Building a Community-Focused Data Platform

Matthew J. Turk
NCSA
University of Illinois
Urbana, IL 61801
matthewturk@gmail.com

Michael S. Warren
LANL
Theoretical Division
Los Alamos, NM 87545
mswarren@gmail.com

Samuel W. Skillman
KIPAC
Stanford University
Stanford, CA 94305
samskillman@gmail.com

> It does not matter how awesome your product is or your presentation or your post. Your awesome thing matters *only* to the extent that it serves the user's ability to be a little more awesome. –Kathy Sierra

Abstract

The National Data Service (NDS) presents a broad-reaching vision of integrating discovery, publishing, and sharing of data. Metcalfe's Law states that the value of a network increases with the number of *connections* in the network, not the number of *nodes*. The NDS must develop and steward connections not only between data, as is in its vision statement, but between individuals as well. We propose that this be accomplished by explicitly developing it as a platform on which data exploration can be constructed by *individual researchers*.

1 Introduction

The NDS presents a unique opportunity for the scientific community: by harnessing and collecting disparate data repositories provided by member institutions, the unification of data and representation can be accomplished in a single environment. This can reduce the distance between raw data and knowledge discovery, which can only further increase the ability of multiple data sources to be simultaneously processed and correlated.

Participation and engagement from both data providers and data users will be necessary for long-term success of the NDS; this requires not only the prescriptive policies set in place by the data repositories and the NDS, but also responsiveness to the needs of researchers, and most importantly active stewardship of their participation. The largest factor in job satisfaction for technical workers is "operational autonomy" — the self-management of decisions, of actions, and self-direction. The single most powerful way to steward engagement and participation from researchers is to focus strongly on *empowering* these researchers with operational autonomy, not only to find, examine and process data on their own, but to develop and deploy tools to share with others to examine this data.

The most important thing the NDS can do is to enable individuals to direct their own inquiry. This will require a concerted effort to construct data interfaces without becoming bogged down in cumbersome standards processes, to enable small, self-contained client-side and shareable tools ("widgets") to be deployed on top of NDS resources, and most importantly to utilize existing systems, communities and APIs, rather than mandating new development. The NDS must find ways to take advantage of existing networks of people rather than attempt to build new networks from whole cloth. We focus on the development and deployment of widgets for exploring data; developing such tools is particularly attractive for an initiative such as the NDS as the cost-benefit ratio is particularly favorable. They usually require only

minimal server-side maintenance, can be immersive, interactive and communicative, and they typically provide safe methods of moving the computational cost to the client-side.

Wikipedia describes fundamental elements of the scientific method: "**Another basic expectation is to document, archive, and share all data and methodology so they are available for careful scrutiny by other scientists, giving them the opportunity to verify the results by attempting to reproduce them**." The complexity and scale of modern experiments and computations and the limits imposed by the cost of data transfer and permanent archival data storage puts both of these non-negotiable elements of science in jeopardy. The NDS must effectively address these issues within a cost-constrained environment. Without the near-seamless ability to archive and share large amounts of raw data to enable effective peer review, the trend toward toward "on-the-fly" analysis means that much of the this data will be thrown away at the earliest possible moment.

2 Connected Platforms

Existing data *storage* platforms can be used for data (versioned and unversioned), sharing of results, and selective transmission of data to other researchers and collaborators. Figshare (`figshare.com`) provides the ability to create and assign DOIs to datasets and documents, and they have a select number of widgets that can visualize data inline (e.g., FITS files). GitHub and BitBucket (`github.com` and `bitbucket.org`, respectively) provide data versioning; both support diffing of a select number of filetypes, displayed in inline widgets. Sketchfab (`sketchfab.com`) supports uploading of vertex and texture files and displaying these as inline 3D WebGL models in the browser, but does not support the downloading or remixing of these models. The most exciting aspect of these widgets is their simplicity and their lack of server-side computation; once the data has been ingested by the storage platform, no additional computation is necessary and all display, manipulation and exploration of the data occurs on the client web browser.

Relying on the client for presentation, display and manipulation both vastly simplifies the server requirements to virtually none, and in doing so vastly increases the ability of the individual to explore data. Platforms such as Science Gateways *do* foster collaboration through sharing of widgets and tools, but do so by heavy reliance on server-side computation and the attendant security, provisioning and monitoring requirements.

Despite this potential, none of these platforms allow users to develop their own data widgets, export those widgets, or allow others to apply them to the data. As researcher-developers, if we want to build a javascript widget for browsing data from a cosmological simulation, we must do so outside of existing systems. On no existing data platform can a researcher upload a dataset, and enable collaborators to create their own data-exploration widgets.

By virtue of its unique position as a unifying, overarching force in this ecosystem, the NDS itself has the potential to effect great change simply by enabling such collaborative development. There are hundreds of visualization libraries for javascript, languages such as Python, C/C++, Ruby and Perl have been ported to work within the web browser, and it is the single most deployed platform for data access, processing, and visualization in existence. By opening up the NDS as a platform for collaboratively developing data exploration widgets, the NDS will benefit from the most powerful network effect in history. Alternately, by attempting to force new, incompatible or burdensome standards on the research community — such as domain specific languages, cumbersome XML standards, or obscure technologies requiring proprietary or incompatible software — it could become eclipsed by smaller, less feature-rich, but overall *simpler* platforms.

The deployment of software stacks, whether on NDS systems or on personal computers designed for small-scale analysis, often results in error-prone and time-consuming cycles of trial-and-error. Many solutions have been proposed and implemented — package managers (Homebrew, apt, RPM, Hashdist, Anaconda), modules (LMOD), all the way down to shell scripts. These diverse, largely non-overlapping

ecosystems each bring their own collections of received wisdom, their own levels of customizability, and their own levels of applicability to each specific deployment system. Much like many of the other systems relevant to NDS, deployment solutions are subject to difficulties arising from interface coupling and requirements management. As a simple example, the development of a graphical user interface (GUI) that works across NDS platforms and does not require the installation of a nearly-complete widgets toolkit is essentially an unsolvable problem; when examining questions of reproducibility, the difficulties of deployment are coupled inextricably with the need to ensure a repeatable software and hardware platform on which software is deployed.

Perhaps the single greatest effort in the history of computation has been put forth in the development of the World Wide Web (WWW); the web itself has become a platform on which computation can be deployed. Technologies like WebGL, WebCL, Asm.js and the (discussed below) Portable Native Client have transformed the web browser into the single most stable, reliably deployed platform in existence. While at present this cannot solve the problem of extremely large computational requirements, the technologies developed for web browsing — sandboxing of executables, isolation of requirements and well-defined transport layers — can be directly applied to the integration of components within the NDS.

3 Building in Collaboration

We propose three guidelines for policy and technology for development and deployment of data exploration widgets. Data exploration tools must be:

- Deployable by individuals
- Decoupled from datasets
- Able to be remixed and shared

Each of these aspects develops a different aspect of community growth and stewardship. Most importantly, individuals should be able to develop, test, and deploy widgets with as little friction as possible. This is the single most powerful means of empowerment available to individual researchers: directing their own inquiry. By enabling the deployment of widgets focused on client-side, sandboxed execution within the confines of the web browser, meaningful, detailed questions can be asked of data accessible through the NDS.

However, tying the data to its representation could potentially short-circuit the entire purpose of the NDS by undercutting access to the underlying data. To circumvent this, widgets must not be tied to specific datasets. For example, if someone develops a widget which enables the exploration of NetCDF files related to oceanography, others should be able to find and discover that widget (much like the NDS will enable discover of data itself) and apply it to additional, disparate datasets of similar or identical formats. In this way, the NDS can provide access to suites of data exploration tools that do not require installation, that do not require complex downloading procedures, and that leverage the network effect of the many users of NDS resources.

Finally, these tools must be sharable and modifiable. Development of widgets empowers individual researchers; remixing and modifying tools is a means of empowering *communities* of researchers, of leveraging Metcalfe's Law as it applies to people, and of building the means to exponentially increase the utility of collaboration between individuals and disciplines.

4 Pathfinding with `yt`

The BrownDog project is designed to curate the uncurated, to make available the "long tail" of data. Much like BrownDog aims to do for unstructured data, the `yt` project (`yt-project.org`) aims to accomplish for organized, regular or semi-regular data. Originally applied to astrophysical simulations, it now processes and visualizes data from nuclear engineering, seismology, and has made tentative steps toward processing data from the domains of weather, hydrology, and neurology. It is python-based, has simple methods for both ingesting data from a variety of sources and applying meaningful, inquiry-driven processes to it, and can be deployed on ranging from Linux, Mac OSX, Windows. Recently, we have

even ported it to run completely within the Chrome browser utilizing Portable Native Client (PNaCl).

Because `yt` is built on a semantically-aware foundation of indexing, physical understanding and inquiry-driven analysis and visualization, it can construct reduced data products that enable exploration of data that tie together both the representation and the underlying data. To enable this type of exploration, in past years the `yt` community has developed a "data hub" for sharing and analyzing data products.

We intend to develop the `yt` community as pathfinders for user-driven development within the platform provided by the NDS, rather than as an example of the fragmentation of data storage platforms. We will develop first-class functionality for services provided by the NDS within `yt` (both importing from the NDS and exporting codified "data products" to NDS-affiliated repositories), mentor and encourage younger researchers to build upon NDS platforms and technologies, and initiate projects to build collaborative data exploration widgets that utilize the enormous potential of the NDS. Taking advantage of Metcalfe's Law, we will utilize the network effect of the existing `yt` community to build a software platform to interact with, extend, and foster the growth of the National Data Service.

5 Conclusions

The National Data Service has a window of opportunity to act as a unifying system for enabling publishing, sharing, discovery and reuse of data. We assert that by enabling the development, deployment and sharing of data exploration widgets and tools built on web technologies, it could act as a platform for empowering, sophisticated collaboration around data. Identifying pathfinder communities on the part of infrastructure is critical, and has already begun; we believe that identifying pathfinder communities of scientists, such as `yt`, is equally necessary to ensure its success. Enabling these communities to self-direct their growth will build engagement and foster deeper relationships between NDS and its constituents.

How systems and institutions facilitate the scientific method is in need of a revision, to accommodate the massively cooperative, incredibly powerful and inherently collaborative network that is the modern community of scientists. Scientists are now able to communicate, directly transfer technology, and develop meaningful new questions to ask of data with minimal overhead. By taking simple steps not to control this web of people, but to instead allow them to do what comes naturally — collaborate, share, conduct inquiry — the NDS can lead a revolution in how the scientific method is realized.