

튜닝을 통한 User Response Time 감소 기법 - SAP R/3 ERP System 을 중심으로

이정재⁰, 송관호*, 최진영*
고려대학교 디지털정보공학과, *고려대학교 컴퓨터학과
e-mail : mcseguy@korea.ac.kr, *{ghsong, choi}@formal.korea.ac.kr

A Decreasing Methodology of User Response Time - Mainly Focused On SAP R/3 System

Jung-Jae Lee, Gwann-ho Song, Jin-Young Choi
Dept of Digital Information Technology, Korea University
*Dept of Computer Science, Korea University

요 약

갈수록 복잡화, 대량화되고 있는 시스템의 성능은 유지하면서 User Response Time 을 단축시키는 Tuning 은 투자 대비 이익률(ROI : Return On Investment)에 탁월한 효과를 가져온다는 점에서 중요시 되고 있다.

본 연구는 SAP R/3 System 환경에서 Tuning 통한 User Response Time 감소를 위하여 Query 분석 및 Table Index 생성, Program Logic 수정, Parallel Processing, SAP R/3 System 의 고유 기능인 Notes 적용을 통한 User Response Time 감소 기법을 제시한다.

1. 서론

ERP(Enterprise Resource Planning)는 생산, 판매, 인사, 회계, 자금 등 전 부문에 걸쳐 경영자원을 통합 시스템으로 재구축함으로써 생산성을 극대화 하려는 대표적인 기업 리엔지니어링 기법이다[1].

ERP(Enterprise Resource Planning)System 의 사용률은 해외의 경우 2000 년 포춘(Fortune)지가 선정한 세계 500 대 기업의 80% 이상의 기업이 ERP 를 도입하여 사용중인 것으로 발표[1] 하였으며, 국내의 경우 SAP KOREA 의 통계[2]에 따르면, 2005 년 12 월 12 일 기준 100 대 기업 중 67 개의 기업이 도입한 것으로 발표 하였다.

본 연구에서는 세계 3 대 독립 소프트웨어 공급업체 SAP 社의[3] ERP Package SAP R/3 System 을 중심으로 Query 분석을 통한 Table Index 생성, Program Logic 수정, Parallel Processing, SAP R/3 System 의 고유 기능인 Notes 적용 등의 기법을 통한 User Response

Time 을 감소시키고, DB 서버 성능의 최적화를 통하여 ERP System 의 소유 총비용(TCO : Total Cost of Ownership) 삭감 방향을 제공하고자 한다.

본 연구의 구성은 다음과 같다. 제 2 장에서는 SAP R/3 System 의 이론적 배경을 소개한다. 제 3 장에서는 User Response Time 감소를 위한 Tuning 관점을 소개한다. 제 4 장에서는 테스트 환경을 통한 실증연구 및 결과를 분석한다. 마지막으로 제 5 장에서는 결과와 연구의 시사점 및 향후 연구 과제를 기술한다.

2. 이론적 배경

2.1 SAP R/3 System 의 개요

SAP R/3 는 독일 Walldorf 에 본사를 두고 있는 SAP AG 社에서 개발 및 판매하고 있는 ERP Package 의 하나로써, 1998 년 전세계 ERP Package 시장의 42%를 점유하고 있다[2]. SAP ERP 는 모든 Business Area 통합

할 수 있는 4 개의 모듈 구성체계와 12 개의 기본 모듈로 구성되어 있으며, <표 1>에서는 기본 4 개의 모듈 구성도만 기술하기로 한다.

<표 1> SAP 모듈 구성도[2]

모듈 구성체계			
Accounting	Logistics	Human Resources	Cross-Applications

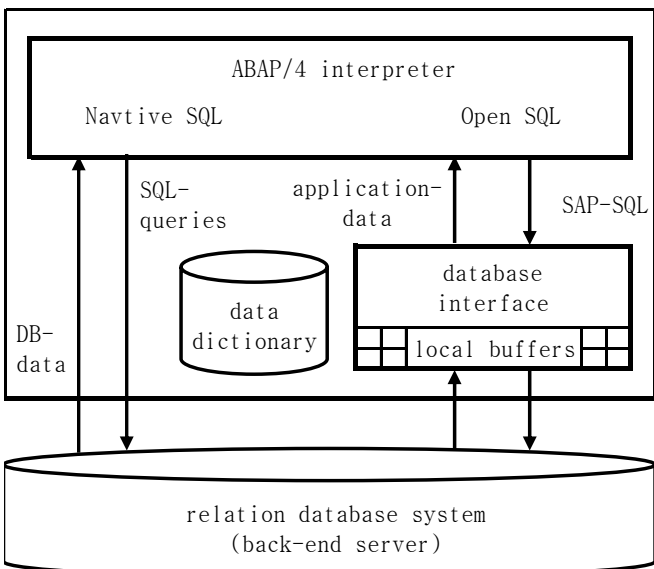
2.2 SAP/3 System 의 특징

SAP R/3 System 은, 사용자의 변경을 허용하지 않는 Standard 부분과, SAP 社의 독자적인 4GL 언어인 ABAP/4 를 이용한 개발 및 에드온(Add-on)되어진 CBO (Customer Built On) 부분으로 구분된다.

또한 각각의 Transaction 은 Business Flow 에 따라 상호 연계되어 사용되며, Standard Table 및 Program 과의 밀접한 연관을 가지고 있다.

2.3 SAP R/3 시스템의 Database Interface

SAP R/3 System 의 Database Interface 방식은 (그림 1)에서 보임과 같이, Native SQL 과 Open SQL 이라는 다른 Interface 방법으로 Database 로 접근한다[4]. Native SQL 은 EXEC SQL 이라 불리어 지며, Database Server 로의 접근에 SAP-Internal data dictionary 를 거치지 않으므로, Open SQL 과 비교하여 보다 빠른 Data Query 가 가능하나, Overhead 의 증가로 SAP R/3 에서는 권유하지 않는다[5].



(그림 1) ABAP/4 의 Database Interface[5]

2.4 SAP R/3 System 의 Tuning 개념 및 절차

튜닝이란 OS, CPU, Memory, Disk, I/O Controller, BUS, Disk Controller, Network 등을 이해하고 점검하여[6], Transaction 실행 시간을 향상시키는 일련의

작업을 말한다.

갈수록 복잡화, 대량화되고 있는 시스템을 그대로 유지하면서 User Response Time 의 실행시간을 단축시키는 Tuning 은 투자 대비 이익률(ROI : Return On Investment)에 탁월한 효과를 가져온다는 점에서 중요시 되고 있다. 또한 튜닝을 보다 체계적으로 실시하기 위해서는 적절한 Tuning Scenario 가 필요 하며 [7] 본 논문에서는 다음과 같은 Scenario 를 바탕으로 튜닝을 실시 하였다.

<표 2> Tuning 절차[7]

비즈니스 튜닝	비즈니스-튜닝
디자인 튜닝	데이터-디자인 튜닝
어플리케이션 튜닝	응용 프로그램 설계의 튜닝
	논리적 구조의 튜닝
	DB 운영상의 튜닝
	실행 결로의 튜닝
서버 튜닝	메모리 구조의 튜닝
	I/O 와 물리적 구조 튜닝
시스템 튜닝	자원에 대한 결합 튜닝
	OS 의 기본구조 튜닝

3. User Response Time 감소를 위한 Tuning 관점

SAP R/3 System 은 26 만개의 Table 과 120 만개의 Program 을 기본으로 제공하며, 생성되는 data 양의 방대함은, OLTP(On-Line Transaction Processing) Database 에 저장되는 정보의 크기와는 비교할 수 없는 정도로 대용량이므로[8], 성능의 향상을 위해 특별한 준비가 있어야 한다 [8,9,10,11,12,13].

3.1 Table Index 생성

Index 는 빠른 검색을 위해 사용되는 독립된 객체이며, Index 를 모든 칼럼에 계획 없이 작성하면 오히려 성능을 저하시킬 수 있으므로[7], 본 연구에서는 특정 Table 을 사용하는 모든 프로그램의 Query 문을 분석, Matrix 를 작성 하여, Program 에서 공통적으로 사용 가능한 Index 를 우선적으로 생성하였다. 본 논문에서는 Query Matrix 작성요령은 기술하지 않기로 한다.

3.2 Program Logic 수정

입력 데이터의 모든 쌍에 대한 연산이 요구되는 경우, 흔히 발생하는 시간 복잡도 $O(n^2)$ 는 $O(n)$ 과 비교, n 이 두 배로 증가할 경우 수행 시간은 4 배로 증가 [15]하는 시간 복잡도를 고려하여, 다음과 같이 기준으로 Program Logic 수정을 실행하였다.

- (1) 중첩된 Loop 문의 사용을 지양 하였다.
- (2) Full Scan 방식의 Query 문은, 필요한 data 를 memory 로 load 후 binary search 를 이용하여, 탐색 속도를 향상 시켰다.
- (3) 조건절의 누락으로 생성된 Index 를 이용하지 못하는 경우, 조건절의 해당 필드를 추가 하였다.

3.3 관련 Notes 적용

SAP R/3 System 은, ABAP/4 를 이용하여 개발된 CBO(Customer Built On) Program 및 Table 을 제외하고는 사용자의 직접적인 수정을 허용하지 않는다. 그러나 SAP社에서는 Standard Program 의 Performance 향상 등을 위하여 비정기적으로 Notes 를 제공하며 [16], 점검 대상 Program 과 관련된 적합한 Notes 를 검색하여 적용하였다.

3.4 Parallel Processing

Response Time 의 대부분이 Processing Time 이며 Program Logic 개선 또는 Table Index 의 생성으로 User Response Time 의 개선이 기대될 수 없는 경우, Application Server 의 Parallel Processing 을 이용하여 Processing Time 를 감소하였다.

4. 실증 분석

4.1 환경의 구성

본 연구에서는 각각 1 기로 구성된 Database Server, Application Server, Presentation Server 의 환경에서 테스트를 진행하였으며 Database Server 와 Application Server 는 1000Mbps, Application Server 와 Presentation Server 는 100Mbps Ethernet 으로 구성되었다. 각 Server 의 H/W 내역과 S/W 구성은 <표 3> 과 같다.

<표 3> 실험 H/W 및 S/W 사양

	Database Server	Application Server	Presentation Server
CPU	PA-8700 750Mhz*8	Ultra Sparc III 450Mhz*2	Intel P-4 2Ghz*1
MEM	12GB	2GB	1GB
HDD	666GB	36GB	120GB
OS	HP-Unix 11	Solaris 8	Windows XP
DBMS	Oracle 8.1.7	-	-

본 논문의 Tuning 대상 Program 목록은 <표 4>와 같다.

<표 4> Tuning 대상 Program 목록

	전체 P/G	사용 P/G	점검 P/G
Count	874	524	362

4.2 결과 및 분석

아래의 결과는 Database Server 의 Tuning 전·후 1 개월의 CPU, Memory, Physical Disk I/O Block 을 측정 한 값들이다. User Response Time 으로는 Database request time 과 Response Time 의 값을 측정하였다.

(1) CPU 및 Memory 사용현황

CPU 사용률은 기존 19%의 사용률에서 14%로 약 26.31%의 성능 개선 효과를 보였으며, Memory 사용률은 93%의 사용률에서 90%로 약 3.22%의 성능 개선 효과를 보였다. 그 결과는 <표 5>와 같다.

<표 5> Database Server CPU 및 Memory 사용률

	CPU 사용률	Memory 사용률
Tuning 전	19%	93%
Tuning 후	14%	90%
개선 효과	26.31%	3.22%

CPU 사용률에 비해 Memory 사용률이 개선 효과 비율이 떨어지는 것은, Table Index 추가에 따른 Memory Overhead 의 증가 때문인 것으로 추측된다.

(2) Physical Blocks 사용현황

<표 6>은 Database Server 의 Physical Block 사용현황이다. 기존 평균 12,341,000 Block 의 사용률에서 8,848,800 Block 의 사용으로 28.29%의 성능 개선 효과를 보였다. 이는 Tuning 시 생성된 Index 를 통하여 Access 하는 경우, 일상적으로 Access 되는 Table 에 대해서는 Index 의 Root Block 과 Branch Block 은 항상 메모리에 Cache 되어 있을 확률이 높으므로 Physical I/O 를 유발하는 비율이 감소 되었기 때문이다. 이는 생성된 Index 가 바람직하게 구현된 것임을 나타낸다.

<표 6> Database Server Physical Block 사용률

	Physical Block 사용량
Tuning 전	12,341,000
Tuning 후	8,848,800
개선 효과	28.29%

(3) Response Time 과 Database request time 측정현황

<표 7>은 Response Time 과 Database request time 측정현황으로 515.5ms 로 측정된 평균 Response Time

은 492.0ms 으로 4.55%의 성능 개선 효과를 보였으며, 120.6ms 로 측정된 평균 Database request Time 은 87.5ms로 27.44%의 성능 개선 효과가 측정되었다.

이는, Table Index 생성, Program Logic 수정, 관련 Notes 적용, Parallel Processing 을 이용한 Tuning 작업이 긍정적인 결과를 도출하였다고 판단된다.

< 표 7> Response Time 과 Database request time 측정현황

	Response Time	Database request time
Tuning 전	515.5ms	120.6ms
Tuning 후	492.0ms	87.5ms
개선 효과	4.55%	27.44%

5. 결론

5.1 연구 결과

본 연구의 목적은 User Response Time 을 향상시키고, 나아가 Database Server 의 Resource 사용률 개선 기법을 SAP R/3 System 환경에서 실증적으로 검증해 내는 것이다. User Response Time 을 향상을 위하여 본 연구는 상대적으로 느린 Database Request 횟수를 줄이고, Processing Time 비중을 증가시킴으로써, 전체 User Response Time 의 감소를 가져온다는 관점에서 문제에 접근하였고, 이를 검증하기 위해 전체 874 본의 CBO Program 중 사용중인 524 본의 Program 과 362 개의 Table 을 대상으로 실증적 분석을 실시 하였다.

분석 환경은 Database Server, Application Server, Presentation Server 를 구축하였으며, 연구 결과를 요약하면 다음과 같다.

Table Index 생성 55EA, Program Logic 수정 12EA, Notes 적용 5EA, Parallel Processing 3EA 를 통한 CPU 사용률 26.31%, 메모리 사용률 3.22%, Physical Block 사용률 28.29%, Response Time 4.55%, Database request time 27.44% 를 각각 개선하였다.

5.2 연구의 시사점

본 연구에서 제안하고 있는 두 가지는 다음과 같다. 첫째, Standard Table 및 Program 에서도 적극적인 성능향상을 위한 노력이 필요하다.

사용자의 수정을 허용하지 않는 Standard Program 의 경우에도, Table Indexing, Application Server Resource Sharing, SAP Notes 적용 등의 기법을 이용하여 성능을 향상시킬 수 있다.

둘째, CBO(Customer Built On) Program 의 경우 DB 설계 및 개발단계에서 최적화된 알고리즘의 사용으로 Time Complexity(시간 복잡도) 감소를 통해서 성능을 향상시킬 수 있다.

5.3 연구의 한계 및 향후 연구 방안

26 만개의 Table 과 120 만개의 Program 으로 이루어진 SAP R/3 System 의 경우, Tuning 을 통하여 모든 Transaction 의 성능을 향상시키는 것은 불가능하며 또한 Standard Table 과 Program 의 경우 사용자의 변경을 허용하지 않으므로 본 논문에서는 524 개의 Table 과 362 개의 Program 으로 국한된 Tuning 을 실시 하였다.

향후 관련 연구는 상기와 같은 본 연구의 한계점을 극복하기 위하여 ABAP/4 를 이용한 Standard Program 의 적극적인 재개발, Data Warehouse 구축 등을 이용하여 Standard Program 의 성능 개선을 통하여 user response time 을 보다 줄일 수 있는 tuning 방법론을 제시하는 것이다.

참고문헌

- [1] 남궁남, "2004 기업의 ERP 도입에 따른 ERP 교과 과정 연구"
- [2] SAP KOREA, <http://www.sap.co.kr>
- [3] Top 50 Software Support Service Providers Worldwide, 2005
- [4] B.Matzke, "ABAP/4-Die Programmiersprache des SAP-System R/3.Addison-Wesley,Reading,MA," USA. 1996
- [5] Jochen, Thomas, Alfons, Donald "Database Performance in the Read World"
- [6] 이원조, "관계형 데이터베이스에서 셀프조인 쿼리를 위한 성능평가" 2001
- [7] 김광은, "관계형 데이터베이스 시스템 성능향상을 위한 튜닝 방안" 2003 pp12, 24, 25
- [8] Lyman Do et al, "Issues in Developing Very Large Data Warehouse." In Proc. Int' l Conf. on VLDB.
- [9] Wu,M-C and Buchmann,A.P. "Research Issues in Data Warehousing", In Proc. Int' l Conf. ACM SIGMOD, 1997.
- [10] Wu, M-C and Buchmann,A.P. "Reserch Issues in Data Warehousing.Submitted for publication"
- [11] Wu, M.C. and Buchmann, Alejandro P." Encoded itmap Indexing for Data Warehouse. In Proc. Int' l Conf. ICDE 1998
- [12] George Collia. "OLAP, Relational and Multidimensional Database System". In ACM SIGMOD Record. 25(3)" 1996
- [13] Prasad Deshpande et al. "Cubing Algorithms, Storage Estimation and Storage and processing Alternative for OLAP", IEEE Data Engineering Bulletin 20(1):3-11, 1997.
- [14] James.Martin, "DB2 Concepts, Design, and Programming, Kathleen Kavanogh Chapman Joe Leban" 1989
- [15] 황중선 정영식 공저. "C 언어로 설명한 알고리즘" 2005
- [16] SAP Service Marketplace, <http://service.sap.com/message>