LOGPOINT

Technical White Paper:

Adaptative Memory Management with Shenandoah GC (Garbage Collector)

www.logpoint.com

ABSTRACT

This White Paper discusses the integration and utilization of the Shenandoah garbage collector for Java services within the Logpoint framework. It outlines the advantages of enabling the Shenandoah Garbage collection in Logpoint, while also providing insights on its proper usage. Shenandoah is an advanced garbage collector specifically designed for dynamic workloads. The introduction of Shenandoah aims to simplify the tuning of Logpoint services and ensure optimal system performance and stability. This White paper addresses a common challenge faced by System Administrators when manually adjusting heap settings, during periods of high demand or sudden spikes in activity.

TABLE OF CONTENTS

INTRODUCTION

The Logpoint platform relies on Java services to carry out its essential functions, such as storing logs and generating analytics that include alerts, dashboards, and search features. As Java is a garbage collection language, it requires careful tuning of Logpoint's Java services to adjust the heap sizes dynamically based on the current workload. Managing heap size efficiently is critical since overallocation can lead to unnecessary memory reservations by some services, which can prevent the proper functioning of others that need resources.

On the other hand, under-allocation can lead to inadequate heap sizes, causing Java services to experience extended pause times during full garbage collection. This can introduce latency in search and log ingestion and in extreme cases, it can create severely impact on log collection and analytics. The significant challenge System Administrators face is addressing these issues to ensure optimal system resource utilization and stability.

SHENANDOAH GARBAGE COLLECTOR OVERVIEW

Shenandoah is a garbage collector designed to work seamlessly in systems with dynamic workloads. It distinguishes itself by reducing pause times, making it particularly useful in applications where responsiveness is critical. Shenandoah's integration with Logpoint's Java services reduces the need for manual tuning and re-tuning heap size.

The Shenandoah garbage collector will automatically use the necessary heap during spikes and return memory to the operating system if there is no load in services, allowing other services to make the best use of the available memory resources. The behavior of the Shenandoah garbage collector alleviates the need for manual tuning and the challenge of determining the optimal heap size.

CHALLENGES WITH PREVIOUS GARBAGE COLLECTOR

- Regular, manual adjustment to Java services heap size.
- Reduced performance of Logpoint due to long pause time.
- Significant decrease in throughput of specific services due to garbage collection loops, affecting log collection and analytics pipeline.
- Overall system performance impact when more fine-tuning is during periods of high- demand or when there are sudden spikes.
- Unnecessary memory reservation with previous garbage collection method as it infrequently releases memory back to the operating system, like during full GC.

SHENANDOAH IN ACTION

There are multiple advantages introduced through the integration of Shenandoah into Logpoint services:

- Ability to concurrently perform garbage collection while the application is running ensures system responsiveness even during peak or burst loads.
- Shenandoah's reduced pause time ensures optimal application performance.
- Returning memory to the operating system ensures optimal resource allocation for other services.
- After Shenandoah GC integration, Java services no longer require manual heap size adjustment.

BENEFITS FOR SYSTEM ADMINISTRATORS

Shenandoah significantly impacts System Administrators who manage Logpoint services. The automated and concurrent garbage collection reduces the need for frequent manual adjustments of heap size, thus simplifying the tuning process. This leads to improved Logpoint stability and reliability, especially during resource-intensive and dynamic workloads.

SEAMLESS USAGE OF SHENANDOAH

The Shenandoah garbage collector can be easily enabled for all core Logpoint services with a single command accessible to the li-admin user. To do so, follow these steps:

- Go to the Logpoint console or create a ssh-session using the li-admin user.
- Run the command shenandoah_manager enable.

RECOMMENDATIONS

We recommend using the Shenandoah garbage collector only in systems with low CPU utilization or a low load average. Before enabling Shenandoah, check Logpoint's CPU utilization under System -> System Monitor. If CPU utilization exceeds 85%, it's not advisable to enable Shenandoah. After enabling Shenandoah, monitor your system's memory utilization, CPU utilization, and load average using the system monitor dashboard for a few days. If you observe any unusual behaviour, please revert to the previous garbage collector. Switching is this simple:

- Go to the Logpoint console or create a ssh-session using the li-admin user.
- Run command shenandoah_manager disable.

OBSERVED IMPROVEMENTS

After enabling Shenandoah GC in our Logpoint environments, what we found is shown in the following image:

Memory Usage (Previous Garbage Collection Method)

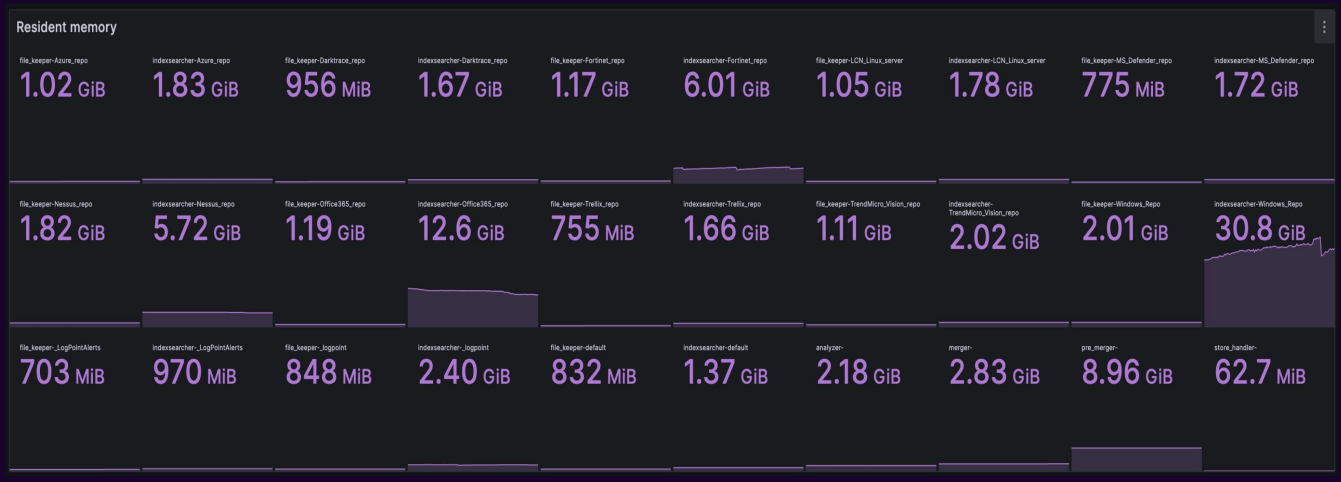

Fig. 1: Memory usage of core Java services inside Logpoint for one of our Logpoint environments with previous garbage collection method.

Memory Usage (Previous Garbage Collection Method)

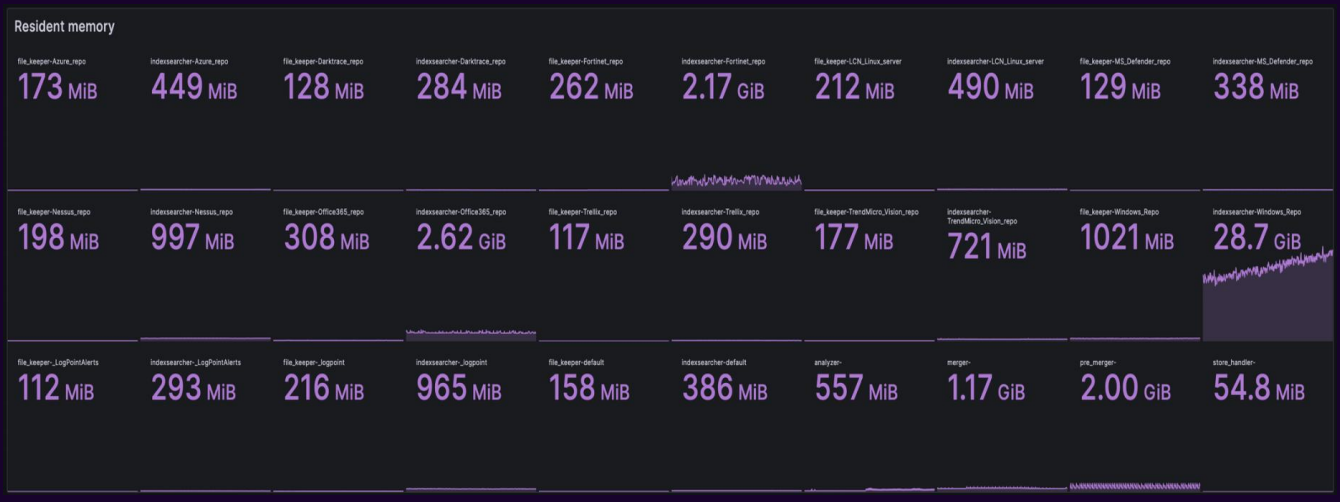

Fig. 2: Memory usage of core Java services inside Logpoint for one of our Logpoint environment after enabling Shenandoah.

Zooming into the memory usage pattern of one Service before and after enabling Shenandoah:

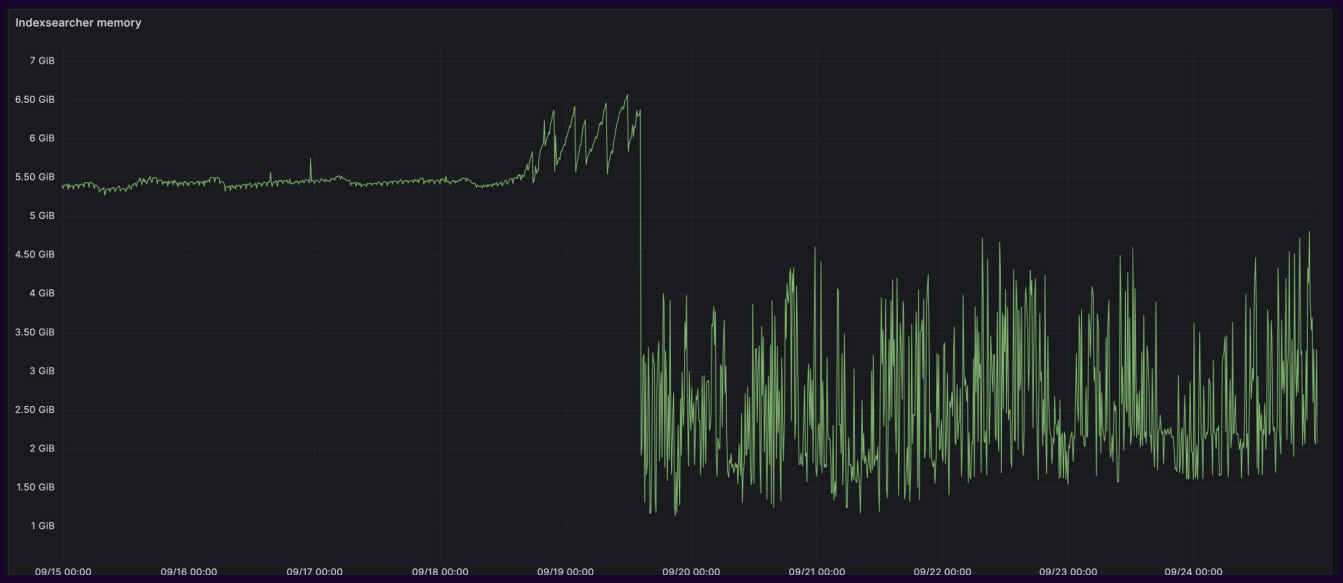

Fig. 3: This is the memory usage pattern of a service before and after enabling the Shenandoah (Shenandoah was enabled on 9/29).

We can see that prior to 9/29 the memory was reserved by the service, and it wasn't released back to OS. After enabling Shenandoah, we can see memory is being used where it is needed and is being released back to the OS accordingly.

CONCLUSION

Adopting the Shenandoah garbage collector is a significant step forward in optimizing Logpoint's Java services. It addresses the challenges associated with manual heap tuning in previous version of Logpoint and empowers System Administrators to ensure peak system performance and stability. By embracing solutions like Shenandoah, Logpoint underscores its commitment to innovation and delivering enhanced user value as it continues to evolve.

Note: The implementation behaviour of previous garbage collection (G1GC) mentioned in this white paper is of Logpoint version 7.4.0 and before.

LOGPOINT.COM