RAID

Thomas Lavergne
lavergne@lisn.fr

Disques RAID

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

Redundant Array of Inexpensive Disks

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

Redundant Array of Inexpensive Disks

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

RAID 1

Principe

Utiliser la redondance pour améliorer la fiabilité

- Mirroring: données dupliquées
- Shadowing: données recopiées

Il faut 2 disques de capacité n pour stocker n :

RAID 1

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!

RAID 1

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

RAID 1

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!

RAID 1

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

Exemple

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

```
0101 1111

0011 0010

0111 1010

0001 0111
```

Construction

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

Exemple

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

```
0101 1111

0011 0010

0111 1010

0001 0111
```

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

Construction

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

Exemple

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

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

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

- Perte de l'octet 32 : 5F $\oplus$ 7A $\oplus$ 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.

Code de Hamming

Principe

Generalisation de la parité à m disques de correction.

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

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

Code de Hamming

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

Avantages et inconvénients

✓ Performance + fiabilité

X Écriture par bit...

$\longrightarrow$ obsolète

RAID 3 et 4

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

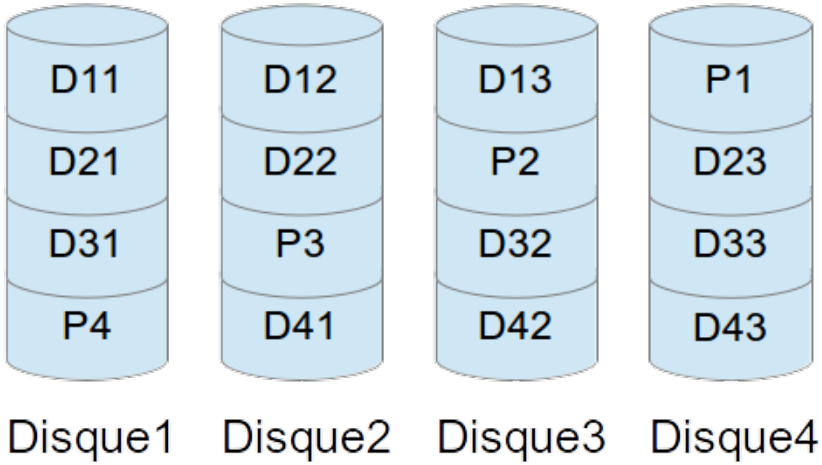
✓ Si n'importe quel disque tombe en panne il peut être reconstruit

X Les disques de parité sont beaucoup plus sollicités

→ pannes plus fréquentes!

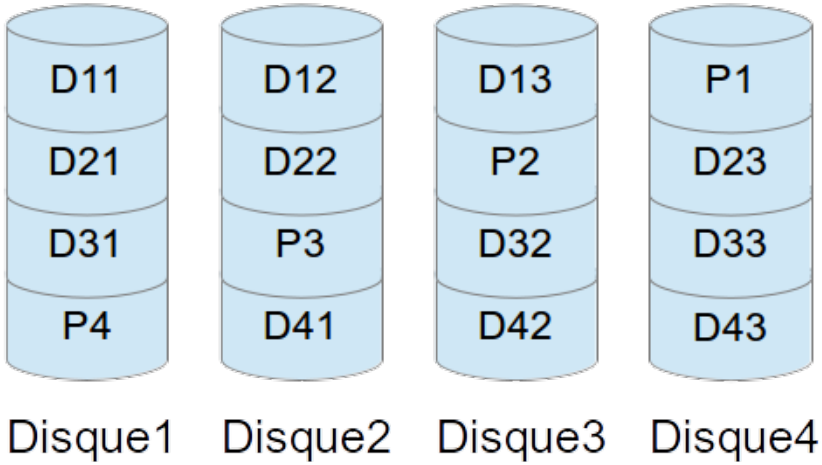
Principe

Agrégation par bande mais blocs de parité répartis



Principe

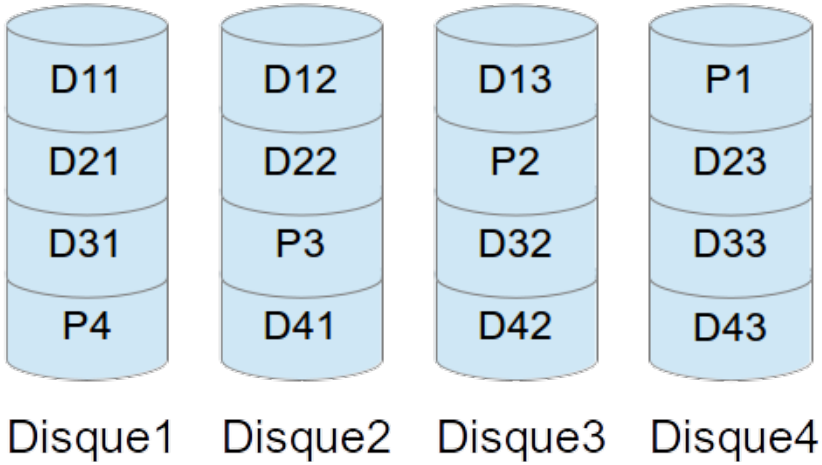
Agrégation par bande mais blocs de parité répartis



✓ 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