



SINE NOMINE
ASSOCIATES

Introducing a common interface to access AFS statistics

Marcio Barbosa
2019 OpenAFS Workshop



AGENDA

- Motivation
- Problem
- Solution
- StatsStore
- StatsStore and OpenAFS
- Other platforms
- Collectd
- Collectd and OpenAFS
- Links



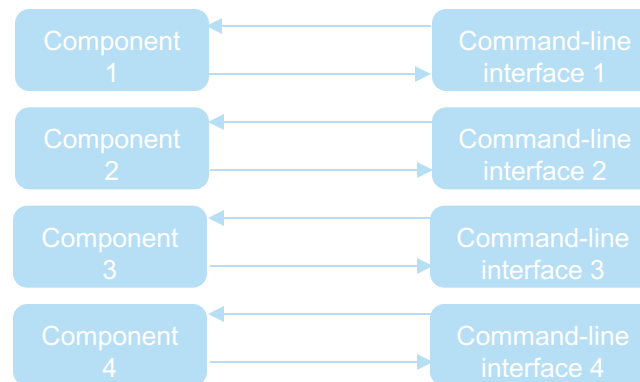
MOTIVATION

- Plenty of reasons why collecting stats about your system is a good idea;
 - Troubleshooting;
 - Tracking down bottlenecks;
 - Analyzing long-term trends;
 - Measure and monitor application performance;
 - Identify ways to optimize performance;
 - Others;



PROBLEM

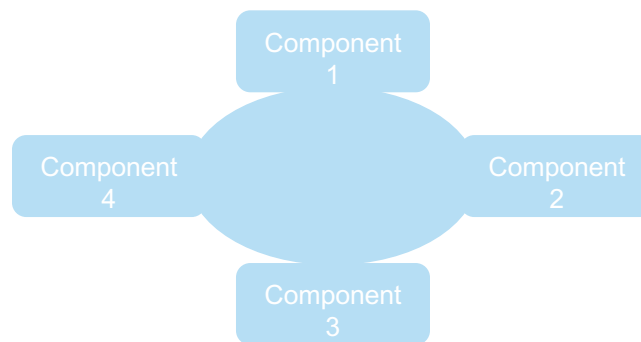
- Different interfaces to get stats from different components of the system;
- Getting stats from one component of your system might not be enough;
 - Unable to correlate data from various applications;
 - Unable to connect events to certain system states;
- Difficulties to build a holistic view of the circumstances surrounding an event;





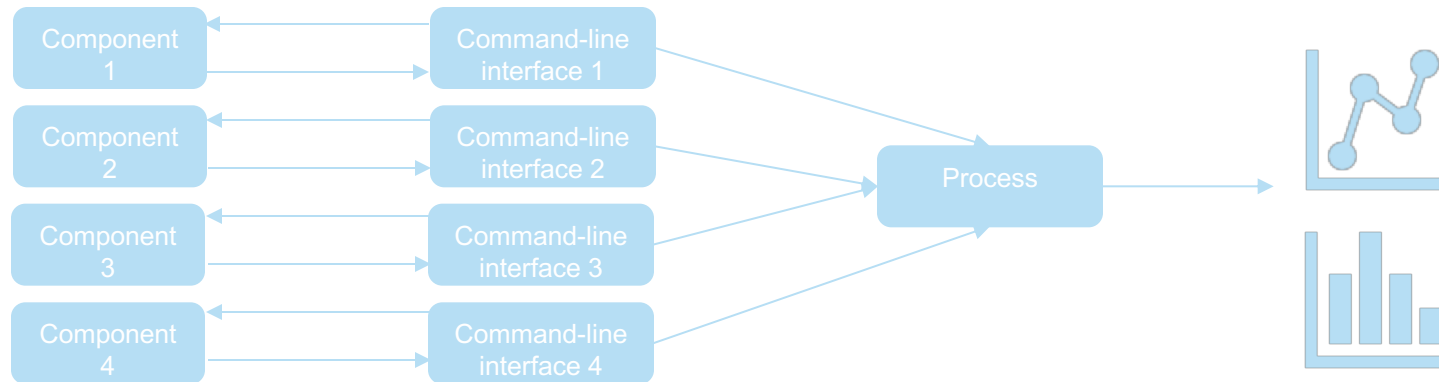
PROBLEM

- Tracking data about the system as a whole could:
 - Bring together apparently disparate pieces of system data;
 - Help us understand what the environment looked like exactly at the time of the problem;
- See our system as a system, instead of as a loose set of unrelated components;
- Unfortunately, each component provides statistics in a different way;





PROBLEM





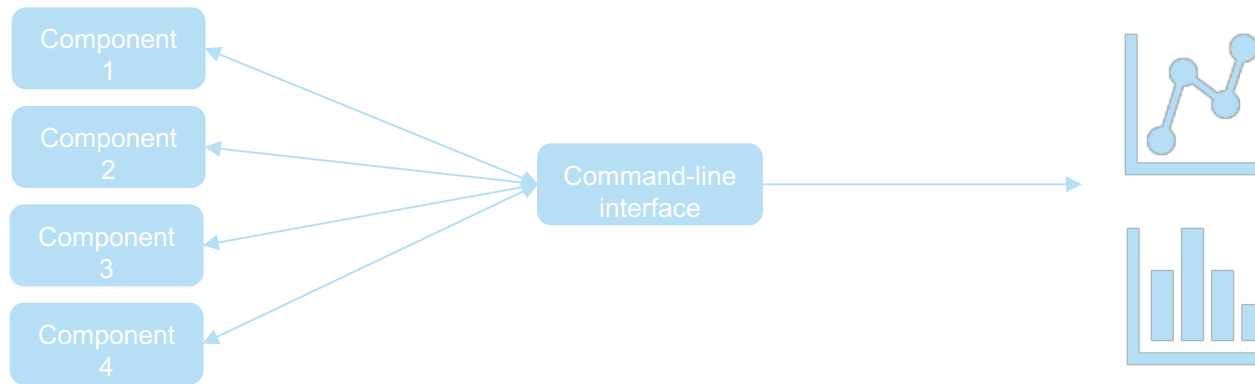
SINE NOMINE
ASSOCIATES

SOLUTION

- A common namespace to access performance data from a variety of system sources;
- Ideally, same command-line interface and API;
- A shared namespace across statistics enables you to easily explore all available information for a given system;



SOLUTION





STATSSTORE

- Introduced in Oracle Solaris 11.4;
- StatsStore unifies the broad set of Oracle Solaris observability technologies under one set of naming rules;
- This consolidated view of data is available through the interactive System Web Interface and through CLIs and APIs;

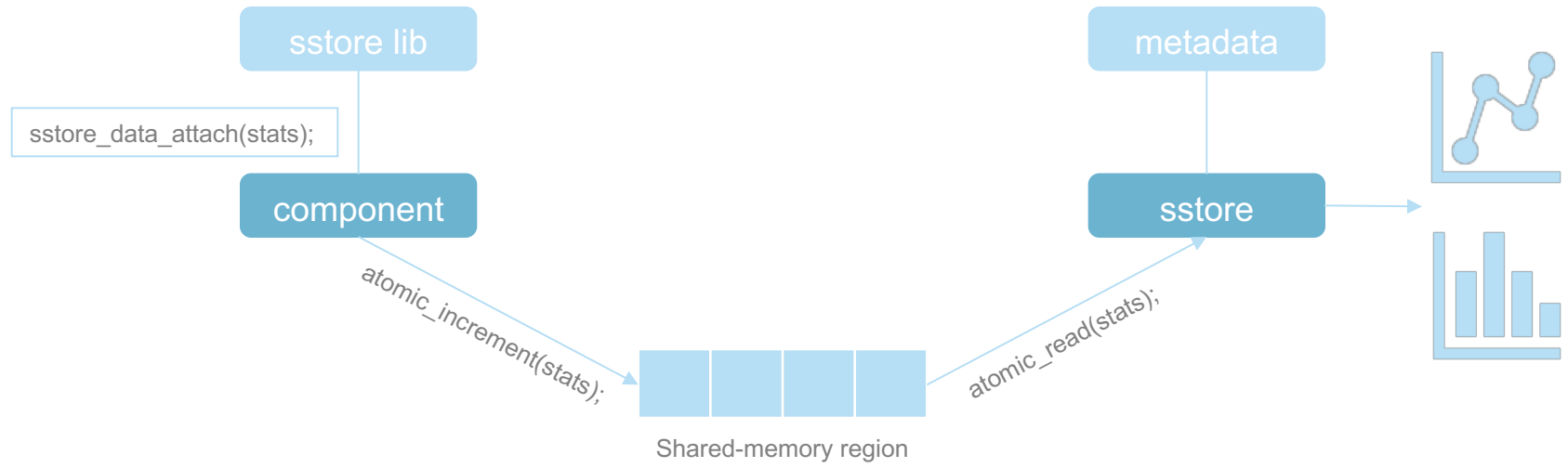


STATSSTORE

- Create metadata files that define your statistics;
- Modify your application to update values for the statistics that you created in metadata;
- Interfaces are available for both C and Python;
- Interface creates a shared memory region between sstored and the client process;
 - Supports only integer statistic values;
 - Values in this shared memory region are initialized to 0;
 - To update the statistics store, update the shared memory region array element for that statistic;



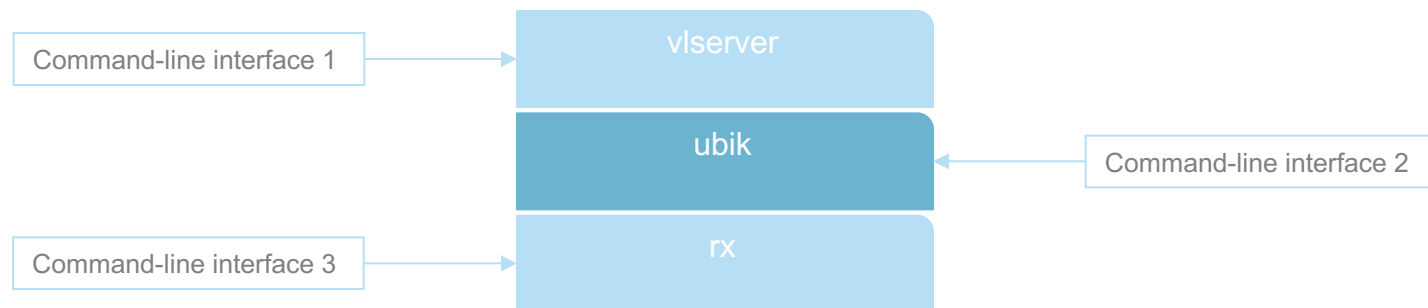
STATSSTORE





STATSSTORE AND OPENAFS

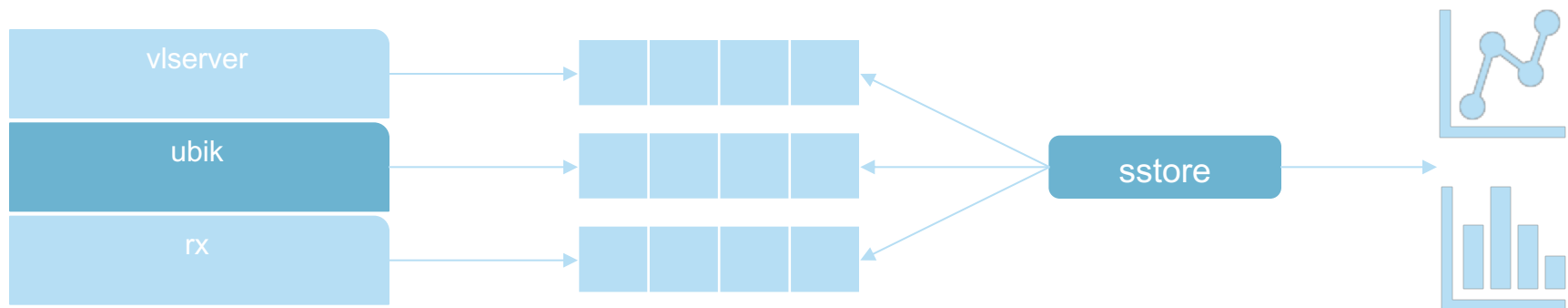
- Different ways to get statistics from different processes;
 - Different command-line interfaces;
 - Signals;
 - Fileserver, VL server, PT server, Volume server, etc.;
- Different ways to get statistics from the same process;





STATSSTORE AND OPENAFS

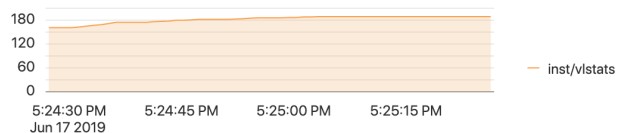
- Code refactoring:
 - Move related stats to the same struct;
 - Use stats store library to create shared memory region for each struct;
 - Update counters normally;



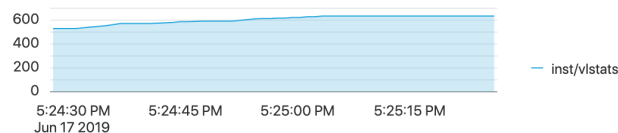


STATSSTORE AND OPENAFS

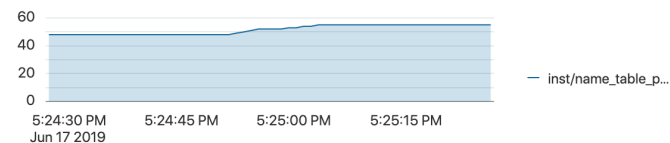
FindByNames Number of FindByNames (operations)



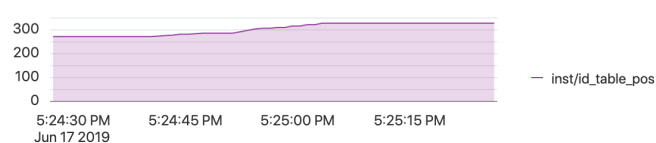
FindByIDs Number of FindByIDs (operations)



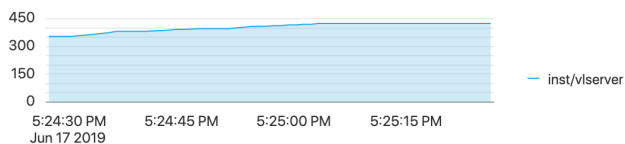
NameTablePos Number of volumes found in memhash (operations)



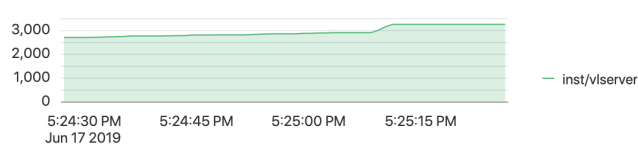
IdTablePos Number of volumes found in memhash (operations)



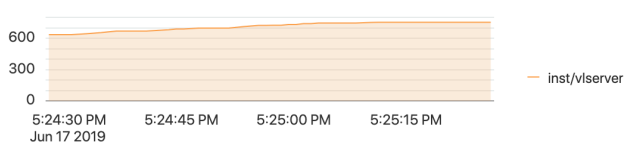
DFlushes Number of DFlushes (operations)



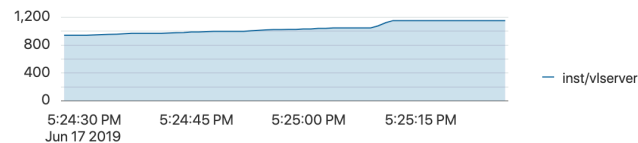
DReads Number of DReads (operations)



Data packets read Number of unique data packets read off the wire (packets)



Data packets sent Number of unique data packets sent (packets)



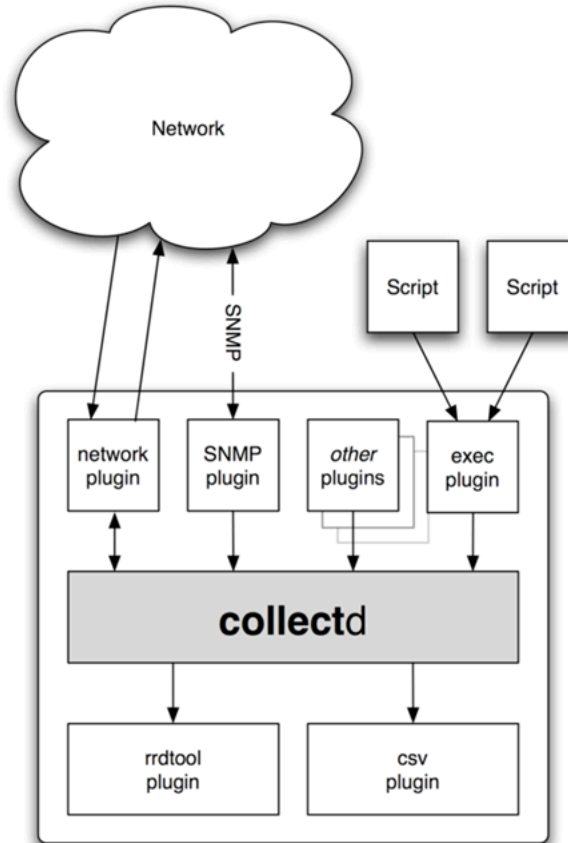


OTHER PLATFORMS

- StatsStore is Solaris specific;
- Alternative for Linux;
 - Collectd;
- Collectd is an open source daemon that collects system and application performance metrics;
- Collects, transfers and stores system performance statistics;
 - Data acquisition and storage handled by plugins;



COLLECTD





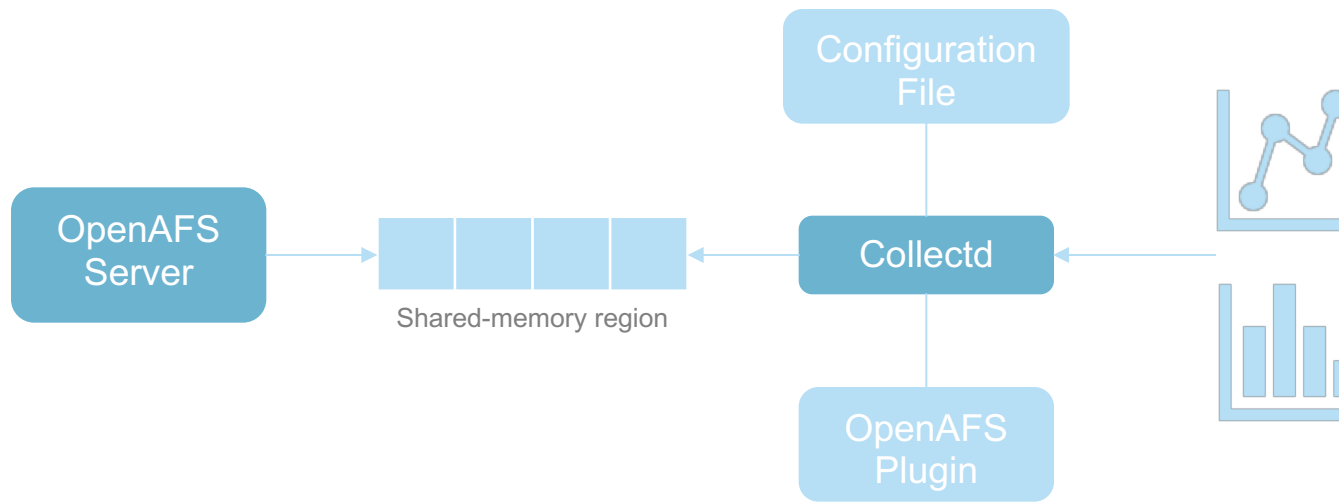
COLLECTD

- Benefits:
 - Open source;
 - Extensible;
 - Free;
 - Lightweight;
 - Lots of plugins (over 130 plugins);
 - Widely supported (Linux, Mac OS X, AIX, FreeBSD, NetBSD, OpenBSD, etc.);
 - More;



COLLECTD

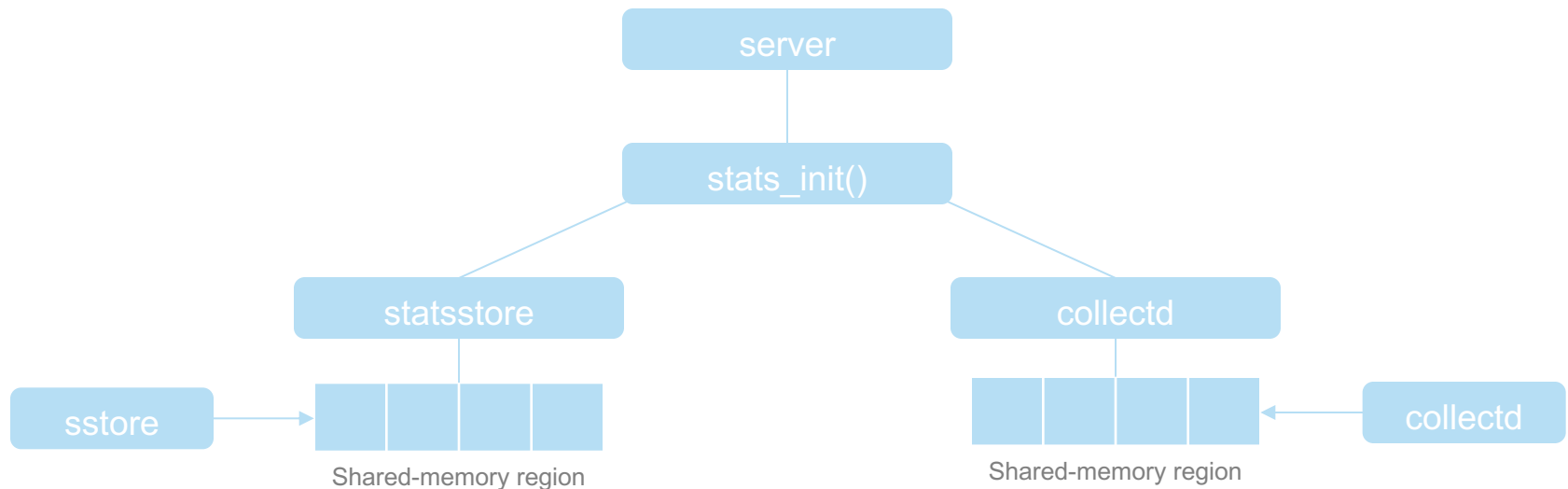
- Everything in collectd is done in plugins;
- Each plugin has their own unique settings;
- Plugin for OpenAFS developed;
 - Creates shared-memory region for each group of stats;
 - Group of stats specified in the configuration file;





COLLECTD AND OPENAFS

- OpenAFS uses the same interface used by StatsStore;
 - But with different implementation;
 - Collectd plugin: github.com/marciobarbosa/collectd/tree/mbarbosa/afs-stats-3





COLLECTD AND OPENAFAFS





SINE NOMINE
ASSOCIATES

Thank you!