

호스트 메인프레임 환경의 데이터베이스를 위한 객체(Object) 최적화 관리 시스템

정성원
고려대학교 정보통신대학원
76dragon@korea.ac.kr

A Management System for Object Optimization in HOST Mainframe

Sungwon Jung
Dept. of Computer & Information, Korea University

요 약

데이터베이스는 매우 복잡한 성격을 지니고 있으며, 일반적으로 한 조직의 모든 데이터를 모아놓으므로 대형화되는 것이 보통이다. 따라서 데이터베이스는 항상 많은 데이터들이 생성 및 갱신을 반복한다. 이에 데이터베이스의 구성요소인 객체 또한 생성 및 변경을 거듭한다. 즉, 이러한 데이터의 증가 및 갱신으로 인하여 처음 생성되었을 때의 객체의 성능 및 저장 공간에 대한 이용은 향후 지속적인 관리가 없이는 보장받지 못한다. 이에 데이터베이스를 관리하는 사람들은 많은 인적 물적 자원을 할애하여 객체를 관리하고 최적의 상태로 만들기 위하여 노력한다. 본 논문에서는 이러한 일련의 행위들을 시스템에서 자동으로 자원을 수집하고 수집된 자원을 분석하여 시스템 특성을 반영한 관리기준에 부합하는 객체를 선별하여, 온라인 REORG(on-line reorganization) 작업을 통한 객체의 관리를 자동화 하여 보다 안정적인 데이터베이스 관리 및 데이터베이스의 관리의 효율성을 높이고, 데이터베이스 관리자의 생산성을 향상시키는데 기여 하고자 한다.

1. 서 론

데이터베이스란 자료 데이터파일을 조직적으로 통합하여 자료 항목의 중복을 없애고, 자료를 구조화하여 기억시켜 놓은, 자료의 집합체이며, 논리적으로 연관된 하나 이상의 자료의 모음으로 그 내용을 고도로 구조함으로써 검색과 갱신의 효율성을 꾀한다. 이런 데이터베이스를 효율적이고 안정적으로 사용하기 위하여 DBA(Database Administrator)는 항상 최적의 객체 상태를 유지하는데 많은 관리의 초점을 맞추게 된다. 어플리케이션의 성능개선을 통하여 사용자들의 사용에 대한 성능을 개선하기도 하지만, 항상 그 기본은 데이터베이스의 객체의 최적화에 있다고 하겠다.

이에 본 논문에서는 데이터베이스 객체관리를 위하여 객체 관련 정보를 시스템에 의하여 수집하고 그 수집된 자료를 가공하여 DBA가 설정하여 놓은 기준에 부합하는 객체를 선별하고, 선별된 객체가 최적의 상태가 될 수 있도록 제반의 작업을 수행하는 작업을 시스템에 의하여 자동적으로 수행되게 하는 객체 최적화 관리 시스템을 구축하고자 한다. 2장에서는 현재의 DBA가 운영하고 있는 현황을 정리하고 이에 관련된 문제점을 기술한다. 그리고 문제점에 대한 개선방향 및 추진목표를 기술한다. 3장에서는 객체 최적화 관리 시스템의 구축방법에 대해서 기술한다. 4장에서는 시스템 구축 후 최적화 작업 수행 결과에 대하여 기술한다. 5장에서는 시스템 구축으로 인한 데이터베이스 사용에 대한 기대효과를 기술한다. 6장에서는 결론 및 향후 과제에 대하여 논의한다.

2. 관련연구

2.1 데이터베이스 객체 최적화 관리의 필요성

데이터베이스관리에 있어 객체관리는 많은 시간을 필요로 하면서, 관리가 적절하지 못할 경우 많은 장애의 요인이 될 수도 있다. 객체의 관리가 부적절할 경우 사용자가

데이터를 입력 및 갱신 또는 조회하려고 할 때 적당한 서비스를 할 수 없게 된다. 또한 최적화가 되어 있지 않다면 장시간 기다리거나 서비스를 받지 못하게 된다. 이에 데이터베이스 관리자는 항상 안정적인고 효과적인 데이터베이스를 유지 관리하기 위하여 객체 관리에 많은 시간을 소비하고 있다.

2.2 현황 및 문제점

2.2.1. 시스템 구축 이전의 현황

시스템이 구축되지 전의 객체관리란 DBA에게 있어 단순하면서도 많은 시간이 소비되는 그런 작업이었다. 현재 사용 중인 저장 공간 디스크에 대한 전체적인 스캔 작업 이후 디스크에 존재하는 객체가 당일 사용가능하지를 일일이 점검하는 방식이었다. 이런 작업이다 보니, 날이 점검할 수 없고 일주일에 한번 작업이 이루어진다.

2.2.2. 문제점

위의 현황에 나온 작업들을 사람이 직접 확인하고, 계산하고, 유추하다 보니 작업의 실수로 인하여 정확한 현상파악이 되지 않을 뿐더러, 정확하지 못한 값으로 인하여 적지 않은 장애가 발생하게 되었다. 그리고 객체관리에 대한 기준이나, 어떠한 체계도 잡지 못하는 문제점을 가지고 있었다.

2.3. 개선방향

관리시스템의 구축으로 저장 공간 디스크의 모든 정보를 해당 데이터베이스와 비교하여 이를 데이터베이스화하고, DBA의 객체관리 기준에 의거된 관리 대상 건을 선별하여 이를 ALTER 및 REORG작업으로 항상 최적의 객체 상태가 될 수 있도록 한다.

2.4. 추진목표

데이터베이스 객체관리에 관련된 모든 작업을 관리시스템에서 자동으로 관리토록 하며, DBA는 관리 기준 및

체계를 확립하여, 정기적인 관리, 체계적인 관리를 하여 보다 안정적이고, 효율적인 데이터베이스 객체 최적화를 구축하는 것을 그 목표로 한다.

3. 시스템구축

3.1 객체 최적화 관리 시스템 구축 환경

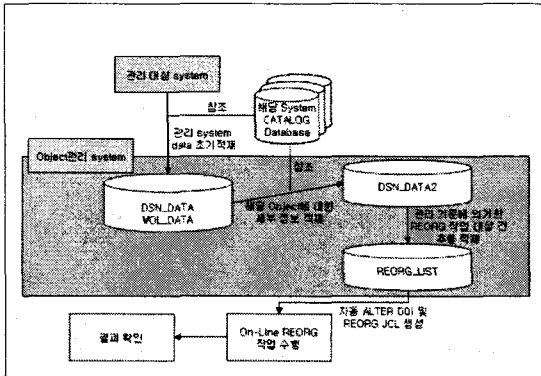
본 논문에서 구축하여 관리하고자 하는 현 시스템의 환경은 아래와 같다.

Seq	시스템	Version
1	데이터베이스	HOST IBM DB2 Version 7
2	데이터베이스 객체	테이블스페이스 : 2,300개 인덱스스페이스 : 4,298개 총 객체 수 : 6,598개
3	저장 공간 디스크	3,289본(총 9.80tera byte)
3	OS	Z/OS Version 1.2
4	어플리케이션 프로그램	COBOL, REXX

<표1> 현 운영 시스템의 환경

3.2 객체 최적화 관리 시스템 전체 구성도

본 논문에서 구축하고자 하는 객체 최적화 관리 시스템의 전체 구성은 아래와 같다.



<그림1> 객체 최적화 관리 시스템 전체 구성도

3.3 저장 공간 디스크내의 DB2 VSAM 데이터추출

디스크의 VTOC을 읽으면 아래의 정보를 추출할 수 있다.

3.3.1. VSAM 정보 추출

현재 사용 중인 디스크의 전체 목록을 파악하여 해당 디스크의 VTOC을 읽어서 저장되어 있는 VSAM의 정보를 추출한다.

Seq	컬럼 이름	설명
1	Dataset name	해당 볼륨에 저장되어 있는 데이터 셋의 명칭
2	Volume	데이터 셋이 저장되어 있는 저장 공간 디스크
3	Allocation Prim	최초 할당받은 데이터 셋의 크기
4	Allocation Sec	Extend 발생 시 할당하게 될 크기
5	Tracks Alloc	데이터 셋의 트랙으로 환산한 크기
6	Tracks Idle	최초 할당받은 것에서 남아 있는 크기
7	Tracks Ext	데이터 셋의 Extend 발생 횟수
8	Credate	데이터 셋의 최초 생성일
9	Usedate	데이터 셋의 마지막 사용일

<표2> 디스크안의 VTOC에서 추출가능 정보1

3.3.2. 디스크안의 사용 가능 공간 정보 추출

해당 디스크의 VTOC을 읽어서 그 중 가장 큰 사용가능 공간, 두 번째 세 번째 큰 사용공간에 대한 정보를 추출한다.

Seq	컬럼 이름	설명
1	Volume	해당 디스크 ID
2	Tracks Device	해당 디스크 전체 할당된 크기
3	Tracks Alloc	현재 할당되어 있는 총 크기
4	Tracks Avail	사용가능한 크기
5	Tracks Idle	총 할당된 크기에서 미 사용한 크기
6	Free space extents	디스크안의 총 데이터 셋의 Extend 횟수
7	Free space three largest	디스크안의 여유 공간 (순서대로 3개)

<표3> 디스크안의 VTOC에서 추출가능 정보2

3.2 추출 정보 관리 데이터베이스에 적재

추출된 각 각의 정보를 해당 데이터베이스 테이블에 적재한다. 이때 시스템의 카탈로그 데이터베이스를 같이 조회하여 객체의 기본적인 정보를 같이 취합하여 입력한다.

3.2.1. VSAM 정보 적재

데이터베이스의 객체는 생성 또는 변경에 있어 데이터베이스에서 정의된 한계 값을 가지게 된다. 이에 Extendable 즉 확장 발생 가능 횟수가 산출되고, 최초의 할당될 수 있는 크기, Extend발생 시 할당하게 될 크기 등, 기본적으로 객체생성 및 변경에 대한 정의된 값을 프로그램에 반영하여 값을 산출 후 아래의 테이블에 적재하게 된다.

Seq	컬럼 이름	설명
1	DSN_NAME	데이터베이스 객체 VSAM 이름
2	DSN_GUBUN	TB:테이블스페이스, IX:인덱스스페이스
3	DSN_DBNAME	객체 데이터베이스 이름
4	DSN_TSNAME	객체 테이블스페이스 이름
5	DSN_TSNAME	객체 테이블 이름
6	DSN_STG	객체 저장 그룹 이름
7	DSN_PARTNO	총 테이블스페이스 파티션
8	DSN_PARTNUM	해당 객체의 테이블스페이스 파티션
9	DSN_PRI_ALLOC	최초 할당받은 데이터 셋의 크기
10	DSN_SED_ALLOC	Extend 발생 시 할당하게 될 크기
11	DSN_USED	현재 사용 중인 크기
12	DSN_IDLE	최초 할당받은 것에서 남아 있는 크기
13	DSN_EXT	데이터 셋의 Extend 발생 횟수
14	DSN_AVAIL	해당 저장 그룹 내 Extend발생 가능한 크기
15	DSN_CRE_DT	객체 VSAM 최초 생성일
16	DSN_EXTENDABLE	해당 저장 그룹 내 Extend 발생 가능한 횟수
17	DSN_TYPE	L : Large 타입 테이블스페이스
18	DSN_CREATOR	객체 Owner
19	DSN_PRIM_EXT	객체 VSAM Primary Extend 가능여부

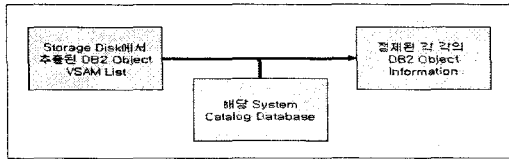
<표4> 객체 VSAM 테이블 컬럼 설명

3.2.1. 저장 공간 디스크 정보 적재

Seq	컬럼 이름	설명
1	VOL_ID	디스크 ID
2	VOL_DEVICE	디스크 전체 할당된 크기
3	VOL_ALLOC	현재 할당되어 있는 총 크기
4	VOL_AVAIL	사용가능한 크기
5	VOL_IDLE	할당된 크기에서 사용가능 크기
6	VOL_EXT	디스크안의 총 데이터 셋의 Extend 횟수
7	VOL_LARGE1	디스크안의 가장 큰 여유 공간 크기
8	VOL_LARGE2	디스크안의 두 번째 큰 여유 공간 크기
9	VOL_LARGE3	디스크안의 세 번째 큰 여유 공간 크기

<표5> 저장 공간 디스크 테이블 컬럼 설명

3.3 시스템의 카탈로그 정보를 이용하여 각 각의 객체에 대한 현황을 비교



<그림 2> 객체 정보 비교

디스크에서 추출하여 적재된 DSN_DATA의 각 객체의 한 데이터는 객체 VSAM의 각 한개 한 개의 정보를 담고 있다. 하지만 최초의 객체 VSAM이 생성되고 Extend가 발생하다 보면 최초의 저장 공간 디스크외의 다른 디스크로 VSAM이 Extend를 발생시키기도 한다. 따라서 같은 데이터베이스의 같은 테이블스페이스나 인덱스스페이스는 GROUP BY를 하여 합으로 계산하여야 한다. 해당 SQL의 구문은 아래와 같고 이에 정제되어 적재되는 DSN_DATA2의 테이블의 형태는 DSN_DATA 테이블의 형태와 일치한다.

3.4 해당 데이터베이스에서 당일 객체 관리 자동화 시스템 관리 대상 객체를 추출하여 데이터베이스에 적재 DSN_데이터2에 적재된 데이터를 DBA의 객체관리에 대한 기준에 의거하여 당일 관리 대상 객체를 선별하고 이를 REORG_LIST 테이블에 저장하게 된다. 이때 REO_TYPE은 DBA의 관리 기준에 대한 type이 들어가며 이는 DBA의 의사결정에 의하여 유동적으로 변경될 수 있다.

Seq	컬럼 이름	설명
1	REQ_DATE	추출된 날짜
2	REQ_DBNAME	객체 데이터베이스 이름
3	REQ_TSNAME	객체 테이블스페이스 이름
4	REQ_TBNAME	객체 테이블 이름
5	REQ_STG	객체 저장 그룹 이름
6	REQ_PRI_ALLOC	최초 할당받은 데이터 셋의 크기
7	REQ_SEC_ALLOC	Extend 발생 시 할당하게 될 크기
8	REQ_USED	현재 사용 중인 크기
9	REQ_IDLE	최초 할당받은 것에서 남아 있는 크기
10	REQ_EXT	데이터 셋의 Extend 발생 횟수
11	REQ_AVAIL	해당 저장 그룹 내 Extend발생 가능한 크기
12	REQ_EXTENDABLE	해당 저장 그룹 내 Extend발생 가능한 횟수
13	REQ_GUBUN	TB : 테이블스페이스 IX : 인덱스스페이스
14	REQ_TYPE	EX : Extend 과다 발생기준 초과 CR : Cluster ratio 기준 미달 EA : Extendable 가능 횟수 기준미달 ET : 기타 대상 객
15	REQ_PARTNUM	해당 객체의 테이블스페이스 파티션
16	REQ_CREATOR	객체 Owner

<표6> REORG 대상 객체 테이블 컬럼 목록 및 설명

아래의 표는 DBA가 객체를 관리하는데 있어 가장 기본이 되는 기준 몇 가지를 기술하였다.

Seq	REQ_TYPE	설명
1	EX (Extend)	Extend 과다 발생으로 관리 기준 초과 DSN2_DATA의 DSN_EXT의 함한 값으로 산출할 수 있다.
2	CR (Cluster ratio)	Cluster ration한 테이블 생성 시 정의된 데이터 적재 순서를 의미하며 시스템 카탈로그 데이터베이스에서 clustering은 "Y"이나 clustered가 "N"인 것들이 그 대상이 된다.
3	EA (Extendable)	데이터베이스 객체가 Extend를 발생하고자 할 때 그 발생 가능한 횟수가 DBA관리 기준에 미달되는 것을 그 대상으로 하며

		DSN2_DATA의 DSN_EXTENDABLE의 함한 값으로 산출할 수 있다.
4	ET (Etc)	기타 값이 많은 데이터가 생성 및 변경된 테이블이나 오랜 기간 관리지 못한 테이블 등 관리가 필요한 기타 요인들의 객체를 그 대상으로 한다.

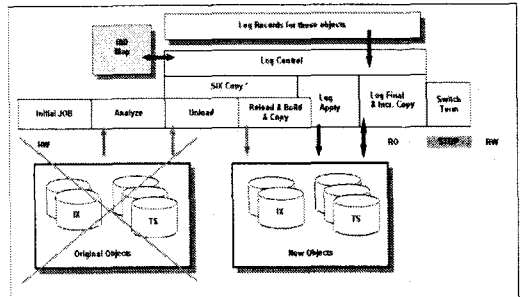
<표7> DBA 객체관리 기준에 대한 몇 가지 예

3.5 자동화 작업 DDL 및 JCL 생성
3.5.1 크기 ALTER DDL 생성

현재 사용 중인 객체의 사용 크기를 확인 후 해당 저장 그룹에 ALTER 가능 크기를 확인하고 현재 객체의 사용 크기에 1.3을 곱하여 크기를 증가시켜 주는 DDL을 생성한다. 이때 사용 크기에 1.3을 곱하여 주는 이유는 다년간 데이터베이스를 운영하여 온 많은 이들의 경험을 바탕으로 본 논문에서 임의로 결정한 값이다. 이 값을 관리자에 따라 유동적으로 증가 또는 감소하여 적용할 수 있다. 또한 ALTER DDL 작성 시 데이터베이스의 객체 한계 값을 고려하여 정의된 한계 값을 넘지 않는 범위 내에서 크기 조절이 이루어 지야한다. 또한 ALTER 크기 후 생성될 크기가 저장 그룹에 생성 가능한지를 파악하여 생성이 불가능 할 경우 이는 ALTER DDL이 아닌 작업 예러 디스플레이 창에 표현될 수 있도록 프로그램에 반영시킨다.

3.5.2 온라인 REORG JCL

REORG 작업이란 DBA가 객체를 관리하는 방법에 있어 기본이 되는 유틸리티 작업이다. 작업의 단계는 우선, 작업이 들어가는 시점의 로그 RBA값을 확정하고 그 RBA 값 이후의 객체에 발생하는 데이터의 변동이나 생성에 대하여 별도의 메모리에 RID값을 기록하여 놓는다. 그리고 데이터를 UNLOAD 하여 새로운 객체에 LOAD한 후, 메모리에 저장해 놓은 RBA값 이후의 데이터 변경 및 생성에 관한 로그를 반영한다. 그리고 마지막으로 해당 로그를 반영하는 동안 발생한 로그를 객체를 READ ONLY 상태로 변경 후 최종 로그 반영을 한다. 마지막으로 객체를 stop한 후 새로운 객체 VSAM과 과거의 VSAM객체의 이름을 바꿔준 후 과거 객체 VSAM을 삭제한다. 위의 작업에 대한 흐름을 그림으로 표현하면 아래와 같다.



<그림 3> 온라인 REORG 작업 프로세스 흐름도

3.6 시스템 JOB 자동 스케줄에 의한 동화 작업 수행
위의 과정을 거쳐 만들어진 ALTER DDL 및 온라인 REORG JCL은 시스템에서 매일 매일 수행되는 JOB 스케줄에 등록하여 일정 시간에 수행토록 한다.

3.6.1. 사용자 배치작업에 대한 통제

온라인 REORG 작업이 객체에 대한 READ/WRITE를 보장하여 준다고 하여도 REORG 아키텍처 구조상 객체의 최종 로그 반영시점과 VSAM 이름 변환시점에서는 객체에 대한 READ ONLY 및 STOP이 불가피 하다. 온라인 Thread는 이로 인한 피해가 없다고 하나 배치 작업의

경우에는 배치작업의 커서가 오픈 되어 패치가 이루어지고 있을 경우 객체에 대한 STOP이 불가능하다. 따라서 그로 인하여 객체 최적화 작업이 예러가 발생할 수 있고, 배치작업에도 -904의 예러가 발생할 수 있다. 따라서 하루에 일정 시간만큼은 사용자들의 배치작업을 통제하여 원활한 객체최적화 작업이 이루어질 수 있도록 하여야 한다.

3.6.2. JOB의 구성

작업은 DB2 자원의 경합을 최소화 하기위하여 하나의 작업으로 수행하며, 하나의 작업에는 크게 ALTER DDL과 온라인 REORG 작업이 하나의 큰 단락을 이루며 각각의 DDL 과 REORG 작업은 스텝으로 이루어진다.

3.6.3. JOB 리턴코드

상위 ALTER DDL 스텝에서 리턴코드 08 이상의 결과가 디스플레이 된다면 아래의 온라인 REORG 작업 스텝은 건너뛰게 되어 다음 객체의 ALTER DDL 스텝으로 넘어가게 된다.

3.6.4. 당일 예러 작업 객체 대상 건

그날의 예러로 인하여 작업이 수행되지 못한 객체의 작업은 익일 다시 추출되어 작업이 시도 되므로 익일 작업 시 예러가 다시 발생하지 않도록 알맞은 조치를 취하여 준다.

4. 시스템 구축 후 객체 최적화 작업 수행 결과

객체 관리 시스템의 환경 구축 및 관리 데이터베이스 생성 완료 후 데이터를 추출하고 추출된 데이터를 가공하여 데이터베이스에 데이터를 적재하고 이 데이터를 특정 기준에 의거하여 당일 객체 최적화 작업 대상 건을 추출한다. 그리고 최적화 작업을 수행하여 객체를 최적화 하는데 아래와 같은 작업의 단계에 2시간 3분 38초가 소요되었다. 2시간의 작업으로 데이터베이스의 1,865개의 모든 객체의 상태를 점검하고 기준에 적합하지 않은 객체를 추출하여 최적화 작업을 완료하였다.

구분	작업소요시간	내용
저장 공간 디스크 VTOC read	19분 24초	3,289(총 9.80tera byte) 디스크 탐색 총 10,076건의 객체 VSAM 정보추출
VOL_DATA 데이터 적재	03분 06초	3,289건의 디스크 데이터 시스템 카탈로그 데이터베이스와 비교 작업 이후 VOL_DATA 테이블에 적재
DSN_DATA 데이터 적재	08분 10초	10,076건의 객체 VSAM 데이터 시스템 카탈로그 데이터베이스와 비교 작업 이후 DSN_DATA 테이블에 적재
DSN_DATA2 데이터 적재	01분 03초	1,865개의 데이터베이스 객체 별 정보 생성 후 DSN_DATA2 테이블에 적재
REORG_LIST 데이터 적재	02분 31초	각각의 DBA 기준에 의거하여 객체 관리 대상 건 추출 및 REORG_LIST 테이블에 적재
객체 최적화 작업 JCL 생성	01분 12초	추출된 대상 건에 대한 크기 ALTER DDL 및 온라인 REORG 작업 JCL생성
최적화 작업 수행	1시간 23분 12초	대상 건으로 추출된 총 테이블스페이스: 4건과 총 인덱스스페이스: 12건에 대한 작업 수행
작업결과 확인 및 관리 시스템 점검	5분	수행결과 확인 및 관리 시스템 정상 가동 유무 확인

<표8> 각 단계별 작업 소요시간 및 작업 내용

5. 기대효과

5.1 DBA(Database Administrator)가용성 확보

객체 크기 증가 및 cluster ratio의 저하 등으로 인한 트랜잭션 Thread의 성능의 저하는 DBA들에게 있어 적지 않은 시간을 투자하여 관리하여 하는 직무이다. 상기 위의 시스템의 개발로 인하여 객체 최적화가 자동화된다면 DBA는 기존의 객체관리에 투자하던 많은 시간을 다른 생산적인 일에 투자할 수 있다.

5.2. 장애발생의 사전대처

객체의 데이터의 증가로 인하여 가용한 저장그룹에 적당량의 여유 공간이 없다면 또한 객체 VSAM이 Extend를 발생가능 횟수를 초과하였다면 이는 데이터저장에서의 장애를 의미하며, 이는 현 운영중인 시스템의 중단을 의미하기도 한다. 이런 장애를 사전에 인지하여 조치함으로써 현 운영중인 시스템의 가용성을 보장받을 수 있다.

5.3 사용자 성능 보장

객체의 최적화가 되어 있지 않다면 현재 운영중인 트랜잭션의 성능에 적지 않은 악 영향을 미치게 된다. 이에 항상 객체의 최적화를 만들어 놓는다면 사용자의 트랜잭션의 성능을 항상 일정하게 보장할 수 있다.

6. 결론 및 향후과제

본 논문에서는 데이터베이스의 객체를 보다 안정적이고 효율적으로 관리하기 위하여, 객체 관리시스템의 구축을 완성하였다. 이는 DBA의 업무 중에 있어 가장 중요하고 핵심이 되는 객체 관리에 대한 기준과 체계를 만들어 주는 바탕이 되었으며, 객체 관련 장애의 감소, 사용자에 대한 성능 보장 등 데이터베이스를 운영에 안정성과 효율성을 확보하는데 큰 기여를 하였다.

추후의 연구과제로서는 DBA의 데이터베이스 객체 관리에 대한 시스템의 구축뿐 아니라 객체의 성능에 관한 데이터를 수집, 가공하여 객체관리 뿐 아니라 객체 성능관리도 자동으로 관리되는 시스템을 구축 하고자 한다.

7. 참고문헌

- Mohana K. Lakhamraju, Rajeev Rastogi, S. Seshadri, S. Sudarshan "On-line reorganization in object databases " ACM Press 2000
- Gary H. Sockut, Robert P. Goldberg "Database organization Principles and Practice " ACM Press December 1979
- Mahmut Kandemir, Seung Woo Son, Guangyu Chen "Low power software design and sensing: An evaluation of code and data optimizations in the context of disk power reduction " ACM Press August 2005
- L. Manderson "Fill-in-the-blank JCL: why we did it and how" ACM Press October 1989
- M. K. Rajaraman "Performance evaluation through job scheduler modeling" ACM Press August 1982
- Edu Metz, Raimondas Lencevicius, Teofilo F. Gonzalez "Performance data collection using a hybrid approach " ACM Press September 2005
- John Mellor-Crummey, Robert Fowler, David Whalley "Tools for application-oriented performance tuning" ACM Press June 2001