Storage Performance Analysis Based on Kernel and Userspace Traces

Houssem Daoud

Progress Report meeting
May 2019

École Polytechnique de Montréal
Agenda

- **Introduction**
- **Mass storage**
  - Performance analysis of local storage devices
  - Performance analysis of distributed storage systems
- **Main memory storage**
  - Monitoring kernel memory usage
  - Performance analysis of automatic memory management mechanisms
- **Conclusion and future work**
Introduction

- Using faster processors doesn't always improve the performance of the system.

A 2× faster processor gives an acceleration rate of 1.14x

- Storage operations are a major bottleneck in high-performance computing systems.

- Many mechanisms have been developed to improve the performance of storage operations (disk schedulers, memory allocators).
The complexity of those mechanisms makes them difficult to debug using traditional tools.

**Benchmarking**
- Synthetic workloads
- Doesn't help in finding the origin of the problem

**Tracing**
- Analyzes the behavior of real workloads
- Offers a more accurate insight into the internals of the storage subsystem

LTTng: a low overhead tracing framework
Introduction

- Tracing overhead can affect the normal behavior of the system (High frequency events).

- The amount of data generated by tracing is huge and needs to be post-processed

Objectives

- Tracing the storage subsystem with a minimal overhead
- Analyzing the performance of the different storage systems
- Providing a comprehensive visualization system
Agenda

- Introduction
- **Mass storage**
  - Performance analysis of local storage devices
  - Performance analysis of distributed storage systems
- **Main memory storage**
  - Monitoring kernel memory usage
  - Performance analysis of automatic memory management mechanisms
- Conclusion and future work
Performance analysis of local storage devices

Proposed architecture
Performance analysis of local storage devices

Data Analysis

Stateful Analysis

The state of the system is kept in a historical database built incrementally in a single pass over the trace.

I/O request life cycle
Performance analysis of local storage devices

Metrics computation

Latency

Latency = Preparation Time + Waiting Time + Service Time
Performance analysis of local storage devices

Visualization

Disk waiting queues

Queue length

Debit

Latency distribution
Performance analysis of local storage devices

Use case: Investigating a high latency

Configuration problem: the backup process has a higher priority than the web server

Fixing the priority problem
Agenda

- Introduction
- **Mass storage**
  - Performance analysis of local storage devices
  - **Performance analysis of distributed storage systems**
- Main memory storage
  - Monitoring kernel memory usage
  - Performance analysis of automatic memory management mechanisms
- Conclusion and future work
Performance analysis of distributed storage systems

Proposed architecture

- Node 1
  - Ceph OSD process
  - Block Layer
  - Disk Driver
  - Lttng Tracer
  - Trace

- Node 2
  - Ceph OSD process
  - Block Layer
  - Disk Driver
  - Lttng Tracer
  - Trace

- Client
  - Application
  - Filesystem
  - Librados
  - Ceph Block device
  - Lttng Tracer
  - Trace

Synchronization + Analysis

Graphical views
Data collection

- The lightweight tracing session traces a small number of events and analyzes them on the fly in order to detect unusual behaviors.
- The exhaustive tracing session writes the complete trace temporarily in a circular buffer.
- The trace is only written if a problem is detected.
Performance analysis of distributed storage systems

Visualization

```

Name
write-zipkin
  write-zipkin/compute1.ce
    17231
    17607
msg
  write-zipkin/compute2.ce
    436
    516
  msg
write-zipkin/compute3.ce
write-zipkin/compute4.ce

sdd
  Driver_queue
    0
    1
    2
    3
    4
    5
    6

Disk queue
  write_request
  write_request
  write_request
  write_request
  write_request
  write_request

Waiting for OSD

Ceph client request

OSD thread

The OSD inserts I/O requests into the disk queue

Wr...

The OSD sends an ACK

The I/O requests are successfully processed
```
Performance analysis of distributed storage systems

Use case: Impact of a slow disk

The client has to wait until the replication is successfully completed in all the secondary OSDs.
Agenda

- Introduction
- Mass storage
  - Performance analysis of local storage devices
  - Performance analysis of distributed storage systems
- Main memory storage
  - Monitoring kernel memory usage
  - Performance analysis of automatic memory management mechanisms
- Conclusion and future work
Monitoring kernel memory usage

**Dynamic Trace-based Sampling Algorithm**

- An event is triggered if memory variability exceeds a certain threshold.

- Implemented as a Kernel module.

- Lock-free data structures are used to provide a good scalability.
Monitoring kernel memory usage

**Use cases**

Firefox memory usage at startup using Massif / Dynamic Sampling Algorithm

Totem video player memory usage using Massif / Dynamic Sampling Algorithm
Agenda

- Introduction
- Mass storage
  - Performance analysis of local storage devices
  - Performance analysis of distributed storage systems
- Main memory storage
  - Monitoring kernel memory usage
  - Performance analysis of automatic memory management mechanisms
- Conclusion and future work
Performance analysis of automatic memory management mechanisms

Proposed architecture

Diagram showing the proposed architecture with components such as Profiling Agent, Userspace Tracer, Kernel Tracer, and Analysis Module connected to visual components like Java Application, Java Virtual Machine, Operating System, and Visualization System.
Performance analysis of automatic memory management mechanisms
Performance analysis of automatic memory management mechanisms

This view shows on which CPU each Java thread is running

The view shows that a GC thread is being preempted by other processes
Agenda

- Introduction
- Mass storage
  - Performance analysis of local storage devices
  - Performance analysis of distributed storage systems
- Main memory storage
  - Monitoring kernel memory usage
  - Performance analysis of automatic memory management mechanisms
- Conclusion and future work
Conclusion

- Recovering Disk Storage Metrics from Low-level Trace Events (journal - published)
- Performance Analysis of Distributed Storage Clusters Based on Kernel and Userspace Traces (journal - submitted)
- Dynamic Trace-based Sampling Algorithm for Memory Usage Tracking of Enterprise Applications (conference - published)
- Multilevel Analysis of The Java Virtual Machine Based on Kernel and Userspace Traces (journal - submitted)

Future work

- Live tracing support
- Using machine learning algorithms to detect and classify performance problems