

巨大天体カタログと任意天体カタログとの ポジションマッチアップ

山内 千里

宇宙航空研究開発機構 宇宙科学研究所

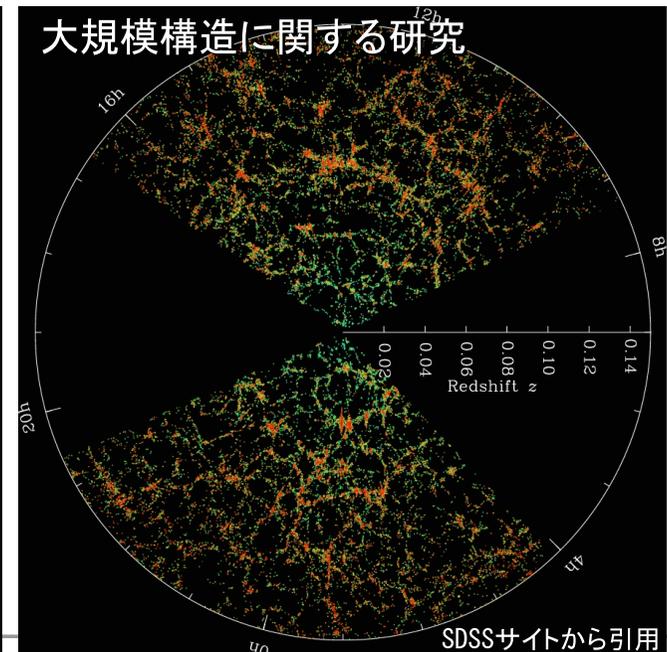
今日のお話し

- 天体カタログとは
- 天体カタログのデータ公開サービスとは
- 今回の開発に至るまでの経緯
- マッチアップの手法
- PostgreSQL-8.4 を使った実装の基本
- PostgreSQL-8.4 を使った大規模カタログ対応
- まとめ

天体カタログとは(1)

天体カタログの目的

- 新天体(超新星・彗星・小惑星等)の搜索
- 地上観測・衛星観測での位置較正・等級の較正
- データ解析による天文学研究における利用



天体カタログとは (2)

一般的な天体カタログ

- プライマリキー・座標 (経度, 緯度; 浮動小数点値)・等級などをカラムに持ち, 1行1天体
- カタログにより, 位置精度がかなり異なる
- 大規模カタログでは「天体名」のカラムは無い
2つ以上のカタログで同一天体の情報を得たい場合, 位置精度を考慮して座標値に近いものを, 自分で選ぶ必要がある
データ解析ではこの“マッチアップ”の需要が高い

objID	run	rerun	camcol	field	obj	type	ra	dec	u	g	r	i	z	Err_u	Err_g	Err_r	Err_i	Err_z
587726032792191572	1462	40	4	468	596	3	195.50238232	2.50383736	23.091534	22.536518	23.219761	22.217064	23.195219	0.722462	0.178235	0.471561	0.312275	1.253866
587726032792191560	1462	40	4	468	584	6	195.49777861	2.50480047	22.719301	22.687466	23.136376	23.397287	22.204723	0.443629	0.170543	0.365079	0.627285	0.908129
587726032792191161	1462	40	4	468	185	6	195.50006152	2.51001307	21.011089	19.536037	19.004444	18.808895	18.725103	0.097735	0.013818	0.012109	0.013682	0.047131
587726032792191253	1462	40	4	468	277	6	195.50338135	2.49517526	23.128822	20.874578	19.441227	18.213102	17.557289	0.563357	0.034428	0.016544	9.686179E-3	0.01959
587726032792191570	1462	40	4	468	594	3	195.50238936	2.49009438	22.046368	21.817022	21.469481	20.75507	20.43574	0.265911	0.084232	0.090651	0.072998	0.247657
587726032792191571	1462	40	4	468	595	3	195.50083343	2.49040539	24.028608	23.507906	22.505405	21.311026	21.146154	1.426794	0.428831	0.271606	0.148105	0.565282
587726032792191163	1462	40	4	468	187	6	195.50849796	2.50856538	19.913969	17.416637	16.164482	15.577018	15.262423	0.043291	4.951692E-3	4.107814E-3	4.403433E-3	5.968257E-3
587726032792191283	1462	40	4	468	307	3	195.49188468	2.50686136	22.927683	21.857718	20.031809	19.118488	18.679014	0.808918	0.121613	0.037246	0.02641	0.078184
587726032792191538	1462	40	4	468	562	6	195.48761378	2.49976186	24.988539	22.894716	22.585751	22.611437	21.340548	1.166349	0.196533	0.21838	0.32883	0.481173
587726032792191544	1462	40	4	468	568	3	195.491917	2.49209086	22.899073	22.263603	21.014359	20.509218	19.83617	0.671795	0.149571	0.075209	0.076365	0.191662
587726032792191545	1462	40	4	468	569	3	195.49348996	2.49174067	24.478172	23.049887	22.784391	22.094995	21.681513	1.235819	0.223935	0.25769	0.209802	0.937999
587726032792191553	1462	40	4	468	577	3	195.49516849	2.5094435	22.386639	21.979746	21.646896	21.417463	21.018116	0.411743	0.110988	0.124472	0.160534	0.506803
587726032792191564	1462	40	4	468	588	3	195.50169935	2.51222927	24.859159	21.349638	20.093365	19.532784	19.21265	1.42937	0.059345	0.029941	0.027789	0.092352
プライマリキー	観測情報			天体の位置			天体の等級 (5バンド)					等級のエラー						
587726032792191149	1462	40	4	468	173	3	195.48343364	2.50006848	23.808319	23.009523	22.297615	21.886116	22.822298	0.894942	0.204041	0.157389	0.160374	0.908237

Sloan Digital Sky Survey の例

天体カタログとは(2)

一般的な天体カタログ

- プライマリキー・座標(経度, 緯度; 浮動小数点値)・等級などをカラムに持ち, 1行1天体
- カタログにより, 位置精度がかなり異なる
- 大規模カタログでは「天体名」のカラムは無い
2つ以上のカタログで同一天体の情報を得たい場合, 位置精度を考慮して座標値が近いものを, 自分で選ぶ必要がある
データ解析ではこの“マッチアップ”の需要が高い

objID	run	rerun	camcol	field	obj	type	ra	dec	u	g	r	i	z	Err_u	Err_g	Err_r	Err_i	Err_z
587726032792191572	1462	40	4	468	596	3	195.50238232	2.50383736	23.091534	22.536518	23.219761	22.217064	23.195219	0.722462	0.178235	0.471561	0.312275	1.253866
587726032792191560	1462	40	4	468	584	6	195.49777861	2.50480047	22.719301	22.687466	23.138376	23.397287	22.204723	0.443629	0.170543	0.365079	0.627285	0.908129
587726032792191161	1462	40	4	468	185	6	195.50006152	2.51001331	21.011089	19.536037	19.004444	18.808895	18.725103	0.097735	0.013818	0.012109	0.013682	0.047131
587726032792191253	1462	40	4	468	277	6	195.50338135	2.49517526	23.128822	20.874578	19.441227	18.213102	17.557289	0.563357	0.034428	0.016544	9.686179E-3	0.01959
587726032792191570	1462	40	4	468	594	3	195.50238936	2.49009438	22.046368	21.817022	21.469481	20.75507	20.43574	0.265911	0.084232	0.090651	0.072998	0.247657
587726032792191571	1462	40	4	468	595	3	195.5008343	2.										
587726032792191163	1462	40	4	468	187	6	195.50849796	2.										
587726032792191281	1462	40	4	468	307	3	195.49188468	2.										
587726032792191538	1462	40	4	468	562	6	195.48761378	2.49976186	24.988539	22.894716	22.585751	22.611437	21.340548	1.166349	0.196533	0.21838	0.32883	0.481173
587726032792191544	1462	40	4	468	568	3	195.491917	2.49209086	22.899073	22.263603	21.014359	20.509218	19.83617	0.671795	0.149571	0.075209	0.076365	0.191662
587726032792191545	1462	40	4	468	569	3	195.491917	2.49209086	22.899073	22.263603	21.014359	20.509218	19.83617	0.671795	0.149571	0.075209	0.076365	0.191662
587726032792191553	1462	40	4	468	577	3	195.491917	2.49209086	22.899073	22.263603	21.014359	20.509218	19.83617	0.671795	0.149571	0.075209	0.076365	0.191662
587726032792191564	1462	40	4	468	584	6	195.49777861	2.50480047	22.719301	22.687466	23.138376	23.397287	22.204723	0.443629	0.170543	0.365079	0.627285	0.908129
587726032792191813	1462	40	4	468	837	3	195.50296088	2.51021388	22.815901	23.016315	21.205267	20.259922	19.969349	0.587908	0.270476	0.082926	0.055284	0.192164
587726032792191148	1462	40	4	468	172	6	195.48256013	2.5012255	24.076654	23.495977	21.281994	20.256006	19.905582	1.073476	0.319855	0.069453	0.041908	0.136506
587726032792191149	1462	40	4	468	173	3	195.48343364	2.50006648	23.808319	23.009523	22.297615	21.886116	22.822298	0.894942	0.204041	0.157389	0.160374	0.908237

天球上の天体の位置(経度・緯度): ra・dec

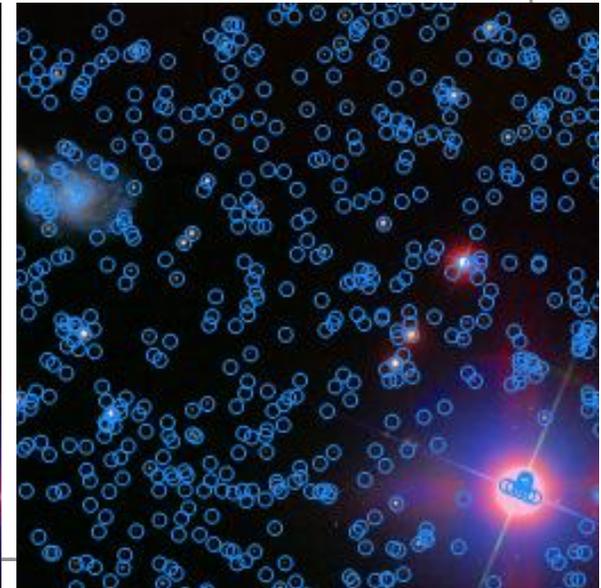
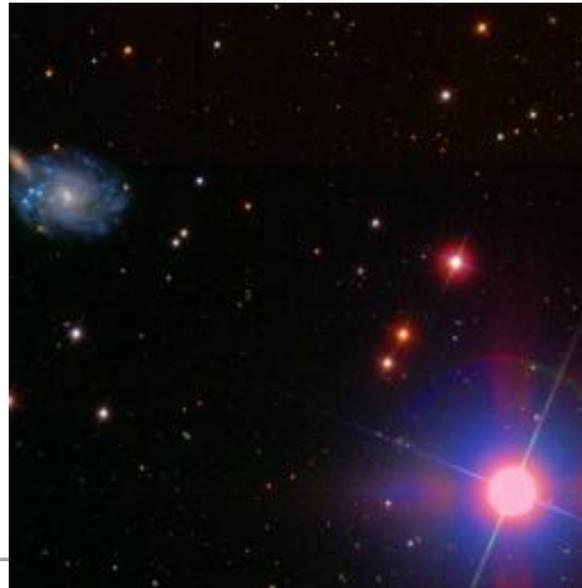
プライマリキー(天体ID): objid

Sloan Digital Sky Survey の例

天体カタログとは（3）

天体カタログはどのようにして作られる？

- 主役はサーベイ観測による天体カタログ
 - 観測プロジェクトが望遠鏡を長時間独占し，広い天域を掃きながら同一条件で写真を撮りまくる．
 - それらの写真を専用のプログラムで解析し，星・銀河などを自動検出してカタログ化する．



天体カタログとは (4)

天体カタログのデータ規模

- USNO-B1.0 (アメリカ海軍天文台; 約 10 億天体)
過去の観測による多数の乾板をスキャンして作られた
- **2MASS 点源カタログ** (米国の複数の研究機関; 約 4.7 億天体)
地上望遠鏡による近赤外天体カタログ
- 「あかり」中間赤外全天カタログ (JAXA; 約 87 万天体)
衛星による国産初の全天サーベイ天体カタログ
- 小規模なカタログは無数にある

天体カタログ用のデータ公開サービス

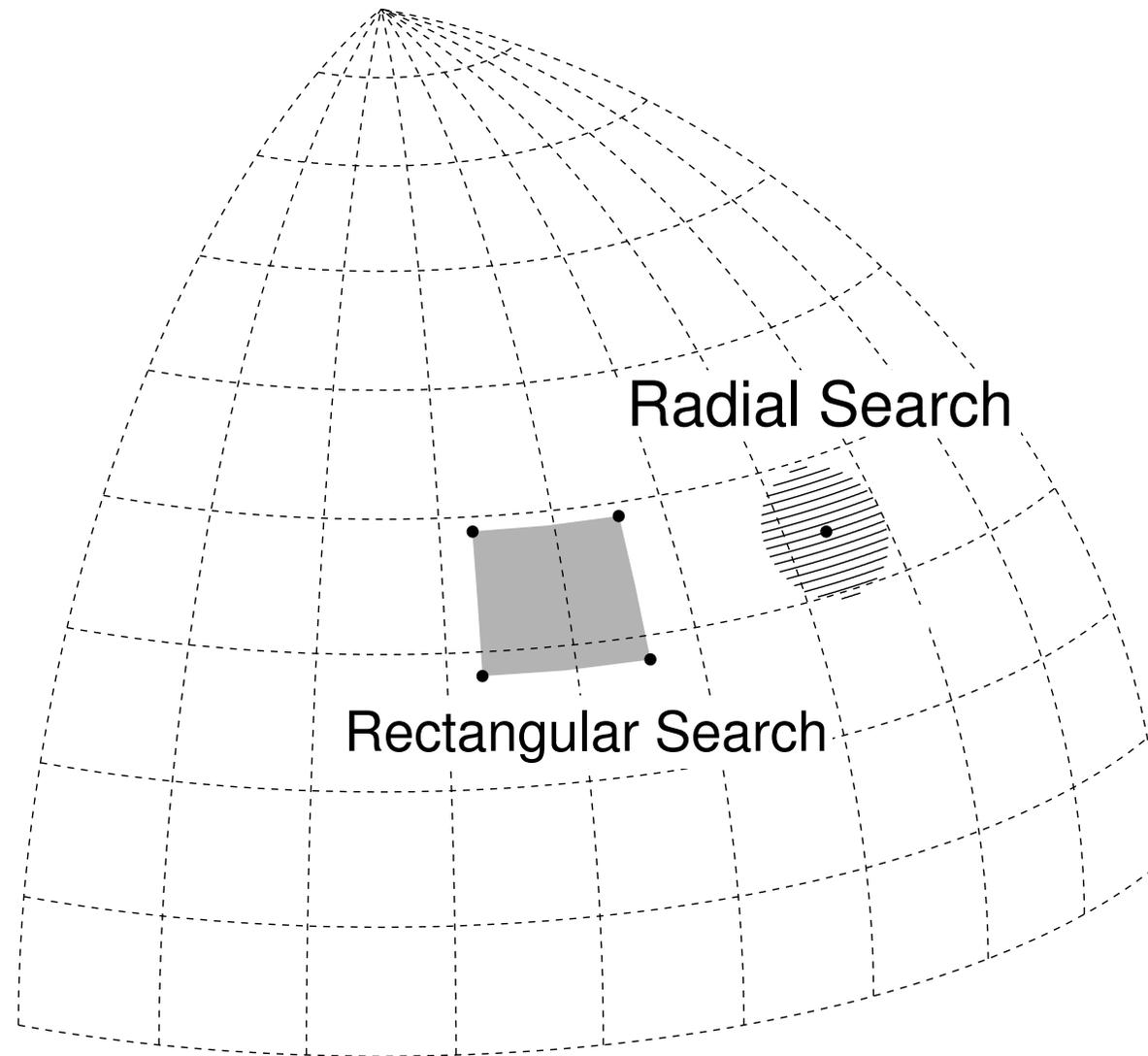
- 天文学者が容易に活用できるように、特に大規模カタログでは専用の WebDB が作られる事が多い

天体カタログのデータ公開サービスとは(1)

RDBMS に天体カタログを登録し，WebDB を開発して公開するのが一般的．次のようなサービスが求められる．

- **ポジション(経度・緯度)による高速検索**
 - Radial Search
1つの経度・緯度から 半径 r 以内 の天体を検索
 - Rectangular Search
経度・緯度の範囲 (矩形領域) で天体を検索
 - Cross-ID(マッチアップ・クロスマッチ)
任意の天体カタログやユーザが用意したカタログのポジションすべてについて Radial Search を行ない，その結果を返す
- 様々な条件による検索・結合・統計情報の取得
- 画像や，他サイトへの動的リンクの提供

天体カタログのデータ公開サービスとは(1)



天体カタログのデータ公開サービスとは(2)

Radial Search のページ遷移の例 — 検索条件入力ページ

The screenshot shows the 'Radial Search' page of a database service. A sidebar on the left contains a navigation menu with items like 'Instruments', 'Pointing Data Archives', 'Observational Database', 'Search Tool', 'Table Information', 'Data Access', 'FTP', 'HTTP', 'Catalog Archives (CAS)', 'Update Information', 'Search Tools', 'Radial Search', 'Rectangular Search', 'Object Cross-ID', 'Match-up with SIMBAD/NED', 'SQL Search', 'Download Local Tools', 'Visual Tools', 'Explore', 'Image List', 'Database Documents', 'SQL Schema', 'Tables', 'Views', 'Functions', 'Help', and 'SQL Tutorial'. The 'Radial Search' item is highlighted in red.

The main content area is titled 'Radial Search' and includes the following text: 'This page provides an easy interface to search FIS or IRC objects within a radius (arcmins) of a point. The point can be set by J2000, B1950, Ecliptic or Galactic coordinates.' Below this is a form with three sections: 'Coordinate', 'Object Name', and 'Radius'. The 'Coordinate' section has a dropdown menu set to 'J2000' and two input fields for 'longitude*' (210.804) and 'latitude*' (54.3506). The 'Object Name' section has a text input field and two radio buttons for 'by SIMBAD:' and 'by NED:'. The 'Radius' section has an input field for '10' and a dropdown menu for 'arcmins'. Below the form is a note: '* [±]12:34.56', [±]12 34 56.78' and [±]12 34.56' styles are also allowed.'

Below the form is a table titled 'Catalogs and Constraint Settings:'. The table has columns for 'Catalogs', 'Enable?', 'Min', 'Quality Indicator', and 'Max'. The 'FIS' catalog is checked, and the 'IRC' catalog is unchecked. The 'Quality Indicator' column lists 'fQual_65', 'fQual_90', 'fQual_140', 'fQual_160', 'fQual_09', and 'fQual_18'. The 'Min' and 'Max' columns have input fields with the value '3'.

At the bottom of the page, there are several controls: 'Database: DR1 (Fis:V1 Irc:V1)', 'Columns: all digest', 'Rows: all max 10', 'Format: HTML TEXT delimiter: newline: LF', and 'reset submit' buttons.

Callouts in the image point to the 'Coordinate' dropdown (座標系), the 'longitude*' and 'latitude*' input fields (経度・緯度), and the 'Radius' input field (サーチ半径).

天体カタログのデータ公開サービスとは (3)

Radial Search のページ遷移の例 — 結果ページ



Your SQL command was:

検索に使ったSQLステートメントの表示

```
SELECT o.*, n.distance
FROM fGetNearbyObjCel('Fis', 'j2000', fLonStr2Deg('210.804'), fLatStr2Deg('54.3506'), 10)
WHERE n.objID = o.objID
ORDER BY n.distance
LIMIT 10
```

Click number of objid column to open the Explore Tool

objid	objname	経度	緯度	天体の明るさ					
objid	objname	ra	dec	flux_65	flux_90	flux_140	flux_160	fqual_65	fqual_90
3364337	1403130+542102	210.804371246985	54.3506309910272	2.8365	4.04652	7.47104	5.01612	1	3
3069767	1403285+542121	210.868669009971	54.3558418521147	2.13147	3.8933	7.90386	8.29088	1	3
3247076	1402598+542329	210.749134742117	54.3912917527775	0.230672	0.957795	2.48693	1.55927	1	3
3212824	1403257+541830	210.856964677903	54.3082862533846	0.483016	1.05094	0.450301	null	1	3
3226372	1403143+541750	210.809476623473	54.297316742893	0.551029	1.44699	4.86652	4.73227	1	3
3177663	1403274+542410	210.864224989732	54.4028831375719	0.961737	2.22215	4.44592	3.41695	1	3
3204272	1402541+541809	210.725407609363	54.3024232414104	0.429494	0.597733	null	0.974106	1	2
3242217	1402583+541714	210.743058962091	54.2872755183917	0.427679	0.689045	0.981898	1.23627	1	3
3246837	1403410+541900	210.920911221898	54.3166065085316	7.59306	10.8848	10.8923	10.7543	3	3
3203296	1402364+542203	210.651506063977	54.3675959346663	0.659961	0.79246	3.13666	null	1	3



天体カタログのデータ公開サービスとは(4)

Radial Search のページ遷移の例 — 画像確認ページ

DARTS/Akari at ISAS/JAXA - Mozilla

http://darts.jaxa.jp/astro/akari/cas/tools/explore/obj.php?objid=3364337&scale=4&grid=true

AKARI-FIS-V1 J1403130+542102 Shrink Enlarge **AKARI-CAS Explore Tool**

Database=DR1, Catalog=Fis, objID=3364337 Get all data

ra	210.80437125 [14:03:13.05]	dec	54.35063099 [+54:21:02.3]
ra1950	210.36201910 [14:01:26.88]	dec1950	54.59023065 [+54:35:24.8]
lambda	174.36492920 [11:37:27.58]	beta	59.79933167 [+59:47:57.6]
l	102.03713804 [06:48:08.91]	b	59.76921867 [+59:46:09.2]

FIS Image Under construction IRC Image Under construction SDSS i,r,g (SDSS CAS)

IRIS 100μ,60μ,25μ (SkyView) 2MASS K,H,J (SkyView) DSS2 IR,R,B (SkyView)

Zoom: Image scale: 4 "/pix Draw grid Invert image update

flux_65	flux_90	flux_140	flux_160
2.8365	4.04652	7.47104	5.01612
fErr_65	fErr_90	fErr_140	fErr_160
1.85784	0.16795	0.912732	1.14301
fQual_65	fQual_90	fQual_140	fQual_160

Input form:

Database: Catalog: Search radius: arcmins

Coordinate system: Longitude [deg or 'hh:mm:ss']:

Latitude [deg or '[e]dd:mm:ss.s']:

Name to identify by SIMBAD (e.g., M101, NGC4300, etc.):

Name to identify by NED (e.g., M101, NGC4300, etc.):

ObjID:

submit

You can select version of FIS and IRC catalogs by Database selector.

SDSS i,r,g image is provided by SDSS SkyServer of the SDSS project.

SkyView has been developed by HEASARC at the NASA / GSFC Astrophysics Science Division.

天体カタログのデータ公開サービスとは（5）

開発者視点での、天体カタログ WebDB の特殊性

- 各テーブルは Read Only である
最高の検索速度を狙う事ができる
- RDBMS のテーブルには、読み取られて困る情報がない
 - “SELECT” から始まるユーザ SQL ステートメントの受付
 - テーブル・ビュー・ストアド関数の仕様公開
 - 想定外のアクセスが行なわれないように、セキュリティ設定が必要

天体カタログのデータ公開サービスとは(6)

SQL ステートメント入力ページ例

» Instruments
» Pointing Data Archives
- Observational Database
- Search Tool
- Table Information
- Data Access
- FTP
- HTTP
» Catalog Archives (CAS)
- Update Information
- Search Tools
- Radial Search
- Rectangular Search
- Object Cross-ID
- Match-up with SIMBAD/NED
- **SQL Search**
- Download Local Tools
- Visual Tools
- Explore
- Image List
- Database Documents
- SQL Schema
- Tables
- Views
- Functions
- Help
- SQL Tutorial
- SQL General Reference
- FAQ [1/5/17](#)
- Query Limits
- Web API
» Data Archives (DAS)
» Links
» JUDO

SQL Search

You can directly submit a SQL (Structured Query Language) query to the 'AKARI CAS' of DARTS database server.

We have prepared a lot of example calls to learn the 'SELECT' statement of SQL. Let's click the buttons in table, then an explanation and an example SQL statement will be set in the command field. You will easily understand the basic of SQL statement.

If you think that using SQL calls is difficult, see also our [Tutorial page](#) or [SDSS SQL Tutorial page](#).

Example SQL calls (Label on buttons indicate 'thema' of calls):

	Typical Calls	Radial Search	Rectangular Search
FIS	<input type="button" value="=<>"/> <input type="button" value="Digest"/> <input type="button" value="All"/> <input type="button" value="AND-OR"/> <input type="button" value="Bit-Flags"/> <input type="button" value="Sort"/> <input type="button" value="Random"/>	<input type="button" value="J2000"/> <input type="button" value="Ecliptic"/> <input type="button" value="Galactic"/> <input type="button" value="Input 'hh:mm:ss'"/> <input type="button" value="IRC CrossID"/> <input type="button" value="IRC CrossID 2"/> <input type="button" value="Advanced"/>	<input type="button" value="J2000"/> <input type="button" value="Ecliptic"/> <input type="button" value="Galactic"/> <input type="button" value="Input 'hh:mm:ss'"/> <input type="button" value="Random"/> <input type="button" value="IRC CrossID"/> <input type="button" value="IRC CrossID 2"/> <input type="button" value="Advanced"/>
IRC	<input type="button" value="=<>"/> <input type="button" value="Digest"/> <input type="button" value="All"/> <input type="button" value="AND-OR"/> <input type="button" value="Bit-Flags"/> <input type="button" value="Sort"/> <input type="button" value="Random"/>	<input type="button" value="J2000"/> <input type="button" value="Ecliptic"/> <input type="button" value="Galactic"/> <input type="button" value="Input 'hh:mm:ss'"/> <input type="button" value="FIS CrossID"/> <input type="button" value="FIS CrossID 2"/> <input type="button" value="Advanced"/>	<input type="button" value="J2000"/> <input type="button" value="Ecliptic"/> <input type="button" value="Galactic"/> <input type="button" value="Input 'hh:mm:ss'"/> <input type="button" value="Random"/> <input type="button" value="FIS CrossID"/> <input type="button" value="FIS CrossID 2"/> <input type="button" value="Advanced"/>
CrossID		<input type="button" value="RC3 with FIS+IRC"/> <input type="button" value="IRAS with FIS+IRC"/> <input type="button" value="FIS with IRAS PSC+FSC"/>	

Press buttons to import examples ;-)

```
-- Basic of SELECT statement of SQL:  
-- - SELECT col1 [as c1], col2 [as c2], ...  
--   Set required columns after SELECT keyword.  
--   You can define aliases after 'as' keyword.  
-- - FROM table1 [t1], table2 [t2], ...  
--   Set table or view names after FROM keyword.  
--   t1 and t2 are aliases of each table.  
-- - WHERE conditions...  
--   Set conditions after WHERE keyword.  
--  
-- This is an example query to obtain the first 10  
-- rows of FIS catalog using a flux cut.  
SELECT objID, objName, ra, dec,  
       fDeg2LonStr(ra) as ra_hms,  
       fDeg2LatStr(dec) as dec_dms,  
       fQual_65, fQual_90, fQual_140, fQual_160,  
       flux_65, flux_90, flux_140, flux_160  
FROM FisObjALL  
WHERE  
       19.5 < flux_90 AND flux_90 <= 20.0  
LIMIT 10
```

Database: Check Syntax Only

Format: HTML TEXT delimiter:

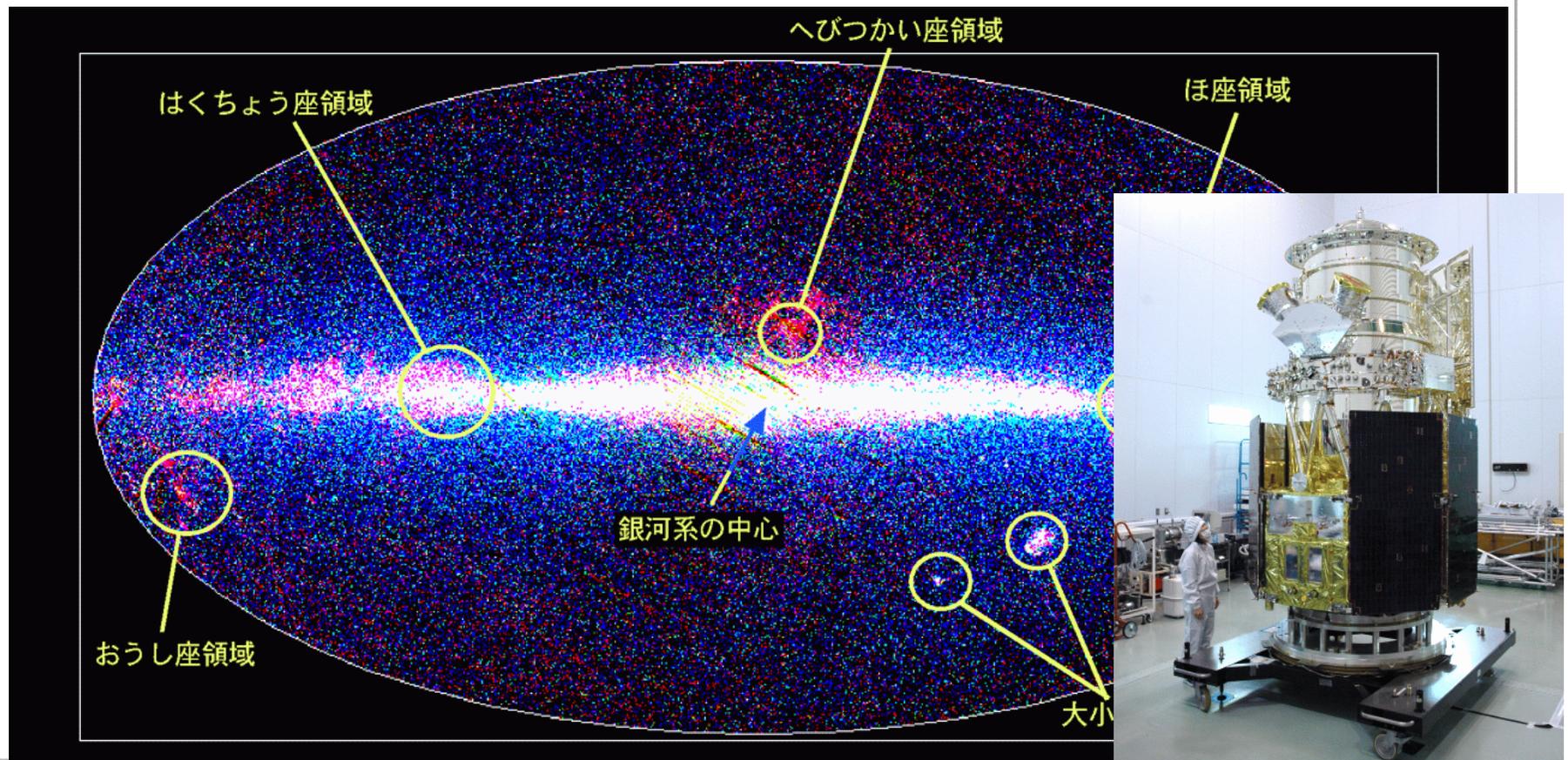
天体カタログのデータ公開サービスとは(7)

開発者視点での、天体カタログ WebDB の特殊性

- 素人でも SQL ステートメントが書ける配慮が必要
 - 文字列・数値の変換や特別なアルゴリズムによる検索は、あらかじめストアド関数に実装
 - RDBMS 製品は、ストアド関数の機能が強力である事
 - 特に、複数の行を返すストアド関数が必須
 - OSS では PostgreSQL がベスト!
 - 検索結果と同時に、SQL ステートメントを表示
 - ドキュメントの充実
- WebAPI も公開
 - wget 等でのアクセスも想定
 - コマンドラインでのインタフェースも開発し、公開

今回の開発に至るまでの経緯 (1)

- JAXA の「あかり」衛星全天サーベイによる日本初の2つの赤外線天体カタログ (合計約 130 万天体) が作成され, 2010 年 3 月末に一般に公開



今回の開発に至るまでの経緯（2）

- 「あかり」カタログ専用の WebDB の開発を担当する事になり，上記公開に合わせて開発・リリース
- 「あかり」カタログと 2MASS カタログ (約 4.7 億天体を含む近赤外カタログ) とのマッチアップや，2MASS カタログの高速検索の要望が複数の天文学者から寄せられる

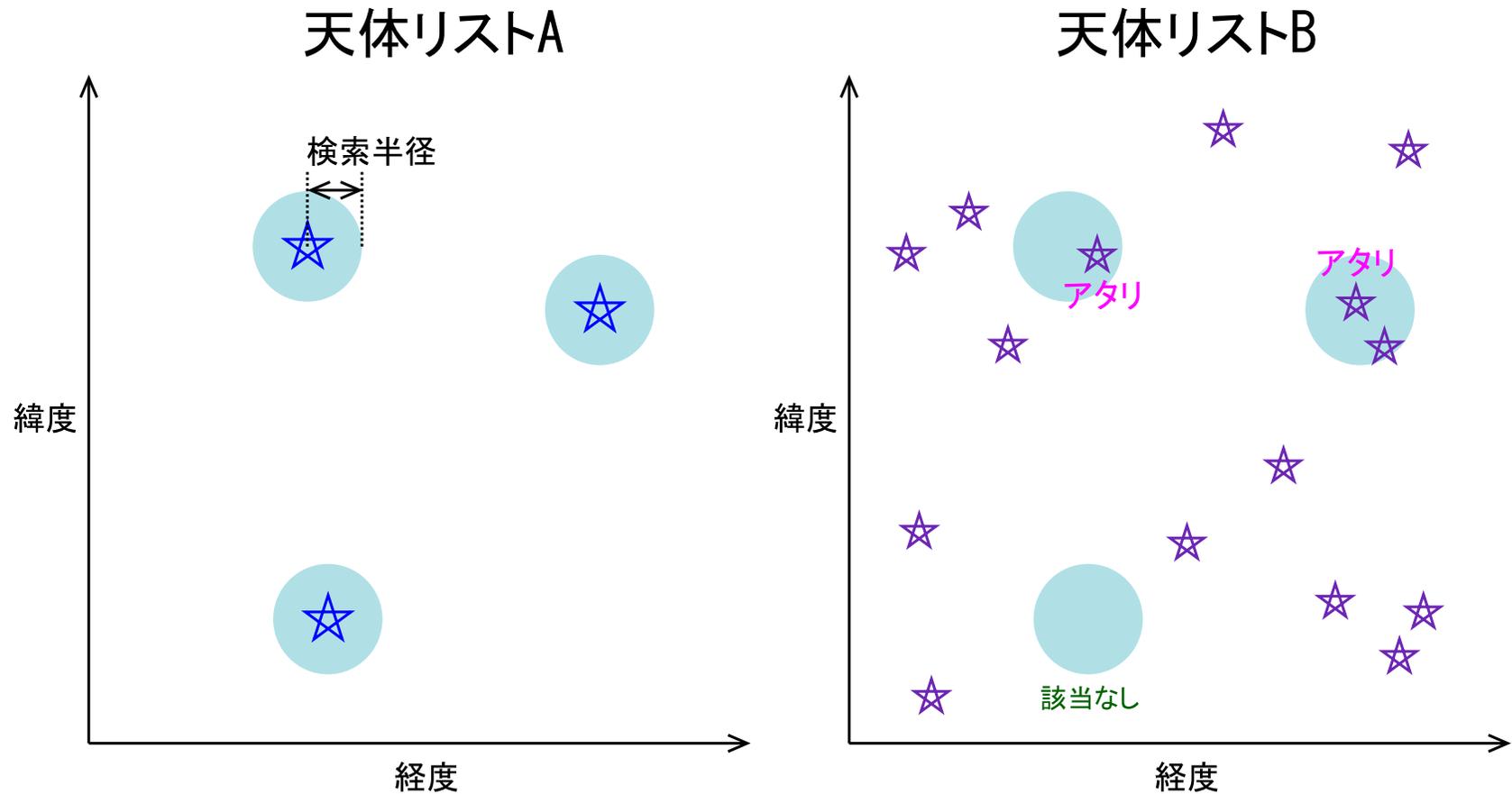
The screenshot displays the DARTS (Data Archives and Transmission System) web interface. The main content area is titled "AKARI (ASTRO-F) Catalogue Archive Server". It provides information about the AKARI FIS and IRC all-sky survey and offers search tools. A section titled "SQL Search" includes an example query to retrieve data from the FIS catalog. Below this, there are several astronomical images and data tables. One table shows coordinates for AKARI-IRC-V1 J1403124+542056, and another table lists various data points for this object, such as flux, magnitude, and scan parameters.

Database-DR1	Catalog-irc	objID	200283152		
ra	210.60194267	[140312.47]	dec	54.34911933	[+54:20:56.8]
ra1950	210.35955518	[1401:26:30]	dec1950	54.58872503	[+54:35:19.4]
lambda	174.96490496	[11:37:27.59]	beta	59.79726080	[+59:47:50.1]
l	102.03786080	[06:48:09.09]	b	59.77125735	[+59:46:16.5]

flux_09	0.140449	flux_18	
fErr_09	0.0121946	fErr_18 <td></td>	
fQual_09	3	fQual_18	0
flags_09	0b00001000	flags_18	0
nScanC_09	3	nScanC_18	0
nScanP_09	9	nScanP_18	5
mCent_09	1	mCent_18	0
nDens_09	0	nDens_18	0
extended_09	0	extended_18	0
meanAB_09	2.28483	meanAB_18	
nData_09	3	nData_18	0

マッチアップの手法 (1)

復習 ... マッチアップとは?



マッチアップの手法 (2)

基本的な考え方

- 2つの天体リストのそれぞれの座標値について、2つの座標間の角度 r が一定値以下の場合に、同一天体の候補とし、最も小さい r のものを「アタリ」とする。
- 天体リスト A, 天体リスト B, それぞれの天球上での位置を直交座標系 $(x_{A_n}, y_{A_n}, z_{A_n})$, $(x_{B_m}, y_{B_m}, z_{B_m})$ で表わすと、2つの天体間の角度 r は、次の式で求まる。

$$r = \arccos \left(\frac{2 - \{(x_{B_m} - x_{A_n})^2 + (y_{B_m} - y_{A_n})^2 + (z_{B_m} - z_{A_n})^2\}}{2} \right)$$

マッチアップの手法 (3)

マッチアップの手法いろいろ

- 総当たりで r を計算
巨大カタログでは現実的でない。
- 天球の分割 + 分割した領域ごとに総当たり
分割した領域の「端」付近の天体の扱いが面倒
- Plane Sweep
緯度方向に、リスト A、リスト B を徐々に同時に掃きながら r の計算を最小限にする方法。現在、ベストと言われる方法だが、RDBMS 向きではない。
- Multiple Radial Search (今日のお話はコレ)
RDBMS 向け。「Radial Search」の検索をチューニングするだけなので、システム全体をコンパクトに実装でき、SQL により柔軟な応用が可能 ユーザにとって重要

PostgreSQL-8.4 を使った実装の基本 (1)

Multiple Radial Search によるマッチアップのステートメント

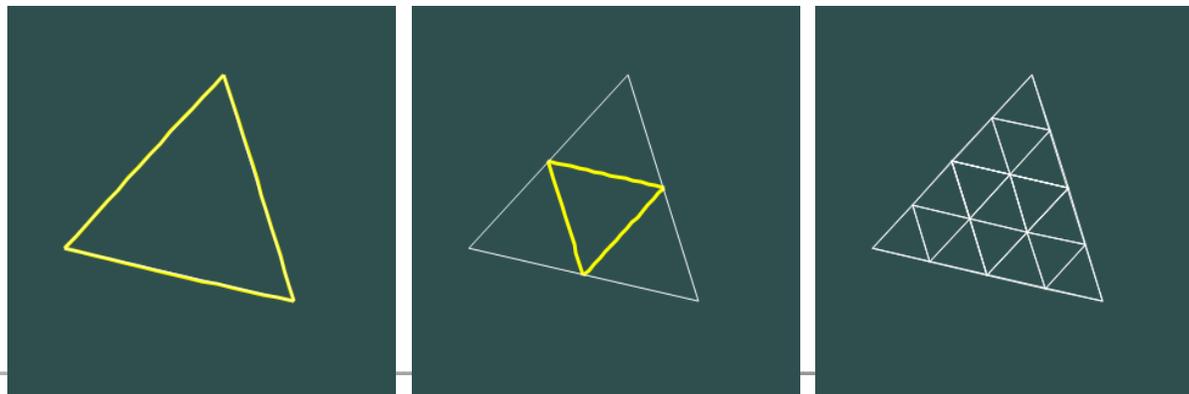
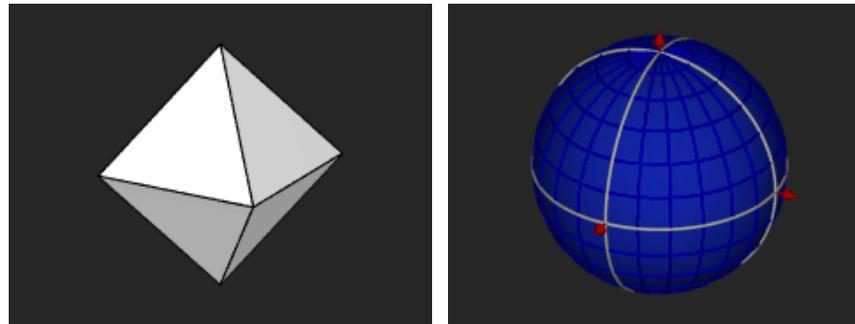
```
SELECT A.*, B.*      -- 天体リスト A と天体リスト B とを結合
FROM
(
  SELECT *
  FROM objlist_A      -- 天体リスト A を
  ORDER BY dec        -- 緯度の順でソート
) A
JOIN
  objlist_B B
ON      -- 検索半径 0.25 で最も近いものを自作関数で検索
  FBGetNearestObjIDEq(A.ra, A.dec, 0.25) = B.objid;
```

Radial Search を行なう `FBGetNearestObjIDEq()` を最適化せよ!

PostgreSQL-8.4 を使った実装の基本 (2)

高速 Radial Search のアルゴリズム

- 天球分割による方法 — 天文業界での標準
 - 何らかの手法で天球を非常に細かく分割
 - 分割したそれぞれの領域に unique な「天域 ID」を付与



SDSS SkyServer
から引用

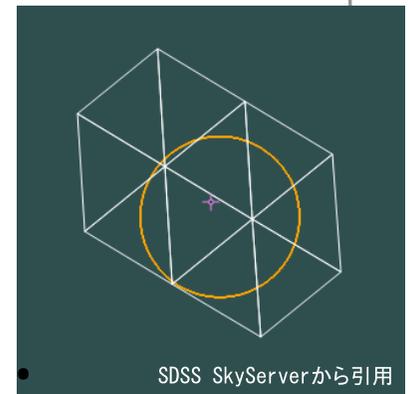
PostgreSQL-8.4 を使った実装の基本 (3)

高速 Radial Search のアルゴリズム

- 天球分割による方法(つづき)
 - 天体リストのテーブルに「天域 ID」のためのカラムを追加し, 値の登録を行ない, インデックスを作成

objid	run	rerun	camcol	field	obj	type	ra	dec	u	g	r	i	z	htmid
587728950388917007	2141	40	6	37	783	8	148.17756873	0.88436601	20.83951	26.711596	21.887098	27.524469	19.659977	15557397048548
587729149846946411	2188	40	2	197	619	3	151.61429452	-2.66710795	22.597839	23.288174	21.801199	21.318817	20.390635	10563805553406
587729149846946412	2188	40	2	197	620	3	151.61831949	-2.52546925	24.111555	22.772299	21.672077	21.842289	21.848372	10563788450971
587729149846946413	2188	40	2	197	621	3	151.61564824	-2.64874101	23.032095	24.46582	22.253138	21.883474	21.012527	10563791891207

- 検索する時は, 必要とする領域を取りこぼさないように「天域 ID」を計算して求め, 「天域 ID」でインデックス・スキャンした後, r を求めて厳密な条件を満たすものだけ取り出す.
- 「天域 ID」の計算にやや時間がかかるため, 小さい検索領域での多数の検索は苦手である.



PostgreSQL-8.4 を使った実装の基本 (4)

高速 Radial Search のアルゴリズム

- 直交座標系 (xyz) を利用する方法 — 今回の方法
 - 天体リストのテーブルに, 直交座標系での位置のためのカラムを追加し, 値を登録

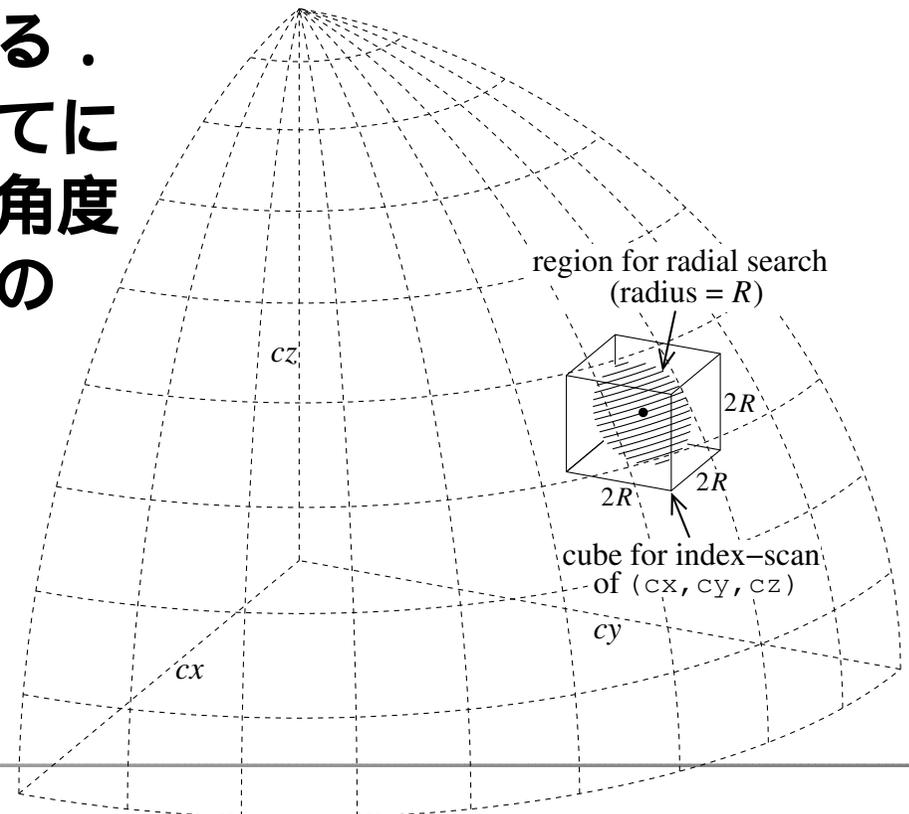
objid	objname	ra	dec	cx	cy	cz	ra1950
3348597	1743519-275919	265.966445271449	-27.988681956023	-0.062113652292042	-0.880853046662423	-0.46929713869197	265.1782205058
3262992	1744043-280156	266.018014728745	-28.0321225797054	-0.0612960831920363	-0.880553388877471	-0.469966508936239	265.2295089636
3340415	1744114-275814	266.047546657882	-27.9705722369879	-0.0608769764630452	-0.881088021139706	-0.469018008972827	265.2594171102
3310580	1743495-280430	265.956279647876	-28.0751276241634	-0.0622199340233321	-0.880134715297639	-0.470628901299184	265.1675196442
3295406	1743439-275604	265.932892129237	-27.9345380942852	-0.0626609062426132	-0.881258491361244	-0.468462466193993	265.1450107534
3047782	1744229-275746	266.095577548034	-27.962781723904	-0.060142683414582	-0.881202360810143	-0.468897917391754	265.3074863007
3329273	1743382-280250	265.909276332352	-28.0473329448933	-0.0629582216535493	-0.880310997182525	-0.470200819401382	265.1206994117
3042519	1743364-275754	265.901769757421	-27.964909149044	-0.063121830409747	-0.880976515679813	-0.468930712735239	265.1137066090
3046313	1744053-275356	266.021988745505	-27.8988550132819	-0.0613106762921765	-0.881645743979488	-0.467912153182039	265.2343097030
3298038	1743320-275944	265.883201996399	-27.9955794801456	-0.0633892942662041	-0.880705517301218	-0.469403439663181	265.0949524225

- xyz の組み合わせで, 複合インデックスを作成
`CREATE INDEX objlist_xyz ON objlist (cx,cy,cz);`

PostgreSQL-8.4 を使った実装の基本 (5)

高速 Radial Search のアルゴリズム

- 直交座標系 (xyz) を利用する方法 (つづき)
 - サーチ半径 R の時, サーチポジションを中心とする一辺 $2R$ の立方体の中にある天体を, インデックス・スキャンで取り出し「候補」とする.
 - 上記「候補」の天体すべてについて, 検索位置からの角度を求め, 条件を満たすもののみ取り出す.
 - インデックス・スキャンまでの計算がほとんど無く, 小さいサーチ半径では効率的



PostgreSQL-8.4 を使った実装の基本 (6)

Radial Search の実際の SQL テートメント

```
SELECT o.*
FROM (
  SELECT objid,cx,cy,cz,
         fDistanceArcMinXYZ($1,$2,$3,cx,cy,cz) AS distance
  FROM IrcObjAll
  WHERE (cx BETWEEN $1 - fArcMin2Rad($4) AND $1 + fArcMin2Rad($4))
        AND
        (cy BETWEEN $2 - fArcMin2Rad($4) AND $2 + fArcMin2Rad($4))
        AND
        (cz BETWEEN $3 - fArcMin2Rad($4) AND $3 + fArcMin2Rad($4))
) o
WHERE o.distance <= $4
```

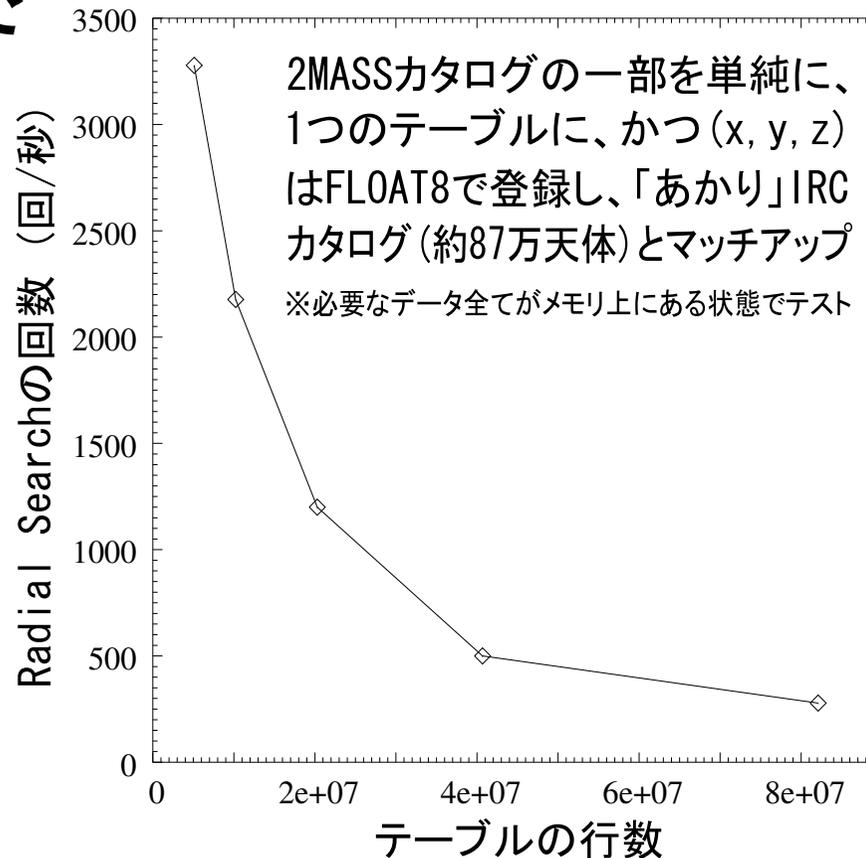
\$1 , \$2 , \$3 ... 検索位置 (直交座標系)

\$4 ... 検索半径

PostgreSQL-8.4 を使った大規模カタログ対応 (1)

「浮動小数点値の (x, y, z) に複合インデックス」の限界

- 1つのテーブルで単純に実装できるのは、せいぜい1000万天体まで



計測に用いたマシン: DP Opteron 2384(2.7GHz) 32GB メモリ

PostgreSQL-8.4 を使った大規模カタログ対応 (2)

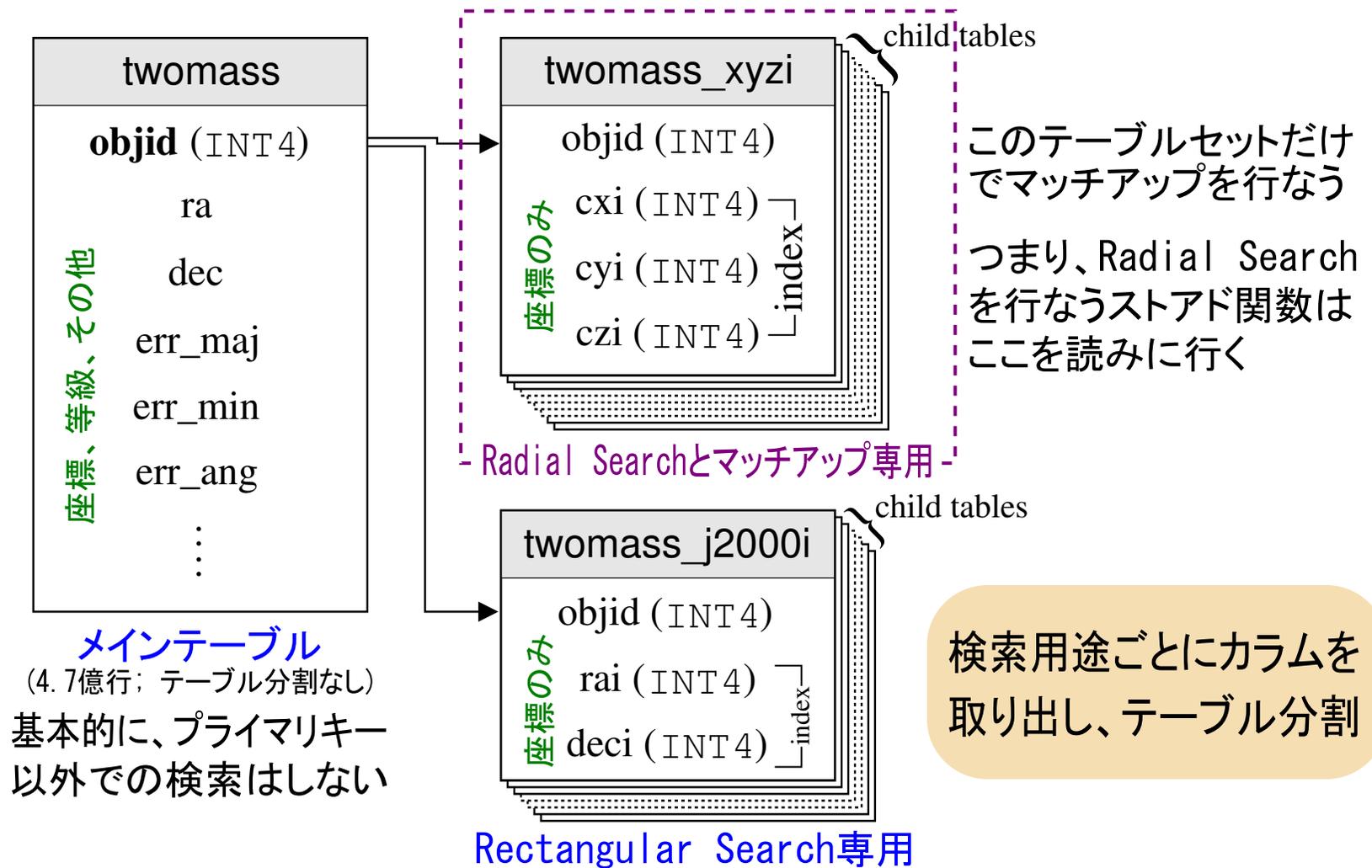
2MASS カタログ (4.7 億天体) のための戦略

- 天体の直交座標系での位置 (x, y, z) 専用のテーブルを作成
- テーブルパーティショニング (緯度で分割)
- CE を使わず, 自作ストアド関数 (PL/pgSQL+C) で子テーブルを選択 (子テーブルが 100 個を越えても高速に動作)
- 天体の直交座標系での位置 (x, y, z) を, 4 バイト整数でスケールリングして格納 約 20GB
- (x, y, z) を 2 バイトに丸めた値に対して, 複合インデックスを作成 (複合式インデックス) 約 10GB
- SSD を活用

【インデックスの深さを抑え, ディスク I/O を最小化】

PostgreSQL-8.4 を使った大規模カタログ対応 (3)

2MASS カタログ検索システムのためのテーブル設計



PostgreSQL-8.4 を使った大規模カタログ対応 (4)

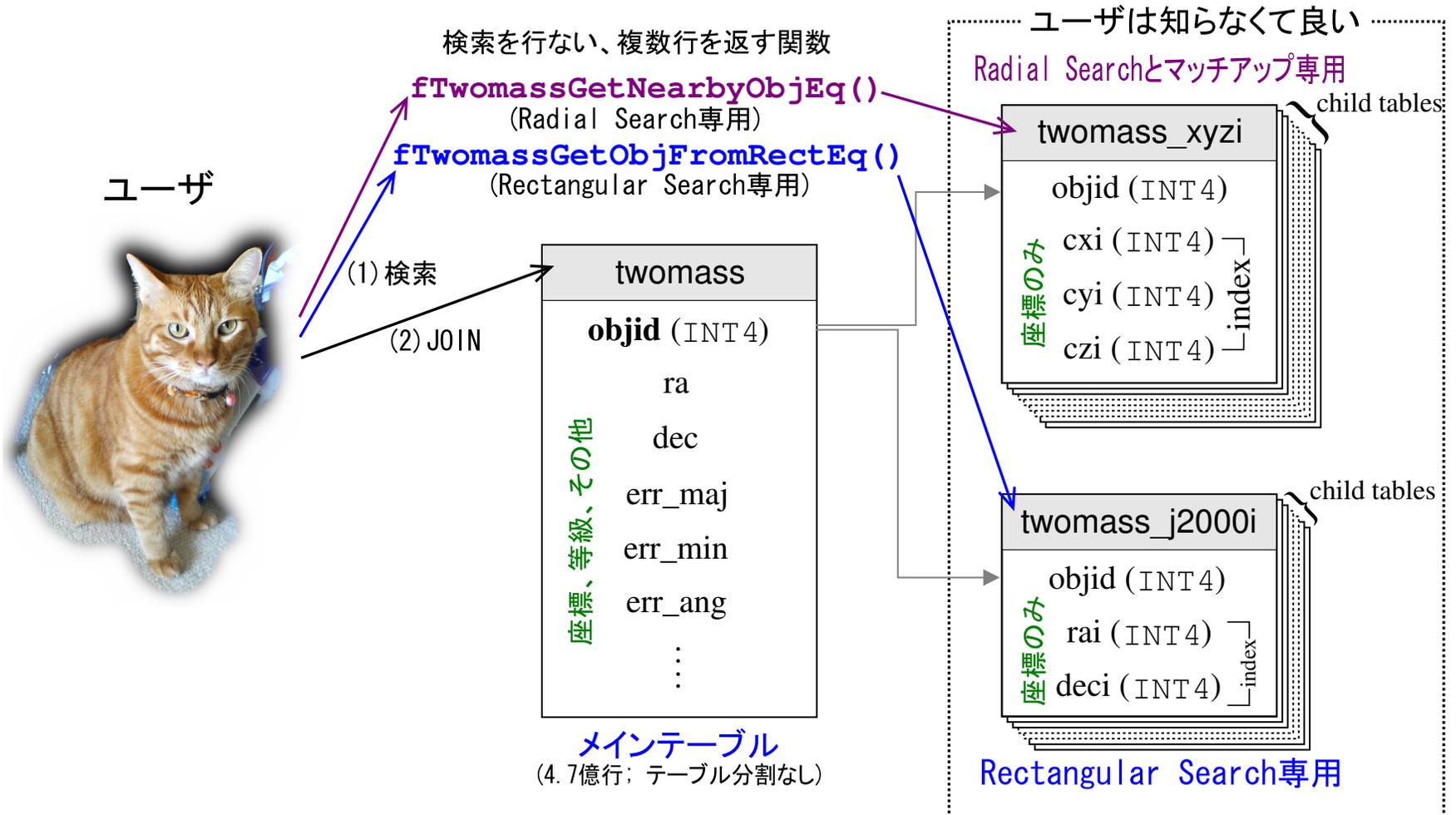
巨大テーブルパーティショニングのメリット・デメリット

- **パーティショニングをする場合**
何らかの実装により, 特定のカラムについてのみ, 検索の高速化が可能.
 - × プライマリキーでの JOIN は, パーティショニングをしない場合よりも遅くなった.
- **パーティショニングをしない場合**
プライマリキーでの検索・JOIN は高速.
 - × unique ではないカラムの検索は苦手.

パーティショニングをする場合・しない場合それぞれのメリットをとる事が重要らしい.

PostgreSQL-8.4 を使った大規模カタログ対応 (5)

ユーザはどのように使うのか?



PostgreSQL-8.4 を使った大規模カタログ対応 (6)

2MASS カタログ検索システムでの検索例

- マッチアップ (個数を得る)

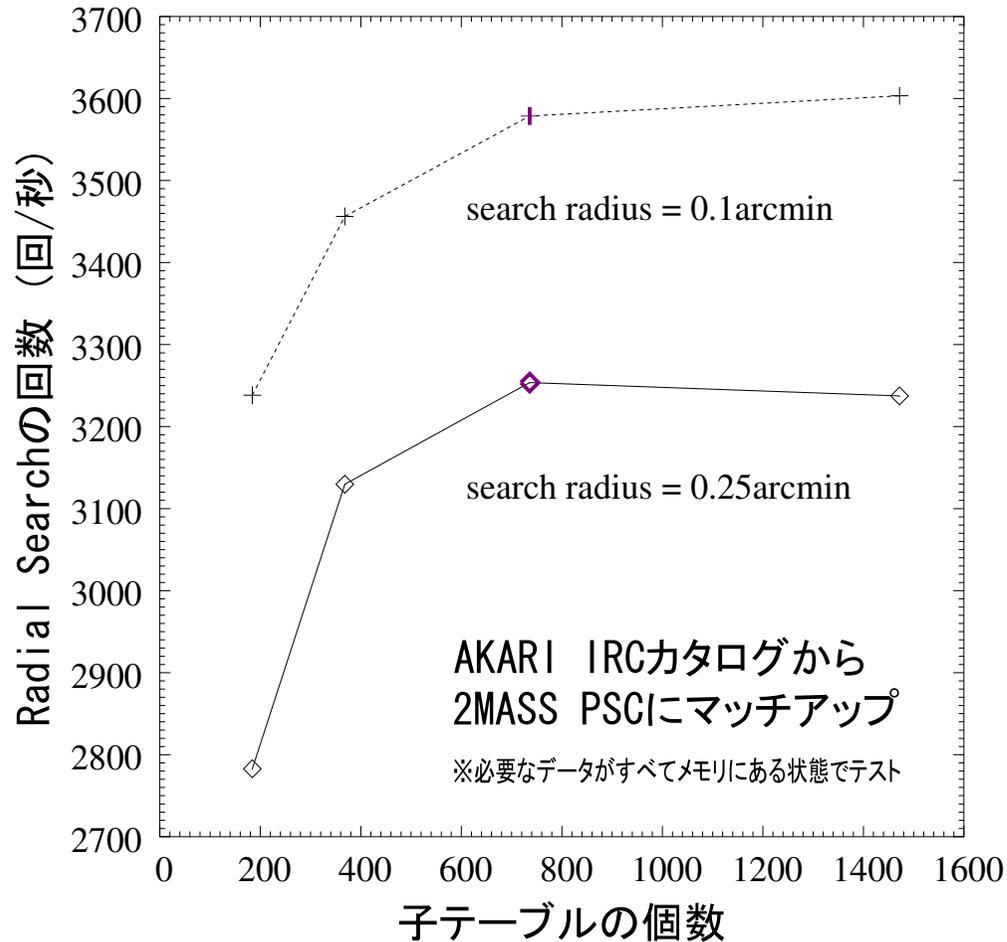
```
SELECT count (fTwomassGetNearestObjIDEq(o.ra,  
                                         o.dec, 0.1))  
FROM ( SELECT * FROM akari_irc ORDER BY dec ) o;
```

- Radial Search(ワンタイムサーチ)

```
SELECT M.*           -- 天体情報をメインテーブルから取得  
FROM  
(                  -- Radial Search を行なう  
  SELECT *         -- カラム (objid, 座標, 距離) の表が返る  
  FROM fTwomassGetNearbyObjEq(1.2, 3.4, 10)  
) F  
JOIN Twomass M     -- メインテーブル  
ON   F.objid = M.objid;
```

PostgreSQL-8.4 を使った大規模カタログ対応 (7)

2MASS カタログのための最適なパーティション数



計測に用いたマシン: DP Opteron 2384(2.7GHz) 64GB メモリ

PostgreSQL-8.4 を使った大規模カタログ対応 (8)

普通の PC でのパフォーマンス (マシン起動直後に計測)

カタログ	サーチ半径	ディスク	経過時間	Radial-Search 回数
AKARI-IRC PSC	0.25arcmin	RAID1(HDD×2)	18.2 分	798 回/秒
AKARI-IRC PSC	0.25arcmin	SATA2 SSD	8.2 分	1778 回/秒
AKARI-IRC PSC	0.25arcmin	SATA3 SSD	7.0 分	2078 回/秒
Tycho-2	0.25arcmin	SATA3 SSD	15.9 分	2666 回/秒
Tycho-2	0.1arcmin	SATA3 SSD	15.2 分	2788 回/秒

上記カタログから 2MASS カタログに対しマッチアップした .

AKARI-IRC PSC は約 87 万天体 . Tycho-2 は約 250 万天体 .

計測に用いたマシン: Core2Quad 3.0GHz 8GB メモリ + Crucial C300 SSD

PostgreSQL-8.4 を使った大規模カタログ対応 (9)

いろいろな PC + SSD での結果 (マシン起動直後に計測)

ハードウェア	経過時間	Radial-Search 回数
Q9650 (3GHz) + X48 + U3S6	15.2 分	2788 回/秒
XEON E5540 SMT:OFF (2.53GHz) + X58 + U3S6	17.6 分	2405 回/秒
PhenomII X4 945 (3GHz) + 890GX	18.0 分	2351 回/秒

Tycho-2 カタログ (約 250 万天体) から 2MASS カタログに対しマッチアップ
検索半径は 0.1 arcmin

ディスクインタフェースはすべて SATA3 . SSD は Crucial C300 を使用

PostgreSQL-8.4 を使った大規模カタログ対応 (10)

UP XEON E5540 + SSD で , マルチコア&SMT を活用:

1 つの PostgreSQL サーバ上で , N 個同時に SQL ステートメントを実行

プロセス数 N	経過時間	Radial-Search 回数 (平均)
1	17.7 分	2395 回/秒
2	10.8 分	3926 回/秒
4	5.57 分	7597 回/秒
8	3.23 分	13094 回/秒

Tycho-2 カタログ (約 250 万天体) から 2MASS カタログに対しマッチアップ
検索半径は 0.1 arcmin

・ 2 プロセスの場合の SQL ステートメント

```
SELECT count (ftwomassGetNearestObjIDEq(o.ra, o.dec, 0.1))  
FROM ( SELECT * FROM tycho2 WHERE objid % 2 =  ORDER BY dec ) o;  
                                     ↑  
                                     0 or 1
```

PostgreSQL-8.4 を使った大規模カタログ対応 (11)

細かなチューニング

- CPU 等の動的クロッキングの停止
 - インテル系では, BIOS の C1E は必ず無効に (1 割以上改善) . EIST は影響が小さいが, 止めた方が改善する事も .
 - AMD 系では, BIOS の powernow または OS の cpuspeed を止める (最近では AMD 系でも C1E があるので注意) .
- `effective_cache_size` を 1GB 以下に設定
2GB 以上にすると, パフォーマンスが低下
- SSD に対する先読み設定
`hdparm` で 1024 程度にする . これで数パーセント改善
- SSD は `ext4` でフォーマット
`ext3` や `xfs` よりも数パーセント改善

PostgreSQL-8.4 を使った大規模カタログ対応 (12)

2MASS Catalog Server Kit の紹介

- 誰でも 2MASS カタログの高速検索サーバが構築可能
google で「"2mass kit"」で検索



ファイル(E) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

http://www.irisas.ac.jp/~cyamauch/2masskit/index.ja.html

2MASS Catalog Server Kit (日本語版)

2010年11月 Version 1.0

山内@宇宙研

Special Thanks: 瀧田怜様, 大藪進喜様(Linuxでのテスト), 池田紀夫様(MacOSXへの対応), 板由房様(ドキュメントのレビュー)

[イントロダクション](#) / [必要条件](#) / [ダウンロード](#) / [インストール手順](#) / [PostgreSQLの各種設定](#) / [チューニング](#) / [使い方](#) / [ストアドファンクション](#) / [FAQ](#)

イントロダクション

2MASS Kitは、誰でも簡単に [2MASS](#)カタログ (約4億7千万天体) の高性能検索サーバを構築可能にした、ソフトウェアキットです。座標検索 (Radial Search, Rectangular Search, Cross-ID) についてチューニングを施し、単一テーブルで単純にインデックスを作成する方法に比べて、一桁以上の高速化を実現しています (もちろん、WCSTOOLS の scat よりもずっと高速です)。特に他カタログとのマッチアップ(Cross-ID)に関しては 徹底的にチューニングしており、最良の条件下においては秒間3000回以上のCross-IDが可能です。

このキットの大きな特徴は、「チューニングの柔軟性」です。各テーブル・インデックスを 7つのテーブルスペース (別々のディレクトリ) に登録する事により、必要な部分だけを高速デバイスに移動する事が簡単に行なえるようになっていました。特に最近のSSDの速度性能の進化は凄まじいものがあり、予算に応じてテーブル・インデックスの一部、またはすべてを 高速SSDに移せば、これまでデータセンターでしか実現できなかったような高性能サーバを 低コストで構築できます。

http://www.irisas.ac.jp/~cyamauch/2masskit/index.ja.html#TUNE

まとめ

- 天体カタログは，新天体の搜索，観測での較正，データ解析による研究で使われ，データ解析ではカタログ間でのマッチアップの需要が高い．
- PostgreSQL-8.4 と最近の SSD を活用し，マッチアップにおいて，2MASS カタログ (4.7 億天体) に対して秒間 2000 ~ 3000 回の位置検索を，普通の PC で可能にするシステムを開発した．
- CE を使わず，自作関数で子テーブルを選べば，約 1000 個のパーティションまでパフォーマンスアップを狙える．
- 直交座標系の位置については，2 バイト整数値に対してインデックスを作成する事で，ディスク I/O を最小化
- 1 台の PC でも，SSD+マルチコア&SMT の活用で 2MASS カタログに対して秒間 10000 回以上の検索が可能

おまけ

- **低価格対決 Crucial C300 v.s. Intel X25-M では , SATA-2 インタフェースの場合でも , C300 の勝ち .**
- **SATA-3 インタフェースカードは , PCIe x4 以上のものを買いたい .**
- **安価な SATA-3 インタフェースカードとして , ASUS U3S6 がある . パフォーマンスは抜群だが , 相性問題あり .**
 - X38 , X58
 - X48
- × **MCP55Pro**
- **6Gbps SAS カード LSI 9211-4i が C300 との組み合わせで良い結果を得ている . C300 × 3 で RAID0 を組めば , シーケンシャルリード 1GB/秒を軽く越えてくる .**