Possibilities for Named Data Networking in HEP

Duncan Rand, Simon Fayer and David J. Colling

Imperial College London, Blackett Laboratory, London, SW7 2AZ, UK

E-mail: duncan.rand@imperial.ac.uk

Abstract. Named Data Networking is a novel networking architecture which places emphasis on the naming and signing of data. Once so named, the location of data sources becomes less important and the emphasis moves from host to host transfers to pulling data from the network. Furthermore data are cached en route to their destination. We believe this approach has interesting possibilities for High Energy Physics (HEP) and report on work we have done in this area including the development of a scalable repository, a ROOT plugin and a small local testbed, the submission of jobs to a grid cluster and some ideas on an authentication system for LHC VOs.

1. Introduction

The Large Hadron Collider (LHC) at CERN is currently the world's largest scientific endeavour. Data volumes at the LHC are already large and are set to grow. For example, the CMS data set is currently of the order of tens of petabytes in size and, with the switching on again of the LHC in 2015, is likely to increase to hundreds of petabytes over the next decades. Similarly, data volumes in other data-centric sciences such as bioinformatics, climate science and geophysics are also likely to increase in the future. The Square Kilometer Array, which should come into operation in 2024, will produce considerably more data than the LHC. So, overall, the trend in data volumes in big science is likely to increase significantly and this will require efficient and effective techniques for the transferring of data. It is not clear that the existing networking solutions based around TCP/IP will be able to scale to the levels of throughput required. More intelligent and advanced networking techniques, such as the pervasive and automatic caching of popular data, are likely to be needed in order to reduce demands on network infrastructure.

LHC experiment computing sites such as the Tier-2 in the HEP Group at Imperial College, with a 4000 core compute cluster and approximately 3 petabytes of disk storage act, in effect, as large caches. Currently there are two main methods whereby LHC experiment data is transferred to such sites for processing. The first is data pre-placement using data management coordinators such as PhEDEx for the CMS experiment and Rucio for ATLAS. Data expected to be popular for analysis is pre-selected for caching at the site. The pre-selection process is not perfect however and it is likely that much data sent out to sites remains unread before deletion. A refinement of this approach is for datasets found to be popular to be distributed in a more dynamic manner, but again it is unknown for how long they are likely to remain popular once transferred. Space at sites has to be carefully monitored and maintained and any unwanted data deleted in a controlled and coordinated manner. The second main method is the direct reading of data by a job from remotely located storage using the XRootD protocol (i.e. AAA (Any data, Any time, Anywhere) for the CMS experiment and FAX for ATLAS). Whilst this removes the requirement to estimate data popularity in advance, the data are not cached (although steps are being taken in this area) and such data, once read, are deleted at the end of the job and therefore unavailable for future reading at that site. Overall, these two approaches have evolved into a complex overlay of software into which considerable effort goes both in monitoring and in support to ensure reliable operation. It would be sensible to explore any possibly simpler methods for the distribution of data that might be applicable. For example, there is one conceptually similar to the remote accessing of data from federated storage, in that the prediction of the popularity of data in advance is not attempted, but in which data, once it has actually been transferred, is then cached at the site. That approach is Named Data Networking (NDN) and we think it offers interesting possibilities for HEP.

2. Named Data Networking

Named Data Networking is a one of five projects funded under the United States National Science Foundation's (NSF) Future Internet Architecture Program [1]. Its aim is to develop a novel Internet architecture centred around data or content rather than host to host communication. Data are secured by requiring producers to cryptographically sign each data packet. Access to confidential data can also be controlled through the use of encryption. Rather than data being transferred between two hosts, data are retrieved out of the network by requesting them by name. Thus in NDN content is location-agnostic, once named and signed it loses any association with a host or location. An NDN network consists of a number of nodes which communicate with each other via intermediate routers as in the existing Internet. Each data block is addressed by a unique name which consists of a hierarchical path, a name and attributes, all separated by "/". An example of a valid address could be "/ndn/uk/ac/imperial/ph/hep/data/somefile/1" or "/grid/cms/user/ndn-testing/file1". Crucially, as the data packets traverse the network they are cached in routers. It is this caching of data which should reduce read latencies and overall network usage in a large distributed computing project such as an LHC virtual organisation (VO).

2.1. Data transfer in NDN

The reader is referred to [1] for a detailed description of NDN. Briefly, a consumer, interested in a named segment of data, sends out a request to the network for that data in the form of an 'interest packet'. As the interest packet is routed towards a copy of the data, a check is made to see if that router has a copy of the segment in its Content Store (cache). If it does it serves the data back to the interface that the interest packet came from. If not, it is stored in the NDN router's Pending Interest Table (PIT) and the interest packet is forwarded until it reaches a copy. The resulting collection of PIT entries acts as a 'trail of crumbs' for the data packet to follow back to the consumer. If two consumers request the same data segment from a router only one interest packet is forwarded. The forwarding strategy is programmable, e.g. it is possible to send interest packets to multiple routers, if one route fails the data can be served by the second one, thereby building in reliability.

2.2. Named Data Networking realisation

There is a collaboration that has been set up to develop and promote NDN (`http://named-data.net`). Quoting from the website, the NDN collaboration is "A Collaborative Effort to Promote and Sustain the NDN Future Internet Architecture" and "It aims to provide a practically deployable set of protocols replacing TCP/IP that increases network trustworthiness and security, addresses the growing bandwidth requirements of modern content, and simplifies the creation of sophisticated distributed applications." The NDN collaboration produce the NDN platform with the ndn-cxx library and Named Data Networking Forwarding Daemon (NFD).

The NFD handles NDN packet transfers and acts as an NDN router. The software is available for Ubuntu and MacOSX and we have also built NFD RPMs for CentOS7, our preferred distribution.

3. NDN scalable repository: 'repo-se'

Along with data caching in routers NDN has the concept of a more permanent store known as a repository. We have written an application in C++ to provide repository services called repo-se (we pronounce it 'repose'). The software consists of server and client.

3.1. Server

The server application links against the lib-ndncxx library to connect it to NDN and to backend filesystem libraries librados (part of Ceph) and libcurl. The backends allow files to be served into the NDN namespace from either a conventional POSIX filesystem, Ceph or HTTP source. The repo-se server has been designed with scalability in mind.

3.2. Client

A small static client library, librepoclient, is also included within the repose codebase which provides a minimal POSIX-like interface (open(), read(), close(), ...) and is designed to be linked into shared library plugins for other applications such as ROOT and GFAL2. There is also a minimal "getfile" application for testing, which fetches an entire file from NDN and writes it to the local disk using the client library. The current implementation is limited to read-only access for basic demonstrations and testing. Once a plan for authentication is finalised the repository software can then be extended to include full remote I/O (i.e. read-write). The original version of the client requested and retrieved data segments one at a time. We are currently working on improving the client to send out multiple interest packets simultaneously. One critical aspect of this appears to be the speed of signing and verifying data packets.

4. Imperial College HEP NDN Testbed

We have installed an NDN testbed running the NFD software and comprising of a User Interface host, two NDN routers, two repositories running 'repo-se' for data storage and a Ceph storage cluster with four 16 TB servers. At the momment the routing paths are added in manually. In the future we intend to install the NDN route advertisement software NLSR. As an example of operation a client on the UI might request data segments from a repo-se instance by sending out interest packets to the intermediate routers. The passage of the interest packets and returning data packets can be monitored using the 'ndndump' software (`https://github.com/zhenkai/ndndump`). We are currently using the testbed to build up experience with NDN.

5. ROOT plugin

ROOT (`https://root.cern.ch`) is an integral part of most HEP experiment analysis software. We have coded up a prototype NDN ROOT plugin using C++. On the NDN testbed we are now able to open a ROOT file stored in our Ceph cluster from the NDN UI via an NDN router although this access is currently unoptimised and rather slow.

6. Grid Testing

The testbed NFD software is built for the CentOS7 operating system but the production grid cluster worker nodes run CentOS6. We have built the NFD software for CentOS6 in order to test staging of files to worker nodes. We have submitted grid jobs to our production cluster which include a tarball of the NDN software in the input sandbox. Once running, the job unpacked the tar file and started up the NFD daemon on the worker node. A route to the local NDN router was added. Files were staged to the worker node scratch area using the client from the repo-ng repository (`https://github.com/named-data/repo-ng`). This had the added effect of demonstrating inter-operability between another NDN repository client and our NDN repository software. In future it is planned to use a CVMFS software area to store the NDN daemon files. Once transfer performance has been improved it is planned to test the staging of such files at scale with a well-resourced host acting as an NDN core router. It is also planned to test the reading and caching of ROOT files over longer distances.

7. Possible use with an LHC VO

The main use case envisaged is chaotic user analysis where it is difficult to predict what data is to be read. A version of the experiment's analysis code would be built with the NDN ROOT plugin. VO datasets could be encrypted using an individual symmetric key per dataset. Accessing the data would involve two steps, the client would request a) the encrypted data and b) the encryption key from a DataSet Key Server (DSKS) using a signed interest packet. The key server would check the user's membership of the VO and if valid would encrypt the dataset key with the user's public key and send it back. On receipt of the encrypted key and data packets the application would decrypt the key with user's private key and use that to access the data.

8. Scaleable routers

NDN routers mentioned in the literature discuss the caching of data in memory. With LHC experiment ROOT files roughly 2 GB in size and increasing, the viability of such in-memory caching is questionable. We envisage the need to build a scalable router similar to our scalable repository. The router would likely have several layers of caching including Memcached, SSD and Ceph base storage. Indeed it is not unlikely that this will evolve to a hybrid router-repository at experiment sites.

9. Conclusion

We feel it makes sense to examine the viability for HEP of this new networking paradigm. Clearly it is early days, there are many unsolved issues both in Named Data Networking itself and in its application to the distribution of High Energy Physics data. We have made a start by coding-up a scalable repository and ROOT plugin. We are building our experience with Named Data Networking on our local testbed. We would like to expand this to include other interested sites.

Acknowledgments

The authors would like to acknowledge the support provided by GridPP.

References

[1] Zhang L, Afanasyev A, Burke J, Jacobson V, claffy kc, Crowley P, Papadopoulos C, Wang L and Zhang B 2014 Named data networking, *SIGCOMM Comput. Commun. Rev.* **44** 66-73