EVOLVING JASMIN: HIGH PERFORMANCE ANALYSIS AND THE DATA DELUGE

Neil Massey[1], Philip Kershaw[1], Matt Pritchard[1], Matt Pryor[1], Sam Pepler[1], Jonathan Churchill[2] and Bryan Lawrence[3]

1. Centre for Environmental Data Analysis, Science and Technology Facilities Council, Rutherford Appleton Laboratory
2. Scientific Computing Department, Science and Technology Facilities Council, Rutherford Appleton Laboratory
3. National Centre for Atmospheric Science, University of Reading

ABSTRACT

JASMIN is a highly successful data analysis system, which is used by thousands of academics and their industrial partners to analyse many petabytes of environmental data. The rapidly increasing volume of data stored on JASMIN, and the steadily increasing number of users, is making it necessary to investigate and implement new methods of providing computing resources to the users, storing the data that they produce from their analyses and storing and maintaining a very large archive of environmental data. To achieve this, two main areas of research are described. Firstly, providing users with virtualised services to best utilise the computing resources available. Secondly, using object storage to provide a large, yet affordable, data store and providing the users with tools and interfaces to common environmental data formats, so as to not unduly affect their current work flows.

1. INTRODUCTION

We describe key avenues of development for the next evolution of JASMIN [1], a hosted processing and data analysis facility for the UK environmental sciences community and its work with international partners. Now in its sixth year of operation, JASMIN provides a large curated archive of earth observation and climate science datasets, group workspaces for users to store their data, together with a batch compute environment (LOTUS) and a community cloud; all hosted on an infrastructure customised for high bandwidth and low latency between compute and storage.

2. DEVELOPMENT OF JASMIN

There are three main development areas: supporting the ongoing migration of compute workloads to containers, developing a cluster-as-a-service provision, and delivering additional tiers of storage. These all arise from the primary purpose of JASMIN: to provide a “data commons” where users can see existing data, add their own, and exploit their own computational environments to manipulate the data. Usage of JASMIN is growing along three independent axes: number of users, volume of data stored, and number of communities supported. Throwing more hardware at the problem alone (whether in-house at JASMIN, or in the public cloud) cannot be afforded, and more sophisticated approaches are necessary, particularly to support less computationally mature user communities.

3. USER ENVIRONMENTS

One key challenge is the ability to make most effective use of parallelism in order to best utilise the computational resources available. This is impacted by usability: for some users, the technical expertise needed to using traditional batch compute is too high or they don’t know how to refactor their code to make it work in parallel. Additionally, there is a need for interactive and graphical application environments in order to analyse data. Technologies such as Jupyter [2] and Zeppelin [3] Notebooks and Dask [4] and Apache Spark [5], when used in combination, provide a new opportunity to provide interactive data analysis and a means to exploit parallelism which to some degree abstracts the complexities from the user.

In order to deliver these environments for our user communities, we need a means to rapidly deploy the underlying virtualised infrastructure a so-called *cluster-as-a-service*. Our experiences with earlier projects on JASMIN such as the ESA-funded OPTIRAD [6] project have shown the potential for container technologies used in conjunction with such services. Recent work with Kubernetes [7] has demonstrated the ability to rapidly deploy environments, make services elastic, more effectively manage the allocation of resources between application and traditional cloud virtualisation tiers and also more easily port between different platforms. Nevertheless, this and other application scenarios must address challenges around storage: interfaces to it, performance and scaling.

doi: 10.2760/383579

4. DATA STORAGE

Most of the current JASMIN storage is delivered using a parallel file system. This gives both users and administrators an easily accessible, highly performant storage environment. However, data volumes are increasing rapidly, arising from both new satellite missions, and more environmental simulation, as well as an even more rapid increase in user-created data. With growing data volume comes the increasing need to make better use of tiered (and cheaper) storage for less frequently accessed data and/or data for which high-performance access is not required. At the same time, the new modes of interaction (from containers, and both private and public cloud) require new methods of interacting with the data. Recent work exploring the use of object storage, alongside new tape caching systems, has shown the potential to address many of these issues. In particular, object stores with simple HTTP REST interfaces such as Amazon S3 [8], provide an approach which avoids the limitations of mounting file systems and managing root privileges, yet can support higher level semantics and rich metadata with reasonable performance.

Even so, S3 presents a fundamental change in how users access data and presents challenges in how best to provide efficient access to data stored using binary formats such as the HDF5/NetCDF4 [9] data model. Work underway with the EU ESiWACE [10] project and collaboration with the HDFGroup [11] in the US has shown how it is possible to make an efficient system to store and access such data by distributing the storage of files across multiple objects with each object representing a fragment of data from the file. In such a way, it is possible to create a subsetting interface to extract individual content without downloading the whole of an object to a client. Using the HDF REST API [12] or the OPeNDAP [13] specification it is possible to integrate this interface with the standard NetCDF client libraries so that the interface presented to the user is largely unchanged.

In addition to HDF5/NetCDF4 files, JASMIN holds a large quantity of both structured and unstructured data, with a large volume of zip files in particular. Additional work will concentrate on how to exploit the internal structure of the structured files to enable them to be split across multiple objects, and finally to provide a solution for the storage of unstructured data. In all cases, deriving and holding rich metadata for each object is a primary goal, to enable searching both the data archive and user data without fetching whole objects.

5. ROADMAP TO IMPLEMENTATION

In order to implement an object storage based solution, a hierarchy of interfaces will be offered to users, with the most simple being implemented first:

1. Object storage used in conjunction with POSIX file system cache i.e. whole files stored in object store and retrieved and operated on by applications from the cache - analogous to tape retrieval. A client utility will be provided to the users
2. Implement a user focused variant of the netCDF4 Python library enabling the splitting a netCDF file into smaller netCDF files using the Climate Format Aggregator (CFA) [14] conventions.
3. Implement a server focused solution to serving netCDF files from the archive, using the HDFGroup Highly Scalable Data Service (HSDS) [15].
4. Provide object storage support for other file formats, determined on which is most prevalent in the archive and user data.

6. REFERENCES

[1] http://www.jasmin.ac.uk

[2] http://jupyter.org

[3] https://zeppelin.apache.org

[4] https://dask.pydata.org

[5] https://spark.apache.org

[6] http://www.esa.int/Our_Activities/Space_Engineering_Technology/Shaping_the_Future/Optical_Multisensor_Radiance_Data_Fusion_Techniques_OPTIRAD

[7] https://kubernetes.io

[8] http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html

[9] http://www.unidata.ucar.edu/software/netcdf/docs/index.html

[10] https://www.esiwace.eu/

[11] https://www.hdfgroup.org/

[12] https://support.hdfgroup.org/projects/hdfserver/ - rest

[13] https://www.opendap.org/

[14] http://www.met.reading.ac.uk/~david/cfa/0.4/index.html

[15] https://support.hdfgroup.org/projects/hdfserver/

doi: 10.2760/383579