

Index Global VizieR

François-Xavier Pineau

¹CDS, Observatoire Astronomique de Strasbourg

Réunion Index Global VizieR du 07 juillet 2011



L'idée

- Une idée PAS nouvelle : au moins 6 ans ?
- Indexer l'ensemble des positions des entrées VizieR
- Information stockée pour chaque entrée :
 - Identifiant de table
 - Identifiant de l'entrée dans la table
 - Coordonnées équatoriales J2000 de l'entrée
- But : accès rapide à l'ensemble des entrées VizieR pour une région du ciel donnée

Exemples d'utilisation

- Aladin / VizieR / Portail CDS : amélioration de l'expérience utilisateur :
 - Accès rapide à la liste des catalogues présents dans le FOV
 - Nombre exact de sources à charger par catalogue
- Cross-match multi-catalogues

Structure physique de l'index (sur le HDD)

Un répertoire racine contenant :

- Un fichier .csv d'informations sur les tables (id, name, ...)
- 48 s-répertoires (ordre 1, 29.3°)
- 3072 ss.-éléments, 64 par s.-répertoire (ordre 4, 3.66°)

Un élément (ordre n) peut-être :

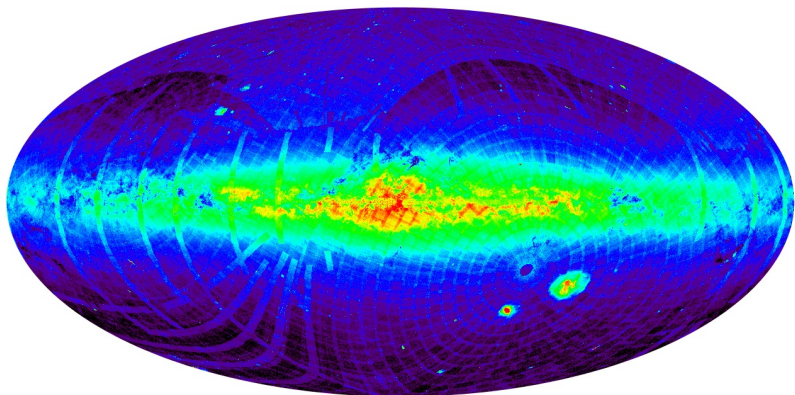
- Un fichier binaire contenant :
 - index Healpix : 4 kB, ordre $n + 5$ (\rightsquigarrow ordre minimum 9, 6.87')
 - entrées triées par indice Healpix (+ entrées vides)
 - taille max : 250 MB (va changer : 50, 75, 100 MB)
- Un dossier composé de 4 s.-éléments d'ordre $n + 1$

Pourquoi cette structure ?

- Nombre de fichier : possibilité de précharger leur index
- 48 répertoires d'ordre 1 : affichage terminal + commandes unix
- ordre $n + 5$: blocks de 20 kB à max 250 kB (si répartition uniforme)

Contenu de l'index

- 9 538 tables VizieR
- 6 193 294 968 entrées
- Taille : 122 GB
- Il manque essentiellement les gros catalogues qui ne sont pas dans le XMatch :
 - deprecated : SDSS DR4, DR5, DR6
 - GSC 1.x, 2.1, 2.2
 - ...
- Temps de création : $\approx 6h$



URL : <http://cdsxmatch.u-strasbg.fr/ViziServlet/vizidex.html>

Vizier Global Index Interface

Sky area

Cone search
Center:
Radius:
 Healpix cell (galactic)
Nside:
Index:

Print option

Catalogues counts Index Entries

VizieR Index entries/counts

Show entries Search:

Table ID	Entry ID	RA	Dec
I/252/out	239405978	14.293509	+57.999259
I/284/out	494031725	14.303242	+57.997439
I/284/out	494031759	14.304073	+58.000367
I/284/out	494031814	14.293400	+57.999325
I/284/out	494031815	14.293542	+57.998625
I/284/out	494031847	14.297770	+58.000984
I/297/out	541577472	14.303242	+57.997439
I/297/out	541577507	14.304072	+58.000367
I/297/out	541577563	14.293400	+57.999325
I/297/out	541577564	14.293542	+57.998625

Showing 1 to 10 of 23 entries

Mises à jour

- 33% d'espace libre ($tabID = -1$)
- Ajout de table :
 - calcul des indices Healpix (ordre 13), tri
 - parcours de l'arborescence, ajout dans les emplacements libres
 - si plus d'emplacements libres : réécriture du fichier / split (TBD)
- Suppression d'une table :
 - parcours exhaustif, $tabID \rightsquigarrow -1$
 - à l'avenir footprint ?

Problèmes liés à la mise à jour : disponibilité du service

Comment gérer les mises à jour en même temps que l'utilisation de l'index ?

Démon et gestion des accès concurrents

- Gestion centralisée par un démon (servlet)
 - préchargement des classes Java
 - préchargement de l'arborescence, pause de verrous de fichiers
 - préchargement des index des fichiers
- Gestion des accès concurrents (à tester !)
 - 2 verrous bloquants par fichier : 1 en écriture, 1 en lecture
 - Pas de limite au nombre de lecteurs, lectures simultanées
 - Une seule écriture à la fois, pas de lecture pendant une écriture
 - Demande de lecture et/ou écriture en attente pendant une écriture
 - Demande d'écriture prioritaire
 - Plus d'info : <http://download.oracle.com/javase/6/docs/api/java/util/concurrent/locks/ReadWriteLock.html>

Problèmes liés à la méthode

Une requête pourrait retourner les sources d'un catalogue dans une région du ciel alors que les entrées sont supprimées pour une autre.

Mais : incidence sur les grosses requêtes principalement (touchant plusieurs fichiers).