SimFS: A Simulation Data Virtualizing File System Interface

Salvatore Di Girolamo
Dept. of Computer Science
ETH Zurich
digirols@inf.ethz.ch

Torsten Hoefler
Dept. of Computer Science
ETH Zurich
htor@inf.ethz.ch

ACM Reference Format:
Salvatore Di Girolamo and Torsten Hoefler. 2018. SimFS: A Simulation Data Virtualizing File System Interface.

1 POSTER DESCRIPTION

Reliable long-term data archiving is very costly. For example, storing 10 TiB for 10 years costs between $2,400 and $6,000 on Microsoft's Azure. The only practical scheme to mitigate these costs, besides deletion, is (lossy or lossless) compression of the data and it is fundamentally constrained by the tradeoff between data size and quality. When taking a closer look at *how data is generated*, we discover two fundamentally different modes: (1) data collected by sensors or terminals that observe non-deterministic environments or (2) data generated by deterministic simulations that model complex and potentially chaotic systems. *We observe, that the latter could be recomputed on demand instead of stored, given the right data retrieval system.*

Many simulation applications produce vast amounts of data that is today stored in large filesystems or databases. For example, the European Centre for Medium-Range Weather Forecasts (ECMWF) alone had an archive of 100 PiB in 2015, experiencing an annual growth rate of 45% [5]; by 2020, their archive will reach a Zettabyte. Climate model data is used by countries and insurances to make critical decisions thus repeatability of analyses is mandated by international regulatory bodies. Astrophysics simulations are another example where data volumes grow with the compute capabilities, creating more than 20 PiB of data each [8]. Thousands of such simulations are collected in virtual observatories, mainly limited by the storage costs [3, 9]. Those two examples outline a clear trend: As we proceed into the age of simulation [11], *big (simulation) data* will soon be required for many real-world decisions.

The data produced by large simulations is commonly used by thousands of analysts and scientists over the course of decades. They are used in analysis workflows where the data is stored in files or databases. Specifically, these workflows address two requirements: (1) data can conveniently be analyzed with any access pattern (e.g., time-reverse or random access) and (2) the exact same data can (often years) later be re-analyzed to reproduce the results. This makes the data-backed analysis a de-facto standard for today's simulation data analytics.

SC'18, Nov. 2018, Dallas, TX, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

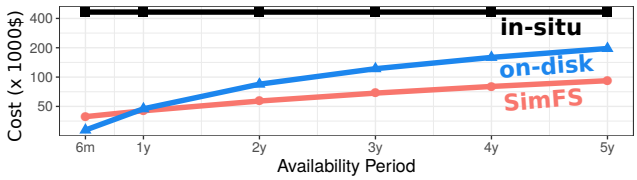

Figure 1: Cost of multiple analyses of the same simulation output. We express the cost as function of the amount of time over which the analyses are executed and for different storage solutions (on-disk, in-situ, SimFS).

We propose *SimFS*, a file system that virtualizes simulation output data for analysis tools. SimFS avoids storing the whole simulation output data but stores checkpoints to re-start parts of the simulation to produce missing files *on demand*. A virtualized view, similar to virtual memory, provided to the analysis tools enables them to work *as if* all output data exists as files. This way, SimFS can precisely optimize the tradeoff between inflexible *in-situ* simulations where all analyses are running together with the simulation and no data is stored, and the full output of all data with later analysis. Figure 1 teases the expected costs for performing 100 analyses equally spaced over varying data availability periods for a real-world climate simulation scenario. It shows that SimFS can reduce the costs for a five-year period from more than $200,000 for an on-disk solution to less than $100,000. We also show "in-situ", which re-runs the whole simulation for each analysis as comparison.

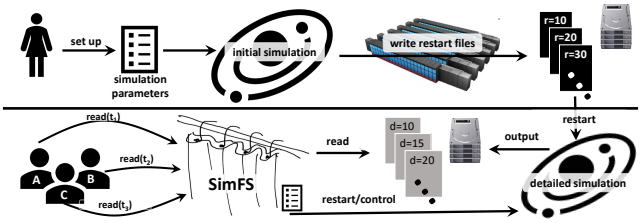

Figure 2: Overview of SimFS

Implementing SimFS poses interesting challenges that we describe in the following. Figure 2 shows the abstract workflow of SimFS: The simulation is set up by a scientist initially (top left of Figure 2) and then run to completion while producing restart files (black files, top right). First in-situ analyses may be performed during the initial simulation but we focus on later analyses in this paper. Later, analysis tools from different clients (e.g., researcher A, B, and C in the lower left) access the virtualization layer through standard data-access interfaces such as HDF5 [4], netCDF [10], or ADIOS [6]. Alternatively, analysis tools can be made aware of the virtualized envirorment by using the SimFS APIs, gaining more control and information about the virtualization. SimFS manages the

simulations to re-create output data (gray files, bottom) on demand and delivers it to the analysis tools. Simulation contexts define the specific simulation configuration: e.g., which files to produce and with which frequency; where the output is stored; how to run the simulation. We remark that simulations can be restarted on different devices than the original simulation, e.g., smaller GPU systems, because the simulated time intervals are less demanding.

SimFS requires that the simulation can be re-started from checkpoints and delivers a bitwise-identical output to the original run. While checkpoint/restart facilities are already needed to deal with limited compute time and failures, bitwise reproducibility may not generally be available. However, it should generally be used for good scientific practice (repeatability) and can be achieved with a set of standard techniques without significant performance penalty [2, 7].

We argue that SimFS solves a significant part of the big data storage challenge in simulation sciences. We will show how it even improves analysis performance and automatically utilizes available storage resources efficiently. SimFS is currently used in the crCLIM[1] project on the Piz Daint supercomputer, one of largest machines existing today.

REFERENCES

[1] 2017. crCLIM: Convection-resolving climate modeling. http://www.crclim.ch. (2017). Accessed: 2018/07.

[2] Andrea Arteaga, Oliver Fuhrer, and Torsten Hoefler. 2014. Designing Bit-Reproducible Portable High-Performance Applications. In *Proceedings of the 28th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE Computer Society.

[3] Maksym Bernyk, Darren J Croton, Chiara Tonini, Luke Hodkinson, Amr H Hassan, Thibault Garel, Alan R Duffy, Simon J Mutch, Gregory B Poole, and Sarah Hegarty. 2016. THE THEORETICAL ASTROPHYSICAL OBSERVATORY: CLOUD-BASED MOCK GALAXY CATALOGS. *The Astrophysical Journal Supplement Series* 223, 1 (2016), 9.

[4] Mike Folk, Albert Cheng, and Kim Yates. 1999. HDF5: A file format and I/O library for high performance computing applications. In *Proceedings of Supercomputing*, Vol. 99. 5–33.

[5] Matthias Grawinkel, Lars Nagel, Markus Mäsker, Federico Padua, André Brinkmann, and Lennart Sorth. 2015. Analysis of the ECMWF Storage Landscape. In *Proceedings of the 13th USENIX Conference on File and Storage Technologies (FAST'15)*. USENIX Association, Berkeley, CA, USA, 15–27. http://dl.acm.org/citation.cfm?id=2750482.2750484

[6] Jay F Lofstead, Scott Klasky, Karsten Schwan, Norbert Podhorszki, and Chen Jin. 2008. Flexible IO and integration for scientific codes through the adaptable IO system (ADIOS). In *Proceedings of the 6th international workshop on Challenges of large applications in distributed environments*. ACM, 15–24.

[7] Ingo Müller, Andrea Arteaga, Torsten Hoefler, and Gustavo Alonso. 2018. Reproducible Floating-Point Aggregation in RDBMSs. *arXiv preprint arXiv:1802.09883* (2018).

[8] Douglas Potter, Joachim Stadel, and Romain Teyssier. 2016. PKDGRAV3: Beyond Trillion Particle Cosmological Simulations for the Next Era of Galaxy Surveys. *arXiv preprint arXiv:1609.08621* (2016), 1.

[9] Antonio Ragagnin, Klaus Dolag, Veronica Biffi, Marionne Cadolle Bel, Nicolay J. Hammer, Aliaksei Krukau, and Daniel Steinborn Margarita Petkova. 2016. An online theoretical virtual observatory for hydrodynamical, cosmological simulations. *arXiv preprint arXiv:1612.06380* (2016).

[10] Russell K Rew and Glenn P Davis. 1990. The unidata netCDF: Software for scientific data access. In *Sixth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*. 33–40.

[11] Eric Winsberg. 2010. *Science in the age of computer simulation*. University of Chicago Press.