

Vantagens e Desvantagens do Terminal Service como Servidor de Aplicações em Ambientes Windows e Linux

Anderson Mapa da Silva¹, Livia Márcia Silva¹

¹Departamento de Ciência da Computação – Universidade Presidente Antônio Carlos (UNIPAC)

Campus Magnus – Barbacena – MG – Brasil

a.mapa@hotmail.com, livimarcia@yahoo.com.br

Abstract: *This article aims to describe the technology and terminal services all major operating systems used for the application of lightweight terminals. This technology is to make client stations, with limited hardware resources, connect to a server application and run your applications using server resources, making the programs work on the server and the client are shown. Therefore, this paper presents a case study where two operating systems are evaluated, with the aim of showing the best cost benefit.*

Resumo: Este artigo tem como objetivo descrever a tecnologia *terminal services* e os principais sistemas operacionais utilizados para a aplicação de terminais leves. Esta tecnologia consiste em fazer de estações clientes, com poucos recursos de hardware conectarem num servidor de aplicação e executar seus aplicativos, usando os recursos do servidor, fazendo com que os programas trabalhem no servidor e sejam mostrados no cliente. Com isso, este trabalho apresenta um estudo de casos onde são avaliados dois sistemas operacionais, com o objetivo de mostrar o melhor custo benefício.

1. Introdução

O *Terminal Services* é um recurso que os sistemas operacionais modelo cliente/servidor dispõem ou que pode ser aplicado para que estações clientes, com poucos recursos de hardware possam conectar em um servidor de aplicação e executar seus aplicativos, neste modelo os recursos utilizados são do servidor, ou seja, os programas são executados no servidor e mostrados no cliente. *Terminal Services* dispõe de funcionalidades similares aos mainframes: um ambiente no qual múltiplos terminais são conectados a um computador. Cada terminal provê um canal para entrada e saída entre um usuário e o computador central. Um usuário pode se conectar no terminal, e então executar aplicações no PC central, acessando arquivos, bancos de dados, recursos de rede, e assim por diante. Cada sessão de terminal é independente, com o sistema operacional gerenciando conflitos entre múltiplos usuários competindo por recursos compartilhados.

A diferença primária entre o *Terminal Services* e o tradicional ambiente *mainframe* é que os terminais burros do *mainframe* são somente baseados na entrada e saída de caracteres. Enquanto os *Terminais Services* dispõem de uma completa área de interface gráfica que interage com o usuário. As aplicações não são executadas localmente, mas são exibidas nos terminais.

1.1. Justificativa

Em um ambiente de rede tipo cliente/servidor o uso do *Terminal Services* otimiza de certos problemas tais como:

1. Licença de softwares – redução na aquisição de licenças, uma vez que o aplicativo necessita de ser instalado somente no servidor;
2. Informações centralizadas – todos os recursos, aplicativos e acessos a banco de dados são alocados no servidor;
3. Segurança – não são trafegados na rede pacotes contendo informações que não podem ser disponibilizada a pessoas não autorizadas, mas somente informações de vídeo, teclado e mouse;
4. Suporte técnico – uma vez que todos o recurso são disponibilizados e executados no próprio servidor, toda a manutenção se faz necessária somente no servidor.

1.2. Objetivo

As vantagens e desvantagens do uso do *Terminal Services* como servidor de aplicações em ambientes Windows e Linux, e ao mesmo tempo oferecer uma alternativa com um melhor custo-benefício para a redução de investimentos em upgrade de equipamentos e suporte técnico.

2. Como Funciona o *Terminal Services*?

Quando um usuário inicia uma conexão de *Terminal Services*, uma sessão é iniciada no servidor de terminais, sendo uma para cada usuário. Cada sessão é determinada por uma única identificação, e todas as conexões são tratadas e identificadas separadamente. Cada sessão é associada a uma janela interativa, a qual interage com o usuário. Quando o usuário se desconecta do servidor, a sessão é encerrada fechando a janela interativa com o usuário.

Uma vez inicializada a sessão, o terminal envia as ações de teclado e *mouse* executadas localmente pelo usuário, através do protocolo de comunicação, para o servidor e exibe os resultados obtidos na janela do terminal.

2.1. Porque se importar com o *Terminal Services*?

Por que se interessar pela informática com base em servidores? Há três razões para isso: a primeira é que pode-se utilizar hardware inferior e quebrar o ciclo de atualizações de equipamentos. A segunda é poder evitar que usuários configurem errados os computadores, e por fim pode-se instalar aplicativos em um local central em vez de fazê-lo em cada PC cliente. Em geral, o *Terminal Services* permite que se manipule o servidor de outro console, o que se torna conveniente.

2.2. Menor Potência de Processamento é Exigida pelo Cliente

O problema está em realmente precisar de computadores mais potentes para se manter atualizado com a tecnologia de software existente. Muitas pessoas utilizam o computador como uma ferramenta para que o seu trabalho seja feito e não é necessário o computador mais rápido do mundo para processar textos. O *Terminal Services* permite

que um computador de bom desempenho trabalhe para muitas pessoas e não apenas para uma.

Na prática, podemos instalar um cliente de terminal num micro 486, e utilizar todos os novos aplicativos no *Terminal Services*. Não é preciso dedicar a máquina a ele, o computador pode ser utilizado normalmente, e ter apenas um atalho para *Terminal Services*.

Existem dois tipos de clientes os *thin client* (clientes leves) e *fat client* (clientes pesados), onde o primeiro é geralmente um computador sem disco rígido, utilizado em ambientes totalmente centralizados e o segundo é um micro completo, com disco rígido, com sistema operacional instalado.

2.3. Distribuição Centralizada de Aplicativos

Um dos grandes benefícios do *terminal services* é como ele simplifica a distribuição de aplicativos. Instalar aplicativos em um servidor de terminais significa suporte para muitos usuários com uma única licença de software, assim não é necessário instalar aplicativos localmente, desenvolver scripts de instalação isolados para um conjunto de programas e se preocupar com a largura da banda ou com recursos disponíveis localmente quando forem acrescentados os aplicativos ao computador. Ao instalar uma única vez o aplicativo todos que deveriam ter acesso a ele o terão.

2.4. Administração Remota do Servidor

O *Terminal Services* não é apenas conveniente para a base do usuário, também é conveniente para o administrador da rede. Pode ser utilizado como uma ferramenta de acesso remoto, permitindo editar as configurações do servidor de qualquer lugar na rede, ou mesmo pela Internet.

3. Terminal Services do Windows Server

O *Terminal Server* do Windows (WTS) transforma uma máquina em um tipo de *mainframe*. Tudo o que a máquina do usuário irá fazer é proporcionar pressionamentos de teclas e cliques com o mouse e estará recebendo imagens gráficas da área de trabalho. Todos os dados e toda a computação estarão localizados de forma centralizada no servidor Windows 2000.

O suporte para o *Terminal Services* no *Windows Server* não é novo. A novidade sobre o suporte multiusuário do *Windows Server* é ter pela primeira vez o suporte para computadores com base em servidores como uma parte essencial da base do sistema operacional Windows.

3.1. Configuração de Hardware do Cliente

De acordo com Minasi, **Um terminal típico deve ter uma CPU de 200 Mhz e 32 MB de memória instalada.** Nem todos os terminais correspondem com essa definição. Existem empresas que vendem CPU's próprias para serem utilizadas como clientes de *Terminal Services*.

3.2. Configuração de *Hardware* do Servidor

Não existe uma configuração típica para um servidor de terminais, cada ambiente requer uma configuração diferente. Isso varia das aplicações instaladas e quantidade de usuários conectados.

A Microsoft recomenda no mínimo a seguinte configuração para processadores de 32 e 64 bits: Velocidade de CPU de 733MHz, 256MB de RAM e 1,5GB de espaço em disco para a instalação. Outra informação importante é que as versões de 64 bits do *Windows Server 2003, Enterprise Edition* e do *Windows Server 2003, Datacenter Edition* são compatíveis somente com sistemas baseados no processador Itanium de 64 bits da Intel. Elas não podem ser instaladas em sistemas de 32 bits.

3.3. Vantagens e Desvantagens do *Terminal Services* do Windows

Neste tópico, são apresentadas as principais vantagens e desvantagens que envolvem a tecnologia *terminal services* no sistema operacional Windows.

1. Com o uso do *terminal services*, reduz-se ou mesmo anula-se o custo de atualização de hardware para os clientes.
2. Os aplicativos rodam somente no servidor e são mostrados nos clientes ou terminais. Sendo assim, micros antigos podem executar programas que aparentemente não funcionariam num 486, por exemplo.
3. Distribuição centralizada de aplicativos faz com que os programas sejam executados somente no servidor, com isso, não a necessidade de instalar os programas nos terminais.
4. O gerenciamento é feito no servidor de terminais, configurado uma vez nele, todos os demais clientes terão as configurações automaticamente. Isso garante que todas as estações serão idênticas ao servidor.
5. Administrações remotas do servidor, podendo-se gerenciar um servidor remotamente de qualquer cliente, através da rede ou mesmo pela Internet.
6. O *terminal services* é nativo no *Windows Server*, podendo ser ativado e desativado conforme suas necessidades.
7. Usuário familiarizado com softwares da plataforma Windows, tem como principal desvantagem do *terminal services* do Windows a questão de licenciamento. Que funciona da seguinte forma: Cada cópia instalada do software servidor requer a aquisição de uma licença de Servidor Windows;
8. Uma Licença de Acesso de Cliente (CAL) Windows é necessária para utilizar o software servidor;
9. Uma Licença de Acesso de Cliente de *Terminal Server* é necessária para utilizar os Serviços de terminal, ou então terá de alojar uma sessão remota GUI (*Graphical User Interface*), efetuando o caso de uma sessão de console.

4. *Terminal Services* do Linux Kurumin

O Kurumin é uma distribuição Linux desenvolvida por um brasileiro Carlos E. Morimoto, destinando ao uso em *desktops*. Desta forma, não vem com servidores préinstalados. A única exceção é o servidor NFS, que é usado para compartilhar

arquivos com outros micros. Mas também pode ser usado como servidor, instalando os aplicativos desejados através dos ícones mágicos, que são scripts de instalação e configuração adotados no Kurumin, tornando a tarefa de uma instalação mais amigável para um usuário sem muita experiência.

Os scripts dos ícones mágicos tentam facilitar ao máximo a instalação, explicando passo a passo o que precisa ser feito e ajudando a criar uma configuração que já funcione imediatamente depois da instalação. A ideia básica é permitir que você consiga configurar pequenos servidores em alguns minutos, mesmo sem ter muita informação técnica sobre os programas específicos.

Os scripts para instalar e configurar servidores estão no Iniciar > Configuração do Sistema > Configuração dos Servidores e podem ser acessados também através da seção “Servidores” do Painel de Controle de Kurumin.

Em junho de 2003, Morimoto incluiu um ícone mágico no Kurumin para a instalação do LTSP. Com ele podemos fazer de micros antigos, a partir de 486, sem HD nem CD-ROM, apenas 8 MB de RAM, placa de rede e um drive de disquetes (ou uma placa de rede com um chip EPRON) e criar uma próspera rede de terminais leves, onde os terminais dão *boot* através da rede e exibem as imagens dos aplicativos que estarão rodando num servidor mais rápido.

Segundo Morimoto (2003), o Kurumin foi desenvolvido usando como base dois projetos já existentes, o Knoppix e o Debian. O Knoppix entrou com sistema de detecção de hardware e configuração automática e o Debian com toda a base do sistema, como os pacotes e ferramentas de administração como o apt-get (comando de instalação e desinstalação de aplicativos).

4.1. Configuração de *Hardware* do Cliente

O servidor fica com o trabalho, que é executar os programas e armazenar todos os dados. Enviando para os clientes apenas instruções para montar as janelas que serão exibidas e estes enviam de volta os movimentos do *mouse* e as teclas digitadas no teclado.

Segundo Morimoto (2003), o tempo necessário para um *click* do *mouse* ir da estação até o servidor e este enviar de volta a resposta é mínimo, quase imperceptível. Mas, apesar disso, a estação precisa rodar uma versão compacta do Linux com um servidor X e tem o trabalho de montar as janelas baseado nas instruções recebidas do servidor. Se o processador for muito lento a atualização da tela começará a ficar lenta. Um 486 DX-100 demora cerca de 0.5 segundo para redimensionar uma janela (usando o Xfree 4.2 padrão do LTSP), é relativamente rápido.

Mas um 386 demoraria 2 ou 3 segundos para fazer a mesma tarefa, o que já seria incômodo. O ideal é utilizar no mínimo micros 486 DX-100 com uma placa de vídeo PCI. Se você utilizar micros um pouco mais rápidos, a partir de um Pentium 100 a atualização de tela já passará a ser instantânea. Numa rede de 100 megabits pode-se conectar 10, 15 ou até mesmo 20 terminais no servidor antes que a velocidade da rede comece a tornar-se problemática.

4.2. Configuração de *hardware* do Servidor

O que se deve levar em consideração referente à configuração do servidor é a quantidade de memória instalada. Quanto mais conexões de terminais ele tiver, maior terá que ser a quantidade de memória.

Morimoto (2003) recomenda a seguinte configuração: Pentium III ou Athlon com 256 MB de RAM e mais 64 MB para cada dois terminais que forem ser adicionados. Ou seja, se conectar 4 terminais, seria recomendável ter 512 MB, se for conectar 10 terminais ou mais então 1 GB já seria mais apropriado. O processador não precisa ser nenhum topo de linha, o mais importante é ter um HD razoavelmente grande (já que os arquivos de todos os usuários serão armazenados unicamente no servidor) e bastante memória RAM.

O servidor não precisa ser dedicado. Deve-se também desabilitar a opção de desligamento local e remoto para que nenhum usuário consiga desligar o servidor por engano.

4.3. Por Que Usar o Kurumin?

Existem muitas distribuições Linux recomendadas para uso em servidores, como o Debian, Red Hat, Fedora, Mandrake e Slackware, para citar só algumas, mas poucas distribuições com ênfase nos usuários menos avançados. É aqui que chegamos no Kurumin.

O Kurumin difere das outras distribuições por ser desenvolvido pensando no usuário menos avançado e não no superusuário que entende tudo de linha de comando.

A primeira preocupação do desenvolvedor foi com que o sistema funcione diretamente a partir do CD, de forma que se possa usá-lo em qualquer micro, sem precisar fazer backup de tudo, particionar o HD e passar por um processo tedioso de instalação. Simplesmente dá-se boot pelo CD e ele roda sem alterar nada que está gravado no HD.

A segunda é que o sistema seja pequeno, sempre menos de 200 MB, fazendo com que o *download* seja rápido. A ideia é que o sistema venha com os 10% de programas, usados por 90% das pessoas, os melhores dentro de cada categoria.

4.4. Experiência Comprovada

O Kurumin *Terminal Services*, tem sido usado por várias pessoas, muitas usam só como um divertimento, desafio, outras estão implantando em instituições, que é o caso da Prefeitura do Rio Grande – RS, que adotou o Kurumin Linux associado ao LTSP, para uso interno da prefeitura e também para um projeto que disponibiliza Internet gratuita para a população carente (Fonte: www.noticiaslinux.com.br, <http://www.riogrande.rs.gov.br/>).

4.5. LTSP - Projeto Servidor de Terminais Linux

Neste ponto será discutido o projeto de Linux *Terminal Services*, uma solução usada pela comunidade do software livre, onde é contada a história e como se desenvolveu o LTSP (*Linux Terminal Server Project*).

O Linux é uma plataforma para estações de trabalho que podem ser inicializadas a partir de um servidor de rede. O LTSP é um projeto de código aberto (*Open Source*) que cria as ferramentas administrativas para configurar facilmente esses terminais burros.

O LTSP provê uma forma simples de utilizar estações de trabalho de baixo custo como terminais gráficos ou caracter de um servidor Linux. Nas configurações comuns para utilizar uma suite *office*, normalmente há computadores potentes espalhados pela empresa. Cada um deles com alguns gigabytes de espaço em disco e os usuários normalmente guardam suas informações no disco rígido local e as cópias de segurança são raramente feitas.

Usando o LTSP, pode-se utilizar estações de trabalho com poucos recursos, removendo o disco rígido, o drive de disquetes e o CD-ROM e adicionando uma placa de rede com eprom de *boot*. As estações de trabalho recebem as informações de IP e do kernel do servidor e então montam o sistema de arquivos *root* a partir dele via sistema de arquivos de rede (NFS).

As estações podem ser configuradas de 3 modos:

1. Gráficas com X Windows. Usando o X Windows as estações podem ser usadas para acessar qualquer aplicação no servidor, ou em qualquer outro servidor da rede.
2. Caracter baseada em sessões telnet. As estações podem abrir múltiplas seções telnet no servidor. Cada seção será aberta em telas virtuais separadas. Pressionando Alt-F1 até Alt-F9 você poderá alternar entre essas seções.
3. *Prompt Shell*, útil para debugging. A estação pode ser configurada para entrar diretamente em uma shell na console. Isto é muito útil quando se deseja fazer checagem de erros que aconteçam com o *X Windows* ou com o *NFS (Network File System)*.

A grande vantagem é poder ter muitas estações sendo servidas por um único servidor Linux. O número estações vai depender da configuração do servidor e de quais aplicações se pretendem usar.

4.6. Considerações sobre Segurança

A instalação do LTSP habilitará alguns serviços que podem fazer o ambiente ficar vulnerável a algumas tentativas de invasões. Recomenda-se que um servidor LTSP não deverá estar conectado diretamente à internet sem um *firewall*.

O script de instalação modificará alguns arquivos, como por exemplo: */etc/exports*. Uma entrada que permitirá o acesso de máquinas na classe C 192.168.0.0 para montar o diretório */ftpboot/lts/ltsroot* será adicionada. Outras entradas que permitem a montagem de vários diretórios também serão adicionadas, entretanto elas estarão comentadas e poderão criar alguma vulnerabilidade se forem descomentadas.

Outros arquivos também são modificados para diminuir a vulnerabilidade na configuração do ambiente.

4.7. Vantagens de Desvantagens do *Terminal Services* do Linux Kurumin

Neste tópico, são apresentadas as principais vantagens e desvantagens que envolvem a tecnologia *terminal services* no sistema operacional Linux.

A principal vantagem de usar o Linux Kurumin é a questão do software livre, onde não precisa pagar nenhum tipo de licença para ninguém.

Com o uso do Linux *Terminal Services* (LTS) custo de atualização de *hardware* diminui ainda mais, em comparação ao Windows *Terminal Services*, levando em conta que é preciso nem o uso de disco rígido para os clientes. Sendo que os aplicativos rodam somente no servidor e são mostrados nos terminais. Onde são inicializados diretamente pela placa de rede. Considerando que os micros podem conter as mesmas configurações dos usados no WTS.

Distribuição centralizada de aplicativos trabalha da mesma forma do WTS, faz com que os programas sejam executados somente no servidor, com isso, não a necessidade de instalar os programas nos terminais. O gerenciamento é feito no servidor de terminais, configurado uma vez nele, todos os demais clientes terão as configurações automaticamente. Isso garante que todas as estações serão idênticas ao servidor.

Administrações remotas do servidor podendo-se gerenciar um servidor remotamente de qualquer cliente, através da rede ou mesmo pela internet.

Substituição de estação no projeto Linux, quando uma máquina quebra, pode ser substituída por outra no mesmo instante é só colocar no lugar e pronto, não precisa configurar nada, pois tudo já está no servidor.

Interação com Windows o LTSP é capaz de interagir com o WTS e iniciar um *thin client* em Linux, que passa a ser um desktop de um servidor Windows. Isto é muito útil para rodar aplicativos legados que ainda não foram migrados para Linux. Porém esta é uma solução barata, por ser obrigado a adquirir uma licença especial de uso para acessar o Windows.

As principais desvantagens são:

1. O *terminal services* não é nativo no Kurumin, precisa ser instalado e configurado.
2. Receio do usuário em aprender/usar uma nova tecnologia. As pessoas estão acostumadas a trabalhar com o Windows, podendo no início rejeitar o Linux, por não se familiarizarem com ele.

5. Conclusão

O *Terminal Services* permite que computadores tenham acesso remoto a programas que estão instalados no servidor. Ele pode ser usado para administração remota, o qual permite acesso a servidores lógicos ou fisicamente distantes, e também como servidor de aplicação, onde permite um ambiente de múltiplas sessões para vários usuários acessarem simultaneamente seus aplicativos.

No Windows *Server* o *Terminal Services* é nativo, para que o acesso remoto aconteça é necessário o protocolo RDP, que funciona em cima do protocolo TCP/IP. Seu ponto fraco é o consumo elevado de CPU e outros, além da exigência de licenças de acesso separadas.

O Linux Kurumin *Terminal Services* possui um projeto chamado de *LTSP* (*Linux Terminal Services Project*), que consiste num projeto de código aberto funcionando sobre o TCP/IP. É uma solução para a criação de terminais leves com o Linux. Ele utiliza uma combinação de DHCP, TFTP e XDMCP para permitir que as estações não apenas rodem aplicativos instalados no servidor, mas realmente dêem *boot* via rede, baixando todos os softwares de que precisam diretamente do servidor. Não é preciso ter HD nem CD-ROM nas estações, apenas a ROM da placa de rede ou então um disquete de *boot*. Exige menos CPU para seu funcionamento e não requer licença de acesso, sendo totalmente sem custos.

Referências

- SILVA, M. GLAYDSON. *Guia Foca GNU/Linux*, Versão 5. 2001. <http://www.pcs.usp.br/cgi-bin/jkinoshi/link.cgi?b=22:2>. Acesso em: 28/10/2004.
- MACKEY, DAVID. *Securing Windows 2000 Terminal Service*, 2002. <http://www.microsoft.com/technet/prodtechnol/win2kts/maintain/optimize/secw2kts.mspx>. Acesso em: 15/11/2004.
- CORRÊA, WILLIAN. *Rodando Linux em Qualquer Computador Através da Um Servidor*, 2004. <http://www.imasters.com.br/artigo.php?cn=2119&cc=80> . Acesso em: 20/11/2004.
- STAR UNION. *Soluções Completas em Tecnologia – Terminal Services*, 2004. http://www.smartunion.com.br/smart2/terminal%20_server.htm. Acesso em: 18/11/2004.
- INFOWESTER. *A História do Linux*, 2003. <http://www.infowester.com/linux5.php#TOPO>. Acesso em: 16/10/2004.
- NOAL, LUIZ. *Linux Terminal Services – LTS*, 2004. <http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1078>. Acesso em: 18/10/2009.
- MINASI, M.; ANDERSON, C.; SMITH, B. M.; TOOMBS, D. *Dominando o Microsoft Windows 2000 Server*. São Paulo: Pearson Education do Brasil, 2001.
- TANENBAUM, Andrew S.; *Redes de Computadores*. Rio de Janeiro: Campus, 1997
- TORRES, G. *Redes de Computadores Curso Completo*. Rio de Janeiro: Axcel Books, 2001.
- NEMETH, EVI; SNYDER, GARTH; HEIN, TRENT R.. *Manual Completo do Linux: Guia do Administrador*, 1ª Edição. São Paulo: Pearson Brasil, 2004.
- NORTON, PETER; *Guia Completo do Linux*. São Paulo. Berkeley, 2000.
- HAUFMAN, LAR; WELSN, MATT; *Dominando o Linux*. 1ª edição. Ciência Moderna. 1997
- OLIVEIRA, GISELE; MISAGHI, MEHRAN; *Desempenho de Servidores Metaframe com Aplicações Corporativas*. Artigo. Revista do Instituto Superior Tupy. Ano 4 N° 05, 2004.
- SANTOS, MOISES JOSE DOS; CAMPOS, AUGUSTO; *Economize com Software Livre*. Artigo. Revista do Linux. Ano 2 N° 23, 2001.

- MORIMOTO, CARLOS E; *Entendendo e Dominando o Linux*, 7ª Edição, 2003
<http://www.guiadohardware.net>. Acesso em: 18/10/2009.
- MORIMOTO, CARLOS E; *Kurumin: Desvendando seus segredos*, 2003.
<http://www.guiadohardware.net>. Acesso em: 18/10/2009.
- MCQUILLAN, JAMES. *LTSP - Linux Terminal Server Project - v3.0*, 2001.
<http://www.ltsp.org>. Acesso em: 18/10/2009.
- MCQUILLAN, JAMES; DUTRA, MARLON. *Baixando e instalando o LTSP, versão 3.0*, 2003. <http://www.ltsp.org/documentation/instructions-br-3.0.html>. Acesso em: 18/10/2009.
- HOCHMUTH. PHIL; *Diskless Linux terminals*, 2000.
<http://www.nwfusion.com/newsletters/linux/0410linux1.html>. Acesso em: 18/10/2009.
- REQUISITOS DE SISTEMA; *Base de conhecimentos Microsoft*.
http://www.microsoft.com/brasil/servidores/windowsserver2003/requisitos_sistema.aspx. Acesso em 17/12/2010.