

Системи датотека засновани на логовима

Оперативни системи 2

Факултет техничких наука Косовска Митровица

Драгиша Миљковић

Системи датотека засновани на логовима



Зашто су потребни системи датотека засновани на логовима?

Величина меморије се повећавала.

Велика разлика у перформансама насумичног I/O и секвенцијалног I/O.

Постојећи системи датотека имају лоше перформансе код неких уобичајених типова оптерећења.

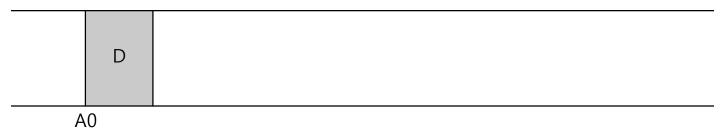
Системи датотека немају "свест" о RAID-у.



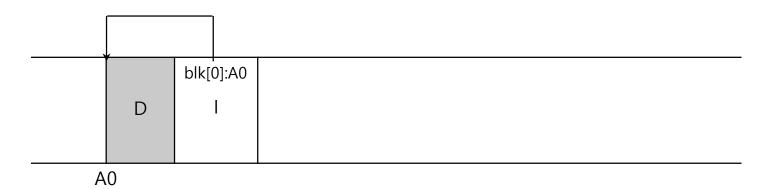
Секвенцијално уписивање на диск

Како да трансформишемо сва ажурирања у систем датотека у серију секвенцијалних уписивања на диск?

ажурирање података



■ метаподаци такође треба да буду ажурирани (нпр. inode)

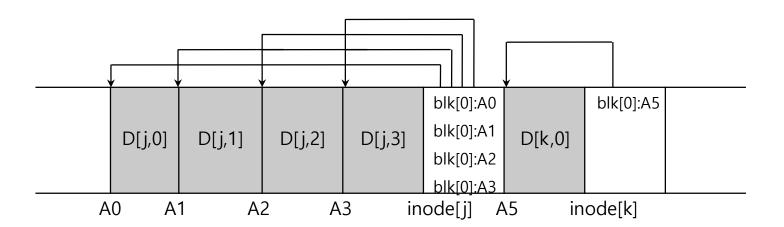




Секвенцијално и ефикасно уписивање на диск

Ефикасно уписивање са баферовањем уписивања.

- Води се евиденција ажурирања бафера меморије.
- Сва уписивања на диск се извршавају одједном (након што се у баферу сакупи довољан број ажурирања).





Колико података треба баферовати?

 $T \downarrow write = T \downarrow position + D/R \downarrow peak$

 $R \downarrow effective = D/T \downarrow write = D/T \downarrow position + D/R \downarrow peak$

 $R \downarrow effective = D/T \downarrow position + D/R \downarrow peak = F \times R \downarrow peak$

 $D = F \times R \downarrow peak \times (T \downarrow position + D/R \downarrow peak)$

 $D = (F \times R \downarrow peak \times T \downarrow position) + (F \times R \downarrow peak \times D/R \downarrow peak)$

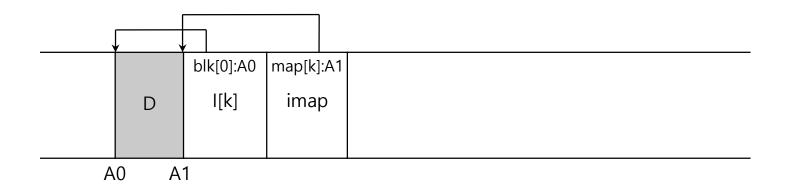
 $D = F/1 - F \times R \downarrow peak \times T \downarrow position$



Проналажење индексног чвора

Како пронаћи индексни чвор у FFS? FFS дели inode таблицу на комаде и смешта групе индексних чворова.

Решење је у индирекцији помоћу "inode мапе".



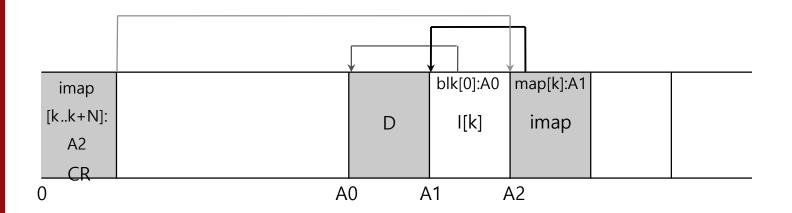


Регион контролне тачке

Како пронаћи inode мапу (она је разбацана по диску)?

 LFS систем датотека мора да садржи фиксирану локацију на диску од које започиње потрагу за датотеком.

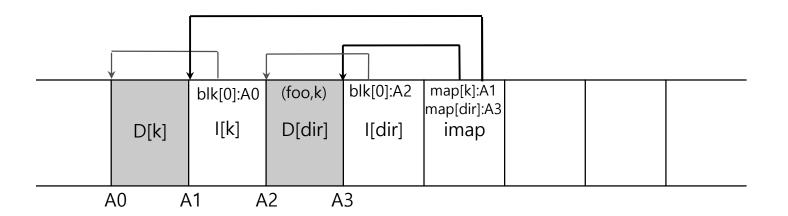
Регион контролне тачке садржи показиваче на најскорију inode мапу.





Шта са директоријима?

Како LFS чува податке о директоријуму?





Garbage collection

LFS непрестано уписује новије верзије датотека.

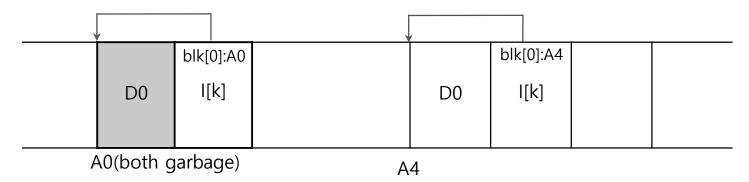
Ипак, LFS оставља старије верзије структура датотека по диску, ово се назива "ђубретом" (garbage).



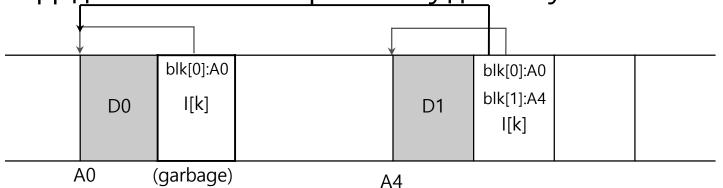
Garbage collection (наставак)

Пример сакупљања ђубрета.

Уписивање преко блокова података.



■ Додавање блока на оригиналну датотеку k.



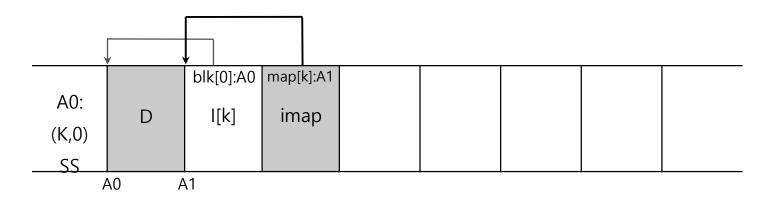


Одређивање живота блока

LFS на сваки сегмент додаје информације које описују сваки од блокова.

 Нпр. LFS укључује за сваки од блокова података његов inode број и његов офсет.

Ове информације се чувају у блоку резимеа сегмента.





Интегритет података и заштита



Режими отказа диска

	Cheap	Costly
LSEs	9.40%	1.40%
Corruption	0.50%	0.05%



Управљање латентним грешкама сектора

Латентне грешке сектора се лако откривају и њима се лако управља.

Коришћење механизама редудантности:

• У **mirrored RAID-**у или RAID-4 или RAID-5 системима заснованим на парности, систем треба да реконструише блок из других блокова у групи парности.



Откривање "корумпираних" података: контролна сума

Како клијент може да утврди да ли је блок "корумпиран"?

Коришћењем механизма контролне суме:

 Ово је просто резултат извршавања функције где се као улаз узима део података, а након извршавања функције над тим подацима се, као резултат, добија мали резиме садржаја тих података.



Изглед контролне суме

Изглед диска без контролне суме:

D0 D1 D2 D3 D4 D5 D6

Изглед диска са контролном сумом:

D0 [10] D1	[20] D2 [20]	D3 5 D4
------------	--------------	---------

■ Контролна сума се смешта у блокове од по 512 бајтова:

C[D0] C[D1] C[D2] C[D3] C[D4]	D0	D1	D2	D3	D4
---	----	----	----	----	----



Коришћење контролне суме

Када се чита блок D, клијент чита његову контролну суму са диска. Сачувана контролна сума – **Cs(D)**

Израчунавање контролне суме над преузетим блоком D. Израчуната контролна сума – **Сс(D)**

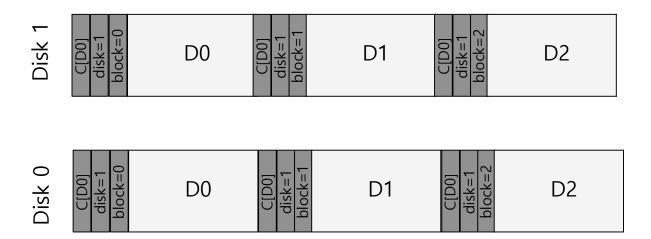
Упоређују се сачувана и израчуната контролна сума;

- Уколико су једнаке (Cs(D) == Cc(D)), подаци су исправни.
- Уколико се разликују (Cs(D) != Cc(D)), подаци су промењени у односу на последњи пут када су сачувани (јер сачувана контролна сума приказује вредност за податке који су постојали тада).



Проблеми контролне суме

Погрешно усмерена уписивања настају код дискова и RAID контролера код којих се подаци исправно упишу, али на погрешној локацији.





Проблеми контролне суме (наставак)

Изгубљена уписивања настају када уређај информише горњи слој да је уписивање извршено, а заправо није.

Scribbling ("шкрабање").

Додатни трошкови код контролног сумирања.

