

Construction of a Taiwan biodiversity genetic database in cloud environments

Han-Yu Lin, Chien-Ming Chen, Yu-Lun Chen, Ya-Hui Chang, and Tun-Wen Pai*

Department of Computer Science and Engineering

National Taiwan Ocean University, Taiwan

* twp@mail.ntou.edu.tw

Abstract—Biodiversity is commonly used to measure the degree of variation of life forms and health conditions within an area or ecosystem. However, reckless human behaviors and dramatic climate changes inevitably cause species extinction. It is therefore important and urgent to establish genetic databases worldwide for preserving precious genome sequences of diversified organisms and perhaps restoration for endangered species. To integrate such abundant genetic data, it requires a robust, stable, and secure mechanism for reliable data storage and retrieval. Here, we have constructed a Taiwan biodiversity genetic database within a cloud environment. The database contains Taiwan related gene sequences from NCBI and local researchers, and it is daily updated for data synchronization. We also collected genome data of various model species from Ensembl for cross-species comparison and functional annotation. To realize the proposed database for such extremely large amount of genome data, cloud computing techniques indeed provide an adequate solution and a mature environment for the exponentially increasing genetic sequences. Besides, important issues for constructing a biodiversity genome database such as data integrity, confidentiality efficiency, availability, and reliability are concerned and discussed in this paper.

Keywords– Biodiversity, cloud computing, genetic database, security, authentication.

I. INTRODUCTION

In 1992, the Convention on Biological Diversity (CBD) was proposed at the Earth Summit [5] to highlight the importance of biological diversity (abbreviated to biodiversity) and more than 193 countries signed to support the treaties. Biodiversity represents the variability of living organisms from terrestrial, marine, aquatic and air ecosystems. Since each species is interlinked with others in their ecosystem, vanishment of certain species will inevitably endanger the survival of closely related ones or even the health of human beings. Accordingly, the delicate balance of biodiversity strongly affects the health of ecosystems and mankind. Generally speaking, the biodiversity is categorized into three levels including habitat, species, and genetic diversity. In this paper, our application of cloud computing techniques focuses on the genetic level of biodiversity, and we have discussed the problems of data storage and security issues for biological researchers. Here, each individual organism is characterized by deposited genetic sequences obtained through various DNA sequencing techniques. The stored DNA sequences contain four fundamental nucleotide bases, adenine(A), guanine(G),

cytosine(C), and thymine(T) within a specific combination and sequential order. Variant sequence numbers, lengths and contents of DNAs are expected for different species, even for the same kind of species but different individual entities. Hence, the problems of collecting huge amount of genome data from variant resources, protecting unpublished data for genetic data suppliers, and identifying the original ownership authentically are raised for consideration and exploration in a cloud environment.

For the past few years, biological data such as genome sequences have been dramatically generated due to advance techniques of next generation sequencing (NGS) [12, 16, 19] including Roche 454, Illumina Solexa, ABI SOLiD and Helicos tSMS. Each sequencer generates high throughput data with low cost and provides precious biological knowledge in molecular level. Processing and storing such large data and performing computational algorithms become a big challenge for single workstation. Another important and critical issue is how to protect unpublished genetic data from being plagiarized, and which is always the major concerns of unwillingness of data deposits for public use.

One of possible solutions for processing huge amount data is adopting the cloud computing techniques [3, 23] which provide an on-demand service model based on virtualization and distributed computing technologies. A significant property of cloud computing exploits the abstraction of computing, network and storage infrastructure, and the advantages of utilizing cloud computing architectures include scalability, flexibility, mobility, reduced cost and shared resources in terms of software, hardware and database subjects. Generally speaking, the categories of cloud computing can be classified into three types of development including infrastructure as service (IaaS), platform as a service (PaaS), and software as a service (SaaS). All these various services are designed and utilized in this proposed application regarding to a biodiversity genome database. Regarding to the issue of security in cloud environments, Kamara and Lauter [13] described several architectures combining cryptographic primitives to construct a secure cloud storage service in a public cloud infrastructure for those customers who do not completely trust the service provider. In their architectures, a data processor (DP) and a data verifier (DV) are responsible for processing and verifying the integrity of data, respectively. Another token generator (TG) enables the cloud storage provider to retrieve segments of customer data. When storing a data entry to the cloud, the DP

initiates an index and then encrypts it under a unique symmetric key. The index and the symmetric key are also encoded through a searchable encryption scheme and an attribute based encryption mechanism, respectively. Consequently, the DV verifies the data integrity by using a technique called proof of storage. In 2011, Cao *et al.* [1] designed and implemented a cloud data storage system for supporting both OLTP and OLAP workloads efficiently. They developed a unified indexing framework including hash indices for exact queries and distributed multi-dimensional indices for multi-dimensional queries. Their system was built based on the epiC project which aims to develop an elastic power-aware data-intensive cloud computing platform which can provide general-purpose large-scale services. They also conducted extensive experiments to demonstrate the performance of their storage system. From the perspective of real world service environments, Park *et al.* [18] also discussed issues on the network security in cloud computing. According to their surveys, the security risks in cloud computing include privileged user access, regulatory compliance, data location, data segregation, recovery in the case of disaster, investigative support and long-term viability. Considering these threats, several groups and organizations have devoted themselves to improve the cloud computing security. These organizations include: Cloud Security Alliance (CSA), Open Cloud Consortium (OCC), Storage Networking Industry Association (SNIA), European Network and Information Security Agency (ENISA), and National Institute for Standards and Technology (NIST). Based on the identity-based hierarchical model for cloud computing (IBHMCC), Li *et al.* [15] proposed a new identity-based authentication protocol for cloud computing and services. The proposed protocol with great scalability is suitable for massive-scale clouds, and the simulation tests have shown that the proposed protocol outperforms the traditional SSL authentication protocol (SAP).

According to previous discussions, to construct a large-scale biodiversity genetic database in cloud environments, the issues of data security, on-line tool computing efficiency, and system reliability should be considered simultaneously. Here, we report how we constructed the Taiwan biodiversity genetic database within a cloud environment and we also discuss security essentials of the large-scale genetic database in the following sections.

II. CLOUD BIODIVERSITY GENOME DATABASE

Since the first NGS, 454 GS 20, was released in 2005, worldwide sequence data have been exponentially generated. For example, the current 454 FLX sequencer can generate an average of 100 megabases of average read lengths of 250 bases per 8-hour run [20]. Adopting NGS techniques to provide large-scale sequenced data in genomics research, many practical problems are accompanied. To confront such abundant amount of genetic data, we adopted the software of Apache Hadoop [9] and constructed a reliable, scalable and distributed environment for cloud computing. The software framework of MapReduce [6] enables distributed processes of large data to promote efficiencies. Additionally, the distributed database, HBase [10], is suggested to store biodiversity annotated information. Figures 1 and 2 depict the HBase architecture and the system architecture of the proposed cloud

biodiversity genome database. In the HBase architecture, HMaster is responsible for managing all HRegion Servers while the latter can control its own HRegions to insert, modify and delete tables, etc. Therefore, all database procedures can be processed in a parallel version within all distributed HRegion Servers to gain better efficiencies. The HBase system is originally designed for the hadoop HDFS systems, and it is suitable for a petabyte level data storage. Furthermore, the column based storage mechanism is very flexible for various types of biodiversity annotation data. However, these advantages are not obviously revealed for limited model species. First, the number of well-studied model species is much smaller than the number of species included in biodiversity research. Second, the biological annotations for model species are usually well organized without rapidly scheme changes. Therefore, the traditional relational databases are relatively preferred for limited amount of model species information. It should be noted that the proposed database in this paper at current stage, most biological annotation terms are acquired from public model species databases. Due to the consideration of quantity, all of these annotations are currently implemented by utilizing the relational database mechanism to gain better efficiencies, and the HBase will be applied when the amount of growing biodiversity data reaches at a petabyte level in the near future.

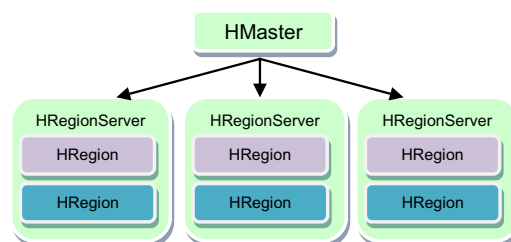


Figure 1: The HBase architecture.

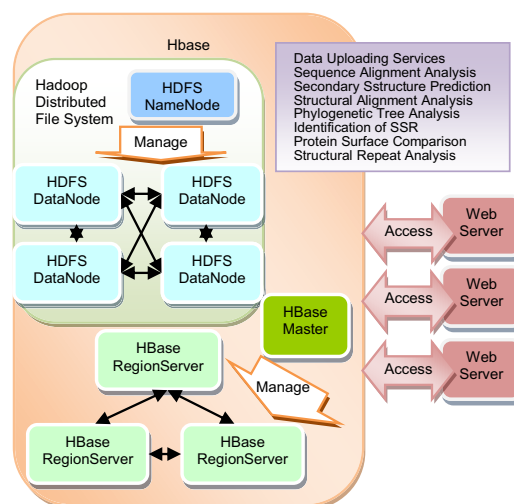


Figure 2: The system architecture of the cloud biodiversity genome database.

In Figure 2, we established the Hadoop distributed file system (HDFS) in each clustered computer. The HDFS is basically the client-server architecture composed of NameNode and DataNode entries. The former is responsible for managing the later elements. The storage space in HDFS can be extended by simply adding new DataNode server into the network, which provides great extendibility and suitable for the continually growing biological sequencing throughput. The combination of the HDFS and MapReduce framework also provide the solutions of massive sequence analysis.

According to different levels of bioinformatics data, we established the database schema in Figure 3, which includes possible fields of biological features for future biocuration processes and workflows. Each rectangle represents a relation and its corresponding attributes, and each solid line shows a foreign key matched with another relation. For example, the relation of “Sequence” possesses 11 attributes (including ID, User, Species and etc.). Among them, the attribute of User in the “Sequence” relation is also a foreign key referencing to the attribute of ID in the “User” relation. Our system is not only a repository for biological sequences and structures, but also provides various online analysis tools. These bioinformatics tools include: data uploading services, sequence alignment analysis, secondary structure prediction analysis, structural alignment analysis, phylogenetic tree analysis, identification of short sequence repeats (SSR), protein surface comparison, structural repeat analysis.

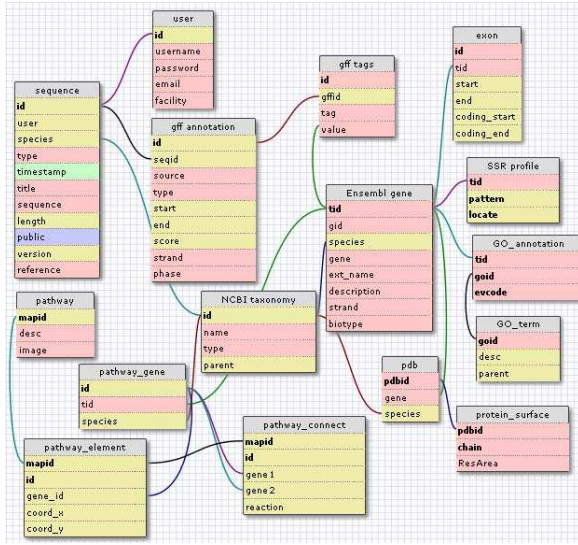


Figure 3: The schema of cloud biodiversity genome database.

III. SECURITY STRATEGIES OF CLOUD BIODIVERSITY GENOME DATABASES

To confront the security issues of biodiversity genome databases in cloud computing environments, we proposed several security strategies to ensure confidentiality, data integrity and availability of the genome data, and these subjects are discussed in the followings:

A. Encryption of Bioinformatics Data:

To ensure the confidentiality of uploaded data, it is necessary to transmit them through SSL channels and then encrypt sensitive fields with robust cryptographic algorithms. It is recommended to utilize symmetric algorithms such as 3DES or AES [17]. The minimum key size should be 128 bits to resist possible active attacks. As genetic data like DNA sequences are usually very large, asymmetric cryptosystems [7, 8, 21, 22] which are based on some intractable mathematical assumptions might not be suitable here.

B. Encryption/Decryption by Standalone Crypto Server:

There are two options to perform data encryption: the first option adopting DBMS and the other one employing an independent crypto server. The advantages of the former approach avoid extra communication overheads and hardware equipment. However, it will increase the burden of DBMS. On the contrary, if we adopt an additional crypto server, the encryption/decryption procedures are off-loaded from DBMS and the selection of encryption algorithms becomes more flexible. The data flow of this architecture is depicted in Figure 4.

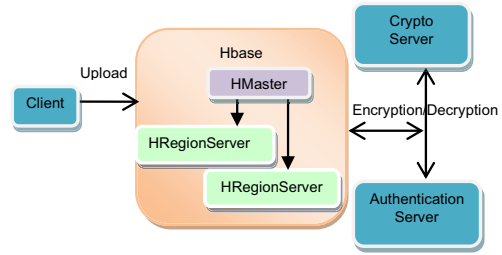


Figure 4: The data flow of standalone crypto server.

C. Separation of Keys from Ciphertexts:

According to Kerckhoffs principle [22], crypto algorithms are not secure, but only the key holds secrecy. Consequently, the storage and management of keys are critical to data confidentiality. To reduce the risk of key exposure, it is recommended to separate the keys from ciphertexts, i.e., to store the keys in other hardware to prevent malicious administrators from direct accessions. The second protection mechanism is to store the hashed keys. A crypto hash function provides the properties of compression, efficiency and one-way characteristics. A strong collision-resistant hash function $h(\cdot)$ [4, 17] further satisfies that it is infeasible to find any x and y , with $x \neq y$ such that $h(x) = h(y)$. Hence, even if an adversary obtains the hashed key files successfully, he would have no way to convert it.

D. Strong User Authentication:

Since it is impossible to convert a hashed file, an adversary might attempt to impersonate legitimate users. We must ensure that only authenticated users can access the key. Traditionally, authentication can be verified based on three major types [22]:

(a).Something you know: For example, a password.

(b).Something you have: This is usually a physical thing in your possession, e.g., a smartcard, a constrained IP address, or a specific MAC address with respect to your network card adapter.

(c).Something you are: It refers to biometric identification such as fingerprinting, iris scanning and facial recognition, etc.

Each of the above mentioned approaches possesses its corresponding merits and drawbacks. To enforce a stronger authentication mechanism, at least a two-factor authentication is suggested [22], i.e., two out of three mentioned verification approaches should be adopted simultaneously. Here, to build a secure cloud biodiversity genome database, we have employed a password along with a specific IP address to identify authenticated users.

E. Privacy Issues:

In a cloud environment, the security and privacy are often two major concerns. The data confidentiality can be ensured by adopting secure encryption algorithms while the protection of user privacy has to be enforced by strict access control and management. An access control matrix storing all subjects (users) in rows and objects (resources) in columns is a common approach. Before access to any resource is allowed, the system has to check this control matrix initially. However, it might be difficult to manage a large access control matrix. A practical solution is to split the matrix into more manageable pieces. One way called access control lists (ACLs) is to store the matrix by columns, and the other called capabilities (C-Lists) is to store it by rows. An obvious difference between these two approaches is that the former is object-oriented while the latter belongs to subject-oriented. In practice, since ACLs often cause the problem of confused deputy [22], C-Lists provides an easy way to delegate authority. In our cloud biodiversity genome database, we also adopt the C-Lists to enforce access control for protecting privacy of users.

F. Event Logging:

Event logging is a basic security protection mechanism. The logs can be viewed as important digital evidences for possible computer forensic procedures[2, 11, 14]. According to the data flows depicted in Figure 1, a system manager should regularly monitor suspicious activities and audit logs in all hardware including HBase, crypto, and authentication servers.

G. Traceability of Stolen Biological Data:

Once partial unpublished biological sequence data, say an important DNA sequence, is unauthorizedly released from our database system or intentional plagiarized by immoral persons. The system should show its ability to convince anyone of its origin. The first step is to publish the Message Authentication Code (MAC) [17, 22] of incoming data whenever a legitimate user uploads his research data. A MAC is computed as the Cipher Block Chaining (CBC) residue. The system announces the information of (ID || IV || MAC) where ID is the identity of the uploading user and IV is a chosen initialization vector. The corresponding key is managed by the server. To claim that part

of deposited biological data is stolen from our database system, we just release the key information and therefore anyone can verify the correctness of MAC with the key. The detailed steps are shown as follows:

Step 1: Let a user U_a associated with the identity ID_a upload an unpublished data $m = m_0 || m_1 || m_2$ to the server S .

Step 2: S first chooses a random vector IV_a and an encryption key K_a to compute

$$c_0 = E(IV_a \oplus m_0, K_a), \quad (1)$$

$$c_1 = E(c_0 \oplus m_1, K_a), \quad (2)$$

$$c_2 = E(c_1 \oplus m_2, K_a) = MAC_a. \quad (3)$$

Note that the symbol 'E' is the adopted encryption algorithm such as AES or 3DES.

Step 3: S publishes ($ID_a || IV_a || MAC_a$) or delivers the information to a trusted third party (TTP).

Step 4: When the secret data m is accidentally compromised, U_a can provide his key K_a to the TTP. The authentication representative can then utilize stored IV_a to derive a new MAC_a^* by Eqs. (1) to (3). If MAC_a^* is equal to MAC_a , the original of data m is authenticated.

Another possible solution is to take the concept of steganography to embed hidden messages into the stored biological data. For example, one can add timestamp information together with the identity of uploading users into the deposited data. These hidden messages must not be directly perceivable to avoid easy removal by plagiarisms. Moreover, the adopted techniques of steganography should be robust to tolerate any on-line comparative and analytical bioinformatics tools.

IV CONCLUSIONS

In this paper, we have discussed and developed a system architecture with secure aspects for a biodiversity genome database in cloud computing environments. With respect to the gene collection task, we have retrieved 53 updated model species and their corresponding 5,617,510 sequences from Ensembl. Besides, 38,665 gene sequences from NCBI deposited by Taiwanese researchers were also retrieved and stored in the constructed database. The biological data for model species from Ensembl is updated according to the newest version, while the data from NCBI is updated automatically by daily checking increased files. Several security strategies employed in this study to ensure data integrity, confidentiality and availability are also proposed and discussed. We expect that our designed mechanism and hypothetical discussions explore potential benefits to the important construction of biodiversity genome databases, and which can help system developers to process increasingly large-scale genetic data with a more efficient and secure way.

ACKNOWLEDGMENT

This work was sponsored by National Science Council, Taiwan, R.O.C. (NSC 100-2321-B-019-004 to T.-W. Pai).

REFERENCES

- [1] Y. Cao, C. Chen, F. Guo, D. Jiang, Y. Lin, B. C. Ooi, H. T. Vo, S. Wu and Q. Xu, "A cloud data storage system for supporting both OLTP and OLAP," *Proceedings of IEEE International Conference on Data engineering (ICDE)*, 2011, pp. 291-302.
- [2] E. Casey, *Handbook of Computer Crime Investigation*, Academic Press, 2002.
- [3] Cloud computing.
http://en.wikipedia.org/wiki/Cloud_computing
- [4] S. Contini, A.K. Lenstra and R. Steinfeld, "VSH, an efficient and provable collision-resistant hash function," *Advances in Cryptology – EUROCRYPT'06*, pp. 165-182, 2006.
- [5] Convention on Biological Diversity: List of Parties.
<http://www.cbd.int/information/parties.shtml>
- [6] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, Vol. 51, No. 1, 2008, pp. 107-113.
- [7] H. Delfs and H. Knebl, *Introduction to Cryptography: Principles and Applications*, Springer, 2002.
- [8] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. IT-22, No. 6, 1976, pp. 644-654.
- [9] Hadoop. <http://hadoop.apache.org/>
- [10] HBase. <http://hbase.apache.org/>
- [11] A. Hellany, H. Achi and M. Nagrial, "An overview of digital security forensics approach and modeling," *Proceedings of International Conference on Computer Engineering & Systems (ICCES 2008)*, IEEE Computer Society, 2008, pp. 257-260.
- [12] R. A. Holt and S. J. M. Jones, "The new paradigm of flow cell sequencing," *Genome research*, Vol. 18, No. 6, 2008, pp. 839-846.
- [13] S. Kamara and K. Lauter, "Cryptographic cloud storage," *Financial Cryptography and Data Security*, Vol. 6054, 2010, pp. 136-149.
- [14] M. Karyda and L. Mitrou, "Internet forensics: legal and technical issues," *Proceedings of the Second International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007)*, 2007.
- [15] H. Li, Y. Dai, L. Tian and H. Yang, "Identity-based authentication for cloud computing," *Proceedings of the 1st International Conference on Cloud Computing (CloudCom 2009)*, LNCS 5931, 2009, pp. 157-166.
- [16] E.R. Mardis, "The impact of next-generation sequencing technology on genetics," *Trends Genet*, Vol. 24, No. 3, 2008, pp. 133-141.
- [17] A. Menezes, P. Oorschot and S. Vanstone, *Handbook of applied cryptography*, CRC Press, Inc., 1997.
- [18] T. H. Oh, S. Lim, Y. B. Choi, K. R. Park, H. Lee and H. Choi, "State of the art of network security perspectives in cloud computing," *Communications in Computer and Information Science*, Vol. 78, 2010, pp. 629-637.
- [19] M. Pop and S. L. Salzberg, "Bioinformatics challenges of new sequencing technology," *Trends Genet*, Vol. 24, No. 3, 2008, pp. 142-149.
- [20] J. M. Rothberg and J. H. Leamon, "The development and impact of 454 sequencing," *Nature biotechnology*, Vol. 26, No. 10, 2008, pp. 1117-1124.
- [21] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 5th Ed., Pearson, 2011.
- [22] M. Stamp, *Information Security: Principles and Practice*, Wiley, 2006.
- [23] L. J. Zhang and Q. Zhou, "CCOA: cloud computing open architecture," *Proceedings of 2009 IEEE International Conference on Web Services (ICWS)*, 2009, pp. 607-616.