Load Evaluation Algorithm of Cloud Database based on Shannon Entropy

Chen Qing, Yong Zhong, Liuming Xiang
Chengdu Institute of Computer Applications
Chinese Academy of Science, Chengdu
China

ABSTRACT: *Due to the two-phase commit protocol, all transactions of DDBS (distributed database system) will roll back if one of distributed nodes was overload, finally make the DDBS difficultly adapting to the Big Data's environment, whose data has the characters of dynamic and randomness. In order to solve this problem, Shannon entropy is proposed to evaluate system's load, using the maximum entropy principle of entropy with the objective function and constraints to balance the load and maximize resource's utilization on the demand of user's QoS. Overloaded node's data will be migrated to other suitable nodes under the guidance of algorithm based on Shannon entropy, and make a step to the further design of Cloud database system. Experimental results show that the load evolution algorithm based on Shannon entropy can evaluate the load in Big Data's environment, avoid single-node bottlenecks, and improve system's performance.*

Keywords: Two-phase Commit Protocol, Big Data, Cloud Database, Shannon Entropy

Received: 17 August 2015, Revised 19 September 2015, Accepted 27 September 2015

1. Introduction

As the representatives of the Sensor systems just as RFID are widely used, global data are growing explosively. These data have characters of complexity, spontaneity and randomness. How to effectively manage these big data has become a big challenge. Database as an effective data management tools are widely used in various industries, but due to the traditional databases are not designed for Big Data environments, making it difficult to adapt to new environment. Designing a new database system—Cloud Database(CloudDB) has become an important thing. At present, the design of Cloud Databases adapting to Big Data environment have two main ideas. One is the database based on key/value structural, which belongs to non-relational databases, and the database system can easily sale out due to no correlation between the data, this design greatly enhanced the level of scalability, but because of lacking of transaction management, it can only be used in a particular scene, such as Social Networking Services(SNS). The other idea of design is based on traditional distributed relational database (DDBS), because this system retains the management of transaction, how to improve the ability of scale-out to adapt Big Data environment is the difficulty in designing.

Microsoft's SQL Azure [1,2] uses a idea of avoiding the 2PC protocol to design Cloud Database, and the transaction limited to a signal node. ElasTras [3, 4] supports elastic transactions, but limits transactional to a sub-region to support mini-transaction [5]. Deuteronomy [6] splits database storage engine into transactional components (TC) and data components (DC). TC provides logical concurrency control and undo/ redo recovery, and need not to know the physical location of data; DC is

responsible for caching data, and knowing data's physical organization, including access methods, operating interface, without knowing the management of transaction. Its main feature is supporting across-nodes' transactional operation by responsibility of DC, and whose TC is responsible for the transaction of the submission and ensures data changing. CloudTPS [7] splits transaction manager into multiple sub-transaction manager to ensure that each sub-transaction manager accessing a single data node. Calvin [8]'s core idea is operating on a global transaction sequence defined schedule, trying to avoid a single node of failure and conflicted operations of distributed transaction.

The thesis based on the design of traditional distributed relational database. In order to avoid a single-node's failure which will cause rollbacks of all the transactions, we propose a new transaction load evaluation algorithm which uses entropy optimization model for evaluating the load of the nodes participating in the transaction, and assess the candidate set of nodes to avoid choosing overload node which will be bottle-neck in the system's operation. Using Load Evaluation Algorithm, we can automatically adjust system's load through migrating overload node's data, enhance the ability of scale out on the other hand, and make a step for further designing of CloudDB.

2. Load Evolution Algorithm

The concept of entropy originated in classical thermodynamics, the entropy used to measure the degree of disorder. Shannon used entropy to define a randomness degree of a discrete random variable [9], and gave the name Shannon entropy or Information entropy.

Literature [10, 11] uses deformation of Shannon entropy to study uncertain information's measurement of rough sets and rough relational database. In 1968, Zadeh first proposed the concept of fuzzy entropy, and was used to quantify the fuzziness of fuzzy sets .In 1972 de Luca and Termini [12] introduced axiomatic construction method of fuzzy entropy, and gave a fuzzy distance measure, similarity measure relations of the fuzzy set. Literature [13] gave new fuzzy entropy and effectively applied to fuzzy rough sets based on the division of fuzziness. Liang Jiye had proposed information entropy has a feature of complementary, and gave its conditional entropy and mutual information, pointing out that it is one kind of fuzzy entropy, applying it to measure the rough sets and fuzzy rough classification [14]. Qian Yuhua and others introduced the non-complete combination of the concept of entropy in information systems, the information's complementary function had possible knowledge content features, and was used to measure incomplete information of uncertain system [15]. While the Big Data environment has characters of randomness and uncertainly, so using Shannon entropy which has ability of measuring uncertain is quite well, we proposed ShannonEval algorithm based on Shannon entropy for evaluating the load of distributed relational database. In order to enhance the database's ability to adapt to a dynamic environment, and further provide theoretical support for the design of the CloudDB.

The traditional evaluation indicators are mostly using static or predictable physical indicators, such as CPU's computing capacity, storage capacity, network bandwidth, etc, but in Big Data which has dynamic environments, these indicators have characters of uncertainty and non-iconic. Therefore, these static indicators were difficult reflecting actual service capabilities of the node. In order to describe the dynamic state of the transaction load and the actual available capacity, the thesis takes the amount of the transaction request, the node's service capabilities and service node's strength as indicators, using the monitoring component to monitor, track and count dynamic information of resources (such as the average number of transaction requests, the node average serving time), and uses curve fitting technique to establish the probability distribution model of service capabilities and resources' amounts of requests, finally uses the maximum entropy principle and the dynamic properties of resources as its constraints to obtain resource assessment of the objective function.

2.1 Evalution Indicators

In order to describe the dynamic load state and the actual resources available capacity, we use following three parameters as dynamic indicators, assuming virtual resources included in the set U of n resources, namely $U = \{U1, U2, ..., Un\}$:

(1) The amount of the resource's request r: unit time of resource serving request is received, the average number of resource nodes. If Ui $(1 \leqslant i \leqslant n)$: the amount of resource's requests is r_i $(1 \leq i \leq n)$, then the amount of resource's requests U is:

$$r_n = \sum_{i=1}^{n} r_i .$$

(2) The service capabilities of resources h: The greater the average time to complete the unit number of service's requests h is, and the stronger resource's service capability and the higher resource's prices are. If the request for the amount of resource's nodes is hi $(1 \le i \le n)$, then then resource's capacity of U is: $h_n = n \Big/ \sum_{i=1}^{n} \frac{1}{h_i}$.

(3) The strength of resource's service q: the ratio of the average time of completing a service's request and the average time interval of the service's request, and $q = \frac{r}{C \cdot h}$, In which, C is resource's parallel service 's capabilities. If parallel service's capabilities of resource node Ui $(1 \le i \le n)$ is $Ci(1 \le i \le n)$, and parallel service's capabilities of resource's node Ui is: $C = C_1 + C_2 + \ldots C_n$.

Strength of resource's service reflects the relationship between resources load pressure situation and available capacity. If strength of resource's service tends to 1ÿindicates strength of resource's service tend to full load. Strength of resource's service U is: $q_N = r_N \Big/ \left(h_N \cdot \sum_{i=1}^{n} C_i \right)$.

2.2 Resource Assessment Model

We use monitoring component to monitor, track and count dynamic information of resources (such as the average number of service requests, average serving time of resources), establish the probability distribution model of amount of requesting resources and service's capabilities by curve-fitting techniques, use probability distribution model and maximum entropy principle resource asset the resource and establish the objective function, and use dynamic properties of resources as its constraints.

By studying on the prediction on the input events of cloud computing and grid computing, we knew input events (resource request quantity r) was mostly Poisson distribution, and the probability distribution is:

$$p(X = r) = \frac{\lambda^n}{r!} e^{-\lambda}, r = 0, 1, \ldots \quad (1)$$

According to the probability distribution function of the resource's request, obtain the value:

$$S = \lambda - \lambda \log \lambda + \sum_{k=0}^{\infty} \log \Gamma(k+1) p(k), \lambda > 0, \quad k = 0, 1, \ldots \quad (2)$$

Through its entropy and principle of maximum entropy seeking the maximum entropy [16], derived (3), and using the amount of resource's requests and resource service's capabilities as its constraints, derived (6); Also according to the principle of minimum entropy, i.e. when entropy is minimum, the system can be maximized, therefore, using the objective function of user selecting resources proposed in the literature [19] $F_u = \alpha \cdot P(r,h) + \beta \cdot W_s(r,h)$ and objective function of providers providing resources $F_p = h \cdot P(r,h) + K \cdot V(r,h)$ (α, β are adjustment factors reflecting the user's preference for the costs and the deadline, P is the price of resource, Ws is sum of resources' serving request waiting time in queue and the actual service time, K is cost of idling resource to be paid per unit time, $V(r, h)$ is the resource idling rateÿderived (7) as a constraint ,and combining with other constraints obtain the user's optimal QoS in current state of the system. In this case, we call this Entropy optimization model, which combines Maximum entropy principle and Minimum entropy principle.

$$\max S_n(P) = -\sum_{i=1}^{n} p_i \ln p_i \quad (3)$$

$$s \cdot t \cdot \sum_{i=1}^{n} P_i = 1 \quad (4)$$

$$p_i > 0, i = 1, \ldots, n \quad (5)$$

$$\sum_{i=1}^{n} P_i q_{Nj}(r_i) = E[q_{Nj}], j = 1, 2, ..., m \quad (6)$$

$$| d_0 F_U + d_0 F_P | > d_0 F'_p + \sum_{k=1}^{n} F_{uk} \quad (7)$$

In which, n is the number of resources, r_i is the value of the possible states of the resource request, qNj (.) is Computing function of the resource, F'_p is part of the resource dynamics entropy, $\sum_{k=1}^{n} F_{uk}$ is relevant parts of the user requesting the entropy change. The process of the algorithm is as follows:

Get the Distribution of r_i by Fitting Curves;
IF the Distribution of r_i is Poisson
THEN Calculate the Probability of r_i, the Entropy of S as (2);
ELSE
Calculate the Entropy of S by other Distribution of r_i;
ENDIF
Calculate the mean, variance of r_i and q_i, as (4) to (6);
Get (7) include F_u, F_p;
Calculate the Max of Entropy S_{max} by (4);
S_{max} Subject to the Constraint as (4) to (7).

3. Load Evolution Experiments

In order to verify the proposed Load Evaluation Algorithm, we use CloudSim3.0 to simulate, Monitor resources on the Linux system. Using shell command vmstat to monitor CPU's utilization, Virtual memory using situation and count the whole system's usage; also using iostat command can monitor disk and I/O usage. The overall state of the resource is dynamic, these information need timing statisticsÿto provide a basis for analysis and evaluation of resources.

In experiment, we compare ShannonEval with AEMJE [17](availability degree enhanced model for job execution)and other two algorithm as followsÿ

(1) HA-JES(highly available job execution service) [18]: According to the user's requirements to schedule se resources which will be in task, and schedule the task to one or more set of resources in order to improve resource's utilization which has low utilization.

(2) ACT (availability check technique) [19]: One kind of usability inspection techniques, used to check and update the status of the resources required, the necessary resources are available when the state began to schedule tasks.

Three techniques are used Min-Min scheduling algorithm to schedule tasks, each of which is using 40 times operation's results to compare with. Their performance evaluation indicators are as follows:

(1) Makespan: Reflecting the task completion time , and is a measure of an important indicator of user QoS, depending on the maximum completion time of compute nodes, the equation calculated as follows:

$$\text{Makespan}=\max \{CT\_node_i\}. \quad (8)$$

(2)System utilization: The proportion of the tasking number and all nodes in large system, which means that more idling nodes used and more efficient using of resources, and thus the better balancing of system's load, the equation calculated as follows:

Utilization = exe_node/all_node. (9)

The main indicators of users' QoS are the completion time and costs, in which the costs has been reflected in the objective function F_u, so the experiment only validates performance at completion time; Also, the nodes which do not meet the user QoS and have low availability, will be improved by data migrating; because performance of the load balancing corresponds to system's utilization, so the experiment only need to validate task's completion time and the performance of system's utilization. And experimental parameters are set as follows:

(1) Task number sets 500, and the node number grows from 100 to 1000; node number sets 500, and task number grows from 100 to 1000.

(2) Task length sets from 1000 to 2000, and the units MI; Amount of data transfer sets form 1000 to 2000; Amount of storage sets from 1000 to 2000.

(3) Processing speed of node sets from 10 to 200, and the units MIPS; Node's capacity of storage sets from 10000 to 20000.

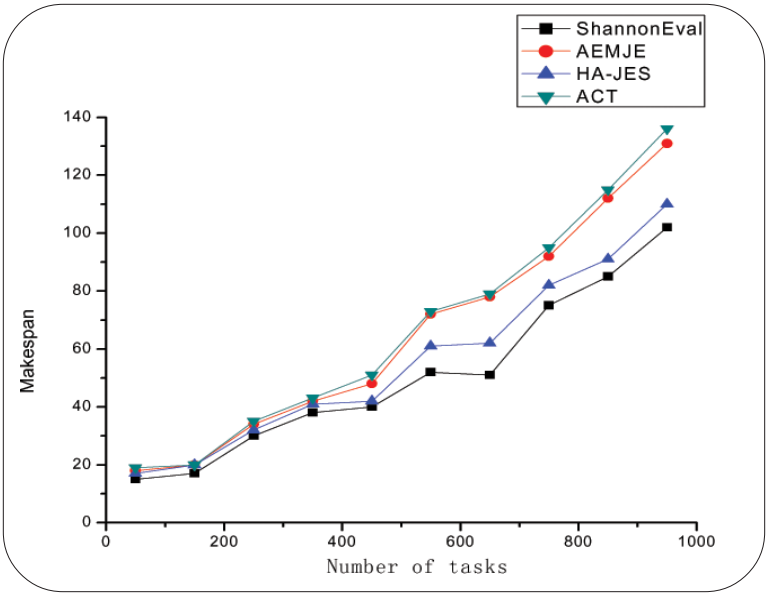

Figure 1. The Performance of Makespan

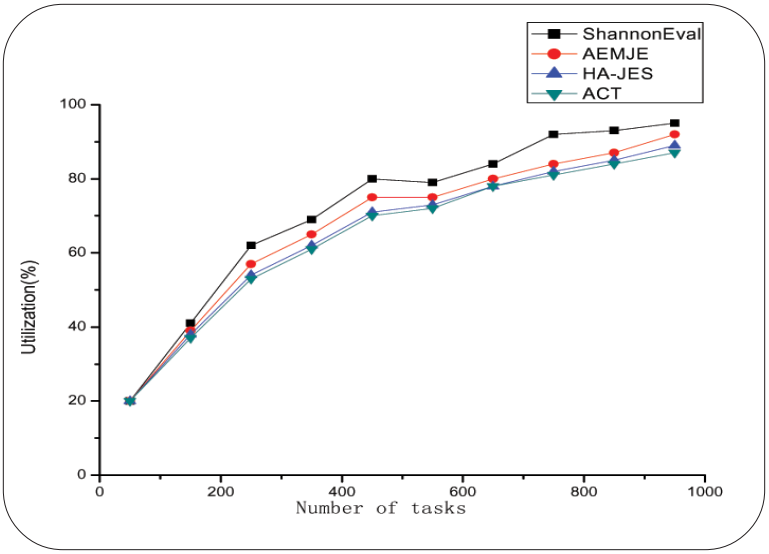

Figure 2. The Performance of Utilization

When number of tasks are differentÿthe performance which comparing with other three are just as “Fig. 1” and “Fig. 2.” We can find that ShannonEval performances better than other three in Makespan with increasing number of tasks, and plays best when tasks’ number are about 670. ACT performances similar to AEMJE, and the HA-JES plays the second best in Makespan. In performance of utilization, ACT, AEMJE and HA-JES play not far-off. ShannonlEval overall dominate the other three, and plays best when tasks’ number are about 450 and 780. And the reason is that ShannonEval using Shannon entropy to evaluate load under the premise of meeting user’s QoS.

4. Conclusion

In Big Data era which data changes dynamically and randomly, spawned a cloud database development and meanwhile a variety of new database were designing. Because of weaknesses of the traditional distributed relational database, we proposed an algorithm based on Shannon entropy. The algorithm filtered optimal collection of nodes under the premise of meeting user’s QoS and maximizing the system’s utilization, made guidance for dynamic data migration to achieve load balancing, and finally lay the foundation for Cloud Database. Through the experiments, the algorithm based on Shannon entropy performances better than the other load balancing algorithms, it has an ability of evaluating the load of cloud database in Big Data’ environment.

References

[1] Campbell, D. G., Kakivaya, G.. Ellis, N. (2010). Extreme scale with full SQL language support in Microsoft SQL azure, *In*: Proceedings of the SIGMOD. New York: ACM Press, 1021-1023.

[2] Bernstein, P. A., Cseri, I. et al. (2011). Adapting Microsoft SQL server for cloud computing, *In*: Proceedings of the ICDE. IEEE, 1255-1263.

[3] Das, S., Agarwal, S., Agrawal, D. (2010). ElasTraS: An elastic, scalable, and self managing transactional database for the cloud, *Technical Report* 2010-4. Santa Barbara University of California.

[4] Aguilera, M. K., et al. (2007). Sinfonia:A new paradigm for building scalable distributed systems, *In*: Proceedings of the SOSP. New York: ACM Press.

[5] Levandoski, J. J., Lomet, D., Mokbel, M. F., Zhao, K. K. (2011). Deuteronomy: Transaction support for cloud data, *In*: Proceedings of the CIDR. USA, 123-133.

[6] Wei, Z., Pierre, G., Chi, C. H. (2010). CloudTPS: Scalable transaction for Web applications in the cloud, *Technical Report* IR-CS-053. Amsterdam: Vrije Universiteit.

[7] Thomson, A., Diamond, T., Weng, S. C. (2012). Calvin: Fast distri-buted transactions for partitioned database systems, *In*: Proceedings of the SIGMOD. New York: ACM Press.

[8] Shannon, C. E. (1948). The mathematical theory of communication, *Bell Syst Technol Journal* 27 (3,4) 373-423, 623-656.

[9] Beaubouef, A. T., Petry, F. E. (2000). Fuzzy rough set techniques for uncertainty processing in a relational database, *Int Journal Intel Syst*, 15 (5), 389-424.

[10] Beaubouef, A. T., Petry, F. E., Arora, G. (1998). Information theoretic measures of uncertainty for rough sets and rough rela-tional databases, *Inform Sciences*, 109 (1-4), 185-195.

[11] De Luca, A., Termini, S. (1972). A definition of nonprobilistic en-tropy in the setting of fuzzy sets theory, *Inform Contr*, 20, 301-312.

[12] Mi, J. S., Leung, Y., Wu, W. Z. (2005). An uncertainty measure in partition based fuzzy rough sets, *Int Journal Gen Syst*, 34 (1), 77-90.

[13] Liang, J. Y., Chin, K. S., Dang, C.Y. et al. (2002). A new method for measuring uncertainty and fuzziness in rough set theory, *Int J Gen Syst*, . 31(4), 331-342.

[14] Qian, Y. H., Liang, J. Y. (2005). Combination entropy and combination granulation in incomplete information system, *Lect Notes Artif Intel*, 4062, 184-190.

[15] Hu, Z. J. (2010). The resource availability evaluation in service grid environment for QoS, [Ph.D. Thesis]. Changsha: Central South University, (in Chinese with English abstract).

[16] Zuo, L.Y., Cao, Z. B., Dong, S. B. (2005). Virtual resource evaluation model based on entropy optimized and dynamic weighted in cloud computing, *Journal of Software*, 24 (8), 1937-1946 (in Chinese).

[17] Buyya, R., Murshed, M. (2002). Gridsim: A toolkit for the mode-ling and simulation of distributed resource management and scheduling for grid computing, Concurrency and Computation: *Practice and Experience*, 14 (12), 1175 -1220.

[18] Düntsch, I., Gediga, G. (1998). Uncertainty measures of rough set prediction, *Artif Intel*, 106, 283-297

[19] Hu, Z. J. (2010). The resource availability evaluation in service grid environment for QoS, [Ph.D. Thesis]. Changsha: Central South University.

[20] Iosup, A., Jan, M., Sonmez, O. O., Epema, D. H. J. (2007). The characteristics and the performance of groups of jobs in grids, *Lecture Notes on Computer Science*, 4641 (8), 382-393.