UNDERSTANDING OBSERVABILITY PIPELINES 01

Cribl

WHITE PAPER

Six Techniques to Control Log Volume

BY **CLINT SHARP, CO-FOUNDER & CEO, CRIBL**

WHITE PAPER

Six Techniques to Control Log Volume

by Clint Sharp, Co-Founder & CEO, Cribl

Introduction

Log Analytics tools are one of the most expensive categories of observability and security tooling, and costs are continuing to increase dramatically. According to an IDC-Seagate Report in 2018, global data stores are increasing at 61% per year. This is accompanied by an explosion of log data being collected and stored. With costs at large customers already in the 10s of millions each year, increasing at double-digits year over year means material costs in the millions.

Log Analytics system costs are a function of the volume of data they are ingesting and storing. Historically, the amount of infrastructure required for a log analysis system scaled linearly with the amount of data stored. Newer innovations separate storage and compute in these systems which allows for longer retention with less cost, but in every architecture, more data requires more compute to process the data at query time. Quite simply, more data, more cost.

At the same time, administrators of these systems have long known that much of the data they are asked to store is junk. Error storms or just bad logging often result in repeated messages with duplicative content. Legacy systems often encode high volume metric type data as logs. Compliance often drives treating log analysis tools also as a system of record, where data which is never queried is stored in the log analysis tool in case of a breach or lawsuit. Individual events often contain superfluous information, usually in the form of key value pairs in structured logs which are set to null or are not valuable to the consumer. All of these add up to a ton of wasted ingest, driving up the cost of log tooling dramatically while providing little value.

WE'VE OBSERVED ORGANIZATIONS ACHIEVING 25% REDUCTION IN DAILY INGESTION VOLUME WITH RELATIVELY LOW EFFORT, AND MORE AGGRESSIVE PROGRAMS CAN HIT 50% OR MORE.

With a little bit of work, administrators can often dramatically trim ingestion rates. We've observed organizations achieving 25% reduction in daily ingestion volume with relatively low effort, and more aggressive programs can hit 50% or more depending on the environment and types of data. Let's review a few techniques for how these administrators are achieving savings.

1. FILTERING

The oldest and simplest technique for controlling log volume is simply to not store it at all. Most log analysis systems have some kind of mechanism for dropping data administrators find not to be valuable, although the techniques for selecting which data to be dropped often leave a lot to be desired.

2. ROUTING TO LOWER COST STORAGE

Log Analytics systems are very expensive systems of record. Storing data in an inverted index balloons storage requirements, and software license costs can be a major factor. Data collected for compliance or breach investigation use cases without day to day use cases is probably best stored in low cost object storage, like Amazon S3 or MinIO. Rather than sending to a log analysis tool, administrators can divert data to an object store, stored in well organized directories. Data can then be ingested later into their log analytics tool of choice or into query environments like Amazon Athena or Snowflake.

3. DROPPING UNNECESSARY FIELDS

The developer who originally builds a log message has a strong incentive to stuff that message with as much information as they have, to avoid having to go back and add additional information later. However, as a consumer of this data, especially from devices and software outside of the organizations control, much of this information is unnecessary. Cisco eStreamer has dozens of key value pairs in every message which are set to NULL or N/A. Microsoft PowerShell logs contain a full copy of the script in every log message. Preprocessing this data to remove unnecessary fields from these messages can result in dramatic savings.

4. CONVERTING LOGS TO METRICS

AGGREGATE THOSE LOGS INTO SUMMARY METRICS ... OFTEN WITH A DRAMATIC REDUCTION IN EVENT COUNTS AND DATA VOLUME.

Many of the highest volume data sources are really metrics in disguise: web activity logs, network flow logs, or custom application telemetry. If a measurement value in the log is the reason for ingesting it, it makes sense to aggregate those logs into summary metrics. Metrics can still be stored in a logging tool, often with dramatic reduction in event counts and data volume, or metrics can be sent to a dedicated time series database like InfluxDB, Circonus or DataDog for efficient storage and retrieval. Some logging tools such as Splunk also provide efficient metrics storage as well.

5. DEDUPLICATING EVENT STREAMS

Log data contains a lot of duplicative information. In a distributed system, many worker processes will emit errors based on the same underlying problem. Many applications will emit a series events which are useful but often more frequent than is necessary. Deduplicating these streams to emit less frequent messages, along with a count of dropped messages, can still give the investigator the same data at a significantly lower volume of data.

6. SAMPLING

SAMPLING CAN BE USED TO GREAT EFFECTIVENESS ON HIGH VOLUME DATA STREAMS TO GIVE A REPRESENTATIVE VIEW.

Sampling is a technique used under the covers by many analysis systems to speed queries or provide fast summary views. During an election, pollsters don't call every voter, they call a representative sample to get a view on the electorate with an estimated margin of error. Sampling can be used to great effectiveness on high volume data streams to give a representative view, still allowing for read time aggregation, while storing a fraction of the original events.

Implementing These Techniques

Log Analytics tools often provide support for some kind of filtering, but most of the rest of these techniques will require some kind of pre-processing step. Our recommendation is of course our product, Cribl LogStream™, which helps our customers implement all of these techniques and more. Below are some examples.

> *A global financial services firm wanted to use Splunk to analyze DNS logs, but adding a terabyte of data per day to their existing license was deemed too expensive. By using LogStream to enrich data with a top internet domains list, they were able to drop uninteresting logs from trusted domains and reduced one terabyte per day to about 50 gigabytes – well within their budget.*

> *A leading Asia-Pacific energy supplier wanted to collect and send data from 11 offshore oil platforms. Bandwidth at these remote locations made that nearly impossible. They chose LogStream to filter, aggregate, and sample data before sending it compressed over their satellite links back to their centralized log system for analysis.*

In addition to LogStream, we've seen people implement techniques in their Observability Pipelines. There are a number of different tools we've seen implement these techniques; check out our post on **building an Observability Pipeline on top of open source technologies** for more information on build options.