Analyzing Parallel I/O BoF Summary

Philip Carns
Argonne National Laboratory
Julian Kunkel
German Climate Computing Center

December 12, 2016

Abstract

Parallel application I/O performance often does not meet user expectations. Additionally, slight access pattern modifications may lead to significant changes in performance due to complex interactions between hardware and software. These challenges call for sophisticated tools to capture, analyze, understand, and tune application I/O. In this BoF, we will highlight recent advances in monitoring tools to help address this problem. We will also encourage community discussion to compare best practices, identify gaps in measurement and analysis, and find ways to translate parallel I/O analysis into actionable outcomes for users, facility operators, and researchers.

1 Organization

The workshop was organized as a series of short talks followed by a discussion about the future of I/O analysis tools. A summary of the talks and discussion is provided in this document.

2 Attendees

During the BoF, we counted roughly 100 attendees.

Informal polling suggests that the majority of the attendees were computer science researchers, though there were also a significant number of facilities staff. There were only a few application developers and end users in attendance.

3 Talks

In the first talk, Phil Carns gave an overview of the BoF scope and motivated the needs for analysis tools.

Then Shane Snyder introduced the new features of Darshan. Darshan now provides a modular instrumentation to support additional I/O layers, allows retrieval of characterization data when applications terminate ungracefully, and enables detailed tracing of I/O. The benefit of detailed instrumentation of Lustre is demonstrated on the HACC-IO code, where misbehaving Lustre servers could be identified.

In the talk Characterizing burst buffers at extreme scale using the TOKIO framework, Glenn Lockwood described the data collection and analysis implemented as part of the TOKIO project. Using RabbitMQ for scalable collection and Spark for analysis of various data sources, performance behavior of NERSCs DataWarp burst buffer can be understood better, for example by revealing write-back behavior. The project now investigates the use of standardized file formats to keep trace data and identify relevant views.

In the talk Characterizing Parallel I/O Behaviour Based on Server-Side I/O Counters, Salem El. Sayed described research to identify performance issues using file system counters, in particular observed read/write throughput on storage servers. He also proposed several metrics to characterize I/O traffic. This tool has been used to investigate observed burstiness, confirming that most I/O occurs in short bursts and available throughput is not well utilized.

In the talk SIOX In-Situ Optimization and Virtual Laboratory, Jakob Lttgau described the SIOX framework. As a prototype, it offers recording and replay of I/O calls and manipulation of I/O operations at runtime to evaluate alternative I/O strategies without changing applications. A performance analysis of these light-weight changes in application scope compared to changes on VFS layer in FUSE show that this can be more than two orders of magnitude faster.

In the short talk Status Update Monitoring at DKRZ, Julian Kunkel first described the monitoring infrastructure at DKRZ based on OpenTSDB and Grafana. It uses the Lustre feature to attribute SLURM JOBIDs to RPCs allowing detailed analysis of performance counters on the server side. Then in the talk Statistical File Characterization, he described a method and tool to determine file characteristics such as compression ratio on large data volumes without scanning the complete data set.

In the talk, Mining Supercomputer Jobs' I/O Behavior from System Logs, Xiaosong Ma described I/O analysis that has been applied at the OLCF. The system records performance counters on Lustre OSTs with RAID granularity in a MySQL database, but the shared storage access patterns must be attributed to individual applications in order to gain insight into application behavior. This is performed by applying wavelet transformation on multiple runs of the application. This strategy was used to identify I/O intensive applications and propose scheduling enhancements to improve efficiency. She also observed that only a relatively small number of applications are able to make optimal use of the storage system.

4 Discussion

The workshop organizers prepared several questions to stimulate the discussion. With respect to overhead, attendees suggested that even a recognizable overhead might be tolerable, if the users and researchers gain something (knowledge) in return.

The most common concern expressed by the attendees was how to turn tools and data collection into user outreach, i.e., how do you identify problems automatically and get feedback to users of the system to improve it. There was no clear consensus on how to solve this problem; it is an open issue. Potentially, commercial support for open source tools could be offered to relieve some of this burden from data centers.

Even the most widely used tools have a big learning curve and aren't very polished from a presentation/support/documentation standpoint. The path forward on this issue is to work with vendors to incorporate support for more I/O analysis tools. Staff should support this development by explicitly asking for tools during the procurement of a new system.

The audience expressed considerable interest in methods to combine parallel I/O analysis into a "big picture" of I/O behavior. Many people would like to have this capability at their facility.