임베디드 DBMS의 전력 기반 질의 최적화를 위한 비용 모델에 관한 연구*

김도윤○, 박원주, 장주연, 박성환, 박상원
한국외국어대학교 컴퓨터및정보통신공학과
e-mail:{dykim wjpark, jyjang, shpark, swpark}@dislab.hufs.ac.kr

A Study on Cost Models for Energy-based Query Optimization on Embedded DBMS

Do-Yun Kim○, Wonjoo Park, Ju-Yeon Jang, Sung-Hwan Park, Sangwon Park
Dept. of Computer and Information Communication Engineering,
Hankuk University of Foreign Studies

요 약

PC 및 서버 급에서 DBMS가 아주 폭넓게 사용되어지고 있으며 그 뿐 아니라 컴퓨팅 파워가 높아짐에 따라서 임베디드 시스템에서도 DBMS가 필요해졌다. 임베디드 시스템에서 DBMS가 충분히 동작할 만큼의 성능을 발휘하게 되었고, 이에 따라 임베디드 시스템에서 동작하는 응용프로그램들도 임베디드 DBMS를 사용하게 되었다. 임베디드 시스템이 점차 플래시 메모리를 사용하는 추세에 맞추어 플래시 기반 임베디드 DBMS 기술 개발이 중요하다. 플래시 메모리의 특성에 맞춘 임베디드 DBMS를 개발하지 않으면, 결과적으로 플래시 메모리의 성능을 저하시키며, 수명도 단축시키는 결과를 초래하게 될 것이다. 특히 임베디드 환경에서는 전기 에너지 자원이 한정되어 있기 때문에 전력 소모를 줄이는 것이 관건이다. 따라서 임베디드 DBMS에서 디스크에서 정의한 비용 모델을 따르는 것은 한계가 있다. 본 논문은 임베디드 DBMS에서 전력 기반 비용 모델을 새롭게 제시하고, 디스크 기반 비용 모델과 비교하여 제시한 비용 모델과의 차이를 보인다.

1. 서론

플래시 메모리는 휴대용 기기의 저장 매체로 각광받고 있다. 플래시 메모리의 용량이 커지면서 기존의 하드 디스크를 대체하고 있다. 배터리로 구동되는 PDA나 휴대 전화 그리고 휴대용 미디어 재생기(portable media player)와 같은 기기의 경우 기기가 사용할 수 있는 전기 에너지 자원이 배터리에 한정되어 있기 때문에 전력 소모를 줄이는 것이 가장 중요한 요소라고 할 수 있다. 또한 플래시 메모리는 읽기에 비해서 쓰기의 비용이 훨씬 많이 든다. 뿐만 아니라 플래시 메모리는 덮어쓰기(overwrite)를 할 수 없다. 이러한 요소들은 데이터베이스의 전력 소비에 대한 비용 모델을 변경하여야 함을 의미한다. 본 논문에서는 플래시 메모리에 적합한 새로운 전력 소비에 대한 비용 모델을 제시하고 실험을 통해서 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 플래시 메모리와 FTL(Flash Translation Layer)[1] 시스템에 대해 살펴보고, 3장에서는 질의 처리 시뮬레이션 시스템의 설계에 대해 설명한다. 4장에서는 플래시 메모리의 비용 계산에 대해서 설명하고 5장에서는 기존 디스크에서 비용 모델과 플래시 메모리에서 새롭게 정의한 비용 모델을 살펴본다. 6장에서 실험을 통한 결과를 분석하며, 7장에서 결론을 맺고 향후 과제에 대해 살펴본다.

* 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력 핵심기술개발사업[2006-S-040-01, Flash Memory 기반 임베디드 멀티미디어 소프트웨어 기술 개발]과 과학기술부 및 대구경북과학기술연구원의 연구개발사업의 일환으로 수행하였음.

2. 관련 연구

2.1. 플래시 메모리와 FTL 시스템

최근 플래시 메모리는 휴대용 장비에 필수적으로 탑재되는 메모리가 되었다. 플래시 메모리는 데이터에 대한 신뢰성, 갱신의 용이성, 낮은 전력 소모, 전원 차단시의 안정성이 큰 특징이다. 플래시 메모리는 읽기, 쓰기와 소거 연산의 수행 시간과 단위가 다르다. 또한 플래시 메모리는 쓰기를 수행하기 위해서 쓰기 전에 소거 연산이 선행되어야 한다. 또한 각 소거 유닛은 약 100만 번까지 소거가 가능하고, 이보다 많은 소거 연산이 발생하게 되면 데이터의 무결성을 보장하지 못한다. 때문에 소거 횟수 평준화(wear-leveling)[2]기법이 필요하다.

<표 1> FTL 알고리즘의 특성

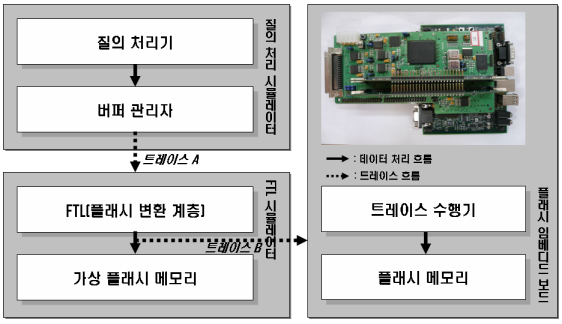
	복사블록 기법	여유공간 기법	로그블록 기법
데이터 블록	고정 섹터 방식	고정 섹터 방식	고정 섹터 방식
매핑	블록 매핑	블록 매핑	혼합 매핑
덮어쓰기	고정 섹터 방식→변동 섹터 방식	변동 섹터 방식	변동 섹터 방식
특징	각 블록 당 복사블록을 사용	블록내의 여유 영역을 사용	한 블록에 로그 블록 하나를 할당.

2.2. FTL(Flash Translation Layer)

플래시 메모리는 여러 단점을 극복하기 위해 FTL이라는 시스템 소프트웨어를 사용한다. FTL은 여러 매핑 방법을 통해서 비효율적인 덮어쓰기 연산을 효율적으로 처리한다. 플래시 메모리는 FTL 모듈을 사용함으로서 전체

적인 성능과 수명을 증진시킬 수 있다. FTL은 매핑 방법이나 물리적인 위치에 쓰는 방식에 따라 표 1과 같은 방법들로 분류할 수 있다. 표 1에서는 FTL 중 복사블록 기법[3], 여유공간 기법[4]과 로그블록 기법[5]을 설명한다.

3. 질의 처리 시뮬레이션 시스템



(그림 1) 질의 처리 시뮬레이션 시스템 구성

본 논문에서 질의 처리 비용을 측정하기 위해 질의 처리 시뮬레이터, FTL 시뮬레이터, 플래시 임베디드 보드로 이루진 질의 처리 시뮬레이션 시스템을 구현하였다. 그림 1은 질의 처리 시뮬레이터와 실험에 필요한 모듈에 대한 전체 시스템 구조를 나타낸다.

질의 처리 시뮬레이터란 DBMS가 수행하는 질의 처리를 가상으로 수행하는 모듈을 말한다. 질의 처리기는 버퍼 관리자를 통해 데이터베이스 페이지를 읽고 쓰는 연산에 대한 트레이스 A를 만든다.

FTL 시뮬레이터는 트레이스 A를 입력으로 받아서 가상 플래시 메모리에 연산을 수행한다. 이 때 가상 플래시 메모리에 요구되는 연산을 트레이스 B를 추출한다. FTL이 필요한 이유는 앞에서 설명한 것과 같이 질의 처리 시뮬레이터는 일반적인 DBMS와 같이 디스크를 저장장치로 간주하기 때문에 플래시 메모리에 동작할 수 있도록 하기 위해서이다. 그 이유는 디스크가 읽기와 쓰기 연산만 존재하는 반면 플래시 메모리는 소거 연산이 필요하다는 특성이 있기 때문이다.

플래시 임베디드 보드는 FTL 시뮬레이터의 트레이스 B를 읽어서 플래시 메모리에서 질의 처리에 의해 발생된 읽기, 쓰기와 소거 연산의 전력을 측정할 수 있다. 플래시 메모리에서 소요되는 시간을 측정하기 위한 계측 시스템은 DAQ(Data AcQuisition) 장비를 사용하였다. DAQ 장비는 데이터 수집 장치이며 플래시 임베디드 보드와 연결하여 시간과 전력을 측정할 수 있다.

4. I/O 연산에 대한 추가 비용비

E_r	: 플래시 메모리 한 개의 페이지를 읽을 때 드는 에너지
E_w	: 플래시 메모리 한 개의 페이지를 쓸 때 드는 에너지
E_e	: 플래시 메모리 한 개의 페이지를 소거할 때 드는 에너지

하드 디스크에서는 데이터베이스의 I/O 하나당 디스크에 하나의 I/O가 발생한다. 하지만, 플래시 메모리에서는 하나의 데이터베이스 페이지 I/O 요청에 추가적인 연산이 발생될 수 있다. 그러므로 플래시 메모리에서 질의 처리를 위한 실제 비용을 계산하는 방법[6]이 필요하다.

$$k = \frac{S_P}{S_{fp}\, m} \tag{1}$$

한 개의 데이터베이스 페이지의 I/O 시에 플래시 메모리에 요청되는 플래시 메모리 페이지 개수 k는 수식 1과 같다. S_p는 데이터베이스의 한 페이지 크기(K 바이트), S_{fp}는 플래시 메모리의 한 페이지의 크기(K 바이트)이고 m은 플래시 메모리에서 칩 인터리빙의 수를 나타낸다. 다음은 읽기와 쓰기 연산에 대한 추가적인 비용의 비, λ(lambda)와 μ(mu)를 구하는 수식을 보여준다.

$$p_r = k \times E_r \tag{2}$$

$$P_r = \sum_{i=1}^{n} p_{r_i} = k \times n_{dr} \times E_r \tag{3}$$

$$P_r' = n_{rr} \times E_r + n_{rw} \times E_r + n_{re} \times E_e = n_{rr} \times E_r \quad \because\ n_{rw},\ n_{re} = 0 \tag{4}$$

수식 2는 데이터베이스의 한 페이지를 읽을 때 플래시 메모리에서 소요되는 이상적인 비용(p_r)을 나타낸다. 수식 3은 데이터베이스에서 n_d번의 읽기 연산이 일어날 때 최적의 비용(P_r)를 의미한다. 하지만 플래시 메모리는 FTL에 따라서 추가적인 비용까지 고려해야하기 때문에 플래시 메모리에서 실제 소요된 비용은 수식 4와 같다. 하지만 읽기는 추가적인 쓰기나 소거가 읽어나지 않기 때문에 수식 4와 같은 식이 도출된다.

$$\lambda = \frac{P_r'}{P_r} = \frac{n_{rr} \times E_r}{n_d \times p_r} = \frac{\frac{n_{rr}}{n_d} \times E_r}{p_r} \tag{5}$$

위 수식들을 이용해서 수식 5의 최적 비용에 대한 실제 비용의 비(λ)를 구할 수 있다. 즉, λ는 데이터베이스의 페이지 읽기 연산을 처리하는데 있어서 실제로 발생한 비용을 예상 비용으로 나누어 그 비율을 나타낸 것이다.

읽기와 마찬가지로 쓰기에 대한 추가적인 비용의 비(μ)를 구할 수 있다. 수식 8에서 플래시 메모리에 실제 소요된 비용은 FTL에 따라 추가적인 읽기, 쓰기와 소거를 포함하게 된다.

$$p_w = k \times E_w \tag{6}$$

$$P_w = \sum_{i=1}^{n} p_{w_i} = k \times n_{dw} \times E_w \tag{7}$$

$$P_w' = n_{wr} \times E_r + n_{ww} \times E_w + n_{we} \times E_e \tag{8}$$

$$\mu = \frac{P_w'}{P_w} = \frac{n_{wr} \times E_r + n_{ww} \times E_w + n_{we} \times E_e}{n_{dw} \times p_w} = \frac{\frac{n_{wr}}{n_{dw}} \times E_r + \frac{n_{ww}}{n_{dw}} \times E_w + \frac{n_{we}}{n_{dw}} \times E_e}{p_w} \tag{9}$$

이때 λ와 μ는 p_r과 p_w의 영향을 받는다. p_r과 p_w값은 칩 인터리빙에 따라 달라지는 값으로, 하드웨어 성능을 고려한 값이 된다. 즉, λ와 μ는 데이터베이스의 한 페이지를 I/O하는데 플래시 메모리에서 드는 비용을 비율로 나타낸 값이다. 표 2와 3은 칩 인터리빙이 1일 때 각 FTL에서의 λ와 μ를 나타낸 것이다. 이는 TPC-A 벤치마크를 MySQL[7]에서 동작시켜 얻어낸 트레이스 A를 실험 시스템의 FTL 시뮬레이터에서 수행하여 각 FTL에 대한 평균적인 λ와 μ를 구한다.

<표 2> 플래시 메모리에서의 λ와 μ (CIL=1)

FTL	λ	μ
복사블록 기법	1.66	17.86
여유공간 기법	34.21	23.7
로그블록 기법	1.01	10.29

5. 플래시 메모리에서 전력 기반 조인 비용 모델

디스크 기반 비용 모델은 I/O 횟수를 토대로 정의하지만, 임베디드 DBMS에서의 플래시 기반 비용 모델은 읽기, 쓰기와 소거에 따른 전력 소모에 따라서 정의해야 한다. 질의 최적화를 위한 가장 큰 비용은 조인 연산의 비용

이라고 할 수 있다. 따라서 조인 연산의 플래시 기반 비용은 새롭게 정의할 필요가 있다. 이런 비용 모델을 수식화하기 위해 다음과 같이 변수를 정의한다.

E_{disk}	:	디스크 기반 비용
E_{flash}	:	플래시 메모리에서 전력 기반 비용
E_{CPU}	:	질의 처리하는데 필요한 CPU 에너지
B	:	데이터베이스 페이지의 크기(bytes)
M	:	버퍼의 크기(데이터베이스 페이지의 수)
b_r, b_s	:	r, s 릴레이션의 데이터베이스 페이지 수
n_r, n_s	:	r, s 릴레이션의 데이터베이스 레코드 수

이 때 디스크 기반 비용 모델에 따라서 플래시 메모리에서 전력기반 비용 모델을 정의하기 위해 플래시 메모리에서 데이터베이스 페이지를 I/O하는 비용을 계산할 필요가 있다.

E_{rB}	:	플래시 메모리로부터 데이터베이스 페이지를 읽을 때 드는 에너지
E_{wB}	:	플래시 메모리로 데이터베이스 페이지를 쓸 때 드는 에너지

위와 같이 정의할 때 하나의 데이터베이스 페이지를 플래시 메모리로 읽고 쓰는 비용은 수식 10과 같다. 디스크의 경우 두 비용이 같지만 플래시 메모리의 특성상 연산의 비동기적인 특징을 가지고 있기 때문에 수식 10이 도출된다.

$$E_{rB} = k \times p_r = k \times \lambda \times E_r \quad (10)$$
$$E_{wB} = k \times p_w = k \times \mu \times E_w$$

위의 변수를 이용하여 대표적인 네 가지 조인방법인 블록 중첩 반복 조인, 색인된 중첩 반복 조인, 합병 조인과 해시 조인의 플래시에서 전력 기반 비용 모델을 제시한다.

5.1. 블록 중첩 반복 조인(bnlj, block nested-loop join)

$$E_{disk}(bnlj) = b_r \times b_s + b_r + E_{CPU} \quad (11)$$
$$(\text{if } M > b_r \text{ or } M > b_s, \text{then } E_{disk}(bnlj) = b_r + b_s + E_{CPU})$$

$$E_{flash}(bnlj) = E_{rB} \times (b_r \times b_s + b_r) + E_{CPU} \quad (12)$$
$$(\text{if } M > b_r \text{ or } M > b_s,$$
$$\text{then } E_{flash}(bnlj) = E_{rB} \times (b_r + b_s) + E_{CPU})$$

수식 11은 블록 중첩 반복 조인의 디스크 기반 비용 모델을 나타낸다. 수식 12는 플래시 메모리에서 전력 기반 비용 모델을 의미한다. 블록 중첩 반복 조인은 $b_r \times b_s + b_r$번의 읽기 연산이 필요하다. 따라서 플래시 메모리에서 전력 기반 비용은 $b_r \times b_s + b_r$번의 데이터베이스 페이지 당 몇 개의 플래시 메모리 페이지를 읽는지를 계산하여 정의한다. 따라서 수식 12처럼 디스크 기반 비용 모델에 E_{rB}를 곱해서 구할 수 있다.

5.2. 색인된 중첩 반복 조인(inlj, indexed nested-loop join)

$$E_{disk}(inlj) = b_r + n_r * d_s + E_{CPU}, \; d_s = \log_{f_B} n_s \quad (13)$$

$$E_{flash}(inlj) = E_{rB} \times (b_r + n_r \times d_s) + E_{CPU} \quad (14)$$

색인된 중첩 반복 조인은 s 릴레이션에 색인이 존재한다고 가정한다. 수식 13은 외부 릴레이션인 r 릴레이션을 읽고(b_r), r 릴레이션의 하나의 레코드에 해당하는 조인 결과를 찾기 위해 색인을 검색하는 디스크 기반 비용이다. 조인의 결과를 찾는 비용은 s 릴레이션에 구축된 B+트리의 깊이(d_s)만큼 찾으므로 $n_r \times d_s$이다. 플래시 메모리에서 전력 기반 비용인 수식 14의 비용은 수식 13에 E_{rB}를 곱한 것이다.

5.3. 합병 조인(mj, merge join)

$$E_{disk}(mj) = b_r + b_s + E_{sort-disk} + E_{CPU} \quad (15)$$
$$E_{sort-disk} = 2 \times b_r \times \left(\left\lceil \log_{M-1} \frac{b_r}{M} \right\rceil + 1 \right) + 2 \times b_s \times \left(\left\lceil \log_{M-1} \frac{b_s}{M} \right\rceil + 1 \right) + E'_{CPU}$$

$$E_{flash}(mj) = E_{rB} \times (b_r + b_s) + E_{sort-flash} + E_{CPU} \quad (16)$$
$$E_{sort-flash} = (E_{rB} + E_{wB}) \times b_r \times \left(\left\lceil \log_{M-1} \frac{b_r}{M} + 1 \right\rceil \right) + (E_{rB} + E_{wB}) \times b_s \times \left(\left\lceil \log_{M-1} \frac{b_r}{M} + 1 \right\rceil \right) + E'_{CPU}$$

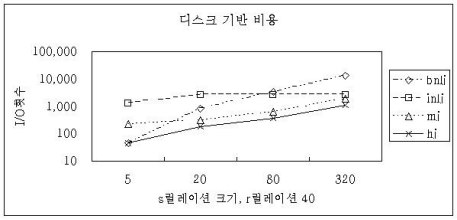
수식 15의 합병 조인에 대한 디스크 기반 비용 모델은 r과 s 릴레이션을 정렬하는 비용($E_{sort-disk}$)과 정렬된 결과를 조인하기위해서 읽는 비용으로 이루어진다. $E_{sort-disk}$는 런을 만들기 위해 릴레이션을 읽고 쓰는 비용과 $\lceil \log_{M-1}(b_r/M) \rceil$의 단계만큼 합병 정렬(merge sort)하는데 읽고 쓰는 비용을 더해서 구한다. 따라서 합병 정렬에 대한 플래시 메모리에서의 전력 기반 비용($E_{sort-flash}$)은 우선 런을 생성하는데 드는 비용을 읽기와 쓰기에 대한 비용으로 나누어 구한다. 따라서 $E_{sort-flash}$는 런을 생성하는 비용과 합병 정렬에서 각 단계에 대한 읽기와 쓰기에 대한 비용을 더해서 구할 수 있다. 마지막으로 플래시 메모리에서 두 릴레이션을 조인하기 위해서 정렬된 결과들을 읽는 비용을 포함해서 수식 16과 같이 정의한다.

5.4. 해시 조인(hj, hash join)

$$E_{disk}(hj) = b_r + b_s + E_{partition-disk} + E_{CPU} \quad (17)$$
$$E_{partition-disk} = 2 \times (b_r + b_s) \times (\lceil \log_{M-1} b_s \rceil - 1) + E'_{CPU}$$

$$E_{flash}(hj) = E_{rB} \times (b_r + b_s) + E_{partition-flash} + E_{CPU} \quad (18)$$
$$E_{partition-flash} = (E_{rB} \times b_r + E_{wB} \times b_s) \times (\lceil \log_{M-1} b_s \rceil - 1) + E'_{CPU}$$

수식 17은 합병 조인에서처럼 해시를 구축하는 비용($E_{partition-disk}$)과 해싱(hashing)된 결과를 조인하기 위해서 읽는 비용을 합하여 구한다. 릴레이션의 크기가 커서 재귀 분할(recursive partitioning)이 필요한 경우 분할의 각 단계에서 분할 영역의 크기를 M-1의 비율로 줄이게 된다. 따라서 분할에 필요한 단계의 수는 $\lceil \log_{M-1} b_r - 1 \rceil$이 된다. $E_{partition-disk}$은 재귀 분할의 각 단계에서 릴레이션의 모든 데이터베이스 페이지를 읽고 쓰는 비용을 말한다. 분할 단계에서 비용($E_{partition-disk}$)은 읽기와 쓰기에 대해서 E_{rB}와 E_{wB}를 각각 곱해서 구할 수 있다. 그리고 수식 18의 해시 조인을 위한 플래시 기반 전체 비용은 해싱된 결과를 읽는 비용을 더해서 구한다. 이와 같이 플래시 기반 비용 모델은 수식 18와 같이 도출된다. 총 비용은 이렇게 해싱을 통해서 얻은 결과를 조인하기 위해서 읽는 비용을 포함한다.

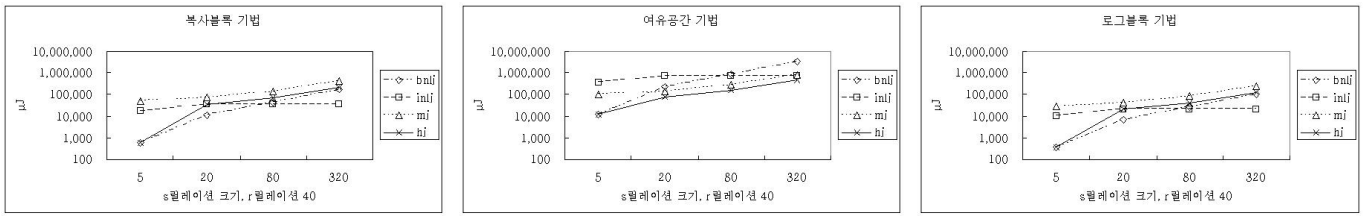


(그림 2) 디스크 기반 비용

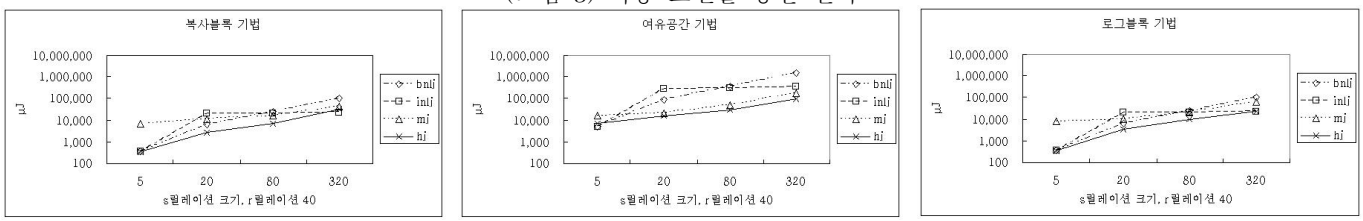
6. 실험

6.1. 실험 환경

본 논문에서는 위의 비용 모델을 검증하기 위해서 디



(그림 3) 비용 모델을 통한 결과



(그림 4) 플래시 임베디드 보드에서의 결과

스크와 플래시 기반 비용 모델을 표 3의 실험 데이터를 토대로 비용을 측정 및 실험한다.

<표 3> 실험 데이터

변수	데이터	단위
b_r	40	8KB
b_s	5, 20, 80, 320	8KB
M	20	8KB
S_r	256	바이트
조인알고리즘	bnlj , inlj, mj, hj	
FTL알고리즘	복사블록 기법, 여유공간 기법, 로그블록기법	
플래시메모리 크기	r, s 릴레이션을 합한 크기의 1.25배	

6.2. 비용 모델을 기반으로 한 예측 결과

그림 2는 디스크 기반 비용 모델에 따른 예측 결과이다. x축은 b_s이고 y축은 디스크 I/O 횟수를 나타낸다. 그림 2에서 b_s가 320 블록인 경우 해시 조인이 작은 비용을 가지며, 320 블록을 초과할 경우 색인된 중첩 반복 조인이 작은 비용을 가진다. 이런 디스크 기반 비용 모델을 플래시 기반으로 전환할 경우 그림 3의 결과로 예측될 수 있다. 이때 그림 3의 결과는 표 2의 TPC-A 벤치마킹을 통해 얻은 λ와 μ를 통해 계산된 결과이다. 그림 3은 각 FTL 알고리즘에 따른 조인 알고리즘별로 전력 기반 비용을 나타낸다. 플래시 기반 비용모델의 경우 여유공간 기법을 제외하고 b_s가 80 블록일 때부터 색인된 중첩 반복 조인이 작은 비용을 보인다. 하지만 여유공간 기법의 경우 λ가 매우 크기 때문에 색인된 중첩 반복 조인이 해시 조인보다 좋지 못한 결과를 보인다.

6.3. FTL 시뮬레이터와 플래시 임베디드 보드에서 결과

<표 4> FTL 시뮬레이션 결과에서의 λ와 μ (CIL=1)

FTL	λ	μ
복사블록 기법	2.29	3.05
여유공간 기법	29.11	5.23
로그블록 기법	1.00	3.47

표 4는 FTL 시뮬레이션에서 발생한 읽기, 쓰기와 소거의 횟수를 기반으로 하여 λ와 μ의 평균으로 계산한 결과이다. 이를 통해 플래시 임베디드 보드를 통해 얻은 결과는 그림 4와 같다. 비용 모델을 통해 계산된 것과 플래시 임베디드 보드에서의 결과는 유사한 결과를 보인다. 하지만 색인된 중첩 반복 조인은 b_s가 M보다 훨씬 작기 때문에 B+-트리와 s 릴레이션이 모두 버퍼에 올라오기 때문에 계산된 값과 다른 결과를 보인다.

7. 결론 및 향후과제

최근 이동성이 중요한 요소로 차지하는 기기들이 등장하면서 플래시 메모리가 각광을 받고 있다. 보조기억장치로 디스크보다 많은 장점을 가지고 있다. 따라서 DBMS도 플래시 메모리를 저장 장치로 사용할 것이라고 예상된다. 하지만 휴대용 기기들은 제한된 자원을 가지며 전력에 의존적이다. 따라서 디스크와 다른 특성을 가진 플래시 메모리에서의 전력 기반 비용 모델이 필요하며 질의 최적화가 필수적이다.

앞으로 본 논문에서 다른 조인 질의뿐 아니라 다른 질의에 대해서도 비용 모델을 세우고 질의 최적화를 할 필요가 있다. 또한 각 FTL 알고리즘에 따른 λ와 μ를 제공받지 못하면 DBMS는 λ와 μ를 계산하여 자신의 플래시 메모리에 맞는 비용을 예상해야 한다. 따라서 이 λ와 μ는 점진적으로 수정해 나아가며 적당한 값으로 맞추어 나가는 방법에 대해서도 연구해야겠다.

참고문헌

[1]박원주, 박성환, 박상원, 윈도우즈 기반 플래시 메모리의 플래시 변환 계층 알고리즘 성능 분석, 한국정보과학회, 정보과학회논문지 : 컴퓨팅의 실제, Vol. 13, 2007. 11, pp. 213~225

[2]김도윤, 박상원, KM-평준화: NAND 플래시 메모리를 위한 레벨 기반 소거 횟수 평준화 기법, 한국정보과학회, 한국정보과학회 학술발표논문집 한국정보과학회 2007 한국컴퓨터종합학술대회 논문집(B), Vol. 34, 2007. 6, pp. 321~326

[3]Amir Ban. Flash file system optimized for page-mode flash technologies, 1999. US Patent, no. 5,937,425.

[4]Takayuki Shinohara. Flash memory card with block memory address arrangement, 1999. US Patent, no. 5,905,993.

[5]Jesung Kim, Jong Min Kim, Sam H. Noh, Sang Lyul Min, and Yookun Cho. A space-efficient flash translation layer for compact flash systems. IEEE Transactions on Consumer Electronics, 48(2), 2002.

[6]박성환, 장주연, 서영주, 박원주, 박상원, 플래시 변환 계층에 대한 TPC-C 벤치마크를 통한 성능분석, 한국정보과학회, 한국정보과학회 학술발표논문집 한국정보과학회 2007 한국컴퓨터종합학술대회 논문집(A), Vol. 34, pp. 21 ~ 22, 2007. 6.