LAB4 보고서

2016311821 한승하

Ftl Simulator Lab Report입니다.

< Data Stucture and Ftl Open >

```c
#include "ftl.h"
#include "nand.h"
typedef struct l2ptable
{
        u32 ppn;
        char v;
}table;
table* l2p;
typedef struct ppntable
{
        u32 p_age;
        char v;
}ptable;
ptable *ppn;
u32 *ptr;
int *freeblocks;
u32 ltime = 1;
```

위는 이번 Simulator에서 Table, 등의 용도로 사용된 Data 구조체들의 선언 부분입니다.

```c
void ftl_open()
{
        freeblocks = (int*)malloc(sizeof(int)*N_BANKS);
        int k;
        for(k=0;k<N_BANKS;k++) freeblocks[k] = BLKS_PER_BANK;
        u32 freepages = N_BANKS*BLKS_PER_BANK*PAGES_PER_BLK;
        nand_init(N_BANKS,BLKS_PER_BANK,PAGES_PER_BLK);
        l2p = (table*)malloc(sizeof(table)*freepages);
        ppn = (ptable*)malloc(sizeof(ptable)*freepages);
        u32 i;
        for(i=0;i<freepages;i++)
        {
                l2p[i].v = 'n';
                ppn[i].v = 'f';
        }
        ptr = (u32*)malloc(sizeof(u32)*N_BANKS);
        for(i=0;i<N_BANKS;i++) ptr[i] = 0;
        return;
}
```

위 구조체들은 ftl_open() 함수에서 Memory Allocation을 진행해 주었습니다.

Freeblocks는 Bank당 Freeblock 수를 관리하기 위해 Bank만큼의 index를 가진 int array 형식으로 사용하였고, ptr은 각 Bank당 Sequential Write를 위해 Bank만큼의 index를 가지고 각 Bank당 기록할 위치를 Pointing 하게 하기 위해 사용하였습니다. Freeblocks는 Bank당 초기 Freeblock인 BLKS_PER_BANK으로, ptr은 0으로 초기화 시켜 주었습니다.

l2p는 N_BANKS*BLKS_PER_BANK*PAGES_PER_BLK 만큼 존재하는 Logic Address 하나하나에 Physical Address를 대응시키기 위해 선언해 주었습니다. l2p table은 struct로 선언해 주었는데, 이는 logic address에 처음 mapping 하는 경우, 쓰레기 값을 읽는 것을 판단, 방지하기 위하여 char 형 valid flag하나와, ppn(physical page address) 값을 가지는 struct의 array 형식으로 사용하였습니다. Valid flag는 n (해당 lpn에 쓰여진적 없음), w (해당 lpn은 ppn과 mapping 되어 있음) 두 상태로 관리해 주었습니다. initial에는 모든 lpn의 상태를 n으로 초기화 시켜 주었습니다.

ppn은 각 physical page당 상태 flag, page age(기록된 시점)을 기록하기 위하여 l2p와 동일한 struct array형식으로 사용하였으며, valid flag 값은 f (Free page), i (Invalid page) , v (Valid page) 3가지 값으로 관리했습니다. Page age는 후에 Cost-benefit Policy에서 block age를 계산하기 위하여 사용하였습니다. initial때는 모든 ppn의 상태를 f으로 초기화 시켜 주었습니다.

ltime은 Logic time으로 매 write operation당 1씩 증가하게 해 주었으며, 가장 쓰여진지 오래 된 Data를 판단하기 위하여 사용하였습니다. 1부터 시작되도록 1로 초기화 시켜 주었습니다.

< Ftl Read >

```
void ftl_read(u32 lpn, u32 *read_buffer)
{
        int bank = lpn%N_BANKS;
        u32 addr = l2p[lpn].ppn;
        int blk = (addr - bank*BLKS_PER_BANK*PAGES_PER_BLK)/PAGES_PER_BLK;
        int page = (addr - bank*BLKS_PER_BANK*PAGES_PER_BLK - blk*PAGES_PER_BLK);
        u32 spare;
        nand_read(bank,blk,page,read_buffer,&spare);
        return;
}
```

Ftl read 함수입니다.

Stripping에 따라 Bank는 lpn%N_BANKS로 정의해 주었습니다.

따라서 ppn 상에서 각 BANK DATA의 시작점은 BANK * (PAGE PER BANK) 입니다. 이는 이후 Write, GC에서 blk number와 page number를 계산하기위해 사용됩니다.

l2p table을 이용하여 ppn을 가져온 후 ppn을 이용하여 blk number와 page number를 계산합니다. 이후 nand_read함수를 사용하여 해당 위치의 data를 읽어 read_buffer로 넘겨주었습니다. spare값은 read에선 사용되지 않으나, read인자로 필요하여 사용해 주었습니다.

< Ftl Write >

```
void ftl_write(u32 lpn, u32 *write_buffer)
{
        int bank = lpn%N_BANKS;
        if(freeblocks[bank] == 1) garbage_collection(bank);
        if(l2p[lpn].v != 'n')
        {
                u32 addr;
                addr = l2p[lpn].ppn;
                ppn[addr].v = 'i';
        }
        if((ptr[bank]+1)%(BLKS_PER_BANK) == 0) freeblocks[bank]--;
        int blk = ptr[bank]/PAGES_PER_BLK;
        int page = ptr[bank]%PAGES_PER_BLK;
        l2p[lpn].ppn = bank*BLKS_PER_BANK*PAGES_PER_BLK + blk*PAGES_PER_BLK + page;
        l2p[lpn].v = 'w';
        nand_write(bank,blk,page,write_buffer,lpn);
        ppn[bank*BLKS_PER_BANK*PAGES_PER_BLK+ptr[bank]].v = 'v';
        ppn[l2p[lpn].ppn].p_age = ltime;
        ltime++;
        if(ptr[bank] == BLKS_PER_BANK*PAGES_PER_BLK-1)
        {
                ptr[bank] = 0;
                while(ppn[bank*BLKS_PER_BANK*PAGES_PER_BLK + ptr[bank]].v != 'f') ptr[bank]++;
        }
        else ptr[bank]++;
        return;
}
```

Ftl write입니다.

Read와 동일하게 lpn을 이용하여 bank number를 구하고, bank의 freebolck 개수를 확인하여 1개일 경우 GC를 진행합니다.

또한 l2p table에서 해당 lpn의 상태가 n이 아닐 경우 즉 이미 mapping 된 ppn이 있을 경우, 해당 page가 invalid 되었음을 ppn table에 표시해 줍니다.

Freeblocks는 ptr+1 즉 다음에 write가 호출되었을 때 기록될 위치가 BLK의 처음인 경우 즉 한 BLK의 마지막 page에 기록하고 있을 경우 줄어들게 해 주었습니다.

Ptr을 이용하여 현재 기록하고 있는 page의 ppn을 계산하고 이를 l2p table에 기록해 줍니다. 이후 해당 table의 v값을 w로 바꿔주어 mapping된 lpn임을 표시해 줍니다.

이후 nand_write를 이용하여 nand에 기록해 줍니다. 이때 GC때 사용하기 위해 spare 영역엔 lpn을 기록해 주었습니다.

Write가 끝난 후 ppn table에 해당 page가 valid page임을 기록해주고, ltime을 기록한 뒤

Itiem++를 해줍니다.

마지막 부분은 ptr을 증가시키는 과정인데 sequential write로 Bank의 마지막 부분을 가르키고 있을 때, 가장 위에 있는 free blk의 시작을 가르키도록 조정해 주었습니다.

< Garbage Collection - Greedy >

```
#ifndef COST_BENEFIT
        int i,j,count,st = 0;
        int nfree = 0;
        int victim = 0;
        int freeblk = 0;
        u32* spare = (u32*)malloc(sizeof(u32));
        u32 r_buffer[SECTORS_PER_PAGE];
        memset(r_buffer, 0, DATA_SIZE);
        u32 s_point = bank*BLKS_PER_BANK*PAGES_PER_BLK;
        for(i=0;i<BLKS_PER_BANK;i++)
        {
                count = 0;
                nfree = 0;
                for(j=0;j<PAGES_PER_BLK;j++)
                {
                        if(ppn[s_point+i*PAGES_PER_BLK+j].v == 'i') count++;
                        if(ppn[s_point+i*PAGES_PER_BLK+j].v == 'f') nfree++;
                }
                if(count>st)
                {
                        victim = i;
                        st = count;
                }
                if(nfree == BLKS_PER_BANK){
                        freeblk = i;
                }
        }
        ptr[bank] = freeblk*PAGES_PER_BLK;
        for(j=0;j<PAGES_PER_BLK;j++)
        {
                if(ppn[s_point+victim*PAGES_PER_BLK+j].v == 'v')
                {
                        nand_read(bank,victim,j,r_buffer,spare);
                        nand_write(bank,freeblk,ptr[bank]%PAGES_PER_BLK,r_buffer,*spare);
                        ppn[bank*BLKS_PER_BANK*PAGES_PER_BLK + ptr[bank]].v = 'v';
                        l2p[*spare].ppn = bank*BLKS_PER_BANK*PAGES_PER_BLK + freeblk*PAGES_PER_BLK + (ptr[bank]%PAGES_PER_BLK);
                        ptr[bank]++;
                        s.gc_write++;
                }
                ppn[s_point+victim*PAGES_PER_BLK+j].v = 'f';
                ppn[s_point+victim*PAGES_PER_BLK+j].p_age = 0;
        }
        nand_erase(bank,victim);
        freeblocks[bank]++;
        free(spare);
```

Greedy Policy Garbage Collection입니다.

해당 Bank의 모든 blk를 돌며 필요한 blk를 찾습니다.

먼저 i 수를 세서 invalid page가 가장 많은 victim blk를 찾습니다 (greedy policy)

또 f 수를 세서 valid page를 복사할 free blk를 찾은 후 해당 bank의 ptr을 freeblk의 시작 부분으로 조정해 줍니다.

Victim blk와 free blk가 찾아지면 victim blk를 돌며 valid page일 경우 data를 nand_read로 read buffer와 spare로 읽어옵니다. 이후 nand_write를 이용하여 freeblk에 기록해 줍니다. Spare으로 읽

어 온 lpn을 이용하여 해당 lpn의 l2p table의 ppn값을 새로 적힌 page로 바꾸어 주고 valid 처리 해줍니다. 이후 victim blk의 모든 page는 free 처리해주고 age도 초기화 시켜준 후 해당 blk를 nand_erase를 사용하여 지워 주었습니다.

이후 freeblock 개수를 1 증가시켜주고 spare의 allocation을 풀어줍니다.

< Garbage Collection – Cost Benefit >

```
#else
        int i,j,count = 0;
        float st = 0;
        float cost_b = 0;
        int nfree = 0;
        int victim = 0;
        int freeblk = 0;
        int age = 0;
        u32* spare = (u32*)malloc(sizeof(u32));
        u32 r_buffer[SECTORS_PER_PAGE];
        memset(r_buffer, 0, DATA_SIZE);
        u32 s_point = bank*BLKS_PER_BANK*PAGES_PER_BLK;
        for(i=0;i<BLKS_PER_BANK;i++)
        {
                count = 0;
                nfree = 0;
                age = 0;
                for(j=0;j<PAGES_PER_BLK;j++)
                {
                        if((ppn[s_point+i*PAGES_PER_BLK+j].v == 'v') && (ppn[s_point+i*PAGES_PER_BLK+j].p_age > age))
                        age = ppn[s_point+i*PAGES_PER_BLK+j].p_age;
                        if(ppn[s_point+i*PAGES_PER_BLK+j].v == 'i') count++;
                        if(ppn[s_point+i*PAGES_PER_BLK+j].v == 'f') nfree++;
                }
                if(count == PAGES_PER_BLK)
                {
                        victim = i;
                        st = count * ltime;
                }
                else    cost_b = (count*(ltime-age))/(PAGES_PER_BLK - count);
                if(cost_b>st)
                {
                        victim = i;
                        st = cost_b;
                }
                if(nfree == BLKS_PER_BANK){
                        freeblk = i;
                }
        }
        ptr[bank] = freeblk*PAGES_PER_BLK;
        for(j=0;j<PAGES_PER_BLK;j++)
        {
                if(ppn[s_point+victim*PAGES_PER_BLK+j].v == 'v')
                {
                        nand_read(bank,victim,j,r_buffer,spare);
                        nand_write(bank,freeblk,ptr[bank]%PAGES_PER_BLK,r_buffer,*spare);
                        ppn[bank*BLKS_PER_BANK*PAGES_PER_BLK + ptr[bank]].v = 'v';
                        ppn[bank*BLKS_PER_BANK*PAGES_PER_BLK + ptr[bank]].p_age = ltime;
                        ltime++;
                        l2p[*spare].ppn = bank*BLKS_PER_BANK*PAGES_PER_BLK + freeblk*PAGES_PER_BLK + (ptr[bank]%PAGES_PER_BLK);
                        ptr[bank]++;
                        s.gc_write++;
                }
                ppn[s_point+victim*PAGES_PER_BLK+j].v = 'f';

                ppn[s_point+victim*PAGES_PER_BLK+j].p_age = 0;

        }
        nand_erase(bank,victim);
        freeblocks[bank]++;
        free(spare);
#endif
```

Cost benefit policy입니다. 전체적인 흐름은 Greedy와 같습니다.

```
for(i=0;i<BLKS_PER_BANK;i++)
{
        count = 0;
        nfree = 0;
        age = 0;
        for(j=0;j<PAGES_PER_BLK;j++)
        {
                if((ppn[s_point+i*PAGES_PER_BLK+j].v == 'v') && (ppn[s_point+i*PAGES_PER_BLK+j].p_age > age))
                age = ppn[s_point+i*PAGES_PER_BLK+j].p_age;
                if(ppn[s_point+i*PAGES_PER_BLK+j].v == 'i') count++;
                if(ppn[s_point+i*PAGES_PER_BLK+j].v == 'f') nfree++;
        }
        if(count == PAGES_PER_BLK)
        {
                victim = i;
                st = count * ltime;
        }
        else    cost_b = (count*(ltime-age))/(PAGES_PER_BLK - count);
        if(cost_b>st)
        {
                victim = i;
                st = cost_b;
        }
        if(nfree == BLKS_PER_BANK){
                freeblk = i;
        }
}
```

Victim blk을 탐색하는 과정에서 (1-u)*age/2u를 간략화 하면 k*invalid page*age/valid page (k는 상수) 가 됩니다. 따라서 invalid page number와 age 값이 필요합니다. Count로 invalid page개수를 계산하였고, age로 최근에 적힌 page를 탐색하였는데 이는 age에 대한 policy를 blk에 write된 시간이 오래 될수록 age가 크게 하기 위해 가장 최근에 blk에 적힌 시점을 탐색하였습니다.

Ltime이 계속 증가하므로 age가 가장 큰 page가 가장 최근에 적힌 page입니다. 따라서 이후 ltime (current logic time) – age를 통해 쓰여진 이후 지난 logic time을 계산하여 사용했습니다.

얻어낸 정보로 blk당 계산을 하여 가장 값이 큰 blk을 선정하게 하였습니다. 예외로 한 blk의 모든 page가 invaild되어 있을 경우 우선적으로 선정할 수 있도록 victim으로 선정 후 st에 들어 갈 수 있는 최댓값을 부여하여 바뀌지 않도록 하였습니다.

```
ptr[bank] = freeblk*PAGES_PER_BLK;
for(j=0;j<PAGES_PER_BLK;j++)
{
        if(ppn[s_point+victim*PAGES_PER_BLK+j].v == 'v')
        {
                nand_read(bank,victim,j,r_buffer,spare);
                nand_write(bank,freeblk,ptr[bank]%PAGES_PER_BLK,r_buffer,*spare);
                ppn[bank*BLKS_PER_BANK*PAGES_PER_BLK + ptr[bank]].v = 'v';
                ppn[bank*BLKS_PER_BANK*PAGES_PER_BLK + ptr[bank]].p_age = ltime;
                ltime++;
                l2p[*spare].ppn = bank*BLKS_PER_BANK*PAGES_PER_BLK + freeblk*PAGES_PER_BLK + (ptr[bank]%PAGES_PER_BLK);
                ptr[bank]++;
                s.gc_write++;
        }
        ppn[s_point+victim*PAGES_PER_BLK+j].v = 'f';

        ppn[s_point+victim*PAGES_PER_BLK+j].p_age = 0;

}
nand_erase(bank,victim);
freeblocks[bank]++;
free(spare);
```

이후 과정은 Greedy와 거의 동일합니다. 다른 부분은 greedy에선 고려하지 않았던 age를 copy 과정에서 새로운 page에도 기록해 주었습니다.

아래는 Result 창입니다

Greedy Policy (Random)

```
han@han-VirtualBox:~/ES/lab2$ ./ftl_sim
Bank: 2
Blocks / Bank: 32 blocks
Pages / Block: 32 pages
OP ratio: 7%
Physical Blocks: 64
User Blocks: 56
OP Blocks: 8
PPNs: 2048
LPNs: 1792
Workload: Random
FTL: Greedy policy

[Run 1] host 1792, valid page copy 0, GC# 0, WAF 1.00
[Run 2] host 3584, valid page copy 1321, GC# 92, WAF 1.37
[Run 3] host 5376, valid page copy 4633, GC# 251, WAF 1.86
[Run 4] host 7168, valid page copy 9566, GC# 461, WAF 2.33
[Run 5] host 8960, valid page copy 15640, GC# 707, WAF 2.75
[Run 6] host 10752, valid page copy 22138, GC# 966, WAF 3.06
[Run 7] host 12544, valid page copy 28856, GC# 1232, WAF 3.30
[Run 8] host 14336, valid page copy 35511, GC# 1496, WAF 3.48
[Run 9] host 16128, valid page copy 42427, GC# 1768, WAF 3.63
[Run 10] host 17920, valid page copy 49308, GC# 2039, WAF 3.75
[Run 11] host 19712, valid page copy 55928, GC# 2302, WAF 3.84
[Run 12] host 21504, valid page copy 63003, GC# 2579, WAF 3.93
[Run 13] host 23296, valid page copy 69822, GC# 2848, WAF 4.00
[Run 14] host 25088, valid page copy 76665, GC# 3118, WAF 4.06
[Run 15] host 26880, valid page copy 83644, GC# 3392, WAF 4.11
[Run 16] host 28672, valid page copy 90299, GC# 3656, WAF 4.15
[Run 17] host 30464, valid page copy 97210, GC# 3928, WAF 4.19
[Run 18] host 32256, valid page copy 103966, GC# 4195, WAF 4.22
[Run 19] host 34048, valid page copy 110907, GC# 4468, WAF 4.26
[Run 20] host 35840, valid page copy 117876, GC# 4742, WAF 4.29
```

```
[Run 165] host 295680, valid page copy 1111320, GC# 43907, WAF 4.76
[Run 166] host 297472, valid page copy 1118169, GC# 44177, WAF 4.76
[Run 167] host 299264, valid page copy 1125177, GC# 44452, WAF 4.76
[Run 168] host 301056, valid page copy 1132056, GC# 44723, WAF 4.76
[Run 169] host 302848, valid page copy 1138778, GC# 44989, WAF 4.76
[Run 170] host 304640, valid page copy 1145596, GC# 45258, WAF 4.76
[Run 171] host 306432, valid page copy 1152347, GC# 45525, WAF 4.76
[Run 172] host 308224, valid page copy 1159196, GC# 45795, WAF 4.76
[Run 173] host 310016, valid page copy 1166014, GC# 46064, WAF 4.76
[Run 174] host 311808, valid page copy 1172863, GC# 46334, WAF 4.76
[Run 175] host 313600, valid page copy 1179639, GC# 46602, WAF 4.76
[Run 176] host 315392, valid page copy 1186462, GC# 46871, WAF 4.76
[Run 177] host 317184, valid page copy 1193592, GC# 47150, WAF 4.76
[Run 178] host 318976, valid page copy 1200440, GC# 47420, WAF 4.76
[Run 179] host 320768, valid page copy 1207383, GC# 47693, WAF 4.76
[Run 180] host 322560, valid page copy 1214299, GC# 47965, WAF 4.76
[Run 181] host 324352, valid page copy 1221053, GC# 48232, WAF 4.76
[Run 182] host 326144, valid page copy 1227994, GC# 48505, WAF 4.77
[Run 183] host 327936, valid page copy 1234746, GC# 48772, WAF 4.77
[Run 184] host 329728, valid page copy 1241689, GC# 49045, WAF 4.77
[Run 185] host 331520, valid page copy 1248318, GC# 49308, WAF 4.77
[Run 186] host 333312, valid page copy 1255099, GC# 49576, WAF 4.77
[Run 187] host 335104, valid page copy 1261942, GC# 49846, WAF 4.77
[Run 188] host 336896, valid page copy 1268605, GC# 50110, WAF 4.77
[Run 189] host 338688, valid page copy 1275482, GC# 50381, WAF 4.77
[Run 190] host 340480, valid page copy 1282489, GC# 50656, WAF 4.77
[Run 191] host 342272, valid page copy 1289404, GC# 50928, WAF 4.77
[Run 192] host 344064, valid page copy 1296346, GC# 51201, WAF 4.77
[Run 193] host 345856, valid page copy 1303195, GC# 51471, WAF 4.77
[Run 194] host 347648, valid page copy 1310099, GC# 51743, WAF 4.77
[Run 195] host 349440, valid page copy 1316982, GC# 52014, WAF 4.77
[Run 196] host 351232, valid page copy 1323832, GC# 52284, WAF 4.77
[Run 197] host 353024, valid page copy 1330715, GC# 52555, WAF 4.77
[Run 198] host 354816, valid page copy 1337722, GC# 52830, WAF 4.77
[Run 199] host 356608, valid page copy 1344575, GC# 53100, WAF 4.77
[Run 200] host 358400, valid page copy 1351546, GC# 53374, WAF 4.77

Results ------
Host writes: 358400
GC writes: 1351546
Number of GCs: 53374
Valid pages per GC: 25.32 pages
WAF: 4.77
han@han-VirtualBox:~/ES/lab2$
```

Greedy Policy (Hot - Cold)

```
han@han-VirtualBox:~/ES/lab2$ ./ftl_sim
Bank: 2
Blocks / Bank: 32 blocks
Pages / Block: 32 pages
OP ratio: 7%
Physical Blocks: 64
User Blocks: 56
OP Blocks: 8
PPNs: 2048
LPNs: 1792
Workload: Hot 90 / Cold 10
FTL: Greedy policy

[Run 1] host 1792, valid page copy 0, GC# 0, WAF 1.00
[Run 2] host 3584, valid page copy 142, GC# 55, WAF 1.04
[Run 3] host 5376, valid page copy 472, GC# 122, WAF 1.09
[Run 4] host 7168, valid page copy 952, GC# 192, WAF 1.13
[Run 5] host 8960, valid page copy 1646, GC# 270, WAF 1.18
[Run 6] host 10752, valid page copy 2574, GC# 355, WAF 1.24
[Run 7] host 12544, valid page copy 3729, GC# 447, WAF 1.30
[Run 8] host 14336, valid page copy 5125, GC# 547, WAF 1.36
[Run 9] host 16128, valid page copy 6737, GC# 653, WAF 1.42
[Run 10] host 17920, valid page copy 8615, GC# 768, WAF 1.48
[Run 11] host 19712, valid page copy 10698, GC# 889, WAF 1.54
[Run 12] host 21504, valid page copy 13068, GC# 1019, WAF 1.61
[Run 13] host 23296, valid page copy 15629, GC# 1155, WAF 1.67
[Run 14] host 25088, valid page copy 18511, GC# 1301, WAF 1.74
[Run 15] host 26880, valid page copy 21718, GC# 1457, WAF 1.81
[Run 16] host 28672, valid page copy 25104, GC# 1619, WAF 1.88
[Run 17] host 30464, valid page copy 28703, GC# 1787, WAF 1.94
[Run 18] host 32256, valid page copy 32535, GC# 1963, WAF 2.01
[Run 19] host 34048, valid page copy 36755, GC# 2151, WAF 2.08
[Run 20] host 35840, valid page copy 41235, GC# 2347, WAF 2.15
[Run 165] host 295680, valid page copy 1247290, GC# 48156, WAF 5.22
[Run 166] host 297472, valid page copy 1255963, GC# 48483, WAF 5.22
[Run 167] host 299264, valid page copy 1264476, GC# 48805, WAF 5.23
[Run 168] host 301056, valid page copy 1273276, GC# 49136, WAF 5.23
[Run 169] host 302848, valid page copy 1281849, GC# 49460, WAF 5.23
[Run 170] host 304640, valid page copy 1290487, GC# 49786, WAF 5.24
[Run 171] host 306432, valid page copy 1298999, GC# 50108, WAF 5.24
[Run 172] host 308224, valid page copy 1307616, GC# 50433, WAF 5.24
[Run 173] host 310016, valid page copy 1316282, GC# 50760, WAF 5.25
[Run 174] host 311808, valid page copy 1324888, GC# 51085, WAF 5.25
[Run 175] host 313600, valid page copy 1333628, GC# 51414, WAF 5.25
[Run 176] host 315392, valid page copy 1342489, GC# 51747, WAF 5.26
[Run 177] host 317184, valid page copy 1351162, GC# 52074, WAF 5.26
[Run 178] host 318976, valid page copy 1359706, GC# 52397, WAF 5.26
[Run 179] host 320768, valid page copy 1368318, GC# 52722, WAF 5.27
[Run 180] host 322560, valid page copy 1377210, GC# 53056, WAF 5.27
[Run 181] host 324352, valid page copy 1385943, GC# 53385, WAF 5.27
[Run 182] host 326144, valid page copy 1394525, GC# 53709, WAF 5.28
[Run 183] host 327936, valid page copy 1403259, GC# 54038, WAF 5.28
[Run 184] host 329728, valid page copy 1412187, GC# 54373, WAF 5.28
[Run 185] host 331520, valid page copy 1420862, GC# 54700, WAF 5.29
[Run 186] host 333312, valid page copy 1429469, GC# 55025, WAF 5.29
[Run 187] host 335104, valid page copy 1438077, GC# 55350, WAF 5.29
[Run 188] host 336896, valid page copy 1446814, GC# 55679, WAF 5.29
[Run 189] host 338688, valid page copy 1455608, GC# 56010, WAF 5.30
[Run 190] host 340480, valid page copy 1464285, GC# 56337, WAF 5.30
[Run 191] host 342272, valid page copy 1472823, GC# 56660, WAF 5.30
[Run 192] host 344064, valid page copy 1481464, GC# 56986, WAF 5.31
[Run 193] host 345856, valid page copy 1490042, GC# 57310, WAF 5.31
[Run 194] host 347648, valid page copy 1498620, GC# 57634, WAF 5.31
[Run 195] host 349440, valid page copy 1507320, GC# 57962, WAF 5.31
[Run 196] host 351232, valid page copy 1516027, GC# 58290, WAF 5.32
[Run 197] host 353024, valid page copy 1524539, GC# 58612, WAF 5.32
[Run 198] host 354816, valid page copy 1532987, GC# 58932, WAF 5.32
[Run 199] host 356608, valid page copy 1541660, GC# 59259, WAF 5.32
[Run 200] host 358400, valid page copy 1550203, GC# 59582, WAF 5.33

Results ------
Host writes: 358400
GC writes: 1550203
Number of GCs: 59582
Valid pages per GC: 26.02 pages
WAF: 5.33
han@han-VirtualBox:~/ES/lab2$
```

Cost-Benifit Policy (Random)

```
han@han-VirtualBox:~/ES/lab2$ ./ftl_sim
Bank: 2
Blocks / Bank: 32 blocks
Pages / Block: 32 pages
OP ratio: 7%
Physical Blocks: 64
User Blocks: 56
OP Blocks: 8
PPNs: 2048
LPNs: 1792
Workload: Random
FTL: Cost-Benefit policy

[Run 1] host 1792, valid page copy 0, GC# 0, WAF 1.00
[Run 2] host 3584, valid page copy 1301, GC# 91, WAF 1.36
[Run 3] host 5376, valid page copy 4693, GC# 253, WAF 1.87
[Run 4] host 7168, valid page copy 9661, GC# 464, WAF 2.35
[Run 5] host 8960, valid page copy 15961, GC# 717, WAF 2.78
[Run 6] host 10752, valid page copy 22683, GC# 983, WAF 3.11
[Run 7] host 12544, valid page copy 29656, GC# 1257, WAF 3.36
[Run 8] host 14336, valid page copy 36632, GC# 1531, WAF 3.56
[Run 9] host 16128, valid page copy 43677, GC# 1807, WAF 3.71
[Run 10] host 17920, valid page copy 50943, GC# 2090, WAF 3.84
[Run 11] host 19712, valid page copy 57976, GC# 2366, WAF 3.94
[Run 12] host 21504, valid page copy 64894, GC# 2638, WAF 4.02
[Run 13] host 23296, valid page copy 72055, GC# 2918, WAF 4.09
[Run 14] host 25088, valid page copy 79227, GC# 3198, WAF 4.16
[Run 15] host 26880, valid page copy 86207, GC# 3472, WAF 4.21
[Run 16] host 28672, valid page copy 93204, GC# 3747, WAF 4.25
[Run 17] host 30464, valid page copy 100281, GC# 4024, WAF 4.29
[Run 18] host 32256, valid page copy 107319, GC# 4300, WAF 4.33
[Run 19] host 34048, valid page copy 114394, GC# 4577, WAF 4.36
[Run 20] host 35840, valid page copy 121469, GC# 4854, WAF 4.39
```

```
[Run 165] host 295680, valid page copy 1145854, GC# 44986, WAF 4.88
[Run 166] host 297472, valid page copy 1152922, GC# 45263, WAF 4.88
[Run 167] host 299264, valid page copy 1159960, GC# 45539, WAF 4.88
[Run 168] host 301056, valid page copy 1166903, GC# 45812, WAF 4.88
[Run 169] host 302848, valid page copy 1174045, GC# 46091, WAF 4.88
[Run 170] host 304640, valid page copy 1181146, GC# 46369, WAF 4.88
[Run 171] host 306432, valid page copy 1188344, GC# 46650, WAF 4.88
[Run 172] host 308224, valid page copy 1195319, GC# 46924, WAF 4.88
[Run 173] host 310016, valid page copy 1202362, GC# 47200, WAF 4.88
[Run 174] host 311808, valid page copy 1209625, GC# 47483, WAF 4.88
[Run 175] host 313600, valid page copy 1216728, GC# 47761, WAF 4.88
[Run 176] host 315392, valid page copy 1223771, GC# 48037, WAF 4.88
[Run 177] host 317184, valid page copy 1230874, GC# 48315, WAF 4.88
[Run 178] host 318976, valid page copy 1237981, GC# 48593, WAF 4.88
[Run 179] host 320768, valid page copy 1244896, GC# 48865, WAF 4.88
[Run 180] host 322560, valid page copy 1251998, GC# 49143, WAF 4.88
[Run 181] host 324352, valid page copy 1258842, GC# 49413, WAF 4.88
[Run 182] host 326144, valid page copy 1265917, GC# 49690, WAF 4.88
[Run 183] host 327936, valid page copy 1272762, GC# 49960, WAF 4.88
[Run 184] host 329728, valid page copy 1279868, GC# 50238, WAF 4.88
[Run 185] host 331520, valid page copy 1286911, GC# 50514, WAF 4.88
[Run 186] host 333312, valid page copy 1293882, GC# 50788, WAF 4.88
[Run 187] host 335104, valid page copy 1300986, GC# 51066, WAF 4.88
[Run 188] host 336896, valid page copy 1308087, GC# 51344, WAF 4.88
[Run 189] host 338688, valid page copy 1315037, GC# 51617, WAF 4.88
[Run 190] host 340480, valid page copy 1322068, GC# 51893, WAF 4.88
[Run 191] host 342272, valid page copy 1329084, GC# 52168, WAF 4.88
[Run 192] host 344064, valid page copy 1336344, GC# 52451, WAF 4.88
[Run 193] host 345856, valid page copy 1343360, GC# 52726, WAF 4.88
[Run 194] host 347648, valid page copy 1350397, GC# 53002, WAF 4.88
[Run 195] host 349440, valid page copy 1357497, GC# 53280, WAF 4.88
[Run 196] host 351232, valid page copy 1364570, GC# 53557, WAF 4.89
[Run 197] host 353024, valid page copy 1371804, GC# 53839, WAF 4.89
[Run 198] host 354816, valid page copy 1378942, GC# 54118, WAF 4.89
[Run 199] host 356608, valid page copy 1386008, GC# 54395, WAF 4.89
[Run 200] host 358400, valid page copy 1392863, GC# 54665, WAF 4.89

Results ------
Host writes: 358400
GC writes: 1392863
Number of GCs: 54665
Valid pages per GC: 25.48 pages
WAF: 4.89
han@han-VirtualBox:~/ES/lab2$
```

Cost-Benifit Policy (Hot - Cold)

```
han@han-VirtualBox:~/ES/lab2$ ./ftl_sim
Bank: 2
Blocks / Bank: 32 blocks
Pages / Block: 32 pages
OP ratio: 7%
Physical Blocks: 64
User Blocks: 56
OP Blocks: 8
PPNs: 2048
LPNs: 1792
Workload: Hot 90 / Cold 10
FTL: Cost-Benefit policy

[Run 1] host 1792, valid page copy 0, GC# 0, WAF 1.00
[Run 2] host 3584, valid page copy 155, GC# 55, WAF 1.04
[Run 3] host 5376, valid page copy 482, GC# 122, WAF 1.09
[Run 4] host 7168, valid page copy 990, GC# 194, WAF 1.14
[Run 5] host 8960, valid page copy 1686, GC# 271, WAF 1.19
[Run 6] host 10752, valid page copy 2633, GC# 357, WAF 1.24
[Run 7] host 12544, valid page copy 3812, GC# 450, WAF 1.30
[Run 8] host 14336, valid page copy 5196, GC# 549, WAF 1.36
[Run 9] host 16128, valid page copy 6803, GC# 655, WAF 1.42
[Run 10] host 17920, valid page copy 8637, GC# 768, WAF 1.48
[Run 11] host 19712, valid page copy 10739, GC# 890, WAF 1.54
[Run 12] host 21504, valid page copy 13050, GC# 1018, WAF 1.61
[Run 13] host 23296, valid page copy 15572, GC# 1153, WAF 1.67
[Run 14] host 25088, valid page copy 18390, GC# 1297, WAF 1.73
[Run 15] host 26880, valid page copy 21493, GC# 1450, WAF 1.80
[Run 16] host 28672, valid page copy 24851, GC# 1611, WAF 1.87
[Run 17] host 30464, valid page copy 28441, GC# 1779, WAF 1.93
[Run 18] host 32256, valid page copy 32308, GC# 1956, WAF 2.00
[Run 19] host 34048, valid page copy 36374, GC# 2139, WAF 2.07
[Run 20] host 35840, valid page copy 40756, GC# 2332, WAF 2.14
```

```
[Run 165] host 295680, valid page copy 1221018, GC# 47335, WAF 5.13
[Run 166] host 297472, valid page copy 1229469, GC# 47655, WAF 5.13
[Run 167] host 299264, valid page copy 1237850, GC# 47973, WAF 5.14
[Run 168] host 301056, valid page copy 1246396, GC# 48296, WAF 5.14
[Run 169] host 302848, valid page copy 1254750, GC# 48613, WAF 5.14
[Run 170] host 304640, valid page copy 1263256, GC# 48935, WAF 5.15
[Run 171] host 306432, valid page copy 1271741, GC# 49256, WAF 5.15
[Run 172] host 308224, valid page copy 1280252, GC# 49578, WAF 5.15
[Run 173] host 310016, valid page copy 1288605, GC# 49895, WAF 5.16
[Run 174] host 311808, valid page copy 1297083, GC# 50216, WAF 5.16
[Run 175] host 313600, valid page copy 1305657, GC# 50540, WAF 5.16
[Run 176] host 315392, valid page copy 1314077, GC# 50859, WAF 5.17
[Run 177] host 317184, valid page copy 1322553, GC# 51180, WAF 5.17
[Run 178] host 318976, valid page copy 1331036, GC# 51501, WAF 5.17
[Run 179] host 320768, valid page copy 1339610, GC# 51825, WAF 5.18
[Run 180] host 322560, valid page copy 1348351, GC# 52154, WAF 5.18
[Run 181] host 324352, valid page copy 1356828, GC# 52475, WAF 5.18
[Run 182] host 326144, valid page copy 1365438, GC# 52800, WAF 5.19
[Run 183] host 327936, valid page copy 1373887, GC# 53120, WAF 5.19
[Run 184] host 329728, valid page copy 1382493, GC# 53445, WAF 5.19
[Run 185] host 331520, valid page copy 1391094, GC# 53770, WAF 5.20
[Run 186] host 333312, valid page copy 1399416, GC# 54086, WAF 5.20
[Run 187] host 335104, valid page copy 1407996, GC# 54410, WAF 5.20
[Run 188] host 336896, valid page copy 1416606, GC# 54735, WAF 5.20
[Run 189] host 338688, valid page copy 1425116, GC# 55057, WAF 5.21
[Run 190] host 340480, valid page copy 1433404, GC# 55372, WAF 5.21
[Run 191] host 342272, valid page copy 1441790, GC# 55690, WAF 5.21
[Run 192] host 344064, valid page copy 1450364, GC# 56014, WAF 5.22
[Run 193] host 345856, valid page copy 1459039, GC# 56341, WAF 5.22
[Run 194] host 347648, valid page copy 1467614, GC# 56665, WAF 5.22
[Run 195] host 349440, valid page copy 1476251, GC# 56991, WAF 5.22
[Run 196] host 351232, valid page copy 1485013, GC# 57321, WAF 5.23
[Run 197] host 353024, valid page copy 1493595, GC# 57645, WAF 5.23
[Run 198] host 354816, valid page copy 1502172, GC# 57969, WAF 5.23
[Run 199] host 356608, valid page copy 1510780, GC# 58294, WAF 5.24
[Run 200] host 358400, valid page copy 1519451, GC# 58621, WAF 5.24

Results ------
Host writes: 358400
GC writes: 1519451
Number of GCs: 58621
Valid pages per GC: 25.92 pages
WAF: 5.24
han@han-VirtualBox:~/ES/lab2$
```