The SDSS Catalog Archive Server

Ani Thakar (thakar@jhu.edu), October 1, 2014

CAS Architecture

The SDSS catalog data is served to the outside world by the SDSS Catalog Archive Server (CAS) [1]. The heart of the CAS is a commercial database management system (DBMS) that provides the essential functions of data storage, data integrity and data recovery as well as high-performance data access via a versatile SQL query engine [2]. The CAS system also encompasses several other components. The data-loading pipeline used to ingest data into the DBMS is the sqlLoader; the Web portal that provides synchronous online browser-based access to the CAS is the SkyServer; and the asynchronous (batch) query workbench that enables users to run complex, time-consuming SQL queries and write output to their own personal server-side relational databases called "MyDBs" is CasJobs. The CAS architecture is illustrated in Fig. 1, which shows the 3 "pillars" of data access resting on the foundation of a complex and highly optimized science schema built into a relational DBMS. We also built a spatial index library into the database – the Hierarchical Triangular Mesh (HTM) – to facilitate fast searching for spatial data [3]. Calls to HTM functions are compiled right into the database code via the C# CLR (common language runtime) feature in SQL Server.

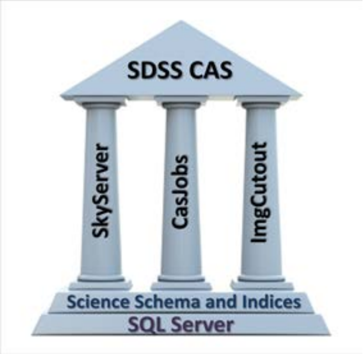


Figure 1. The SDSS "CAStle", built upon a foundation of a commercial DBMS and science-specific schema and indices, and resting on the 3 pillars of synchronous web access (SkyServer), asynchronous batch access (CasJobs), and visual browsing (ImgCutout).

A Relational DBMS

We chose Microsoft's SQL Server, a Windows- based relational DBMS, as the CAS's data repository. We started with an object-oriented DBMS, with the idea that an object data model would provide a much better conceptual match to the SDSS data as well as superior performance compared to the commercial relational DBMS technology available at the time (early-to-mid 1990s). However, even though the object DBMS's performance was initially satisfactory, it soon became insufficient for the SDSS project's data mining needs [4]. Over the same period, relational DBMS technology advanced to the point where it overtook object DBMSs in terms of ease of use, performance, and reliability features.

The decision to select SQL Server was mostly pragmatic and based on our immediate needs and resources, rather than a comparative evaluation of relational DBMS products. Nevertheless, SQL Server is highly rated for its ease of use and administrative features, and has one of the best query optimizers in the business. The features that are particularly important to SDSS from a performance perspective are the CLR that enables compiled code to run in the database, the ability to write complex user-defined functions and stored procedures to build critical science and data access functionality right into the database schema, and the extensive support for floating point data types.

The unit of data storage in a relational DBMS is a two-dimensional table of rows and columns. As such, all the CAS data goes into tables in the DBMS.

The CAS Data Model

Figure 2 shows the SDSS CAS schema for DR12, which is divided by survey (SDSS legacy, BOSS, SEGUE-2, APOGEE and MARVELS) and function into groups of tables: photometric data tables (Photo group), spectroscopic data tables (Spectro group), and metadata tables that contain documentation and other schema information.

Photo Tables and views

The Photo group of tables holds the SDSS photometric pipeline's outputs—that is, the parameters computed from data taken with the SDSS imaging camera. The main table in this group is the PhotoObjAll table, which contains the photometric parameters for each astronomical object that the photometric pipeline identifies. This superset of all observations recorded by the SDSS camera also includes repeat observations of objects. For objects that have multiple recorded observations, the

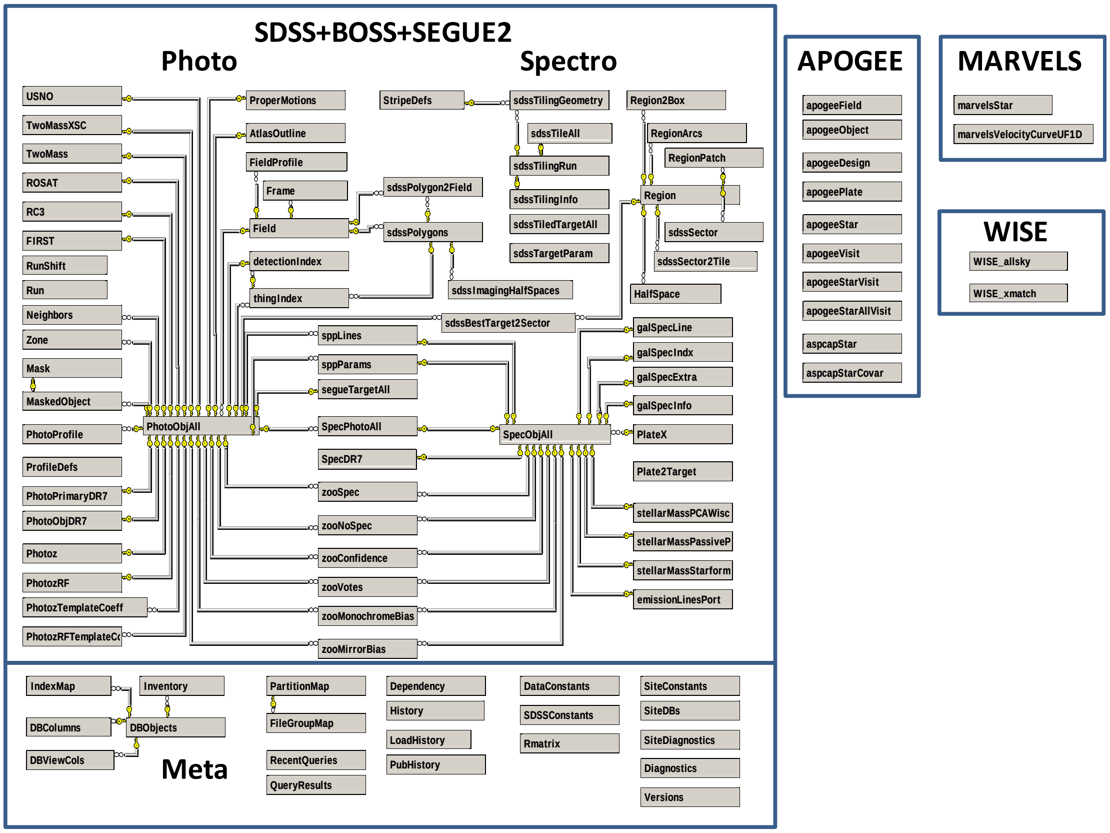


Figure 2. The SDSS DR12 schema showing tables and their relationships grouped by survey and function.

best one is marked as primary in the SDSS photometric pipeline; other observations are marked secondary if they're good enough for science or family for anything else.

Spectro Tables and views

The outputs from the one-dimensional Spectro pipeline, spectroscopic targeting, and tiling information are contained in the Spectro group of tables. The main table in this group is the SpecObjAll table, which is analogous to the PhotoObjAll table in the Photo group but with spectra instead of images. Also analogous to PhotoObj is the SpecObj view of the SpecObjAll table, which contains unique spectra determined by the SDSS to be fit for science.

Table 1 shows the sizes of the largest tables in BestDR10. The PhotoObjAll table, with more than 1.2 billion rows, is by far the largest table in the entire database, and its total size (including the database indices defined on it) exceeds 4 TB in DR10.

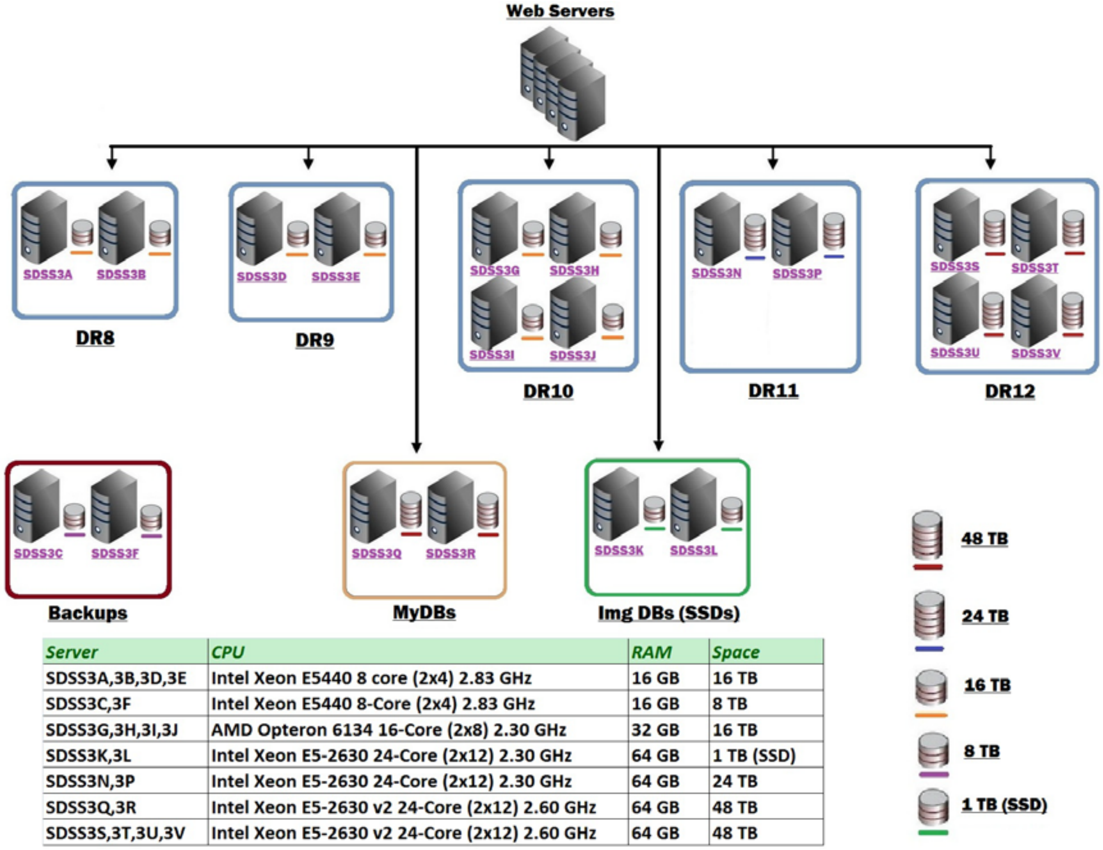
Table 1. Rows and sizes in GB for the largest tables in DR10 (size > 10 GB).

Name	Rows	Data GB	Index GB	Total GB
PhotoObjAll	1,231,051,050	3130.7	1579.1	4709.8
PhotoProfile	44,563,999,574	1137.1	6	1143.1
AtlasOutline	1,219,412,987	1072.1	3.6	1075.8
Frame	3,752,184	989.8	0.5	990.3
Neighbors	25,578,382,962	942.7	6.4	949.1
WISE_allsky	563,921,584	537.8	149.9	687.7
SpecObjAll	3,358,200	185.9	1.1	187
PhotoObjDR7	364,857,538	111.3	7.1	118.4
PhotoPrimaryDR7	305,789,541	93.3	0.4	93.8
SegueTargetAll	453,975,934	51.7	8.4	60.1
Zone	794,035,877	49.7	0.4	50
thingIndex	563,688,948	35.8	10.3	46.1
detectionIndex	932,891,133	26.5	16.9	43.4
FIRST	510,679,510	40.2	0.1	40.3
USNO	253,732,084	38.7	0.2	38.9
Photoz	208,478,448	31.8	0.2	32
ProperMotions	336,954,036	29.2	0.1	29.4
TwoMASS	103,577,231	14.4	11.7	26.1
WISE_xmatch	495,003,196	16.9	8.9	25.8
PhotozRF	208,478,448	25.2	0.1	25.4
PhotozTemplateCoeff	517,961,077	12.2	0.1	12.3
PhotozRFTemplateCoeff	505,398,590	11.9	0.1	12

The PhotoProfile table, which has the largest number of rows (more than 44 billion), clocks in at a little over a TB as the second largest table in the database. The Frame table, containing all the JPEG images for the ImgCutout service, is almost a TB.

Metadata Tables

The Metadata group includes the contents of the documentation pages in the CAS Web sites (SkyServer and CasJobs), which we can broadly divide into the Help and Schema Browser pages. Together, the metadata tables provide an application-independent description of a DBMS that we can use as a template for other scientific archives. Indeed, the Schema Browser group of tables was used in the Virtual Observatory community as a template for SkyNodes in the Open SkyQuery federated query service [5].



Server	CPU	RAM	Space
SDSS3A,3B,3D,3E	Intel Xeon E5440 8 core (2x4) 2.83 GHz	16 GB	16 TB
SDSS3C,3F	Intel Xeon E5440 8-Core (2x4) 2.83 GHz	16 GB	8 TB
SDSS3G,3H,3I,3J	AMD Opteron 6134 16-Core (2x8) 2.30 GHz	32 GB	16 TB
SDSS3K,3L	Intel Xeon E5-2630 24-Core (2x12) 2.30 GHz	64 GB	1 TB (SSD)
SDSS3N,3P	Intel Xeon E5-2630 24-Core (2x12) 2.30 GHz	64 GB	24 TB
SDSS3Q,3R	Intel Xeon E5-2630 v2 24-Core (2x12) 2.60 GHz	64 GB	48 TB
SDSS3S,3T,3U,3V	Intel Xeon E5-2630 v2 24-Core (2x12) 2.60 GHz	64 GB	48 TB

Figure 3. SDSS-III server farm for DR8-DR12, showing the different types of servers supporting web applications, MyDBs, image databases and SDSS DR databases.

CAS Hardware

Figure 3 shows our production cluster for SDSS-III releases DR8 through DR12 (projected). We have 2-4 database (DB) servers per release, so that the public and collab users as well as CasJobs users can be adequately supported, and SkyServer, ImgCutout and CasJobs queries can be load-balanced on different boxes. For each data release that is of maximal interest to users (usually the most recent one or two data releases), we maintain two CasJobs contexts, one that is publicly accessible and one that is restricted to collaboration users (see CasJobs section below for access restrictions).

We dedicate one server (with an additional warm backup server) to ImgCutout queries for multiple releases, since these queries can be quite large in number and intensive.

At any given time, we have 2 or more DB servers in production per release, and the rest are used for backups, testing and data loading purposes. Generally the most recent data release that receives the maximum user queries is hosted on 3 or 4 DB servers. From past experience, we have found that for the most critical datasets, it was necessary to keep as many as 6 copies of the database (including one copy in cold storage) for maximum security, performance and fault-tolerance [6]. This means that 5 spinning copies on disk – the 4 online copies plus 1 on a development or load server – are recommended for the most recent data release.

Server-level load balancing is our most basic performance measure. Our earliest analyses of the SkyServer traffic logs [7] showed that a few data-intensive queries slowed down the performance for the vast majority of users running quick queries that skimmed or accessed a small part of the dataset, so we needed to segregate the two types of query traffic. As a result, we always follow these three rules for our production database servers:

1. Quick and long CasJobs queries are pointed to two separate servers.
2. SkyServer queries are pointed to either a separate server or share the server that fields the quick CasJobs queries.
3. Public and collaboration long queries in CasJobs are pointed to different servers. This is because one server can only support a few concurrent intensive queries.

For the most heavily used (i.e., the most recent) data release, this dictates that we allocate 4 servers for the best performance and load balancing. For older data releases, we can get away with 3 or 2 servers (the latter if there is no longer a collab-only context for that data release).

One of the two MyDB servers will be in production and the other one is used for warm backup (MyDBs are backed up daily to the backup machine).

CAS Data Release Cycle

The CAS issues data releases roughly once every year. The steps that ensure a successful data release are as follows (they are roughly in chronological order).

1. Set cutoff date for deciding which data will make into CAS for next DR.
2. Create or update schema for CAS data in the sas/sql svn product.
3. Update CAS sqlLoader schema with the changes in sas/sql (this is not automatic, on purpose).

4. If there is a new data product or data table involved in step 3, this will also require updates to the data ingest scripts in the sqlLoader pipeline, such as adding the capability to ingest new types of CSV files and adding data validation tests.
5. Generate CSVs on the SAS site for ingesting a subset of the new/updated data into a CAS TestLoad first. There is a FITs-to-CSV guide on the wiki to facilitate this.
6. Create a Testload database for that DR on a designated server in the CAS cluster.
7. Download the CSVs from the SAS to the CAS TestLoad server.
8. Load the CSV data into the TestLoad database for that DR (e.g. TestDR10). This is made available to a select group of testers in the collaboration via the TestDRx context in CasJobs.
9. Test the new/updated data in the TestDRx CasJobs context.
10. Repeat steps 5-8 until we are happy that the data is correctly loaded.
11. Generate CSVs for the full dataset to be loaded into CAS for that DR.
12. Create a BestDRx database (e.g. BestDR12) for that DR on the designated load server in the CAS cluster. When a DR involves a large data increment (more than a TB) or is reloaded from scratch (e.g., DR2, DR8), there is more than one load server for parallelism and faster loading, and one of the load servers is designated as the (load+)publish server that contains the merged BestDRx database.
13. Download CSVs for each data product (= separate load task) for that DR.
14. Initiate sqlLoader tasks to load the CSVs into the BestDRx database.
15. Once all load tasks for the DR have completed successfully, run the FINISH sqlLoader task to complete the steps for merging and consolidating the data, such as creating any computed tables and indices on the full dataset.
16. Set the version of that DR in the Versions table of the BestDRx database.
17. Create CasJobs context(s) for collaboration or public access to the DR, if it doesn't already exist.
18. Create instance of test or production SkyServer web interface for synchronous access to the DR, if it doesn't already exist.
19. Create instance of test or production ImgCutout web service for visual browsing of DR data, if it doesn't already exist.
20. Announce the availability of the data to the collaboration.
21. Repeat steps 11-20 for every data product intended for that DR. This typically happens over the course of months, and the collaboration gets proprietary access to every data product for a few months before the release goes public.
22. Make multiple copies of BestDRx on the CAS cluster for fault tolerance, load balancing and performance.
23. Announce DR to the public.
24. After the DR goes public, any corrections to the data or schema that are approved by the data management team are applied as patches to the databases using a SQL stored procedure that applies the patch, reruns the DB diagnostics and updates the DB version number. The trac ticket system documents all these changes as well. Patch files (SQL scripts) are checked into the sqlLoader patch directory in SVN, so mirror sites can apply the patches locally.

25. Bug fixes and updates to the CAS data access tools and interfaces are also tracked via the trac ticket system and applied as necessary. There are development and test versions of the SkyServer, CasJobs and ImCutout products maintained on separate dev and test servers at JHU. Each change is first applied to the dev version and tested internally first. It is then propagated to the test version where it is tested by appropriate members of the collaboration. The change is deployed to production using automated scripts to push the changes to the production web servers. The SkyServer and CasJobs sites each display the current (SVN or Git) version on their front pages.

sqlLoader Data Loading Pipeline

The sqlLoader pipeline consists of an ASP web application that runs SQL Server and Visual Basic scripts on the database servers to automate as far as possible the task of loading data into the SDSS databases. The sqlLoader makes use of the SQL Agent – a task scheduler available in SQL Server – to schedule and run loading tasks in the proper sequence. The SQL Agent also enables control to be handed over from one phase of the loading to the next. In addition to largely automating the tedious task of loading terabytes of data, the sqlLoader encapsulates all the schema (DDL files) for the database along with all the data validation and integrity checks, it incorporates the index creation logic and the publishing steps for all the different groups of tables in the database, and it maintains a rich set of statistics and logs that are available in perpetuity for analysis, reporting and troubleshooting.

The pipeline is described in detail in the CiSE 2008 special SDSS issue [8]. The web application that lets the database administrator manage the data loading is known as the Load Monitor, and it maintains a complete history of each loading step from the initial creation of the database through successive data releases. For SDSS-III, we have the complete data loading history since the initial loading of the data for DR8 in 2008-9. The sqlLoader product is checked into the SDSS-III SVN repository.

CAS Data Access

The CAS data access system design evolved from our analysis of the SkyServer traffic logs in the first two years of the project, which showed us that usage was primarily split between three different types of users:

- a large majority of casual users who ran quick queries that executed in less than a minute,
- a small minority of "power" users that ran queries that took hours to complete, and
- a significant but important population of "visual users" that browsed the image and catalog data visually.

Accordingly, as illustrated in Figure 1, we built our system around three main applications: SkyServer, CasJobs and ImgCutout.

SkyServer Web Portal

The SkyServer Web portal is an Microsoft Windows-based ASP.NET application that provides a multi-level, multi-language Web interface to the SDSS CAS. Since it first went live in June 2001, it has handled nearly 1.5 billion Web hits, averaging about 15 million hits/month in recent years. The SkyServer was originally developed in Javascript on the ASP (Active Server Pages) platform [9], and it was recently (for DR10) ported to C# in ASP.NET.

The SkyServer web application was originally maintained in the SDSS CVS repository at FermiLab. In 2008, it was moved to the SDSS-III SVN repository at NYU. Earlier this year, the newly reengineered ASP.NET version was moved to the SciServer GitHub respository maintained at JHU. The SkyServer web interface layout (stylesheet) was redesigned in 2007 prior to the launch of DR7, but since then has remained basically unchanged. It will be significantly redesigned for SDSS-IV.

The basic structure of the SkyServer web tree is shown in Figure 4. The site provides synchronous browser-based access to the SDSS catalog data for multiple levels of users, from the casual member of the general public to professional astronomer,

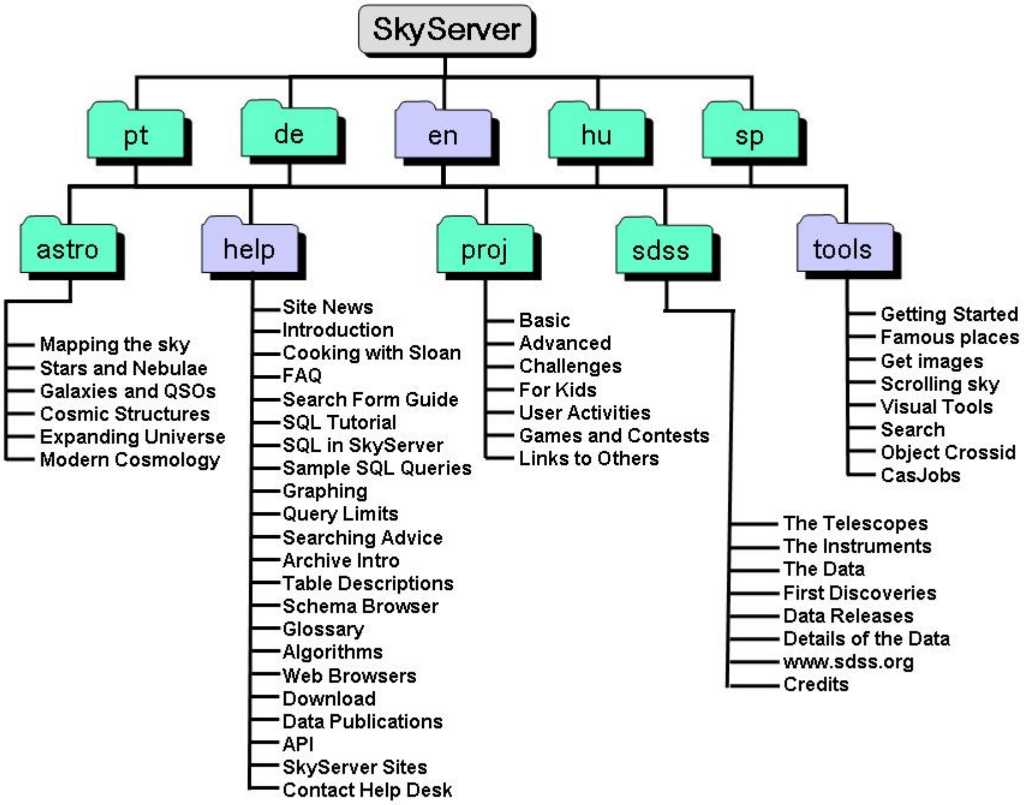


Figure 4. Structure of the SkyServer web tree with multi-language support (pt=Portuguese, de=German, en=English (default), hu=Hungarian, sp=Spanish). The folders shown in green are meant for educational and explanatory content about astronomy and the SDSS project that is of more interest to members of the general public, and hence is often translated to other languages.

including a rich set of exercises, games and informational pages for K-12 educators and students. The "tools" branch of the website contains various levels of query tools ranging from simple form queries to raw SQL. A visual tools section also allows browsing of the image data using the ImgCutout web service.

Multi-language support is provided for the educational and background content intended for non-astronomers. Although only the English version of the site is currently available, multiple language versions were supported for SDSS-II (DR1-7) and will again be supported again for DR13 and beyond.

CasJobs/MyDB Batch Query Workbench

CasJobs is the workhorse of the SDSS CAS data access stable, and it is designed to handle the bulk of the SQL queries submitted to the CAS by the "power users", i.e. those who submit complex and time-consuming queries that often retrieve very large amounts of data, from a few hundred MB to a few GB per query.

CasJobs has two main components – a back-end web service and a web application front end. Both are implemented in the ASP.NET framework using C#. There is also a command-line client written in Java that is downloadable from the web client. CasJobs is a realization of the primary Big Data access axiom that we have adopted from the very beginning: *bring the analysis (program) to the data, not vice-versa*. It gives every user their own server-side SQL Server database called MyDB, and query output for all batch queries is automatically directed to database tables in the user's MyDB. This keeps the analysis close to the data (on the server) and lets the user download only the data that they ultimately need.

CasJobs offers the following main functional features:

- Personal SQL Server database called MyDB that is 0.5 GB by default and can be increased in size to a few GB upon user request.
- Multiple query contexts (data sources) and the ability to run queries against them from a single site.
- A synchronous 1-minute query execution mode for quick queries
- A batch execution mode for intensive queries that typically allows 8 hours for the query to complete (this is configurable per site).
- A scheduler to control the number of concurrent queries per queue and per user to ensure adequate performance and fair use. A user may submit multiple concurrent queries in different queues but only one query per queue, and may not run consecutive queries in a given queue if other users are waiting.
- An Import page to allow users to upload their own data tables into their MyDB, so these data can then be used in queries against SDSS data.
- A searchable job history that records every query and output (download) job that the user ever submitted. Resubmitting a previous job can be done with a couple of clicks.

- A MyDB/Schema Browser page that allows users to browse their own data tables as well as switch the context to other databases (e.g. SDSS DR10) and browse those tables.
- Users can view the query that created any given MyDB table.
- Users can view the source code for View definitions, user-defined functions and stored procedures in the schema browser.
- A Groups feature that allows users to share their data with other CasJobs users by creating groups and inviting selected users to become members. Group members can view group tables on their MyDB page and use group tables in queries just like their own tables.
- Access control via Groups that allows CasJobs administrators to restrict access to certain contexts to members of specified groups.

CasJobs is described in detail in the CiSE 2008 special SDSS issue [10], and it has been adapted for several other astronomical and even non-astronomy applications, including GALEX, Pan-STARRS, Palomar Quest and 1000 Genomes.

ImgCutout JPEG Cutout Web Service

ImgCutout is also an ASP.NET/C# web service that uses the SkyServer visual tools web pages as a client to serve up 3-color JPEG versions of the SDSS image data to users in the form of a Finding Chart service, a Google Maps-style Navigate tool, and a postage-stamp style Image List service. The Explore tool in SkyServer that allows single-object browsing also uses the ImgCutout service to display a JPEG cutout of the object in question.

To allow rapid data visualization, images are stored in JPEG format. During the preprocessing stage, FITS images in 3 of the original 5 bands are combined to create a color JPEG image. The JPEGs are stored in the database at 5 zoom levels, along with associated astrometry information. This preprocessing facilitates rapid recovery of the frames and geometric transformations needed to create dynamically a mosaic of any region of interest. Mosaics can be as big as 2048 × 2048 pixels, covering approximately 34 × 34 degrees.

The ImgCutout service receives user requests for JPEG cutouts along with overlay options (issued from the SkyServer clients) and translates them into SQL queries against the Frame and associated tables to return the image cutouts with the appropriate overlay markup. Many interesting plots can be overlaid on top of the images. The Field option displays the boundaries of the original FITS images used to construct the mosaic. The Plates checkbox shows the boundaries of the aperture plates used to obtain spectra chosen from a list of potential Targets. Selecting SpecObj or PhotoObj highlights objects in the image with spectroscopic or photometric data respectively. Outline and BoundingBox options mark the pixels associated with each object.

Using the Image List tool, users can simultaneously view thumbnails of SDSS images for objects specified by a list of coordinates. The thumbnails provide a visual index for objects specified by the user or returned by an SQL query of the database. More detailed data views can be interactively selected from the Image List tool allowing creation of Finding Charts, for example.

The most interesting feature of this tool is that users can generate a list of coordinates and associated thumbnails by querying the database directly using SQL. For example, visual inspection of the thumbnails generated in response to the following search for high-redshift quasars:

```
SELECT specObjId, ra, dec
FROM SpecObj
WHERE
  class = ‘QSO’ and z > 3
```

yields a potential lensed quasar. Selecting the thumbnail brings up a more detailed image in the Navigator. Inverting the color map improves the visibility of faint background objects. Selecting PhotoObjs shows 5 closely spaced objects present in the photometric catalog. The Explore tool then allows detailed investigation of colors, morphological type, and spectrum (see Figure 5). Based on similar colors for some of the objects, follow-up observations were requested to test whether or not this particular group of sources is indeed a lensed quasar.

In general terms, the system is constructed as follows:

1. Combine and convert FITS into JPEG.
2. Store JPEGs with several zoom levels in the database.
3. Store astrometry for frames and objects.
4. Store other relevant information (masks, plates, etc).
5. Construct a spatial index (HTM).

The database provides images and object data to the image cutout service in response to SQL queries. The cutout service assembles the images into mosaics and displays the mosaics with the selected overlays.

The service can easily be adapted to work with non-SDSS data, as we recently demonstrated for DR10 by incorporating a monochrome cutout service for 2MASS data. Other web applications can use the ImgCutout service using the web service API to get SDSS images and cutouts, for example the WordWide Telescope (www.worldwidetelescope.org).

CAS URLs

The following is a list of CAS URLs currently in existence:

1. SkyServer (DR1-DR7): http://skyserver.sdss.org/drX or http://cas.sdss.org/drX, e.g. http://skyserver.sdss.org/dr7.
2. SkyServer (DR8-DR12): http://skyserver.sdss3.org/drX, e.g., http://skyserver.sdss3.org/dr10.

3. CasJobs: http://skyserver.sdss3.org/casjobs or http://skyserver.sdss.org/casjobs.
4. SkyServer admin/download site: http://skyserver.org/.
5. ImgCutout JPEG download (example call for DR10 cutout): http://skyservice.pha.jhu.edu/DR10/ImgCutout/getjpeg.aspx?ra=132.749&dec=11.656&scale=0.79224&width=400&height=400&opt=GST&query=SR(10,20).
6. sqlLoader User Guide: http://skyserver.org/help/sqlLoader.htm.
7. JHU SDSS site: http://www.sdss.jhu.edu/.

References

1. Thakar, A.R. 2008: "The Sloan Digital Sky Survey: Drinking from the Fire Hose", Computing in Science and Engineering, 10, 1 (Jan/Feb 2008), 9.
2. Thakar, A.R., Szalay, A.S., Fekete, G., and Gray, J. 2008: "The Catalog Archive Server Database Management System", Computing in Science and Engineering, 10, 1 (Jan/Feb 2008), 30.
3. Kunszt, P.Z., Szalay, A.S., Csabai, I., and Thakar, A.R. 2000, "The Indexing of the SDSS Science Archive", Astronomical Data Analysis Software and Systems IX, ASP Conference Proceedings, Vol. 216, edited by Nadine Manset, Christian Veillet, and Dennis Crabtree. Astronomical Society of the Pacific, ISBN 1-58381-047-1, 2000, 141.
4. Thakar, A.R., Szalay, A.S., Kunszt, P.K., Gray, J. 2003, "The Sloan Digital Sky Survey Science Archive: Migrating A Multiterabyte Archive from Object to Relational Databases", Computing in Science and Engineering, vol. 5, no. 5 (Sep/Oct 2003), 16.
5. O'Mullane, W.; Budavári, T.; Li, N.; Malik, T.; Nieto-Santisteban, M. A.; Szalay, A. S.; Thakar, A. R.. 2005, "OpenSkyQuery and OpenSkyNode - the VO Framework to Federate Astronomy Archives", Astronomical Data Analysis Software and Systems XIV ASP Conference Series, Vol. 347, Proceedings of the Conference held 24-27 October, 2004 in Pasadena, California, USA. Edited by P. Shopbell, M. Britton, and R. Ebert. San Francisco: Astronomical Society of the Pacific, 2005., p.341.
6. Thakar, A.R. 2008: "Lessons Learned From The SDSS Catalog Archive Server", Computing in Science and Engineering, 10, 6 (Nov/Dec 2008), 65.
7. Singh, V.; Gray, J.; Thakar, A.; Szalay, A. S.; Raddick, J.; Boroski, W.; Lebedeva, S.; Yanny, B. 2007, "SkyServer Traffic Report - The First Five Years," Microsoft Technical Report, MSR-TR-2006-190.
8. Szalay, A.S., Thakar, A.R., and Gray, J. 2008: "The sqlLoader Data Loading Pipeline", Computing in Science and Engineering, 10, 1 (Jan/Feb 2008), 38.
9. Szalay, A. S.; Gray, J.; Thakar, A. R.; Kunszt, P. Z.; Malik, T.; Raddick, J.; Stoughton, C.; vandenBerg, J. 2001, "The SDSS SkyServer: Public Access to the Sloan Digital Sky Server Data", Microsoft Research Technical Report MSR-TR-2001-104, Nov 2001.
10. Li, N. and Thakar, A.R. 2008: "CasJobs and MyDB: A Batch Query Workbench", Computing in Science and Engineering, 10, 1 (Jan/Feb 2008), 18.