

SeedMe: A scientific data sharing and collaboration platform

Amit Chourasia

University of California, San Diego
9500 Gilman Dr., MC 0505
La Jolla, CA – 92093
amit@sdsc.edu

Mona Wong

University of California, San Diego
9500 Gilman Dr., MC 0505
La Jolla, CA – 92093
mona@sdsc.edu

Dmitry Mishin

University of California, San Diego
9500 Gilman Dr., MC 0505
La Jolla, CA – 92093
dmishin@sdsc.edu

David R. Nadeau

University of California, San Diego
9500 Gilman Dr., MC 0505
La Jolla, CA – 92093
nadeau@sdsc.edu

Michael Norman

University of California, San Diego
9500 Gilman Dr., MC 0505
La Jolla, CA – 92093
mlnorman@sdsc.edu

ABSTRACT

Rapid secure data sharing and private online discussion are requirements for coordinating today's distributed science teams using High Performance Computing (HPC), visualization, and complex workflows. Modern HPC infrastructures enable fast computation, but the data produced remains within a site's storage and network environment tuned for performance rather than broad easy access. To share data and visualizations among distributed collaborators, manual efforts are required to move data out of HPC environments, stage it locally, bundle it with metadata and descriptions, manage versions, encode videos from visualization animations, and finally post it all online somewhere for secure access and discussion among project colleagues. While some of these tasks can be scripted, the effort remains cumbersome, time-consuming, and error prone. Thus, a more streamlined approach with a persistent infrastructure is needed.

In this paper we describe *SeedMe* – the *Stream Encode Explore and Disseminate My Experiments* platform for web-based scientific data sharing and discussion. *SeedMe* provides streamlined data movement from HPC and desktop environments, metadata management, data descriptions, video encoding, secure data sharing, threaded discussion, and, optionally, public access for education, outreach, and training.

CCS Concepts

- **Human-centered computing** **Visualization**
- **Information systems applications** **Collaborative and social computing system tools**
- **World Wide Web** **Web interfaces** **Web services**

Keywords

Data sharing; Collaboration; Visualization; Video encoding

1. INTRODUCTION

High Performance Computing (HPC) is an essential tool for current large-scale simulation and data analysis. Large systems like *Comet* at the San Diego Supercomputer Center, or *Blue Waters* at the National Center for Supercomputing Applications, use carefully tuned hardware with fast processors, memory, interconnects, and storage. However, the high user demand, security mechanisms, and expense of these systems makes them a poor choice for hosting the simpler needs of web servers and support tools focused more on user communications than getting the highest possible compute performance.

Instead of hosting web tools directly atop HPC infrastructures, convention places them on separate low cost systems that do not share HPC's expensive processors, networks, and storage. Sharing data among collaborators entails moving data out of the HPC environment and onto these separate systems where the data needs to be transcoded, annotated, visualized, packaged, and posted for secure on-line distribution among collaborators. Since HPC data is often large, complicated, and distributed across many thousands of files, the administrative effort to do this properly can be surprisingly significant. Time spent doing it for run after run of big iterative HPC jobs is annoying and wastes time a researcher could better use doing science. *Ad hoc* scripts can automate some of this, but the process remains laborious and error prone.

In this paper we describe the *SeedMe* platform that includes a web front-end, a data management back-end engineered to streamline secure scientific data sharing, and clients for command-line and programmatic interaction. Section 2 briefly surveys related work, Section 3 discusses *SeedMe* features from a user perspective, while section 4 describes the technology developed for this platform. Section 5 concludes with notes about current usage patterns, while Section 6 discusses future directions.

2. RELATED WORK

On-line data collaboration tools fall into a few broad and overlapping categories. *Cloud middleware*, like Globus Online [11], provide extensive high-end features such as HPC job scheduling, data movement between HPC storage systems, and access controls using centralized authentication services. But powerful products like these also require significant expertise to install and maintain, and the product itself doesn't provide storage, discussion capabilities, data viewing, or transcoding tools. These products also may not be available on the breadth of computing platforms researchers use for different stages of their research work. Cloud middleware is certainly useful throughout HPC, but it doesn't quite address the data sharing tasks at hand.

Software repositories like GitHub [13], and many others, offer more focused collaboration services to handle code directory trees, revision control, bug tracking, documentation, wikis, threaded discussion, color-coded file differences, and usage tracking. These are desirable features for software development, but are a misfit for collaborative science where HPC input and output data sets are much larger, binary, and not amenable to revision control, bug tracking, or presentation as text in a web browser with color-coded differencing.

File hosting (a.k.a. *file sharing*) services provide a simpler approach that focuses on cloud-based folders to store files for quick sharing among friends and colleagues, or syncing across multiple devices. Dropbox [6] is perhaps the canonical example, but there are many others, such as Box [4] and Google Drive [14], Apple iCloud[15], and Microsoft OneDrive [21]. Services like these typically provide a web-browser interface to drag-and-drop files from a desktop computer and view text, image, and document file types within a browser or mobile app. These are certainly useful services for collaboration, but they are again a misfit for computational science where it is not practical to manually drag-and-drop hundreds or thousands of files into a browser or mobile app window. These services usually do not support data descriptions, metadata, or threaded discussions of the hosted data. Storage size limits are also uncomfortably low for handling the number and size of data files generated by HPC jobs.

Content Management Systems (CMS), such as Drupal [7], Joomla [17], MediaWiki [19], and WordPress[26], offer rich and well-supported web-based platforms to present text, images, and videos for blogs and project websites. File hosting sites like Figshare [10] (originally based on MediaKit) leverage CMS abilities to provide data publication oriented services primarily aimed at dissemination. HubZero [18] (based on Joomla) provides a compute, content management, and collaboration environment focused on education. In each case, CMS platforms provide a good starting point for building a web-based presence, but none of them include out-of-the-box features for scientific data management, programmatic posting of data files from an HPC environment, or transcoding and video management services.

In the HPC space, *scientific data sharing* needs to consider how to move large numbers of files programmatically, rather than manually in a web browser. Sensitive research data and metadata needs to be transferred and stored securely. Transcoding and viewing features need to handle common tasks, such as visualization video encoding for streaming to desktop and mobile platforms. And a service needs to support threaded discussion as collaborators review results and plan further research. Additional tools need to support integration with existing research workflows. Scientific data sharing is more than a web-accessible

folder and files – it is a tool for collaboration. The *SeedMe* Platform addresses these issues to meet the needs of the computation community.

3. SEEDME FEATURES

SeedMe is an acronym for *Stream Encode Explore and Disseminate My Experiments* [5, 23]. The name reflects a project history that began with automated web-based video encoding, transcoding, and streaming tools for visualization animations assembled from HPC data. The project has since expanded into a general scientific data sharing platform.

Principal user-visible features of *SeedMe* include:

- Arbitrary numbers of files organized into collections
- File and collection descriptions
- Metadata tables of key-value pairs
- Time-stamped “ticker” messages to report job progress
- Video encoding, transcoding, and streaming
- Threaded discussion among collaborators
- Access controls
- Secure viewing and data transfers
- Federated login via a user's institution
- Multiple client tools to post and retrieve content

3.1 Files and Collections

At its core, *SeedMe* manages an arbitrary number of data files of arbitrary type. Data files are named and organized into *collections* owned by a principal user. Users may have any number of collections and they may grant view access to selected collaborators.

Key/Value Pairs	Files	Images	Sequences	Tickers	Videos
Title	Filename	Modified date			
Bonaire 2008, Pink Beach diver and photo frame Documentation of photo frame using a PVC rectangle base, struts, and a mounted camera aimed downward through frame base. The base includes tape to note the data collection location and date, along with white, black, and colored tapes used for a color reference.	IMG_3198.jpg	2016-06-01 18:04			
Bonaire 2008, Pink Beach 20m transect, DSC 4076 Seafloor and photo frame partially obscured by soft coral. Color references visible. No color balance or contrast enhancement.	DSC_4076.JPG	2016-06-01 17:46			
Bonaire 2008, Cliff 10m transect, DSC 2766 Variety of corals and other reef inhabitants. Photo frame is visible with color references. No color balancing or contrast enhancement.	DSC_2766.JPG	2016-06-01 17:39			
Bonaire 2008, Cliff 5m transect, DSC 3045 Coral amidst highly silted sea floor indicative of the area. Photo frame is visible with color swatches. No color balancing or contrast enhancement.	DSC_3045.JPG	2016-06-01 17:38			

Credits:
D. Stokes, J. Leichter (UCSD/Scipps Institute of Oceanography)
R. Sinkovitz, D. Nadeau (UCSD/San Diego Supercomputer Center)

Figure 1. A *SeedMe* collection for a research project.

Every collection has a title, descriptive text, metadata, author access controls, and a set of images, animations, videos, and other data files (Figure 1). When a collection contains data actively being generated by a running HPC job, status “tickers” may record job progress with a time-stamped sequence of messages posted by the job periodically.

3.2 Programmatic File Posting Tools

Files may be manually posted to a collection through the web interface, individually or in groups with drag and drop. However, most collection files are posted programmatically using a variety of tools available from *SeedMe* and third parties.

SeedMe provides a Windows command-line utility, a Java client, and a Python module which doubles as a command-line tool for Linux, OS X, and Windows platforms (Figure 2). The Python module may be invoked manually to post files and folders, but it is more often invoked within HPC job scripts to automatically post files as they are generated. This enables collaborators to monitor the progress of long-running HPC jobs by periodically checking the *SeedMe* web site for posted results.

```
seedme.py -auth_path ~/seedme.txt \
-update 1234 -file_path "*.png"
```

Figure 2. A *SeedMe* command line to post a set of images.

There are also multiple third-party tools for posting files to *SeedMe*. For instance, the *Kepler* [1] scientific workflow manager comes with integrated support to post files to *SeedMe* during workflow execution. The *OpenMM* [22, 24] macromolecular simulation toolkit has a *SeedMe* plugin to report simulation progress using ticker messages, and the *VisIt* [25] visualization software includes the *SeedMe* Python module to post files and images directly to *SeedMe*.

All of these posting methods reference collections by a unique numeric ID assigned to each collection (e.g. “1234” in Figure 2). These IDs enable scripts to update collections without repeatedly referencing a collection’s human-readable title, which is often long and may include embedded spaces and punctuation that can be problematic on a command line.

3.3 Metadata

Collections may include a metadata table of user-defined key-value pairs (Figure 3). For example, collections may use metadata to document HPC job parameters, software version numbers, the number of cluster nodes, cores, and threads for a job, start times, run times, job script names, and so forth.

Collection metadata may be posted from the command-line, so a common use is to post it directly from an HPC job script, as the job runs, to record job parameters (Figure 4).

Key	Value	
Average tile processing time	3.2718 sec	<input checked="" type="checkbox"/> <input type="checkbox"/>
Image tile size	64 x 64	<input checked="" type="checkbox"/> <input type="checkbox"/>
Number of Images processed to date	17952	<input checked="" type="checkbox"/> <input type="checkbox"/>
Number of nodes	128	<input checked="" type="checkbox"/> <input type="checkbox"/>
Number of threads/node	16	<input checked="" type="checkbox"/> <input type="checkbox"/>
Number of tiles/image	512	<input checked="" type="checkbox"/> <input type="checkbox"/>

Figure 3. Collection metadata showing job parameters.

```
seedme.py -auth_path ~/seedme.txt \
-update 1234 \
-keyvalue "Number of nodes:128"
```

Figure 4. A *SeedMe* command line to update metadata.

3.4 Job Progress “tickers” Messages

Collections may include logs of time-stamped progress/debug messages, called “tickers”. These are typically posted periodically from the command line by long-running HPC jobs to report on job progress (Figure 5).

```
seedme.py -auth_path ~/seedme.txt \
-update 1234 -ticker "25% done"
```

Figure 5. A *SeedMe* command line to add a ticker.

A ticker is similar to a conventional *print* statement reporting job progress. But where text from a *print* statement usually goes into an HPC job log file that isn’t easily accessible until after a job completes, *SeedMe* tickers can be posted during a job run and are immediately available on the collection’s page (Figure 6). The principal user and their collaborators may monitor the job as it runs without logging into the HPC system.

Key/Value Pairs	Files	Images	Sequences	Tickers	Videos
Ticker				Timestamp	+
Classifying Bonaire 2008: DSC_4075.jpg				06/14/16 - 10:39:33	<input checked="" type="checkbox"/> <input type="checkbox"/>
Classifying Bonaire 2008: DSC_4074.jpg				06/14/16 - 10:39:30	<input checked="" type="checkbox"/> <input type="checkbox"/>
Classifying Bonaire 2008: DSC_4073.jpg				06/14/16 - 10:39:27	<input checked="" type="checkbox"/> <input type="checkbox"/>
Classifying Bonaire 2008: DSC_4072.jpg				06/14/16 - 10:39:24	<input checked="" type="checkbox"/> <input type="checkbox"/>
Classifying Bonaire 2008: DSC_4071.jpg				06/14/16 - 10:39:21	<input checked="" type="checkbox"/> <input type="checkbox"/>
Classifying Bonaire 2008: DSC_4070.jpg				06/14/16 - 10:39:17	<input checked="" type="checkbox"/> <input type="checkbox"/>

Figure 6. Collection tickers showing job runtime progress.

3.5 Image Sequences and Video Encoding

SeedMe’s roots are in automated video encoding to assemble large image sequences into videos at multiple resolutions and compression levels for streaming to desktops and mobile devices. These features remain a key part of the project (Figure 7).

Images may be posted to the site manually via the web interface or using any of the previously discussed command-line and third-party tools. HPC jobs using *in situ* visualization may generate and post images on-the-fly, while a large simulation continues to run, to provide a visual record of a simulation’s progress.

Images may be viewed individually or in sequences from a web browser, or automatically encoded to video for streaming playback. Video encoding can be requested manually from the web browser, or triggered automatically from the command-line (Figure 8).

Video encoding automatically creates videos at several quality levels to tune bit rates for playback on desktops and mobile devices with varying bandwidth limits. Videos may be streamed directly from *SeedMe*, or downloaded for later playback.

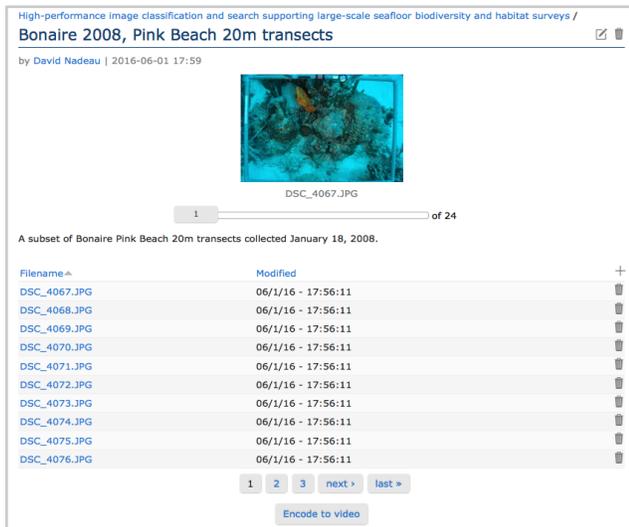


Figure 7. An image sequence ready to encode to video.

```
seedme.py -auth_path ~/seedme.txt \
  -update 1234 \
  -sequence_path "images*.png" \
  -sequence_title "Animation" \
  -sequence_frame_rate 10 \
  -sequence_encode
```

Figure 8. A *SeedMe* command line to post an image sequence then encode it to video.

3.6 Downloads

Any file posted to *SeedMe* is available through the web browser front end, and via the command line. Collections, data files, images, sequences, and encoded videos all can be downloaded programmatically, individually or as a group, to pull data back from *SeedMe* and save it locally for further processing, visualization, or archiving (Figure 9).

```
seedme.py -auth_path ~/seedme.txt \
  --download 1234 all mydir
```

Figure 9. A *SeedMe* command line to download all files in a collection to a local directory.

3.7 Threaded Discussion

SeedMe supports threaded forum-style discussions attached to any collection. Discussion posts are organized into threads and under control of the collection's owner.

3.8 Access Controls

Each collection has access controls that default to only granting the owner access to a collection. At the owner's discretion, one or more collaborators may be granted access to view and discuss a collection's contents.

Optionally, a user may open up their collections for public viewing, such as for education, outreach, and training. Public collections may document a project and/or provide a cache of data with which students may explore the project's topic on their own.

3.9 Secure Access

All *SeedMe* content is maintained on servers that limit content creation and discussion to users with validated accounts. All

account requests are vetted and validated to mitigate spurious users.

SeedMe's optional single sign-on features use the *CILogon* [3] service, which in turn relies on the *InCommon Federation* [16]. *CILogon* enables users to log in using their home institution's login credentials, without ever revealing them to *SeedMe* or any other site. Currently, *CILogon* enables federated authorization from over 175 universities, organizations, and government laboratories in the United States.

In addition to secure site logons, *SeedMe* maintains encrypted connections through the browser and for all data transferred to and from the platform from the command line or through third-party tools.

4. SEEDME TECHNOLOGY

SeedMe is implemented as a web front-end to an integrated suite of back-end open source software that includes:

- *Apache* [2] web servers
- A *Drupal* [7] content management system
- A *MySQL* [20] database
- The *Gearman* [12] job scheduler
- The *FFmpeg* [8] video encoding tools

Back-end software is integrated using custom *Drupal* modules and glueware written in PHP, and *SeedMe* clients are written in Java and Python. The site runs on a cluster of primary and backup Linux servers and Mac minis for video encoding.

4.1 Content Management

The *Drupal* content management system is a large PHP code base invoked on-demand by an *Apache* web server in response to incoming URLs. *Drupal* parses the URL and invokes the appropriate back-end code to issue one or more queries to a *MySQL* database containing administrative and textual content and inserts that content into selected web page templates. The formatted page and associated images, videos, scripts, and style sheets are returned to the web user for display.

Drupal's rich feature set, extensibility, and broad community support make it a compelling development platform for web-based projects like *SeedMe*. However, all but the simplest of *Drupal* sites invariably require custom configurations and custom extensions. *SeedMe*'s customizations principally include:

- *CILogon* support for secure logons based upon a user's home institution credentials
- Specialized "content types" for collections, files, image sequences, videos, and user accounts
- Access control mechanisms to enable users to grant selective view only access to a collection for specific collaborators
- User-base management to handle account requests, automatic account validation, and announcements
- Collaboration features to issue new user account invitations, and send email notifications when watched content is changed
- REST (Representational State Transfer) services for adding, deleting, updating, downloading, and listing content from command-line and third-party tools

- Editing features to manage integrated content with custom dialogs, drag-and-drop file handling, and other specialized user interface elements
- Video encoding services that queue long-running jobs to back-end services that run independent of the web front-end

Virtually all *Drupal* sites, *SeedMe* also includes community contributed modules that are configured to meet the site's needs. Finally, *SeedMe* also employs custom page layouts, a specialized look and feel, and extensive tweaks to provide improved usability for the specific needs of *SeedMe*'s users.

4.2 Security

Apache, *Drupal*, *MySQL*, and the underlying *Linux* servers are all configured to provide a secure environment. Private networks connect back-end servers and storage systems, encryption is used on all external connections to users, and multiple security layers are used to watch access patterns, lock out bad agents, and block spam.

Two security features are of particular note:

4.2.1 Password-less Access via API Keys

SeedMe issues each user an "API Key" that contains a unique encrypted identifier. Command-line and third-party tools use this API Key and pass it to *SeedMe* on each request, where it is validated to confirm authorized access to site content. This approach insures that command-line and third-party tools never know a user's logon name or password and that this information is never available in an unencrypted form.

While API Key and command-line tools do allow necessary access to a user's files, they do not grant login access to the site itself via a web browser. This insures that a compromised API Key cannot be used to steal an account, change login credentials or otherwise lock out or spam the user and their collaborators.

API Key can be reset and regenerated at any time by the user by logging in to their account via a web browser. Generating a new key invalidates any prior key.

4.2.2 Validated File Access

SeedMe stores all data, image, and video files in a private file system that is not directly served by *Apache*. All files are accessed via *Drupal* and *SeedMe*'s account validation procedures. Furthermore, stored files are given "munged" names that do not match their original file names and insure that file names cannot be guessed and cannot collide with existing files.

SeedMe handles the mapping between munged and user defined file names automatically using relationships stored in the *MySQL* database. Munged names and the munging algorithm are never visible to the user and files cannot be accessed directly without going through *SeedMe*'s security mechanisms.

4.3 Web Services

Command-line and third-party tool access to *SeedMe* requires web services that do not involve a web browser. Implemented using *REST* (Representational State Transfer) [9], these services leverage the web's standardized protocols to create, update, retrieve, delete, and query content.

Drupal's base support for REST services has been extended by *SeedMe* to support particular usage patterns and *SeedMe*'s security approach (e.g. API Keys). Additionally, all data transfers

to and from *SeedMe* using REST services are end-to-end encrypted.

4.4 Video Encoding

Encoding an image sequence into a video is a time-consuming process that usually cannot be completed within the few seconds available to respond to a user during web interaction. Instead, *SeedMe* queues video encoding jobs using the *Gearman* job scheduler running across back-end servers.

Gearman jobs retrieve image files and execute video encoding using the open source *FFmpeg* tools. Each video is created in the MPEG-4 video format to support a variety of web browsers. Videos are also created at multiple compression levels, from high-quality to low-quality, in order to support smooth playback on a variety of desktop and mobile devices with different network bandwidths.

Upon job completion, *SeedMe* adds the encoded videos to the user's collection. Thereafter, users may stream the videos in their web browser, or download videos at a desired quality for later local playback.

5. DISCUSSION & CONCLUSIONS

SeedMe is a scientific data sharing and collaboration service that exists at a mid-point between file hosting services and social networks. It supports remote browser-based access to shared files and collections in a way that is loosely similar to file hosting services like DropBox. At the same time, *SeedMe* provides collaboration and discussion features that are loosely similar to social networks. But unlike both of these, *SeedMe* provides metadata management, video encoding, and command-line and third-party tools to programmatically post, query, and retrieve data using HPC job scripts.

SeedMe's user base spans a broad array of disciplines and its unique feature set is in active use by researchers as a tool to share computed and experimental data among distributed collaborators, track the status of long-running jobs, and encode visualization images into videos used to quickly review computation job results.

While *SeedMe* collections are often kept private to a project's collaborators, *SeedMe* is also used to host public collections containing data and discussion aimed at education, outreach, and training. Students may download data from *SeedMe* collections and do their own computation and visualizations, then contribute to ongoing threaded discussions for the collection.

SeedMe's usage is inherently bursty, as users run big HPC jobs and quickly post hundreds or thousands of files. After the job completes, collaborators pour over the data for a time as they plan their next job run. When that job is ready and runs, another burst of files are posted to *SeedMe*. This bursty pattern repeats across multiple research groups and leads to an unpredictable load on the system. To accommodate it, the underlying servers, networks, and storage systems are intentionally over-engineered.

In conclusion, the *SeedMe* platform provides an important and useful infrastructure for computational work. It fulfills several unmet research needs for easily describing, organizing, and sharing data and metadata, encoding videos, and discussing ongoing and completed work in a secure collaborative environment.

6. FUTURE WORK

While *SeedMe* has proven to be a popular science collaboration tool, the current project design has limitations that we are addressing in a 2.0 design. This new version expands collections to support arbitrary directory trees of files, metadata, and threaded discussions so that there are more options for hierarchically organizing content. The next generation of *SeedMe* also integrates science data viewing features to quickly build plots, charts, tables, and diagrams from data posted in a variety of microformats, from XML to JSON, CSV, and raw text. All of these features are being developed as modular building blocks that can be separately used in other *Drupal*-based sites or in even non-*Drupal* projects.

7. ACKNOWLEDGMENTS

We'd like to thank Andrew Ferbert, Andrea Zonca, Sebastian Amara, Brad Whitlock and Cyrus Harrison. We would also like to thank Apple Inc. for providing test hardware and software during the conceptualization phase. This material is based upon work supported by the National Science Foundation under Grant No. ACI-1235505 and ACI-1443083.

8. REFERENCES

- [1] Altintas, I. Berkley, C. Jaeger, E. Jones, M. Ludascher, B. and Mock, S. 2004. *Kepler: an extensible system for design and execution of scientific workflows*. Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on. Pages: 423-424.
- [2] Apache. 2016. *The Apache HTTP Server Project*. Retrieved Jun 2, 2016 from <http://httpd.apache.org/>
- [3] Basney, J. Fleury, T. and Gaynor, J. 2014 "CILogon: A Federated X.509 Certification Authority for CyberInfrastructure Logon," *Concurrency and Computation: Practice and Experience*, Volume 26, Issue 13, pages 2225-2239, September 2014. <http://dx.doi.org/10.1002/cpe.3265>
- [4] Box. 2016. *Box – Secure Content & Online File Sharing for Business*. Retrieved Jun 2, 2016 from <https://www.box.com/>
- [5] Chourasia, A. Wong-Barnum, M. and Norman, M. *SeedMe Preview: Your Results from Disk to Device* In Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery (XSEDE '13). ACM, New York, NY, USA, Article 35, 4 pages.
- [6] Dropbox. 2016. *Dropbox*. Retrieved Jun 2, 2016 from <http://dropbox.com/>
- [7] Drupal. 2016. *Drupal – Open Source CMS*. Retrieved Jun 2, 2016 from <http://drupal.org/>
- [8] FFmpeg. 2016. *FFmpeg*. Retrieved Jun 2, 2016 from <http://ffmpeg.org/>
- [9] Fielding, R. T. and Taylor, R. N. 2002. "Principled Design of the Modern Web Architecture", *ACM Transactions on Internet Technology (TOIT)* (New York: Association for Computing Machinery) 2 (2): 115–150, May 2002
- [10] Figshare. 2016. *figshare – credit for all your research*. Retrieved Jun 2, 2016 from <https://figshare.com/>
- [11] Foster, I. 2016. *Globus Online: Accelerating and Democratizing Science through Cloud-Based Services*. In *IEEE Internet Computing*, vol. 15, no. 3, pp. 70-73, May-June 2011. doi: 10.1109/MIC.2011.64
- [12] Gearman. 2016. *Gearman Job Server*. Retrieved Jun 2, 2016 from <http://gearman.org/>
- [13] GitHub. 2016. *GitHub*. Retrieved Jun 2, 2016 from <https://github.com/>
- [14] Google Drive. 2016. *Google Drive – Cloud Storage & File Backup for Photos, Docs & More*. Retrieved Jun 2, 2016 from <https://www.google.com/drive/>
- [15] iCloud. 2016. *iCloud*. Retrieved Jun 2, 2016 from <https://www.icloud.com>
- [16] InCommon Federation. 2016. *InCommon Federation*. Retrieved Jun 2, 2016 from <https://www.incommon.org/federation>
- [17] Joomla. 2016. *Joomla – The CMS Trusted By Millions for their Websites*. Retrieved Jun 2, 2016 from <https://www.joomla.org/>
- [18] McLennan, M. and Kennell, R. 2010. *HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering*, 12(2), pp. 48-52, March/April, 2010.
- [19] MediaWiki, 2016. *MediaWiki*. Retrieved Jun 2, 2016 from <https://www.mediawiki.org/>
- [20] MySQL. 2016. *MySQL*. Retrieved Jun 2, 2016 from <http://www.mysql.com/>
- [21] OneDrive. 2016. *Microsoft OneDrive*. Retrieved Jun 2, 2016 from <https://onedrive.live.com/about/en-us/>
- [22] OpenMM. 2016. *OpenMM*. Retrieved Jun 2, 2016 from <http://openmm.org/>
- [23] SeedMe. 2016. *SeedMe (Stream Encode, Explore and Disseminate My Experiments)*. Retrieved Jun 2, 2016 from <https://www.seedme.org/>
- [24] SeedMe OpenMM plugin. Retrieved Jun 2, 2016 from <https://github.com/paesani/seedmeopenmm>
- [25] VisIt. 2016. *VisIt*. Retrieved Jun 2, 2016 from <https://wci.llnl.gov/simulation/computer-codes/visit>
- [26] WordPress. 2016. *WordPress – Create your site today*. Retrieved Jun 2, 2016 from <https://wordpress.com>.