PLATAFORMA DE DESENVOLVIMENTO PINHÃO PARANÁ

ANÁLISE DE DESEMPENHO DO REPOSITÓRIO SCRIBA

Outubro – 2008

Sumário de Informações do Documento

Tipo do Documento: Relatório

Título do Documento: Análise de Desempenho do Repositório Scriba

Estado do Documento: Elaborado

Responsáveis: Leslie Harlley Watter

Palavras-Chaves: ANÁLISE, DESEMPENHO, PROTOAGENTE, PARÂMETROS

Resumo: Análise de influência dos parâmetros do jackrabbit no protoagente Scriba.

Número de páginas: 44

Software utilizados: OpenOffice

Versão	Data	Mudanças
1.0	27/10/2008	Elaboração
1.1	24/11/2008	Atualização – Respostas à questionamentos e Dados sobre Configurações Recomendadas

SUMÁRIO

1 Motivação

A necessidade de planejar o repositório de documentos utilizado pelo protoagente Documentador e fornecer o melhor desempenho na sua utilização levaram à realização dos testes relatados nesse documento.

O objetivo dos testes foi identificar a relação entre os diversos valores para os parâmetros de configuração que o *jackrabbit*[1] fornece para que fosse possível identificar os melhores valores a serem utilizados em nossa configuração.

As seções a seguir descrevem o ambiente de testes e o resultado dos testes propriamente ditos, tanto para as operações de inclusão de arquivos em lote como para os testes de pesquisa. A seção 5 traz os resultados obtidos para o consumo de espaço em disco quando da indexação de um número maior de palavras (ou *tokens*) por arquivo. Ao final, as seções 6 e 7 apresentam os resultados obtidos e a conclusão dos testes.

2 Hardware Utilizado

processador

- AMD Athlon™ XP 2600+
- 1.9 GHz

disco rígido

- SAMSUNG SP0842N
- modo de operação UDMA5
- velocidade leitura/gravação 54.55 MB/sec

Memória RAM

- Total: 1024MBytes

Sistema Operacional

- Debian Linux Etch

Configurações da VM utilizada pelo Jboss

- -Xms128m

1 O jackrabbit (http://jackrabbit.apache.org/) é uma implementação para repositório de conteúdos da JSR170 e JSR283.

- -Xmx512m
- -XX:MaxPermSize=128m
- -XX:+AggressiveOpts
- -Dsun.rmi.dgc.client.gcInterval=3600000
- -Dsun.rmi.dgc.server.gcInterval=3600000

2.1 Observações Importantes

2.1.1 Configuração Inicial dos Testes

Para todos os testes executados, partiu-se da seguinte configuração para os parâmetros do jackrabbit:

bufferSize = 10
cacheSize = 1000
extractorPoolSize = 0
maxFieldLength = 10000
mergeFactor = 10
minMergeDocs = 100

A única exceção à essa regra foi para a medida do parâmetro *mergeFactor* onde o valor utilizado para *minMergeDocs* foi de 200.

Essa configuração foi escolhida como configuração base por ser a configuração atualmente utilizada no servidor da CELEPAR.

2.1.2 Gráficos

Os gráficos apresentados nas seções 3 e 4 tem como *eixo X* um número arbitrário, utilizado apenas para separar as execuções de teste, enquanto que no *eixo Y*, as variáveis encontram-se expressas em milisegundos. A legenda abaixo dos gráficos é composta de p-VAL-NUMa, onde VAL indica o valor aplicado ao parâmetro para aquele teste e NUM indica o número de arquivos utilizados em cada teste.

2.1.3 Tempo de Inclusão de Arquivos

O tempo de inclusão de arquivos apresentado é a somatória dos tempos da inclusão dos arquivo individualmente, não considerando tempo de inicialização do jboss, nem mesmo tempo de espera entre uma operação de inclusão e outra.

2.1.4 Operação de Inclusão

A execução dos testes foi executada com a seguinte configuração: uma única máquina contendo os arquivos a serem incluídos e a estrutura do servidor, composta por servidor de banco de dados e servidor de aplicação. Essa configuração certamente influencia os tempos de inclusão de arquivos, uma vez que a taxa de transferência de dados utilizando o WebService é a máxima possível. Por outro lado, é necessário salientar que existe concorrência na utilização do disco rígido quando da inserção de arquivos, por parte do servidor de aplicação e do servidor de banco de dados.

2.1.5 Operação de Pesquisa

O tempo médio de pesquisa foi obtido através da média aritmética de 30 pesquisas consecutivas com itens diferentes mas que seriam encontrados nos mesmos arquivos dentro do repositório. Dessa maneira elimina-se o atraso da busca ao repositório na primeira vez e obtém-se o tempo médio de pesquisa para um item que esteja na memória cache do servidor de aplicação, sem contanto desconsiderar o tempo de pesquisa. O teste foi executado procurando-se por trinta padrões diferentes, mas que poderiam ser encontrados no mesmo arquivo. Uma vez carregado este arquivo na memória do servidor de aplicação, com todos os seus *tokens*, as próximas pesquisas se valem da cache para retornar mais rapidamente os primeiros resultados. Como a abrangência da busca utilizada na pesquisa envolvia todo o repositório, acredita-se que somente para repositórios de tamanho significativo (onde os índices do repositório não caibam na memória RAM) ocorra diferença significativa nos resultados.

3 Análise de Resultados – Inclusão de Arquivos

As próximas seções analisam os parâmetros de configuração do jackrabbit em conjunto com o protoagente Scriba onde:

bufferSize
: indica o número máximo de documentos que são mantidos em uma fila de pendências até que sejam adicionados ao índice.

cacheSize
: tamanho da cache de documentos. Essa cache mapeia uuids para os números de documento do lucene.

extractorPoolSize
: indica o número de threads a ser utilizado para indexação do conteúdo dos arquivos incluídos no repositório.

maxFieldLength
: indica o número máximo de palavras que serão indexadas por propriedade.

mergeFactor
: indica a freqüência de agregação de segmentos de índices.

minMergeDocs
indica o número mínimo de arquivos a serem agregados em uma única transação para escrita do índice.

3.1 Influência do parâmetro bufferSize

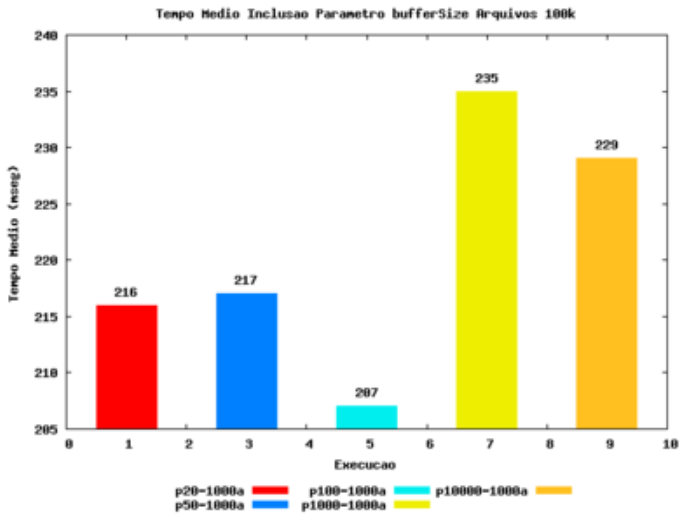

Figura 1: Tempo Médio de Inclusão bufferSize 100KB 1000 arquivos

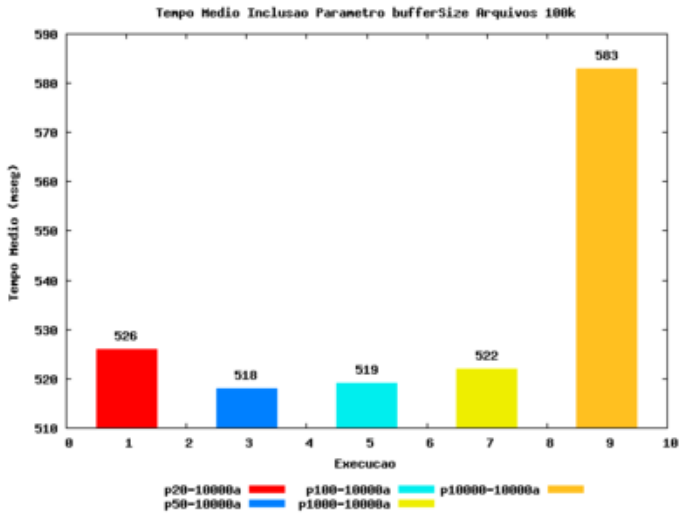

Figura 2: Tempo Médio de Inclusão bufferSize 100KB 1000 arquivos0

Tam/Param	20	50	100	1000	10000
100k	3	3	3	3	3
1M	10	9	9	10	11

10M	43	42	44	43	46

Tabela 1: Tempo de Inclusão de 1000 Arquivos (em min) – bufferSize

Analisando as figuras 1 e 2 observa-se que o menor tempo de inclusão é obtido quando o parâmetro bufferSize tem o valor 100. Esta diferença de 10 milisegundos entre os tempos de inclusão com valores 50 e 100, respectivamente, não se verifica quando aumentamos a quantidade de arquivos incluídos de 1.000 para 10.000. Ocorre, entretanto, uma mudança de comportamento, onde o tempo médio de inclusão de arquivos para a configuração com parâmetro igual a 50 é menor que o tempo médio de inclusão com parâmetro igual a 100. Embora essa diferença não seja significativa para arquivos com tamanho entre 100 e 300 KBytes, observa-se que a diferença se intensifica à medida em que aumenta-se o número de arquivos incluídos juntamente com o tamanho dos arquivos; vide figura 3,figura 4 e figura 5.

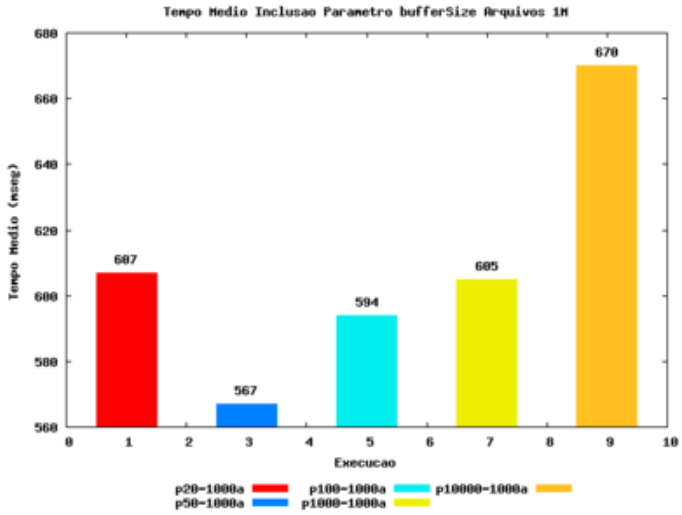

Figura 3: Tempo Médio de Inclusão bufferSize 1MB 1000 arquivos

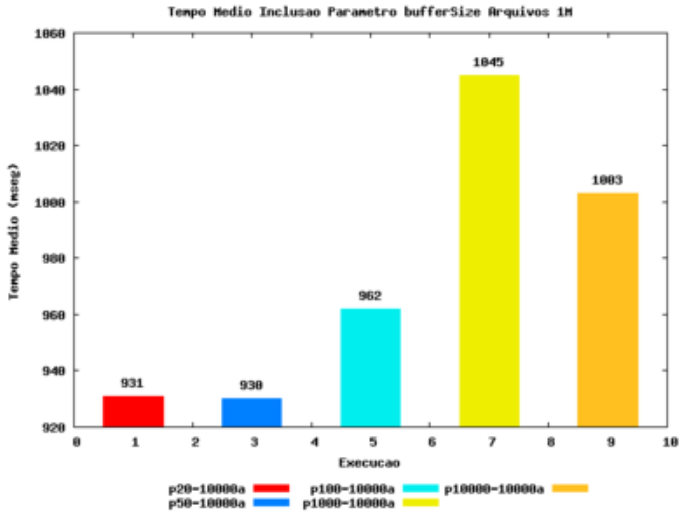

Figura 4: Tempo Médio de Inclusão bufferSize 1MB 10000 arquivos

Tam/Param	20	50	100	1000	10000
100k	87	86	86	87	97
1M	155	155	160	174	167

Tabela 2: Tempo de Inclusão de 10000 Arquivos (em min) – bufferSize

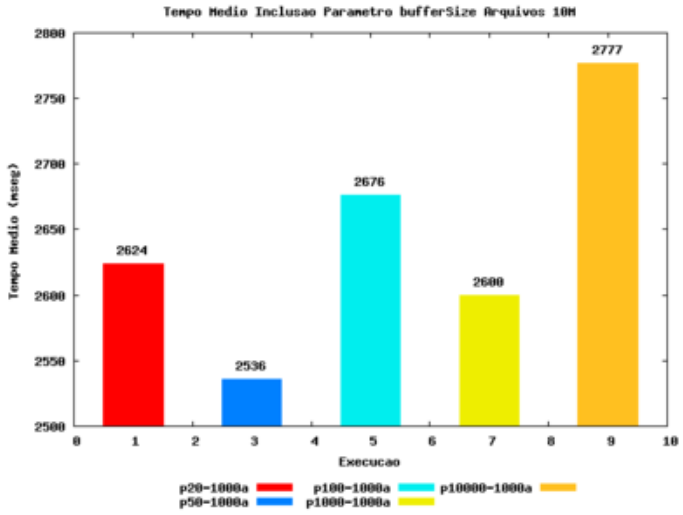

Figura 5: Tempo Médio de Inclusão bufferSize 10MB 1000 arquivos

As diferenças no tempo médio de inclusão estão diretamente relacionadas com o tamanho e quantidade dos arquivos a serem incluídos. Observando as figuras 1 e 5, é possível verificar a relação de tamanhos 100kBytes-10Mbytes e tempos de inclusão da ordem de 216-2600milisegundos respectivamente.

A partir das observações acima, observa-se que para a operação de inserção de arquivos no repositório, independente do número de arquivos a serem incluídos no repositório, para arquivos com tamanhos médios de 100KBytes, 1MBytes e 10MBytes o melhor valor para o parâmetro *bufferSize* é 50.

3.2 Influência do parâmetro cacheSize

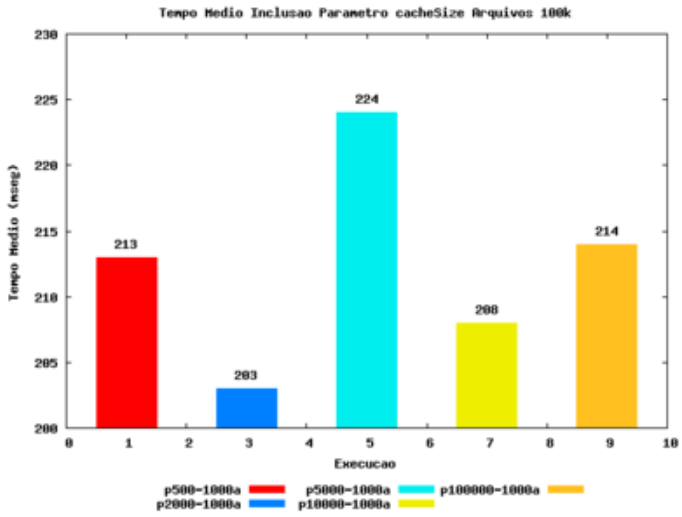

Figura 6: Tempo Médio de Inclusão cacheSize 100KB 1000 arquivos

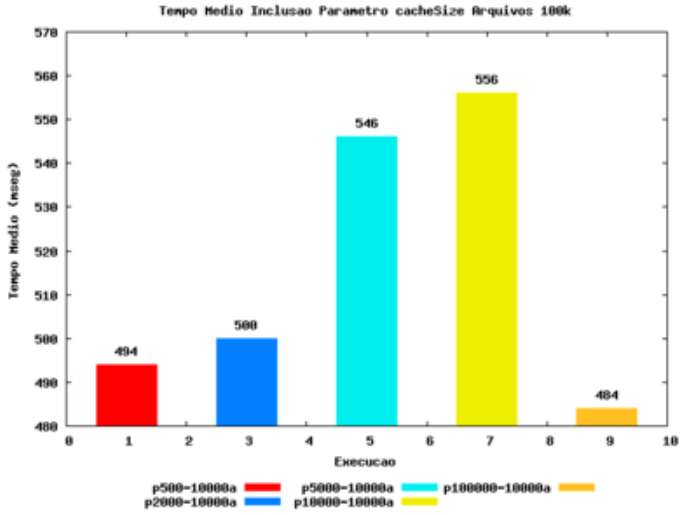

Figura 7: Tempo Médio de Inclusão cacheSize 100KB 10000 arquivos

Tam/Param	**500**	**2000**	**5000**	**10000**	**100000**
100k	3	3	3	3	3
1M	10	10	10	10	10
10M	43	42	44	45	40

Tabela 3: Tempo de Inclusão de 1000 Arquivos (em min) – cacheSize

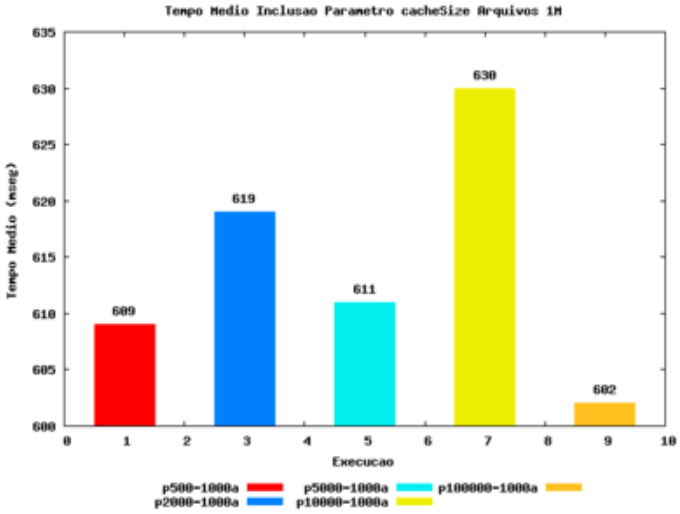

Figura 8: Tempo Médio de Inclusão cacheSize 1MB 1000 arquivos

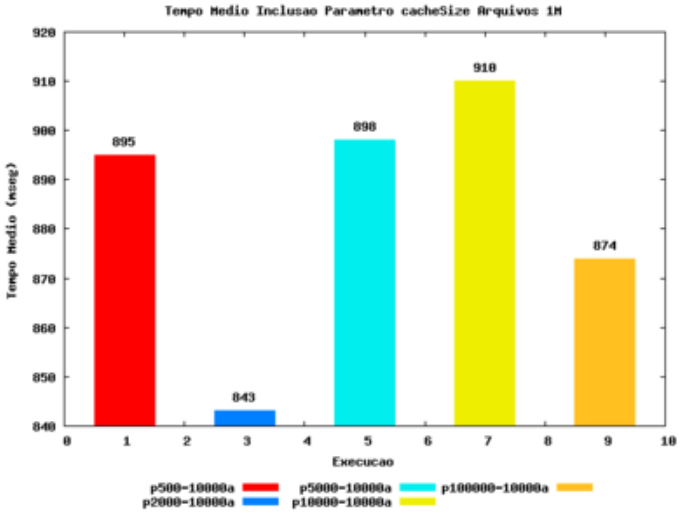

Figura 9: Tempo Médio de Inclusão cacheSize 1MB 10000 arquivos

Tam/Param	**500**	**2000**	**5000**	**10000**	**100000**
100k	82	83	91	92	80
1M	149	140	149	151	145

Tabela 4: Tempo de Inclusão de 10000 Arquivos (em min) – cacheSize

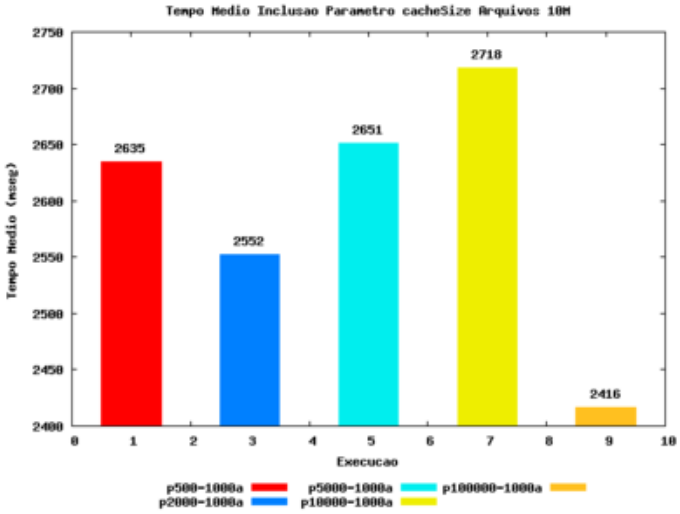

Figura 10: Tempo Médio de Inclusão cacheSize 10MB 1000 arquivos

Observa-se das figuras 7, 8 e 10 que quanto maior o volume/tamanho dos dados, o menor tempo médio de inclusão é obtido utilizando-se o maior valor para o parâmetro *cacheSize*, independente da quantidade/tamanho de arquivos utilizados.

Porém, observa-se que na figura 9 o menor tempo ocorre para o valor 2.000. Esta divergência pode ser explicada por uma coincidência do tamanho médio de arquivos e memória com o tamanho do cache especificado, maximizando o resultado obtido. Desconsiderando-se essa medida, observa-se que, para os demais valores associados ao parâmetro, confirma-se a tendência de que, quanto maior o tamanho do cache, menor o tempo médio de inclusão, para quantidade e tamanho de arquivos que preencham a cache completamente.

3.3 Influência do parâmetro extractorPoolSize

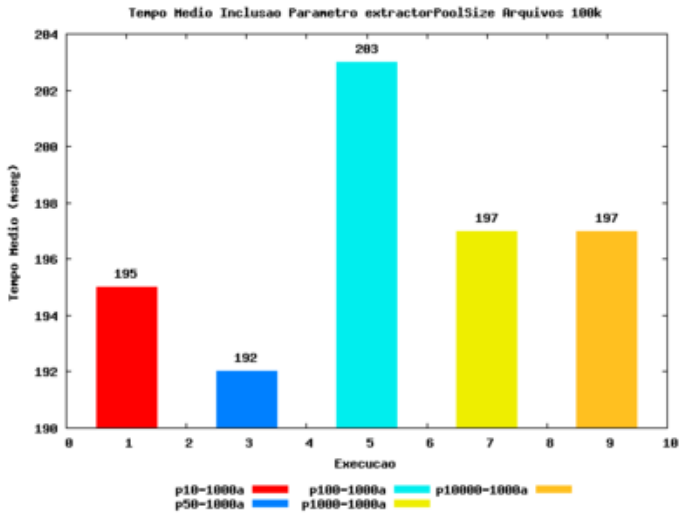

Figura 11: Tempo Médio de Inclusão extractorPoolSize 100KB 1000 arquivos

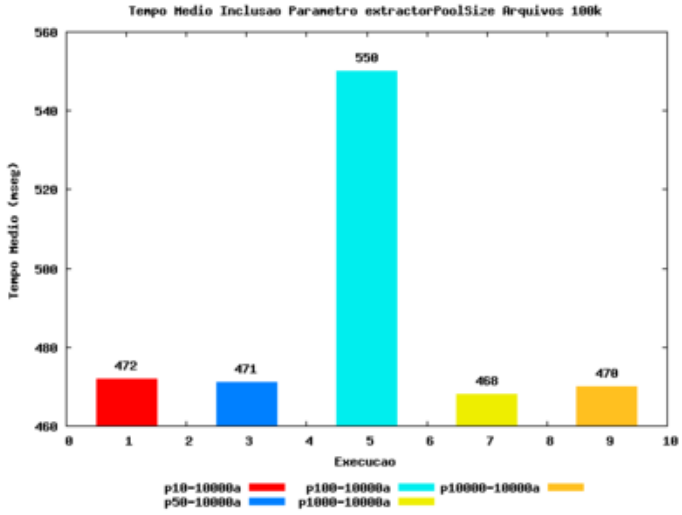

Figura 12: Tempo Médio de Inclusão extractorPoolSize 100KB 10000 arquivos

Tam/Param	**10**	**50**	**100**	**1000**	**10000**
100k	3	3	3	3	3
1M	8	8	8	8	8
10M	41	41	40	40	40

Tabela 5: Tempo de Inclusão de 1000 Arquivos (em min) – extractorPoolSize

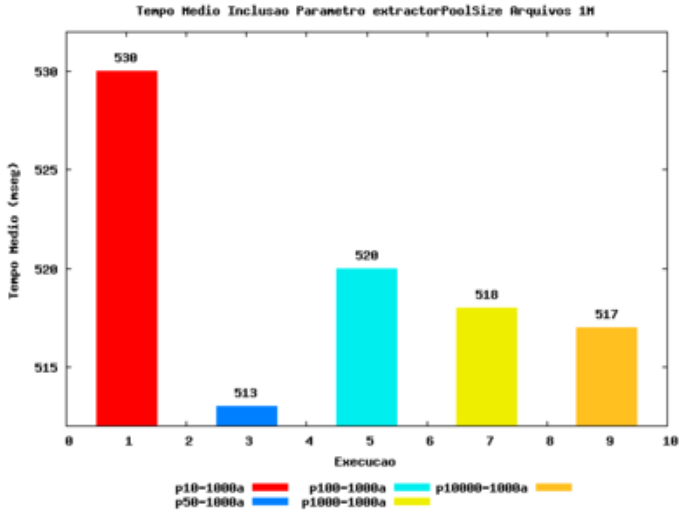

Figura 13: Tempo Médio de Inclusão extractorPoolSize 1MB 1000 arquivos

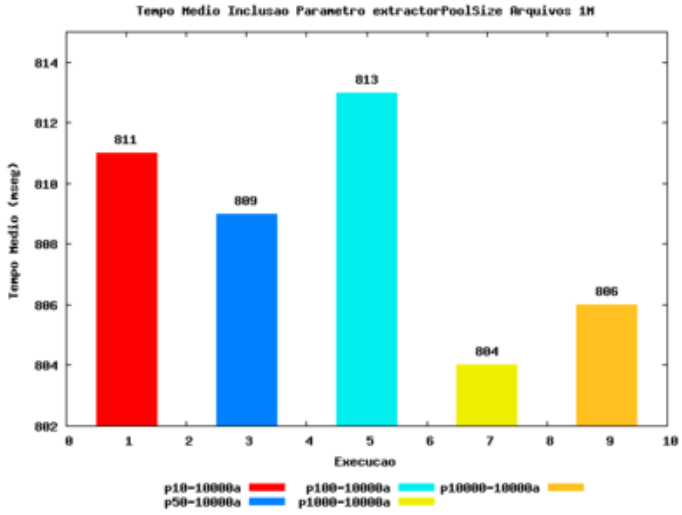

Figura 14: Tempo Médio de Inclusão extractorPoolSize 1MB 10000 arquivos

Tam/Param	**10**	**50**	**100**	**1000**	**10000**
100k	78	78	91	78	78
1M	135	134	135	134	134

Tabela 6: Tempo de Inclusão de 10000 Arquivos (em min) – extractorPoolSize

Analisando as figuras 11 e 13 em conjunto com as figuras 12 e 14 observamos que o menor tempo médio de inclusão de arquivos está relacionado diretamente com o número de arquivos a serem incluídos. Quanto maior o número de arquivos a serem incluídos, o menor tempo médio de inclusão ocorre com um número maior de threads. Observe que, para o tamanho de 100KBytes com 1.000 arquivos o menor tempo médio é obtido com 50 threads, enquanto que para o mesmo tamanho, porém para 10.000 arquivos, o menor tempo médio é obtido utilizando-se 1.000 threads. O mesmo comportamento ocorre para o tamanho de arquivos ao redor de

1MByte.

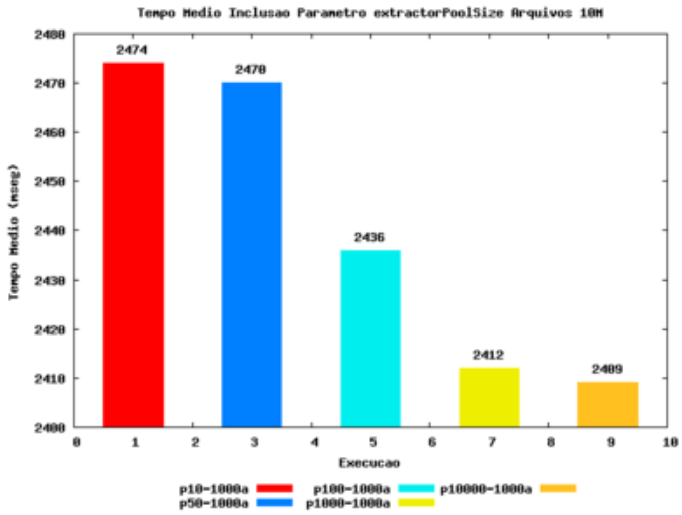

Figura 15: Tempo Médio de Inclusão extractorPoolSize 10MB 1000 arquivos

Observa-se também que o menor tempo médio para arquivos de 10MBytes é obtido utilizando-se o maior valor de threads (10.000) testado. Desse comportamento podemos inferir que, à medida em que o número e tamanho dos arquivos aumenta, se faz necessário utilizar mais threads para indexar os arquivos com o menor tempo possível.

3.4 Influência do parâmetro maxFieldLength

Nota: O valor máximo para o parâmetro *maxFieldLength* é $2^{31}-1$ ou 2147483647 ou ainda a constante **Integer.MAX_VALUE** do Java. Atribuindo esse valor ao parâmetro *maxFieldLength* o único limitador na indexação de documentos será a quantidade de memória disponível.

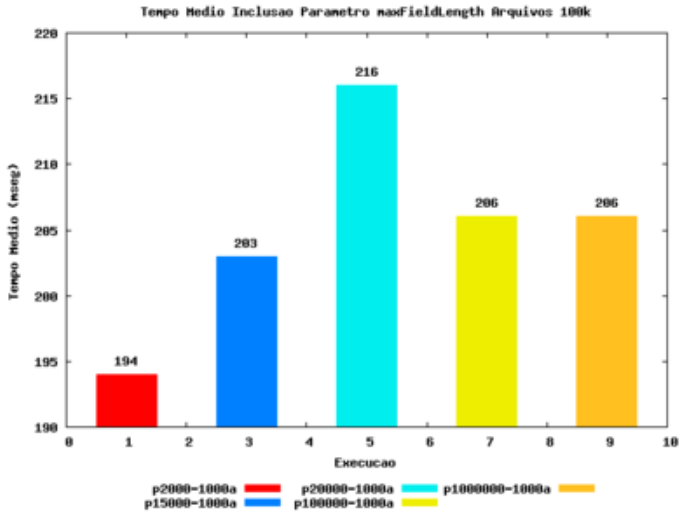

Figura 16: Tempo Médio de Inclusão maxFieldLength 100KB 1000 arquivos

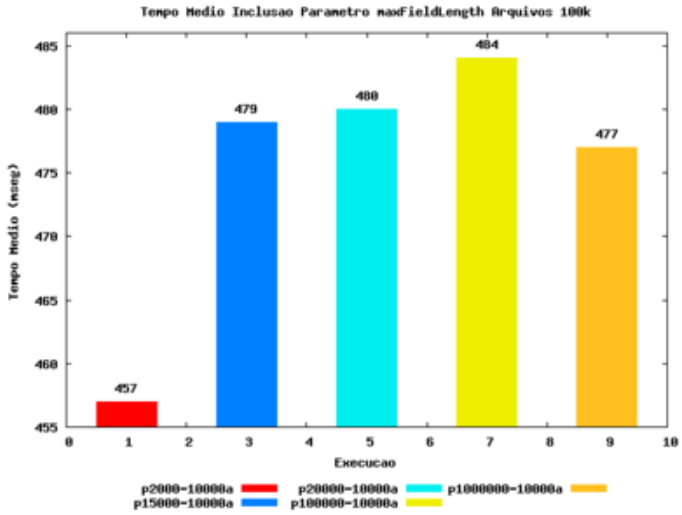

Figura 17: Tempo Médio de Inclusão maxFieldLength 100KB 10000 arquivos

Tam/Param	2000	15000	20000	100000	1000000
100k	3	3	3	3	3
1M	9	9	10	11	13
10M	41	42	42	42	48

Tabela 7: Tempo de Inclusão de 1000 Arquivos (em min) – maxFieldLength

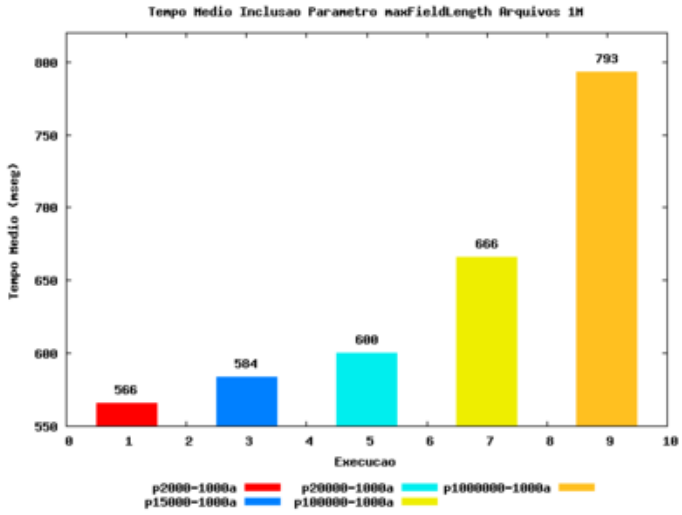

Figura 18: Tempo Médio de Inclusão maxFieldLength 1MB 1000 arquivos

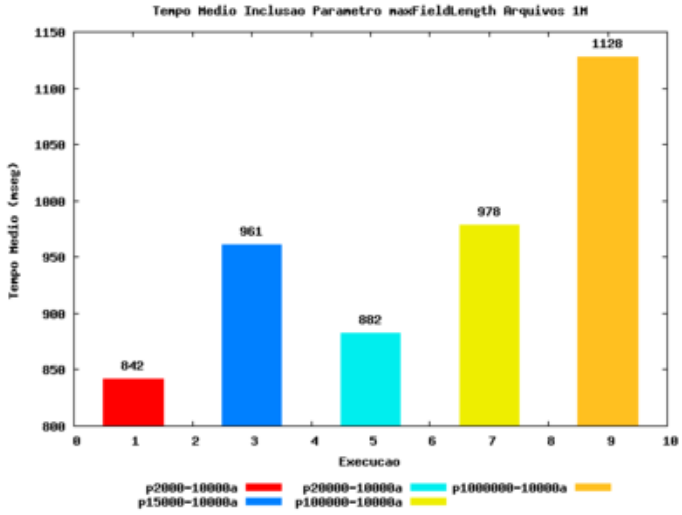

Figura 19: Tempo Médio de Inclusão maxFieldLength 1MB 10000 arquivos

Tam/Param	**2000**	**15000**	**20000**	**100000**	**1000000**
100k	76	79	80	80	79
1M	140	160	147	163	188

Tabela 8: Tempo de Inclusão de 10000 Arquivos (em min) – cacheSize

De maneira unânime em todos os testes realizados figuras 16, 17, 18, 19 e 20, o menor tamanho utilizado para o parâmetro *maxFieldLength*, respondeu com mais rapidez, independente do tamanho e quantidade de arquivos incluídos. Observa-se também que a diferença entre o menor tempo e o segundo menor tempo cresce à medida em que aumenta-se o número de arquivos e o tamanho destes (passa de 9mseg para 20 mseg para 1000 arquivos de 100Kbytes ao passo que passa de 18mseg para 40mseg quando o tamanho de arquivos cresce

para 1MByte). Aparentemente todo o arquivo é analisado e tem seus tokens extraídos, independente do tamanho do arquivo, porém o valor do parâmetro determina quantos desses *tokens* serão armazenados.

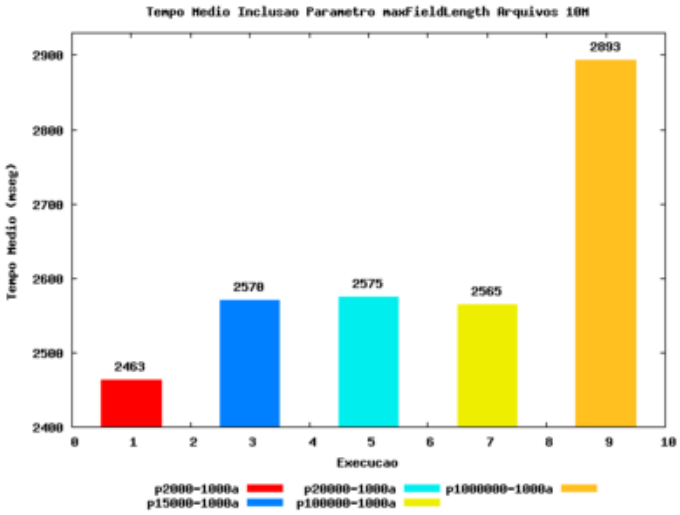

Figura 20: Tempo Médio de Inclusão maxFieldLength 10MB 1000 arquivos

3.5 Influência do parâmetro mergeFactor

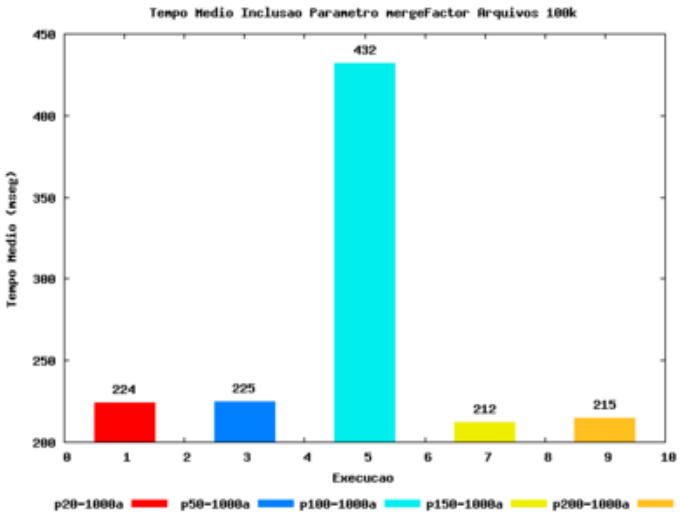

Figura 21: Tempo Médio de Inclusão mergeFactor100KB 1000 arquivos

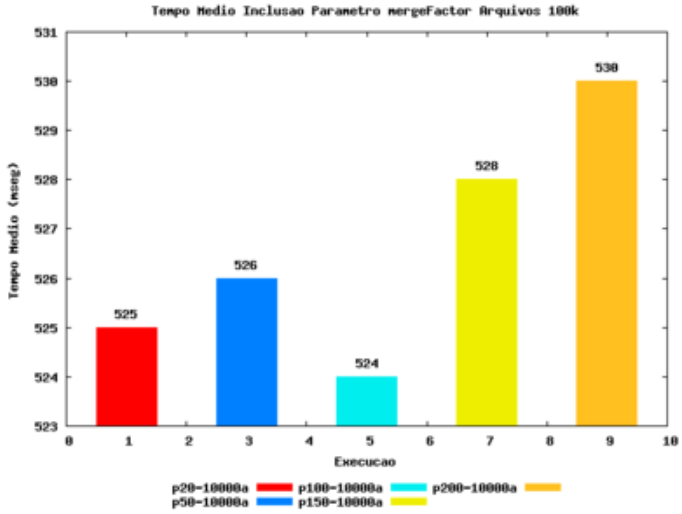

Figura 22: Tempo Médio de Inclusão mergeFactor 100KB 10000 arquivos

Tam/Param	20	50	100	150	200
100k	3	3	7	3	3
1M	9	9	10	10	9
10M	41	42	44	45	44

Tabela 9: Tempo de Inclusão de 1000 Arquivos (em min) – cacheSize

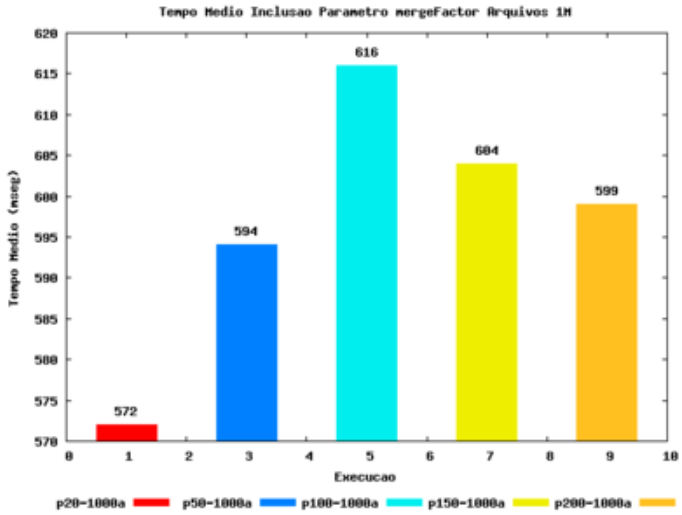

Figura 23: Tempo Médio de Inclusão mergeFactor 1MB 1000 arquivos

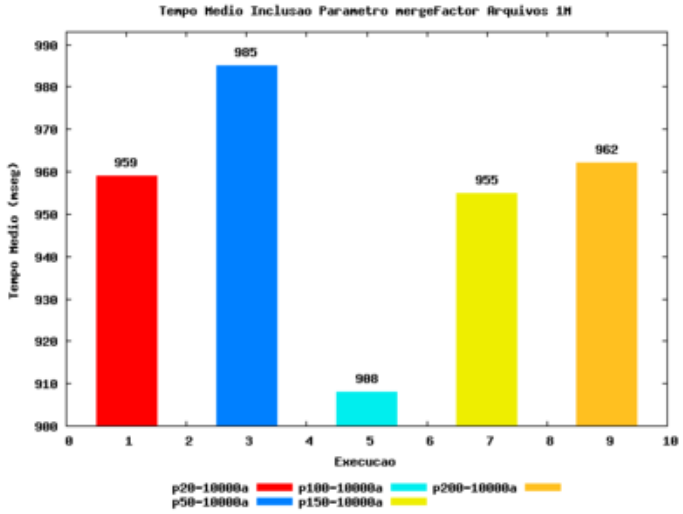

Figura 24: Tempo Médio de Inclusão mergeFactor 1MB 10000 arquivos

Tam/Param	20	50	100	150	200
100k	87	87	87	88	88
1M	159	164	151	159	160

Tabela 10: Tempo de Inclusão de 10000 Arquivos (em min) – cacheSize

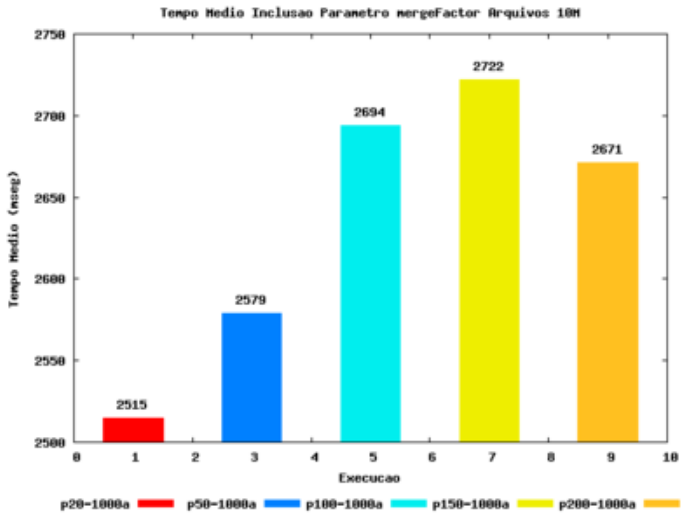

Figura 25: Tempo Médio de Inclusão mergeFactor 10MB 1000 arquivos

Observando as figuras 21, 23 e 25 é possível estabelecer a relação de que, quanto maior o tamanho do arquivo a ser incluído, um parâmetro menor obtêm um tempo menor de inclusão, sendo mais apropriado.

3.6 Influência do parâmetro minMergeDocs

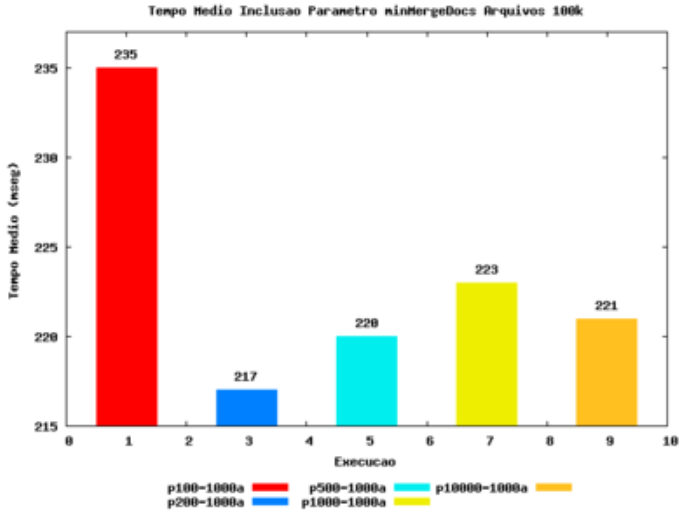

Figura 26: Tempo Médio de Inclusão minMergeDocs 100KB 1000 arquivos

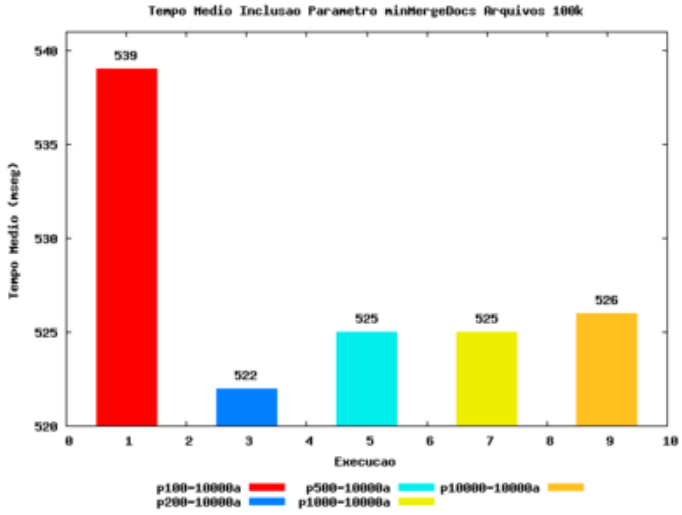

Figura 27: Tempo Médio de Inclusão minMergeDocs 100KB 10000 arquivos

Tam/Param	**100**	**200**	**500**	**1000**	**10000**
100k	3	3	3	3	3
1M	12	9	9	9	9
10M	45	43	43	42	43

Tabela 11: Tempo de Inclusão de 1000 Arquivos (em min) – cacheSize

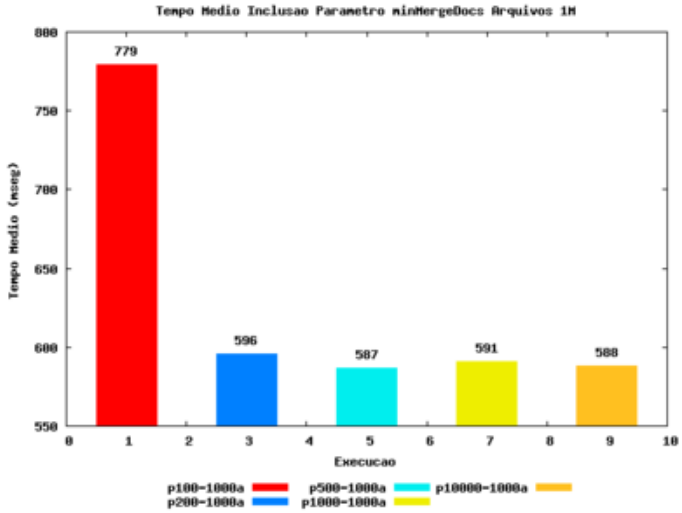

Figura 28: Tempo Médio de Inclusão minMergeDocs 1MB 1000 arquivos

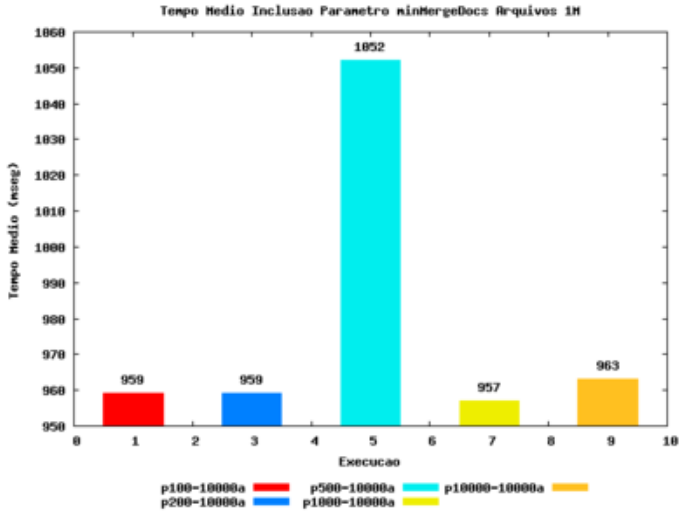

Figura 29: Tempo Médio de Inclusão minMergeDocs 1MB 10000 arquivos

Tam/Param	100	200	500	1000	10000
100k	89	87	87	87	87
1M	159	159	175	159	160

Tabela 12: Tempo de Inclusão de 10000 Arquivos (em min) – cacheSize

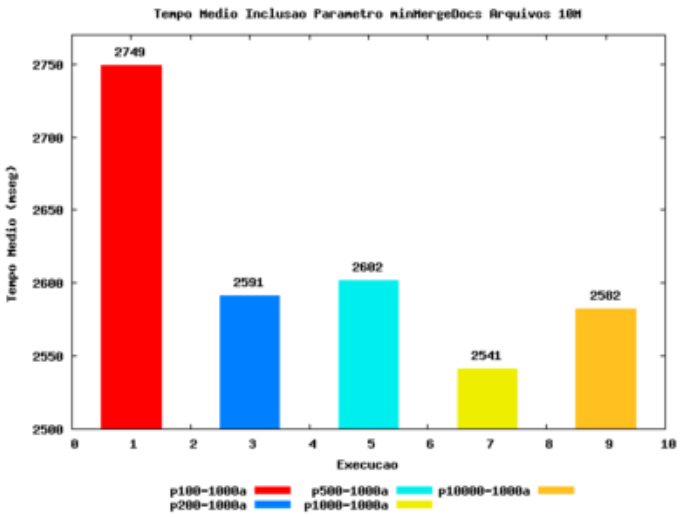

Figura 30: Tempo Médio de Inclusão minMergeDocs 10MB 1000 arquivos

Ao observar as figuras 26, 27, 28 e 30 conclui-se que independente do valor escolhido para minMergeDocs, com exceção do valor 100, que fornece o pior resultado, todos os outros valores testados não apresentam diferença significativa para tamanhos de arquivo menores que 10MBytes.

A figura 29 apresenta um padrão diferente das demais, porém, após a análise do ambiente, observa-se que justamente o parâmetro 500 para arquivos de tamanho 1MByte reflete o pior caso. Como *minMergeDocs* indica o número mínimo de arquivos a serem tratados para que o índice seja escrito e a máquina de testes possui aproximadamente 500MBytes de memória disponível para a execução deste teste, supõe-se que seja esse o caso onde é ocupada a maior quantidade de memória e o índice seja gravado nesse momento.

4 Análise de Resultados – Pesquisa de Arquivos

4.1 Influência do parâmetro bufferSize

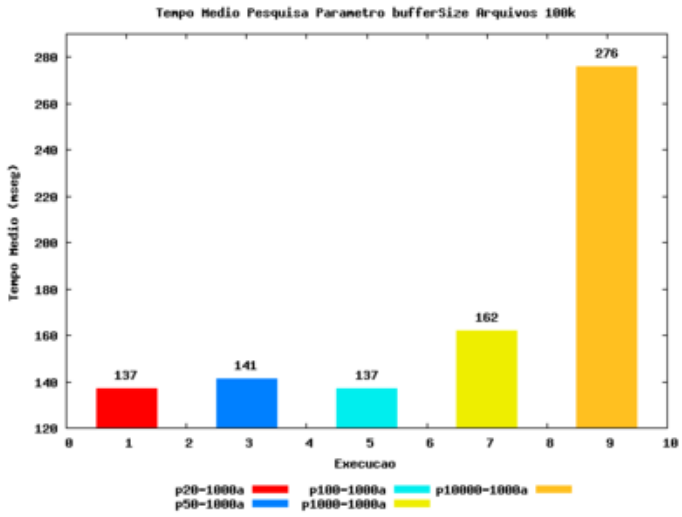

Figura 31: Tempo Médio de Pesquisa bufferSize 100KB 1000 arquivos

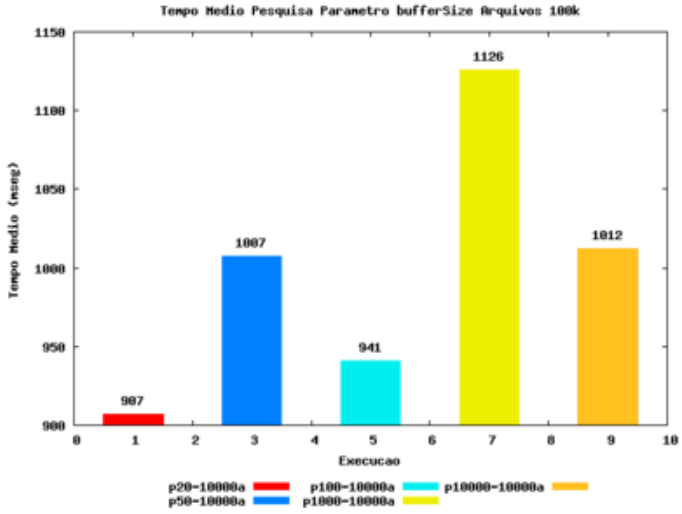

Figura 32: Tempo Médio de Pesquisa bufferSize 100KB 10000 arquivos

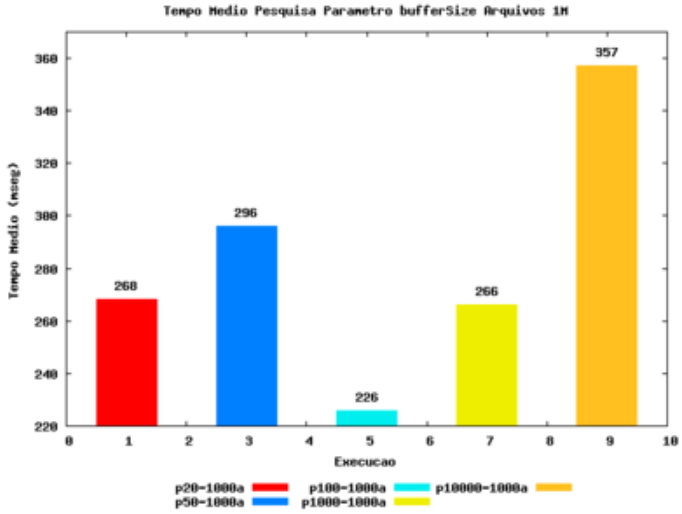

Figura 33: Tempo Médio de Pesquisa bufferSize 1MB 1000 arquivos

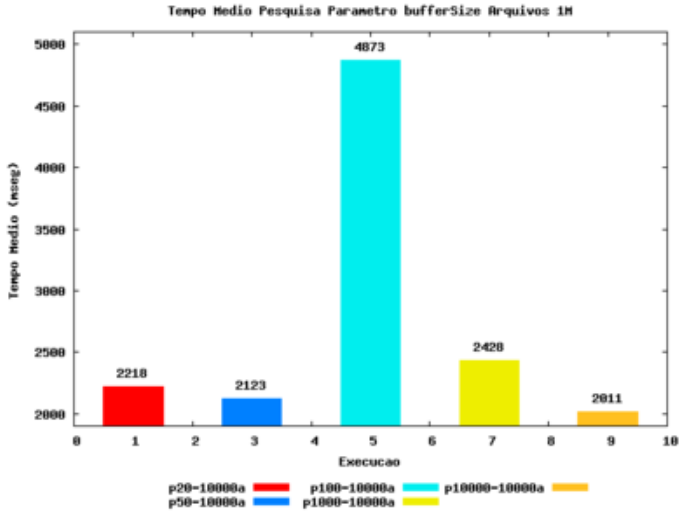

Figura 34: Tempo Médio de Pesquisa bufferSize 1MB 10000 arquivos

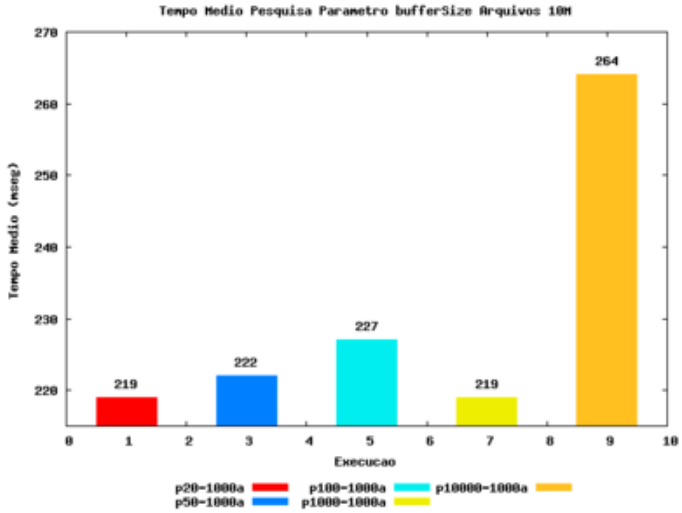

Figura 35: Tempo Médio de Pesquisa bufferSize 10MB 1000 arquivos

O tempo de pesquisa aumenta proporcionalmente ao número de arquivos incluídos no repositório. Por outro lado, o número de arquivos não influencia diretamente o tempo de busca quando mantêm-se o mesmo valor para o número de palavras indexadas (valor de *maxFieldLength*). Acredita-se que esse comportamento ocorra por causa do uso de memória cache para manter o conteúdo dos índices.

4.2 Influência do parâmetro cacheSize

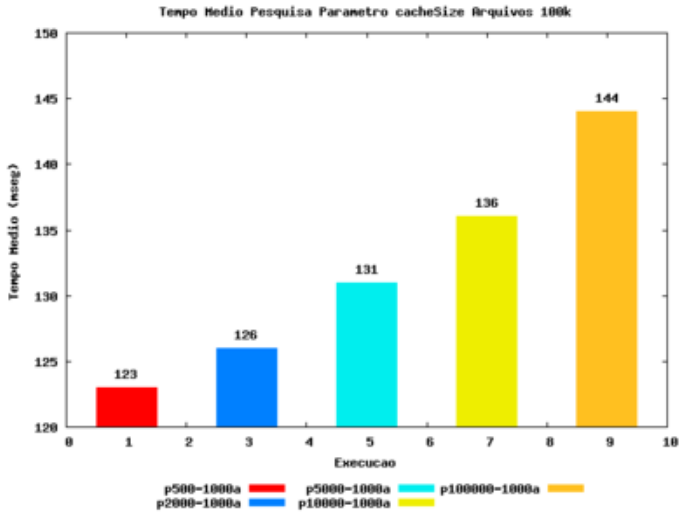

Figura 36: Tempo Médio de Pesquisa cacheSize 100KB 1000 arquivos

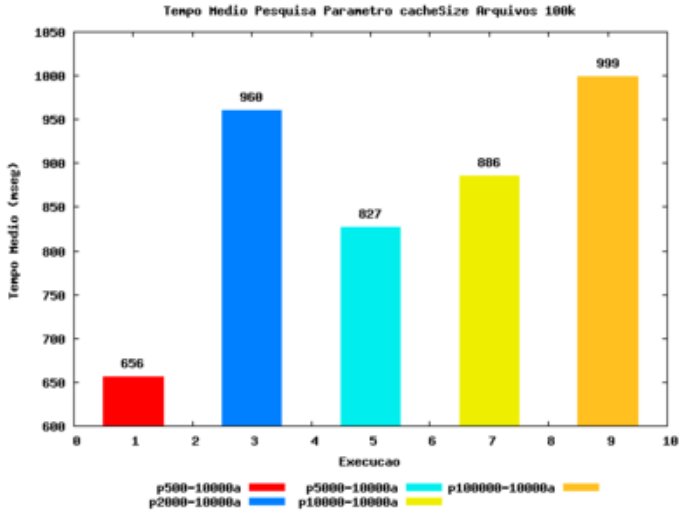

Figura 37: Tempo Médio de Pesquisa cacheSize 100KB 10000 arquivos

Observando-se a figura 36 é possível identificar o custo impingido pelo gerenciamento de uma memória cache maior para arquivos pequenos. Embora a diferença entre o menor valor (123) e o maior valor (144) seja pequena para um repositório pequeno, observa-se na figura 37 que essa diferença aumenta proporcionalmente ao número de arquivos dentro do repositório, mantendo um mesmo[1] padrão de crescimento.

1 Acredita-se que fatores externos tenham contribuído para o desvio de padrão do valor 2.000 na figura 37.

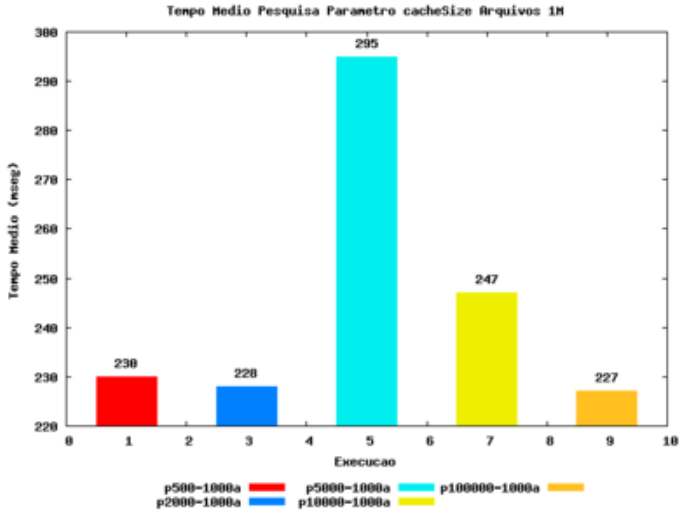

Figura 38: Tempo Médio de Pesquisa cacheSize 1MB 1000 arquivos

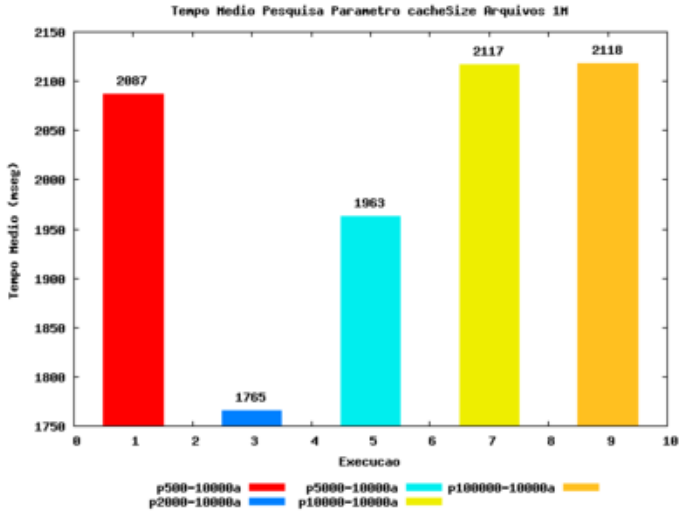

Figura 39: Tempo Médio de Pesquisa cacheSize 1MB 10000 arquivos

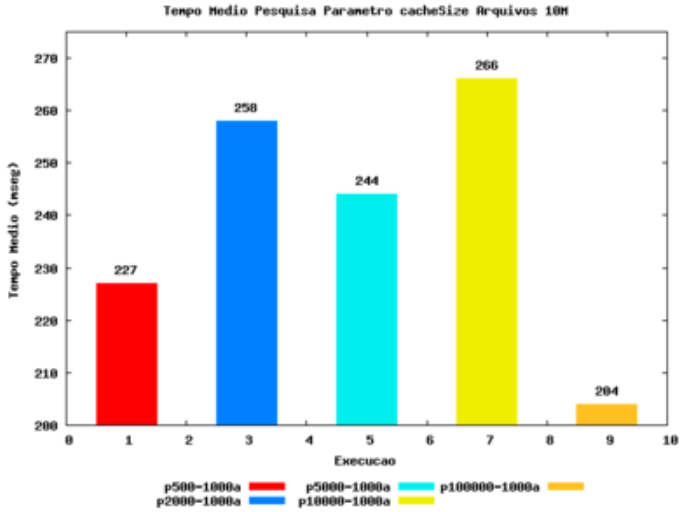

Figura 40: Tempo Médio de Pesquisa cacheSize 10MB 1000 arquivos

O tamanho do arquivo influencia diretamente o tamanho do parâmetro *cacheSize* que obtêm o melhor resultado segundo a relação: quanto maior o tamanho do arquivo, o maior valor do parâmetro obtêm o melhor resultado. A tabela 13 ilustra o melhor valor de *cacheSize* relacionado com o tamanho médio dos arquivos.

Tamanho	**Valor**
100KBytes	500
1MByte	2.000
10MBytes	10.000

Tabela 13: Comparação entre tamanho de arquivo e melhor valor para cacheSize

4.3 Influência do parâmetro extractorPoolSize

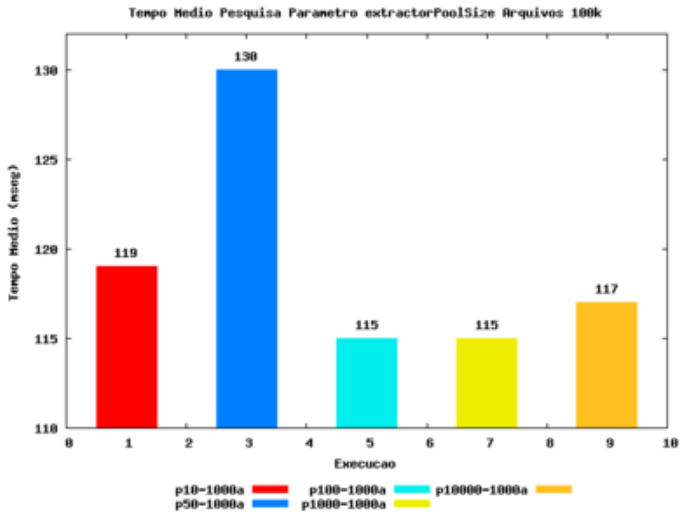

Figura 41: Tempo Médio de Pesquisa extractorPoolSize 100KB 1000 arquivos

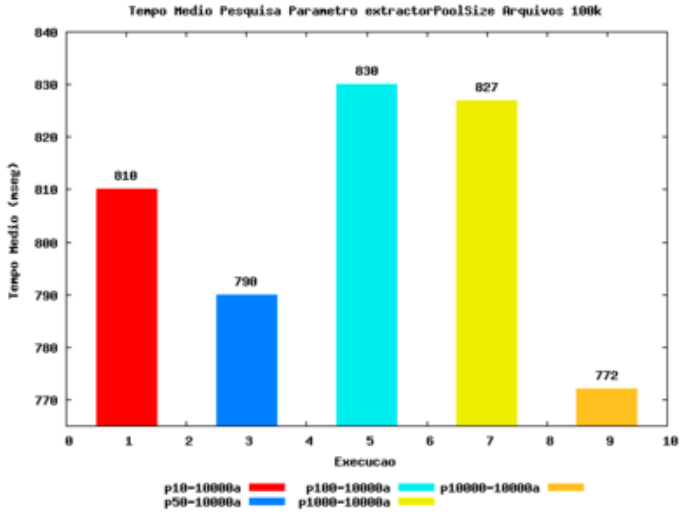

Figura 42: Tempo Médio de Pesquisa extractorPoolSize 100KB 10000 arquivos

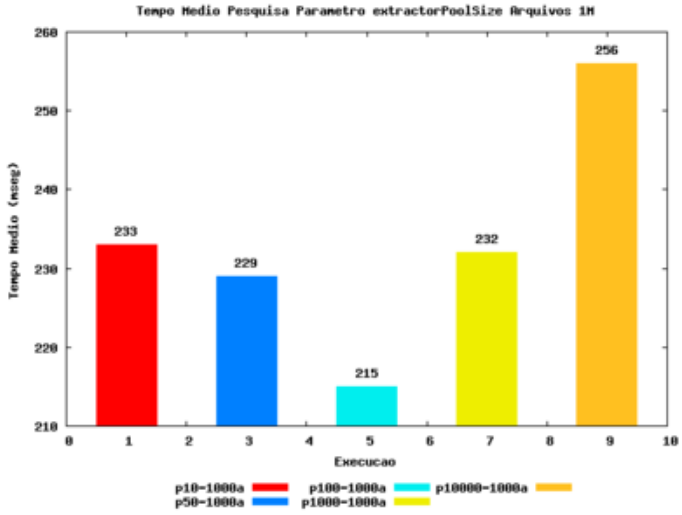

Figura 43: Tempo Médio de Pesquisa extractorPoolSize 1MB 1000 arquivos

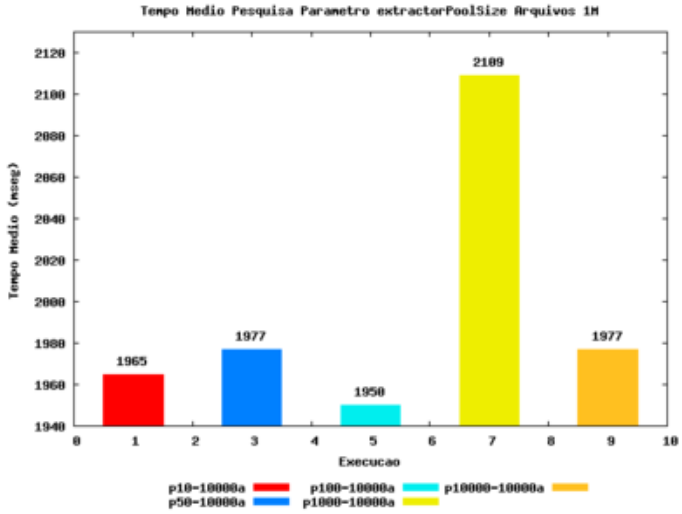

Figura 44: Tempo Médio de Pesquisa extractorPoolSize 1MB 10000 arquivos

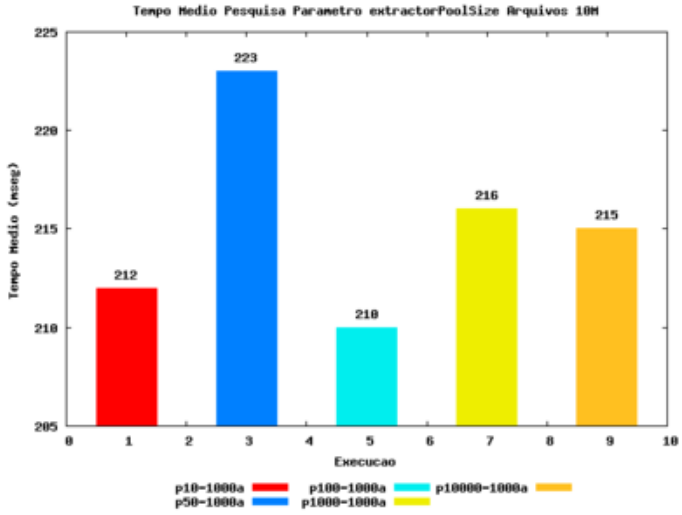

Figura 45: Tempo Médio de Pesquisa extractorPoolSize 10MB 1000 arquivos

Independente do número e tamanho dos arquivos pesquisados, o melhor valor para o parâmetro *extractorPoolSize* é 100. Analisando-se os parâmetros e ambiente observa-se uma relação (100 threads para ↔ 1.000 e 100 threads para ↔ 10.000 arquivos) entre o número de arquivos e o número de threads no pool, onde acredita-se que essa relação atinja sua melhor distribuição com o valor mencionado.

Acredita-se que eventos externos influenciaram o resultado obtido na figura 42. Independentemente, a diferença de tempo entre o melhor e o pior parâmetro é de apenas 58 millisegundos.

4.4 Influência do parâmetro maxFieldLength

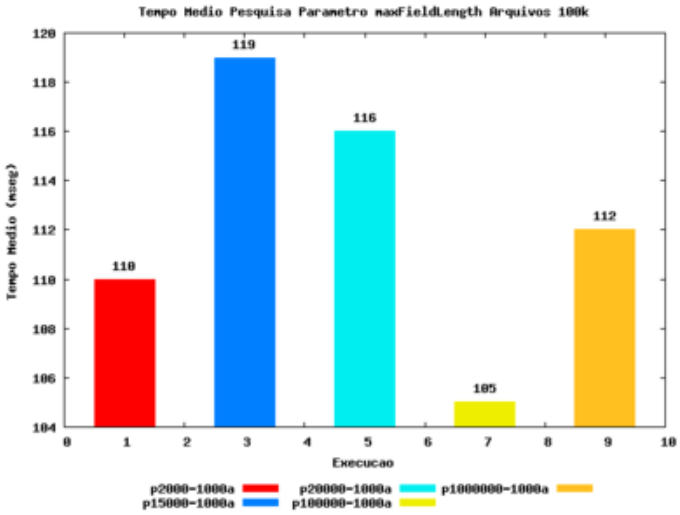

Figura 46: Tempo Médio de Pesquisa maxFieldLength 100KB 1000 arquivos

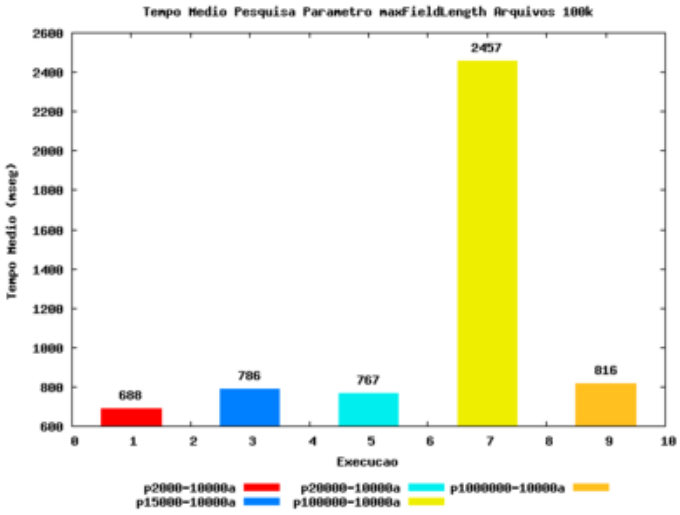

Figura 47: Tempo Médio de Pesquisa maxFieldLength 100KB 10000 arquivos

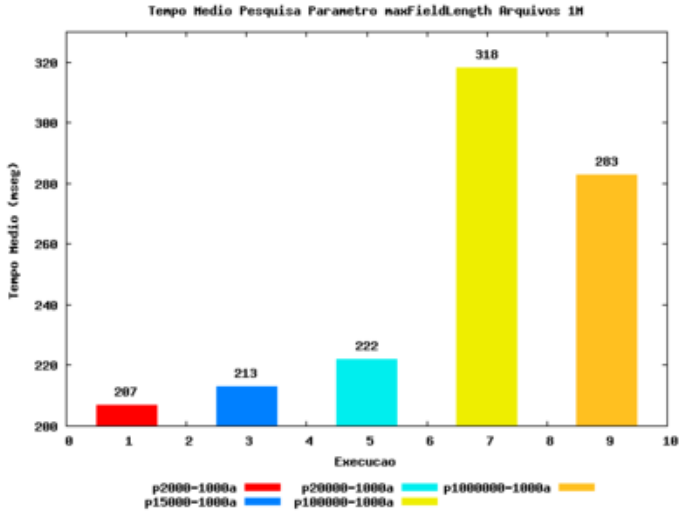

Figura 48: Tempo Médio de Pesquisa maxFieldLength 1MB 1000 arquivos

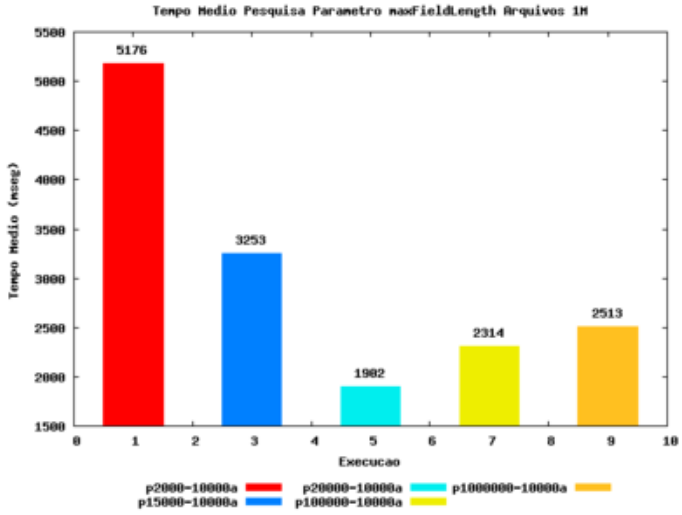

Figura 49: Tempo Médio de Pesquisa maxFieldLength 1MB 10000 arquivos

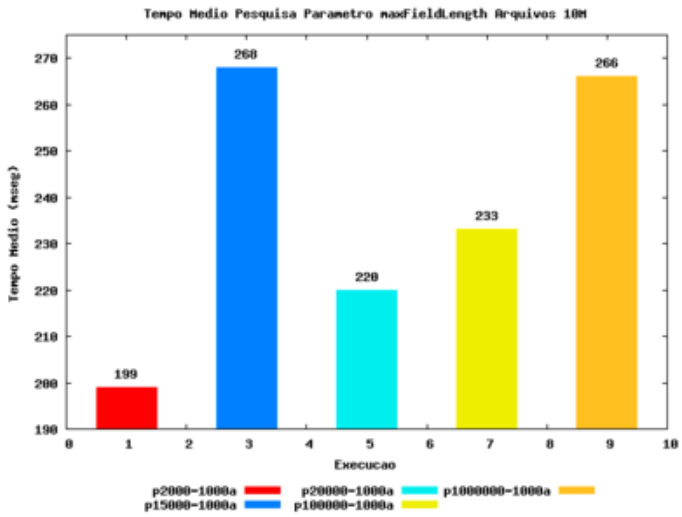

Figura 50: Tempo Médio de Pesquisa maxFieldLength 10MB 1000 arquivos

O parâmetro *maxFieldLength* estabelece a relação entre o tamanho do arquivo e os itens indexados na pesquisa. O preço a ser pago para arquivos de 100KB é de aproximadamente 7ms em média, aumentando para 134ms quando o tamanho do repositório aumenta. Esses valores foram obtidos pela diferença entre o menor tempo de resposta, que nesse caso não necessariamente quer dizer que encontrou o item, pelo tempo de execução com o maior valor de maxFieldLength, que indica que o maior número de itens dentro do documento serão indexados aumentando a chance de encontrar um termo pesquisado.

A tabela 14 ilustra as diferenças encontradas.

Tamanho / Número	1.000	10.000
100KBytes	7ms	134ms
1MByte	76 ms	611ms
10MBytes	67ms	n.d.

Tabela 14: Diferenca entre menor tempo de pesquisa e tamanho de palavras indexadas

4.5 Influência do parâmetro mergeFactor

Quanto menor o tamanho do fator de agregação, menor o tempo de resposta da pesquisa.

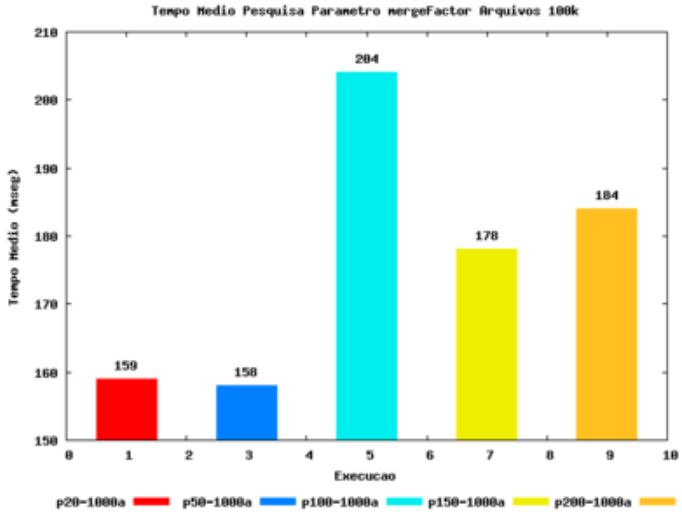

Figura 51: Tempo Médio de Pesquisa mergeFactor 100KB 1000 arquivos

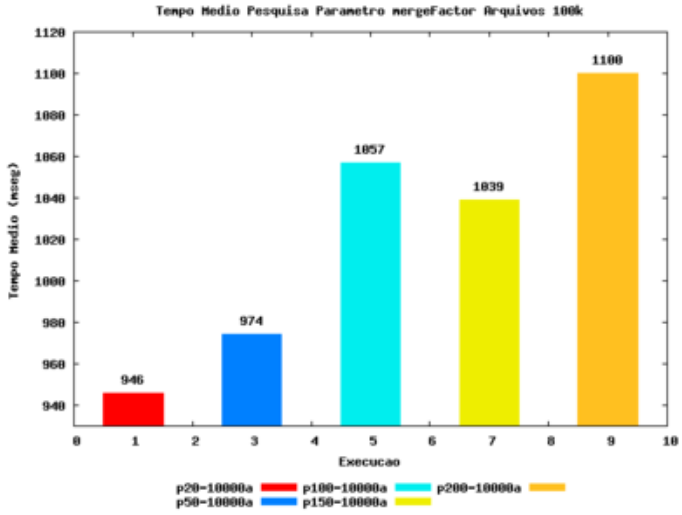

Figura 52: Tempo Médio de Pesquisa mergeFactor 100KB 10000 arquivos

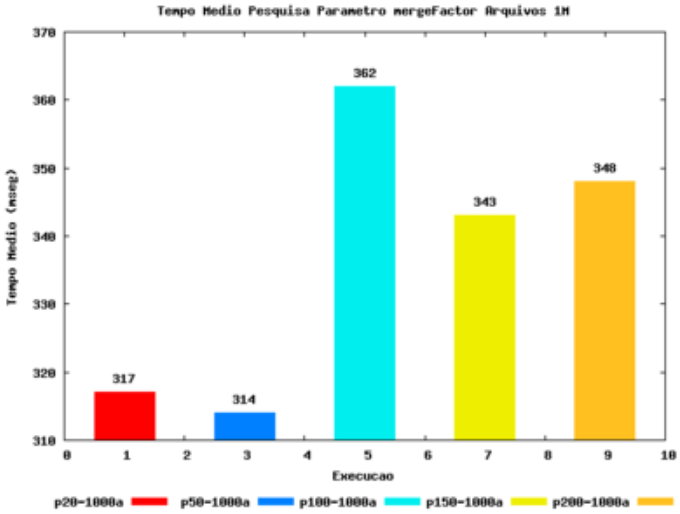

Figura 53: Tempo Médio de Pesquisa mergeFactor 1MB 1000 arquivos

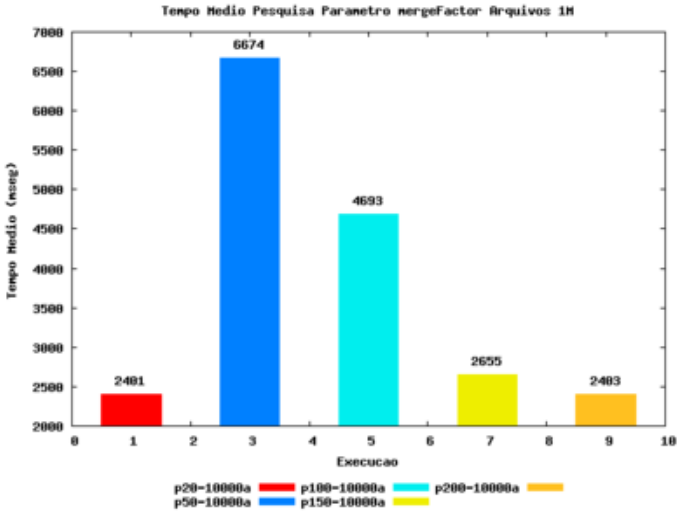

Figura 54: Tempo Médio de Pesquisa mergeFactor 1MB 10000 arquivos

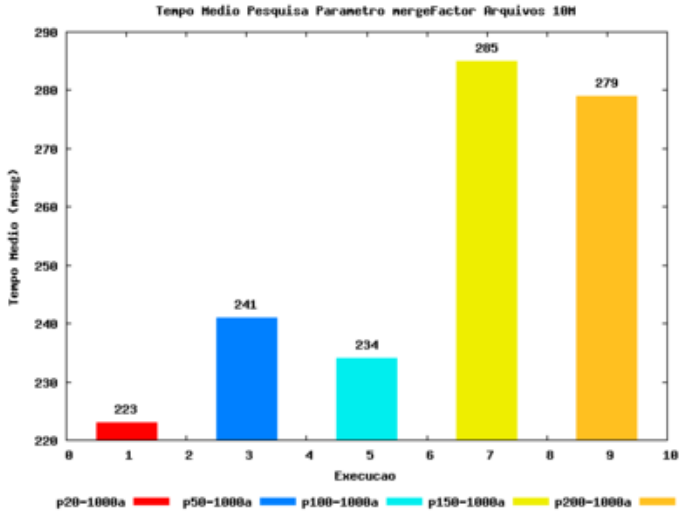

Figura 55: Tempo Médio de Pesquisa mergeFactor 10MB 1000 arquivos

4.6 Influência do parâmetro minMergeDocs

À medida em que o tamanho e número de arquivos crescem, o valor atribuído ao parâmetro deve ser aumentado proporcionalmente para obter-se melhor eficiência.

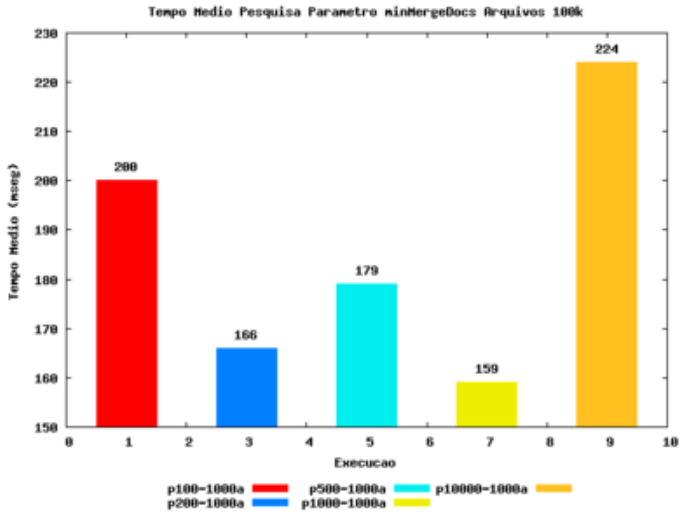

Figura 56: Tempo Médio de Pesquisa minMergeDocs 100KB 1000 arquivos

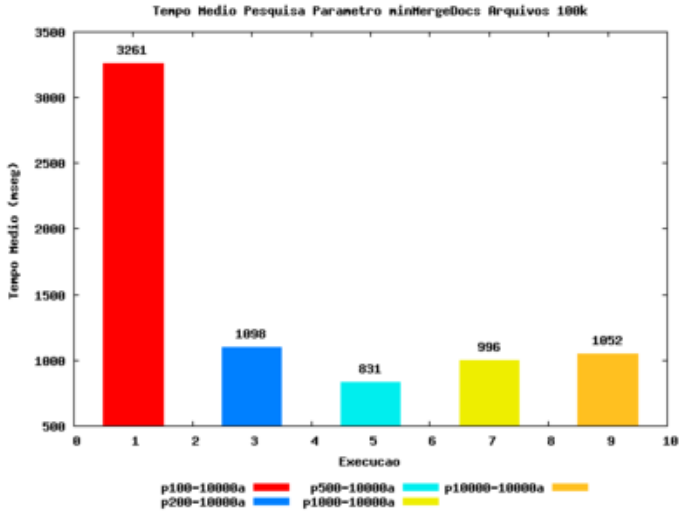

Figura 57: Tempo Médio de Pesquisa minMergeDocs 100KB 10000 arquivos

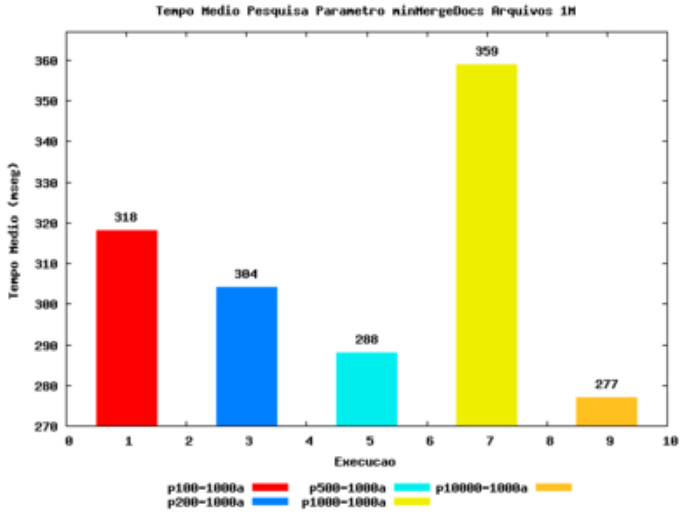

Figura 58: Tempo Médio de Pesquisa minMergeDocs 1MB 1000 arquivos

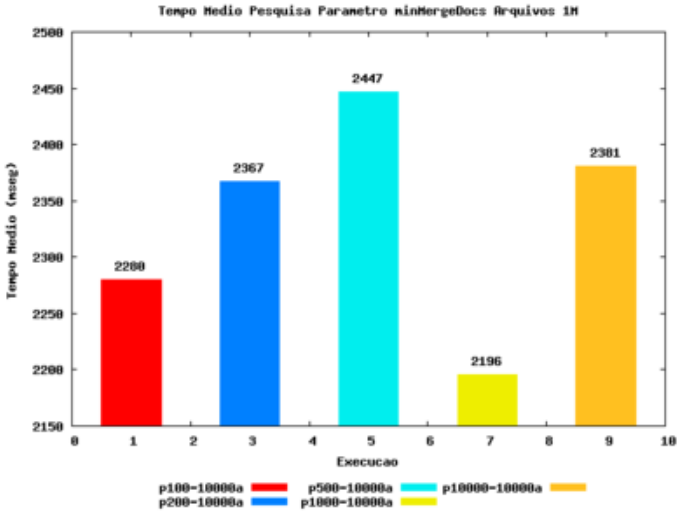

Figura 59: Tempo Médio de Pesquisa minMergeDocs 1MB 10000 arquivos

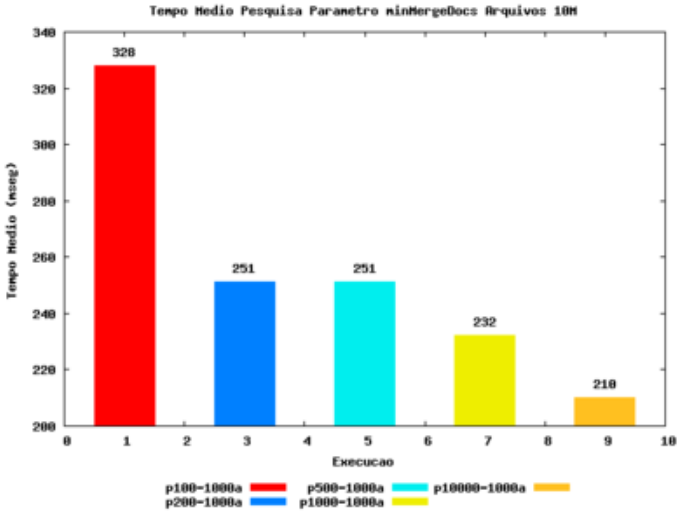

Figura 60: Tempo Médio de Pesquisa minMergeDocs 10MB 1000 arquivos

Não se chegou a uma relação a ser seguida para obter o melhor resultados para o parâmetro *minMergeDocs*. Observou-se, entretanto, que o tamanho do arquivo (100KB, 1MB, 10MB) tem pequena influência no tempo de resposta (200ms,318ms,320ms) do resultado final quando comparado com a quantidade de arquivos. Sugere-se refazer esse teste com uma configuração onde haja maior variação do número de arquivos incluídos para melhor análise do resultado.

5 Avaliação de Infra-Estrutura

5.1 Tamanhos de Repositório e Índices

A tabela 15 ilustra os resultados obtidos para diferentes tamanhos de arquivo e valores para o parâmetro *maxFieldLength*. O parâmetro *maxFieldLength* foi escolhido para este teste por influenciar o número de palavras que serão armazenadas nos índices de busca.

A influência de um parâmetro *maxFieldLength* maior se faz presente tanto no tamanho do repositório de BLOBs quanto na memória utilizada(vide tabela 17).

Tam./Núm. Arquivos	**1**	**10**	**100**	**1000**	*Param*
100KBytes	324K	1,4M	12M	115M	*2000*
100KBytes	340K	1,5M	13M	125M	*100000*
1MByte	1,3M	11M	103M	1017M	*2000*
1MByte	1,4M	12M	115M	1,2G	*100000*

Tabela 15: Tamanho de Arquivos versus Tamanho de BLOBs

Um aumento de 50 vezes no parâmetro causou um aumento de aproximadamente 20% no tamanho do armazenamento de BLOBs, dependendo do tamanho de arquivo utilizado. Esse comportamento era esperado, uma vez que para o maior valor (100.000) é atingido o final de arquivo antes de processar o número de *tokens* especificado.

A tabela 16 mostra o resultado do tamanho do banco de dados para o teste com os tamanhos de arquivo e valores diferentes para o parâmetro *maxFieldLength* do jackrabbit.

A mudança no valor do parâmetro *maxFieldLength* não influenciou o tamanho do banco de dados, como ocorreu com o tamanho do repositório de BLOBs.

Tam./Núm. Arquivos	**1**	**10**	**100**	**1000**	*Param*
100KBytes	76K	144K	816K	7,4M	*2000*
100KBytes	76K	144K	816K	7,4M	*1000000*
1MByte	76K	144K	816K	7,4M	*2000*
1MByte	76K	144K	816K	7,4M	*1000000*

Tabela 16: Tamanho de Arquivos versus Tamanho do Banco de Dados

Utilizando os valores presentes nas tabelas 15 e 16 é possível projetar aproximadamente a utilização de disco para um volume de arquivos.

5.2 Quantidade de Memória RAM Utilizada

Os valores a seguir foram obtidos executando-se um dump da memória utilizada pelo jboss na execução do teste. Os tamanhos referem-se exclusivamente ao processo do jboss. O momento “antes” foi definido como o ponto imediato antes da execução do teste de inserção de arquivos, enquanto que o momento “depois” refere-se ao tempo imediatamente posterior à execução do teste.

A tabela 17 ilustra a diferença da utilização total de memória quando utilizado o valor 1.000.000 e o valor 2.000. Essa tabela ilustra também a progressão de consumo de memória (em kilobytes) ao aumentar o número de inserções e o tamanho dos arquivos. É possível observar que a diferença entre a memória utilizada pode chegar a 38MBytes (100 arquivos,1MByte) dependendo da configuração utilizada.

Tam./Núm. Arquivos	**1**	**10**	**100**	**1.000**	**10.000**	*Param*
100KBytes	326 267	3890 1216	15785 10136	35225 64167	84592 49538	*1.000.000* *2.000*
Diferença(p 1M-2k)	**59**	**2.674**	**5.649**	**-28.902**	**35.054**	–
Tam./Núm. Arquivos	**1**	**10**	**100**	**1.000**	**10.000**	*Param*
1MByte	320 333	16321 1013	51931 13433	76848 63938	82325 43661	*1.000.000* *2.000*
Diferença(p 1M-2k)	**-13**	**15.308**	**38.498**	**12.910**	**38.664**	–

Tabela 17: Comparação entre diferença de parâmetro e Quantidade de Memória RAM utilizada - em KBytes

Utilizando-se esses valores é possível calcular aproximadamente o consumo de memória durante a inclusão de arquivos e operação do sistema Scriba. Observe que esses valores são *apenas para os dados utilizados*, descartada a memória utilizada pelo servidor de aplicação.

6 Resultados Obtidos

Os testes serviram para mostrar que a correta configuração dos parâmetros do jackrabbit influencia bastante no desempenho e resultados obtidos, tanto na inclusão de arquivos, quanto na busca dos mesmos e nos resultados de busca.

Para maximizar os resultados obtidos na pesquisa, é necessário utilizar o maior valor possível para o parâmetro *maxFieldLength*, o que implica em um tempo de resposta mais lento na inclusão e um pequeno atraso na pesquisa, porém agregando mais informações no resultado da pesquisa. Um parâmetro *maxFieldLength* grande reduz grandemente as possibilidades de o usuário dizer: “Não Encontrei o texto X que está no arquivo Y quando pesquisei no Documentador”.

Quanto maior o tamanho da memória disponível para o servidor de aplicação menor o tempo de resposta obtido nas pesquisas para valores de *cacheSize*.

7 Conclusão

O planejamento do repositório de dados requer uma estimativa prévia da quantidade *inicial* de documentos a serem incluídos no repositório[1].

Antes porém de estabelecer o tamanho médio dos arquivos é importante definir qual abordagem será utilizada na indexação de termos dos documentos:

indexação máxima
: onde atribui-se um maior valor para o número de palavras a serem indexadas (*maxFieldLength*), resultando em uma busca mais ampla nos documentos armazenados, porém com um tempo de busca maior.

indexação intermediária
: onde atribui-se um valor arbitrário para o número de palavras a serem indexadas, onde o tempo de resposta à pesquisa tende a ser mais rápido, porém com partes dos documentos inseridos não indexadas. O revés dessa abordagem é que o usuário pode procurar por um termo que esteja em um documento armazenado e não encontrá-lo.

Após definida essa abordagem, estima-se o tamanho e quantidade média dos arquivos para atribuir-se um valor inicial para as configurações. Obtendo-se um valor médio aproximado para o tamanho e quantidade dos arquivos, é possível, extrapolando os dados apresentados nesse documento, formular a melhor relação para os parâmetros de configuração do *jackrabbit* e assim otimizar o tempo de inserção de arquivos em lote.

A seção 5 fornece dados iniciais para o planejamento da ocupação de espaço em disco nos servidores.

Para repositórios já implantados, uma maneira de melhorar a performance é estabelecer um novo valor para os parâmetros do *jackrabbit* e proceder a re-indexação do repositório. Como observado nos testes, com o crescimento do número de arquivos alguns parâmetros, tais como *bufferSize*, *cacheSize*, *extractorPoolSize* e *maxFieldLength*, têm seus valores ótimos deslocados para mais ou para menos dependendo do parâmetro.

7.1 Configurações Recomendadas

A tabela 18 apresenta uma estimativa de espaço ocupado médio para o repositório e tamanho aproximado de arquivo.

1 Essa estimativa é necessária apenas para agilizar a inclusão de grandes volumes de dados.

Número Arquivos	Tamanho	BLOBs	Índice	TOTAL
1.000	100 KB	130 MB	10 MB	≈ **150 MB**
	1 MB	1.5 GB	15 MB	≈ **1.7 GB**
10.000	100 KB	1.6 GB	71 MB	≈ **1.8 GB**
	1 MB	16 GB	72 MB	≈ **18 GB**

Tabela 18: Estimativa de Tamanho Necessário

A seguir listamos as configurações recomendadas para uma máquina com as características apresentadas na seção 2.

7.1.1 Repositório com Tamanho Médio de Arquivos 100-300KBytes

Com base nos dados presentes no repositório do Documentador, sugere-se a mudança de configuração para os parâmetros do jackrabbit apresentados a seguir.

Parâmetro	**Valor Sugerido**
bufferSize	100
cacheSize	100.000
extractorPoolSize	1.000
maxFieldLength	2.147.483.647
mergeFactor	150
minMergeDocs	200

Tabela 19: Configuração Sugerida para Valores do Jackrabbit – Arquivos de Tamanho Médio 100KB

7.1.2 Repositório com Tamanho Médio de Arquivos 1-3MBytes

Parâmetro	**Valor Sugerido**
bufferSize	50
cacheSize	2.000
extractorPoolSize	50
maxFieldLength	2.147.483.647
mergeFactor	100
minMergeDocs	1.000

Tabela 20: Configuração Sugerida para Valores do Jackrabbit – Arquivos de Tamanho Médio 1MB

7.1.3 Repositório com Tamanho Médio de Arquivos 10MBytes

Parâmetro	**Valor Sugerido**
bufferSize	50
cacheSize	100.000
extractorPoolSize	10.000
maxFieldLength	2.147.483.647
mergeFactor	20
minMergeDocs	1.000

Tabela 21: Configuração Sugerida para Valores do Jackrabbit – Arquivos de Tamanho Médio 10MB

7.1.4 Tempo Utilizado – Configurações Recomendadas

A tabela 22 ilustra o tempo utilizado para a inclusão de arquivos em lote no scriba para tamanhos de arquivo de 100KB, 1MB e 10MB para a máquina mencionada na seção 2. Comparam-se os tempos de inclusão para o parâmetro maxFieldLength com valor *1.000.000* e *2.147.483.647* (que é o maior valor possível para esse parâmetro).

NumArq/maxField Length	**1.000.000**	**2.147.483.647**	**Tamanho**
1.000	5 38 45	5 39 55	100KB 1MB 10MB
10.000	89 199	88 230	100KB 1MB

Tabela 22: Tempo de Inclusão Arquivos (em min) – configuração recomendada

7.1.5 Memória RAM utilizada para Configurações Recomendadas

A quantidade de memória expressa na tabela 23 indica apenas a quantidade de memória necessária para os dados, sem contar a memória utilizada para o servidor de aplicação jboss e para a aplicação scriba.

Tam./Núm. Arquivos	**1.000**	**10.000**	
100KBytes	63.282	95.231	
1MBytes	37.702	142.894	
10MBytes	125.805	–	

Tabela 23: Quantidade de memória RAM utilizada (em KBytes) para as configurações recomendadas, com valor de parâmetro maxFieldLength igual 2.147.483.647

7.2 Perguntas Frequentes

1. **Qual infraestrutura deverá ser tratada com a GTI para atender determinada demanda ao utilizar o Documentador ?**

   R:

   A infraestrutura irá depender da demanda específica. Os resultados apresentados neste documento ajudam o analista a estabelecer uma previsão de espaço ocupado e memória utilizada no servidor.

2. **Existe um controle de performance?**

   R:

   Os controles de performance são os seguintes:

   * Ambiente

     A. Quantidade de Memória Disponível para o servidor JBoss

     B. Quantidade de Memoria Disponível para a aplicação Scriba

     C. Número de Conexões ao Banco de Dados

   * Aplicação

     D. Melhoria nas Consultas executadas pela aplicação.

     E. Configurações do Jackrabbit apresentadas neste documento(seção 7.1)

3. **Vou poder garantir ao cliente que ele terá um bom tempo de resposta?**

   R:

   O acesso do cliente ao documentador vai depender de uma série de fatores que a CELEPAR não tem como controlar, por exemplo, largura de banda até a máquina cliente.

   O que pode ser garantido é que, o acesso no servidor e aplicação será rápido e eficiente, porém não se pode garantir (sem um contrato com SLA com a operadora de telefonia) qualquer tipo de velocidade de acesso ao cliente.

   O tempo de resposta que podemos informar é o tempo de resposta da busca no servidor.

4. **Posso ter certeza de que ao informar determinada palavra, o sistema de busca me trará todos os resultados possíveis e sem falhas?**

   R:

   É possível ter uma grande probabilidade de encontrar a palavra, mas não é possível garantir que o resultado da busca seja bem sucedido em 100% das vezes. O resultado da busca depende dos termos indexados e do algoritmo utilizado para indexação desses termos quando o documento é incluído no scriba, algoritmo esse

que varia de acordo com o tipo de arquivo utilizado.

O que podemos fazer (e estamos fazendo) é aumentar a quantidade de elementos que são elegíveis para indexação e busca, de modo a reduzir ao mínimo a possibilidade de uma busca ser mal-sucedida.

5. **Quando publico, X documentos automaticamente, qual o meu tempo de resposta?**

   R:
   O tempo de inclusão pode ser estimado estrapolando-se os números fornecidos neste documento (seção 3) para a realidade do cliente. Esses números irão variar dependendo da massa de dados (número e tamanho de documentos) a serem inseridos.

6. **Qual infraestrutura preciso ter para ter uma resposta X (rápida) durante a busca?**

   R:
   Em primeiro lugar é preciso determinar o que é uma resposta rápida. A seção 4 ilustra os resultados de busca para os diferentes parâmetros de configuração do jackrabbit. O tempo de resposta é composto de vários fatores, dentre eles:

   * tempo de comunicação do cliente com o servidor.
   * tempo de comunicação do servidor scriba com o banco de dados.
   * tempo de busca do banco de dados.
   * tempo de retorno da busca para o servidor scriba.
   * tempo de recuperação do arquivo no disco no servidor scriba (no caso de ter um BLOB associado).
   * tempo de retorno da resposta do servidor ao cliente.

   Vários fatores interferem no tempo de resposta do servidor, sendo que a quantidade de memória disponível para a aplicação (Scriba) e a velocidade de acesso ao banco de dados (número de conexos no *pool*) são fatores determinantes.

7. **Qual a melhor infraestrutura para cargas de Documentos um a um e com bom tempo de resposta na recuperação da busca?**

   R:
   A infraestrutura vai variar de acordo com o número de usuários acessando simultâneamente.

8. **Qual a melhor infraestrutura para carga de documentos em massa e com bom tempo de resposta, tanto na carga quanto na busca?**

   R:
   Um fator importante é a quantidade de memória disponível para o servidor de aplicação.

9. **Qual a melhor infraestrutura para ter uma boa performance geral no uso do**

Documentador?

R:

São vários os itens a serem seguidos:

i. Verificar o número de usuários simultâneos e aumentar o número de conexões no *pool* de acesso ao banco de dados.

ii. Seguir a orientação para configuração do jackrabbit recomendada neste documento na seção 7.1 de acordo com o número e tamanho de arquivos no repositório.