U.S. Department of Energy Best Practices Workshop on
File Systems & Archives
San Francisco, CA
September 26-27, 2011
Position Paper

Richard Hedges
Lawrence Livermore National Laboratory
hedges1@llnl.gov

ABSTRACT / SUMMARY

This position paper discusses issues of usability of the large parallel file systems in the Livermore Computing Center. The primary uses of these file systems are for storage and access of data that is created during the course of a simulation running on an LC system.

INTRODUCTION

The Livermore Computing Center has multiple, globally mounted parallel file systems in each of its computing environments. The single biggest issue of file system usability that we have encountered through the years is to maintain continuous file system responsiveness. Given the back end storage hardware that our file systems are provisioned with, it is easily possible for a particularly I/O intensive application or one with particularly inefficiently coded I/O operations to bring the file system to an apparent halt.

The practice that we will be addressing is one of having an ability to indentify, diagnose, analyze and optimize the I/O quickly and effectively.

Tools applied

LMT: LMT[1] (Lustre monitoring tool) is run by the Livermore Computing system administration staff to monitor operation of the Lustre file systems. It is generally used to probe and isolate reported problems rather than to identify the problem before or as it develops.

Having an earlier version of LMT accessible to users proved problematic. The particular issue was that some users would "cry wolf" when they saw periods of heavy usage of a file system. These notifications were generally self serving and counter productive, so presently LMT is available for system administrators only.

In daily operations, Lustre system administrators may have running instances of LMT, but would not necessarily be tracking the output, unless a problem (such as a file system being sluggish or unresponsive) had been reported. Due to the architecture of Lustre, LMT leads one down an indirect path in identifying the source of a file system load. Load is observed on storage or metadata servers, next correlated with client activity, and finally (hopefully) identified with a single users job. This detective work can take some time, so it can be a challenge to do all of the tracing while the offending code instance is still active.

Darshan, strace: Since a single application program with inefficiently coded I/O operations can have center wide negative impact on parallel file system function and usability, It is critical to be able understand the sequence of I/O operations that a code is generating and to understand the effects of those on file system behavior.

For some time we have been in the business of profiling file system I/O for selected applications to diagnose and resolve performance problems causing center wide impact. Initially profile data was extracted exclusively from strace [2], and

application runs and analyzed essentially by manually reviewing the data.

We generally trace with the options "strace -tt -etrace=file,read,write,close,lseek,ioctl" which provides time stamped system call traces to standard error for the I/O related system calls identified. We can collect the system call traces on a per process basis. It is possible to trace a running process, or to incorporate the tracing in a job run script.

More recently we also use Darshan[3] from Argonne National Laboratory. Darshan is a petascale I/O characterization tool. Darshan is designed to capture an accurate picture of application I/O behavior, including properties such as patterns of access within files, with minimum overhead. Darshan includes scripting to analyze and aggregate the data.

A code can be instrumented with Darshan by utilizing wrapper scripts, or by interposing the libraries using LD_PRELOAD. Being as lightweight as it is makes it suitable for full time deployment, although we at LC do not apply it in that manner.

These methods are available to users, but have primarily been applied by an LC staff member on behalf of a user or application team. Note that these methods are also applied in the case where the performance issues impact the user's productivity, even if the center-wide impact is minimal.

User training and documentation

Training specific to application I/O performance issues is summarized in two documents maintained on the clusters in /usr/local/docs: (1) Lustre.basics and (2) Lustre.striping. We have also included I/O specific discussions in user oriented system status meetings on a regular basis. Consulting is available, and is offered on a general and on an intervention basis.

Let me interject a personal comment here related to user training, because I am eager to see if others at the workshop have observed similar. Relative to other parts of parts of a complex HPC system (e.g. processor architecture, code parallelization) I find that our user community seems generally more resistant to learning about the I/O architecture and how to use and code to it effectively. I suspect that this is a historically bias that the CPU processing is the valuable resource and the I/O bandwidth and storage capacity are free. We at the center may have reinforced this, if subtly, by our accounting and allocation policies.

CONCLUSIONS

For the key initial step of identifying a code or user who, by their I/O actions are impacting the user, we have a workable tool. LMT provides an path, albeit indirect to associate system load with a particular root cause.

The strace and Darshan offer approaches (some overlapping and some complementary) to analyze the I/O execution of an HPC application. They allow one to identify and localize a problem in a code.

We have an issue on the user training side. Some code teams have taken on the challenge of understanding good application I/O practices. Others have been motivated only when I/O performance was an insurmountable hurdle.

REFERENCES

1. LMT github site https://github.com/chaos/lmt/wiki
2. strace: stardard Linux command to "trace system calls and signals" see "man strace"
3. Darshan: Petascale I/O Characterization Tool *http://www.mcs.anl.gov/research/projects/darshan/*