Database Migration from Separated to an Integrated Database

ChulSu Lim*, Seungwu Han*, Wongoo Lee*, Kang-Ryul Shon*
*Korean Institute of Science and Technology Information
E-mail : {cslim, swhan, wglee, krshon}@kisti.re.kr

1. Introduction

A modular design of a database has many advantages in system architecture configuration especially in initial stages of a system. This approach reduces complexity of the system which could be a crucial decision factor. The complexity affects access control, user interface of the application.

But as the system grows in years, the focus of the design issue should be modified to reflect current situations. When the number of involved objects and organizations grow, the components of the database are also increased. This induces new problems in respect to handling cost and efficiency of the management of the databases. We consider above mentioned cases with a complicated database configuration in NTIS.

NTIS (National Science and Technology Information Service) provides overall Korean national R&D information [1, 2]. This service provides more than 107 million information items related to national program information gathered from 17 ministries and offices. Each organization manages the information to support their business needs[3].

2. Individual databases versus integrated databases

In designing a system with many components, databases could be configured as sets of individual databases. This modular design approach has many generic advantages including simplicity, scalability, and reusability. On the other hand, integrated approach also has its advantages. In case of NTIS, as system grows, needs for integrated databases prevailed. Figure 1 shows many services which could be efficient with integrated database support. In these services, the listing require full scan of many databases with similar schemas. The pros and cons for the databases configurations are listed in table 1. The environment of the system tends to select preferable configurations.

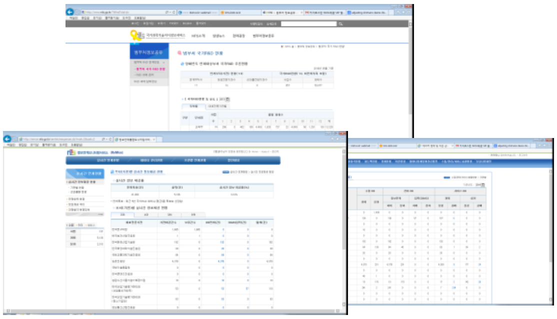

Figure 1. Needs from applications

[Table 1] Pros and cons of database configuration

	pros	cons
Individual DB	data ownership access control customization to specific needs	schema maintenance cost need union operations searching globally
Integrated DB	easy to maintain schema full scan efficiency efficient use of indexes application program development	need key integrity check precise access control if needed big index/table hard to customize to specific needs

3. Changes in accessing database tables

Figure 2 shows configurations before the integration and after the integration. In this case as much as 17 databases could be integrated as 1 database. The managing cost of database schema could be shrinked to 17 times smaller.

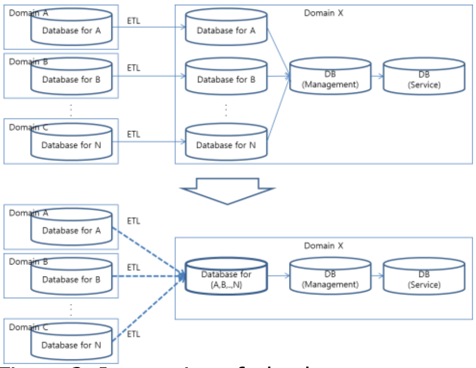

Figure 2. Integration of databases

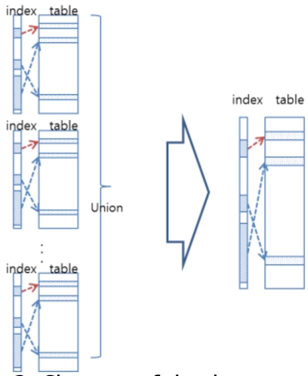

Figure 3. Changes of database record access

Figure 3 shows the changes of database table accesses for the full listing for each organization. Instead of individual index lookups and union operations to sum up, a big index and table performs the required task more efficiently.

4. Steps to integrate databases

Table 2 summarizes steps for the database integration process with plan-do-see stages. Before conducting any tasks, many aspects of the service should be checked. The influence on the applications could be checked with the help of concurrent versions system(CVS) and proper utilities like SQL management system. Figure depicts a performance check result in plan stage(check performance). For the experiment, we use 3 interconnected DBs and 3 test sets with varying number of records(A: separated configuration, B: integrated configuration). This experiment confirmed that integrated model does not suffer from big index and tables. We could conjecture that, the performance could even be better with the help of fine-tuned indexes.

[Table 2] Steps for the database integration process

stage	Actions	Details
Plan	Check influence on applications	- web service - function/procedures, job, sql
	Check DB/utility capacity	- concurrency - security utility compatibility - ETL compatibility
	Check Key consistency	- select primary key(uniqueness)
	Check security requirement	- access control
	Check performance	- tuple size, index, query optimization
	Check DB schema	- schema compatibility
Do	Modify DB, Schema, Application	- Schema modify - Data migration - Application modify
See	Check Service & Monitor	- Function/Service check - Performance check

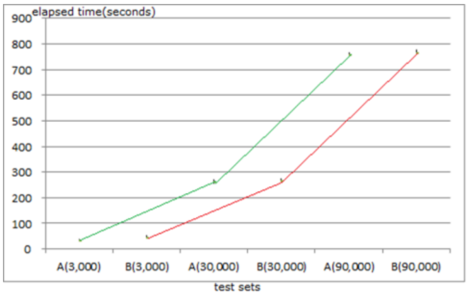

Figure 4. A test for the performance check

5. Conclusions and Future research

We examine evaluation factors to compare benefits of different database design configuration. Then we design a migration steps involved in the change of database configurations.

This research was supported by the Sharing and Diffusion of National R&D Outcome funded by the Korea Institute of Science and Technology Information.

6. References

[1] Byung-Min Lee. 2006. “Basic Strategies and Projects of the NTIS”. Journal of Korea Technology Innovation Society, pp. 245-262.
[2] ChulSu Lim, et al., “Development of NTIS Data Infrastructure for Convergence Service”, Korea convergence society, Summer conference, Jeju Korea
[3] ChulSu Lim, et al.,“Data Content Constraint Management for National R&D Data Quality Improvement”, 2011, ICCC