STERN SOFTWARE

O Dilema Multinúcleo

julho de 2010

CITO Research
Tell Us a Question.

Conteúdo

Introdução

Compre um sistema novo e melhore o desempenho. De notebooks a servidores, essa ideia sempre foi um axioma. É verdade que a insatisfação com o desempenho muitas vezes determina as compras de hardware.

Mas e os processadores multinúcleo? Esses processadores, utilizados há muito tempo em servidores de topo de linha, chegaram ao território dos servidores PC e dos desktops. O problema é que os compradores corporativos das pequenas e médias empresas continuam seguindo a regra que funcionava no passado: “compre um sistema novo e melhore o desempenho”. Mas a relação entre desempenho de aplicativos e processadores multinúcleo é muito mais complexa e abrange várias camadas do aplicativo e da pilha de software do sistema operacional. Para certos aplicativos de negócios, os processadores multinúcleo não melhoram o desempenho; pior ainda, esses processadores podem até deixar certos tipos de aplicativo mais lentos.

A CITO Research tomou conhecimento desse problema e estudou formas de resolvê-lo. Este artigo explica como os processadores multinúcleo afetam negativamente o desempenho de certos aplicativos de negócios e quais são as opções disponíveis para que os desenvolvedores desses aplicativos atenuem o problema.

Alguns temas discutidos neste artigo são:

- O que são processadores multinúcleo?
- Por que eles exigem mudanças na arquitetura?
- Por que certos aplicativos ficam mais lentos em processadores multinúcleo?
- E os aplicativos que ficam mais rápidos em processadores multinúcleo?
- O que são aplicativos multithreaded e paralelos?
- Por que os aplicativos multithreaded nem sempre ficam mais rápidos em processadores multinúcleo?
- Como os processadores multinúcleo afetam o trabalho dos desenvolvedores?
- Que opções os desenvolvedores têm para melhorar o desempenho dos aplicativos em processadores multinúcleo?

O contexto deste artigo

Dan Woods, chefe de tecnologia da CITO Research, começou a escrever sobre o impacto no desempenho dos processadores multinúcleo e sobre as maneiras de acelerar o processamento de grandes quantidades de dados em sua coluna na Forbes.com em 2009. A Pervasive está patrocinando este artigo Explainer para aumentar a conscientização sobre o impacto potencial no desempenho dos processadores multinúcleo. A última seção deste artigo contém uma análise da Pervasive sobre esse problema.

Desenvolvedores e hardware: Perspectiva histórica

No princípio, os programadores criavam software utilizando instruções de nível de hardware, escrevendo diretamente em código de montagem. A arquitetura do processador estava sempre na mente do programador. Os sistemas operacionais e as linguagens de alto nível aliviaram essa tarefa. Os desenvolvedores passaram a utilizar linguagens como Basic, COBOL, C, Pascal e Fortran para escrever seu código que era submetido a um compilador, que por sua vez traduzia o código para instruções de máquina.

A programação paralela ainda era uma atividade especializada. Embora os programadores que criavam sistemas operacionais tivessem que trabalhar com programação paralela, os programadores de aplicativos não precisavam se preocupar em tornar paralelas as principais áreas de seus aplicativos.

Hoje, o apetite por computação rápida é insaciável. Os fabricantes de processadores como AMD e Intel atingiram o limite das velocidades de clock (o aumento da velocidade basicamente aqueceria tanto os processadores que eles queimariam). A inclusão de vários núcleos no mesmo chip tornou-se a maneira mais fácil de aumentar o poder de processamento. Isso leva diretamente à nossa questão central: se o aumento do clock sempre aumentou a velocidade do software, o aumento do número de núcleos terá o mesmo resultado?

A relação complexa entre multinúcleos e desempenho

Os usuários de computadores multinúcleo percebem que eles conseguem rodar mais programas ao mesmo tempo e que a mudança entre um programa e outro é mais rápida. Mas o aumento de desempenho dos programas em si não é tão evidente. Para melhorar o desempenho de um aplicativo em um sistema multinúcleo, os desenvolvedores precisam escrever código paralelo que utilize os vários núcleos simultaneamente.

Embora muitos aplicativos sejam multithreaded, a utilização de threads muitas vezes limita-se a áreas completamente isoladas. Por exemplo, as interfaces de usuário muitas vezes são tratadas por um thread separado para melhorar a usabilidade, mas a lógica do aplicativo é implementada por uma sequência de operações que não pode ser facilmente paralelizada. Assim, o primeiro requisito para a melhoria do desempenho em sistemas multinúcleo é que exista algum potencial de paralelização no aplicativo. Isso pode ser difícil de se conseguir nos aplicativos projetados e criados sem qualquer consideração ao processamento paralelo.

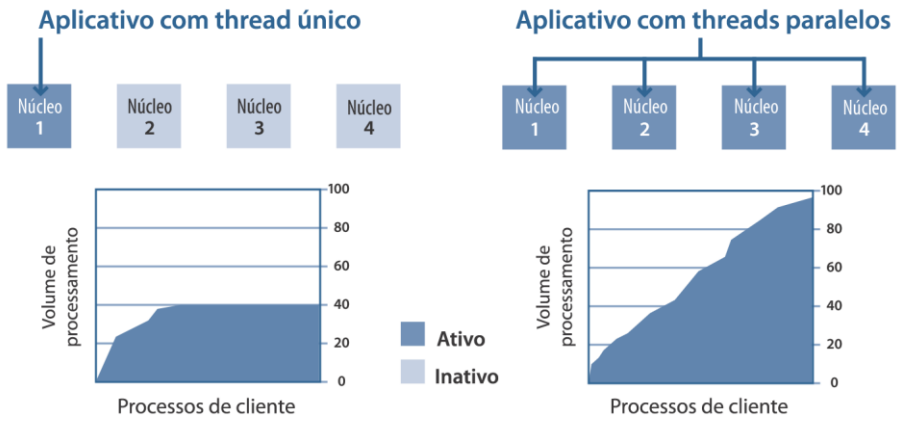

Figura 1: Aplicativos com thread único e com threads paralelos

A segunda barreira é a qualificação profissional necessária. A programação eficaz em processadores multinúcleo é difícil e apresenta novos desafios mesmo para os programadores com experiência em programação multithreaded.

Os desenvolvedores de aplicativos têm as seguintes opções:

- Modificar a arquitetura do código para atender à necessidade de paralelismo utilizando ferramentas e ambientes de desenvolvimento criados com essa finalidade.
- Substituir certos componentes do aplicativo, como o banco de dados, quando isso ajudar a mitigar o problema.

Essas opções não são mutuamente exclusivas. Para obter o máximo desempenho no mundo multinúcleo, os desenvolvedores terão que modificar a arquitetura de seus aplicativos para adequá-los a esses processadores. A substituição de elementos do aplicativo como o banco de dados, quando viável, pode melhorar o desempenho e ganhar tempo para que os desenvolvedores reescrevam seus aplicativos para adequá-los ao hardware multinúcleo.

Um dos desafios para a análise do dilema multinúcleo é que é difícil oferecer um conselho geral. Para avaliar a urgência da reformulação de determinado aplicativo ou parte dele, temos que entender as várias dimensões do aplicativo, da arquitetura do microprocessador e do sistema operacional:

- Os processadores multinúcleo não são idênticos. Os dois principais fabricantes de microprocessadores, Intel e AMD, abordam de maneiras diferentes o projeto de processadores multinúcleo. A arquitetura do chip multinúcleo pode afetar a escolha do processador. Determinados aplicativos podem rodar melhor em processadores Intel, enquanto outros podem rodar melhor em processadores AMD.
- Diferentes versões do mesmo sistema operacional diferem em sua capacidade de suportar threads simultâneos. Por exemplo, o escalonador de tarefas Windows Server 2008, presente tanto no Windows Vista quanto no Windows Server 2008, sofreu uma melhoria significativa.
- Os aplicativos podem ficar mais rápidos ou mais lentos em processadores multinúcleo dependendo da sua forma de operação.

É necessário manter em mente essas dimensões do problema à medida que conhecemos mais detalhes do dilema multinúcleo.

Quando os multinúcleos aumentam o desempenho

Em determinadas situações, o processamento multinúcleo pode aumentar dramaticamente o desempenho. Considere a execução simultânea de vários aplicativos. O impacto dos processadores multinúcleo é claro e imediato quando percebemos que é possível jogar um videogame enquanto um projeto de vídeo está sendo codificado no mesmo sistema desktop.

Alguns aplicativos respondem bem ao processamento multinúcleo. Os aplicativos cujo trabalho pode ser dividido naturalmente em várias partes que podem ser executadas em paralelo e agregadas ao final da operação podem se beneficiar do processamento multinúcleo. Note porém que o aplicativo precisa ser reformulado para rodar dessa forma, talvez com o auxílio de ferramentas para tornar paralelo o trabalho do aplicativo. Tarefas que exigem muito do processador, como o tratamento de áudio e vídeo, aplicativos de modelagem financeira e científica e renderização de projetos CAD são exemplos desse tipo de aplicativo. Muitos sistemas de computação de alto desempenho e grande volume de processamento levam essa ideia ao limite extremo e dividem tarefas como a renderização de animações de computador em milhares de partes que são executadas por computadores organizados em um gigantesco plantel.

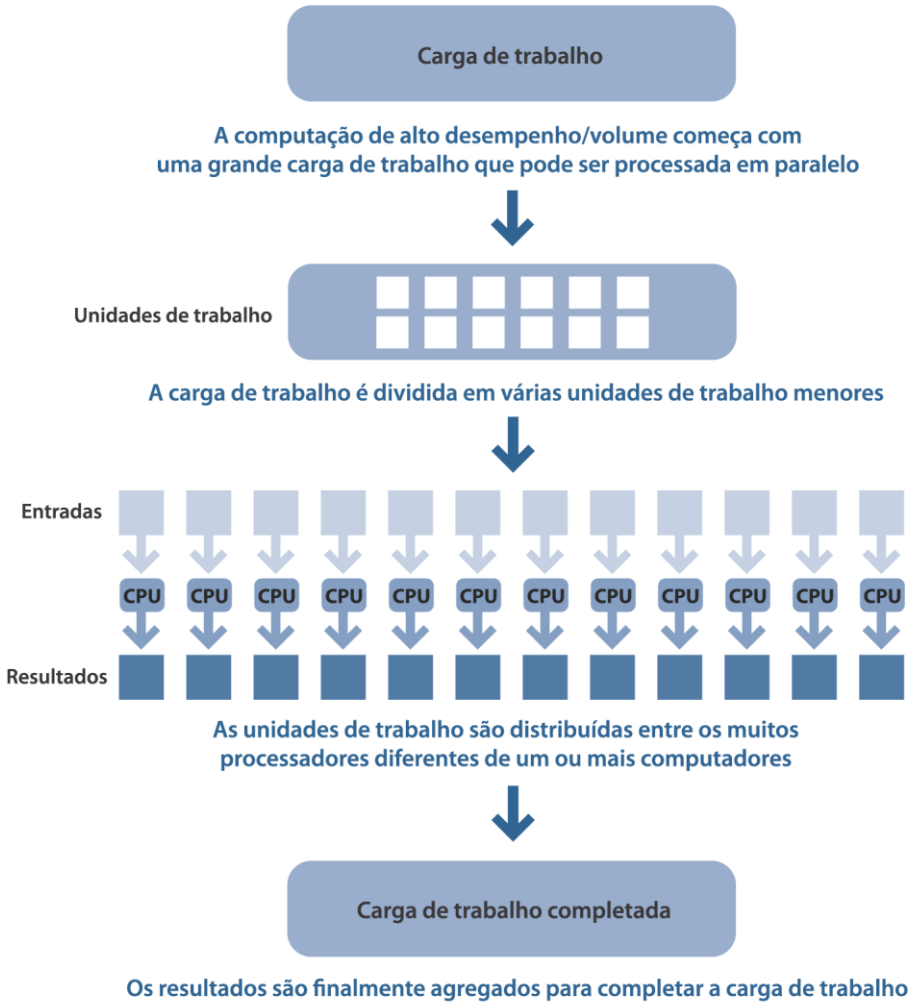

Figura 2: Estrutura básica para computação de alto desempenho e grande volume de processamento

Quando os multinúcleos prejudicam o desempenho

O que não é bem compreendido pela comunidade de desenvolvimento de software é que os processadores multinúcleo podem prejudicar o desempenho nas seguintes situações:

- Quando as operações do aplicativo não podem ser divididas porque é necessário manter muitas informações na memória ao mesmo tempo.
- Quando determinada operação, como acessar o banco de dados, é feita por muitos usuários ao mesmo tempo.
- Quando o aplicativo utiliza banco de dados e depende da estabilidade do estado dos dados para obter resultados.
- Quando o aplicativo compartilha dados.

Nesses casos, os processadores multinúcleo podem reduzir o desempenho do aplicativo.

Exemplos do primeiro caso, de aplicativos cujas operações não podem ser facilmente divididas, são os modelos financeiros dinâmicos nos quais os parâmetros podem ser modificados pelo usuário em tempo de execução (para obter várias versões do modelo, por exemplo) enquanto os dados (como preços de ações ou commodities) continuam chegando em tempo real.

Lojas virtuais têm muitos usuários simultâneos. Em certas épocas do ano, a demanda por determinados produtos é alta. As operações de consulta ao estoque, inclusão na lista de compras e confirmação de pedidos evidentemente competem por recursos específicos, o que pode provocar problemas de desempenho e insatisfação dos clientes. Para esse tipo de aplicativo, o processamento multinúcleo pode ser problemático.

Bancos de dados utilizam muitos threads e muitos processos simultâneos cuja interdependência é complexa, o que significa que alguns processos precisam esperar que outros se completem, dificultando a paralelização. O estado dos dados precisa ser administrado cuidadosamente para assegurar a integridade transacional. Aplicativos complexos e multiusuários muitas vezes utilizam bancos de dados. Esses aplicativos estão sujeitos à diminuição do desempenho em processadores multinúcleo.

Características do aplicativo	Desempenho provável em processadores multinúcleo sem modificação do aplicativo
Vários aplicativos rodando ao mesmo tempo.	Melhor
Pode ser dividido em várias etapas independentes (jogos, simulação, modelagem, renderização). Nessas situações, os dados são estáticos e o processo pode ser completamente executado sem interrupção nem atualização.	Igual ou melhor
Utiliza um banco de dados em que a integridade transacional deve ser mantida e tem muitos usuários.	Pior
Compartilha dados e tem muitos usuários.	Pior

Tabela 1: Características e desempenho de aplicativos em processadores multinúcleo

Como a maioria dos aplicativos que compartilham dados são aplicativos de negócios, o impacto provável é o seguinte:

- Usuários de sistemas desktop observam um impacto positivo com os processadores multinúcleo.
- Usuários de pequenas empresas observam um impacto positivo com os processadores multinúcleo quando utilizam aplicativos de produtividade como o Microsoft Office.

- Usuários de pequenas empresas observam uma degradação significativa do desempenho de seus aplicativos de negócios com os processadores multinúcleo.

Os usuários de pequenas e médias empresas, por sua vez, manifestam sua insatisfação a seus fornecedores de software e exigem soluções para seus problemas. Uma análise das chamadas de suporte aos fornecedores de software nos últimos dois anos revelou um padrão. Aplicativos que rodavam bem sofreram uma degradação do desempenho quando o hardware foi atualizado para multinúcleo. Assim, uma atualização de hardware no cliente transformou-se em um problema de suporte no fornecedor de software.

O desafio multinúcleo

Agora que conhecemos o problema em termos gerais, podemos entender como, pela primeira vez, uma mudança importante do hardware do PC representa um problema para o setor de software. Os processadores multinúcleo tornam certos aplicativos mais rápidos, mas podem deixar outros mais lentos. As metodologias de desenvolvimento, depuração e teste também mudaram fundamentalmente. Vamos descobrir como essa redução de desempenho acontece e como ela pode ser evitada.

Os processadores multinúcleo também oferecem uma oportunidade: se ajustados corretamente, os aplicativos que de outra forma perderiam desempenho podem utilizar melhor os processadores multinúcleo e apresentar um aumento significativo de desempenho. Em alguns casos, a substituição de certas partes do aplicativo como o banco de dados pode resolver o problema sem que os desenvolvedores precisem fazer alterações imediatas. Os desenvolvedores devem considerar essa abordagem como uma opção de baixo risco para ganhar tempo até que seja necessário reescrever os aplicativos. A figura 3 mostra as três opções.

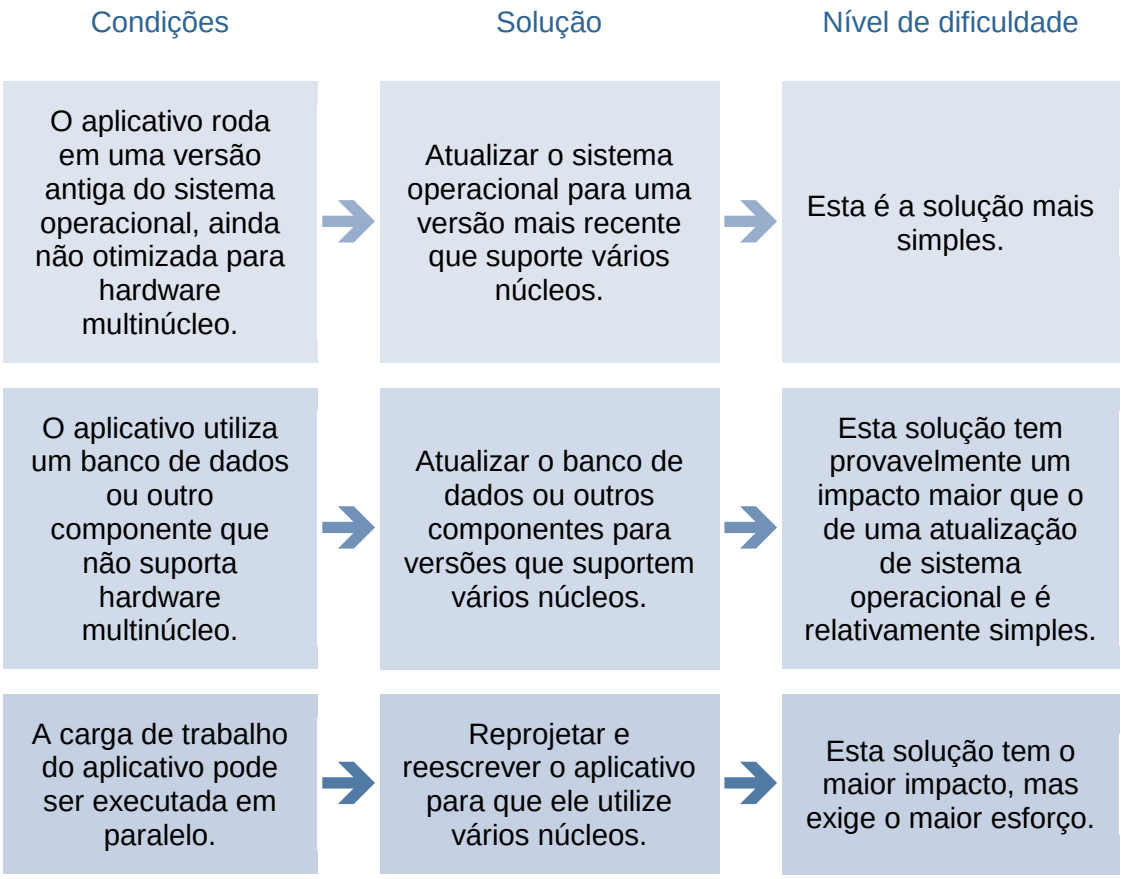

Figura 3: Matriz de decisão para otimizar aplicativos para processamento paralelo

Por que aplicativos multithreaded ficam mais lentos em multinúcleos

À primeira vista, pode parecer pouco intuitivo que um aplicativo multithreaded rode mais lentamente em sistemas multinúcleo. Os motivos para que isso aconteça são complexos e merecem uma explicação mais detalhada.

Nos aplicativos multithreaded que compartilham dados, a sincronização entre threads consome muitos recursos do sistema em máquinas multinúcleo. O compartilhamento de dados é o flagelo da computação paralela. O acesso de vários threads aos mesmos dados precisa ser sincronizado. Quando sincronizamos o acesso aos dados, a parte do código que faz o acesso não pode ser executada por mais de um thread de cada vez, por isso deixa de ser concorrente. Essa parte do código torna-se uma passagem estreita e todos devem fazer fila para passar por ela.

A utilização de cache piora o problema em vez de amenizá-lo. Quando os caches de vários núcleos ou processadores estão apontando para os mesmos dados e um dos núcleos modifica os dados, os caches dos demais núcleos deixam de ser válidos e precisam ser sincronizados com os novos dados.

O acúmulo de processamento extra devido a todas essas operações de sincronização é significativo. Em última análise, isso significa que o desempenho em processadores multinúcleo pode ser pior que em processadores de núcleo único, que não precisam desse tipo de sincronização.

Sempre que possível, cada núcleo deve trabalhar com seus próprios dados. Caso contrário, a necessidade de sincronização leva ao processamento extra que pode reduzir significativamente o desempenho.

Problemas comuns em aplicativos multithreaded

Outras dificuldades enfrentadas pelos programadores nessa transição são os eternos problemas da programação paralela que se manifestam em sistemas multinúcleo, como:

- O estouro da manada
- Compartilhamento falso
- Competição por memória

Vamos examinar esses problemas.

O estouro da manada

O problema do estouro da manada ocorre quando o escalonador ativa todos os threads que tenham a possibilidade de atender determinado evento (uma conexão Web, por exemplo) para determinar qual deles deve prosseguir, depois desativa novamente todos os demais threads cuja vez de rodar ainda não chegou.

É como se o alarme de incêndio tocasse no quartel dos bombeiros sempre que o processador termina uma tarefa e fica pronto para começar outra. Os threads que estão em espera são como os bombeiros que estão dormindo em seus leitos. O processo de acordar todos os bombeiros para descobrir quais precisam se vestir e subir no caminhão para combater o incêndio não é nada eficiente. Esse processo acaba dificultando a execução do serviço e prejudica o desempenho, principalmente se o número de threads for grande. O resultado costuma ser chamado de *thrashing*.

Em sistemas multinúcleo, o escalonador precisa lidar com mais processadores, por isso o problema do estouro da manada normalmente presente nos aplicativos multithreaded torna-se ainda mais sério.

Compartilhamento falso

Uma das vantagens da programação multithreaded é que várias tarefas podem ser executadas ao mesmo tempo. Para que essas tarefas funcionem como se espera, porém, seu acesso às linhas de cache para obter dados da memória é controlado segundo sua prioridade. Dados diferentes em posições diferentes da memória podem ser acessíveis por meio da mesma linha de cache. Assim, mesmo que cada thread esteja acessando dados diferentes, não compartilhados, eles podem acabar competindo pela mesmas linhas de cache, tomando posse dos recursos alternadamente em uma longa batalha. Para utilizar eficientemente o processamento multinúcleo, os aplicativos devem ser escritos de modo que esse tipo de luta pelas linhas de cache não ocorra.

Competição por memória

Programas escritos para funcionar em ambientes multinúcleo precisam levar em conta os processos simultâneos que buscam informações na memória. Programas que não foram otimizados para o funcionamento em ambientes multinúcleo (que, vamos admitir, são maioria) acessam informações serialmente, por isso podem cair em uma situação chamada "competição por memória", em que os processadores verificam repetidamente o cache para garantir que eles têm dados atualizados. A finalidade desse procedimento de verificação é impedir que o thread de determinado processador utilize dados que foram desatualizados pelos dados produzidos por outro thread em outro processador. Embora o procedimento evite erros, ele também deixa a máquina consideravelmente mais lenta, na medida em que o bus de memória coloca novas requisições em espera.

Esse problema não pode ser resolvido pelo aumento do número de processadores. Ao contrário, a presença de mais processadores faz com que mais requisições sejam feitas à memória para atualizar os caches. Quanto maior o número de caches a serem verificados, maior o tráfego. Mesmo sem cache, os vários threads podem entrar em competição pelo acesso à memória central.

Os aplicativos multithreaded precisam ser reprojetados para dividir bloqueios gerais em bloqueios com mais granularidade, balancear a carga de processamento e reduzir a competição entre threads por recursos compartilhados como a memória.

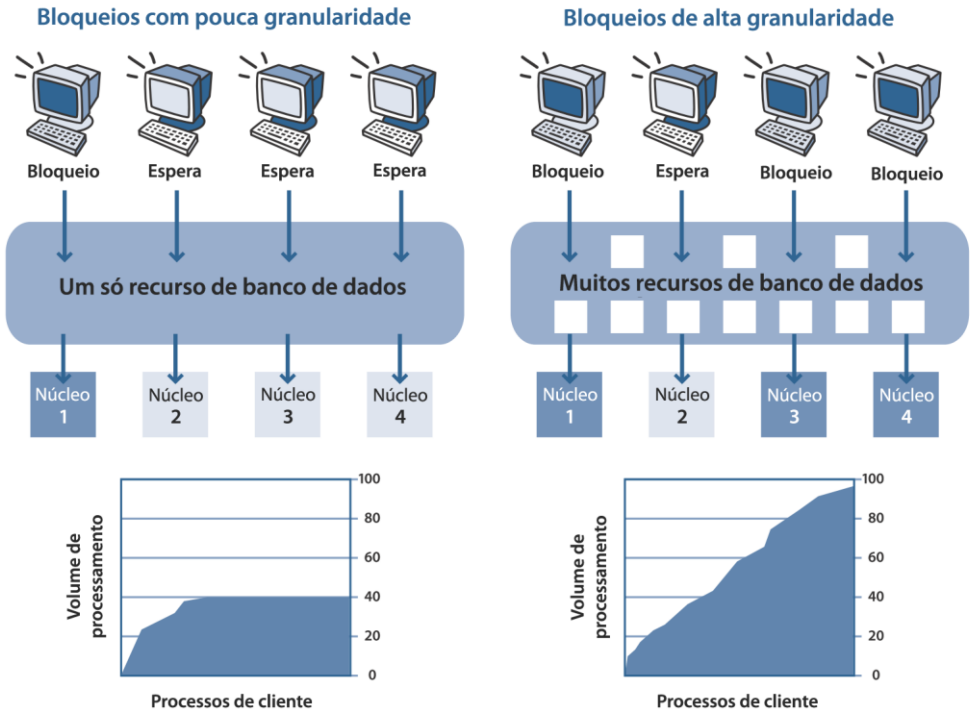

Figura 4: A granularização de bloqueios gerais aumenta o volume de processamento

O papel do sistema operacional

O sistema operacional é responsável pela resolução das situações de competição por recursos. Na experiência dos desenvolvedores consultados pela CITO Research, os sistemas operacionais também demoram mais para resolver as situações de competição entre threads nos sistemas multinúcleo. Ou seja, os sistemas operacionais de máquinas multinúcleo levam mais tempo para resolver os pontos de competição. Isso acontece mesmo no Windows Server 2008 e no Windows Vista. Nas versões mais antigas do Windows, a lentidão em sistemas multinúcleo é ainda mais pronunciada.

Isso derruba o mito do sistema operacional multinúcleo. A instalação de um sistema operacional otimizado para máquinas multinúcleo não ajuda a solucionar o seu problema se os seus aplicativos continuam a fazer requisições em fila indiana ao sistema operacional.

Imagine que o seu sistema operacional é um guarda de trânsito em um cruzamento movimentado. Todos os veículos (threads) querem atravessar o cruzamento imediatamente, mas estão aguardando instruções do guarda. Imagine se, em vez de avaliar de forma independente qual faixa de tráfego deve ser liberada, o guarda consultasse cada um dos motoristas para saber se ele está pronto para seguir viagem para depois liberá-lo ou não. O congestionamento ficaria pior do que se o guarda não estivesse presente. Essencialmente, é esse o tipo de “congestionamento” que ocorre quando o sistema operacional recebe requisições de um aplicativo que não contém instruções para processamento multinúcleo. Não é surpresa que o congestionamento no nível do sistema operacional seja percebido pelo usuário como um problema de desempenho do aplicativo.

Aplicativos multinúcleo fornecem instruções que permitem que o sistema operacional administre os recursos compartilhados e determine a prioridade do acesso a esses recursos. É isso que muitos fornecedores de software querem dizer quando dizem que seus aplicativos têm uma implementação sem bloqueios. As requisições de informação são organizadas de forma que não entrem em competição por linhas de cache ou por acesso à memória central.

Opções para os desenvolvedores

O que os fornecedores de software podem fazer diante de todos esses problemas?

Os aplicativos terão que ser reprojetados um dia para rodar com a máxima eficiência em máquinas multinúcleo. Estão surgindo ferramentas que facilitam essa tarefa, como a biblioteca Parallel FX da Microsoft, que inclui PLINQ (que está sendo incorporado ao Visual Studio 2010), Intel Parallel Studio e Open MPI. Essas ferramentas oferecem aos desenvolvedores uma estrutura preparada para o processamento paralelo sobre a qual seus aplicativos podem ser desenvolvidos. Novas linguagens como Google Go também estão sendo desenvolvidas.

A realidade é que o processo vai demorar anos, em certos casos. Cada fornecedor de software deve determinar a melhor maneira de enfrentar esse desafio para evitar um impacto negativo para seus clientes.

Em alguns casos, é possível substituir certos componentes do aplicativo, como o banco de dados. Essa é uma opção simples para enfrentar o dilema multinúcleo e ganhar tempo para modificar o seu aplicativo para que ele utilize os poderosos sistemas multinúcleo, que certamente vieram para ficar. Se você utiliza o Pervasive PSQL ou outro tipo de sistema de armazenamento de dados que possa ser facilmente convertido para PSQL (como o ISAM), a migração para a versão otimizada para multinúcleos do PSQL pode aumentar o desempenho do aplicativo sem a necessidade de recompilação ou conversão de arquivos, na maioria dos casos.

Sobre o Pervasive PSQL™ v11

A Pervasive patrocinou este artigo Explainer da CTO Research porque a empresa considera que o mercado não está suficientemente bem informado sobre esse problema e porque ela tem uma solução para um determinado tipo de fornecedor de software.

A Pervasive está respondendo ao dilema multinúcleo concentrando esforços na otimização do seu sistema de banco de dados, o Pervasive PSQL v11, para utilizar os sistemas multinúcleo com a máxima eficiência.

A versão mais recente do Pervasive PSQL foi projetada para gerar threads paralelos que executam atividades semelhantes, permitindo que o banco de dados utilize vários núcleos durante a execução de tarefas. O resultado é que o desempenho multiusuário do sistema de banco de dados melhora nos ambientes multinúcleo. Além da melhoria significativa da operação multinúcleo, o desempenho do banco de dados melhora com o aumento do número de núcleos. Nos testes da Pervasive feitos em instalações de clientes, o aumento típico de desempenho em relação ao Pervasive PSQL v10 foi de 50%, com alguns aplicativos chegando a 400%.

Em resumo, estas são as características do Pervasive PSQL v11 que podem ajudá-lo a enfrentar imediatamente o dilema multinúcleo:

- O PSQL v11 tem uma implementação verdadeiramente paralela que é capaz de utilizar os processadores multinúcleo e acelerar dramaticamente a maioria dos aplicativos multiusuário.
- O Pervasive PSQL v11 apresenta vários aprimoramentos nos mecanismos de sincronização de baixo nível da interface transacional. Vários usuários podem ler as mesmas páginas de arquivo em cache ao mesmo tempo e suas operações podem ser executadas em núcleos independentes.
- Outras atividades, como gerenciamento de checkpoints e logs também podem utilizar núcleos extras.
- Muitos gargalos do sistema de banco de dados foram removidos ou atenuados. Por exemplo, vários usuários acessando arquivos independentes podem ser atendidos por núcleos independentes. O sistema de banco de dados também pode suportar um número maior de usuários com menos sobrecarga, resultando em um maior volume de processamento.
- Não é necessário recompilar os aplicativos. Os aplicativos Pervasive PSQL existentes continuam a funcionar no PSQL v11 sem alterações.

A adoção do PSQL melhora imediatamente o desempenho multiusuário em sistemas multinúcleo, seja evitando a lentidão, seja adiando a necessidade de criar uma versão paralelizada do aplicativo.

Outros motivos para adotar o PSQL v11 também precisam ser mencionados:

- Trata-se de um banco de dados embutido extremamente rápido.
- Ele oferece um excelente desempenho ISAM e gerenciamento de I/O de disco.
- Ele é provavelmente o banco de dados ideal para aplicativos COBOL que necessitem de um aumento de desempenho.
- Ele inclui um sistema SQL de 64 bits, além de provedores ODBC e ADO.NET de 64 bits.
- Ele pode ser utilizado por todos os aplicativos baseados em Pervasive PSQL ou Btrieve.

Para os desenvolvedores cujos aplicativos utilizem o banco de dados Pervasive PSQL, a atualização para o PSQL v11 pode resolver o problema do desempenho multinúcleo pela simples substituição da camada de banco de dados. Essa opção também pode ser escolhida para os aplicativos baseados em Btrieve e para os aplicativos escritos em COBOL que utilizam bancos de dados ISAM (no fundo, o PSQL é um ISAM muito rápido). A atualização é uma excelente opção para os fornecedores de software que utilizam um dos métodos de acesso a banco de dados suportados pelo PSQL, como ADO.NET, ODBC, JDBC, JCL, PDAC, OLE DB e ActiveX.

Conclusão

Pela primeira vez na história da computação, um hardware mais novo deixa os aplicativos mais lentos. Alguns aplicativos ficam apenas um pouco mais lentos, mas os aplicativos multiusuário que controlam recursos compartilhados e nos quais o gerenciamento de estados é muito importante (transações, por exemplo) podem ficar significativamente mais lentos.

A paralelização de aplicativos é uma tarefa difícil, demorada e arriscada. A programação, a depuração e os testes são difíceis. Mesmo que a sua equipe sempre tenha cumprido os prazos dos ciclos de atualização, isso pode mudar quando ela tiver que enfrentar os muitos espinhos da rosa multinúcleo. O problema é amplificado quando você tem que gerenciar não só o conflito entre threads, mas o conflito entre threads em vários núcleos, cada um dos quais com seu próprio grupo de threads. Se um programa multithreaded tiver qualquer ponto fraco ou ineficiência, esses problemas são amplificados nos sistemas multinúcleo. O gerenciamento de recursos feito pelo sistema operacional não é adequado para permitir que a maioria dos aplicativos se beneficiem dos múltiplos núcleos.

Porém, como vimos, é possível substituir componentes do aplicativo, como o banco de dados. Essa é uma opção simples para enfrentar o dilema multinúcleo e ganhar tempo para trabalhar no reprojeto do seu aplicativo para que ele se beneficie dos poderosos sistemas multinúcleo que já se tornaram metas padrão de atualização de hardware. Se você utiliza o Pervasive PSQL ou um sistema de dados que possa ser facilmente migrado para PSQL, a utilização da versão mais recente do PSQL pode aumentar o desempenho do seu aplicativo e evitar a lentidão multinúcleo.