

**Mestrado em Engenharia Informática**

*Engenharia dos Sistemas de Computação - 2014/2015*

Análise de Traçados com *strace*

14 de Maio de 2015

**Resumo**

Este trabalho resultará num relatório de desempenho equivalente ao que é produzido diretamente pela aplicação iozone e produzir gráficos que representem os padrões temporais de acesso aos dados, para as operações de leitura/escrita e busca (lseek).

**Fábio Gomes** pg27752

# **Índice**

[**Índice** 2](#_Toc419826071)

[Introdução 3](#_Toc419826072)

[Strace e Parâmetros de teste 4](#_Toc419826073)

[Estatísticas 5](#_Toc419826074)

[*Tempo* 5](#_Toc419826075)

[*Total de Operações E/S* 5](#_Toc419826076)

[*Banda utilizada pelo write* 6](#_Toc419826077)

[*Tamanho de Blocos de Ficheiros no write* 6](#_Toc419826078)

[*Resumo do write* 7](#_Toc419826079)

[*Banda utilizada pelo read* 8](#_Toc419826080)

[*Tamanho de Blocos de Ficheiros no read* 8](#_Toc419826081)

[*Resumo do read* 9](#_Toc419826082)

[Conclusão 9](#_Toc419826083)

# Introdução

O *strace* é uma ferramenta para a depuração de programas cujos traçados quando filtrados e analisados podem também ser usados para estudar o padrão de execução das aplicações. Monitoriza interações entre os programas e o kernel do Linux, como chamadas ao sistema, sinais e mudanças no estado do processo. O trabalho do *strace* só é possível devido ao *ptrace*.

Com a ajuda do ficheiro *refazStFd.py* disponibilizado pelo professor foi possível obter melhores informações sobre a utilização dos vários ficheiros temporários criados pelo *iozone*.

# Strace e Parâmetros de teste

Para este trabalho utilizei o strace versão 4.5.19 instalada no cluster. Primeiramente foi necessário utilizar o *strace* da seguinte forma num job sobre o iozone com 256 MB:

strace -T -ttt -o strace.out /opt/iozone/bin/iozone –R -a -i0 -i1 -i2 -i5 -g 256M

Com o ficheiro output *strace.out* passei-o pelo *refazStFd.py* para tratar dos ficheiros temporários.

/share/apps/IOAPPS/refazStFd.py < strace.out > ref.rsf

O *ref.rsf* já está tratado e então é altura de o passar pelo *strace\_analyzer* para gerar algumas estatísticas para melhor compreender o que se passou naquela execução do *iozone*.

/share/apps/IOAPPS/strace\_analyzer\_ng\_0.09.pl ref.rsf > stanREF.txt

# Estatísticas

Tendo já o ficheiro pronto para análise, é possível obter as seguintes informações.

## *Tempo*

O programa na totalidade demorou cerca de 191.96 segundos a concluir, sendo que 83.65% desse tempo foi em operações de E/S ao sistema perfazendo 160.587 segundos.

|  |  |
| --- | --- |
|  |  |
| **Elapsed Time for run** | **191.959035 (secs)** |
| Total IO Time | 160.587606 (secs) |
| Total IO Time Counter | 324461 |
| Percentage of Total Time | 83.657227% |

## *Total de Operações E/S*

As seguintes chamadas ao sistema via operações de E/S foram contabilizadas da seguinte forma:

|  |  |
| --- | --- |
| Command | Count |
| access | 1 |
| lseek | 95254 |
| stat | 404 |
| unlink | 234 |
| open | 941 |
| close | 1175 |
| creat | 117 |
| fstat | 6 |
| fsync | 1404 |
| read | 126931 |
| write | 97920 |

Como se trata de um teste de escrita em disco, é normal que os comandos *lseek*, *read* e *write* sejam os mais utilizados.

## *Banda utilizada pelo write*

Com o histórico da chamada ao sistema write, foi possível gerar o seguinte gráfico onde podemos analisar a quantidade de informação escrita por segundo (*Banda*/*Bandwidth*) ao longo da execução. Os dados obtidos resultaram em 97921 linhas, resultando num gráfico ilegível, portanto fiz uma média a cada 512 linhas para ficar à volta dos 191 segundos, tempo de duração da execução. Ficando assim um gráfico mais percetível e de melhor interpretação.

## *Tamanho de Blocos de Ficheiros no write*

A tabela seguinte representa a dispersão entre os tamanhos de blocos utilizados nas chamadas ao sistema em operações de escrita.

|  |  |
| --- | --- |
| IO Size Range | Number of syscalls |
| 0KB < < 1KB | 2901 |
| 1KB < < 8KB | 24528 |
| 8KB < < 32KB | 18396 |
| 32KB < < 128KB | 27639 |
| 128KB < < 256KB | 12285 |
| 256KB < < 512KB | 6141 |
| 512KB < < 1000KB | 3069 |
| 1000KB < < 10MB | 2868 |
| 10MB < < 100MB | 93 |
| 100MB < < 1GB | 0 |
| 1GB < < 10GB | 0 |
| 10GB < < 100GB | 0 |
| 100GB < < 1TB | 0 |
| 1TB < < 10TB | 0 |

## *Resumo do write*

Concluídas as análises anteriores à chamada *write* é possível tirar uma conclusão geral e algumas estatísticas interessantes com os valores obtidos.

|  |  |
| --- | --- |
| -- WRITE SUMMARY -- |  |
| Total number of Bytes written | 14,796,940,713 (14,796.940713 MB) |
| Number of Write syscalls | 97920 |
|  | |
| Average (mean) Bytes per syscall | 151,112.548131127 (bytes) (0.151112548131127 MB) |
| Standard Deviation | 718,723.650115704 (bytes) (0.718723650115704 MB) |
| Mean Absolute Deviation | 686,634.90476273 (bytes) (0.68663490476273 MB) |
| Median Bytes per syscall | 65,536 (bytes) (0.065536 MB) |
| Median Absolute Deviation | 142,811.593045343 (bytes) (0.142811593045343 MB) |
|  | |
| Time for slowest write syscall (secs) | 0.164958 |
| Line location in file | 304974 |
|  | |
| Smallest write syscall size | 1 (Byte) |
| Largest write syscall size | 16777216 (Bytes) |

No total foram escritos 14 796.9 MB (+/- 14 GB), num total de 97920 chamadas de escrita (como o *write*), como mostra a primeira estatística (*Total de Operações E/S*). A Mediana de Bytes por chamada ao sistema está nos 65536, valor esse presente nos blocos de teste do *iozone*, não sendo um valor estranho às estatísticas.

## *Banda utilizada pelo read*

Com o histórico da chamada ao sistema read, foi possível gerar o seguinte gráfico onde podemos analisar a quantidade de informação lida por segundo (*Banda*/*Bandwidth*) ao longo da execução. Tal como no *Write*, o gráfico gerado diretamente a partir dos dados ficaria gigantesco, portanto fiz uma média a cada 650 linhas para ficar à volta dos 191 segundos, tempo de duração da execução. Ficando assim um gráfico mais percetível e de melhor interpretação.

## *Tamanho de Blocos de Ficheiros no read*

A tabela seguinte representa a dispersão entre os tamanhos de blocos utilizados nas chamadas ao sistema em operações de escrita.

|  |  |
| --- | --- |
| IO Size Range | Number of syscalls |
| 0KB < < 1KB | 3 |
| 1KB < < 8KB | 32724 |
| 8KB < < 32KB | 24564 |
| 32KB < < 128KB | 36896 |
| 128KB < < 256KB | 16404 |
| 256KB < < 512KB | 8210 |
| 512KB < < 1000KB | 4112 |
| 1000KB < < 10MB | 3884 |
| 10MB < < 100MB | 134 |
| 100MB < < 1GB | 0 |
| 1GB < < 10GB | 0 |
| 10GB < < 100GB | 0 |
| 100GB < < 1TB | 0 |
| 1TB < < 10TB | 0 |

## *Resumo do read*

Concluídas as análises anteriores ao *read* é possível tirar uma conclusão geral e algumas estatísticas interessantes com os valores obtidos.

|  |  |
| --- | --- |
| -- READ SUMMARY -- |  |
| Total number of Bytes read | 20,131,020.718 (20,131.020718 MB) |
| Number of Read syscalls | 126,931 |
|  | |
| Average (mean) Bytes per syscall | 158,598.141651764 (bytes) (0.158598141651764 MB) |
| Standard Deviation | 749,136.288122469 (bytes) (0.749136288122469 MB) |
| Mean Absolute Deviation | 716,227.68995956 (bytes) (0.71622768995956 MB) |
| Median Bytes per syscall | 65,536 (bytes) (0.065536 MB) |
| Median Absolute Deviation | 148,001,871520747 (bytes) (0.148001871520747 MB) |
|  | |
| Time for slowest read syscall (secs) | 0.006456 |
| Line location in file | 199947 |
|  | |
| Smallest read syscall size | 832 (Bytes) |
| Largest read syscall size | 16777216 (Bytes) |

No total foram lidos 20 131 MB (+/- 20 GB), num total de 126931 chamadas de leitura (como o *read*), como mostra a primeira estatística (*Total de Operações E/S*). A Mediana de Bytes por chamada ao sistema está nos 65536, valor esse presente nos blocos de teste do *iozone*, não sendo um valor estranho às estatísticas.

# Conclusão

Com a ferramenta *strace* foi possível ter uma perspetiva diferente de como o *iozone* trabalha. Analisando de um ponto de vista de chamadas ao sistema, *syscalls*, como *lseek*, *write* ou *read*. Podendo assim obter melhores informações do seu funcionamento e até detetar erros ou anomalias.