

LDMS Darshan Connector: Integrating LDMS Streams for Execution-Time Diagnosis of HPC Application I/O Performance

Abstract—**Might need to redo this later** Improving I/O performance usually depends on a large number of components such as the applications access pattern, the computing system architecture, I/O libraries, file system, mode of access, data size, and the storage configuration and layout. Changes in these components can create high variations in I/O performance which could lead to many negative effects such as network congestion, poor scheduling and wasted resources on the system. Having this many variables makes it difficult to detect the root cause of variations in I/O performance and most of the time strong correlations of I/O analyses across different applications are required to identify a possible solution. This paper introduces a unique methodology that provides the opportunity for low latency monitoring of I/O performance variations during execution-time. This is done through the implementation of a system-level infrastructure that continuously collects I/O application data from an existing I/O characterization tool. This allows insights into the I/O application performance and the components affecting it through the analyses and visualizations. This methodology addresses the challenge of poor understanding of throughput for system-specific behaviors and variations of I/O performance of similar applications across a system. This paper demonstrates the implementation and assessments of this method and how it can be used on various HPC applications.

I. INTRODUCTION

As more scientific applications are developed and used, the need for improved fidelity and throughput is more pressing than ever and much design effort and investments are being put into improving not only the application but also the system components. Being able to identify, predict and analyze I/O behaviors are critical to ensuring parallel storage systems are utilized efficiently. However, I/O performance continues to show high variations on large-scale production conditions in many cases **Cite NE**. This variation makes it difficult to determine the root cause of I/O related problems and have a thorough understanding of throughput for system-specific behaviors and I/O performance in similar applications across a system. Further, not knowing what the cause is will directly affect the user and developer as unwanted time, effort and investment will need to be put into solving the issue.

Variations in I/O can be caused by the file system (e.g. buffering, file transfer, interrupt handling), network congestion, system resource contentions, or the access patterns of the application itself.

Generally, the I/O performance is analyzed post-run by application developers, researchers and users in the form of regression testing or other I/O characterization tools that capture the applications I/O behavior. An example of one of

these tools is *Darshan* which captures I/O information on access patterns from HPC applications. Detailed information along with the LDMS integration will be covered in the *Approach* section. In the case where an I/O performance problems are observed, efforts to identify this usually come from any identified correlation between similar analyses of various applications. However, this analysis approach does not take into account the file system, network congestion and system resource contentions affect on the I/O performance. In order to make these associations, the *absolute timestamps* is required the post-run approach and I/O characterization tools usually lack.

Explain the structure of the paper. Can do this when it's mostly complete

II. RELATED WORK

Look for papers that used a similar approach

III. PROBLEM AND APPROACH

- **Address current I/O performance analysis and implications (i.e. refer to Emily's SC21 presentation)**

Lesson Learned: Different applications experience high and low performance variations at separate and disjoint time periods. These time periods are shared across different applications and clusters.

How LDMS will help: It is suggested that a simple I/O monitoring data collection from Darshan will be useful in identifying these time periods. LDMS will be able to provide the run time I/O monitoring data collection which will help them in identifying these time periods.

Lesson Learned: Clusters running on weekends observe some of the highest performance variations. It is assumed that this could be from I/O intensive application runs during the weekends.

How LDMS will help: The I/O timeseries data generated by LDMS will provide NERSC researchers with new data they can utilize to further their research.

- **Benefits of having time series data:**

Better understand how application I/O variability correlates with overall system behavior

Understand how time of day affects the I/O performance

Understand how the I/O pattern within a run affects the I/O performance of read and write. E.g. perhaps an application that has nodes writing data at disjoint times have less I/O performance variation than one that has all nodes writing data at the same time.

Understand why read and write I/O performance patterns are different and independent of each other. E.g. read and write may exhibit different behavior during the run, which affects the performance of each in unique ways

Discuss the issues with not knowing the IO application data. How are the users and developers affected?

A. Darshan and LDMS Overview

- Explain how Darshan works. What is it, who created it, what does it do exactly? What areas does it lack? (i.e. no timestamps)
- Explain how LDMS works (high level overview – same as Darshan). What capabilities does it have that we will be applying to Darshan to transport the data.
- Explain how SOS works and why we are using it.
- Explain how Grafana works with SOS and why we are using it.

B. Integration

- Explain how LDMS is integrated into Darshan. Give an overview on how we implement LDMS Streams into the Darshan DXT section to collect execution-time I/O data and push to a JSON file that then gets aggregated by LDMS and stored to SOS. Then explain how this stored data is then queried and displayed in a Grafana dashboard.
- Add a pic of current Darshan LDMS integration setup (.png)

IV. USE CASES

In this section, we discuss the correlation of I/O performance variation with network congestion and file system contention and describe our design and implementation of *Darshan LDMS Integration* at a high level.

Explain the different scenarios we will be testing:

- Applications: SWFFT, sw4, sw4lite. Standard base-line: mpi-test.
- Explain what each application does, why it's being tested and how the test was performed (i.e. number of nodes, etc.)
- Explain the analysis used to analyze the I/O data and how they address the correlation between I/O performance variation and network congestion/file system contention.

Figure out what type of analyses address this correlation. Reach out to Emily and Ana (possibly) about this.

- show Grafana graphs, Darshan output (maybe) JSON and any tables (if applicable).

A. Results

- Analyse the results of the use cases
- What is the significance of the results?
- Throw in a picture of the Grafana Dashboard for each use case
- How do these results satisfy and solve the problems described in the the "Problems and Approach" section.

V. FUTURE WORK

- Explain the purpose of the future work.
- What are our plans for this connector? Will we be implementing *LDMS Streams* across other I/O characterization tools, expanding the streams capabilities on Darshan, testing across other applications, etc.?

A. Figures and Tables

a) *Positioning Figures and Tables*: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text.

TABLE I: Table Type Styles

Table Head	Table Column Head		
	<i>Table column subhead</i>	<i>Subhead</i>	<i>Subhead</i>
copy	More table copy ^a		

^aSample of a Table footnote.



Fig. 1: Example of a figure caption.

REFERENCES

FIX REFERENCES