

슈퍼컴퓨팅 기술의 활용 3

무결성이 깨어진 빅 데이터 시대

남범석 교수 (UNIST 전기전자컴퓨터공학부)

위키백과(Wikipedia)에 의하면 빅데이터는 “기존 데이터베이스 관리도구(DBMS)의 역량을 넘어서는 데이터로부터 가치를 추출하고 결과를 분석하는 기술”이라고 정의되어 있다. 데이터베이스 관리도구는 1970년 IBM에서 개발된 이후 40년동안 대용량 데이터를 빠르게 처리할 수 있도록 진화되어 왔고 지금도 그 성능이 개선되고 있다. 그런데 최근 몇 년 전부터 오히려 데이터베이스를 사용하면 대용량 데이터를 처리하지 못한다고 주장하면서 데이터베이스 대신 새로운 시스템을 개발하자는 움직임이 NoSQL 혹은 빅데이터(Big Data)라는 이름으로 불리며 화두가 되고 있다. 도대체 데이터베이스의 어떠한 문제점을 발견하였길래 대용량 데이터를 위해 40년간 진화해 온 데이터베이스를 대체할 시스템을 개발하자는 움직임이 생겼을까?

데이터베이스 관리도구인 DBMS의 소비자는 일반 대중이 아닌 기업이었다. 90년대에 들어서야 대중들이 컴퓨터와 인터넷을 집에서 사용하게 되었으니, 70년대부터 연구 개발되어 온 데이터베이스 연구 개발은 자연스럽게 기업의 대용량 비즈니스 데이터를 처리하기 위해 최적화되어 왔다. 비즈니스 데이터를 처리하는 데이터베이스 관리도구는 흔히 ACID(Atomicity, Consistency, Isolation, Durability) 라고 불리는 데이터 무결성을 최우선 원칙으로 삼고 있다.

예를 들면 유니스트 학생이 전공을 전기 전자 트랙에서 컴퓨터 공학 트랙으로 변경하는 과정 중에 전기전자 컴퓨터 공학부의 총 학생수는 일정하게 유지되어야 한다는 것이다. 만일 그 학생을 전기전자 트랙 명단에서 먼저 삭제한 후 컴퓨터 공학 트랙으로 이제 막 등록하려는 순간에 누군가 전기전자 컴퓨터 공학부의 총 학생수를 확인하려 한다면 학생 수가 한 명이 줄어든 것으로 보일 수 있다. 데이터베이스는 이러한 순간적인 오류를 허용하면 안된다. 데이터베이스 관리 도구에 논리적인 오류가 있다는 것은 결국 그 데이터베이스 관리 도구를 사용하는 기업의 경제적인 손실로 이어질 수 있기 때문

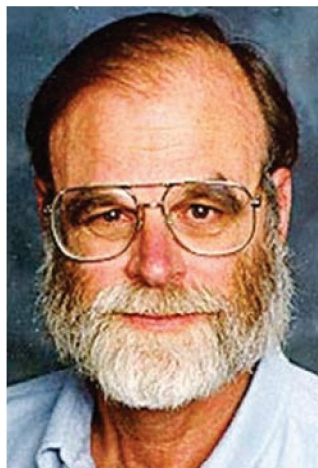


사진 설명 : 데이터베이스의 ACID 원칙을 정의한 Jim Gray. 그는 ACID 원칙 등 DBMS의 트랜잭션 프로세싱에 관한 많은 중요한 연구로 1998년 Turing 상을 수상하였다. 그러나 2007년 1월 샌프란시스코 근처에서 요트 항해 중 실종되었는데, 만일 그가 실종되지 않았더라면 현재의 NoSQL 연구 방향에 어떠한 영향을 주었는지 매우 궁금하다. (사진 출처 : Wikipedia)



이다. 40년동안 대용량 데이터를 처리하는데 있어 ACID 원칙은 구글(Google)이라는 거대 인터넷 기업이 탄생하기 까지 어느 누구도 의문을 가지지 않은 절대 원칙으로 자리 잡게 되었다.

1990년대말 인터넷의 폭발적인 성장은 Yahoo!와 같은 기존의 인터넷 검색 엔진 기업들이 감당할 수 없는 수준에 이르렀고, 검색 결과가 사용자들이 원하는 수준에 이르지 못하자 이들 기업은 검색 인덱스를 주제별로 나누어 사용자들이 주제별 목록(Directory)을 계층구조로 따라 내려가 선택하도록 하는 방식으로 서비스하는 선택을 하였다. 반면 구글(Google)은 페이지순위(PageRank)라는 새로운 검색 기법을 사용하여 단 하나의 텍스트박스만으로 검색 인터페이스를 단순화하는 동시에 사용자들의 검색 결과 기대치를 만족시키며 거대 인터넷 검색 엔진 기업으로 성장하였다. 구글은 자사의 서비스를 위해 개발한 기술들을 매년 컴퓨터 공학 학회에서 조금씩 공개하고 있는데, 구글에서 발표하는 논문들은 컴퓨터 공학의 연구 방향을 바꾸어놓을 만큼 큰 영향력을 보여주고 있다. 2004년 OSDI 학회에서 발표한 논문에서 구글은 자사의 대용량 데이터 처리를 위해 개발한 맵리듀스(MapReduce)라는 분산 병렬 프로그래밍 모델을 공개하였다. 맵리듀스(MapReduce)는 미국 메릴랜드 대학교에서 90년대 후반에 개발한 ADR (Active Data Repository)이라는 분산 병렬 프로그래밍 모델을 사용하고 있는데, 메릴랜드 대학교에서 개발된 ADR과는 달리 맵리듀스 프로그래밍 모델은 구글이라는 거대 인터넷 기업의 이름을 등에 업고 컴퓨터 업계의 지대한 관심을 받게 된다.

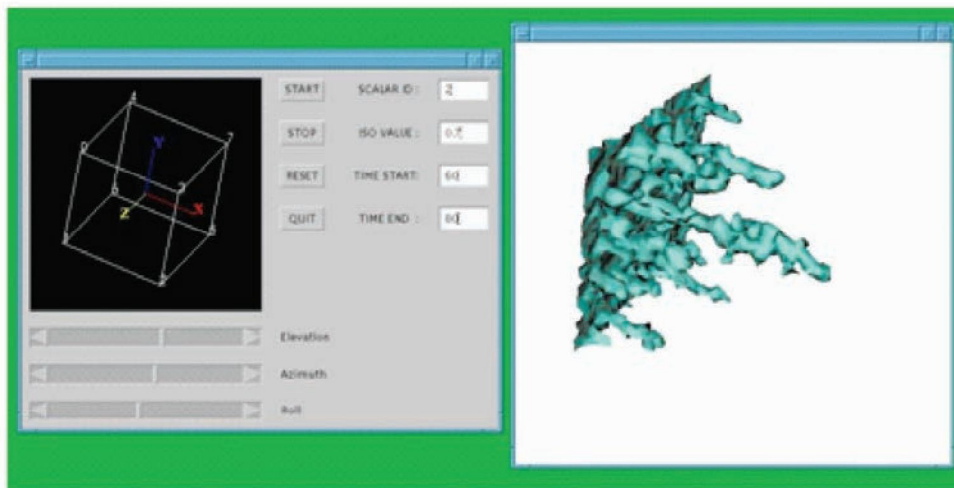


사진 설명 : 메릴랜드 대학교에서 개발한 Active Data Repository를 사용하여 석유 oil reservoir의 iso-surface를 동적으로 가시화하는 프로그램. 석유, 환경, 채광 등 지표하의 방대한 데이터를 분산 저장하고 병렬 처리하여 빠르게 가시화한다.

맵리듀스 혹은 ADR은 구글의 첫 페이지와 같이 매우 쉽고 단순하지만 폭발적으로 성장한 인터넷의 대용량 데이터를 처리하기에 적합한 컴퓨팅 모델이었다. 데이터는 많고 컴퓨터의 처리 속도는 한정되어 있으니, 데이터를 작게 나누어서 여러개의 컴퓨터에서 동시에 처리한 뒤 결과값을 합치는 구조이다. 맵리듀스는 관계형 데이터베이스의 SQL과 사용법의 용이함에서 공통점을 찾을 수 있다. 맵리듀스는 프로그래머가 데이터가 어떻게 분산되어 저장되었고, 어느 디렉토리를 접근하여 읽을지 등 세세한 부분을 고려할 필요가 없도록 한다. 단지 분할 정복 기법의 알

고리즘을 Map 함수와 Reduce 함수만을 사용하여 작성함으로써 확장성이 높은 분산 병렬 프로그램을 완성하도록 하여준다. 물론 맵리듀스 프로그래밍 모델에서 데이터베이스에서 이야기하는 ACID 원칙은 고려도 하지 않았다. 어떻게 하면 컴퓨팅 연산을 단순하게 만들고 확장성을 높여서 더 많은 데이터를 한 번에 처리할 수 있을지에 모든 초점이 맞추어져 있다. ACID 원칙은 구글 검색 엔진에는 전혀 필요가 없는 거추장스러운 원칙일 뿐이다. 물론 검색된 데이터를 DBMS에 저장하여 서비스하는 검색 엔진(Enterprise Search)을 개발하여 팔고 있는 오라클이라는 회사도 있긴 하나, 그 검색 대상이 인터넷 스케일의 대용량 데이터는 아닌 것으로 판단된다. 맵리듀스는 검색 로봇이 인터넷에서 긁어 모은 웹페이지들을 분석해서 어떠한 웹페이지가 더 중요한지, 어떠한 웹페이지가 어떠한 내용을 담고 있는지를 파악하는데 사용되었는데, 웹페이지는 비즈니스 데이터와는 달라서 ACID에서 요구하는 데이터 무결성이 전혀 중요하지 않다. 페이지순위(PageRank) 를 정하는데 있어 무결성이 깨진 웹페이지의 분석은 검색 결과에 아주 미미한 영향만을 줄 뿐이다.

사람들은 구글이라는 거대 인터넷 검색 엔진 업체가 처리하고 있을 엄청난 데이터의 양에 관심을 가졌다. 맵리듀스가 처리할 수 있는 데이터가 웹페이지가 아닌 다른 데이터들이어도 될 것으로 판단한 것이다. 구글과 경쟁관계에 있지만 망해가는 Yahoo!는 하둡(Hadoop)이라는 이름으로 MapReduce 프로그래밍 모델을 지원하는 공개 소스코드를 개발하였다. 하둡은 빠르게 여러 분야에 적용이 되었고, 야후!는 하둡을 사용하여 슈퍼컴퓨터나 계산할 수 있을 법한 π (π) 의 2,000,000,000,000,000 번째 숫자를 찾아내는 성과를 올리기도 하였다. 하둡 역시 맵리듀스와 같이 ACID 원칙을 지키지 않고 있기에 비즈니스 데이터를 저장하여 관리하는 용도로는 적합하지 않았다. 대신 무결성과는 상관없는 단발성 대용량 데이터 분석 - 예를 들면 로그 분석을 위한 용도로 많이 사용되어졌다. 하둡의 확장성과 높은 성능이 아마존(Amazon)의 클라우드 컴퓨팅 플랫폼 EC2(Elastic Compute Cloud)서비스와 함께 점점 인기를 끌게 되었는데, 하둡에 관심을 갖는 사용자 수가 많아지면서 동시에 맵리듀스의 단순한 프로그래밍 모델이 아닌 데이터베이스 수준의 더 많은 기능을 갖추면서 동시에 높은 고성능과 확장성을 갖는 새로운 시스템에 대한 요구가 생겨나게 되었다. 구글은 2006년 빅테이블(Big Table)이라는 데이터베이스와 같은 정형화된 데이터를 저장

할 수 있는 시스템을 개발하여 OSDI 학회에 발표하였고, 구글과 함께 최근 클라우드 컴퓨팅을 비롯하여 IT 업계의 트렌드를 만들어 가는 아마존(Amazon)은 자사의 인터넷 쇼핑몰 데이터를 처리하기 위한 다이نام오(Dynamo)라는 P2P 분산 저장 시스템을 개발하여 2007년에 발표하였다. 야후! 역시 2008년 자사의 웹서비스를 위한 PNUTS 라는 데이터 처리 미들웨어를 개발하였고, 페이스북 역시 뒤질세라 2009년과 2010년 Hive(HiveQL), Cassandra 시스템을 각각 발표하였다.

대용량 데이터는 모두 DBMS에 저장하고 처리해야하는 줄 알았던 시대가 가고, 기존의 DBMS를 대체하는 새로운 미들웨어(Middleware)들이 NoSQL 이라는 통칭으로 불리며 속속 등장하고 있다. NoSQL이라는 용어는 데이터베이스에 대적하는 의미를 갖



사진 설명 : NoSQL 은 데이터베이스에 대적하는 의미로 써의 No! SQL 이라는 뜻이 아니다. 그러나 MIT의 마이클 스톤브레이크 교수는 NoSQL이라는 이름이 주는 부정적인 이미지에 불만을 표현한 적이 있다. (사진 출처 : Wikipedia)

는 것처럼 보이나 그 원 뜻은 Not-Only-SQL 로써, SQL을 보완한다는 의미를 갖는다. 데이터의 무결성이 반드시 필요한 비즈니스 응용 프로그램들에는 데이터베이스를 사용하는 것 이외에는 대안이 있을 수 없으나, 그 외의 대용량 데이터를 처리하는데에는 DBMS의 ACID는 너무 가혹한 요구조건이므로 데이터의 일관성(Consistency)를 포기하는 대신 기존의 DBMS들의 단점인 낮은 확장성을 보완하여 성능을 개선하는 선택을 하는 미들웨어들이 속속 개발되어지고 있다. 최근 개발되고 있는 NoSQL미들웨어들은 DBMS의 SQL의 편리함마저 가져와 NoSQL 미들웨어 위에 개발자들이 쉽게 응용 프로그램을 작성할 수 있도록 SQL 과 비슷한 선언적 프로그래밍 언어 (Declarative Programming Language)도 개발하고 있다.

40년간 데이터베이스 성공시대를 구가하였던 컴퓨터 과학자들 중 일부는 이런 급격한 변화가 불편한 듯 하다. UC 버클리에서 30년간 데이터베이스 연구의 선두에 서있었던 Michael Stonebraker (현 MIT 교수는 “NoSQL 진영은 기존의 병렬 데이터베이스 연구부터 다시 공부하고 오라” 며 그 불편한 심경을 토로한 바 있다. NoSQL 진영에서 이야기하는 확장성은 이미 분산 병렬 데이터베이스 연구자들이 오래전부터 연구하였던 주제인데 NoSQL 진영이 마치 새로운 주제인양 처음부터 다시 반복하고 있다는 불만을 토로한 것이다. 그는 2009년 SIGMOD 학회에서 하둡과 병렬 데이터베이스의 성능을 비교한 결과를 발표하며 병렬 데이터베이스의 성능이 하둡보다 월등히 빠르다고 주장한다. 그러나 그의 실험(을 포함한 대부분의 데이터베이스 옹호자들의 실험)은 소수의 서버를 사용한 것으로(SIGMOD 논문에서는 100대만을 사용하였다.) 사용한 것으로 NoSQL 진영에서 말하는 수천 수만 대의 서버를 돌리며 수십억 사용자들의 작업을 수행하는 환경과는 거리가 있다. ACID 무결성을 지키는 DBMS는 한 명의 사용자 작업을 위해 수 많은 사용자의 작업들을 정지시키는 일이 발생하곤 하는데, 구글, 야후, 아마존, 페이스북과 같은 웹서비스에서는 있을 수 없는 일이다. 그는 후에 한 발 물러서서 NoSQL은 단발성의 데이터 분석(ETL: Extract-Transform-Load)에 적합하지만 데이터 무결성을 요구하는 응용 프로그램들을 위해서는 DBMS 도구를 사용해야만 한다며, NoSQL과 데이터베이스는 서로 적어 아니라 공존해야 하는 친구라고 NoSQL 움직임일 일부 인정하는 자세를 취하였다. 또한 무결성을 보장하면서 동시에 NoSQL과 같은 뛰어난 확장성을 가지는 DBMS를 새로 연구개발할 것을 데이터베이스 커뮤니티에 주문하기도 하였다.



사진 설명: 빅데이터NoSQL 진영은 ACID 원칙 중 강한 일관성 (Strong Consistency)을 포기하고 최종 일관성 (Eventual Consistency) 만을 지원함으로써 빅데이터 처리를 위한 확장성을 높이는 선택을 한다. (사진 출처 : Sesame Street)

확장성과 무결성은 공존할 수 있는 것일까? 데이터베이스 이론 중에 CAP 정리라는 것이 있다. CAP, 즉 일관성 (Consistency), 가용성(Availability), 분할내성(Partition-tolerance) 이 세가지는 동시에 절대 만족할 수 없다는 것이 증명되어 있다. Michael Stonebraker는 컴퓨터들의 고장은 천재지변으로도 발생할 수 있듯 불가항력이므로 CAP 중에 한 가지를 포기하려면 분할내성(Partition-tolerance)를 포기하고 일관성, 즉 무결성은 포기하면 안된다고 말한다. 반면 NoSQL 진영은 그들의 웹서비스 특징을 고려하여 무결성을 포기하고 확장성, 가용성을 선택하는 것이 맞다고 주장한다. 예를 들어 페이스북은 데이터에 접근하는 속도를 무결성보다 더 중요시 하여, 데이터를 여러 서버에 복



사진 설명 : 영화 매트릭스에 나오는 Deja-Vu 고양이, 매트릭스 세상의 데이터를 처리하는 시스템도 분명 NoSQL 처럼 무결성보다는 확장성에 초점을 맞춘 듯 하다. 데자뷰 고양이는 ACID 원칙을 만족하는 기존의 데이터베이스 시스템에서는 발생할 수 없다.

제(Replication)하여 관리한다. 복제본을 많이 만들면 사용자들의 수 많은 요구 작업이 여러 복제본에 분산되므로 개별 서버의 로드가 줄고 작업을 처리하는 속도가 빨라진다. 대신 복제본이 여러 곳에 존재하므로 사용자가 업데이트하는 내용이 다른 복제본에까지 반영이 되려면 시간이 오래 걸리는데, 그 동안 복제본들이 서로 다른 내용을 갖고 있게 되어 어떤 페이스북 친구들은 업데이트된 내용을 보는 반면 다른 친구들은 업데이트된 내용을 보지 못하고 있는 상황이 발생할 수 있다. 페이스북과 같은 소셜네트워크 사이트는 내가 오늘 저녁에 페이스북에 올리는 사진을 친구들이 즉시 볼 수 있게 하는 것보다는 사진이 페이스북에 올라가는 속도를 더 중요시 하는 것이다. 또한 데이터의 복제는 복제본들을 가지고 있는 모든 서버가 죽지 않는 이상 데이터를 필요로 하는 사용자들에게 서비스를 수행할 수 있어, 서비스 장애 시간이 줄어들고 서비스업체

의 매출이 증가하게 된다. 우리나라의 일부 인터넷 서비스들은 매주 특정 요일 새벽만 되면 서비스 정기 점검 메시지를 당연한 듯 띄워놓는데, 이는 국내 소프트웨어 업계의 뒤쳐진 실력을 그대로 보여주는 사례라고 할 수 있겠다.

데이터의 무결성은 빅데이터를 처리하는데 있어 걸림돌이기만 한 것일까? 그 답은 응용 분야에 따라 다르다. 일반인들을 대상으로 하는 웹서비스 혹은 단발성 빅데이터 처리에는 무결성이 필요하지 않은 경우가 많지만 데이터의 무결성을 요구하는 경우도 없지는 않다. 예를 들어 페이스북에 가족들이 보면 안되는 사진을 올려 친구들에게 자랑을 하고 싶은 경우를 생각해보자. 우선 가족들이 내 페이스북을 볼 수 없도록 권한을 막고 난 후에 사진을 올리겠지만, 무결성을 보장하지 않는 NoSQL 미들웨어들은 접근 권한을 막는 작업과 사진을 올리는 작업, 이렇게 두 개의 작업을 서로 독립적으로 처리하게 되며 그 작업 순서가 지켜지지 않아 가족들이 보면 안되는 사진을 보는 경우가 발생할 수 있다. 기업의 비즈니스 데이터와는 좀 다른 성격이라 그 영향이 크게 문제가 되지는 않겠으나, NoSQL 관련 연구가 발전하면서 최근에는 이러한 문제점들도 어느정도 해결하고자 하는 움직임이 생기고 있다. 야후에서 개발한 PNUTS는 야후 메시지 브로커(Yahoo Message Broker)라는 메시지 큐잉 시스템(Message Queuing System)을 사용하여 한 사용자의 작업은 그 순서를 지켜서 데이터에 반영이 되도록 하고 있으며, 구글에서 2011년에 발표한 Megastore 역시 이러한 무결성을 데이터 엔터티 그룹(Entity Group, DBMS의 테이블과 거의 비슷한 개념) 단위로 해결하고 있다. 엔터티 그룹이라고 불리는 데이터의 집합 내부에서는 기존의 데이터베이스에서 제공하는 것과 동일한 ACID 원칙을 만족시키나, 여러 엔터티 그룹에 접근하는 사용자 작업에 대해서는 다른 NoSQL 미들웨어들과 마찬가지로 약한 일관성(Weak Consistency)만을 지원한다. 이러한 설계 바탕에는 여러 엔터티그룹을 동시에 접근하는 사용자의 작업은 그리 많지 않다는 전제가 깔려 있다. 수많은 사용자들에게 서비스를 제공하고 있는 구글에서만 얻을 수 있는 노하우를 기반으로 한 결정이 아닐까 생각된다.

NoSQL은 데이터베이스에서 지켜왔던 무결성 원칙을 버리는 전략을 통해 빅데이터 시대의 대용량 데이터를 처

리 가능하도록 만들었다. 그 놀라운 처리 성능으로 인해 응용 분야가 넓어지는 과정에서 자연스럽게 다시 데이터의 무결성이 문제로써 부각되면서 확장성과 무결성 사이에 절충안을 찾고자 하는 노력이 새로 시작되고 있다. 조만간 머지 않아 데이터베이스와 빅데이터 사이에서 무결성도 지원하고 확장성도 높은 BigSQL이라는 새로운 기술이 탄생하게 되지 않을까 기대해본다.