

ERP 환경에서 MRP 부품전개의 성능향상을 위한 연구

이형곤*(서울대 산업공학과), 나홍범(서울대 산업공학과), 박진우(서울대 산업공학과 교수)

Performance improvement study for MRP part explosion in ERP environment

H. G. Lee(Industrial. Eng. Dept., SNU), H. B. Na(Industrial Eng. Dept., SNU),
J. W. Park (Prof. at Industrial Eng. Dept., SNU)

ABSTRACT

There have been many studies to improve the performance of a database system focused on modifying data structure, data partitioning, and materializing strategy. The main contribution of this study is to propose a new alternative towards improving database performance by designing single table schema or processing queries virtually in main memory space. Material Requirement Planning(MRP) part explosion process has shown almost 2 times shorter under DB schema we suggested, and even more than 10 times shorter when separating and filtering policy of DB archiving process are assumed. Several experimental results are shown to illustrate the excellence of our solution.

Key Words : ERP(전사적 자원 관리), MRP(자재소요계획), Part Explosion(부품전개), Single Table Schema(단일 테이블 스키마), virtual main memory DB(메인메모리 가상 DB).

1. 서론

최근들어 기업들은 생산성 향상이나 이익의 최대창출과 같은 목적 달성을 위하여 고객만족이라는 패러다임을 받아들이고 있다. 이러한 기업들은 이전과는 다른 경영 지원도구를 요구하게 된다. 기업의 다양한 인적, 물적, 시간적 자원들을 전사적으로 관리하게 해주는 전사적 자원관리(ERP)가 그 대표적인 예라 할 수 있다. ERP 는 그 시초가 제조업이며, 제조업에서의 전사적 자원관리는 기준생산계획(MPS)의 수립과 이를 근거로 자재소요계획(MRP) 전개과정을 거쳐 생성되는 생산 및 구매주문의 산출이 핵심이다. 그러나, 제조품목이 수 천 개가 넘는 글로벌 제조기업들의 경우 MRP 전개과정을 수행하는 데에 많은 시간과 전산 자원이 소요되어 작업현장에 실시간으로 적용될 수 있는 생산계획을 수립하기가 어려운 실정이다. MRP 결과는 상위부품의 계획변경으로 인해 하위 부품의 계획이 변경된다. 그리고, 이 효과는 하위 단계로 내려갈수록 증가하게 된다. 이러한 MRP nervousness 는 MRP 자체에 대해 신뢰도를 감소시켜왔다.

MRP 전개방식의 개선에 관한 연구는 부품전개

방식과 MRP nervousness 를 줄이기 위한 방법으로 나누어진다. 이 둘은 불가분의 관계에 있고, 특히 MRP nervousness 는 MRP 가 해결해야 할 우선 과제라고 볼 수 있다. 그러나, 이에 대한 접근들은 대부분이 lot-sizing 기법으로 해결하려 하였다. lot-sizing 기법은 생산환경에 의해 조정되며, 최적을 찾기란 어렵다. 그리고, 고객의 주문 형태가 다양해 질수록 다품종 소량 생산의 경우에는 lot-sizing 기법을 적용하기가 더 어렵게 된다. 이에 비해 MRP 전개과정을 컴퓨터상에서 처리하는 절차나 방법론에 대한 연구는 거의 없는 실정이다.

MRP 부품전개과정은 데이터베이스 오퍼레이션의 관점에서 보았을 때 질의(query) 패턴이 일정하며, 지속적인 반복이 발생한다는 특성을 지닌다. 또한, 품목정보, BOM 정보 등 전개과정 중 데이터의 변경이 거의 발생하지 않는 기준정보(master data)와 연산결과와 시간주기에 따라 값이 변하는 거래정보(transaction data)로 크게 이원화되어 있어, 이용빈도에 따른 DB 접근방식을 개선할 여지가 있다. 본 연구에서는 부품전개 소요시간을 획기적으로 감소시킬 수 있는 단일 테이블 스키마 및 메인메모리 상의 가상 DB 의 활용을 제안하며 제시된 기법의 우

수성을 실험을 통해서 검증하고자 한다.

2. 접근방법

2.1 기존방법

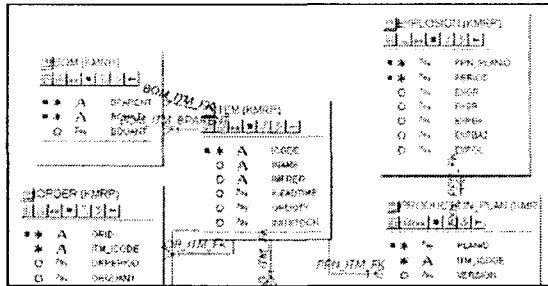


Fig. 1 기존의 관계형 데이터베이스 모델

MRP에 기반을 두고 발전한 ERP는 SAP R/3 나 Oracle Application 과 같은 대형 벤더들에 의하여 보다 방대하고 다양한 능력을 수반하게 되었다. 따라서 기업의 생산과정 전반에 걸쳐 다양한 분야의 프로세스들을 처리할 수 있게 되었으나, 정작 생산계획의 중심이 되는 MRP의 속도 향상에 대한 연구는 많지 않다.

Fig. 1은 대부분의 ERP 제품들이 MRP부품전개에 관련하여 따르고 있는 관계형 데이터베이스 모델이다. 품목정보(ITEM)와 자재명세서(BOM)가 분리되어 있으며, 기준계획이나 긴급주문을 입력받는 주문(ORDER)테이블, 부품전개결과를 저장하는 전개(EXPLOSION)테이블, 그리고 계획결과의 버전관리를 목적으로 하는 생산계획(PRODUCTIONPLAN)테이블을 상정하였다. 모든 단계의 MRP 로직이 적용될 때마다 ITEM-BOM-PRODUCTION_PLAN-EXPLOSION 테이블간에 조인연산이 발생하며, BOM테이블을 통해 자식품목이 무엇인지가 검색되며, 계획결과가 EXPLOSION 테이블에 저장되어야 자식

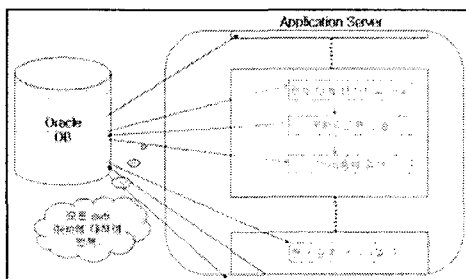


Fig. 2 DB 작동 측면에서 본 3-tier 형 ERP에서의 부품전개

품목의 부품전개를 수행할 수 있게 된다.

Database Server와 Application Server가 분리되어 있는 3-tier 형 ERP에서 부품전개과정 동안 발생하는 DB 조작과정을 개념적으로 나타내면 Fig.2와 같다.

2.2 단일 테이블 스키마

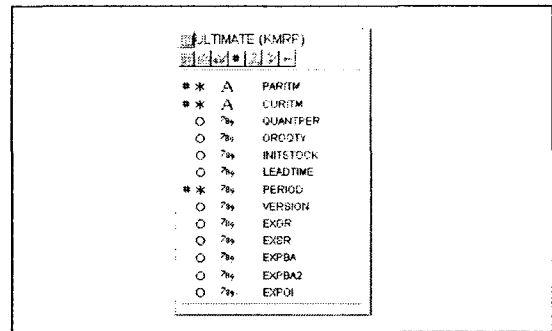


Fig. 3 단일 테이블형 모델

앞의 관계형 데이터베이스 모델에서 빈번하게 발생하는 join 연산이 야기하는 속도감소요인을 개선한 모델이다. ITEM, BOM, PRODUCTION_PLAN, EXPLOSION 등의 내용이 모두 ULTIMATE 라는 한 테이블에 표현되어 있다. 그러나, 이러한 Cartesian 테이블을 구성하기 위해서는 중복적인 데이터의 입력이 수반된다.

관계형 데이터베이스에서 연산속도와 데이터중복성의 trade-off에 관한 논의는 정규화(Normalization)라는 이론으로 오랫동안 있어왔다. 특히 빈번하게 발생하는 다양한 쿼리에 대하여 잘 정규화된 데이터베이스는 오히려 좋은 성능을 보이기 힘들다.¹⁾

부품전개속도의 향상이라는 실질적인 목표에서만 본다면 정규화의 원칙을 깨고 위와 같은 극단적인 예를 상정할 수 있겠으나, 방대한 자료가 갱신이 요구되는 ERP에서 위와 같은 모델은 데이터의 과도한 중복으로 인하여 발생하는 데이터 유지에 대한 비용 증가로 현실성이 떨어진다고 할 수 있다.

데이터베이스의 속도향상을 위해서 방대한 데이터베이스에 부분적인 접근과 수정을 가하는 방법으로 materialized view³⁾를 활용하는 방법과 data partitioning⁴⁾ 방법이 있다. 이는 데이터베이스의 일부분에 대한 검색 질의 및 수정 질의에 대하여 보다 빠른 수단을 제공할 수 있지만 연속적으로 발생하는 데이터베이스 전체의 갱신 및 수정 작업에 대해서는 오히려 계산의 복잡성을 야기할 수 있다.

빈번하게 발생하는 질의에 대하여 질의 패턴에 맞는 데이터베이스의 구조를 구성하는 연구도 있다.

¹⁾ 이는 관계형 데이터베이스에서 관계의 구조 설정을 통한 속도향상을 꾀하는 방법이다.

본 연구에서는 데이터의 구조나 추출 및 쓰기 방식에 대한 개선이 아닌, 컴퓨터 내에서 질의를 수행하는 물리적 위치의 변경을 통하여 데이터베이스의 질의처리 속도의 향상을 도모한다.

2.3 메인메모리상의 가상 DB 모델

데이터베이스 오퍼레이션의 효율성 측면에서 메인메모리 DBMS(MMDBMS)의 구현과 응용에 대한 연구가 최근 활발하게 이뤄지고 있다. SAP 와 같은 ERP 벤더들은 물론, Oracle, Informix 와 같은 데이터베이스 벤더들도 MMDBMS 기술을 보유한 업체들과의 기술 제휴를 모색하는 추세이다. 그러나, MMDBMS 는 메인메모리의 확장에 큰 제약이 따르기 때문에 대용량의 데이터를 처리하는 데에 한계가 있고, 부품전개의 관점에서 보았을 때 DB 연산 측면에서는 속도가 증가하겠으나, DB 접근의 측면에서는 개선의 여지가 없다. 한편, 컴퓨터 성능의 비약적인 향상에 힘입어 어플리케이션 서버 측의 메모리의 잉여를 활용, 데이터베이스 서버의 부담을 줄이고 서버 간의 비효율적인 트랜잭션을 줄이는 접근을 생각해 볼 수 있다.

본 연구에서 제안하고자 하는 메인메모리상의 가상 DB 는 Application Server 상의 메모리를 활용하며, 부품전개 과정에서 단계별로 요구되는 핵심이 되는 데이터만을 추출하여 메인메모리상에 보관한다는 아이디어가 적용되었다. 이를 개념적으로 묘사하면 Fig. 4 와 같다.

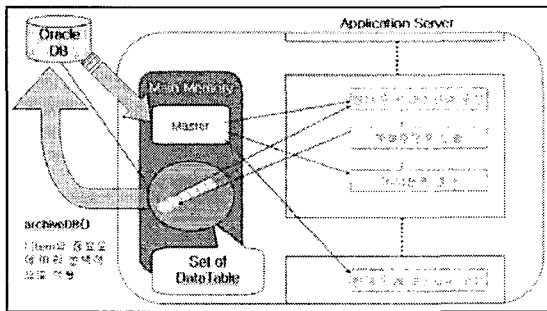


Fig. 4 메인메모리상의 가상 DB 모델

기존의 방식에서는 하나의 제품에 대한 계획 주문량(planned order)을 산출할 경우, master(ITEM 과 BOM 의 JOIN 정보)검색, child 검색, plan 결과 저장 등 모든 제품단계별로 세 차례의 DB 접근이 요구되었으나, 제안된 방식에서는 부품전개 초기에 최종제품의 생산계획 초기값과 master 만 DB Server 에서 가져오면 나머지 부품전개 과정은 메모리상에서 완료된다.

모든 정보의 통합적 관리를 지향하는 ERP 에서 는 부품전개의 결과 또한 대부분의 경우 저장되는 실정이다. 그러나, 제안된 방식에서는 부품전개과정 과 DB 저장(archive DB)과정을 분리시켰으며 이는 회사의 정책에 따라 중요부품에 대해서만 DB Server 에 저장하기를 선택할 경우 이를 부품전개과정에 반영할 수 있는 여지가 생긴다. 이는 MRP 부품전개방식이 최종제품의 계획이 먼저 결정되고 하위부품들의 계획이 차례로 결정되는 push 방식인 데 비해, 제조과정을 감안할 때 가장 먼저 생산계획을 전달받아야 할 대상은 부품벤더라는 점에서 현실적 의의가 크다고 할 수 있다. 즉, 부품전개 이후 DB 저장과정이 하루 혹은 이틀 정도로 길어진다 하더라도 메모리상에서 완료된 부품전개결과를 부품공급자에게 신속하게 전달할 수 있는 가능성이 열리게 된다.

제안된 메인메모리 가상 DB 방식을 구현하기 위해 구체적으로는 BOM 이 계층성을 띤다는 것을 착안, 각 제품에 대한 EXPLOSION 정보들이 서로 계층적으로 묶일 수 있도록 Microsoft .Net 에서 제공하는 DataTable 클래스의 배열을 활용함으로써 계층형 정보가 관계형 데이터베이스 스키마에 담겨있을 때 가져올 수 있는 제약들을 극복하였다.²⁾

3. 실험설계 및 결과

3.1 실험 데이터

실제 제조현장에서 사용하는 BOM 및 기준정보로 실험하는 것이 가장 이상적이겠지만, 정보유출의 문제로 인해서 실제 데이터를 얻을 수 없었다. 또한, 속도실험에 주안점을 둔 본 연구의 취지로 미루어 볼 때 가상 BOM 을 사용하더라도 큰 문제가 없었다.

BOM 의 경우 Fig. 5 와 같이 이진트리방식 즉, 하나의 부모품목이 두 개의 자식 품목을 갖도록 구성하여 depth 2 부터 depth 7 까지 구성하였다.

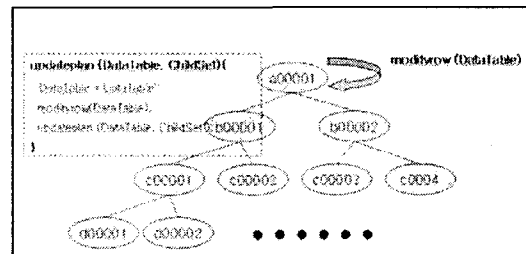


Fig. 5 이진트리 BOM 의 구성

한편, MRP 의 lot-sizing 방식은 고정주문량 방식을 사용, 제품별 주문량이 ITEM 테이블에 지정되어

있는 것으로 가정하였으며, 리드타임이나 초기재고 값들도 ITEM 테이블에서 임의로 설정하였다.

3.2 실험환경

대부분의 ERP 패키지들이 지향하고 있는 3-tier 방식을 따랐으며, ITEM 정보와 BOM 정보 등을 담고 있을 Main DB 로는 Oracle10g 가 사용되었으며, MRP 엔진은 Microsoft .net 으로 구현하였다. Table 1 은 실험에 사용된 DB Server 와 Appl. Server 의 주요 사양이다.

Table 1 시스템 사양

	DB Server	Appl. Server
CPU	1.26 GHz	2.4 GHz
Memory	1572MB	510MB
OS	Windows 2000	Windows XP

3.3 실험결과

2 장에서 소개되었던 관계형 데이터베이스 모델을 실험군(일반관계형)으로, 단일 테이블 스키마(통합형), 쓰기기능만을 메모리상에서 수행(MMDBW) 하는 경우와 읽기와 쓰기 기능을 모두 메모리상에서 수행(MMDBR&W)하는 경우를 대조군으로 두었다. 또한, 메인메모리상에서 부품전개가 완료되었다 하더라도 이를 다시 DB Server 상에 저장해야 실험군과 비교가 될 것이라는 점에서 아카이빙 프로세스가 포함된 MMDBW(DB)와 MMDBR&W(DB)군을 구성하였다. 실험은 BOM depth 2 부터 7 까지 차례로 수행하였으며 그 결과는 다음과 같다(Table 2).

Table 2 부품전개 실험결과

depth	(단위 : Sec)					
	2	3	4	5	6	7
일반관계형	0.74	1.39	3.64	6.56	23.25	54.15
통합형	0.73	1.30	1.83	5.64	19.56	33.79
MMW	0.06	0.13	0.28	0.36	1.03	3.24
MMW(DB)	0.31	0.61	1.42	2.70	5.31	31.16
MMA&W	0.03	0.07	0.10	0.23	0.51	1.19
MMA&W(DB)	0.28	0.56	1.27	2.25	4.70	22.69

메인메모리상에서만 부품전개를 수행하는 MMW 이나 MMA&W 는 BOM 단계별로 DB Server 를 접근하는 일반관계형이나 통합형에 비해 월등하게 속도가 향상되었음을 볼 수 있으며, DB Archiving 을 고려하더라도 두 배 가까이 빨라졌음

을 알 수 있다. 이는 같은 DB 읽기 및 쓰기 프로세스라 하더라도 batch 방식이 더 빠르다는 이론적 근거에 부합한다.

또한, 다소 변동성이 존재하지만 모든 정보를 하나의 테이블에 담았던 통합형의 경우도 일반관계형에 비해 우수한 성능을 보이고 있음을 알 수 있다.

4. 결론

본 연구에서는 단일 테이블 스키마와 메인메모리 가상 DB 구현이라는 아이디어를 적용하여 부품전개의 속도를 획기적으로 향상시킬 수 있었다. 특히, DB Archiving 을 부수적인 요소로 간주할 경우 메인메모리 방식은 10 배 이상의 획기적인 시간감소가 가능해졌다.

본 연구에서 사용된 구현 아이디어는 다음과 같은 점에서 이론상의 의의를 지닌다.

- DB 접근 횟수의 획기적 감소.
- 정보의 계층성을 관계형 데이터베이스 상에서 구현.
- 부품전개 프로세스와 DB Archiving 프로세스의 분리.

감사의 글

본 연구는 산업자원부에서 추진하는 차세대신기술개발사업의 하나로 수행되고 있는 '글로벌 정보공유 및 지식기반의 차세대 생산시스템 개발' 과제 의 지원을 받아 수행되었습니다.

참고문헌

1. Andrew N., Paulo B., Alok G., and James R., "Database design in the modern organization ? identifying robust structures under changing query patterns and arrival rate conditions? Decision Support Systems 37(2004) 435-447
2. Jagadish, H., "Incorporating Hierarchy in a Relational Model of Data," ACM SIGMOD, Vol. 18, Issue 2, pp. 78 - 87, 1989.
3. Jonathan, G., and Per-Ake, L., "Optimizing Queries Using Materialized Views," ACM SIGMOD, Vol. 30, Issue 2, pp. 331 - 342, 2001.
4. Ceri S., Negri M., and Pelagatti G., "Horizontal Data Partitioning in Database Design? Proceedings of the 1982 ACM SIGMOD international conference on Management of data, pp.128 ? 136, 1982.