

# TODO:title Improved Parsing of Unstructured Logs using XYZ

Yang Su, Amos Hebb  
*University of Toronto*

## Abstract

TODO:Abstract Your abstract text goes here. Just a few facts. Whet our appetites. Not more than 200 words, if possible, and preferably closer to 150.

## 1 Introduction

CLP [1] is a tool capable of compressing unstructured text logs. The CLP implementation of log ingestion TODO:FluffThisUp. This paper presents an implementatnion of an improved log ingestion process TODO:ProcessName . TODO:ProcessName achieves over TODO:Nx increase in ingestion performance. TODO:ProcessName does so without sacraficing compression ratio, or search performance. TODO:ProcessName's gains come from TODO:PapersGimmick.

## 2 Design Overview

TODO:DesignOverview

## 3 Parsing Messages

TODO:ParsingMessages

### 3.1 Handling Ambiguous Tokens

### 3.2 Handling Special Cases

TODO:keepOrRemoveThese

## 4 Evaluation

TODO:Evaluation We explore: 1) Change in compression speed; 2) Change in compression ratio; and 3) Change in resource efficiency

## 4.1 Experiment Setup

We used the Hadoop-14TB logs [2] were generated by three Hadoop clusters, each containing 48 data nodes, running workloads from the HiBench Benchmark Suite [25] for a month. Note that the dataset generated by a benchmarking tool may be artificially uniform, as benchmarks do not always capture the randomness of real-world deployments. However, this should not affect our claims as we compare CLP with TODO:ProcessName relative to CLP's default log processing on the same dataset.

## 4.2 Compression Speed

To examine the change in single-node ingestion speed, a 30GB subset of the Hadoop corpus was written to a tempfs RAM disk. The RAM Disk minimizes I/O overhead to fully expose algorithmic differences between the two versions of CLP. We use CLP's default compression level for both tests.

## 4.3 Compression Ratio

TODO:CompressionRatio The new parser produces a nearly identical output, except for TODO:ABC. To verify that compression ratio was not impacted, the entire 14TB Hadoop corpus was compressed. With the improved parser, a compression ratio of TODO:CompressionRatioImproved was acheived, within TODO:NPercent of the default parser.

## 4.4 Resource Efficiency

CLP paper asserts "Compression and search are embarrassingly paralelizable". I don't think they actually demonstrate parallel compression, but do have a whole para about whacky IO. We'll need to work out how they measure this, or maybe we just drop the whole section. This feels pretty thin on benchmarking though and I get the feeling DrYuan is big on 'actually running stuff'.

## 5 incoherent ranting

I have downloaded CLP.

There's a few spots where it reaches out looking for x86 specific binaries, seems like turning most of these off works.

I've managed to get it installed and running on a docker image using ubuntu:latest.

It runs, and when I compress about 2GB logs from my macbook's disk

1. it takes about 18 seconds to do heuristic based, roughly 100MB/s, 5x slower than ramdisk in paper.
2. I currently just get flaming segmentation faults whenever I try to use schemas.

## Acknowledgments

TODO:Scrap this section? It's from the USENIX template. If we get access to the cluster or something may we could praise that, feels awkward to ack the existing paper though.

## Availability

HDFS Corpus is available at [2]. The fork of CLP is available at TODO:Repo.

## References

- [1] Kirk Rodrigues, Yu Luo, and Ding Yuan. CLP: Efficient and scalable search on compressed text logs. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*, pages 183–198, 2021.
- [2] Kirk Rodrigues, Yu Luo, and Ding Yuan. hadoop-14tb-part1, September 2022.