Astronomical Data Analysis Software and Systems XX
ASP Conference Series, Vol. 442
Ian N. Evans, Alberto Accomazzi, Douglas J. Mink, and Arnold H. Rots, eds.

Service Infrastructure for Cross-Matching Distributed Datasets Using OGSA-DAI and TAP

Mark Holliman,[1] Tilaye Alemu,[2] Alastair Hume,[2] Jano van Hemert,[3] Robert G. Mann,[1] Keith Noddle,[1] and Laura Valkonen[3]

[1]*WFAU, Institute for Astronomy, University of Edinburgh*

[2]*EPCC, University of Edinburgh*

[3]*School of Informatics, University of Edinburgh*

Abstract. One of the most powerful and important goals for VO developers has been to enable cross-match queries between disparate datasets for end users. This has only been achieved within the VO using the early SkyNode infrastructure and has not been reproduced using current IVOA standards. To remedy that situation, the Wide Field Astronomy Unit (WFAU) has worked with the Edinburgh Parallel Computing Center (EPCC) in leveraging the OGSA-DAI grid middleware to enable cross catalog queries on distributed VO services. We have achieved this goal by building a three layer service stack that places the OGSA-DAI software above multiple individual services implementing the IVOA's new Table Access Protocol (TAP), and then a single TAP service is placed above this and presented to the end users. Users can then execute ADQL queries that cross-match between the disparate datasets as though they were in the same database with acceptable performance rates on the resulting data flow. The OGSA-DAI software is able to interrogate any compliant TAP service to acquire the necessary metadata for insertion into the single federated TAP service used for cross-match queries. We are currently testing this distributed infrastructure using the TAP services provided by WFAU for the UKIDSS DR3 and SDSS DR7 datasets in combination with the TAP service available from the Canadian Astronomy Data Center (this last without requiring any action from CADC staff). This forms the basis for a large-scale distributed data mining workflow and similar activities can be readily implemented as more TAP services come online. Future work will involve releasing this infrastructure to the greater astronomical community as an IVOA compliant service for users.

Cross-Matching Distributed Datasets

At present there is no singular tool for cross-matching distributed datasets using current IVOA standards. It is with this aim that we have investigated leveraging the distributed query processing (DQP, Dobrzelecki et al. 2010) functionality of the OGSA-DAI data access and integration middleware (Jackson et al. 2007) to enable cross catalog queries over the IVOA ratified Table Access Protocol (TAP, Dowler et al. 2010) services.

Cross-matching distributed datasets is a non trivial problem due to the difficult nature of query planning and execution over infrastructure that is both disparate and heterogeneous. Astronomy datasets are housed by many individual institutions that often use different backend hardware and software platforms for storing and serving their data. The problems are further compounded by the growing sizes of astronomical datasets which cannot be transported easily or loaded entirely into memory for query execution. The combination of these problems has led to a situation whereby there is no straightforward method for joining datasets offered by different services.

Fortunately there are IVOA standards that address query language and data access which help minimize the problems caused by heterogeneous dataset backend infrastructure. In particular the ADQL standard (Ortiz et al. 2008) unifies the query language necessary for writing queries to astronomical data services. Furthermore, the TAP standard defines a simple yet powerful interface for uniformly accessing astronomical datasets using ADQL queries and producing VOTable output. OGSA-DAI is a data access and integration middleware solution that has been developed to provide a standard interface to distributed underlying databases, and it includes powerful functionality for federating these databases in such a way that they can be accessed as though they were a single database. We have worked on building a service infrastructure that utilizes the powerful dataset federation and query handling capabilities of the OGSA-DAI middleware alongside the uniform TAP service interfaces for obtaining data in order to build a service infrastructure that enables distributed dataset cross-matching.

OGSA-DAI and DQP

The OGSA-DAI server software sits above the database(s) and allows users to build workflows containing queries which it then executes to return query results. OGSA-DAI includes a component called Distributed Query Processing that executes queries (including join operations) across distributed databases. DQP performs the tasks of parsing the query, and builds, optimizes, and executes the query plan. It federates the schemata from the underlying databases into a single schema that acts as though all the tables are within a single database. In order to improve the efficiency of the query execution DQP optimizes the query plan such that as many of the expensive operations as possible are pushed down to the database level for final execution. This ensures that the amount of data that needs to be moved for performing the join operations is minimized.

DQP query plans are executed on one or more OGSA-DAI servers. OGSA-DAI offers a range of possible join algorithms and it is the task of the DQP query plan optimization to choose the most appropriate algorithms for a given query. The performance of these different joins varies significantly depending on the statistics and sizes of the underlying databases. The available join algorithms include:

- In-memory join: One side is transferred and stored in memory, the other side is streamed,
- Partial in-memory join: Gets first results quickly, then all data are stored to disk for joining,
- Ordered merge join: Both inputs are ordered by the join attribute thus supporting an efficient fully streamed join,
- Parallel hash equi-join: Multiple OGSA-DAI servers can be used to implement the join in parallel,
- Batch joins using IN clauses: e.g. SELECT * FROM foo WHERE bar IN (x,y,z).

Table Access Protocol

TAP is a recently approved IVOA protocol for accessing tabular data in a standardized way. Current implementations allow users to submit queries in ADQL through standard HTTP GET and POST methods, and return results in VOTable format. Users can access TAP services using simple clients like web browsers or "wget", or software like VODesktop (AstroGrid 2010) and TAPsh (Demleitner 2010). There are a limited number of TAP services currently published on the VO, but more are expected in the near future. TAP is anticipated to be highly utilized as it offers a more powerful interface to underlying datasets than the current "simple" IVOA protocols like cone-search and image access protocol.

Cross-Matching Infrastructure

We have designed a 3-layer architecture that places the OGSA-DAI middleware above an arbitrary set of TAP services. OSGA-DAI combines the metadata from the underlying datasets, and this is then exposed through a single TAP interface at the top layer, enabling users to perform cross-match queries in ADQL on all of the federated datasets underneath.

We implemented a test infrastructure using an OGSA-DAI deployment (seen in Figure 1) that federates the UKIDSS DR3 TAP service and the Canadian Astronomy Data Center's (CADC) TAP service for CAOM data (CADC 2010). These are independent TAP services, and no coordination or assistance was required from CADC in building the test system. We then submitted several scientifically relevant ADQL queries that cross-matched the different datasets to determine problems and performance for the system. These queries utilized a variety of different join types and expected result set sizes in order to properly stress the service infrastructure. Overall the test system

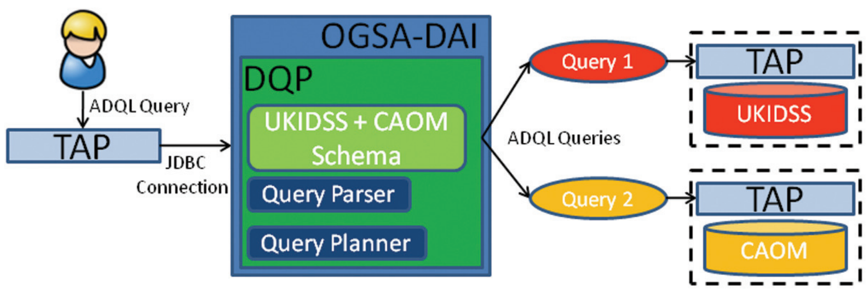

Figure 1. Cross-match testbed diagram.

successfully demonstrated the viability of the service architecture, returning results for most queries. We have noted those queries that failed and initiated investigation into the causes of their failure to determine at what point in the service infrastructure the failure occurs. The majority of failures were due to query language differences between ADQL and the subset of SQL utilized by the DQP component of OGSA-DAI.

The actual query times varied greatly depending on a number of factors, which included the join algorithm utilized, size of tables to be joined, and result set size. Though not as fast as native systems, the query times were acceptable given the distributed nature of the data and the novelty of the cross-matching capability. For some queries the query plans built by DQP are not as efficient as hand written OGSA-DAI workflows. This is primarily because DQP does not choose the most suitable join algorithm due to the currently limited set of statistics regarding the sizes and attribute value distributions of the datasets. Our main test query selected a portion of the UKIDSS Large Area Survey and then cross-matched those objects against the entire CAOM, and it completes in approximately 70 minutes. Using a hand optimised OGSA-DAI workflow this time can be reduced to around 3 minutes. The main optimisation not currently performed by DQP is the use of a range join algorithm that uses a two sided range index ($x < y$ AND $x > z$) to match values rather than a single-sided range ($x < y$). The other important optimisation is the use of a more compact data transfer format rather than the default XML VOTable representation. It will not be difficult to update DQP to perform these optimisations. This time could be further reduced by placing an OGSA-DAI server geographically close to the CAOM TAP server which would reduce the overall time spent transferring data over the internet.

Future Development

The demonstrated success of our test infrastructure points to a number of possibilities for future development. One ultimate goal is to deploy a registered TAP service utilizing the OGSA-DAI infrastructure to federate data from all the TAP services published on the VO. Such a service would be one of a kind in allowing users to cross-match all the databases and catalogs available. Other possibilities for user services that utilize this system are being explored with input from VO developers and astronomers.

In the meantime we are working on addressing the issues discovered through testing to improve the functionality as well as the robustness of the service infrastructure. In particular we would like to address the issues of error handling through the service layers, and also to better optimize query plans for complex joins of distributed databases. All of this will be investigated; in addition, we plan on implementing more join algorithms, improving join algorithm choice and execution, adding support for ADQL spatial functionality, and improving performance for joins of large (multi-terabyte) databases. A demonstration webpage can be found at: `http://www2.epcc.ed.ac.uk/~ally/aida/Demo1.html`.

References

AstroGrid 2010, Vodesktop. URL `http://www.astrogrid.org/`

CADC 2010, Caom. URL `http://cadcwww.dao.nrc.ca/caom/`

Demleitner, M. 2010, Tapsh. URL `http://vo.ari.uni-heidelberg.de/soft/tapsh`

Dobrzelecki, B., et al. 2010, Phil. Trans. Royal Soc., 368, 4133

Dowler, P., Rixon, G., & Tody, D. 2010, Table Access Protocol, Version 1.0. URL `http://www.ivoa.net/Documents/TAP/`

Jackson, M., et al. 2007, in UK e-Science All Hands, edited by S. J. Cox (Edinburgh: National e-Science Center), vol. 2007 of UK e-Science All Hands

Ortiz, I., et al. 2008, Ivoa astronomical data query language, version 2.0. URL `http://www.ivoa.net/Documents/latest/ADQL.html`