

Atelier d'Optimisation de Base de Données Cinématographique

Analyse d'Optimisation d'une Requête Complexe sur une Base de Données de Films

Objectif : Analyser en détail le plan d'exécution d'une requête complexe impliquant plusieurs jointures sur une base de données cinématographique.

Méthodologie :

Examen approfondi d'un plan d'exécution généré par le SGBD, incluant :

- Temps de réponse et d'exécution pour chaque étape
- Nombre de tuples traités
- Mémoire allouée
- Techniques de jointure utilisées
- Utilisation du parallélisme

Schéma de la base :

Film (**idFilm**, titre, année, genre, résumé, *idRéalisateur*, *codePays*)

Pays (**code**, nom, langue)

Artiste (**idArtiste**, nom, prénom, annéeNaiss)

Rôle (**idFilm**, **idActeur**, nomRôle)

Internaute (**email**, nom, prénom, région)

Notation (**email**, **idFilm**, note)

Métriques analysées :

Temps d'exécution (en millisecondes)

Mémoire allouée (en octets)

1) Analyse des plans d'exécution 1^{ier} cas d'une petite, 2^{ier} cas d'une moyenne table, et 3^{ier} cas d'une grande base de données pour la requête:

Select titre from Film

Plan d'exécution sur la base films

Petite base des films

Arbre des opérateurs

- Parcours séquentiel de film (temps de réponse:0.00 ; temps d'exécution:18.25 ; nombre de nuplets:325 ; mémoire allouée:16)

Grande base des films

Plan d'exécution sur la base maxifilms

Les temps sont exprimés en millisecondes et la mémoire allouée en octets.

Arbre des opérateurs

- Parcours séquentiel de film (temps de réponse: 0.00 ; temps d'exécution:17688.92 ; nombre de nuplets:325392 ; mémoire allouée:16)

Grande base avec index

Plan d'exécution sur la base maxindex

Les temps sont exprimés en millisecondes et la mémoire allouée en octets.

Arbre des opérateurs

- Parcours séquentiel de film (temps de réponse :0.00 ; temps d'exécution :17684.00 ; nombre de nuplets:325000 ; mémoire allouée:16)

2) Analyse des plans d'exécution 1^{ier} cas d'une petite, 2^{ier} cas d'une moyenne table, et 3^{ier} cas d'une grande base de données pour la requete:

Quel metteur en scène a tourné dans ses propres films ? Donner le nom, le rôle et le titre des films ;

```
select prénom, nom, nomRôle
from   Rôle as r, Film as f1, Film as f2, Artiste as a
where  f1.idFilm = r.idFilm
and    f2.idRéalisateur = a.idArtiste
and    r.idActeur = a.idArtiste
```

Plan d'exécution sur la base films

Les temps sont exprimés en millisecondes et la mémoire allouée en octets.

Résultats :

Arbre des opérateurs

- Jointure de hachage (temps de réponse:91.21 ; temps d'exécution:143.07 ; nombre de nuplets:325 ; mémoire allouée:31)
- condition de jointure (r.idfilm = f1.idfilm)
- -> jointure de hachage (temps de réponse : 68.90 ; temps d'exécution :119.90 ; nombre de nuplets :325 ; mémoire allouée:35)
 - Condition de jointure (a.idartiste = f2."idréalisateur")
 - -> parcours séquentiel d'artiste a (temps de réponse :0.00 ; temps d'exécution :39.36 ; nombre de nuplets :2236 ; mémoire allouée :22)
 - -> hachage (temps de réponse :64.84 ; temps d'exécution :64.84 ; nombre de nuplets :325 ; mémoire allouée :25)
 - -> jointure de hachage (temps de réponse :22.31 ; temps d'exécution :64.84 ; nombre de nuplets :325 ; mémoire allouée :25)
 - Condition de jointure (r.idacteur = f2."idréalisateur")
 - -> parcours séquentiel de "rôle" r (temps de réponse :0.00 ; temps d'exécution :32.11 ; nombre de nuplets :1911 ; mémoire allouée :21)
 - -> hachage (temps de réponse :18.25 ; temps d'exécution :18.25 ; nombre de nuplets :325 ; mémoire allouée :4)
 - -> parcours séquentiel de film f2 (temps de réponse :0.00 ; temps d'exécution :18.25 ; nombre de nuplets :325 ; mémoire allouée :4)
 - -> hachage (temps de réponse :18.25 ; temps d'exécution :18.25 ; nombre de nuplets :325 ; mémoire allouée :4)
 - -> parcours séquentiel de film f1 (temps de réponse :0.00 ; temps d'exécution :18.25 ; nombre de nuplets :325 ; mémoire allouée :4)

Grande base des films

Plan d'exécution sur la base maxindex

Les temps sont exprimés en millisecondes et la mémoire allouée en octets.

Résultats :

Arbre des opérateurs

- Gather (temps de réponse:54504.61 ; temps d'exécution:117184.26 ; nombre de nuplets:325000 ; mémoire allouée:34)
- workers planned: 2
- -> parallel jointure de hachage (temps de réponse:53504.61 ; temps d'exécution:83684.26 ; nombre de nuplets:135417 ; mémoire allouée:34)
 - condition de jointure (a.idartiste = f2."idréalisateur")
 - -> parallel parcours séquentiel de artiste a (temps de réponse:0.00 ; temps d'exécution:26121.67 ; nombre de nuplets:931667 ; mémoire allouée:25)
 - -> parallel hachage (temps de réponse:51811.89 ; temps d'exécution:51811.89 ; nombre de nuplets:135417 ; mémoire allouée:21)
 - -> parallel jointure de hachage (temps de réponse:27461.83 ; temps d'exécution:51811.89 ; nombre de nuplets:135417 ; mémoire allouée:21)
 - condition de jointure (r.idfilm = f1.idfilm)
 - -> parallel jointure de hachage (temps de réponse:17480.88 ; temps d'exécution:41475.47 ; nombre de nuplets:135417 ; mémoire allouée:25)
 - condition de jointure (r.idacteur = f2."idréalisateur")
 - -> parallel parcours séquentiel de "rôle" r (temps de réponse:0.00 ; temps d'exécution:20426.71 ; nombre de nuplets:800971 ; mémoire allouée:21)
 - -> parallel hachage (temps de réponse:15788.17 ; temps d'exécution:15788.17 ; nombre de nuplets:135417 ; mémoire allouée:4)
 - _____ -> parallel parcours séquentiel de film f2 (temps de réponse:0.00 ; temps d'exécution:15788.17 ; nombre de nuplets:135417 ; mémoire allouée:4)
 - -> parallel hachage (temps de réponse:8288.24 ; temps d'exécution:8288.24 ; nombre de nuplets:135417 ; mémoire allouée:4)
 - -> parallel index only scan using film_pkey on film f1 (temps de réponse:0.42 ; temps d'exécution:8288.24 ; nombre de nuplets:135417 ; mémoire allouée:4)

Interprétation des résultats :

1. Comparaison des performances selon la taille de la base et l'utilisation d'index :

a) Petite Base de Données :

- Performances généralement bonnes, même sans optimisation
- Temps d'exécution rapides pour la plupart des requêtes simples
- Utilisation mémoire modérée

b) Grande Base de Données sans Index :

- Augmentation significative des temps d'exécution, particulièrement pour les requêtes complexes
- Consommation mémoire élevée pour les opérations de tri et de jointure
- Performances dégradées pour les recherches sur des colonnes non-clés

c) Grande Base de Données avec Index :

- Amélioration drastique des temps d'exécution pour les requêtes ciblées par les index

- Réduction notable de la consommation mémoire pour certaines opérations
- Gain de performance particulièrement marqué pour les requêtes impliquant des jointures et des recherches sur des colonnes fréquemment utilisées

2. Analyse détaillée d'une requête complexe (basée sur le plan d'exécution fourni) :

- Temps de réponse total : 54504.61 ms
- Temps d'exécution total : 117184.26 ms
- Nombre total de tuples traités : 325000
- Utilisation de techniques d'optimisation avancées :
 - Parallélisme avec 2 processus parallèles
 - Jointures de hachage à plusieurs niveaux
 - Utilisation d'un 'index only scan' sur la clé primaire de la table Film

3. Observations spécifiques sur la requête complexe :

- Efficacité des jointures de hachage pour gérer de grands volumes de données
- Impact positif du parallélisme sur le temps de traitement global
- Identification de goulots d'étranglement potentiels, notamment dans les parcours séquentiels de grandes tables"

2. Performances :

- Temps de réponse total : 54504.61 ms
- Temps d'exécution total : 117184.26 ms
- Nombre total de tuples traités : 325000

3. Techniques d'optimisation observées :

- Parallélisme : Utilisation de 2 processus parallèles planifiés
- Jointures de hachage : Employées à plusieurs niveaux pour optimiser les jointures
- Index : Utilisation d'un 'index only scan' sur la clé primaire de la table Film

4. Points d'attention :

- Parcours séquentiels sur les tables Artiste et Rôle, potentiellement coûteux
- Temps d'exécution élevé pour certaines étapes, notamment le parcours de la table Artiste

5. Opportunités d'optimisation :

- Évaluer la possibilité d'ajouter des index sur les colonnes fréquemment utilisées dans les jointures
- Analyser la pertinence du parallélisme et potentiellement ajuster le nombre de processus parallèles
- Considérer la restructuration de la requête pour réduire le nombre de jointures si possible

Conclusions :

1. L'analyse du plan d'exécution révèle une requête complexe bien optimisée par le SGBD, utilisant le parallélisme et des techniques de jointure avancées.
2. Malgré l'optimisation, certaines opérations restent coûteuses, suggérant des opportunités d'amélioration supplémentaires.
3. L'utilisation judicieuse des index et l'optimisation continue des requêtes sont importantes pour maintenir de bonnes performances sur des bases de données volumineuses.

Cette analyse démontre l'importance d'une compréhension approfondie des plans d'exécution pour identifier et implémenter des optimisations efficaces dans les systèmes de gestion de bases de données complexes."