

Fausto Guzzo da Costa	nUsp: 5230736
Filipe Del Nero Grillo	nUsp: 5378140
Vinicius Augusto Tagliatti Zani	nUsp: 5118935

Avaliação de Sistemas de Arquivos

(Ext4, ReiserFS e XFS)

São Carlos - SP, Brasil

20 de junho de 2011

Fausto Guzzo da Costa	nUsp: 5230736
Filipe Del Nero Grillo	nUsp: 5378140
Vinicius Augusto Tagliatti Zani	nUsp: 5118935

Avaliação de Sistemas de Arquivos *(Ext4, ReiserFS e XFS)*

Monografia apresentada para conclusão
da disciplina de Sistemas Operacionais da
pós-graduação do ICMC-USP em 2011

INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
UNIVERSIDADE DE SÃO PAULO
ICMC - USP

São Carlos - SP, Brasil

20 de junho de 2011

*“É impossível avaliar a força que possuímos sem
medir o tamanho do obstáculo que podemos vencer,
nem o valor de uma ação sem sabermos
o sacrifício que ela comporta.”*

H. W. Beecher

Resumo

resumo aqui!

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 9
1.1	Contexto e Motivação	p. 9
1.2	Objetivos	p. 10
1.3	Organização do trabalho	p. 10
2	Fundamentação	p. 11
2.1	Considerações iniciais	p. 11
2.2	XFS	p. 11
2.2.1	História	p. 11
2.2.2	Arquitetura	p. 12
2.3	Ext4	p. 12
2.3.1	História	p. 13
2.3.2	Ext4	p. 14
2.4	ReiserFS	p. 16
3	Métodos e Ferramentas	p. 18
3.1	IOzone	p. 18
3.2	Ambiente	p. 19
4	Planejamento e execução	p. 20

4.1	Variáveis de resposta, fatores e níveis	p. 20
4.2	Experimentos	p. 21
5	Resultados Obtidos	p. 23
5.1	Etapa 1: comparação entre Ext4 e XFS	p. 23
5.2	Etapa 2: comparação entre EXT4 e ReiserFS	p. 29
5.3	Etapa 3: comparação entre XFS e ReiserFS	p. 34
6	Conclusões	p. 39
6.1	Etapa 1: comparação entre Ext4 e XFS	p. 39
6.2	Etapa 2: comparação entre Ext4 e ReiserFS	p. 39
6.3	Etapa 3	p. 40
6.4	Considerações finais	p. 40
	Referências	p. 41
	Anexo A – Scripts utilizados	p. 42
A.1	Execução dos experimentos	p. 42
A.2	Extração dos dados do log	p. 43
A.3	Geração dos gráficos de barra	p. 44
	Anexo B – Dados coletados	p. 47
B.1	Ext4	p. 47
B.2	XFS	p. 50
B.3	ReiserFS	p. 53

Lista de Figuras

1	Detalhes de um <i>Allocation Group</i> do XFS [1].	p. 13
2	Uso de <i>extents</i> no sistema de arquivos Ext4 [2].	p. 16
3	Modelo de árvore utilizado pelo ReiserFS com blocos internos representados pelos blocos retangulares e blocos folha representados por esferas.	p. 17
4	Gráfico comparando os experimentos da Etapa 1 na operação de Escrita.	p. 24
5	Gráfico comparando os experimentos da Etapa 1 na operação de Leitura.	p. 24
6	Gráfico comparando os experimentos da Etapa 1 na operação de Leitura aleatória.	p. 25
7	Gráfico comparando os experimentos da Etapa 1 na operação de Escrita aleatória.	p. 26
8	Gráficos com as influências dos fatores calculadas para as operações avaliadas na Etapa 1: (a) - Escrita, (b) - Leitura, (c) - Leitura aleatória e (d) - Escrita aleatória	p. 28
9	Gráfico comparando os experimentos da Etapa 2 na operação de Escrita.	p. 29
10	Gráfico comparando os experimentos da Etapa 2 na operação de Leitura.	p. 30
11	Gráfico comparando os experimentos da Etapa 2 na operação de Leitura aleatória.	p. 31
12	Gráfico comparando os experimentos da Etapa 2 na operação de Escrita aleatória.	p. 32
13	Gráficos com as influências dos fatores calculadas para as operações avaliadas na Etapa 2: (a) - Escrita, (b) - Leitura, (c) - Leitura aleatória e (d) - Escrita aleatória	p. 33
14	Gráfico comparando os experimentos da Etapa 3 na operação de Escrita.	p. 34

15	Gráfico comparando os experimentos da Etapa 3 na operação de Leitura.	p.35
16	Gráfico comparando os experimentos da Etapa 3 na operação de Leitura aleatória.	p.36
17	Gráfico comparando os experimentos da Etapa 3 na operação de Escrita aleatória.	p.37
18	Gráficos com as influências dos fatores calculadas para as operações avaliadas na Etapa 3: (a) - Escrita, (b) - Leitura, (c) - Leitura aleatória e (d) - Escrita aleatória	p.38

Lista de Tabelas

1	Fatores e níveis escolhidos para a Etapa 1.	p.21
2	Fatores e níveis escolhidos para a Etapa 2.	p.21
3	Fatores e níveis escolhidos para a Etapa 3.	p.21
4	Níveis utilizados em cada um dos experimentos para a Etapa 1	p.22
5	Níveis utilizados em cada um dos experimentos para as Etapa 2 e 3 . .	p.22
6	Tabela com os valores calculados de influência dos fatores para a Etapa 1 da avaliação de desempenho separados para o tipo de operação. . . .	p.27
7	Tabela com os valores calculados de influência dos fatores para a Etapa 2 da avaliação de desempenho separados para o tipo de operação. . . .	p.32
8	Tabela com os valores calculados de influência dos fatores para a Etapa 3 da avaliação de desempenho separados para o tipo de operação. . . .	p.35

1 *Introdução*

1.1 Contexto e Motivação

A complexidade crescente presente nos computadores, motivada pelas necessidades cada vez mais apuradas de um mercado tecnológico em plena evolução, faz com que diversas funcionalidades que antigamente atendiam plenamente às necessidades se tornem obsoletas. Tais mudanças podem ser observadas no âmbito de hardware e software. Para os softwares, mais especificamente dentro do domínio de sistemas operacionais, a evolução também se mostra verdadeira, pois esse conjunto de sistemas é responsável por grande parte da integração que ocorre entre equipamentos de diversos fabricantes, além de prover abstração para que os componentes possam ser utilizados por terceiros, por via de softwares executáveis no contexto dos sistemas operacionais. Portanto, torna-se crescente a necessidade de melhores técnicas de utilização dos componentes periféricos, ao passo que a organização interna dos sistemas operacionais deve ser mantida coesa, segura e eficiente, objetivando uma evolução de longo prazo ao mesmo tempo em que as soluções para os problemas atuais são elaboradas.

No contexto de utilização de disco, o sistema operacional encapsula as diversas marcas de diversos fabricantes em chamadas de rotina simples e genéricas que permitem uma utilização transparente pela camada de aplicações. Dessa forma o desenvolvedor de aplicações para o sistema operacional em questão deve saber somente o conjunto de chamadas genéricas que lhe permite executar as operações de leitura, escrita, abertura e fechamento de arquivos. Cabe assim ao sistema operacional gerir o fluxo de informações e sua organização interna, abstraindo os arquivos lógicos — aqueles visíveis para as aplicações — em arquivos físicos — mapeados em hardware. A gestão de arquivos lógicos e seu mapeamento em partes físicas é foco de estudo desde há muito tempo no domínio de sistemas operacionais, pois é sabido que o armazenamento secundário é um grande gargalo a ser enfrentado em aplicações que o utilizam.

Assim sendo, diversas propostas têm sido feitas para gestão interna de arquivos de sistemas operacionais. No contexto de sistemas operacionais de código aberto, pode-

mos encontrar propostas de sistemas de arquivos. Nesse âmbito Ext4, XFS e ReiserFS foram lançados ao público como sistemas de arquivos eficientes para o sistema operacional Linux, baseado em Unix, e têm tido adoção crescente como consequência das funcionalidades que oferecem para realização de gestão de arquivos.

1.2 Objetivos

Este trabalho tem como objetivo avaliar o desempenho dos sistemas de arquivos Ext4, XFS e ReiserFS em comparação uns aos outros. Esta avaliação pretende explicitar vantagens e desvantagens destes sistemas de arquivos e esclarecer em que situações cada um possui melhor desempenho.

1.3 Organização do trabalho

O Capítulo 2 descreve os sistemas de arquivos que serão avaliados e apresenta as características necessárias para o correto entendimento da execução dos experimentos desse trabalho. O capítulo 3, as ferramentas utilizadas para execução dos experimentos e o ambiente no qual foram executados são apresentados. No Capítulo 4 descreve-se em detalhes o planejamento e a execução dos experimentos. São definidos os fatores e níveis que serão avaliados, a lista de todos os experimentos e o planejamento adotado. O capítulo 5 traz os resultados obtidos através da coleta realizada nos experimentos e possui gráficos comparativos dos experimentos bem como análise da influência dos fatores. Por fim, o Capítulo 6 traz as conclusões sobre cada uma das etapas do experimento e considerações finais.

2 *Fundamentação*

2.1 Considerações iniciais

Este trabalho envolve técnicas de Avaliação de Desempenho e os Sistemas de Arquivos XFS, EXT4 e ReiserFS. Este capítulo descreve os conceitos e teorias dos tópicos citados.

2.2 XFS

Nessa seção será primeiramente apresentada a história do desenvolvimento do XFS, e após a arquitetura do mesmo.

2.2.1 História

O sistema de arquivos XFS foi desenvolvido para o sistema operacional IRIX, da Silicon Graphics em 1994. Ele foi planejado para resolver os problemas do sistema de arquivos que era utilizado, o EFS (*Extents File System*). Os problemas apresentados na época eram decorrentes de novas aplicações e novos *hardwares* que estavam surgindo. Aplicações para edição de vídeo descomprimidos, por exemplo, precisavam alocar centenas de *gigabytes* de espaço, e o EFS suportava apenas 8 GB de arquivos, sendo que cada arquivo podia ter no máximo 2 GB. Além disso, o acesso ao disco rígido precisava ser otimizado, pois havia muito *overhead* [3].

O XFS foi distribuído na licença GNU *General Public License* em 2000, e o primeiro suporte à esse sistema de arquivos foi em 2001, no kernel 2.4 do Linux. Hoje ele está disponível para quase todas as distribuições Linux.

2.2.2 Arquitetura

Um sistema de arquivos XFS é dividido em vários Allocation Groups (AGs) de mesmo tamanho. Um AG pode ser pensado como sendo um sistema de arquivos autônomo, podendo ter até um *terabyte* de tamanho, e tem as seguintes características:

- (i) Contém um superbloco descrevendo as suas informações;
- (ii) Capacidade de gerenciar o seu espaço livre;
- (iii) Aloca e busca por *inodes*.

A ideia do desenvolvimento do XFS com AGs, é que conforme o acesso concorrente aumenta no sistema de arquivos, o XFS consegue paralelizar as várias operações sem degradar o desempenho.

Cada AG é dividido em 4 partes como mostra a figura 1. A primeira parte é o superbloco, responsável por guardar todas as informações do AG, como tamanho dos blocos, dos setores e dos *inodes*, versão do XFS, contadores para número de *inodes* alocados e livres, etc. Esse superbloco é replicado dentro do AG, então caso haja uma falha, ele pode ser recuperado.

A segunda parte do AG é o gerenciamento de espaço livre, que é feito através do uso de duas árvores B+. Por motivos de otimização, uma delas é ordenada por número do bloco, e a outra por quantidade de espaço contíguo.

A terceira parte é responsável pelo gerenciamento de *inodes* no sistema de arquivos, sendo responsável por alocar e buscar *inodes*. Os *inodes* são alocados em grupos de 64, e são buscados através de uma árvore B+.

A quarta parte é responsável por reservar espaço para possível crescimento das árvores B+ descritas previamente. Esse espaço não pode ser alocado por nenhum outro tipo de dado.

2.3 Ext4

Nesta seção será apresentada a história do sistema de arquivos Ext até a chegada em sua última versão, a Ext4. Em seguida essa última versão será estudada em mais detalhes.

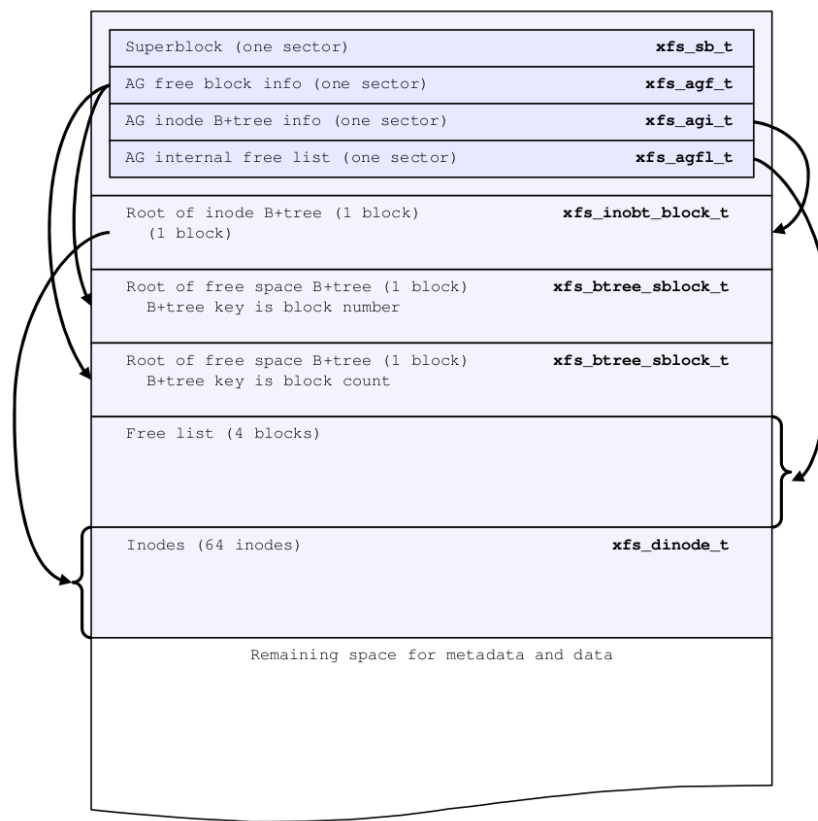


Figura 1: Detalhes de um *Allocation Group* do XFS [1].

2.3.1 História

O sistema de arquivos Ext — *Extended File System* — nasceu através de pesquisas realizadas por Rémy Card et al. em 1992 [4], e foi integrada ao Linux em sua versão 0.96c. Esse novo sistema de arquivos resolveu duas grandes limitações impostas pelo sistema de arquivos Minix [5]:

- (i) O limite de 64 MB¹ para tamanho de arquivo: Ext passou a permitir arquivos de até 2 GB²; e
- (ii) O limite de 14 caracteres para o nome do arquivo: Ext passou a permitir nomes de arquivos de até 255 caracteres.

Entretanto, alguns problemas ainda estavam presentes nessa implementação. Por exemplo, não havia acesso separado, modificação de *inodes* e marcação de datas de modificação³. Este sistema de arquivos utilizava listas encadeadas para manter controle dos

¹Mega Bytes: 1024 Bytes.

²Giga Bytes: 1024 MB.

³O termo utilizado em inglês é *timestamp*.

blocos livres, e isso degradava a performance [4]: à medida que o sistema operacional era utilizado, a lista se tornava não-ordenada, e o sistema de arquivos se tornava fragmentado.

Como resposta a estes problemas, dois novos sistemas de arquivos foram desenvolvidos em versão alfa, em Janeiro de 1993: o sistema de arquivos Xia e o Ext2 — *Second Extended File System*. Enquanto o Xia representava algumas melhorias para o sistema Minix, o Ext2 era baseado no código do Ext e trazia melhorias como [4]:

- (i) Suporte a padrões de arquivos Unix padrão;
- (ii) Capacidade de gerir sistemas de arquivos criados em partições de até 4 TeraBytes, eliminando a necessidade de várias partições para utilização de discos grandes;
- (iii) Capacidade de nomes de arquivos de até 1012 caracteres; e
- (iv) Reserva de blocos para super usuário (*root*), em geral 5%. Isso permite com que administradores de sistema recuperem o sistema em situações onde o disco fica cheio.

Em novembro de 2001 foi introduzida uma nova versão do sistema de arquivos Ext, a Ext3. Com muitas novidades, essa versão foi rapidamente integrada às versões mais populares do Linux [2]. Entre as inovações apresentadas por essa versão está a introdução de *journaling* como ferramenta para aumento da confiabilidade de sistema, em casos onde há interrupção repentina do funcionamento do sistema operacional. A função do *journaling* é salvar em um *log* as operações a serem executadas. Esse *log* é utilizado na inicialização do sistema para restaurá-lo a um estado válido. Com o passar do tempo, o sistema de arquivos Ext3 foi tornando-se limitado para as necessidades de utilização, tanto pessoal quanto na indústria, especialmente em relação ao seu limite de tamanho de arquivos. Assim uma atualização desse sistema de arquivos foi feita em 2006 [2]. Em Dezembro de 2008 surgiu o Ext4. Esse sistema será detalhado na sub-seção 2.3.2.

Por fim, vale observar que as versões mais recentes do Ext sempre suportam as versões mais antigas, o que justifica a facilidade de adoção pelo mercado: não há esforço em migração de dados para novos formatos, pois o sistema pode operar ao mesmo tempo com partições de versões anteriores.

2.3.2 Ext4

O sistema de arquivos Ext4 vem sendo utilizado no Linux desde a sua versão 2.6.19. Seus grandes objetivos são de resolver problemas de escalabilidade e de gargalo do seu

antecessor — o Ext3, detalhado na sub-seção 2.3.1. São avanços notados nesse sistema de arquivos [2]:

1. Utilização de 48 bits para mapeamento de blocos, frente aos 32 bits utilizados pelo Ext3. Isso permite com que sistemas de arquivos de até 256 TB sejam mapeados;
2. Utilização de *extents* para gravação de arquivos. Um *extent* é uma posição contígua no disco, representada por início e fim em um descritor contido em um *inode*, num limite de quatro *extents* por *inode*. Assim, arquivos fragmentados exigem mais *extents* que um arquivo pequeno ou contíguo em disco. Um *extent* pode ter até 2^{15} blocos, ou 128 MB quando os blocos são de 4 KB. Sua utilização otimiza tanto leitura quanto escrita em disco para arquivos grandes. A solução implementada no Ext3 — mapeamento indireto de bloco — era boa para arquivos pequenos e esparsos, mas ineficiente para casos onde arquivos grandes eram utilizados. Esse mapeamento de *extents* é observado na Figura 2;
3. Utilização de bits reservados de *inode* para mapear blocos de arquivos, elevando a limitação de tamanhos de arquivos de 2 TB (Ext3) para 16 TB (Ext4). Há um planejamento para que o limite seja elevado nas melhorias futuras desse sistema de arquivos;
4. Eliminação de restrição de número de subdiretórios contidos em um único diretório, e indexação de diretórios usando uma árvore H^4 de profundidade fixa. O sistema de arquivos Ext3 possuía um limite de 32000 arquivos;
5. Pré-alocação persistente, onde os blocos são reservados fisicamente no disco para arquivos que ainda não os ocuparam. Essa característica é útil para minimizar fragmentação de arquivos;
6. Alocação atrasada de blocos múltiplos, onde as operações de escrita são mantidas em *cache* até que ocorra um *flush* de página, e ele seja de fato aplicado em disco. Essa característica melhora utilização de disco e minimiza fragmentação, fazendo melhor uso dos *extents*;
7. Melhorias na forma como a integridade do sistema de arquivos é verificada (usando a ferramenta *e2fsck*). O ganho de performance é visível quando comparado à versão Ext3; e
8. Utilização de *checksum* CRC32 nos metadados utilizados para recuperação do sistema de arquivos — o *journal*. Assim o sistema não confia cegamente nos dados de reparação, e pode evitar danos de recuperação indevida.

⁴Uma árvore H é uma implementação de árvores B usando *hashes* de 32 bits.

As características acima descritas permitem obter uma melhor visão do sistema de arquivos Ext4, utilizado na avaliação de desempenho conduzida como foco desse trabalho.

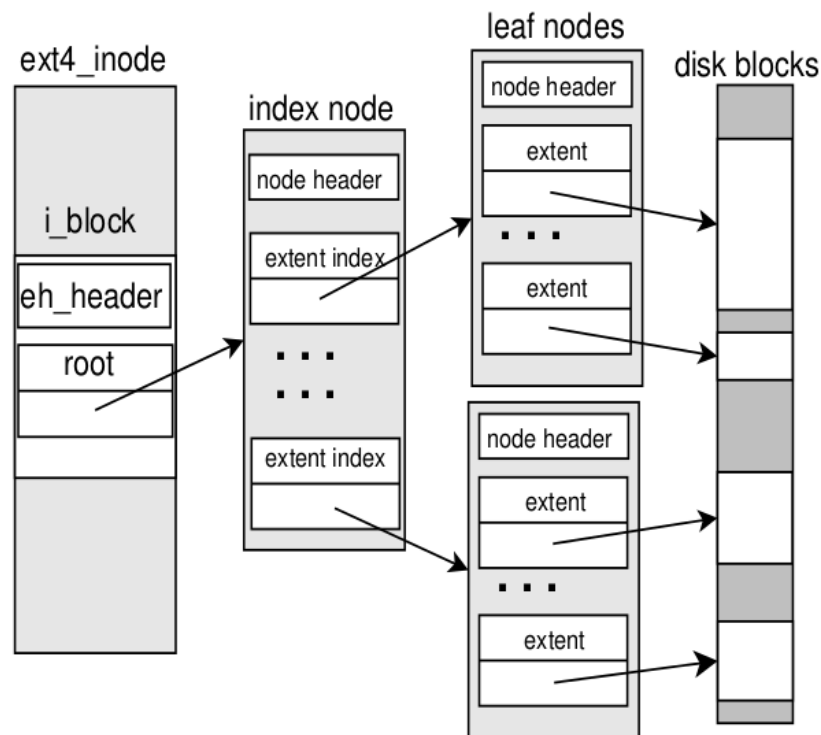


Figura 2: Uso de *extents* no sistema de arquivos Ext4 [2].

2.4 ReiserFS

O ReiserFS foi criado em 1996 por Hans Thomas Reiser e se tornou muito famoso, entre outras coisas, por ser o primeiro sistema de arquivos a utilizar a técnica de *journaling* para melhorar a confiabilidade do sistema. No caso do ReiserFS, o *journaling* embora sejam guardadas informações sobre todo o sistema de arquivos, ele tem a intenção de garantir a integridade dos metadados dos arquivos, ou seja, ele não é capaz de reconstruir todo o sistema de arquivos a partir do *journal*.

Passou a ser suportado oficialmente pelo kernel do linux a partir de sua versão 2.4 e chegou a ser adotado como sistema de arquivos padrão da distribuição comercial da Novell, o SUSE Linux Enterprise. Sua adoção foi baseada no fato do Reiser ser o único sistema de arquivos da época que possuía *journaling* e também em acordos de suporte com a empresa de seu criador a Namesys.

Este sistema utiliza árvores B* para melhorar o desempenho em operações de busca. De acordo com a documentação encontrada sobre o sistema[6] ele pode ser utilizado com quatro opções de tamanho de bloco: 4096 (padrão), 512, 1024 e 8192 KBytes. No entanto, durante testes preliminares com o sistema Kubuntu 10.10 não foi possível criar um sistema de arquivos Reiser com tamanho de bloco diferente de 4096 KBytes. Em função disso, nas etapas 2 e 3 da avaliação o fator B - Tamanho do bloco não foi utilizado.

Á estrutura de árvore utilizada possui dois tipos de nós: Nós internos e nós folhas e cada nó é um bloco do disco. O primeiro tipo é utilizado para manter a estrutura de árvore, apontando para outros blocos. Os nós folha são os itens propriamente ditos. No ReiserFS os itens podem representar arquivos, diretórios ou *stat item*. Os arquivos podem ser de dois tipos, itens diretos ou indiretos, dependendo do tamanho do arquivo.

Cada diretório ou arquivo é sempre precedido por um *stat item* que contém metadados sobre o arquivo ou diretório que o sucede.

A Figura 3 mostra um exemplo da estrutura de árvore utilizada pelo ReiserFS para organizar os arquivos no disco.

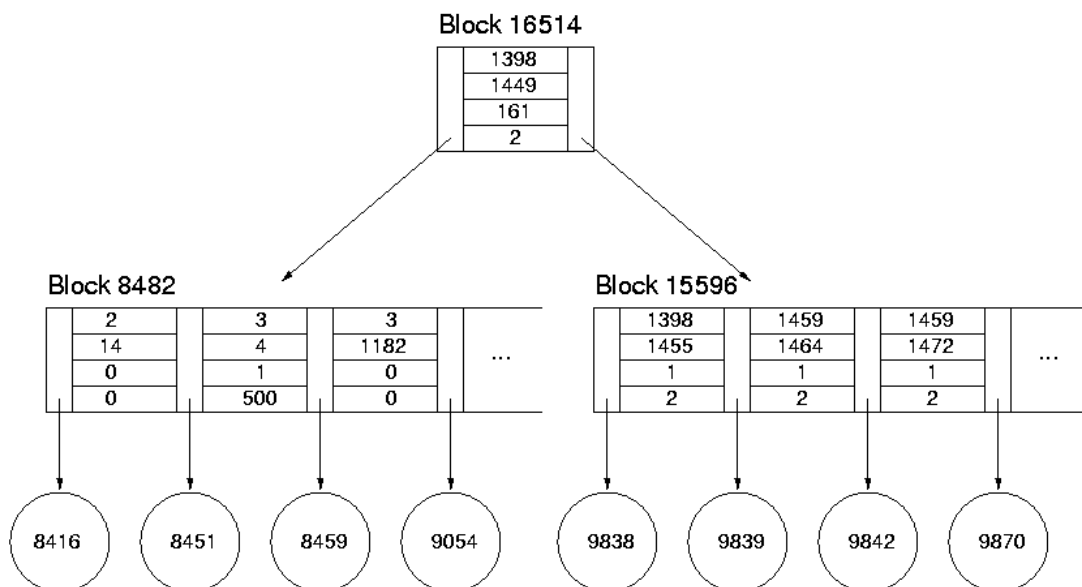


Figura 3: Modelo de árvore utilizado pelo ReiserFS com blocos internos representados pelos blocos retangulares e blocos folha representados por esferas.

3 *Métodos e Ferramentas*

A seguir encontra-se uma descrição detalhada da ferramenta utilizada e o ambiente no qual os experimentos foram executados.

3.1 IOzone

A ferramenta IOzone é uma ferramenta de avaliação de desempenho de sistemas de arquivo [7]. Esta ferramenta foi disponibilizada para diversas versões de sistemas operacionais, incluindo Kubuntu Linux, a distribuição Linux utilizada para realização da avaliação de desempenho desse trabalho. Quando a ferramenta é utilizada, ela simula a utilização dos discos de uma máquina, ao passo em que faz medições de tempo, utilização e taxa de entrada/saída para os algoritmos utilizados. Assim, são operações utilizáveis e mensuráveis:

- Read;
- Write;
- Re-read;
- Re-write;
- Read backwards;
- Read strided;
- Fread;
- Fwrite;
- Random read;
- Pread;
- Mmap;
- Aio read; e
- Aio write.

No contexto desta avaliação de desempenho foram utilizadas somente medidas para *Read*, *Random read*, *Write* e *Random write*.

3.2 Ambiente

Os experimentos foram executados computador desktop com as seguintes configurações:

- Processador: AMD Athlon 64bits X2 Dual Core Processor 4000+
- Memória: 1GB/2GB (Dois pentes de 1GB) DDR2 800Mhz
- HD: Samsung SP2004C 200GB 7200rpm 8MB de cache
- Sistema operacional: Kubuntu Linux 10.10 32bits, Kernel 2.6.35-28-generic

Para execução das repetições de cada experimento foi criado um *script* bash que roda automaticamente todas as variações de fatores para a configuração de memória RAM da máquina. Então é necessário alterar a configuração de memória manualmente e depois executar o *script* novamente. O *script* pode ser encontrado no Anexo A.

4 *Planejamento e execução*

Neste capítulo são especificados o modelo de avaliação que foi adotado, quais as variáveis de resposta medidas, fatores e níveis considerados e como os experimentos foram divididos em etapas.

4.1 Variáveis de resposta, fatores e níveis

Para a escolha das variáveis de resposta foram escolhidas as operações mais comuns utilizados pelo sistemas operacionais que são escrita e leitura. Portanto, as variáveis de resposta escolhida foram:

- Leitura
- Escrita
- Leitura aleatória
- Escrita aleatória

Para a avaliação foi adotado o planejamento fatorial completo 2^4 na etapa 1 e fatorial completo 2^3 para as etapas 2 e 3 para manter a complexidade dos experimentos em um nível aceitável foram utilizados apenas 4 fatores que acredita-se que poderiam ter maior influência sobre os resultados dos experimentos e cada um deles com 2 níveis de variação.

Para simplificar a execução dos experimentos e ainda assim comparar todos os sistemas uns com os outros, a avaliação foi dividida em 3 etapas. Na etapa 1 foram comparados os sistemas de arquivos Ext4 e o XFS. A Tabela 1 mostra os fatores e níveis escolhidos para a etapa 1.

Na etapa 2 foram comparados os sistemas de arquivos Ext4 e ReiserFS e a Tabela 2 mostra os fatores e níveis adotados para a etapa 2. Como foi dito anteriormente, o

Fator	Nível 1	Nível -1
A - Sistema de arquivo	Ext4	XFS
B - Tamanho do bloco	4KB	1KB
C - Memória RAM	1GB	2GB
D - Tamanho do arquivo	64KB	64MB

Tabela 1: Fatores e níveis escolhidos para a Etapa 1.

ReiserFS não permite a mudança do tamanho do bloco, este fator não foi considerado para as etapas 2 e 3 e o bloco ficou fixado em 4096 bytes para ambos os sistemas comparados nessas etapas.

Fator	Nível 1	Nível -1
A - Sistema de arquivo	Ext4	ReiserFS
C - Memória RAM	1GB	2GB
D - Tamanho do arquivo	64KB	64MB

Tabela 2: Fatores e níveis escolhidos para a Etapa 2.

Na etapa 3 foram comparados os sistemas XFS e ReiserFS. A Tabela 3 mostra os fatores e níveis adotados para esta etapa.

Fator	Nível 1	Nível -1
A - Sistema de arquivo	XFS	ReiserFS
C - Memória RAM	1GB	2GB
D - Tamanho do arquivo	64KB	64MB

Tabela 3: Fatores e níveis escolhidos para a Etapa 3.

4.2 Experimentos

Na etapa 1 da avaliação foram realizados 16 experimentos para cobrir todas as variações possíveis entre os 2 níveis de cada um dos 4 fatores definidos anteriormente. Para as etapas 2 e 3 foram realizados 8 experimentos, novamente para cobrir todas as variações entre os 2 níveis e 3 fatores definidos anteriormente. Para cada experimento foram realizadas 10 repetições resultando em um total de 160 execuções na etapa 1, 80 execuções na etapa 2 e 80 execuções na etapa 3 em função de possuírem 1 fator a menos do que a etapa 1.

A tabela 4 mostra os níveis utilizados em cada um dos 16 experimentos realizados na etapa 1 e a Tabela 5 mostra os níveis utilizados em cada um dos 8 experimentos

tanto da etapa 2 quanto da etapa 3. A definição dos fatores A, B, C e D podem ser encontradas nas Tabelas 1, 2 e 3

Experimento	A	B	C	D
1	1	1	1	1
2	1	1	1	-1
3	1	1	-1	1
4	1	1	-1	-1
5	1	-1	1	1
6	1	-1	1	-1
7	1	-1	-1	1
8	1	-1	-1	-1
9	-1	1	1	1
10	-1	1	1	-1
11	-1	1	-1	1
12	-1	1	-1	-1
13	-1	-1	1	1
14	-1	-1	1	-1
15	-1	-1	-1	1
16	-1	-1	-1	-1

Tabela 4: Níveis utilizados em cada um dos experimentos para a Etapa 1

Experimento	A	C	D
1	1	1	1
2	1	1	-1
3	1	-1	1
4	1	-1	-1
5	-1	1	1
6	-1	1	-1
7	-1	-1	1
8	-1	-1	-1

Tabela 5: Níveis utilizados em cada um dos experimentos para as Etapa 2 e 3

XXX EXPLICAR COMO FOI A EXECUÇÃO XXX

5 *Resultados Obtidos*

Este capítulo apresenta os resultados obtidos nos experimentos nas Etapas 1, 2 e 3. Os gráficos de barra utilizados foram agrupados de modo que cada par de experimentos que possuam de variação entre eles apenas o fator Sistema de arquivo sejam exibidos juntos para facilitar a comparação de um sistema de arquivo em relação ao outro.

5.1 **Etapa 1: comparação entre Ext4 e XFS**

Na primeira etapa do trabalho os sistemas de arquivos Ext4 e XFS foram analisados em conjunto, a partir dos dados de experimentos de análise de desempenho realizados separadamente, conforme descrito no Capítulo 4.

Como pode ser observado na Figura 4, o sistema de arquivos XFS teve melhor desempenho para todos os experimentos envolvendo tamanho de blocos de 4 KB. Entretanto, quando o bloco passa a ser de 1 KB, nada pode se afirmar sobre qual dos dois sistemas de arquivos tem melhor desempenho, pois seus intervalos de confiança se sobrepõem. Nota-se também que o melhor desempenho de saída (MB/s) foi obtido utilizando XFS com blocos de 4 KB e arquivos de tamanho de 64 MB, onde nada pode se afirmar sobre a memória, pois para ambos os valores de 1 GB e 2 GB as taxas de saída com seus respectivos intervalos de confiança se sobrepõem.

Ao comparar XFS e Ext4 para operação de leitura — exemplo da Figura 5 — podemos notar que em dois experimentos o Ext4 possui melhor desempenho que XFS: na leitura de blocos de 1 KB, com memória de 2 GB e tamanho de arquivos de 64 KB; e em uma situação similar onde os tamanhos de arquivos são de 64 MB. Apesar da diferença ser significativa para o primeiro caso, para o segundo não é. Nos demais casos nada se pode afirmar sobre um melhor desempenho, pois os resultados obtidos possuem intervalos de confiança sobrepostos quando comparados.

Para a operação de leitura randômica, XFS e Ext4 obtiveram desempenhos similares, com diferenças tênues para grande parte dos casos, como visto na Figura 6. Um

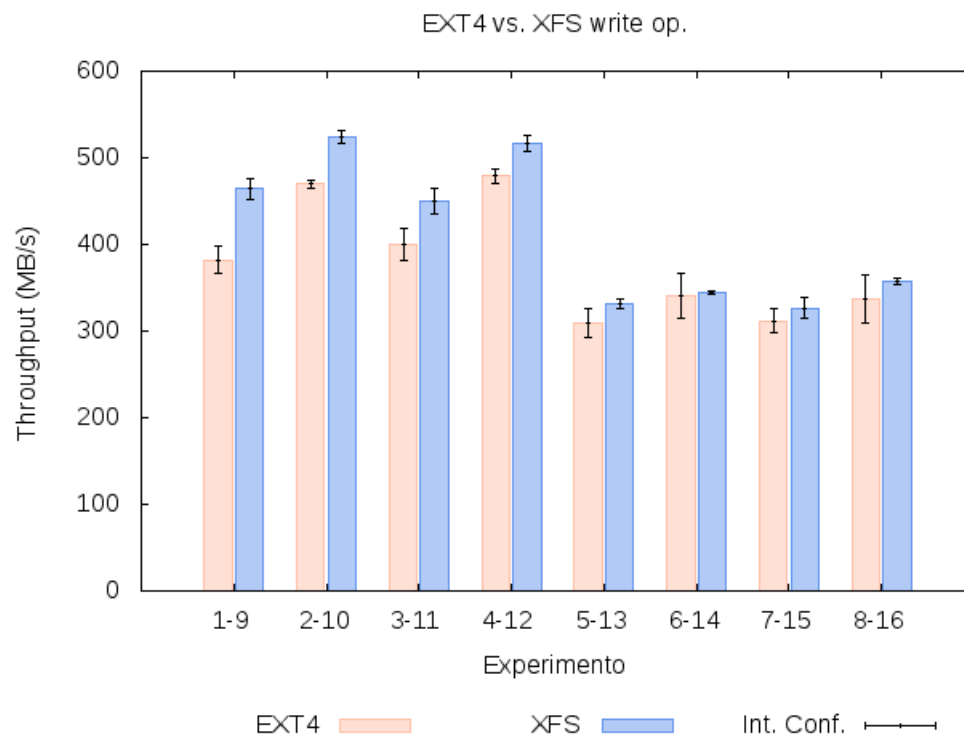


Figura 4: Gráfico comparando os experimentos da Etapa 1 na operação de Escrita.

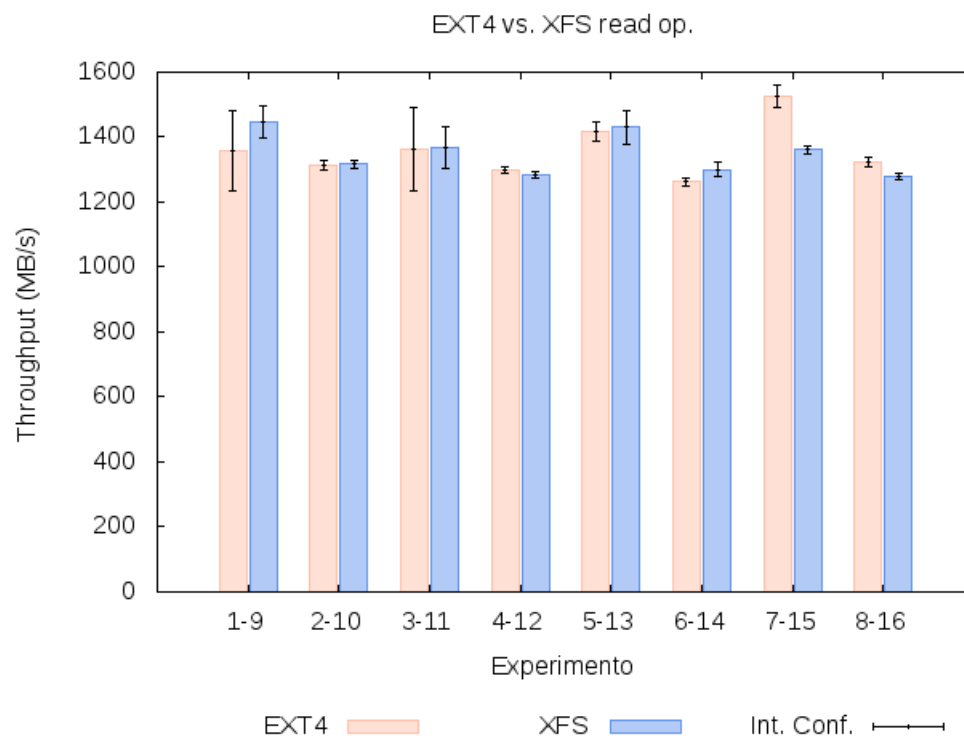


Figura 5: Gráfico comparando os experimentos da Etapa 1 na operação de Leitura.

dos resultados com grande diferença — mas também grandes intervalos de confiança — foi observado quando blocos de 4 KB foram utilizados com 1 GB de memória com arquivos de tamanho de 64 KB. Nesse experimento, o Ext4 teve um desempenho *médio* inferior ao XFS. Entretanto, os intervalos de confiança se sobrepõem nos limites, fazendo com que não seja possível afirmar com a certeza desejada qual se comporta da melhor forma sob tais circunstâncias.

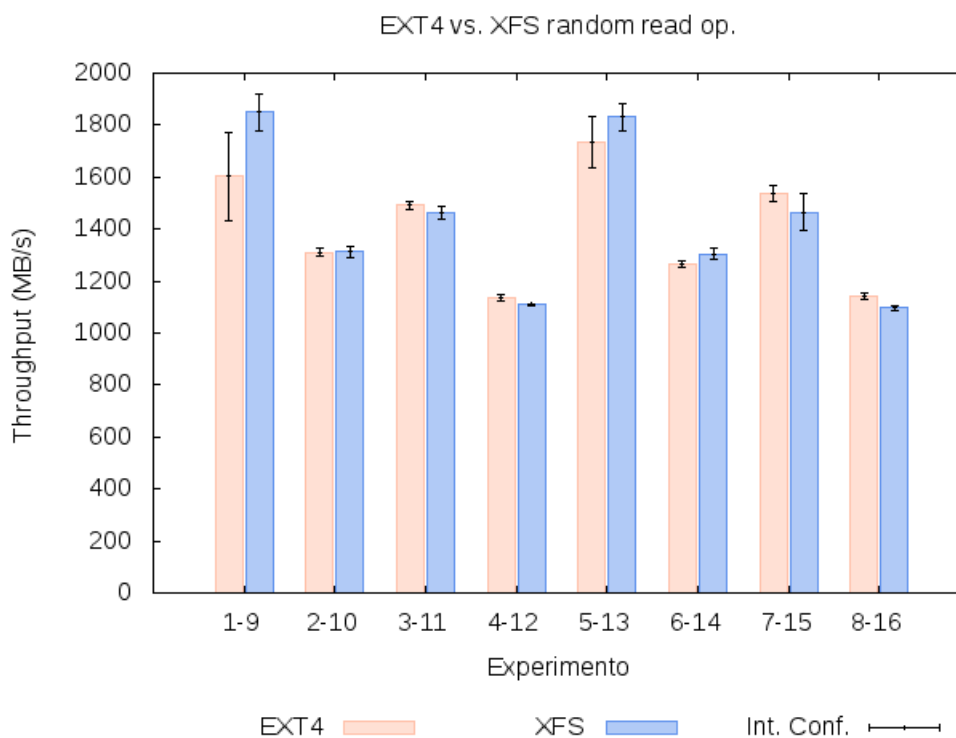


Figura 6: Gráfico comparando os experimentos da Etapa 1 na operação de Leitura aleatória.

Por fim, na comparação utilizando operações de escrita randômica (vista na Figura 7) o XFS teve um desempenho visivelmente superior que o Ext4 quando os blocos eram de 4 KB, a memória de 2 GB e arquivos de 64 KB. Uma diferença um pouco menor, com *throughput* similar, pode ser notada quando blocos de 1 KB, memória de 2 GB e arquivos de 64 MB são utilizados. Os maiores desempenhos de *throughput* são observados quando a seguinte configuração é adotada: blocos de 4 KB, memória de 1 GB e tamanho de arquivos de 64 KB. Ainda, uma variação dessa configuração com blocos de 1 KB obtém performance mais previsível, ou seja, com menor intervalo de confiança. De forma geral nesse teste os sistemas de arquivos se comportam de maneira similar, à parte exceções detalhadas previamente.

Por fim, ao utilizar os dados das medições, foi possível calcular as influências dos fatores, apresentados na Tabela 6.

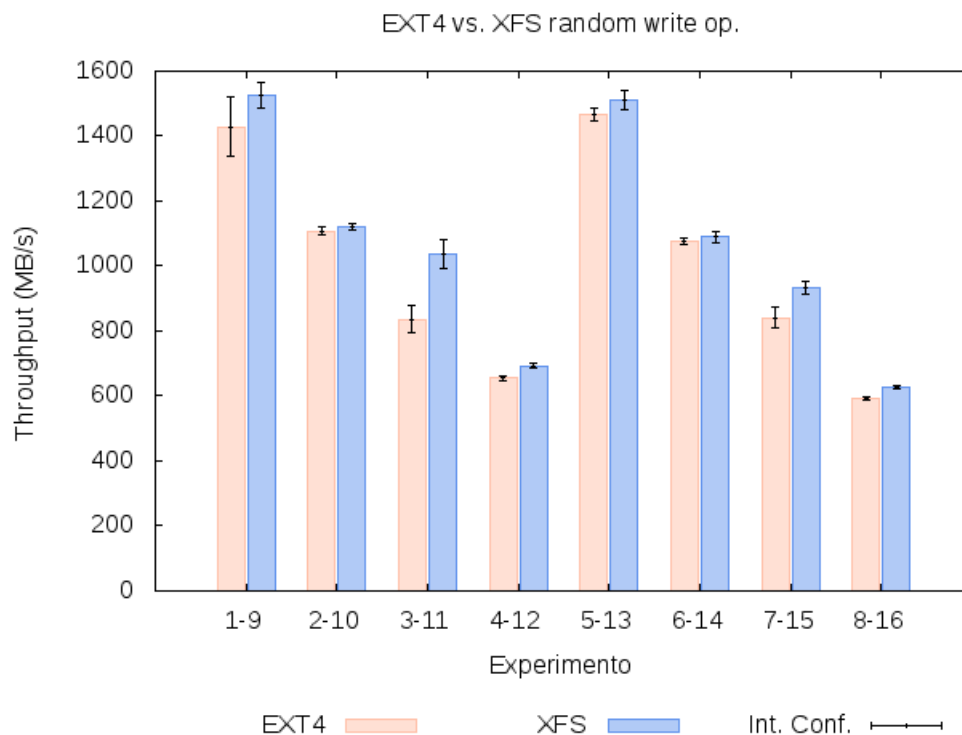


Figura 7: Gráfico comparando os experimentos da Etapa 1 na operação de Escrita aleatória.

Nos gráficos gerados a partir dessa tabela (vide Figura 5.1), é possível notar que o tamanho de blocos é um fator que possui grande influência quando comparamos o desempenho de Ext4 e XFS para a operação de escrita. Esse fator faz com que o desempenho ambos os sistemas de arquivos seja visivelmente degradado quando blocos pequenos são utilizados. Também podemos notar que o tamanho de arquivos é um fator de grande influência quando comparamos Ext4 e XFS para as operações de leitura, leitura randômica e escrita randômica. Arquivos maiores (64 MB) degradaram o desempenho de ambos os sistemas de arquivos. Por fim, a quantidade de memória disponível é também um fator de grande importância para operações de leitura randômica e escrita randômica. Nesse caso, com o aumento da memória disponível há uma degradação no desempenho de ambos os sistemas de arquivos, onde essa degradação é mais visível para o Ext4.

Fator(es)	Write	Read	Random Read	Random Write
iA	6.00	0.44	28.97	1.15
iB	77.29	1.82	0.02	0.29
iC	0.01	0.11	26.29	68.52
iD	11.27	64.16	35.92	28.06
iAB	1.94	4.86	0.50	0.11
iAC	0.11	10.86	7.35	0.16
iAD	0.19	0.13	0.00	0.45
iBC	0.00	3.03	0.09	0.14
iBD	2.74	5.25	0.15	0.05
iCD	0.01	0.06	0.14	0.88
iABC	0.25	2.04	0.03	0.01
iABD	0.06	5.05	0.23	0.11
iACD	0.12	2.07	0.11	0.05
iBCD	0.01	0.04	0.15	0.01
iABCD	0.00	0.09	0.03	0.01

Tabela 6: Tabela com os valores calculados de influência dos fatores para a Etapa 1 da avaliação de desempenho separados para o tipo de operação.

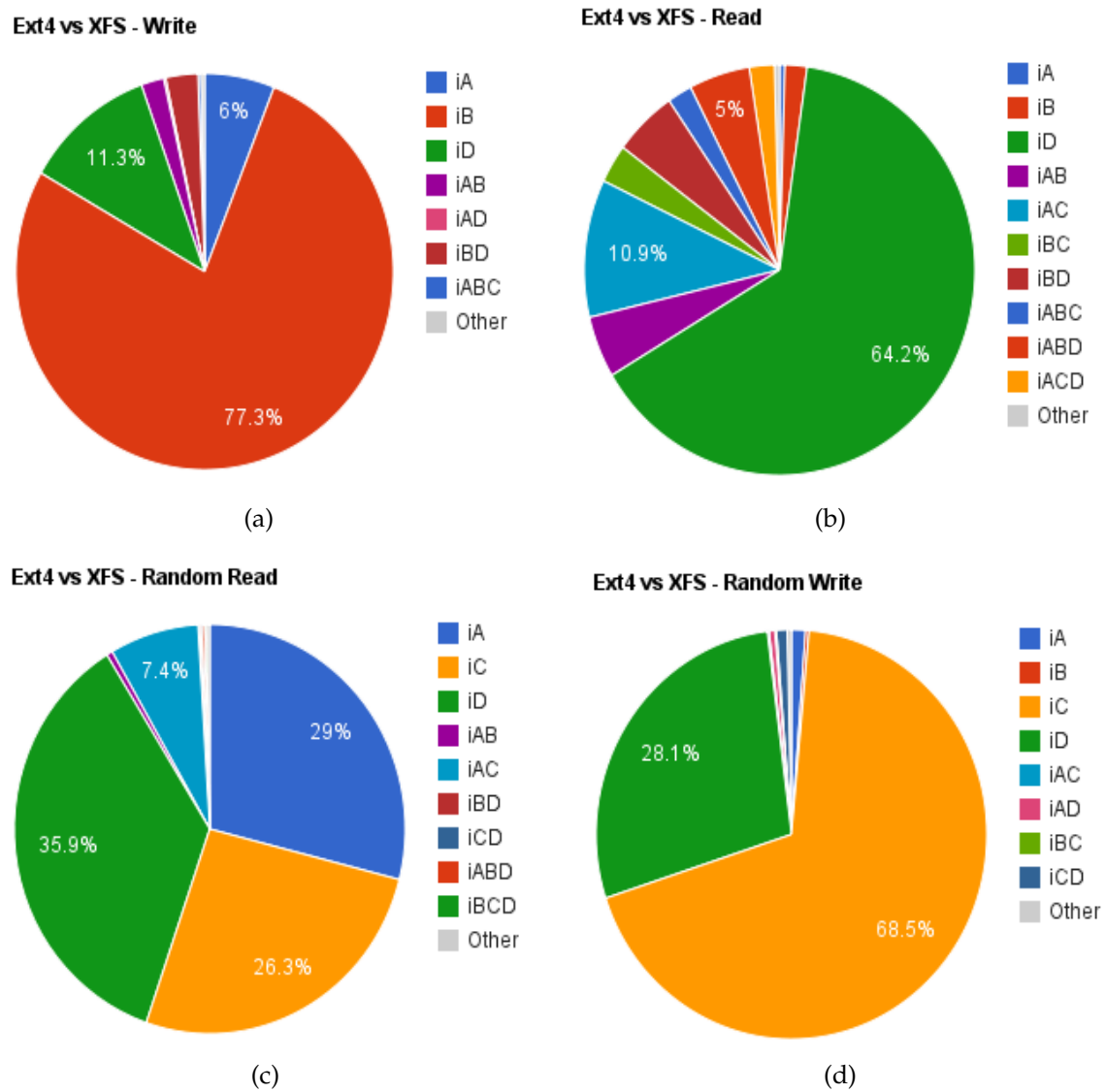


Figura 8: Gráficos com as influências dos fatores calculadas para as operações avaliadas na Etapa 1: (a) - Escrita, (b) - Leitura, (c) - Leitura aleatória e (d) - Escrita aleatória

5.2 Etapa 2: comparação entre EXT4 e ReiserFS

Na segunda etapa do trabalho os sistemas de arquivos Ext4 e ReiserFS foram analisados em conjunto, a partir dos dados de experimentos de análise de desempenho realizados separadamente, conforme descrito no Capítulo 4. Vale ressaltar que o ReiserFS não disponibiliza opção de utilização de blocos de tamanho de 1 KB. Portanto, o Fator B, conforme apresentado na Tabela 2, não foi utilizado.

Como pode ser observado na Figura 9, o sistema de arquivos Ext4 possui melhor desempenho que o ReiserFS para todos os experimentos realizados com a operação de escrita.

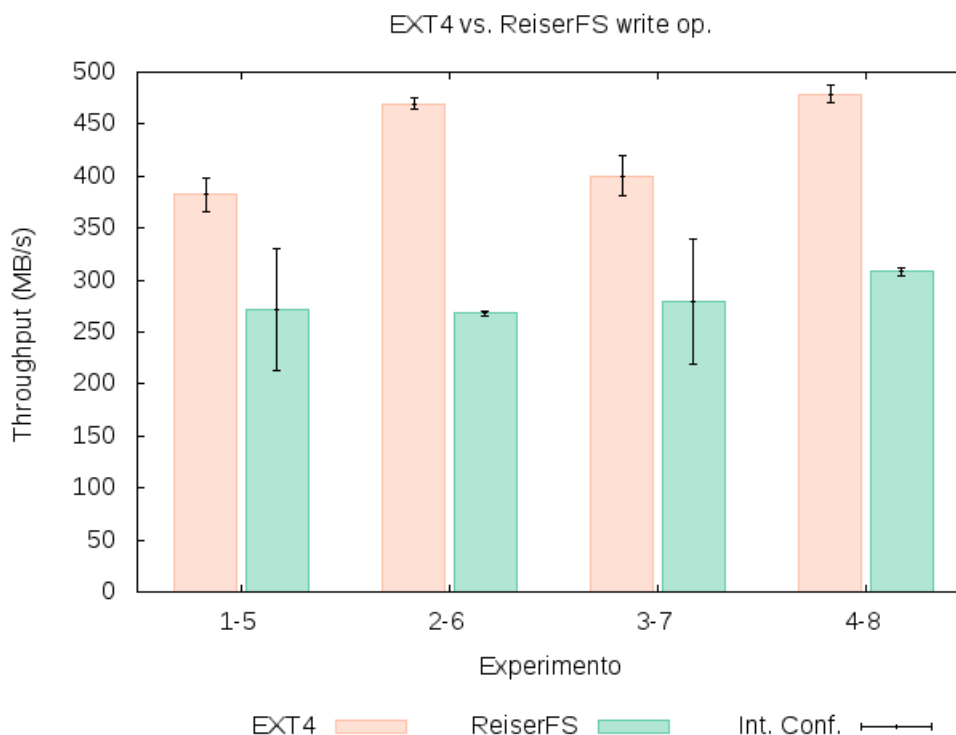


Figura 9: Gráfico comparando os experimentos da Etapa 2 na operação de Escrita.

Na Figura 10 — que representa experimentos realizados para a operação de leitura — o ReiserFS apresentou em geral um melhor desempenho que o Ext4, à parte o experimento 2, onde o segundo obteve um melhor desempenho. Nesse caso, a configuração utilizada foi de 1 GB de memória e arquivos de 64 MB. Portanto, torna-se visível a degradação de desempenho do ReiserFS ao lidar com arquivos de grande tamanho.

Para a operação de leitura aleatória — representada na Figura 11 — os resultados não são conclusivos de um modo geral: o ReiserFS tem melhor desempenho para arquivos menores (64 KB) e memória de 1 GB, enquanto para uma mesma configuração

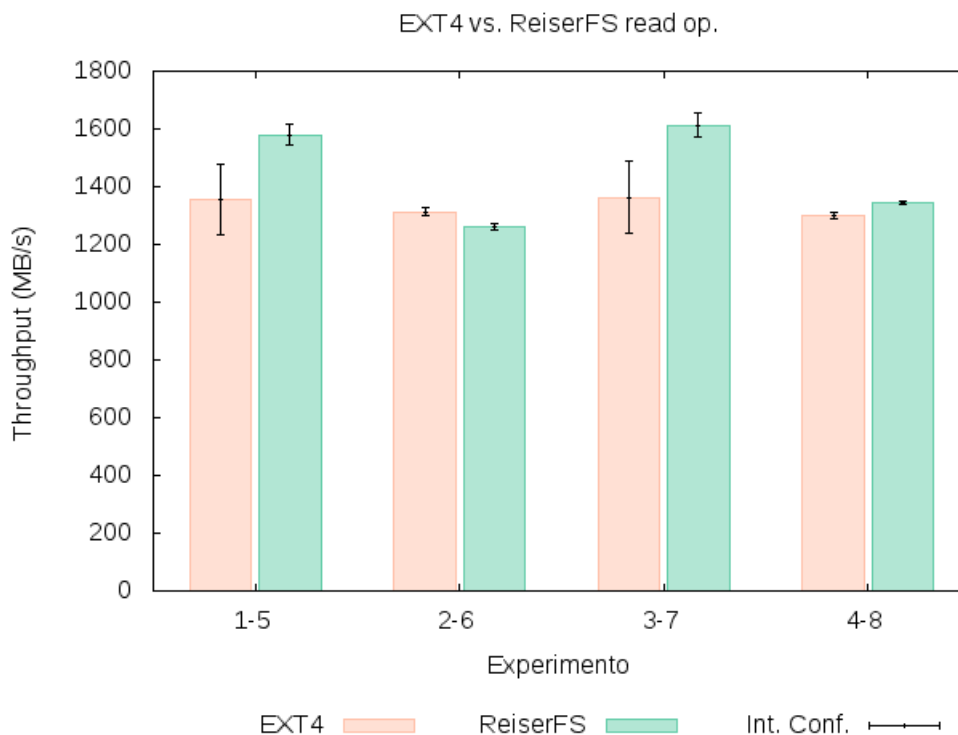


Figura 10: Gráfico comparando os experimentos da Etapa 2 na operação de Leitura.

com arquivos maiores (64 MB) o Ext4 o supera. Quando há mais memória disponível — fator importante para *caching* — não se pode concluir sobre qual é o sistema de arquivos mais eficiente, pois os intervalos de confiança se sobrepõem.

Por fim, para a operação de escrita randômica (vide Figura 12), o ReiserFS obteve um melhor desempenho quando a configuração de fatores envolvia mais quantidade de memória (2 GB). Por outro lado, quando havia menor quantidade de memória (1 GB), nada se pode afirmar sobre o desempenho de ambos, pois os intervalos de confiança se sobrepõem.

Por fim, ao utilizar os dados das medições, foi possível calcular as influências dos fatores, apresentados na Tabela 5.2.

Nos gráficos gerados a partir dessa tabela (vide Figura 5.2), é possível notar que o tamanho de arquivos é um fator que possui grande influência quando comparamos o desempenho de Ext4 e ReiserFS para as operações de leitura, leitura randômica e escrita randômica. Esse fator faz com que o desempenho do ReiserFS seja visivelmente degradado quando arquivos grandes são utilizados. Também podemos notar que a quantidade de memória disponível é um fator de grande importância quando comparamos o desempenho de Ext4 e ReiserFS para operações de leitura randômica e escrita randômica. Nesse caso, com o aumento da memória disponível há uma degradação no

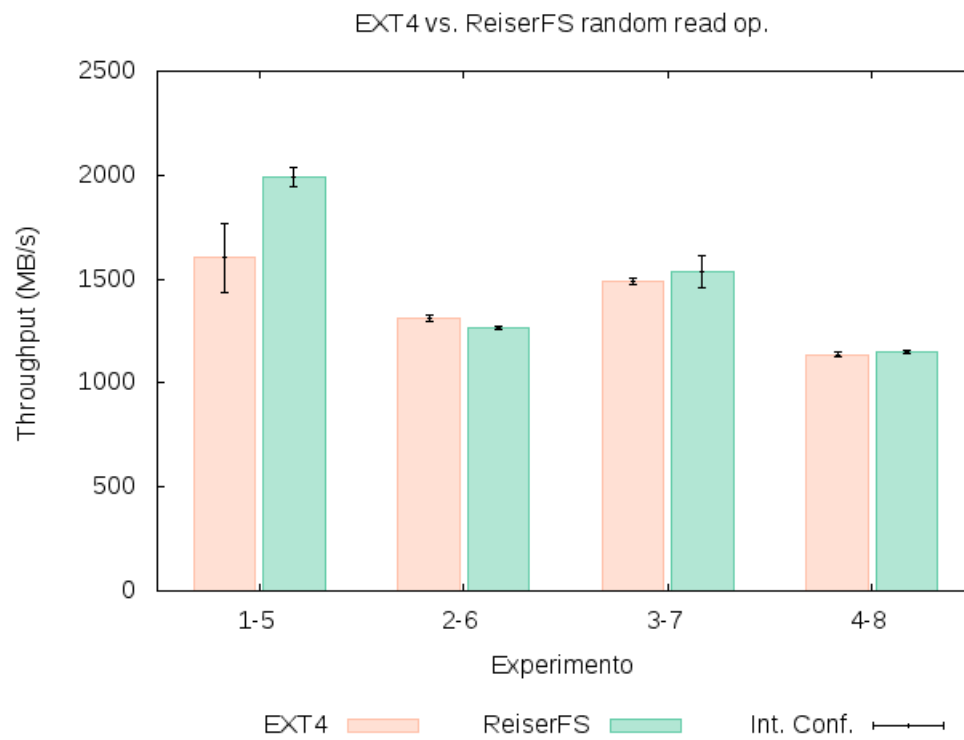


Figura 11: Gráfico comparando os experimentos da Etapa 2 na operação de Leitura aleatória.

desempenho de ambos os sistemas de arquivos, onde essa degradação é mais visível para o ReiserFS.

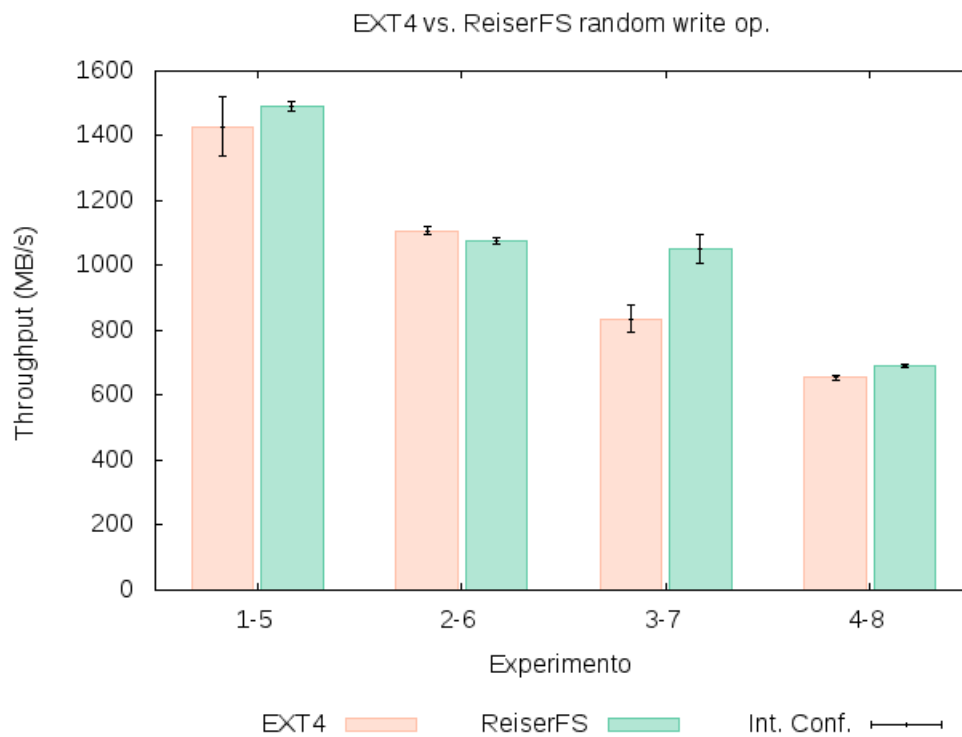


Figura 12: Gráfico comparando os experimentos da Etapa 2 na operação de Escrita aleatória.

Fator(es)	Write	Read	Random Read	Random Write
iA	84.93	22.67	3.49	0.08
iC	1.32	1.32	16.17	56.45
iD	8.55	49.85	68.35	43.08
iAC	0.09	1.60	1.79	0.14
iAD	4.57	23.98	4.81	0.01
iCD	0.13	0.09	1.70	0.25
iACD	0.41	0.49	3.69	0.00

Tabela 7: Tabela com os valores calculados de influência dos fatores para a Etapa 2 da avaliação de desempenho separados para o tipo de operação.

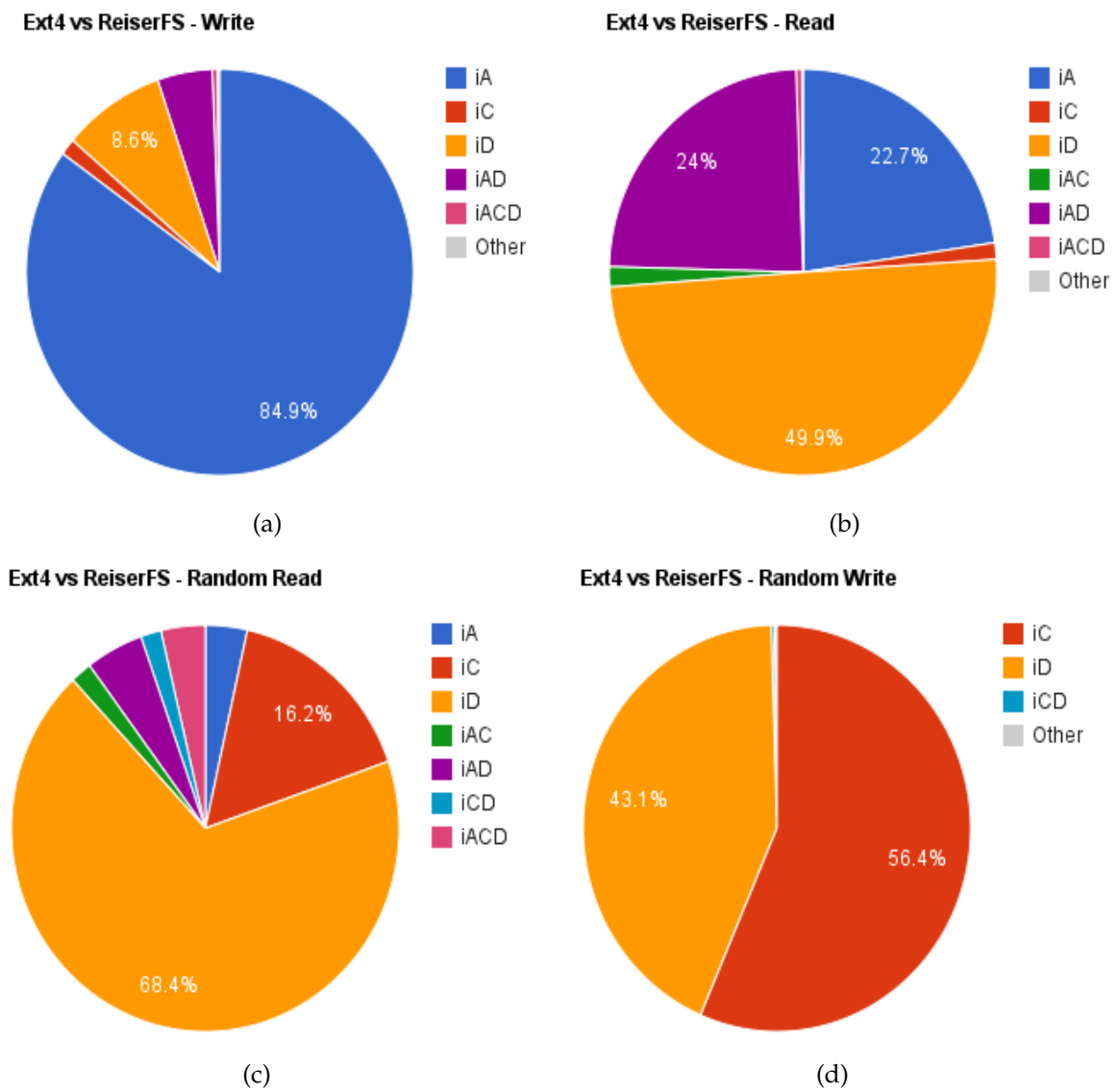


Figura 13: Gráficos com as influências dos fatores calculadas para as operações avaliadas na Etapa 2: (a) - Escrita, (b) - Leitura, (c) - Leitura aleatória e (d) - Escrita aleatória

5.3 Etapa 3: comparação entre XFS e ReiserFS

A terceira etapa do trabalho é a comparação entre os sistemas de arquivo XFS e ReiserFS. Note que nessa etapa não foi considerado a mudança de tamanho de blocos, já que o ReiserFS não possibilitava a sua utilização com 1 KB. As discussões sobre os resultados obtidos nos gráficos seguintes serão, portanto, considerados sempre com tamanho de bloco de 4 KB.

A partir do gráfico 14, podemos concluir que o XFS desempenha a operação de escrita melhor que o ReiserFS. Sendo que os melhores resultados aparecem nos experimentos 2 e 4, ou seja, onde estava sendo utilizado arquivos grandes, de 64 MB. Note que a memória RAM não influencia muito esses resultados.

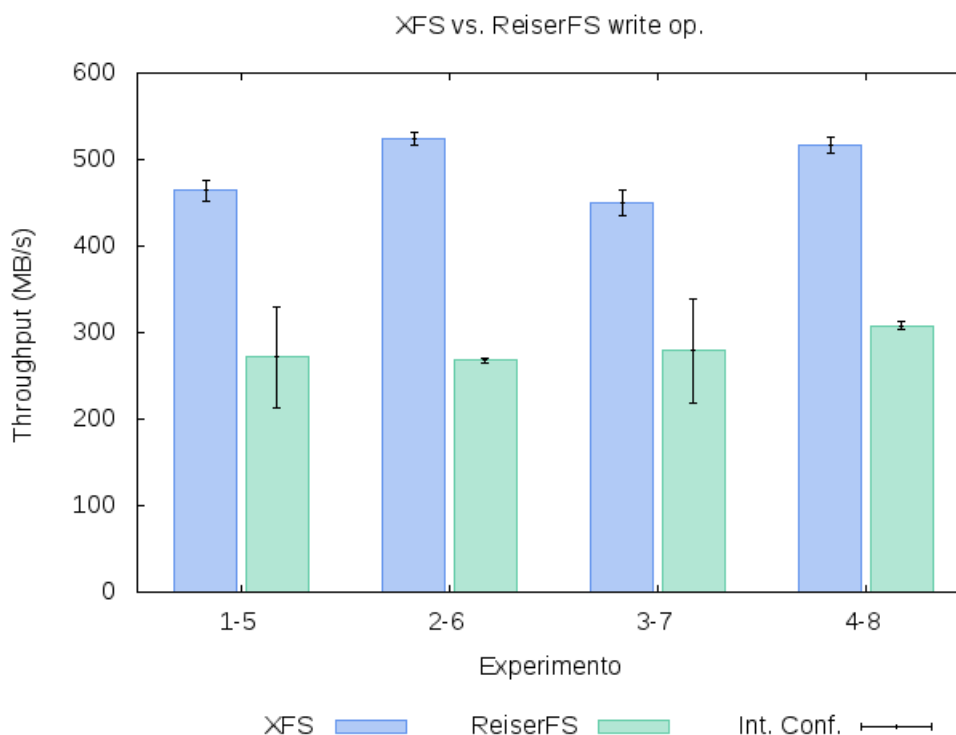


Figura 14: Gráfico comparando os experimentos da Etapa 3 na operação de Escrita.

Na operação de leitura, mostrada no gráfico 15, há um ganho em quase todas as comparações do ReiserFS. Nota-se, principalmente, um ganho nos experimentos 5 e 7, onde foi considerado o teste com tamanho de arquivos pequenos e memória RAM variável. Novamente não há grande influência do uso da memória principal.

As operações de escrita randômica e leitura randômica ficam bem próximas entre os sistemas de arquivos comparados, como é visto nos gráficos 17 e 16, respectivamente. Através desses gráficos é possível visualizar que o melhor resultado é obtido

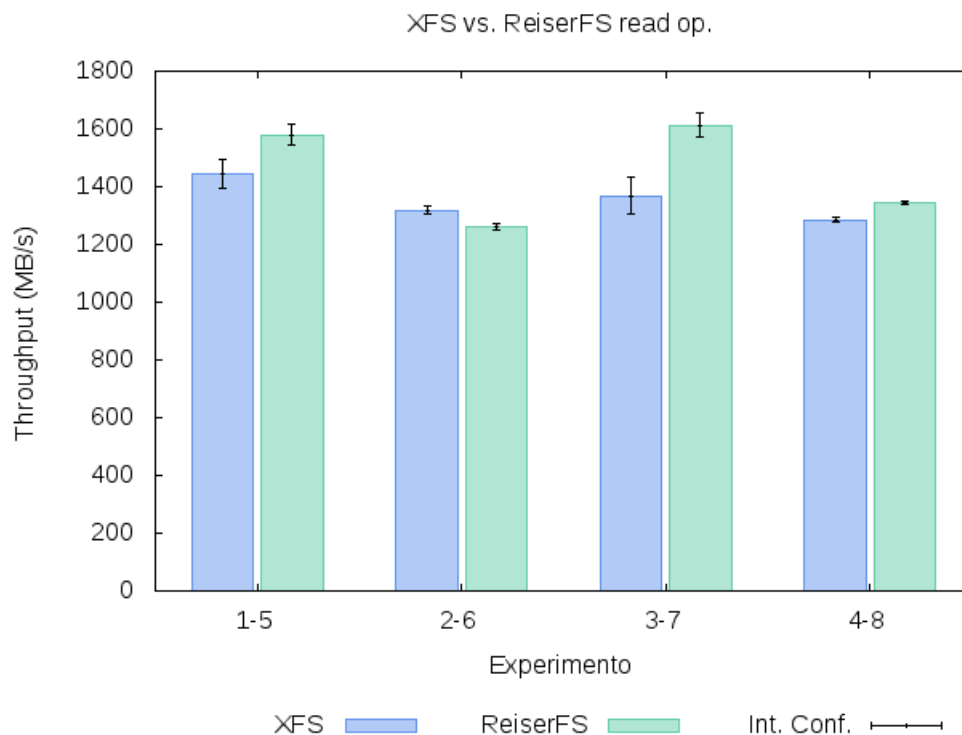


Figura 15: Gráfico comparando os experimentos da Etapa 3 na operação de Leitura.

nos experimentos 1 ou 5 em ambas operações. Estes experimentos utilizavam tamanhos de arquivos pequenos, de 64 KB e a memória principal de 2 GB. Note que a memória influencia muito nesses resultados.

Para o cálculo da influência dos fatores, seguem a tabela 5.3 com os dados e os gráficos de pizza na figura 5.3.

Fator(es)	Write	Read	Random Read	Random Write
iA	94.36	14.81	0.74	0.08
iC	0.10	0.01	23.52	56.45
iD	3.19	64.68	69.60	43.08
iAC	0.63	5.26	0.01	0.14
iAD	1.40	14.35	0.88	0.01
iCD	0.22	0.88	4.80	0.25
iACD	0.09	0.00	0.45	0.00

Tabela 8: Tabela com os valores calculados de influência dos fatores para a Etapa 3 da avaliação de desempenho separados para o tipo de operação.

Nos gráficos gerados a partir dessa tabela (vide Figura 5.3), é possível notar que os próprios sistemas de arquivos têm influência quando comparados para a operação de escrita. Também podemos notar que o tamanho de arquivos é um fator de grande

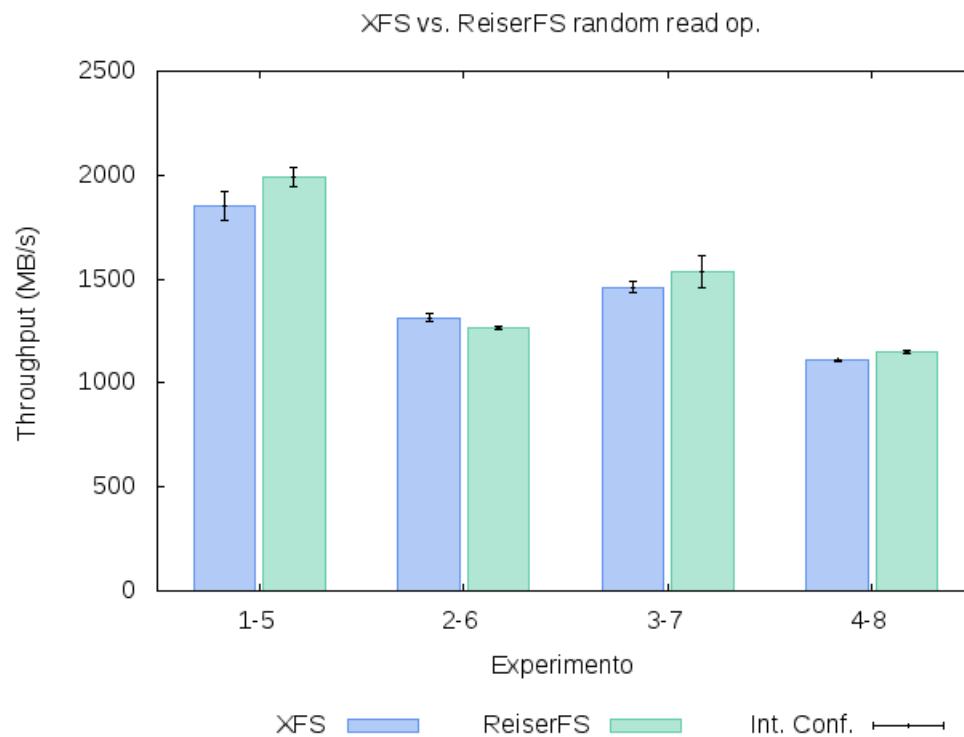


Figura 16: Gráfico comparando os experimentos da Etapa 3 na operação de Leitura aleatória.

influência quando comparamos XFS e ReiserFS para as operações de leitura, leitura randômica e escrita randômica. Arquivos maiores (64 MB) degradaram o desempenho de ambos os sistemas de arquivos. Por fim, a quantidade de memória disponível é também um fator de importância para operação de leitura randômica. Nesse caso, com o aumento da memória disponível há uma degradação no desempenho de ambos os sistemas de arquivos.

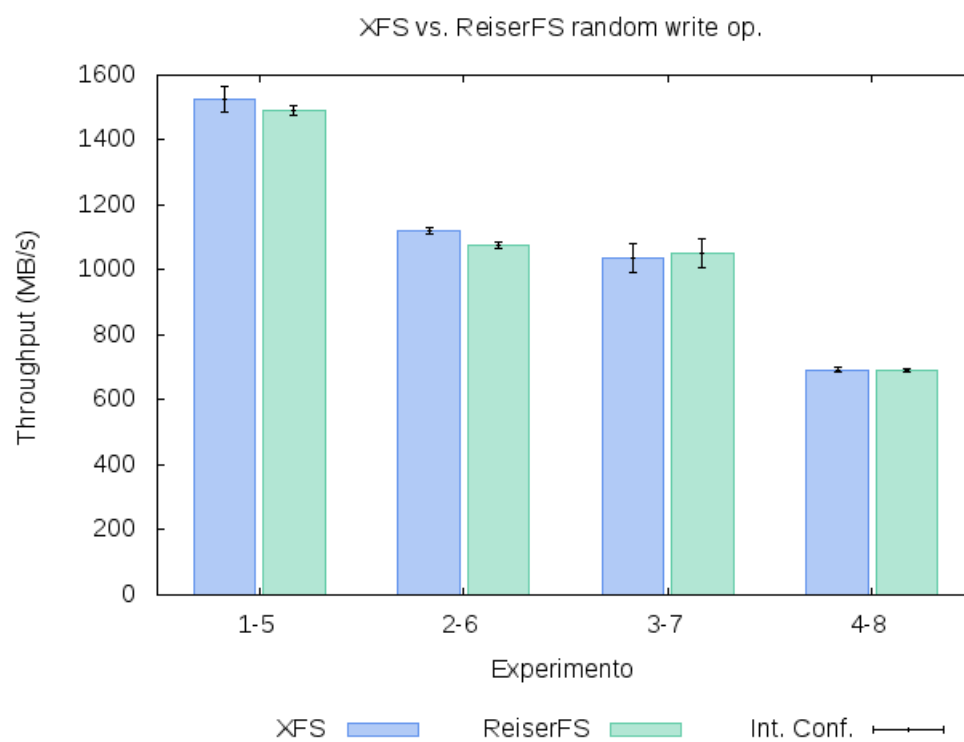


Figura 17: Gráfico comparando os experimentos da Etapa 3 na operação de Escrita aleatória.

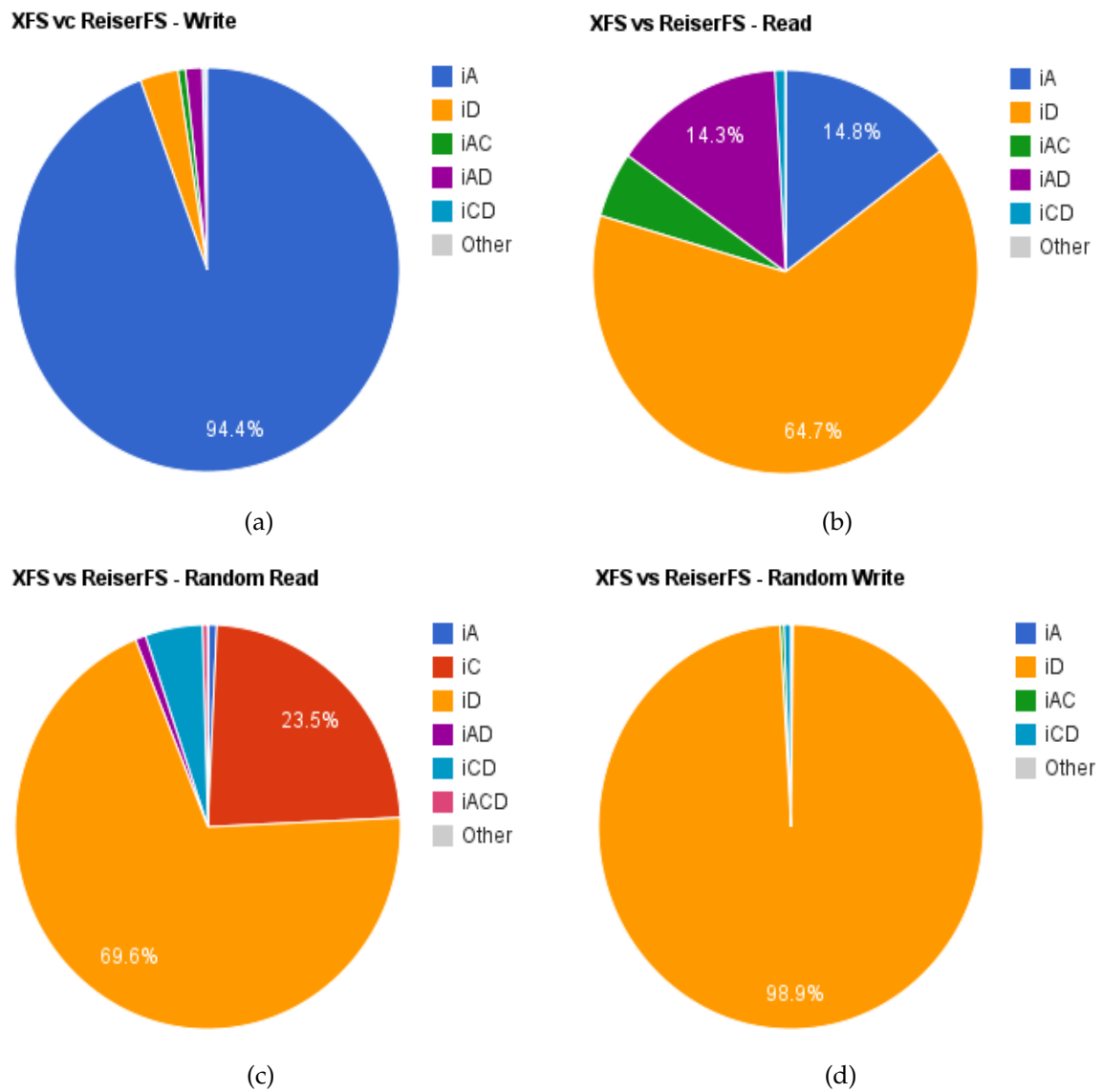


Figura 18: Gráficos com as influências dos fatores calculadas para as operações avaliadas na Etapa 3: (a) - Escrita, (b) - Leitura, (c) - Leitura aleatória e (d) - Escrita aleatória

6 *Conclusões*

Neste capítulo serão apresetadas as conclusões para as observações feitas a partir dos experimentos realizados e apresentados nos capítulos anteriores. Será feita uma análise para cada etapa de experimentos, seguida de uma conclusão final.

6.1 **Etapa 1: comparação entre Ext4 e XFS**

Na primeira etapa do trabalho os sistemas de arquivos Ext4 e XFS foram analisados em conjunto. A partir dessa análise, algumas conclusões são possíveis:

- O sistema de arquivos XFS teve melhor ou igual desempenho para todos os experimentos realizados envolvendo blocos de tamanho 4 KB;
- Para demais operações XFS e Ext4 obtiveram desempenhos similares, com diferenças tênues para grande parte dos casos;
- O tamanho de blocos é um fator que possui grande influência quando comparamos o desempenho de Ext4 e XFS para a operação de escrita; e
- O tamanho de arquivos é um fator de grande influência quando comparamos Ext4 e XFS para as operações de leitura, leitura randômica e escrita randômica.

6.2 **Etapa 2: comparação entre Ext4 e ReiserFS**

Na segunda etapa do trabalho os sistemas de arquivos Ext4 e ReiserFS foram analisados em conjunto. A partir dessa análise, algumas conclusões são possíveis:

- O sistema de arquivos Ext4 possui melhor desempenho que o ReiserFS para todos os experimentos realizados com a operação de escrita;
- O sistema de arquivos ReiserFS apresentou em geral um melhor desempenho que o Ext4 em experimentos envolvendo a operação de leitura para arquivos menores (64 KB);

- O sistema de arquivos ReiserFS apresentou em geral um melhor desempenho que o Ext4 em experimentos envolvendo a operação de escrita para configurações onde havia maior quantidade de memória (2 GB);
- Para demais operações Ext4 e ReiserFS obtiveram desempenhos similares, com diferenças tênues para grande parte dos casos;
- O tamanho de arquivos é um fator que possui grande influência quando comparamos o desempenho de Ext4 e ReiserFS para as operações de leitura, leitura randômica e escrita randômica — há queda de desempenho para arquivos maiores (64 MB);

6.3 Etapa 3

6.4 Considerações finais

De uma forma geral, quando utilizamos uma formatação padrão de um sistema de arquivos — que utiliza blocos de 4 KB — torna-se mais interessante o uso do sistema de arquivos XFS frente ao sistema de arquivos Ext4.

Referências

- [1] GRAPHICS, S. *XFS Filesystem Structure*. 2006. Disponível em: http://xfs.org/docs/xfsdocs-xml-dev/XFS_Filesystem_Structure/tmp/en-US/html/index.html.
- [2] MATHUR, A. et al. The new ext4 filesystem: current status and future plans. In: *Proceedings of the Linux Symposium*. [s.n.], 2007. Disponível em: <http://kernel.org/doc/ols/2007/ols2007v2-pages-21-34.pdf>.
- [3] SWEENEY, A. *Scalability in the XFS File System*. 2006. Disponível em: http://oss.sgi.com/projects/xfs/papers/xfs_usenix/index.html.
- [4] CARD, R.; TS'O, T.; TWEEDIE, S. Design and implementation of the second extended filesystem. In: *Proceedings of the First Dutch International Symposium on Linux*. [s.n.], 1994. Disponível em: <http://e2fsprogs.sourceforge.net/ext2intro.html>.
- [5] SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. *Operating System Concepts*. 8th. ed. [S.l.]: Wiley Publishing, 2008. ISBN 0470128720.
- [6] BUCHHOLZ, F. *The structure of the Reiser file system*. , <http://homes.cerias.purdue.edu/~florian/reiser/reiserfs.php> Último acesso em 12/06/2011.
- [7] IOZONE Filesystem Benchmark. 2006. Disponível em: <http://www.iozone.org/>.

ANEXO A – Scripts utilizados

A.1 Execução dos experimentos

```

1  #!/bin/bash
2
3  TEMP_DEVICE=/dev/sda9
4  TEMP_MOUNT=/tmp/test_fs
5  TEMP_FILE=/tmp/test_fs/temp_file
6  LOG_DIR=/home/fgrillo/Documents/sisop11/logs
7  IOZONE=/usr/bin/iozone
8  IOZONE_PARAMS="-R -r 4k -i 0 -i 1 -i 2 -i 8 -tu -f $TEMP_FILE"
9  NUM_EXPERIMENTS=10
10
11  function log {
12      echo [ `date '+%d/%m/%Y %X'` ] $1
13  }
14
15  if [ ! -d $TEMP_MOUNT ]; then
16      mkdir $TEMP_MOUNT
17  fi
18
19  if [ ! -d $LOG_DIR ]; then
20      exit 1
21  fi
22
23  for fs in ext4 reiserfs xfs; do
24      for bs in 4096 1024; do
25          if [ $fs = "xfs" ]; then
26              MKFS_PARAM="-f -b size=$bs"
27          fi
28          if [ $fs = "ext4" ]; then
29              MKFS_PARAM="-b $bs"
30          fi
31          if [ $fs = "reiserfs" ]; then
32              if [ $bs = 1024 ]; then
33                  continue
34              fi
35              MKFS_PARAM="-f -b $bs"
36          fi
37
38          log "Formatting $TEMP_DEVICE with fs:$fs bs:$bs"
39          mkfs.$fs $MKFS_PARAM $TEMP_DEVICE && /tmp/trace
40          if [ $? != 0 ]; then

```

```

41         exit 1
42     fi
43
44     log "Mounting $TEMP_MOUNT"
45     mount $TEMP_DEVICE $TEMP_MOUNT && /tmp/trace
46     if [ $? != 0 ]; then
47         exit 1
48     fi
49
50     for size in 64k 64m; do
51         for ((i=0; i<$NUM_EXPERIMENTS; i++)); do
52             log "Executing test for fs:$fs bs:$bs fsize:$size exp:$i"
53
54             $IOZONE $IOZONE_PARAMS -s $size > $LOG_DIR/experiment_${fs}_${bs}_${size}_${i}.txt
55             if [ $? != 0 ]; then
56                 exit 1
57             fi
58         done
59     done
60
61     log "Unmounting $TEMP_MOUNT"
62     umount $TEMP_MOUNT && /tmp/trace
63     if [ $? != 0 ]; then
64         exit 1
65     fi
66 done
67 done

```

Listagem A.1: Script bash utilizado para execução dos experimentos. Foi executado uma vez com a máquina com 2GB de memória RAM e outra com 1GB de memória RAM.

A.2 Extração dos dados do log

```

1  #!/bin/bash
2
3  LOG_DIR=/home/fgrillo/Documents/sisop11/logs
4
5  mp=$1
6  fs=$2
7  bs=$3
8  fsize=$4
9
10 echo 'KB reflen write rewrite read reread read write'
11 for file in `ls $LOG_DIR/${mp}GB/experiment_${fs}_${bs}_${fsize}_*`; do
12     grep -A1 -E '^\\s*KB' $file | tail -1 | awk '{print $1,$2,$3,$4,$5,$6,$6,$7 }'
13 done

```

Listagem A.2: Script bash utilizado para extrair os dados dos log dos experimentos para um formato separado por vírgulas que possibilitava a importação para uma

planilha do OpenOffice.

A.3 Geração dos gráficos de barra

```

1  set boxwidth 0.3
2  set xrange [0:9]
3  set yrange [0:]
4  set xlabel "Experimento"
5  set ylabel "Throughput (MB/s)"
6  set xtics ("1-9" 1, "2-10" 2, "3-11" 3, "4-12" 4, "5-13" 5, "6-14" 6, "7-15" 7, "8-16" 8)
7  set key below
8  set terminal png
9
10 set style fill solid 0.5
11
12
13 # EXT4 vs. XFS
14
15 set title "EXT4 vs. XFS write op."
16 set output "ext4_xfs/ext4_xfs_write.png"
17 plot "ext4.dat" using ($1-0.17):($5/1024) title "EXT4" with boxes lc rgb "#FFC1AA",\
18      "xfs.dat" using ($1+0.17):($5/1024) title "XFS" with boxes lc rgb "#6495ED",\
19      "ext4.dat" using ($1-0.17):($5/1024):($6/1024) title "Int. Conf." with yerrorbars lc ←
20      rgb "black" pt 1 ps 0.3,\
21      "xfs.dat" using ($1+0.17):($5/1024):($6/1024) notitle with yerrorbars lc rgb "black" pt←
22      1 ps 0.3
23
24 set title "EXT4 vs. XFS read op."
25 set output "ext4_xfs/ext4_xfs_read.png"
26 plot "ext4.dat" using ($1-0.17):($7/1024) title "EXT4" with boxes lc rgb "#FFC1AA",\
27      "xfs.dat" using ($1+0.17):($7/1024) title "XFS" with boxes lc rgb "#6495ED",\
28      "ext4.dat" using ($1-0.17):($7/1024):($8/1024) title "Int. Conf." with yerrorbars lc ←
29      rgb "black" pt 1 ps 0.3,\
30      "xfs.dat" using ($1+0.17):($7/1024):($8/1024) notitle with yerrorbars lc rgb "black" pt←
31      1 ps 0.3
32
33 set title "EXT4 vs. XFS random read op."
34 set output "ext4_xfs/ext4_xfs_randread.png"
35 plot "ext4.dat" using ($1-0.17):($9/1024) title "EXT4" with boxes lc rgb "#FFC1AA",\
36      "xfs.dat" using ($1+0.17):($9/1024) title "XFS" with boxes lc rgb "#6495ED",\
37      "ext4.dat" using ($1-0.17):($9/1024):($10/1024) title "Int. Conf." with yerrorbars lc ←
38      rgb "black" pt 1 ps 0.3,\
39      "xfs.dat" using ($1+0.17):($9/1024):($10/1024) notitle with yerrorbars lc rgb "black" ←
40      pt 1 ps 0.3
41
42 set title "EXT4 vs. XFS random write op."
43 set output "ext4_xfs/ext4_xfs_randwrite.png"
44 plot "ext4.dat" using ($1-0.17):($11/1024) title "EXT4" with boxes lc rgb "#FFC1AA",\
45      "xfs.dat" using ($1+0.17):($11/1024) title "XFS" with boxes lc rgb "#6495ED",\
46      "ext4.dat" using ($1-0.17):($11/1024):($12/1024) title "Int. Conf." with yerrorbars lc ←
47      rgb "black" pt 1 ps 0.3,\
48      "xfs.dat" using ($1+0.17):($11/1024):($12/1024) notitle with yerrorbars lc rgb "black" ←
49      pt 1 ps 0.3

```

```

43
44
45 set xtics ("1-5" 1, "2-6" 2, "3-7" 3, "4-8" 4)
46 # EXT4 vs. ReiserFS
47
48 set title "EXT4 vs. ReiserFS write op."
49 set output "ext4_reiser/ext4_reiserfs_write.png"
50 plot [0.5:4.5]\
51     "ext4.dat" using ($1-0.17):($5/1024) title "EXT4" with boxes lc rgb "#FFC1AA",\
52     "reiserfs.dat" using ($1+0.17):($5/1024) title "ReiserFS" with boxes lc rgb "#66CDAA",\
53     "ext4.dat" using ($1-0.17):($5/1024):($6/1024) title "Int. Conf." with yerrorbars lc ←
54     rgb "black" pt 1 ps 0.3,\
55     "reiserfs.dat" using ($1+0.17):($5/1024):($6/1024) notitle with yerrorbars lc rgb "←
56     black" pt 1 ps 0.3
57
58 set title "EXT4 vs. ReiserFS read op."
59 set output "ext4_reiser/ext4_reiserfs_read.png"
60 plot [0.5:4.5]\
61     "ext4.dat" using ($1-0.17):($7/1024) title "EXT4" with boxes lc rgb "#FFC1AA",\
62     "reiserfs.dat" using ($1+0.17):($7/1024) title "ReiserFS" with boxes lc rgb "#66CDAA",\
63     "ext4.dat" using ($1-0.17):($7/1024):($8/1024) title "Int. Conf." with yerrorbars lc ←
64     rgb "black" pt 1 ps 0.3,\
65     "reiserfs.dat" using ($1+0.17):($7/1024):($8/1024) notitle with yerrorbars lc rgb "←
66     black" pt 1 ps 0.3
67
68 set title "EXT4 vs. ReiserFS random read op."
69 set output "ext4_reiser/ext4_reiserfs_randread.png"
70 plot [0.5:4.5]\
71     "ext4.dat" using ($1-0.17):($9/1024) title "EXT4" with boxes lc rgb "#FFC1AA",\
72     "reiserfs.dat" using ($1+0.17):($9/1024) title "ReiserFS" with boxes lc rgb "#66CDAA",\
73     "ext4.dat" using ($1-0.17):($9/1024):($10/1024) title "Int. Conf." with yerrorbars lc ←
74     rgb "black" pt 1 ps 0.3,\
75     "reiserfs.dat" using ($1+0.17):($9/1024):($10/1024) notitle with yerrorbars lc rgb "←
76     black" pt 1 ps 0.3
77
78 set title "EXT4 vs. ReiserFS random write op."
79 set output "ext4_reiser/ext4_reiserfs_randwrite.png"
80 plot [0.5:4.5]\
81     "ext4.dat" using ($1-0.17):($11/1024) title "EXT4" with boxes lc rgb "#FFC1AA",\
82     "reiserfs.dat" using ($1+0.17):($11/1024) title "ReiserFS" with boxes lc rgb "#66CDAA",\
83     "ext4.dat" using ($1-0.17):($11/1024):($12/1024) title "Int. Conf." with yerrorbars lc ←
84     rgb "black" pt 1 ps 0.3,\
85     "reiserfs.dat" using ($1+0.17):($11/1024):($12/1024) notitle with yerrorbars lc rgb "←
86     black" pt 1 ps 0.3
87
88 # XFS vs. ReiserFS
89
90 set title "XFS vs. ReiserFS write op."
91 set output "xfs_reiser/xfs_reiserfs_write.png"
92 plot [0.5:4.5]\
93     "xfs.dat" using ($1-0.17):($5/1024) title "XFS" with boxes lc rgb "#6495ED",\
94     "reiserfs.dat" using ($1+0.17):($5/1024) title "ReiserFS" with boxes lc rgb "#66CDAA",\
95     "xfs.dat" using ($1-0.17):($5/1024):($6/1024) title "Int. Conf." with yerrorbars lc rgb ←
96     "black" pt 1 ps 0.3,\
97     "reiserfs.dat" using ($1+0.17):($5/1024):($6/1024) notitle with yerrorbars lc rgb "←
98     black" pt 1 ps 0.3
99

```

```

91 set title "XFS vs. ReiserFS read op."
92 set output "xfs_reiser/xfs_reiserfs_read.png"
93 plot [0.5:4.5]\
94     "xfs.dat" using ($1-0.17):($7/1024) title "XFS" with boxes lc rgb "#6495ED",\
95     "reiserfs.dat" using ($1+0.17):($7/1024) title "ReiserFS" with boxes lc rgb "#66CDAA",\
96     "xfs.dat" using ($1-0.17):($7/1024):($8/1024) title "Int. Conf." with yerrorbars lc rgb↵
97     "black" pt 1 ps 0.3,\
98     "reiserfs.dat" using ($1+0.17):($7/1024):($8/1024) notitle with yerrorbars lc rgb "↵
99     black" pt 1 ps 0.3
100
101 set title "XFS vs. ReiserFS random read op."
102 set output "xfs_reiser/xfs_reiserfs_randread.png"
103 plot [0.5:4.5]\
104     "xfs.dat" using ($1-0.17):($9/1024) title "XFS" with boxes lc rgb "#6495ED",\
105     "reiserfs.dat" using ($1+0.17):($9/1024) title "ReiserFS" with boxes lc rgb "#66CDAA",\
106     "xfs.dat" using ($1-0.17):($9/1024):($10/1024) title "Int. Conf." with yerrorbars lc ↵
107     rgb "black" pt 1 ps 0.3,\
108     "reiserfs.dat" using ($1+0.17):($9/1024):($10/1024) notitle with yerrorbars lc rgb "↵
109     black" pt 1 ps 0.3
110
111 set title "XFS vs. ReiserFS random write op."
112 set output "xfs_reiser/xfs_reiserfs_randwrite.png"
113 plot [0.5:4.5]\
114     "xfs.dat" using ($1-0.17):($11/1024) title "XFS" with boxes lc rgb "#6495ED",\
115     "reiserfs.dat" using ($1+0.17):($11/1024) title "ReiserFS" with boxes lc rgb "#66CDAA"↵
116     ,\
117     "xfs.dat" using ($1-0.17):($11/1024):($12/1024) title "Int. Conf." with yerrorbars lc ↵
118     rgb "black" pt 1 ps 0.3,\
119     "reiserfs.dat" using ($1+0.17):($11/1024):($12/1024) notitle with yerrorbars lc rgb "↵
120     black" pt 1 ps 0.3

```

Listagem A.3: Script do Gnuplot utilizado gerar os gráficos de barra para comparação dos experimentos. Ele utiliza arquivos de dados que foram extraídos das tabelas que podem ser vistas no Anexo B

ANEXO B – Dados coletados

B.1 Ext4

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ext4	4096	1GB	64	367747	524351	1047367	1186265	1186265	1122807
ext4	4096	1GB	64	361811	470229	1085345	1232105	1232105	1523810
ext4	4096	1GB	64	347888	520818	1231068	1390185	1390185	1422222
ext4	4096	1GB	64	407736	620992	1488832	1733638	1733638	1523806
ext4	4096	1GB	64	385363	587512	1360757	1731165	1731165	1361701
ext4	4096	1GB	64	410152	604135	1454257	1686988	1686988	1422222
ext4	4096	1GB	64	405042	659035	1491918	1685741	1685741	1488372
ext4	4096	1GB	64	423643	702723	1638474	1943148	1943148	1523806
ext4	4096	1GB	64	418571	696422	1565425	1877324	1877324	1560976
ext4	4096	1GB	64	383372	633140	1523493	1941083	1941083	1684212
Total				3911325	6019357	13886936	16407642	16407642	14633934
Média				391132.5	601935.7	1388693.6	1640764.2	1640764.2	1463393.4
Desvio padrão				25843.39	77346.21	203012.63	277870.29	277870.29	148917.31
Intervalo confiança				16017.6	47938.8	125826.22	172222.63	172222.63	92298.21

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ext4	4096	1GB	65536	478250	720762	1370473	1370761	1370761	1147459
ext4	4096	1GB	65536	481861	686125	1346980	1336544	1336544	1150238
ext4	4096	1GB	65536	472015	666700	1312345	1307974	1307974	1096268
ext4	4096	1GB	65536	479699	681331	1343003	1340585	1340585	1114103
ext4	4096	1GB	65536	474458	726458	1339275	1339193	1339193	1145614
ext4	4096	1GB	65536	495370	720555	1371248	1374413	1374413	1166473
ext4	4096	1GB	65536	475577	678146	1310272	1308993	1308993	1136002
ext4	4096	1GB	65536	476227	669615	1315957	1315457	1315457	1114804
ext4	4096	1GB	65536	477900	715543	1356177	1361162	1361162	1130184
ext4	4096	1GB	65536	496203	723635	1377967	1374413	1374413	1129016
Total				4807560	6988870	13443697	13429495	13429495	11330161
Média				480756	698887	1344369.7	1342949.5	1342949.5	1133016.1
Desvio padrão				8382.24	24477.17	25170.9	26411.23	26411.23	20698.92
Intervalo confiança				5195.27	15170.83	15600.8	16369.55	16369.55	12829.09

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ext4	4096	2GB	64	392409	570972	1491676	1831017	1488372	864864
ext4	4096	2GB	64	378612	688642	1598075	1933403	1523810	914286
ext4	4096	2GB	64	470504	890335	1526921	1945039	1523806	914286
ext4	4096	2GB	64	355655	673268	1520817	1887929	1523810	888889
ext4	4096	2GB	64	410127	702477	1562538	1730908	1523810	876712
ext4	4096	2GB	64	424141	659234	1101787	1888046	1560976	761905
ext4	4096	2GB	64	410313	614939	1392445	1886206	1488372	914286
ext4	4096	2GB	64	429259	694913	1102373	1881223	1560972	761905
ext4	4096	2GB	64	412982	703400	1104466	1886653	1523810	771084
ext4	4096	2GB	64	415799	680871	1556865	1832848	1560976	901408
Total				4099801	6879051	13957963	18703272	15278714	8569625
Média				409980.1	687905.1	1395796.3	1870327.2	1527871.4	856962.5
Desvio padrão				30772.05	82706.38	209258.65	60744.7	26854.32	65618.51
Intervalo confiança				19072.36	51261	129697.47	37649.27	16644.18	40670.03

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ext4	4096	2GB	65536	497737	733860	1322064	1325993	1164277	675399
ext4	4096	2GB	65536	485963	727094	1312344	1314873	1172190	680193
ext4	4096	2GB	65536	486024	723587	1311294	1317915	1178302	682581
ext4	4096	2GB	65536	494652	734444	1321024	1321265	1157573	677682
ext4	4096	2GB	65536	500890	733499	1347756	1348311	1174775	672005
ext4	4096	2GB	65536	451294	715764	1354247	1356404	1151491	638391
ext4	4096	2GB	65536	496756	738025	1342182	1345127	1179893	668727
ext4	4096	2GB	65536	497945	726209	1349201	1351591	1156347	669636
ext4	4096	2GB	65536	498353	743029	1340834	1355088	1124078	671805
ext4	4096	2GB	65536	493821	727562	1313533	1320676	1171352	676221
Total				4903435	7303073	13314479	13357243	11630278	6712640
Média				490343.5	730307.3	1331447.9	1335724.3	1163027.8	671264
Desvio padrão				14614.73	7836.19	16973.98	16942.25	16783.27	12388.28
Intervalo confiança				9058.14	4856.83	10520.39	10500.72	10402.19	7678.19

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ext4	1024	1GB	64	304617	633659	1451534	1824097	1824097	1523810
ext4	1024	1GB	64	249071	609403	1451504	1881545	1881545	1523810
ext4	1024	1GB	64	329760	633028	1452347	1782640	1782640	1523806
ext4	1024	1GB	64	310758	582196	1388321	1826819	1826819	1488372
ext4	1024	1GB	64	334961	520797	1360552	1883021	1883021	1454545
ext4	1024	1GB	64	321714	616007	1524583	1336090	1336090	1523810
ext4	1024	1GB	64	340552	640109	1454745	1773439	1773439	1454545
ext4	1024	1GB	64	312253	610016	1487044	1880831	1880831	1488372
ext4	1024	1GB	64	343945	660175	1522556	1826339	1826339	1560976
ext4	1024	1GB	64	326526	608933	1422836	1734923	1734923	1488368
Total				3174157	6114323	14516022	17749744	17749744	15030414
Média				317415.7	611432.3	1451602.2	1774974.4	1774974.4	1503041.4
Desvio padrão				27297.88	38386.08	52439.75	162022.27	162022.27	34044.23
Intervalo confiança				16919.09	23791.5	32501.9	100420.6	100420.6	21100.44

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ext4	1024	1GB	65536	229375	625940	1313345	1316989	1316989	1129424
ext4	1024	1GB	65536	362959	617216	1297381	1298591	1298591	1095974
ext4	1024	1GB	65536	369977	616716	1289874	1291220	1291220	1096781
ext4	1024	1GB	65536	367743	606618	1285271	1287344	1287344	1098252
ext4	1024	1GB	65536	350396	589485	1296357	1300551	1300551	1080293
ext4	1024	1GB	65536	370804	644024	1321422	1324388	1324388	1116875
ext4	1024	1GB	65536	352679	586451	1271110	1283963	1283963	1077345
ext4	1024	1GB	65536	351942	592699	1264267	1260211	1260211	1108393
ext4	1024	1GB	65536	352408	593806	1276287	1272419	1272419	1089162
ext4	1024	1GB	65536	377159	635340	1309305	1316166	1316166	1108562
Total				3485442	6108295	12924619	12951842	12951842	11001061
Média				348544.2	610829.5	1292461.9	1295184.2	1295184.2	1100106.1
Desvio padrão				42944.86	20255.47	18763.86	20421.76	20421.76	16134.43
Intervalo confiança				26617.01	12554.24	11629.74	12657.3	12657.3	10000.04

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ext4	1024	2GB	64	321671	673437	1523358	1878652	1523810	735632
ext4	1024	2GB	64	265507	666104	1602398	2006144	1599996	853333
ext4	1024	2GB	64	337018	673298	1602674	1941309	1600001	914284
ext4	1024	2GB	64	300302	634038	1487072	1780487	1560976	853333
ext4	1024	2GB	64	333249	687664	1643475	1937683	1600001	901408
ext4	1024	2GB	64	324794	615333	1522384	1886297	1488372	888889
ext4	1024	2GB	64	326594	639309	1564779	2072453	1684207	901408
ext4	1024	2GB	64	347852	666484	1487148	1933982	1560976	853333
ext4	1024	2GB	64	303481	688814	1598243	1773594	1560972	853333
ext4	1024	2GB	64	333204	680505	1595275	1935078	1560976	864864
Total				3193672	6624986	15626806	19145679	15740287	8619817
Média				319367.2	662498.6	1562680.6	1914567.9	1574028.7	861981.7
Desvio padrão				23830.08	24692.65	54393.71	91470.32	52390.91	50376.53
Intervalo confiança				14769.77	15304.38	33712.95	56692.85	32471.62	31223.12

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ext4	1024	2GB	65536	229029	663877	1305134	1315613	1190092	608166
ext4	1024	2GB	65536	387065	672204	1371792	1375041	1179617	596823
ext4	1024	2GB	65536	384130	680631	1369641	1371531	1159744	593112
ext4	1024	2GB	65536	339093	663386	1340618	1329414	1134370	626233
ext4	1024	2GB	65536	342750	661752	1381360	1373488	1184349	612423
ext4	1024	2GB	65536	338150	655432	1368812	1373951	1190914	593736
ext4	1024	2GB	65536	344284	661318	1343996	1344384	1175048	610580
ext4	1024	2GB	65536	391980	690369	1378343	1379966	1194560	614213
ext4	1024	2GB	65536	346931	668468	1352623	1366413	1171415	610972
ext4	1024	2GB	65536	343956	672199	1341986	1334718	1141046	605022
Total				3447368	6689636	13554305	13564519	11721155	6071280
Média				344736.8	668963.6	1355430.5	1356451.9	1172115.5	607128
Desvio padrão				46005.47	10360.62	23367.96	23186.15	20918.91	10314.11
Intervalo confiança				28513.96	6421.46	14483.35	14370.66	12965.43	6392.63

B.2 XFS

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
xfs	4096	1GB	64	467331	736215	1487136	1939508	1939508	1641027
xfs	4096	1GB	64	467189	673322	1487792	1944151	1944151	1560972
xfs	4096	1GB	64	477423	726431	1491785	1935704	1935704	1560976
xfs	4096	1GB	64	485285	702560	1563685	1996490	1996490	1641022
xfs	4096	1GB	64	495677	710644	1563347	1995867	1995867	1641022
xfs	4096	1GB	64	426676	726906	1528007	1824775	1824775	1523806
xfs	4096	1GB	64	484882	711635	1487748	1942476	1942476	1523806
xfs	4096	1GB	64	481527	726514	1524587	1938619	1938619	1600001
xfs	4096	1GB	64	477682	680786	1330937	1637835	1637835	1454545
xfs	4096	1GB	64	488201	718542	1336041	1779864	1779864	1488372
Total				4751873	7113555	14801065	18935289	18935289	15635549
Média				475187.3	711355.5	1480106.5	1893528.9	1893528.9	1563554.9
Desvio padrão				19193.72	20622.25	82661.95	113031.27	113031.27	66701.2
Intervalo confiança				11896.17	12781.57	51233.47	70056.22	70056.22	41341.07

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
xfs	4096	1GB	65536	537748	751438	1374499	1377180	1377180	1159600
xfs	4096	1GB	65536	551798	775289	1378805	1380575	1380575	1144134
xfs	4096	1GB	65536	531068	766341	1364537	1369015	1369015	1158924
xfs	4096	1GB	65536	536371	767795	1349200	1352314	1352314	1175575
xfs	4096	1GB	65536	521094	722764	1314268	1258274	1258274	1139538
xfs	4096	1GB	65536	518710	719037	1348505	1345624	1345624	1146014
xfs	4096	1GB	65536	523216	718509	1329603	1336514	1336514	1139300
xfs	4096	1GB	65536	550986	777182	1355171	1359417	1359417	1135884
xfs	4096	1GB	65536	545142	761621	1356378	1340342	1340342	1117560
xfs	4096	1GB	65536	545696	766636	1326641	1334009	1334009	1143475
Total				5361829	7526612	13497607	13453264	13453264	11460004
Média				536182.9	752661.2	1349760.7	1345326.4	1345326.4	1146000.4
Desvio padrão				12308.5	23563.93	20948.16	34760.18	34760.18	15756.28
Intervalo confiança				7628.75	14604.8	12983.57	21544.19	21544.19	9765.66

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
xfs	4096	2GB	64	429752	744544	1491226	1778120	1488372	1084745
xfs	4096	2GB	64	415749	703545	1423286	1680315	1454545	1049180
xfs	4096	2GB	64	481046	727996	1424606	1783039	1488372	1066666
xfs	4096	2GB	64	457121	727801	1454953	1834328	1488372	1084745
xfs	4096	2GB	64	470667	743393	1453096	1828891	1560972	1122805
xfs	4096	2GB	64	444749	711770	1102912	1779542	1523810	853333
xfs	4096	2GB	64	470296	761842	1424772	1824013	1488368	1084745
xfs	4096	2GB	64	488356	726990	1423688	1826054	1560976	1122807
xfs	4096	2GB	64	473949	726908	1394225	1772633	1488372	1084745
xfs	4096	2GB	64	477619	762627	1425219	1777723	1422222	1066666
Total				4609304	7337416	14017983	17884658	14964381	10620437
Média				460930.4	733741.6	1401798.3	1788465.8	1496438.1	1062043.7
Desvio padrão				23751.28	19405.07	108208.62	45573.89	43034.12	76893.77
Intervalo confiança				14720.93	12027.17	67067.17	28246.47	26672.34	47658.38

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
<u>xfs</u>	4096	2GB	65536	538769	828856	1333764	1335342	1129171	710078
<u>xfs</u>	4096	2GB	65536	540918	810408	1303003	1305548	1145453	703266
<u>xfs</u>	4096	2GB	65536	515410	805387	1335097	1337797	1146135	712440
<u>xfs</u>	4096	2GB	65536	539076	793356	1317490	1325511	1118266	699520
<u>xfs</u>	4096	2GB	65536	527639	807769	1311115	1313582	1133664	707472
<u>xfs</u>	4096	2GB	65536	545501	797955	1323635	1328117	1136455	704976
<u>xfs</u>	4096	2GB	65536	507103	783043	1326556	1330894	1138331	710463
<u>xfs</u>	4096	2GB	65536	539590	804517	1283835	1312662	1140867	700283
<u>xfs</u>	4096	2GB	65536	503902	803245	1313636	1319347	1138825	742348
<u>xfs</u>	4096	2GB	65536	537810	787949	1313502	1316538	1120177	704529
Total				5295718	8022485	13161633	13225338	11347344	7095375
Média				529571.8	802248.5	1316163.3	1322533.8	1134734.4	709537.5
Desvio padrão				15252.34	12874.78	15242.97	10641.4	9605.11	12299.9
Intervalo confiança				9453.32	7979.72	9447.51	6595.48	5953.2	7623.42

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
<u>xfs</u>	1024	1GB	64	333532	743493	1643386	1888175	1888175	1560976
<u>xfs</u>	1024	1GB	64	345947	688358	1491809	1834371	1834371	1560976
<u>xfs</u>	1024	1GB	64	338733	660550	1363626	1830870	1830870	1600001
<u>xfs</u>	1024	1GB	64	336979	673419	1452950	1935187	1935187	1523810
<u>xfs</u>	1024	1GB	64	337000	688862	1490285	1935888	1935888	1560972
<u>xfs</u>	1024	1GB	64	333261	673194	1450749	1886899	1886899	1560972
<u>xfs</u>	1024	1GB	64	359783	560912	1418947	1877202	1877202	1488368
<u>xfs</u>	1024	1GB	64	338567	681331	1485208	1884493	1884493	1560976
<u>xfs</u>	1024	1GB	64	336682	681094	1334876	1680084	1680084	1454545
<u>xfs</u>	1024	1GB	64	334947	652288	1524202	1996816	1996816	1600001
Total				3395431	6703501	14656038	18749985	18749985	15471597
Média				339543.1	670350.1	1465603.8	1874998.5	1874998.5	1547159.7
Desvio padrão				7959.78	45545.28	86131.25	84439.01	84439.01	46089.03
Intervalo confiança				4933.43	28228.74	53383.72	52334.88	52334.88	28565.75

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
<u>xfs</u>	1024	1GB	65536	353050	697673	1311977	1340372	1340372	1084153
<u>xfs</u>	1024	1GB	65536	353621	682894	1350505	1353799	1353799	1085914
<u>xfs</u>	1024	1GB	65536	355027	670534	1328034	1333845	1333845	1139974
<u>xfs</u>	1024	1GB	65536	346300	687053	1343336	1334907	1334907	1128374
<u>xfs</u>	1024	1GB	65536	356396	687831	1229893	1229889	1229889	1120273
<u>xfs</u>	1024	1GB	65536	349919	682630	1352680	1353689	1353689	1148102
<u>xfs</u>	1024	1GB	65536	355806	703523	1355643	1360514	1360514	1067987
<u>xfs</u>	1024	1GB	65536	349997	689751	1348119	1351929	1351929	1142379
<u>xfs</u>	1024	1GB	65536	356656	717064	1332032	1343805	1343805	1135471
<u>xfs</u>	1024	1GB	65536	352247	692082	1358402	1361307	1361307	1098049
Total				3529019	6911035	13310621	13364056	13364056	11150676
Média				352901.9	691103.5	1331062.1	1336405.6	1336405.6	1115067.6
Desvio padrão				3352.3	12752.14	38366.91	38684.75	38684.75	28654.7
Intervalo confiança				2077.74	7903.71	23779.62	23976.62	23976.62	17760.04

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
<u>xfs</u>	1024	2GB	64	279417	603903	1452067	1832103	1488368	955224
<u>xfs</u>	1024	2GB	64	345801	703455	1451542	1823728	1488372	984616
<u>xfs</u>	1024	2GB	64	338742	710725	1455454	1775020	1488372	969696
<u>xfs</u>	1024	2GB	64	336959	702871	1453634	1733009	1560976	1000001
<u>xfs</u>	1024	2GB	64	338832	696211	1389998	1725725	1488372	941175
<u>xfs</u>	1024	2GB	64	340585	718140	1422899	1783670	1523810	969696
<u>xfs</u>	1024	2GB	64	340543	689029	1455805	1783149	1523810	984616
<u>xfs</u>	1024	2GB	64	344275	703508	1362734	1780054	1454545	969696
<u>xfs</u>	1024	2GB	64	343891	702670	1068569	1776649	1488372	790124
<u>xfs</u>	1024	2GB	64	334948	711027	1420472	1773954	1488372	999999
Total				3343993	6941539	13933174	17787061	14993369	9564843
Média				334399.3	694153.9	1393317.4	1778706.1	1499336.9	956484.3
Desvio padrão				19613.04	32715.74	19613.04	32715.74	118446.28	33099.21
Intervalo confiança				12156.07	20277.05	12156.07	20277.05	73412.42	20514.73

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
<u>xfs</u>	1024	2GB	65536	369244	742500	1323290	1325458	1141961	631556
<u>xfs</u>	1024	2GB	65536	371884	756032	1312347	1316330	1132469	659634
<u>xfs</u>	1024	2GB	65536	360658	757572	1324518	1334121	1135254	642635
<u>xfs</u>	1024	2GB	65536	371447	759211	1300445	1304562	1102464	635667
<u>xfs</u>	1024	2GB	65536	360194	740293	1314403	1317733	1139855	643386
<u>xfs</u>	1024	2GB	65536	372846	760216	1315718	1315559	1127520	644265
<u>xfs</u>	1024	2GB	65536	368500	750380	1286582	1291858	1121826	645732
<u>xfs</u>	1024	2GB	65536	366897	759679	1306720	1311452	1120273	650804
<u>xfs</u>	1024	2GB	65536	358506	742423	1286712	1293261	1092885	626371
<u>xfs</u>	1024	2GB	65536	362435	764465	1322196	1324255	1109050	642075
Total				3662611	7532771	13092931	13134589	11223557	6422125
Média				366261.1	753277.1	1309293.1	1313458.9	1122355.7	642212.5
Desvio padrão				5366.27	8733.63	14069.36	13646.79	16437.66	9444.89
Intervalo confiança				3325.98	5413.06	8720.12	8458.21	10187.98	5853.9

B.3 ReiserFS

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ReiserFS	4096	1GB	64	3539	984965	1489671	2001644	2001644	1523810
ReiserFS	4096	1GB	64	303267	900884	1686739	1933138	1933138	1560976
ReiserFS	4096	1GB	64	306233	954289	1683480	2134177	2134177	1523806
ReiserFS	4096	1GB	64	309036	901485	1563218	1999656	1999656	1523810
ReiserFS	4096	1GB	64	310652	876257	1595469	1942963	1942963	1488372
ReiserFS	4096	1GB	64	303391	926644	1637488	2126089	2126089	1560972
ReiserFS	4096	1GB	64	309233	927962	1638650	2128048	2128048	1560976
ReiserFS	4096	1GB	64	309010	900559	1595719	2004183	2004183	1523810
ReiserFS	4096	1GB	64	306148	875329	1639280	2060561	2060561	1523810
ReiserFS	4096	1GB	64	319887	902663	1642071	2068064	2068064	1488372
Total				2780396	9151037	16171785	20398523	20398523	15278714
Média				278039.6	915103.7	1617178.5	2039852.3	2039852.3	1527871.4
Desvio padrão				96564.54	34253.44	58989.53	74878.08	74878.08	26854.32
Intervalo confiança				59850.22	21230.11	36561.41	46409.06	46409.06	16644.18

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ReiserFS	4096	1GB	65536	270991	720571	1283763	1285697	1285697	1102687
ReiserFS	4096	1GB	65536	269487	714312	1303578	1302565	1302565	1123866
ReiserFS	4096	1GB	65536	274031	725196	1284110	1285599	1285599	1097406
ReiserFS	4096	1GB	65536	271655	703893	1270494	1269193	1269193	1086508
ReiserFS	4096	1GB	65536	277916	732647	1278502	1285551	1285551	1093287
ReiserFS	4096	1GB	65536	274166	727336	1284138	1282753	1282753	1096873
ReiserFS	4096	1GB	65536	278445	762763	1302436	1306984	1306984	1120675
ReiserFS	4096	1GB	65536	271065	725878	1285550	1289115	1289115	1094784
ReiserFS	4096	1GB	65536	281142	769933	1311792	1314687	1314687	1095022
ReiserFS	4096	1GB	65536	274476	736806	1311588	1312375	1312375	1135924
Total				2743374	7319335	12915951	12934519	12934519	11047032
Média				274337.4	731933.5	1291595.1	1293451.9	1293451.9	1104703.2
Desvio padrão				3782.3	20406.33	14510.35	14836.81	14836.81	16224.76
Intervalo confiança				2344.25	12647.74	8993.44	9195.78	9195.78	10056.03

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
ReiserFS	4096	2GB	64	3785	1050722	1641493	2067147	1599996	876712
ReiserFS	4096	2GB	64	320170	1047191	1643912	2006070	1600001	1084745
ReiserFS	4096	2GB	64	300447	1047708	1636090	1934159	1600001	1084745
ReiserFS	4096	2GB	64	321757	986179	1726912	2003796	1600001	1122805
ReiserFS	4096	2GB	64	329701	954849	1729999	1995932	1641027	1103448
ReiserFS	4096	2GB	64	310834	1034011	1561278	1936086	1599996	1066666
ReiserFS	4096	2GB	64	316937	940055	1681816	1940604	1560972	1122807
ReiserFS	4096	2GB	64	316900	1067327	1527421	1999293	1600001	1066666
ReiserFS	4096	2GB	64	312350	984403	1733785	2072195	1684212	1142857
ReiserFS	4096	2GB	64	321715	999770	1641429	2063957	1230767	1084745
Total				2854596	10112215	16524135	20019239	15716974	10756196
Média				285459.6	1011221.5	1652413.5	2001923.9	1571697.4	1075619.6
Desvio padrão				99278.26	44187.19	69572.36	53531.45	124106	74328.69
Intervalo confiança				61532.17	27387	43120.6	33178.53	76920.28	46068.55

File system	Block size	Memory	File Size	write	rewrite	read	reread	rand read	rand write
<u>ReiserFS</u>	4096	2GB	65536	322675	803718	1375851	1374872	1169867	704294
<u>ReiserFS</u>	4096	2GB	65536	306320	739141	1357979	1352906	1154554	702610
<u>ReiserFS</u>	4096	2GB	65536	310353	776253	1388152	1394149	1180020	699856
<u>ReiserFS</u>	4096	2GB	65536	317175	798595	1386920	1383932	1186923	714725
<u>ReiserFS</u>	4096	2GB	65536	319231	794317	1368497	1374639	1180360	697837
<u>ReiserFS</u>	4096	2GB	65536	321574	797769	1391360	1389415	1196873	728380
<u>ReiserFS</u>	4096	2GB	65536	318748	782378	1380548	1383641	1166390	696369
<u>ReiserFS</u>	4096	2GB	65536	314656	795695	1380487	1380025	1182981	706001
<u>ReiserFS</u>	4096	2GB	65536	302953	713417	1375074	1385100	1182938	711736
<u>ReiserFS</u>	4096	2GB	65536	321659	808348	1369355	1364568	1170390	704415
Total				3155344	7809631	13774223	13783247	11771296	7066223
Média				315534.4	780963.1	1377422.3	1378324.7	1177129.6	706622.3
Desvio padrão				6851.4	30897.94	10271.44	12214.59	11998.35	9545.64
Intervalo confiança				4246.46	19150.39	6366.19	7570.54	7436.52	5916.34