**배치프로그램 튜닝**

1. **배치프로그램 튜닝 개요**

1) 배치프로그램 : 사용자와 상호작용이 없이 대용량의 데이터를 처리하는 일련의 작업들을 묶어 정기적으로 반복수행하거나 정해진 규칙에 따라 자동 수행되며 3가지로 구분이 된다.

- 정해진 시점에 수행하는 정기배치, 조건을 충족할 경우 수행되는 이벤트성 배치, 명시적 요구가 있을 때 행하는 ON-DEMAND 배치가 있다.

2) 배치 환경의 변화 : 요즘 시간 배치 작업의 비중이 증가하며 분 배치 작업이 일부 존재하며 ON-DEMAND 배치를 제한적이나마 허용

3) 성능 개선 목표 설정

- 온라인 프로그램 : 최초 응답속도 최적화 목표

- 배치프로그램 : 항상 전체 처리속도 최적화를 목표로 설정

4) 배치프로그램 구현 패턴과 튜닝 방안

- 절차형 프로그램 : 어플리케이션 커서를 열고 루프 내에서 다른 SQL 반복 처리 형태. 반복적인 DBMS CALL 발생하며 RANDOM I/O 위주로 동일데이터를 중복 액세스

- ONE SQL 위주 프로그램 : ONE SQL로 구성하거나 집합적으로 정의된 여러 SQL를 단계적으로 실행. INDEX 스캔보다는 FULL TABLE 방식으로 HASH 조인 방식으로 처리

->ONE SQL 이 훨씬 더 나음

1. **병렬처리 활용**

- SQL이 수행해야 할 작업 범위를 여러 개의 작은 단위로 나누어 여러 프로세스가 동시 처리

- 여러 프로세스가 동시에 작업하므로 대용량 데이터를 처리할 때 극적인 수행속도 절감

- PARALLEL 힌트를 사용할 때는 반드시 FULL 힌트도 함께 사용하는 습관이 필요

- 옵티마이저가 인덱스 스캔을 선택하면 PARALLEL 힌트는 무시

- PARALLEL\_INDEX 힌트를 사용할 때는 반드시 INDEX / INDEX\_FFS 힌트 사용.

- 옵티마이저가 FULL TABLE SCAN을 선택하면 힌트는 무시

1) QUERY COORDINATOR와 병렬 서버 프로세스

- QC : 병렬 SQL문을 발행한 세션

- 서버 프로세스 : 실제 작업을 수행하는 개별 세션들

- QC 역할 : 병렬도와 오퍼레이션 종류에 따라 하나 또는 2개의 병렬서버 집합 할당. 서버 풀로부터 필요한 만큼 서버 프로세스를 확보 및 부족분 신규 생성하며 병렬로 처리하도록 사용자가 지시하지 않는 테이블은 QC가 직접 처리. 산출물을 통합하며 최종 결과집합을 사용자에게 전송하며 쿼리 결과를 전송하는 단계에서 수행되는 스칼라 서브 쿼리도 QC수행.

2) INTRA-OPERATION / INTER-OPERATION PARALLELISM

- INTRA : 서로 배타적인 범위를 독립적으로 처리. 절대 프로세스간 통신 발생 X

- INTER : 반대편 서버 집합에 분배하는 작업. 정렬된 결과를 QC에 전송. 프로세스 간 통신이 발생하며 통신채널이 필요하다,

3) 테이블 큐

- 서버 집합간 또는 QC와 쿼리 서버 집합 간 데이터 전송을 위한 연결된 PIPELINE.

- 테이블 큐에는 병렬도의 제곱만큼의 프로세스 필요

4) 데이터 재분배

- RANGE : ORDER BY 또는 SORT GROUP BY를 병렬로 처리할 때 사용. 두 번째 서버 집합의 프로세스 별로 처리 범위 지정. QC는 프로세스에 작업 범위를 할당하고 정렬 작업 참여 X

- HASH : 조인이나 HASH GROUP BY를 병렬로 처리할 때 사용. 각 키 값을 해시 힘수에 적용하고 리턴 된 값에 따라 데이터를 분배하는 방식

- BROADCAST : 크기가 매우 작은 테이블이 있을 때 사용. 첫 번째 집합 데이터를 두 번째 서버 집합에 속한 모든 병렬 프로세스에게 전송하는 방식

- KEY : PARTIAL PARTITION-WISE 조인 : 한쪽만 PARTITION이 되어 있을 때 사용. 특정 칼럼들을 기준으로 테이블 또는 인덱스를 파티셔닝 할 때 사용하는 분배 방식.

- ROUND-ROBIN : 반대편 병렬 서버에 무작위로 데이터를 분배할 때 사용.

5) PQ\_DISTRIBUTE 힌트

- 이 힌트는 조인에 앞서 데이터를 분배하는 과정만 관여하며 조인을 결정하는 힌트는 아님

- 이 힌트를 사용했을 때 프로세스가 몇 번 일어나는지가 주요 포인트

- PQ\_DISTRIBUTE ( INNER TABLE, OUTER\_DISTRIBUTE, INNER DISTRIBUTE) 방식

- FULL-PARTITION WISE 조인일 경우 PQ\_DISTRIBUTE (INNER, NONE, NONE)인 경우 DEGREE + DEGREE 로 프로세스 계산.

- HASH HASH 방식인 경우 각 DEGREE X 2를 더할 것. NONE은 그냥 DEGREE만.

- NONE을 제외한 것들은 대부분 DEGREE X 2로 더할 것.

6) 병렬처리 시 주의사항

- 과도하게 사용시 시스템을 마비시킬 수 있다. (병행 처리 남발 X. 10분을 5분으로 X)

- 무조건 병렬도를 높인다고 성능이 선형적으로 좋아지지 않음. CPU 개수의 80%정도로

- 시스템 리소스의 최대한 사용이 필요한 시점

- 병렬 DML 수행시 LOCK이 걸려 트랜잭션이 활발한 주간에 사용 금지 (OLTP X)