Real-Time I/O-Monitoring of HPC Applications with SIOX, Elasticsearch, Grafana and FUSE

Eugen Betke, Julian Kunkel

Research Group
German Climate Computing Center
22-06-2017

DKRZ
DEUTSCHES
KLIMARECHENZENTRUM

1. Introduction
2. On-line Monitoring Framework
   - Components
   - Architecture
3. Evaluation
   - Scalability
   - Overhead
4. Summary

Introduction

Why monitoring?

Monitoring is important to find inefficient applications

What I/O levels to monitor?

- node I/O
  - Overview of total I/O traffic on a node
  - Available in user space
- file I/O
  - Filtered I/O traffic for a specific file
  - Available in user space
- mmap I/O
  - I/O traffic done by virtual memory in the background
  - Hidden in the kernel space

How do monitoring tools get data?

- Capturing of proc-files statistics
- Instrumentation code injection
  - Approach: static
  - Idea: Injection of new compiled C code into a binary executable or dynamic library file
  - Drawback: Re-compilation necessary
- Interception with `LD_PRELOAD`
  - Approach: dynamic
  - Idea: Overloading of I/O functions
  - Drawback: Statically linked functions can not be manipulated

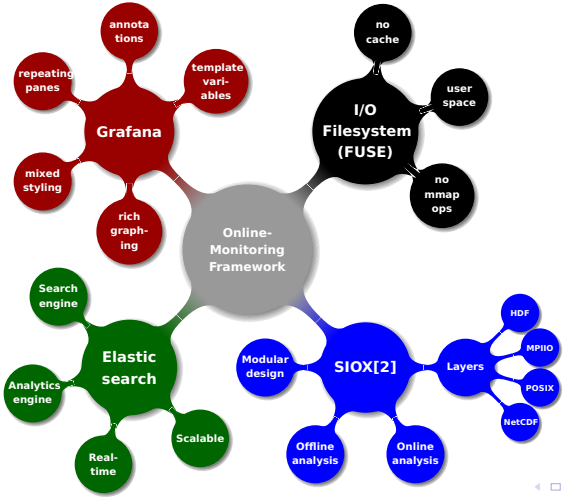
On-line Monitoring Framework Components

Visualization

Database

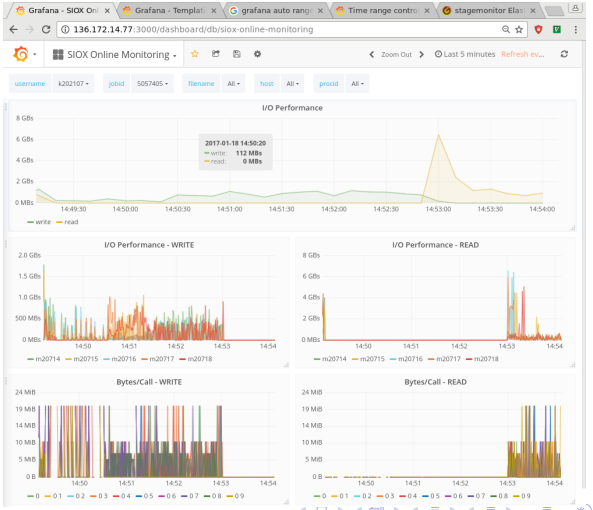
Monitoring of MMAP I/O

Instrumentation

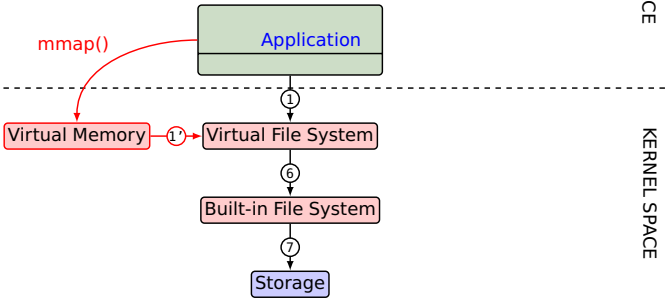


Grafana Web-Interface (User Perspective)

- Online monitoring
  - Visualization while application runs
  - Delay in order of 1sec
- Interactive web interface
  - Zoom, time shift, filtering, . . .
- Elaborated filtering
  - Based on templates
  - Auto update of templates



Existing I/O paths



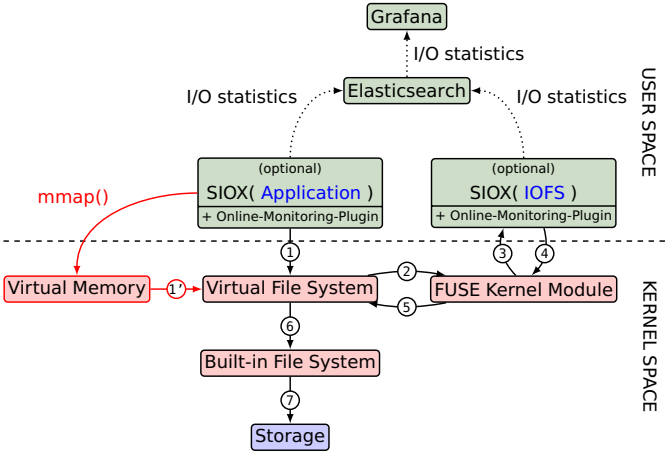
1 → 6 → 7

Existing file I/O path. Typically, supported by most instrumentation tools.

1' → 6 → 7

Existing mmap I/O path. Tools cannot trace this.

On-line Monitoring Architecture



Key features

1. Traditional monitoring of file I/O

   1 → 6 → 7

   File I/O calls are intercepted directly in application by SIOX

2. Monitoring of mmap I/O (Novelty)

   1' → 2 → 3 → 4 → 5 → 6 → 7

   Redirected mmap I/O path from kernel to user space allows SIOX to intercept the I/O calls within elevated privileges.

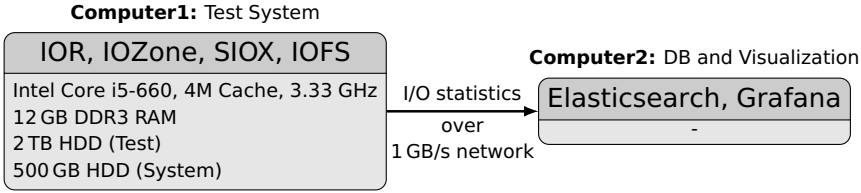
Elasticsearch performance

- Elasticsearch was deployed on an office PC
- Test setup
  - Nodes: 10
  - Processes per Node: 20
  - Metrics were
    - generated on our HPC “Mistral” with a python script
    - sent in 100 metrics packages
- Result
  - 100 x 7500 metrics per second

Package

```
{
  'metric1': '1',
  'metric2': '2',
  'metric3': '3',
  ...
  'metric100': '100'
}
```

Overhead - Test Setup

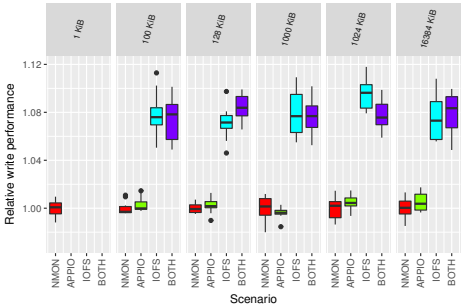


Experiment configuration

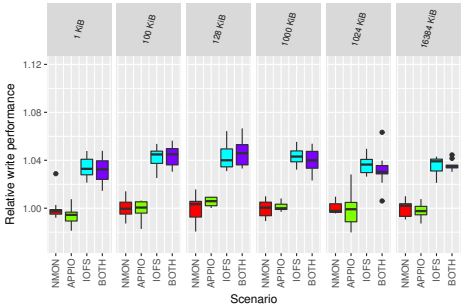
- 4 GiB test file
- 1 nodes and 1 processes per node
- Block sizes 1 KiB, 100 KiB, 128 KiB, 1000 KiB, 1024 KiB, 16384 KiB
- 10 test runs for each block size
  - IOR for file I/O
  - IOZone for mmap I/O
- Scenarios without monitoring and with monitoring (application, mount point, both)

Overhead [1/4] - Write

FILE I/O



MMAP I/O



$$P_{rel} = \frac{mean(P_{no\_monitoring})}{P_{<scenario>}}$$

Scenarios

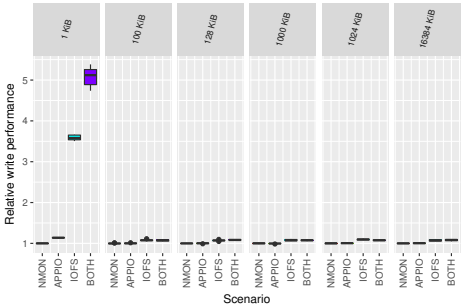
NMON	no monitoring
APPIO	monitoring of application
IOFS	monitoring of mount point
BOTH	APPIO and IOFS

Exp. configuration

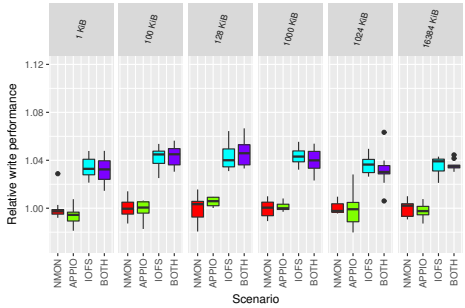
nodes/processes per node	1/1
test file	4 GiB
test runs	10

Overhead [2/4] - Write (zoomed)

FILE I/O



MMAP I/O



$$P_{rel} = \frac{mean(P_{no\_monitoring})}{P_{<scenario>}}$$

Scenarios

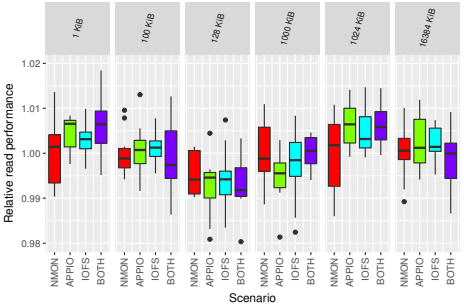
NMON	no monitoring
APPIO	monitoring of application
IOFS	monitoring of mount point
BOTH	APPIO and IOFS

Exp. configuration

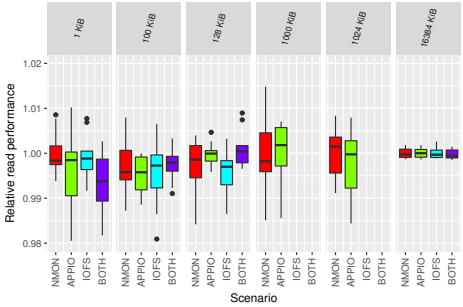
nodes/processes per node	1/1
test file	4 GiB
test runs	10

Overhead [3/4] - Read

FILE I/O



MMAP I/O



$$P_{rel} = \frac{mean(P_{no\_monitoring})}{P_{<scenario>}}$$

Scenarios

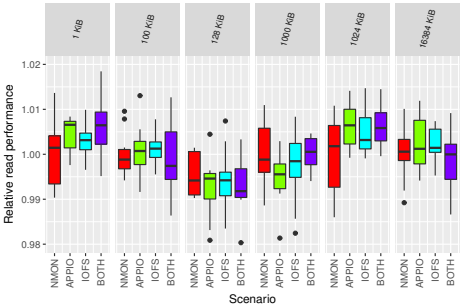
NMON	no monitoring
APPIO	monitoring of application
IOFS	monitoring of mount point
BOTH	APPIO and IOFS

Exp. configuration

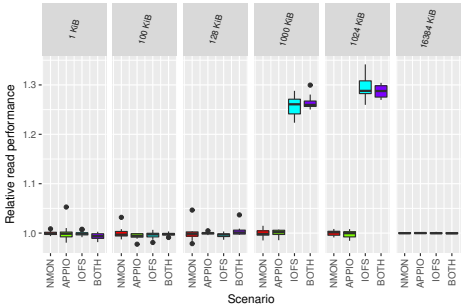
nodes/processes per node	1/1
test file	4 GiB
test runs	10

Overhead [4/4] - Read (zoomed)

FILE I/O



MMAP I/O



$$P_{rel} = \frac{mean(P_{no\_monitoring})}{P_{<scenario>}}$$

Scenarios

NMON	no monitoring
APPIO	monitoring of application
IOFS	monitoring of mount point
BOTH	APPIO and IOFS

Exp. configuration

nodes/processes per node	1/1
test file	4 GiB
test runs	10

Summary

- Non-intrusive On-line Monitoring Framework
  - is built on top of open source software: FUSE, SIOX, Elasticsearch, Grafana
  - provides near real-time on-online monitoring
  - supports file and mmap I/O
    - file I/O: Detailed information about file accesses
    - mmap I/O: Non-intrusive way for instrumenting virtual memory (novelity)
- Scalability (office PC)
  - 750,000 metrics/second
- Overhead (office PC) for monitoring of
  - applications: mostly <1%
  - I/O file system: <1% (read) and <10% (write) + outliers
- Results for our HPC “Mistral” [1] are coming soon

References

HLRE-3 "Mistral". https://www.dkrz.de/Klimarechner/hpc. Accessed: 2017-03-22.

SIOX. https://wr.informatik.uni-hamburg.de/research/projects/siox. Accessed: 2017-03-22.