

대규모 전력계통해석을 위한 연결형자료저장기법 적용에 관한 소개

김선군, 허성일
한국전력거래소

Application of Linked List Data Storage Scheme into Large Scale Power System Analysis

SEON GU KIM, SEONG IL HUR
Korea Power Exchange

Abstract - 실계통에서 운영되는 에너지관리시스템(EMS)의 계통해석 기능을 수행하기 위해서는 대규모 자료의 처리가 불가피하며 이에 따른 메모리의 사용에 대한 효율성, 즉 물리적인 자원의 제약성도 고려되어야 한다. 이러한 이유로 인하여 대규모 자료처리가 요구되는 실시간 시스템의 개발에서는 데이터에 대한 저장의 효율성과 접근의 용이성을 제공할 수 있는 자료설계 (Database Design)가 매우 중요하다 [1]. 이를 위하여 EMS에서는 고속의 연산속도에 대한 요구를 만족하며, 효율적이고 빠른 데이터 처리가 가능한 데이터저장기법이 적용된다. 본 논문에서는 상업용 EMS와 현재 개발중인 한국형EMS에서 적용되고 있는 자료저장기법인 연결형데이터저장기법 (Linked List Data Storage Scheme)에 대한 기능원리와 실제 적용에 대하여 소개하고자 한다.

1. 배경 및 필요성

물리적 상태의 전력계통에 대한 정상상태해석 (Steady State Analysis)의 일반적인 해의 접근방법은 비선형적 특성을 가지는 전력계통 방정식의 간략화 또는 선형화를 통하여 전력계통 상태변수들에 대한 선형적 관계를 가지는 선형연립방정식으로 변환하여 가우스소거법이나 뉴턴법 등의 수치해석기법이 사용되고 있으며, 이들 수치해석은 대부분 행렬의 처리를 기반으로 하고 있다. 특히 뉴턴법에 의한 해석은 그 과정에서 연산메모리를 많이 필요로 하는 역행렬 계산이 수반된다. 그러나 실제 전력계통의 경우 데이터의 크기가 매우 대규모이기 때문에 컴퓨터에 의한 해석과정에서 연산자원의 물리적 한계에 큰 영향을 받게 된다.

한편, 전력계통은 물리적인 특성상 네트워크의 연결구성은 자기모순과 몇몇의 인근 모선과의 연결로 구성되는 점을 감안하며 실제계통과 같은 대규모의 전력계통의 경우라도 전력방정식을 구성하는 어드미턴스행렬이 영이 아닌 값 행렬의 요소가 매우 적은 희소행렬 (Sparse Matrix)로 구성되고 나머지 대부분의 행렬요소가 제로가 된다는 점을 감안하면 조류계산과 오장계산 등과 같은 전력계통 상태해의 계산과정에서 행렬의 '0'요소들을 메모리에서 배제하는 데이터 저장기법 (Data Storage Scheme)을 활용한다면 컴퓨터의 저장자원인 메모리 사용감소는 물론 데이터 접근속도 향상을 통한 연산속도를 증가시킬 수 있다 [2-3].

본 논문에서는 실계통 적용을 위하여 개발중인 한국형EMS에서 전력계통해석을 위한 자료설계에서 적용된 연결형자료저장기법 (Linked List Data Storage Scheme)에 대한 원리와 실제 운전중인 전력거래소 EMS와 K-EMS에서의 적용에 대하여 소개하고자 한다.

2. 연결형자료저장기법 (Linked List Data Storage Scheme)

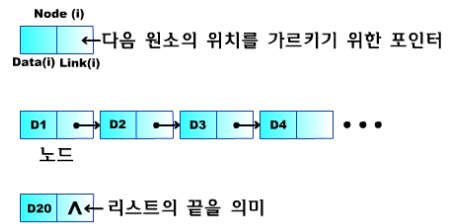
2.1 기본개념 및 원리

방대한 규모의 자료를 저장하기 위해서는 일반적으로 배열에 의한 순차적 저장방법이 사용된다. 이 방법은 구성요소간 논리적 관계가 순차적으로 인접하여 있고, 주기억장치에 저장된 자료가 물리적으로 인접된 성질을 이용하는 것이 순차적인 자료저장방법이다. 그러나 순차적인 자료저장은 자료의 요소가 추가, 삭제되는 경우 배열의 전체요소를 재배열하여야 하므로 자료의 처리속도가 많이 요구되거나 확정된 배열의 크기를 초과하는 과잉상태(Overflow) 현상이 발생할 수가 있다. 따라서 대규모의 자료저장에서는 자료의 연속성보다는 자료의 연결정보인 위치정보를 이용하면 자료를 접근하는 데 있어 처리속도와 자원(기억장소)관리에 성능을 증대시킬 수 있는 연결형 자료구조를 사용한다. 따라서 연결형 자료저장은 자료요소의 값 자체뿐만 아니라 다음 요소를 찾기 위한 위치정보인 포인터 또는 연결정보를 동시에 포함하여야 한다. 즉 자료의 요소간 연결 포인터를 이용한 연결형 또는 연결리스트 (Linked List) 방법이 자료구조처리에서 많이 이용되고 있다 [4].

연결리스트는 일정한 순서를 가지는 데이터 항목들을 표현하는 방법의 하나로서 배열과 같은 순차적 표현 방법과는 달리 데이터 항목들의 논리적인 순서만 유지되고 기억장소 내에서는 각 항목들의 임의의 위치

를 가지도록 하는 자료구조이다. 따라서 연결리스트에서는 각 데이터 항목들이 기억장소내의 어떤 위치에 어떤 항목이 있는 지를 표시해 주어야 한다. 즉, 데이터 항목에는 값뿐만 아니라 다음 항목의 위치 정보도 함께 저장해둔다. 이러한 데이터 값들 사이의 연결관계성이나 위치정보를 저장할 위해서는 포인터라는 개념이 사용된다.

연결리스트에 의한 자료저장에서 포인터는 각 항목의 다음 순서항목이나 앞 순서항목의 위치를 알려 주는 일종의 지시정보인 것이다. 그리고 연결리스트에 의한 저장기법에서 자료의 첫 번째 항목에는 이전 순서인 항목이 없고, 마지막 항목에는 다음 순서에 해당하는 항목이 없기 때문에 이들 값들이 갖는 포인터 정보는 Null값이나 기타 처음과 마지막이라는 정보를 알 수 있는 플래그 성격의 포인터 정보를 가지고 있다. 아래 그림은 연결리스트를 이용한 자료저장구조에 대한 기본개념을 표현하고 있으며, 노드(Node)는 배열의 레코드(튜플)과 동일한 의미로 사용된다.



<그림 1> 연결형자료구조의 기본개념

2.2 연결형자료저장기법의 종류

연결리스트를 이용한 자료저장은 포인터에 의한 자료의 연결방식에 따라서 단순연결형 (Single Liked List), 이중연결형 (Doubly Liked List) 및 환형연결형 (Circular Liked List)등으로 나누어 진다.

단순연결형 (Singly Liked List)은 위에서 설명한 바와 같이 각 노드는 노드의 값 (Data Field)과 이 노드가 연결될 노드에 대한 포인터 정보 (Linked Field)를 가지고 있으며, 이외에 첫 번째 노드인 Head Node와 마지막 노드에는 이를 인식하기 위한 특정한 플래그정보를 부여하는 것이 일반적이다. 마지막 노드의 표시를 위해서는 보통 '0'값, 즉 Null Point가 사용된다. 단순연결형은 연결부분이 오직 하나로 구성된다. 따라서 단순연결형 자료구조는 오직 한 쪽 방향 (오른쪽)으로만 자료의 접근과 탐색이 가능하다. 원형연결형(Circular Linked List)은 단순연결형과 같이 연결리스트의 시작을 알리는 Head 포인트에 대한 정보는 가지지만, 제일 마지막 노드가 Null Point 대신 첫 번째 노드에 대한 포인터 정보를 갖도록 하연 자료구조이다. 단순연결형과는 달리 원형연결형은 임의의 특정 노드에서 자료의 검색이 빠르나, 시작되는 특정 노드가 존재하지 않는 경우 무한 루프에 빠질 수 있는 단점도 가지고 있다. 마지막으로 이중연결형(Doubly Linked List)은 각 노드에 선행연결노드와 후행연결노드에 대한 정보를 모두 가지고 있는 자료구조형이다. 그리고 양쪽 끝에 있는 마지막 노드의 왼쪽 또는 오른쪽 연결부분에 Null Point가 입력된다. 이중연결형은 자료의 접근검색에 있어 매우 유연성을 제공하는 하나, 별도의 필드를 추가해야 하므로 메모리가 낭비될 수 있는 가능성이 있다.

3. 사례소개 : AREVA EMS의 데이터베이스 모델링

3.1 자료저장의 기본구조

현재 전력거래소에서 운영중인 AREVA EMS에서는 사용되는 데이터의 사용목적에 따라서 다양한 테이블 또는 행렬로 분류되고 각 테이블에는 설비의 물리적인 규모만큼의 레코드로 구성되어있다. 또한 테이블들은 자료의 연결관계성에 따라서 계층형과 비계층형으로 다시 분리하여 자료를 설계하였다. 전력계통의 물리적 구성요소와 같이 저장구조에

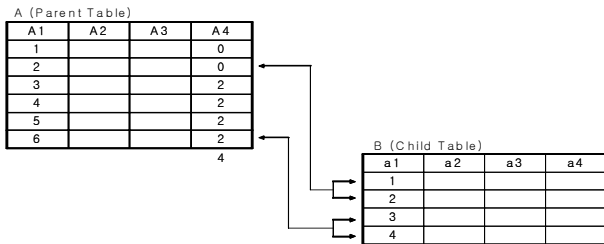
변동사항이 없거나 데이터간의 연결고리가 분명한 자료들은 계층형 구조(Hierarchical Structure)로 모델링 되었으며, 계층적 구조의 테이블에 대한 세부 설계방법은 오랜 기간의 개발경험에서 도출된 결과로서 설계계통에 대한 계통해석이나 화면처리를 지원하기 위한 접근속도를 향상을 위한 최적 구조로 설계되어 있다. 그리고 자료의 성격이 동적으로 생성되는 테이블의 경우 비계층적 구조(Non-Hierarchical Structure)로 설계되어 있다.

계층적 자료의 저장구조는 특정 테이블간의 상호관계성과 이 관계성을 연결하는 포인터로 구성되며 이 포인터값을 이용하여 테이블간 연결리스트가 만들어 진다. 현재 EMS에서 사용하고 있는 관계성에는 두 테이블에서 각각 1개의 튜플이 서로 연결되는 1:1 관계성과 테이블의 특정 튜플이 다른 테이블의 여러 튜플과의 연결성을 갖도록 하는 1:N 관계성을 적용하고 있다. 테이블사이의 관계성을 연결하기 위한 포인터는 Child 포인터, Parent 포인터 및 Indirect 포인터 등이 사용되고 있다.

첫째, Child Pointer는 상위계층(Parent)의 테이블에서 연결되는 하위계층(Child)의 테이블의 연결 데이터를 정의하는 포인터 정보이며, 1개의 상위계층 데이터가 여러 개의 하위계층 데이터와 연결 관계를 정의할 수 있다. 둘째, Parent 포인터는 Child 포인터와 반대의 개념으로 Parent 테이블의 튜플의 위치를 정의한다. 마지막으로 Indirect 포인터는 1:1 관계성을 두 테이블인 Source 테이블과 Object 테이블간 연결관계를 정의한다. Indirect 포인터는 Source 테이블 튜플이 연결되는 Object 테이블의 튜플의 위치값을 표시한다. Source 테이블의 튜플은 Object 테이블의 튜플과 중복하여 관계성을 가질 수 있으며, 연결정보가 없는 경우에는 '0'의 값을 입력한다. Indirect 포인터는 역으로 Object 테이블의 튜플은 Source 테이블로 연결되는가 가능하다. 즉, Forward/Backward 연결이 가능하다.

3.2 테이블설계 및 자료접근 알고리즘

이번 항목에서는 AREVA EMS에서 적용중인 연결형자료구조에 대한 설계방법과 Child 포인터를 이용한 특정자료의 접근방법을 소개하고자 한다. 아래 그림 2는 Child 포인터를 이용한 두 테이블간의 자료접근을 위한 설계사례를 도시하였다 [5].



〈그림 2〉 AREVA EMS의 Child Pointer의 설계사례

Parent 테이블의 각 튜플은 Child 포인터 필드인 A4 필드에 각 튜플에 대한 Child 테이블과의 연결정보를 가진다. Parent 테이블의 Child 포인터 필드에 입력된 연속된 Child 포인터값은 Parent 테이블 각 튜플의 Child 테이블의 참조영역을 정의한다. 그런데 이러한 포인터 참조방식은 마지막 튜플은 다음 값을 가지지 못하기 때문에 마지막 튜플이 참조할 두 번째 값을 EMS 내부에서 자동으로 만들어 준다. 그림2에서 보면 A테이블의 첫 번째 튜플은 자신의 Pointer 값(0)과 두 번째 튜플의 Pointer 값(0)을 Child 테이블의 Entry로 참조한다. 이와 같은 방법으로 하면 두 번째 튜플은 (0,2), 세 번째는 (2,2), 네 번째는 (2,2), 다섯 번째는 (2,2), 여섯 번째는 (2,4)을 Child 테이블의 참조영역으로 값을 갖게 된다. 그리고 실제 데이터 접근을 위해서는 Child 테이블의 첫 번째 튜플을 '0'으로 사용하였기 때문에 표1에서 보는 바와 같이 Child 테이블의 참조영역은 첫 번째 Child 포인터에서 1을 증가한 값을 사용한다. Child 테이블의 첫 번째 튜플의 위치값을 '0'으로부터 시작하는 데는 특별한 이유는 없이 오랜 동안 사용해온 프로그래밍 습관이다. 그리고 마지막으로 참조된 Child 포인터값의 유효성을 검증하기 위하여 두 번째 값이 첫 번째 값보다 반드시 커야 한다. 그리고 결과적으로 Parent 테이블의 튜플들의 Child 포인터값을 정리하면 아래와 같다.

〈표 1〉 Parent 테이블의 Child 포인터 참조결과

튜플	Pointer 필드정의	실제 Child 참조영역	유효성
1	(0,0)	(1,0)	무효
2	(0,2)	(1,2)	정상
3	(2,2)	(3,2)	무효
4	(2,2)	(3,2)	무효
5	(2,2)	(3,2)	무효
6	(2,4)	(3,4)	정상

표1에서 보는 바와 같이 Parent 테이블중에서 튜플2와 튜플6만이 정상적으로 Child 테이블과 관계성을 가지고 해당 영역을 참조하게 된다. 이 관계성을 이용하여 Parent 테이블에서 Child 테이블의 값을 접근하여 사용하기 위한 알고리즘 아래 표2와 같다.

〈표 2〉 Child 테이블 자료접근을 위한 Pseudo 알고리즘

```

Do      A = 1, LVA
      ! Child Pointer 값의 유효성 검사
      Do      B = A(A4), A+1(A4)
            ! Child 테이블 자료 처리
      End Do
End Do
  
```

4. 현장 적용 및 맺음말

전력거래소에서 현재 운전중인 AREVA EMS는 2001년부터 도입, 운영하여 온라인상태에서의 전력계통해석을 수행하여 계통운전을 위한 인 프라스시스템으로 사용하고 있다. 특히, 해석기능들이 연결형자료구조에 기반한 전력계통모델인 NETMOM을 기반으로 상태추정 및 상정고장해석과 같은 계산주기가 매우 빠른 프로그램은 물론, 최적조류계산과 같은 고급기능의 프로그램들이 성공적으로 운영되어오고 있다.

그리고 최근에는 한국형EMS개발에서 전력계통해석 응용프로그램에 대한 공통의 자료구조설계모델인 ACM(Application Common Model)에서도 연결형자료저장구조를 적용하여 상태추정과 급전원조류계산 프로그램 개발하여 현장시험단계까지 성공적으로 수행하였다 [6-7]. 향후, ACM 모델의 개선을 병행하면서 계통해석의 고급기능들이 상정고장해석, 최적조류계산 및 전압안정도를 고려한 송전가능용량계산등의 프로그램들에 대한 개발이 진행중에 있다.

마지막으로, 연속적인 접근방법 (Serially Access Method)에 기반을 두고 있는 최소행렬에 대한 연결형자료저장구조가 처음 제안된 이후 [8], 현재까지 전력계통해석분야에서 그 기법이 성공적으로 적용되고 있으며, 최근에는 보다 데이터 접근속도의 향상을 위한 랜덤접근법 (Random Access Method)에 기반을 둔 보완된 기법이 제안되고 있다 [9]. 그리고 최근 국제적으로 대두되고 있는 전력계통운용기술에 대한 새로운 패러다임의 대한 현장적용과 이를 위한 기술력의 확보, 및 더 나아가 개발기술의 글로벌화를 위해서는 새로운 도입되는 전력기기에 대한 전력계통모델링에 대한 개발과 이에 대한 효과적인 자료구조설계와 저장기법에 대한 기술개발이 향후 지속적으로 병행되어야 할 것이다.

참고 문헌

- [1] Jian Wu, Yong Cheng, and Noel N. Schulz, "Overview of Real-Time Database Management System Design for Power System SCADA System", 38th Southeastern Symposium on System Theory, March 4-6, 2006.
- [2] F. L. Alvarado, W. F. Tinney, and M. K. Enns. "Sparsity in Large-Scale Network Computation". In C. T. Leondes, editor, Advances in Electric Power and Energy Conversion System Dynamics and Control, volume 41 of Control and Dynamic Systems, pp. 207-272. Academic Press, 1991. Part 1.
- [3] S. A. Soman, S. A. Khaparde, Shubha Pandit, "Computational Methods for Large Sparse Power Systems: An Object Oriented Approach", Springer, 2001.
- [4] 김해곤, "임베디드 시스템 개발을 위한 자료구조", 그린, 2006.
- [5] AREVA, "Hdb Programmer's Guide", 2001.
- [6] 윤상운, 조운성, 손진만, 이옥화, 이진, 김홍래, 신만철, 허성일, 김성규, 이효상, "한국형 EMS의 Application S/W를 위한 Common DB의 설계에 관한 연구", 전력IT추진기술대회, 2007.
- [7] 윤상운, "KEMS Baseline DB 개발", KEMS ACM 설명회 자료, 05, 2008.
- [8] D. E. Knuth, "The Art of Computer Programming: Vol.1, Fundamental Algorithms", Addison-Wesley, 1968.
- [9] Krishna M. Sambarapu, S. Mark Halpin, "Sparse Matrix Techniques in Power Systems", 39th Southeastern Symposium on System Theory, March 4-6, 2007.