

Implementation of Kerberos method on DDAS system and search data speedily from extracted Zip data

Suraj Gulhane^{1*} and Sonali Bodkhe²

^{1*,3}*Department of Computer Science and Engineering ,
Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur, India*

www.ijcseonline.org

Received: Jul /12/2015

Revised: Jul/20/2015

Accepted: Aug/17/2015

Published: Aug/30/ 2015

Abstract— Deploying application over the web is increasing day by day. Such deployed application useful for client to store as well as retrieve database to/from particular server. Over the web data stored in distributed manner so flexibility, reliability, scalability and security are important aspects need to be considered while constructed data management system. After analyzing Distributed data aggregation service(DDAS) system which maintain a catalog which is relying on Blobseer it found that it provide a good performance in aspects such as data storage as a Blob (Binary large objects) and fast retrieval of data by data aggregation process. For highly complex analysis and instinctive mining of scientific data, Blobseer act as a repository backend for easy retrieval of data. By using this Kerberos method client will be able to done a secure authentication as using this method only authorized clients are able to access distributed database. Kerberos consist of 4 steps i.e. Authentication Key exchange, Ticket granting service Key exchange, Client/Server service exchange and Build secure communication. After that aggregation of data carried out and aggregated data catalog is generated. From that catalog user is able to search a required data and this data in zip file form saved at client side. For zipping purpose Adaptive Huffman method is used (also referred to as Dynamic Huffman method) which is based on Huffman coding. It permits compression as well as decompression of aggregated data.

Keywords— Adaptive Huffman Method; Blobseer; Distributed Database; Kerberos; Data Aggregation

I. INTRODUCTION

There is exponential information growth correlative with associate increasing want of quick and reliable information access moreover as information management that are key problem that reduces applications performance. Application performance is looking on such properties of the data management service. So as to be obtainable for more use information-management systems specifically work with high volumes of structured data that must be hold on expeditiously and accurately. However, as a result of the exponentially growing information is said to with associate increasing would really like for fast and reliable information access, knowledge/Data management continues to be a key issue that very impacts on the performance of applications, because the overall application performance is very obsessed on the properties of the data management service [2]. Whereas managing and aggregation system should bear in mind of the complexness and issue of configuration such a system must be consistent and for that these high volumes of distributed information are continuously structured in a very explicit kind [6].

What is more there's have to be compelled to applied some ways for locating and retrieving information in a very secure and effective manner. What is more of these options have to be compelled to guarantee high levels of fault tolerance and information consistency, whereas at identical

time it'll minimize access and computation prices. For that there'll be Distributed information Aggregation Service (DDAS) wishing on a distributed information management system, particularly BlobSeer fulfilling higher than objectives. The service will utilize multiple choices of BlobSeer like info uncovering, distributed info management and versioning-based concurrency management and might be implemented with respect all the needs and constraints obligatory by data-intensive applications [7].

DDAS are going to be designed to confirm quantifiability, fault tolerance and information retrieval performance. The precise objectives think about main options and specific eventualities for DDAS. Firstly, all manipulated information during this system is pictured in long sequences of bytes of unstructured information that's browse and written into binary massive objects (BLOBS). As this application's main goal is automatic mining of scientific information and to own a repository back-end for complicated analysis, as its read of the info received from client's are going to be structured. So for creating the proper translation between the 2 views, a large effort is going to be place into developing a brand new meta-data management layer. Second associate absolute analysis of many of BlobSeer's attributes must be done to determine the most effective preparation configuration for a particular information aggregation pattern. This may be reach by

learning characteristics like the distribution of connected and unrelated information between BLOBS or the mounted size of a blob's page. A compression algorithmic rule i.e. adaptation Huffman writing is applied on information whereas information retrieval operation in order that it'll minimize access and computation prices. Owing to compression and decompression algorithmic rule information reduction happens that minimizes computation prices moreover as security increased [16].

Kerberos methodology is going to be used for authentication in order that solely licensed purchaser's are able to access the info over server. In Kerberos technique, service requesting consumer not send Secret positive identification to application server Instead it Request initial price tag from authentication server and these initial ticket and encrypted request sent to application server. There are four steps namely-Authentication Key exchange, price tag granting service Key exchange, Client/Server service exchange and Build secure communication.

In Authentication key exchange, service requesting consumer request authentication server for initial price tag to initial ticket Granting Server (TGS), authentication server look for the service requesting consumer at intervals the data and if found then generate session key(SK1) to be used between service requesting consumer and initial TGS. Writing has been distributed by Kerberos it inscribe the SK1 victimization client's secret key. Now, Authentication server (AU) can turn out and send initial price tag granting ticket (TGT) to client/user by victimization TGS's secret key.

In price tag granting service Key exchange, The writing has been distributed at service requesting consumer facet so as that the service requesting consumer decrypt the message and recover the session key then uses it to form consumer critic containing the user name, IP address and timestamp. Afterwards service requesting consumer sends this consumer critic beside initial TGT to the initial TGS for requesting access to the requested target server. 1st of all, initial TGS decrypt the initial TGT afterwards uses this SK1 at intervals the initial TGT to decrypt consumer critic. Verification of licensed purchasers is vital so it verifies the info at intervals the consumer critic if everything is matches then it precedes the request. Therefore, afterwards initial TGS turn out new session key i.e. SK2 (Session key2) for service requesting consumer and target server, first inscribe it victimization SK1 thus sends it to service requesting consumer.

In Client/Server key exchange, the service requesting consumer distributed writing of the message and gets SK2. Then service requesting consumer turn out new appraiser encrypted with SK2 and afterwards the service

requesting consumer send session initial price tag and consumer critic to the target server. Afterwards this target server decrypts and checks the initial price tag, consumer critic, service requesting consumer address and timestamp. In Secure communication step, the target server is attentive to the service requesting consumer as a result of its verified in earlier steps and share writing key for secure communication as a results of exclusively target server and repair requesting consumer share this key.

II. RELATED WORK

The following sections justify the survey of varied papers. Completely different ways are used for extracting sentiments or opinions from the given sentences by several researchers. Following section conjointly justify completely different ways that are accustomed rank or classify these opinions extracted from the sentences.

There's an information storage system provided by BlobSeer [1]. This knowledge storage system represents knowledge as BLOBS taking into thought that almost all knowledge are unstructured in circulation. This same BLOB is used to store huge amounts of knowledge/information by maintaining the offset of the BLOB that make certain the possibility of quantifiability thus allowing application-level correspondence as AN older version are going to be scan whereas a additional trendy version is generated however this DDAS system isn't a lot of secured and a few new ways OR algorithms needed to supply security to knowledge aggregation service.

For Aggregation solutions, the first system in [5] a way for aggregating internet service knowledge is conferred. The fundamental abstract structure employs a gaggle of interconnected aggregation nodes that collaborate with each other thus on execute service requesting customer requests. This aggregation system provides high outturn and nice response times once client's requests involve an oversized range of aggregation nodes with all for handling a large range of requests. If the low numbers of aggregation nodes are use and request load isn't distributed uniformly then it achieves a lot of lower performance and outturn.

For knowledge Storage Solutions, S3 storage system projected by Amazon [8] aims to supply a charging model i.e. easy 'pay-as-you-go' charging model, storage price is incredibly low, extremely out there service. Some helpful aspects of this model are the worldwide availableness of the system and also the high-level access management. It will but have several unfavorable aspects like having the likelihood of losing complete hold on knowledge if one thing as easy as like your email account is compromised and conjointly not having any style of Service Level Agreement (SLA) to keep up hold on knowledge.

There's together exploring the solution given in [9] where the information storage atmosphere is enforced to handle a very high write product and together scale with the amount of users. Although the system has many blessings just like the ability to scale incrementally, victimization replication to verify high availableness and strength and failure detection, vaticinator together should shock certain issues like non-uniform data and cargo distribution.

The system in [10] evaluates the implementations and interfaces for user made public aggregation in such a giant quantity of state of the art distributed computing systems. User outlined aggregation in Hadoop implementations build user answerable for understanding all outlined sorts and use casts or access functions for filling the specified fields. This should adds plenty of quality to the trivial computation, so for the extra subtle aggregation functions the overhead of casting between the system varieties are less noticeable, and conjointly the advantage of obtaining access to a full featured source language, throughout this case Java language, are extra smart. The interfaces of the User outlined aggregation of information show the advantage that in-built sound unit functions have once writing an aggregation methodology, however conjointly there's AN limits of information languages once user-defined functions and sorts are extra advanced.

In [11], there's AN energy economical Aggregation formula for WSNs i.e. secure and powerful against malicious executive attack by any faulty or compromised node among the network and each node, among the projected formula, so instead of uncasting its detected knowledge to its parent, its estimate ought to be broadcasts to all or any or any its neighbours. Due to that the protocol thanks to that it become extra fault-tolerant and can increase the information availability among the network. Though it's secure against malicious corporate executive attack however not secure against alternative style of security threats and conjointly it's time overwhelming method.

In [12], there's a use of cloud computing for the retrieval of knowledge from distributed information. However, during this cloud computing setting, typically failure is happens and thanks to that it causes some unfortunate errors i.e. nodes could also be accessorial, upgraded, replaced within the system. That causes a haul of load imbalance within the system. For resolution this drawback, load rebalancing algorithmic rule is enforced during this paper in order that overload mustn't get occurred in central node. Therefore, the implementation is completed in hadoop distributed classification system. The Hadoop Distributed classification system is that the main file system in Hadoop, master/slave design is gift within the Hadoop Distributed classification system within which one device controls the opposite devices. As apache hadoop is employed, security problems

are arises. For resolution these security problems and to extend security, Kerberos authentication protocol is use i.e. include authentication server and price tag granting server for ticket and session key exchange. This Kerberos methodology is enforced to handle multiple nodes. The advantage of this approach is that one might still use the tokens to supplement a unique primary authentication mechanism. The disadvantage of this approach is that load rebalancing algorithmic rule cause's reduction in knowledge retrieval speed.

In [13], AN Anonygator system is employed for knowledge aggregation and supply knowledge retrieval theme to the clients/users. AN Anonygator technique is AN anonymity-preserving data aggregation service for the large-scale distributed applications. AN Anonygator uses ANonymous routing that offer user obscurity as host that generate a range of messages can separate those messages. This technique can stop knowledge malicious users uploading Brobdingnagian and disproportionate quantity of useless pretend knowledge by using a light-weight accounting theme. Finally, AN Anonygator maintains overall system quantifiability by employing an utterly distinctive distributed tree-based data aggregation procedure that's sturdy to pollution attacks. Therefore, All of these components are tied on and tuned by a customization engine, with a scan to achieving pollution resistance, specific obscurity, and efficiency goals.

In [14], a comprehensive taxonomy is taken under consideration to elucidate distributed organization architectures and use this stratified system to survey existing DFS system implementations in very huge scale network computing systems like Search Engines, Grids, etc. throughout this approach, taxonomy and conjointly the survey results is printed to identify field approaches that haven't been completely explored in previous distributed classification system analysis. Security will still be a major issue related to distributed data storage.

In [15], BlobSeer example approach is incontestable. BlobSeer example is a cost-effective and ascendable large-scale storage service that serves the requirements of data-intensive applications. as a result of the storage service i.e. data storage could also be a key issue that impacts overall performance of data-intensive applications, there is have to be compelled to understand issues involved and coming up with a method so as that it's crucial to beat these problems. This work provides economical support for choices and techniques normally used to exploit correspondence at data level likewise on explore a gaggle of recent choices and techniques that will be leveraged to extra improve parallel data access.

III. PROBLEM DEFINITION

In [1], this knowledge storage resolution provided by BlobSeer and this DDAS system isn't a lot of secured and a few new ways OR algorithms needed to supply security to knowledge aggregation service.

In [5], For Aggregation solutions methodology of aggregating net service knowledge is given however its disadvantage is, if the low numbers of aggregation nodes square measure use and request load isn't distributed uniformly then it achieves a lot of lower performance and outturn.

In [8], For knowledge Storage Solutions a S3 storage system is projected by Amazon however it will but have several unfavorable aspects like having the likelihood of losing complete hold on knowledge if one thing as easy as like your email account is compromised and additionally not having any sort of Service Level Agreement (SLA) to keep up hold on knowledge.

In [10], system evaluates the implementations and interfaces for user outlined aggregation however there's a limits of info languages once user-defined functions and sort's square measure extra advanced.

In [11], there's AN implementation of energy economical Aggregation algorithmic rule for WSNs (wireless device network). Though it's secure against malicious business executive attack however not secure against different sort of security threats and additionally it's time intense method.

In [12], implementation of load rebalancing algorithmic rule is applied. The hadoop distributed classification system is employed for implementation of this technique however the disadvantage of this approach is that load rebalancing algorithmic rule cause's reduction in knowledge retrieval speed.

In [13], AN Anonygator system is employed for knowledge aggregation. This system can stop knowledge malicious users uploading immense and disproportionate quantity of useless pretend knowledge but this method is time overwhelming and cause's reduction in knowledge uploading and knowledge retrieval speed.

In [15], BlobSeer paradigm approach is incontestable. This BlobSeer storage service isn't a lot of secured and a few new ways OR algorithmic rules needed to supply security to the current storage service additionally a compression algorithm should be applied on info whereas knowledge retrieval method that reduces value of knowledge retrieval and enhance retrieval speed.

IV. PROPOSED WORK AND EXPERIMENTAL REVIEW RESULTS

A. Proposed Work

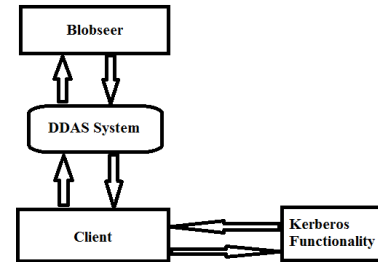


Figure 1. Architecture of implementation of Kerberos method on DDAS. In this system, the info backend storage system communicates with the DDAS'extended consumer through write and browse requests to retrieve and store specific objects in their serialized kind. DDAS acts as an intermediary between all different data-management applications and BlobSeer that need quick and reliable storage and retrieval of information.

Here the most plans are to map every object to a minimum of one theme. The theme is pictured by a gaggle of key-value pairs that represents the properties of the objects. Main theme of AN object should contain AN entry that's a stream that represents the article, a key that unambiguously identifies the article and every one the properties that the objects have in common.

For Object Storage, DDAS maintains a catalog for of these mappings referred to as the article catalog. It conjointly maintains a catalog of all the aggregation and growth themes that are every mapped to a listing of object meta data that unfold to it scheme. This catalog is understood because the data catalog.

Kerberos methodology that is employed for authentication so solely approved shoppers are able to access the info over server conjointly a compression and Decompression formula is applied on information whereas information retrieval operation. once a Store request for brand spanking new information is formed, the new object entry is hold on within the article catalog and its entry is dilated to any or all the prevailing schemes within the data catalog during this means this new object fits.

B. Experimental Review Results

1) Blobseer:

Using BlobSeer for Associate in nursing aggregation resolution permits the convenient putting of objects to guide the DDAS for future operations. The system organizes information in BLOBS, marking every storage operation with a brand new version, it will prove terribly elective within the complicated aggregation method With a model that collects objects that match common attributes, our service can send information to BlobSeer specified all retrieval and aggregation operations for a septic pattern are

going to be reduced to reading information from the proper location in BlobSeer. This resolution represents information as BLOBS taking into thought that almost all information in circulation is unstructured.

This gives the chance of guaranteeing measurability mistreatment constant BLOB to store massive amounts of data by solely maintaining the onset of the BLOB. alongside these options BlobSeer conjointly provides the user with a versioning-oriented access interface for manipulating BLOBS, therefore allowing application-level similarity as Associate in Nursing older version may be browse whereas a more recent version is generated. To move with BlobSeer all that's needed could be a handle that points to a specific BLOB from that information is extracted or to that information is hold on. Minimizing the quantity of handles that may be created throughout a call for participation was a serious priority in our implementation. BlobSeer is Associate in nursing economical distributed information management service specifically designed to deliver a high output below significant access concurrency. Information is abstracted in BlobSeer as vast sequences of bytes known as BLOBS (Binary massive Objects).

Each BLOB is manipulated through an easy versioning access interface that allows fine-grained reads, writes and appends of subsequences of bytes from/to the BLOB. Three key style factors change BlobSeer to attain a high throughput below significant access concurrency: information marking, distributed information management and versioning-based concurrency control. Every BLOB is split into chunks that area unit distributed among information suppliers, that area unit accountable to store the chunks. To keep up information availableness in spite of failures, each chunk is replicated on multiple distinct suppliers. A configurable chunk distribution strategy is used once writes and appends area unit issued so as to optimize chunk placement in such approach that accesses to completely different chunks area unit as much as doable handled by completely different machines, effectively distributing the I/O work.

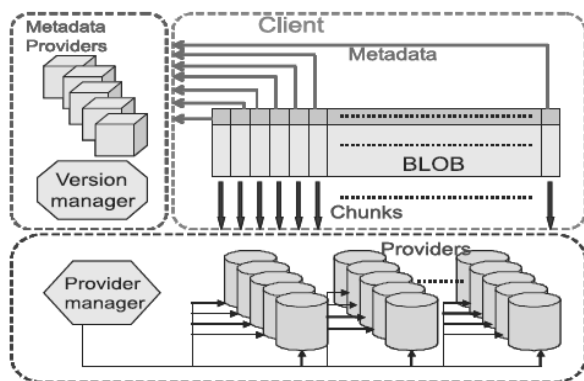


Figure 2. Design of the BlobSeer system.

Metadata decentralization: BlobSeer uses a distributed metadata management theme to avoid the bottleneck of accessing constant centralized node and forestall one purpose of failure. Versioning-based concurrency control: Writes and appends to a BLOB ne'er modify its contents, however rather generate a new exposure of it that appears and acts just like the original BLOB excluding the applied update. Solely the distinction is physically held on, with unrestricted components shared. This approach enables the concurrency management to isolate updates in their own exposure, so avoiding the necessity for synchronization significantly higher than lock-based approaches, that greatly improves achieved output

A typical size for a chunk at intervals a BLOB is often smaller that one MB, whence the challenge of coping with many thousands of chunks relief to only one BLOB. BlobSeer provides efficient support for heavily-concurrent access to the keep knowledge, reaching a outturn of a vi.7 GB/s collective bandwidth for a configuration with sixty data providers, ninety knowledge suppliers and 360 coincident writers.

Few tests obtained by deploying BlobSeer on the

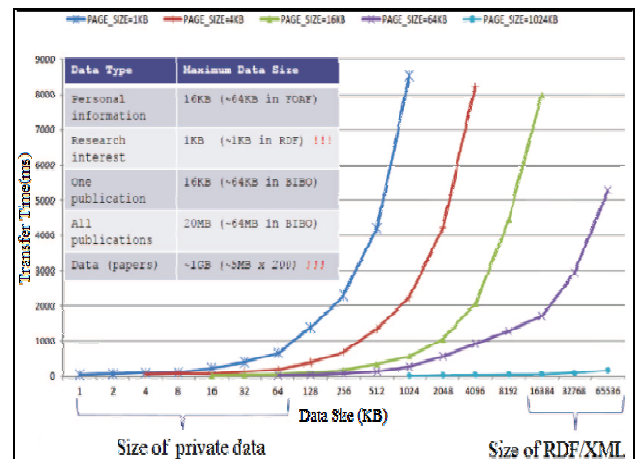


Figure 3. BlobSeer analysis Results for various data kind and size.

2) DDAS Interaction:

So as for a client application to act with the DDAS there'll be got to offer a correct description of the format of the input and output of schemes and objects. There's got to do this through that of XML files. The foremost theme that represents academic degree object is made of a root tag and inner tags that represent the object's properties with their values as text content. All objects, freelance of the applying, ought to have the "key" and "entry" tags. Similarly, academic degree enlargement theme incorporates a similar XML format, however we tend to tend to say that it's significantly smaller in size as a result of it does not contain a tag with the entire object. These 2 types of XML files area unit input whenever a storage operation is required with the foremost object theme as a compulsory argument and thus

the enlargement theme no obligatory. The next is academic degree example of academic degree XML file with the theme for a BibTex object

The aggregation theme is that the argument required by the retrieval request and its XML format contains a root tag with many inner tags with specific names. First, the “select” tag contains inner tags with properties and values that objects should match. Second, the “exclude” tag has inner tags that represent the values for properties that objects shouldn't have so as to be retrieved. Finally, the aggregation theme can contain multiple “function” tags that in turn have inner tags describing the property there on the perform is applied, the kind of the property, the number of operands thereon the perform works and thus the result kind.

3) Kerberos:

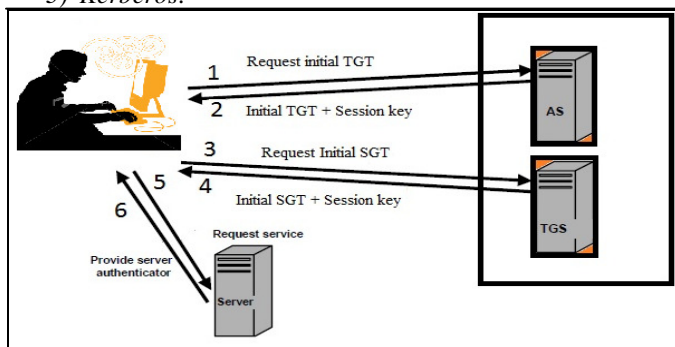


Figure 4. Overview scenario of the Kerberos process/actions.

a) Kerberos 4 message exchange:

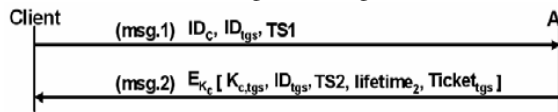


Figure 5. Authentication Service key Exchange to obtain initial ticket granting ticket.

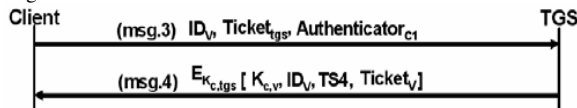
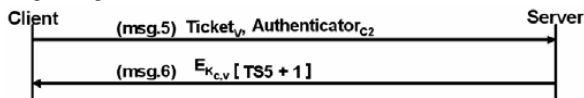


Figure 6. Ticket-Granting Service key Exchange to obtained initial service-granting ticket



$$\begin{aligned} \text{Ticket}_{TGS} &= E_{K_{C,TGS}} [K_{C,TGS}, ID_C, AD_C, ID_{TGS}, TS2, lifetime_2] \\ \text{Ticket}_V &= E_{K_{C,V}} [K_{C,V}, ID_C, AD_C, ID_V, TS4, lifetime_4] \\ \text{Authenticator}_{C1} &= E_{K_{C,TGS}} [ID_C, AD_C, TS3] \\ \text{Authenticator}_{C2} &= E_{K_{C,V}} [ID_C, AD_C, TS5] \end{aligned}$$

Figure 7. Client Server Authentication key Exchange to obtain service.

b) Kerberos 5 message exchange:

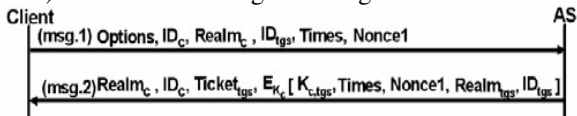


Figure 8. Authentication Service key Exchange to obtain initial ticket granting ticket.

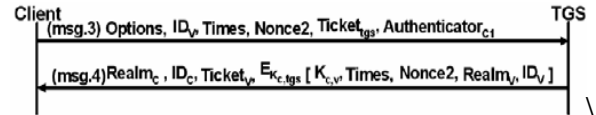
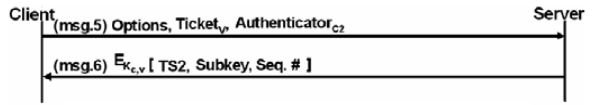


Figure 9. Ticket-Granting Service key Exchange to obtained initial service-granting ticket.



$$\begin{aligned} \text{Ticket}_{TGS} &= E_{K_{C,TGS}} [\text{flags}, K_{C,TGS}, \text{Realm}_C, ID_C, AD_C, \text{Times}] \\ \text{Ticket}_V &= E_{K_{C,V}} [\text{flags}, K_{C,V}, \text{Realm}_C, ID_C, AD_C, \text{Times}] \\ \text{Authenticator}_{C1} &= E_{K_{C,TGS}} [ID_C, \text{Realm}_C, TS1] \\ \text{Authenticator}_{C2} &= E_{K_{C,V}} [ID_C, \text{Realm}_C, TS2, \text{Subkey}, \text{Seq. \#}] \end{aligned}$$

Figure 10. Client Server Authentication key Exchange to obtain service.

TABLE I. COMPARISON BETWEEN KERBEROS 4 AND KERBEROS 5

No.	Comparison Item	Type of kerberos	
		Kerberos 4	Kerberos 5
1	Encryption technique	DES	Encryption key is tagged with type and length
2	Session key	1/lifetime	Client and server may negotiate for sub-session Key(1/connectionn)
3	Network address	IPv4	Any (network address is tagged with type)

4) Graph Result

TABLE II. SAMPLE VALUES GENERATED

No. of Record	Differential Values		
	Aggregation Threshold	Percentage space save	Time
1000	2	93.6	0.001428
2000	2	96.8	0.001482
5000	2	98.72	0.00146
10000	2	99.36	0.001439

Experimental result by varying the number of records with percentage space save shows that as number of records increases the percentage of space save increases proportionally as shown in graph as follows;

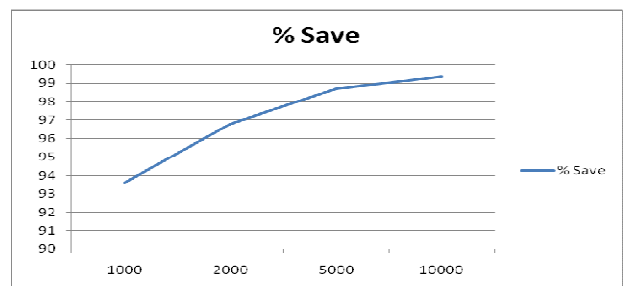


Figure 11. graph showing percentage space saves

Experimental result by varying the number of records with time required to search data shows that first of all delay

time to search data from record will first increases and then after some threshold value it will decreases as number of records increases as shows in the graph as follows;

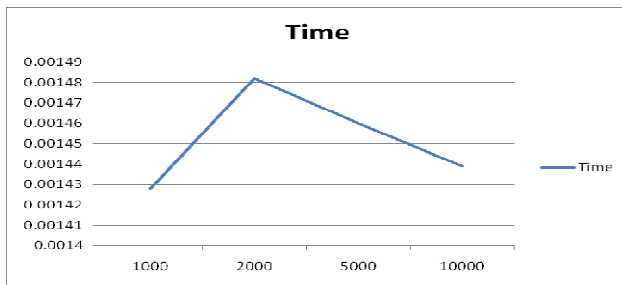


Figure 12. graph showing time required to search data.

V. CONCLUSION

As aggregation and information management continues to evolve during a wide spectrum of analysis domains and necessities become complicated and a lot of specific. A Distributed Data Aggregation Service (DDAS) depends on BlobSeer function a intermediary between distributed information and User that reduces User information looking out time from distributed information and will increase information retrieval speed.

Implementing the conception of DDAS victimization Kerberos will increase the safety and solely licensed User is ready to access distributed information and Zip method is applied on aggregated data which will increases data retrieval speed. This complete system will saved the space for generating aggregated data and will reduces time required to search data.

REFERENCES

- [1] Suraj Gulhane, Sonali Bodkhe "DDAS using Kerberos with Adaptive Huffman Coding to enhance data retrieval speed and security" in *Proceedings of the IEEE International Conference on International Conference on Pervasive Computing*, IEEE Computer Society, **2015**.
- [2] Florin Pop, Gabriel Antoniu, Vlad Serbanescu, Valentin Cristea, "Architecture of Distributed Data Aggregation Service" in *Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications*, IEEE Computer Society, **2014**.
- [3] K. Aamodt *et al*, "The ALICE experiment at the CERN LHC," *JINST*, vol. 3, p.S08002, August 14, **2008**.
- [4] S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, B. Quetier, O. Richard, F. Cappello, E. Caron, M. Dayde, F. Desprez, Y. Jegou, P. Primet, and E. Jeannot, "Grid'5000: A large scale and highly reconfigurable grid experimental testbed," in *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, ser. GRID'05. Washington, DC, USA: IEEE Computer Society, **2005**. [Online]. Available: <http://dx.doi.org/10.1109/GRID.2005.1542730>.
- [5] X. Pennec, T. Glatard, and J. Montagnat, "Efficient services composition for grid-enabled data-intensive applications," in *Proceedings of the IEEE International Symposium on High Performance and Distributed Computing*, Jun. **2006**. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00683201>.
- [6] S. Dustdar, W. Hummer, and P. Leitner, "Ws-aggregation: distributed aggregation of web services data," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC'11. New York, NY, USA: ACM, **2011**.
- [7] S. Leo power presentation on Python MapReduce Programming with Pydoop, "Pydoop: a python mapreduce and hdfs api for hadoop", **2014**
- [8] G. Antoniu, L. Boug'e, D. Moise, A. Carpen-Amarie, and B. Nicolae, "Blobseer: Next-generation data management for large scale infrastructures," Author manuscript, published in "Journal of Parallel and Distributed Computing" **71,2(2011)**.
- [9] M. Ripeanu, S. Garfinkel, M. R. Palankar, and A. Iammitchi, "Amazon s3 for science grids: a viable solution?" in *Proceedings of the 2008 international workshop on Data-aware distributed computing*, ser. DADC '08. New York, NY, USA: ACM, **2008**.
- [10] K. Ramamohanarao, S. Venugopal, and R. Buyya, "A taxonomy of data grids for distributed data sharing, management, and processing," *ACM Computer Survey*, vol.38, June 2006.
- [11] M. Isard, Y. Yu, and P. K. Gunda, "Distributed aggregation for data parallel computing: interfaces and implementations," in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, ser. SOSP '09. New York, NY, USA: ACM, **2009**.
- [12] Jaydip Sen "A Robust and Secure Aggregation Protocol for Wireless Sensor Networks" Innovation Lab, TCS Ltd, **2015**
- [13] N. Chiwande, Prof. Animesh R. Tayal, and Ms. Vidya, "An Approach to Balance the Load with Security for Distributed File System in Cloud," in *Proceedings of 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies*.
- [14] Ranjita Bhagwan, Venkata N. Padmanabhan, Krishna P. N. Puttaswamy, "Anonymity-Preserving Data Aggregation using Anonygator," Computer Science Department, UCSB, †Microsoft Research, India, **2014**
- [15] Eunmi Choi, Subaji Mohan, Pilsung Kim, SangBum Kim, and Tran Doan Thanh, "A Taxonomy and Survey on Distributed File Systems" in *Proceedings of Fourth International Conference on Networked Computing and Advanced Information Management*, **2008**.
- [16] Bogdan Nicolae, "BlobSeer: Efficient Data Management for Data Intensive Applications Distributed at Large-Scale," University of Rennes IRISA, France, **2015**
- [17] Gonzalo Navarro, Nieves Rodriguez Brisaboa, "New Compression Codes for Text Databases" University of Coruna (España), **2013**