A Prefetching Technique using HMM Forward Chaining for the DFS in Cloud

V.Thilaganga, Dr.M.Karthika, Dr.X.Josphin Jasiline Anitha
Research scholar, Assistant professor, HOD
MCA Department, NMSSVN College

***Abstract* - In recent applications of Distributed File System for cloud have managed in providing high security. The Hidden Markov Model (HMM) has played a vital role in prediction analysis in various areas. Hidden Markov Model and Distributed File System(DFS) applications have been in a variety of problems in cloud. The HMM includes Evaluation, Decoding and Learning. The HMM problems can be used to solve various sequences analysis problems, such as forward, backward and Viterbi algorithms. Cloud computing synchronizes the operation of all systems, with techniques such as data center networking, the map-reduce framework with supports data intensive computing applications in parallel and distributed systems that provide lively resource allocation, permittingvarious operating system to exists on the same physical server. So Distributed File System provides more reliable services. The paper represents client and server process, which is a storage server, can directly pre-fetch the data from client machines. The prefetching techniques, uses Hidden Markov Model Forward chain and work Distributed File System for cloud. The prefetching techniques can be implemented to analyze the I/O from the client machine and then send the prefetching data to the relevant machine proactively. Finally, the storage server can forward the prefetching data, in prior to the request from the client machine.**

***Keywords* - Distributed File System, Hidden Markov Model, Storage Server.**

I. INTRODUCTION

In a cloud computing application, one of the core technologies is distributed file systems. A file system is a file management activity such as organization, storing, retrieval, naming, sharing and protection of file. A distributed file system for cloud is a file system that toleratesnumerous 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[1]. According to the EMC_IDC Digital Universe, the data amount is created, replicated and consumed in states and may increase double every three years through the end of this decade.

There are several methods to share documents in a distributed architecture: all solution must be fit for a certain type of application, contingent on how complex the application. In fact, the distributed file system employs 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[2][3][4][5][6]. In a remote file system access, the distributed file systems measures the numerically and geographically, the network delay is becoming the majestic factor.

The mobile device generally have limited processing power, battery life and storage, but cloud computing provides temptation of infinite computing resources. For combining the mobile devices and cloud computing to create a new infrastructure, the mobile cloud computing research field emerged [12].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 [7][9].This type of prefetching is used for local file system.

The Hidden Markov Model is a statistical and finite 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 [13].In Hidden Markov Model the core problems are evaluation, decoding and learning. Before 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. The prefetching technique is Hidden Markov Model algorithm. In this paper the proposed technique is combined with distributed file systems and Hidden Markov Model for cloud computing.

II. CLOUD COMPUTING

The term Cloud refers to a Network or Internet. That Cloud is something, which is current at isolated location. Cloud can provide services over network, on public networks or on private networks, Widespread Area Network, Limited Area Network andSimulated Private Network.Applications like as e-mail, mesh conferencing, client relationship management (CRM), all run in cloud. Cloud figuring is, which consist of virtualization, scattered computing, networking, software and network services.

Cloud Figuringdenotes in manipulating, constructing, and retrieving the applications online. The aforementioned offers online data storage, substructure and application. Server supports to calculate the resource sharing and proposition other services like as source allocation and de allocation, observing resources, safety, etc.

PROCESS OF CLOUD COMPUTING

Cloud computing abstracts the details of system implementation from users and developers. Applications run on physical systems that aren't specified, data is stored in locations that are unknown, administration of systems is outsourced to access by users is ubiquitous. Cloud figuring virtualizes organizations by merging and sharing resources. Systems and storage can be provisioned as needed from a centralized infrastructure, costs are assessed on a metered basis, multi-tenancy enabled and resources are scalable with agility. Cloud computing is an abstraction based notion of pooling of physical resources and presenting them as a virtual resource. This one is a new model for provisioning sources, for presentation applications, and designed for platform-independent user contact to services. Clouds come in different types, and the services and applications that run on clouds may or may not be delivered by a cloud service provider.

SAMPLES OF CLOUD COMPUTING

a. YouTube is the best specimen of cloud storage which masses millions of user uploaded audiovisual files.

b. Picasa and Flickr host lots of digital snapshotspermitting their users to create photo albums online by uploading images to their facility's servers.

c. Google Docs is extra best specimen of cloud computing that tolerates users to upload presentations, conversation documents and worksheets to their data servers. Google Docs permits users manage files and publish their papers for other users adjacent read or make edits.

III. DISTRIBUTED FILE SYSTEM FOR CLOUD

DISTRIBUTED FILE SYSTEM (DFS)

A distributed file system is a file system with data stored on a server. The data is read and processed as if it was kept on the local client machine. Automatic Prefetching uses past file accesses to predict future file system requests [10].The DFSmarks it suitable to share information and files midst users on a network in anorganized and authorized way. The server permits the client users to share files and store data fair like they are storing the data locally.

DFS 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 arekept 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 essentialis ensured. Confidentiality, availability and integrity are the foremost keys designed for a secure system. Distributed file systemallowsseveral 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 beenthusiastically 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.

Distributed File System in clouds such as GFS and HDFS rely on central or master servers or nodes (Master for GFS and Name Node for HDFS) to manage the metadata and the load balancing. The master rebalances replicas periodically: data must be moved from one Data Node or chunk server to another if free space on the first server falls below a certain threshold. However, this federal approach container become a bottleneck for people master servers, if they converted unable to manage a large number of file accesses, as it growths their previously heavy loads. The load rebalance difficult is NP-hard.A great deal of work on disk architectures, there has been very little work measuring actual low-level disk access [11].In order to get large number of chunk servers to work in collaboration, and to solve the problem of load balancing in distributed file systems, several approaches have been proposed, such as reallocating file chunks so that the chunks can be distributed as uniformly as possible while reducing the movement cost as much as possible.

Distributed File System provides excellent performance and reliability.DFS answers the encounters of dealing with enormous files and directories, organizing the movement of thousands of disks, so long as parallel access to metadata on anenormous scale, handling both scientific and general-purpose workloads, validating and scrambling on a huge scale, and swelling or decreasing dynamically in arrears to frequent device decommissioning, device letdowns, and cluster expansions.

CACHING

Caching is a process of store the frequent data from the machine. Service caching improve system performance. Around four places in a distributed system where hold data:Preceding the server's disk, in a cache in the server's memory, in the client's reminiscence and on the client's disk. Many standard techniques caching and predictive prefetching help somewhat, but provide little or no assistance for personal data that is needed only by a single user[8].The paramount two seats are not issuessubsequently any interface to the server check the federal cache. It is in the latter two spaces that problems ascend and have to consider the dispute of store consistency. Attitudes should take unified control. Server keeps path of what open in which mode. Support a formalfull system and deal with indicating traffic.

SERVERACCUMULATING

Server caching is spontaneous at the server in that the similar buffer accumulation is used as for all further files on the server. The variance for DFS-related composes is that they are all write-through toeludeunpredicted data defeat if the server dies.

CLIENT ACCUMULATING

The goal of client accumulating is to decrease the amount of isolated operations. Three methods of information are collected at the client: case data, casetrait information, and pathname findings.NFScollections the grades of read, declaim link, getattr, lookup, and reader processes. The hazard with accumulating is that irregularitiescould arise. NFS tries to shunconflictsor growth performance withvalidation. If collecting one or extra blocks of a file, protect a stage stamp. When a file is released or if the server is communicated for a fresh data block, match the last variation time. If the out-of-the-way modification period is more recent, invalidate the cache.Authentication is performed each three seconds on exposed files.Collected data blocks are supposed to stayeffective for three seconds.Cached reference book blocks are implicit to be usable for thirty seconds.At any time a page is modified, it is obviousmurky and slated to be printed (asynchronously). The page is reddenedwhile the case is closed.Transferences of data are complete in huge chunks; the evasion is 8K bytes. As shortly as a chunk is received, the client nearlyappeals the following 8K-byte chunk. This is recognized as read-ahead. The conjecture is that most file entrees are consecutive and potency as well fetches the next wedge of data while working on our present block, antedating that possible need it. This way, through the time user do, it will one or the other be don't have to pause too extensive for it subsequently it is on the situation way.

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. A transition distribution, which describes the distribution for the next state given the current state [18]. Recently, hidden Markov models have been comprehensive to pairwise Markov models and threesome Markov models which permitthought of more difficult data structures and the forming of non-stationary data.

TYPES OF ALGORITHMS IN HMM

Evolutions among the states are administrated by a set of likelihoods called transition probabilities. In a specific state theresult or observation can be generated, conferring to the associated possibility distribution. It is single the outcome, not the state visible to an exterior observer and therefore conditions are unknown to the outside; henceforth the name Hidden Markov Model.The hidden variable represents the risk state, which is assumed to be common to all bonds within one particular sector and region [15]. A Hidden Markov Model is unique in which perceive a series of emissions, but do not recognize the series of states the model pass awayover to generate the emissions. The emissions can be observed, thus giving some information, for instance about the most likely underlying hidden state sequence which led to a particular observation [20].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**.**Aroundare 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 isReal-world has constructions and processes which have or yieldrecognizable outputs. Example: speech signals

APPLICATIONS OF HMM

The trained HMM is used to identify the locate similar patterns in the historical data [16].Hidden Markov models are particularly known for their application temporal pattern respect such as speech,handwriting, signal recognition,harmoniousscore following, fractional discharges and bioinformatics. Cryptanalysis, Communicationproduction, Part of speech tagging, Text Separation in glance at solutions, Device translation, Incomplete discharge, Gene prediction, Arrangement of bio-sequences, Period Series Analysis, Action recognition, Protein folding, Metamorphic WormDiscovery and DNA Motif Discover.

EVALUATING HIDDEN MARKOV MODELS

- **a.** Producing a Test Sequence
- **b.** Assessing the State Sequence
- **c.** Approximating Transition and Release Matrices
- **d.** Estimating Posterior State Probabilities
- **e.** Altering the Initial State Supply

Information and Appliance Learning Toolbox meanings related to hidden Markov models are:

hmmgenerate — Produces a sequence of states and emissions from a Markov model.

hmmestimate — Evaluationsextreme likelihood estimates of transition and emission probabilities from a series of emissions and a recognizedseries of states.

hmmtrain — Computessupreme likelihood estimates of transition and emission prospects from a series of emissions.

hmmviterbi — Determines the bestfeasible state path for a hidden Markov model.

hmmdecode — Reckons the subsequent state possibilities of a series of emissions.

This segmentdisplays how to use these utilities to examine hidden Markov models.

EXAMPLE

Coin toss: Headsand backsseriespertwo sides in a coin. You stay in a room, with a wall. Somebodybehind schedule wall flip-flops coin tells result. Coin assortment and fling is hidden. Cannot perceive events, single output (heads, tails) after events. Difficult is then to form a model to describe observed series of heads and tails.

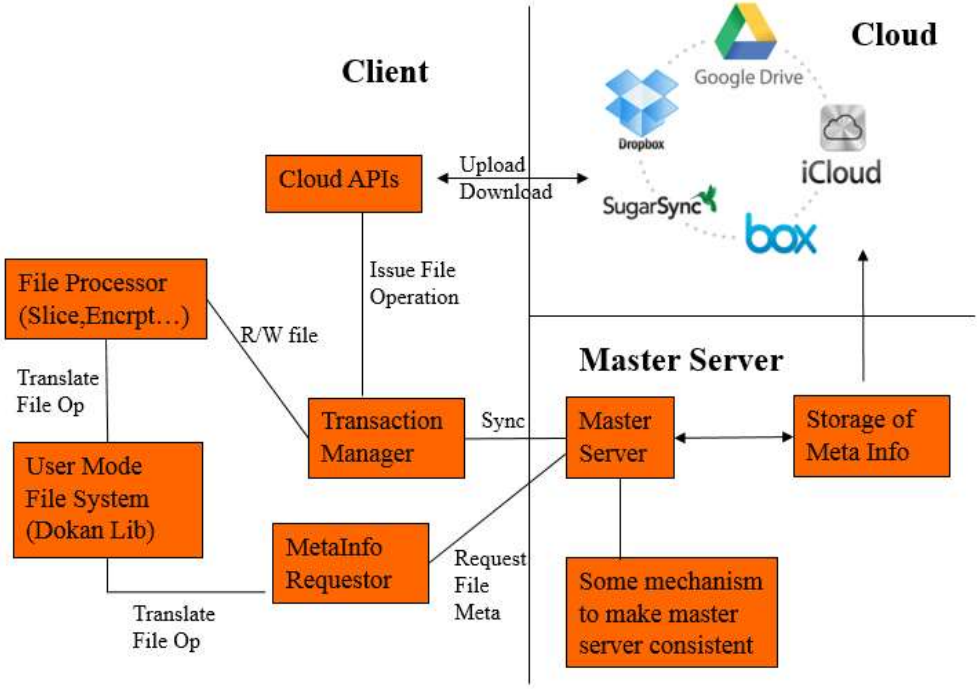
V. CLOUD COMPUTING AND DISTRIBUTED FILE SYSTEM

The necessary difference is that cloud computing is almost infrastructure whereas distributed system is about accomplishment and communication sandwiched between processes. Let explain:Cloud is nearby elasticity, multi-tenancy. Fundamentally on-demand compute or storageCloud is in what manner procurescalculate and storage. Could be a SaaS (Software as a Service) where unique also secures the processing. A distributed system is somewhere the compute is vault across multiple processes - in the similar

computer or crosswise different computer systems. Distributed systems run in a cloud infrastructure by way of well.Typically distributed processing is encrusted over certain form of IPC (Inter Process Communication) Protocol - CORBA, DCOM et al in previousyears and currently REST, JNI, et al. Furthermore the distributed processing containeris compute parallelism or information parallelism.

SYSTEM ARCHITECTURE AND IMPLEMENTATION

SYSTEM ARCHITECTURE



Around three fragments in system architecture client, cloud and master server. The client is answerable for charitable the file system abstraction to end user, control file amendment and file preprocessing. The master server is intended at sustaining the metadata information for files, charting chunks to different location, control data replication and keeping data in reliable status. The cloud portion is mainly about storage. The major server pull Meta information (pardon chunk does this storage) and the client send request to cloud storage service.

IMPLEMENTATION

In an existing system the files prefetching I/O disk actions from the numerous clients via the storage server. Afterwards analyzing the client I/O actions are dispatched to the relevant machines completed by the storage server. The prefetching data can be pushed to the relevant client machine from the storage server [14].For prediction the I/O antiquity details, used chaotic time series prediction and linear regression prediction.

The Hidden Markov Model (HMM) model offers the dynamic programming algorithms such as Forward, Backward and Viterbi algorithm. The Forward algorithm defines the probability that observed in the series was created by the Hidden Markov Model and ponders all paths that could have produced by the observed sequence. The backward algorithm can be calculated in the similar way as possibility of most likely path. The Viterbi algorithm regulates which explanation is supremepossible and finds the path maximum likely to have produced the observed series.

We have pragmatic the proposed mutual with the data prefetching in Distributed File System for cloud and Forward Hidden Markov Model algorithm for forecast the I/O events from the client machine. The persistence of using this Forward Chain is to fetch the data for belief state which all probabilities. Using forward algorithm, likelihood value compare with other time series prediction values.

VI. ANALYSIS OF FORWARD ALGORITHM

The Forward Algorithm, in the context of a Hidden Markov Model is used to calculate a belief state. The probability of producing $\mathbf{Q_i}$, $\mathbf{Q_{t-1}}$while ending up in state **s**.

$$\alpha_i(t) = p(Q_1, _{t-1}, X_t = i) \quad \text{— (1)}$$

N represents the number of states. T is the period of observations used to estimate the model parameters [17]. **T**signifies the amount of Observations. Q_i is emission parameter for an observation associated with state **i**.

Initialization:

$$\alpha_i(1) = \pi_i \quad \text{— (2)}$$

Induction:

$$\alpha_j(t+1) = p(Q_1, _t, X_{t+1} = j) \quad \text{— (3)}$$

Termination

$$£_i\ \alpha_i(t)\ a_{ij}\ b_{ij}\ o_t \quad \text{— (4)}$$

The solution to problem (2) is given by the Forward Algorithm. Once we have the solutions to the above problems, the tools helps to shape based clustering of Hidden Markov Model are available[13].

For each sequence in the training set of sequence:

a)Calculate Forward probabilities with the forward algorithm.
b)Analyzebackwardlikelihoods with the backward algorithm.
c) Calculate the contributions of the current sequence to the transitions of the model; calculate the current sequence to the emission probabilities of the model.
d) Calculate the new model parameters (start probabilities, transition probabilities, and emission probabilities).
e)Analyze the original log likelihood of the model.
f)Stop when the change in log like hood is smaller than a given threshold or when a maximum number of iterations are passed.
The forward-backward algorithm has very important applications to both hidden Markov models (HMMs) and conditional random fields (CRFs)[19].Preparation problem:Assumed a model arrangement and a set of series, find the model that greatest fits the data.

VII. CONCLUSION

The proposed, implemented and evaluated in data prefetching Distributed File System for Cloud, which the client machines can receive relevant data proactively through by the storage server in a cloud environment. The storage servers are capable to analyze and predict the client I/O events and then they proactively push data to the relevant client machines for satisfying client's future applications requests. The purpose of forward I/O events and regarding client machine information's are piggybacked and then transferred to corresponding storage server from the client nodes.
The Hidden Markov Model Froward Algorithm computes probability much more efficiently than the naïve approach, which very quickly ends up in combinational explosion. It can provide the probability of a given emission or observation at each position in the sequence of observations.
The current implementation of proposed data prefetching process in Distributed File System for cloud using Hidden Markov Model Froward prediction algorithms for proactively fetch disk I/O events from the client nodes. Using forward chain the data is trained and increases the learning process. In future work of this paper, there are different workloads happening in the system by client, classifying block access patterns from the block I/O events are traced and resulted by several workloads with using Viterbi algorithm.

REFERENCES

[1] Jianwei Liao, Francois Trahay, Guoqiang Xiao, Li Li, Yutaka Ishikawa Member, ' Performing Initiative Data Prefetching in Distributed File Systems for Cloud Computing'.
[2] 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.
[3] 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
[4] N. Nieuwejaar and D. Kotz.The galley parallel file system. Parallel Computing, 23(4-5):447–476, 1997.
[5] 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.
[6] 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.
[7]X. Ding, S. Jiang, F. Chen, K. Davis, and X. Zhang. DiskSeen: Ex- ploiting 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] A. Reda, B. Noble, and Y. Haile. Distributing private data in challenged network environments. In Proceedings of the 19th international conference on World wide web (WWW '10). ACM, pp. 801–810, 2010.
[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(3), Article 10, 23 pages, 2013.
[10] J. Griffioen, and R. Appleton.' Reducing file system latency using a predictive approach'. In Proceedings of the 1994 USENIX Annual Technical Conference (ATC '94), pp. 197-107, 1994.
[11] S. Narayan, J. A. Chandy. Trace Based Analysis of File System Ef- fects on Disk I/O. In Proceedings of 2004 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '04), 2004.
[12] M. S. Obaidat. QoS-Guaranteed Bandwidth Shifting and Re- distribution in Mobile Cloud Environment. IEEE Transac- tions on Cloud Computing, Vol.2(2):181–193, April-June 2014, DOI:10.1109/TCC.2013.19
[13] 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.
[14] A.B. Poritz, 'Hidden Markov models: a guided tour´, in Proc. of ICASSP, pp. 7-13, 1988.
[15] Giacomo Giampieri, Mark Davis and Martin Crowder, 'A Hidden Markov Model of Default Interaction', wwwf.imperial.ac.uk.
[16] Md.Rafiul Hassan,Baikunth Nath and Michael Kirley, 'A fusion model of HMM, ANN and GA for stock market forecasting', https://pdfs.semanticscholar.org.
[17] SZYu,H Kobayashi,'An Efficient forward backward algorithm for an explicit duration Hidden Markov Model',www.hisashikobayashi.com.

[18]RameshSridharan,' HMMs and the forward-backward algorithm',https://people.csail.mit.edu.
[19] Michael Collins,' The Forward-Backward Algorithm',www.cs.columbia.edu.

[20] Christian Kohlschein,' An introduction to Hidden Markov Models',www.tcs.rwth-aachen.de.