Data Movement Categories

Chin Fang
Zettar Inc., Mountain View, California, USA

R. "Les" A. Cottrell
SLAC National Accelerator Laboratory
2575 Sand Hill Road, Menlo Park, California, USA

Abstract -- We have endeavored to classify the commonly seen data movement needs, as observed in data-intensive institutions (both commercial and non-profit), into four categories. Knowing how to map a data movement task into one of the four categories helps select proper data mover tools. For each category, how the data storage is involved, high-level examples and the nature of typical solutions are described. Finally, some general remarks are provided to help further orient readers new to this field - the 4th IT pillar[1].

Keywords -- data storage; intrinsically scale-out; data movements; firewall; data compression; software is hard; software lifetime.

SLAC National Accelerator Laboratory, Stanford University, Stanford, CA 94309

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-76SF00515

[1] "High Performance Data Transfer for Distributed Data Intensive Sciences". https://slac.stanford.edu/pubs/slactns/tn05/slac-tn-16-001.pdf. Accessed Dec. 15, 2020.

Table of Contents

Introduction **3**

1. Completely automated data movements **4**
    - Data storage 4
    - Examples 4
    - Solutions 4

2. Partially automated data movements, end user controlled **4**
    - Data storage 4
    - Examples 5
    - Solutions 5

3. Partially automated data movements, but controlled by professionals **6**
    - Data storage 6
    - Examples 6
    - Solutions 6

4. Strictly interactive data movements and end user oriented **6**
    - Data storage 6
    - Examples 7
    - Solutions 7
        - An example: multiple individual collaborators with a repository 7
        - General comments 7
            - GUI is not equal to user friendliness 7
            - Watch out for the firewall's negative impact! 8
            - Compression is not the panacea to gain high transfer speed! 8
            - Big brand and large group are not the same as quality 8
            - Respect a successful software tool 8
            - Machine-generated data is the dominant modern factor to consider 8
            - Achieving high data movement rates is non-trivial 9
            - Like any engineering product, software has a life-time too 9
            - Using the right tool for the job 9

Acknowledgments **9**

References **10**

Introduction

Analysts forecast that by 2025 data will exponentially grow by 10 times according to Intel DCG EVO Navin Shenoy[2]. Highlighted by a number of shocking statistics, e.g. only about 1% of the data is utilized and acted upon, this exponential data growth has caused many new data management problems. In addition, the following few key reasons further hinders many parties from tackling data movements effectively.

1. Many people extrapolate their end user experience to the more demanding scenarios.
2. Almost all vendors create "specialized tools" for "different" problems (*e.g. large scale backup, replication, migration*), causing users to think this is the way it is. However, logically, problems in categories 1, 2, and 3 below are all large-scale data movement problems. Therefore, in principle, they could be solved in a unified manner.
3. Data movement tools have not been viewed as a critical area by most IT people. As a result, they are seldom aware of the advances in this space.

All such reasons result in often disappointing results or costly failures.

The purpose of this monograph is to classify the data movement needs in many distributed data-intensive institutions (*both commercial and non-profit*) into the following categories:

1 Completely automated data movements
2 Partially automated data movements, end user controlled
3 Partially automated data movements, controlled by professionals
4 Strictly interactive data movements and end user oriented

Knowing how to map a data movement task into one of the four categories helps select proper data mover tools. In passing we note that *describing and comparing different data mover products or related technologies are completely outside of the scope and not the purpose of this monograph*. Before proceeding, it should be of interest to note that attempting to use solutions for categories **1, 2,** and **3** for category **4** is like trying to use a Kenworth T800[3] as a Fiat 500[4] - i.e. a mismatch. Similarly, trying to stretch a solution designed for **2** and **3** for **1** is unlikely to be fruitful. The following explains why.

FIG. 1 A Kenworth T800 pulling a fuel tank (picture credit: Kenworth)

FIG. 2 A Fiat 2019 500 model (picture credit: Fiat S.P.A)

[2] "Innovating for the 'Data-Centric' Era". https://newsroom.intel.com/editorials/data-centric-innovation-summit/. Accessed Dec. 8, 2020.
[3] "Kenworth T800, The Ultimate Workhorse". https://www.kenworth.com/trucks/t800/. Accessed Dec. 8, 2020.
[4] "2019 Fiat 500". https://www.fiatusa.com/2019/500-low-inventory.html. Accessed Dec. 8, 2020.

1. Completely automated data movements

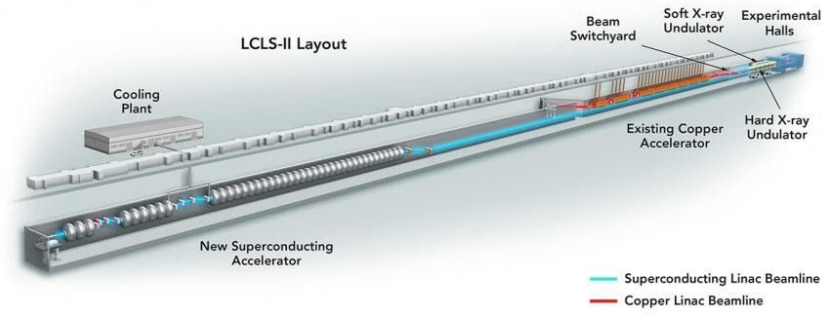


FIG. 4 The LCLS-II design (picture credit: SLAC National Accelerator Laboratory

FIG. 3 SKA mid-Africa (picture credit: skatelescope.org)

Data storage

Data is always stored in institutional storage. The required data volumes and data transfer rates are so large that only completely automated data movements are feasible.

Examples

The Square Kilometer Array (SKA) [5] and LCLS-II[6] are two excellent real-world examples. For such use cases, creating a custom and robust automation system is mandatory. Such a task needs highly skilled professionals. The progress and status of transfers are always monitored using professionally designed and implemented monitoring systems (e.g. with Prometheus[7] + Grafana[8]).

Solutions

The data mover employed for such scenarios is extremely challenging to design and implement. To the best of our knowledge, there are very few choices in the world which can do this kind of job - none of the well-known commercial data mover solutions are capable of meeting such levels of requirements. The data mover also needs to provide many high-end features not found in end-user class products, such as high-availability, intrinsically scale-out, check-pointing, API, data integrity assurance, parallel encryption, append-streaming, very high efficiency etc. Such data movers always run as daemons in clusters, never a single Command Line Input (CLI) or a Graphical User Input (GUI) based utility operated by end-users.

2. Partially automated data movements, end user controlled

Data storage

Data is always stored in institutional storage. There are surprisingly many use cases in this category. In the U.S. DOE national lab circle, this is called "Third Party Mode". Please see this U.S. DOE ESnet technical report: "mdtmFTP and Its Evaluation on ESNET SDN Testbed"[9] subsection **3.3 c**. In such scenarios, data is always

[5] "Square Kilometer Array". https://www.skatelescope.org/. Accessed Dec. 9, 2020.
[6] "LCLS-II Design and Performance". https://lcls.slac.stanford.edu/lcls-ii/design-and-performance. Accessed Dec. 9, 2020.
[7] "From metrics to insight". https://prometheus.io/. Accessed Dec. 9, 2020.
[8] "The open observability platform". https://grafana.com/. Accessed Dec. 9, 2020
[9] "mdtmFTP and Its Evaluation on ESNET SDN Testbed". https://mdtm.fnal.gov/downloads/mdtmFTP.pdf. Accessed Dec. 9, 2020.

moved among institutional storage (*possible cross institutions*), but a movement is initiated by end users. This is a case where a Data Transfer Portal (DTP) approach is an excellent fit. A DTP typically is designed and implemented as a Web portal application. It front-ends both the employed data mover tools and the involved storage services. It is an application of the Fundamental Theorem of Software Engineering[10], mainly covering the following:

1. User login to the web portal, likely authenticated and authorized using the institution's directory service[11] (e.g. LDAP[12])
2. List both local and remote files and/or objects in the familiar file/folder structure with last modify date/time, owner, group, and permission
3. Schedule data movement jobs (one time or periodically) from local to remote and vice versa using a data mover tool that's most appropriate for the job
4. List pending and finished jobs
5. Restore selected file/folder from remote to local
6. Download (optionally compressed) selected remote files/objects.

FIG. 5 an actual genome sequencer farm (picture credit: Genome Research Limited)

Examples

A researcher moves the output of a genome sequencer farm's data to the local imported storage, to be collected via more automated means later on. See **FIG. 5.** and **6**.

Solutions

There have been a reasonable number of choices (both free and commercial, but quality varies!) for data movers employed for such scenarios. In all cases, it's much better to front end these data movers with a DTP so as to make users' life easier and reduce the support burden of IT admins.

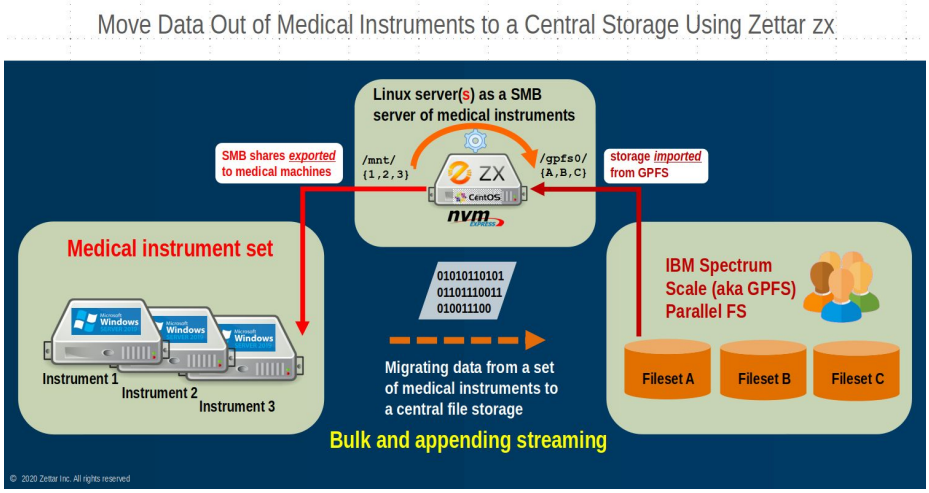
A well designed DTP improves security and workflow progress. For example, it eliminates the need to grant access to the servers running data movers to users. It also allows easy task management and scheduling by end users. A DTP also eliminates the need to include elaborate user authentication and authorization functionalities in the data mover. Note that for such use cases, data movers can be running either as daemons or CLI tools. Daemons in general offer the potential of much better efficiency than CLI tools.

Designing and implementing a generic DTP is highly unlikely to succeed. The needs vary from institution to institution. It's almost impossible for an unrelated third party to come up with something to meet site-specific needs. A reasonable approach is to find a project on github[13] and then customize it for an institution's particular needs.

---

[10] "Fundamental theorem of software engineering". https://en.wikipedia.org/wiki/Fundamental_theorem_of_software_engineering. Accessed Dec. 16, 2020.
[11] "Directory service". https://en.wikipedia.org/wiki/Directory_service. Accessed Dec. 15, 2020.
[12] "Lightweight Directory Access Protocol". https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol. Accessed Dec. 15, 2020
[13] "GitHub: Where the world builds software". https://github.com/. Accessed Dec. 9, 2020.

3. Partially automated data movements, but controlled by professionals



Data storage

Data is always stored in institutional storage. e.g. Network Attached Storage (NAS) or cluster file systems - This is also a large category!

FIG. 6 A high-level view on how to migrate data generated by instruments in a biomedical research lab to a parallel file system. Data is owned by multiple users

All data-intensive enterprises must execute tasks in this category periodically, e.g. for tech refresh, storage capacity balancing, file system migration. In such scenarios, data is always moved among institutional storage (possible cross institutions too), but movements are initiated by IT professionals. The scale of the data movement, the operation complexities involved, and the duration almost always make some kind of custom automation necessary.

Examples

Collecting data from instruments in a biomedical research lab as illustrated in **FIG. 6**.

Solutions

There has been a reasonable number of choices (both free and commercial, but quality varies!) for data movers employed for such scenarios. For such use cases, data movers can be running either as daemons or CLI tools. Furthermore, due to the need to preserve ownership and permissions of user data, the data mover employed usually may need to run under the `root` user.

4. Strictly interactive data movements and end user oriented

Data storage

Data is stored in institutional storage and personal computing devices. *This is where many people get confused and attempt to use data movers designed for categories 1, 2, and 3!* Note that even this may "seem to be" the most "visible" category, but when IoT devices are considered, the category is not really the most "active" category[14]. Categories consisting of moving machine generated data tend to be far more "active", contributing to most of the world's data growth.

[14] "Leading the IoT. Gartner Insights on How to Lead in a Connected World". https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf. Accessed on Dec. 14, 2020.

Examples

End users wish to download or send data to a "data repository" (*which should always be operated and managed by professional IT people*). A realistic case: the nurses and doctors in a hospital send collected patient information to the hospital's central data repository or another collaboration institution's data repository.

Solutions

From the 1990s to today, there have been numerous solutions available, both free and commercial, for various use cases. Owing to space limitations, only the solutions for a popular one - individual collaborators need to exchange data with a central data repository - are illustrated below. Other than this one, many other use cases also exist that may require different solutions. But for almost every known use case, there are many choices as far as we know.

An example: multiple individual collaborators with a repository

For this type of need, using a Web portal to front end the data repository is a popular approach. The portal provides the following:

- Create, delete, rename, preview, and edit files and folders.
- Upload and download files and folders.
- Create multiple users with their own directories. Each user can have a distinctive directory to keep his or her data.
- We can use it either as a standalone application or a middleware.
- Web-based.
- Cross-platform. Works well on GNU/Linux, Windows and Mac OS X.

FIGs 7, **8**, and **9** show what is possible for such use cases. It is done with freeware.

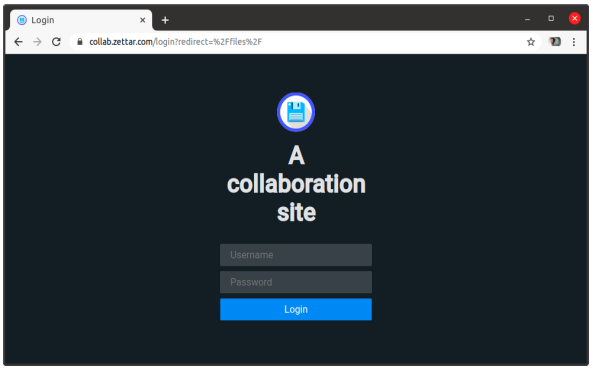


FIG. 7 The login page

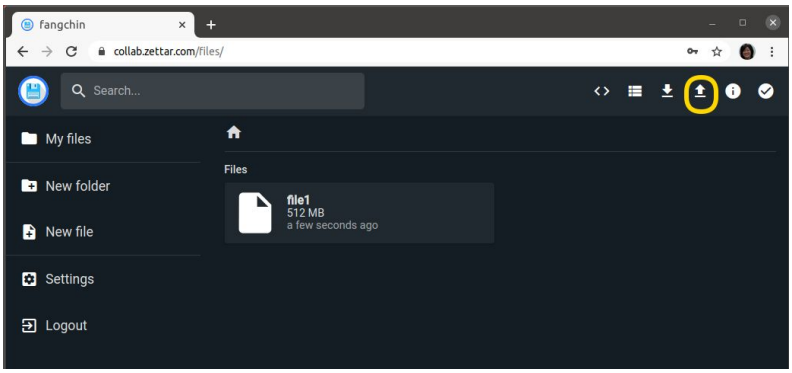


FIG. 8 Upload a 512 MiB file. Note the upload up-arrow

General comments

GUI is not equal to user friendliness

The majority of end users, even degreed researchers but whose specialization does not involve IT, tends to equate the existence of a fancy-looking GUI or the ability to use Web browsers to manage data movement tasks as “ease of use”. In fact, for categories **1**, **2**, and **3**, the ease of automating the data mover, e.g. with its REST APIs, actually is far more important than having the availability of a UI! Often, depending on the user’s computer literacy, a well-designed and implemented CLI tool can be “easier to use” than one that comes with a confusing UI!

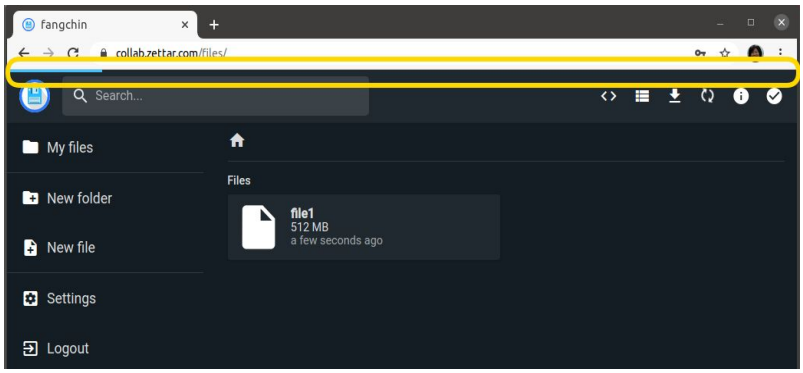


FIG. 9 Uploading a directory that consists of a 10GB file. Note the progress bar

Watch out for the firewall's negative impact!

Firewalls, even these running on personal computing devices, can be an invisible hindrance of a satisfactory data movement experience. This is the reason why Eli Dart, U.S. DOE Energy Sciences Network, embarked on the Science DMZ approach in 2010[15]. So, if your data movement doesn't seem to be fast enough, instead of blaming the data mover application immediately - as most end users would do - check the firewall settings first (*although this is probably beyond most of end users' capabilities*).

Compression is not the panacea to gain high transfer speed!

Many people think that with compression, even in low bandwidth situations, e.g. WiFi, they could obtain "much faster" transfer rates. Data compression actually is a very vast field full of different algorithms, which may do better or worse depending on the nature of data. For example, pictures and videos taken by an end user device such as a mobile phone are not amenable to compression. Implementations also differ, even for the same algorithm, and thus speed may differ. But all compression operations take CPU cycles and time. Some algorithms also work well only if there is enough computer memory. It's definitely not as simple as "if compression is used, we can move data faster". Not always true - you may see slower transfer rates!

Big brand and large group are not the same as quality

The widely used ESnet network test tool, `iperf3`[16], is actually maintained by a single ESnet software engineer Bruce Mah[17]. Likewise, when choosing data mover tools, the brand name and size of a vendor should not take the highest priority. In fact, all major U.S. DOE funded heavy duty data movers are created and maintained by small teams with just a few engineers.

Respect a successful software tool

The thought "Oh, just have someone code it up" is common! Please keep in mind what the world-renowned computer scientist, Dr. Donald Knuth, Professor Emeritus at Stanford University, stated: "*The most important lesson I learned during the past nine years is that* ***software is hard****; and it takes a long time. From now on I shall have significantly greater respect for every successful software tool that I encounter. …* ***The amount of technical detail in a large system is one thing that makes programming more demanding than book-writing****. Another is that programming demands a significantly higher standard of accuracy.* ***Programs don't simply have to make sense to another human being, they must make sense to a computer.***"[18]

Machine-generated data is the dominant modern factor to consider

As an example, the data generated from a single week-long experiment of LCLS-II is the same amount of data produced by a person taking six photos per minute on a cell phone, where each photo is around 700KB in size, for 1.6 million man-days. Transferring a few GB or even TB worth of data is really not that demanding. In the

---

[15] "ESnet's Science DMZ Breaking Down Data Barriers, Speeding up Science". https://www.es.net/about/esnet-history/esnets-science-dmz-breaking-down-data-barriers-speeding-up-science/. Accessed Nov. 8, 2020.
[16] "iperf2/iperf3". https://fasterdata.es.net/performance-testing/network-troubleshooting-tools/iperf/. Accessed Dec. 10, 2020.
[17] "Bruce Mah". https://www.es.net/about/esnet-staff/software-engineering/bruce-mah/. Accessed Dec. 10, 2020.
[18] Knuth, D.E. Theory and Practice (Report No. STAN-CS-89-1284). Palo Alto, CA: Computer Science Department, Stanford University, 1989.

modern digital age, in terms of contribution to the world's data growth and as causes of data movements, end-users are insignificant. Today, machine-generated data is the dominant factor.

Achieving high data movement rates is non-trivial

To better understand the challenges posed by moving data at speed and scale, a water transport system analogy[19] is very helpful. It is of interest to note that at the time of this writing, to attain a 20+Gbps point-to-point data rate is still considered non-trivial. Most parties achieve high data movement rates by aggregating independent data traffic with network traffic routing, e.g. Google's B4 project[20]. A major goal of B4 is to maximize the bandwidth utilization of the links connecting various Google data centers. It should be evident that in each data center, there are numerous data source and destination "sets" that are independent from each other. The data traffic associated with such source and destination sets are routed intelligently over selected network links operated by Google. This is very different from a point-to-point data transport solution does, dealing with single source and single destination, for example: the fast feedback storage of LCLS-II to the Lustre storage at NERSC[21].

Like any engineering product, software has a life-time too

Creating software is both an art and engineering. The engineering part makes the lifetime of a piece of software finite if not actively maintained. In general, every 10 - 20 years, for most software if there is not a major overhaul of the fundamental architecture, the risk of not matching the newer demand and hardware advances becomes higher and higher. So, the longer a piece of software has been in existence, the more important to know whether it has undergone any major overhaul. GUI improvements, minor polishes, changing product names based on marketing considerations, like naming a product something202X, do not count. For example, the well-known scientific software BLAS (Basic Linear Algebra Subprograms)[22] has been around since 1979. But, the algorithms that the library uses are often researched and enhanced. Furthermore, its uses are often done via BLAS-compatible but newer libraries such as LAPACK[23]. So BLAS' long life-time is not an exception but actually an excellent example of the above software lifetime remarks.

Using the right tool for the job

It's understandable that not everyone is interested in learning computing in-depth. Nevertheless, in this digital age, data and its appropriate processing are important. So, the saying "Using the right tool for the job" applies here too. Depending on what you wish to do, spending some effort to learn about the appropriate data mover(s) that are designed for the data movement categories is strongly recommended. A bit of such "up-front" investment enhances your efficiency, productivity, and reduces stresses and wasted time.

Acknowledgments

We would like to give our thanks to two colleagues at SLAC National Accelerator Laboratory: Dr. Amedeo Parrazo, for his support and encouragement; Dr. Wilko Kroeger for discussions about the content and helpful comments. In addition, Dr. Ezra Kissel, ESnet, for his valuable critiques and thoughtful suggestions; Andrey O Kudryavtsev, Intel NSG, for his insights about using storage.

---

[19] Learn the water transport analogy. https://youtube.com/watch?v=Ve1mIOSVsUY. Accessed December 19, 2020.
[20] Sushant Jain et. al., B4: Experience with a Globally-Deployed Software Defined WAN. https://research.google/pubs/pub41761/. Accessed Dec. 19, 2020
[21] Jana Thayer, Data Processing at the Linac Coherent Light Source, https://docs.google.com/presentation/d/1QuEV3ShXPwV7KowgGG8qC-BRDFz6hREwhtRE8a5_fBY/. Accessed Dec. 19, 2020.
[22] BLAS (Basic Linear Algebra Subprograms). http://www.netlib.org/blas/. Accessed Dec. 21, 2020.
[23] LAPACK. http://www.netlib.org/lapack/. Accessed Dec. 21m 2020.

References

1. Chin Fang, R. "Les" A. Cottrell, Andrew B. Hanushevsky, Wilko Kroeger, Wei Yang, High Performance Data Transfer for Distributed Data Intensive Sciences". SLAC Technical Notes TN-16-001. https://slac.stanford.edu/pubs/slactns/tn05/slac-tn-16-001.pdf. Accessed Dec. 15, 2020.
2. Naveen Shenoy, Innovation for the 'data-centric' era, August 8, 2018. https://newsroom.intel.com/editorials/data-centric-innovation-summit/ Accessed Dec. 8, 2020.
3. Kenworth T800, The Ultimate Workhorse. https://www.kenworth.com/trucks/t800/. Accessed Dec. 8, 2020.
4. 2019 Fiat 500. https://www.fiatusa.com/2019/500-low-inventory.html. Accessed Dec. 8, 2020.
5. Square Kilometer Array. https://www.skatelescope.org/. Accessed Dec. 9, 2020.
6. LCLS-II Design & Performance, Linac Coherent Light Source, SLAC National Accelerator Laboratory. https://lcls.slac.stanford.edu/lcls-ii/design-and-performance. Accessed Dec. 9, 2020.
7. From metrics to insight. https://prometheus.io/. Accessed Dec. 9, 2020.
8. The open observability platform. https://grafana.com/. Accessed Dec. 9, 2020
9. Liang Zhang, Wenji Wu, Phil DeMar, Eric Pouyoul, mdtmFTP and Its Evaluation on ESNET SDN Testbed". https://mdtm.fnal.gov/downloads/mdtmFTP.pdf. Accessed Dec. 9, 2020.
10. Fundamental theorem of software engineering. https://en.wikipedia.org/wiki/Fundamental_theorem_of_software_engineering. Accessed Dec. 16, 2020.
11. Directory service. https://en.wikipedia.org/wiki/Directory_service. Accessed Dec. 15, 2020.
12. Lightweight Directory Access Protocol. https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol. Accessed Dec. 15, 2020
13. GitHub: Where the world builds software. https://github.com/. Accessed Dec. 9, 2020.
14. Mark Hung, Leading the IoT. Gartner Insights on How to Lead in a Connected World. https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf. Accessed on Dec. 14, 2020.
15. ESnet's Science DMZ Breaking Down Data Barriers, Speeding up Science. https://www.es.net/about/esnet-history/esnets-science-dmz-breaking-down-data-barriers-speeding-up-science/. Accessed Nov. 8, 2020.
16. iperf2/iperf3. https://fasterdata.es.net/performance-testing/network-troubleshooting-tools/iperf/. Accessed Dec. 10, 2020.
17. Bruce Mah. https://www.es.net/about/esnet-staff/software-engineering/bruce-mah/. Accessed Dec. 10, 2020.
18. Knuth, D.E. Theory and Practice (Report No. STAN-CS-89-1284). Palo Alto, CA: Computer Science Department, Stanford University, 1989.
19. Learn the water transport analogy. https://youtube.com/watch?v=Ve1mIOSVsUY. Accessed December 19, 2020.
20. Sushant Jain et. al., B4: Experience with a Globally-Deployed Software Defined WAN. https://research.google/pubs/pub41761/. Accessed Dec. 19, 2020
21. Jana Thayer, Data Processing at the Linac Coherent Light Source, https://docs.google.com/presentation/d/1QuEV3ShXPwV7KowgGG8qC-BRDFz6hREwhtRE8a5_fBY/. Accessed Dec. 19, 2020.
22. BLAS (Basic Linear Algebra Subprograms). http://www.netlib.org/blas/. Accessed Dec. 21, 2020.
23. LAPACK. http://www.netlib.org/lapack/. Accessed Dec. 21m 2020.