



Mestrado Integrado em Engenharia Informática

Análise de padrões I/O em sistemas de gestão de dados

Pedro Miguel Borges Rodrigues

Orientadores: João Paulo, Ricardo Vilaça e Tânia Esteves

Outubro 11, 2023

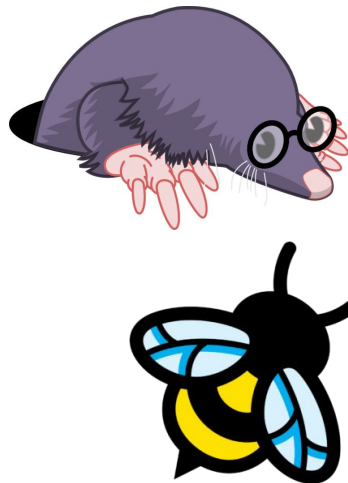
Motivação

- Crescimento da indústria tecnológica
- Grande quantidade de dados digitais
 - Garantir confiabilidade e eficiência
- Otimizações dos sistemas de gestão de dados
 - Cache
 - Replicação
 - Deduplicação



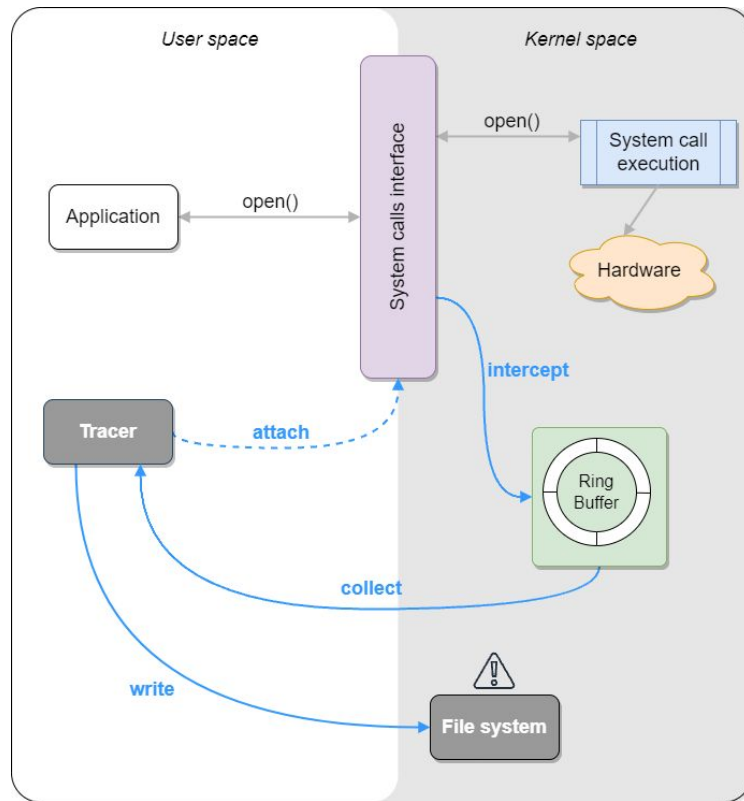
Análise de chamadas ao sistema

- Instruções para comunicar com o sistema operativo
- Visão completa de todas as instruções
- Ferramentas de tracing:
 - strace
 - eBPF
 - LTTng



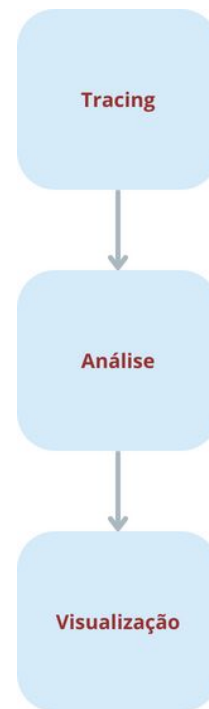
Problemas da análise de chamadas ao sistema

- Impacto no desempenho da aplicação
 - Deve ser minimizado
- Impacto no sistema de ficheiros
- Momento de análise
- Complexidade dos resultados
 - Necessidade de correlacionar



Pipeline de tracing e análise

- Componente de tracing
 - Capturar chamadas ao sistema
- Componente de análise
 - Guardar e analisar dados
- Componente de visualização
 - Visualizar dashboards e gráficos



DIO [1]

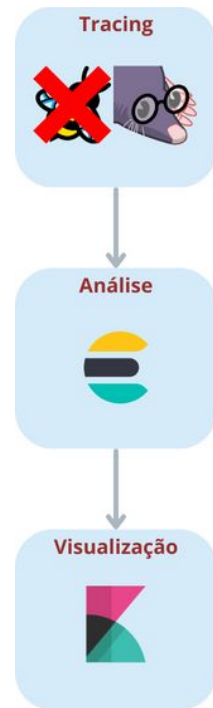
- Componente de tracing
 - eBPF
- Componente de análise
 - Elasticsearch
- Componente de visualização
 - Kibana



[1] T. Esteves, R. Macedo, R. Oliveira and J. Paulo, "Diagnosing applications' I/O behavior through system call observability," in 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Porto, Portugal, 2023 pp. 1-8

Objetivo

- Componente de tracing
 - eBPF LTTng
- Componente de análise
 - Elasticsearch
- Componente de visualização
 - Kibana

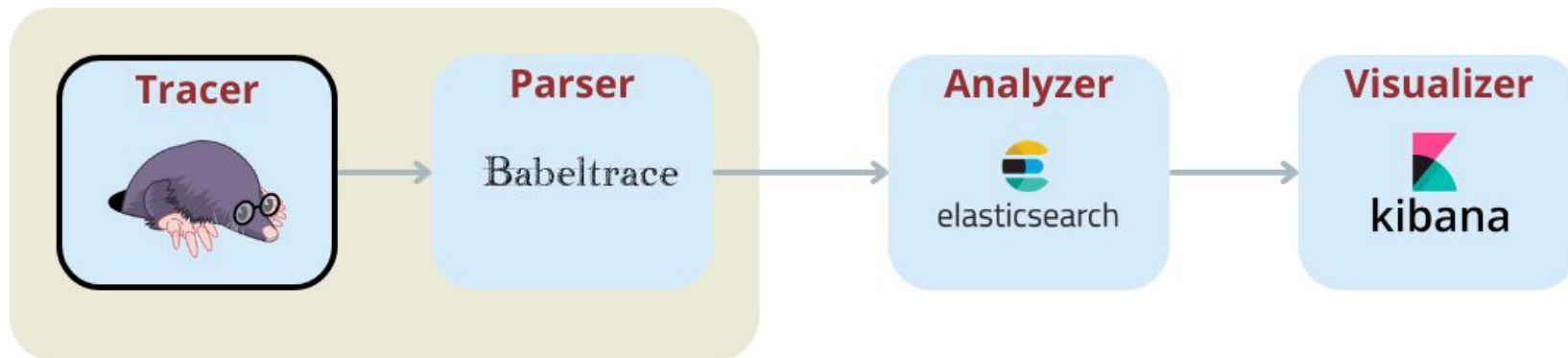


LTTng

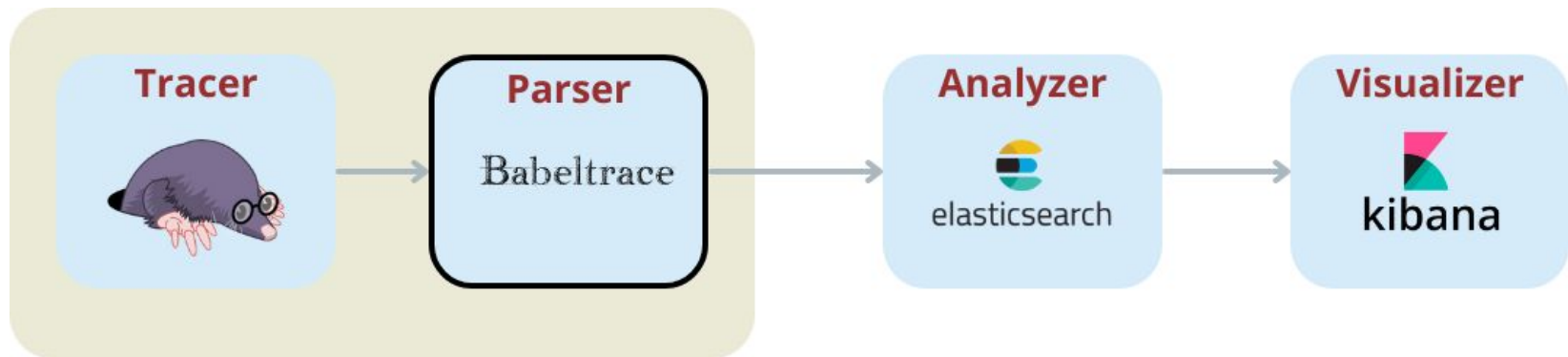
- Ferramenta de tracing
- Moderno, fácil de usar e bastante customizável
- Menor impacto no desempenho sobre a aplicação alvo



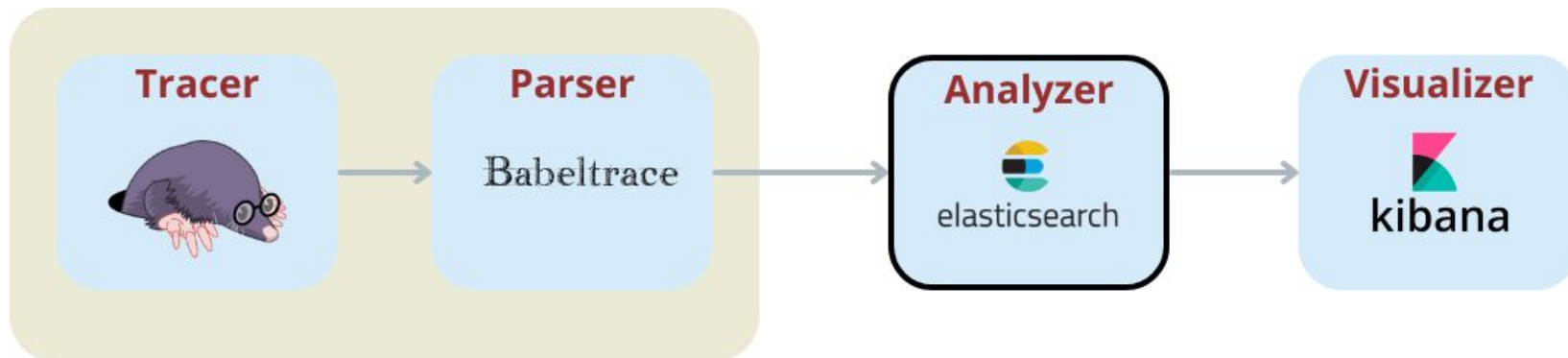
Arquitetura



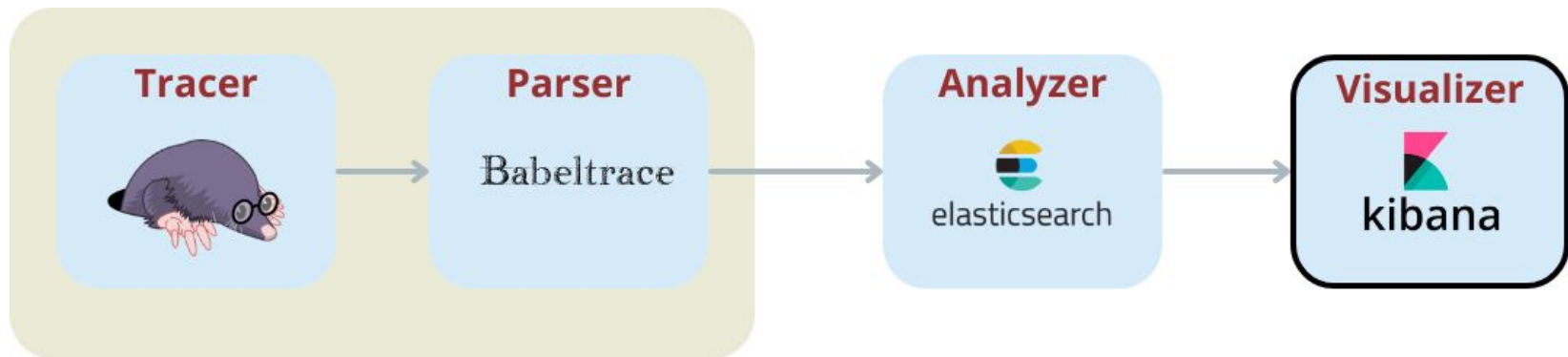
Arquitetura



Arquitetura



Arquitetura

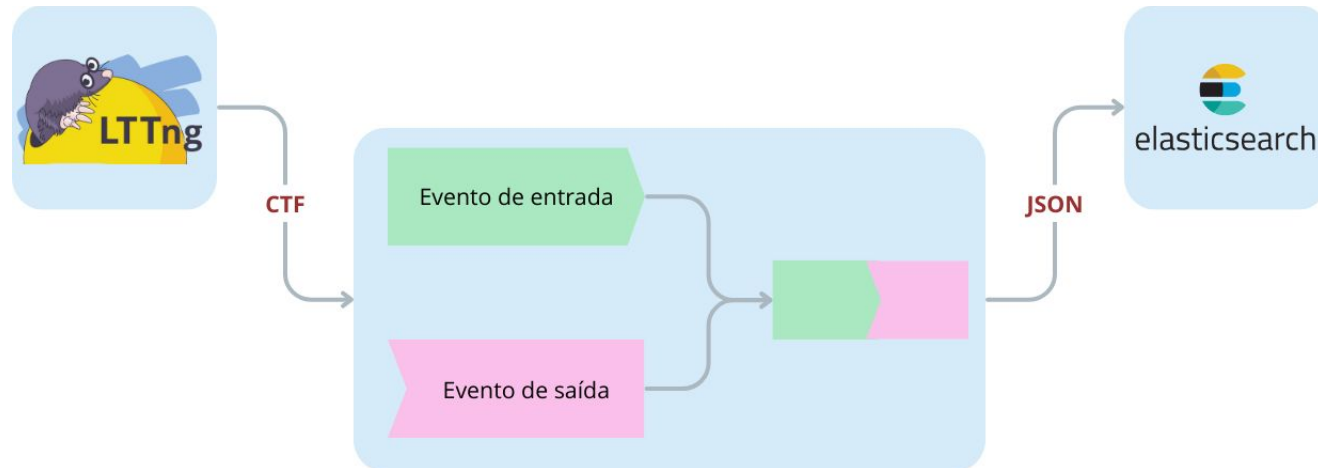


LTTng - Filtragem de processos

- Necessidade de correr o tracing sobre um comando
 - Ativar tracing para processo principal e possíveis filhos
- *lttng-noty*
 - Adicionar processos filho à lista de processos a analisar
- *lttng-trace*
 - Lançar o *lttng-noty*
 - Ativar envio de notificações
 - Executar comando recebido e correr LTTng sobre respetivo processo

Babeltrace2 - Integração com pipeline

- Gerar JSON e enviar para o Elasticsearch
- Juntar evento de entrada com respetivo evento de saída



Avaliação

- Testes de desempenho
 - Validar impacto da solução sobre aplicação alvo
- Testes funcionais
 - Validar utilidade da solução



Testes de desempenho

- Filebench com 4 threads durante 20 minutos
 - Operações I/O em 10 000 ficheiros
- 55 chamadas ao sistema colecionadas
- Captura de diferentes métricas
 - CPU
 - RAM



Testes de desempenho

- Impacto no número de operações do Filebench
 - 11% pelo LTTng
 - 30% ao integrar com a pipeline
- Percentagem de perda de eventos
 - Inferior a 0.03%
 - Menor com plugin desenvolvido



Testes funcionais

- Compreender a aplicação alvo e analisar acessos a ficheiros

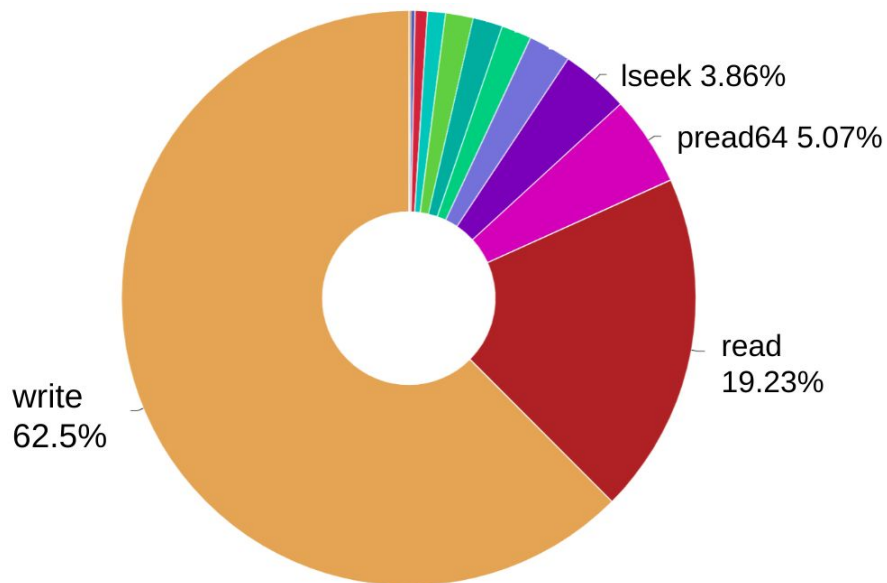


- Identificar causas de um problema conhecido

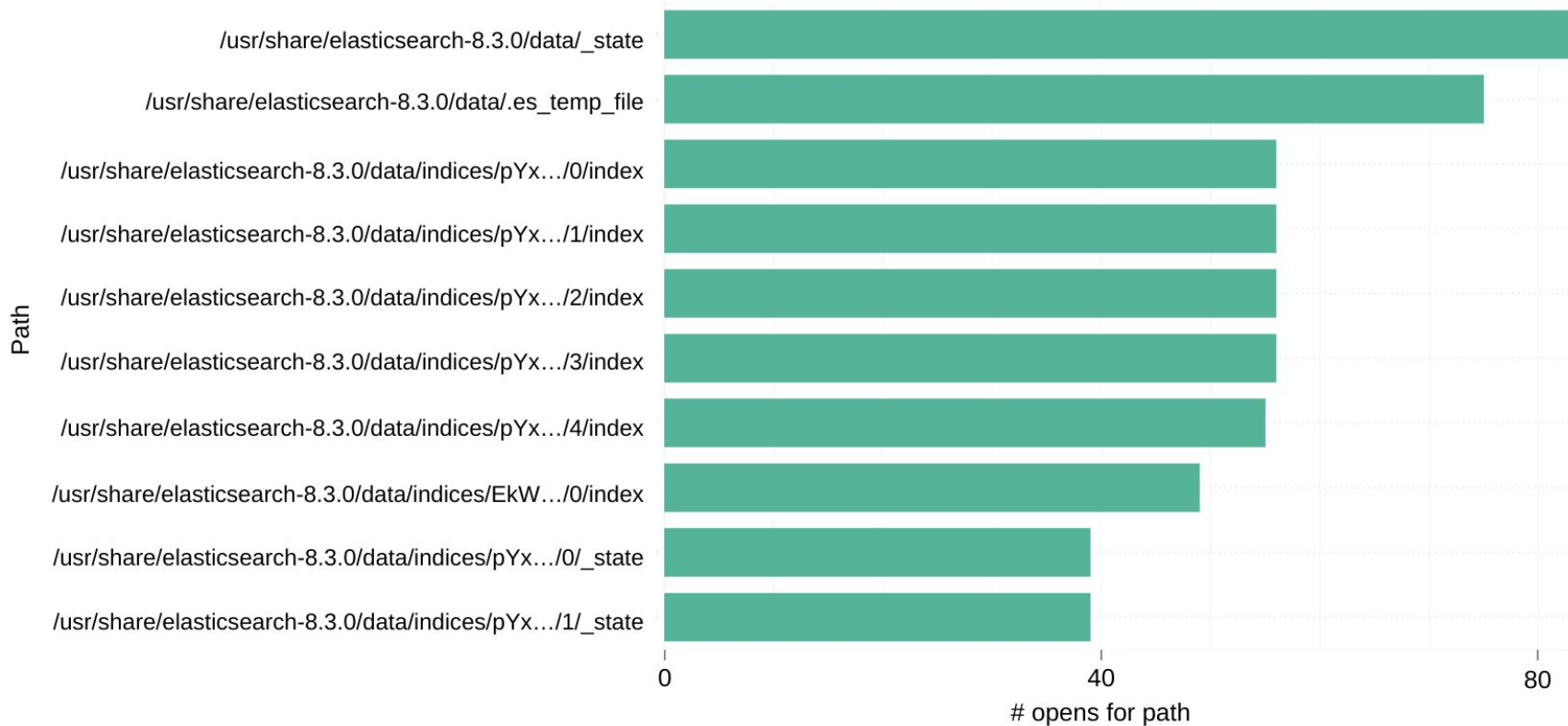


Testes funcionais - Elasticsearch

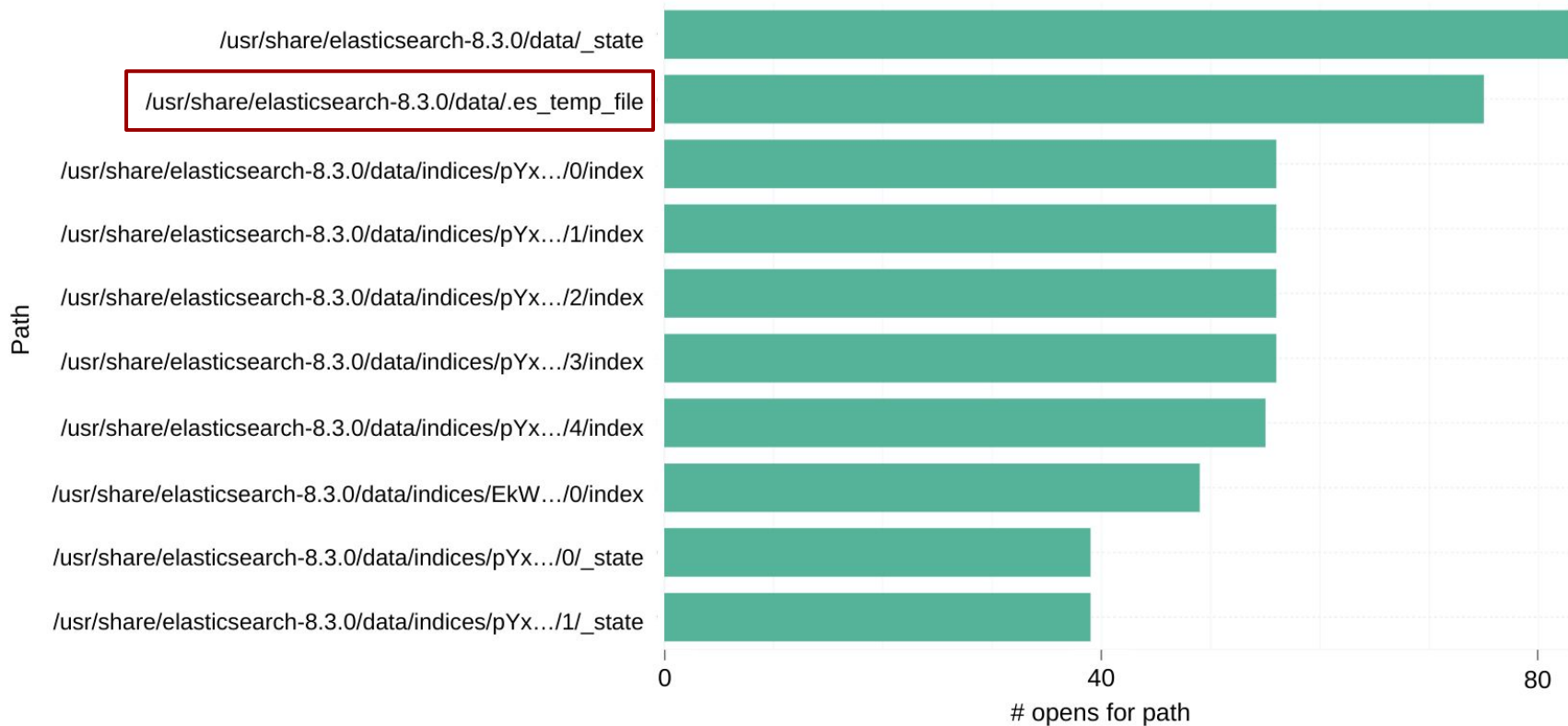
- Dataset com mais de 11 milhões de documentos e 8 clientes concorrentes
 - 1 milhão de eventos colecionados



Testes funcionais - Elasticsearch

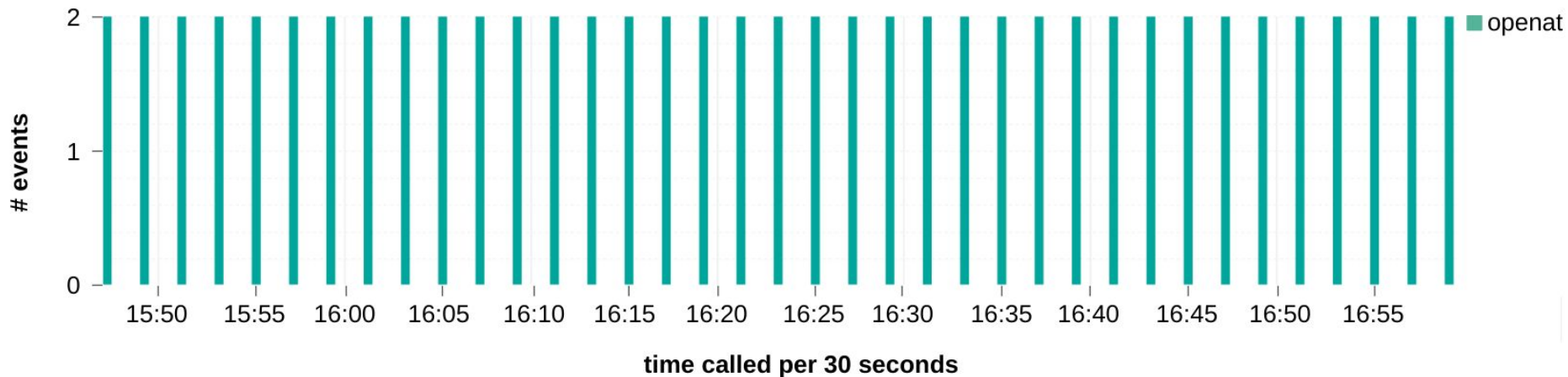


Testes funcionais - Elasticsearch

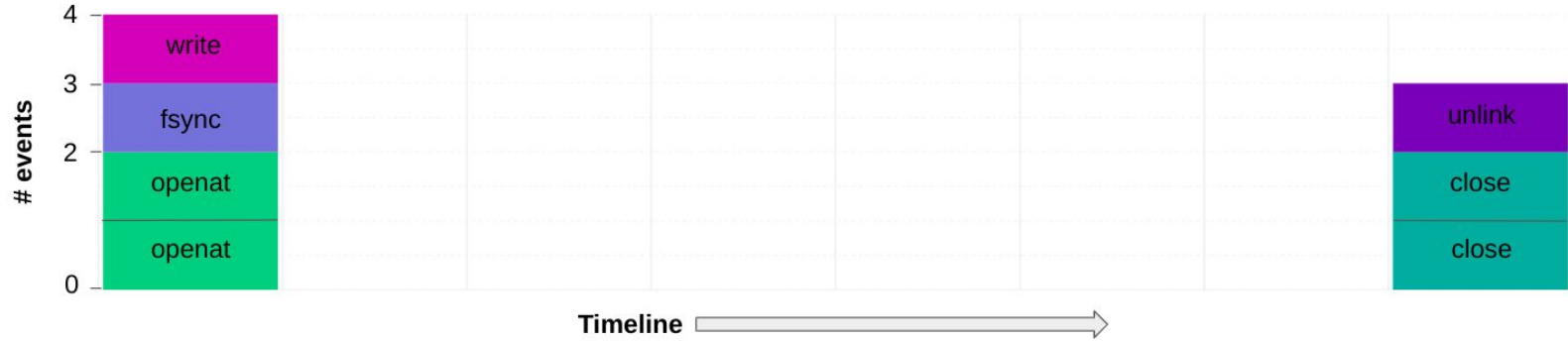


Testes funcionais - Elasticsearch

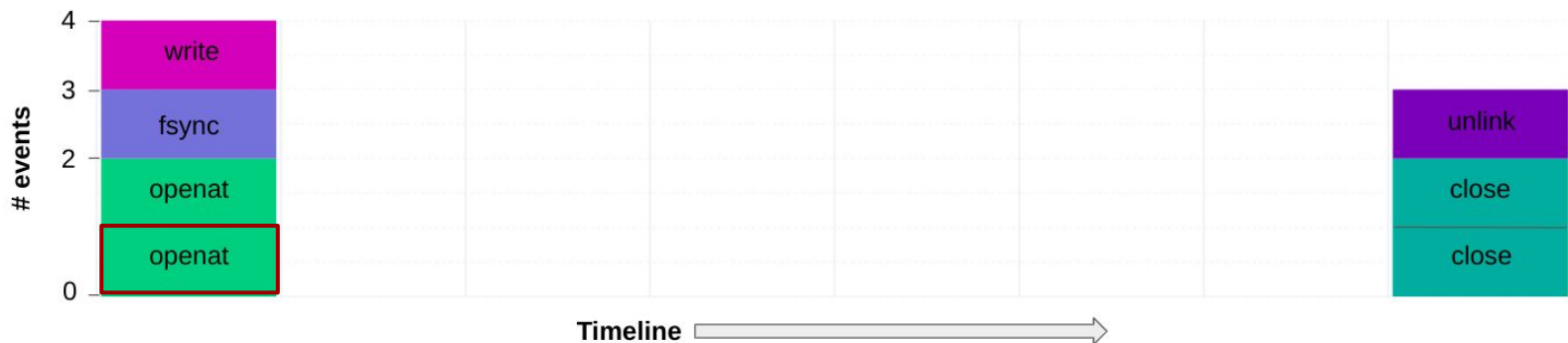
- Ficheiro é aberto 2 vezes a cada 2 minutos



Testes funcionais - Elasticsearch



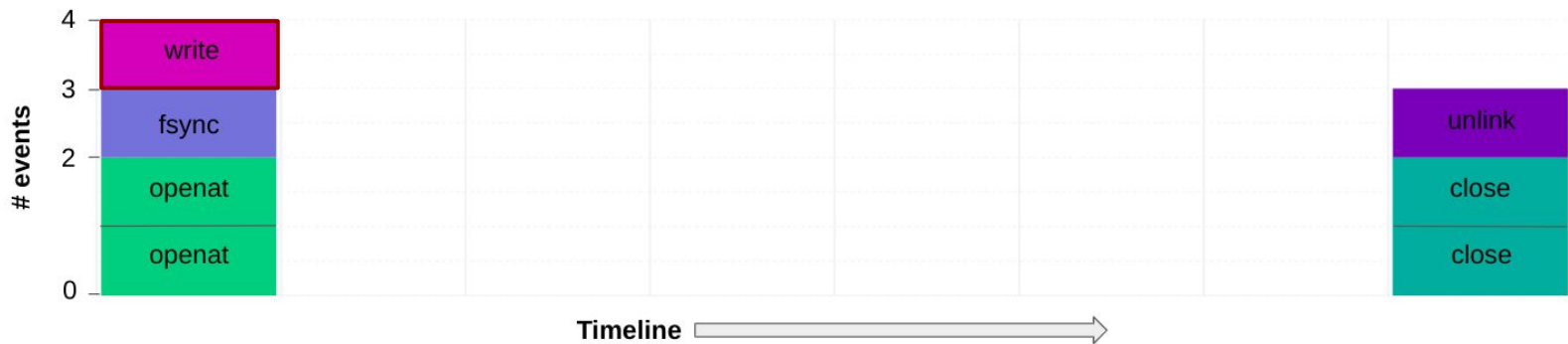
Testes funcionais - Elasticsearch



FsHealthService.java

```
168 try (OutputStream os = Files.newOutputStream(tempDataPath, StandardOpenOption.CREATE_NEW)) {  
169  
170  
171  
172
```

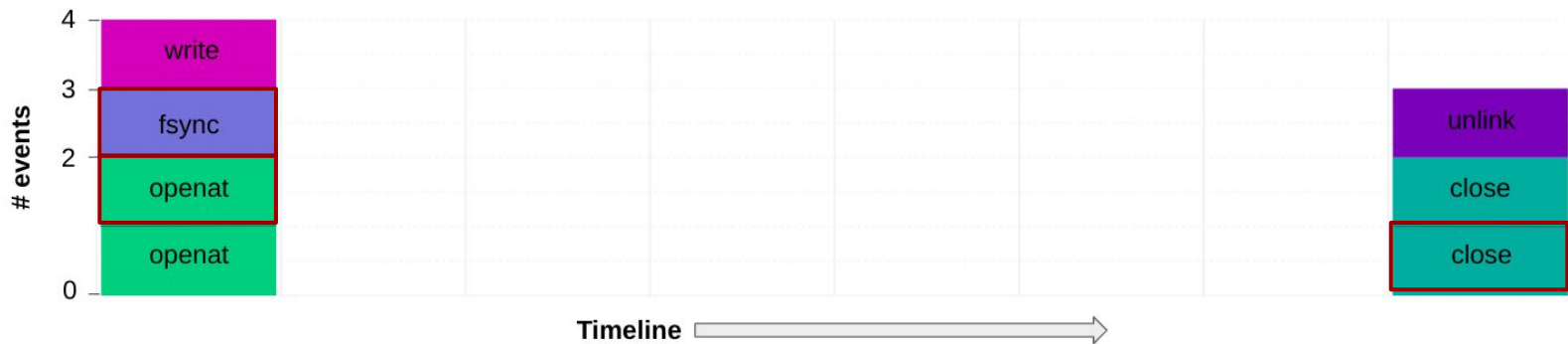

Testes funcionais - Elasticsearch



FsHealthService.java

```
168 try (OutputStream os = Files.newOutputStream(tempDataPath, StandardOpenOption.CREATE_NEW)) {  
169     os.write(bytesToWrite);  
170  
171  
172
```

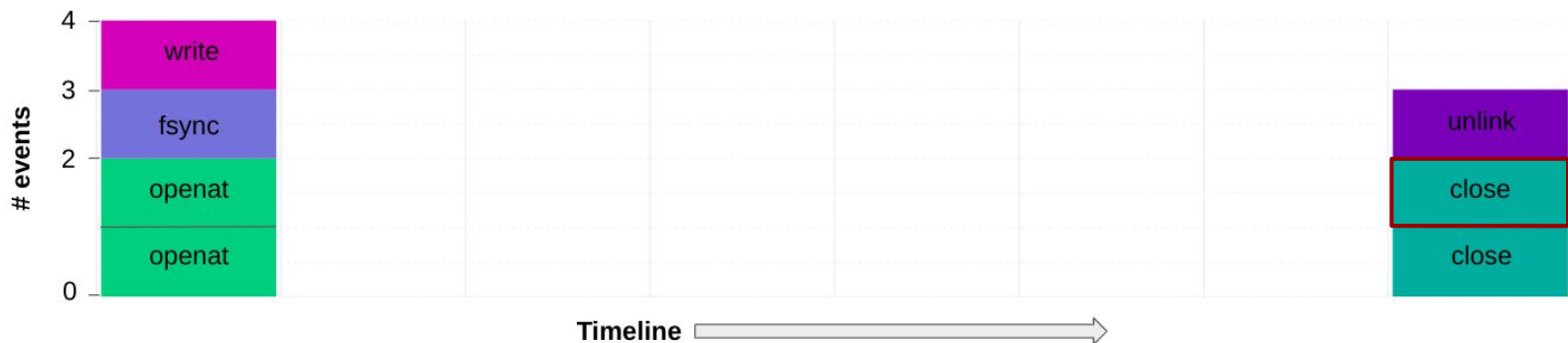
Testes funcionais - Elasticsearch



FsHealthService.java

```
168 try (OutputStream os = Files.newOutputStream(tempDataPath, StandardOpenOption.CREATE_NEW)) {  
169     os.write(bytesToWrite);  
170     IOUtils.fsync(tempDataPath, false);  
171  
172
```

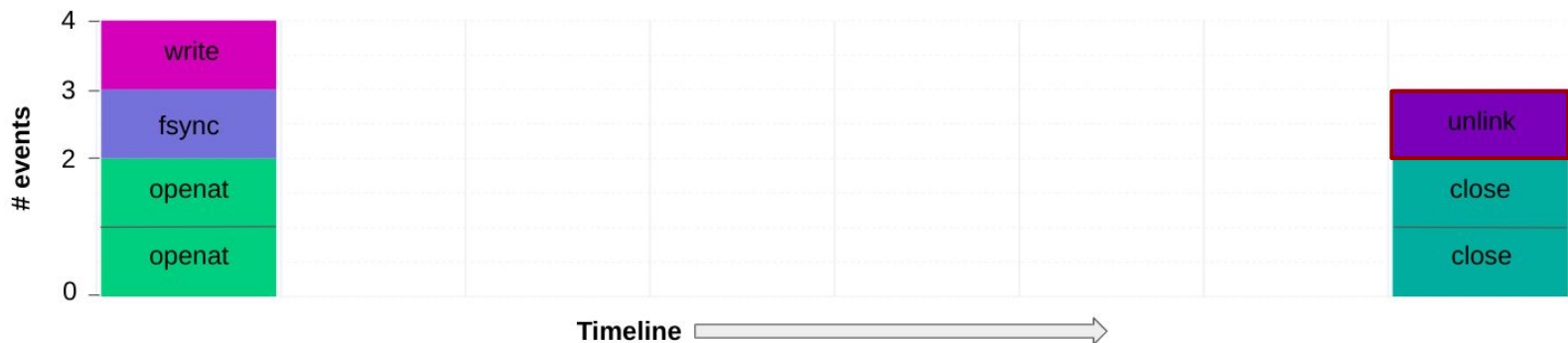
Testes funcionais - Elasticsearch



FsHealthService.java

```
168 try (OutputStream os = Files.newOutputStream(tempDataPath, StandardOpenOption.CREATE_NEW)) {  
169     os.write(bytesToWrite);  
170     IOUtils.fsync(tempDataPath, false);  
171 }  
172
```

Testes funcionais - Elasticsearch



FsHealthService.java

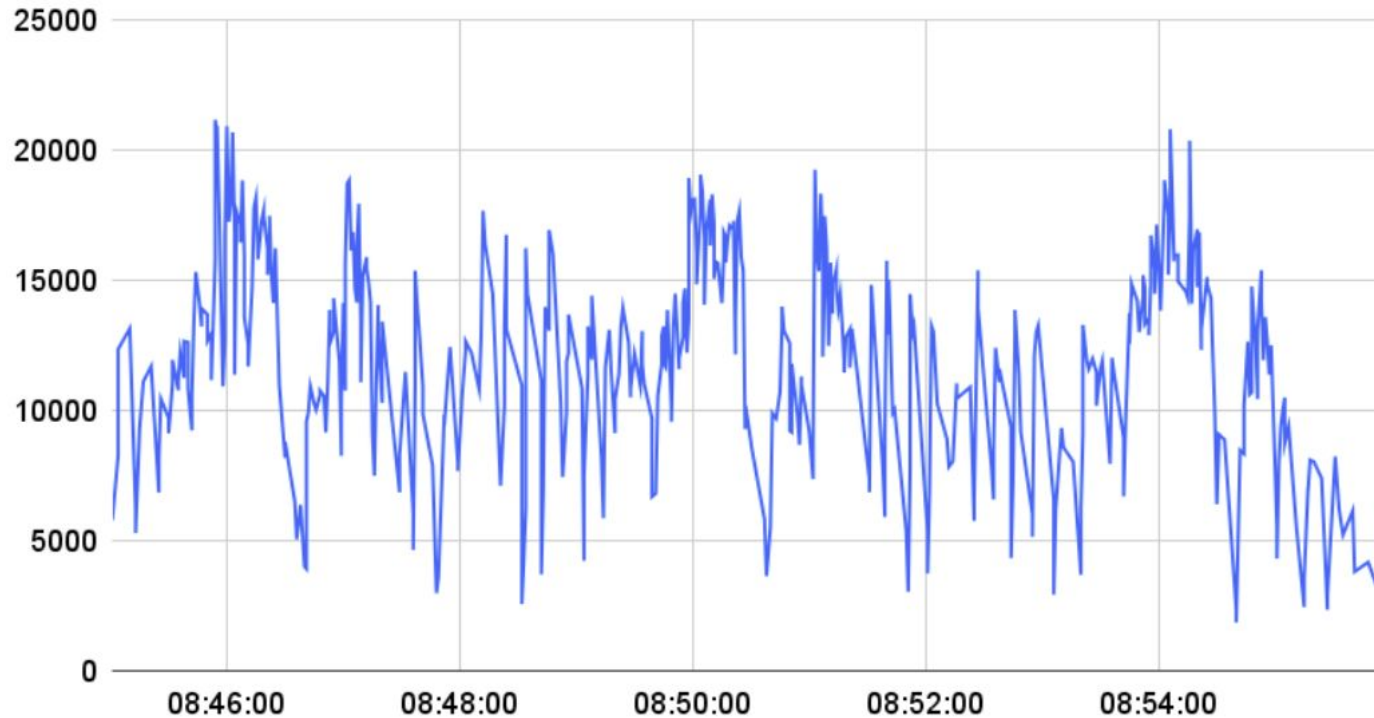
```
168 try (OutputStream os = Files.newOutputStream(tempDataPath, StandardOpenOption.CREATE_NEW)) {
169     os.write(bytesToWrite);
170     IOUtils.fsync(tempDataPath, false);
171 }
172 Files.delete(tempDataPath);
```

Testes funcionais - RocksDB

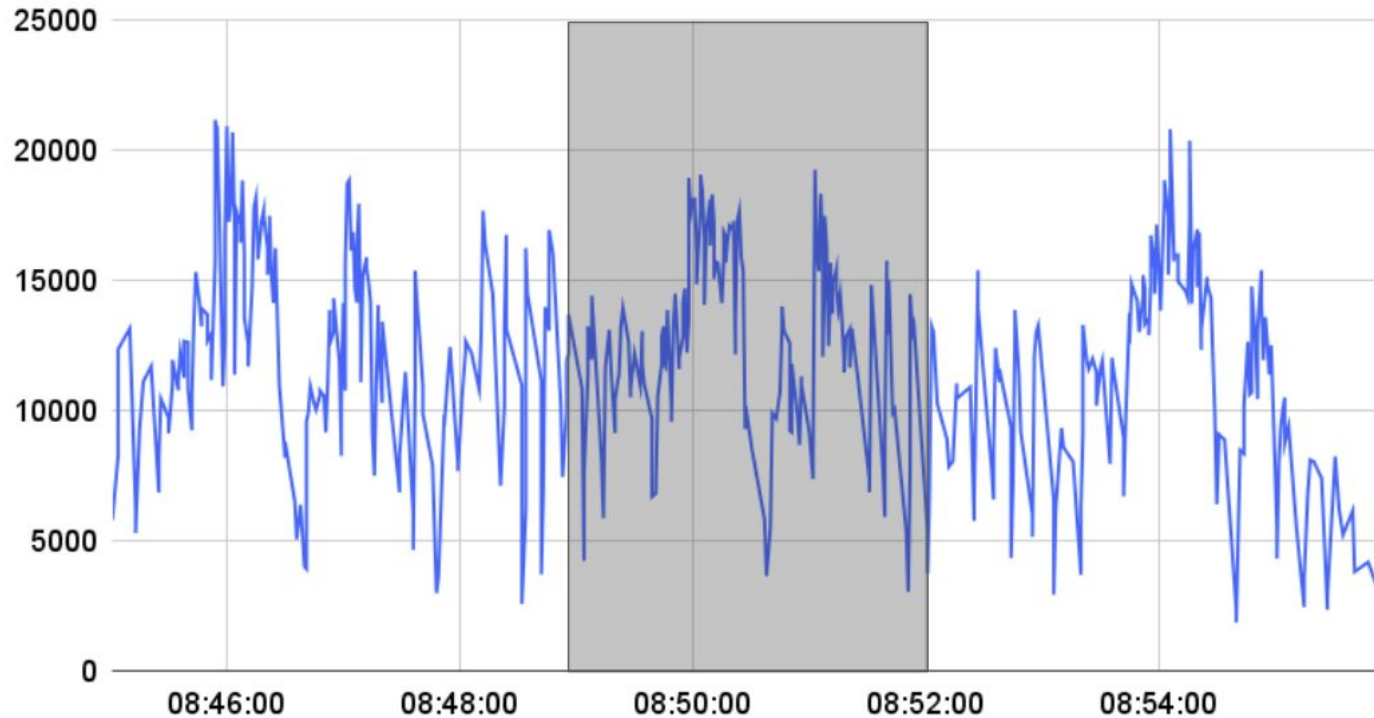
- Identificar razão de picos de latência em pedidos de clientes
- Benchmark com 8 threads de clientes
- RocksDB com 8 threads
 - 1 de alta prioridade
 - 7 de baixa prioridade



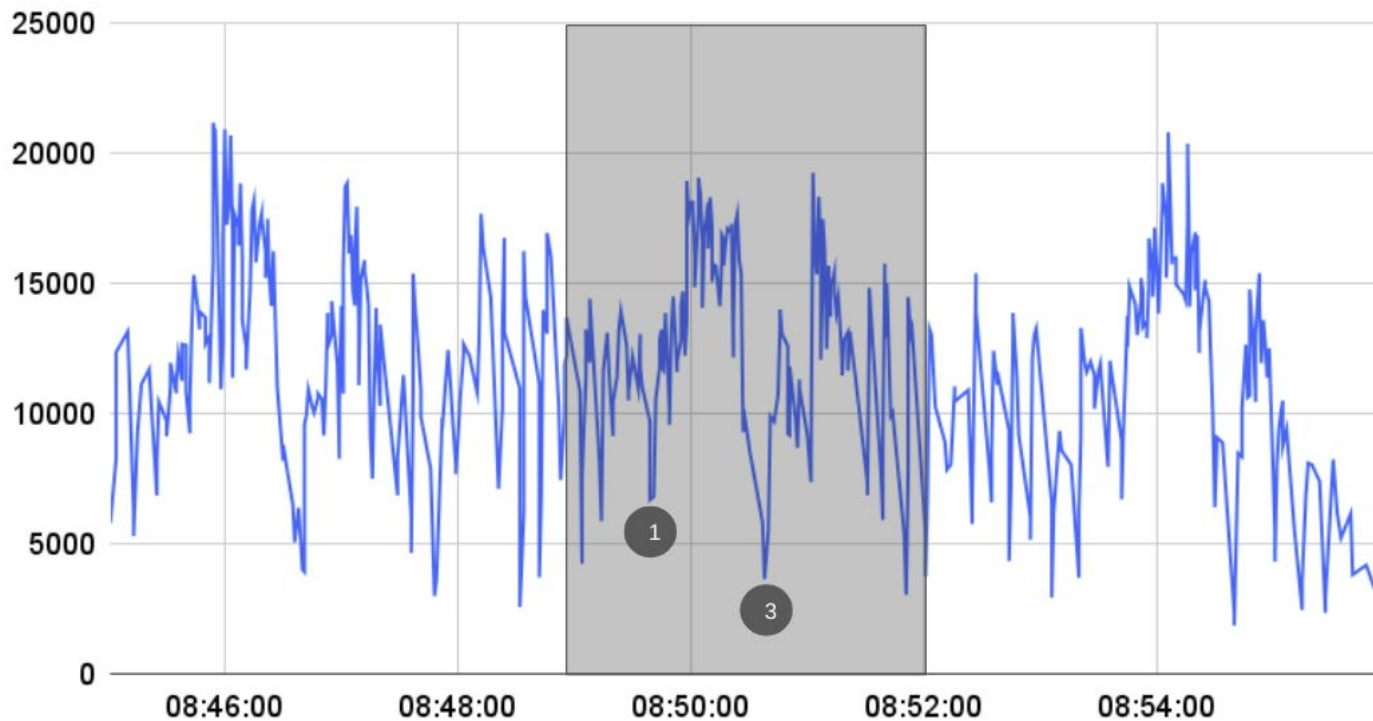
Testes funcionais - RocksDB



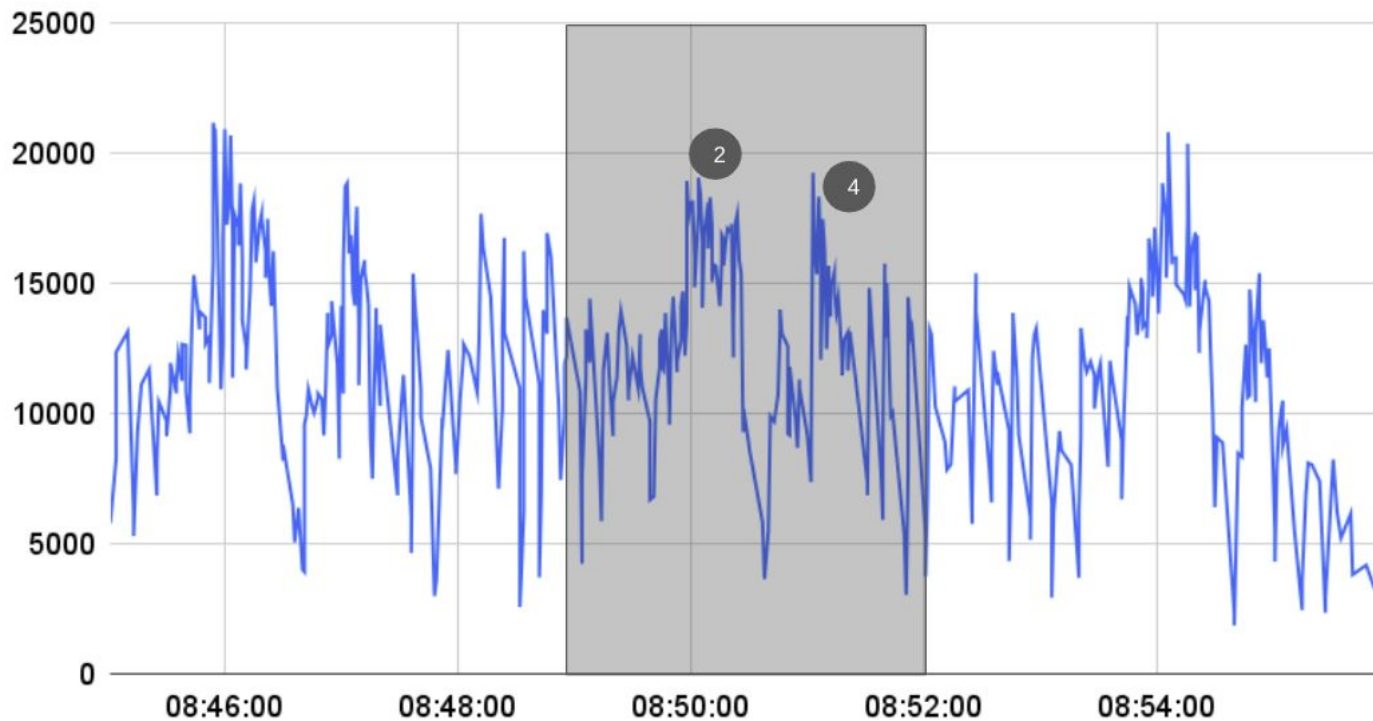
Testes funcionais - RocksDB



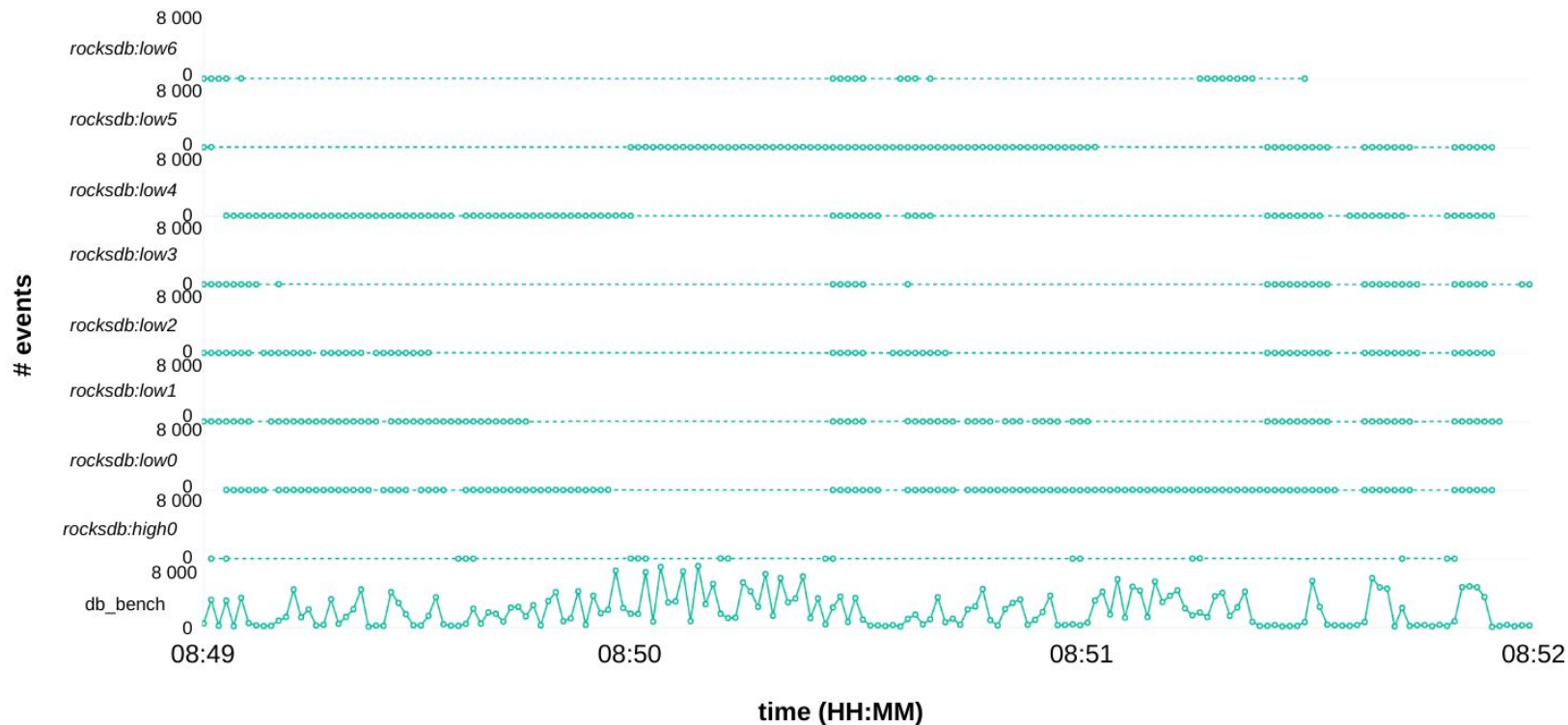
Testes funcionais - RocksDB



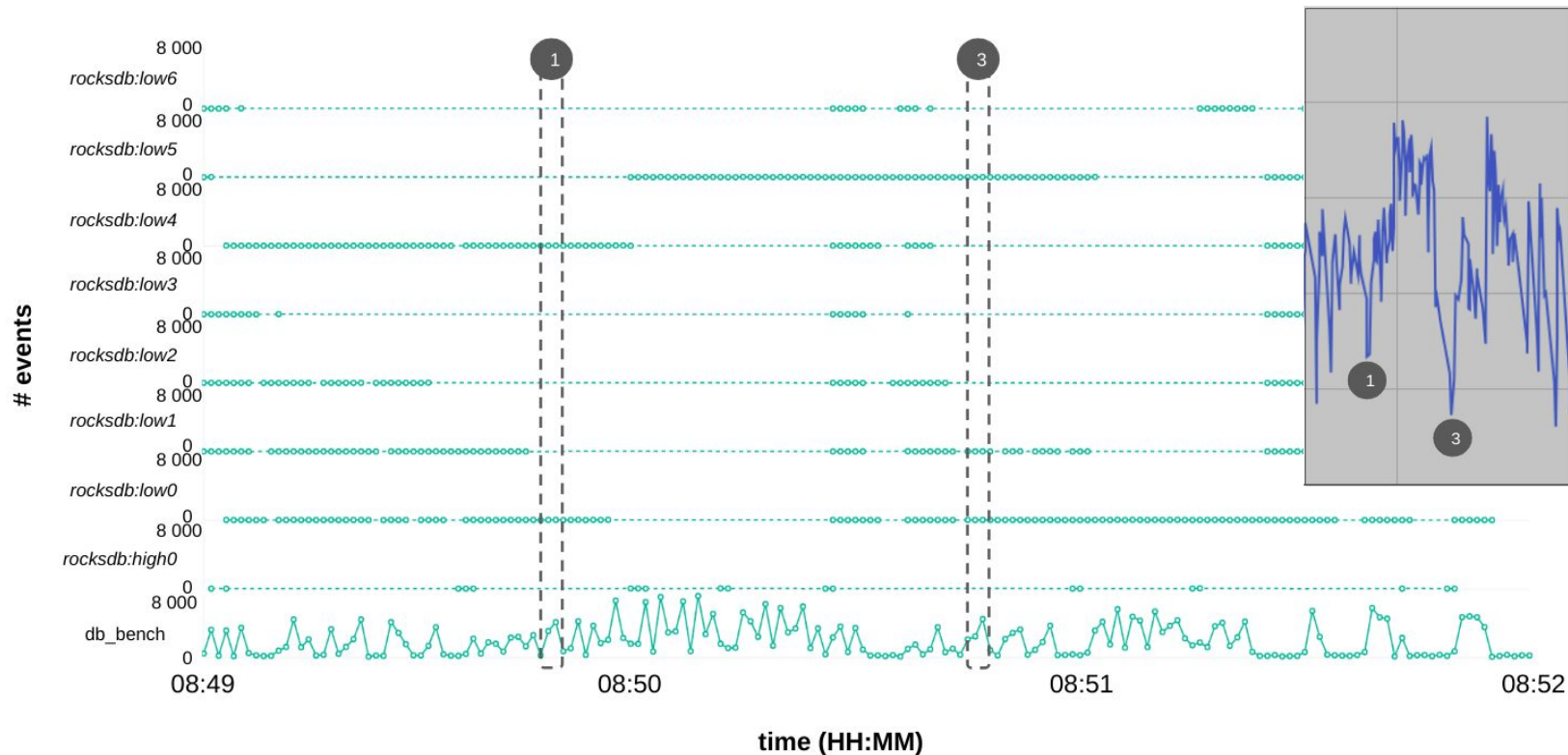
Testes funcionais - RocksDB



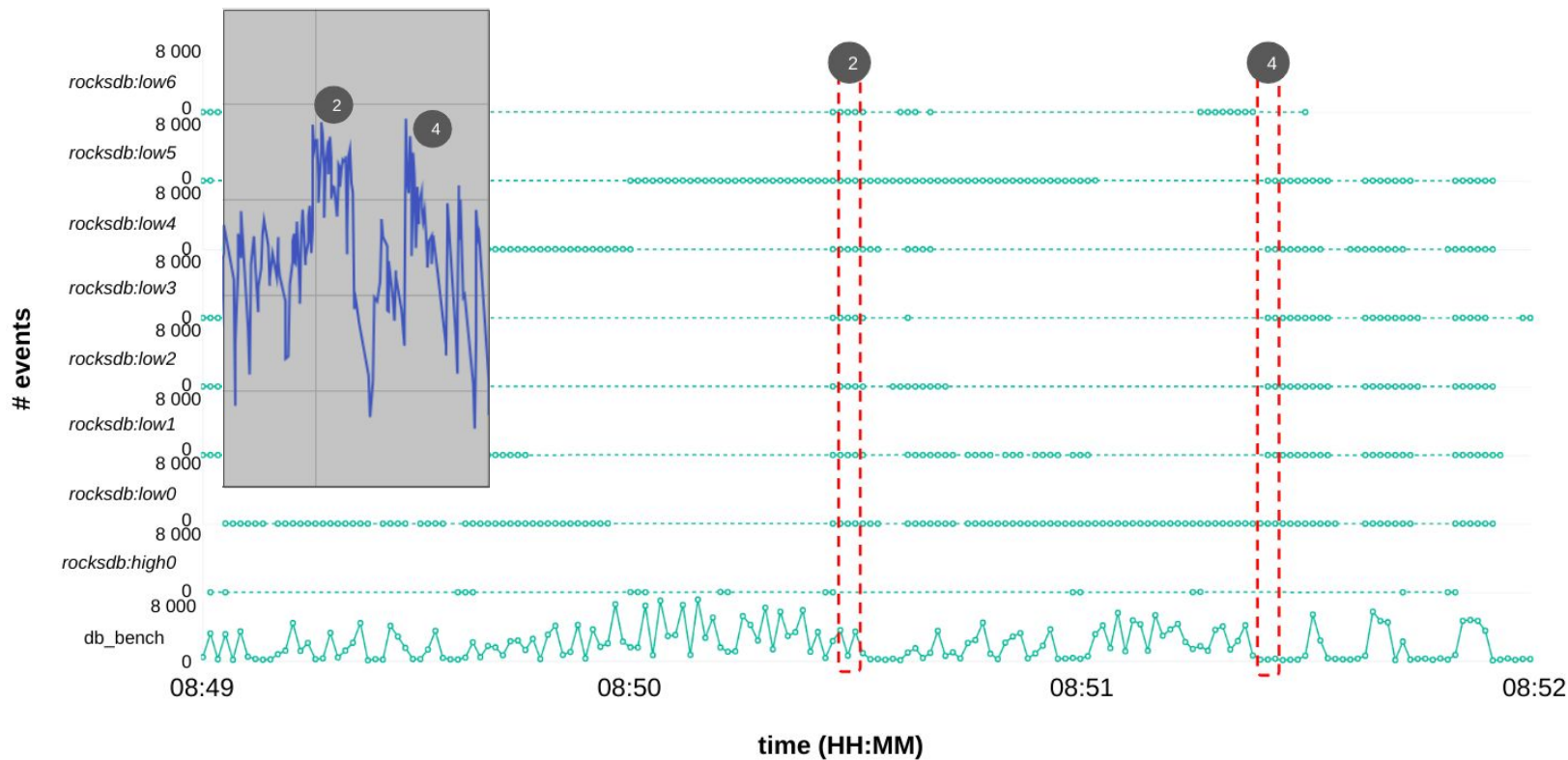
Testes funcionais - RocksDB



Testes funcionais - RocksDB



Testes funcionais - RocksDB



Conclusão

- Explorar LTTng para construir uma plataforma de diagnóstico de aplicações
- 38% impacto total e 0.004% de eventos perdidos
- Facilita a compreensão da aplicação
 - Identificar a causa para problemas conhecidos
- Possíveis melhorias
 - Adicionar informação dos ficheiros aos eventos
 - Capturar conteúdo das referências de memória
 - Detetar padrões problemáticos automaticamente





Mestrado Integrado em Engenharia Informática

Análise de padrões I/O em sistemas de gestão de dados

Pedro Miguel Borges Rodrigues

Orientadores: João Paulo, Ricardo Vilaça e Tânia Esteves

Outubro 11, 2023