Adaptive Tier Selection for NetCDF and HDF5

Extended Abstract

Jakob Luettgau
Deutsches Klimarechenzentrum
Hamburg, Germany
luettgau@dkrz.de

Eugen Betke
Deutsches Klimarechenzentrum
Hamburg, Germany

Olga Perevalova
Universität Hamburg
Hamburg, Germany

Julian Kunkel
Deutsches Klimarechenzentrum
Hamburg, Germany

Michael Kuhn
Universität Hamburg
Hamburg, Germany

ABSTRACT

Scientific applications on supercomputers tend to be I/O intensive. To achieve portability and performance data description libraries such as HDF5 and NetCDF are commonly used. Unfortunately, the libraries often default to sub optimal access pattern for reading and writing data to multi-tier distributed storage systems. This work explores the feasibility of adaptive selection of storage tiers depending on an application's I/O behavior.

KEYWORDS

Storage, System Software, Performance, HDF5, NetCDF

ACM Reference format:
Jakob Luettgau, Eugen Betke, Olga Perevalova, Julian Kunkel, and Michael Kuhn. 2017. Adaptive Tier Selection for NetCDF and HDF5. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Scientific discovery to a large extent relies on high performance computing (HPC). Generation, post processing and preservation of scientific data routinely requires reading and writing large amounts of data. Technological and budgetary constraints have lead to complex storage hierarchies. Rapid advances in compute capability, achieved by exploiting distributed parallelism within and across nodes of a cluster computer[9], have also influenced storage systems to adopt distributed approaches to provide matching I/O performance.

As application developers are adapting their codes to take advantage of the next-generation exascale systems, the I/O bottleneck becomes a major challenge[1, 4, 6] because storage systems struggle to absorb data at the same pace as it is generated. Especially, simulation codes such as climate and weather forecasts periodically experience bursty I/O, as they are writing so called checkpoints to achieve fault tolerance and for data analysis.

The increasing complexity of storage systems adds extra burden to the already complicated task of developing scientific applications. As a result many applications use data description libraries such as NetCDF[3] and HDF5[2] for performance and portability.

Unfortunately, high-level information about the structure of the data and also the programmers intent are lost as data traverses the storage stack.

As part of the Excellence in Simulation of Weather and Climate in Europe (ESiWACE)[1] a middleware for earth system data is developed to improve efficiency and productivity of numerical weather and climate simulation on HPC platforms by supporting the end-to-end workflow of global Earth system modeling. The middleware aims to provide access to the same shared data but using different APIs, e.g., NetCDF, HDF5 or GRIB. In addition, transparently to the user the middleware should apply data-center specific optimizations such as cost/performance optimized data placement by using different storage backends adaptively.

It follows a short introduction of HDF5 and it's virtual object layer in Section 2. The approach of addaptive tier selection and resulst are presented in Section 3 and Section 4. Section 5 introduces related work and Section 6 concludes with a short summary.

2 HDF5 VIRTUAL OBJECT LAYER

The proof of concept prototype makes uses of development branch of HDF5. HDF5 is an open source, hierarchical, and self-describing format that combines data and metadata. The format is widely used by many scientific applications due to its versatile data model, the portable file format and build in support for a wide range of computational platforms up to massively parallel systems.

Virtual Object Layer (VOL):. An enabling technology that allows for optimizations to benefit a wide range scientific applications is provided by the HDF5 Virtual Object Layer (VOL). The VOL is an experimental fork of HDF5 that is being developed by the HDF Group to allow for more flexibility to exploit arbitrary storage backends. The changes introduced with VOL are described in detail in a request for comments[5]. VOL intercepts all API calls and forwards those calls to plugin object drivers. The VOL exports an interface that allows third party plugin development.

Plugin for Seperate Metadata Handling: A VOL plugin was developed that separates and handles data and metadata differently. To this end, metadata is stored in an SQLite3 database and data in a shared file but a full fledge RDBMS or a NoSQL based backend is also thinkable. Generated simulation data is routinely published, but at the moment not automatically cataloged With the VOL plugin it is possible to store summaries extracted from the metadata

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

for available datasets to make them available for search in a catalog as a dataset is written.

3 ADAPTIVE TIER SELECTION

One opportunity to optimize performance and to lower total operation cost is provided by making better use of the available tiers. The underlying assumption is to avoid unnecessary data movements if possible by exploiting higher level knowledge about the data that is written by an application. The general approach for tier selection boils down to the following three steps:

(1) Establish ground truth about expected performance of available tiers either via automatic benchmark (e.g. hourly or before an I/O phase) or via a configuration file.
(2) Make use of runtime information and of the intent of an application, e.g., as exposed by the description of a HDF5 Dataset/Datapace and the MPI domain decomposition.
(3) Consult a decision component that incorporates knowledge from 1) and 2) before I/O is redirected to a subsystem.

Adaptive tier selection is realized as depicted in the architecture illustration below. The prove of concept decision component accounts for two runtime information made available by MPI and HDF5: 1) domain decomposition and 2) the domain description of a dataset. The tier selection policies used by the performance model are based on benchmark measurements obtained at an earlier time.

4 EVALUATION

For evaluation the NetCDF Performance Benchmark Tool[7] is used to mimic a parallel application that is using NetCDF for checkpoint/restart. Each configuration consists of a checkpoint size `{xdim,ydim,zdim}` which is read and written `t` times, a domain decomposition `{nodes+ppn}` and the I/O path (SHM, SSD, Lustre, adaptive). Measurements were performed on the Mistral supercomputer of the German Climate Compute Center (DKRZ), #38 (June'17). The following tiers types have been used and evaluated for the prototype:

- *Shared Memory/RAM*: Tier for small random I/O local to the node. Even though RAM today is not persistent, it is considered in expectence of burst buffers that should offer comparable performance in many situations.
- *Local SSDs*: For medium size random access I/O. This tier is for data not shared with other nodes. In most cases node local NAND and disk storage will perform worse than a parallel file system in terms of latency and throughput. Positive is that node local storage allows to take load of the network.
- *Parallel File System (Lustre)*: For large sequential file I/O as in large high resolution checkpoint/restart data. Also when a single large file holding the whole dataset and multiple variables is required e.g. to exchange with other researchers.

5 RELATED WORK

Multiple efforts are working towards performance portability and transparent performance optimization in HPC environments. The Adaptive Input/Output System (ADIOS) [8] uses what is called componentization to separate the I/O transfer methods from application codes. Using XML configuration files it is then easy to exchange the I/O methods when compiling for another system. ADIOS also uses a binary data format for efficient data storage. While impressive I/O performance improvements can be achieved using ADIOS, applications have to be developed with the ADIOS APIs in mind.

The Distributed Asynchronous Object Storage (DAOS)[6] is a software stack aiming to provide a new paradigm for exascale systems and for big data analytics. The architecture aims to exploit NVRAM and NVMe enabled storage. The project provides many attractive semantics such as transactions, global address spaces and built-in resilience, but assumes technologies that are not yet available in many data centers.

The NEXTGenIO[4] focuses on the exploitation of non-volatile memory technologies to overcome the I/O bottleneck. In an co-design effort a hardware prototype is developed along with software to drive the system and tools for performance analysis. The project also works on topology awareness and the integration with workload scheduling and modeling of I/O workloads using simulation for faster more educated prototyping.

6 SUMMARY

Adaptive tier-selection is a viable approach for performance optimization of I/O performance. As storage systems become more heterogeneous in the wake of burst buffers and non-volatile memory I/O middleware can help to avoid exposing unnecessary complexity to users. In future work a formalism for decision components for tier-selection based on access patterns is developed. The integration of various storage tiers is also continued as part of the ESiWACE project. In particular the following backends deserve further exploration:

- Object storage mappigns for short term storage of working sets as an alternative to parallel file systems.
- Tape and other nearline storage for affordable long-term archival and as cold storage for infrequently accessed data.

Besides storage backends, the integration of scientific workflows with workload manager requires investigation as it offers new opportunities to automatically reduce data movements by exploiting spacial and temporal data locality.

ACKNOWLEDGMENT

This work is part of the ESiWACE project which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 675191.

REFERENCES

[1] [n. d.]. ESiWACE | Centre of Excellence in Simulation of Weather and Climate in Europe. ([n. d.]). https://www.esiwace.eu/
[2] [n. d.]. HDF5 | Hierarchical Data Model. ([n. d.]). https://www.hdfgroup.org/hdf5/
[3] [n. d.]. NetCDF | Network Common Data Format. ([n. d.]). https://www.unidata.ucar.edu/software/netcdf/
[4] [n. d.]. NEXTGenIO | Next Generation I/O for the Exascale. ([n. d.]). http://www.nextgenio.eu/
[5] Chaarawi Mohamad and Quincey Koziol. 2014. RFC: Virtual Object Layer. (Sept. 2014).
[6] Intel, The HDF Group, EMC, and Cray. 2014. Fast Forward Storage and I/O - Final Report. (June 2014).
[7] joobog. 2017. Netcdf-Bench: NetCDF Performance Benchmark Tool (NetCDF-Bench). (March 2017). https://github.com/joobog/netcdf-bench
[8] ORNL. 2017. ADIOS. (2017). https://www.olcf.ornl.gov/center-projects/adios/
[9] Top500. 2017. Top500 Supercomputer Sites. (2017). http://www.top500.org/