opentext™

OpenText™ eDirectory CE 24.4 (v9.3)

Guia de Ajuste

Janeiro de 2025

Informações Legais

Copyright 2009 – 2024 Open Text Todos os direitos reservados.

As únicas garantias para os produtos e serviços da Open Text e de suas afiliadas e licenciantes (“Open Text”) são as que possam estar descritas nas declarações de garantia expressa que acompanham esses produtos e serviços. Nenhuma informação nos termos deste documento deve ser interpretada como garantia adicional. A Open Text não será responsável por erros técnicos ou editoriais e nem por omissões contidos neste documento. As informações constantes neste documento estão sujeitas à mudança sem aviso prévio.

Índice

Sobre este manual e a biblioteca **5**

1 Visão geral **7**

Pré-requisitos 7

2 Subsistemas do OpenText eDirectory **9**

Banco de dados do FLAIM 9
Ponto de verificação 10
Índices 10
Registro de transações individuais 11
Containerização do atributo FLAIM 11
Pool de thread 12

3 Analisando gargalos do sistema **15**

Subsistema de E/S de disco 15
Subsistema da CPU 16
Subsistema de memória 16
Subsistema de rede 17

4 Ajuste de subsistemas do OpenText eDirectory **19**

Banco de dados do FLAIM 19
Escolhendo índices 20
Ajustes de atualizações 20
Pool de thread 21
ACLs 21
Melhorando pesquisas e leituras do OpenText eDirectory 21
Desativando modelos ACL 22
Replicação 24
SSD (Solid State Disk - Unidade de Estado Sólido) 25
Intervalo de atualização de login do NMAS 26
Overhead do SSL 26
ICE (Import Convert and Export) 26
ldif2dib 26
Tamanho do pacote de NCP melhorado 26

5 Configuração do OpenText eDirectory **29**

Configurando o subsistema FLAIM 29
Limite de cache físico 29
Ajustando o limite dinamicamente 29
Modificando as configurações de cache de FLAIM 29
Modificando as configurações de cache de FLAIM pelo OpenText iMonitor 30

Modificando as configurações de cache do FLAIM pelo _ndsdb.ini 31

Sobre este manual e a biblioteca

Este documento descreve como analisar e ajustar o produto OpenText™ eDirectory CE 24.4 (v9.3) para proporcionar um desempenho superior em todas as suas implantações.

Para obter a versão mais recente do *Guia de Ajuste do OpenText™ eDirectory*, consulte o site da Documentação Online do OpenText eDirectory (https://www.netiq.com/documentation/edirectory-93/).

Público-alvo

Ele é dirigido aos administradores de rede.

Outras informações na biblioteca

A biblioteca fornece os seguintes recursos informativos:

Guia de Administração do XDASv2

Descreve como configurar e usar o XDASv2 para auditar o OpenText eDirectory e o OpenText Identity Manager.

Guia de Instalação

Descreve como instalar o OpenText eDirectory. Destina-se a administradores de rede.

Guia de administração

Descreve como gerenciar e configurar o OpenText eDirectory.

Guia de Solução de Problemas

Descreve como resolver problemas do OpenText eDirectory.

Guia de novidades

Descreve os novos recursos do OpenText eDirectory.

Esses guias estão disponíveis no site na web de documentação do OpenText eDirectory (https://www.netiq.com/documentation/edirectory-93/).

Para obter informações sobre o utilitário de gerenciamento do eDirectory, consulte o Guia de Administração do OpenText Identity Console (https://www.netiq.com/documentation/identity-console/).

1 Visão geral

O OpenText eDirectory 9.3 é uma solução de serviços de diretório de alto desempenho, altamente escalável, tolerante a falhas, compatível com várias plataformas e em conformidade com as normas. Este guia fornece informações sobre como ajustar o ambiente do OpenText eDirectory para obter um desempenho aprimorado.

Ajustar o desempenho é uma atividade complexa. Ela exige compreensão do OpenText eDirectory e dos subsistemas do sistema operacional. Ela envolve também monitorar o sistema para identificar gargalos e corrigi-los um por vez. Muitas vezes os recursos são limitados e o ajuste fica restrito ao OpenText eDirectory e ao sistema operacional.

Neste guia, leia a seção Pré-requisitos antes de tentar realizar qualquer tipo de ajuste e, então, prossiga para as outras seções. O capítulo Subsistemas do OpenText eDirectory descreve os subsistemas primários que influenciam o desempenho do OpenText eDirectory. O capítulo Analisando gargalos do sistema descreve diversos recursos do sistema e sua influência sobre o desempenho do OpenText eDirectory. O capítulo Ajuste de subsistemas do OpenText eDirectory descreve como analisar e ajustar o OpenText eDirectory em diversas condições e ambientes. Por fim, o capítulo Configuração do OpenText eDirectory descreve como configurar os diversos parâmetros ajustáveis.

Pré-requisitos

Certifique-se de que os pré-requisitos gerais a seguir sejam atendidos antes de tentar ajustar o desempenho do sistema:

- Um bom design de árvore do OpenText eDirectory pode aperfeiçoar o desempenho do programa. As considerações a seguir podem ser aplicáveis:
  - Os aplicativos leem todas as informações localmente no servidor sem precisar encadear as solicitações.
  - O OpenText eDirectory lida automaticamente com as referências de objetos de maneira eficiente. Se possível, os objetos de um servidor não devem fazer referência a objetos não localizados neste servidor, pois isso pode exigir mais tempo para manter as referências de objetos não locais. Se tal referência existir, os backlinks deverão ser mantidos. Isso pesará em grandes implantações.
  - Se você precisar de um grupo com 10 mil membros ou mais, será recomendado usar grupos dinâmicos. Isso permite evitar o overhead associado à manutenção de referências para tantas pessoas. Escolha suas configurações de grupo dinâmico com cuidado, pois usar vários grupos dinâmicos com critérios de pesquisa inadequados poderá sobrecarregar o servidor e reduzir seu desempenho geral. Se uma operação de pesquisa levar muito tempo para ser concluída, o índice escolhido poderá ser ineficaz. Minimize o uso de grupos regulares (estáticos), pois isso pode aumentar o andamento das árvores ao efetuar login.

- Use ACLs (Access Control List - Lista de Controles de Acesso) com eficiência. Por exemplo, use [Este] trustee e atribua-o no nível de container em vez de usar um modelo de ACL que atribui direitos a si mesmo. Quanto menor o número de ACLs, melhor o desempenho. Para obter mais informações sobre ACLs, consulte "Direitos do OpenText eDirectory" no *Guia de Administração do OpenText™ eDirectory CE 24.4 (v9.3)*.
- Distribua a carga em diversos servidores de réplicas.
- Embora um bom design de árvore minimize a necessidade de andamento de árvore, algumas vezes, isso ainda pode ser necessário. Você pode considerar o "Custo de referência avançada" no *Guia de Administração do OpenText™ eDirectory CE 24.4 (v9.3)*.
- Se os logins estiverem lentos, você poderá desabilitar as atualizações de login. Existem maneiras separadas de desabilitar atualizações de login para logins NDS (Novell Directory Services) e do NMAS (NetIQ Modular Authentication Service — Serviço de Autenticação Modular do NetIQ). Contudo, é importante compreender as implicações de segurança (http://www.novell.com/documentation/nmas33/admin/data/bg8dphs.html).

- Execute verificações de funcionamento por meio do iMonitor. Para obter mais informações, consulte "Exibindo a saúde do servidor do OpenText eDirectory" no *Guia de Administração do OpenText™ eDirectory CE 24.4 (v9.3)*. Verifique o seguinte:
  - O tempo está em sintonia em todos os servidores de réplicas.
  - A sincronização de réplica e os processos em segundo plano estão funcionando bem.

2 Subsistemas do OpenText eDirectory

Esta seção aborda os subsistemas do OpenText eDirectory.

- “Banco de dados do FLAIM” na página 9
- “Pool de thread” na página 12

Banco de dados do FLAIM

O OpenText eDirectory usa o FLAIM como seu banco de dados. O FLAIM (Flexible Adaptable Information Manager — Gerente de Informações Adaptável e Flexível) é usado para informações tradicionais, voláteis e complexas. Ele é um mecanismo de banco de dados altamente escalável que suporta múltiplos leitores e um modelo de apenas um gravador simultâneo. Leitores não bloqueiam gravadores e gravadores não bloqueiam leitores.

Fisicamente, o FLAIM organiza os dados em blocos. Alguns blocos são tipicamente mantidos na memória. Eles representam o cache do bloco. O cache de entrada (por vezes chamado de cache de registro) retém as entradas lógicas do banco de dados. As entradas são construídas de itens no cache do bloco. O FLAIM mantém tabelas de hash para ambos os caches. O tamanho do compartimento de memória de hash é ajustado periodicamente com base no número de itens.

Por padrão, o OpenText eDirectory usa um tamanho de bloco de 4 KB. O tamanho do cache do bloco para armazenar o DIB total é igual ao tamanho do DIB, e o tamanho necessário para o cache de entrada é cerca de duas a quatro vezes o tamanho do DIB.

Ao obter uma entrada, o FLAIM verifica primeiramente a entrada no cache de entrada. Se ela existir nele, não será necessário ler o cache do bloco. Ao obter um bloco do disco, o FLAIM verifica primeiramente os blocos no cache. Se o bloco existir, não será necessário realizar uma operação de leitura de disco.

Quando uma entrada é adicionada ou modificada, os blocos correspondentes desta entrada não estão diretamente comprometidos com o disco, por isso o disco e a memória podem não estar em sintonia. Contudo, as atualizações na entrada são registradas no registro de transações individuais (RFL). Um RFL é usado para recuperar transações após uma falha do sistema.

Usado mais recentemente (LRU) é o algoritmo usado para substituir itens no cache.

- “Ponto de verificação” na página 10
- “Índices” na página 10
- “Registro de transações individuais” na página 11
- “Containerização do atributo FLAIM” na página 11

Ponto de verificação

Um ponto de verificação traz a versão presente no disco do banco de dados para o mesmo estado coerente que o banco de dados na memória (armazenado). O FLAIM pode realizar um ponto de verificação durante a atividade de atualização mínima no banco de dados. Ele é executado a cada segundo e grava os blocos modificados (cache modificado) no disco. Os blocos que são modificados no cache, porém ainda não são escritos no disco, são chamados de "blocos modificados". O FLAIM bloqueia o banco de dados e realiza o máximo de trabalho possível até que o ponto de verificação seja concluído ou outro thread esteja aguardando para atualizar o banco de dados. Para evitar que os bancos de dados no disco fiquem muito fora de sincronia, existem condições sob as quais é forçada a realização de um ponto de verificação mesmo se existirem threads aguardando para atualizar o banco de dados:

- Se um thread de ponto de verificação não puder concluir um ponto de verificação dentro do intervalo de tempo especificado (o padrão é três minutos), ele será forçado e o cache modificado será apagado.
- Se o tamanho do cache modificado for maior do que `maxdirtycache` (se definido), um ponto de verificação será forçado a reduzir o tamanho do cache modificado para o definido em `mindirtycache` (se definido) ou para zero.

Índices

Um índice é um conjunto de chaves organizado de maneira a acelerar significativamente a tarefa de encontrar qualquer chave específica dentro do índice. As chaves de índice são formadas ao extrair o conteúdo de um ou mais campos (atributos) das entradas. Os índices são armazenados no cache do bloco. Qualquer mudança nos atributos indexados exigirá mudanças nos blocos do índice.

O OpenText eDirectory define um conjunto padrão de índices para atributos do sistema (campos). Atributos do sistema, como `parentID` e `ancestorID`, são usados em pesquisas de um nível e de subárvore. Esses índices não podem ser suspensos ou apagados. O diretório os utiliza internamente. Os índices padrão são definidos para atributos, como `CN`, `Sobrenome` e `Nome`, entre outros. Índices podem ser do tipo de presença, valor e subcadeia. Esses índices podem ser suspensos. Ao serem excluídos, eles são automaticamente recriados.

Você pode usar o OpenText Identity Console ou o ndsindex do utilitário LDAP (Lightweight Directory Access Protocol) para criar índices. Índices (http://www.netiq.com/documentation/edir83/) são específicos de cada servidor.

Ao habilitar a tag do Gerenciador de armazenamento (`StrMan`) no DSTrace (ndstrace), você poderá ver o índice escolhido nas pesquisas.

O exemplo a seguir mostra um registro de DSTrace para uma pesquisa de subárvore usando `"cn=admin"`, `CN`.

```
3019918240 StrMan: Iter #b239c18 query ((Flags&1)==1) &&
((CN$217A$.Flags&8=="admin") && (AncestorID==32821))

3019918240 StrMan: Iter #b239c18 index = CN$IX$220
```

O exemplo a seguir mostra um registro de DSTrace para uma pesquisa de subárvore usando `"Description= This is for testing"`, `AncestorID`.

```
2902035360 StrMan: Iter #83075b0 query ((Flags&1)==1) &&
((Description$225A$.Flags&8=="This is for testing") &&
(AncestorID==32821))

2902035360 StrMan: Iter #83075b0 index = AncestorID_IX
```

Para melhorar o desempenho da pesquisa com a classificação do lado do servidor, use a opção -a para prefixar o atributo `AncestorID` na lista de atributos passada ao se criar um novo índice.

A tabela abaixo explica a tendência de aumentar o tamanho do DIB ao criar índices simples/compostos para um determinado número de objetos. Recomendamos que você consulte estes dados de teste ao criar índices simples/compostos:

Número de Objetos	Aumento no tamanho do DIB com índice simples contendo tamanho de atributo de 20 caracteres	Aumento no tamanho do DIB com índice simples contendo tamanho de atributo de 200 caracteres	Aumento no tamanho do DIB com índice composto contendo tamanho de atributo de 200 caracteres
23 mil	24 KB	3 MB	7 MB
47 mil	50 KB	6 MB	15 MB
0.5 milhão	4 MB	55 MB	100 MB
1 milhão	9 MB	111 MB	210 MB

Registro de transações individuais

O FLAIM registra operações para cada transação de atualização em um arquivo de registro de transações individuais (RFL). Um RFL é usado para recuperar transações de uma falha do sistema ou ao fazer restauração de um backup. O arquivo RFL é truncado após cada ponto de verificação ser concluído, exceto se ativado em (`rflkeepfiles`) usando um backup dinâmico contínuo (http://www.netiq.com/documentation/edir93/).

Containerização do atributo FLAIM

Para garantir uma utilização eficiente do cache de entrada e obter um desempenho melhorado das operações de pesquisa de atributo, o FLAIM armazena atributos com valores maiores ou com um número maior de valores em uma localização separada, ou seja, o Container de atributos. Por padrão, os atributos serão movidos automaticamente para o container quando o atributo:

- tiver mais de 25 valores
- tiver um valor maior do que 2048 bytes

Para desabilitar a containerização automática de atributos, adicione `disablemovetoattrcontainer=1` no arquivo `_ndsdb.ini` e reinicie o OpenText eDirectory.

O OpenText eDirectory proporciona a flexibilidade de programação da transferência do atributo. Primeiro você deve ver os atributos que estão prontos para ser transferidos e, em seguida, programar as respectivas transferências de acordo com sua conveniência.

Para ver o número de atributos que estão prontos para transferência para o Container de atributos, execute o comando `ndscheck`. Para ver os detalhes dos atributos, use o atributo **dsContainerReadyAttrs** do iMonitor nos objetos servidores Pseudo na **Configuração de agente**. O usuário também pode encontrar esses atributos, que estão marcados para indexação na saúde do agente.

Para iniciar a containerização de atributos, use a opção de conserto de objeto único de `ndsrepair` para o objeto Pseudo servidor. Para containerizar um atributo, emita o comando `ndsrepair` com o novo switch avançado `-am` seguido do nome do atributo, conforme mostrado abaixo:

```
ndsrepair –J <Pseudo server object ID> –Ad –AM/–am <attribute name>
```

Após transferir um atributo para o Contêiner de Atributos, o OpenText eDirectory cria um índice de sistema com o nome do atributo. Após a containerização de um atributo, não é possível transferi-lo de volta para o container original.

Observação: Se um atributo tiver um valor maior do que 2.048 bytes, a containerização ainda acontecerá, mas o OpenText eDirectory não criará nenhum índice do sistema.

Pool de thread

O OpenText eDirectory tem múltiplos threads por questões de desempenho. Em multi-threading, quando o sistema está ocupado, mais threads são criados para lidar com a carga e alguns são encerrados para evitar overhead extra. Criar e destruir threads frequentemente é custoso e ineficaz. Em vez de criar novos threads e destruí-los para cada tarefa, certos threads são iniciados e colocados em um pool. O sistema aloca os threads do pool de threads para diversas tarefas conforme necessário. As tarefas são separadas em dois tipos de filas:

- Tarefas que precisam de programação imediata são mantidas na fila Prontas.
- Tarefas que precisam de programação posterior são mantidas na fila Em espera.

Nem todos os módulos usam pool de threads. O número real de threads do processo é maior do que o número existente no pool de threads. Por exemplo, o FLAIM gerencia seus threads em segundo plano separadamente.

Executar o comando `ndstrace -c threads` exibe as seguintes estatísticas do pool de thread:

- O número total de threads que são criados, encerrados e estão inativos.
- O número total de threads de trabalho atuais e o número máximo de threads de trabalho.
- O número de tarefas e número máximo de tarefas na fila Prontas.
- O número mínimo, máximo e médio de microssegundos transcorridos na fila Prontas.
- O número atual e máximo de tarefas na fila Em espera.

Um exemplo de uma amostra do pool de thread:

```
Thread Pool Information

Summary      : Spawned 42, Died 5
Pool Workers : Idle 8, Total 37, Peak 37
Ready Work   : Current 0, Peak 10, maxWait 67436 us
Sched delay  : Min 14 us, Max 1052004 us, Avg: 792 us
Waiting Work : Current 17, Peak 21
```

Existem certos parâmetros do pool de thread:

- **n4u.server.max-threads:** O número máximo de threads que podem ficar disponíveis no pool.
- **n4u.server.idle-threads:** O número máximo de threads inativos que podem ficar disponíveis no pool.
- **n4u.server.start-threads:** Número de threads iniciados.

Execute os comandos `ndsconfig get` e `ndsconfig set` para obter e definir o tamanho do pool de thread.

3 Analisando gargalos do sistema

Existem diversos recursos do sistema que influenciam o desempenho do OpenText eDirectory. Além disso, atualizar para a versão mais recente do sistema operacional melhora o desempenho.

- "Subsistema de E/S de disco" na página 15
- "Subsistema da CPU" na página 16
- "Subsistema de memória" na página 16
- "Subsistema de rede" na página 17

Subsistema de E/S de disco

O subsistema de disco é o gargalo mais comum. A E/S leva relativamente muito tempo com filas mais longas, resultando em alta utilização de disco e ciclos de CPU inativos. Use a ferramenta iostat durante os picos esperados para determinar os indicadores de tempo de resposta média.

As operações de leitura, gravação e atualização de disco podem ser sequenciais ou aleatórias. Leituras e atualizações aleatórias são o padrão de acesso mais comum em implantações do OpenText eDirectory.

Algumas soluções para cargas de trabalho aleatórias:

- Aumente a memória RAM. Isso permite armazenar em cache os dados frequentemente usados ou os dados de leitura antecipada na camada do sistema de arquivos. Ela também permite armazenar em cache o DIB dentro do subsistema do FLAIM.
- Use volumes dedicados para o DIB. O desempenho do sistema de arquivos melhora para volumes criados mais próximos ao fuso. Use volumes dedicados para o RFL e outros registros.
- À medida que o disco desenvolve latência crescente com o passar do tempo devido à fragmentação, ele deverá ser desfragmentado.
- Adicione unidades de disco separadas para RFL do FLAIM. Esse tipo de registro pode ser realizado em discos de alta velocidade.
- Use um ambiente RAID 10(1+0) com mais unidades de disco.

Os arquivos criados pelo OpenText eDirectory podem ser expandidos até 4 GB. Os sistemas de arquivos otimizados para lidar com grandes arquivos trabalham bem com o OpenText eDirectory.

- Para Solaris™, o sistema de arquivos Veritas* VxFS é um sistema de arquivos baseado em extensão no qual os metadados do sistema de arquivos são otimizados para grandes arquivos. O sistema de arquivos UFS é indiretamente baseado em blocos, no qual os metadados do sistema de arquivos são armazenados em um grande número de blocos. Ele pode até mesmo ser dividido para grandes arquivos, o que torna o UFS mais lento para grandes arquivos.
- Para Linux™, o sistema de arquivos Reiser é um sistema JFS rápido que trabalha melhor que o sistema de arquivos ext3 em grandes conjuntos de DIB. Contudo, o modo de registro em diário de gravação inversa do ext3 é compatível com o desempenho do sistema de arquivos Reiser, embora o modo padrão ofereça maior consistência de dados. O XFS é um sistema JFS de alto

desempenho, capaz de trabalhar com grandes arquivos e oferecer transferências de dados suaves. O OpenText eDirectory 9.3 é compatível com SLES 15 32 e com as plataformas de 64 bits que têm o sistema de arquivos XFS.

O FLAIM suporta um tamanho de bloco de 4 KB a 8 KB. O padrão é 4 KB. Esse é o mesmo tamanho de bloco padrão do Linux (`tune2fs -l device`). Contudo, no Solaris, o sistema de arquivos UFS é criado com um tamanho de bloco padrão de 8 KB (`df -g mountpoint`). Se o tamanho de bloco do FLAIM for menor do que o do sistema de arquivos, poderá ocorrer gravação parcial de bloco. Se o tamanho de bloco do banco de dados for maior do que o tamanho de bloco do sistema de arquivos, leituras e gravações individuais de blocos serão divididas em uma série de operações de E/S físicas separadas. Portanto, é necessário sempre manter o tamanho de bloco do FLAIM igual ao do sistema de arquivos.

O tamanho de bloco pode ser controlado apenas durante a criação do DIB. Acrescente uma linha "blocksize=8192" ao `_ndsdb.ini` para criar o DIB com tamanho de bloco igual a 8k.

Escolher o tamanho correto de bloco depende do tamanho médio do registro do FLAIM das suas implantações. É necessário realizar testes empíricos no conjunto correto de dados de teste para determinar qual tamanho de bloco é o melhor para sua implantação.

Subsistema da CPU

O OpenText eDirectory foi criado em uma arquitetura altamente escalável. O desempenho aumenta de acordo com o aumento do número de processadores. Um maior throughput é observado até que, pelo menos, o 12º processador esteja sob carga pesada. Contudo, esse aumento está sujeito ao desempenho de outros recursos durante o aumento da carga do sistema. Os servidores geralmente possuem configuração inferior com relação a discos e memórias. Você deverá adicionar mais processadores apenas nas seguintes circunstâncias:

- Se a carga média dos processadores usados anteriormente estiver acima de 75% de utilização. Se a utilização atual da CPU estiver abaixo de 75%, adicionar mais CPU poderá não melhorar o desempenho.
- Se houver um aumento satisfatório no desempenho.

Se o OpenText eDirectory estiver configurado com muitos threads, uma quantidade considerável de tempo da CPU será gasta em alternância de contexto. Neste caso, uma diminuição nos threads pode gerar um melhor throughput.

Subsistema de memória

Os aplicativos do servidor podem ser executados significativamente melhor com o aumento da RAM. Armazenar em cache o banco de dados do OpenText eDirectory no sistema de arquivos ou no cache do FLAIM pode levar a um melhor desempenho nas operações de pesquisa e modificação. Contudo, não será possível armazenar em cache o DIB em sua totalidade em grandes implantações. Evite passar as páginas mesmo se isso significar reduzir os tamanhos de entrada de FLAIM e cache do bloco. Use a ferramenta vmstat para encontrar mais informações sobre o subsistema de memória.

À medida que o OpenText eDirectory usa a memória, cada thread do pool de threads usa 1 MB de RAM para sua pilha. Por padrão, o tamanho de cache FLAIM é definido como 200 MB.

Diversos módulos carregáveis são iniciados quando o OpenText eDirectory é iniciado, mas a arquitetura do módulo carregável do OpenText eDirectory permite reduzir o tamanho da memória no processo ao deixar de carregar os módulos não usados (como SecretStore, LDAP ou eMBox). Além disso, produtos como o IDM contam com alguns módulos que são executados dentro do OpenText eDirectory.

A memória usada pelo OpenText eDirectory pode parecer crescer. Embora a memória seja liberada por um processo do OpenText eDirectory, ela não poderá ser liberada para o pool livre do sistema, pois o gerenciador de memória usado internamente pelo OpenText eDirectory tenta otimizar as alocações de memória para o futuro. Esta é uma das razões pelas quais não recomendamos a configuração dinâmica do FLAIM. Use a ferramenta Top para encontrar o tamanho de memória virtual aproximado do processo ndsd em sua implantação.

A memória máxima que pode ser alocada em um processo é limitada de diversas maneiras. Certa quantia de RAM é usada pelo sistema operacional e outros processos do sistema. O sistema operacional pode impor limitações sobre quanto de RAM física um processo pode usar.

Subsistema de rede

Implantações típicas possuem largura de banda suficiente para lidar com picos de carga de rede. Uma largura de banda adequada reduz erros, colisões e pacotes perdidos. Use a ferramenta netstat para determinar as estatísticas de rede.

Diversos sistemas operacionais oferecem parâmetros TCP/IP ajustáveis para ajustar servidores intensivos de rede. Para obter mais informações, consulte a documentação dos sistemas operacionais.

Se a rede é o gargalo, é necessário aumentar a largura de banda. Configurar uma rede privada dedicada entre os servidores do aplicativo e o do OpenText eDirectory também pode ajudar a reduzir o congestionamento na rede.

4 Ajuste de subsistemas do OpenText eDirectory

Esta seção inclui as seguintes informações:

- “Banco de dados do FLAIM” na página 19
- “Pool de thread” na página 21
- “ACLs” na página 21
- “Replicação” na página 24
- “SSD (Solid State Disk - Unidade de Estado Sólido)” na página 25
- “Intervalo de atualização de login do NMAS” na página 26
- “Overhead do SSL” na página 26
- “ICE (Import Convert and Export)” na página 26
- “ldif2dib” na página 26
- “Tamanho do pacote de NCP melhorado” na página 26

Banco de dados do FLAIM

O tamanho do cache é possivelmente o fator mais importante que afeta o desempenho geral do OpenText eDirectory. Quanto maior o número de itens (blocos e entradas) que são armazenados, melhor o desempenho geral. A porcentagem de vezes que os blocos e entradas são achados no cache é chamada de razão de acerto. Uma razão maior resulta em melhor desempenho. O OpenText iMonitor também pode ser usado para ver a razão de acerto.

O cache do bloco é mais útil para operações de atualização. O cache de entrada é mais útil para operações que realizam uma pesquisa de uma entrada com escopo. Contudo, ambas as pesquisas com escopo de um nível e subárvore usam o cache de entrada e o do bloco. O cache do bloco é usado para recuperar índices. Crie o tipo correto de índices conforme necessário; para obter mais informações, consulte “Escolhendo índices” na página 20.

Uma falha no cache do bloco pode resultar em uma operação de leitura de disco. As leituras de disco sempre são dispendiosas, porém podem ser evitadas se um bloco for recuperado no cache do sistema de arquivos.

A quantidade de memória necessária para armazenar o banco de dados completo no cache do bloco é quase o tamanho do banco de dados no disco, e o tamanho de memória necessário para armazenar o banco de dados completo no cache de entrada é quase de duas a quatro vezes o tamanho do banco de dados no disco. Se você possui pouca memória em um sistema, tente obter um cache de entrada menor e um cache de bloco ou de sistema de arquivos muito maior.

Se as leituras estiverem localizadas em um conjunto de entradas do diretório, você deverá aumentar o cache de entrada, desde que isso resulte em um aumento da razão de acertos do cache de entrada.

Se o padrão de leitura for totalmente aleatório e o DIB for muito maior do que a RAM disponível, você deverá possuir um cache do bloco ou de sistema de arquivos maior do que o de entrada.

Qualquer método usado para ajustar o OpenText eDirectory para obter melhor desempenho demandará testes empíricos. Uma boa razão de cache de entrada versus bloco para ambientes sensíveis a pesquisas é 2:1. Certifique-se de que deixa memória livre suficiente para outros processos. Evite passar as páginas mesmo se isso significar reduzir os tamanhos dos caches de entrada e bloco do FLAIM.

Como o FLAIM fornece armazenamento pré-alocado, a memória alocada no cache do OpenText eDirectory nunca é fragmentada pelo gerenciador de memória nativo do sistema operacional.

Escolhendo índices

Os índices têm por objetivo melhorar o desempenho de pesquisas com escopo de um nível ou de subárvore. Grupos dinâmicos também usam pesquisas com escopo de um nível ou de subárvore. Os índices não são usados com pesquisas com escopo de base.

Como o índice de Presença não diferencia entre os valores presente e ausente (apagado), ele é usado principalmente para fins internos. Se os aplicativos executarem uma consulta de pesquisa do tipo Presença, esse índice nunca será usado, por isso, os índices de Presença não deverão ser criados para os aplicativos.

Os aplicativos podem criar um índice Valor para um atributo, o que é suficiente para a maioria das pesquisas. O FLAIM pode usar um índice Valor para realizar ambas as pesquisas de Presença e de Subcadeia nos atributos.

Um índice Subcadeia pode desacelerar significativamente as atualizações realizadas em um atributo. O número de blocos de índice necessários para dar suporte a um índice Subcadeia é muito grande se comparado ao índice Valor. Isso significa que é necessário ter mais cache do bloco para armazená-los. Crie um índice de Subcadeia apenas quando necessário. Um índice Valor deverá ser suficiente para a maioria das pesquisas. Contudo, se as pesquisas de Subcadeia não gerarem resultados de desempenho aceitáveis com o índice Valor, você poderá criar um índice Subcadeia nesses atributos.

Se uma operação de pesquisa levar muito tempo para ser concluída, independentemente do índice escolhido, você poderá introduzir um novo índice de valor em um dos atributos do filtro de pesquisa. Escolha o atributo que gera os melhores resultados quando indexado.

Ajustes de atualizações

O cache do bloco é mais útil para operações de atualização. Os índices também residem no cache do bloco. Embora os índices ajudem em pesquisas mais rápidas, ter muitos índices requer que o servidor fique ocupado para mantê-los. Os índices são modificados se os valores do atributo forem modificados, adicionados ou apagados. Durante grandes operações de upload, os índices podem ser desabilitados para um upload mais rápido.

Manter o diretório RFL em um disco diferente que o diretório do DIB melhora o desempenho.

Um limite aceitável do tempo de resposta de uma operação de atualização pode ser controlado usando `maxdirtycache`. Por exemplo, se um limite aceitável de resposta do servidor for 5 segundos e a velocidade de gravação de disco aleatória for 20 MB por segundo, então, o

`maxdirtycache` deverá ser definido como 20x5 = 100 MB. Certifique-se que o cache do bloco possa conter esses blocos modificados na memória. Consulte o "Modificando as configurações de cache do FLAIM pelo _ndsdb.ini" na página 31 para obter mais informações.

Pool de thread

Por padrão, o número máximo de threads que podem estar disponíveis no pool de thread é 256. Esse número deverá ser suficiente para a maioria das implantações. Ele pode ser aumentado para 512 threads em grandes implantações. Você deverá aumentar o número de threads no pool nos seguintes casos:

- Se o número de threads inativos for frequentemente zero.
- Se a quantidade média de tempo gasto pela tarefa na fila Prontas for alta e crescente.
- Se o número de tarefas na fila Prontas for alto e crescente.

Continue aumentando o número máximo de threads se o desempenho do servidor aumentar. Isso também deverá resultar em uma maior utilização de CPU.

Para obter mais informações sobre como visualizar estatísticas do pool de threads, consulte "Visualizando estatísticas dos pools de threads" no *Guia de Administração do OpenText™ eDirectory CE 24.4 (v9.3)*.

ACLs

- "Melhorando pesquisas e leituras do OpenText eDirectory" na página 21
- "Desativando modelos ACL" na página 22

Melhorando pesquisas e leituras do OpenText eDirectory

Uma pesquisa LDAP no OpenText eDirectory exibe resultados dependendo do número de atributos exibidos para um usuário (`inetOrgPerson`).

Quando um objeto é criado no OpenText eDirectory, os ACLS padrão podem ser adicionados ao objeto. Isso depende dos modelos de ACL na definição do esquema para o objectClass à qual o objeto pertence. Por exemplo, na configuração padrão do `inetOrgPerson`, podem existir até seis ACLs adicionados ao objeto Usuário. Quando a solicitação de pesquisa de LDAP é feita de maneira a exibir esse objeto Usuário com todos os atributos, demora um pouco mais para exibir esse objeto ao cliente do que exibir o objeto Usuário sem os atributos ACL.

Embora os ACLs padrão possam ser desativados, os administradores podem não querem fazê-lo, pois eles são necessários para um melhor controle de acesso. Contudo, é possível melhorar o desempenho de pesquisas ao não solicitá-los ou torná-los atributos filtrados como leitura. Essas mudanças não impactam os aplicativos, pois a maioria deles usa privilégios efetivos e não conta com ACLs específicos.

Não solicitação de ACLs: Um atributo ACL não é necessário para muitos aplicativos, por isso eles podem ser modificados para solicitar os atributos específicos pelos quais se interessam. Isso resulta em um melhor desempenho nas pesquisas LDAP.

Marcar um ACL como filtrado para leitura: Se um aplicativo não pode ser modificado, o `arf_acl.ldif` pode ser usado pelo administrador para marcar o atributo ACL como atributo filtrado para leitura. Quando o ACL é marcado como atributo filtrado para leitura, o servidor não o exibe na entrada se todos os atributos forem solicitados. Contudo, se a pesquisa LDAP for realizada para exibir os atributos operacionais ou se a solicitação especificamente demandar atributos ACL, o atributo marcado será exibido. É possível usar `rrf_acl.ldif` para desabilitar a flag de filtrado para leitura em um atributo ACL. Esses LDIFs (LDAP Directory Interchange Format) afetam o atributo ACL no esquema, de maneira que apenas um usuário com privilégios de Supervisor na raiz da árvore poderá estendê-los.

Por padrão, um ACL não é marcado como filtrado para leitura, por isso não é visto o benefício, no desempenho, de as solicitações exibirem todos os atributos.

A tabela a seguir exibe o local dos arquivos `arf_acl.ldif` e `rrf_acl.ldif` em diferentes plataformas.

Plataforma	Local
Linux	◆ `/opt/novell/eDirectory/lib64/nds-schema/`
Windows	◆ `<unzipped_location>\eDirectory\windows\x64\NDSonNT\ndsnt\nds`

Desativando modelos ACL

Você pode desabilitar os modelos de ACL (lista de controles de acesso) para aumentar o desempenho em massa. A implicação disso é que alguns dos ACLs faltarão. Contudo, você pode resolver isso ao adicionar os ACLs necessários ao arquivo LDIF ou aplicando-os posteriormente.

1 Execute o seguinte comando:

```
ldapsearch -D cn_of_admin -w password -b cn=schema -s base
objectclasses=inetorgperson
```

A saída do comando seria:

```
dn: cn=schema
```

```
objectClasses: (2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
organizationalPerson STRUCTURAL MAY (groupMembership $ ndsHomeDirectory
$ loginAllowedTimeMap $ loginDisabled $ loginExpirationTime $
loginGraceLimit $ loginGraceRemaining $ loginIntruderAddress $
loginIntruderAttempts $ loginIntruderResetTime $
loginMaximumSimultaneous $ loginScript $ loginTime $
networkAddressRestriction $ networkAddress $ passwordsUsed $
passwordAllowChange $ passwordExpirationInterval $
passwordExpirationTime $ passwordMinimumLength $ passwordRequired $
passwordUniqueRequired $ printJobConfiguration $ privateKey $ Profile $
publicKey $ securityEquals $ accountBalance $ allowUnlimitedCredit $
minimumAccountBalance $ messageServer $ Language $ UID $
lockedByIntruder $ serverHolds $ lastLoginTime $ typeCreatorMap $
higherPrivileges $ printerControl $ securityFlags $ profileMembership $
Timezone $ sASServiceDN $ sASSecretStore $ sASSecretStoreKey $
sASSecretStoreData $ sASPKIStoreKeys $ userCertificate
$ nDSPKIUserCertificateInfo $ nDSPKIKeystore $ rADIUSActiveConnections
```

```
$
rADIUSAttributeLists $ rADIUSConcurrentLimit $ rADIUSConnectionHistory
$ rADIUSDefaultProfile $ rADIUSDialAccessGroup $ rADIUSEnableDialAccess
$ rADIUSPassword $ rADIUSServiceList $ audio $ businessCategory $
carLicense $ departmentNumber $ employeeNumber $ employeeType $
givenName $ homePhone $ homePostalAddress  $ initials $ jpegPhoto $
labeledUri $ mail $ manager $ mobile $ pager $ ldapPhoto $
preferredLanguage $ roomNumber $ secretary $ uid $ userSMIMECertificate
$ x500UniqueIdentifier $ displayName $ userPKCS12) X-NDS_NAME 'User' X
-NDS_NOT_CONTAINER '1' X-NDS_NONREMOVABLE '1' X-NDS_ACL_TEMPLATES
('2#subtree#[Self]#[All Attributes Rights]' '6#entry#[Self]#loginScript'
'1#subtree#[Root Template]#[Entry Rights]'
'2#entry#[Public]#messageServer' '2#entry#[Root Template]#groupMembership'
'6#entry#[Self]#printJobConfiguration' '2#entry#[Root
Template]#networkAddress'))
```

2. Na saída observada na etapa anterior, apague as informações marcadas em negrito.
3. Salve a saída revisada como um arquivo LDIF.
4. Acrescente as seguintes informações ao arquivo LDIF recém-salvo:

```
dn: cn=schema

changetype: modify

delete: objectclasses

objectclasses: (2.16.840.1.113730.3.2.2)

-

add:objectclasses
```

Por isso, seu LDIF deverá agora ser semelhante a este:

```
dn: cn=schema

changetype: modify

delete: objectclasses

objectclasses: (2.16.840.1.113730.3.2.2)

-

add:objectclasses
```

```
objectClasses: (2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
organizationalPerson STRUCTURAL MAY (groupMembership $ ndsHomeDirectory
$ loginAllowedTimeMap $ loginDisabled $ loginExpirationTime $
loginGraceLimit $ loginGraceRemaining $ loginIntruderAddress $
loginIntruderAttempts $ loginIntruderResetTime $
loginMaximumSimultaneous $ loginScript $ loginTime $
networkAddressRestriction $ networkAddress $ passwordsUsed $
passwordAllowChange $ passwordExpirationInterval $
passwordExpirationTime $ passwordMinimumLength $ passwordRequired
$ passwordUniqueRequired $ printJobConfiguration $ privateKey $ Profile
$
publicKey $ securityEquals $ accountBalance $ allowUnlimitedCredit $
minimumAccountBalance $ messageServer $ Language $ UID $
lockedByIntruder $ serverHolds $ lastLoginTime $ typeCreatorMap $
higherPrivileges $ printerControl $ securityFlags $ profileMembership $
Timezone $ sASServiceDN $ sASSecretStore $ sASSecretStoreKey $
sASSecretStoreData $ sASPKIStoreKeys $ userCertificate $
nDSPKIUserCertificateInfo $ nDSPKIKeystore $ rADIUSActiveConnections $
rADIUSAttributeLists $ rADIUSConcurrentLimit $ rADIUSConnectionHistory
$
rADIUSDefaultProfile $ rADIUSDialAccessGroup $ rADIUSEnableDialAccess
$ rADIUSPassword $ rADIUSServiceList $ audio $ businessCategory $
carLicense $ departmentNumber $ employeeNumber $ employeeType $
givenName $ homePhone $ homePostalAddress  $ initials $ jpegPhoto $
labeledUri $ mail
$ manager $ mobile $ pager $ ldapPhoto $ preferredLanguage $ roomNumber
$ secretary $ uid $ userSMIMECertificate $ x500UniqueIdentifier $
displayName $ userPKCS12) X-NDS_NAME 'User' X-ND S_NOT_CONTAINER '1' X
-NDS_NONREMOVABLE '1')
```

5 Digite o seguinte comando:

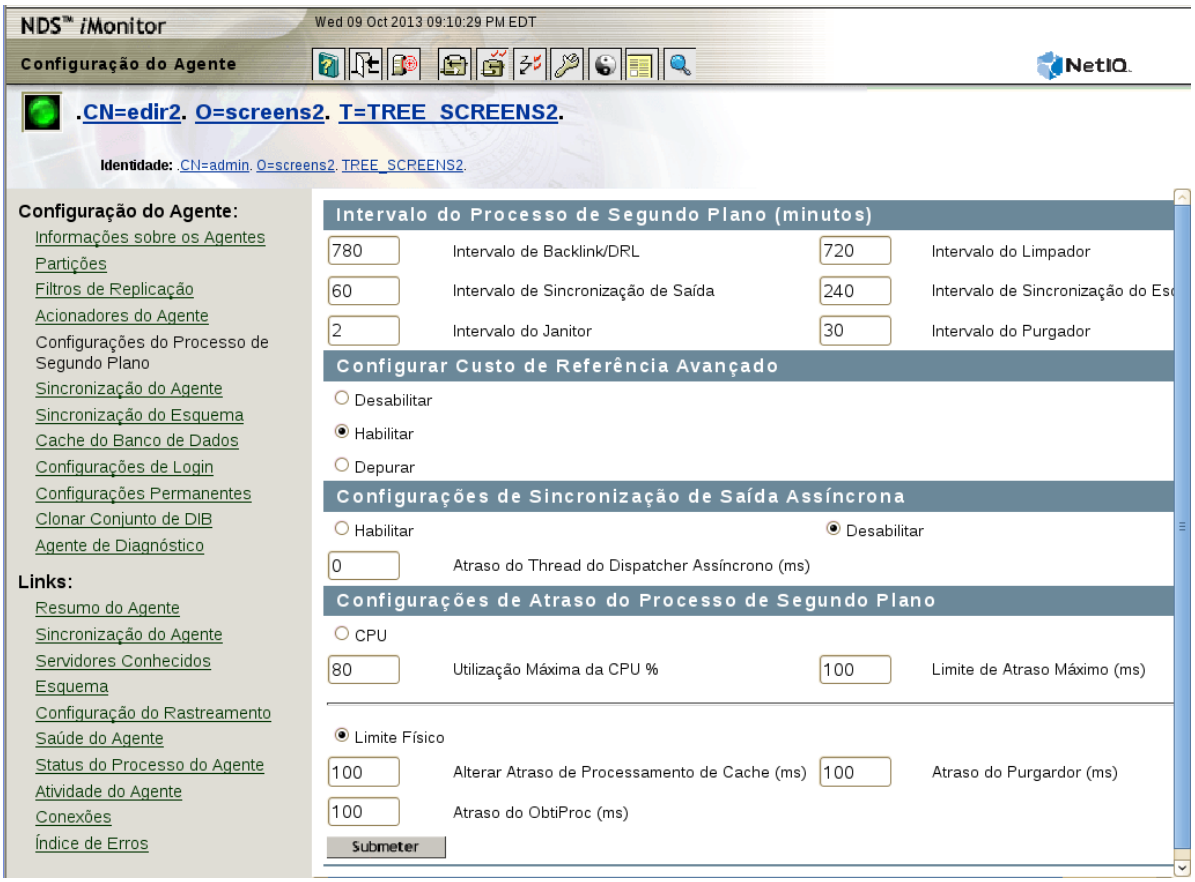
```
ldapmodify -D cn_of_admin -w password -f LDIF_file_name
```

Replicação

Nesta versão, alguns processos em segundo plano foram reprojetados para atender a ambientes dinâmicos maiores. Para obter mais informações, consulte “Gerenciando o processo em segundo plano” no *Guia de Administração do OpenText™ eDirectory CE 24.4 (v9.3)*.

Recomendamos definir o limite físico para *5ms* e *habilitar* a Sincronização de saída assíncrona. Contudo, se a utilização da CPU aumentar, aumente a duração do modo adormecido. Figura 4-1 mostra os valores definidos para as **Configurações de atraso de processo em segundo plano**.

Figura 4-1 *Configurações do Processo em Segundo Plano*



Testes internos em laboratório foram realizados em uma configuração de 10 servidores com as seguintes configurações: Limite físico- *0ms*, Sincronização de saída assíncrona - *habilitada* e Atraso de thread de dispatcher assíncrona - *0ms*. Os testes demonstraram que a replicação é sete vezes mais rápida do que as configurações padrão. Durante este teste, nenhuma outra operação do cliente foi realizada.

Observação: Para aproveitar os melhores benefícios de desempenho dos seus sistemas com essas melhorias de escalabilidade, é necessário executar o OpenText eDirectory 9.1 ou superior em todos os servidores. Mesmo se houver alguma versão mais antiga do anel de réplicas, haverá melhoria no desempenho.

SSD (Solid State Disk - Unidade de Estado Sólido)

Esta versão suporta SSD empresarial para uma melhor operação de IO. O Tabela 4-1 na página 26 mostra a melhoria no desempenho de conserto no SSD em nossa configuração de teste:

Tabela 4-1 *Desempenho de conserto*

Tamanho do DIB (GB)	HDD (tempo em minutos)	SSD (tempo em minutos)	% de melhoria
11	80	53	33,75
24	277	169	38,98
34	542	296	45,38
75	1383	618	55,31
98	3171	1023	67,73

Intervalo de atualização de login do NMAS

Para obter mais informações, consulte "Usando os atributos sasUpdateLoginInfo e sasUpdateLoginTimeInterval" no *Guia de Administração do OpenText™ eDirectory CE 24.4 (v9.3)*.

Overhead do SSL

O LDAP sobre SSL representa uma carga adicional na CPU devido a seus requisitos de criptografia. Um estudo de desempenho em laboratório mostra acerto de desempenho maior que 10% devido ao overhead de criptografia.

ICE (Import Convert and Export)

O utilitário OpenText Import Convert and Export (ICE) usa um protocolo de atualização em massa otimizado chamado LBURP para fazer upload de dados para o OpenText eDirectory. Esse protocolo é significativamente mais rápido do que fazer upload dos dados usando um comando `ldapmodify` simples. Para obter mais informações, consulte Utilitário de upload em massa offline no *Guia de Administração do OpenText™ eDirectory CE 24.4 (v9.3)*.

ldif2dib

Para obter mais informações sobre os ajustes de desempenho do OpenText durante um upload em massa offline usando o utilitário ldif2dib, consulte Ajustando o ldif2dib, no *Guia de Administração do OpenText™ eDirectory CE 24.4 (v9.3)*.

Tamanho do pacote de NCP melhorado

Para estabelecer comunicação entre vários servidores, o OpenText eDirectory usa o Protocolo Central do Netware (NCP) como o protocolo de comunicações. Nas versões anteriores, o tamanho máximo do pacote permitido pelo NCP era de 64 KB, o que limitava a throughput máxima quando os dados eram transferidos através do NCP. Esta versão melhora a capacidade do NCP de processar tamanhos do pacote de até 1 MB, permitindo ao OpenText eDirectory sincronizar até 1 MB de dados em um único pacote. O OpenText eDirectory inicia a sincronização com um pacote de 64 KB e

aumenta o tamanho do pacote de acordo com os dados restantes a serem sincronizados. Esse processo melhora o desempenho da replicação de forma significativa. Se ambos os seus servidores executam a versão 9.2, você não precisa realizar outras configurações para tirar proveito desse aprimoramento.

5 Configuração do OpenText eDirectory

Esta seção inclui as seguintes informações:

- “Configurando o subsistema FLAIM” na página 29
- “Modificando as configurações de cache de FLAIM” na página 29

Configurando o subsistema FLAIM

Para atender a uma ampla gama de implantações e configurações, dois mecanismos de controle de consumo de cache de memória são fornecidos no OpenText eDirectory. Esses mecanismos são mutuamente exclusivos.

- “Limite de cache físico” na página 29
- “Ajustando o limite dinamicamente” na página 29

Limite de cache físico

Você pode especificar um limite de memória física da seguinte maneira:

- Como número fixo de bytes.
- Como porcentagem da memória física.
- Como porcentagem da memória física disponível.

Quando um limite físico é especificado usando o segundo ou terceiro método, ele sempre é traduzido como um número fixo de bytes. Isso significa que para o segundo método, o número de bytes é a porcentagem da memória física detectada quando o OpenText eDirectory é iniciado. Para o terceiro método, o número de bytes é a porcentagem da memória física disponível detectada quando o OpenText eDirectory é iniciado.

Ajustando o limite dinamicamente

Um ajuste dinâmico faz com que o OpenText eDirectory ajuste periodicamente seu consumo de memória em virtude do consumo de memória variável de outros processos. Embora ajustar a memória dinamicamente funcione bem em cenários típicos, esse mecanismo não é recomendado para um desempenho ideal do OpenText eDirectory em plataformas Linux devido à grande diferença de padrões de uso e alocadores de memória em tais plataformas.

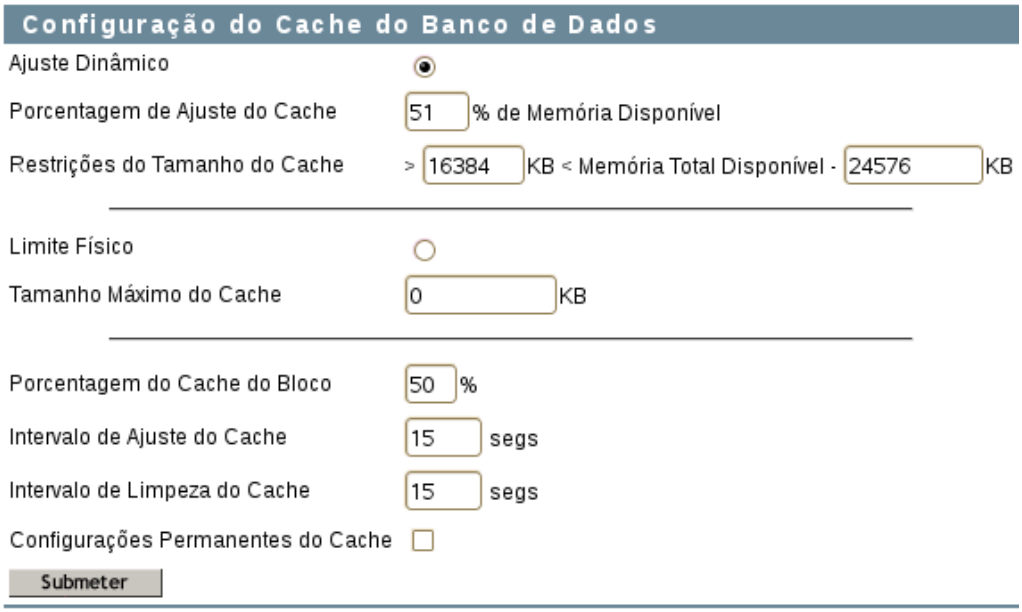
Modificando as configurações de cache de FLAIM

- “Modificando as configurações de cache de FLAIM pelo OpenText iMonitor” na página 30
- “Modificando as configurações de cache do FLAIM pelo _ndsdb.ini” na página 31

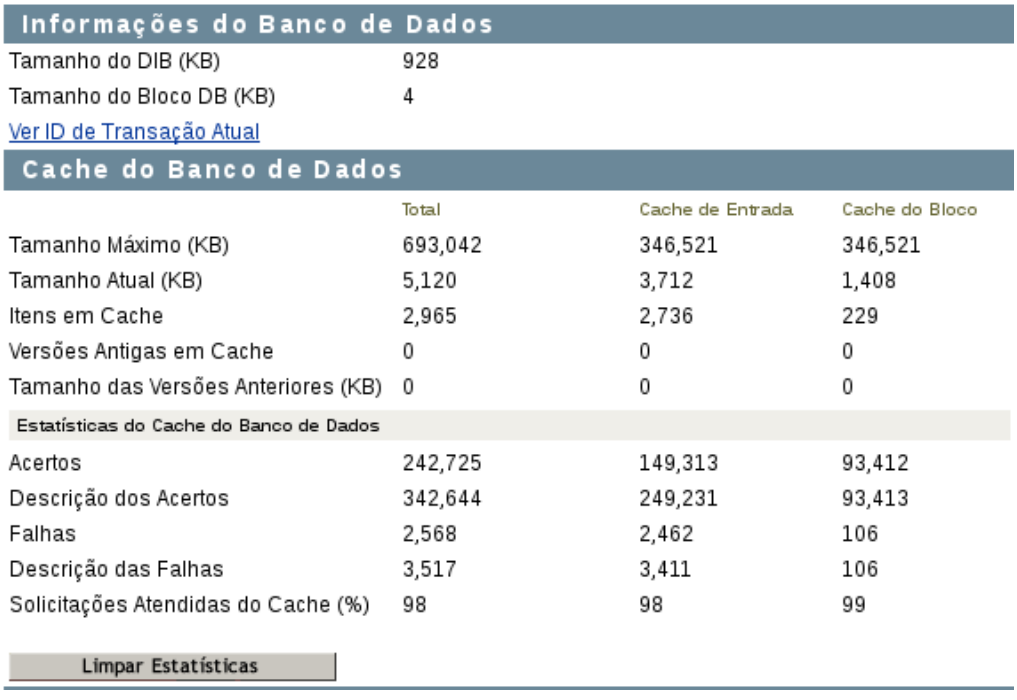
Modificando as configurações de cache de FLAIM pelo OpenText iMonitor

Você pode usar o OpenText iMonitor para fazer o seguinte:

- Ver ou mudar as configurações de cache.



- Monitorar as estatísticas de cache.



Consulte o cache do Banco de dados na Configuração de agente do OpenText iMonitor para obter as informações acima.

Informações de cache de banco de dados	Descrição
Tamanho Máximo do	O tamanho máximo (em KB) permitido que o cache especificado atinja.
Tamanho Atual	O tamanho atual (em KB) do cache especificado.
Itens em Cache	O número de itens no cache especificado.
Versões Antigas em Cache	O número de versões antigas no cache especificado. Versões antigas de itens de cache são retidas para manter a consistência de transações de leitura no banco de dados. Em outras palavras, se um thread estiver em uma transação de leitura e outro em transação de gravação, as versões antigas dos blocos modificados pelo gravador serão mantidas em prol do leitor. Isso ocorre para garantir que os resultados do leitor produzam uma visualização consistente pela duração da transação mesmo que ocorram modificações no momento.
Tamanho das Versões Anteriores	O tamanho (em KB) dos itens armazenados da versão antiga.
Acertos	O número de vezes que um item foi acessado com êxito do cache especificado.
Descrição dos Acertos	O número de itens observados no cache antes que um item tenha sido acessado com êxito do cache especificado. A razão de acerto de observação versus acerto é a medida da eficiência da pesquisa do cache. Normalmente, esta razão deverá estar próxima a 1:1.
Falhas	O número de vezes que um item não foi encontrado no cache especificado e teve que ser obtido em um cache de nível inferior ou do disco.
Descrição das Falhas	O número de itens observados no cache antes de ser determinado que o item desejado não estava presente no cache especificado. A razão de falha de observação versus falha é uma medida da eficiência da pesquisa do cache. Normalmente, esta razão deverá estar próxima a 1:1.

Modificando as configurações de cache do FLAIM pelo _ndsdb.ini

As configurações de cache do FLAIM e outras configurações deste podem ser alteradas ao modificar o arquivo `_ndsdb.ini` existente no diretório do DIB. Reinicie o OpenText eDirectory após mudar o arquivo `_ndsdb.ini`.

É possível definir o limite de ajuste dinâmico ou limite de cache físico. As opções de cache são listadas abaixo. Várias opções podem ser especificadas, em qualquer ordem, separadas por vírgulas. Todas são opcionais.

- **DYN ou HARD** - Ajustar dinamicamente um limite ou limite físico.
- **% : porcentagem** - Porcentagem da memória física ou disponível para uso.

- **AVAIL ou TOTAL** - A porcentagem especifica a memória disponível ou a memória física total. Aplica-se somente ao limite físico e é ignorado pelo limite de ajuste dinâmico, pois este sempre é calculado com base na memória física disponível. Por padrão, esse recurso é AVAIL.
- **MIN:bytes** - Número mínimo de bytes.
- **MAX:bytes** - Número máximo de bytes.
- **LEAVE:bytes** - Número mínimo de bytes para deixar.

  Por exemplo:

  ```
  cache=HARD,%:75, MIN:200000000
  ```

  ```
  cache=500000000
  ```

- **preallocatecache: verdadeiro/falso** - Essa configuração faz com que o OpenText eDirectory aloque previamente a quantidade de memória especificada pelo limite de cache físico.
- **rfldirectory -** É possível especificar um caminho diferente para os arquivos RFL.
- **cpinterval** - Número de segundos após os quais o FLAIM força um ponto de verificação. O padrão é 3 minutos.
- **maxdirtycache** - Número máximo de bytes de cache modificado.
- **lowdirtycache** - Número mínimo de bytes de cache modificado.
- **blockcachepercent** - Porcentagem do cache do FLAIM usado para o cache do bloco.
- **cacheadjustinterval** - Intervalo em segundos para ajustar dinamicamente o cache.
- **cachecleanupinterval** - Intervalo em segundos para apagar versões antigas de entradas e blocos do cache.