ON MAKING HPC STORAGE SYSTEMS MALLEABLE

Designing dynamic and reactive distributed storage systems

ADMIRE

Adaptive Multi-tier Intelligent Data Manager for Exascale is a EuroHPC project aiming at addressing upcoming I/O challenges in Exascale systems. It gathers thirteen European partners to pioneer new ways of doing HPC I/Os.

Objective

The main objective of the ADMIRE project is the creation of an active I/O stack that dynamically adjusts computation and storage requirements through intelligent global coordination, elasticity of computation and I/O, and the scheduling of storage resources along all levels of the storage hierarchy, while offering quality-of-service (QoS), energy efficiency, and resilience for accessing extremely large data sets in very heterogeneous computing and storage environments.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 under Grant Agreement number: 956748-ADMIRE-H2020-JTI-EuroHPC-2019-1.

ADMIRE
malleable data solutions for HPC

I/O Challenges in HPC

In High-Performance Computing (HPC) data movements are one of the biggest challenges. Indeed, large computation is necessarily leading to large datasets. Current HPC workflows favor a feed-forward way of launching programs, loading their dataset, and then storing the result in persistent storage for later post-processing. This is done by separate jobs without any form of collaboration. Moreover, the I/O backend is so critical that it generally runs separated from the machine in a service island, being dimensioned for the whole system. **What if the I/O subsystem and the application started collaborating to perform better?** This is the question ADMIRE tries to respond to. This EUROHPC project has taken the ambitious goal of experimenting with a holistic I/O management approach. In the project's framework, it translates into a feedback loop and careful job and service reconfiguration to handle I/O resources globally in the computing center. This should translate to a lower dependency on the I/O backend reconfiguring nodes to act as an ad-hoc file-system — reducing the need for large and expensive I/O backplanes. In this context, being able to precisely describe what is taking place on the system is crucial, and to this matter, a new real-time monitoring system was developed by project partners.

EuroHPC Joint Undertaking

Malleable ad-hoc storage systems

ADMIRE intends to significantly extend existing ad-hoc storage systems, making them malleable. Storage systems, e.g., file systems, are generally available for users to store their data. In the context of HPC, large parallel file systems, such as Lustre or GPFS, are used for long-term data storage. To reduce application interference, so-called *burst buffers* are used, offering a temporary storage environment to accelerate the I/O performance of applications.

The advantages

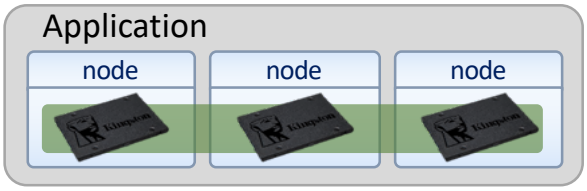

Fig I. Illustration of combining the I/O capabilities of three nodes by using a burst buffer file system

Nowadays, compute nodes offer node-local storage, e.g., SSDs, that can be used as burst buffers. Unfortunately, they often remain unused because they only provide local storage which is not useful for distributed applications that rely on a distributed namespace. Burst buffer file systems, such as GekkoFS, BurstFS, dataClay, or Hercules, fill that niche and can combine node-local burst buffers into a single global namespace. They further accumulate the I/O performances and capacities of the SSDs.

Because these kinds of burst buffer file systems are often collocated with compute nodes, they are usually deployed *ad-hoc*. Such file systems are also called ad hoc file systems, and their ability to be quickly deployed and destroyed in a job context is particularly important not to waste precious computational resources. Because these ad hoc burst buffer file systems are often accessed by a single application, they can heavily optimize for an application's requirements. Further, this allows applications to run in isolation with regards to the I/O, which is only accessed to import input datasets (*stage-in*) for the application's used ad-hoc file system or to export its results for long-term storage to the parallel file system (*stage-out*).

If possible, such a file system could even relax its consistency guarantees to offer higher performance and scalability than the parallel file system can provide.

The challenges and solutions for usability and malleability

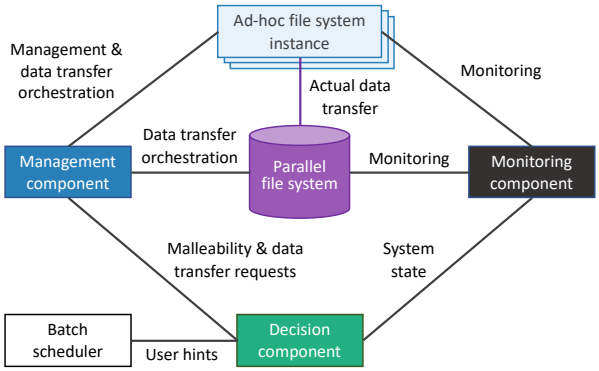

Fig II. ADMIRE's proposal for a malleable HPC stack

Typically, ad-hoc file systems are user-space file systems allowing them to be used by any non-privileged user. What is more, many support the standard POSIX file system interface so that an application does not need to be modified. However, before an application can use the ad-hoc file system, users need to launch the system as well as stage-in input data before the application starts, and stage-out any non-intermediate output data once the application finishes.

The ADMIRE framework makes this process entirely transparent to the user, who only needs to hint stage-in and stage-out paths when

submitting a batch job, while the ADMIRE framework manages the ad-hoc file systems and the corresponding data transfer. This topic will be briefly revisited in a later blog post.

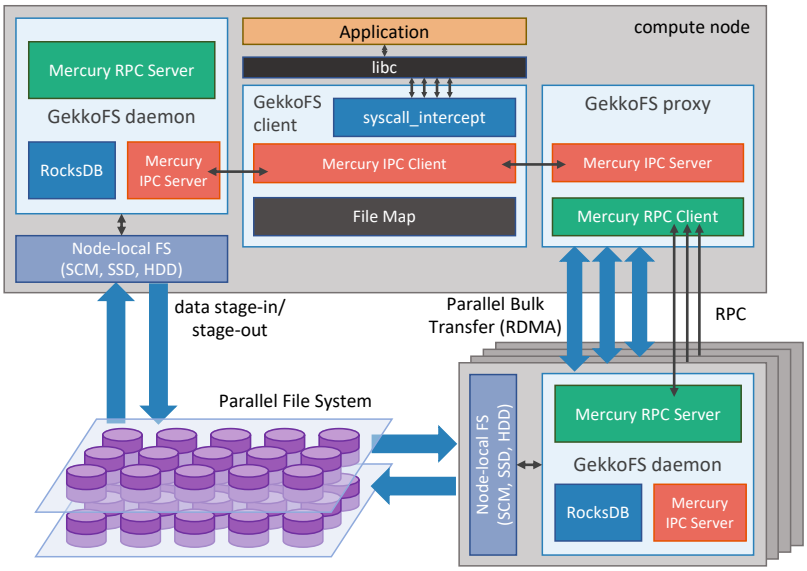

Fig III. GekkoFS's extended architecture

The challenges are considerably larger for enabling file system malleability, that is, molding it to the application's and HPC system's requirements. Although users can offer hints to the storage system w.r.t. the application's I/O requirements, the state of the system is out of their reach. As a result, the ad-hoc storage systems are integrated into the ADMIRE framework such that it can transparently trigger the ad-hoc file system's malleable options when required.

What's next...

To support any actions decided by the ADMIRE framework, the file systems are currently significantly extended, as outlined on the right. Further, on-going tasks include trace analysis of the I/O behavior of HPC applications to optimize them as best as possible.

admire_eurohpc@uc3m.es

Admire EURO-HPC Project

@admire_eurohpc

Possible malleable options

Several malleable options are considered for the ad-hoc storage systems. The following presents a small outline of them.

Dynamic sizing of storage resources

Depending on the state of the HPC system, usage of the parallel file system, and an application's I/O requirements in different phases, an ad-hoc file system's size, i.e., its number of used storage nodes, can be dynamically increased and decreased. This allows general malleability and bandwidth control.

Data distribution

Generally, ad-hoc file systems try to use all I/O nodes as efficiently as possible by including all of them in an I/O operation. However, this is not necessary in some cases, e.g., when processes only read the data they have written. Therefore, our ad-hoc file systems offer several data distribution policies.

Quality-of-Service and consistency protocols

The ad-hoc file systems support QoS mechanisms for more fine-grained control to manage the used computational and storage resources. Further, they can be configured to relax consistency guarantees, decreasing file system communication and, as a result, increasing I/O performance.