

Desempenho do Pervasive[®] PSQL v10

Principais Recursos de Desempenho do Pervasive PSQL

Pervasive PSQL v10 White Paper

Junho de 2008

CONTEÚDO

Introdução	3
Desempenho básico: mais memória e menos disco e I/O de rede	3
Suporte a 64 bits: mais memória.....	3
Métodos de cache de sistemas de gerenciamento de banco de dados (SGBD) e de sistemas de arquivos.....	4
Utilização mais eficiente da memória: XIO e Cache Dinâmico	4
Escritas e leituras eficientes: Compactação de página e registro, Turbo Write Accelerator	8
Redução do I/O de rede: Cache de cliente.....	9
Resumo	11
Apêndice A – Configurações de desempenho	12
Apêndice B — Comparação entre cache estático e cache dinâmico.....	14
Informações para contato	15

INTRODUÇÃO

O banco de dados Pervasive PSQL tem a merecida reputação de possuir características particularmente adequadas às necessidades das pequenas e médias empresas: baixa manutenção, auto-otimização, auto-regulagem e facilidade de instalação e integração com opções de implementação. Além disso, sua velocidade já é conhecida há muito tempo. Os aplicativos que rodam no Pervasive PSQL oferecem sempre uma ótima experiência ao usuário porque passamos 25 anos implementando aperfeiçoamentos de desempenho a cada nova versão. Este artigo destaca alguns dos mais importantes aperfeiçoamentos de desempenho do Pervasive PSQL e apresenta algumas sugestões de configuração para que você possa otimizar seu banco de dados. Para um exame mais detalhado dos ajustes de desempenho do PSQL, consulte o Capítulo 5 do Guia de Operações Avançadas do Pervasive PSQL Summit™ v10.

A filosofia da Pervasive para melhorar o desempenho é aperfeiçoar o desempenho geral dos aplicativos sem prender-se a um conjunto de testes arbitrários. O objetivo deste artigo é explicar as melhorias no desempenho do Pervasive PSQL e servir como guia para que desenvolvedores e fornecedores independentes de software possam compreender essas melhorias e avaliar os benefícios resultantes, tanto nos seus próprios testes de desempenho quanto no uso de seus aplicativos pelos usuários.



DESEMPENHO BÁSICO: MAIS MEMÓRIA E MENOS DISCO E I/O DE REDE

Há três fatores principais relacionados ao banco de dados que afetam o desempenho dos aplicativos. São eles: a latência das operações em disco, o I/O da rede e a utilização do processador e da memória. Como o acesso à memória RAM é mais que 100.000 vezes mais rápido que o acesso ao disco, o aumento da quantidade de memória e da eficiência na sua utilização normalmente melhoram o desempenho dos aplicativos. O Pervasive PSQL contém vários aperfeiçoamentos para facilitar essas duas estratégias.

SUPORTE A 64 BITS: MAIS MEMÓRIA

Uma maneira simples de reduzir as operações de I/O de disco é maximizar a quantidade de RAM disponível. É fácil perceber que o desempenho dos aplicativos melhora quando os dados são mantidos na memória em vez de serem lidos do disco. Afinal, a velocidade do acesso a disco é prejudicado pela presença de peças móveis, já que essa é uma das poucas funções mecânicas que fazem parte do processamento. Do ponto de vista do software, o acesso a disco exige uma "chamada ao sistema" relativamente dispendiosa em termos de desempenho.

Acesso a disco e acesso a RAM

	Latência média = 4,2 ms Tempo médio de leitura = 9 ms Total = 13 ms = 0,013 segundo.
	Tempo de acesso = 70 ns = 0,00000007 segundo.

MÉTODOS DE CACHE DE SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS (SGBD) E DE SISTEMAS DE ARQUIVOS

Teoricamente, as máquinas de 32 bits da Intel e da AMD podem endereçar até 4GB de RAM. Nas máquinas que rodam Windows, esses 4GB são divididos entre o sistema operacional e os aplicativos. Isso quer dizer que os aplicativos podem acessar no máximo 2GB de memória. Assim, aumentar a memória de uma máquina Windows de 32 bits que tenha 4GB de RAM não tem qualquer efeito no desempenho. Com a utilização da API Address Windowing Extension (AWE) da Microsoft no Windows 2000 Server e no Windows Server 2003, os aplicativos de 32 bits podem acessar mais memória endereçável. Entretanto, o aplicativo precisa ser criado com suporte à AWE e essa abordagem afeta outras partes do sistema operacional porque reduz a quantidade de memória disponível para o sistema operacional e para os drivers.

O Pervasive PSQL Summit v10 tem suporte à arquitetura de 64 bits e ao Windows Vista™. Os aplicativos que utilizam grandes conjuntos de dados são beneficiados pela utilização de servidores e sistemas operacionais de 64 bits porque, em muitos casos, eles podem carregar o banco de dados inteiro na RAM. O Windows Vista 64 suporta entre 8GB e 128GB de memória (dependendo da versão), por isso os aplicativos podem fazer o carregamento prévio de uma maior quantidade de dados na memória virtual, permitindo um acesso rápido e um grande aumento no desempenho dos aplicativos. A tabela seguinte resume o espaço de endereçamento disponível para os aplicativos Windows de 32 e 64 bits.

Arquiteturas de 32 bits e de 64 bits

ESPAÇO DE ENDEREÇAMENTO	WINDOWS 64 BITS	WINDOWS 32 BITS
Memória virtual	16 terabytes	4 gigabytes
Arquivo de paginação	512 terabytes	16 terabytes
Cache do sistema	1 terabyte	1 gigabyte

O Pervasive PSQL Summit v10 é a escolha lógica para desenvolvedores que estão prontos para colher os benefícios oferecidos pelas plataformas de 64 bits. Os usuários que adquirirem plataformas e sistemas operacionais de 64 bits poderão alavancar os recursos extras de processamento e memória, mesmo com aplicativos de 32 bits. O Pervasive PSQL suporta três tipos de configuração para otimização do desempenho:

Bom

Aplicativos de 32 bits utilizando um banco de dados PSQL de 32 bits rodando em uma plataforma de 64 bits, permitindo ao sistema operacional utilizar a configuração interna e o poder de processamento nativo.

Melhor

Aplicativo de 32 bits acessando um banco de dados PSQL de 64 bits em uma plataforma de 64 bits, ampliando a memória de banco de dados para o aplicativo.

Ótimo

Aplicativo de 64 bits acessando um banco de dados PSQL de 64 bits rodando em uma plataforma de 64 bits. Isso permite que todos os componentes sejam executados em uma configuração nativa para um desempenho ótimo.

UTILIZAÇÃO MAIS EFICIENTE DA MEMÓRIA: XIO E CACHE DINÂMICO

XTREME I/O

O Xtreme I/O (XIO) é um recurso do Pervasive PSQL Summit v10 que melhora o desempenho agilizando o acesso aos arquivos de dados do Pervasive PSQL no disco. O XIO é implementado como um device driver para aplicativos de servidor do Pervasive PSQL em plataformas Windows de 32 bits. Um dos requisitos de sistema é a instalação de 2GB de RAM antes da instalação do Pervasive PSQL. O Guia de Operações Avançadas tem mais detalhes.

O XIO e o sistema de banco de dados trabalham em conjunto para aumentar o desempenho. O sistema notifica o XIO quando um arquivo de dados é aberto. A partir desse ponto, o XIO acelera o acesso ao arquivo de dados no disco. Por exemplo, considere um aplicativo Pervasive PSQL inserindo vários registros em uma tabela. Essas inserções produzem requisições aleatórias de I/O a várias regiões do arquivo de dados. O XIO organiza e otimiza as inserções para reduzir o número total de requisições de I/O de escrita e armazenamento. Isso reduz a carga do controlador de armazenamento e do bus PCI e, por causa da carga reduzida do processador, libera mais capacidade para outros tipos de tráfego.

O XIO pode utilizar diretamente a RAM disponível para reservar (ou atribuir) seu próprio cache de banco de dados totalmente gerenciável. O XIO otimiza as operações de escrita e utiliza algoritmos de compactação inteligentes para aumentar a capacidade de armazenamento de dados. O funcionamento do XIO é transparente — nenhuma intervenção do usuário ou do aplicativo é necessária.

Cache do XIO

A maioria dos sistemas de cache suportam todos os aplicativos do sistema e procuram otimizar o funcionamento do cache de maneira geral. O cache do XIO foi projetado para armazenar somente os arquivos acessados pelos aplicativos através do banco de dados Pervasive PSQL. Ele também permite que o banco de dados amplie o cache do sistema até atingir a quantidade de memória física disponibilizada pelo Windows, ou seja, mais que o limite prático de 2GB da maioria dos aplicativos de 32 bits. O XIO utiliza primeiro a memória estendida acima dos 4GB para acesso por meio da extensão de endereço físico (physical address extension — PAE), se disponível. O XIO retém a memória estendida até o desligamento do sistema; o tamanho do cache permanece constante e não é reduzido nem ampliado. Se não houver memória estendida, o XIO cria seu cache a partir da memória padrão (RAM até 4GB).

Por exemplo, o Windows 2000 Advanced Server com PAE permite o acesso a até 8GB de memória física. Com o XIO, o cache do sistema de banco de dados é ampliado utilizando a memória física disponibilizada pelo Windows. Sem o XIO, o tamanho do cache do banco de dados nunca ultrapassa 1,5GB (ou o que resta do limite de 2GB do Windows 32 bits depois da carga dos executáveis).

Quando utiliza a memória padrão, o XIO contrabalança as exigências de memória do sistema de banco de dados e os requisitos de todo o sistema operacional e outros aplicativos que não utilizam o banco de dados. O cache do XIO pode ser reduzido e ampliado dinamicamente à medida que outros recursos do sistema requisitam e liberam memória. Isso proporciona um alto nível de desempenho para os servidores que executam vários aplicativos acessados simultaneamente por muitos usuários.

Compactação XIO.

O XIO contém vários algoritmos de compactação, cada qual adequado para um tipo diferente de dado. Em vez de utilizar um único compactador para todos os dados de entrada, o XIO utiliza um algoritmo de leitura antecipada que seleciona o algoritmo de compactação adequado para os dados. Essa compactação dos dados como parte do processo de cache permite que o XIO utilize com mais eficiência os recursos de cache. Por exemplo, com uma taxa de compactação de 2:1, o XIO pode armazenar o dobro de dados no cache.

Escrita agilizada XIO

O XIO utiliza duas técnicas para melhorar a eficiência das escritas em disco: agregação de escrita e ordenamento de escrita. Agregando várias requisições de escrita para formar uma só requisição maior, o XIO reduz a carga do controlador de disco e do bus PCI. Ordenando as requisições, o XIO reduz a quantidade de operações de busca executadas pela cabeça do disco rígido para completar a escrita. A redução do número de escritas e a racionalização da ordem das escritas melhora bastante os tempos de escrita.

Quando utilizar o XIO

A tabela seguinte identifica as características de aplicativos que podem produzir um aumento do desempenho quando se utiliza o XIO:

FATOR	DETALHE	UTILIZAR XIO
O aplicativo tem um padrão aleatório de acesso a dados (o acesso aos dados tende a não ser seqüencial)	Essa característica é típica de aplicativos que utilizam sistemas de gerenciamento de banco de dados.	Sim
O conjunto de dados não cabe completamente no cache de sistema do Windows.	Quanto mais o cache de sistema do Windows é excedido, maior é a quantidade de leituras e escritas em disco. O XIO é capaz de colocar as leituras e escritas em cache. Se os dados residirem inteiramente no cache de sistema do Windows, o XIO não processa a requisição de leitura ou escrita.	Sim
O aplicativo faz leituras e escritas freqüentes em disco.	O acesso é, muitas vezes, um fator limitante do desempenho do aplicativo.	Sim
Você já utiliza outro programa para acelerar o I/O de disco.	A utilização de vários programas para acelerar o I/O de disco não é aconselhável porque eles podem interferir uns com os outros e produzir resultados imprevisíveis.	Não

Para obter mais informações sobre o Xtreme I/O, consulte o capítulo Desempenho do Guia de Operações Avançadas do Pervasive PSQL Summit V10.

CACHE DINÂMICO

É um princípio largamente reconhecido na computação que os dados recentemente requisitados tendem a ser requisitados novamente. O Pervasive.SQL e o Btrieve sempre incluíram um recurso de cache integrado que mantém blocos de dados (páginas) que também são armazenados em disco. A partir do Pervasive SQL V8, foi implementado um segundo nível de cache, um cache dinâmico preparado para operar em condições variáveis de memória e para aproveitar a memória não utilizada.

A alocação estática de cache (um bloco contínuo de memória definido na configuração) pode ser aceitável em servidores dedicados cuja utilização de memória pode ser monitorada e prevista, mas não é adequada para outras situações porque a) o cache não pode ser ampliado para aproveitar a memória não utilizada do sistema, e b) o tamanho do bloco contínuo pode ser limitado por fragmentação de memória ou pelo sistema operacional. Em sistemas onde a disponibilidade de memória varia constantemente, como as estações de trabalho e os servidores com vários aplicativos, a capacidade de adaptação dinâmica aos requisitos memória que mudam à medida que os usuários abrem e fecham outros processos é essencial.

Se poucos processos que utilizam intensivamente a memória estiverem sendo executados no computador, pode haver uma quantidade significativa de memória livre que poderia ser usada para o cache de dados, aumentando assim o desempenho geral do banco de dados. As primeiras versões dos servidores PSQL eram configuradas para utilizar 20% da memória física. Em servidores de banco de dados dedicados, esse padrão deixava uma considerável quantidade de RAM ociosa. Além de permitir a configuração do tamanho estático do cache, o Pervasive PSQL utiliza o cache do sistema (para ler e escrever a partir do cache do sistema operacional, não do disco). Como não há troca de informações entre o cache do Pervasive PSQL e o cache do sistema e como os dois caches utilizam os mesmos algoritmos LRU (Least Recently Used: a memória utilizada menos recentemente tem preferência para ser removida do cache) para determinar o que deve ser mantido e o que deve ser descartado, a maior parte dos dados no cache são redundantes. Em outras palavras, se o cache do Pervasive PSQL utiliza 20% da memória total disponível, o cache do sistema pode utilizar até 80%. Mas a utilização redundante de memória para manter no cache as páginas que já estão no cache do Pervasive PSQL pode chegar a 20% do cache do sistema.

Cache Dinâmico: Utilização eficiente de memória e otimização simplificada

Outro motivo para implementar o cache dinâmico no Pervasive PSQL é a eliminação da necessidade de otimização da configuração do cache por parte do usuário final. O método estático exigia que o usuário alterasse a configuração para melhorar a eficiência da utilização dos recursos do sistema em cada ambiente diferente e sempre que o ambiente era modificado.

Finalmente, o cache estático do banco de dados exigia um bloco contínuo de memória do sistema operacional. O cache dinâmico não precisa de um bloco contínuo de memória, por isso pode utilizar com mais eficiência a memória disponível no sistema.

Arquitetura do cache dinâmico

O Pervasive PSQL utiliza um cache com duas camadas: L1 e L2. A camada L1 é similar ao cache estático utilizado nas versões anteriores do banco de dados. A camada L2 é o novo cache totalmente dinâmico. As páginas são movidas entre L1 e L2 conforme necessário. A camada L1 contém todas as páginas que estejam sendo ativamente acessadas ou modificadas. A camada L2 contém as páginas que foram acessadas recentemente, mas que não estão ativas. A separação entre o cache ativo L1 e o cache inativo L2 significa que o gerenciamento do cache dinâmico pode ser realizado em segundo plano, independentemente da atividade específica do kernel.

Com a utilização de cache dinâmico, o banco de dados não necessita mais de alocações do cache do sistema. Isso melhora o desempenho do acesso de leitura de duas maneiras: Primeiro, a quantidade de memória em uso pode ser controlada com base nas necessidades de dados do Pervasive PSQL. Segundo, não são feitas chamadas ao sistema e o cache L2 pode presumir que apenas um processo está acessando seu conteúdo. O cache do sistema, ao contrário, precisa presumir que vários processos podem usar seus recursos ao mesmo tempo.

Benefícios de desempenho do cache dinâmico

Os benefícios de desempenho do cache dinâmico dependem de fatores como disponibilidade de memória no servidor, tamanho do banco de dados e padrões de acesso. Com essas ressalvas, porém, podemos delinear uma métrica geral de melhoria de desempenho com base na utilização relativa de memória.

Vamos começar com um exemplo de duas máquinas: uma delas utiliza o cache dinâmico e a outra utiliza o cache estático. Vamos supor que tanto o cache L1 da máquina com cache dinâmico quanto o cache estático da outra máquina estejam configurados para utilizar 20% da memória disponível. Como o cache do sistema utiliza os mesmos algoritmos LRU, ele terá muita redundância em relação aos caches estático e L1.

Cache estático: Em um servidor com 4GB, a quantidade de memória utilizada pelo banco de dados é de 20%, ou 800MB. (embora o cache do sistema utilize mais 800MB por causa do banco de dados, ele não melhora o desempenho porque seus dados são os mesmos armazenados no cache estático).

Cache Dinâmico: Em um servidor com 4GB, o cache L1, de 20%, utiliza 800MB, deixando 3,2GB disponíveis. O tamanho máximo do cache L2 é determinado por outra opção de configuração, Max Memory Usage, que é o percentual da memória total utilizada pelos caches L1+L2. Se configurarmos essa opção para utilizar 50% (2GB) nesse sistema, a quantidade de memória disponível para L2, descontada a memória utilizada por L1, será de 2GB - 800MB = 1,2GB.

Nesse simples exemplo, aumentamos a memória disponível para o cache do banco de dados de 800MB para 2GB, um fator de 2,5x. Para um cálculo detalhado dos efeitos do cache dinâmico sobre o desempenho, por favor consulte o Apêndice B.

Pode-se dizer que, em alguns casos, resultados similares podem ser obtidos através do ajuste do servidor de banco de dados, eliminando o cache do sistema e maximizando L1. Entretanto, esse ajuste exige que o sistema como um todo seja basicamente estático e dedicado como servidor de banco de dados. Os usuários que não tenham servidores de banco de dados dedicados nem administradores de bancos de dados (DBA) que possam fazer constantes ajustes no banco de dados ficarão satisfeitos em saber que o Pervasive PSQL pode fazer o ajuste automático do cache, melhorando a utilização de memória nas instalações de servidores e estações de trabalho. Esse é um benefício importante para pequenas e médias empresas que buscam obter a melhor configuração e desempenho de aplicativos.

ESCRITAS E LEITURAS EFICIENTES: COMPACTAÇÃO DE PÁGINA E REGISTRO, TURBO WRITE ACCELERATOR

Compactação

O Pervasive PSQL tem dois tipos de compactação de dados: registro e página. Eles podem ser utilizados separadamente ou em conjunto. A compactação melhora o desempenho porque reduz o tamanho dos arquivos de dados e aumenta a velocidade das leituras e escritas no cache.

Compactação de registros

O Pervasive PSQL tem um recurso para compactar em 3 bytes as seqüências de 5 ou mais caracteres idênticos. Como resultado, dependendo do tipo de registro dos dados, o espaço de armazenamento necessário pode ser significativamente reduzido. A compactação de registros proporciona os melhores resultados nas seguintes situações:

- Os registros a serem compactados estão estruturados de modo a maximizar os benefícios do uso da compactação de dados.
- A eficiência da utilização do espaço em disco é mais importante que o possível aumento do tempo de processamento e do tempo de acesso necessários para os arquivos compactados.
- O computador que contém o banco de dados tem memória suficiente para os *buffers* de compactação utilizados para expandir o registro para leitura.

A compactação de registros resulta em maior eficiência quando cada registro pode conter uma grande quantidade de caracteres repetidos. Por exemplo, os registros podem conter vários campos inicializados com espaços pela sua tarefa antes da inserção no arquivo. A compactação é mais eficiente se esses campos estiverem lado a lado no registro, não separados entre si por campos contendo outros valores.

Compactação de páginas

Internamente, os bancos de dados Pervasive PSQL são conjuntos de páginas de vários tipos. A compactação de páginas controla a compactação e a descompactação das páginas de dados de um arquivo.

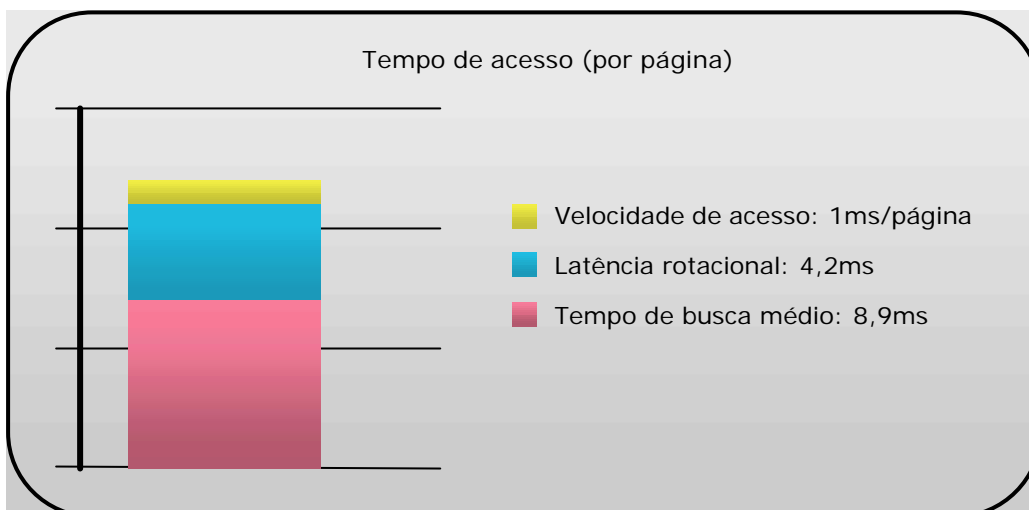
Quando o arquivo é lido do disco, suas páginas são descompactadas e colocadas no cache. As leituras e atualizações de registros são feitas nos dados descompactados presentes no cache. Quando ocorre uma operação de escrita, a página de dados é compactada e gravada no disco. Dependendo do gerenciamento do cache, a página compactada é mantida na memória até que seja acessada novamente. A compactação de páginas resulta em maior eficiência nas seguintes condições:

- Os dados são altamente compactáveis com algoritmos do tipo ZIP. Quando o tamanho do arquivo pode ser reduzido por um fator de 4:1 ou mais, o desempenho do arquivo pode aumentar significativamente.
- O aplicativo executa uma grande quantidade de leituras em relação a outras operações (inserções, atualizações e remoções).

Para obter mais informações sobre a utilização da compactação de registros e páginas, consulte o Guia do Programador do Pervasive PSQL (capítulo 5) e o Guia de Operações Avançadas (capítulo 13).

Turbo Write Accelerator

Uma das características da escrita em disco é que, uma vez iniciada, é muito mais rápido continuar escrevendo do que parar, reposicionar a cabeça em uma nova trilha física e reiniciar a escrita. Em outras palavras, a escrita contínua é muito mais rápida que a descontínua. Além disso, as chamadas de escrita exigem interação com o sistema operacional. A execução de uma única escrita grande em lugar de várias pequenas pode diminuir significativamente o tempo necessário para gravar os dados. O TWA (Turbo Write Accelerator) pré-aloca espaços vagos no arquivo físico de forma que várias páginas possam ser escritas como uma só página combinada. Isso melhora o desempenho de I/O porque reduz a fragmentação (e, portanto, o tempo de acesso) em arquivos atualizados com frequência.



Os benefícios de desempenho do TurboWrite Accelerator dependem da quantidade de operações de escrita sendo executadas, do tamanho da página de dados e das configurações de transação do sistema. Os arquivos com páginas menores são mais beneficiados que os arquivos com páginas maiores, pois os dados nas páginas maiores já estão combinados.

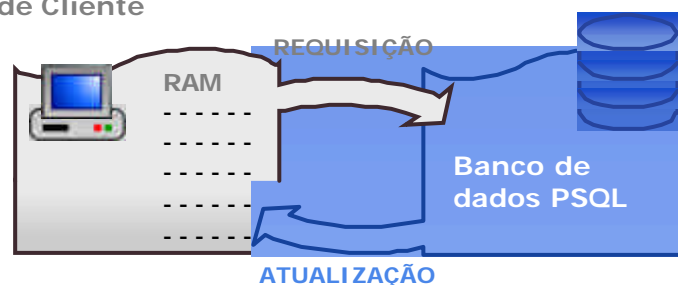
O desempenho das escritas em disco quando se utiliza o Turbo Write Accelerator aumenta com o número de páginas livres no arquivo devido ao aumento da possibilidade de escrita contínua de páginas. A opção para ajustar o percentual de páginas livres chama-se File Growth Factor. Para obter mais detalhes sobre a opção File Growth Factor, consulte o capítulo 4 do Guia de Operações Avançadas.

REDUÇÃO DO I/O DE REDE: CACHE DE CLIENTE

Cache de cliente

Para reduzir o I/O de rede, o Pervasive PSQL utiliza um cache no cliente (na configuração cliente/servidor). Além de responder às requisições dos clientes registro a registro, o servidor do Pervasive PSQL inclui o conceito de servidor de páginas. Segundo esse conceito, as chamadas de I/O de disco do sistema local são transferidas para o servidor remoto. O cliente Pervasive PSQL contém um sistema que pode utilizar um cache de páginas local. Entretanto, as leituras com bloqueio, as escritas e as operações dentro de transações do usuário são enviadas ao servidor como operações sobre registros, não sobre páginas. O cache de cliente e o servidor de páginas gerenciam o acesso simultâneo ao cache, detectando e alternando dinamicamente entre leituras do cache de cliente e leituras por requisição de registro e entre a taxa de acerto do cache de cliente e o nível geral de tráfego do sistema.

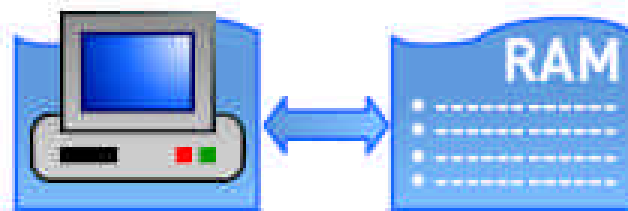
Sistema de Cache de Cliente



Do ponto de vista do aplicativo, os dados acessados com mais frequência sempre estão disponíveis no cache do cliente. Todas as operações update, insert delete e lock e as transações são passadas para o servidor. Assim, o cache de cliente não afeta diretamente o desempenho de escrita. O sistema de cache de cliente também suporta o chaveamento dinâmico entre o modo servidor de página/cache de cliente e o modo cliente/servidor baseado em registros caso o custo de desempenho do cache de cliente supere seu benefício para um determinado arquivo.

Quando o cache de cliente é útil

O benefício principal do cache de cliente é a manutenção de um cache local para o aplicativo, o que reduz o tráfego de rede e o I/O de disco quando os mesmos dados são lidos várias vezes. Os maiores benefícios são obtidos em operações que executam muitas leituras, como a geração de relatórios. Nos casos em que um conjunto de dados relativamente pequeno e estático é acessado repetidamente (como, por exemplo, uma lista de contas em uma tabela de contabilidade) ou quando há uma grande probabilidade de que a mesma página seja lida várias vezes, os dados podem estar disponíveis no cache de cliente e podem ser fornecidos rapidamente ao aplicativo, sem a necessidade de I/O de disco ou rede.



RESUMO

Este artigo destacou os aperfeiçoamentos de desempenho mais significativos criados pela Pervasive em seus produtos de banco de dados ao longo dos últimos 25 anos. Muitos desses recursos são exclusivos do banco de dados Pervasive PSQL e estão patenteados. Outros são exemplos da capacidade de otimização do Pervasive PSQL utilizando avanços tecnológicos no nível da plataforma, como a mudança para 64 bits. Há muitos outros avanços não descritos aqui. Otimizações SQL, utilitários aprimorados, melhores ferramentas para o desenvolvedor e outros avanços fazem parte do processo de melhoria do desempenho dos desenvolvedores e aplicativos. As pequenas e médias empresas clientes da Pervasive não se caracterizam pela abordagem clássica da TI corporativa para a solução de problemas de desempenho: aumentar a quantidade de hardware. O orçamento limitado das pequenas e médias empresas não permite a aquisição de hardware. Por isso a Pervasive tornou-se criativa na busca de diferentes maneiras de melhorar o desempenho dos bancos de dados e dos aplicativos. O fio condutor de todas as alterações é o avanço constante do Pervasive PSQL na busca pelo máximo desempenho possível com o hardware e os sistemas operacionais utilizados por nossos clientes e a certeza de que novas plataformas com desempenho superior possam ser adotadas facilmente quando for possível.

APÊNDICE A – CONFIGURAÇÕES DE DESEMPENHO

A tabela seguinte mostra algumas recomendações gerais de configuração que dependem da RAM disponível.

OPÇÕES	PADRÃO	S.O. DE 32 BITS <2GB RAM	S.O. DE 32 BITS DE 2 A 4 GB RAM	S.O. DE 64 BITS DE 2 A 4 GB RAM	S.O. DE 64 BITS >4GB RAM
Cache Allocation Servidor não-dedicado	20% da RAM	20%	20%	20%	20%
Cache Allocation Servidor de banco de dados dedicado	20% da RAM	40%	40% até o máximo de 500MB	40%	Tamanho do conjunto de dados ¹
XIO ²	Ligado	N/A	Ligado	N/A	N/A
Max Microkernel Memory Usage	60% da RAM	60%	0% ³	0%	0% ¹
I/O Threads ⁴	32	32	32	64	64
Allocate Resource at Startup	Desligado	Ligado	Ligado	Ligado	Ligado
Back to Minimal State	Desligado	Desligado	Desligado	Desligado	Desligado
Index Balancing	Desligado	Desligado	Desligado	Desligado	Desligado
Limit Segment Size to 2GB	Ligado	Desligado	Desligado	Desligado	Desligado
Log Buffer Size	1MB	1MB	1MB	1MB	1MB
Transaction Log Size	2MB	2MB	2MB	2MB	2MB
Tracing	Desligado	Desligado	Desligado	Desligado	Desligado
Cache do sistema	Desligado	Desligado	Desligado ⁵	Desligado	Desligado

Notas:

- 1) Em sistemas de 64 bits, pode haver RAM suficiente para manter no cache todo o conjunto de dados utilizado. Apenas certifique-se de deixar RAM suficiente para o sistema operacional. Note, porém, que é necessária a versão v10 Service Pack 1 para utilizar mais que 16GB de cache. Se não houver RAM suficiente para todo o conjunto de dados utilizado, recomendamos ajustar Cache Allocation em 20% da RAM e Max Microkernel Memory Usage em 60%.
- 2) O XIO funciona somente em 32 bits e precisa de pelo menos 2GB de RAM. O padrão do XIO é On.
- 3) Com o XIO, o cache L2 fica desligado automaticamente.
- 4) O sistema operacional Windows aloca 1MB de memória por thread. Para um melhor resultado, utilize um thread para cada 8 arquivos abertos.
- 5) Se o XIO não estiver instalado ou estiver desabilitado, ligue o cache do sistema.

Outra opção importante no ajuste de desempenho é File Growth Factor, que tem como base o tamanho total do banco de dados. A seguinte tabela mostra recomendações que servem de ponto de partida para a configuração da opção File Growth Factor:

TAMANHO DO BANCO DE DADOS	FILE GROWTH FACTOR
Até 1GB	5%
De 1GB a 5GB	2%
10GB ou mais	1%

Para um exame mais detalhado dos ajustes de desempenho do Pervasive PSQL, consulte o Capítulo 5 do Guia de Operações Avançadas do Pervasive PSQL v10.

APÊNDICE B — COMPARAÇÃO ENTRE CACHE ESTÁTICO E CACHE DINÂMICO

Com base em um conjunto de medições de velocidade, determinamos que o tempo médio de leitura de uma página do cache L1 é de aproximadamente 110µs. As tarefas de gerenciamento do cache L2 consomem em média mais 210µs, resultando em 320µs de tempo total por página. O custo de leitura do disco dessa mesma página é de 1900µs, ou cerca de 18 vezes maior que o custo de leitura de uma página do cache. Naturalmente, estes tempos valem para uma configuração específica de hardware e sistema operacional e não devem ser generalizados, mas servem para ilustrar a diferença entre o cache dinâmico e o cache estático usado em versões anteriores do Pervasive PSQL.

Com a utilização do cache estático, considere que 20% das leituras serão atendidas pelo cache do Pervasive PSQL (L1), idealmente 20% pelo cache do sistema e 60% exigirão I/O de disco. No caso ideal em que o cache L1 e o cache do sistema não contenham dados redundantes e o acesso ao cache do sistema não tenha um custo extra, o tempo de 1000 leituras seria a soma do tempo de acesso ao cache ($400 \times 110\mu\text{s}$) = 0,044s e do tempo de acesso ao disco ($600 \times 1900\mu\text{s}$) = 1,14s, ou 1,184s. Na verdade, o cache L1 e o cache do sistema provavelmente teriam bastante redundância, de modo que apenas 20% dos dados estariam no cache e o tempo de acesso seria de $(200 \times 110\mu\text{s}) + (800 \times 1900\mu\text{s}) = 1,542\text{s}$.

Com o cache dinâmico, temos 20% de probabilidade de que uma leitura seja atendida por L1, 50% de que seja atendida por L2 e 30% de que seja necessário I/O de disco. Assim, 1000 leituras levariam $(200 \times 110\mu\text{s}) + (500 \times 320\mu\text{s}) + (300 \times 1900\mu\text{s}) = 0,022\text{s} + 0,16\text{s} + 0,57\text{s} = 0,7522\text{s}$, um desempenho entre 1,5 e 2 vezes melhor que o do cache estático.

Informações para contato

Brasil

Latin America Distributor

Stern Software
www.stern.com.br
info@stern.com.br

Estados Unidos

Pervasive Software Inc.
www.pervasive.com
info@pervasive.com

©2008 Pervasive Software Inc. Todos os direitos reservados. Todas as marcas e nomes de produtos Pervasive são marcas registradas ou comerciais de Pervasive Software Inc. nos Estados Unidos e em outros países. Todas as outras marcas pertencem a seus respectivos proprietários.