Bulletin of Electrical Engineering and Informatics
Vol. 10, No. 5, October 2021, pp. 2707~2715
ISSN: 2302-9285, DOI: 10.11591/eei.v10i5.2212

Efficient incremental data backup of unison synchronize approach

Prakai Nadee, Preecha Somwang
Faculty of Engineering and Architecture, Rajamangala University of Technology Isan, Thailand

Article Info

Article history:

Received Feb 11, 2020
Revised Apr 29, 2021
Accepted Jul 13, 2021

Keywords:

Asynchronous
Data backup
Information technology
Synchronous
Unison

ABSTRACT

Data communication and computer networks have enormously grown in every aspect of businesses. Computer networks are being used to offer instantaneous access to information in online libraries around the world. The popularity and importance of data communication has produced a strong demand in all sectors job for people with more computer networking expertise. Companies need workers to plan, use and manage the database system aspects of security. The security policy must apply data stored in a computer system as well as information transfer a network. This paper aimed to define computer data backup policies of the Incremental backup by using Unison synchronization as a file-synchronization tool and load balancing file synchronization management (LFSM) for traffic management. The policy is to be able to perform a full backup only at first as a one time from obtaining a copy of the data. The easiest aspect of value to assess is replacement for restoring the data from changes only and processing the correct information. As a result, the new synchronization technique was able to improve the performance of data backup and computer security system.

Corresponding Author:

Prakai Nadee
Faculty of Engineering and Architecture
Rajamangala University of Technology Isan
744 Suranarai Road, Muang, Nakhon Ratchasima, 30000, Thailand
Email: prakai@rmuti.ac.th

1. INTRODUCTION

Nowadays, the internet has grown into a production communication system that connects people around the world [1]. The growth of the global computer network has an economic impact as well [2]. The problem of computer system is the risk of being attacked by intruders via computer network or computer viruses which damages data become and makes it unusable. In addition, computer data backup and data security are very important tools for all business sectors [3]. Computer data backup is a complex subject in many technologies, and each one has different features from the other [4]. Commercial data backup products and services that use the technologies in unconventional ways have been created. There are three importnant aspects that make data secure over a computer network; confidentiality, integrity and availability [5].

Confidentiality is protection against unauthorized data access via snooping or wiretapping from an intruder [6]. Integrity is protection from altering the data so that it arrived at the recipient exactly as it was sent. Availability is the prevention of service interruption that keeps data accessible for legitimate uses. However, the major problem in full backup is that data takes a long time to complete backup due to the need for large storage resources. It is very hard to restore an archived data which needs collection of the information. As a result, incremental backup plays an important role to data backup in computer network system [7]. To perform incremental backup, various techniques have been widely applied, such as synchronous and asynchronous

techniques. Nevertheless, existing techniques have limitations for duplicate data storage techniques [8]. This is because duplicate data storage offers the most assorted set of data in terms of backup execution, implementation, and dynamics [9]. Thus, this paper proposes a new data backup technique mainly focusing on Incremental backup by using Unison synchronize technique. The proposed technique combines the Unison used for a file-synchronization tool and load balancing file synchronization management (LFSM) for traffic management and content analysis. This article presents methods for improving the efficiency of the Rajamangala University of Technology Isan backup environment and adapting to changing systems. The rest of this paper is organized as; section 2 discusses background and related works of data backup system. Section 3 explains the synchronize technique. Section 4 describes the methodology, Unison and LFSM. Experiment results are shown in section 5. Finally, conclusion is in section 6.

2. BACKGROUND AND RELATED WORKS

Many techniques of data backup system have been proposed and categorized as full-backup, differential backup and incremental backup techniques.

2.1. Full-backup

A full backup is the starting point and required for all other backup methods, because it contains a complete copy of all the folders and files in the storage space. It considered to the best storage management in terms of a single file in faster and simpler restoration operations [10]. However, making full backups all the time imposes considerable workload on the computer network. Full backup is a process limited to a weekly or monthly schedule due to the required large volume of data storage to be copied. The data storage space needs to have a largestorage capacity in the backup repository, as shown in Figure 1 [11].

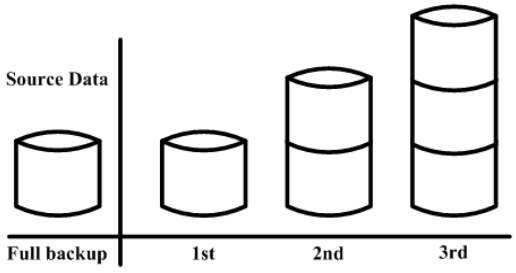


Figure 1. Full-backup method

2.2. Differential backup

Differential backup is a data backup procedure that makes a full backup initially save the data changes made since the last full backup. It serves a fast recovery time because it requires only a full backup and the last differential backup to restore the entire data repository. Thus, differential backup is faster than full backup given that the backup operation only requires the latest differential backup. However, its restore operation is slower than full backup because it requires two pieces of backup ;between the full backup and the latest differential backup, that needs to be restored, as shown in Figure 2 [12].

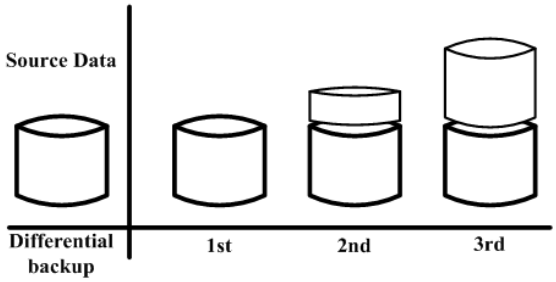


Figure 2. Differential backup method

2.3. Incremental backup

Incremental backup makes one full backup first of the data source stored in a single backup file that copy only the portions that have changed since the last backup operation. It serves to reduce the amount of time and requires less storage space since it is only backing up changed files. The recovery process requires that the most recent full backup has been completed as well as all incremental backups up to the restore point. The problem of incremental backup is if one increment data in the chain is missing or corrupted, it will be impossible to perform full recovery, as shown in Figure 3 [13].

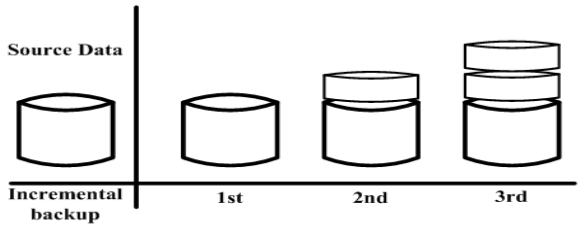


Figure 3. Incremental backup method

There are many problems with making a trade-off with the cost of performing normal backup operations and the cost of performing recovery after faulty occurrences [14]. Overall, a full backup requires a lot of free storage media and the recovery process takes a long time. Nakamura *et al.*, proposed a stochastic model base on incremental backup which describe the behaviour of a database system. The proposed method can improve appropriate rough to actual database system from the point of view of the cost of backup operations [15]. Incremental backups of files are easier to restore for entering the file recovery process. Incremental database recovery involves two steps to restore a full backup version of the backup that then read the latest version of additional files [16]. This paper aimed to propose a synchronization technique of backup model with incremental backup. The experiment operated on the scheme that the cost of backup operation could be lower than that of the other backup methods in a normal condition.

3. SYNCHRONIZE FRAMEWORK

Data synchronization is a process widely used by specialized software back up data as well as make sure multiple venues contain the same data [17]. There are two types of data transmission techniques; synchronous and asynchronous data transmisson [18].

3.1. Synchronous

Synchronous data transmission is a data trans method between sender and receiver where it takes some time before the exchange is made. usually in synchronous transmission, a communication between the sender and receiver must be established and an agreement on which party is going to be in control is established [19]. Once the session is established, the two parties also ensure there is give and take conversation occurs in actualtime. the same timing for internal clock pulses of transmitter and receiver share a common clock pulse as well as having synchronization in communication. After the connection is correctly synchronized, immediate response on data transmission may begin. The receiver counts how many bits are sent over a periodof time then reassemble them into bytes. Thus, synchronous transmission modes work well when large amounts of data must be transferred very quickly from one site to the another. Data synchronization is the entering process for synchronizing data between two or more devices and updating changes automatically between them to maintain consistency within uniformity of data systems [20].

3.2. Asynchronous

Asynchronous transmission is a type of data transmission that follows a non-synchronized and does not allow continuous data flow in communication. In addition, sender and receiver do not define the parameters of the data exchange. However, the sender inserts an extra bit of data before and after each split that indicates when each split starts and ends the transfer. Thus start and stop bits are required to do intimate the receiver packet of data about the beginning and end of the data stream [21]. Asynchronous transfers work well as timing is not an important factor as transmitter and receiver operate at different clock frequencies. Asynchronous transmission is used widely for communications over a physical medium and transfers work well when using reliable transfer media [22].

4. METHODOLOGY

The proposed data synchronization system was developed by using a send of commands from the shell script with Unison program [23]. Load balancing file synchronization management (LFSM) for distributing network traffic across multiple computing resources is introduced as shown in Figure 4.

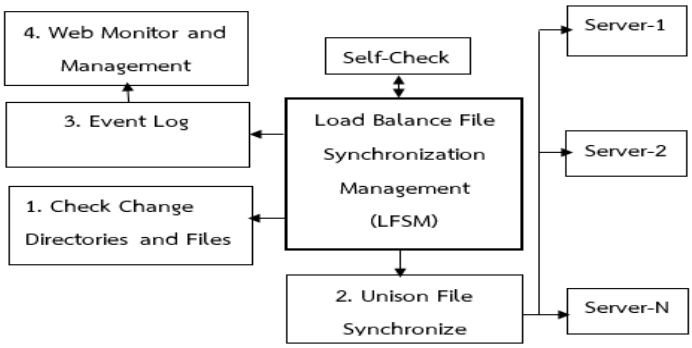


Figure 4. Backup system framework

The incremental backup system designed runs on a Linux operating system environment is Debian server 8.10 software. The storage system type of the backup is Intel Xeon CPU, 2.00 GHz, 32 GB of RAM, SAS 360 GB of Hard Disk. The implemented incremental backup and recovery functions for file data is Ext4 file system type. Synchronization of data files to each server were configured with the following authentication algorithms:

```
sync (O, A, B) =
        If A = B then (A,B);            if matching : finish
        else if A = O then (B,B);               if A = O : check B
        else if B = O then (A,A);               if B = O : check A
        else if A = MISSING then (A,B);         delete/edit
        else if B = MISSING then (A,B);         delete/edit
        else if ATOMIC in (dom(A)) U dom(B))
                and dom(A) <> dom(O)
                and dom(B) <> dom(O)
                and dom(A) <> dom(B)
                Then (A,B)
        else
for each child k, let
                (Ak,Bk) = sync(O(k), A(k), B(k)) in
                Let A' = { k -> AK } in
                Let B' = { k -> BK } in
                (A', B')
```

Unison has the command structure as a follows:
unison [Source Directory] ssh: // [Server IP Address] /[Destination Directory]

The process of calculating the size of a new file [24], Checksum, sends data that new block has added from new file to old file as shown in Figure 5.

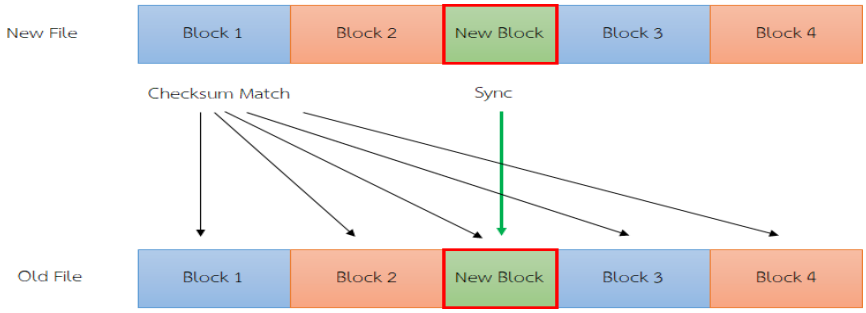


Figure 5. Checksum of synchronization data files

The synchronization system will carry out inspections with the Rolling Checksum process to check the accuracy of the information within the file [25]. The comparison of data for data checking are done by using (1).

$$a(k.l) = (\sum_{i=k}^{I} X_i) \text{mod } M \quad (1)$$

Examination is a comparison stage of the size of the source data and the destination to compare different sizse of data that check the data changes in terms of Re-calculation of the file size and sending the resulting difference to the destination *mod M* to speed up the comparison $M=2^{16}$ as (2):

$$b(k,l) \sum_{i=k}^{i} (I - i + 1)X_i) \text{mod } M \quad (2)$$

$$S\ (k,l) = a\ (k,l) + 2^{16} b(k,l) \quad (3)$$

Suppose where *s (k,I)* is the result of the validation of all data X_k X_i is data backup [26]. LFSM system status checking uses shell scripting methods to check the name status of operations as check_service.sh of name service in the proposed method. The process operates on a condition to check the LFSM system every 10 seconds. Then shell script will send the record to a status.txt file name that will decide whether the normal is 0 and abnormal is 1 of connection status as a follows.

```
service=LFSM_service.sh
while true do
   if (($(ps -ef | grep -v grep | grep $service | wc -l)>0)) then
     echo "$service is running!!!"
     echo "0" > $spath/status.txt
   else
     echo "$service is not running!!!"
     echo "1" >$spath/status.txt
   fi sleep 10 done
```

5. EXPERIMENT RESULTS

The performance of the proposed system was tested by determining the efficiency of synchronization duration. The new data backup technique of LFSM instruction was set for file quality checking and control of the workload distribution system as shown by the message log in Figure 6.

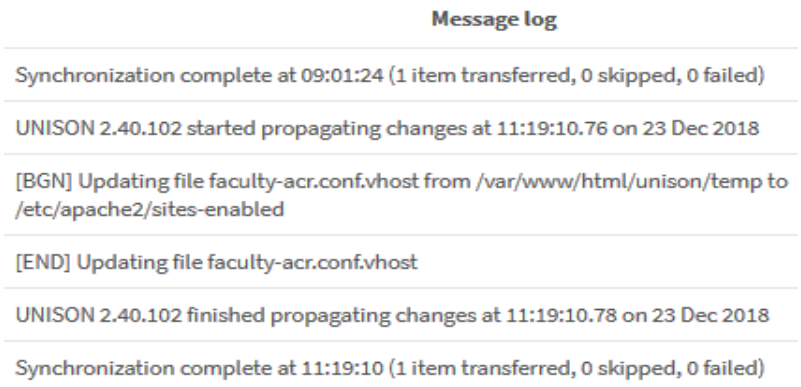


Figure 6. Synchronizes message log

The validation step was to find the changes to the files from the source machine that automatically use a command set using shell script language. The data validation method synchronized the file properties to every server in the workload distribution system by sending commands through the Unison program base on secure shell (SSH) protocol. The test compared two files format between the size of a single file and multiple data files with sizes of 1MB, 5MB, 10MB, 100MB, 200MB, 500MB, 700MB, and 1GB, respectively. The specific test data of proposed method are shown in Figure 7.

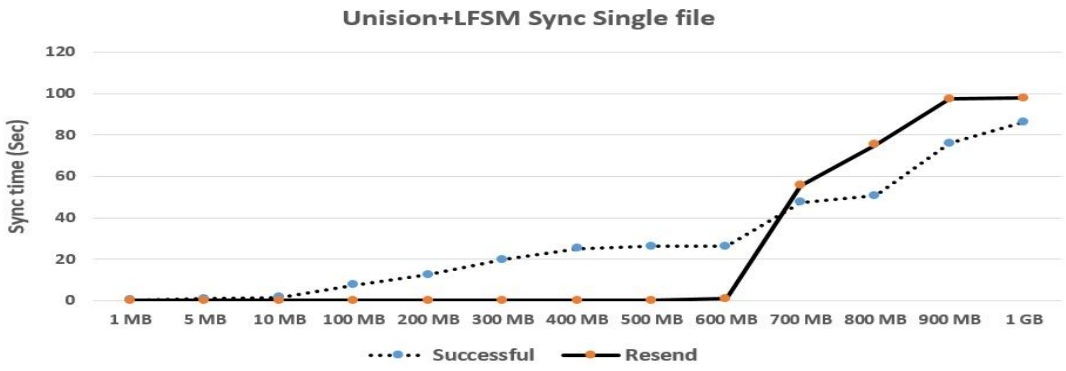
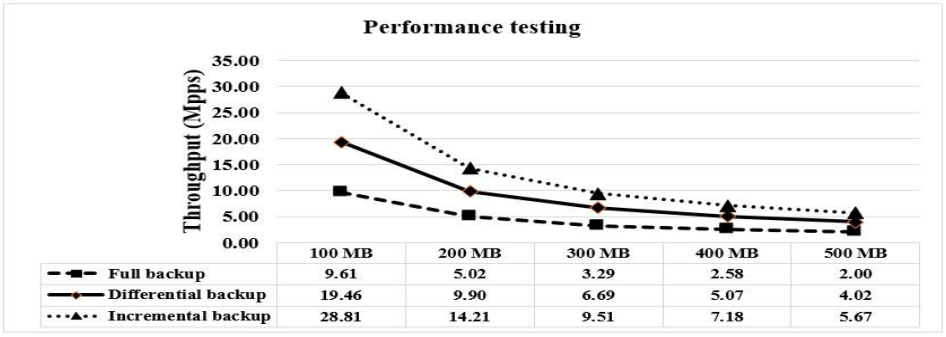


Figure 7. Single file transfer rate

The results of the single file data transfer rate of the test demonstrated the effectiveness of the most effective size was 600 MB data file. The analysis found that due to the timing of the synchronization at 60 seconds that the data transmission or data synchronization to the server was not complete according to the correct file size. The data transmission took longer than the specified time resulting in the program cancelling the original synchronization and having to send the same data file again.



	100 MB	200 MB	300 MB	400 MB	500 MB
Full backup	9.61	5.02	3.29	2.58	2.00
Differential backup	19.46	9.90	6.69	5.07	4.02
Incremental backup	28.81	14.21	9.51	7.18	5.67

Figure 8. Results of throughput

It explains a method that improve in computing terminology of throughput [27], [28] of the workload also packets per second (pps) that can be transferred from one place to another over a specified period of time. The multiple-file synchronization test was already 1MB of original data inside that number of files for each test. The specific test data of multiple-file of proposed method are shown in Figure 9.

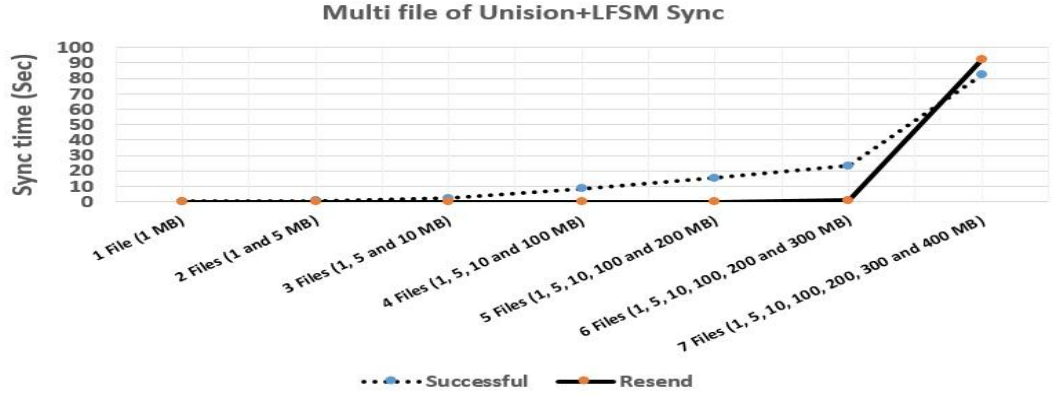


Figure 9. Multiple-file transfer rate

The result of the trial of sending multiple files showed that the number of files did not affect data transmission. The amount of data is an important factor that significantly affects data transmission too much which can affect the system and cause errors in the process. Comparison backup time consistency between the proposed method and other techniques are shown in Figure 10.

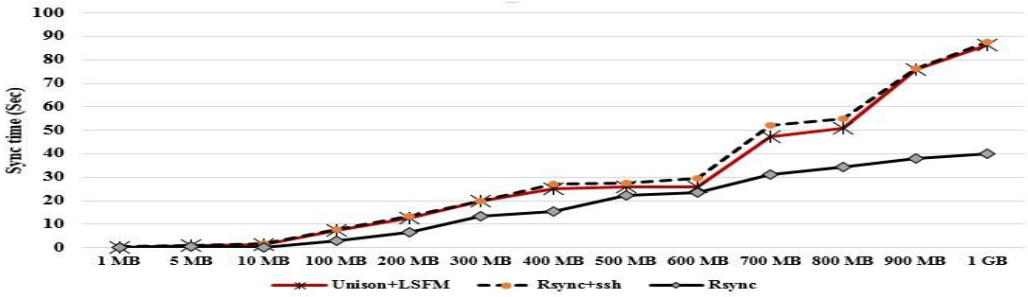


Figure 10. Comparison of performance test

Comparison of backup methods between the proposed methods and other methods. The test on database contains the initial 112 MB of files. The proposed method could improve the performance of data storage and data files backup as shown in Table 1.

Table 1. Performance of data storage

Testing	Source Data 112 MB			
	Size per day (MB)	Full backup	Differential backup	Incremental backup
Day 1	30	142	142	142
Day 2	40	324	212	182
Day 3	30	536	312	212
Day 4	38	786	450	250
Day 5	42	1,078	630	292
Day 6	36	1,406	846	328
Day 7	39	1,773	1,101	367

6. CONCLUSION

The novel proposed method of the backup system in this paper was data synchronization using a set of shell script commands in LFSM with Unison. The novel of this work is the process of checking the correctness of data file changes before synchronization to help solve errors in backup systems. File size in the experiment is a synchronization test that 1 MB to 1 GB only increase the data size of a file. As the results showed, the proposed method could improve performance of data storage and data backup system by using an average 0.238 sec to sync 1 MB file size.

ACKNOWLEDGEMENTS

The novel proposed method of the backup system in this paper was data synchronization using a set of shell script commands in LFSM with Unison. The novel of this work is the process of checking the correctness of data file changes before synchronization to help solve errors in backup systems. File size in the experiment is a synchronization test that 1 MB to 1 GB only increase the data size of a file. As the results showed, the proposed method could improve performance of data storage and data backup system by using an average 0.238 sec to sync 1 MB file size.

REFERENCES

[1] J. Wang, X. Chen, X. Huang, I. You, Y. Xiang, "Verifiable Auditing for Outsourced Database in Cloud Computing," *IEEE Transactions on Computers*, vol. 64, no. 11, pp. 3293-3303, 1 Nov. 2015, doi: 10.1109/TC.2015.2401036.

[2] D. Nallaperuma *et al*., "Online Incremental Machine Learning Platform for Big Data-Driven Smart Traffic Management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679-4690, Dec. 2019, doi: 10.1109/TITS.2019.2924883.

[3] A. Alwam, H. Ibrahim, N. Udzir, F. Sidi, "Missing Values Estimation for Skylines in Incomplete Database," *The International Arab Journal of Information Technology,* vol. 15, no. 1, pp. 66-75, 2018.
[4] A. Saxena, D. Claeys, H. Bruneel, B. Zhang, J. Walraevens, "Modeling data backups as a batch-service queue with vacations and exhaustive policy," *Journal of Computer Communications,* vol. 128, pp. 46-59, 2018, doi: https://doi.org/10.1016/j.comcom.2018.07.014.
[5] B. Hu, H. Gharavi, "Smart Grid Mesh Network Security Using Dynamic Key Distribution With Merkle Tree 4-Way Handshaking," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 550-558, March 2014, doi: 10.1109/TSG.2013.2277963.
[6] G. Wang, J. Yu, Q. Xie, "Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 294-302, Feb. 2013, doi: 10.1109/TII.2012.2215877.
[7] K. Renuga, S. S. Tan, Y. Q. Zhu, T. C. Low, Y. H. Wang, "Balanced and Efficient Data Placement and Replication Strategy for Distributed Backup Storage Systems," *International Conference on Computational Science and Engineering*, 2009, pp. 87-94, doi: 10.1109/CSE.2009.27.
[8] A. A. Nasr, N. A. El-Bahnasawy, A. El-Sayed, "A New Duplication Task Scheduling Algorithm in Heterogeneous Distributed Computing Systems," *Bulletin of Electrical Engineering and Informatics*, vol. 5, no. 3, pp. 373-382, 2016, doi: 10.11591/eei.v5i3.559.
[9] Y. Qin, B. Hoffmann, D. J. Lilja, "HyperProtect: Enhancing the Performance of a Dynamic Backup System Using Intelligent Scheduling," *IEEE 37th International Performance Computing and Communications Conference IPCCC*, 2018, pp. 1-8, doi: 10.1109/PCCC.2018.8711182.
[10] M. Misaki, T. Tsuda, S. Inoue, S. Sato, A. Kayahara, S. Imai, "Distributed database and application architecture for big data solutions," *International Symposium on Semiconductor Manufacturing ISSM*, 2016, pp. 1-4, doi: 10.1109/ISSM.2016.7934509.
[11] C. Qian, Y. Huang, X. Zhao, T. Nakagawa, "Optimal Backup Interval for a Database System with Full and Periodic Increment Backup," *Journal of Computers*, vol. 5, no. 4, pp. 557-564, 2010, doi: 10.4304/jcp.5.4.557-564.
[12] Brandon Hoffmann, "Performance Trade-offs in Dynamic Backup Scheduling," Thesis of Master of Science, Faculty of the graduate school, University of Minnesota, 2015.
[13] X. Yin, J. Alonso, F. Machida, E. C. Andrade, K. S. Trivedi, "Availability Modeling and Analysis for Data Backup and Restore Operations," *IEEE 31st Symposium on Reliable Distributed Systems*, pp. 141-150, doi: 10.1109/SRDS.2012.9.
[14] L. Ma, W. Su, X. Li, B. Wu, X. Jiang, "Heterogeneous data backup against early warning disasters in geo-distributed data center networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 4, pp. 376-385, 2018, doi: https://doi.org/10.1364/JOCN.10.000376.
[15] Y. Chao, X. Wen, W. Guohui, Q. Xinge, L. Sai, Z. Lele, "Incremental local data backup system based on bacula," *IEEE International Conference of Safety Produce Informatization IICSP)*, 2018, pp. 429-432, doi: 10.1109/IICSPI.2018.8690434.
[16] F. Gulagiz, S. Eken, A. Kavak, A. Sayar, "Idle Time Estimation for Bandwidth-Efficient Synchronization in Replicated Distributed File System," *The International Arab Journal of Information Technology*, vol. 15, no. 2, pp. 177-185, 2018.
[17] M. Faiz, U. Shanker, "Data synchronization in distributed client-server applications," *IEEE International Conference on Engineering and Technology ICETECH*, pp. 611-616, doi: 10.1109/ICETECH.2016.7569323.
[18] R. Yu, R. Proietti, S. Yin, J. Kurumida, S. J. B. Yoo, "10-Gb/s BM-CDR Circuit With Synchronous Data Output for Optical Networks," *IEEE Photonics Technology Letters*, vol. 25, no. 5, pp. 508-511, March1, 2013, doi: 10.1109/LPT.2013.2242461.
[19] T. Rahman, S. M. A. Motakabber, M. I. Ibrahimy, A. H. M. Z. Alam, "Design and implementation of a series switching SPSI for PV cell to use in carrier based grid synchronous system," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 2, pp. 349-366, 2019, doi: 10.11591/eei.v8i2.1507.
[20] H. Hu, G. Wu, X. Ding, J. Chen, "SFSW Time Transfer Over Branching Fiber-Optic Networks With Synchronous TDMA," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1802-1805, Sept. 2018, doi: 10.1109/LCOMM.2018.2828079.
[21] Z. Cai, S. Ji, J. He, L. Wei, A. G. Bourgeois, "Distributed and Asynchronous Data Collection in Cognitive Radio Networks with Fairness Consideration," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2020-2029, Aug. 2014, doi: 10.1109/TPDS.2013.75.
[22] S. Ji, Z. Cai, "Distributed Data Collection in Large-Scale Asynchronous Wireless Sensor Networks Under the Generalized Physical Interference Model," *IEEE/ACM Transactions on Networking*, vol. 21, no. 4, pp. 1270-1283, Aug. 2013, doi: 10.1109/TNET.2012.2221165.
[23] X. Lin, Z. Du, J. Yang, "The simple optimization of WLC algorithm based on LVS cluster system," *IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, 2012, pp. 279-282, doi: 10.1109/CCIS.2012.6664412.
[24] C. Xu, K. Wang, Y. Sun, S. Guo, A. Y. Zomaya, "Redundancy Avoidance for Big Data in Data Centers: A Conventional Neural Network Approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 104-114, 1 Jan.-March 2020, doi: 10.1109/TNSE.2018.2843326.
[25] R. Kiyohara, S. Mii, K. Tanaka, Y. Terashima, H. Kambe, "Study on binary code synchronization in consumer devices," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 254-260, February 2010, doi: 10.1109/TCE.2010.5439153.

[26] X. Wang, M. Veeraraghavan, H. Shen, "Evaluation study of a proposed hadoop for data center networks incorporating optical circuit switches," *Journal of Optical Communications and Networking*, vol. 10, no. 8, pp. 50-63, 2018, doi: https://doi.org/10.1364/JOCN.10.000C50.
[27] D. A. G. Ramirez, C. Hernandez, F. Martinez, "Throughput in cooperative wireless networks," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 2, pp. 698-706, 2020, doi: 10.11591/eei.v9i2.2025.
[28] W. Zhang, Z. Zhang, S. Zeadally, H. Chao, V. C. M. Leung, "Energy-efficient Workload Allocation and Computation Resource Configuration in Distributed Cloud/Edge Computing Systems With Stochastic Workloads," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1118-1132, June 2020, doi: 10.1109/JSAC.2020.2986614.

BIOGRAPHY OF AUTHOR

Prakai Nadee received the B.Eng. degree of Computer Engineering from the Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand in 1995, and the M.Eng. Computer Engineering, Kasetsart University, Bangkok, Thailand in 2003. His area of interest includes Computer Network, Operating System and Programming Language.

Preecha Somwang received the B.S. degree from the Nakhon Ratchasima College, Nakhon Ratchasima, Thailand in 2009, and the M.S. degree from the same college in 2011. His area of interest includes Computer Network and Intrusion Detection.