데이터 쿼리연산 종류에 따른 RDBMS 처리 성능 분석

Performance Analysis on the Data Query Operations of RDBMS

저자
(Authors)

이영석, 김용희, 김응모
YeongSeok Lee, YongHee Kim, Ung-Mo Kim

출처
(Source)

한국통신학회 학술대회논문집 , 2015.11, 370-371 (2 pages)
Proceedings of Symposium of the Korean Institute of communications and Information Sciences , 2015.11, 370-371 (2 pages)

발행처
(Publisher)

한국통신학회
Korea Institute Of Communication Sciences

URL

http://www.dbpia.co.kr/Article/NODE06564716

APA Style

이영석, 김용희, 김응모 (2015). 데이터 쿼리연산 종류에 따른 RDBMS 처리 성능 분석. 한국통신학회 학술대회논문집 , 370-371.

데이터 쿼리연산 종류에 따른 RDBMS 처리 성능 분석

이영석, 김용희, 김응모*
성균관대학교, 성균관대학교, *성균관대학교

alhaq@skku.edu, dkfaustj6@skku.edu, *ukim@skku.edu

Performance Analysis on the Data Query Operations of RDBMS

YeongSeok Lee, YongHee Kim, Ung-Mo Kim*
Sungkyunkwan Univ., Sungkyunkwan Univ., *Sungkyunkwan Univ.

요 약

현재 RDBMS 는 가장 많이 쓰이는 데이터베이스 시스템이며, 데이터들을 관계적으로 표현하고 데이터들 간 연산 작업이 편리하다는 이점을 가진다. 존재하는 다양한 RDBMS 에서 사용자가 가진 데이터에 맞는 RDBMS 를 선택하는데 도움을 주기 위하여 데이터 연산 종류에 따른 RDBMS 분석에 초점을 맞췄다. 본 논문에서는 데이터 연산 종류에 따른 RDBMS 성능 분석을 위해 대표적으로 사용되는 RDBMS 를 가지고 실제로 사용되는 데이터들에 대한 스키마를 구성하였다. 각 RDBMS 에 INSERT/SELECT/JOIN 연산을 구현하여 연산에 대한 성능을 분석하였다. 이러한 성능 비교를 통해 사용자 데이터에 적합한 RDBMS 를 선택 할 수 있을 것이다.

Ⅰ. 서 론

RDBMS 는 관계적 데이터 표현과 테이블 간의 연산 작업이 편리하기 때문에 다양한 데이터를 체계적으로 저장할 수 있다. ACID(원자성, 일관성, 고립성, 지속성)의 성질을 가지기 때문에 데이터베이스 트랜젝션이 안전하게 수행되는 것을 보장한다.[1] 반면, NoSQL 은 ACID 의 일관성과 고립성(CI)을 제외하고 대신 속도와 분산 작업에 중점을 둠으로써 RDBMS 에 비해 데이터의 관계성과 안정성이 떨어진다. [2]

RDBMS 에서 JOIN 연산은 서로 다른 테이블의 관계를 이용해 결합된 정보를 제공한다. JOIN 연산은 현실의 데이터들의 관계를 표현하기 위해 필수적으로 사용되는 연산이다. 하지만 NoSQL 에서는 위에서 말한 ACID 중 CI 성질을 가지지 않으며 JOIN 연산을 지원하지 않기 때문에 데이터들을 관계적으로 저장하는 것에 적합하지 않다.

본 논문에서는 여러가지 RDMBS 를 동일한 조건에서 JOIN 을 포함한 다양한 연산 작업의 성능 분석을 통해, 사용자가 원하는 적절한 RDBMS 를 선택하는데 도움을 주는 것을 목적으로 한다. 가장 대표적으로 사용되는 RDBMS 로 MySQL, PostgreSQL, SQLite 를 가지고 테스트 할 것이다. 2 장인 본문에서 각 연산에 대한 소개와 RDBMS 에 따른 성능 실험 결과를 제시하며 3 장에서 결론을 맺는다.

Ⅱ. 본론

본 논문에서는 아래와 같은 연산을 통해 성능 분석을 할 것이다. 데이터를 저장하고 처리하는데 가장 많이 사용되는 대표적인 연산으로 INSERT/SELECT/JOIN 연산 들이 있다.[3] 해당 연산들의 성능에 따라서 사용자가 결과값을 얻는데 걸리는 시간에 큰 영향을 미치기 때문에, 각 RDBMS 는 해당 연산을 빠르게 처리하는 것이 매우 중요하다. RDBMS 별로 해당 연산에 대한 처리 시간을 측정해 성능을 비교 및 분석할 것이다.

1. 연산종류

1) INSERT
데이터베이스에 저장할 데이터를 삽입하는 연산이다. INSERT INTO table 명 VAlUES (value1, value2, …) 형태로 사용한다.

2) SELECT
데이터베이스에 저장되어 있는 테이블로부터 데이터를 검색하여 출력하는데 사용하며, 일반적으로 SELECT column_name FROM table WHERE condition 형태로 사용한다.

3) JOIN
서로 다른 테이블의 상관 관계를 이용하여 두 테이블의 결합된 정보를 제공하는 연산이다. 두 테이블의 상관관계를 확인해 일치되는 것을 출력해야 하기 때문에 부하가 많이 걸리는 연산 중 하나이다.

2. 성능 측정

INSERT/SELECT/JOIN 연산에 대해서 테스트 데이터들을 통한 실제 수행 시간을 측정한다. 약 100 만개의 데이터를 이용해 동일한 환경에서 각 연산에 대한 시간을 측정하였다.

Id	minutes_past	radardist_km	ref
0	0	0.0	0.00
1	3	10.0	0.00
1	16	10.0	0.00
1	25	10.0	0.00
1	35	10.0	0.00
1	45	10.0	0.00
1	55	10.0	0.00
2	1	2.0	9.00
2	6	2.0	9.99

그림 1. U.S. National Weather Service Polarimetric 데이터

그림 1 은 U.S. National Weather Service 에서 제공한 Polarimetric 데이터 중 일부이며, 테스트 데이터로 테이블 A 에 저장 될 것이다. 전체 ROW 수는 1048576 개 이며, 데이터 크기는 15.5MB 이다.

1) INSERT(삽입) 시간

테이블 A 에 대한 INSERT 연산을 실행 하였다. 그림 2 에서 각 RDBMS 에 따라서 테이블 A 에 모든 데이터를 삽입하는데 걸리는 시간을 그래프로 보여준다.

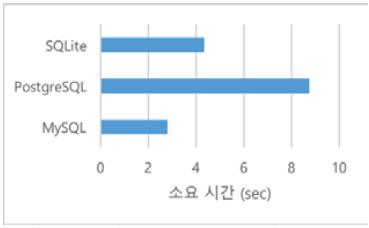

그림 2. 테이블 A 에 대한 RDBMS 삽입 소요시간

약 100 만개의 데이터를 삽입하는데 MySQL 이 2.79 초로 가장 빨랐으며, PostgreSQL 이 8.738 초로 가장 느리게 나타났다. 차이가 약 3 배 정도로 눈에 띄는 차이가 났음을 알 수 있다.

2) SELECT(조회) 시간

테이블 A 에 대해서 각 RDBMS 별 조회 시간을 측정하였다. 그림 3 은 테이블 A 에 대한 조회 소요시간이며 PostgreSQL 이 1.685 초로 가장 빨랐으며, MySQL(2.095 초) SQLite(3.53 초) 순이였다.

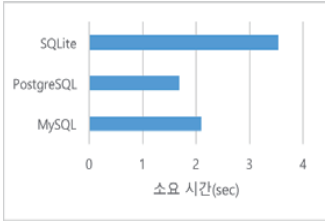

그림 3. 테이블 A 에 대한 RDBMS 조회 소요시간

3) JOIN 연산 시간

JOIN 연산은 두 테이블간의 같은 관계를 가진 값들끼리 연결해 보여주는 연산이다. 따라서 또 다른 테이블 B 를 추가해 테이블 A 와 JOIN 연산에 걸리는 시간을 측정하였다. 그림 4 는 기대값(Expected) 데이터 중 일부이며, ROW 수는 717626 개, 용량은 12.3MB 이다.

Id	expected
0	0.00000000
1	0.08576453
2	0.00000000
3	1.59400433
4	6.91338020

그림 4. U.S. National Weather Service 기대값 데이터

JOIN 연산 중에서 INNER JOIN 연산과 NATURAL JOIN 연산을 각각 수행하였다.

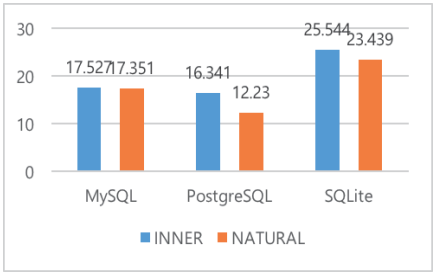

그림 5. 테이블 A 와 B 에 대한 JOIN 연산 소요시간

그림 5 를 보았을 때, JOIN 연산은 PostgreSQL 이 16.341 초 12.23 초로 가장 빠르게 나타났으며 특히 NATURAL JOIN 연산을 할 때 성능이 약 30% 되었다. MySQL 이나 SQLite 는 PostgreSQL 에 비해 느리고 연산을 편하게 도와주는 NATURAL JOIN 에서 별다른 성능 향상을 볼 수 없었다.

Ⅲ. 결론

본 논문에서는 데이터 연산에 따라서 동일한 환경 및 스키마에서 각 RDBMS 별 소요 시간을 측정하였다. 성능 분석 결과 PostgreSQL 이 SELECT 및 JOIN 연산이 가장 빨랐고, INSERT 연산은 MySQL 이 가장 빨랐다. 이것으로 사용자는 INSERT 가 빈번히 일어나는 데이터는 MySQL 을 사용하면 이점을 가지며, SELECT/JOIN 연산을 자주 수행하는 데이터들은 PostgreSQL 을 사용하는 것이 좀 더 이점이 있을 것으로 기대된다. 사용자는 데이터에 자주 일어나는 연산 종류에 적합한 RDBMS 를 선택하여 시스템을 만들 수 있다.

ACKNOWLEDGMENT

본 연구는 미래부가 지원한 2013 년 정보통신.방송(ICT) 연구개발사업의 연구결과로 수행되었음(1391105003). 이 논문은 2013 년도 정부(교육부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(NRF-2013R1A1A2008578).

참 고 문 헌

[1] Michael Kruchenberg and Jay Pipes. Pro MySQL Appress, 2005.

[2] Christof Strauch, NoSQL Databases. 2012.

[3] E. F. Codd. "A Relational Model of Data for Large Shared Data Banks", IBM Research Laboratory, San Jose, California.