



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년04월05일
(11) 등록번호 10-1027112
(24) 등록일자 2011년03월29일

(51) Int. Cl.

G11C 16/10 (2006.01) G11C 16/16 (2006.01)

G11C 16/06 (2006.01)

(21) 출원번호 10-2008-0062735

(22) 출원일자 2008년06월30일

심사청구일자 2008년06월30일

(65) 공개번호 10-2010-0002731

(43) 공개일자 2010년01월07일

(56) 선행기술조사문헌

Mei-Ling Chiang et al., "Using Data Clustering to Improve Cleaning Performance for Flash Memory", Softw. Pract. Exper., 29(3), pages 267-290, 1999.

Jeong-Uk Kang et al., "A Superblock-based Flash Translation Layer for NAND Flash Memory", Proceedings of the 6th ACM & IEEE International conference on Embedded software, pages 161-170, Seoul, Korea, 0

전체 청구항 수 : 총 13 항

(73) 특허권자

한국산업기술대학교산학협력단

경기 시흥시 정왕동 2121 한국산업기술대학교

(72) 발명자

장지웅

서울특별시 송파구

오륜동울림픽션수촌아파트(301~328동)320동 701호

박세미

서울특별시 성동구 사근동 176-1 307호

(뒷면에 계속)

(74) 대리인

특허법인명문

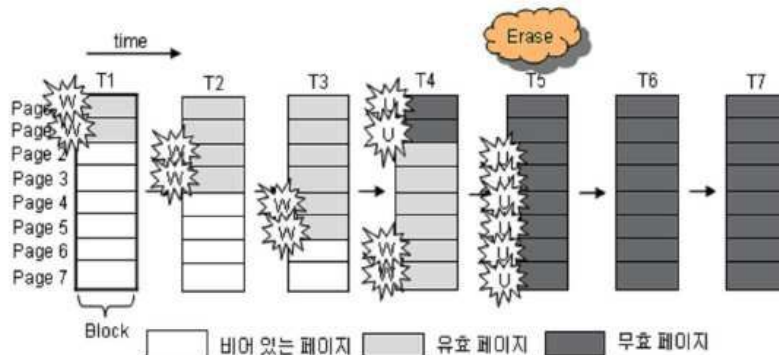
심사관 : 손윤식

(54) 플래시 메모리의 클러스터링 장치 및 그 클러스터링 방법

(57) 요약

플래시 메모리의 클러스터링 장치 및 그 클러스터링 방법이 개시된다. 본 발명에 따른 플래시 메모리의 클러스터링 장치는, 업데이트 시점이 유사한 페이지들끼리 모아 동일한 블록에 쓰기 연산을 실행하는 것을 특징으로 한다. 이로써, 플래시 메모리의 쓰기 성능이 향상되고, 플래시 메모리의 수명을 연장시킬 수 있게 된다.

대표도 - 도2



(72) 발명자

배덕호

서울특별시 성동구 사근동 186-4 201호

김상욱

서울특별시 성동구 행당2동 한진아파트 106-301

특허청구의 범위

청구항 1

플래시 메모리의 데이터 클러스터링 장치에 있어서,

페이지의 예상 업데이트 시점이 기 설정된 시간 범위에 있는 페이지들을 모아 동일한 블록에 쓰기 연산을 실행하며,

상기 페이지의 예상 업데이트 시점은 상기 페이지를 구성하는 다수의 데이터에 대응되는 다수의 업데이트 주기 중 최단주기를 기반으로 설정되는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 장치.

청구항 2

제 1항에 있어서,

동일한 페이지 내에 서로 다른 업데이트 주기를 갖는 데이터들이 있는 경우, 그 페이지의 업데이트 주기는 가장 짧은 업데이트 주기를 갖는 데이터의 주기와 동일하게 되는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 장치.

청구항 3

제 1항에 있어서,

쓰기 연산이 요청될 경우, 그룹 관리 테이블에 기초하여 데이터를 저장할 페이지를 할당하는 할당부(allocator); 및

소거 연산이 요청될 경우, 소거 정책(erase policy)에 의해 선택된 블록을 소거하는 소거부(cleaner)를 포함하는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 장치.

청구항 4

제 3항에 있어서,

상기 그룹 관리 테이블은 일정 개수의 그룹 엔트리를 관리하고, 그룹 업데이트 타임과 그룹에 해당하는 블록들의 리스트로 구성되는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 장치.

청구항 5

제 4항에 있어서,

상기 그룹은 업데이트 타임이 상기 그룹 각각에 대해 기 설정된 시간 범위에 있는 페이지들이 저장된 블록들의 집합이며, 하나의 그룹은 하나의 그룹 관리 테이블 엔트리와 대응되는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 장치.

청구항 6

제 5항에 있어서,

특정 페이지 P_j 에 업데이트가 발생할 경우, 상기 P_j 는 다음의 식을 만족하는 그룹 엔트리 E_i 에 저장되는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 장치:

$$E_{i-1}.updatetime < P_j.updatetime \leq E_i.updatetime$$

청구항 7

플래시 메모리의 데이터 클러스터링 방법에 있어서,

데이터 쓰기가 요청되는 경우, 새로운 데이터 쓰기인지 이전 데이터의 업데이트 인지 확인하는 단계;

상기 확인하는 단계에 의해 이전 데이터의 업데이트인 것으로 확인되면, 상기 이전 데이터가 저장된 페이지를 무효화하고 해당 페이지의 업데이트 타임을 계산하는 단계;

계산된 상기 업데이트 타임에 해당하는 그룹이 존재할 경우, 상기 그룹에 빈 페이지가 존재하는지의 여부를 확인하는 단계; 및

빈 페이지가 존재할 경우에 상기 빈 페이지에 업데이트된 데이터를 저장하는 단계를 포함하며,

상기 페이지의 예상 업데이트 타임은 상기 페이지를 구성하는 다수의 데이터에 대응되는 다수의 업데이트 주기 중 최단주기를 기반으로 설정되는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 방법.

청구항 8

제 7항에 있어서,

상기 그룹에 빈 페이지가 없을 경우 새로운 블록을 할당하여 상기 그룹에 추가하는 단계를 더 포함하며,

추가된 상기 새로운 블록의 빈 페이지에 상기 업데이트된 데이터를 저장하는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 방법.

청구항 9

제 7항에 있어서,

계산된 상기 업데이트 타임에 해당하는 그룹 엔트리가 존재하지 않는 경우, 상기 그룹 엔트리에 포함되지 않은 블록의 빈 페이지에 상기 업데이트된 데이터를 저장하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 방법.

청구항 10

제 7항에 있어서,

새로운 데이터 쓰기인 것으로 판단되면, 그룹 엔트리에 포함되지 않은 블록의 빈 페이지에 데이터를 저장하는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 방법.

청구항 11

플래시 메모리의 데이터 클러스터링 방법에 있어서,

데이터 쓰기가 요청되는 경우, 이전 데이터가 저장된 페이지를 무효화하고 해당 페이지의 업데이트 타임을 계산하는 단계; 및

계산된 상기 업데이트 타임에 해당하는 그룹이 존재하는지를 검색하여 해당 그룹이 존재할 경우 상기 해당 그룹에 포함된 페이지들 중 비어있는 페이지에 업데이트된 데이터를 저장하는 단계를 포함하며,

상기 페이지의 예상 업데이트 타임은 상기 페이지를 구성하는 다수의 데이터에 대응되는 다수의 업데이트 주기 중 최단주기를 기반으로 설정되는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 방법.

청구항 12

제 11항에 있어서,

상기 해당 그룹에 빈 페이지가 없을 경우 새로운 블록을 할당하여 상기 해당 그룹에 추가하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 방법.

청구항 13

제 11항에 있어서,

상기 해당 그룹이 존재하지 않는 경우 그룹에 포함되지 않은 블록에 상기 업데이트된 데이터를 저장하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리의 데이터 클러스터링 방법.

명세서

발명의 상세한 설명

기술분야

[0001] 본 발명은 플래시 메모리의 클러스터링 장치 및 그 클러스터링 방법에 관한 것으로서, 보다 상세하게는 블록 복사 비용을 최소화시킬 수 있는 플래시 메모리의 클러스터링 장치 및 그 클러스터링 방법에 관한 것이다.

배경기술

[0002] 플래시 메모리는 저 전력, 비휘발성, 휴대성 등의 특징으로 인해 휴대용 전자기기의 저장 매체로 많이 이용되고 있다. 플래시 메모리는 동일한 크기의 여러 개의 연속적인 블록으로 구성되며, 하나의 블록은 동일한 크기의 여러 개의 연속적인 페이지로 이루어진다. 플래시 메모리에서 수행 가능한 연산은 읽기, 쓰기, 소거의 세 가지이다. 이때, 플래시 메모리에서의 읽기와 쓰기 연산은 페이지 단위로 이루어지며, 소거 연산은 블록 단위로 이루어진다.

[0003] 이와 같은 플래시 메모리는 일반적인 저장 매체와 달리 플래시 메모리만의 고유한 특성을 갖는다. 첫째, 디스크는 읽기 연산과 쓰기 연산 간 수행 속도의 차이가 없는데 반하여, 플래시 메모리는 두 연산 간의 수행 속도의 차이가 매우 크다. 즉, 플래시 메모리의 쓰기 연산은 읽기 연산에 비해 매우 느리다. 둘째, 플래시 메모리에는 일반적인 저장 매체에는 없는 새로운 연산인 소거 연산(erase operation)이 존재한다. 플래시 메모리는 물리적 특성상 저장된 데이터를 직접 갱신(in-place update)하는 것이 불가능하다. 데이터를 갱신하기 위해서는 데이터가 저장된 페이지를 무효화하고, 갱신 사항이 반영된 새로운 페이지를 플래시 메모리의 다른 영역에 기록하는 방법을 사용한다. 이때, 기존의 페이지를 무효화하는 연산이 소거 연산이다. 소거 연산을 수행하기 위해서는 해당 블록 안에 존재하는 유효한 데이터들을 다른 블록으로 복사하여야 하며, 이를 블록 복사(block copy)라고 한다. 이와 같이 소거 연산은 오버헤드가 매우 큰 연산으로 잦은 쓰기 연산에 의해 발생한다. 셋째, 플래시 메모리는 하나의 블록에 수행 가능한 소거 연산의 횟수가 보통 10만 번 이하로 제한되어 있다. 제한된 수만큼 소거 연산을 수행한 블록은 더 이상 사용할 수 없다. 즉, 잦은 소거 연산은 플래시 메모리의 수명을 단축시키는 중요한 요인이 된다. 이와 같은 특성들로 인하여 플래시 메모리 환경에서 쓰기 연산을 감소시키는 것이 매우 중요하다.

[0004] 쓰기 연산을 감소시키는 방법으로 자주 업데이트 되는 데이터들을 동일한 페이지 또는 블록에 저장하는 핫 데이터 클러스터링(hot data clustering) 방법이 있다. 여기서, 핫 데이터는 자주 업데이트가 되는 데이터를 말한다. 핫 데이터가 저장된 페이지는 업데이트가 자주 일어나므로 무효화될 확률이 높다. 그러므로 이러한 페이지들을 동일한 블록에 저장하면 소거 연산 수행 시 해당 블록의 블록 복사 비용을 줄일 수 있다. 이로 인해 쓰기 연산의 횟수가 줄어들고, 소거 연산의 횟수도 줄어들게 되어 전반적인 플래시 메모리의 성능이 향상된다.

[0005] 하지만, 핫 데이터가 모여 있는 블록이라고 할지라도 소거 연산이 요청되었을 때 유효한 페이지(valid page)가 많이 존재하는 경우가 발생한다. 예를 들어, 도 1에 도시한 바와 같이 하나의 블록은 총 8개의 페이지로 구성되어 있으며 페이지의 주기는 3으로 모두 동일하고 핫 데이터는 페이지 단위라고 가정하면, 시간이 지남에 따라 블록에는 페이지가 순차적으로 쓰이지만 T5에서 소거 연산이 요청되었을 때 Page 0부터 Page 2까지 무효한 페이지이며, Page 3부터 Page 6까지는 유효한 페이지이다. 이때, 소거 연산이 수행되면, 블록 내의 절반에 해당하는 유효한 페이지를 다른 블록으로 복사하여야 하기 때문에 블록 복사 비용이 많이 발생하게 된다. 또한, 이와 같은 잦은 블록 복사로 인하여 플래시 메모리의 성능이 저하되며, 소거 연산을 많이 유발하여 플래시 메모리의 수명을 단축시키게 된다.

발명의 내용

해결 하고자하는 과제

[0006] 본 발명은 상기와 같은 문제점을 해결하기 위하여 창안된 것으로서, 플래시 메모리의 쓰기 성능을 향상시키고, 플래시 메모리의 수명을 연장시킬 수 있는 플래시 메모리의 데이터 클러스터링 장치 및 그 클러스터링 방법을 제공하는 것을 목적으로 한다.

과제 해결수단

[0007] 상기의 목적을 달성하기 위한 본 발명에 따른 플래시 메모리의 데이터 클러스터링 장치는, 업데이트 시점이 기 설정된 시간 범위에 있는 페이지들끼리 모아 동일한 블록에 쓰기 연산을 실행하는 것을 특징으로 한다.

[0008] 여기서, 동일한 페이지 내에 서로 다른 업데이트 주기를 갖는 데이터들이 있는 경우, 그 페이지의 업데이트 주기는 가장 짧은 업데이트 주기를 갖는 데이터의 주기와 동일하게 되는 것이 바람직하다.

[0009] 또한, 상기의 플래시 메모리의 데이터 클러스터링 장치는, 쓰기 연산이 요청될 경우, 그룹 관리 테이블에 기초하여 데이터를 저장할 페이지를 할당하는 할당부(allocator); 및 소거 연산이 요청될 경우, 소거 정책(erase policy)에 의해 선택된 블록을 소거하는 소거부(cleaner)를 포함할 수 있다.

[0010] 또한, 상기 그룹 관리 테이블은 일정 개수의 그룹 엔트리를 관리하고, 그룹 업데이트 타임과 그룹에 해당하는 블록들의 리스트로 구성될 수 있다.

[0011] 또한, 상기 그룹은 업데이트 타임이 상기 그룹 각각에 대해 기 설정된 시간 범위에 있는 페이지들이 저장된 블록들의 집합이며, 하나의 그룹은 하나의 그룹 관리 테이블 엔트리와 대응되는 것이 바람직하다.

[0012] 여기서, 특정 페이지 P_j 에 업데이트가 발생할 경우, 상기 P_j 는 다음의 식을 만족하는 그룹 엔트리 E_i 에 저장되는 것이 바람직하다.

[0013]
$$E_{i-1}.updatetime < P_j.updatetime \leq E_i.updatetime$$

[0014] 한편, 상기의 플래시 메모리의 데이터 클러스터링 장치는, 데이터 쓰기가 요청되는 경우, 새로운 데이터 쓰기인지 이전 데이터의 업데이트 인지를 확인하는 단계; 상기 확인단계에 의해 이전 데이터의 업데이트인 것으로 확인되면, 상기 이전 데이터가 저장된 페이지를 무효화하고 해당 페이지의 업데이트 타임을 계산하는 단계; 계산된 상기 업데이트 타임에 해당하는 그룹 엔트리가 존재할 경우, 해당 그룹에 빈 페이지가 존재하는지의 여부를 확인하는 단계; 및 빈 페이지가 존재할 경우에 상기 빈 페이지에 업데이트된 데이터를 저장하는 단계를 포함하는 플래시 메모리의 데이터 클러스터링 방법을 제공한다.

[0015] 여기서, 상기의 플래시 메모리의 데이터 클러스터링 방법은, 상기 그룹 엔트리의 해당 그룹에 빈 페이지가 없을 경우 새로운 블록을 할당하여 상기 해당 그룹에 추가하는 단계를 더 포함할 수 있다. 이 경우, 추가된 상기 새로운 블록의 빈 페이지에 상기 업데이트된 데이터를 저장하는 것이 바람직하다.

[0016] 또한, 상기의 플래시 메모리의 데이터 클러스터링 방법은, 계산된 상기 업데이트 타임에 해당하는 그룹 엔트리가 존재하지 않는 경우, 상기 그룹 엔트리에 포함되지 않은 블록의 빈 페이지에 상기 업데이트된 데이터를 저장하는 단계를 더 포함할 수 있다.

[0017] 여기서, 새로운 데이터 쓰기인 것으로 판단되면, 그룹 엔트리에 포함되지 않은 블록의 빈 페이지에 데이터를 저장하는 것이 바람직하다.

[0018] 한편, 상기의 플래시 메모리의 데이터 클러스터링 방법은, 데이터 쓰기가 요청되는 경우, 이전 데이터가 저장된 페이지를 무효화하고 해당 페이지의 업데이트 타임을 계산하는 단계; 및 계산된 상기 업데이트 타임에 해당하는 그룹이 존재하는지를 검색하며, 해당 그룹이 존재할 경우 상기 해당 그룹에 포함된 페이지들 중 비어있는 페이지에 업데이트된 데이터를 저장하는 단계를 포함하는 제2의 플래시 메모리의 데이터 클러스터링 방법을 제공한다.

[0019] 이와 같은 제2의 플래시 메모리의 데이터 클러스터링 방법은, 상기 해당 그룹에 빈 페이지가 없을 경우 새로운 블록을 할당하여 상기 해당 그룹에 추가하는 단계가 더 포함될 수 있다.

[0020] 또한, 제2의 플래시 메모리의 데이터 클러스터링 방법은, 상기 해당 그룹이 존재하지 않는 경우 그룹에 포함되지 않은 블록에 상기 업데이트된 데이터를 저장하는 단계가 더 포함될 수도 있다.

효과

[0021] 본 발명에 따르면, 플래시 메모리의 쓰기 성능이 향상되며, 플래시 메모리의 수명 또한 연장된다.

발명의 실시를 위한 구체적인 내용

[0022] 이하, 첨부된 도면을 참조하여 본 발명에 따른 플래시 메모리의 클러스터링 장치 및 그 클러스터링 방법을 상세하게 설명한다.

[0023] 도 2는 본 발명에 따른 플래시 메모리의 클러스터링 방법을 설명하기 위해 도시된 도면이다. 한 블록은 총 8개의 페이지로 구성되어 있으며, 모든 페이지는 T5 시간 이전에 업데이트 되었다고 가정한다. 이때, 업데이트 된 각각의 페이지는 업데이트 타임(update time)을 기반으로 클러스터링 된다. 페이지의 업데이트 타임은 해당 페이지에 수정이 발생하게 될 예상 시각을 의미하며 페이지 내의 데이터들의 업데이트 주기 중에서 가장 짧은 주기를 가진 데이터를 기반으로 업데이트 타임을 예측한다. 예를 들어, 도 3에 도시한 바와 같이 한 페이지에 6개의 각각 다른 업데이트 주기를 가진 데이터가 있다고 가정하면, 그 중 제일 작은 업데이트 주기를 가진 데이터 1의 업데이트 주기가 된다. 블록 복사의 비용을 줄이기 위해서는 소거 연산을 수행하는 해당 블록에 존재하는 유효 페이지의 수가 적어야 한다. 이를 위하여 본 발명에서는 업데이트될 예상 시점이 유사한 데이터를 동일한 블록에 저장한다. 즉, 한 페이지가 업데이트되었을 때, 다음 업데이트 타임을 알고 있다면 해당 페이지의 업데이트 타임과 가장 유사한 예상 업데이트 완료 시간을 가지는 블록에 이 페이지를 쓴다. 한 페이지의 예상 업데이트 시간이란, 한 페이지가 업데이트된 후 다음 업데이트가 발생하게 될 예상 시각을 의미한다. 또한, 블록의 업데이트 완료 시간은 해당 블록 내에 저장되어 있는 페이지들이 모두 업데이트된 직후의 시간을 말한다. 따라서, 본 발명에 따른 예상 업데이트 시간 기반 클러스터링은 블록의 업데이트 완료 시간을 바탕으로 페이지의 예상 업데이트 시간을 고려하여 저장하는 방법이다.

[0024] 서로 업데이트 주기는 다르지만 업데이트 시점이 비슷한 페이지들끼리 동일한 블록에 모아쓰게 되면, 그 블록은 업데이트 완료 시간이 지난 후 블록 내의 모든 페이지가 무효화된 상태로 존재하게 된다. 이러한 블록이 T5에서 소거 대상 블록으로 선정된다면 블록 내의 대부분의 페이지가 무효화된 상태로 존재하므로 블록 복사 비용을 현저히 줄일 수 있다.

[0025] 도 4는 본 발명에 따른 플래시 메모리의 클러스터링 장치의 구조를 개략적으로 나타낸 도면이다. 본 발명에 따른 클러스터링 장치(10)는 크게 할당부(allocator)(12) 및 소거부(cleaner)(14)로 구성된다.

[0026] 할당부(12)는 쓰기 연산이 요청되었을 때, 데이터를 저장할 페이지를 할당해주는 역할을 하며, 이를 위해 그룹 관리 테이블(group management table)을 관리한다. 여기서, 그룹 관리 테이블은 업데이트된 페이지들이 저장될 위치를 할당하기 위해 블록들을 그룹 단위로 관리하는 테이블을 말한다.

[0027] 그룹은 업데이트 타임이 유사한 페이지들이 저장된 블록들의 집합으로서, 하나의 그룹은 하나의 그룹 관리 테이블 엔트리와 대응된다. 그룹 관리 테이블은 일정한 개수의 그룹 엔트리를 관리하며, 하나의 그룹 엔트리는 그룹 업데이트 타임과 그룹에 해당하는 블록들의 리스트로 구성된다. 그룹 업데이트 타임은 해당 그룹에 쓰인 페이지들의 업데이트 타임의 최대값을 의미한다.

[0028] 특정 페이지 P_j에 업데이트가 발생할 경우, P_j는 수학적 1을 만족하는 그룹 엔트리 E_i에 저장된다.

수학식 1

$$E_{i-1}.updatetime < P_j.updatetime \leq E_i.updatetime$$

[0029]

[0030] 시간이 지남에 따라 현재시각 이전의 업데이트 타임을 가지는 그룹 엔트리는 테이블에서 삭제되고, 새로운 그룹 엔트리가 생성된다. 이때, 그룹 엔트리의 업데이트 타임은 설정된 업데이트 타임 간격에 의해 자동적으로 부여된다. 업데이트 타임 간격은 데이터의 성격에 따라 사용자가 부여하는 값이다. 업데이트 타임 간격을 넓게 설정하면 동일한 그룹에 속하게 될 페이지간의 업데이트 타임 간격이 증가하여, 소거 연산 수행 시 해당 블록에 속한 유효 페이지들이 많이 존재할 가능성이 높다. 이로 인해 블록 복사 비용이 증가한다. 반면에 업데이트 타임 간격을 좁게 설정하면 그룹 업데이트 타임이 세분화 되어, 동일한 그룹에 속하게 될 페이지들의 수가 줄어들게 된다. 이로 인해, 빈 페이지가 많이 남아 있는 채로 소거되는 블록이 많아지게 되므로 적절한 값을 설정하는 것이 중요하다.

[0031]

소거부(14)는 소거 연산이 요청될 때 소거 정책(Erase policy)에 의해 선택된 블록을 소거 한다. 특정 블록에 무효 페이지가 많을수록, 유효 페이지가 적은 수록, 소거 횟수가 적을수록 선택될 확률이 높아진다.

[0032]

도 5는 본 발명에 따른 플래시 메모리의 클러스터링 방법의 전체적인 쓰기 과정을 나타낸다. 도면을 참조하여 본 발명에 따른 플래시 메모리의 클러스터링 방법을 보다 상세하게 설명한다.

[0033]

데이터 쓰기가 요청되면, 클러스터링 장치(10)는 그룹 관리 테이블을 이용하여 해당 페이지의 업데이트 타임과 각 그룹의 업데이트 시간 값을 비교하여 그 요청된 대상 데이터가 새로운 데이터 쓰기인지 이전 데이터의 업데이트인지 확인한다(S101). 만약 새로운 데이터의 쓰기 일 경우 업데이트 타임을 예측할 수 없으므로 그룹에 포함되어 있지 않은 블록에 저장한다(S103, S105). 이전 데이터의 업데이트의 경우에는 먼저 이전 데이터가 저장된 페이지를 무효화하고 해당 페이지의 업데이트 타임을 계산한다(S107). 또한, 할당부(12)는 그룹 관리 테이블에 계산된 업데이트 타임에 해당하는 그룹 엔트리가 존재하는지를 검색하며, 계산된 업데이트 타임에 해당하는 그룹 엔트리가 존재할 경우, 해당 그룹에 빈 페이지가 존재하는지 여부를 확인한다(S109, S111). 해당 페이지에 빈 페이지가 존재할 경우 해당 페이지에 업데이트된 데이터를 저장한다(S113). 해당 그룹에 빈 페이지가 없을 경우, 새로운 블록을 할당하여 할당된 블록을 그룹에 추가한 후, 업데이트 된 데이터를 저장한다(S115, S117). 계산된 업데이트 타임에 해당하는 그룹이 없을 경우, 그룹에 포함되어 있지 않은 블록에 저장한다.

[0034]

도 6은 쓰기 요청된 페이지의 업데이트 타임이 그룹 관리 테이블내의 엔트리에 해당 될 때 쓰는 과정을 나타낸 도면이다. 여기서, 테이블의 엔트리 개수는 6개이고 각 그룹 엔트리간의 업데이트 타임 간격은 10분이라고 가정한다. 이때, 쓰기가 요청이 온 페이지의 업데이트 타임이 1시 18분일 때 그룹의 업데이트 타임이 1시 20분인 테이블의 두 번째 엔트리에 해당 페이지가 할당 된다.

[0035]

그룹 관리 테이블에서는 그룹의 개수와 그룹 업데이트 타임의 간격을 사용자가 임의로 설정해 주어야 한다. 전체 플래시 메모리의 데이터를 그룹으로 관리하는 것은 오버헤드가 너무 크고, 업데이트가 거의 발생하지 않은 데이터의 경우 그룹으로 관리하는 것은 의미가 없다. 그러므로 일정 기간 내에 업데이트될 데이터를 유한개의 그룹으로 관리하는 것이 바람직하다. 이때, 업데이트가 일어나지 않는 데이터와 업데이트가 되더라도 예상 업데이트 타임 값이 커 그룹에 속하지 못한 데이터는 일반적인 쓰기 연산 방법을 따른다. 본 발명에서는 실험을 통해서 그룹의 개수에 따른 플래시 메모리의 클러스터링 방법의 성능 변화를 측정하였다. 그룹 업데이트 타임의 간격은 각 그룹의 업데이트 타임을 설정하는 값으로 그룹 업데이트 타임 간격을 넓게 설정하면 동일한 그룹에 속하게 될 페이지 간의 업데이트 타임 간격이 증가하여, 소거 연산 수행 시 해당 블록에 속한 유효 페이지들이 많이 존재할 가능성이 높다. 이로 인해 블록 복사 비용이 증가한다. 반면에 그룹 업데이트 타임 간격을 좁게 설정하면 그룹 업데이트 타임이 세분화되어, 동일한 그룹에 속하게 될 페이지들의 수가 줄어들게 된다. 이로 인해, 빈 페이지가 많이 남아있는 채로 소거되는 블록이 많아지게 되므로 적절한 값을 설정하는 것이 중요하다.

[0036]

본 실험에서는 플래시 메모리의 용량을 1G NAND 플래시 메모리를 기준으로 설정하였으며, 소거 대상 블록 선택 방법(cleaning policy)은 CAT를 적용하였다. 실험에 사용되는 데이터는 일정한 업데이트 타임을 가지는 데이터를 기반으로 하였다.

[0037]

다양한 데이터를 대상으로 성능을 측정하기 위하여 4가지 로컬리티가 있는 데이터(locality data)와 랜덤 데이

터(random data)를 대상으로 한다. 로컬리티가 있는 데이터 90/10, 70/10, 50/10, 30/10과 10/10의 랜덤 데이터를 사용한다. 90/10은 전체 10%의 데이터에 90%의 연산이 집중되고, 90% 데이터에 나머지 10%의 연산이 집중되는 것을 말하며, 로컬리티 70/10은 전체 10%의 데이터에 70%의 연산이 집중되고, 90%의 데이터에 나머지 30%의 연산이 집중되는 것을 말한다.

[0038] 로컬리티가 있는 데이터를 생성하기 위하여 지수 확률 변수(exponential random variable)의 확률 밀도 함수를 이용하였다. 확률 밀도 함수는 수학식 2와 같다.

수학식 2

$$f_x(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

[0039] 수학식 2의 지수 확률 변수의 확률 밀도 함수를 적분하여 λ 값을 찾아 각 페이지마다 빈도(frequency)를 설정하고 빈도의 역수의 크기를 조정하여 로컬리티가 있는 데이터를 생성하였다. 도 7은 1기가바이트 환경에서 페이지 번호에 따른 빈도수를 나타낸다.

[0041] 본 실험에서 플래시 메모리의 초기화를 위하여 데이터를 순차적으로 전체 플래시 메모리 크기의 90%를 저장한다. 그리고 플래시 메모리의 최소 2%를 빈 공간으로 유지한다. 실험 수행 시간은 페이지가 40,000,000회 업데이트된 후를 기준으로 한다. 실험의 성능 척도로 페이지가 40,000,000회 업데이트된 후, 소거 연산의 총 횟수와 평균 블록 복사 비용을 측정한다. 또한, 쓰기 연산과 소거 연산의 횟수의 가중치(쓰기:1, 소거:10)를 고려하여 수학식 3의 소거 비용을 이용한다.

수학식 3

[0042] 소거비용 = (평균블록복사비용 + 10) X 소거연산의 횟수

[0043] 또한, 본 발명에 따른 플래시 메모리의 클러스터링 방법과 DAC(Dynamic data Clustering)의 성능을 비교하기 위한 실험으로, 플래시 메모리의 기존 데이터 클러스터링 방법인 핫 데이터 클러스터링 방법과 예상 업데이트 타임 기반 클러스터링 방법의 성능을 측정한다. 본 발명에 따른 플래시 메모리의 클러스터링 방법의 비교 대상으로서 핫 데이터 클러스터링 방법 중 DAC를 채택한다. DAC는 리전의 개수와 시간 임계값에 따라서 성능 차이가 크기 때문에 총 6개의 환경을 설정하였다. 매개 변수 설정은 표 1을 통해 알 수 있다. DAC1의 방법은 리전을 구분하지 않으므로 클러스터링을 하지 않은 일반적인 플래시 메모리의 쓰기 방법을 대표한다. 예상 업데이트 타임 기반 클러스터링 방법(UTC)은 테이블의 크기를 16, 그룹의 업데이트 타임 간격은 20으로 설정하였다. 그 결과는 도 8 내지 도 10을 통해 알 수 있다.

표 1

| DAC 매개 변수 | DAC1 | DAC2 | DAC3 | DAC4 | DAC5 | DAC6 |
|--------------|------|------|------|------|------|------|
| 리전의 개수 | 1 | 4 | 2 | 4 | 8 | 8 |
| 시간 임계값 | - | 30 | 120 | 120 | 120 | 60 |

[0044] 도면에 나타난 바와 같이 본 발명에 따른 클러스터링 방법이 DAC에 비하여 평균 블록 복사 비용은 최대 54%, 소거 연산의 횟수는 최대 59%, 소거 비용은 최대 77%의 성능 향상을 보였다. 또한, DAC는 로컬리티에 따라서 성능 차이가 많은 반면에 본 발명에 따른 클러스터링 방법은 로컬리티에 따른 성능 차이가 작음을 알 수 있다. 로컬리티 10/10인 모든 데이터의 주기가 DAC2의 시간 임계값보다 크기 때문에 DAC2의 리전의 개수 지정과 상관없이 DAC1의 결과와 같다. 또한, DAC3, DAC4, DAC5는 시간 임계값이 같으므로 전반적으로 비슷한 패턴을 보인다.

[0046] 환경 변수에 따른 예상 업데이트 타임 기반 클러스터링 방법의 성능 변화를 측정하기 위한 실험에서는 다양한 실험을 통해 정확한 성능을 측정하기 위하여 환경 변수를 두어 성능 변화를 측정하였다. 표 2는 실험에 사용한 그룹 업데이트 타임(갱신 시간)의 간격과 그룹의 개수의 값을 정리한 것이다.

표 2

| 환경 변수 | 환경 변수 값 |
|-------------|--------------------------------|
| 그룹 갱신 시간 간격 | 1, 5, 10, 20, 40, 80, 160, 320 |
| 그룹의 개수 | 1, 4, 8, 16, 32, 64, 128 |

[0047]

[0048]

표 2에서 그룹 갱신 시간 간격 20 및 그룹의 개수 16는 그 매개 변수를 대표하는 값으로 특정한 하나의 매개 변수 값을 변화시켜 가는 동안 다른 환경 변수 값은 그 대표 값으로 고정된다. 예를 들어, 그룹 업데이트 타임 간격을 1, 5, 10, 20, 40, 80, 160, 320으로 변화시키며 실험을 할 때 그룹의 개수는 16으로 고정된다. 본 실험에서 데이터는 로컬리티 90/10인 데이터를 대상으로 한다.

[0049]

첫 번째 실험은 그룹의 업데이트 타임 간격을 고정하고 그룹의 개수의 변화에 따른 예상 업데이트 타임 기반 클러스터링 방법의 성능 변화를 측정한다. 실험 결과는 도 11 내지 도 13을 통해 알 수 있다. 여기서, 그룹의 개수가 16개까지 급격한 성능 차이를 보이지만 그 이후로 성능 차이가 크지 않다. 그룹의 개수가 많을수록 그룹 관리 테이블에서 관리할 수 있는 블록의 수가 많으므로 클러스터링의 효과가 높을 것이라고 예상하였지만, 예상 업데이트 타임이 짧은 일부 데이터를 관리하는 것만으로도 충분히 큰 클러스터링 효과를 보인다. 이로부터 그룹의 개수에 따라 성능의 차이가 크지 않음을 알 수 있다.

[0050]

두 번째 실험은 그룹의 개수를 고정하고 그룹의 업데이트 타임 간격의 변화에 따른 예상 업데이트 타임 클러스터링 방법의 성능 변화를 측정한다. 실험의 결과는 도 14 내지 도 16을 통해 알 수 있다. 그룹 업데이트 타임 간격이 20과 40에서 성능이 가장 좋으며, 그 이후 값이 커질수록 성능이 완만하게 증가하는 것을 볼 수 있다. 이것은 그룹 업데이트 타임 간격이 큰 값을 가지면 최악의 경우에는 클러스터링 하지 않은 것과 같이 때문이다. 그룹 업데이트 타임 간격은 데이터의 로컬리티에 따라 사용자가 적절한 값을 지정해주는 것이 중요하다.

[0051]

이상에서는 본 발명의 바람직한 실시 예에 대해서 도시하고 설명하였으나, 본 발명은 상술한 특정의 실시 예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시가 가능한 것은 물론이고, 그와 같은 변경은 청구범위 기재의 범위 내에 있게 된다.

도면의 간단한 설명

[0052]

도 1은 핫 데이터 클러스터링에서 블록 복사가 발생하는 예를 나타낸 도면이다.

[0053]

도 2는 본 발명에 따른 플래시 메모리의 클러스터링 방법을 설명하기 위해 도시된 도면이다.

[0054]

도 3은 서로 다른 업데이트 주기를 가진 데이터들을 구비한 페이지의 예를 나타낸 도면이다.

[0055]

도 4는 본 발명에 따른 플래시 메모리의 클러스터링 장치의 구조를 나타낸 도면이다.

[0056]

도 5는 본 발명에 따른 플래시 메모리의 클러스터링 방법의 전체적인 쓰기 과정을 나타낸 흐름도이다.

[0057]

도 6은 쓰기 요청된 페이지의 업데이트 타임이 그룹 관리 테이블내의 엔트리에 해당 될 때 쓰는 과정을 나타낸 도면이다.

[0058]

도 7은 로컬리티가 있는 데이터의 빈도수를 나타낸 그래프이다.

[0059]

도 8은 로컬리티에 따른 평균 블록 복사 비용을 나타낸 그래프이다.

[0060]

도 9는 로컬리티에 따른 소거 연산의 총 횟수를 나타낸 그래프이다.

[0061]

도 10은 로컬리티에 따른 소거 비용을 나타낸 소거 비용을 나타낸 그래프이다.

[0062]

도 11은 그룹의 개수의 변화에 따른 평균 블록 복사 비용을 나타낸 그래프이다.

[0063]

도 12는 그룹의 개수에 따른 소거 연산의 총 횟수를 나타낸 그래프이다.

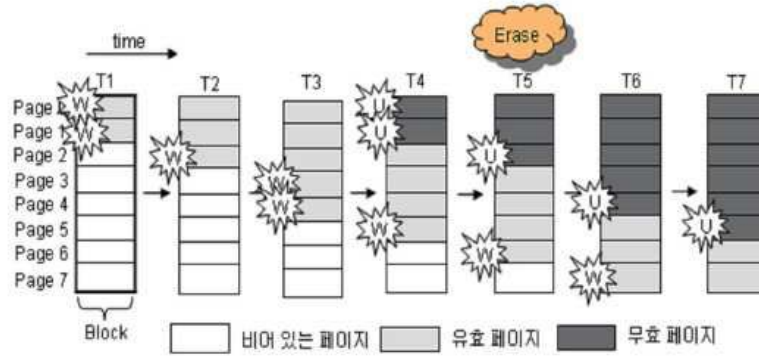
[0064]

도 13은 그룹의 개수에 따른 소거 비용을 나타낸 그래프이다.

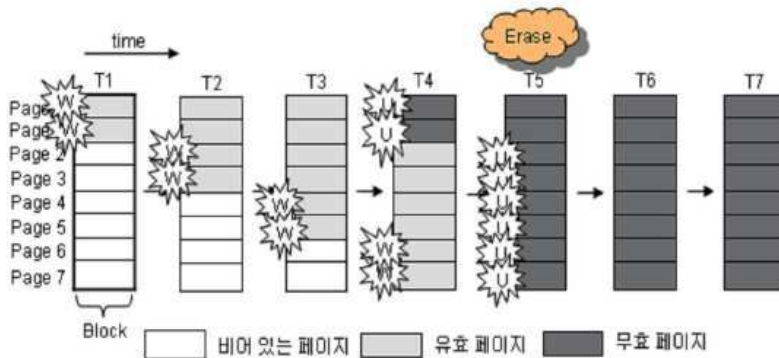
- [0065] 도 14는 그룹의 업데이트 타임 간격에 따른 평균 블록 복사 비용을 나타낸 그래프이다.
- [0066] 도 15는 그룹의 업데이트 타임 간격에 따른 소거 연산의 총 횟수를 나타낸 그래프이다.
- [0067] 도 16은 그룹의 업데이트 타임 간격에 따른 소거 비용을 나타낸 그래프이다.

도면

도면1



도면2

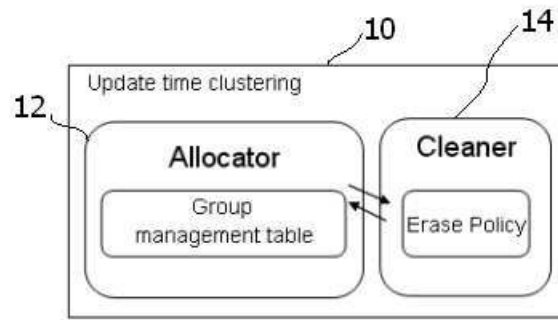


도면3

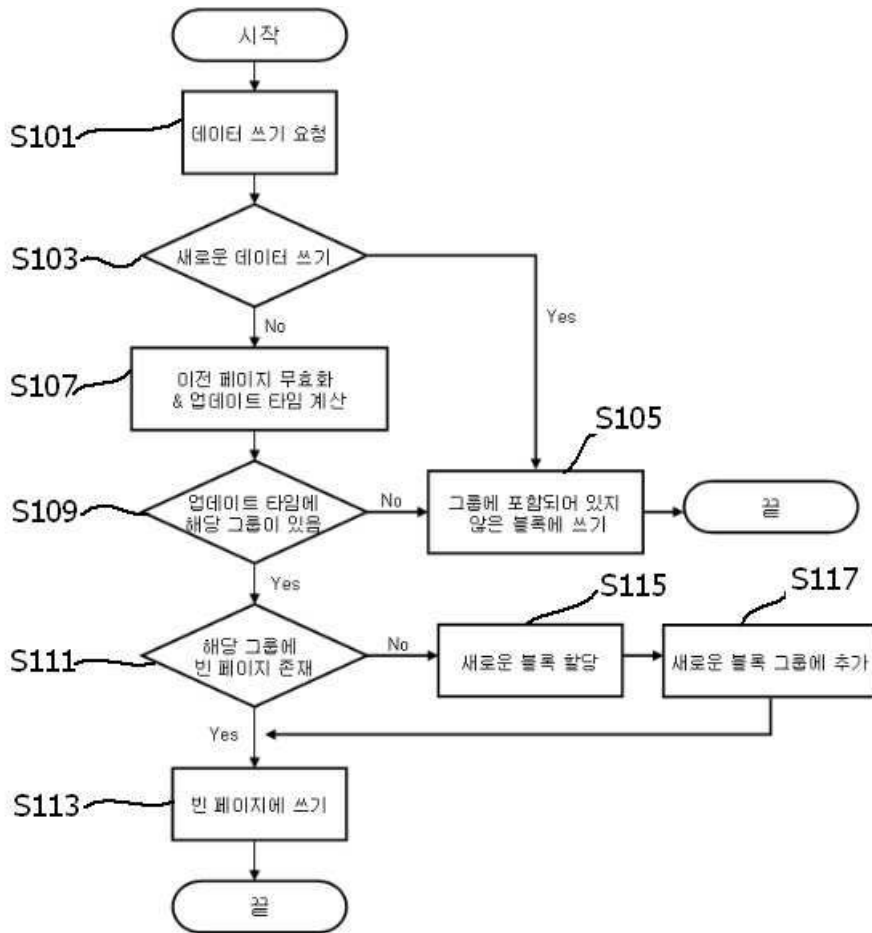
| | | |
|---------------------|---------------------|---------------------|
| 데이터 0 업데이트 주기: 7 | 데이터 1 업데이트 주기: 4 | 데이터 2 업데이트 주기: 7 |
| 데이터 3 업데이트 주기: 8 | 데이터 4 업데이트 주기: 5 | 데이터 5 업데이트 주기: 6 |

Page 1의 업데이트 주기: 4

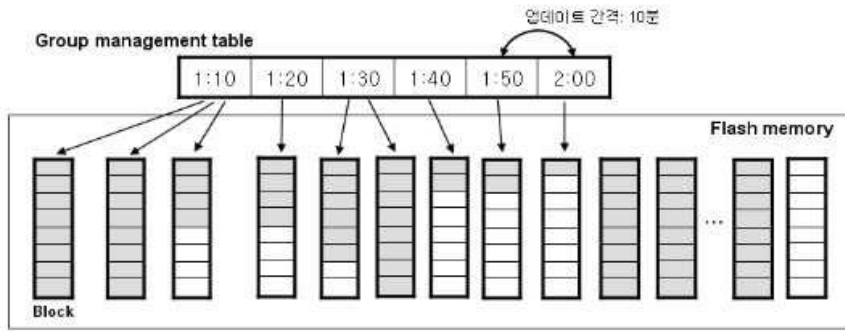
도면4



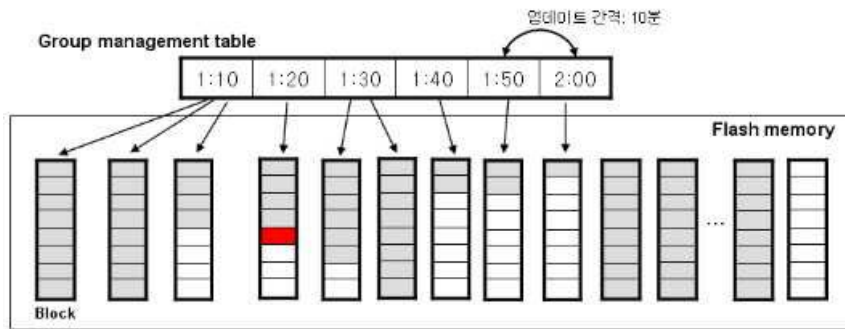
도면5



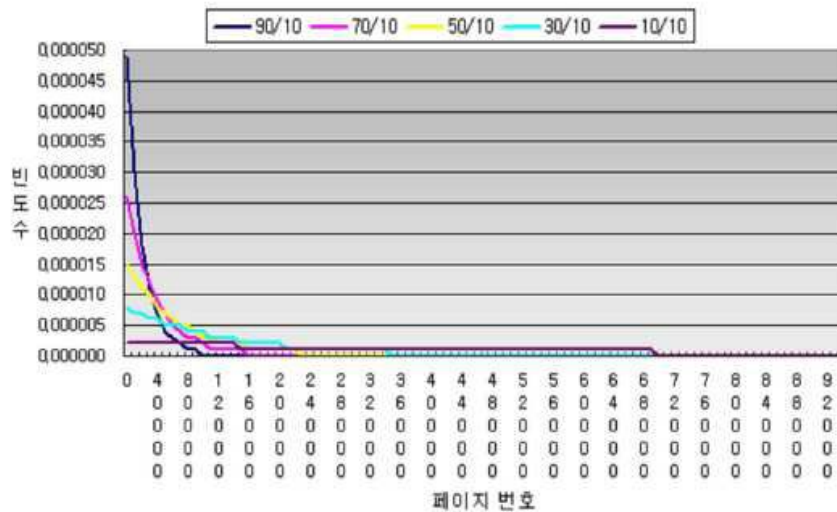
도면6a



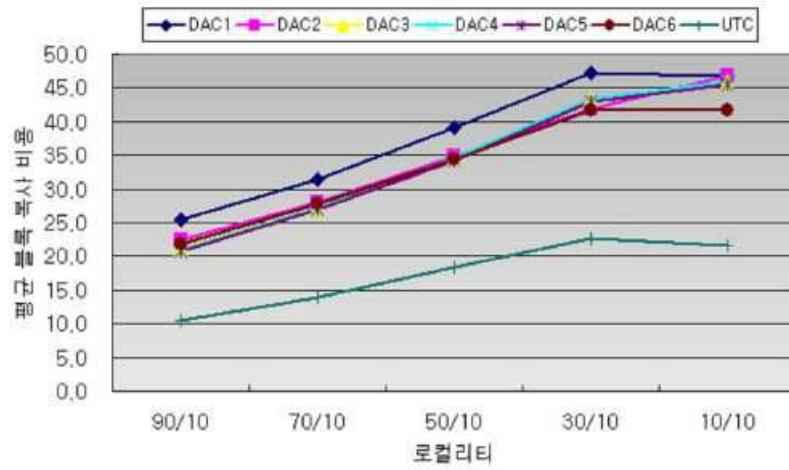
도면6b



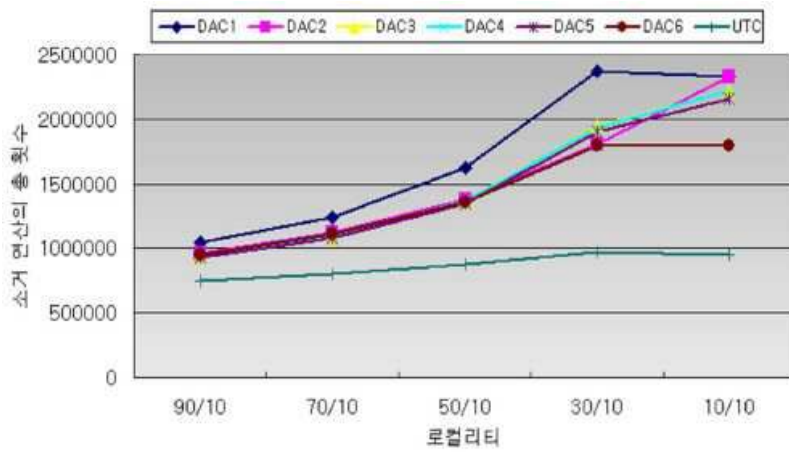
도면7



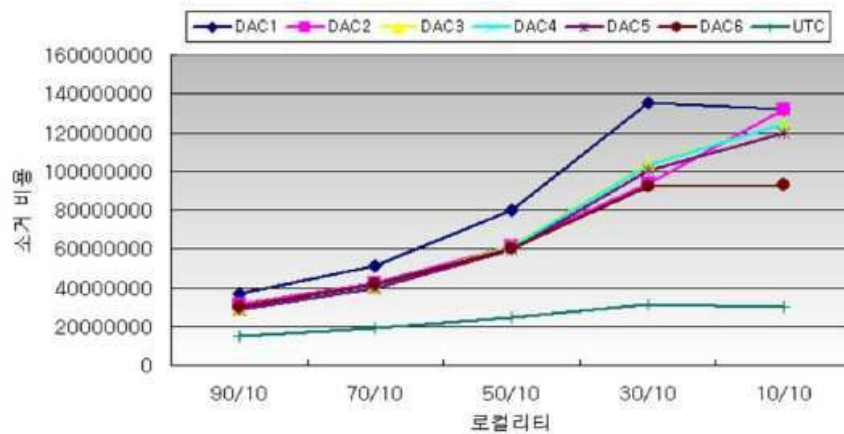
도면8



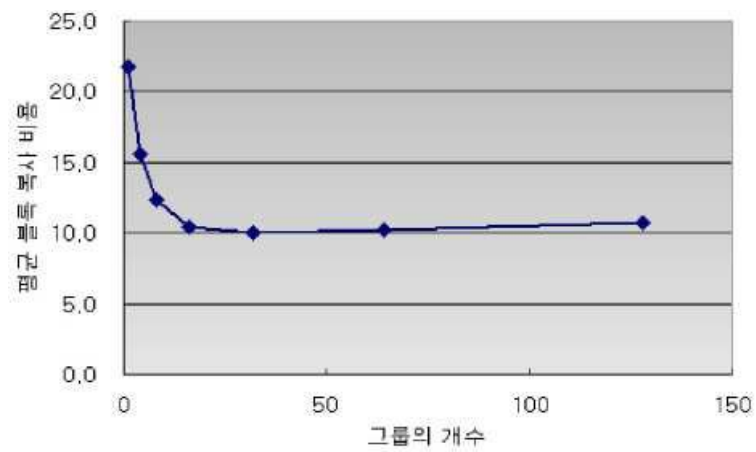
도면9



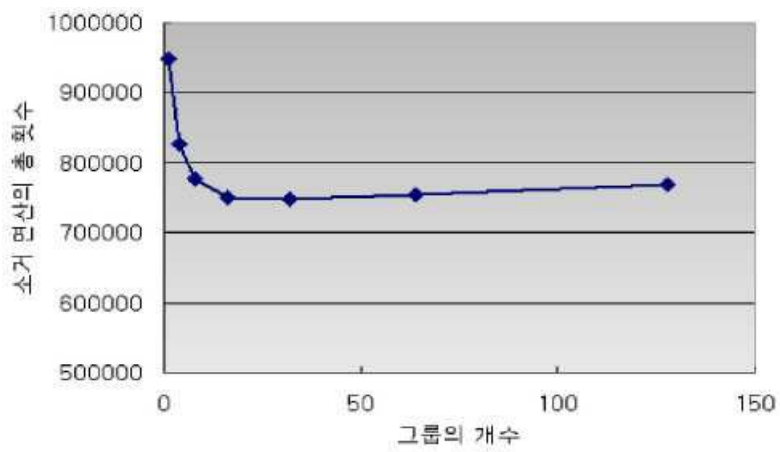
도면10



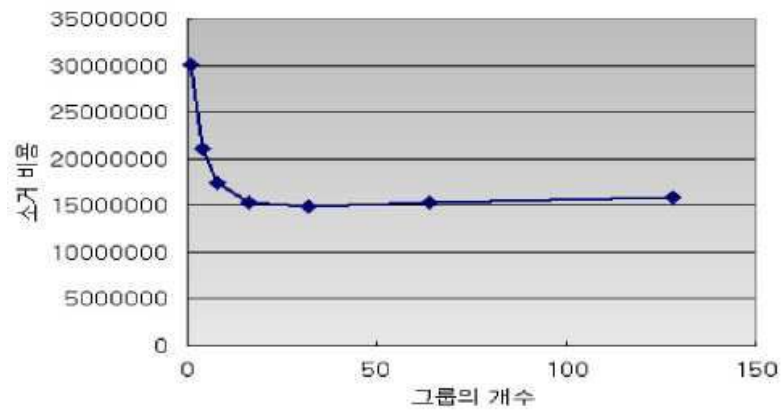
도면11



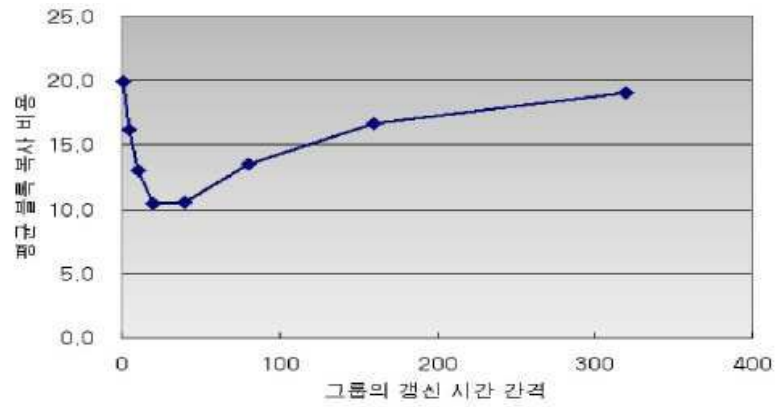
도면12



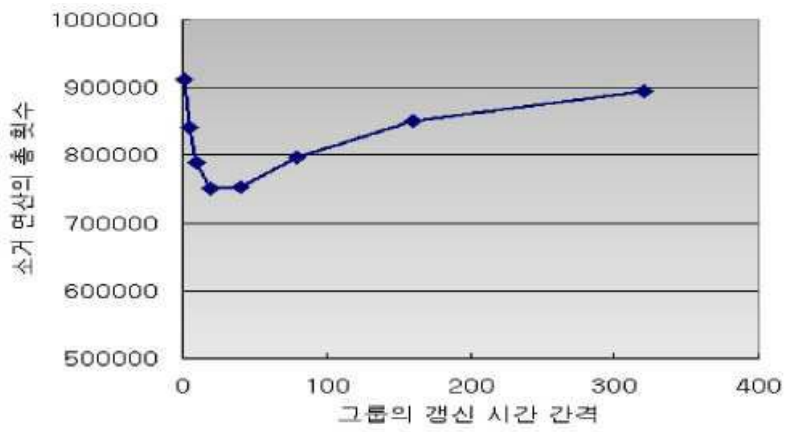
도면13



도면14



도면15



도면16

