HYBRID INDEXING SCHEME FOR PRESERVING MEDICAL DATA IN MOBILE CLOUD ENVIRONMENT

[1]Sanjay.H.M, [2]Dr.GuruPrakash.C.D,
[1]Assistant Professor, [2]Professor
[1]Department of Computer Science and Engineering, PESCE, Mandya , India
[2]Department of Computer Science and Engineering, SSIT, Tumkur. India

Abstract : The enormous growth of medical data sequencing found to be very challenging in medical technologies that makes available of the medical and health care data in the nearby future. The ratio of adapting mobile smart devices in distributed environment provides a scope in accessing the medical applications at user finger tips. For example medical database contains necessary set of information over sensitive data (eg. Ancestry data, artery diseases and so on). It's necessary to use the medical applications in a sensitive way by adapting user's sensitivity and privacy concerning to access these set of information to avoid leakage of information. As we knew that the smart devices varies with respect to heterogeneity factors like service provider, Infrastructure providers and their own physical attributes. To process this type of sensitive medical application in this paper we try to provide a hybrid selection and indexing scheme framework, which takes care of processing the information at the users end to extract their required medical data from cloud. A user can perform a confidential test by deploying this medical information in a hybrid way on their smart devices. Here we deploy a homogeneity control law on cloud that reports proper feedback among different clusters, by deploying Kubernetes system which supports single and multinode replications. From our earlier contribution of control input that provisions feedback between multiple agents by periodic indexing. The sensitive medical user information we have considered in this work is Single Nucleotide Polymorphism (SNP).

IndexTerms: **SNP,Kubernetes, GlusterFS,CSP,Docker**

I.INTRODUCTION

Cloud Computing has offered several applications that can be hosted at the user finger tips. One among them in the current era is medical application that is processed for identifying and predicting various medical diseases in clinical operation for example SNP sequencing. Considering the cost for sequencing such medical procedures had dropped when compared with other intellectual research for example Moore's Law. The amount of characterization a medical application needs calculates over millions of associated agents or companies in certifying it. But speaking over individual medical information it contains its own uniqueness and very sensitive information that is treated highly confidential and private in health care as it carries ethical properties and issues. Processing a medical application as a private set is quite challenging in distributed environment and dynamic environment, has it is affected by vulnerabilities in various dimensions. Research is been carried out on efficient manner to process the medical information to maintain its privacy at multiple agents (like storage unit, medical center, certified institute and so on) Down the decade accessing information in distributed environment over sensitive applications lead with higher complexity without the support of smart devices. Mean while the current era, the situation as changed due to enormous usage of mobile applications that are integrated with remote services like cloud, grid and so on. According to survey report the number of mobile devices or smart devices crosses over the human population by 2020 [1]. On the same time the integration of mobile devices on health care application has reached the sky limit for its regular instances in day to day life. Processing medical applications appeals the mobile platform with respect to time and cost saving respectively.

According to the existing privacy policies that have initiated in accessing the medical information mainly does on SNP sequencing. In order to compare the SNP sequences several proposes have been executed like jha et al. garbled circuiting proposed on privacy technique [2], Canim et al. biomedical information securing with respect to use of cryptographic hardware [5], Ayday et al. for enhancing the privacy policy in order to protect patients sensitive medical information on Cloud Storage Provider (CSP)[6].

In this paper we try to introduce a Hybrid scheme among multiples stakes of cloud that provides an authenticated decision making between CSP and Medical data user. The task that's been considered as follows

1. The medical data stored at the CSP is in the encrypted form, the CSP is formed as group of regional clusters. If the same medical information is present among different cluster which are replicated with same set of encrypted information issued to CSP by an Authorized biological lab.
2. A Cluster analysis is made by considering local and global assignments with the help of control input linear method from our earlier work Sanjay et al (ref) . Than a feedback is provided among the cluster. Simultaneously a user can

retrieve authorization from cloud, later can decrypt his/her medical information on their smart devices.

3. Kubernetes an existing Google frame work is deployed in the architecture for container management which supports in identifying replicated data sets and supports a Master behavior in order to orchestra the processed information sets to CSP.

II. BACKGROUND

2.1 BACKGROUND OF SENSITIVE MEDICAL INFORMATION LIKE SNP

Medical science has provisioned the sequencing of SNP information to an upper bound in order to calculate human medical. Typically it is encoded in to dual standard of bio molecules represents SNP. A literature says that a human medical contains tentatively 3 billion sequences of letters and also these characters are identical for at least 99.5% individuals. The standard nucleotide consists of regular units followed as (A, C, G, and T). For example a medical ancestry contains at least 60 Single Nucleotide Polymorphism (SNP) that can be used as computational processing. SNP are typically measured using two conditional variants. (i) The rare nucleotide which can be said as minor allele (ii) Frequently viewed major allele. The frequency of every SNP is assigned with at least one major and minor allele respectively. For an instance in figure 1 we consider two alleles indexing the positions of SNP [2].

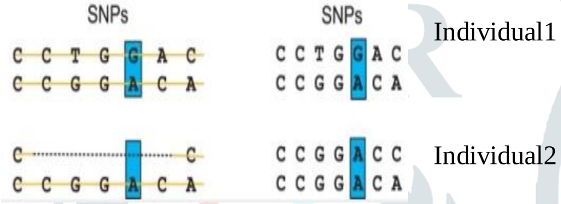


Fig 1.1 Representation of Minor and major Allele [7]

Each individual of course carries one allele for SNP positioning by both the father and mother. It's always variants whenever a SNP drives with one minor allele.

Most of the regular consumer's intensions in choosing cloud for deploying their sensitive medical information are for the following criteria's

1. Privacy issues, as they require distributed environment to make the information strange

2. Application on demand characteristic restricting to deploy the entire architecture set in the same infrastructure provider

3. Consumer expecting upgraded trust, low risk and efficiency

4. Amount of transparency in satisfying the law towards data protection

5. Sensitive applications like medical applications stands at a higher security sight in consumer view as the information are set at cloud environment

6. As the application to be deployed among different multiple provider's storage based on the availability and on-demand features.

2.2 KUBERNETES OVERVIEW

The application deployed over a distributed environment need a fault tolerant and requires access transparency in order to operate within the sand or deployed environment. The use of Kubernetes increases the level of access transparency in managing the distributed application by use of containers. It allows the PaaS service of cloud to regenerate the foreign code, scaling, load balancing, logging, and monitoring [8]. It tries to console and preserve user options and scale up the need of flexibility as it provides the developers a building platform. Few characteristics of Kubernetes to addressed are (a) Never limits the application type and data processing, (b) Continuous Integration, (c) Service brokering among different entities , (d) API provided are declarative and targeted , (e) Fault tolerant and self healing [9]. When we consider the older way of representing an application for its deployment is only on its host by operating system packages. It has a demerit of compatibility with the libraries, configurations and executable operations with other host operating system. An essential rollout and roll backs are required whenever a virtual machine images are created for predictive analysis. But in the new way deployment of containers are completely dependent on operating system virtualization. The containers are remote mutually and from their hosts [8].

III. BRIEF OF PROPOSED SCHEME

In the proposed solution we would like to introduce Kubernetes to integrate with processing of medical information for an efficient service. In this architecture, system represents a container solution by adding up hybrid selection process of nodes in order to serve the authenticated medical user. The architecture represents the following stake holders such as Certified Bio lab [CBL], GlusterFs (Gluster file system), Medical Health Care [MHC], Mobile User, Kubernetes, Gateway Server and Nodes. The advantage of using Kubernetes is it deals with replicated information. As cloud is a distributed environment the medical data may get replicated in different clusters without the knowledge of authorized user. In our proposed work Kubernetes analyzes and orchestra this information to their master leader of each cluster by generating tokens and applies hybrid selection procedure on each cluster. This carries the set of accessed information in to a container. Meanwhile the party who is accessing the medical information is strange to the cloud environment. When a token is generated it is delivered to appropriate authorized user. If an unauthorized access was made on any node which contains the replicated medical information, sequence of tokens is generated among the clusters and tries to match with user identity inside the container. If no token is matched, than remaining clusters been informed not to provision the access for unauthorized party. And a query is generated by the user for both MHC and CSP to check the disease prediction based on his information. GlusterFS is used for storing the accessed information on cloud by a dedicated IP address and node interaction ratio with its master. The GlusterFS tries to stripe the information and backups the node for knowing the service is been issued back to the user without any uncertain access.

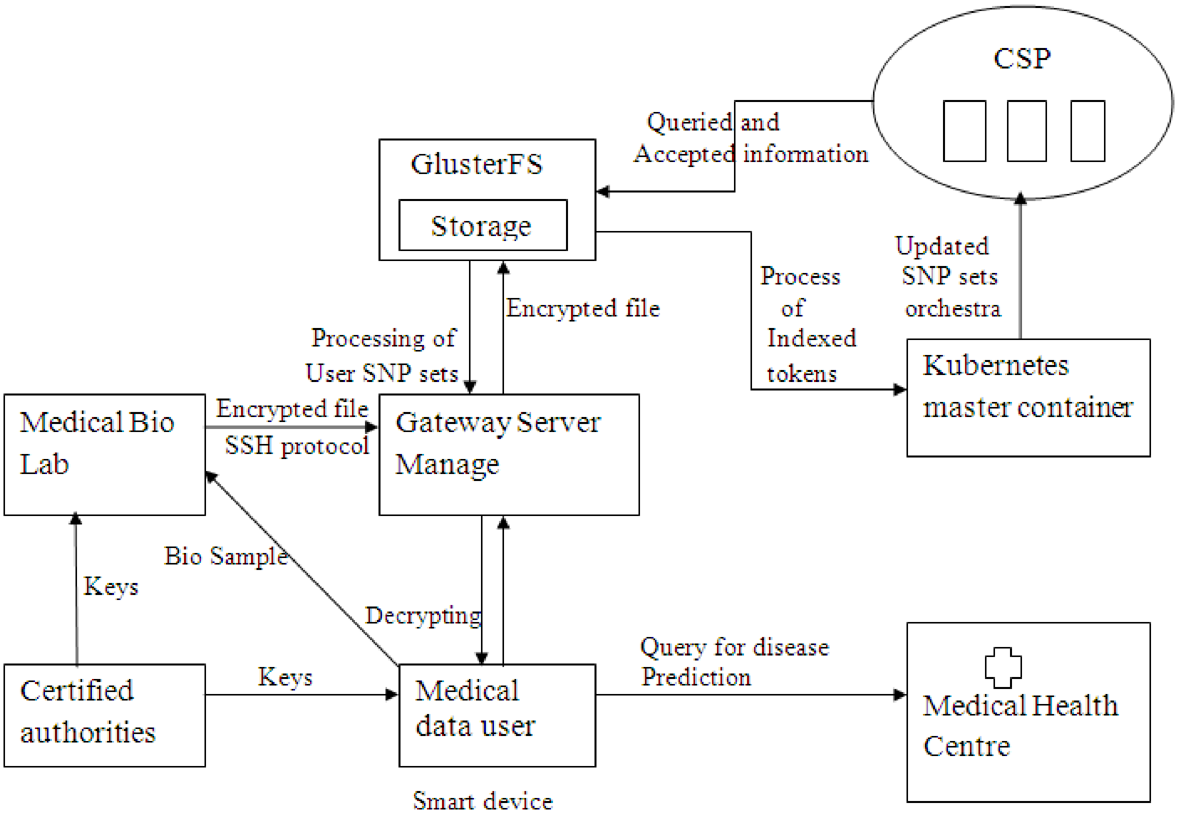


Fig 3.1: Proposed System Architecture

3.1 CONTROL OVER SNP DATA

For this situation a real SNP is being utilized as genetic variants. The SNP is retrieved by the CSP, later stored in GlusterFS and it is issued to Kubernetes container section to host the encrypted file among all the clusters, as Kubernetes tries to manage the replicated information with its charming master and slave method.

Consider an individual/user who is trying to conduct a search on his information that exists in distributed cloud. Let us consider there exist a disease Z, that defines by resting with limited SNPs {SNP$_1$, SNP2, SNP3}. Alleles associated with SNP are represented as to different sets with unique variable representation α_1 and α_2. Now we apply the homogeneity control input law for linear sequencing to identify the individual uniqueness for predictive test function after obtaining a feedback by the manager which is running at containers. For each and every individual a container is launched with appropriate SNP tokens defining the base SNP and respective alleles. The encrypted information is set back to the gateway manager again along with the generated tokens are stored using GlusterFS. In this case SNP information along with tokens is not easier for an attacker to retrieve the SNP straight away as the tokens are distributed among containers and there is no provision for nodes to get compromised in order to serve the unauthorized party. The SNP sequencing is managed by creating image service that manifests the base representation of major and minor SNP. The CSP develops a SNP service using a face of IDE (Integrated development Environment) that later calls the Image Service (IS). The IDE supports the service manifesting. The SNP services are in the form of VM images. Figure 3 represents service manifest and Image Service.

1. The CSP uses GlusterFS for manifesting its service.
2. Control input is applied among all the nodes which holds the SNP images and also over same replicated images.
3. The GlusterFS interfaced with gateway server in order to process tokens to Kubernetes.

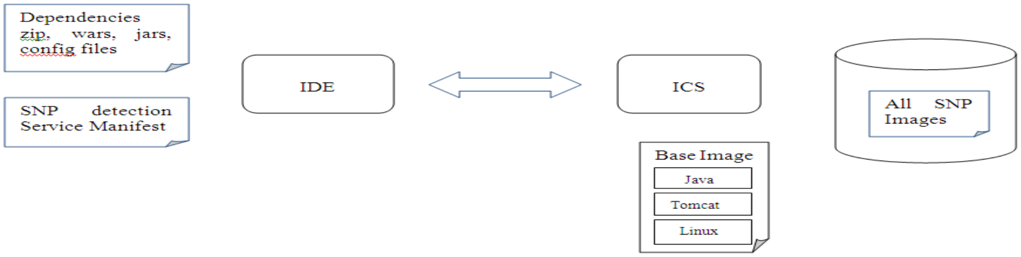

Dependencies zip, wars, jars, config files
SNP detection Service Manifest
IDE
ICS
Base Image
Java
Tomcat
Linux
All SNP Images

Fig 3.2: Image Service Manifest

In order to upload the data the user commonly interacts with gateway manager. Later after receiving the data the gateway manager checks for same SNP sequence image along with token whether it is present or not. If the image is not present the information is carried to GlusterFS to query the information to Kubernetes. There after Kubernetes launches a container by announcing it to cloud that requested user image SNP is not authorized and it is informed to all the clusters masters which are holding original and replicated images without provisioning the service. The replicated SNP's of an individual among different clusters are linearly sequenced in Kubernetes container system as it provides Platform as a Service(PaaS) by applying the control law input of homogeneous feature. Below Algorithm shows the Pseudo of Hybrid Scheme

Algorithm1: Represents Hybrid indexing procedure

```
CSP→SNP                                    //Retrieve SNP from CSP
$A_s$←SNP                                  // SNP from cloud to GlusterFS
Send $A_s$ to $K_s$ container              // to host encrypted file among all clusters
Consider an user trying to search
Z={snp1,snp2,snp3}                         // Disease with limited SNPs
H(a,b) ($\alpha_i$+$x_j$(t+1))             //Apply Homogeneity Control Input Law //
for i=0 to n                               // n is number of users
            Launch container with SNP and tokens
end for
$G_m$← (Encrypted information, tokens)     //Gateway manager
Create image service                       // representation of major and  minor SNP
CSP develops SNP service using IDE
Gateway manager←data
if SNPs==1 && tokens==0                    // Same SNP and token does not present
  then
      $A_s$←$snp_i$                        // List to be indexed
      $K_s$ ←CSP                           // orchestra of  indexed replicated sets
end if
```

3.2 CONSTRUCTION OF HOMOGENEITY CONTROL INPUT ON SNP DATA

3.2.1 Preliminaries: we consider a linear method of homogeneity principle that defines a rank stating *H*(a,b) *H* defines homogeneity term a and b are the two ranks defining with respect to token representation. For private encryption two functions are sequenced i.e., F and G, two searchable arrays denoted as K_s and A_s associated with Kubernetes and GlusterFS respectively. An assumption is made for each file f_i that holds at least one snp_i holds good for all SNP sets.

3.2.2 Hybrid Indexing: we consider medial bio lab that normalize the key pseudonym and encryption of user data over a collection of files indexing f, that are driven to gateway manger using SSH protocol. A list is created for rank monitoring of 0's and 1's which indicates both homogeneity and heterogeneity substituting on a and b. A list associated with SNP is defined as L_{snpi} related to all files f_{snpi}. The sets are located and stored at random locations of cloud cluster as nodes (here we consider VM's). The Kubernetes container launches or hosts the SNP file information along with tokens to cloud with individual node addressing defined as N_i where N_i=<*id, addr(a,b)*+N_{i+1}, here id states the file identifier that contains all snp_i and *(a,b)* are the ranks declaring the probability of *(0,1), (1,0)* and *(1,1)* , therefore address of N is update in the table when all the probability of rank is done.

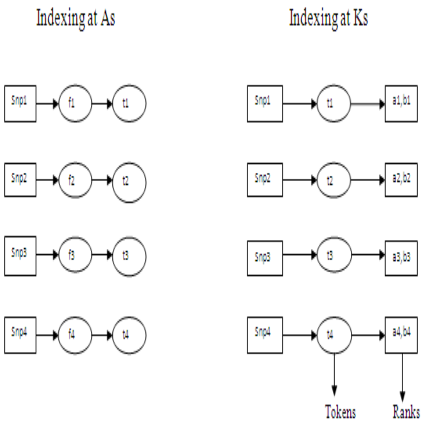
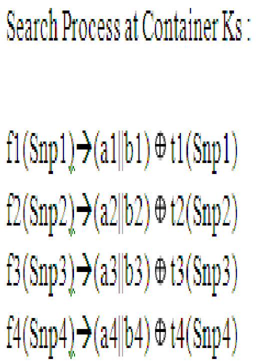


Fig 4.1: Table indexing GFS and Kubernetes



Search Process at Container Ks :

f1(Snp1)→(a1||b1)⊕t1(Snp1)

f2(Snp2)→(a2||b2)⊕t2(Snp2)

f3(Snp3)→(a3||b3)⊕t3(Snp3)

f4(Snp4)→(a4||b4)⊕t4(Snp4)

Fig 4.2: Search Process at container

For all snp_i a list is established in order to select its list L_{snpi}, as a result we have created two indexing scheme for snp sets , they are Kubernetes searchable K_s and A_s for GlusterFS. Indexing of snp starts from snp_1 to snp_n relatively with their functions f_1 to f_n. The indexing is formed by deploying homogeneity control law to four SNP in our example. Suppose an individual wants retrieve the predictive test on his/her device. The conduction of test is associated with A_s searchable index of GlusterFS that inputs four SNPs randomly retrieved from CSP. The user forwards four tokens to the gateway server that is related to randomly generated snps. The GlusterFS accepts the tokens through SSH and verifies existing matches of four snps. The GFS starts the search index A_s with accepted token. If the user token is matched rank is assigned as 1 and treated as true index by delivering it to Kubernetes along with matched SNP and token. Once the rank is assigned the K_s indexing starts to perform its check by applying XOR operation. The control input rank factors a and b are considered to perform this operation in order to check the heterogeneity and homogeneity data. If ***'a'*** is **'0'** than ***'b'*** must be **'1'** this signifies no replication of additional snp has taken on the container, if both the ranks are zero or one on ***(a,b)*** respectively than replication of snp sets to be found and is orchestra to the cloud by Kubernetes master container launcher. The launched information sets from container swarms on the cluster of the cloud by creating index table for local and global clustering sets. The cluster containing the SNP information stores all the entries in it which is launched by container and provisions local target leader represented as T_l that communicates with its neighboring cluster targets represented as global T_g to look up with SNP sets. A searching is done for all the entries of four SNP's in our example, if it is found than primary sets of snp is generated to the local target $T_{l.}$. The local target generates its entire updated index to GFS. There by the user can retrieve his required information sets by decrypting at his end through gateway server.

IV. SECURITY ANALYSIS BASED ON PROPOSED SCHEME

4.1 DENIAL OF SERVICE (DOS)

DoS can be vulnerable in CSP when a user wants to access sensitive medical information. We consider two types of attackers : (1) Fraggle attackers tries to spoof the nodes as if they have already communicated with CSP where original users are avoided to communicate with CSP (2) eavesdroppers who tries to include their time bound by eavesdropping inside the communication channel between multiple entities like MHC and CSP

4.2 CSP'S RESISTANCE

The user's medical information stored at CSP can be encrypted by secret keys at users end; as a result the sensitive or original content of the information can't be accessed by CSP. Even at a level most the CSP is strange about the user's identity as pseudonyms will be generated for any queries among MHC or Biological labs. As a result the SNP's are retrieved at encrypted form from the CSP to GFS and tokens being exchange among Kubernetes and CSP in hybrid manner, so the CSP is not provisioned in any instance in order to obtain the user identity in any manner.

4.3 RESISTANCE TO VULNERABLE ATTACKS OF THIRD PARTIES.

The cloud can be said as a honest sector but very curious hunter on users data. There may be situations for cloud to get compromised with strange users as they try to dominate over user information sets. In our system the medical SNP information of a user is been indexed in a hybrid manner at multiple level by token generation, by applying homogeneity ranks, as it provides the users privacy at middle transactions by proper decision making process.

VI. EVALUATION

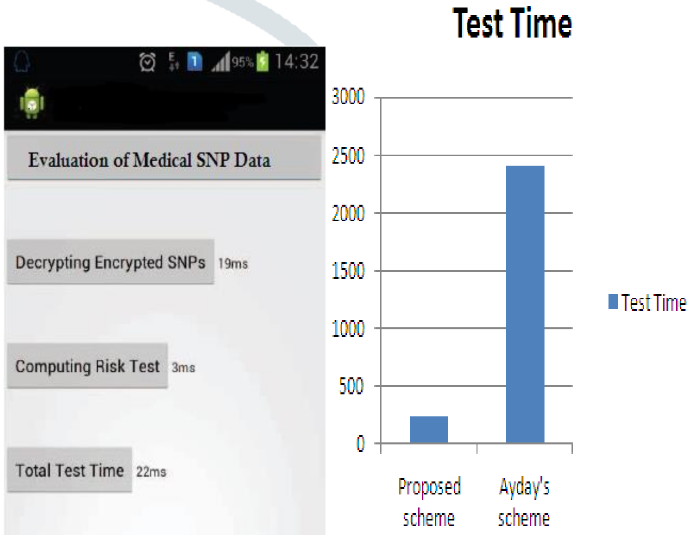
6.1 SIMULATION PHASE

The Hybrid scheme is been evaluated by functioning the processed information at users end using a smart android application. For implementing Kubernetes we have considered Linux Operating system in order to launch the containers by creating single node and multiple threads. The android launching is favored by JAVA platform to compute risk test and function the decryption process. The homogeneity law applied in the assignment of SNP functioning is evaluated using 12 SNPs. In our implementation we have used AES key for decryption process which is of 256 bit for all the encrypted SNPs which are retrieved from CSP.

6.2 RESULTS

The experiment we conducted proves that when the law of homogeneity is indexed the replicated information orchestra to the cloud and retrieving that information and decrypting it takes a lesser time, as it already encrypted SNP sets are sequenced in a hybrid manner i.e. Indexed at GFS and Kubernetes. As it is already indexed at GFS the Gateway manager processes it to the user with in lesser time. In our test we have considered 12 SNPs and their decryption time at users end. Later after we compute the time cost based on the risk assessment from our security analysis discussed above. The Container is launched as soon as user generates their token so the load of the container is also to be considered in order to transact for the entire users request with the cloud. By using this Control homogeneity law a single node container is launched efficiently that manages with at least one cluster. When the user request are at higher rate than the rate of CPU consumption hits with higher data risk rate.

The below figures represents the android User Interface(UI) and Kubernetes Container launcher. The term images in the figure refer to SNP encrypted images discussed in the section (Control over SNP data) creation of Image services. In our iteration we consider only two SNP images for every 0.1 sec. totally we have come across 6 iterations for all our 12 SNP sets. The evaluation result is compared with the Ayday's protocol [6] which specifies the maximum time utilization for decrypting the SNP information. Our proposed solution consumes a minimal time of 19ms to decrypt the provided encrypted SNP sets from the service provider. The risk test is performed by considering synthetic allele information of various users in order to compute the risk for each and every user SNP's referring to [16]. In the figure 6 it reports total test time when compared with Ayday's work i.e., our proposed solution provides 22 ms after launching the Kubernetes container.



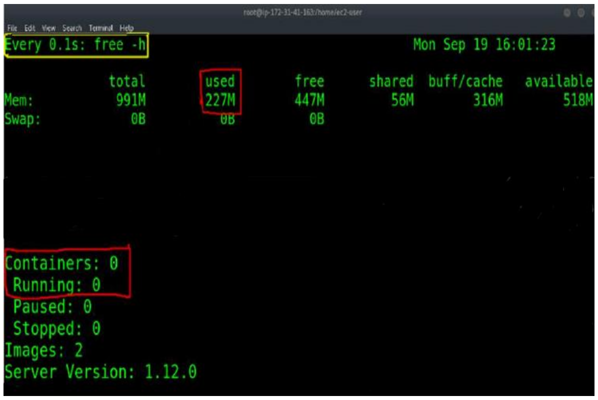
Evaluation of Medical SNP Data

Decrypting Encrypted SNPs 19ms

Computing Risk Test 3ms

Total Test Time 22ms

Fig 6.1 Mobile device indicating Decrypting and Test time

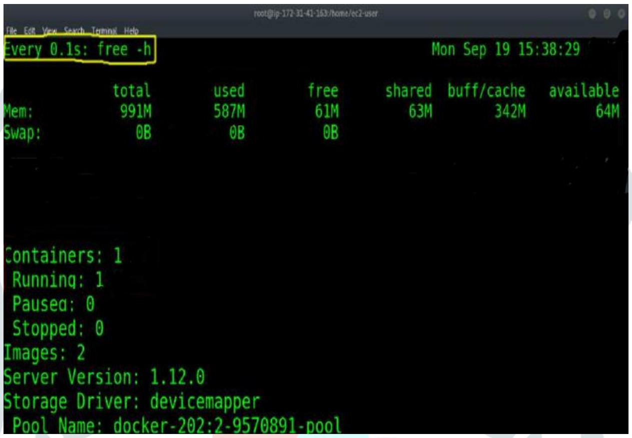


```
root@ip-172-31-41-163:/home/ec2-user
File Edit View Search Terminal Help
Every 0.1s: free -h                                    Mon Sep 19 16:01:23

              total        used        free      shared  buff/cache   available
Mem:           991M        227M        447M         56M        316M        518M
Swap:            0B          0B          0B

Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 2
Server Version: 1.12.0
```

Fig 6.2 : VM Images without Orchestra of SNP sets



```
root@ip-172-31-41-163:/home/ec2-user
File Edit View Search Terminal Help
Every 0.1s: free -h                                    Mon Sep 19 15:38:29

              total        used        free      shared  buff/cache   available
Mem:           991M        587M         61M         63M        342M         64M
Swap:            0B          0B          0B

Containers: 1
 Running: 1
 Paused: 0
 Stopped: 0
Images: 2
Server Version: 1.12.0
Storage Driver: devicemapper
 Pool Name: docker-202:2-9570891-pool
```

Fig 6.3: VM Images with Orchestra of SNP sets

VII. CONCLUSION

Cloud Computing has evolved in acquiring prior position in distributed computing forum for processing and storing of users information. Recent trends of e-commerce and medical information processing has started their attentions on cloud forum in order to process their information for business processing. Likewise in our work we have recommended a Control input law that provides homogeneous decision making in hybrid manner by deploying Kubernetes system to identify replication sets of SNP medical information. The proposed scheme satisfies the user security concerns with lower computation rates and minimal decryption process at their smart device. Also the proposed scheme provides a hybrid behavior indexing by proving privacy of individual medical user. The results show that our scheme takes a lesser computation time to run both decryption and risk test when compared with Ayday’s results.

VIII. Acknowledgement

We convey special thanks to our Research Centers SAHE and PET , and also convey our thanks to GIMS Bangalore.

REFERENCES

[1] “Cisco visual networking index: Global mobile data traffic forecast update, 2013-2018.”

[2] S. Jha, L. Kruger, and V. Shmatikov, “Towards practical privacy for genomic computation,” in *the 2008 IEEE Symposium on Security and Privacy (IEEE SP 2008)*, Oakland, California, USA, 18-21 May,2008.

[3] F. Bruekers, S. Katzenbeisser, K. Kursawe, and P. Tuyls, “Privacypreserving matching of dna profiles.” *IACR Cryptology ePrint Archive*, vol. 2008, p. 203, 2008.

[4] P. Baldi, R. Baronio, E. De Cristofaro, P. Gasti, and G. Tsudik, “Countering gattaca: efficient and secure testing of fully-sequenced human genomes,” in *the 18th ACM conference on Computer and communications security(ACM CCS 2011)*, Chicago, Illinois, USA, October 17-21,2011.

[5] M. Canim, M. Kantarcioglu, and B. Malin, “Secure management of biomedical data with cryptographic hardware,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 1, pp. 166–175, 2012.

[6] E. Ayday, J. L. Raisaro, M. Laren, P. Jack, J. Fellay, and J.-P. Hubaux,“Privacy-preserving computation of disease risk by using genomic, clinical,and environmental data,” in *the 2013 USENIX Security Workshop on Health Information Technologies (HealthTech 2013)*, Washington, D.C,USA, August 12, 2013.

[7] Xiaosan Lei, Xiaoyan Zhu, Haotian Chi, Shunrong Jiang *“*Privacy-Preserving Use of Genomic Data on Mobile Devices” IEEE/CIC ICCC 2015 Symposium on Privacy and Security in Communications.

[8] https://www.tutorialspoint.com/kubernetes/kubernetes_tutorial.pdf

[9] https://people.redhat.com/~eparis/kubernetes/kube.pdf

[10] Con Wang, Qian Wang, Kui Ren Wenjing, “Ensuring Data Storage Security in Cloud Computing”,Quality ofService,2009.IWQoS. 17th International Workshop,DOI:10.1109/IWQoS.2009.5201385

[11] P.G. Dorey, A. Leite, “Commentary: Cloud computing A security problem or solution?”,Journal Information Security tech.Report[archive],August 2011

[12] Rohit Bhadauria , Sugata Sanyal,”Survey on Security Issues in Cloud Computing and Associated Mitigation Techniques”,International Journal of Computer Applications,

[13] Rohit Bhadauria,Rituparna,Chaki,Nabendu Chaki,Sugata Sanyal,”A Survey on Security Issues in Cloud Computing “,IEEE Communications Surveys and Tutorials, 1-15.2011

[14] S. Subashini , V. Kavitha,”A survey on security issues in service delivery models of cloud computing ”Journal of Network and Computer Applications[archive],January 2011

[15] Mohamed Al Morsy, John Grundy, Ingo MÃ¼ller,”An Analysis of The Cloud Computing Security Problem.”,2010 Asia Pacific Cloud Workshop,January 2010

[16] http://www.ncbi.nlm.nih.gov/projects/SNP/.

[17] Shun-Sheng Wang, Kuo-Qin Yan, Shu-Ching Wang,”Achieving efficient agreement within a dualfailure cloud-computing environment”, Expert Systemswith Applications: An International Journal(archieved), Volume 38 Issue 1, January, 2011. pp:906-915, ISSN: 0957-4174.

[18] Brototi Mondal,Kousik Dasgupta,ParamarthaDutta,”Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach”, 2nd International Conferrence on Computer, Communication, Control and InformationTechnology(C3IT-2012), February 25-26,2012, volume4, pp 783-789,

[19] Weiwei Lin, Deyu Qi,James Z. Wang ,Chen Liang ,”A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing.”Procedia Engineering 23(2011) 695-703

[20] Latifa Ben Arfa Rabai, Mouna Jouini,Anis Ben Aissa,Ali Mili,”A cybersecurity model in cloud computing environments.”Journal of King Saud University-Computer and Information Sciences,vol-25,issue 1,January 2013

[21] Divya Muntimadugu, Anjana Suparna Sriram, “Red Hat Storage 2.0 Administration Guide”.

[22]Arun gupta “Clustering Using Docker Swarm 0.2.0”, April 23,2015,http://blog.arungupta.me/clusteringdocker-swarm-techtip85

[23] Dr. Yong Yu, Prof. Atsuko Miyaji, Dr. Man Ho Au**,** Prof. Willy Susilo, **“**Special Issue on Cloud Computing Security and Privacy: Standards and Regulations”