Disques SSD

Thomas Lavergne
lavergne@lisn.fr

Principe

Support physique

- Sans support magnétique (*vs disque dur*)
- Réinscriptible de nombreuse fois (*vs CD*)

Principe

Support physique

- Sans support magnétique (*vs disque dur*)
- Réinscriptible de nombreuse fois (*vs CD*)

Un peu d'histoire

- Apparition des clefs USB fin années 90
  $\rightarrow$ premiers SSD
- Cartes mémoires début années 2000
- Smartphones : cartes mémoires comme disque

Évolution très rapide

- Jusqu'en 2013, plus cher et plus lent qu'un DD
  - 110 Mo/s pour un DD 500 Go à 80 euros
  - 50 Mo/s pour un SSD 120 Go à 80 euros

Évolution très rapide

- Jusqu'en 2013, plus cher et plus lent qu'un DD
  - 110 Mo/s pour un DD 500 Go à 80 euros
  - 50 Mo/s pour un SSD 120 Go à 80 euros
- Mais limites différentes des DD :
  - Pas de rotation + mouvement de bras

Évolution très rapide

- Jusqu'en 2013, plus cher et plus lent qu'un DD
  - 110 Mo/s pour un DD 500 Go à 80 euros
  - 50 Mo/s pour un SSD 120 Go à 80 euros
- Mais limites différentes des DD :
  - Pas de rotation + mouvement de bras

Évolution des SSD

- 2013 : 500Go à 500 Mo/s pour 1000 euros
- 2015 : 1To à 500 Mo/s pour 150 euros
- 2019 : 3Go/s en NVMe

Comment ça marche ?

Idée

On souhaite faire des supports amovibles sans bande magnétique

Comment ça marche ?

Idée

On souhaite faire des supports amovibles sans bande magnétique

On veut imiter le principe d'une RAM...

Comment ça marche ?

Idée

On souhaite faire des supports amovibles sans bande magnétique

On veut imiter le principe d'une RAM...

... mais sans avoir besoin de l'alimenter !

Comment ça marche ?

Idée

On souhaite faire des supports amovibles sans bande magnétique

On veut imiter le principe d'une RAM...

... mais sans avoir besoin de l'alimenter !

Maintenir chargé un bit sans bascule RS ?

Comment ça marche ?

Idée

On souhaite faire des supports amovibles sans bande magnétique

On veut imiter le principe d'une RAM...

... mais sans avoir besoin de l'alimenter !

Maintenir chargé un bit sans bascule RS ?

Mécanisme

Charger une grille électromagnétique de manière à ce que les électrons ne s'échappent pas

Effet tunnel (Fowler-Nordheim)

Une particule peut (parfois) franchir une grille même si elle a un potentiel trop faible

- *Effet de mécanique quantique*
  $\rightarrow$ *non déterministe*

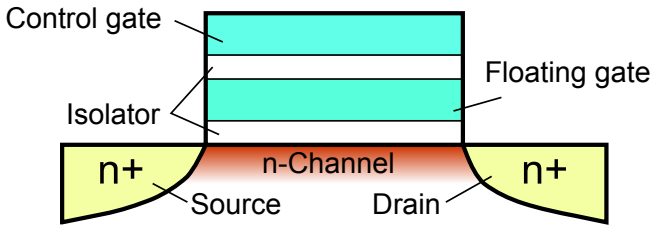
Effet tunnel (Fowler-Nordheim)

Une particule peut (parfois) franchir une grille même si elle a un potentiel trop faible

- *Effet de mécanique quantique*
  $\longrightarrow$ *non déterministe*

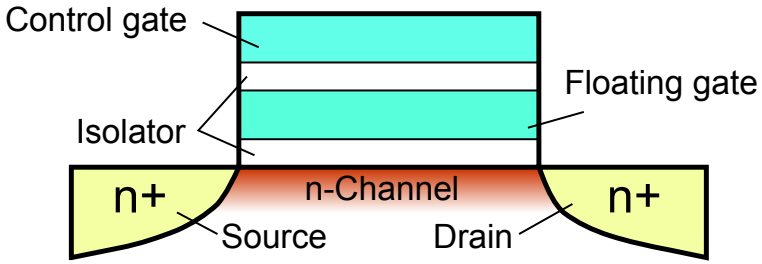
Bit SSD : mémoire flash NAND

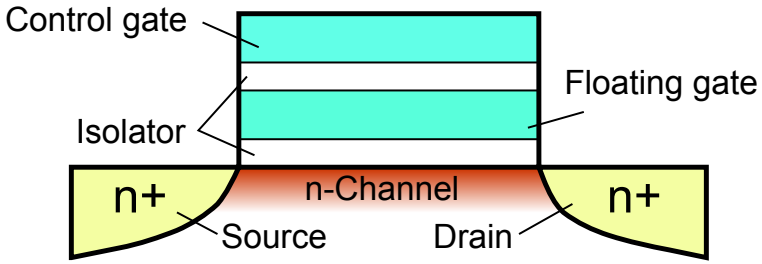
Transistor avec une grille flottante qui bloque les électrons



source : Widipedia

Le n-tunnel forme un substrat pour des électrons.

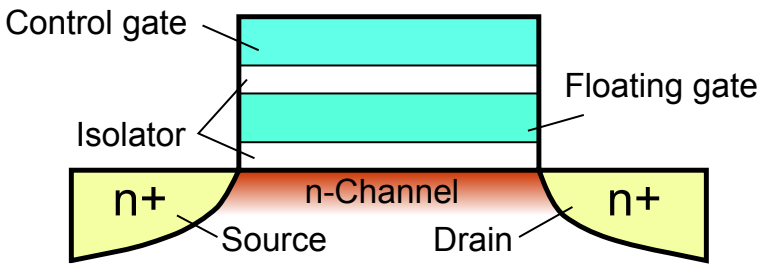

Control gate
Floating gate
Isolator
n+
n-Channel
n+
Source
Drain



Grille saturée

Le transistor est un isolant

$\longrightarrow$ Lecture d'un bit 0



Grille saturée

Le transistor est un isolant

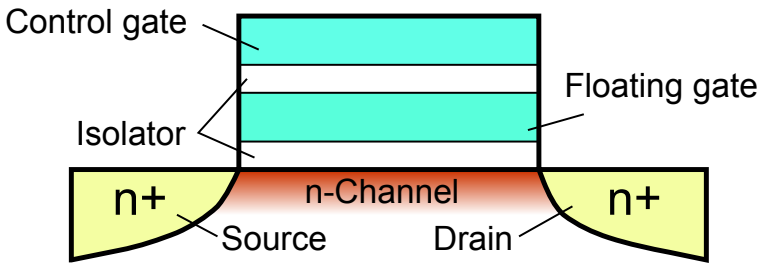
$\longrightarrow$ Lecture d'un bit 0

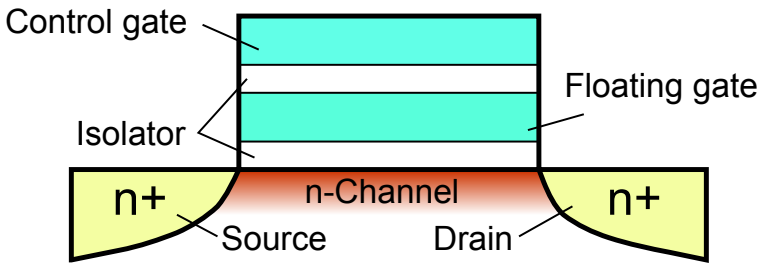
Grille vidée

Le transistor est un conducteur (effet tunnel)

$\longrightarrow$ Lecture d'un bit 1

Écriture


Control gate
Floating gate
Isolator
n+
n-Channel
n+
Source
Drain



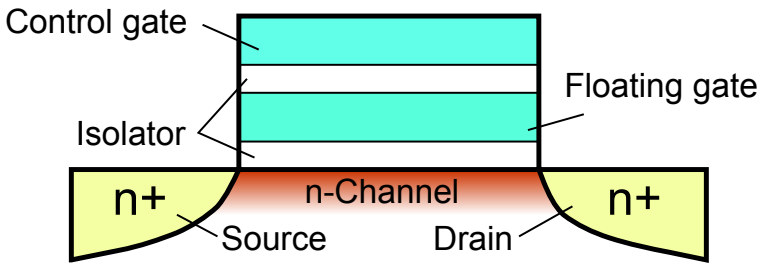
Écriture

Faible tension entre la source et le drain

Forte tension sur la grille

$\rightarrow$ Chargement de la grille

Écriture



Écriture

Faible tension entre la source et le drain

Forte tension sur la grille

$\longrightarrow$ Chargement de la grille

Éffacement

Faible tension entre le drain et la source

Forte tension sur la grille

$\longrightarrow$ Vidage de la grille

Pages *(blocs pour les DD)*

Unité de base du SSD, environ 4Ko

- Peut être lue
- Peut être programmée (si la page est vide)

Pages *(blocs pour les DD)*

Unité de base du SSD, environ 4Ko

- Peut être lue
- Peut être programmée (si la page est vide)

Blocs

Groupe de pages, environ 128 pages par bloc

- Peut être éffacé
- Contiens un code d'erreur

Pages *(blocs pour les DD)*

Unité de base du SSD, environ 4Ko

- Peut être lue
- Peut être programmée (si la page est vide)

Blocs

Groupe de pages, environ 128 pages par bloc

- Peut être éffacé
- Contiens un code d'erreur

Tampon

Le controleur dispose d'une RAM interne

- Lecture d'un bloc complet vers la RAM interne
- Transfert entre la RAM interne et le système

Problème d'usure

Grille flottante

Données = (dé)saturation de la grille flottante

Grille flottante

Données = (dé)saturation de la grille flottante

- Il reste toujours des électrons...

Au fur et à mesure, la grille se désature de moins en moins

Usure

Entre 10 000 et 100 000 cycles d'écriture maximum

Grille flottante

Données = (dé)saturation de la grille flottante

- Il reste toujours des électrons...

Au fur et à mesure, la grille se désature de moins en moins

Usure

Entre 10 000 et 100 000 cycles d'écriture maximum

Politique de gestion de l'usure

Allocation des blocs

Stratégie de base

Bloc logique $\rightarrow$ adresse physique (secteur)

Allocation des blocs

Stratégie de base

Bloc logique $\rightarrow$ adresse physique (secteur)

Inconvénients

X Effacer le secteur + recopier nouvelle valeur

$\rightarrow$ lent !

Allocation des blocs

Stratégie de base

Bloc logique $\rightarrow$ adresse physique (secteur)

Inconvénients

- ✗ Effacer le secteur + recopier nouvelle valeur
  $\rightarrow$ lent !
- ✗ Fichiers souvent modifiés (ex : données d'état)
  $\rightarrow$ secteurs plus vite inutilisables

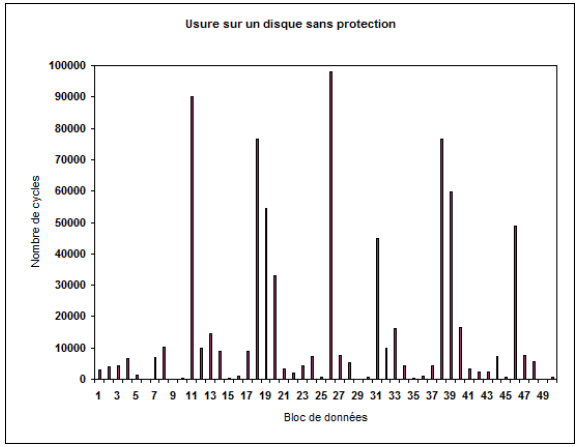
Allocation des blocs

Stratégie de base

Bloc logique $\rightarrow$ adresse physique (secteur)

Inconvénients

- ✗ Effacer le secteur + recopier nouvelle valeur
  $\rightarrow$ lent !
- ✗ Fichiers souvent modifiés (ex : données d'état)
  $\rightarrow$ secteurs plus vite inutilisables



source : http://tomsharedware.fr

Gestion de l'usure

ou *wear levelling*

Gestion de l'usure

ou *wear levelling*

Disque

Le disque est inutilisable dès que X blocs ont atteint leur limite, quelque soit l'état des autres blocs !

Gestion de l'usure

ou *wear levelling*

Disque

Le disque est inutilisable dès que X blocs ont atteint leur limite, quelque soit l'état des autres blocs !

Remarques

Pour les SSD pas de problème de temps d'accès :

tous les secteurs sont accessibles à la même vitesse

Plus de correspondance directe nécessaire :

adresse logique $\neq$ adresse physique

Wear Leveling Dynamique

Wear Leveling dynamique

Principe

Changement de secteur lors de l'éciture

Wear Leveling dynamique

Principe

Changement de secteur lors de l'éciture

Avantages

✓ Répartir l'écriture sur les secteurs libres

$\rightarrow$ Moins d'usure

Wear Leveling dynamique

Principe

Changement de secteur lors de l'éciture

Avantages

✓ Répartir l'écriture sur les secteurs libres

→ Moins d'usure

✓ Libérer les secteurs quand le périphérique est inutilisé

→ Gain de performances

WL dynamique : fonctionnement I

Table d'adresse des blocs logiques

- Adresse physique (secteur)
- Adresse Logique (n° *bloc ou « libre »*)

- Usure (nombre d'écriture)

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	bloc :	1	4	2	6	0	L	9	3	L	L	5	7	8	L	L	L	
	date :	53	41	106	99	72	87	55	31	80	58	64	67	91	61	0	0	
	usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0	
	invalid :	F	F	F	F	F	F	F	F	T	F	F	F	F	T	F	F	

WL dynamique : fonctionnement I

Table d'adresse des blocs logiques

- Adresse physique (secteur)
- Adresse Logique (*n° bloc ou « libre »*)
- Date dernière modification
- Usure (nombre d'écriture)

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	bloc :	1	4	2	6	0	L	9	3	L	L	5	7	8	L	L	L
	date :	53	41	106	99	72	87	55	31	80	58	64	67	91	61	0	0
	usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0
	invalid :	F	F	F	F	F	F	F	F	T	F	F	F	F	T	F	F

WL dynamique : fonctionnement I

Table d'adresse des blocs logiques

- Adresse physique (secteur)
- Adresse Logique (*n° bloc ou « libre »*)
- Date dernière modification
- Usure (nombre d'écriture)
- Bit d'invalidité du secteur

secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc :	1	4	2	6	0	L	9	3	L	L	5	7	8	L	L	L
date :	53	41	106	99	72	87	55	31	80	58	64	67	91	61	0	0
usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0
invalid :	F	F	F	F	F	F	F	F	T	F	F	F	F	T	F	F

WL dynamique : fonctionnement I

Table d'adresse des blocs logiques

- Adresse physique (secteur)
- Adresse Logique (n° *bloc ou « libre »*)
- Date dernière modification
- Usure (nombre d'écriture)
- Bit d'invalidité du secteur

Fonctionnement

- Écriture : choix nouvelle adresse physique
  + marquer secteur invalide et libre

WL dynamique : fonctionnement I

Table d'adresse des blocs logiques

- Adresse physique (secteur)
- Adresse Logique (n° *bloc ou « libre »*)
- Date dernière modification
- Usure (nombre d'écriture)
- Bit d'invalidité du secteur

Fonctionnement

- Écriture : choix nouvelle adresse physique
  + marquer secteur invalide et libre
- Temps libre : nettoyer secteurs invalides

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	logique :	1	4	2	6	0	L	9	3	L	L	5	7	8	L	L	L	
	usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0	

Demandes d'écriture : A

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	logique :	1	4	2	6	0	L	9	3	L	L	5	7	8	L	L	L	
	usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0	

Demandes d'écriture : A

En cas d'égalité, je prend le premier secteur libre

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	logique :	1	4	2	6	0	L	9	3	L	L	5	7	8	L	A	L
	usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	0

Demandes d'écriture : A 1

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
logique :	1	4	2	6	0	L	9	3	L	L	5	7	8	L	A	L
usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	0

Demandes d'écriture : A 1

Libérer le secteur contenant le bloc 1

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
logique :	L	4	2	6	0	L	9	3	L	L	5	7	8	L	A	1
usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1

Demandes d'écriture : A 1 7

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	logique :	L	4	2	6	0	L	9	3	L	L	5	7	8	L	A	1	
	usure :	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1	

Demandes d'écriture : A 1 7

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
logique :	7	4	2	6	0	L	9	3	L	L	5	L	8	L	A	1
usure :	3	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1

Demandes d'écriture : A 1 7 1

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	logique :	7	4	2	6	0	L	9	3	L	L	5	L	8	L	A	1	
	usure :	3	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1	

Demandes d'écriture : A 1 7 1

Même si mon secteur est moins usé, je déplace

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	logique :	7	4	2	6	0	L	9	3	L	L	5	L	8	1	A	L	
	usure :	3	8	7	3	6	5	8	6	7	9	6	8	8	4	1	1	

Demandes d'écriture : A 1 7 1 2

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	logique :	7	4	2	6	0	L	9	3	L	L	5	L	8	1	A	L	
	usure :	3	8	7	3	6	5	8	6	7	9	6	8	8	4	1	1	

Demandes d'écriture : A 1 7 1 2

WL dynamique : fonctionnement II

Choix d'une nouvelle adresse physique

Liste triée des secteurs libres *(et valides)* par usure

- Choisir le secteur libre le moins usé

Exemple :

	secteur :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	logique :	7	4	L	6	0	L	9	3	L	L	5	L	8	1	A	2	
	usure :	3	8	7	3	6	5	8	6	7	9	6	8	8	4	1	2	

Demandes d'écriture : A 1 7 1 2

Usure des *blocs*

Seule l'usure des blocs remplacés est égalisée :

✗ les blocs lus n'usent pas !

WL dynamique : limites

Usure des *blocs*

Seule l'usure des blocs remplacés est égalisée :

✗ les blocs lus n'usent pas !

Problème

Beaucoup de données ne sont jamais réécrites

- Fichiers de l'OS
- Archives
- Photos, vidéos, musiques…

WL dynamique : limites

Usure des *blocs*

Seule l'usure des blocs remplacés est égalisée :

✗ les blocs lus n'usent pas !

Problème

Beaucoup de données ne sont jamais réécrites

- Fichiers de l'OS
- Archives
- Photos, vidéos, musiques…

Elles sont statique

Wear Leveling statique

Wear Leveling statique

Principe

Déplacer les blocs qui ne sont pas écrits

$\longrightarrow$ choisir systématiquement le secteur le moins usé

Wear Leveling statique

Principe

Déplacer les blocs qui ne sont pas écrits

$\longrightarrow$ choisir systématiquement le secteur le moins usé

Implémentation

Table des blocs logiques :

- secteur, bloc, bit d'invalidité, usure,
- date (en nombre d'écritures sur le SSD)

Wear Leveling statique

Principe

Déplacer les blocs qui ne sont pas écrits

$\longrightarrow$ choisir systématiquement le secteur le moins usé

Implémentation

Table des blocs logiques :

- secteur, bloc, bit d'invalidité, usure,
- date (en nombre d'écritures sur le SSD)

Bloc statique/dynamique

Un bloc modifié depuis moins de t_l pas de temps est considéré dynamique.

$\longrightarrow$ Il ne peut pas être déplacé.

Algorithme

Chercher le secteur le moins utilisé

- Il est libre $\longrightarrow$ l'utiliser

Algorithme

Chercher le secteur le moins utilisé

- Il est libre $\rightarrow$ l'utiliser
- Il contient un bloc dynamique $\rightarrow$ choisir un autre secteur

Algorithme

Chercher le secteur le moins utilisé

- Il est libre $\rightarrow$ l'utiliser
- Il contient un bloc dynamique $\rightarrow$ choisir un autre secteur
- Il contient un bloc statique $\rightarrow$ déplacer le bloc (suivant le même algorithme)

WL statique : algorithme

Algorithme

Chercher le secteur le moins utilisé

- Il est libre $\longrightarrow$ l'utiliser
- Il contient un bloc dynamique $\longrightarrow$ choisir un autre secteur
- Il contient un bloc statique $\longrightarrow$ déplacer le bloc

(suivant le même algorithme)

Implémentation

Liste des secteurs libres ou contenant des blocs statiques... *triée par usure croissante*

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	1	2	6	0	L	9	3	L	L	5	7	8	L	L	L
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	0	0
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	1	2	6	0	L	9	3	L	L	5	7	8	L	L	L
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	0	0
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0

Demandes d'écriture : A

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	1	2	6	0	L	9	3	L	L	5	7	8	L	L	L
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	0	0
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	0	0

Demandes d'écriture : A

En cas d'égalité, je prend le premier secteur

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	1	2	6	0	L	9	3	L	L	5	7	8	L	A	L
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	107	0
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	0

Demandes d'écriture : A 1

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	1	2	6	0	L	9	3	L	L	5	7	8	L	A	L
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	107	0
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	0

Demandes d'écriture : A 1

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	L	2	6	0	L	9	3	L	L	5	7	8	L	A	1
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	107	108
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1

Demandes d'écriture : A 1 7

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	L	2	6	0	L	9	3	L	L	5	7	8	L	A	1
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	107	108
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1

Demandes d'écriture : A 1 7

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	L	2	6	0	L	9	3	L	L	5	L	8	L	A	1
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	107	108
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1

Demandes d'écriture : A 1 7

Le bloc 7 est retiré

Les blocs dynamiques sont protégés

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	4	L	2	6	0	L	9	3	L	L	5	L	8	L	A	1
date	53	41	106	92	72	87	55	31	80	58	64	67	91	61	107	108
usure	2	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1

Demandes d'écriture : A 1 7

Le bloc 4 est statique

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	6	0	L	9	3	L	L	5	L	8	L	A	1
date	109	41	106	92	72	87	55	31	80	58	64	67	91	61	107	108
usure	3	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1

Demandes d'écriture : A 1 7 4

Il faut maintenant replacer le bloc 4 !

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	6	0	L	9	3	L	L	5	L	8	L	A	1
date	109	41	106	92	72	87	55	31	80	58	64	67	91	61	107	108
usure	3	8	7	3	6	5	8	6	7	9	6	8	8	3	1	1

Demandes d'écriture : A 1 7 4

En cas d'égalité, on préfère un secteur libre

plutôt qu'un bloc statique qui serait déplacé. . . dans ce même bloc !

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	6	0	L	9	3	L	L	5	L	8	4	A	1
date	109	41	106	92	72	87	55	31	80	58	64	67	91	109	107	108
usure	3	8	7	3	6	5	8	6	7	9	6	8	8	4	1	1

Demandes d'écriture : A 1 7 4

On peut considérer qu'on est toujours au cycle 109...

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	6	0	L	9	3	L	L	5	L	8	4	A	1
date	109	41	106	92	72	87	55	31	80	58	64	67	91	109	107	108
usure	3	8	7	3	6	5	8	6	7	9	6	8	8	4	1	1

Demandes d'écriture : A 1 7 4 1

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	6	0	L	9	3	L	L	5	L	8	4	A	1
date	109	41	106	92	72	87	55	31	80	58	64	67	91	109	107	108
usure	3	8	7	3	6	5	8	6	7	9	6	8	8	4	1	1

Demandes d'écriture : A 1 7 4 1

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	1	0	L	9	3	L	L	5	L	8	4	A	L
date	109	41	106	110	72	87	55	31	80	58	64	67	91	109	107	108
usure	3	8	7	4	6	5	8	6	7	9	6	8	8	4	1	1

Demandes d'écriture : A 1 7 4 1 6

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	1	0	L	9	3	L	L	5	L	8	4	A	L
date	109	41	106	110	72	87	55	31	80	58	64	67	91	109	107	108
usure	3	8	7	4	6	5	8	6	7	9	6	8	8	4	1	1

Demandes d'écriture : A 1 7 4 1 6

Un secteur libre n'est jamais « dynamique » !

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	1	0	L	9	3	L	L	5	L	8	4	A	6
date	109	41	106	110	72	87	55	31	80	58	64	67	91	109	107	110
usure	3	8	7	4	6	5	8	6	7	9	6	8	8	4	1	2

Demandes d'écriture : A 1 7 4 1 6 2

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	2	1	0	L	9	3	L	L	5	L	8	4	A	6
date	109	41	106	110	72	87	55	31	80	58	64	67	91	109	107	110
usure	3	8	7	4	6	5	8	6	7	9	6	8	8	4	1	2

Demandes d'écriture : A 1 7 4 1 6 2

WL statique : exemple

Principe

Choisir le secteur le moins utilisé et déplacer si bloc statique

Exemple : $t_l = 10$ et date $= 107^e$ écriture

sect.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
bloc	7	L	L	1	0	2	9	3	L	L	5	L	8	4	A	6
date	109	41	106	110	72	111	55	31	80	58	64	67	91	109	107	110
usure	3	8	7	4	6	6	8	6	7	9	6	8	8	4	1	2

Demandes d'écriture : A 1 7 4 1 6 2

Wear Leveling statique

Avantage

Bien meilleure répartition de l'usure

- ✓ Écriture sur le disque
  - → les blocs sont déplacés sur tout le support
- ✓ L'écart d'usure maximum est t_l
- ✓ Moyenne d'utilisation faible
  - → mais pas homogène

RAID

Disques RAID

- *Redundant* : duplication
- *Array* : en parallèle, données réparties
- *Inexpensive* : pas cher

Disques RAID

- *Redundant* : duplication
- *Array* : en parallèle, données réparties
- *Inexpensive* : pas cher

Principe

Utilisation en parallèle de disques sur lesquels les données sont réparties et dupliquées

Disques RAID

- *Redundant* : duplication
- *Array* : en parallèle, données réparties
- *Inexpensive* : pas cher

Principe

Utilisation en parallèle de disques sur lesquels les données sont réparties et dupliquées

Améliorer

✓ La performance

✓ La fiabilité

Coût du matériel

Le coût d'un disque croît de manière exponentielle avec sa performance.

Si un disque qui traite n requêtes à la seconde coûte m euros, un disque qui traite $2n$ requêtes à la seconde coûte m^2 euros.

Coût du matériel

Le coût d'un disque croît de manière exponentielle avec sa performance.

Si un disque qui traite n requêtes à la seconde coûte m euros, un disque qui traite $2n$ requêtes à la seconde coûte m^2 euros.

Disques RAID

Utiliser plusieurs disques en parallèle :

- Un contrôleur réparti les requêtes sur les disques

Un tel système coûte $2m + \epsilon$ euros

RAID 0

Principe

Utiliser la redondance pour améliorer la performance

- Entrelacement des données sur les disques

Chaque bloc est écrit sur un disque différent, modulo n disques

RAID 0

Principe

Utiliser la redondance pour améliorer la performance

- Entrelacement des données sur les disques

Chaque bloc est écrit sur un disque différent, modulo n disques

Disque virtuel

Le disque RAID fonctionne comme un disque avec des blocs n fois plus grands (ou n fois plus rapides)

RAID 0

Principe

Utiliser la redondance pour améliorer la performance

- Entrelacement des données sur les disques

Chaque bloc est écrit sur un disque différent, modulo n disques

Disque virtuel

Le disque RAID fonctionne comme un disque avec des blocs n fois plus grands (ou n fois plus rapides)

Avantages

✓ Temps de traitement des « petits » accès

✓ Temps de traitement des « grands » accès

Panne de matériel

Un disque tombe en panne toutes les 100 000 heures env. (11 ans).

$\longrightarrow$ Dans un parc de 100 disques indépendants, une panne tous les 41 jours environ !

Panne de matériel

Un disque tombe en panne toutes les 100 000 heures env. (11 ans).

$\rightarrow$ Dans un parc de 100 disques indépendants, une panne tous les 41 jours environ !

Sauvegarde des données

Sauvegarde à intervalle de temps réguliers

X Perte des données non-encore sauvegardées

Panne de matériel

Un disque tombe en panne toutes les 100 000 heures env. (11 ans).

$\longrightarrow$ Dans un parc de 100 disques indépendants, une panne tous les 41 jours environ !

Sauvegarde des données

Sauvegarde à intervalle de temps réguliers

X Perte des données non-encore sauvegardées

Disques RAID

Stocker *plus* pour récupérer les pannes

- Données redondantes
- Code correcteur

Principe

Utiliser la redondance pour améliorer la fiabilité

- Mirroring : données dupliquées
- Shadowing : données recopiées

Il faut 2 disques de capacité n pour stocker n :

Principe

Utiliser la redondance pour améliorer la fiabilité

- Mirroring : données dupliquées
- Shadowing : données recopiées

Il faut 2 disques de capacité n pour stocker n :

simple mais coûteux !

Principe

Utiliser la redondance pour améliorer la fiabilité

- Mirroring : données dupliquées
- Shadowing : données recopiées

Il faut 2 disques de capacité n pour stocker n :

simple mais coûteux !

Avantage

✓ Réduit effectivement les pannes...

dépannage = 10h $\rightarrow$ 1 panne toutes les 57 000 ans

Principe

Utiliser la redondance pour améliorer la fiabilité

- Mirroring : données dupliquées
- Shadowing : données recopiées

Il faut 2 disques de capacité n pour stocker n :

simple mais coûteux !

Avantage

✓ Réduit effectivement les pannes...

dépannage = 10h $\rightarrow$ 1 panne toutes les 57 000 ans

...si elles sont indépendantes !

Principe

Utiliser la redondance pour améliorer la fiabilité

- Mirroring : données dupliquées
- Shadowing : données recopiées

Il faut 2 disques de capacité n pour stocker n :

simple mais coûteux !

Avantage

✓ Réduit effectivement les pannes...

dépannage = 10h $\rightarrow$ 1 panne toutes les 57 000 ans

...si elles sont indépendantes !

X 1 disque logique = 2 disques physiques

Principe

- Code d'erreur $\longrightarrow$ détecter secteur défectueux
- Code correcteur $\longrightarrow$ détecter et réparer !

Principe

- Code d'erreur $\longrightarrow$ détecter secteur défectueux
- Code correcteur $\longrightarrow$ détecter et réparer !

Exemple

Tripler toute l'information :

$$0 \longrightarrow 000 \qquad 1 \longrightarrow 111$$

- 3 bits différents $\longrightarrow$ erreur
- Vote majoritaire $\longrightarrow$ corriger

Très coûteux en espace ($\times 3$) !

Construction

Code correcteur $= \mathrm{XOR}(V_1, V_2, \ldots V_n)$

Construction

Code correcteur = XOR(V_1, V_2, ... V_n)

Exemple

- octets 5F 32 7A
- 5F $\oplus$ 32 $\oplus$ 7A =

Construction

Code correcteur = XOR(V_1, V_2, ... V_n)

Exemple

- octets 5F 32 7A
- 5F $\oplus$ 32 $\oplus$ 7A =

```
0101 1111

0011 0010

0111 1010
```

Construction

Code correcteur = XOR(V_1, V_2, ... V_n)

Exemple

- octets 5F 32 7A
- 5F $\oplus$ 32 $\oplus$ 7A =

```
0101 1111

0011 0010

0111 1010

0001 0111
```

Construction

Code correcteur $= \text{XOR}(V_1, V_2, \ldots V_n)$

Exemple

- octets 5F 32 7A
- $5\text{F} \oplus 32 \oplus 7\text{A} = 17$

```
0101 1111

0011 0010

0111 1010

0001 0111
```

Construction

Code correcteur $= \mathrm{XOR}(V_1, V_2, \ldots V_n)$

Exemple

- octets 5F 32 7A
- $5\mathrm{F} \oplus 32 \oplus 7\mathrm{A} = 17$

```
0101 1111

0011 0010

0111 1010

0001 0111
```

- Perte de l'octet 32 : $5\mathrm{F} \oplus 7\mathrm{A} \oplus 17 =$

Construction

Code correcteur $= \mathrm{XOR}(V_1, V_2, \ldots V_n)$

Exemple

- octets 5F 32 7A
- $5\mathrm{F} \oplus 32 \oplus 7\mathrm{A} = 17$

```
0101 1111
0011 0010
0111 1010
0001 0111
```

```
0101 1111
0001 0111
0111 1010
```

- Perte de l'octet 32 : $5\mathrm{F} \oplus 7\mathrm{A} \oplus 17 =$

Construction

Code correcteur $= \mathrm{XOR}(V_1, V_2, \ldots V_n)$

Exemple

- octets 5F 32 7A
- $5\mathrm{F} \oplus 32 \oplus 7\mathrm{A} = 17$

```
0101 1111
0011 0010
0111 1010
0001 0111
```

```
0101 1111
0001 0111
0111 1010
0011 0010
```

- Perte de l'octet 32 : $5\mathrm{F} \oplus 7\mathrm{A} \oplus 17 = 32$

Code de Hamming

Principe

Generalisation de la parité à m disques de correction.

Principe

Generalisation de la parité à m disques de correction.

Exemple : code de Hamming $(7,4)$

$$D_0, D_1, D_2, D_3, C_0, C_1, C_2$$

Principe

Generalisation de la parité à m disques de correction.

Exemple : code de Hamming $(7,4)$

$$D_0, D_1, D_2, D_3, C_0, C_1, C_2$$

Paritées :

- C_0 code la parité de $D_0 \oplus D_1 \oplus D_3$
- C_1 code la parité de $D_0 \oplus D_2 \oplus D_3$
- C_2 code la parité de $D_1 \oplus D_2 \oplus D_3$

Principe

Generalisation de la parité à m disques de correction.

Exemple : code de Hamming $(7,4)$

$$D_0, D_1, D_2, D_3, C_0, C_1, C_2$$

Paritées :

- C_0 code la parité de $D_0 \oplus D_1 \oplus D_3$
- C_1 code la parité de $D_0 \oplus D_2 \oplus D_3$
- C_2 code la parité de $D_1 \oplus D_2 \oplus D_3$

Vérification :

- $C_0 + D_0 + D_1 + D_3 = 0$
- $C_1 + D_0 + D_2 + D_3 = 0$
- $C_2 + D_1 + D_2 + D_3 = 0$

Principe

RAID 0 (volume en bande) + code correcteur de Hamming(n,m) sur chaque « bande »

- m disques de données $\longrightarrow$ $n - m$ disques de code correcteur
- Écriture de bande bit par bit

RAID 2

Principe

RAID 0 (volume en bande) + code correcteur de Hamming(n,m) sur chaque « bande »

- m disques de données $\rightarrow$ $n - m$ disques de code correcteur
- Écriture de bande bit par bit

Avantages et inconvénients

✓ Performance + fiabilité

X Écriture par bit...

$\rightarrow$ obsolète

Principe

- RAID 0 en bande
  - RAID 3 = bande par octets
  - RAID 4 = bande par blocs
- Disque de « parité »

Principe

- RAID 0 en bande
  - RAID 3 = bande par octets
  - RAID 4 = bande par blocs
- Disque de « parité »

Code correcteur

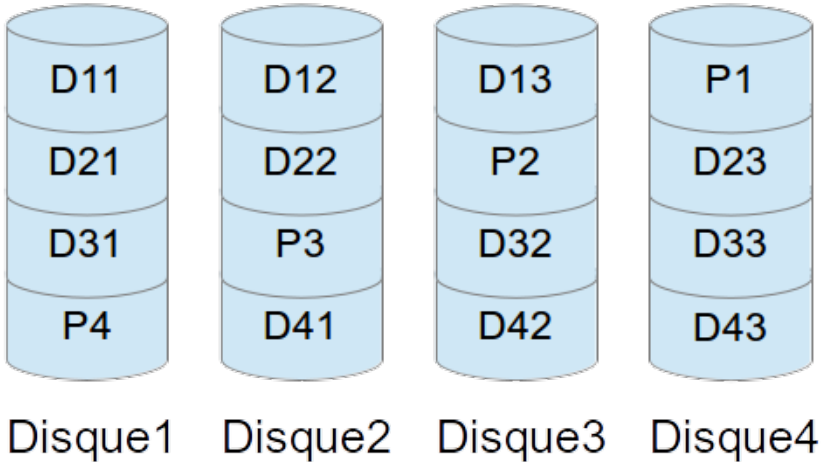
✓ Si n'importe quel disque tombe en panne il peut être reconstruit

X Les disques de parité sont beaucoup plus sollicités

→ pannes plus fréquentes !

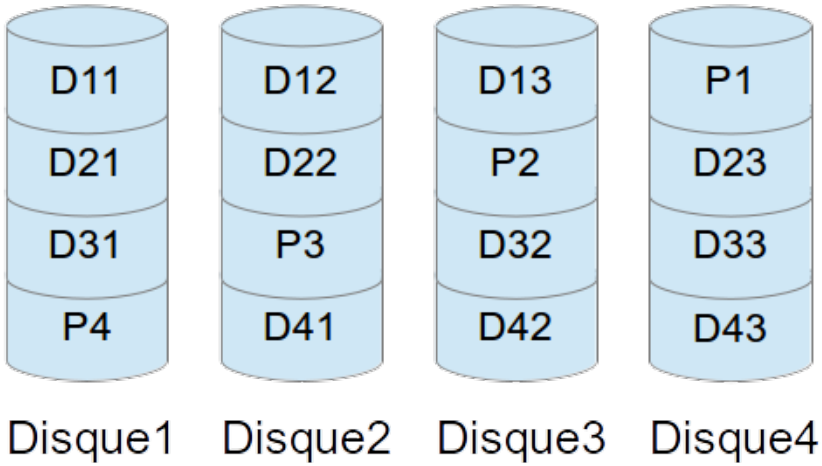
Principe

Agrégation par bande mais blocs de parité répartis



Principe

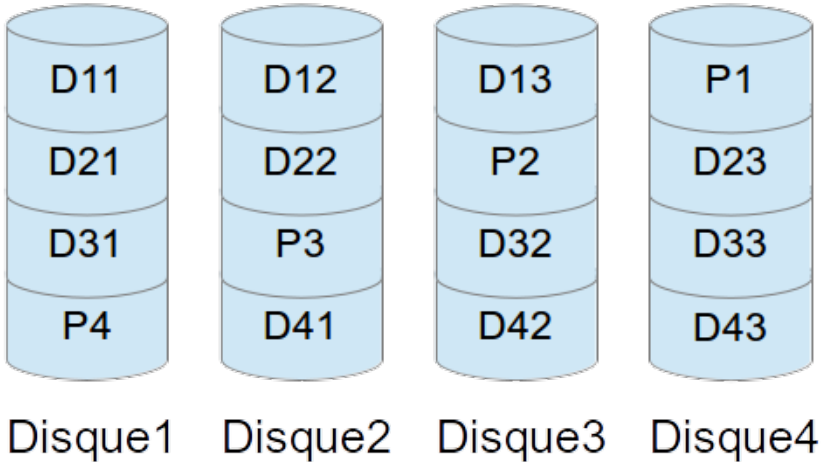
Agrégation par bande mais blocs de parité répartis



✓ Répartition de l'usure sur tous les disques

Principe

Agrégation par bande mais blocs de parité répartis



✓ Répartition de l'usure sur tous les disques

RAID 6

Même principe avec m disques de parité au lieu de 1 seul

$\rightarrow$ supporte la perte de $m - 1$ disques