Int. J. Advance Soft Compu. Appl, Vol. 6, No. 3, November 2014
ISSN 2074-8523

A New Database Archiving Approach for Effective Storage and Data Management: A Case Study of Data Warehouse Project in a Korean Bank

Seung Ryul Jeong[1], Yoosin Kim[1], Imran Ghani[2], and Jae Hwa Kim[1]

[1]Graduate School of Business IT, Kookmin University, Seoul, South Korea
e-mail: srjeong@kookmin.ac.kr, trust@kookmin.ac.kr, kjh871030@naver.com
[2]Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia
e-mail: Imran@utm.my

Abstract

Today, the amount of data stored in applications and large data warehouse increases exponentially every year. Thus, organizations face numerous related problems, such as rapid increases in management load, service deterioration, increasing storage costs, shorter server upgrade cycles, and so on. To solve these problems, this paper presents the information lifecycle management (ILM)-based new database archiving approach. To examine the feasibility and compare the performance of the proposed approach, a case study method is employed that analyzes an actual data warehouse project conducted at a large Korean commercial bank.

Keywords: *Information Lifecycle Management, Archiving System, Data Warehouse separated by commas.*

1 Introduction

Today, all companies build systems that require databases. In particular, enterprise-wide data warehouse systems built for financial applications store tens of terabytes to hundreds of terabytes of data, and the volume of data increases exponentially every year beyond normal levels [1,2]. According to Informatica (2010), the amount of data stored in applications and large data warehouses, such as ERP and CRM systems, is constantly increasing—often by more than 65%. In this situation, the characteristics of and problems in information environments that most organizations face or are expected to face as a result of the rapid increase in the volume of stored data are, first, rapid increases in management load, service deterioration, increases in storage costs, and shortening of server upgrade cycles. Second, the very complicated and diversified needs of interested parties using the information and the excessive cost to generate the information ultimately lead to deterioration in an organization's competitiveness. Third, the possibility of incorrect decision-making increases as a result of deterioration in information quality. Fourth, the phenomenon of absence of information ownership by worksite personnel is become increasingly serious and, as a result, communication costs and errors are increasing during business development. Finally, efforts to increase the value of enterprise-wide information assets are weighted, thus causing the problem of increasing information management costs [3]. To solve this problem, companies make various efforts; however, in reality, they repeatedly face the same problem over time. In terms of data management, most companies continue to store all rapidly increasing data on disk simply for the economic factor of storage price. These routine procedures are simply repeated without assessing the inherent value of the data and overlook the data's utilization and quality—the most basic properties of information generation [4]. In particular, the most significant concern is whether the data warehouse is understood to be the unconditional storage area for all of the data and all data are stored in the data warehouse. If so, massive storage management costs are incurred and the volume of data increases to the extent that its management is impossible. Literally, the data warehouse plays the role of warehouse without the ability to conduct analysis, which is, in many cases, the unique feature of data warehouses [1].

Information Lifecycle Management (ILM) is defined as a new management paradigm that includes policies, processes, implementation strategies, services, and tools to manage—in the most optimized and cost-effective infrastructure—the creation of information at its disposal depending on the business value of the information [5]. ILM is an appropriate approach for enterprise-wide information management and provides the necessary measures to understand the characteristics of the new information environment discussed to this point and to solve the resulting issues [12]. Responding in the future to the explosive growth of digital data will be impossible without enterprise-wide management policies and an implementation infrastructure for information management; otherwise, the ultimate goals of enterprise information management—information competitiveness and TCO reduction—will be difficult to achieve [4, 15, 16].

The purpose of this paper is to present an ILM-based new database archiving system model for effective storage of large capacity databases, such as data warehouses, improvements to business support processes, and enhancements to management operating efficiency. Thus, this study attempts to present a new model by analyzing both actual cases and its feasibility. Chapter 2 examines ILM and database archiving and discusses current problems and solutions by analyzing the operating environment of the current system and the defined functional worksite requirements. Chapter 4 presents an ILM-based database archiving system as a system improvement plan and examines related archiving policies and detailed technical elements, and then analyzes performance through a comparative review using the model proposed in the next chapter and the existing environment.

2 Theoretical Background

2.1 Information Lifecycle Management

Changes in the information management environment resulting from a sudden increase in corporate data support the claim that investment type should be modified from unilateral investment in the traditionally maintained IT infrastructure to investment in the quality and value of the information itself [6,7]. This change intends to maintain data management cost efficiency and companies' consistent maintenance of their business by defining lifecycle management policies based on the value of the information and by differentiating the data storage and utilization method depending on the policy because corporate data can no longer be managed in the traditional manner [8]. Moreover, a need exists to introduce ILM-based enterprise-wide information management to replace existing information management types [8, 13].

ILM may be defined as a comprehensive information management mechanism composed of various policies related to the organization's information management, standard procedures, and solutions and services to support them to utilize and manage information in the most efficient and economical manner, depending on the business value of each step—from the creation of information to its final disposal [5,9]. Therefore, an IML-based information management strategy requires various tasks to implement the most cost-effective management services and data storage method depending on (1) the various policies for the maintenance of a company's information assets and functional implementation and (2) the value of information based on the company's existing data that were systematically maintained and managed for decades [3, 12, 14, 15, 16]. Of these tasks, data classification must be implemented to enable a company's mass data management, and may be executed very differently depending on the purpose of the use [12]. The most common factor related to ILM is classification of active data and inactive data. That is, classifying frequently utilized data and intermittently used data and storing frequently used active data on expensive, high-performance disks and rarely used mass data on inexpensive, capacity-oriented disks secures efficient storage utilization and economic feasibility [8, 14].

2.2 Database Archiving

Archives refer to document bureaus and recordkeeping places that possess and store official and private documents of the government, public offices, and other organizations. Archiving is about digitizing information. However, some of the information may be lost through degradations in quality or information that was scattered over time during permanent recording, conservations, or usage. Digital archives represent a type of database that transforms the collection of the data to a digital format, stores information, and maintains and manages the relationships among the data. The purpose of a digital archive is to accumulate information by systemizing the process to enable the efficient use of the information in various ways and not just to accumulate the information, which differs from regular databases [10].

Database archiving, e-mail archiving, and ERP archiving solutions, among others, have been implemented in real business situations. Database archiving is an archiving system that manages file loading by extracting and loading data of low value and access frequency from an operating system database. ERP archiving extracts infrequently used ERP data from completed business processes, transfers them to an archiving system, and supports easy utilization if necessary. E-mail archiving is a system that effectively manages the preservation and retrieval of e-mail by compressing sent and received e-mails and storing them in data storage. Of these methods, database archiving is one way to increase the flexibility of the rapidly growing volume of data within a company. IT selects data records in an operations database that are expected to not be referred to again, and transfers/stores them in an archiving system, allowing users to view the data if necessary. According to IDC, the average increase in a database’s size is 42% and occurs two times during every two years. Companies invest significant funds every year to upgrade and operate complicated databases related to their core business applications, and these databases store significant data for business operations and decision-making. As a result, a database that holds too much data degrades performance and limits the availability of the application’s functions for which it was designed for use. The method presented as a fundamental solution for these issues is the database archiving system that extracts inactive data from a database and stores it.

This database archiving is classified as active and in-active archiving depending on whether used within the same application that accesses the original database. Active archiving targets data, for which a transaction is terminated within the database, provides reports and analysis information and enables the use of existing applications for data storage policies and, in particular, compliance.

In-active archiving targets rarely used data because transactions are terminated within the database and is used when long-term storage and property maintenance of the original data are needed. In-active archiving is also used to store data for compliance and auditing, and passes data when required, for example, by the operational database and the active archival area. In-active archiving cannot use the same application but provides immediate access to the data through SQL or XML. This method, the most efficient one for mass, long-term data, has few retrieval requirements, does not require rapid response time, and provides data immediately if necessary. Moreover, using compression techniques saves significant disk space [17].

3 Analysis of the Operating System Environment

The purpose of this study is to identify problems by analyzing the operating environment of an actual large-capacity database from an information lifecycle perspective and to present database archiving-based alternatives. For this purpose, this section attempts to analyze the system operating environment of A Bank, a top financial institution in Korea. The target system is a large capacity database that supports various analyses, statistics, and reporting tasks required at the worksite. This study first identified the data status and disk structure, and analyzed the business support processes of the data warehouse.

3.1 Status of Data Warehouse Data

A Bank's data warehouse (see Table 1) contains 1,907 tables and has a total database size of 7.80 terabytes (TB). The top 10% of the tables accounts for approximately 91.1% of the total data, and the top 20% of the tables accounts for almost all of the warehouse's capacity, or 98.2%.

Table 1: A Bank data warehouse data usage

Total	**Number of tables**	**Size (TB)**	**Table size of monthly maximum data**
	1,910	7.80 TB	502 GB
Top 10%	**Number of tables**	**Size (TB)**	**Share**
	191	7.11 TB	91.1%
Top 20%	**Number of tables**	**Size (TB)**	**Share**
	382	7.66 TB	98.2%

Table 2: Disk configuration and database size in GB

Classification	**Disk unit**	**Number of disks**	**Disk size (Physical)**	**RAID1 disk size (DB)**	**Number of disks (DB)**	**Used DB size**
Clique 0	36	204	7,344	3,672	192	3,197
Clique 1	36	212	7,632	3,816	192	3,197
Clique 2	73	128	9,244	4,672	128	4,264
Clique 3	73	88	6,424	3,212	88	2,930
TOTAL		632	30,744	15,372	600	13,588

Total disk size is 30.7 TB and 15.3 TB, and the actual physically usable area from configuration of the disk as RAID-1 was assigned to the database. The size of the actual database (for example, data, index, temporary area) is 13.5 TB. As surveyed in Table 1, the size of the genuinely used data is approximately 7.81 TB, and the remaining 6.54 TB is used for the index and in other areas. The entire database space use rate is 86%.

3.2 Analysis of Data Warehouse Business Support Process

Approaches to using the A Bank data warehouse are divided into two methods. The first method is a formal inquiry of pre-defined summary tables that give general users immediate use and that provide data. The second method is unstructured work, such as performing a search based on specific requirements or creating a new summary table not previously defined. Super staff directly performs such unstructured work using SQL sentences. However, sometimes a re-analysis of the original data reveals that such data no longer exist in the data warehouse. In such a situation, a data recovery service request (SR) is made to the data warehouse operators and system operators, who provide such immediate services. However, rapid response time is difficult because of the time required and the procedures that need to be followed to process such SRs. Time to respond to services increases rapidly if the data do not exist in the data warehouse, creating a structural limitation. However, blindly increasing storage space to solve this problem increases both costs and management.

Type A) Processing procedure in the presence of data in the data warehouse

STEP-1: A general user requests services to a super user.

STEP-2: The super user derives the data result through an unstructured search of the data warehouse.

STEP-3: The results are transferred to the general user.

Type B) Processing procedure in the absence of data in the data warehouse

STEP-1: A general user requests services to a super user.

STEP-2: The super user requests the data to a data warehouse operator.

STEP-3: The data warehouse operator sends the data to be recovered to a system operator and asks the operator to recover it in a general disk area.

STEP-4: The system operator finds the media containing the data, conducts the recovery using backup equipment, and copies the data to a general disk area.

STEP-5: The system operator notifies the data warehouse operator of the completion of the recovery.

STEP-6: The data warehouse operator loads the data into the data warehouse.

STEP-7: When loading completes, the data warehouse operator notifies the super user of the results.

STEP-8: The super user processes the requested operation and sends the results to the general user.

When comparing the processing steps for Types A and B, if no data are in the data warehouse, additional work steps are required. Additionally, the primary work cannot be completed given the time limitations resulting from the availability of the backup equipment, the data warehouse operating hours (8:00 a.m. to midnight), and the daily batch working hours (midnight to 8:00 a.m.); thus, such work is accomplished on the weekends and only urgent recovery work is performed in the middle of batch time. Data recovered from the monthly backup media occurred on average 15 times. The size of the data recovered at this time is approximately 3 TB and the time needed from recovery service request to service response time is an average of three to five days for more than 100 GB and on average shorter than two days for less than 100 GB (see Table 3).

Table 3: Monthly average number of recoveries and processing time

Unit	**Number of recovery requests**	**100 GB or more**	**100 GB or less**
One month	Average 15 times	3-5 days	Less than 2 days

Backup is divided into full backup and backup by table. Full backup is carried out once a week, and takes an average of 16 hours. Two types of backup exist: 1) tables, the ledger table, and daily closing tables are primarily backed up every day, and 2) tables created during monthly finishing are backed up once a month on a regular basis. In particular, important trading specification tables manage daily-accumulated data as monthly tables, indicating that the amount of statistical tables created during the monthly closing process is not significant.

An examination of A Bank shows that, first, excessive upgrade costs may be incurred to maintain performance given the increase in the volume of data. Second, the data access constraint is serious given the increase in the volume of data. Third, lack of storage results in additional work from regularly repeating restores of the backup data, backing up some data to free up disk space, and recovering the data again after such work is completed. Fourth, if not having the layout of that time, additional time and money are needed to recover backed up data. Moreover, when switching to the current operating table, data conversion costs are added separately. Fifth, when storing historical data by table on the backup tape, previously backed up data become redundant during the monthly backup. That is, the problems of longer backup time and greater consumption of backup media are intensified because redundant backup data also increase.

3.3 Current System Challenges

The challenges derived from the previously discussed data status and business process analysis are summarized in the following five points. First, database storage space represents 10% of total costs; therefore, either hardware should be expanded quickly or data should be backed up or deleted. Additionally, of data currently stored in online, the storage cycle of large tables is only an average of three months, indicating that more data should be stored and accessibility should be immediately improved. Second, if no data exist online, business processing becomes complicated, requiring eight total steps and a processing period from two to five days; therefore, more efficient business processes are required. Third, frequent recovery work may deteriorate database availability. That is, during the actual recovery work, input to the table / edit / delete and normal lookup service is impossible, a situation that must be improved. Fourth, hardware expansion should be suppressed. Because significant hardware expansion costs and system maintenance costs are incurred every year, expansion costs should be suppressed as much as possible by considering utilization and value of data and by implementing a new storage policy and new storage system. At this time, expansion resulting from natural increases in the volume of data is excluded. Fifth, a data compression function is required because a large amount of data should be stored and access to such data should be convenient even in the compression state.

4 System Improvements

An analysis of the operating environment of A Bank's data warehouse revealed that securing data storage space, reducing frequent recovery operations, and shortening SR processing time are required to increase data availability. Thus, this study attempted to solve these problems by implementing a database archiving system based on the information lifecycle.

4.1 Establishment of Database Archiving Policy

The data warehouse of A Bank managed data by classifying them into three- to 12-month permanent storage periods depending on the nature and amount of the data and frequency of use. Most of the data stored for three months were frequently deleted after backup because of poor access frequency of the data itself; sometimes, the data were deleted temporarily because of inadequate data warehouse size. Thus, archiving target data was selected as a focus of this study because the activity generates the most data and numerous recovery requests are made each month. This study measured the size of the entire data warehouse based on the tables needed to develop a database archiving system (see Table 4).

Table 4: Archiving target data

Classification	**Number of tables**	**Size (GB)**	**Share table**	**Share database**
Total tables	1,910	7,809	-	-
Archiving target tables	124	4,874	6%	62.24%

Approximately 500 GB of initial data is to be implemented based on the online database size (1.1 TB in a flat file), and approximately 9 TB is needed to back up and store the data to a compression archival file based on FALT FILE. The data storage period in the data warehouse was established as three-month, 12-month, and permanent storage, similar to the existing storage policy. Data archived for one month are stored in a newly introduced archive database, data archived for 12 months are stored in a compression archive area, and data archived for more than 12 months are permanently stored on backup media.

4.2 Compression Archiving System

Compression archiving compresses and stores data in the form of an archive file. That is, the compression method reduces the size of the data to store larger amounts in the same storage size using an in-active archiving technique. Compression archiving is primarily used for the long-term storage of data or when storing data for compliance purposes. Therefore, unlike regular compression, the data should be readable immediately and the original property value of data should be maintained. Properly storing the original data is one of the elements required to recover the data intact. Most commercial archiving software packages support the creation of XML or compression archive files. Additionally, compression archiving must be able to extract only certain necessary areas when maintaining large amounts of data. A search for transactions from a particular account number from 10 years of data with two years of data in the operational database and two to four years of data stored in the active archiving database requires the remaining six years of data to be recovered. However, the compression archiving system can efficiently recover only specific data, such as data based on account number, from the compression archive file and provide them to the database. Shorter backup time is accomplished because the compressed in-active archive file is backed up in its compression state.

To summarize, the following results are obtained from a compression archiving system. First, a compression archiving system plays a data storage role in storing a large volume of data for a long period. Second, because the properties of the original data (column information, length, number of cases, time of creation) are stored, data is recoverable to the time when the archival file was created. Third, the compression archiving system plays the role of a source of data supply, providing desired past data to the operational database and the active archive database. Fourth, the system is used to meet compliance, self-counterfeiting, and tamper-proofing requirements, and can be interlocked with WORM (write once, read many) storage. Fifth, the system may be used as a backup of the original data against a failure of the operational database and the archiving database. The compression archive file may be used more efficiently when searching for a specific record from a mass volume of data because the data may also be searched directly using ANSI SQL.

4.3 Virtual Database

The basic concept of a virtual database system is not to have an actual DBMS engine and to save schema information stored in each different actual database in one local database file to show it to users as if a real database [11]. If conducting modification or re-analysis work by recovering all or part of the data needed from compression archive files, the space used for recovery should be secured in an operational database and an archiving database and be deleted again after use. A virtual database plays a role in these temporary operations without loading the operational database and the archive database; thus, if not attempting to view integrated data, the virtual database does not affect the operational database and the archive database is useful.

In a virtual database, a program creator creates an application without needing to know the location of the data in the database. If a compression archive file needs to be modified, using a virtual database in the archiving system means modifying and reanalyzing this file by loading it into the database; therefore, if the instance is shut down after the work is completed, the existing original database is not changed. The virtual database operates by running a server engine (instance) in one database. If a user accesses this instance to add/modify/delete data, the contents are created in temporary files. The changed content is viewed when this instance is running but if this instance is shut down, temporary files are deleted and do not affect the original database.

4.4 Automation System

Performing archival functions through night batch work is common. Moreover, an individual cannot monitor the extracted data or create and perform archiving commands. A person would also have difficulty browsing the archive file to create and perform de-archiving commands. If an increasing number of personnel and more time are provided for such work, administrative costs would increase.

Therefore, implementing an automated processing system through certain rules and schedules is necessary to operate regular archiving and de-archiving work. Management efforts should be minimized by implementing a GUI to control such an automation system. The automation system should include functions such as archiving, database building, creating and recovering compression archive files, and retrieving and removing archived files whose storage period has ended. The automation system plays an important role in optimizing the utilization of storage resources.

5 Proposed System Configuration and Test Results

5.1 Proposed System Configuration

This proposed system stores data by classifying them into four storage steps in terms of the information lifecycle based on the data's lifecycle and storage policy. First, data extracted from the data warehouse goes through a compression process in the compression archive area and an archive file is created. A one-month portion of the archive file stored in this manner is reloaded into the archive database. At this point, the existing one-month portion that used to be in the archive database is deleted automatically. Because the deleted data already exists as a compression archive file, this file need not be recreated. Of the data stored in the compression archive area, data for more than 12 months is separated from the compression archive area and stored permanently in backup equipment. Extracting and creating compressed archive files, creating archive databases, and other tasks are performed automatically through the administrator screen. Figure 1 shows the configuration of the proposed system.

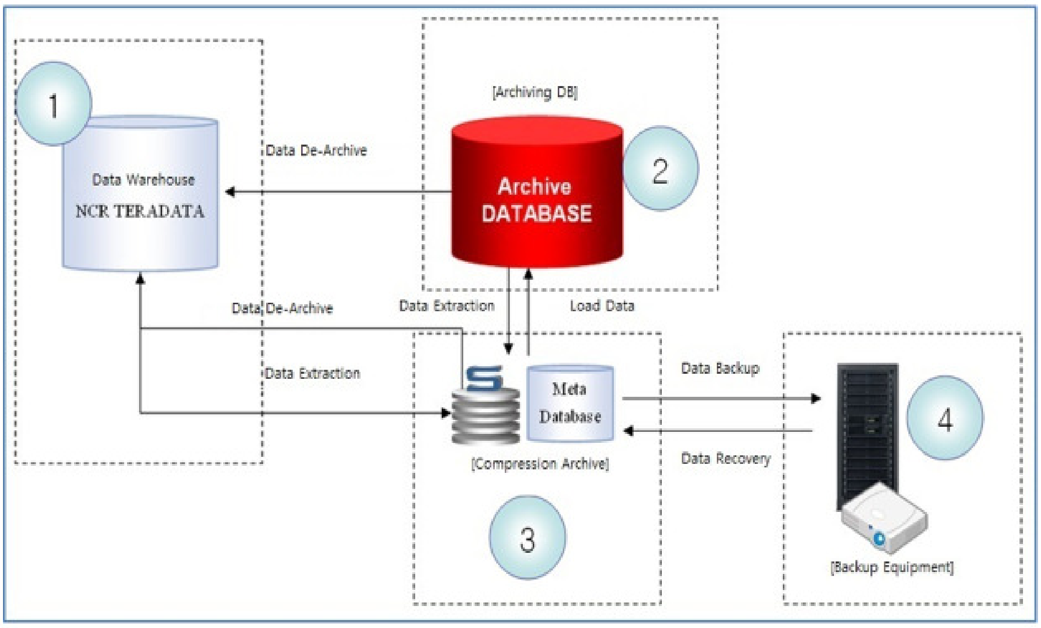

Fig.1 Configuration Diagram of Proposed System

5.2 Test Results

To evaluate the effectiveness of an ILM-based database archiving system, storage savings before and after system building and improvements in existing SR processing steps were analyzed. Moreover, processing time was compared using the existing recovery time and archiving process, and the scenarios in which the archive system was developed by expanding the existing data warehouse were compared and analyzed. Additionally, by considering the recovery of mass data for re-analysis, recovery time and time to build a virtual database were compared and assessed. Finally, the expected effects were summarized based on the results from monitoring the system for one week after its implementation.

5.2.1 Reduction effects of data warehouse storage use

By initially implementing SR-related data for one month of archived target information in the data warehouse, the proposed system secured approximately 500 GB of storage space. Moreover, by recovering a table with the highest recovery request frequency of all SR-related tables from the backup tape, approximately 9.81 TB based on the original was stored in the compression archiving area. The average compression rate of the compression files is 93.3%, based on the recovered flat file; approximately 770 GB of storage was used and approximately 9.04 TB of storage was saved, compared with the original files. Storing these data in the data warehouse would save approximately 5 TB of storage. This value was used because the NCR's Teradata Database has a compression rate of approximately 50% through its self-compression function. A savings of 5 TB affects the pure data size except for the index and other areas.

5.2.2 Rapid response speed improvement

In the absence of data in the data warehouse, the complexity of the SR processing steps and processing time problems were improved as follows.

1) When data are necessary before one month: Processing service within a few seconds to one hour by searching data from the archive database (excluding the mass re-analysis requirement) reduced processing complexity from eight steps to one step and improved service response time.

2) When retrieving transactional information for one year:

2-1) When retrieving a simple statement: Retrieving data from the compression archive system and sending the results

Processing time: Processing within a few seconds to one hour (retrieval and sending mass data is excluded).

2-2) When retrieving a complex statement: By retrieving data using the compression archive system and loading the results into a data warehouse or archive database, processing is done by connecting to related tables.

Processing time: Processing within a few seconds to two hours (except for question time after recovery).

In summary, the actual processing steps were reduced from at least one step to a maximum of three steps, simplifying the existing process that goes through eight steps; the SR processing time improved by 36 times to up to 4,319 times compared with the existing process (see Table 5).

Table 5: Comparison of before and after service request processing time building

Classification	100 GB or more	100 GB or less
Before building	3-5 days	Less than 2 days
After building	1 minute-2 hours	10 seconds-1 hour

5.2.3 Existing recovery process response and improvement effects using a virtual database

For a large amount of data that needs to be recovered to meet SR requirements, the recovery time for an existing data warehouse and the time to configure a virtual database were compared. Existing recovery requests were made 15 times per approximately one month and the size of the data was approximately 3 TB. Such recovery is done by table rather than recovery of the entire database. The improvement effects were compared by actually preparing for these types of situations and measuring the time to build the virtual database (see Table 6). For data warehouse recovery time, recovering 100 GB using the existing recovery process was set to one day and those figures were compared based on approximately three hours to recover 100 GB from a virtual database—a period during which work occurred sequentially. That is, this time to sequentially process and recover the data to disk from the backup media and then reload it into the database and a virtual database is also the time needed to sequentially process the targeted archive files. The meaning of recovery time in a data warehouse is the generalized time needed to receive data from the backup equipment and load them into the database. The virtual database building time is the total time needed to build a virtual database from a compression archive file.

Table 6: Comparison of recovery time before and after building a virtual database

Classification	Based on 100 GB	Number of cases	When building 3 TB
Before building	Approx. one day	15	Approx. 30 days
After building	Approx. three hours	15	Approx. 45 hours

5.2.4 Reduction in the number of recoveries

The number of recoveries, which used to be 15 cases per month in the existing system, was significantly reduced. The system was monitored for approximately one week after its initiation and showed that the number of accesses of the compression archive system and retrievals was from six to eight cases a day. Query requirements were monitored through two major aspects.

-Data Extraction: two to three cases per day of receiving a query result in a file and reloading it into the data warehouse

-Data Query: four to five cases per day of no further action at the end of the actual data query

During this period, no requests were made to recover data from the backup media. However, we cannot conclude that recovery requests were reduced by 100% because the storage cycle determined was based on data utilization for the past year and was investigated to account for most SR work; however, recovery requests were certainly reduced by a significant amount. Given that a lot of other work occurred after recovering the existing data, the access frequency of the compression archive system may be further increased.

5.2.5 Other improvement effects

In the existing system, the administrator is significantly burdened by the operations because he or she must continue to monitor and give work orders during the process of finding data and determining on which media they are stored when recovering data, loading data from the media onto disk, and reloading them onto the database. Compared with this burdensome process, the proposed database archiving system enables searching the data through built metadata and registering work, making the administrator’s job convenient. Additionally, the automation program automates the process of creating compression archive files based on a schedule, thus reducing the operator’s actual administration time and costs.

6 Conclusion

Annual growth in application data and more than 80% of these data being inactive have resulted in performance degradation of application programs, deterioration of availability, and increases in the maintenance costs of IT Infrastructure. Two methods may be selected in such a situation. One method is to maintain performance by continuously investing in existing infrastructure. The other method is to develop a new process to separate out the inactive data. By presenting a new approach to the build process and the infrastructure that effectively maintains and manages these inactive data through an ILM policy, this paper attempts to improve the large capacity data management problem that many companies face.

The improvement method proposed in this study is an ILM-based database archiving system that determines the storage cycle of the data by considering the data access characteristics of the data warehouse. To check the validity and effectiveness of the proposed model, this study carried out a comparative analysis with the existing database archiving technique in terms of cost and performance. The improvement effects of the proposed system compared with the existing system are as follows. First, the capacity for 500 GB of data in the existing data warehouse was secured by initially conducting SR-related data for one month among the archiving target data in terms of storage. Approximately 90% of storage-saving effects were obtained by storing data of the same period using the compression archiving technique.

This scenario is expected to result in approximately 5 TB of storage savings when storing the same data in the data warehouse. Additionally, with respect to backup time, backup in preparation for a failure of the data warehouse is bound to maintain the existing system; however, in the case of backup by table, backup time seems to be significantly reduced because of the 90% compression effect experienced when receiving compressed archive files.

Second, in SR processing, significant time saving in service and process simplification were achieved. The actual processing steps were reduced from at least one step to a maximum of three steps, simplifying the existing process of eight steps. Such simplification of the processing steps results in an improvement in processing time of at least 36 times, or up to 4,319 times relative to the existing process.

Third, if a large amount of data needs to be recovered to meet SR requirements, assuming 15 recoveries and approximately 3 TB of data each month, the time needed by the existing recovery method and to configure the virtual database were improved by 45 hours to 30 days. Additionally, given the number of recoveries based on the results of monitoring the improved system for one week, recoveries—previously 15 cases per month in the existing system—were eliminated, and the number of retrievals through access of the compression archive system turned out to be six to eight cases a day.

Finally, given that an administrator’s continuous monitoring and work activities are required in the existing system, analysis of the degree of an administrator’s operational burden showed that the actual administrative burden and operator costs were reduced significantly by utilizing built metadata data or an automated program in the proposed system.

The limitations of this study include the following. Improvement effects could not be measured because the backup system currently in operation could not be changed immediately to the backup method after building the archiving system, and additional long-term results could not be obtained given the short monitoring period.

Additionally, the ILM-applied data warehouse archive system was implemented but ILM-based data warehouse archiving that enables true data integration management seems to be built only when achieving integrated queries through the connection between this model's databases. Nevertheless, the results of this study show that an ILM-based database archiving system effectively responds to the physical limitations of large-capacity database systems of existing organizations.

References

[1] H.Y. Kim and C.K. Youn. 2012. A Case Study for the Application of Storage Tiering based on ILM through Data Value Analysis, *The Journal of Digital Policy & Management*, Vol.10, No.8, 159-172

[2] J.H. Park, S.H. Choi, and B.K. Kim. 2012. A Study on Utilization of Korea Science Citation Database(KSCD) Based on Data Mining Techniques, *Journal of information management*, Vol.43, No.4, 191-210

[3] S.R. Jeong. 2009. A Methodology for Implementing the Information Lifecycle Management System, *Korea Information Technology Research*, Vol.15

[4] S.R. Jeong. 2007. ILM based Information Management, *Korea Information Technology Research*, Vol.13

[5] Data Management Forum. 2004. ILM Definition and Scope, at http://www.snia.org/dmf

[6] C. Boone and D. Wheeldon. 2005. Information Lifecycle Management, *Service Talk*, No.75, 7-9

[7] J.H. Im, C.G. Lee, and Y.J. Lee. 2005. The Information value-based document management technique using the Information Lifecycle Management Theory, *Journal of the Korea Society for Simulation*, Vol.1, No.4, 19-30

[8] S.R. Jeong and D.W. Shin. 2009. Improving the Utilization and Efficiency of Enterprise Architecture through the Implementation of information Lifecycle Management-based EA System, *Korea Institute of information technology Architecture*, Vol.6, No.2, 107-121

[9] Sun microsystems. 2005. Information Lifecycle Management

[10] Naver Encyclopedia, digital archiving at http://terms.naver.com/entry.nhn?docId=440945&cid=441&categoryId=441

[11] Naver Encyclopedia, virtual database at http://terms.naver.com/entry.nhn?docId=1598776&cid=2955&categoryId=2955

[12] S. Al-Fedaghi. 2008. On Information Lifecycle Management, *Proceedings of Asia-Pacific Services Computing Conference, 2008. APSCC '08. IEEE*, pp. 335-342.

[13] H. Liu, X. Wang, and Q. Quan. 2009. Research on the Enterprise' Model of Information Lifecycle Management Based on Enterprise Architecture, Proceedings of 9th International Conference on Hybrid Intelligent Systems, pp. 165-169.

[14] M.F. Wang, W.T. Lin, C.H. Tang, and M.F. Tsai. 2010. Constructing Storage Capacity Migration Policies for Information Lifecycle Management System, Proceedings of 12th IEEE International Conference on High Performance Computing and Communications (HPCC), pp.503-508.

[15] T. Tanaka, R. Ueda, T. Aizono, K. Ushijima, I. Naitoh, and N. Komoda. 2005. Proposal and Evaluation of Policy Description for Information Lifecycle Management, Proceedings of International Conference on Intelligent Agents, Web Technologies and Internet Commerce, pp. 261-267.

[16] M. Beigi, I.M. Devarakonda, R. Jain, M. Kaplan, D. Pease, J. Rubas, U. Sharma, and A. Verma. 2005. Policy-based information lifecycle management in a large-scale file system, Proceedings of 6th IEEE International Workshop on Policies for Distributed Systems and Networks, pp.139-148.

[17] S. Li-zhen. 2010. Research on hierarchical storage of digital library based on the information lifecycle management, Proceedings of 2nd IEEE International Conference on Information Management and Engineering (ICIME), pp. 64-66.