Towards a Task-Based I/O System

Anthony Kougkas
Department of Computer Science
Illinois Institute of Technology
Chicago. USA
akougkas@hawk.iit.edu

Hariharan Devarajan
Department of Computer Science
Illinois Institute of Technology
Chicago. USA
hdevarajan@hawk.iit.edu

Xian-He Sun
Department of Computer Science
Illinois Institute of Technology
Chicago. USA
sun@iit.edu

I. Abstract

In the era of data-intensive computing, with ever growing dataset sizes and an exploding core count, storage systems are a critical component in achieving computational efficiency. Large scale applications, in both scientific and the BigData communities, demonstrate unique I/O requirements leading to a variety of storage solutions which are often incompatible to one another. Each new architecture has been accompanied by new software for extracting performance on the target hardware. Further, each new storage device added to the system architecture of large-scale computing machines has led to new paradigms. Parallel file systems (PFS) are the de-facto storage solution in most HPC machines. As the name implies, a PFS deals with data in the form of files. PFSs obey certain standards, such as POSIX, to offer portable guarantees and strong data consistency. PFS manipulate data in a certain sequence of operations known as streamlined I/O. In cloud environments the storage scene is different. Innovation is driven by the wide popularity of computing frameworks. As a result, the cloud community has developed a wide variety of storage solutions tailored to serve specific purposes. To navigate this vast and diverse set of contradictory I/O requirements, the software landscape is filled with custom, highly specialized storage solutions varying from high-level I/O libraries to custom data formats, interfaces, and ultimately storage systems. Modern storage systems must efficiently support a diverse and conflicting set of features.

In this study, we aim to explore a novel way to view the I/O needs of modern applications. We propose TABIOS, a new, distributed, scalable, and adaptive Task-based I/O System. In TABIOS, all I/O requests are transformed to a configurable unit of I/O, called DataTask, which is a tuple of operation and a pointer to user data. Datatasks are pushed from the application to a distributed queue which is served by a scheduler. TABIOS workers (i.e., storage servers) execute datatasks independently. The entire TABIOS architecture is decoupled. TABIOS' datatasks offer great flexibility to both the applications, which simply express their I/O workload in the form of an I/O task, and to the storage infrastructure that can now be malleable (i.e., elastic storage resources). Using datatasks, Most importantly, TABIOS can support a wide variety of diverse conflicting I/O workloads, from scientific computing to BigData analytics, on a single platform. We envision TABIOS to be agile, capable of power-capped computing, and reactive to the environment offering storage QoS guarantees and tunable concurrency control.

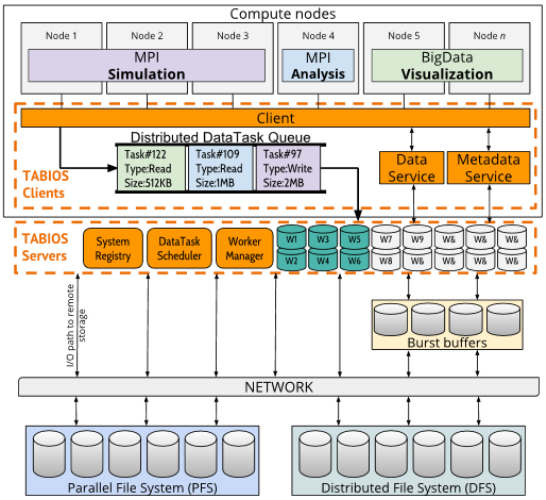

Figure 1: TABIOS High-level Architecture

TABIOS demonstrates:

1. the effectiveness of **storage malleability**, where resources can grow/shrink based on the workload.
2. how to effectively use **asynchronous I/O** with the mixed media and various configurable storage options.
3. how to support **resource heterogeneity** based on the targeted hardware configuration supporting a variety of storage resources under the same platform.
4. the effectiveness of **data provisioning**, enabling in-situ data analytics and process-to-process data migration.
5. how to support a diverse set of conflicting I/O workloads, from HPC to BigData analytics, on a single platform, through effective **storage consolidation**.

The following figure shows the read and write datatask operation decomposition expressed as time percentage and divided by each TABIOS component. All results are the average time of 10K datatasks. As expected the most time is spent on the workers performing the disk operations.

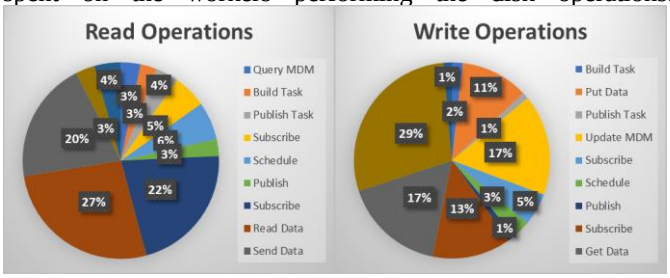

Figure 2: Read/Write Operations Decomposition