Improving Directory Entry Insertion in EXT File Systems for O(1) Time Complexity

이 름 윤동현 지도교수 김상훈

2024-2 AJOU SOFTCON

연구배경

Ext4 파일 시스템은 Linux 환경에서 널리 사용되며, 디렉터리 검색 효율성을 높이기 위해 HTree 구조를 사용하여 시간 복잡도를 O(N)에서 O(log N)으로 개선하였다. 하지만, Ext4의 디렉토리 엔트리 관리 방식은 디렉토리 구조가 클 때, 삽입 과정에서 여전히 최악의 경우 O(N)의 시간복잡도를 필요로 한다. 이는 다수의 디스크 접근을 요구하여 성능 저하를 유발한다.

본 연구에서는 Slotarray와 이중 연결 리스트(doubly linked list)를 활용한 새로운 자유 공간 관리 기법을 제안한다. 이를 통해 디렉터리 엔트리 삽입 및 삭제 작업에서 O(1) 시간 복잡도를 달성하고, 블록 I/O 작업을 최소화하며, 공간 단편화를 줄이는 것을 목표로 한다. 초기 실험은 구조가 간단한 Ext2에서 진행되었으며, 이를 통해 Ext4로의 최적화 가능성을 제시한다. 이러한 연구는 대규모 디렉터리를 다룰 때 파일 시스템의 확장성과 성능을 개선할 수 있는 잠재력을 가진다.

자료구조

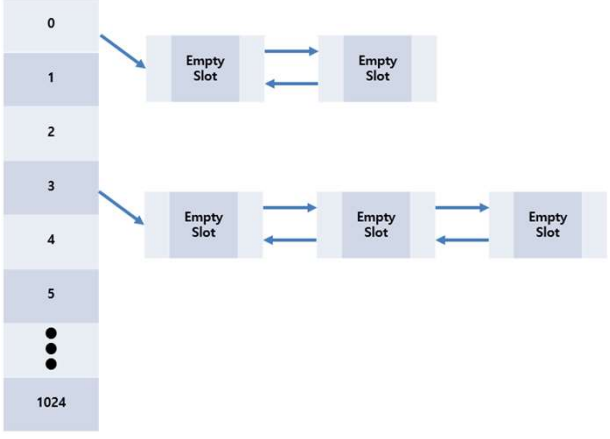
Slotarray

- 큰 디렉토리 구조에서 디렉토리 엔트리 삽입 시 시간복잡도를 줄이기 위한 구조이다.
- 한 블록(이 연구에서는 4KB를 기준으로 하였음)을 할당하고, 배열의 각 엔트리의 사이즈는 4Byte이다.
- 삽입하려는 디렉토리 엔트리 사이즈를 4로 나눈 값이 배열이 인덱스 값이 된다.
- 각 엔트리에는 새 디렉토리를 삽입할 위치 정보(블록넘버+오프셋)가 들어있다. 위치 정보는 삽입할 빈 공간의 이중 연결 리스트를 가리키게 된다.

Doubly linked list

- Slotarray는 추가하려는 디렉토리 엔트리의 사이즈에 맞는 블록을 찾아주는 자료구조이나, 해당 사이즈에 대해 빈 공간이 하나만 있다는 보장이 없다.
- 동일 사이즈를 가지는 빈 공간들에 대해 이중연결 리스트를 통해 관리를 한다.
- 리스트는 기본적으로 LIFO구조를 이용한다.

아래는 두 자료구조를 도식화한 것이다.



알고리즘

변경된 알고리즘은 디렉토리 엔트리 삭제와 삽입이다.

Directory Deletion

1. 삭제할 디렉토리 엔트리를 찾아낸다. 삭제할 디렉토리 엔트리의 이전 엔트리와 이후 엔트리가 빈 공간인지 확인한다.
2. 인접한 엔트리가 빈 공간일 경우, 이들을 병합하는 과정을 진행한다. 병합에 쓰이는 빈 공간들은 연결 리스트의 요소이므로 리스트로부터 제거한다.
3. 상기 과정을 거쳐 새로 생성된 빈 공간을 빈 공간의 사이즈에 맞는 Slotarray가 가리키는 연결리스트의 맨 앞에 추가하고 Slotarray의 엔트리값을 업데이트한다.

Directory Insertion

1. 새로 삽입할 디렉토리 엔트리에 필요한 사이즈를 계산한다. 계산된 사이즈를 기준으로 Slotarray의 엔트리를 찾는다.
2. 엔트리가 가리키는 값이 있을 경우, 해당 위치정보로 이동한다. 가리키는 값이 없을 경우, 인덱스를 증가시키며 적합한 슬롯을 탐색한다. 인덱스의 첫 증가시에만 3 증가시키고, 이후에는 1씩 증가 시켜가며 탐색한다(디렉토리 엔트리의 최소 사이즈가 12Byte이기 때문에 Internal Fragmentation을 방지하기 위함).
3. Slotarray 탐색 완료 후에도 빈 공간이 없을 경우, 새 페이지를 할당하여 디렉토리 엔트리를 저장할 공간을 마련한다.
4. 마련된 공간에 디렉토리 엔트리를 삽입한다. 해당 빈 공간은 연결리스트에서 삭제되고, 삽입 이후에 빈 공간이 더 있을 경우 남은 공간의 사이즈에 해당하는 Slotarray와 연결리스트를 업데이트하여 자료구조를 유지한다.

결과 및 분석

환경

- Debian 기반 QEMU 머신에서 Ext2 파일 시스템으로 실험
- Sync mode를 통해 모든 쓰기 작업을 즉시 디스크에 기록하도록 하여 정확한 블록 I/O 측정을 함
- Blktrace를 사용하여 블록 I/O 횟수 분석

실험 방법

- 파일 이름 길이가 10~16자인 10,000개의 파일로 구성된 디렉토리를 상대로 실험
- 디렉토리 엔트리 삽입 시 발생하는 블록 I/O 횟수 분석

결과적으로 최적화 이전 Ext2에서는 495회, Slotarray와 이중 연결 리스트를 적용한 Ext2에서는 87회의 블록 I/O로 82%가량 I/O 횟수가 줄어들었다.

Slotarray와 이중 연결 리스트를 활용한 최적화 알고리즘은 디렉터리 삽입 및 삭제에서 O(1) 시간 복잡도를 달성하여 블록 I/O 작업을 크게 줄이고, 대규모 파일 시스템 관리의 성능 병목현상을 완화할 잠재력을 보여준다.