대용량 데이터 관리에 필요한 데이터 베이스의 중요 특징

최소이
SungKyunKwan UNIV, Sofware dpt
2014311892
010 - 2879 - 1278
ithdli@naver.com

ABSTRACT

이 논문은 대용량의 정보를 관리하기 위해서 핵심적으로 필요한 기술에 대해서 논하고 이에 따른 DBMS(Data Base Manage System)의 필요성에 대한 내용을 담고 있다. DBMS 의 설계에 있어서 필요한 여러 기능들에 대해서 생각하고 이러한 기능들이 왜 필요한지 기술하였다.

Keywords

데이터의 독립성(data dependency), query language, transaction, concurrency control, recovery.

1. INTRODUCTION

'정보의 바다'라는 용어마저 낯설게 느껴질 정도로 사회에 많은 정보가 유입되기 시작한 지 오랜 시간이 지났다. 따라서 "BIG DATA", "DATA MANAGING"같은 대용량의 데이터를 분석하고 사용하는 기술에 대한 관심이 대두되고 있다.

다양한 데이터를 분석하기 위해서는 데이터를 관리하는데 있어서 발생할 수 있는 여러 가지 문제 상황을 해결할 수 있어야 한다.. 보통 큰 용량의 데이터 관리 시에 발생하는 문제는 하나의 데이터에 대한 다양한 사용자의 동시 다발적인 접근으로 인해서 이뤄지는데 이에 대한 해결책 즉 요구사항을 만족해야 대용량의 데이터를 잘 처리할 수 있다.

이러한 문제를 해결하는 데 있어서 file system 과 같은 우회 경로에 대해서 생각해 볼 수도 있다 하지만 file system 이 가지는 몇몇 가지 한계가 있어 대용량의 데이터 관리에는 부적합하다. 이는 추후에 다루도록 하겠다. 따라서 개발된 데이터 베이스가 가지고 있는 여러 가지 특징들이 존재한다.

데이터 베이스의 중요한 특징들은 DBMS 의 발전에 있어서 중요한 motivation 이 된다. 후에 이 논문에서는 abstraction 과 introduction 에서 언급한 내용들을 뼈대로 왜 데이터 베이스의 몇 가지 특징들이 중요한지에 대해서 다룰 것이다.

2. REQUIREMENT

대용량의 데이터를 다루는데 있어서 다양한 기술들이 필요하다. 이러한 기술들에 대해서 알아보고 다른 우회경로에 대한 논의를 하겠다.

2.1 DATA MANAGE REQUIREMENT

대용량의 데이터를 저장하는 문제를 생각해보자. 방대한 크기의 데이터는 보통 disk 나 tape 에 저장이 된다. 하지만 이에 대한 접근 속도는 매우 느리다. 32-bit 컴퓨터의 경우 데이터를 4GB 의 용량씩만 참조할 수 있기 때문에 데이터의 접근 시에 많은 시간이 소요된다. 이에 대해 일련의 데이터를 조금씩 확인하는 것이 아니라 필요로 하는 데이터로 한 번에 접근하여서 참조할 수 있도록 할 수 있는 기술이 요구된다.

또한 Data concurrency 문제에 대해서도 고려해야 한다. 하나의 데이터에 다양한 사용자들이 동시에 접근한다. 이 때 data concurrency 가 지켜지지 않으면 어떠한 데이터의 수정 중에 발생한 invalid 한 데이터를 사용하게 되는 혼란이 생길 수 있기에 이러한 문제를 막을 수 있어야 한다.

Data concurrency 를 포함해서 다양한 사용자가 동시에 일련의 데이터에 접근한다는 특징으로 인해서 직면하는 문제들을 해결하는 것이 데이터의 관리에 있어서 매우 중요하다. 이러한 문제로 인해서 요구되는 사항이 접근 권한관리와, 보안문제이다. 많은 사용자에 대해 유연한 보안 정책을 가지고 있어야 하며 접근에 대한 관리도 필요하다.

정리하면, 대용량의 데이터에 대한 사용자들의 동시다발적 접근을 효율적으로 관리해야 좋은 DATA MANAGING 을 할 수 있다. 이 때, 데이터에 쉽게 접근할 수 있어야 하며, 그 데이터는 독립적으로 작동해서 빠른 수정이 가능할수록 좋다. 또한 데이터의 동시 접속에 대한 관리와 접근 권한, 보안상의 문제에 대한 관리, 시스템 충돌에 대한 recovery 역시도 대용량의 데이터 처리에 있어서 핵심적인 요구사항이다.

2.2 LIMIATION OF FILE SYSTEM

File system 을 이용한 데이터의 관리에 대해서 종종 언급이 되지만 file system 은 요구 사항들에 대한 한계점들이 있다.

먼저 File system 을 이용할 경우 TAPE 나 DISK 에 저장된 정보에 접근을 할 때 한정된 양씩 scan 을 해야 한다. 당연히 target information 에 바로 접근하는 것 보다 속도가 느려진다.

DATA CONCURRENCY 의 측면에서도 만일 file system 은 정보가 저장된 주소에 대해서 직접적으로 관리를 해야 하기 때문에 구현이 복잡하다. 구현상의 복잡성은 사용자의 접근 권한을 다루는 문제나 보안상의 문제에도 적용이 된다. OS 는 보통 password 를 이용한 보안 정책을 사용하기 때문에 대용량의 데이터를 다루는 데에는 다소 부적합하다..

File system 은 시스템 충돌과 같은 문제가 생겼을 때 훼손된 자료에 대해서 복구하는 등의 내용에 대해서도 취약하다.

3. CORE DATABASE CONCEPT

2.2 에서 언급했듯이 file system 은 대용량의 데이터를 다루는 데에 있어서 한계가 있다. 3 에서는 대용량의 데이터를 다루는데 필요한 특징과 이에 대한 중요성에 대해 논의한다.

3.1 DATA MANAGE REQUIREMENT

위와 같은 문제들을 해결하기 위해 필요한 특징은 “데이터 독립성”, “효율적인 데이터 접근”, “Data Integrity and Security” 그리고 “Concurrent Access and Crash Recovery” 이다.

효율적인 데이터 접근은 접근 시간과 데이터의 처리시간을 줄여준다. 앞서 2.1 에서 데이터를 scan 하여서 생기는 시간을 최소화하기 위해서 데이터를 바로 참조할 수 있도록 하는 구조가 필요하다고 했다. 이는 후의 쿼리문과 관련이 있다.

또한 2 에서 계속 언급한 데이터에 대한 사용자의 동시 접근으로 인해서 생기는 문제에 대해서 자료의 효용성에 대해서 미리 확인을 하는 메커니즘을 가지고 있는 것이 좋은데 이를 DATA INTEGRITY 와 SECURITY 로 볼 수 있다.

Concurrency 에 대한 언급은 앞에서도 했다. 사용자가 바꾸고 있는 invalid data 를 valid 하게 사용함으로써 생기는 문제를 해결할 수 있어야 한다. 따라서 이에 대한 concurrency control 개념이 필요하다. 또한 transaction 이라는 기술을 이용하여 recovery 에 대한 문제도 처리할 수 있어야 한다.

데이터의 독립성 또한 중요한 데이터베이스의 특징이다.. 이는 특정 데이터가 수정되더라도 다른 데이터나 다른 schema 에 영향을 주지 않도록 하는 것을 말한다.

3.2 KEY CONCEPT INFORMATION

3.2.1 Data Independence and Query language

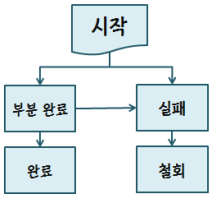
데이터의 독립성은 사용자 스키마, 논리적 스키마, 물리적 스키마를 분리해 각각의 스키마를 독립적으로 관리하기 위한 기술로 특정 스키마가 수정되더라도 다른 스키마에 그러한 영향이 미치지 않도록 한다. 이는 특정 데이터의 수정에 대해서 다른 데이터가 손상되는 문제를 피하게 해주고 성능적인 측면에도 이점이 있어서 대용량의 데이터를 관리하는 데이터 베이스에 필수적인 특징이다.

쿼리는 쉽게 말해 요청을 보내는 것이다. 쿼리문은 이러한 요청을 전달하는 일종의 high level language 이다. 쿼리문은 일반적인 영어 문장과 비슷한 형식을 가지고 있어서 익히기가 용이하다. 또한 쿼리라는 구조를 통해서 대용량의 데이터를 효율적으로 관리해서 대용량의 데이터에 손쉽게 접근할 수 있도록 해서 성능 면에서 큰 효과가 있다.

즉 데이터의 독립성과 쿼리문은 데이터를 효율적으로 관리하여서 성능을 높인다는 점에서 공통적인 이점이 있다.

3.2.2 Transaction, Concurrency Control and Recovery

Transaction 은 쪼갤 수 없는 명령들의 단위이다. 중간에 끊겨서는 안 되는 명령들에 대해서 시스템 충돌과 같은 문제에 직면했을 때 효율적으로 명령을 취소하고 데이터를 안전하게 관리 하기 위해서 필수적인 개념으로 [figure 1]이 transaction 의 핵심 mechanism 을 나타내고 있다.



[Figure 1] concept of transaction

Concurrency Control 은 앞에서 많이 언급한 개념으로 다양한 사용자의 접근으로 인해서 순간적으로 발생하는 invalid data 에 대해서 효율적으로 관리하기 위해서 데이터베이스에 필수적이다. Concurrency control 기술이 있어야 각각의 사용자는 다른 사용자로 인한 데이터의 의도치 않은 변형이 이뤄지지 않는다는 것을 보장받을 수 있다. [table 1]은 concurrency 가 보장되지 않았을 때 직면하는 문제에 대해서 보여준다.

Thread A	Thread B	Contents of i
		5
fetch i into register		5
increase i in register	fetch i into register	5
store i into memory	increase i in register	6
	store i into memory	6

[Table 1] wrong concurrency problem

Recovery 는 말 그대로 데이터를 복구하는 것을 말한다. 시스템 충돌과 같은 의도치 않은 시스템상의 문제가 생겼을 때 데이터에 손상이 날 수도 있다. 하지만 은행과 같은 중요한 데이터를 관리하는 곳에서 이런 데이터 손상은 치명적이다. 따라서 데이터의 손상에 대해서 recovery 를 하는 매커니즘은 데이터 베이스에서 매우 중요하다.

4. REFERENCES

[1] Ramakrishnan, Gehrke, 2003, *Database Management Systems*