LA-UR-20-26122'

# Comparative Analysis of Metric Collecting Software

HPC/USRC Showcase 8/13/2020

**Author**
David Huff

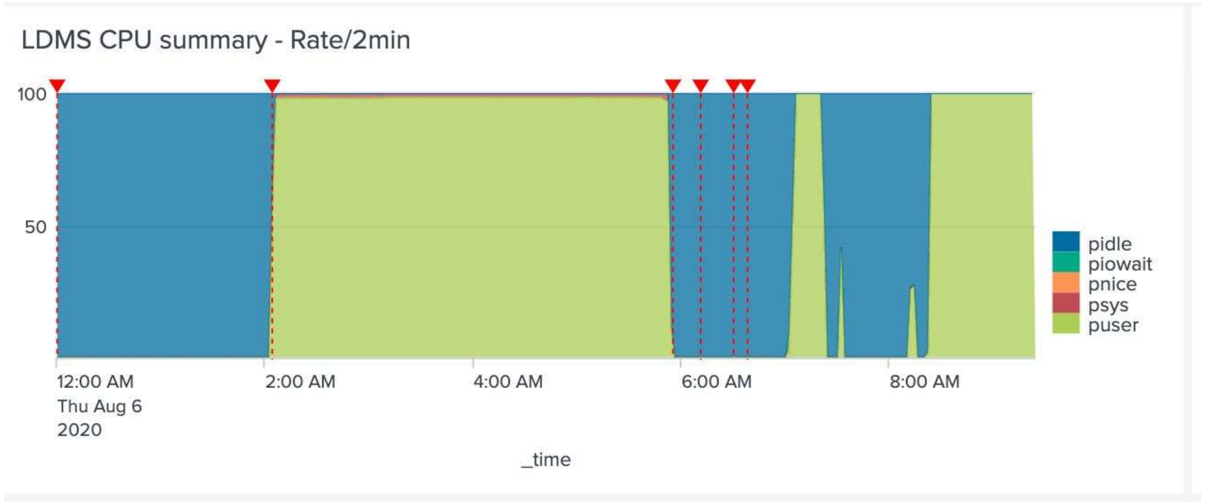
**Mentors**
Daniel Illescas, Mike Mason

## Objective

The objective of this project is to make a Comparative Analysis between popular open-source monitoring software. We are testing four metric collection tools: Lightweight Distributed Metric Service (LDMS), Telegraf, Fluent-bit, and Node Exporter. This project has three phases to help understand the footprint they can have running on a cluster. First phase is to test CPU overhead. The second phase is to test over all network traffic each software produces. The final phase is to run benchmarks on a cluster to test the impact each software has on a HPC job.

## Metrics, How are they Useful?

By default the Linux kernel has the ability to collect and store metrics on each individual node. Metric software enables us to collect these metrics in a centralized location which allows us to correlate and analyze the data.These metrics can span from cpu usage, network info, to power usage. By collecting metrics we are able to get a snapshot of how a node is currently working. And get an understanding how user jobs are using the systems.



However, the priority of the HPC cluster is to run scientific programs, therefore all metric collecting software we use has to be lightweight to minimize impact on the applications.

## LDMS Pro Con

LDMS is a project designed to specifically collect metrics on HPC systems. It's written in C to be lightweight and has an active community of HPC focused contributors. We are currently running LDMS on our production clusters.

Pro

- Lightweight, Written in C
- Robust number of HPC specific plugins
- RDMA transport
- HPC specific tool

Con

- Limited documentation
- Unstable
- Not backwards compatible
- Difficult to build source and package
- Complex configuration
- Limited number of output plugins

## Node Exporter Pro Con

Node Exporter is part of the Prometheus monitoring stack. The project was started to help gain insight into microservices. It is written in go and designed to have a small footprint. Node Exporter is written in Go and compiles into a single binary with no external dependencies, and requires a very minimal memory footprint.

Pro

- Good Documentation
- Robust configuration options
- Simple configuration
- Integrates well with modern monitoring stacks

Con

- Output is only an HTTP endpoint

## Telegraf Pro Con

Telegraf is a plugin-driven server agent for collecting and sending metrics and events from databases, systems, and IoT sensors. Telegraf is written in Go and compiles into a single binary with no external dependencies, and requires a very minimal memory footprint.

### Pro

- Robust plugin architecture
- Easy configuration
- Multiple output plugins
- Good documentation
- Plugins to process, aggregate, and serialize data

### Con

- Large configuration files

## Fluent-Bit Pro Con

Fluent-Bit is an open source and multi-platform Log Processor and Forwarder which allows you to collect data/logs from different sources, unify and send them to multiple destinations. Fluent-Bit is written in C, and has a pluggable architecture supporting around 30 plugins.

Pro

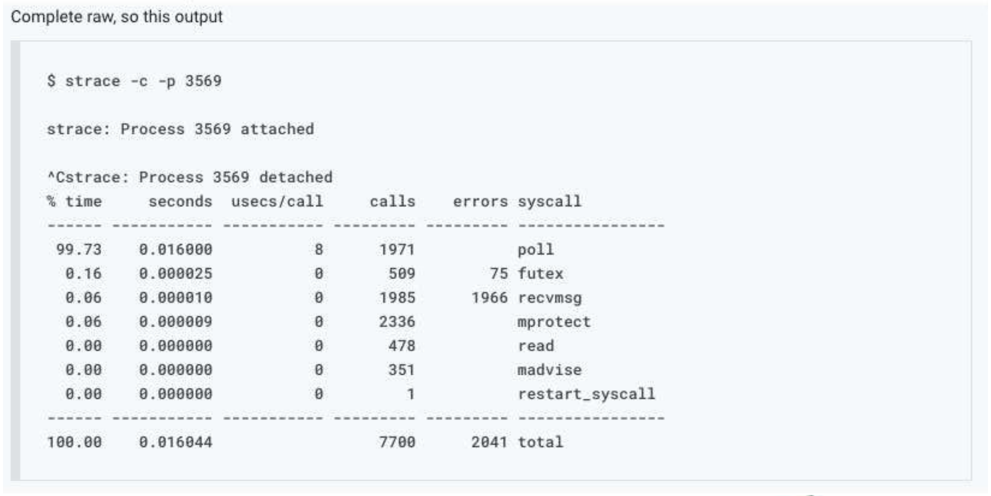
- Lightweight, Written in C
- Good Documentation
- Flexible tools to gather information
- Simple configuration

Con

- Log based rather than metrics based
- Need cmake3 to build
- Large configuration files

## Results - Explanation

- To test each software's CPU usage, we selected four metric sets that all four tools can collect.
- Run the software on a node with three different pull times 1s, 5s, 10s
- Setup a remote server to pull/receive data with the different software outputs
- We ran each test for 10 minutes, and 10 times each to reduce the significance of outliers
- We used the "strace -c -f" command to get a total runtime of each software and child processes
- Because each test runs for 10 minutes we can use the total time from strace to get the percent of CPU time used


Complete raw, so this output

```
$ strace -c -p 3569

strace: Process 3569 attached

^Cstrace: Process 3569 detached
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 99.73    0.016000           8      1971           poll
  0.16    0.000025           0       509        75 futex
  0.06    0.000010           0      1985      1966 recvmsg
  0.06    0.000009           0      2336           mprotect
  0.00    0.000000           0       478           read
  0.00    0.000000           0       351           madvise
  0.00    0.000000           0         1           restart_syscall
------ ----------- ----------- --------- --------- ----------------
100.00    0.016044                  7700      2041 total
```

## Test Configurations

LDMS

- RDMA output
- Socket output

Fluent-Bit

- Stdout output
- TCP output

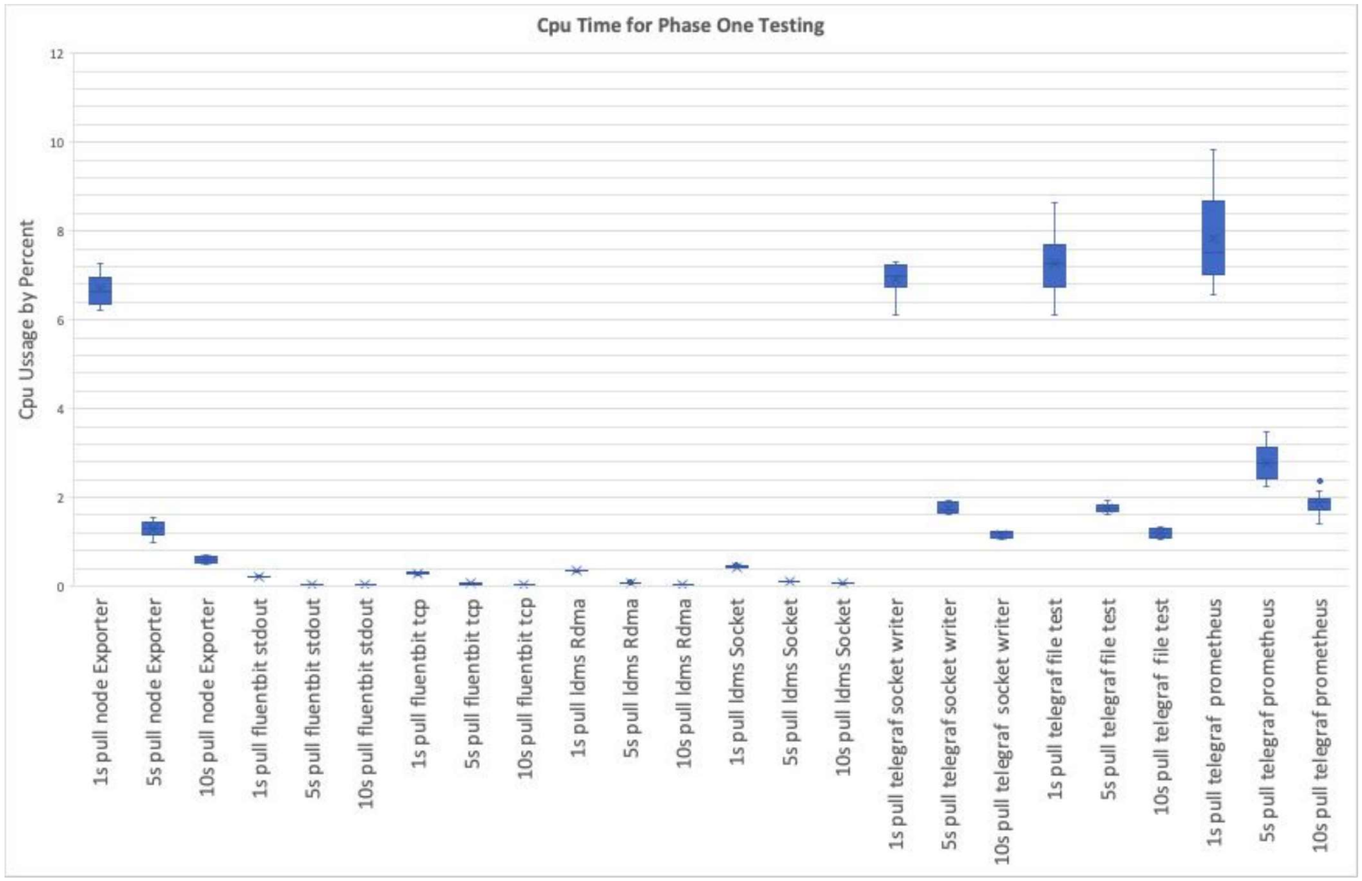
Node Exporter

- HTTP Endpoint

Telegraf

- File output
- HTTP Endpoint
- TCP Output

## Result - CPU Time Box and Whisker Chart



## Test Results: Total CPU Usage

LDMS

| | |
|---|---|
| RDMA output: | **0.0012%** |
| Socket output: | **0.0016%** |

Fluent-Bit

| | |
|---|---|
| Stdout output: | **0.00064%** |
| TCP output: | **0.00084%** |

Node Exporter

| | |
|---|---|
| HTTP Endpoint: | **0.016%** |

Telegraf

| | |
|---|---|
| File output : | **0.033%** |
| HTTP Endpoint: | **0.052%** |
| TCP Output : | **0.032%** |

## Future Work/ Last Remarks

At this point in time only the CPU overhead testing is completed, and we are halfway through the overall network saturation tests that each software creates when we run it at 1s, 10s, and 30s intervals.

From the preliminary results Fluent-Bit is the best performing software out of the four where LDMS is a close second.  However, Node Exporter and Telegraf are also shown to be lightweight using less than 0.05 % of the total cpu.

To better understand the impact this has on applications, the third phase of this project is to test the overall impact the software has when it is running during a job.

LDMS is unique because of its RDMA capability, however, when we compared it to the socket transport we found that it only improves performance by 0.0028% cpu time. And Fluent-bit is lighter weight than both of these configurations.

HIGH PERFORMANCE COMPUTING

## Questions?

HIGH PERFORMANCE COMPUTING

*Over 70 years at the forefront of supercomputing*