

□ 기술예설 □

## 병렬 DBMS 아키텍처 분석

한국전자통신연구소 김양우\* · 박진원\*\* · 임기욱\*\*

● 목	차 ●
1. 서론	4. 상용 병렬 DBMS 제품 분석
2. 병렬 DBMS 아키텍처의 주요 요구사항	4.1 Informix사의 Dynamic Scalable Architecture(DSA)
3. 병렬 DBMS 아키텍처 분석	4.2 Oracle사의 Oracle Parallel Server(OPS)
3.1 OLTP 및 DSS 특성 비교 분석	4.3 Sybase사의 Navigation Server(NS)
3.2 공유메모리 구조(Shared Everything)	5. 맺음말
3.3 공유디스크 구조(Shared Disk)	
3.4 비공유디스크 구조(Shared Nothing)	

### 1. 서론

주로 과학계산 분야에서 사용되던 병렬처리 컴퓨터의 상용분야에 대한 새로운 진출은 시기적인 추세와 사회적 요구에 맞추어 이루어지고 있다. 즉, 병렬처리 컴퓨터는 기존의 메인프레임을 대체하며 클라이언트-서버 모델의 확장에 따른 대형 시스템의 다운 사이징 추세에 시기적으로 맞물렸으며, 사회적으로는 급속한 정보화 사회로의 진입 과정에서 그동안 온라인 트랜잭션 처리로 축적되어온 방대한 양의 데이터를 가공하여 정보화하는 과정에서 그 필요성이 부각되었다. 물론 여기에는 “정보는 곧 돈이다”라는 인식의 확산과 더불어 구소련의 몰락이 물고온 미국 국방예산의 대폭적인 삭감이 병렬 시스템 업체들로 하여금 자의반 타의반으로 상용분야로 눈을 돌리게한 점도 부인할 수 없다. 이와 더불어 Informix, Oracle, Sybase 등과 같은 독자적인 소프트웨어 개발 업체들에 의한 병렬 DBMS 패키지 개발도 결정적인 역할을

하였다. 이러한 일련의 사건들은 최근 2~3년 사이에 이루어 졌고, 국내에서도 한국전자통신연구소의 고속병렬컴퓨터(SPAX: Scalable Parallel Architecture computer based on X-bar network) 개발 사업과 서울대 컴퓨터 신기술 공동연구소 주관으로 AT & T GIS사의 NCR3600 기술 도입 사업 등으로 병렬처리 컴퓨터에 대한 관심이 많아지고 있다.

상용분야에 진출하고 있는 병렬처리 컴퓨터 시스템 업체들의 개발 동향을 보면 아래와 같이 몇 가지로 요약할 수 있다[1, 2, 3]. 첫번째로, 그들은 자신들 고유의 운영체제 대신에 UNIX를 채택하여 개방화를 도모하고 있으며, 두번째로, SMP 기반의 클러스터링 제품들이 속속 발표되고 있음을 볼 수 있다[4]. 이들은 고가용도와 정보처리 성능 향상을 위한 방법으로 주로 Ethernet으로 기존의 SMP들을 엮는 방식과 SMP구조의 노드들을 고유의 고성능 인터컨넥션 네트워크로 묶는 방식이 있다. 세번째로 그 위에 독자적인 소프트웨어 개발 업체들이 제공하는 병렬 관계형 데이터베이스 제품들을 탑재하여 대용량 데이터 처리 솔루션을

\*정회원  
\*\*중신회원

제공하고 있다.

이에 따라 본 고에서는 병렬처리 컴퓨터와 밀접한 관계를 갖는 상용 병렬 DBMS에 대한 여러 사항들을 살펴보고자 한다. 특히, 급변하는 정보화 시대 속에서 클라이언트-서버와 다운사이징, 아웃 소싱 등의 의미를 이해해야 하는 사람들에게 Informix, Oracle, Sybase사에서 거의 매달 개최하는 신제품 발표회들과 기술 세미나들은 그들에 대한 평가에 혼란을 가중시켜왔다고 생각되기에 본 고에서는 병렬 DBMS 아키텍처에 대한 주요 요구사항과 OLTP, DSS 특성에 따른 병렬 DBMS의 기본적 구조를 비교, 분석하며 끝으로 상용화된 주요 병렬 DBMS 최신 제품들의 소개와 이에 대한 분석을 다루고자 한다.

## 2. 병렬 DBMS 아키텍처의 주요 요구사항

병렬 DBMS 아키텍처가 갖추어야 할 몇 가지 주요 요구사항으로는 높은 가용도, 시스템 확장성, 대용량 데이터 및 자원 관리능력, 온라인 복합 처리 등을 들 수 있다. 그러면 이러한 주요 요구사항들이 어떠한 의미를 갖는지 알아보자.

### •가용도 :

관계형 DBMS에서 일반 사용자들의 고가용도에 대한 요구는 갈수록 높아져서 이제는 많은 응용환경에서 24×7×365 수준의 가용도를 요구하고 있다. 이러한 요구는 하드웨어 고장과 복구와 같은 비계획적 운영 상황 뿐만 아니라 주기적인 백업 작업과 같은 계획적 운영 상황을 모두 포함하는 것으로, 이 항목은 빠른 복구 기술과 상시 대기 기술 등과 같은 고가용도를 유지하기 위한 다양한 방안들의 유무와 그들의 성능 및 온라인 처리 가능성 등을 의미한다.

일반적으로 데이터베이스 로드, 백업, 복구, 무결성 확인, 색인 재구성 등을 위한 유틸리티는 모두 온라인으로 실행될 수 있어야 한다. 이를 위해 체크 포인트(check point)는 물론 심지어는 완전한 저널링(full journaling)을 제공해야 하는 경우도 있다. 또한 중복된 하드웨어와 중복된 제어 데이터를 이용하는 소프트웨어

미러링과 중복(replication) 기능들이 있다. 전자의 경우는 DBMS가 어떤 한 단위의 데이터를 다른 물리적 장치에 이중화하는 프로세스로 모든 쓰기는 미러에 투명하게 기록되고, 읽기는 원본과 미러 사이에 분산될 수 있다. 후자의 경우인 중복은 복사본이 원격지에 위치할 수도 있고, 동기식으로 갱신될 필요가 없으며, 전체 데이터베이스를 복사할 수 있다는 점이 미러링과 다를 뿐 기본적으로는 유사하다. 중복은 OLTP 응용에서는 원본을 갱신하고, DSS와 같은 읽기 전용 응용에서는 예비 복사본의 데이터를 사용할 수 있게 하므로 고가용도 측면 뿐만 아니라 OLTP/DSS 혼합 환경에서 큰 이점을 제공한다.

### •시스템 확장성 :

실질적으로 TPC 벤치마크 결과들은 다양한 RDBMS들의 성능을 비교 평가하는데 별 도움이 되지 못하고 있다. 똑 같은 TPC 벤치마크와 하드웨어 그리고 운영체제 환경에서 공평하게 다수의 RDBMS들을 평가한 자료들이 거의 없기 때문에 시스템 확장성을 알기 위해서는 다시 구조적 확장성과 효율적인 멀티 쓰레딩 지원 여부 등을 검토하여야 한다.

구조적 확장성은 병렬 RDBMS의 근본적 구조에 직접적인 연관이 있다. 일반적으로 공유메모리 구조, 공유디스크 구조, 비공유디스크 구조의 순으로 구조적 확장성은 증가한다. 이밖에 테이블 크기, 동시 연결 수, 메모리(버퍼) 크기, 로그 크기 등의 제한 사항이 내장되어 있는 아키텍처는 확장성이 약하며 아키텍처상에 동적으로 조정할 수 있는 기능(예를 들면 CPU 개수, 메모리 용량, 동시 실행 쓰레드 수 등)들이 많을수록 확장성이 유연하다. 또한 다중 쓰레드 방식은 다중 프로세스 방식에 비해 운영체제에 대한 오버헤드가 낮고 보다 높은 정도의 자원 공유의 효율성을 제공한다.

### •대용량 데이터 및 자원 관리능력 :

대용량 RDBMS 구성은 그 시스템의 데이터와 자원 관리 능력에 결부된다. 대용량 데이터를 관리하기 위한 여러가지 도구들과 기능들이 데이터베이스 관리자에게 제공되는지의 여

부와 대용량 데이터와 색인의 분할 및 변경, 데이터 로딩 등의 기능이 제한된 시간내에 온라인으로 이루어질 수 있는지의 여부를 확인하여야 한다.

• 온라인 복합 처리 :

온라인 복합 처리(On-Line Complex Processing)란 OLTP, DSS 및 일괄처리 등이 공통의 데이터 처리 자원을 공유하고 동일 데이터를 조작하는 혼합 환경으로 이전에는 DSS나 OLTP 등 한 환경에서 전용으로 사용되었으나 최근에는 이를 혼합하여 사용하는 경향을 보이고 있다. 따라서 이러한 혼합 환경에서의 자원관리, 튜닝 및 관리는 플랫폼이 어느 한 특정 유형의 처리에 전용될 때보다 동시 처리 문제 및 결합 문제 등으로 인하여 확장성에 제약을 가져오므로 이에 대한 기술적인 해결 여부가 주요 요구 사항이 된다.

3. 병렬 DBMS 아키텍처 분석

본 장에서는 먼저 상용 병렬처리 시스템의 주요 응용분야인 OLTP와 DSS에 대한 DBMS 측면에서의 특성을 비교 검토하며, 이들 분야의 업무 처리를 위한 병렬 DBMS의 3 가지 기본적인 구조에 대해 서로의 특징 및 장,단점들을 검토한다.

3.1 OLTP 및 DSS 특성 비교 분석

상용 병렬처리 시스템들은 범용 시스템으로

가는 중간 단계로 온라인 트랜잭션 처리(OLTP) 또는 의사 결정 지원 시스템(DSS)을 가장 큰 시장으로 보고 집중 공략하는 상태이다. OLTP 사용 분야로는 은행과 같은 금융 서비스 부문이나 기차·비행기 예약 시스템 등과 같이 수많은 트랜잭션들의 빠르고 정확한 처리를 요구하는 서비스 분야가 있고, DSS 사용 분야로는 증권시장 분석이나 각종 대규모 통계자료 분석과 같이 OLTP 시스템에 의해 축적된 대용량 데이터베이스를 분석하고 처리하여 유용한 정보를 만들어 내는 분야가 있다. 시장의 성숙도로 보면 OLTP는 이미 활성화되어 많이 쓰이고 있는 실정이고 DSS 쪽은 응용분야가 많이 개발되어 현재 활성화가 되어가고 있는 상태이다.

크게 보면 OLTP와 DSS 모두 DBMS의 한 유형으로 볼 수 있겠지만 각각의 시스템들이 쓰이는 목적이 다르고 처리하는 작업들의 성격이 크게 다른 관계로 각각의 분야들이 DBMS의 어떠한 특성들을 보이는지 표 1과 같이 비교하여 보았다.

표 1에서 보는 바와 같이 OLTP는 상대적으로 서로 독립적이며 많은 수의 트랜잭션들을 빠르게 처리하는 것이 필수적이다. 또한, 각각의 트랜잭션들이 필요로 하는 데이터들은 이곳 저곳에 분산되어 저장되어 있고 자주 이용되는 DBMS 기능들은 비교적 간단한 선택연산(selection) 및 갱신(update) 등이다. 특히, 이러한 OLTP의 수행은 ACID 특성을 만족시켜야 하는데, 이들은 Atomicity(단위성), Consistency(일관성), Isolation (공유데이터의 확보),

표 1 OLTP와 DSS의 작업 특성 비교[2]

Comparison Category	OLTP	DSS
Market Maturity	High	Medium to High
Real Time Requirement	High	Medium
Commercial DBMS Product	Infomix, Oracle Sybase	Teradata, Infomix Sybase, Oracle
System Availability Requirement	High	Medium
Type of Access	Reads & Writes	Mainly Read
Data Intensity	Medium	High
Complexity	Low to Medium	Medium to High
No. of Joins	Low	Medium to High
Query Selectivity	High	Low
I/O Access Pattern	Random	Mainly Sequential
Transaction Volumes	10-1000tps or more	10-500K queries/day

Durability(견고성)로 그들의 성질은 각각 다음과 같다[5].

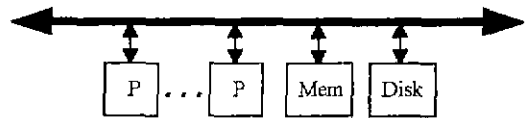
- Atomicity(단위성) : 한 트랜잭션에 연관된 모든 수행 기능들이 모두 성공적으로 수행되든지 아니면 전부 무효화시키는 것을 말한다.
- Consistency(일관성) : 모든 트랜잭션들은 항상 유효상태에 있어야 한다는 의미로 트랜잭션 수행중 수행이 중단되면 항상 그 이전의 유효상태로 돌아갈 수 있어야 한다.
- Isolation(공유 데이터의 확보) : 공유데이터를 사용하는 한 트랜잭션은 그 트랜잭션이 완전히 끝나기 전에는 다른 트랜잭션들로부터 영향을 받지 않는다.
- Durability(견고성) : 일단 성공적으로 수행을 마친 트랜잭션의 결과는 시스템 실패 시에도 살아 있어야 한다.

반면에 DSS는 서로 연관되는 많은 화일들을 가공 처리하여 새롭고 유용한 결과를 도출해내는 것으로 대량의 데이터베이스 화일들을 순차적으로 읽는 작업이 많고, 이렇게 읽혀진 데이터베이스 화일들을 대상으로 이루어지는 데이터베이스의 기능(operation)들은 주로 복잡하고, 특히 많은 결합연산 기능을 필요로 하는 특성을 갖는다[6].

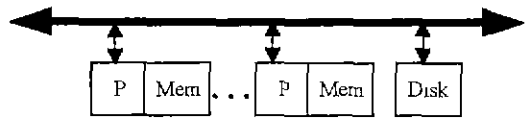
앞에서 살펴본 바와 같이 OLTP와 DSS는 서로 다른 특성을 가지고 있다. 이러한 이유로 기존에는 OLTP와 DSS를 동시에 제공하기 위하여 두개의 서로 다른 시스템과 두개의 데이터베이스를 갖추고 운영하였다. 물론 서로 다른 특성을 가진 작업을 각각의 시스템에서 수행시키면 효율적으로 관리할수 있고 각각의 수행 특성에 맞게 시스템을 최적화하여 운영할 수 있다는 장점이 있다. 그러나 디스크를 포함하여 두개의 값 비싼 시스템을 관리해야 하고, DSS용 시스템은 주로 최신 버전의 데이터가 아닌 데이터를 상대로 수행하여야 하며, OLTP용 시스템의 최신 데이터를 사용하여 주기적으로 DSS용 데이터베이스를 업그레이드 시켜 주어야 한다는 큰 단점이 있다.

또한 OLTP와 DSS를 하나의 시스템에서 동시에 수행시키는 경우에는 DSS를 병렬 수행시키기 위해 많은 컴퓨팅 자원을 배정하면 OLTP

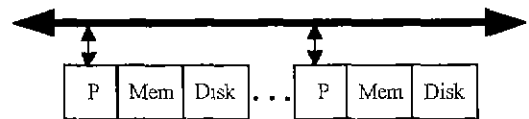
가 희생될 수 있으므로 우선 순위에 대한 특별한 고려가 필요하다. 그리고 많은 트랜잭션들을 동시에 처리하기 위해서는 작은 단위의 록킹을 필요로 하는데 이것은 비교적 많은 화일들을 동시에 접근해야 하는 DSS에 많은 오버헤드를 준다. 반대로 DSS 수행시 잡혀져 있는 록(lock)들은 갱신연산을 요하는 많은 트랜잭션들의 수행을 지연 시킬 수 있다. 이러한 오버헤드들을 감소시키는 방안으로 Unlocked Reads, Transient Versioning 등과 같은 기술들이 제안되어 있다[7, 8].



(a) 공유메모리 구조



(b) 공유디스크 구조



(c) 비공유디스크 구조

그림 1 공유메모리, 공유디스크, 비공유디스크 구조 비교

### 3.2 공유메모리 구조(Shared Everything)

공유메모리 구조는 메모리와 디스크를 여러 프로세서들이 서로 공유하는 형태를 가지고 있다. 그림 1에서 보듯이 이 구조는 대칭형 다중 처리 구조(SMP)와 같은 것으로 기존의 Sequent나 Encore와 같은 SMP 기종들에서 사용되는 구조이다. 이들 외에 IBM 3090에서 쓰였던 DB2, 또 Bull의 DPS8 등이 이에 속한다[8, 9, 10].

이 구조는 작업안배, 단순함 및 가격경쟁력

측면에서 장점이 있으나 낮은 확장성 및 낮은 가용성이 단점이다. 즉, 단일 프로세서 시스템에서 프로그래밍하듯이 프로그래밍이 용이하고, 실제 부하를 실행시(run time)에 바로 프로세서들에게 나누어 주는 등 작업안배가 용이하며, 질의어간의 병렬성(inter-query parallelism) 또한 다중 프로세서들에 의해 쉽게 처리된다. 반면에 공유메모리 접근시 서로간에 충돌 가능성이 높아짐에 따른 성능의 저하 및 낮은 확장성 문제, 그리고 메모리를 모든 프로세서들이 공유함에 따라 오류 확대 가능성이 커지고 이에 따른 낮은 가용성 등이 문제점으로 지적되고 있다. 그리고 시스템의 최대 구성이 대략 30여개의 프로세서들을 수용할 수 있는 수준으로 제한되어 확장성에도 제약이 갖는다. 그러나 이 방식은 가장 먼저 시도된 방식으로 기술적으로도 안정되어 있고 상용 RDBMS 제품도 Informix, Oracle, Sybase 등 여러 업체에서 제공하고 있다.

### 3.3 공유디스크 구조(Shared Disk)

이 구조는 공유메모리 구조와 비공유디스크 구조의 중간 형태로 각각의 프로세서들은 자신만의 비공유메모리를 가지나 디스크는 서로 공유하는 구조를 갖는다. DEC VAXcluster와 Ncube에 올렸던 Oracle, 그리고 IBM IMS/VS의 Data Sharing 등이 이 방식을 취하고 있다.

각각의 프로세서들은 공유디스크에 있는 모든 데이터베이스 페이지들에 접근할 수 있고 이때 같은 페이지로의 접근충돌을 피하기 위하여 시스템 차원의 록킹(global locking)과 캐쉬 일관성 유지 프로토콜 등을 필요로 한다. 작업안배 및 기존의 데이터베이스로부터의 이주는 공유메모리 방식에서와 같이 유리하며, 확장성은 대략 200~300개 프로세서 수준까지 가능하고, 가용성은 오류의 확대가 한 프로세서-메모리 노드로 제한될 수 있어 공유메모리 방식에 비해 높은 가용성과 확장성을 갖는다[8].

성능 향상을 위해서는 각 노드(SMP노드, 또는 한 개의 프로세서-메모리)당 비교적 큰 메모리에 데이터베이스 버퍼 캐쉬를 두어 디스크 입출력을 최소한으로 줄인다. 데이터베이스 버퍼 캐쉬의 운영방식은 일반 캐쉬 관리와 비슷

하고 이와 더불어 디스크 입출력을 줄이기 위해 "Fast Commit", "Group Commit", "Deferred Write" 등의 기술을 사용하여 변경된 데이터 블록을 디스크에 바로 쓰는 대신 일단 "Re-Do Log File"에 변경된 부분들만 모아 쓰고 나중에 시스템 부하가 적을 때 또는 일정 시간마다 주기적으로 디스크에 변경사항을 기록한다. 데이터 일관성 및 공유 데이터의 확보를 위해서는 "행단위 록킹" 및 "다중버전을 이용한 동시처리 제어" 기술 등이 있다. 이러한 기술들을 사용하여 "쓰기 후 읽기", 또는 "읽기 후 쓰기" 등의 데이터 일관성 유지 문제를 해결한다. 클러스터 구조의 소결합 시스템에서는 구성 노드당 DBMS 엔진을 가지며 각각의 노드 내에서는 쓰레드 또는 프로세스 단위로 병렬처리를 적용한다.

공유디스크 방식의 대표적인 상용 제품으로는 Oracle의 Oracle Parallel Server(OPS)가 있으며[12, 16], 이미 NCR, Ncube 등 10여개의 클러스터 및 MPP 구조의 시스템에서 운영중에 있다.

### 3.4 비공유디스크 구조(Shared Nothing)

비공유디스크형 병렬 DBMS 구조는 "shared nothing" 구조라고도 하며 각각의 노드는 자신만을 위한 프로세서(들), 메모리, 그리고 디스크를 갖는다. 그림 1에서 보듯이 그 구조가 분산 DBMS와 같은 형태를 갖기 때문에 기존의 분산 DBMS에서 개발된 기술(분산 트랜잭션 관리, 분산 질의어 처리, 데이터베이스 분산 기술)들을 바로 적용할 수 있다. 비공유디스크 구조를 갖는 DBMS 시스템들로는 Teradata의 DBC, Tandem의 NonStop SQL, 그리고 상대적으로 최근에 발표된 Sybase의 Navigation Server 및 Informix의 Dynamic Scalable Architecture와 같은 상용 제품들이 있고 Bubba, Prisma, Gamma, 그리고 Grace와 같은 시제품들이 있다[11, 12, 14, 18].

비공유디스크 구조는 기본적으로 성능 향상을 위하여 데이터 분할(data partition)에 의한 데이터 병렬성을 추구하는 방식으로 확장성 및 데이터 일관성 유지가 용이하다[11]. 데이터 분할 방식으로는, "순차적 분할", "영역 분할",

“해성 분할”, “스키마 분할” 등이 있고 데이터 사용 패턴에 따라 분할 방식을 지정해 준다. 각각의 노드들은 자신에게 분할되어 디스크에 저장된 데이터에 대한 소유권을 가지며 관리에 대한 책임을 진다. 사용자의 한 트랜잭션 명령은 필요한 데이터가 분할되어 있는 하나 또는 여러개의 노드(들)로 보내지고(split), 각각의 노드들은 자기가 소유하고 있는 데이터에 대해 데이터베이스 작업을 수행하며(parallel processing), 그 결과를 한 노드에 모아(merge) 결과를 취합한다.

이 구조는 데이터 분할에 의한 병렬성 추구로 성능을 증가시키는 방식이나 반면에 데이터 분할 자체가 가장 큰 단점이 된다. 즉, 기존의 대용량 데이터베이스를 비공유디스크 구조의 시스템에서 사용하기 위해서는 데이터 분할이 큰 오버헤드가 되고 또한 프로세싱 노드나 디스크 수의 변화와 같은 시스템 확장시에도 매번 이러한 오버헤드를 겪어야 한다. 또한 차세대 응용분야라 할 수 있는 멀티미디어 및 CAD/CAM 데이터베이스 처리에서는 데이터 분할이 쉽지 않기 때문에 더욱 큰 문제로 대두되리라 예상된다. 이러한 데이터 분배(data allocation)를 위한 데이터 분할은 작업 안배와 통신 오버헤드, 병렬성 정도 간에 서로 상반된 관계를 갖고 있다. 즉, 성능 향상과 빠른 응답시간을 위해선 절의어간 및 하나의 절의어 내의 병렬성을 높여야 하고 이를 위해서는 데이터베이스를 더 많은 숫자로 분할해야(declustering) 하기 때문에 통신 오버헤드를 증가시키고

작업안배를 더욱 어렵게 한다[8].

비공유디스크 방식의 대표적인 상용 제품으로는 Sybase의 Navigation Server가 있으며, 현재 NCR 3600에서 동작되고 있고 IBM SP2에 설치중에 있다. Informix사의 Dynamic Scalable Architecture 또한 기본적으로 비공유디스크 형태를 가진다. 따라서 위에서 설명한 비공유디스크의 기본적 특성들은 모두 적용되고 할 수 있다. 표 2는 병렬 DBMS가 운영되는 3 가지 방식에 대한 상대 비교 내용을 요약한 것이다.

#### 4. 상용 병렬 DBMS 제품 분석

현재까지 SMP급 이상의 UNIX 시스템을 대상으로 솔루션을 제공하고 있는 대표적인 업체와 제품으로는 Informix사의 Dynamic Scalable Architecture(DSA), Oracle사의 Oracle Parallel Server(OPS), 그리고 Sybase사의 Navigation Server(NS) 등을 들 수 있다.

이들은 공통적으로 SMP급 시스템에서는 공유메모리 형태의 구조를 가지고 있고 그 이상의 클러스터 또는 MPP급에서는 Oracle의 경우 공유디스크 구조를, 그리고 Informix와 Sybase는 비공유디스크 구조를 갖고 있다. 다음에 소개되는 제품들의 평가는 객관성을 기하기 위해서 주로 가트너 그룹의 자료[13, 15, 17]들과 데이터 프로의 자료[14, 16, 18]들을 인용하였음을 밝혀 둔다. 그리고 소개하는 순서도 단지 그 회사 이름의 알파벳 순서에 따랐다.

표 2 공유메모리, 공유디스크, 비공유디스크 구조들에 대한 상대 비교

비교 항목	공유메모리 구조	공유디스크 구조	비공유디스크 구조
확장성(프로세서 갯수)	최대 약 30여개	최대 약 300여개	최대 약 1,000여개
고가용성	보통	비교적 높다	높다
부하균형	높다	비교적 높다	보통
기존 DB와의 호환성	높다	비교적 높다	낮다
기술적 성숙도	높다	비교적 높다	보통
메모리 효율성	높다	보통	비교적 높다
상호 간섭성	보통	높다	낮다
대용량 OLTP를 위한 구조적 적합성	비교적 높다	보통	비교적 낮다
대용량 DSS를 위한 구조적 적합성	비교적 낮다	보통	높다
상용 DBMS 업체	Oracle, IBM, Informix, Sybase, Ingres	Oracle, IBM	Teradata, Sybase, Informix

#### 4.1 Informix사의 Dynamic Scalable Architecture(DSA)

Informix DSA는 병렬 플랫폼들을 목표로 디자인된 제품으로 확장성이 우수하고 특히 가격대 성능비에서 타제품에 비해 우수하다. 반면에 비교적 최신 제품으로 아직 상용 시장에 많이 보급되지 못했고, 따라서 500명 이상의 동시 사용자와 100GByte 이상의 OLTP 상용 시장에서 성능을 입증해야 할 필요성이 남아 있다 [13, 14].

DSA는 효율적인 멀티쓰레드 서버 구조를 가지고 있으며 SMP 플랫폼에서는 최대 16개의 프로세서까지도 지원하는 좋은 확장성을 갖는다. 500명 이상의 사용자를 포용하는 환경에서도 비교적 좋은 가격대 성능비를 보이고 있으며 메모리 자원의 소비도 다른 제품들에 비해 비교적 낮다. 기본적으로는 페이지 단위의 록킹을 지원하고 있으나 테이블 단위로 행단위 록킹도 지원한다. 또한 DSA는 트랜잭션 프로세싱(TP) 모니터의 기본적인 기능들을 포함하고 있어 TP 모니터 사용의 필요성을 줄이면서 분산환경 구성에 장점을 제공한다.

대용량 데이터 관리를 위한 데이터 분할 기법도 다양해 순차적 분할, 해성 분할, 영역 분할, 그리고 스키마 분할 기법 등을 제공한다. 성능 조율을 위한 다양한 변수들의 많은 수가 동적으로 온라인 상에서 조절되고 버전 7.1에서는 전체 데이터베이스의 복사본을 원격 서버에 두고 읽기 전용으로 활용할 수 있는 기능도 가지고 있다. 이 경우 로그 정보는 동기적 또는 비동기적으로 전송될 수 있고 주 서버의 실패시 자동적으로 콘트롤이 원격 서버로 넘어간다. '95년도에 발표될 버전 8.0에서는 소결합 클러스터 시스템과 대규모 병렬 시스템들을 지원할 계획으로 있다.

#### 4.2 Oracle사의 Oracle Parallel Server (OPS)

Oracle은 SMP급 기종들에 DBMS 솔루션을 맨 처음 제공함으로써 메인프레임을 대체할 수 있는 시스템 시장을 형성하였고, 많은 수의 다양한 플랫폼들에 장착되어 사용되면서 확장성과 고가용성을 입증하였다[15, 16].

SMP급 플랫폼에서 16개 프로세서까지의 확장은 실제 사용 환경에서 증명되었으나 Oracle에서 사용하는 프로세스 기반의 구조는 경쟁 상대인 쓰레드 기반의 다중 쓰레드 구조에 비해 메모리를 포함한 컴퓨팅 자원의 소비가 상대적으로 크다. 행단위 록킹과 같은 가장 다양한 록킹 레벨의 지원과 멀티 버전 동시처리 제어 기능 등은 타사 제품에 비해 큰 장점이다. 하지만 많은 수의 동시 사용자들을 위한 확장의 제한은 구조적 확장성이나 동시처리 제어 기능보다는 메모리 또는 프로세싱 파워와 같은 자원들의 소비가 큰데서 오고 있다. TP 모니터를 사용하여 분산환경을 구성하면 효과적으로 많은 수의 동시 사용자들을 지원할 수 있다.

가용성 측면에서의 Oracle은 비록 Tandem의 NonStop SQL 수준은 아니지만 클러스터 구조를 갖는 시스템에서 OPS를 이용하여 타사에 비해 상대적으로 높은 가용도를 지원하고 있다. 공유디스크 구조를 지원하는 이 구성은 한쪽의 실패시에도 다른 한쪽은 계속 작동하는 상시 대기 형태의 높은 가용도를 제공한다. 온라인 백업과 복구 기능도 지원하고 있으나 온라인 구성 변경은 제한적으로 이루어지고 있다.

Oracle의 가장 큰 장점은 여러 다양한 플랫폼들에 이미 장착되어 성능, 동시사용자 수, 데이터베이스 처리 용량, 그리고 고가용도 등의 항목들이 상용 시장에서 인정받고 있다는 점이다.

#### 4.3 Sybase사의 Navigation Server(NS)

Sybase는 소형과 중형 클라이언트/서버 응용분야에서 가장 먼저 두각을 나타낸 회사이고 현재 유닉스 기반 RDBMS 시장에서 Oracle에 이어 2위를 차지하고 있다. Sybase 제품에는 SQL Server 10이라 불리우는 SMP용 제품이 있고 이들을 여러개 묶어 사용하는 Navigation Server라고 하는 병렬 플랫폼용 제품이 있는데 현재 NCR 3600에서 동작하고 있다. 이외에 고가용도를 위한 Replication Server와 보안 기능을 B1 등급까지 올릴 수 있는 Secure SQL Server 등이 있다[17, 18]. 아래의 분석은 여러 서버에 공통적으로 포함되며 기본적인 핵심 기능들을 가지고 있는 SQL Server 10에 대

한 것이다.

SQL Server 10은 4개 미만의 프로세서를 갖는 SMP급 시스템에서는 좋은 성능을 내고 있으나 서버 구조의 불균형과 디스크 입출력의 병목현상으로 6개 프로세서 이상의 시스템에서는 DBMS 엔진의 확장성에 문제를 보이고 있다. 차후 버전에서는 이러한 확장성 문제들이 많이 해결되리라 본다. 기본적으로 페이지 단위의 록킹을 지원하고 행단위 록킹은 지원하지 않고 있어 자주 쓰이는 색인의 변경 기능이나 일관성 유지 기능들의 수행시에 경합의 원인이 되기도 한다. 기본적으로는 쓰레드 기반의 다중 쓰레드 구조를 가지고 있으며 특히 메모리 소비율은 무척 좋은 편으로 하나의 서버로 최고 약 800명의 동시 사용자를 지원할 수 있는 것으로 알려져 있다.

SQL Server 10은 큰 용량의 OLTP 데이터 베이스를 분할하고 관리하는데 상대적으로 약간 미흡하다. NS는 보다 나은 기능들을 제공하고 있으나 주로 DSS 처리에 적용되고 있다. 가용도를 높이기 위해선 Replication Server를 사용하여 주서버의 실패시 예비 서버로 바로 연결할 수 있다.

### 5. 맺음말

중대형 컴퓨터 시스템 아키텍처의 발전 동향을 살펴 보면, 가격대 성능비에서 SMP 구조를 갖는 시스템들이 우위를 점하고 있으나 그들의 단점인 확장성을 보완하기 위하여 몇 개의 SMP 시스템들을 엮는 클러스터링 구조로 나아가고 있다. 이러한 클러스터링 구조는 일반적으로 4개 미만의 SMP 시스템들을 엮는 것으로 근본적인 확장성에 역시 제한이 있으나 기존의 응용 프로그램들을 수용할 수 있는 장점과 UNIX의 보편화, 분산처리 기술의 발전, 그리고 클라이언트/서버 추세에 맞추어 이러한 동향은 당분간 계속될 것으로 예상된다.

MPP 시스템들의 상용시장 진출은 세가지 방향에서 진행되고 있는데, 클러스터 구조를 갖는 SMP 구조의 확장이 그 하나이고, 두번째는 전통적 MPP 특성을 갖는 구조에 프로그래밍을 쉽게하기 위한 분산 공유 메모리를 제공하

는 방향이고, 마지막방향은 첫번째와 두번째의 중간 형태를 취하는 것으로 SMP 구조의 노드들을 고유의 고성능 상호 연결망으로 묶어 하나의 시스템 이미지를 제공하는 것이다. 첫번째 방향을 취하고 있는 시스템들로는 Sequent와 Digital VMS등과 같이 기존의 SMP 구조의 중대형 시스템 개발 업체들의 시스템들이 여기에 속하고, 두번째 방향을 취하고 있는 시스템들로는 Kendall Square Research의 KSR1, Cray Research의 T3D, 그리고 Convex의 Exemplar SPP등을 들 수 있으며, 세번째 방향을 취하고 있는 시스템들은 IBM의 SP2, 그리고 현재 개발중인 AT&T GIS의 NCR 3700과 한국전자통신연구소의 SPAX 등을 들 수 있다.

이러한 시스템들은 UNIX 기반 위에 독자적인 소프트웨어 개발 업체들의 병렬 RDBMS 패키지로 솔루션을 제공하고 있다는 공통점을 갖고 있다. 따라서 본 고에서는 병렬 RDBMS가 갖추어야 할 주요 요구사항인 높은 가용도, 확장성, 대용량 데이터 및 자원 관리능력, 온라인 복합 처리 기능 등이 어떠한 의미를 갖는지를 알아보았다. 또한 여러 병렬 DBMS들의 기본적인 아키텍처들을 분석하여 구조적 특성과 장단점들을 비교하였으며, Informix, Oracle, Sybase사와 같은 독자적인 소프트웨어 개발 업체들의 병렬 RDBMS 패키지들을 분석해 보았다.

Informix의 DSA는 상대적으로 가장 최근에 발표된 제품임에도 불구하고 주목을 받고 있다. DSA는 데이터 및 시스템 관리를 위한 다양한 온라인 관리 및 조율 기능들을 제공하고 있으며 효율적인 다중 쓰레드 구조의 DBMS 엔진을 가지고 있다. 또한 가격대 성능비에서 타사에 비해 유리하고 경쟁력이 있으나, OLTP를 위한 데이터의 최대 크기 및 최대 동시 사용자 수 등의 확장성을 실제 사용환경에서 검증 받아야 하는 과제를 안고 있다.

Oracle OPS의 가장 큰 장점은 여러 다양한 플랫폼들에 이미 장착되어 성능, 동시 사용자 수, 데이터베이스 처리 용량, 그리고 고가용도 등의 측면에서 사용자들에게 인정받고 있다는 점이다. 그러나 동시 사용자의 수를 늘리기 위해선 트랜잭션 프로세싱 모니터를 필요로 하고,



DBMS 엔진이 멀티 프로세서에 기반을 둔 구조로 메모리와 컴퓨팅 파워 등 자원의 소비가 많으며 온라인 관리 기능이 비교적 부족한 단점도 갖고 있다.

Sybase의 SQL Server는 6개 미만의 프로세서를 장착한 중소형급 시스템에서는 성능과 자원 사용의 효율성이 매우 좋다. 그러나 이미 언급하였듯이 현재로서는 6개 이상의 프로세서들을 수용하기에는 DBMS 엔진의 확장성에 문제가 있으며, 50GByte를 넘는 대용량 데이터베이스를 처리하기에는 관리 기능이 약하다. NS는 클러스터와 MPP 시스템들을 지원하기 위한 제품으로 OLTP 보다는 DSS에서 보다 효율적으로 운영되는 장점을 갖고 있다.

현재로서는 상용분야로 진출하고 있는 병렬처리 시스템들은 아직 초기 단계로 기술 지원 및 서비스 그리고 기술의 성숙도에서 불안한 면이 없지 않다. 그러나 이 분야의 급속한 기술 개발로 인하여 상용분야에서의 정보처리를 위한 병렬처리 기술들은 '96년까지는 완숙기에 접어들 것으로 예상되고 있다.

## 참 고 문 헌

- [1] 김양우의 2인, "상용 대규모 병렬처리 컴퓨터의 시스템 아키텍처 연구동향(I), (II)," 주간기술동향, Vol. 640 & 641, 한국전자통신연구소, 1994. 3.
- [2] Y. W. Kim, S. W. Oh, J. W. Park, "Design Issues and System Architecture of TICOM-IV, A Highly Parallel Commercial Computer," The 3rd Euromicro Workshop on Parallel & Distributed Processing, San Remo, Italy, pp. 219-226, Jan. 1995.
- [3] 이준석, 김양우, 박진원, 임기욱, "21세기 정보통신망서비스를 위한 고성능 컴퓨팅 서버 동향 분석," 주간기술동향, Vol. 700, 한국전자통신연구소, 1995. 6.
- [4] 이준석, 김양우, 박진원, 임기욱, "SMP 기반의 클러스터링 컴퓨터 시스템 동향 분석 (I), (II)," 주간기술동향, Vol. 695, 696, 한국전자통신연구소, 1995. 6.
- [5] Dennis Byron, "Issues in On-Line Transaction Processing," Midrange Systems, Datapro Corp., Apr. 1992.
- [6] K. K. Ramakrishnan, P. Biswas, R. Karledla, "Analysis of File I/O Traces in Commercial Computing Environments," Performance Evaluation Review, Vol. 20, No. 1, pp. 78-90, June 1992.
- [7] H. Pirahesh et al., "Parallelism in Relational Data Base Systems: Architectural Issues and Design Approaches," Proc. 2nd IEEE Int. Symp. on Databases in Parallel and Distributed Systems, pp 4-29, 1990.
- [8] E. Rahm, "Parallel Query Processing in Shared Disk Database Systems," SIGMOD RECORD, Vol. 22, No. 4, pp 32-37, Dec. 1993.
- [9] S. S. Thakkar, M. Sweiger, "Performance of an OLTP Application on Symmetry Multiprocessor System," Proceedings of Seventeenth Annual International Symposium on Computer Architecture, Seattle, Wash., pp. 228-238, May 1990.
- [10] P. Valduriez, "Parallel Database Systems: the case for shared something," Proc. 9th Conf. on Data Engineering, pp. 460-465, 1993.
- [11] David DeWitt and Jim Gray, "Parallel Database Systems: The Future of High Performance Database Systems," Communications of the ACM, Vol. 35, No. 6, pp. 85-98, June 1992.
- [12] Caroline Long, "Competitive Outlook: Three Approaches to Parallel RDBMS Technology," Midrange Systems, Datapro Corp., May 1994.
- [13] J. Radcliffe, "Informix's DSA: Starting to Prove Itself in the Field," Gartner Group, Research Note, P-600-029, Mar. 22, 1995.
- [14] Caroline Long, "Informix Software, Inc. INFORMIX-Online Dynamic Server," DATAPRO, UNIX & Open System Service, IU20-483-101, 1994. 7.
- [15] J. Radcliffe, "One of the 'Alternative Mainframe' Leaders," Gartner Group, Research Note, P-500-003, Dec. 21, 1994.
- [16] Caroline Long, "Oracle Corp. ORACLE," DATAPRO, UNIX & Open System Ser-

vice, IU20-685-101, 1994. 6.

- [17] J. Radcliffe, "Sybase SQL Server 10; Aiming High, But Lacking Scalability," Gartner Group, Research Note, P-600-036, Apr. 14, 1995.
- [18] Caroline Long, "Sybase, Inc. SYBASE System 10," DATAPRO, UNIX & Open System Service, IU20-861-101, 1994. 7.
- [19] Gary Herman, "Parallel Processing Database Servers," Midrange Systems, Datapro Corp., Apr. 1993.

김 양 우



- 1984 연세대학교 전자공학과, 학사
- 1986 미국 시라큐스 대학 컴퓨터공학, 석사
- 1992 미국 시라큐스 대학 컴퓨터공학, 박사
- 1988~1992 미국 시라큐스 대학 Teaching 및 Research Assistant
- 1992~현재 한국전자통신연구소 컴퓨터연구단 선임연구원

관심분야: 특수전용 컴퓨터시스템 아키텍처(병렬처리, 데이터베이스, 정보검색, 멀티미디어, 오류허용, 평전자 정보처리), 시스템 분석

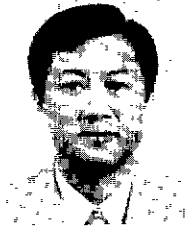
박 진 원



- 1975 서울대학교 산업공학과, 학사
- 1982 미국 오하이오 주립대 산업공학과, 석사
- 1987 미국 오하이오 주립대 산업공학과, 박사
- 1977~1980 한국개발연구원 (KDI) 연구원
- 1987~1988 미국 남콜로라도대학 조교수
- 1988~현재 한국전자통신연구소 컴퓨터연구단 시스템공학연구실장

관심분야: 컴퓨터시스템 아키텍처 설계, 모델링 및 성능 평가, 시스템 분석 및 시뮬레이션

임 기 욱



- 인하대학교 전자공학과, 학사
- 한양대학교 전자계산학과 석사
- 인하대학교 전자계산학과, 박사
- 1977~1988 한국전자통신연구소 시스템소프트웨어연구실장
- 1988~1989 캘리포니아 주립대 (U.C. Irvine) 방문연구원
- 1989~현재 동연연구소 컴퓨터연구단 시스템연구부장

관심분야: 운영체제, 데이터베이스 시스템, 시스템 구조