Astronomical Data Analysis Software and Systems XII
ASP Conference Series, Vol. 295, 2003
H. E. Payne, R. I. Jedrzejewski, and R. N. Hook, eds.

SkyQuery – A Prototype Distributed Query Web Service for the Virtual Observatory

Tamás Budavári, Tanu Malik, Alex Szalay, Ani Thakar

Dept. of Physics and Astronomy, The Johns Hopkins University, Baltimore, MD 21218

Jim Gray

Microsoft Bay Area Research Center, San Francisco, CA 94105

Abstract. We present SkyQuery[1], a distributed query system for astronomical catalogs. Using XML Web Services, SkyQuery federates databases at different locations and provides a programming and user interface to access catalog data as easily as if they were in a single database server. The data nodes, called SkyNodes, make their catalogs available by implementing a Web services interface. They may be written in any programming language and hosted on any platform, since the underlying technology is inherently interoperable. SkyQuery recursively performs an implicit probabilistic spatial join of the catalogs on the fly and returns the selected properties from the chosen surveys along with a best guess spatial position and the probability of the match being correct.

1. Introduction

One of the fundamental goals of the Virtual Observatory is to enable astronomical services to be used cooperatively to answer potentially very complex scientific questions. The idea is that these services would build on top of each other, similar to the packages in IRAF. The *core* services provide access to astronomical data and publish certain basic tools, while the *higher level* services would perform more sophisticated scientific analyzes relying on results from the core services. The hierarchy of services would provide a standard interface to all public astronomical resources.

XML Web Services is an industry standard (W3C) that perfectly suits the needs of the astronomical community. It is built on other standards such as XML, XSD, SOAP and WSDL, which guarantees interoperability between platforms and makes it independent from programming languages. Anyone can implement and consume Web Services using practically any computer and language. Currently there exist two fully functional development environments for Java and the .NET Framework that make programming Web Services very easy.

[1] `http://www.SkyQuery.net/`

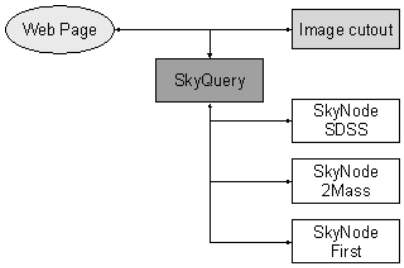

Figure 1. The relation of the different web services in the SkyQuery application. The SkyQuery web service calls the SkyNode web services on the participating archives. The Image Cutout service provides an image display of the search area.

Here we describe a prototype distributed query system for the VO, a hierarchy of Web Services that federates astronomical databases possibly located all over the World.

2. SkyQuery Architecture

SkyQuery is a network of Web Services. The Portal provides an entry point into the distributed query system relying on the metadata and query services of the database SkyNodes. The SkyNodes are the individual databases located at different sites along with their WS wrappers. At present, there are 3 SkyNodes linked into SkyQuery: (1) SDSS, (2) 2MASS and (3) FIRST. Having the SkyNodes registered in the Portal, the complexity of the network can be completely hidden from the user, see Figure 1. A sample user interface is implemented as a Web application on the project site that can submit queries, search the metadata and render the XML DataSet into an HTML table. The client web applications also uses the Sloan Digital Sky Survey's Image Cutout web service to display the composite color image of the sky specified in the query.

The primary entry point to SkyQuery is a Web method that accepts a request in an extended SQL format. The slightly modified syntax was required to specify the target archives and area on the sky and to parameterize the probabilistic cross-matching algorithm. Figure 2 shows a sample query.

3. The SkyNode

The data nodes publish functionalities that are consumed programmatically by the portal. These methods provide access to the data and metadata of the archive. Anyone can publish her data through SkyQuery by implementing a few SOAP methods regardless of how the data are stored. In fact, the 3 methods below are currently the only requirements to register a SkyNode:

- `Info(keyword)` – Return basic facts, e.g., survey name, area coverage or astrometric precision in arcsecs
- `Query(sqlcmd)` – Search the catalog by executing the SQL command and return the results in an XML DataSet

```
SELECT o.objId, o.type, o.i, t.objId, t.j_m
  FROM SDSS:PhotoPrimary o, TWOMASS:PhotoPrimary t
  WHERE XMATCH(o,t)<3.5 AND AREA(181.3,-0.76,6.5)
        AND o.type=3 AND t.j_m>14 AND (o.i-t.j_m)<1
```

Figure 2. The query syntax for SkyQuery is similar to SQL. There are special target designators to specify the archive, and there are two special operators, `AREA` and `XMATCH`, used to constrain the search area and the search accuracy.

- `XMatch(xplan)` – Complex task to retrieve data from another SkyNode and cross-match with own data according to the execution plan

The first three SkyNodes were implemented in C#. The .NET Framework Class Library provides a great set of tools not just for developing Web Services but also to access the SQL Server 2000 database that we used at the back end. The cross-matching algorithm was developed entirely inside SQL server with user defined stored procedures. An HTM based spatial indexing supports the fast matching algorithm.

4. Data Flow

How does it really work? The Portal receives a request and parses the query. After locating the referenced SkyNodes, it submits a simple SQL query in parallel to every SkyNode using the `Query()` method to get an estimate for the number density of the objects satisfying the selection criteria. For example, the sample query in Figure 2 is looking for galaxies in the SDSS survey (`o.type=3`) matched with objects in the 2MASS that are fainter than 14^{th} magnitude in the J band (`t.j_m>14`). Based on the results, the portal arranges the SkyNode into an execution plan so that running the distributed query would minimize the network traffic. The portal then just executes the plan by calling the `XMatch()` method of the first SkyNode in the "stack" and waits for the results to come back from the SkyNodes that can be just relayed back to the user.

The first SkyNode (`SkyNode 1` in Figure 3) looks at the plan and decides if it has the information to satisfy the request. If not, it then recursively calls the next SkyNode with a simpler execution plan and so on until the last node in the plan (`SkyNode 3`) will see a simple SQL query that it can run against the local database. These requests are done by passing only very light-weight objects on the wire but now real data start streaming from one SkyNode to another. Having received the data from the bottom data node, `SkyNode 2` can do its job: first it matches the catalogs using the astrometric precisions and probabilistic thresholds, then applies the selection criteria and returns the data back to one level up. All that is carried out within the database. Only the necessary parameters are propagated that were selected by the user or that are needed to perform the cross identification of the catalogs. The mixed constraints, e.g., `(o.i-t.j_m)<1`, are applied as soon as they can be evaluated. The result from the top level SkyNode is sent to the user.

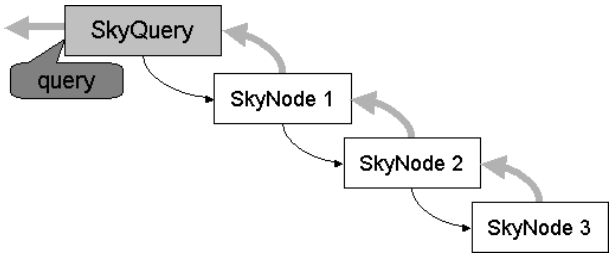

Figure 3. SkyQuery runs the distribution query by creating an execution plan for the SkyNodes that call each other recursively to satisfy the selection criteria. The narrow lines represent the query requests and the wide lines represent the actual data flow.

For extra credit, the system can calculate the best positions of objects based on the positions measured by the individual surveys and it can also quote a probability on the match-up. The web application at the project site automatically adds the columns of these parameters to the result table.

5. Summary and Future Works

The observations of large all-sky surveys are stored in separate databases and due to the rapidly changing technology these data sets cannot be built into a centralized system. SkyQuery can federate databases using XML Web Services. It can cross-match many catalogs on the fly using a probabilistic fuzzy join or it can look for drop-outs in certain catalogs. SkyQuery has proven that astronomical services may be adequately implemented as Web Services.

SkyQuery is a work in progress. The planned enhancements to the prototype demonstrated here include support for complex area specification and an advanced query language that is more flexible, e.g., allows local table joins. Next, survey footprint services will be added to the SkyNodes and the dynamical SkyNode registration to the Portal.

Additional SkyNodes will be added soon. A new SkyNode is on its way at the Institute of Astronomy in Cambridge, UK, to publish the Wide Field Survey catalog of the Isaac Newton Telescope.

Acknowledgments. This work is supported partly by a NASA AISRP 2001 grant NRA-00-01-AISR-035.

References

Szalay, A. S., & Gray, J. 2001, Science, 293, 2037

Szalay, A. S. et al. 2002, Proc. of SPIE, 4846, in press

Kunszt, P. Z., Szalay, A. S., & Thakar, A. 2001, 'The Hierarchical Triangular Mesh' in Mining the Sky: Proc. of the MPA/ESO/MPE workshop, Garching, A. J. Banday, S. Zaroubi, M. Bartelmann (eds.), (Springer-Verlag Berlin Heidelberg), 631