FACILITATION OF THE A POSTERIORI REPLICATION OF WEB PUBLISHED SATELLITE IMAGERY

Mat Kelly
Old Dominion University
Department of Computer Science
Norfolk, Virginia 23529 USA
mkelly@cs.odu.edu
Advisor: Michele C. Weigle

ABSTRACT

Publicly available NASA satellite imagery hosted on NASA Langley Research Center's web servers are relied upon for analysis by atmospheric scientists. Because the data is hosted in a single location, any downtime, data loss, or latency is reliant on a single point of failure. Data redundancy would assist in dissemination of this data but it would then be necessary to ensure that updated versions of the data or fresh data from the same domain remain related with the original data. In this work, I borrow on technologies from other domains of digital preservation and data distribution to allow the progressively growing corpus of imagery at NASA to be harvested and made available on a peer-to-peer basis using facets of the frameworks behind ResourceSync, BitTorrent, and WebRTC.

The Problem

Satellite imagery is constantly being created for analysis by atmospheric scientists. This data is analyzed to detect patterns and to learn about the Earth's cloud conditions. This data set is ever-increasing in size, as past trends need to be analyzed. To make all of this data readily accessible to both the public and the scientists, it is made available in image form and put online[1]. This data set is ever increasing in size and thus only recent data is made readily available. Data older than a month or two is archived in a way where it is better preserved than on spinning disk (like the imagery) but not as accessible without issuing an ad hoc request.

The procedure of copying data from the production environment onto a medium meant for preservation is problematic. Frequently, this data gets mistakenly appended with intermediary data that is not part of the original data set by those doing analysis in-house. This superfluous, often one-off derivative data, does not need to be backed up but frequently is backed up- thereby being an additional burden on the resources meant for "deep storage".

Further, because this data is manipulated and the procedures to preserve it scripted, the backup procedure is prone to error. The current setup does not immediately halt the process on errors, sometimes resulting in incomplete or non-existent backups and offloading of data where another copy may not exist.

This initial study is a hands-on investigation of the potential application of applying digital preservation and file distribution technologies for the use case of the NASA data. In this work, facets of both the ResourceSync and BitTorrent frameworks are adapted and applied to develop a means of distributing public domain satellite imagery data on NASA Langley Research Center (LaRC) servers to an arbitrary collection of web users. Further, this research investigates applying these technologies in an a posteriori fashion where neither the structure nor the size of the target data (i.e., the images) is known. This system agnostic approach allows the products created in this research to be applicable to other systems and not limited to NASA's servers.

Relevant Technologies from Other Domains

Facets of two technologies, ResourceSync and BitTorrent, when joined together to exploit each one's relevance to the problem, provide a potential solution of preserving the accessibility of the satellite imagery.

ResourceSync

The ResourceSync Framework [6, 5] is a specification for synchronizing resources on the web. The framework describes the various entities involving in the process of duplicating resources in terms of advertising contents available, notifying each entity when a change has occurred in the data set, and ensuring that data duplicated maintains bit-level integrity. ResourceSync has been deployed in multiple systems that require data redundancy, namely arXiv.org, a repository containing preprints of scientific papers. The framework relies on an extension of the Sitemap format [7], the basis of which is frequently used by web crawlers to index web resources for search engines. This initial study adapts some of the concepts from ResourceSync.

BitTorrent

[1] `http://www-pm.larc.nasa.gov`

BitTorrent is a peer-to-peer file sharing protocol that is frequently used to share large amounts of data through strategic data segmentation and distribution [3]. Utilizing a distributed system like BitTorrent removes a single point of failure for data availability and allows data to be segmented and distributed and for partial downloads to be resumed and exchanged from other sources that have either the entirety of the data or only a portion. This study utilizes BitTorrent to partition the large data set into chunks that can be distributed and exchanged among clients.

WebRTC

WebRTC (Web Real-Time Communication) is a recently developed protocol for allowing peer-to-peer file sharing using users' respective web browsers without the need for additional browser plug-ins. Though the primary use case for the technology has been in video conferencing in a browser, WebRTC supports the sending of arbitrary data. WebRTC also provides circumvention of Network Address Translation (NAT) restrictions that would otherwise prevent a user on a Local Area Network (LAN), who is assigned an IP address inaccessible to the rest of the Internet, from acting as the data source for external clients.

How ResourceSync and BitTorrent Can Be Utilized in Conjunction

ResourceSync provides a standard for defining endpoints and resources available from a source (e.g., LaRC servers containing imagery data) and destinations (any interested users). BitTorrent is used to chunk large datasets for aggregation when recombined, even when the original source of the data has gone offline.

Certain attributes of each of these technologies are either inappropriate or inapplicable for file distribution from an a posteriori perspective. BitTorrent relies on the hashing and indexing of the partitions of content for the reliable transfer and distribution. ResourceSync's power of data duplication is largely stemmed on the cooperation of the client in distributing data. A for-purpose crawler was built for this study to provide the hashing of content for reliable distribution of data. This same software also acts as the "Adapter" and means of indexing to access content on the target servers (Figure 1).

Data

The target data set is an ever-growing hierarchical set of image data in the form of "cloud products" at the web site of LaRC (an example is shown in Figure 2). The data is temporally organized in a prettified directory structure for security (Figure 3). The top-level directory contains sub-directories of all years available at the moment of access (Figure 3a) with a similar system within for month and days (Figures 3b and 3c, resp.) with the final listing containing links to the images. Each image in the listing is semantically named. For example, the image in Figure 3d resides at `http://cloudsgate2.larc.nasa.gov/prod/goes-west/visst-pixel-gif/2015/04/01/G15.FD.2015091.2000.08KM.RGB.GIF`. The aforementioned date encoding can be seen within the path portion of this URI:

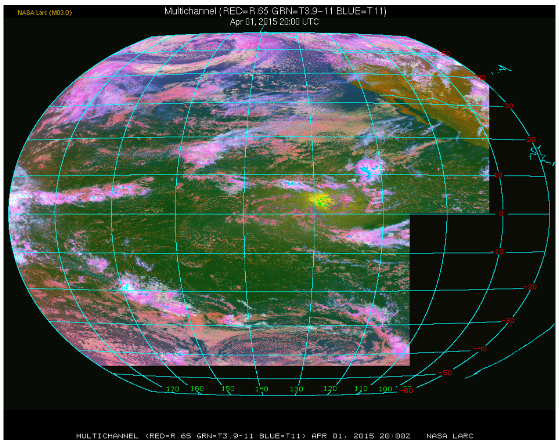

NASA Larc (M03.0)
Multichannel (RED=R.65 GRN=T3.9-11 BLUE=T11)
Apr 01, 2015 20:00 UTC
MULTICHANNEL (RED=R.65 GRN=T3.9-11 BLUE=T11) APR 01, 2015 20:00Z NASA LARC

Figure 2: An example of a payload from one of the images in the directory listing in Figure 3d. This image resides on the LaRC web servers at `http://cloudsgate2.larc.nasa.gov//prod/goes-west/visst-pixel-gif/2015/04/01//G15.FD.2015091.2000.08KM.RGB.GIF`.

- "goes-west/visst-pixel-gif" refers to the cloud product, e.g., the radiation spectrum, target portion of the Earth, etc.
- 2015091 refers to the ninety-first image captured in the year 2015.
- 2000 refers to the time of day the capture represents (assumably, 8 o'clock p.m.).
- 08KM refers to the distance of capture for the satellite above earth.
- RGB is the radiation spectrum of the capture where other spectrums in the list by the same satellite at the same time of capture can be seen in Figure 3d.
- GIF is the file format of the image.

A Posteriori Replication

ResourceSync implicitly relies on the cooperation of a source in duplicating data held within for most schemes. This work assumes an a posteriori perspective on the data, that is, we wish to duplicate the data without being aware of the structure or quantity of the data. By crawling the data to establish a baseline, the system is able to generate the metadata to be consumed by an entity setup for distribution.

Software

For a reference implementation, we collapsed multiple software entities into one codebase for simplicity of access. The software needed was a harvesting agent to crawl the LaRC servers, a post-processor to progressively establish a hash for eventual data integrity, and a BitTorrent-style tracker to route peer requests to one another (Figure 5).

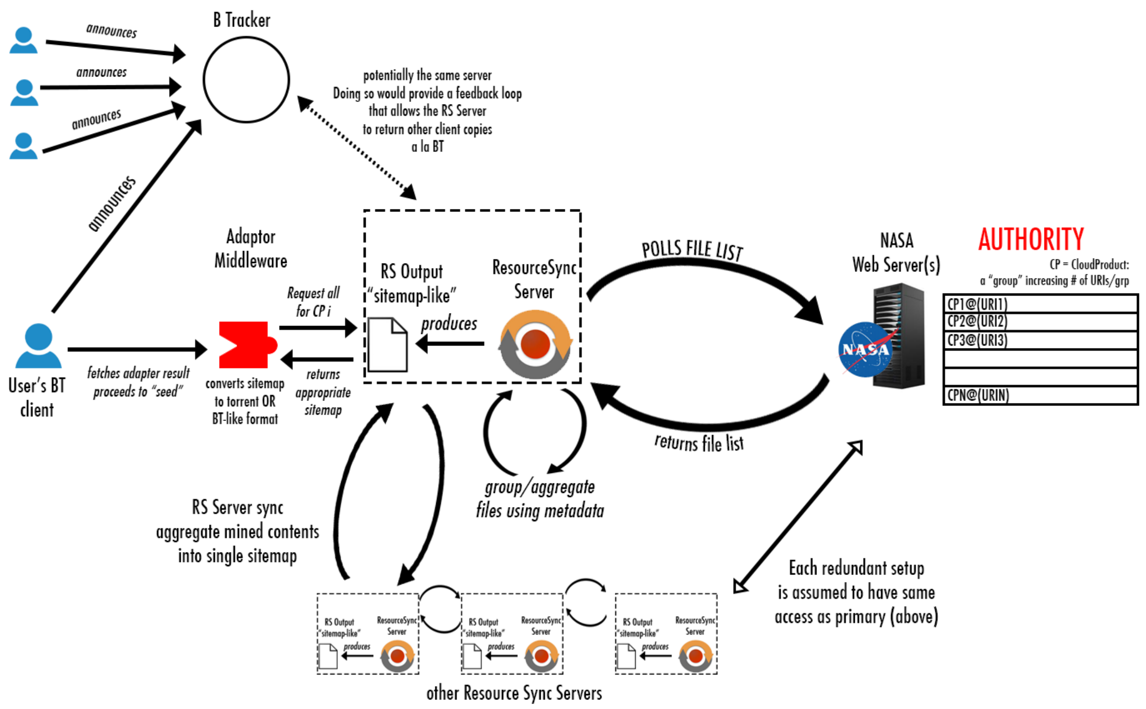


Figure 1: The originally envisioned hierarchy for peers' access to the satellite imagery data files on the LaRC servers.

The Progressive Crawler / Harvesting Component

I created ad hoc web crawler software to discover the resources available on the LaRC servers. This software re-uses the frontier concept from the Internet Archive's Heritrix archival crawler [8] to add the URIs of newly discovered resources to a list of files that make up the corpus. It is assumed that is monotonically increasing in size, though the possibility of redacted older data has not been explored.

To mitigate transferring large payloads, the HTTP HEAD method was used instead of HTTP GET [4] when a resource was identified as having a non-trivial payload (i.e., when the HTTP response body was more than a prettified directory listing). Alam et al. [2] confirmed that support for HEAD is almost exactly correlative to support for the GET method – the method used by web browsers when accessing a web page when a user simply enters a URI . The LaRC servers used to supply the cloud product images support HEAD, though this work ought to be reusable on other target data sets based on Alam's study.

A variety of formats was explored to implement the ResourceSync's sitemap-like structure [1] without relying on XML, which is used by both the Sitemap specification and ResourceSync. Because of the hierarchical nature of the target, YAML[2] (YAML Ain't Markup Language, a human friendly data serialization standard) was chosen due to a lower level of verbosity, a higher signal-to-noise ratio with the absence of structural brackets, and the higher degree of readability compared to XML and JSON (Figure 4). The secondary choice of JSON is a subset of YAML and is planned for native interaction with JavaScript-based clients in future work.

The crawler acts as the "destination" from the perspective of the LaRC servers and the "source" (both in ResouceSync nomenclature) from the perspective of the users. The crawl rate can be tailored based on observing the last-modified HTTP header if present or by performing a re-crawl on the directory where the images reside.

BitTorrent Access Point

From the results generated by the crawler, a secondary service acts as the parser and access point for users to fetch files using BitTorrent-like clients. I used the WebTorrent[3] open source Node.js module for hashing the data files on-demand. This software is a JavaScript-based BitTorrent client with the additional feature of providing access to the protocol via WebRTC. At this time, no standard BitTorrent client supports BitTorrent via WebRTC though the module provides a JavaScript-based include for adding the functionality to a hosted web page.

Middleware

To join the BitTorrent Access Point and our crawler, we introduced middleware into the programmatic flow that reads the crawler's output and produces a web-readable list of URIs as the source of images. The interface to the user allows the resource at the URIs harvested by the crawler to be fetched on-demand. Upon this "Just-in-time" fetching, a hash

[2] `http://yaml.org/`

[3] `https://github.com/feross/webtorrent/`

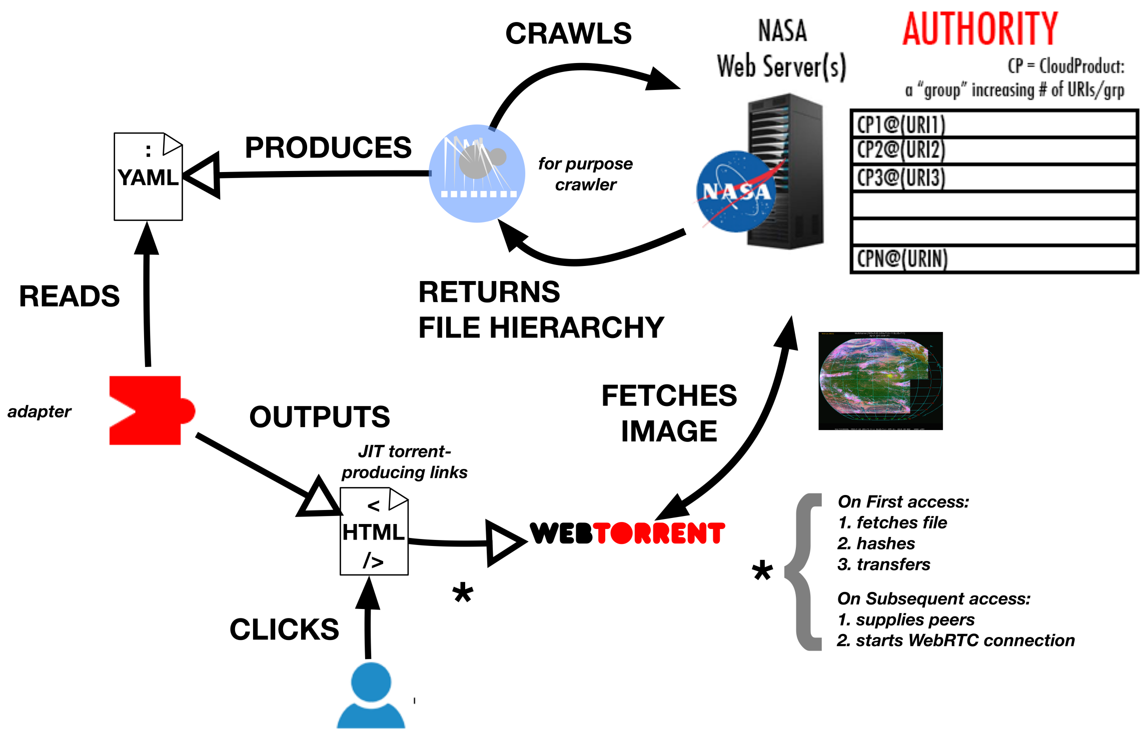


Figure 5: The modified software component of the research is more specific about the external libraries and formats utilized.

is created and the file continues to be "seeded" to other users in a client-to-client form.

Results

Preliminary tests showed that the data is duplicated to clients when an image is selected on a page representative of the subset of data. Multiple clients on disjoint networks also were served data from other clients when accessing the same links rather than the LaRC servers. Further tests were done to enforce partial downloads of individual files from the server to evaluate aggregation of the data solely obtained from clients. These tests initially seemed effective but a limitation in the WebTorrent code appeared to cause inconsistent results when this restriction was enforced.

Future Work

Inherent in the revised version of BitTorrent protocol is the ability to utilize a distributed hash table (DHT), which would allow data cohesion, access, and distribution of data in the event that the primary data source becomes non-existent. This might also assist in the data acquired in the setup described in this paper from existing solely between clients instead of still relying on a centralized entity.

Multiple image formats allow for additional sets of arbitrary data to be stored within an image file without a visual change of the image. Normally, this space is reserved for metadata about the image itself, but it might also be used to facilitate the distribution of the file by storing hash or peer information within this space.

The software created for this study is a proof-of-concept. Further efforts toward improving the software behind the setup used would make the work more publicly useful.

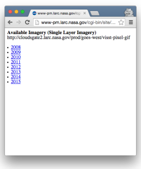
The decision to utilize YAML prevented full use of the ResourceSync framework at the potential advantage of being more human readable. This research resulted in a setup where the YAML was only consumed by machines, which proved its advantage moot. In the future, utilizing ResourceSync's sitemap extension will be investigated with the hope that more features from ResourceSync can be applied.

Conclusion

This research served as a proof-of-concept on using concepts from ResourceSync, BitTorrent, and WebRTC to effectively duplicate publicly available web-based imagery resources to interested web users. A primary contribution of this system was the discovery process of resources allowing for a posteriori duplication of content without being coupled with expected hierarchies in structure of the target data. A portion of the merit of this work is for its potential applicability onto other domains beyond NASA imagery data.

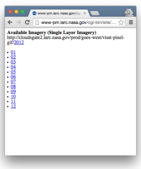
Acknowledgements

This research was supported by the NASA-sponsored Virginia Space Grant Consortium's graduate research fellowship. Further support for discussion, encouragement, and review were provided by Michele C.


Available Imagery (Single Layer Imagery)
http://cloudsgate2.larc.nasa.gov/prod/goes-west/visst-pixel-gif

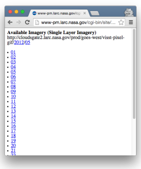
- 2008
- 2009
- 2010
- 2011
- 2012
- 2013
- 2014
- 2015

(a)


Available Imagery (Single Layer Imagery)
http://cloudsgate2.larc.nasa.gov/prod/goes-west/visst-pixel-gif/2012

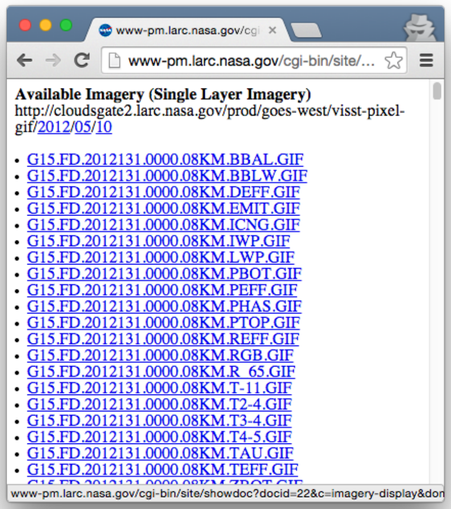
- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08
- 09
- 10
- 11
- 12

(b)


Available Imagery (Single Layer Imagery)
http://cloudsgate2.larc.nasa.gov/prod/goes-west/visst-pixel-gif/2012/05

- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08
- 09
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21

(c)


www-pm.larc.nasa.gov/cgi-bin/site/...
www-pm.larc.nasa.gov/cgi-bin/site/showdoc?docid=22&c=imagery-display&don

Available Imagery (Single Layer Imagery)
http://cloudsgate2.larc.nasa.gov/prod/goes-west/visst-pixel-gif/2012/05/10

- G15.FD.2012131.0000.08KM.BBAL.GIF
- G15.FD.2012131.0000.08KM.BBLW.GIF
- G15.FD.2012131.0000.08KM.DEFF.GIF
- G15.FD.2012131.0000.08KM.EMIT.GIF
- G15.FD.2012131.0000.08KM.ICNG.GIF
- G15.FD.2012131.0000.08KM.IWP.GIF
- G15.FD.2012131.0000.08KM.LWP.GIF
- G15.FD.2012131.0000.08KM.PBOT.GIF
- G15.FD.2012131.0000.08KM.PEFF.GIF
- G15.FD.2012131.0000.08KM.PHAS.GIF
- G15.FD.2012131.0000.08KM.PTOP.GIF
- G15.FD.2012131.0000.08KM.REFF.GIF
- G15.FD.2012131.0000.08KM.RGB.GIF
- G15.FD.2012131.0000.08KM.R_65.GIF
- G15.FD.2012131.0000.08KM.T-11.GIF
- G15.FD.2012131.0000.08KM.T2-4.GIF
- G15.FD.2012131.0000.08KM.T3-4.GIF
- G15.FD.2012131.0000.08KM.T4-5.GIF
- G15.FD.2012131.0000.08KM.TAU.GIF
- G15.FD.2012131.0000.08KM.TEFF.GIF

(d)

Figure 3: The LaRC servers provide a prettified hierarchy of data to access temporally organized images by year (3a), month (3b), day (3c), and finally a listing of images available for the selected temporal parameters.

```
Professor:
name: John Smith
Classes:
- Class: {title: CS418, time: 3pm, days: MW}
- Class: {title: CS101, time: 6pm, day: TR}
```

(a) YAML

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
    <Professor />
    <name>John Smith</name>
    <Classes>
        <Class>
            <title>CS418</title>
            <time>3pm</time>
            <days>MW</days>
        </Class>
    </Classes>
    <Classes>
        <Class>
            <title>CS101</title>
            <time>6pm</time>
            <day>TR</day>
        </Class>
    </Classes>
</root>
```

(b) XML

```
{
    "Professor": null,
    "name": "John Smith",
    "Classes": [
        {
            "Class": {
                "title": "CS418",
                "time": "3pm",
                "days": "MW"
            }
        },
        {
            "Class": {
                "title": "CS101",
                "time": "6pm",
                "day": "TR"
            }
        }
    ]
}
```

(c) JSON

Figure 4: The verbosity and additional syntax for valid YAML, XML, and JSON can be compared as a function of the necessary structural characters to express the same data in the three different formats. YAML was chosen in this study for human readability.

Weigle, Michael L. Nelson, and Sawood Alam, all of the Web Science and Digital Libraries Research Lab at Old Dominion University.

1. REFERENCES

[1] Sitemaps XML format, 2008. `http://www.sitemaps.org/protocol.html`.

[2] S. Alam, C. L. Cartledge, and M. L. Nelson. Support for Various HTTP Methods on the Web. Technical Report arXiv:1405.2330, Old Dominion University, 2014.

[3] Bram Cohen. The BitTorrent Protocol Specification. `http://www.bittorrent.org/beps/bep_0003.html`, January 2008.

[4] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. IETF RFC 2616, June 1999.

[5] M. Klein, R. Sanderson, H. Van de Sompel, S. Warner, B. Haslhofer, C. Lagoze, and M. L. Nelson. A Technical Framework for Resource Synchronization. *D-Lib Magazine*, 19(1/2):3, January/February 2013.

[6] M. Klein, R. Sanderson, R. Van de Sompel, Herbert, S. Warner, B. Haslhofer, M. Nelson, and C. Lagoze. ResourceSync Framework Specification (ANSI/NISO Z39.99-2014). http://www.openarchives.org/rs/resourcesync, 2014.

[7] M. Klein and H. Van de Sompel. Extending Sitemaps for ResourceSync. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 277–280. ACM, 2013.

[8] G. Mohr, M. Stack, I. Rnitovic, D. Avery, and M. Kimpton. Introduction to Heritrix. In *4th International Web Archiving Workshop*, 2004.