Data Logistics: Toolkit and Applications

Micah Beck
Terry Moore
University of Tennessee
mbeck@utk.edu
tmoore@icl.utk.edu

Nancy HF French
Michigan Technological University
nhfrench@mtu.edu

Ezra Kissel
Martin Swany
Indiana University
ezkissel@iu.edu
swany@iu.edu

ABSTRACT

We live in an age of highly distributed collaboration in nearly every field of science coupled with widely disparate technological capabilities. Such heterogeneity makes the movement and management of data that supports scientific and end-user communities a global challenge, particularly when placed in the context of under-served populations without access to advanced network and storage infrastructure. This paper describes an approach known as Data Logistics as a model for exposing generic storage at the network's edge, enabling broader access to data sharing capabilities supporting a wide range of devices and networking infrastructure. We present a survey of the underlying Data Logistics technologies, the software components and packaging, and some representative applications of our approach that highlight its use in different environments.

1 INTRODUCTION

Connecting today's global information infrastructure in a more capable and robust way to all parts of the developing or otherwise under-served world has proved to be an elusive goal for those who seek to apply modern information technology for social good. Achieving this goal is not only a topic of great commercial interest, but also, increasingly, a social and economic necessity. Even people who subsist on very low incomes find it necessary to have connectivity and some basic services. It is widely agreed that the availability of a greater selection of services can improve their lives and open them to new and better opportunities.

The most common approach to this fundamental problem is to try to upgrade the relevant infrastructure to enable high quality broadband connectivity. This approach not only supports asynchronous communication (e.g., fast file transfer) and streaming media, it also enables the most demanding kind of interactive applications, from Voice over IP to Desktop Sharing and collaborative office apps. While this model of development serves the current strategies of many important companies, it tends to be expensive and has proved difficult to implement in many parts of the world, requiring an expensive build-out of wired and wireless infrastructure. Efforts to implement it have even extended to the deployment of balloons or solar-powered drones flying permanently in the stratosphere over the region to be served. Successes on this front have tended to be marginal and halting, as is evident, for example, from the fact that two decades into the 21st century, "rural broadband" remains an unfulfilled promise in the United States.

Our work seeks to combine whatever interactive (high bandwidth, low latency) connectivity is available at an edge network with highly generic storage resources to provide the best quality of service for applications that can make use of both. It does not contradict the idea that wide area applications are extremely powerful and flexible, or that Cloud data centers can provide vast resources. It does ask whether these can be augmented with local storage resources to provide services that may not be available in certain areas (if connectivity to the backbone is challenged). The research challenge is to provide access to data that is open, interoperable, and scalable using mature software. Our term for this approach is Data Logistics, and as with many systems dealing with logistics, our system considers both the *geographic* and *temporal* aspects of the data being managed. Combined with *policy*, we form a Data Logistics Network (DLN) that is concerned with delivering data to storage locations that maximize user access and to make that data available over a period of time that provides the most value given the resource constraints of the underlying infrastructure.

This paper describes our development of a set of software elements that enable Data Logistics in a number of contexts, from nation-wide data sharing using fixed infrastructure to low-power, low-cost mobile deployments that can provide access to data in environments that have previously been difficult to support. We highlight applications of our Data Logistics approach as exemplars of how these elements may be used in practice. Our goal is to provide packaged software components as open source building blocks that will promote further integration in areas with limited network and storage capabilities.

2 BACKGROUND

One of the defining aspects of Internet datagram delivery is statelessness: a send operation initiated by an endpoint does not affect the persistent state of network intermediate nodes. However, some of the most important functions of the network can be most effectively and economically implemented only using storage. An important example of using storage is to deploy edge servers to stage content, and to then serve to clients over a high performance local network. This "electronic library" approach has been used in a variety of commercial applications (e.g. hotel video-on-demand) and for educational purposes (e.g. digital encyclopedias). Such offline solutions tend to suffer from the fact that they are isolated

Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and much more static than online approaches. Updates to the content are not automatic and are often controlled by the owner of the content delivery infrastructure, and thus not subject to competition.

Another important example of the importance of storage is Content Delivery Networks (CDNs) [14, 19], which implement an overlay multicast tree with their servers providing distributed storage and processing resources at strategic locations within the network topology. A DLN can take advantage of the same principles, namely those of locality and load distribution. Indeed a generic CDN could be constructed with as a DLN. However, our approach is open and extensible and made more flexible by exposing the structural metadata of the files and our efforts have made use of this to improve distribution and manipulation of domain-specific data. The most critical difference is that CDNs are offered as commercial services, and the ongoing service is expensive. This raises a host of issues, among them the generally prohibitive long-term cost for scientific and education communities and the lock-in that can occur with long-term storage of "big data." For many institutions it is easier to have a capital expense for adding new storage to a Data Logistics system than to support the operational expense of paying for a CDN. Further, commercial CDNs lack the capability for user-based publication of derived data products, and certainly lack the ability to utilize volunteer or intermittently-connected storage resources, as well as the ability to get portions of a file from various locations.

Peer-to-peer protocols like BitTorrent [17] are also used to efficiently distribute content with no centralized infrastructure, and naturally provide the ability to get chunks of data from multiple locations. Our approach can take advantage of the same performance benefits of simultaneous downloads from many sources and resiliency via distribution. At the same time, our approach is more malleable and allows policy-based distribution and load sharing in contrast to BitTorrent's operational mechanisms like random peer selection, optimistic unchoking and rarest-first distribution. While the benefits are the same, a DLN could not simply "use BitTorrent" as the control in BitTorrent is built-in and designed for P2P file-sharing concerns like preventing free-riding and rapid seeding of rare blocks.

Here, the innovation of Data Logistics is to provide distribution mechanisms that can be configured to behave like CDNs, P2P file sharing networks, parallel filesystems, etc. Our project seeks to embed storage resources at a variety of locations in the network topology, including edge networks, using a form of open, interoperable and scalable networking that includes storage. The intent is that storage should be available as an additional resource to network applications of all kinds, and should not be deployed by the operators of specific services or overlay networks for their commercial use alone. Today, high definition videos can be served from a $400 storage server to a rural school that otherwise has to settle for streaming over a low bandwidth wireless connection. The cost of the server, storage and local networking are cheap compared to wide area broadband connectivity. Using local storage of content in this way cannot replace the "unbounded" content available over a 100Mbps Internet connection, but a well engineered solution that maximizes dynamic update (and uses the available Internet capabilities to implement interactive functions) can implement a high quality content-rich environment. Storage can be used to serve any collection of content limited in size that does not change too suddenly. The challenge is how to make a network that includes such storage scale in a manner analogous to the Internet, and have a significant impact.

2.1 The Data Logistics Toolkit

The foundation of our software infrastructure is the *Data Logistics Toolkit* (DLT), an NSF CC-NIE funded software package that incorporates and extends the software components produced by the research program in *Logistical Networking* (LN). This technology was pioneered by authors [anonymized] for the express purpose of addressing the data logistics problems faced by the large CERN LHC and remote sensing/geospatial communities and by many other data intensive application communities. Below we describe the content and status of the current DLT software package; this easy to install bundle of components forms the basis of work of all the work described here.

2.1.1 *The Internet Backplane Protocol (IBP).*

The IBP Storage Service [3, 15] allocates, reads, writes and manages storage in variable-sized chunks known as allocations. The server that implements the IBP service is called a *depot*. In order to separate access policy from the mechanism that implements it, access to an allocation is governed by read, write, and manage keys, or *capabilities*. An allocation has no externally visible address or name which it can be referred to by IBP clients other than these three capabilities, which are returned by the depot as the result of a successful allocate operation. Designed explicitly to be a better form of storage for data logistics, its leading characteristics can be summarized as follows: 1) *Thin* - IBP is a minimal/primitive interface that includes no features that can be correctly and completely implemented at a higher layer, unless there is a clear benefit such as performance or security; 2) *Open* - IBP can be given policy settings that enable clients to make allocations without any notion of identity or authorization, so that clients can reside anywhere that has connectivity and are not restricted to specific domains (e.g., a LAN or SAN); 3) *Limited* - IBP can enforce policy such as maximum allocation size and maximum duration, but one basic notion is that, since no provision is made for a "filesystem check" to verify that all unreferenced allocations are free, allocations are time limited and must be refreshed; 4) *Non-rendezvous* - To rule out the use of IBP as a rendezvous point (e.g., between a content publisher and consumers) without some other service to create a match, directory listings are hidden and the server chooses long, semantically neutral names (e.g. random strings); 5) *Best Effort* - IBP can be thought of as best effort because there are no guarantees of service quality (correctness, performance, etc); 6) *Generic* - IBP does not include features that restrict the possible implementations; 6) *Third Party Transfer* - IBP allows third-party tranfser between nodes as a primitive; 6) *Multiplexing* - The use of a depot by one user should not be directly visible in any way by other users, but at best only indirectly inferable through its effects.

2.1.2 *The exNode.*

IBP is analogous to a block-level storage service and clients must aggregate these low-level storage allocations into larger structures like those found in file systems. The exNode (modeled on the Unix filesystem *inode*) is the data structure that aggregates storage allocations into a file-like unit of storage [2]. The exNode implements many, but not all, of the attributes that are associated with a file, leaving some, such as naming and permissions, to other services. While the exNode omits some file attributes, it also implements file-like functionality with greater generality than a typical file. For example, the exNode can express wide-area distribution of data replicas, whereas file systems are typically restricted to a local or enterprise network.

2.1.3 Intelligent Data Movement System (IDMS). IDMS was created as part of the NSF GENI [4] program's experimental initiative for deploying the DLT as a persistent storage service for experimenters. IDMS builds upon the Logistical Runtime System (LoRS) library [16] , including functionality known as the *dispatcher*, which distributes and refreshes storage allocations and includes logic to perform data positioning based on arbitrary policies. IDMS performs initial data placement at upload time and can also rebalance based on policy, existing conditions, and demand. In addition, IDMS includes policy-based logic to allocate storage resources on the fly from infrastructure as a service (IaaS) control frameworks, and this currently works with CloudLab [18], where long-running DLT services have been prototyped. Finally, IDMS includes the *harvester* service, which performs policy-based ingestion of external data and is currently running to automatically gather and distribute new remote sensing data products from the USGS [1] and Google Earth Engine [8] machine-to-machine interfaces.

2.1.4 Unified Network Information Service (UNIS). UNIS [7] is responsible for storing network topology, indexing and managing exNodes, associating them with administrative metadata such as a name in a hierarchical namespace, a concept of ownership by a specific user and/or group, and policies as to where replicas should be placed. UNIS combined with IDMS provide the core POSIX IO functions for a user-level filesystem. It can function as a set of distributed and replicated services to ensure its scalability. This includes distributing and replicating content and directory metadata across both local sites and in the wide area to provide an appropriate level of performance and fault tolerance.

2.1.5 The Phoebus Accelerator Service. Phoebus [10] is a WAN acceleration system that can dramatically improve end-to-end throughput by forwarding data via intermediaries, called Phoebus Gateways (PGs), placed at strategic locations in the network . Phoebus has been demonstrated to improve throughput for network applications via years of testing in real networks and in emulated environments, and has demonstrated efficacy for real scientific applications. As a network middlebox, Phoebus may be deployed in an on-demand manner when edge connectivity is severely inhibited and enabled via IBP configuration, but the service is not required to realize an installation of the DLT.

2.1.6 perfSONAR and Periscope. The DLT incorporates perfSONAR [9] and the newer Periscope [7] implementation, network measurement infrastructure, which includes the UNIS component described above. The perfSONAR implementation has seen tremendous adoption in R&E networks and is now in use around the globe. The aim of perfSONAR is to create a framework allowing a variety of network metrics to be gathered and exchanged in a multi-domain, heterogeneous, federated manner. Data from perfSONAR is used to provide network topology and performance information to the DLT dispatcher to inform tasks such as data balancing. The DLT extends perfSONAR to include host and storage metrics.

2.1.7 DLT Packaging. The DLT software has been packaged for a number of operating system distributions to facilitate deployment concerns, and a package repository is maintained and updated at *https://data-logistics.org*. A DLT meta-package is available to end-users that resolves and installs any necessary dependencies needed to deploy an IBP depot node, which may include optional measurement and Phoebus components depending on the desired configuration. Installation instructions and a best-practices document are also maintained and made publically available.

Virtualization technology has also necessitated the ability to easily instantiate and configure DLT on cloud, shared, or multi-tenant infrastructure. To that end, the core DLT services (IBP depot, Phoebus WAN accelerator, IDMS, Periscope) have been packaged as appliance images (VMs) that may be deployed on OpenStack and Emulab-based rack technologies. One such deployment target has been on GENI's flexible cloud infrastructure where DLT has been used extensively in the context of IDMS in extending a storage network the exchange of earth observation and other remote sensing data. The current image descriptors have been made publicly available via GENI and CloudLab and are updated along with the OS distribution packages. Finally, service containerization approaches through technologies such as Docker [13] have enabled rapid prototyping and can significantly reduce the administrative overhead of maintaining new software releases. Our DLT efforts have resulted in parallel versions of each component made available as deployable containers with supporting documentation.

3 APPLICATIONS

Having introduced the key components of our Data Logistics approach, we now describe two areas in which the DLT is being used in practice to facilitate open access to content and one ongoing effort to apply our approach in a rural, under-served community, each case having its own unique data distribution challenges.

3.1 Earth Observation Depot Network

The remote sensing community (e.g. meteorology, climate science, land and water use) is illustrative of the data access and management concerns present in the age of "Big Data" computing: the size of individual data objects tend to be large, and multi-source collections are often immense with complex associated metadata [11]. Despite widespread agreement of its value, many users and communities struggle to get timely and rapid access to the remote sensing data that they need to track ongoing changes in the physical world and to explore the impact of those changes on the human and natural systems that society depends on. In addition to the size and volume of the data sets, the users (both actual and potential) are highly distributed, geographically and socially, with widely varying degrees of access to suitable network bandwidth, and new data is constantly flowing in, so that user collections frequently need to be updated from remote sources.

To address the above challenges we have instantiated a prototype DLN known as the *Earth Observation Depot Network* (EODN).

[1] U.S. Geological Survey https://www.usgs.gov/

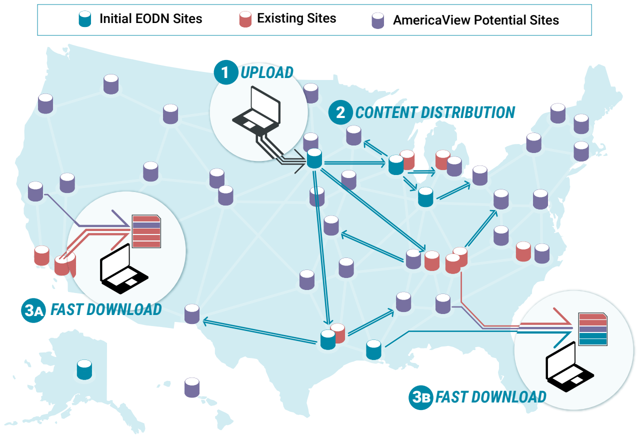

Figure 1: Map of the initial EODN illustrating three key aspects of this model DLT system. At step 1), a data source uploads remote sensing data to a depot in their local network, ensuring that the upload is fast and secure. In step 2), the DLT's IDMS server replicates this data across the EODN, first to core nodes at EODN "leadership partner" sites (in MI, WI, IN and TN), and then to "tier 3" sites at other AV locations. Steps 3a and 3b illustrate multi-stream downloads by individual users, pulling slices of a given data object from several locations at which it's replicated, including (as in 3b) a local "cache"

The EODN concept has been cultivated in collaboration with AmericaView [1], a U.S.-based remote sensing education and advocacy consortium, and we have worked closely with AmericaView community members in multiple states in scoping requirements for a storage and compute infrastructure that benefits their remote sensing science projects. At a high level, the EODN uses the building blocks of the DLT to create a content distribution and publication network that is tailored to the sharing of geospatial data (Fig. 1).

Storage depots (IBP) in EODN are hosted by participating members on a voluntary basis, while founding institutions and partner sites provide persistent, core depot capacity. System requirements may vary significantly, from some sites hosting dedicated hardware to others simply downloading and running a virtual machine or container image and allocating some available storage for EODN. This deployment model certainly relies on a benevolent notion of providing resources for a greater good, but a secondary incentive is that the local depot capacity is designed to benefit those who are geographically adjacent to the storage. Data locality increases the availability and speed of locally staged, or cached, data that is of importance to the hosting institution and its users.

3.1.1 Reducing data acquisition latency. A barrier that exists for facilitating the kinds of workflows necessary to incorporate large imagery data in many science domains is the latency between satellite observation and access to the resulting data. The latency builds at several points along the path from observation to analysis: 1) ground station processing; 2) data discovery; 3) download operations; and 4) manipulation for viewing in meteorological or other analysis software. An example of the type of remote sensing data suffering from these characteristics is Landsat 8 [12] imagery, which is available from the USGS as large "bulk data" bundles after some variable processing lead time.

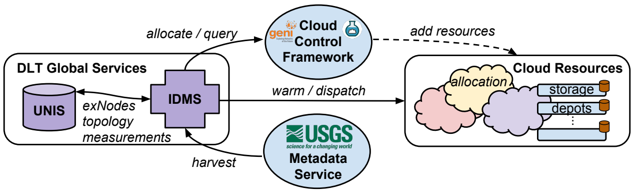

Figure 2: Interactions of DLT services in managing the upload and maintenance of remote sensing data from a source such as the USGS. When deployed alongside R&E cloud services such as GENI [4] and Cloudlab [18] the DLT is able to dynamically provision distributed storage resources to provide on-demand capacity and geographic proximity to staged data.

The EODN has directly addressed issues with acquiring such imagery in a number of ways. First, by applying specific IDMS policies for Landsat imagery of interest—over variables such as row/path, cloud cover, azimuth, etc.—EODN stages the bundled data of interest at depots near the point of computation where the eventual analysis occurs. Second, the UNIS service provides a metadata subscription mechanisms to notify EODN download agents when new data has been "harvested" and locally cached, thereby eliminating the manual interrogation of multiple data sources for new data acquisition. By eliminating steps 2 and 3 in the workflow above with automated policy-based subscriptions to new data, definition of geographic areas of interest, and multi-threaded download, EODN offers an opportunity to treat Landsat data more like readily-available weather data making it more accessible and significantly more relevant to a large scientific community.

3.1.2 Extending EODN with dynamic provisioning. To increase the impact of EODN we have made use of research and education (R&E) cloud services like GENI, and efforts under the NSFCloud program such as CloudLab [18], to host additional DLT services, in particular IBP depots. Such infrastructure allows EODN to make use of dynamic and geographically diverse hosting locations to reach communities that do not have the capacity to bring up their own instances of EODN services but still allows them to reap the benefits of relatively nearby storage. In the U.S., GENI alone provides over 60 locations, known as "aggregates", that cover much of the nation and these sites each have the ability to expose anywhere from tens of gigabytes to a few terabytes of storage through EODN.

Our approach gives IDMS the ability to intelligently request resources through the testbed's control framework, allocating storage at aggregates based on-demand at a particular geographic location if requested through a policy declaration, or perhaps if more storage is simply necessary to accommodate a burst of staged data (Fig. 2). Given that these testbeds are shared resources, the key is to ensure resources are released when no longer needed, which requires adequate monitoring of total EODN usage, replication, and usage patterns. As part of our GENI experiment efforts we have also provided a mechanism by which other users of the testbed may

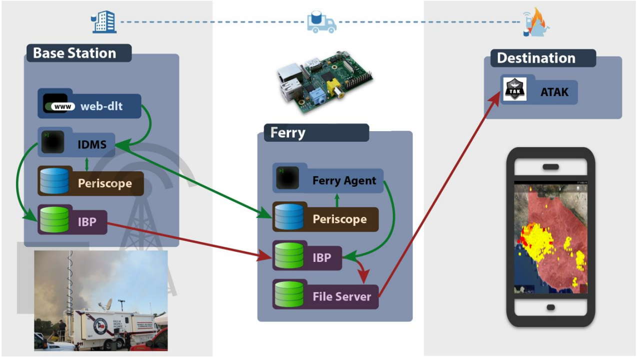

Figure 3: WildfireDLN workflow showing the relevant services and interactions between base stations, data ferries, and mobile clients.

"opt-in" to the EODN allocations and use the storage for their own experimentation when there is sufficient capacity. Finally, access to federated testbeds such as through the work being performed under Fed4FIRE+ [6] gives EODN the opportunity to expand its reach into international communities without siginificant barriers to entry.

3.2 WilfireDLN

The Wildfire Data Logistics Network (WildfireDLN) is a project that employs the DLT to improve access to data and information in the disconnected or poorly connected environment of wildfire incident operations. The work is funded through the National Institute of Standards and Technology Public Safety Communications Research (NIST-PSCR) Public Safety Innovation Program. The idea is to develop a network of DLT nodes placed at key locations to support wildfire operations information needs, such as maps and fire model outputs. A base node would sit at a fixed site, either well-connected (e.g. at a permanent facility) or poorly connected (e.g. a remote Incidence Command Center) location, while portable nodes could be deployed to satellite locations, on airborne resources, or as transportable devices called Data Ferries that work in fully disconnected environments.

Ferries are a realization of Data Logistics that fundamentally serve as mobile data buffers in a frequently disconnected networking environment. Indeed, a key aspect of applying the DLT in this context was to extend services such as IDMS and Periscope to tolerate frequent rediscovery of storage nodes that are available over certain windows of time. One of the central innovations of the DLT is that it focuses on the data buffer as a shared abstraction and the basis of inter-operation. All digital networking makes use of buffers to store data between forwarding operations, and storage and computation is also defined in terms of data stored in buffers, moving between buffers, being transformed or being maintained in some form of memory or storage. The technical insight which we seek to apply is that conventional networking seeks to hide buffers, making them implicit or automatic components of end-to-end forwarding paths. In WildfireDLN we do the opposite, giving higher layers of the service stack the option of explicitly naming and operating on the buffers directly, which are the glue out of which data sharing systems are constructed.

Using Data Ferries, data to be communicated is buffered opportunistically among available ferry nodes while a policy engine is responsible for applying higher level, user-defined distribution considerations (Fig. 3). For example, specific data sets may be of use to first responders at a given geographic location over a specific window of time. Such policy can be applied at the ferry level so that a subset of nodes destined to an appropriate location eventually stage the data selected, and further, the buffers on the ferries backing the data are only allocated for the duration required. This temporal aspect of data availability ensures that even very mobile, resource-constrained nodes can participate in moving important data without requiring manual flushing of stale data. Structural metadata tracking ferry data sets (e.g., duration, allocation size, replication count, etc.) is stored in UNIS that is maintained and refreshed by the deployed policy engine. Running on each Data Ferry is an IBP depot that is responsible for allocating, reading, writing and managing data in variable-sized chunks determined by the data distribution policy.

3.2.1 *Hardware deployments.*

The physical realization of Data Ferries is in small form factor, System-on-a-Chip (SoC)-class systems such as the Raspberry Pi. These embedded platforms provide opportunities for low cost and low power, battery-driven deployments in a multitude of scenarios and at varying levels of environmental hardening. Being physically compact, this class of ferry can be easily carried into the field and transported by numerous unmanned aerial and ground vehicular (UAV, UGV) platforms. With the compute and I/O capabilities of SoC products increasing over time there is additional opportunity to provide in-situ computation services on ferry nodes in addition to running the core DLT software stack.

From a practical standpoint, Data Ferries expose wireless connectivity to other mobile devices (e.g., first responder phones and tablets) within signal range of the integrated radios, e.g. WiFI, Bluetooth, LTE, LoRA, etc. Being mobile, ferries may move among many first responder groups during a deployment cycle, enabling access to the data they carry. In addition to providing standard data access mechanism such as an HTTP server and a Web Map Service (WMS) for tiled map data, Data Ferries have been integrated directly with mobile application software used in many incident response scenarios. As one such example, a WildfireDLN plugin for the civilian version of the Android Team Awareness Kit (ATAK) [5] has been developed and prototyped that enables a close coupling of user tools and the DLT services. Users may automatically download staged data from nearby ferries and receive geographic coordinate information about other nodes in the WildfireDLN.

A number of Data Ferry prototypes, as well as the necessary software for both command center and deployed end user applications, are being developed for use in wildfire operations, which includes environmental hardening (e.g. enclosures, antenna mounting constraints, power considerations) and simplified controls for operational use. We are coordinating with firefighting agencies to field test our approach and incorporate feedback from these experiences to further improve the WildfireDLN system.

3.3 Kenya Open CDN

Many schools in Kenya are in a position to make use of high quality local area networking, as the basic electrical and end-node infrastructure is present. However backbone connectivity is very restricted, either due to the cost or in more rural areas because infrastructure is inadequate or non-existent.

Mary Mount is an example of such a school – it is one of the highest performing secondary schools in Kenya, and it has a computer lab equipped with 18 reasonably up-to-date PCs, a PC in every classroom and overhead projectors throughout. However, the only connection in the lab is a single 3G wireless modem/router. Students can be seen huddled around PCs, viewing videos from low-resolution video streaming services such as YouTube in groups to conserve bandwidth. Mary Mount has no LAN infrastructure connecting the classrooms to the labs or to each other. The lack of adequate bandwidth rules out concurrent use of backbone connectivity by multiple uncoordinated viewers or the viewing of high bandwidth content. This in spite of the fact that the school could fund LAN infrastructure, PC graphics and projectors sufficient to support such use. Without access to content, there is little motivation for such infrastructure improvements.

Author [anonymized] has personal ties to Kenya, and contacts with schools, Universities, the ISOC chapter and Internet providers there. We have proposed applying the DLT to enhance Internet connectivity, using a satellite connection for update while integrating a DLN prototype known as the Kenya Open Content Delivery Network (KOCDN) to deliver content and services that can be supported by local infrastructure.

In the school setting, much of the content accessed by students and teachers is closely linked to a fixed curriculum, and educational publishers provide some that is used on a daily basis. Our method is to make a collection of such content available in stored but periodically updated form on a content delivery server in the school. The content server that will be the Point of Presence for the KOCDN at Mary Mount School will be a generic PC running Linux and loaded with the open source software distributed by the DLT project. The KOCDN server will be attached via Ethernet to the wireless router that serves the school's computer lab, consisting of 18 Windows PC workstations. We anticipate that the school may deploy LAN infrastructure in the later part of the project, enabling content to be used in all classrooms.

The use of a local DLT-enabled server and high quality LAN infrastructure will enable the school to make high bandwidth educational content available with high confidence, insulated against variations in service quality or availability. It will also justify the extension of that LAN infrastructure throughout the school and encourage reliance on it for the services that the local server can support with limited interactive bandwidth. When high performance backbone connectivity becomes available, Mary Mount will be in a much better position to use it, given the experience they will have acquired through this project.

4 CONCLUSIONS

This paper has outlined the motivations and current implementations behind the Data Logistics concept. As a set of deployable software packages, we have shown how the DLT can be applied to very real problems in a number of domains that require data distribution and management in challenging or non-traditional networking scenarios. We hope to expand awareness of Data Logistics and engage directly with additional communities who could benefit from the use of the managed storage and policy-driven data staging exposed through our toolkit.

While specific applications of the DLT have been highlighted the underlying technology is illustrative of the possibilities and the software product is adaptable to other domains and in other infrastructures. In that sense, the DLT can be viewed as a set of building blocks for addressing a wide range of data distribution and content management needs, suitable for running on everything from data center infrastructure to low-cost, low-power mobile devices. As future and ongoing work, we plan to work closely with users and domain scientists to more closely integrate their data worklflows with the libraries and interfaces exposed through the DLT release.

5 ACKNOWLEDGEMENTS

A part of this work (WildfireDLN) was performed under the a financial assistance award (70NANB17H174) from US Department of Commerce, National Institutes of Standards and Technology.

REFERENCES

[1] AmericaView. AmericaView Website. http://americaview.org, 2019.

[2] A. Bassi, M. Beck, and T. Moore. Mobile management of network files. In *Third Annual International Workshop on Active Middleware Services*, pages 106 – 114, August 2001.

[3] Alessandro Bassi, Micah Beck, Terry Moore, James S Plank, Martin Swany, Rich Wolski, and Graham Fagg. The internet backplane protocol: A study in resource sharing. *Future Generation Computer Systems*, 19(4):551–561, 2003.

[4] Mark Berman, Jeffrey S. Chase, Lawrence Landweber, Akihiro Nakao, Max Ott, Dipankar Raychaudhuri, Robert Ricci, and Ivan Seskar. Geni. *Comput. Netw.*, 61(C):5–23, March 2014. ISSN 1389-1286. doi: 10.1016/j.bjp.2013.12.037. URL http://dx.doi.org/10.1016/j.bjp.2013.12.037.

[5] CivTAK. Civilian Android Team Awareness Kit. https://www.takciv.org/, 2019.

[6] Isabella de A. Ceravolo, Diego G. Cardoso, Cristina K. Dominicini, Pedro Hasse, Rodolfo da S. Villaca, Moises R. N. Ribeiro, Magnos Martinello, Reza Nejabati, and Dimitra Simeonidou. O2cmf: Experiment-as-a-service for agile fed4fire deployment of programmable nfv. In *Optical Fiber Communication Conference*, page Tu3D.13. Optical Society of America, 2018. doi: 10.1364/OFC.2018.Tu3D.13. URL http://www.osapublishing.org/abstract.cfm?URI=OFC-2018-Tu3D.13.

[7] Ahmed El-Hassany, Ezra Kissel, Dan Gunter, and Martin Swany. Design and implementation of a Unified Network Information Service. In *10th IEEE International Conference on Services Computing (SCC 2013)*, 2013.

[8] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 2017. doi: 10.1016/j.rse.2017.06.031. URL https://doi.org/10.1016/j.rse.2017.06.031.

[9] A. Hanemann, J. Boote, E. Boyd, J. Durand, L. Kudarimoti, R. Lapacz, M. Swany, S. Trocha, and J. Zurawski. PerfSONAR: A service oriented architecture for multi-domain network monitoring. In *In Proceedings of the Third International Conference on Service Oriented Computing (ICSOC 2005)*, ACM Sigsoft and Sigweb, pages 241–254, December 2005.

[10] Ezra Kissel, Martin Swany, and Aaron Brown. Phoebus: A system for high throughput data movement. *J. Parallel Distrib. Comput.*, 71:266–279, February 2011. ISSN 0743-7315. doi: http://dx.doi.org/10.1016/j.jpdc.2010.08.011. URL http://dx.doi.org/10.1016/j.jpdc.2010.08.011.

[11] Yan Ma, Haiping Wu, Lizhe Wang, Bormin Huang, Rajiv Ranjan, Albert Zomaya, and Wei Jie. Remote sensing big data computing. *Future Gener. Comput. Syst.*, 51(C):47–60, October 2015. ISSN 0167-739X. doi: 10.1016/j.future.2014.10.029. URL http://dx.doi.org/10.1016/j.future.2014.10.029.

[12] Richardson Leslie Koontz S.R. Loomis John Miller, H.M. and 2013 Koontz, Lynne. Users, uses, and value of landsat satellite imagery—results from the 2012 survey of users. U.S. Geological Survey Open-File Report 2013–1269, 51 p., http://dx.doi.org/10.3133/ofr/20131269.

[13] Christopher Negus. *Docker Containers.* Addison-Wesley Professional, 2nd edition, 2015. ISBN 9780134397511.

[14] Erik Nygren, Ramesh K Sitaraman, and Jennifer Sun. The Akamai network. *ACM SIGOPS Operating Systems Review*, 44(3):2–19, August 2010.

[15] J. S. Plank, M. Beck, W. Elwasif, T. Moore, M. Swany, and R. Wolski. The Internet Backplane Protocol: Storage in the network. In *NetStore'99: Network Storage Symposium*. Internet2, http://dsi.internet2.edu/netstore99, October 1999.

[16] J. S. Plank, S. Atchley, Y. Ding, and M. Beck. Algorithms for high performance, wide-area distributed file downloads. *Parallel Processing Letters*, 13(2):207–224, June 2003.

[17] Johan Pouwelse, Paweł Garbacki, Dick Epema, and Henk Sips. The Bittorrent P2P File-Sharing System: Measurements and Analysis. In *Peer-to-Peer Systems IV*, pages 205–216. Lecture Notes in Computer Science, Berlin, Heidelberg, 2005.

[18] Robert Ricci, Eric Eide, and The CloudLab Team. Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications. *USENIX ;login:*, 39(6), December 2014. URL https://www.usenix.org/publications/login/dec14/ricci.

[19] A Vakali and G Pallis. Content delivery networks: Status and trends. *IEEE Internet Computing*, 7(6):68–74, November 2003.