Asian Journal of Computer Science and Technology
ISSN: 2249-0701 Vol. 6 No. 2, 2017, pp.23-26

A Prefetching Technique Using HMM Forward and Backward Chaining for the DFS in Cloud

V.Thilaganga[1], M.Karthika[2] and M.Maha Lakshmi[3]
[1&3]Research Sscholar, [2]Assistant Professor
MCA Department, NMSSVN College, Madurai, Tamil Nadu, India
E-Mail: gangamahi92@gmail.com, mkpartha@yahoo.com
m.lakshmimahi@gmail.com

Abstract - **A general class of temporal probabilistic model have recently developed, which extends the Forward, Backward and Viterbi algorithm for hidden Markov models. The HMM (Midden Markov Model) is a probabilistic model of the joint probability of a collection of random variables with both observations and states. The algorithm is based on shrinking the state space of the HMM noticeably using such chains. The states through which the world passes are hidden, or unobserved. However, at each point in time also gets an observation that in some way reflects on the current state of the world. The Cloud Computing is a big deal for three reasons: It does not need any effort on Clients part to maintain or manage. It's effectively infinite in size, so clients don't need to worry about it running out of capacity. User can access cloud-based applications and services from anywhere all you need is a device with an Internet connection. In this Cloud Computing used the Distributed File Systems (DFS) for sharing and allocating the data during dynamic process .Those process are using some Prediction algorithms here using HMM Forward and Backward Chain. In this paper represents, Cloud Storage Server can Share the data among with the multiple users, using two prediction algorithms such as forward and backward chain in HMM.**
***Keywords*: Distributed FileSystem, Hidden Markov Model, Storage Server**

I. INTRODUCTION

In a cloud computing application, some of the essential technology is distributed file systems. A file system is a file controlling activity such as organization, storing, retrieval, naming, sharing and protection of file. A distributed file system for cloud is a file system that allows several clients to have admission to data and supports applications. The assimilation of distributed computing for search engines, multimedia websites and data intensive applications has brought about the generation of data at unprecedented speed[3]. According to the EMC_IDC Digital Universe, the data volume is created, replicated and consumed in states and possibly will increase twice every three years over the end of this decade.

There are quite a lot of methods to share documents in a distributed architecture: all solution need be fit for a certain category of application, contingent on how complex the application. In fact, the distributed file system services multiple I/O devices by striping file data across the I/O nodes, and uses high aggregate bandwidth to meet the growing I/O requirements of distributed and parallel scientific applications[4][5][6][7][8]. In a remote file system access, the distributed file systems processes the mathematically and geographically, the network delay is becoming the majestic factor.

The portable device normally have partial processing power, battery life and storage, but cloud computing provides attraction of infinite computing resources. For combining the mobile devices and cloud computing to create a new infrastructure, the mobile cloud computing research field emerged [1]. To perform I/O optimization tactics, the I/O events can reveal from the I/O disk tracks and that can offers critical information, certain prefetching techniques have been proposed in succession to read the data on the disk in advance after analyzing disk I/O traces [2] [9]. This category of prefetching is used for limited file system.

The Hidden Markov Model is a numerical and determinate set of states. In this model, the state is not directly visible, but the output, dependent on the state is visible. It’s a machine learning method and can observe output from states, not the states themselves. Create one Hidden Markov Model using the trained data. The training data set is divided into vector of 5 values each, the first 4 values of the sector are treated as the input, and the fifth value is treaded as the output then depending on the log-likelihood values of the input (as obtained from the HMM), the training data is divided into clusters [10]. In Hidden Markov Model the primary problems are evaluation, decoding and learning. Previously solving the applications the said core problems must be solved depending on the real world applications.

The proposed mechanism analyze the client I/O details for storage server that can predict and find the future I/O process in advance and then forwards to relevant client machine for the future potential usage. Here the prefetching technique is using HMM Forward and Backward algorithms. In this paper the proposed technique is combined with distributed file systems and Hidden Markov Model in cloud computing.

II. CLOUD COMPUTING

Cloud Computing refers to manipulating, configuring, and accessing the applications online. It offers online data storage, infrastructure and application. Simply put, cloud computing is the delivery of computing services servers, storage, databases, networking, software, analytics and more over the Internet (“the cloud”). Companies offering these computing services are called cloud providers and typically charge for cloud computing services based on usage, similar to how you are billed for water or electricity at home. Cloud computing is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. Cloud computing is comparable to grid, a type of computing where unused processing cycles of all computers in a network are harnesses to solve problems too intensive for any stand-alone machine. Also denoted to as a network cloud. In telecommunications, a cloud refers to a public or semi-public space on transmission lines that exists between the end points of a transmission. Data that is transmitted across a WAN enters the network from one end point using a standard protocol suite such as Frame Relay and then enters the network cloud where it shares space with other data transmissions. The data emerges from the cloud where it may be encapsulated, translated and transported in myriad ways in the same format as when it entered the cloud. A network cloud exists because when data is transmitted across a packet switched network in a packet, no two packets will necessarily follow the same physical path. The unpredictable area that the data enters before it is received is the cloud. A place where clients can access apps and services, and where client’s data can be stored securely.

III. DISTRIBUTED FILE SYSTEM FOR CLOUD

Cloud uses distributed file system for storage purpose. If one of the storage resource fails, then it can be extracted from another one which makes cloud computing more reliable Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards.

Distributed file system allows to multiple clients to access the data and support operations (create, delete, modify, read, write) on that data. Every data file may be segregated into several parts named chunks. Every chunk might be stored on dissimilar remote machines, facilitating the parallel performance of applications. Typically, documents are kept in files in a hierarchical tree, wherever the nodes denote directories. There are numerous ways to share records in a distributed architecture: each solution must be appropriate for a certain type of application, conditional on how composite the application is. Meanwhile, the safety of the System essential is ensured.

Confidentiality, availability and integrity are the foremost keys designed for a secure system. Distributed file system allows several big, medium, and small originalities to store and access their isolated data as they ensure local data, enabling the use of variable resources.

In a cloud computing environment, failure is the norm and chunk servers may be upgraded, replaced, and added to the system. Documents can be enthusiastically created, deleted, and appended. Those indications to load imbalance in a distributed file system, meaning that the file chunks are not distributed equitably between the servers.

IV. HIDDEN MARKOV MODEL (HMM)

The Hidden Markov Model is a restricted set of states, each one of which is associated with a (commonly multidimensional) possibility distribution. A hidden Markov model can be considered a broad view of a mixture model someplace the hidden variables or dormant variables, which control the mixture constituent to be designated for each observation, are related over a Markov process somewhat than self-determining of each other. Recently, hidden Markov models have been comprehensive to pairwise Markov models and threesome Markov models which permit thought of more difficult data structures and the forming of non-stationary data. A Hidden Markov Model is unique in which perceive a series of emissions, but do not recognize the series of states the model pass away over to generate the emissions. Evaluates of Hidden Markov Models seek to recuperate of states from the detected data. Common HMM Types: major one is Ergodic (completely connected).Every single state of model can be touched in a single step from all other state of the model. Next one is Bakiss (left-right).Such as time increases, states keep from left to right ensure with an HMM, the forward algorithm, backward algorithm, the forward-backward algorithm and the Viterbi algorithms are used**.** Around are situated three core problems in hmm**.** Three problems must be explained for HMMs to be valuable in real-world applications, Evaluation, Decoding and Learning. HMM Incentive is Real-world has constructions and processes which have or yield recognizable outputs.

V. IMPLEMENTATION

Compare the HMM Forward, Backward and Viterbi Algorithm is the best way for find the shortest path without knowing the inside data process. In this Algorithms are used to training for the data that are accessed and processed with frequently. So we get trained data that is used for finding the shortest path with values.

A.Forward Algorithm

The Forward Algorithm is a recursive algorithm for calculating $\alpha_t(i)$ for the observation sequence of increasing length t .

1. Initialization

$$\alpha_1(i) = p_i\ b_i(o(1))\ ,\ i = 1, \ldots, N$$

2.Recursion

$$\alpha_{t+1}(i) = [\textstyle\sum_{j-1}^{N} \alpha_t(j)\,\alpha_{ji}]\, b_i\ (O(t+1))$$

here $i = 1, \ldots, N$, $t = 1, \ldots, T-1$

3.Termination

$$P(O(1)\,O(2) \ldots O(T)) = \textstyle\sum_{j-1}^{N} \alpha_T(j)$$

Use the forward algorithm to calculate the probability of a T long observation sequence. An each of the y is one of the observable set. Intermediate probabilities (α's) are calculated recursively by first calculating α for all states at t=1.

B. Backward Algorithm

The Backward Algorithm calculates recursively backward variables going backward along the observation sequence.

1.Initialization

$$\beta_T(i) = 1\ ,\ i = 1, \ldots, N$$

According to the above definition, $\beta_T(i)$ does not exist. This is a formal extension of the below recursion to $t = T$.

2.Recursion

$$\beta_T(i) = \textstyle\sum_{j-1}^{N} \alpha_t b_j\ (O(t+1))\ \ \beta_{t+1}(j)$$

here $i = 1, \ldots, N$, $t = T-1,\ T-2, \ldots, 1$

3.Termination

$$P(O(1)\,O(2) \ldots O(T)) = \textstyle\sum_{j-1}^{N} p_j\ b_j\ (O(1)\ \beta_1(j)$$

Obviously both Forward and Backward algorithms must give the same results for total probabilities $P(O) = P(o(1), o(2), \ldots, o(T))$.

In this forward and backward are calculating and finding the best path value is equal 0.000804. Which means all the HMM algorithms brings the same value using matrix calculations. In HMM chaining algorithms are finding best solution path with values. It is Combined and worked with Distributed File System for Cloud and bring a best solution for the clients and servers.

The HMM combined with Distributed File Systems Prefetching Technique algorithms are used to fetch the client's data proactively and forward into the storage server. In previously using this prefetching techniques are chaotic time series and regression prediction algorithms. In this paper represents replace the prefetching algorithms such as Forward and Backward algorithms in Hidden Markov Model. It is mostly find the clients data as previously and forward into relevant storage server. The main advantage for using HMM forward and backward chain the data prefetching process and data should be trained and will forward to the nearest storage server as soon as possible. So the prefetching technique processed will take quick time with trained data. Here include the Values for founded path values HMM and DFS and also include the previous Prediction algorithms are a founded value that is below Figure Table I.

TABLE 1 EVALUATION OF PREVIOUS METHODS

Learning algorithm	**Results**
Wang and Mendel [11]	0.091
Kim and Kim [12]	0.026
Liner Predictive Model [11]	0.55
ANFIS [13]	0.007
Hidden Markov Model + Neural Nets	0.0017
Hidden Markov Model + DFS	0.5153
Hidden Markov Model + DFS	0.0315
Hidden Markov Model + DFS	0.003843
Hidden Markov Model + DFS	0.000804

VI. CONCLUSION

The Hidden Markov Model Froward and Backward Algorithm compute probability much more efficiently than the naïve approach, which very rapidly ends up in combinational explosion [11]. It can offer the likelihood of a given emission or observation at each position in the sequence of observations. In this proposed mechanism is, implemented and evaluated in data prefetching Distributed File System for Cloud, which the client engines can collect relevant data proactively through by the loading server in a cloud environs.

The loading servers are capable to analyze and predict the client I/O process and then they proactively push data into the relevant client machines for adequate client's future applications requests. The purpose of forward I/O data and about the client machine information's are piggybacked and then transferred to corresponding storage server from the client nodes [12].

The current implementation of proposed data prefetching process in Distributed File System for cloud using Hidden Markov Model Froward and backward prediction chains for proactively fetch the clients I/O events. Using HMM forward and backward chain the data is trained and growths the learning process [13]. There are different workloads bang up-to-date in the system by client, categorizing block access patterns from the block I/O events are sketched and bring about by several workloads with using HMM Viterbi algorithm.

REFERENCES

[1] M. S. Obaidat. "*QoS-Guaranteed Bandwidth Shifting and Re-distribution in Mobile Cloud Environment*". IEEE Transac- tions on Cloud Computing, Vol.2,pp.181–193, DOI:10.1109/TCC.2013.19, April-June 2014.

[2] E. Shriver, C. Small, and K. A. Smith. *Why does file system prefetching work?* In Proceedings of the USENIX Annual Technical Conference (ATC '99), USENIX Association, 1999.

[3] Jianwei Liao, Francois Trahay, Guoqiang Xiao, Li Li and Yutaka Ishikawa Member, "*Performing Initiative Data Prefetching in Distributed File Systems for Cloud Computing*".

[4] J. Gantz and D. Reinsel. "*The Digital Universe in 2020: Big Data, Bigger Digital Shadows, Biggest Growth in the Far East-United States*". http://www.emc.com/collateral/analyst- reports/idc-digital-universe-united-states.pdf [Accessed on Oct. 2013], 2013.

[5] J. Kunkel and T. Ludwig, "*Performance Evaluation of the PVFS2 Architecture"*, In Proceedings of 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing, PDP '07, 2007.

[6] N. Nieuwejaar and D. Kotz. "*The galley parallel file system*". Parallel Computing, 23(4-5) pp. 447–476, 1997.

[7] X. Ding, S. Jiang, F. Chen, K. Davis, and X. Zhang." *DiskSeen: Exploiting Disk Layout and Access History to Enhance I/O Prefetch*". In Proceedings of In Proceedings of USENIX Annual Technical Conference (ATC '07), USENIX, 2007.

[8] J. Stribling, Y. Sovran, I. Zhang and R. Morris *et al*. Flexible, "*wide-area storage for distributed systems with WheelFS*". In Proceedings of the 6th USENIX symposium on Networked systems design and implementation (NSDI'09), USENIX Association, pp. 43–58, 2009.

[9] S. Jiang, X. Ding, Y. Xu, and K. Davis. "*A Prefetching Scheme Exploiting both Data Layout and Access History on Disk*". ACM Transaction on Storage Vol.9 No.3, Article 10, 23 pages, 2013.

[10] Saurabh Bhardwaj, Smriti Srivastava, Member, IEEE, Vaishnavi S., and J.R.P Gupta, "*Chaotic Time Series Prediction Using Combination of Hidden Markov Model and Neural Nets*".

[11] D. Kim and C. Kim," *Forecasting Time Series with Genetic Fuzzy Predictor ensemble*," IEEE Trans. Fuzzy Syst., Vol. 5, pp. 523-535, Nov.1991.

[12] L.X. Wang and J.M. Mendal , " *Generating Fuzzy rules by learning from Examples* ," IEEE Trans. Syst., Man, Cybern., Vol. 22, pp. 1414-1427, Nov.1992.

[13] J.S.R.Jang, "*ANFIS:Adaptive Network Based Fuzzy Inference System*," IEEE Trans. Syst., Man, Cybern., Vol. 23, pp. 51-630, Nov.1993.