gharaylou (2009)).

3.1 Fundamentals of the cloud model

Let U denote the quantitative domain that are expressed as the exact values, T denote the qualitative concept spatially associated with U, and $X \subseteq U$. If there exists a random number $C_r(x) \in [0,1]$ that shows a stable tendency for $x \in X$, which is known as the membership of x with respect to T, then the numeric field distribution of T's mapping from the domain U to the interval [0,1] is called the cloud. If the domain corresponding to the concept is an n-dimension space, then it can be extended to a n-dimension cloud (ZHANG et al. (2013)).

3.2 Basic numerical characteristics of the cloud model

The cloud consists of drops. There is no sequence among drops. A drop is an implementation of the qualitative concept using the number. A drop alone is negligible, but a large number of drops can represent the characteristics of the qualitative concept overall.

The numerical features of a cloud can be represented using three values: expectation *Ex*, entropy *En* and super-entropy *He.* (ZHANG et al. (2013)). It accomplishes mapping between qualitativeness and quantitativeness and conducts uncertainty conversion between qualitative concept and quantitative representation using specific computer algorithms. Meanwhile, it uncovers the correlation between randomness and fuzziness. The expectation Ex is the point that can best represent the qualitative concept in the numeric field. It is also the best sampling point that can represent the qualitative concept using numbers. It refers to where the peak of the cloud is located in the cloud map. The entropy En is a measure of the uncertainty in the expectation, and denotes range of the numerical field that can be accepted by the qualitative concept, i.e. ambiguity. It is also a measure of uncertainty in the qualitative concept. Usually, the larger the entropy, the more universal the concept. It refers to the width of the cloud in the map. The super-entropy He is a measure of the entropy, indicates how the cloud drops are scattered, represents the randomness in the creation of the drops, and uncovers the correlation between fuzziness and randomness. The larger the super-entropy, the more scattered the drops, and the thicker the cloud in the map (Blahak (2010)).

The cloud map outlines the concept of uncertainty, and makes approximate and flexible conclusions. As we know, a proper level of fuzziness may yield “accuracy”, and desperate pursuit of accuracy may cause “fuzziness”. Consider people’s talks and thinking in their daily lives, where inaccurate terms are usually used. But it neither impedes people’s understanding of the talks nor hinders people from reaching the right conclusion. The cloud model quantifies the language value using three numerical features and this particular algorithm only.

Note that the three numerical features of the cloud model may have more than one dimension to represent multiple properties of a concept. The algorithm can accomplish this by extending the normal generator concatenated twice to a concatenated multidimensional normal generator. The expectation in the cloud model can be extended to an image, a signature, or a segment of the sound. The extended cloud model can provide any number of extended images, extended signatures, extended sounds, or extended numerical codes for this image, signature, or this sound. The extension like this has uncertainty. The extended cloud can represent more extensive and complicated uncertain knowledge.

3.3 Normal cloud

The normal cloud is a type of the symmetric cloud models (Yu et al. (2015)). It is the most fundamental and important cloud model, because the universality of normal distribution has been verified in every branch of the society and natural science.

The normal cloud model generates a quantitative value for the qualitative concept by using the generator which has a particular structure and consists of expectation, entropy and super-entropy to represent uncertainty in the concept. This particular structure relaxes the condition for normal distribution. Instead of accurately determining the membership function, it suffices to construct the expectation function that follows the normal membership distribution. Thus, the normal cloud model is more universal, and has the ability to carry out qualitative-to-quantitative conversion more easily and directly.

Let *U* denote a quantitative domain denoted by a deterministic value, *C* denote a qualitative concept on *U*, and *x* denote a random implementation of the qualitative concept *C*. Consider that x~N(Ex, En'2), En'~N(En,He_2), and the determinacy of *x* to *C* follows the formula below:

$$\mu = e^{-\frac{(x-Ex)^2}{2(En')^2}} \qquad (1)$$

Then, the distribution of x on U is called the normal cloud. Detailed analysis of the universality of the normal cloud in terms of representing uncertain knowledge is performed in [Jackie (2005)].

3.4 3σ in the normal cloud

Each of the drops generated by the forward normal cloud generator makes different contributions to the specific concept. Like the normal distribution, the drops which contributes greatly to the qualitative concept mainly fall within [*Ex* -3 *En*, *Ex* +3 *En*]. And the drops outside [*Ex* -3 *En*, *Ex* +3 *En*] are called the events of small probability. Neglecting them has little impact on the features overall. This is the 3 *En* rule of the forward normal cloud [1,11], equivalent to the 3σ rule of the normal distribution, as shown in Fig. 2.

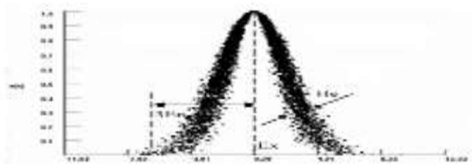

Figure 2: The 3σ rule of the normal distribution

4. The normal cloud based algorithm for aggregating distributed data

4.1 Scenario of the Proposed Algorithm

Consider the carriers' analysis of the 2G and 3G data. The following set of indicators is defined. The weights of these eight indicators concerned are determined based on their importance.

Table 1: Indicators for analysis

Indicators	Weights
Configuration consistency analysis for GSM external communities in TD	0.2
Number of GSM neighboring areas in TD	0.2
Analysis of redundant G external communities	0.1
G neighboring communities in TD	0.1
non-configured neighboring communities in TD	0.1
Parameter consistency analysis for external communities in 3G (UARFCN&CPIZ)	0.1
LAC configuration error analysis for external communities in 3G	0.2

4.2 Implementation method

By following utterly new design ideas, the cloud model is used in this paper to effectively address the problems above.

Changes of the structure: the previous central server is replaced with an ordinary PC, which is responsible for maintaining the set of indicators. It is also responsible for determining which computer node the data of each indicator should be sent to in order to perform data aggregation and analysis.

Details of the method is as follows: for each member u_i of the set U of indicators, the ordinary PC should begin with determining the list $L=\{c_1, c_2, \ldots, c_n\}$ of computer nodes that store the data needed to analyze this indicator. Then, the information (hardware, current load, and the amount of the collected data) about each member of the list L of computer nodes is obtained, as shown in Tab.2. Select a node (denoted by c_k) of the list of computers that is suitable for data aggregation and analysis by using the algorithm in Subsection 3.4. Transmit the relevant data from other members of the list L to c_k to perform analysis of each indicator. For other members of the set U of indicators, send the data to a certain computer node similarly by following the same algorithm to analyze the indicators.

The central ordinary PC obtains the information of each member, as shown in Fig. 3.

If the node PC1 is selected for the analysis of u_i, and PC3 is selected for the analysis of u_j, then the needed data for the analysis of u_i of the other members is transmitted to PC1, and the needed data for the analysis of u_i of the other members is transmitted to PC3, as shown in Fig. 4.

Advantages of this method are as follows:

1) Dynamic traffic balance is achieved across the entire network by using the cloud model-based task assignment algorithm to evenly distribute the network traffic into the network.

2) All of the analysis tasks confronting the previous central server are allocated to different computer nodes in a dynamic and appropriate manner. The performance of the computer nodes is fully exploited to improve the availability of resources for the entire system.

3) The choice of the computer that conducts data aggregation is adapted to the received amount of information and to the variation of loads in each computer node. Thus, the system's ability in adapting to the complicated network is improved.

4) Rather than using the expensive central server, the cheap computer nodes are used to perform data aggregation and analysis, thereby reducing hardware cost of the whole system.

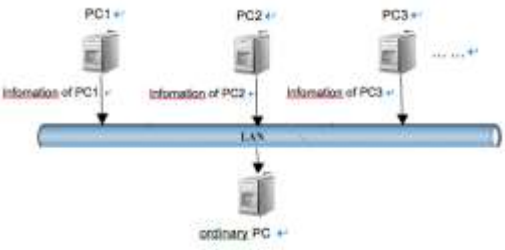

Figure 3: The central ordinary PC obtain the information of each member

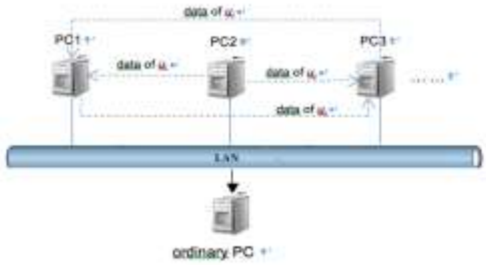

Figure 4: The transmission of data

4.3 Computation of the weight of computing nodes

1) Indicators whose weight needs to be computed

The resources of each computing node are expected to be used optimally to send the data of indicators to the right computer node with high-performance hardware and light loads. To achieve this end, we should consider the hardware and software performance as well as the amount of data about indicators stored in each of the computing nodes.

2) Method for computing the weight

Weighted mean can be computed based on the weight above to obtain the weight (denoted by weight(i)) of the computer nodes.

$$\text{weight}(i) = \text{CPU performance} * 0.2 + \text{memory performance} * 0.2 + \text{hard disk capacity} * 0.2$$
$$+ \text{CPU performance} * 0.1 + \text{memory load} * 0.1 + \text{IO load} * 0.1$$
$$+ \text{the amount of data about the indicators stored in this computer} * 0.2 \qquad (2)$$

4.4 Major algorithm of the proposed model

First, the set of indicators should be defined based on characteristics of the distributed information that need to be aggregated or analyzed and of the cloud model. For each of the indicators in the set, begin with determining the list of computer nodes (denoted by $L=\{c_1, c_2, \ldots, c_n\}$) that store the data needed to analyze this indicator. Follow the following steps to move the needed data to a proper computer node c_k, as shown in Fig. 4: Create a cloud drop (x_i, y_i) for each member c_i of the computer list L. The abscissa x_i of the cloud drop denotes this computer node c_i, the ordinate y_i denotes the weight of the computer node (based on Eq. (2)). The cloud drop is described quantitatively in this way. After all indicators are represented, perform statistical analysis of all cloud drops to determine the optimal data transmission means. Details of the algorithm are given below.

Step 1 Determine the indicators and the corresponding weights

Step 2 Select the indicators in descending order of weights; for a given index, determine the list of computer nodes (denoted by $L=\{c_1, c_2,\ldots, c_n\}$) that have data to be analyzed.

Step 3 For each computer node c_i obtained in Step2, send the data needed to compute the weight of this computer node to the ordinary PC during analysis of the indicators.

Step 4 The ordinary PC analyzes the data from the computer node c_i, and then sets the expectation *Ex*, entropy *En* and super entropy *He*. Generate the cloud drops and the cloud picture based on the 1-D normal cloud algorithm using the cloud generator. According to the situation of the cloud picture, retransmit and distribute relevant data columns stored in the computers of the list L to a certain computer node (denoted by c_j)

Step 5 Aggregate and analyze the data on the computer node c_j, and achieve the results.

Step 6 Follow Steps 2-6 for each indicator.

The pseudo code of the algorithm is as follows:

1) Find an indicator for data analysis get Next Indicator();

2) Determine the list of computer nodes (denoted by $L=\{c_1, c_2,\ldots, c_n\}$) that store relevant data.

3) Compute the weight of each of the N computer nodes based on Eq. (2). Then generate a corresponding 2-D cloud drop (denoted by drop(x_i,y_i)) to record the drop's information.

```
for i= 1 to N do
   weight$_{(i)}$= setWeight( CPU, Mem, Disk, loadCPU,loadMem,loadDisk,digitalCount);
  drop(xi,yi) = generateDrop(max(dataNumber), max(dataNumber) - min(dataNumber), weight$_{(i)}$)
end for;
for i= 1 to N – 1  do  // for the N-1 computer nodes
  Moveto(IP)    // move the data column to the computer node with the specified IP
end for;
```

Some rules for the computation above are as follows:

The expectation E_x is the maximum amount of data stored by all computer nodes pertaining to this indicator.

The value of the entropy E_n is the difference between the maximum and minimum amount of data stored by all computer nodes pertaining to this indicator.

The value of the super entropy He is weight$_{(i)}$, which is computed using Eq.(2).

The cloud picture is created on the central computer using cloud parameters above. Then the part of the picture where the cloud drops are most dense is extracted. The computer node (denoted by c_j) corresponding to this part is the node that should be fed with the data of the indicator. Later on, the data columns of this indicator stored in all other computers are retransmitted and aggregated into the computer node c_j. Matching and analysis of the data on c_j yields the final results.

Table 2: Computation of the weight of computing nodes

First layer indicators	Weights	Second layer indicators	Weights
Hardware	0.5	CPU performance	0.2
		Memory performance	0.2
		Hard disk capacity	0.1
First layer indictors	Weights	Second layer indicators	Weights
Software	0.3	CPU load	0.1
		Memory load	0.1
		IO load	0.1
Data size	0.2	The amount of data about the indicators stored in this computer	0.2

4.5 Result analysis

To verify the effectiveness of the proposed algorithm, several Excel files from carriers that contain 2G and 3G signals collected from the real-world environment are used as the experimental data. A total of 10 simulations are conducted based on the indicators in Tab.3.

Table 3: Final analysis and evaluation results

Indicators	Computing node B	Computing node C	Computing node D
Configuration consistency analysis for GSM external communities in TD	10	0	0
Number of GSM neighboring areas in TD	0	9	1
Analysis of redundant G external communities	0	9	1
G neighboring communities in TD	1	8	1
non-configured neighboring communities in TD	8	0	2
Parameter consistency analysis for external communities in 3G (UARFCN&CPIZ)	0	8	2
LAC configuration error analysis for external communities in 3G	8	1	1

The results in Tab. 3 show that the data needed to analyze the indicator “LAC configuration consistency analysis for GSM external communities in TD” should be transmitted to the computing node B to perform data analysis. Other indicators can be discussed similarly.

5. Conclusions

The normal cloud model is introduced to the distributed data aggregating system to perform data analysis. Inspired by the idea that the quantitative-to-qualitative conversion can be done using the cloud model, the shape of the cloud picture is used to determine the right computing node that should be fed with the data.

References

Blahak U., 2010, Efficient approximation of the incomplete gamma function for use in cloud model applications, Geoscientific Model Development, 3, 329-336, DOI: 10.5194/gmd-3-329-2010

Fan T.S., Zhang Z.Q., 2013, Image scrambling algorithm based on cloud model, Computer Applications, 33, 2497-2500, DOI: 10.3724/sp.j.1087.2013.02497

Gharaylou M., Zawar-Reza P., Farahani M., 2009, A one-dimensional Explicit Time-dependent cloud Model (ETM): Description and validation with a three-dimensional cloud resolving model, Atmospheric Research, 92, 394-401, DOI: 10.1016/j.atmosres.2008.12.008

He W., Wang F.K., 2015, A Hybrid Cloud Model for Cloud Adoption by Multinational Enterprises, Global Information Management, 23, 1-23, DOI: 10.4018/jgim.2015010101

Li D.Y., Du Y., 2007, Qualitative and Quantitative Transform Model-Cloud Model, Artificial Intelligence with Uncertainty, ISBN: 978-1-58488-998-4, 107–151, DOI: 10.1201/9781584889991.ch5

Li G.Z., 2013, Real-coded quantum evolutionary algorithm based on cloud model, Computer Applications, 33, 2550-2552, DOI: 10.3724/sp.j.1087.2013.02550

Li Z.H., Liu C.M., 2014, The Cloud Computing And The GPS's Cloud Model, Green Communications and Networks, 2, 164-170, DOI: 10.2495/gcn130762

Liu Y., Zhang T.W., 2012, P-Order Normal Cloud Model: Walking on the Way between Gaussian and Power Law Distributions, Computer Science, 7664, 467-474, DOI: 10.1007/978-3-642-34481-7_57

Liu Y.T., Li L., Zhang M., 2013, Effectiveness Evaluation of Information Management System Based on Modified Normal Cloud Model, Applied Mechanics and Materials, 411, 231-235, DOI: 10.4028/www.scientific.net/AMM.411-414.231

Shu J, Gao M., Sun L.M., 2012, A Study of Group Mobility Model in Tactical Ad Hoc Network based on Normal Cloud Model, Second International Conference on Instrumentation, Measurement, Computer, Communication and Control, 153-160, DOI: 10.1109/imccc.2012.182

Wang G.Y., Xu C.L., Li D.Y., 2014, Generic normal cloud model, Information Sciences, 280, 1-15, DOI: 10.1016/j.ins.2014.04.051

Wu T., Xiao J., Qin K., Chen Y.X., 2015, Cloud model-based method for range-constrained thresholding, Computers & Electrical Engineering, 42, 33-48, DOI: 10.1016/j.compeleceng.2014.03.016

Xu H.Y., 2015, Normal Cloud Heavy-Tailed Model Research Based on the Semi-Invariantion, Journal of Software Engineering, 9, 276-286, DOI: 10.3923/jse.2015.276.286

Xu Z.H., Shen G., Lin S., 2010, Normal Distribution Data Generating Method Based on Cloud Model, Advanced Materials Research, 171-172, 385-388, DOI: 10.4028/www.scientific.net/AMR.171-172.385

Yang Z.X., Fan Y.F., 2013, Normal Cloud Model Based Reputation Quantification in Trusted Networks, Information Technology Journal, 12, 4523-4528, DOI: 10.3923/itj.2013.4523.4528