

## DBMS의 성능 향상을 위한 Disk I/O에 대한 연구

\*강동필, \*\*문주희  
\*하나로통신(주), \*\*세종대학교

### A Study on Disk I/O for the Improvement of DBMS Performance

\*Dong Pil Kang, \*\*Joo Hee Moon  
\*Hanaro Telecom, Inc., \*\*Sejong University

#### 요약

본 연구는 DBMS에서의 Disk I/O 종류별, 옵션별 특성 및 성능 분석을 하고 이를 실제 시험을 통하여 검증하였다. Raw Device와 Quick I/O 등이 뛰어난 성능을 보여주었으나, UFS(Unix File System)의 사용 및 관리의 용이성을 고려하여 특성에 맞는 Disk I/O 구성 방식이 필요함을 제시하였다.

#### 1. 서론

컴퓨터 기술의 발달에 따라 빠른 처리 능력을 가진 고성능 컴퓨터와 고속의 데이터 전송능력을 지닌 네트워크의 출현으로 고속 정보처리 요구를 만족시키는 컴퓨팅 환경이 마련되고 있다. 그러나 이러한 컴퓨터 시스템의 모든 부분이 고속 정보처리 요구사항을 충족시킬 수 있도록 향상되지는 않고 있다. CPU와 메모리의 처리속도는 최근 급속하게 증가되었고, 향후 이러한 추세는 계속될 것이라는 전망이 우세하다.

따라서, 입출력(I/O) 시스템의 병목현상을 줄이는 것이 전체시스템의 성능을 향상시키는 중요한 요소가 되고 있다.

DBMS의 성능 향상에 있어서도, 역시 가장 중요한 요소가 이러한 입출력(I/O) 시스템의 병목 현상을 줄이는 것이 될 것이다.

따라서, 본 연구에서는 2절에서 Database File의 구성 기반을 살펴보고, 3절에서 File System의 성능 저하 요인, 4절에서 시험을 통하여, 5절에서 효율적인 Disk I/O 구성 방안을 제시코자 한다.

#### 2. Database File 구성 기반

Database의 File을 구성하는 기반은 다음과 같다.

##### 2.1 File System

File System은 전체적인 파일 시스템 트리 또는 디렉토리 구조로 사용자 입장에서는 정보를 조직화하고 저장하기 위해 사용되는 파일과 디렉토리의 집합, 운영체제에서의 파일시스템은 파일과 디렉토리의 위치를 정의하고 있는 테이블 구조를 포함하는 파티션을 의미한다. Database 관리자 입장에서 보면, File System의 다양한 기능을 사용할 수 있으므로, 백업 및 복구, Data file의 생성과 확장 등 Database를 운영하는데 있어서, 관리가 매우 편리하다.

Berkeley Fast File System에 기초한 Unix File System(UFS)와 Veritas File System(VxFS) 등이 있다

##### 2.2 Raw Device

Raw partition이라고도 하며, File System을 통해 마운트되거나, 쓰여지지 않고 Character Special 디바이스 드라이버를 통해 액세스되는 Disk Partition다.

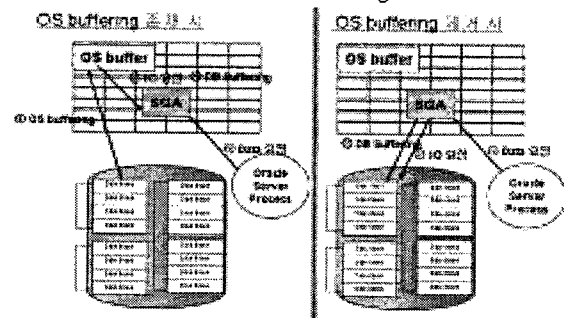
Data가 어떻게 쓰여질지는 어플리케이션에 달려있다. OLTP에서 가장 좋은 성능을 내며, Disk에 Data가 직접 쓰여지기 때문에 Data 손실의 가능성이 낮다. 그러나, 다루기가 까다롭다.

#### 3. File System의 성능 저하 요인

##### 3.1 Double Buffering

대부분의 DBMS는 자체적인 버퍼 캐시를 가지고 있으므로, 파일 시스템의 버퍼 캐시는 불필요하다. 파일 시스템 버퍼에 캐시된 Database Data는 중복되고, 결국 메모리의 낭비와 버퍼를 관리하기 위한 CPU 자원을 추가적으로 사용해야 한다. Direct I/O를 사용함으로써, Data는 DBMS 캐시와 Disk간에 직접 쓰여지므로, CPU자원의 이용을 줄이고, 메모리를 소모를 줄여 트랜잭션 처리 능력을 향상시킬 수 있는 DBMS의 버퍼 캐시로 추가로 사용될 수 있다.

그림 1. Double Buffering



##### 3.2 Single Writer Lock

###### 3.2.1 Unix Single Writer Lock

일반 UNIX 파일 시스템은 파일의 무결성을 유지하기 위해 한 시점에 하나의 프로세스가 Lock을 제어하여 쓰기 작업을 수행한다.

3.2.2 Database Lock

데이터베이스는 데이터에 대해서 로우 레벨과 블록 레벨의 Lock을 제어한다. 데이터베이스에서 제어하는 Lock과 중복하여 파일 시스템의 단일시점 쓰기 Lock에 의한 제한을 받아 실제적으로는 한 시점에 한 프로세스에 의한 쓰기 작업만이 수행된다.

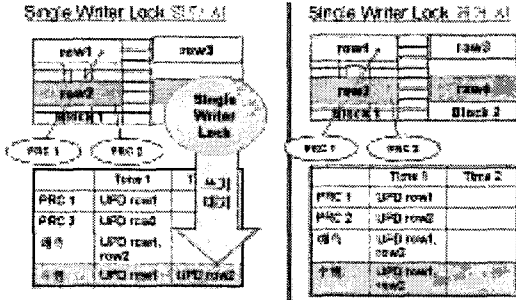


그림 2. Single Writer Lock

3.3 Asynchronous I/O 지원

3.3.1 Asynchronous I/O

System Call이 비동기 방식 : System Call 호출 후 다음 I/O System Call 수행하며, Return을 요구하지 않음. Non-blocking System Level 읽기와 쓰기를 수행하여, System이 병렬 I/O 요청을 동시에 처리할 수 있도록 해준다. Solaris와 같은 운영체제는 Raw Device에 대한 Asynchronous I/O를 커널에서 지원한다. 그러나, File System에 대하여는 Asynchronous I/O를 지원하지 않으므로, DBMS에서 Asynchronous I/O를 지원하여도 Asynchronous I/O로 동작하지 않는다. 일반적으로 Asynchronous I/O로 동작하지 않으면, 상당한 성능 저하를 가져올 수 있다.

3.3.2 Synchronous I/O

System Call이 동기 방식 : System Call 수행 후 Return을 받은 후 다음 I/O System Call 수행.

데이터 파일로의 쓰기 작업이 빈번한 경우 단일 쓰기 작업 시의 성능 향상이 가능하고, 사용자가 많은 경우 쓰기 경합에 대한 지연을 제거하여 성능 향상을 가져온다.

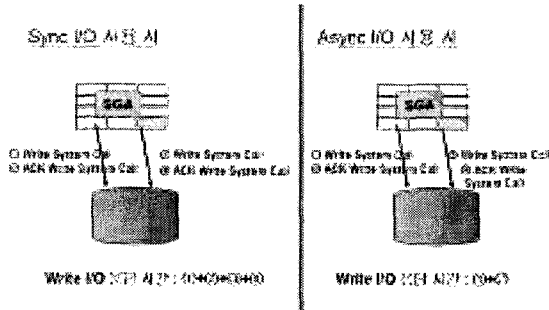


그림 3. Asynchronous I/O

4. 성능 시험

4.1 시험개요

데이터베이스의 Disk I/O에 따른 성능 분석을 위하여, UFS, Raw device, Quick I/O, Cached Quick I/O 상에서 비교

시험하였다.

4.2 테스트 방법

4.2.1 vxbench

파일시스템, Raw Device 대상 순수 입출력 부하 테스트 툴로, 데이터베이스의 I/O 유형에 따른 입출력 부하 테스트에 사용한다.

데이터베이스 성능 검증 트랜잭션 기준치는 TPC-C benchmark가 일반적인 구성이나, TPC-C의 기초 데이터 생성 및 동시 사용자 수 Simulation이 어려우며, 기타 환경적인 영향으로, 순수 I/O에 대한 검증을 위한 vxbench로 대체하였다.

1) 구성 정보

- 비교대상 : UFS, RAW Device, Quick I/O(QIO), Cached Quick I/O(CQIO)
- 볼륨구성 : 각 2GB Volume, 8K block size
- 동시 Thread : 50

2) 비교 요소

- Elapsed Time, I/O 처리량
- CPU 사용량
- 동시사용자 수별 균등한 I/O 속도

4.2.2 Database SQL\*Loader 테스트

1) 구성 정보

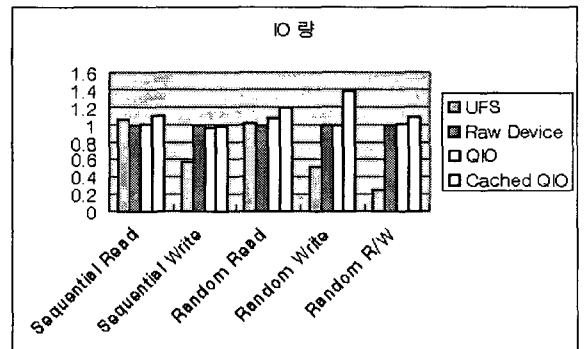
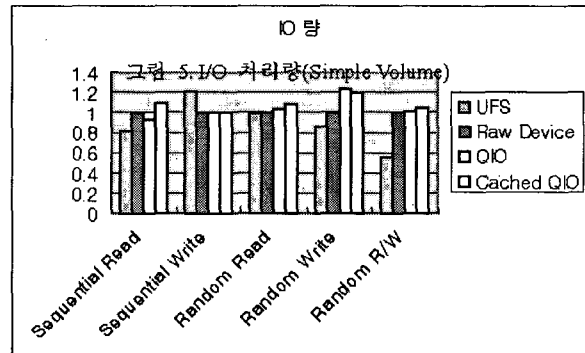
비교대상 : UFS, Raw device, Quick I/O(QIO)

2) 비교 요소

Load 수행 시간

4.3. 시험 결과 분석

4.3.1 I/O 처리량

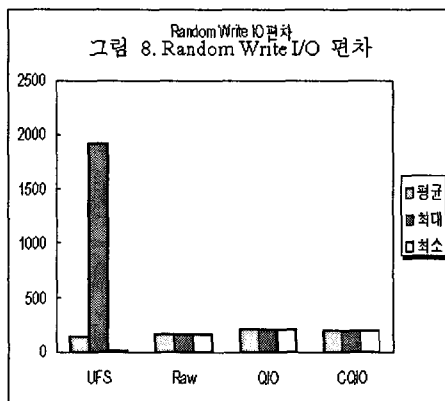
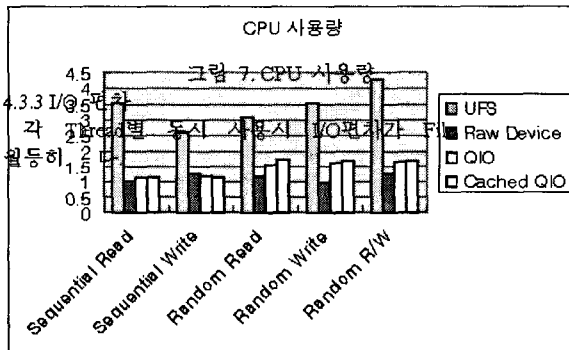


입출력 유형	Simple Volume	Stripe Volume
Read Performance	비슷함	Sequential Read는 비슷하며, Random Read는 UFS 대비 좋음
Write Performance	CQIO가 가장 좋으며, QIO, RAW 동일하며, UFS는 불균등한 IO를 가짐 UFS는 Raw Device 대비 Random Write는 86%, Sequential Write는 120%임 (Sequential Write는 일부 IO 집중으로 인해 UFS 결과가 좋음)	CQIO가 가장 좋으며, QIO, RAW 동일하며, UFS는 Raw Device 대비 Sequential 56%, Random 51%임
Random R/W	UFS가 월등히 나쁨(Raw Device 대비 55%)	UFS가 월등히 나쁨(Raw Device 대비 24.8%)

표 1. I/O 처리량 분석

#### 4.3.2 CPU 사용량

Raw Device, Quick I/O가 월등히 작으며, File System이 2.5배에서 4.5배까지 소요되었다.



#### 5. 결론

최근 하드웨어 기술의 발전에 따라, 고성능 컴퓨팅 환경이 마련되고 있으나 입출력(I/O) 시스템의 병목 현상은 가장 큰 걸림돌이 되고 있다. DBMS에 있어서도 Disk I/O의 병목 현상이 가장 큰 성능 향상 요소가 되고 있다. 본 연구에서는 Disk I/O 구성 방식에 따라, 입출력 유형별로 I/O 처리량, CPU 사용량, I/O 편차를 측정하였다. I/O 처리량은 Read 성능에서는 비슷한 성능을 보여 주었으나, Write 성능에 있어서는 Cached Quick I/O가 가장 뛰어난 성능을 UFS가 가장 조제한 성능을 보여 주었다. 특히, Random R/W에서는 UFS의 성능이 Raw Device에 비하여 월등히 나쁜 결과를 보여주었다. 또한, CPU 사용량이 Raw Device 대비하여 월등히 높으며, UFS는 월등히 큰 I/O 편차를 보여준다. 이러한 UFS의 성능상의 문제점에도 불구하고 사용 및 관리상의 용이성이 뛰어나므로, 각 Database별 환경에 적합한 Disk I/O 구성 방식을 선택하는 것이 중요하다. 또한 UFS의 관리상 편리성과 Raw Device의 우수한 성능을 복합한 새로운 Disk I/O 구성 방식에 대한 추가적인 연구가 필요할 것으로 보인다.

#### 참고문헌

- [1] Steve D. Pate, "Unix Filesystems : Evolution, Design, and Implementation", John Wiley & Sons, Inc., January 2003
- [2] VERITAS Software Corporation, "Veritas Database Edition 3.5 for Oracle : Database Administrator's Guide", July 2002
- [3] Oracle Corporation, "Oracle9i Administrator's Reference Release2(9.2.0.1.0) for UNIX Systems", Part No. A97297-01, May 2002
- [4] Bob Sneed, "Oracle I/O : Supply and Demand", SMI Performance & Availability Engineering, October 2001
- [5] Richard McDougall, Sriram Gummuluru, "Oracle File System Integration and Performance", Sun Microsystems, January 2001
- [6] Jim McKinstry, "I/O Considerations for Database Performance - Overcoming the Database I/O Bottlenecks", MTI, February 2001
- [7] 정광을, "유닉스 환경에서의 클라이언트 서버와 DB 서버의 성능관리 및 튜닝에 관한 연구", 연세대학교 산업대학원, 2000년 6월