Modeling Allocation of Heterogeneous Storage Resources on HPC Systems

Julien Monniot, François Tessier, Gabriel Antoniu
Inria
Rennes, France
{first name}.{last name}@inria.fr

***Abstract*—In recent years, and despite remarkable progress in computing and network performance, HPC platforms have struggled to maintain satisfactory I/O throughput. Various solutions have been proposed to mitigate the contention and variability experienced by more and more concurrent applications, particularly on heavily shared parallel file systems. As a result, many large scale platforms now offer complex hierarchies of storage resources using various architectures (node-local, burst buffers, network-attached storage and so on) based on diversified hardware technologies such as persistent memories or flash for instance. In that context, we propose to study how to efficiently allocate these heterogeneous storage resources for scientific applications and workflows. For that purpose, we developed StorAlloc, a modular and extensible simulator of a storage-aware job-scheduler. We ran a large set of experiments with StorAlloc to investigate storage system designs and resource scheduling algorithms, through the analysis of multiple storage-related metrics. For example, we have been able to determine an estimate of the size of a burst buffers partition sufficient to capture the intensive I/O of a top-tier supercomputer.**

***Index Terms*—storage, heterogeneity, hpc, simulation**

I. Context and motivations

Scientific applications and workflows in domains such as cosmology, climate modeling, genetics or high-energy physics, among others, require the computing performance of increasingly powerful HPC platforms. In recent years, we have witnessed the spread of more and more data-driven applications in these areas, leading to a compute to storage drift sometimes referred to as *data deluge*. With emerging projects like the SKA radio telescope[1] or climate models using high resolution digital twins of earth[2], we expect the scale of processed data to keep increasing and eminently soon reach many exabytes per year, with correlated demands regarding storage and I/O performances. However, looking at platforms in the Top500, we observe a growing gap between computing power and IO performance: on the top 3 supercomputers, the ratio of IO bandwidth to computing power has been reduced by a factor of $\sim$ 10 over the last ten years. Meanwhile, many works have highlighted the proliferation of contention issues and performance variability on traditional large-scale shared storage systems [1], [2]. One solution to this bottleneck is to deepen the storage hierarchy with intermediate storage tiers using existing and emerging technologies. Thus, many HPC platforms now make use of a wide variety of storage solutions between the compute node and the shared parallel file-system, ranging from node-local storage to burst buffers for instance. They often leverage diverse technologies such as persistent memory, NVMe(oF) or regular SSDs, as well as arrays of HDDs and tapes for capacity tiers. These new layers have different characteristics in terms of data lifetime (cache, job-term, medium-term, and so on), scope (node-local to off-site) or I/O throughput that need to be considered. In particular, complex workflows mixing simulation, data analysis and AI require flexible solutions for storage and dissemination of scientific data that can be fulfilled by these tiers.

This rapidly growing heterogeneity in storage solutions brings new challenges for an efficient use of all resources, for instance when allocating dedicated storage space to a job. Some works have already studied ways to leverage this complexity, but most focus on scheduling I/O or optimising a single storage level [3], [4]. Our approach focuses on heterogeneous storage allocations at job level.

[1]https://www.skatelescope.org/the-ska-project/

[2]https://digital-strategy.ec.europa.eu/en/policies/destination-earth

II. Our approach

Among the aforementioned storage-related challenges, we have focused our efforts on optimising the use of heterogeneous resources, through the study of allocation algorithms. In that regard, we have adopted a simulation-based approach.

The motivation is two-fold. First, it is hard to experiment on actual production platforms as it requires to repurpose production resources. Second, simulation offers flexibility to abstract emerging architectures or technologies and assess various storage-aware job scheduling algorithms.

Simulators of distributed platforms already exist but to the best of our knowledge, they mainly aim attention at compute resource allocations and only offer a bird's eye view on storage [5], [6], focusing on modeling I/Os with limited to no support for scheduling storage resources. Our goal is thus to develop a tool that could help design new solutions for allocating storage, while accounting for the impact of the heterogeneity of storage systems being deployed. Our proposal should allow to describe several heterogeneous storage architectures and to evaluate storage-aware scheduling policies using relevant metrics.

III. Solution design: StorAlloc

In this context, we introduce StorAlloc, a DES-based simulator aimed at modeling allocation of heterogeneous storage

resources for HPC jobs. Our tool consists of three main components: the clients that submit requests, the orchestrator that computes the scheduling, and the storage infrastructure that describes the resources.

On the client side, StorAlloc takes as input storage allocation requests (capacity, duration) and submit them to the orchestrator which attempts to allocate them on available intermediate storage resources. Using Darshan[3] I/O traces of jobs executed on an actual HPC platform, a large number of requests can be generated to feed the simulator. It is important to notice that StorAlloc does not replay and schedule individual I/O requests, but rather concentrate on optimizing the placement of storage allocations on resources.

Within the orchestrator, four scheduling algorithms have been implemented. The component-based design of StorAlloc allows to easily extend it with more strategies. Data structures allow to manipulate requests and resources in a fine way.

On the storage infrastructure side, resources are abstracted in a hierarchy composed of one or many logical storage server(s), which manage a pool of resources represented by *storage nodes* and *disks*. Each node and disk can be characterized by multiple fields among which read or write bandwidth, latency, capacity or network bandwidth. At any time, the orchestrator may get the current state of all available resources on the simulated platform, along with the list of running and pending allocations.

Finally, communication between StorAlloc components uses a socket-based messaging protocol. Through this protocol we bring up simulation data to additional logging and visualization components. These components provide the user with simulation-time results of major metrics and detailed post-simulation data for further analysis.

IV. Results

Our first round of experiments led us to compare 192 configurations of the simulator, using different values for the total storage capacity of the infrastructure, the layout of storage nodes and disks and the choice of scheduling algorithm. Depending on the complexity of the running algorithm, we were able to complete each simulation in a range of 5m40s to 1h29m57s, with an average duration of 25m48s. In every run, we used the same dataset as input: 24'000 jobs out of 624'000, selected on I/O criteria from one year of processed Darshan traces from the Theta supercomputer at Argonne National Laboratory[4].We used the simulation data generated by our experiments to analyse storage resources usage and compare algorithms and layouts. In order to conduct our study, we defined metrics such as remaining disk bandwidth given a permanent I/O regime or concurrent allocations at node and disk level over the simulation duration, for example. More can be seamlessly added into StorAlloc. Through this analysis, we demonstrated that StorAlloc can answer questions about the impact of splitting requests into distributed blocks across multiple nodes and disks, the benefits (or not) of storage disaggregation, the effectiveness of a particular scheduling algorithm to minimize contention on intermediate storage resources or the proper sizing of a burst buffer partition. This latter case is set forth on our poster. We show that on Theta, under the right conditions for storage layout and scheduling strategy, a 16TB burst buffer is sufficient to cover 90% of the needs of all selected I/O-intensive jobs.

[3]https://www.mcs.anl.gov/research/projects/darshan/

[4]This data was generated from resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

V. Conclusions and future work

We are developing StorAlloc, an extensible DES-based simulator for allocating heterogeneous storage resources required by I/O intensive jobs on HPC systems. Leveraging the flexibility of simulations, it can be used as a test-bed to validate the usefulness of storage related metrics, explore the representation of storage infrastructures and develop new scheduling algorithms for storage resources. We validated StorAlloc with I/O traces from one year of jobs execution on a HPC system and used experimental data to analyze multiple storage infrastructures. We showed that StorAlloc can be used to answer worthwhile storage-related issues. Our future works shall focus on simulation accuracy and may lead us to integrate StorAlloc concepts with a state of the art framework such as WRENCH [7]. A longer-term objective will be to extend the scheduling capabilities of StorAlloc in order to make it directly collaborate with a batch scheduler, usually devoted to the allocation of compute resources only.

References

[1] E. Costa, T. Patel, B. Schwaller, J. M. Brandt, and D. Tiwari. 'Systematically Inferring I/O Performance Variability by Examining Repetitive Job Behavior'. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 1–15. St. Louis Missouri: ACM, 2021. https://doi.org/10.1145/3458817.3476186.

[2] A.K. Paul, O. Faaland, A. Moody, E. Gonsiorowski, K. Mohror, and A.R. Butt. 'Understanding HPC Application I/O Behavior Using System Level Statistics'. In 2020 IEEE 27th International Conference on High Performance Computing, Data, and Analytics (HiPC), 202–11, 2020. https://doi.org/10.1109/HiPC50609.2020.00034.

[3] H. Khetawat, C. Zimmer, F. Mueller, S. Atchley, S.S. Vazhkudai, and M. Mubarak. 'Evaluating Burst Buffer Placement in HPC Systems'. In 2019 IEEE International Conference on Cluster Computing (CLUSTER), 1–11, 2019. https://doi.org/10.1109/CLUSTER.2019.8891051.

[4] G. Aupy, O. Beaumont, and L. Eyraud-Dubois. 'Sizing and Partitioning Strategies for Burst-Buffers to Reduce IO Contention'. In 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 631–40. Rio de Janeiro, Brazil: IEEE, 2019. https://doi.org/10.1109/IPDPS.2019.00072.

[5] H. Casanova, A. Giersch, A. Legrand, M. Quinson, F. Suter. Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms. Journal of Parallel and Distributed Computing, Elsevier, 2014, 74 (10), pp.2899-2917. http://hal.inria.fr/hal-01017319/

[6] P.F. Dutot, M. Mercier, M. Poquet, O. Richard: Batsim: A realistic language-independent resources and jobs management systems simulator. In: Desai, N., Cirne, W. (eds.) Job Scheduling Strategies for Parallel Processing. pp. 178–197. Springer International Publishing, Cham (2017)

[7] H. Casanova, R. Ferreira da Silva, R. Tanaka, S. Pandey, G. Jethwani, W. Koch, S. Albrecht, J. Oeth, and F. Suter, "Developing Accurate and Scalable Simulators of Production Workflow Management Systems with WRENCH", Future Generation Computer Systems, vol. 112, p. 162-175, 2020.