

read disturb design & reclaim scan design & high level refresh design

- [背景](#)
 - [Read Disturb](#)
 - [reclaim scan](#)
 - [refresh](#)
- [框架](#)
 - [模塊關係](#)
 - [queue](#)
- [input/output queue 的設計](#)
 - [input check queue](#)
 - [output check queue](#)
 - [input refresh queue](#)
 - [output refresh queue](#)
- [check flag/refresh flag 的設計](#)
 - [set check flag](#)
 - [clear check flag](#)
 - [set refresh flag](#)
 - [clear refresh flag](#)
- [read count 的統計](#)
- [read count 的臨界值](#)
- [read operation](#)
 - [step add read count](#)
 - [step check read count](#)
 - [step input queue if need](#)
- [check task](#)
 - [step trigger condition](#)
 - [step get one SPB](#)
 - [step read page check](#)
 - [step input refresh queue if need](#)
 - [step output check queue](#)
- [refresh task](#)
 - [模塊關係](#)
 - [step trigger condition](#)
 - [step get one SPB](#)
 - [step trigger refresh](#)
 - [step wait refresh done](#)
 - [step output refresh queue](#)

- [User Lu gc task / internal Lu gc task](#)
 - [step trigger gc](#)
 - [step source SPB](#)
 - [step gc flow](#)
 - [step gc done](#)
- [reclaim scan task](#)
 - [step trigger condition](#)
 - [step input queue](#)
 - [system block](#)
 - [PON APL case handle](#)

背景

Read Disturb

讀干擾，主要是讀頁操作，會對同一個塊內其他頁造成干擾，隨著 Flash 制程的提高、一個物理塊頁數更多、和單個 Cell 存儲的資訊更多 (TLC、QLC)，此問題會變得更加突出。嚴重的情況下會導致讀回來的數據出錯，導致數據丟失。

reclaim scan

對數據進行巡檢，檢查出有危險的數據，防止數據丟失

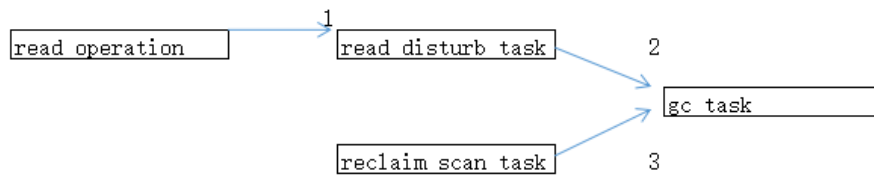
refresh

對有危險數據進行搬移 還到新的 block 上

框架

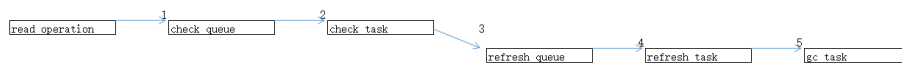
模塊關係

pic1



queue

pic2



check queue

等待進一步檢查的 SPB (或是 sub SPB) queue,

queue size 10 個

refresh queue

等待 refresh 的 SPB (或是 sub SPB) queue,

queue size 10 個

input/output queue 的設計

input check queue

當 SPB (或是 sub SPB) 的 read count 達到設定的臨界值 因為有丟失數據的風險 我們會把此 SPB (或是 sub SPB) 放到 check queue 等待進一步的檢查

除了 read count 之外 我們也會週期性的把 SPB 放到 check queue 等待進一步的檢查

output check queue

當我們檢查完 某 SPB(或是 sub SPB) 就移除該 SPB (或是 sub SPB)

input refresh queue

當讀 User Lu 或是 internal Lu 數據 error bit 數 達到設定的臨界值 我們需要 refresh ,我們會把 User Lu 或是 internal Lu 所在的 SPB (或是 sub SPB) 放到 refresh queue 等待 refresh

output refresh queue

當我們 refresh 完某 SPB(或是 sub SPB) 就移除該 SPB (或是 sub SPB)

check flag/refresh flag 的設計

check flag

等待進一步檢查的 SPB (或是 sub SPB) flag

refresh flag

等待 refresh 的 SPB (或是 sub SPB)flag,

set check flag

當 SPB (或是 sub SPB) 的 read count 達到設定的臨界值 因為有丟失數據的風險 我們會把此 SPB (或是 sub SPB) 放到 check queue 但 check queue 如果 full, 我們會把此 SPB (或是 sub SPB) 設定 check flag

clear check flag

當我們把此 SPB （或是 sub SPB）放到 check queue, clear check flag

set refresh flag

當讀 User Lu 或是 internal Lu 數據 error bit 數 達到設定的臨界值 我們需要 refresh ,我們會把 User Lu 或是 internal Lu 所在的 SPB （或是 sub SPB）放到 refresh queue 等待 refresh 但 refresh queue 如果 full , 我們會把此 SPB （或是 sub SPB）設定 refresh flag

clear refresh flag

當我們把此 SPB （或是 sub SPB）放到 refresh queue, clear refresh flag

read count 的統計

基於以下原因 我們統計 read count 是以 SPB （或是 sub SPB）為顆粒度

1 前面說過 讀頁操作 會對同一個塊內其他頁造成干擾

2 每個顆粒度需要使用接近 4 byte 紀錄 read count 越小的顆粒度 所需要的內存越大

描述SPB的状态信息:

```
Typedef struct _SPB_DESC_INFO
```

```
{
```

```
U32  flags:10;
```

```
    U32  validCnt:22;
```

```
    U32  readCnt:28;
```

```
    U32  subSpb:1;
```

```
    U32  spbType:1
```

```
    U32  poolId:2
```

```
    U32  ecCount;
```

```
    .....  ....
```

```
} SPB_DESC_INFO;
```

```
/* SPB_DESC_INFO */
```

read count 的臨界值

數據的保存能力 和數據所在的 block 相關 基於以下

1 數據保存能力 SLC block > TLC block

2 數據保存能力 隨著 erase count 的增加 而遞減

所以在設計 read count 的臨界值時 需要根據上述兩個參數 進行設定

同時不同的 Nand 廠家的情況 有所不同 所以根據不同廠家 進行設定

TLC erase count	<500	500-1000	1000-2000	2000-2500	>2500
close block					
read count threshold	1000k	800k	600k	400k	200k

open block read count threshold	500k	400k	300k	200k	100k

SLC erase count	<5000	5000-10000	10000-20000	20000-25000	>25000
close block read count threshold	1000k	800k	600k	400k	200k
open block read count threshold	500k	400k	600k	200k	100k

超過臨界值後 我們設定後續的檢查 read count 的臨界值 100k

read operation

step add read count

當 ftl read req 時 不論是 user lu read , internal lu read 我們都可由 req 得到 SPB ID

每一個 ftl read req 都會統計到 該 SPB ID 所對應的 read count

step check read count

檢查 read count 是否有超過 該 SPB 所對應的 read count 臨界值

step input queue if need

如果超過 read count 臨界值 則將此 SPB 放到 check queue 中

如果 check queue full 則 read operation 記錄此 SPB 每次 read operation 都嘗試放到 check queue

如果此 ftl read req 已經發生 error bit 數 達到設定的臨界值 我們需要 refresh ,則將此 SPB (或是 sub SPB) 放到 refresh queue 中

如果 refresh queue full 則 read operation 記錄此 SPB 每次 read operation 都嘗試放到 refresh queue

check task

step trigger condition

check queue 不為空

設置 3 minutes 域值 避免影響 IO 速度

如果 check queue full 縮短 為 1.5 minutes 域值

step get one SPB

從 check queue 拿出一個 SPB

step read page check

對此 SPB 做 read page 的操作 Nand 廠家的情況 有所不同 所以根據不同廠家 進行不同位置的 read page 操作

如果拿不到 table 或者 拿到的顆粒很差 但又想數據可靠性 只能做大量的 check 最壞情況是所有 page

step input refresh queue if need

如果檢查過程當中 ftl read req 發生 error bit 數 達到設定的臨界值 我們需要 refresh ,則將此 SPB (或是 sub SPB) 放到 refresh queue 中

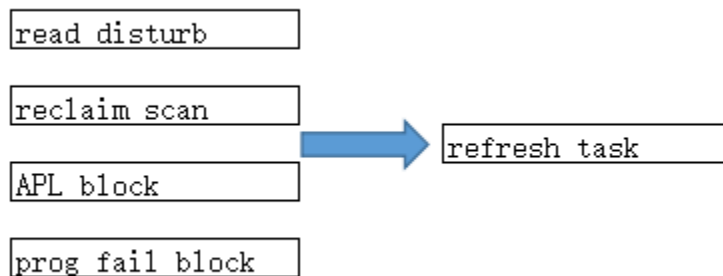
如果 refresh queue full 則 check task 保持在此 step 直到能放到 refresh queue 才能進入下一步

step output check queue

當我們檢查完 某 SPB(或是 sub SPB) 就移除該 SPB (或是 sub SPB)

refresh task

模塊關係



step trigger condition

refresh queue 不為空

step get one SPB

從 refresh queue 拿出一個 SPB

對於多個 SPB 我們需要決定 Refresh 順序

每個 queue node 我們紀錄 urgent flag 或是 error bit

每次從 queue 拿出 SPB 時 需要遍歷出 urgent flag 或是 error bit 最高的 優先取出

1 UECC 優先於 CECC

2 Meta 優先於 host data

step trigger refresh

根据此 SPB 的屬性 做出不同的 refresh 操作

User Lu SPB => 設定 User Lu need refresh flag 與 SPB ID, user lu gc task 會根據此訊息 對此 SPB 進行 refresh

Internal Lu SPB => 設定 internal Lu need refresh flag 與 SPB ID, internal lu gc task 會根據此訊息 對此 SPB 進行 refresh

system block => 調用 system block gc task 對此 system block 進行 refresh

block type	refresh interface	comment
User Lu SPB	user lu gc task	
Internal Lu SPB	internal lu gc task	
frb block		
srb block		
system log block		

step wait refresh done

等待 internal Lu need refresh flag 與 SPB ID, internal lu gc task 通知

step output refresh queue

當 refresh 完某 SPB(或是 sub SPB) 就移除該 SPB (或是 sub SPB)

User Lu gc task / internal Lu gc task

step trigger gc

檢查是否有任何 task 有使用 gc task 的需求

當檢查到 need refresh flag gc task 流程開始

step source SPB

在選擇源 SPB 步驟當中 設定 refresh task 所指定的 SPB 為源 SPB

step gc flow

後面進行同樣步驟

step gc done

當完成時 gc task 發送訊息通知 refresh task 某個 SPB 已完成

reclaim scan task

step trigger condition

除了檢查 read count 之外 我們也會週期性的把 SPB 放到 check queue 等待進一步的檢查

常溫下 設置 10 minutes 域值 避免影響 IO 速度

高溫下 我們線性縮短域值

step input queue

順序的把 SPB 放到 check queue 中

維護一個 SPB 的使用時間 如果能由 host 端獲取系統時間 我們就紀錄系統時間

如果無法獲取系統時間 我們就紀錄 SPB 的使用序號

system block

對於 system block 各個業務模塊提供給 reclaim scan task 需要檢查的 block ID, reclaim scan task 就只會檢查這些 block

PON APL case handle

PON APL 之後 我們做單獨的 reclaim scan

在 PON APL 之後 我們把所有的 SPB 都設置上 check flag 用來檢查所有的 SPB

同時 check task 也需要盡快檢查完所有的 SPB 不限速