From SkyServer to SciServer

Alexander S. Szalay

Institute for Data Intensive Engineering and Science

The Johns Hopkins University, Baltimore, MD 21218, USA

We set out 20 years ago with Jim Gray to build the archive for the Sloan Digital Sky Survey. The project aimed to collect a statistically complete data set over a large fraction of the sky and turn it into an open data resource for the world's astronomy community. There were few examples to learn from and we had to invent much of the system ourselves. Over the years the project has changed astronomy. Now we are faced with the problem of how we can ensure that the data will be preserved and kept in active use for another 20 years. In this process we reengineered the system to be able to serve a much broader set of communities. We discuss the lessons learned, and how the growing sophistication of our users challenged and motivated us to incorporate more of the patterns of data analytics required by contemporary science.

1. Introduction

The unprecedented amounts of observational, experimental and simulation data are transforming the nature of scientific research. As more and more data sets are becoming public, we need to find the right ways not just to make the data public, but accessible and usable. Yet, our techniques have not kept up with this evolution. Even simple things like moving large amounts of data to the computing are becoming difficult; as a result, scientists are learning how to "*move the analysis to the data*". In this digital world we have to rethink not only the "*Data Lifecycle*" – as most data sets are accessible via services, we also need to consider the "*Service Lifecycle*" as well.

The data from **Sloan Digital Sky Survey** (SDSS) has been one of the first examples of a large open scientific data set. It has been in the public domain for more than a decade (Szalay 2000). It is fair to say that the project and its archive have changed astronomy forever. It has shown that a whole community is willing to change its traditional approach, if the data is of high quality and presented in an intuitive fashion. The continuous evolution and curation of the system over the years has been an intense effort, and has given us a unique perspective of the challenges involved in operating open archival systems over a decade.

The SDSS is special – it has been one of the first major open eScience archive. Tracking its evolution helps the whole science community to understand the long term curation of such data, and see what common lessons emerged for other disciplines facing similar challenges. These new archives not only serve flat files and simple digital objects, but they also present complex services. The toolkits change, the service standards evolve, and even though some services may have been cutting edge ten years ago, today they may be dated. In order to support the increasingly sophisticated client-side environments, the services need active curation at regular intervals.

Scientists in many disciplines would like to compare the results of their experiments to data emerging from numerical simulations based on first principles. This requires not only that we are able to run sophisticated simulations and models, but that the results of these simulations are also available publicly, through an easy-to-use portal. We have to turn the simulations into open numerical laboratories where anyone can perform their own experiments. Integrating and comparing experiments to simulations is another non-trivial data management challenge. Not every data set from these simulations has the same lifecycle. Some results are just transient and need to be stored for a short while to analyze, while others will become community references, with a useful lifetime of a decade or more.

In many areas, like environmental science, another challenge is the enormous complexity of the data sets involved. Various physical scales interact in a complex fashion; we have physical, chemical, biological factors all contributing to the observed phenomena. Much of this data resides in small files, such as the spreadsheets and tables collected in laboratories, in contrast to the large data collections like SDSS. These form the "*Long Tail*" of scientific data. Often, scientists would like to cross-correlate the data in these small objects with each other as well as with the large on-line databases.

The progression through these challenges form the story of this paper. We strongly feel that the framework developed for astronomy, and used for over a decade by the SDSS, captures the way we approach science extremely well, and our tools form a set of *generic building blocks* out of which many new applications can be built. In the sections below we will describe these components, how they fit together, and also how they can be generalized to solve a variety of problems.

Our data and services span a wide range in terms of their age: the SDSS data is quite far along in its lifecycle, after 15 years it faces a set of different curation issues than services that have been in operation for 5 years (turbulence, cosmological simulations, sensors), and yet different from new applications just being built. In the sections below we will discuss both the history of our ongoing efforts, describe our goals and the objectives and methods applied to the problems.

Furthermore, SDSS is now the first among the large scale eScience projects that is now approaching the point when its instruments will be turned off, but the data will still be used, possibly for decades to come. We find ourselves again at a place where we have to invent the best solution that serves the long-term needs of our user base.

2. The Origins

2.1 The Sloan Digital Sky Survey

The SDSS was one of the first large scale digital survey of the sky. The goal was to perform a high-resolution five color imaging of the Northern sky, and based upon our own images, collect spectra of the brightest one million galaxies and a few hundred thousand stars and quasars. All data was going to be open and public, after a six-month proprietary period.

The project was started in 1992, and expected to finish in 2000. The total budget was going to be $25M, about $10M on the telescope, $10M on the instruments, $3M on the mountain operations, and $2M on the software. The first light was in 2001, and we finally completed the survey as originally proposed by 2008. The final cost was over $100M, with close to a third of this spent on the software development, data processing and data management. The original projections for the data in the mid-90s were that we will collect about 10TB of raw imaging data, process it as it arrives, with maybe on additional full reprocessing towards the end. We projected the total volume of the database to be about 0.5TB, which was quite a big number in 1992. The final tally of all the low-level data that needs to be preserved for long-term use is about 150TB, the current size of the main database is 15TB, with an additional 20TB of supplementary databases, like the time-domain data from the 'Stripe-82'. This was all made possible with the eventual delays in the survey, where Moore's Law helped us to reprocess the data much easier as we understood various systematic errors better, and Kryder's Law enabled us to store much more of the data as it was collected and processed. One of the lessons learned for future surveys was the fact that in projects like the SDSS the *capital investment is in the software and the computational hardware became disposable.*

The SDSS was amazingly successful. By now there are more than 7,000 refereed papers published, with well over 350,000 citations. Much more papers were from outside the collaboration than by the survey participants. We have outlined many of the possible science projects enabled by the survey data in our 'Black Book', but the published papers have exceeded our imagination.

Initially, there was a lot of distrust in the community whether SDSS will truly release their data as promised. It took several years to convince the astronomers that we stood by our promises – we have never missed a data release. In the end, most of the proprietary periods were close to zero, we were almost always getting the data at the same time as the public. In the beginning, people did not believe that we were able to process the data well enough in an automated environment. Much of the astronomy software at the time still required an astronomer in the loop. This was unacceptable for a uniform processing for a large fraction of the sky, and represented one of the biggest unforeseen challenges for the project. Creating such an automated pipeline required much more code to be written than previously envisaged.

There was very little precedent that we could rely upon, and we had to make things up as we went along, rather quickly, as the data went on-line and usage grew very fast. We had to figure out how to deal with reproducibility. New data was added every few nights. We decided to adopt a model where the public data releases happened once a year, and once released, they never changed. This meant that a query submitted to a particular data release, say DR3, will always return the same result. While DR4 and later contain all of DR3, and more, each of the data releases were treated as a separate editions of a book. When a new 'edition' was released, we have not taken the old ones off the shelf. This enabled students, who started their thesis with a particular data release, to remain consistent and papers published on an earlier data release can be fully reproduced by re-running the queries on the original version of the database. Today we are at DR13 and counting. All data releases are still up and available at JHU.

The data releases presented an interesting challenge. Data comes in at an approximately linear rate, assuming no significant changes are made to the detectors. This means that from year 1 to year 2 the data is doubling, but after that the relative change is much smaller. The total amount of data we have to store is approximately quadratic: $1+2+\ldots N = N(N-1)/2$. Of course we always store several copies of the most current data: right now we are serving 6 different instances of DR13. We can relax this somewhat for the older, less used versions. As the price of storage is constantly dropping, with larger and larger disks, we found that the second year was the hardest to accommodate from the perspective of hardware expenditures. The evolution of storage technologies has far outpaced the rate of 20 year's linear-rate data collection!

2.2 Jim Gray and the SDSS Archive

Alex Szalay and Ani Thakar, in collaboration with Jim Gray (Microsoft) have spent much of the last two decades working on the archive for the Sloan Digital Sky Survey. The archive was originally built on top of an object-oriented database, but after a few years it became clear that the users will want to have a very flexible query environment with a lot of ad-hoc queries. The promised flexible Object Query Language (OQL) was very slow to emerge and turned out to be very fragile. As a result, we have started to develop our own (very limited) query language.

It was around this time, when we met Jim Gray of Microsoft. Jim liked to say, that the "best collaborators are the desperate ones", as they are ready to change the way they approach a problem. We were definitely desperate at this point. After a few meetings Jim advocated that we should consider using a relational database. He made the point that a few programmers in an academic environment cannot successfully compete with the thousands of developers at Microsoft, Oracle and IBM, and we should spend our efforts on creating the additional 'business logic' related to astronomy, and use a commercial platform with a robust SQL query engine. This advice set us on the trajectory that we have followed ever since.

Another major principle behind our approach also came from Jim Gray. When we first met, he asked me to give the "20 queries" astronomers wanted to ask from the database. My first kneejerk reaction was that we would like to ask anything and everything, and this is what science is about. Jim just smiled, and asked me to give the first five that came to mind. I quickly wrote the down. He then asked me to give the next five. It took the better part of an hour to do so. By then I realized that the next five will take much longer, and understood that not every question is the same. In an hour this methodology has taught me a lot of humility, and I learned that there is a priority among all the things that we can conceivably think of. We have since used this heuristic technique in a lot of different projects and settings. The results are always the same: after an initial astonishment the domain scientists very quickly 'get the idea' of establishing clear priorities, and the database architect and domain scientists quickly find the common ground.

The project has revolutionized not just professional astronomy but also the public's access to it. Although a substantial portion of the astronomy community is using the SDSS archive on a daily basis, the archive has also attracted a wide range of users from the public (Singh 2006): a scan of the logs shoed more than 4M distinct IP addresses accessing the site. The total number of professional astronomers worldwide is only about 15K. Furthermore, the collaborative CasJobs interface has more than 8,000 registered users – almost half of the professional astronomy community.

SDSS (2000-2005) and its successors SDSS-II (2005-2008) and SDSS-III (2008-2014) have an extraordinary legacy of mapping structure across a vast range of scales, from asteroids in our own Solar System to quasars over 10 billion light-years away. These surveys have produced data to support the 5,000 papers with more than 200,000 citations. The SDSS has several times been named the highest impact project, facility or mission in the field of astronomy, as judged by number of citations of associated refereed journal articles (Banks 2009, Madrid 2009). The SDSS was the source of the most highly cited astronomy article in the years 2000, 2002, 2005, and 2008 (Frogel 2010). Within the Collaboration there have been over 120 SDSS-based PhD theses, and outside the Collaboration there have been many more. Its publicly available, user-friendly tools have fueled a large number of undergraduate and even high-school projects and research papers.

In 2007 we have played a key role in launching the Galaxy Zoo citizen science project (Lintott 2008) in which online volunteers - members of the public, most of which had no prior experience with scientific research - are asked to visually classify SDSS images of nearly a million galaxies. Today, Galaxy Zoo has more than 200,000 volunteers, who have collectively classified each of the million galaxies between 9 and 50,000 times. Galaxy Zoo has been featured by many of the world's best-known and respected news organizations (BBC, the *New York Times*, *Nature*, etc.), showing how active scientific research can attract a large and involved non-expert population. But Galaxy Zoo users have contributed more than just raw image classifications. One of the most unexpected and successful parts of the project was the way in which citizen scientists used SkyServer tools to learn more about the galaxies they were asked to classify. In two cases, these efforts by citizen scientists led to published original research in astronomy journals (Lintott 2009, Cardamone 2009).

The 2.5-meter Sloan telescope in Apache Point, NM, remains the most powerful wide-field spectroscopic survey facility in the world today. To capitalize on this resource, a collaboration of 186 astronomers and physicists from 65 institutions have organized the SDSS-IV program, conducting a broad survey of our Milky Way Galaxy, the population of nearby galaxies in the local universe and the large-scale structure of the universe as a whole. SDSS-IV will operate from July 2014 to July 2020. It will marshal imaging data from multiple telescopes and wavelength regimes to identify targets for follow-up spectroscopy.

3. Evolution During the Early Years

3.1 The SkyServer Usage Log Database

One of the most useful byproducts of the SDSS data has been the usage logs that we have kept since the very beginning of the project: every Web hit and every single query have been logged since the archive was opened. The log database today is over 3TB, and contains rich historical information about how astronomers learned to access a virtual telescope (Singh 2006). This has resulted in an amazingly rich and useful resource not only for SDSS scientists and project managers, but since the dataset is available to anyone, many other projects and researchers have found it extremely valuable. Next generation large astronomy surveys like Pan-STARRS (Heasley 2007) and LSST (Becla 2006) have used this data to plan their data management infrastructure and services, and several other groups in astronomy and computer science have downloaded the entire dataset for analysis. The SDSS log data was the subject of a Ph.D. dissertation at Drexel University in Human-Computer Interaction research (Zhang 2011). We receive on average one or two requests per month to download the SDSS log data, especially the SQL query logs since this is perhaps the only such large dataset of SQL usage in existence. The SDSS has several mirrors over the world (UK, Brazil, India, China, Hungary). Their logs are harvested every night and aggregated into our main log database. On the main SDSS SkyServer web page there is a link to some cumulative counts from the log DB. The most current values are 2.4B web hits, and 364M free-form SQL queries.

3.2 Parallel Loader Environment

The raw data is transformed into a common loadable format by a set of specific plug-ins. The data comes in blocks, typically a few tens of GB at the time. Each block is transformed into a set of files in a single directory tree, with checksum files stored in each directory. For larger data sets this is a brutally data-parallel operation. The results of the data transformation process are picked up by the Parallel Loader (Szalay 2008). The Loader scales to an arbitrary number of machines. It performs a two-phase load. First, for each block, we create an empty database with the same schema as the main system, and load the whole block. During the load process, broken into tasks, steps and phases, we generate a detailed log at each granularity. The state of all jobs can be tracked visually using the Load Monitor interface.

Some of the tasks in the workflow described by a DAG (Directed Acyclic Graph) perform a very detailed integrity checking and data scrubbing, looking for out-of-band data. The data rows in each block get tagged by a load-ID unique to the block, so that combining this with the loader logs allows us to track each row's provenance. The two-phase load has proven to be invaluable, as data errors were caught well before the bad data could have contaminated the main database. It turns out that the load performance of a typical database server, running on a good file system, is not I/O- but rather CPU-limited, due to the various page formatting and checksum calculations. We found that on a high-end SQL Server machine, using an array of SSDs and 32 cores, we were able to achieve load speeds in excess of 1GBytes per second using thread parallelism. Once data is in a DB page format, copying the DB files to other machines is only limited by the hardware performance.

3.3 The Web Interface

Early on we have decided to move away from the solely form-based interfaces to the archive. We decided to have a highly visual interactive front end, based on the available browsers at the time. This era is still remembered as “browser hell”, as the existing web browsers had rather incompatible functionalities and commands. Internet Explorer and Netscape were still not capable of stylesheets, their javascript functions were rather different. In the end we have implemented the website using our own abstract API for rendering various items, which were mapped onto their native implementations when a web page was loaded. In the end, this approach saved us a lot of headaches. Some of the functions are still there, but we are in the process of gradually replacing them with HTML5 canvas, and other more modern components. Based upon Jim Gray’s Terraserver experience, and the emerging MapQuest, we created a clickable map of the sky, with image mosaics built server side from precomputed false color JPEG images. The interface provided ways to overlay markers corresponding to different object types detected by the survey, also show pixel boundaries and bounding boxes of each object, and show some of the basic geometric elements of the survey (stripes, plates, fields).

3.4 Free-form SQL Queries

After about a year we visited the National Center for Biological Information (NCBI) for a few days. During this visit, David Lipman, the NCBI director has shown us an article arguing against form-based interfaces to biological databases. The author felt that these interfaces, designed by a programmer and not by a scientist at the cutting edge, restrict the patterns how data can be explored, thus limit the scope of science possible. He suggested that there should be a back-door, enabling ‘anything and everything goes’ type creative queries.

We decided to open the database for free-form SQL queries. Many people cautioned us against this, arguing that (a) no astronomer will want to write SQL queries (b) we will be constantly hit with denial of service attacks. We did it anyway and much to our amazement we found that neither of those prediction came true: astronomers embraced SQL remarkably fast, and there were no major abuses of the interface.

In order to help astronomers to learn SQL, we have posted the 20 queries that came out of the early discussions. We have first displayed the original question or problem definition written in plain English, then shown the SQL implementation that executed the query. Finally, in about a page or so we explained why the query was written the way it was shown. This enabled the astronomers to look for a query that was close enough to what they wanted to do, first do a cut and paste, run it, and start modifying it step-by-step, until they arrived at their results. Over the years we have added another 15 query patterns to the pool. We have also built a step-by-step tutorial to teach the basics of the SQL language.

3.5 The SDSS Web Services

The web interface is built on a very small number of atomic services. One simply executes a single SQL query. This service was very carefully written, it parses the body of the query looking for ‘bad’ patterns, like injection queries, DROP TABLE, or ALTER TABLE like DDL statements. This function also heavily interacts with the log, it writes the SQL string into the database when the query is started, then logs the lapse and CPU times, and the number of rows delivered, or the error code if any.

The other major core service is the ImageCutout (Thakar 2008). This is computing the displayed views of the sky dynamically, using a multiresolution set of the more than a million SDSS images. This code has been written is C#, and uses the Microsoft .NET GDI library for the warping of images. The transformation from screen coordinates to world coordinates is done by storing the transformation matrix for each of the 2048x1489 pixel fields separately, so drawing can be done either of the original SDSS pixel coordinates, mapped onto the screen, or standard equatorial coordinates used by astronomers.

Given the enormous dynamic range in the brightness of astronomical objects, we used a special transformation of the pixel values. After the images were mapped into RGB space, the dynamic range was still too high to be properly represented on an 3x8-bit graphics display. We have computed the luminescence, obtained by the combination of the three bands, and compressed its range through a simple mapping, linear at fluxes close to zero (and the sky noise) and logarithmic for the bright objects (Lupton 1999). This preserved the colors across the objects, while displaying all the details of both bright and faint objects.

4. Maturity and Production

4.1 The CasJobs/MyDB Collaborative Environment

As traffic on the SDSS archive grew, many users were running repeated queries extracting a few million rows of data. The DB server delivered such data sets in 10 sec, but it took several minutes to transmit the data through the slow wide-area networks. We realized that if users had their own databases at the server, then the query outputs could go through a high-speed connection, directly into their local databases. This improved system throughput by a factor of 10. Furthermore, we have built an asynchronous (batch) mode that enabled queries to be queued for execution and results to be retrieved later at will.

The CasJobs/MyDB batch query workbench environment was born as a result of combining these "take the analysis to the data" and asynchronous query execution concepts. The name "CasJobs" comes from "CAS (Catalog Archive Server)" and (batch) "jobs". CasJobs builds a flexible shell on top of the large SDSS database. Users are able to conduct sophisticated database operations in their own space: create new tables, perform joins with the main DB, write their own functions, upload their own tables, and extract their value-added data sets to their home environment, to be used with the familiar tools they have been using since their graduate years. The system became an overnight success.

For redundancy, we had three identical servers containing the active databases. By studying the usage patterns, we realized that the query length distribution was well represented by a power law. Hence, we split the traffic into multiple queues served by different servers, each handling the same aggregate workload (O'Mullane 2004). Each query can be submitted to a "fast," "medium:" and "long" queue, returning the result into a MyDB table. The user can then process the derived result further, run a multi-step workflow, or extract the data. Everything that a user does is logged. This set of user-controlled databases form a very flexible tier on top of the rigid schema of the archive. This resolves the long-standing tension between stability and integrity of the core data and the flexibility for user creativity.

As users became familiar with the system, there were requests for data sharing. As a result, we added the ability to create groups and to make individual tables accessible to certain groups. This led to a natural self-organization, as groups working on a collaborative research project used this environment to explore and build their final, value-added data for eventual publication. GalaxyZoo, which classified over a million SDSS galaxies though a user community of 300,000, used CasJobs to make the final results world-visible, and CasJobs also became a de-facto platform for publishing data. We added the capability for users to upload their own datasets and import them into their MyDBs for correlation with the SDSS.

4.2 SQL Extensions

Over the years we have developed a Design Pattern to add domain specific extensions to SQL Server, using CLI integration. Our code for spatial indexing was used in the "shrink-wrap" production version of SQL Server 2005 (Fekete 2006, Budavari 2010). The idea is to take a class library written in one of the .NET languages (C++, Java,C#), store a binary instance of the class as a binary datatype, and expose the object methods as user-defined functions (UDFs). SQL Server makes this very convenient, since unlike many other database platforms like MySQL, it allows for table-valued UDFs. One can then pass the binary object as a parameter to the function and execute the method, or access the property.

We have 236 UDFs supporting detailed astronomy knowledge, like conversion of cosmological coordinates in a curved space to angles and radial distances. Also, we have built an astronomy-specific spatial index, representing spherical polygons with milliarcsec accuracy over the whole sky, with a relational algebra over the regions, and fast indexing capabilities finding several million points per spherical region in a second.

For large numerical simulations much of the data is in multidimensional floating point arrays. We have built such a User Defined Type for SQL Server, which is used for all of our simulation databases (Dobos 2011). We will develop a generic module that repartitions the data in a large array into smaller blocks organized along a space-filling curve, adds the custom metadata header and writes these out in native binary format for optimal SQL Server load performance.

4.3 Schema and Metadata Framework

The schema for the database is contained in a set of DDL files. These files are quite complex, they not only contain the code to generate the database and the associated stored procedures and user defined functions, but in the comment fields of the scripts they contain rich metadata describing the schema elements, including physical units, enumerations, indexes, primary keys, short and long descriptions. A parser can extract this information at different granularities (object, column) and create a set of metadata tables that can be automatically loaded into the database. This ensures that all the schema and related metadata is handled together in an automated fashion, similar to the approach originally employed by Donald Knuth, when he created TeX. The database will then contain all the up-to-date metadata and these can be queried and displayed using simple functions and dynamic web services. This tool is quite robust and mature and has been in use for more than 14 years.

5. Branching Out to Other Disciplines

5.1 The SkyServer Genealogy

The template for the SDSS archive is now being used within astronomy by several projects and institutions beyond JHU (STScI, Fermilab, Caltech, Edinburgh, Hawaii, Portsmouth, and Budapest). The technologies and concepts used for the SDSS archive have since been used beyond astronomy. Using the same template, we have built databases for a growing number of other disciplines. Included are databases for turbulence (Li 2008), radiation oncology (McNutt 2008), environmental sensing and carbon cycle monitoring (Szlavecz 2006) and most recently a prototype for high-throughput genomics (Wilton 2015). The databases built for cosmological simulations are revolutionizing how astronomers interact with the largest simulations.

5.2 Open Numerical Laboratories

World-wide there is an ongoing effort to build an exascale computer. At the same time, fewer and fewer codes will scale to millions of cores, and as a result, fewer people will use these ever larger machines. There will be an increasing gap between the wide science community and the top users. It will be increasingly important to create science products that can be used by a much wider pool of users, otherwise community support will be soon endangered. There is already an increasing demand from the broader science community to access the largest numerical simulations. While only our largest supercomputers are capable of creating such simulations, their analysis, especially if the data will be publicly accessible, requires a different type of architecture.

To date, the usual way of analyzing somebody else's simulation is to download the data. With PB scale data sets, this is obviously not going to work. We are experimenting with a new, immersive metaphor for interacting with large simulations by using a large number of virtual sensors that can be placed in a simulation, anywhere at any timestep. They can also be set to send a data stream in real physical quantities. Imagine how scientists could launch mini accelerometers into simulated tornadoes, emulating the movie "*Twister!*" We have successfully implemented this metaphor for our turbulence data, and are now porting it to the cosmology simulations.

In this approach one can create a so-called *immersive environment* in which the users can insert virtual sensors into the simulation data. These sensors can then feed data back to the user. They can provide a one-time measurement, they can be pinned to a physical (Eulerian) location or they can "go with the flow" as comoving Lagrangian particles. By placing the sensors in different geometric configurations, users can accommodate a wide variety of spatial and temporal access patterns. The sensors can feed back data on multiple channels, measuring different fields in the simulation.

This pattern also enables the users to run time backwards, impossible in a direct simulation involving dissipation. Imagine that the snapshots are saved frequently enough that one can interpolate particle velocities smoothly enough. Sensors can back-track their original trajectory and one can see where they came from, all the way back to the initial conditions. This simple interface can provide a very flexible, yet powerful way to do science with large data sets from anywhere in the world. The availability of such a 4D dataset "at your fingertips" and the ability to make "casual" queries from anywhere is beginning to change how we think about the data. Researchers can come back to the same place in space and time and be sure to encounter the same values.

The "Twister" metaphor mentioned above, has been implemented in the Turbulence DB eight years ago. The Turbulence DB is the first space-time database for turbulent flows, containing the output of large simulations, publicly available to the research community (Perlman 2008). The 27TB database contains the entire time-history of a 1024^3 mesh point pseudo-spectral Direct Numerical Simulation of forced Navier-Stokes equations representing isotropic turbulence. 1024 time-steps are stored, covering a full "large-eddy" turnover time of model evolution. We have built a prototype web service that serves requests over the web for velocities, pressure, various space derivatives of velocity and pressure, and interpolation functions. The data and its interface are used by the turbulence research community and have led to about 100 publications to date. To date we have delivered over 36 trillion (!) data points to the user community. In a recent paper on MHD, trajectories were computed by moving the particles backward in time, impossible to do in an in-situ computation, only enabled by interpolation over the database (Eyink 2013).

A similar transformation is happening in cosmology. The SDSS SkyServer framework was reused for the Millennium simulation database (Lemson 2006). The database has been in use for over 10 years, has hundreds of regular users, and has been used in nearly seven hundred publications. The database contains value added data from a simulation originally only containing 10B dark matter particles. A semi-analytical recipe was used to create mock galaxies in the simulations, and their hierarchical mergers were tracked in the database. The merger history was used to assign a plausible star formation rate to each galaxy which in turn can be used to derive observable physical properties. The database contains several such semi-analytic scenarios and has been expanded with data from three other simulations, one of which is containing 300 billion particles.

5.3 Environmental Science

Environmental data are complex, combining biological, physical, and geological measurements, heterogeneous in space and time. The data is fragmented, as various scientists focus on specific variables and store data in isolated file systems, integration becomes a significant challenge. A great deal of effort has been spent to make environmental data more accessible. A common feature of these networks is that they have largely focused on data accessibility through metadata catalogs where investigators can search data by key word, project name, investigator name, etc.

Our pilot system focused on integrating data on various spatial and temporal scales to answer science questions related to the soil ecosystem. LifeUnderYourFeet (Szlavecz 2006) has been continuously collecting soil moisture and temperature data since 2008, and soil respiration data since 2010. We used the SciServer framework to integrate data at national and local spatial scales and to correlate soil measurements in space and time for various climatic, atmospheric, meteorological, and anthropogenic conditions and scales.

6. Towards a Sustainable Solution: The SciServer

6.1 Consolidating the Evolution

Over the first 12 years of the SDSS archive we have incrementally evolved the system, avoiding major architectural changes. The SDSS data with all the additional science projects have been created at a cost well over $100M. They are widely used by a diverse community, and are generating new papers and supporting original research every day.

Now, as we need to look ahead into the future, the services are showing signs of aging; while the data are still very much alive, they will still be used in 15 years from now. In order to prepare for the future, we need to consolidate and reengineer the services, with the main goal of making them more sustainable and inexpensive to operate. In order to do this, we have endeavored on converting the SkyServer to the SciServer, a more generic, modular set of building blocks that can be connected in several ways.

6.2 New Building Blocks

FileDB

Relational databases have shown their value to the scientific community. The SDSS Database (Thakar 2008) was a forerunner showing how the community was willing to take the step of learning SQL to access the database. However, data volumes are reaching the limits of what can be managed within relational databases with reasonable effort, it takes a week to load a typical Turbulence database. To avoid this bottleneck, we built a system that allows linking to raw data *from* the database, using indexes, without ingesting it *into* the database. We wrote custom functions that can access the file system, but can be called from ordinary SQL. These functions are exposed as table valued, user defined functions and are accessible through standard SQL queries. Their performance is as good as native DB calls.

ScratchDB

We have enabled the CasJobs system to have many other contexts, not just the SDSS data versions (right now we have all the previous data releases from DR1 through DR9), but also other astronomical collections. We will also bring the simulations and environmental data sets into the federation. Uploaded and derived data (and the related metadata) will automatically show up in the user's MyDB. For large scale intermediate data the few GB of user space is not quite enough. For example, a custom cross-match of large astronomical catalogs, like SDSS and GALEX might require several 100 GB if not TBs of disk space. This cannot be done today. We resolve this problem by a new *MyScratch* layer between the static contexts and the MyDBs, with tens of TB of storage, both in flat files and as large database larges.

Advanced Scripting

Our users, both in SDSS and in the Numerical Laboratories have become quite artful in using database queries. They use SQL tools not as a hammer, but rather as a violin, and they generate "nice music". But, with the emergence of Python, lot of sophisticated machine learning algorithms, libraries and packages have become available, and the users are now keen to use these with same ease of interactivity as SQL. A typical use case would start with a SQL query returning tens of thousands of objects with a particular spectral property. However, the user would then like to go back to the raw data (spectra in this case) and run her own tools and algorithm written in Python.

In order to facilitate this we built two add-ons: one is SciServer Compute, a set of servers providing about 100 virtual machines, always available, that can be used to start Jupyter/iPython notebooks, within Docker containers. These are preconfigured with the database interface tools, that users can run their SQL queries out of Python. Furthermore, all the raw data files of SDSS (about 150TB) are wrapped into a data container, so access is trivial. The Jupyter environment also enables Matlab and R, which are more relevant for our engineering and Biostats/genomics users. Several of our interactive Numerical Laboratories (Turbulence, Ocean Circulation, N-body) are now using both Python and Matlab bindings.

7. SDSS Futures

7.1 Consolidation of the SDSS Versions

We aim to integrate the SDSS-IV results with the legacy data from SDSS-I, SDSS-II and SDSS-III, including a large (14,555 square degree) imaging survey of the sky with follow-up optical and near-infrared spectroscopy. Currently, because the SDSS-III project proceeded under a different organizational structure than SDSS-II, the SDSS products have branched into two distribution sites. For SDSS-IV, we plan to reintegrate this distribution under a single archive that includes all of the legacy data and documentation, as well as the new data, integrated under the reengineered and enhanced version of SciServer. The proven flexibility and extensibility of the Sky/SciServer framework makes it possible to integrate this new data in a coherent and scientifically powerful fashion. The total data volume of the survey, combined with the legacy data, by the end of SDSS-IV is projected to be around 400 TB, with the final reduced catalogs around 15 to 20 TB. In addition, these final reduced catalogs will consist of several different flavors of data – optical and near-infrared spectra, single-fiber and fiber-bundle data, and several different types of target imaging data (optical, near- and mid-infrared data). Finally, the combination of imaging and spectroscopic coverage maps will form a complex pattern on the sky that will need to be described quantitatively for science, and that the spatial tools of the SkyServer have been designed to track.

7.2 The Data Lifecycle

We often talk about the Data Lifecycle, and its phases. As the SDSS project is probably nearing its data acquisition, we have think carefully about the long-term sustainability of the data archive, and how it will be curated and preserved. Given that its usage shows no signs of decrease we need to consider that the data will support good science for another 20 years. How can we support such a long lifecycle, where does the support come from, and where will the data reside? It is time to start to thinking about what happens to the data after the sunset of the observations.

We can see three distinct phases. In Phase 1 observations are still going on. So long as the SDSS telescope is still taking data, the archive is part of an active data collection effort, thus the ongoing project funds most of the related expenses. Phase 2 starts once the telescope is shut down. It is clear that the archive needs to be kept alive, but the data does not grow any longer. Over a five year period during this phase we need to consolidate the services as much as possible. This must be done by the team currently operating the archive. During Phase 3, the following 5 years the archive must be handed off to an organization which can operate on a good economy of scale, and whose sustained existence is guaranteed, independent of the individual data sets. One of the possibilities we are considering is to identify a set of University Libraries, which are willing to undertake this task of maintaining the archive and operate a help desk. This phase should continue as long as there is a continued use of the archive and one can justify its existence based upon scientific value generated.

7.3 The Service Lifecycle

However, during the 20 years we have been working on the SDSS Archive and now the SciServer, we have learned about the Service Lifecycle as well. The SDSS archive and now the SciServer is much more than just a simple file-store. The data are served through a set of sophisticated, smart services, which offer a lot of server-side functionality. In 2001 we have built the first web services deployed in a science setting, but by now many of these APIs and interfaces have become obsolete. Computing has undergone several major paradigm shifts. We went from CORBA to the GRID, to Web Services, Grid Services, then the Cloud, and most recently to Data Lakes. No matter what, this dynamic evolution is going to continue, and it is difficult to predict what the world of distributed computing will look like even in 5 years from now.

There is also natural aging. Technology has improved significantly since we built our first services (the first web services in science were built for SDSS by Jim Gray and Alex Szalay). While several improvements have been implemented over time, it is important now to rethink the methodologies in the context of the new Internet. Smarter client-side web interfaces are possible today using HTML5 and JavaScript, which are standard and quickly becoming widely accepted. This will enable our new infrastructure to perform some of the processing steps in the browsers rather than overloading the servers. Smarter clients will work efficiently with resource oriented services. By now, REST has replaced SOAP almost everywhere. Asynchronous messaging protocols will make the infrastructure more robust against the glitches in the communications. Behind the web server we will build a universal application layer that uses proper scheduling mechanisms to handle the large volume of complex user jobs. Load balancing will be realized on all levels by partitioning and parallel execution of the tasks over a cluster of database servers.

From queries to file extractions, everything will be prioritized and executed in the most efficient way by schedulers that keep track of data locality and use the closest copies in the distributed database system. The next generation execution environment will be based on workflows, whose state can be persisted in a database. Thus long-running and expensive scientific analyses can be suspended and resumed, making the framework more resilient and the system management much easier.

All that we can do today is to prepare for these changes to come, and reorganize the underlying services and APIs in such a way, that they are maximally modular and independent, so that future overhauls will be as painless as possible. BY building the database schema to be maximally portable enabled us to move from Versant to Objectivity, then to SQL Server. In the SciServer we are extending this philosophy, and we have further modularized the whole environment, and incorporated design patterns going beyond astronomy. The fact that it was very easy to bring new science use cases into the SciServer validated our approach.

7.4 Community Response

We started from astronomy, building the SDSS archive and then various tools for the Virtual Observatory. In a few years these datasets have earned the trust of the community and started to be heavily used. Starting with the Turbulence project, we have introduced the notion of interactive, database-centric tools into other domains. The initial reaction from the turbulence community by those researchers that themselves do very large simulations was rather skeptical: they felt that they could analyze their own data more effectively than through our database approach. However, the rest of the community could not do so effectively or not at all.

Thus, many researchers started to access the data in our system and do their research in the open numerical laboratory. For instance, experimentalists could place tracer particles as measurement devices inside the numerical space-time data in the numerical laboratory and thus calibrate their measurement techniques. Mathematicians could find seeds of possible singularities in the partial differential equations. These scientists represent a cross-section of the research community that had real difficulties accessing large datasets from simulations prior to the JHTDB. The availability of our open numerical laboratory has led to many results and papers by researchers all over the world, having been used for over 100 published papers on turbulence - (this is in fact a significant number, noting that turbulence is a relatively small field, much smaller than astrophysics or biology - e.g. the total number of papers published in the Journal of Turbulence is around 70 per year). In 2015 the number of points has exceeded 12 trillion, and a few days ago it has reached 39 trillion.

A similar transformation is happening in cosmology. The SDSS SkyServer framework was reused for the Millennium simulation database (Lemson 2006). The database has been in use for over 8 years, and has hundreds of regular users, and has been used in nearly seven hundred publications. The set-oriented SQL query language makes it remarkably easy to formulate very complex aggregate queries over the temporal history of various subsets of galaxies and create samples that can be compared directly to observations. It is clear that there is a similar momentum building in the cosmology community as in turbulence.

8. Conclusions, Lessons Learned

What we see is that in any new community we engage with, it takes about 3-5 years to overcome the initial skepticism of the pundits, and demonstrate that our interactive approach to Petascale problems is more scalable than the traditional ones. Similar to SDSS, ***we have to earn the TRUST of the community the hard way*** – by giving them open access to high quality data and easy to use tools that mesh well with how they analyze their data.

It is also clear, that none of the domain communities understand the subtle differences between the value of data, and the cost of data, and in particular, the cost of archiving. The value of data is relatively easy to grasp, we make new discoveries based upon these data sets, write new papers, share them, combine them with other data sets, and they provide a solid foundation for reproducible results.

It is much harder to define the price of data. On one hand one can argue, the price of data is what it took to build and run the instruments. Many of today's large data collections in this sense have cost hundreds of millions of dollars (SDSS), if not billions (LHC). On the other hand one can argue that a typical NSF grant of $100K/year is considered to be high quality if it produces 2 refereed papers in a high quality journal annually. By this token, the value of a paper is about $50K. Of course not all science support goes into the individual grants, at least an equal amount goes into various national facilities, both physical and computational. Let us double this number, and estimate the value of a good scientific paper to be about $100K. By this measure, the SDSS data has to date resulted in more than 5,000 refereed publications, and this has a "monetarized value" of $500M. At the same time, the total cost of all the SDSS projects has been around $200M, making it very cost efficient.

Now we need to consider the cost of archiving. Again, on one hand we can calculate t6he physical costs, power, disk drives, curation personnel, servers, etc. In astronomy, the typical operating cost of a telescope is around 5-10% of the capital cost. Everyone accepts this. At the same time, we are still shocked if the cost of maintaining an archival data set was a few hundred thousand dollars, often a small fraction of 1% of the capital cost of acquiring it. Yet, these archival data sets will generate a disproportionally high value in terms of new publications, at least for several decades ahead of us.

These large data sets, analyzed by a much broader range of scientists than ever before, using all the tools of the computer age are creating a new way to do science, one that we are just starting to grapple with. We cannot predict where it will exactly lead, but it is already clear that these technologies will bring about dramatic changes in the way we do science and make discoveries.

9. References

Banks, Michael, Impact of Sky Surveys, *Physics World*, p 10, March 2009.

Becla, J., Hanushevsky, A., Nikolaev, S., Abdulla, G., Szalay, A.S., Nieto-Santisteban, M., Thakar, A., Gray, J.: Designing a multi-petabyte database for LSST, *Proceedings of the SPIE*, Volume **6270**, pp. 62700R (2006).

Budavári, T., Szalay, A.S., Fekete, G.: Searchable Sky Coverage of Astronomical Observations: Footprints and Exposures, *Publications of the Astronomical Society of the Pacific*, **122**, 1375-1388 (2010).

Cardamone, C., Schawinski, K., Sarzi, M., Bamford, S.P., Bennert, N., Urry, C.M., Lintott, C., Keel, W.C., Parejko, J., Nichol, R.C., Thomas, D., Andreescu, D., Murray, P., Raddick, M.J., Slosar, A., Szalay, A. and Vandenberg, J., 2009, "Galaxy Zoo Green Peas: Discovery of a Class of Compact Extremely Star-forming Galaxies," *Monthly Notices of the Royal Astronomical Society*, 399, 1191–1205. doi:10.1111/j.1365-2966.2009.15383.x.

Dobos, L., Csabai, I., Milonvanovic, Budavari, T., Szalay, A.S., Tintor, M., Blakeley, J., Jovanovic, A., Tomic, D.: Array Requirements for Scientific Applications and an

Implementation for Microsoft SQL Server, EDBT/ICDT Workshop on Array Databases, Uppsala, Sweden (2011).

Eyink, G., Vishniac, E., Lalescu, C., Aluie, H., Kanov, K., Bürger, K., Burns, R., Meneveau, C., Szalay, A.S. 2013, Flux-freezing breakdown in high-conductivity magnetohydrodynamic turbulence, *Nature*, 497,466.

Fekete, G. Szalay, A.S., Gray, J.: Using Table Valued Functions in SQL Server 2005; MSDN Development Forum (2006).

Frogel, Jay A, Astronomy's Greatest Hits: The 100 Most Cited Papers in Each Year of the First Decade of the 21st Century (2000-2009), Publications of the Astronomical Society of the Pacific, Volume 122, issue 896, pp.1214-1235 (2010).

Heasley J. N., Nieto-Santisteban M., Szalay A., Thakar A.: The Pan-STARRS Object Data Manager Database, *Bulletin of the American Astronomical Society*, **38**, 124 (2007).

Lemson, G. and the Virgo Consortium, 2006, arXiv:astro-ph/0608019.

Lemson, G., Budavari, T., Szalay, A.S.: Implementing a General Spatial Indexing Library for Relational Databases of Large Numerical Simulations, Proc. SSDBMS Conference, Portland OR (2011).

Li, Y., Perlman, E., Wan, M., Yang, Y., Meneveau, C., Burns, R., Chen, S., Szalay, A.S., Eyink, G.: A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence, *Journal of Turbulence*, **9(31)**, pp. 1-29, (2008).

Lintott, C.J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M.J., Nichol, R.C., Szalay, A., Andreescu, D., Murray, P. and Vandenberg, J. (2008), "Galaxy Zoo:

morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey," *Monthly Notices of the Royal Astronomical Society*, 389, 1179–1189. doi: 10.1111/j.1365-2966.2008.13689.x.

Lintott, C. J., Schawinski, K., Keel, W., Van Arkel, H., Bennert, N., Edmondson, E., Thomas, D., Smith, D. J. B., Herbert, P. D., Jarvis, M. J., Virani, S., Andreescu, D., Bamford, S. P., Land, K., Murray, P., Nichol, R. C., Raddick, M. J., Slosar, A., Szalay, A. and Vandenberg, J. (2009), Galaxy Zoo: 'Hanny's Voorwerp', a quasar light echo?. Monthly Notices of the Royal Astronomical Society, 399: 129–140. doi: 10.1111/j.1365-2966.2009.15299.x.

Lupton, Robert H.; Gunn, James E.; Szalay, Alexander S. 1999, A Modified Magnitude System that Produces Well-Behaved Magnitudes, Colors, and Errors Even for Low Signal-to-Noise Ratio Measurements, *Astron.J.,* 118, 1406L.

Madrid, J. P. & Macchetto, F. D. 2009, High-Impact Astronomical Observatories, Bulletin of the American Astronomical Society, Vol. 41, p. 913-914.

McNutt, T. and Nabhani, T. and Szalay, A. and Deweese, T. and Wong, J., Oncospace: EScience Technology and Opportunities for Oncology, *Medical Physics*, 35, 2900 (2008).

O'Mullane, W., Gray, J., Li, N., Budavari, T., Nieto Santisteban, M., Szalay, A.S.: Batch Query System with Interactive local storage for SDSS and the VO, *Proc. ADASS XIII, ASP Conference Series, eds: F. Ochsenbein, M. Allen and D. Egret*, **314**, 372 (2004).

Perlman, E. R. Burns, Y. Li & C. Meneveau, Data exploration of turbulence simulations using a database cluster, In *Proceedings of the Supercomputing Conference* (SC'07), 2007

Singh, V., Gray, J., Thakar, A.R., Szalay, A.S., Raddick, J., Boroski, B., Lebedeva, S., Yanny, B.: SkyServer Traffic Report – The First Five Years, *Microsoft Technical Report* , MSR-TR-2006-190 (2006).

Szalay, A.S., Kunszt, P. Thakar, A., Gray, J., Slutz, D. and Brunner, R.: Designing and Mining Multi-Terabyte Astronomy Archives: The Sloan Digital Sky Survey, *Proc. SIGMOD 2000 Conference*, 451-462 (2000).

Szalay A., Thakar A. R., Gray J.: The sqlLoader Data-Loading Pipeline, *Computing in Science and Engineering*, **10**, 38 (2008).

Szlavecz, K., Terzis,A., Musăloiu-E., R., Cogan, J., Small, S., Ozer, S., Burns, R., Gray, J., Szalay, A.S.: Life Under Your Feet: An End-to-End Soil Ecology Sensor Network, Database, Web Server, and Analysis Service, *Microsoft Technical Report*, MSR-TR-2006-90 (2006), also http://lifeunderyourfeet.org/

Thakar A. R., Szalay A., Fekete G., Gray J. 2008, The Catalog Archive Server Database Management System, *Computing in Science and Engineering*, 10, 30.

Wilton, R., Budavari, T., Langmead, B., Wheelan, S. J., Salzberg, S. L., Szalay, A. S. 2015, Arioc: high-throughput read alignment with GPU-accelerated exploration of the seed-and-extend search space, *PeerJ*, 3, 808.

Zhang, J. Ph.D. Thesis, Drexel University, (2011) https://idea.library.drexel.edu/islandora/object/idea%3A3543