

### 전력계통 수요예측을 위한 데이터베이스 설계

박정도\*, 송경빈\*\*, 백영식\*\*\*

\*위덕대학교 정보전자공학부, \*\*숭실대학교 전기전자공학부, \*\*\*경북대학교 전기전자공학부

#### Database Design for the Power System Load Forecasting

Jeong-Do Park\*, Kyung-Bin Song\*\*, Young-Shik Baek\*\*\*

\*Uiduk University, \*\*Soongsil University, \*\*\*Kyoungbuk University

**Abstract** - 전력계통의 수요예측은 수십 년간의 일별, 주별, 월별, 년도별 자료와 기타 수많은 계수들을 요구하므로, 처리해야할 자료의 양이 방대하여, 수요예측에는 데이터베이스의 사용이 필수이다. 본 연구에서는 수요예측 및 이와 유사한 대규모 자료의 전산화에 적합한 데이터베이스 설계기법을 소개하고, 계산 수행 시 속도 및 운용의 효율성을 기하기 위한 방안을 소개한다. 또한 데이터베이스의 유지보수를 위한 기법과 각종 접근 방법의 예를 들었다.

HDB (Hierarchical Database), NDB (Network DB), RDB (Relational DB), OODB (Object-Oriented DB), RTDB (Realtime DB) 등 다양한 형태로 발전하여왔다[2].

HDB/NDB: 초기 DB의 형태이며 현재 거의 사용되지 않으나, 70-80년대에 구축된 시스템에는 아직도 사용되고 있다.

RDB: 1970년 E.F. Codd에 의해 제안되었으며, 현재 가장 많이 사용되는 형태이다. Oracle, Sybase, MS-SQL Server 등 거의 모든 대규모 상용 DBMS(DB Management System)가 RDBMS의 형태이다. 데이터의 논리적인 관계를 이용하여 데이터를 효율적으로 관리한다.

OODB: RDB에서 발생하는 데이터 변환 문제인 Impedance Mismatch 문제를 해결하고 객체 지향성을 추가한 형태로 GemStone, Illustra (PostGres), UniSQL 등의 상용 DBMS가 존재하나 아직 널리 사용되고 있지는 않다[3].

RTDB: 실시간성을 보장하기 위하여 제안된 DB이며, 주로 시스템의 메모리에 상주하는 형태로 iLog와 같은 상용제품이 있다.

기존의 많은 연구에서 RTDB는 실시간보장의 특성을 기반으로 전력계통의 실시간 실측치 데이터 수집에 사용되었으며, RDB는 전력계통의 실시간 데이터를 제외한 모든 데이터의 관리시스템으로 사용되어왔다. 초기에는 전력계통의 해석에 X-Base(개인용 RDB) 계열의 데이터베이스인 Access, Paradox 등을 사용하였으나, 자료의 방대함과 X-Base에 존재하는 제약점들로 인하여 최근에는 Oracle 등 대용량 상용 DBMS를 사용하여 전력계통 해석 패키지를 구축한 사례[4]가 많았다. 본 연구에서는 상용 대용량 RDBMS인 MS-SQL Server 2k를 사용하여 데이터베이스를 설계하였다.

## 1. 서 론

전력계통은 그 규모가 방대하고 또한 매우 복잡하여, 수많은 종류의 장비들이 상호 유기적으로 연동되어 운전되고 있다. 따라서 전력계통을 해석하기 위해서는 많은 자료의 입력이 요구된다. 특히 전력계통의 수요예측은 수십 년간의 일별, 주별, 월별, 년도별 자료와 기타 수많은 계수들을 요구하므로, 처리해야할 자료의 양이 매우 방대하다. 이와 같이 대규모 전력계통을 해석하기 위해서는 올바른 데이터의 효과적인 취득기술과 수집된 데이터를 효율적으로 제공하는 기술이 필수이다.

그 동안 전력계통의 실무에서는 업무와 관련한 문제에 대한 자료를 별도로 수집하여 파일시스템으로 구축한 후 응용프로그램과 연계하는 개별적 문제해결방법에 익숙해 있었다. 일례를 들면 전력계통의 해석에 널리 사용되고 있는 PSS/E (Power System Simulator / Engineering)의 실무적용 예를 들 수 있다[1]. 이와 같이 개별적인 파일을 사용할 경우, 데이터의 접근에 다양한 제약이 따르므로 접근 속도가 느리며, 보안 및 관리가 힘들고, 자료의 불일치로 인해 동일한 문제에 대하여 서로 다른 결과를 산출하기도 하므로 매우 비효율적이다[2]. 그러므로 전력계통의 다양한 해석에는 여러 응용프로그램들이 동시·다발적으로 접근할 수 있는 대규모 데이터베이스의 개발이 크게 요구되고 있다.

따라서, 본 연구에서는 수요예측 및 이와 유사한 대규모 자료의 전산화에 적합한 대용량 데이터베이스 설계기법을 소개하고, 계산 수행 시 속도 및 운용의 효율성을 기하기 위한 방안을 제시한다.

## 2. 본 론

최근 각종 전산장비, 네트워크, 소프트웨어의 성능 및 데이터 처리 기술(DW:Data Warehousing, DM: Data Mining)의 발달로 인하여 기존의 데이터베이스 설계기법만으로는 최적의 데이터베이스를 설계하는 데에 한계점이 많다. 따라서 데이터베이스 설계 시 고려해야 할 사항들도 많은 변화가 있어야 한다. 본 절에는 대용량 데이터베이스 구축 시, 최근의 추세를 반영한 데이터베이스 설계 방안을 나타내었다.

### 2.1 데이터베이스의 종류와 용도

데이터베이스는 기초적인 파일시스템에서 출발하여

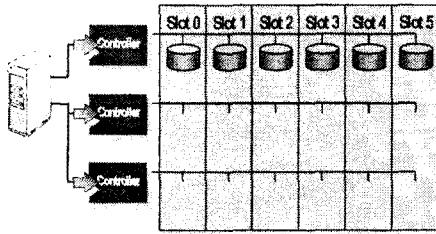
### 2.2 데이터베이스 설계

#### 2.2.1 Hardware 고려사항

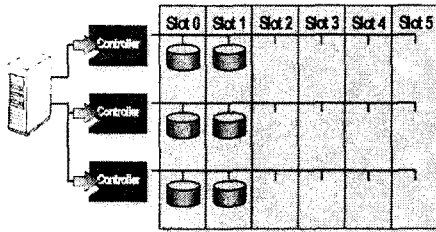
거의 모든 DBMS는 자료 저장소로 컴퓨터의 하드디스크를 사용하므로, 하드디스크 자체의 속도와 하드디스크 컨트롤러 및 하드디스크 배치 여부가 DBMS의 성능을 크게 좌우한다. DBMS는 주로 서버 장비에 설치하며, 일반 PC와 달리 서버 장비들은 다수의 RAID (Redundant Array of Inexpensive Disks) 장치들 통하여 하드디스크를 제어한다. RAID 장치의 특성을 최대한 활용하기 위하여 다음 사항을 고려하였다.

- 운영체제와 DBMS를 서로 다른 디스크, 다른 RAID 컨트롤러에 분리하여 설치
- 운영체제는 안정성을 위하여 RAID 1으로 구성
- DBMS는 경제적·내결함성을 위하여 RAID 5로 구성

- 그림 1 (b)와 같이 RAID를 구성하는 디스크를 모든 RAID 채널을 사용하도록 배치
- 임시 DB는 RAID 0로 구성
- 데이터 DB와 인덱스 DB는 병렬저장 가능한 RAID 5로 구성
- 로그 데이터는 병렬저장이 불가하므로 RAID 5의 이득을 볼 수 없기 때문에 RAID 1로 구성
- 시스템 DB는 잦은 변경에 대한 대비보다는 안정성 측면이 강조되므로 RAID 1로 구성



(a) 모든 슬롯 사용, 하나의 채널 사용



(b) 모든 채널 사용, 두 개의 슬롯 사용  
그림 1. 디스크 배치 비교

여섯 개의 디스크로 RAID 5를 구성하는 경우를 그림 1에 나타내었다. 그림 1의 (b)와 같이 배치하면 (a)에 비해 디스크 접근 속도가 최대 3배 까지 빨라진다.

### 2.2.2 File Group 할당

대부분의 DBMS는 다수의 물리적 파일을 하나의 논리적 파일그룹으로 지정하고, 다시 다수의 논리적 파일그룹을 하나의 데이터베이스 디바이스로 지정하는 기능을 제공한다. 파일그룹을 사용하는 것은 관리적인 측면뿐만 아니라 DBMS의 자원 사용 측면에서도 유리하며, 물리적 디스크 입출력의 분산에도 기여하므로, 결국 데이터베이스의 성능향상에 많은 도움이 된다. 또한, 대용량 데이터베이스의 백업 및 장애 복구 시 소요되는 시간을 획기적으로 단축시키는 이점을 가진다. 파일그룹의 할당에 있어서는 다음 사항을 고려한다.

- 병렬 테이블 스캔, 병렬 인덱스 스캔 또는 병렬 조인 (Join)작업과 같은 기능을 사용하기 위하여 주 파일그룹 이외의 파일그룹을 추가로 생성
- 각 파일그룹을 하나 이상의 파일로 구성
- 다수 개의 디스크를 사용하여 데이터가 분산 저장되도록 파일그룹 설정  
⇒ 이것은 데이터베이스 내의 데이터 및 인덱스를 분리하고, 읽기와 쓰기 유형의 테이블에 물리적 입출력의 분산이 되게끔 한다.
- 추가 데이터 또는 인덱스 파일마다 다른 드라이브나 다른 RAID 배열을 사용
- 시스템 데이터베이스와 사용자 데이터베이스를 분리
- 사용자 인덱스를 다른 디스크에 분리되도록 배치

이상을 고려하면 데이터베이스에 대한 각각의 접근 요청이 서로 다른 디스크나 RAID 스트라이프에서 수행되며, 궁극적으로 저장소 하드웨어가 병렬 쿼리 요청을 최

대로 지원하게 된다.

제시한 고려사항을 반영하여 데이터베이스를 구성하면 그림 2의 형태가 된다.

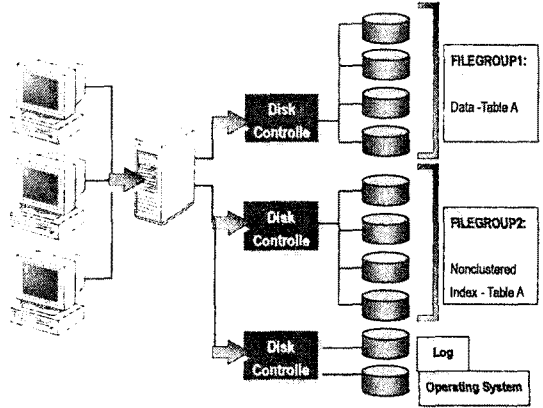


그림 2. 파일그룹을 고려한 데이터베이스

### 2.2.3 데이터베이스 구성 설정

데이터베이스의 구성에는 다음 사항을 고려하여 데이터베이스를 설정한다.

- 로그 파일의 크기를 무한 증가되지 않도록 제한
- 로그 파일의 크기는 데이터베이스 디바이스의 10-20% 로 설정
- 보안을 위한 데이터베이스 접근용 계정을 추가 설정
- 운영체제, 인덱스, 데이터, 로그 파일을 각기 다른 논리적 디스크에 배치
- 병렬처리 옵션 설정
- 데이터의 정렬, 임시테이블 및 join문 사용빈도를 고려하여 임시 데이터베이스의 크기를 증가 시킨다
- 임시 테이블의 사용이 빈번할 경우, 페이지 분열 현상이 발생하므로, 기존의 데이터베이스 내에서 임시 테이블을 분리하기 위해 새로운 데이터베이스를 추가하고, 새로운 데이터베이스 내에 임시 테이블을 배치
- 임시 데이터베이스를 위한 파일그룹을 따로 생성하여 페이지 분열 현상을 방지

### 2.2.4 테이블 생성

생성된 데이터베이스 내에 테이블을 설계하는 과정은 가급적 고전적인 방법을 따르되, 다음 사항에 각별히 유의한다.

- RDB의 특성을 고려하여 가급적 연관관계를 충분히 활용
- 정규화를 통한 테이블 설계
- 정규화를 위한 정규화 배치
- 응용소프트웨어의 접근 방식에 따라 정규화 된 테이블을 비정규화
- 정규화 및 비정규화 된 테이블에 대한 분석 실행

### 2.2.5 인덱스 설정

이미 언급한 방안으로 데이터베이스를 구성하고, 생성된 데이터베이스에 대용량 자료를 수집하는 각종 전력계통 테이블을 생성한 후, 테이블의 용도에 따라 다음을 고려하여 인덱스를 설정한다.

- 클러스터 인덱스는 범위 탐색(range search) 시 사용하며, 비클러스터 인덱스는 하나의 특정 행을 검색하는 경우가 빈번할 때 사용
- 전력계통 수요예측 데이터는 수십 년간의 데이터 중

특정 범위의 데이터를 요청하는 경우가 빈번하므로 클러스터 인덱스를 생성

- 발견기 데이터와 같이 특정 행의 검색이 빈번한 테이블에는 비클러스터 인덱스를 생성
- 조인 작업과 같이 정렬된 결과나 정렬된 중간 결과를 필요로 하는 경우 클러스터된 인덱스를 사용하여 검색 속도를 높인다
- 대규모 열에서 클러스터된 인덱스를 만들거나, 둘 이상의 열이 있는 인덱스를 만들면 B 트리 인덱스 구조가 복잡해지므로, 데이터에 빠르게 접근하기 위하여 클러스터된 인덱스를 가능한 간단하게 유지

### 2.2.6 운영체제 고려사항

운영체제에서는 다음을 고려한다.

- 페이지 파일의 크기는 물리적인 메모리의 1.5배로 설정
- 페이지 파일과 데이터베이스 디바이스를 서로 다른 물리적 디스크에 배치
- 가상 물리 메모리 증설

### 2.3 데이터베이스의 유지 및 보수

데이터베이스를 효과적으로 설계하고 구축하는 것만큼 이미 구축된 데이터베이스를 유지하고 보수하는 것 또한 중요하다. 구축중인 데이터베이스 또는 구축이 완료된 데이터베이스에 대하여 주기적으로 다음 사항을 검사한다.

- 인덱스 분열 상태를 주기적으로 확인
- 분열 상태가 80% 이하일 경우 인덱스 재생성
- 저장 프로시저(SP: Stored Procedure) 및 트리거들이 최신의 정보를 반영할 수 있도록 재 컴파일
- 최신의 쿼리 계획 생성

### 2.4 데이터베이스 구축

대용량 데이터베이스가 최적의 성능을 발휘하기 위해서는 데이터베이스 구축 시 전술한 바와 같이 다양한 측면의 고려사항들이 반영되어야 하며, 제안한 데이터베이스 구축 방안을 도식화하여 나타내면 그림 3과 같다.

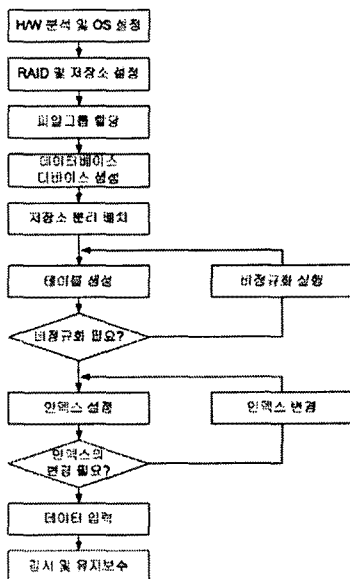


그림 3. 대용량 데이터베이스 설계 절차

### 3. 모의실험

제안한 방안으로 전력계통 수요예측 데이터베이스의 일부를 구축하고 성능 분석을 시행하였다. 성능 분석은 Compaq ML750 4Way 서버장비에서 수행하였고 제안한 방안을 모두 적용하여 HW를 구성하고 데이터베이스를 구축하였다. 모의실험은 일간실적모형 테이블에 대하여 행하였고, 모의 데이터로 1,000,000건의 데이터를 생성하여 입력한 후 이것을 사용하였다. 일간실적모형 테이블에 대하여 쿼리를 수행하고 속도를 측정된 결과를 표1에 나타내었다.

표 1. 동시 쿼리 수 vs. 응답속도

동시 쿼리 수	10	20	30	40	50	60
시간	1초 이내	1초 이내	1초 이내	1초 이내	1초 이내	2초 이내

모의실험 결과에서 알 수 있듯이, 제안한 방안으로 설계된 데이터베이스는 동시 쿼리에 대해서 빠른 응답속도를 나타내었다. 설계된 데이터베이스는 전력계통의 각종 응용소프트웨어가 동시에 접속을 하는 경우에 있어서 계산 수행시간에 도움을 줄 수 있을 것으로 기대된다.

### 4. 결 론

본 연구에서는 수요예측 및 이와 유사한 대규모 자료의 전산화에 적합한 데이터베이스 설계기법을 소개하였다. 또한, 계산 수행 시 속도 및 운용의 효율성을 기하기 위한 방안을 제시하고, 데이터베이스의 유지보수를 위한 기법을 제안하였다. 전력계통 수요예측에 요구되는 일간실적모형 테이블에 1,000,000 건의 모의데이터를 입력하고 동시 쿼리를 증가시키며 응답속도를 측정된 결과를 나타내어 그 타당성을 검증하였다. 제안한 설계 기법은 전력계통 뿐만 아니라 타 분야의 대용량 데이터베이스 설계에도 기본 지침으로 사용될 수 있을 것이다.

### [참고 문헌]

[1] 김인섭, 외, "영역별 수요분석을 위한 데이터베이스의 개발", 전력계통 해석 및 DB구축 연구회 Workshop 논문집, 171-186, 2000  
 [2] Korth Silberschatz, "Database System Concepts", McGraw Hill, 1995  
 [3] Elisa Bertino and Lorenzo Martino, "Object-Oriented Database Systems: Concepts and Architectures", Addison-Wesley, 1993  
 [4] "전력수급계획 및 운용해석 종합시스템 개발에 관한 연구" (최종보고서), 한국전력공사 전력연구원, 1998