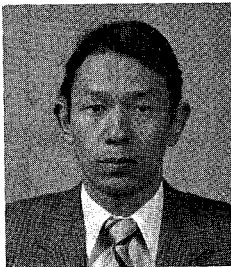


分散 Database의 開發 背景과 推移



文 松 天

韓國科學技術院 電算學科 教授 / 工博

分散 데이터 베이스에 관한 문제들을 해결하기 위한 기술들은 단지 전통적인 데이터 베이스를 위한 기술을 확장함으로써 가능한 것은 아니다. 문제에 따라서는 전혀 다른 방식의 해결책을 필요로 하는 것도 있으며 전통적인 데이터 베이스에서는 찾아볼 수 없는 새로운 문제를 해결해야만 하는 경우도 있다.

1. 分散 Database의 開發 背景

최근 들어서 분산 데이터 베이스 (Distributed Database)는 정보처리에 있어서 중요한 분야로 등장하였으며 이러한 중요성은 앞으로 더욱 증대되어질 것으로 예측된다. 이러한 예측에는 구조적이고도 기술적인 이유가 있다. 즉, 분산 데이터 베이스는 中央集中型 데이터 베이스 (Centralized Database)의 많은 단점을 제거하였으며 또한 비중앙집중형태의 구조를 가진 조직에게 자연스럽게 적용될 수 있는 장점을 갖고 있기 때문이다.

분산 데이터 베이스는 데이터 베이스들의 집합이며 이러한 각각의 데이터 베이스는 자기 Site에 있는 컴퓨터를 통해 액세스되며 전산망 (Computer network)을 통해 다른 데이터 베이스에 서로 연결되어 전체의 데이터가 논리적으로 같은 시스템에 속한 것을 가리킨다. 이와 같은 정의에서 볼 수 있듯이 분산 데이터 베이스에는 두가지 중요한 면이 있다.

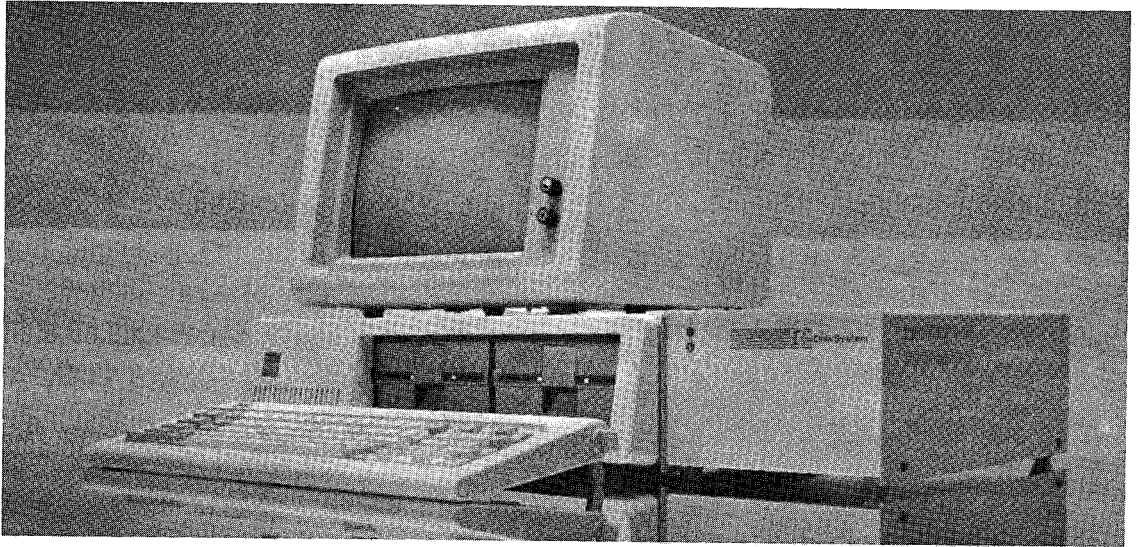
첫째, 분산은 모든 데이터가 하나의 동일한 Site에만 존재하는 것이 아니라는 것이다. 따라서 분산 데이터 베이스는 하나의 Site에만 있는 중앙집중형 데이터 베이스와 구별된다.

둘째, 논리적 연계 즉, 데이터들이 서로 연관되는 성질을 갖고 있다. 이러한 점에서 분산 데이터 베이스는 전산망내의 Site들에 널리 퍼져 있는 로컬 데이터 베이스들을 그대로 모은 것과는 구별된다.

이와 같은 분산 데이터 베이스를 정의하는 경우에 分散과 論理的 連繫라는 두가지 면이 매우 애매한 뜻을 갖고 있으므로 좀 더 자세히 정의하기 위하여 다음의 예를 통해 고찰해 본다.

「3개의 지점을 서로 다른 장소에 갖고 있는 은행을 생각해 보자.

각 지점에는 하나의 컴퓨터가 있는데 이 컴퓨



분산 데이터 베이스는 전산망내의 서로 다른 컴퓨터에 분산되어 있는 데이터들의 집합이다.

터가 그 지점 창구의 터미널들과 計算 데이터 베이스를 관장하고 있다. 한 지점의 컴퓨터와 이에 로컬된 계정 데이터 베이스가 분산 데이터 베이스의 한 Site를 구성하고 있으며 컴퓨터들 끼리는 電算網을 통하여 연결되고 있다. 정상적인 업무처리에 있어서 지점의 터미널로부터 들어오는 작업은 그 지점의 데이터 베이스만을 액세스함으로써 처리된다.

이와 같이 모든 작업은 그 작업이 제기된 지점의 컴퓨터에서 실행되므로 局地的作業(Local application)이라고 부른다. 국지적 작업의 한가지 예로서 작업이 제기된 지점에 보관되어 있는 계정에 대한 入金作業과 出金作業이 있다.」

이러한 예의 경우에 대해 이미 정의한 분산 데이터 베이스의 정의를 적용해 본다면 이 예에서는 논리적 연계라는 면을 찾아볼 수 있는지 판단하기가 힘들게 된다. 따라서 우리는 다음과 같은 질문을 던지게 된다. 각 지점에서 그 지점의 계정에 관한 정보만을 갖고 있으면 충분한가? 이 예를 분산 데이터 베이스로 볼 것인가 아니면 단순한 국지적 데이터 베이스의 집합으로 볼 것인가?

이와 같은 질문에 대답하기 위하여 무엇이 국지적 데이터 베이스의 집합인가, 분산 데이터 베이스에서 데이터가 논리적으로 연계를 갖고 있는 것을 구분할 수 있는가에 대해 초점을 두고

생각해 보자. 기술적인 관점에서 중요한 점은 여러 지점에 있는 데이터들을 액세스하는 작업이 있느냐 하는 것이다. 이러한 작업들을 全地的作業(Gloval application) 혹은 分散作業(Distributed application)이라고 부른다. 이러한 전지적 작업이 존재하느냐 하는 것이 분산 데이터 베이스를 국지적 데이터 베이스와 구별하는 특징이 된다.

전지적 작업의 대표적인 예는 한 지점의 계정에서 돈을 인출하여 다른 지점의 계정으로 입금시키는 것이 있다. 이러한 작업에서는 두개의 지점에 있는 데이터 베이스를 更新하여야 하는 일이 수행된다. 여기서 주의할 점은 이 작업이 단순히 두개의 독립된 지점에서 인출과 입금을 각각 수행하는 것으로 국한되는 것이 아니라는 것인데 왜냐하면 이 두가지 갱신이 모두 수행되든지, 아니면 하나도 수행되지 않든지 해야하기 때문이다.

본 예에서는 컴퓨터들이 지역적으로 서로 다른 장소에 위치했으나 분산 데이터 베이스는 近距離網(Local area network)에서도 구축될 수 있다. 예에서 살펴 본 것을 토대로 하여 분산 데이터 베이스를 자세히 정의하면 다음과 같다.

「분산 데이터 베이스는 전산망내의 서로 다른 컴퓨터에 분산되어 있는 데이터들의 집합이며 전산망내의 각 Site는 자치적으로 처리할 수 있

는 능력을 갖고 있고 국지적 작업을 실행할 수 있다. 각 Site는 또한 적어도 하나 이상의 전지적 작업 즉, 통신을 통하여 여러 Site에 저장된 데이터를 액세스하는 작업을 실행하는데 관여한다.]

2. 分散 Database의 開發 이유

분산 데이터 베이스가 개발되는 이유는 다음과 같이 살펴볼 수 있다.

가. 組織의 構造 및 經濟性

많은 조직이 중앙집중적인 형태보다는 분산화된 형태를 갖고 있기 때문에 분산 데이터 베이스를 도입하는 것은 조직의 구조에 자연적으로 들어맞는다고 볼 수 있다. 분산된 조직의 구조에 관한 문제와 그러한 조직을 위한 情報시스템에 관한 問題는 많은 사람들의 관심이 되어 왔다. 컴퓨터 技術이 최근에 더욱 개발되어 大型의 中央集中型 컴퓨터 센터를 갖는다고 하는 것이 경제적인 각도에서 볼 때 회의적으로 생각되어진다.

여기에서 이러한 것에 관한 논의는 더이상 하지 않겠으나 조직의 구조나 경제적인 이유가 분산 데이터 베이스를 개발하는데 있어서 제일 중요한 이유가 된다고 본다.

나. 既存 데이터 베이스들 相互間의 連結

분산 데이터 베이스는 어떤 조직에 이미 여러 개의 데이터 베이스가 존재하고 있는 상황에서 작업을 全地的으로 처리할 필요가 발생하였을 때 이러한 要求를 자연적으로 만족시켜 줄 수 있는 解決策이다. 이러한 경우에 분산 데이터 베이스가 기존의 局地的 데이터 베이스들을 기반으로 하여 上向式으로 생성된다. 이러한 과정에서는 국지적으로 데이터 베이스의 구조를 어느 정도 변경하는 일이 요구되나 이러한 再構成에 드는 노력은 중앙집중형 데이터 베이스를 완전히 새로이 생성하는데 드는 노력에 비하면 아무것도 아닌 것이다.

다. 漸進的인 擴張

어떤 기관의 조직이 기존의 조직에다 自治的으로 작동되는 조직의 한 단위(예를 들면 신설

된 지점이나 신설된 창고 등)를 새로이 첨가함으로써 점차적으로 확장된다면 분산 데이터 베이스를 도입하여 기존의 組織單位들에게 가장 적은 정도의 영향만을 주면서 그 조직을 점차적으로 성장시키는데에 기여할 수 있다. 中央集中形態의 방식을 사용하면 시스템 초기 설계시에 미리 장래에 있을 확장을 고려하여야 하며 그렇지 않으면 조직을 확장하는 것이 새로운 작업에 뿐만 아니라 기존의 작업에도 중대한 영향을 초래하게 된다. 그런데 장래에 어떻게 확장될 것인가를 미리 予測하는 것은 매우 힘든 일이다.

라. 通信 오버 헤드(Overhead)의 減少

지역적으로 널리 분산된 데이터 베이스에서 많은 작업이 국지적으로 처리가능하면 통신 오버 헤드가 중앙집중형 데이터 베이스보다는 분명히 감소될 것이다. 그러므로 作業의 局地性을 최대화하는 것이 분산 데이터 베이스를 설계하는데 있어서 중요한 目標의 하나이다.

마. 性能 向上

시스템에 자치적으로 작동되는 프로세서가 많이 존재하면 作業處理의 並列性(Parallelism)이 高度化되므로 시스템의 성능이 향상되는 결과를 낳게 된다.

이와 같은 면은 비단 분산 데이터 베이스에서만 볼 수 있는 것이 아니고 일반적으로 多重프로세서(Multiprocessor) 시스템에서도 볼 수 있는 것이다.

바. 信賴度와 可用度

분산 데이터 베이스는 데이터의 重復을 허용함으로써 信賴度(Reliability)와 可用度(Availability)를 높이기 위하여 이용될 수 있다. 한편 이러한 目標을 달성하는 일이 간단하지는 않으며 아직 언급하지 않은 技法을 사용해야 한다. 여러 Site에서 자치적으로 처리할 수 있는 기능 자체만으로 시스템의 높은 신뢰도를 보장할 수는 없으나 그 기능은 시스템의 성능이 저하되더라도 갑자기 저하되지 않고 漸次的으로 低下(Graceful degradation)되도록 保障해 준다.

다시 말하면 분산 데이터 베이스에서는 시스템의 성분이 많기 때문에 중앙집중형 데이터 베이스에서 보다도 더 고장이 잦으나 故障의 波及

효과는 단지 고장난 Site에 저장되어 있는 데이터를 액세스하는 작업에만 국한되는 것이지 전체 시스템의 작동을 중지시키는 데로까지 번지지 않는다는 것이다.

이상에서 살펴본 분산 데이터 베이스에 대한 동기는 새로운 사실이 아니다. 그런데 무슨 이유로 분산 데이터 베이스 시스템 개발을 시작한 것이 얼마되지 않았는가? 이에는 두가지 중요한 이유가 있다. 첫째는 최근에 들어서 소형컴퓨터의 개발이 눈부시게 성장하여 종전에 대형 메인프레임(Mainframe) 컴퓨터에 의해서만 提供되던 機能들이 소형컴퓨터에서도 제공됨으로써 經費의 節減이 가능해졌으며 이들 소형컴퓨터가 분산된 情報시스템을 개발하는 데에 필요한 하드웨어로 사용될 수 있게 되었기 때문이다.

둘째로는 분산 데이터 베이스의 기술이 1970년대에 그 기반을 확고히 다진 電算網의 技術과 데이터 베이스의 기술에 기반을 두고 있기 때문이다. 電算網과 Site마다 있는 局地的DBMS들의 집단위에 분산 데이터 베이스를 구축하는 일은 어렵고 복잡한 일이나 이러한 분산 데이터 베이스의 下部構造를 정하는 성분들이 없이 分散 데이터 베이스를 構築한다는 것은 妥当性이 별로 없다고 볼 수 있다.

3. 分散 Database의 推移

분산 데이터 베이스에 관한 문제들을 해결하기 위한 기술들은 단지 전통적인 데이터 베이스를 위한 기술을 확장함으로써 가능한 것은 아니다. 문제에 따라서는 전혀 다른 방식의 해결책을 필요로 하는 것도 있으며 傳統的인 데이터 베이스에서는 찾아볼 수 없는 새로운 문제를 해결하여야만 하는 경우도 있다.

많은 문제들에 대한 해결책이 마련되어 분산 데이터 베이스 관리체제가 점차 상용화됨에 따라서 분산 데이터 베이스를 구축하거나 사용하는 일이 멀지않은 장래에 많이 일어날 것으로 본다. 현재까지 분산 데이터 베이스 관리체제를 쉽게하는 접근 방법에는 두가지가 있다.

첫째는 기존의 데이터 베이스 관리체제 위에 분산 데이터 베이스 관리체제 부분을 엮어서 설계하는 방법이다. 이 방법은 그간의 문헌상이나 실제 상용화된 시스템에서 채택한 방법이다.

둘째는 분산시스템을 위한 분산운영체제 위에 데이터 베이스 관리체제를 설계하는 방법이다.

두번째 방법은 최근의 시도로서 Locus를 기반으로 한 시스템에서 채택되었다. 앞으로는 두번째 접근 방법을 사용하는 분산 데이터 베이스 관리체제가 많이 설계될 것으로 예측된다.

