

Concurso Público



Analista de Sistemas

Caderno de Questões
Prova Objetiva

2015

SRH SUPERINTENDÊNCIA
DE RECURSOS
HUMANOS
DA UERJ



Cópia Internet - <http://concursos.srh.uerj.br/>



Conhecimentos Específicos do Cargo

01|

No Windows 2008 Server, em um servidor com 2 GB de RAM, quanto, a memória mínima recomendada para definir o tamanho total do arquivo de paginação, em MB, é de:

- a) 1024
- b) 2048
- c) 3072
- d) 4096

02|

O resultado da conversão para binário dos números decimais 10 e -25, usando 8 bits com notação de complemento a dois é:

- a) 00001010 e 00011001
- b) 00001010 e 11100111
- c) 11110110 e 00011001
- d) 11110110 e 11100111

03|

Os Sistemas Operacionais (S.O.) apresentam uma arquitetura que tem a seguinte forma:

- a) de um único programa, que é compilado de forma única, no qual as interações entre os serviços são feitas em camadas ao redor de um núcleo
- b) de um núcleo mínimo, no qual todos os módulos solicitam permissão para interagir por meio de sinalizações entre processos, subprocessos e *threads*
- c) em camadas, oferecendo um conjunto de funções, que podem ser utilizadas apenas pelas camadas superiores, criando restrições funcionais na interação dos módulos
- d) vários módulos que são compilados separadamente e depois linkados, formando um grande e único programa executável, no qual os módulos podem interagir livremente

04|

O sistema de linguagem SQL (*Structured Query Language*, ou Linguagem de Consulta Estruturada) possui a seguinte característica:

- a) está centrado no conceito de adição ou remoção de servidores de banco de dados de forma escalável e sem interrupção dos serviços
- b) está baseado no relacionamento estruturado para endereços de nomes de domínios com seus respectivos endereços de máquina
- c) está baseado na pesquisa declarativa padrão para banco de dados relacional; muitas de suas características foram inspiradas na álgebra relacional
- d) está centrado em bancos de dados distribuídos, onde dados são armazenados em múltiplos pontos de processamento e normalmente em muitos servidores



05|

Suponha um *host* com endereço IP 10.1.1.185 e máscara de sub-rede 255.255.255.128. A sub-rede à qual o *host* pertence, o endereço de broadcast e a faixa de IPs disponível para os *hosts* dessa sub-rede são, respectivamente:

- | | | |
|------------------|------------|-------------------------|
| a) 10.1.1.128/25 | 10.1.1.255 | 10.1.1.129 – 10.1.1.254 |
| b) 10.1.1.128/26 | 10.1.1.255 | 10.1.1.128 – 10.1.1.255 |
| c) 10.1.1.128/26 | 10.1.1.128 | 10.1.1.129 – 10.1.1.254 |
| d) 10.1.1.255/25 | 10.1.1.128 | 10.1.1.128 – 10.1.1.255 |

06|

São portas TCP padrão, sem criptografia, utilizadas pelos protocolos SMTP, IMAP e POP3, respectivamente:

- a) 25, 133 e 110
- b) 25, 143 e 110
- c) 465, 143 e 100
- d) 465, 143 e 111

07|

A conversão do binário 11001111.01 para hexadecimal é:

- a) 8F.2₁₆
- b) DF.1₁₆
- c) CE.4₁₆
- d) CF.4₁₆

08|

Os certificados confiáveis são utilizados para garantir uma comunicação segura na Internet. A responsabilidade da Autoridade de Certificação (CA - *Certification Authority*) na comunicação segura na Internet é:

- a) protocolar, garantindo a conexão segura em redes sem fio e estabelecendo um modelo de chaves baseado no endereço MAC do Cliente para a determinação da chave pública e assimétrica
- b) armazenar chaves para validação dos domínios na Internet; é por meio dos CAs que os navegadores conseguem trocar os nomes dos domínios nos endereços IPs para efetuar a navegação segura na rede
- c) ser o terceiro ente confiável e emitir um certificado em uma conexão segura; o cliente usa o certificado da Autoridade de Certificação para verificar a assinatura do servidor antes de se estabelecer a conexão segura
- d) assumir o papel do cliente na comunicação segura e conhecer antecipadamente o servidor confiável; existem apenas dois entes participantes na comunicação, que estabelecem troca de chaves antes de se estabelecer uma conexão segura



09|

O protocolo Camada Segura de Sockets ou *Secure Sockets Layers* (SSL) é um padrão de segurança que atua da seguinte forma:

- a) definindo o melhor serviço de nomes de domínios (DNS) da rede
- b) aprimorando o TCP, com sigilo, integridade dos dados, autenticação de servidor e cliente
- c) definindo o protocolo de roteamento fim a fim entre todos os equipamentos de rede no caminho
- d) informando o endereço MAC do remetente ao destinatário, o que garantirá confiabilidade na transferência das informações entre cliente e servidor

10|

Em redes de computadores, o Protocolo de Resolução de Endereços (ARP – *Address Resolution Protocol*) está associado à fronteira das seguintes camadas do modelo OSI:

- a) enlace e rede
- b) física e enlace
- c) rede e transporte
- d) transporte e aplicação

11|

Em relação ao endereço IP 172.22.3.65, máscara 255.255.255.192, pode-se afirmar que:

- a) o endereço da rede é 172.22.3.0
- b) esse é o endereço de *gateway* da rede
- c) o endereço de *broadcast* da rede é 172.22.3.127
- d) o último endereço de *host* da faixa é 172.22.3.254

12|

O endereço de *broadcast* da rede a que pertence o endereço IP 145.77.88.4/22 é:

- a) 255.255.255.255
- b) 145.77.255.255
- c) 145.77.88.255
- d) 145.77.91.255

13|

Dentre as tecnologias de armazenamento em redes de computadores “*Network Attached Storage*” (NAS) e “*Storage Area Network*” (SAN), a tecnologia SAN se destaca por:

- a) ter um menor custo na implementação e gestão, quando comparado à outra tecnologia
- b) ter um elevado custo de implementação e de gestão, quando comparado à outra tecnologia
- c) oferecer um sistema de arquivos, não permitindo o gerenciamento dos blocos de armazenamento diretamente por parte do cliente
- d) oferecer interligação com o cliente predominantemente por meio de TCP/IP, não suportando protocolos de baixo nível como o ATA



14|

Na configuração do MTA Postfix, normalmente configurados no arquivo *main.cf*, o parâmetro que limita o tamanho máximo de mensagem que o sistema aceitará é:

- a) *message_size_limit*
- b) *local_recipient_size*
- c) *default_message_limit*
- d) *maximal_size_message*

15|

Para operar com grandes volumes de dados ou garantir uma alta disponibilidade no armazenamento, é necessário conhecimentos em *Redundant Array of Independent Disks* (RAID). Com o objetivo de promover uma alta disponibilidade de dados, um administrador decide investir na confiabilidade do armazenamento de um servidor com 4 (quatro) discos rígidos (*hard disk*) por meio do uso de RAID.

A tecnologia RAID que oferece a maior confiabilidade em caso de falhas simultâneas de disco é:

- a) RAID0
- b) RAID1
- c) RAID6
- d) RAID10

16|

No Windows 2008 Server, o *User Account Control* (UAC) aumenta a segurança impedindo que usuários implementem mudanças inadvertidas nas configurações do sistema e bloqueia o computador para impedir que aplicativos não autorizados se instalem ou executem ações suspeitas.

O principal componente do UAC que determina como os administradores serão avisados da elevação é:

- a) *Admin Alert Mode*
- b) *Admin Switch Mode*
- c) *Admin Approval Mode*
- d) *Admin User Control Mode*

17|

Levando em consideração o endereço 172.22.3.65, referente às redes ipv4, no qual existem redes públicas e privadas, assinale a opção que apresenta a classe a que pertence esse endereço e se é público ou privado.

- a) classe B / público
- b) classe B / privado
- c) classe C / público
- d) classe C / privado

Responda às questões de 18 a 40, considerando o sistema operacional Linux.

18|

O protocolo SSH permite a criação de ferramentas extremamente versáteis. Com ele é possível exportar e usar remotamente aplicativos gráficos e de texto, transportar outros protocolos internamente por meio de túneis estáticos (como o http e https), criar túneis dinâmicos para navegação assistida de *browsers*, entre outras funcionalidades.

Supondo que o gerente de um servidor OpenSSH configure o envio de pacotes automáticos, reduzindo a ociosidade da conexão, por meio do canal encriptado para manter abertas as conexões com os clientes, a diretriz que deve ser usada é:

- a) TCPAlive
- b) TCPKeep
- c) KeepAlive
- d) ClientAliveInterval

19|

A solução correta para que uma consulta sql retorne às agências que possuem média dos saldos aplicados em conta maior que 1200 é:

- a)

```
select nome_agencia, avg(saldo)
from conta
group by nome_agencia
having avg(saldo) > 1200
```
- b)

```
select nome_agencia, avg(saldo)
from conta
where ( having avg(saldo) > 1200 )
```
- c)

```
select nome_agencia, avg(saldo)
from conta
where avg(saldo) > 1200
group by nome_agencia
```
- d)

```
select nome_agencia, avg(saldo)
from conta
group by nome_agencia
having saldo > 1200
```

20|

O nome e o número do sinal que destrói o processo receptor e não pode ser capturado, bloqueado ou ignorado são:

- a) SIGKILL 9
- b) SIGKILL 15
- c) SIGTERM 9
- d) SIGTERM 15



21|

Um analista é chamado para verificar o porquê de um domínio não estar respondendo com o nome apropriado. Ele utiliza, por meio de um *shell*, ferramentas para consultas de DNS. Dentre as ferramentas abaixo relacionadas, a que **não** permite realizar tais verificações é:

- a) dig
- b) host
- c) resolver
- d) nslookup

22|

O primeiro processo a ser executado depois do sistema operacional ser inicializado, que sempre possui o PID igual a 1, é denominado de:

- a) init
- b) getty
- c) telinit
- d) inittab

23|

Em um ambiente jboss 6, pode-se acessar o console *web* de gerenciamento utilizando a seguinte porta padrão, pré-definida na instalação:

- a) 8000
- b) 8080
- c) 8443
- d) 9990

24|

O comando *chmod* modifica as permissões de um arquivo. Observe as permissões do arquivo de nome *arquivo.txt*, listado pelo comando *ls* com opção *-l*:

```
# ls -l arquivo.txt
-rw-rw-r-- 1 root root 23 Jan  3 18:04 arquivo.txt
```

Após a execução do comando *chmod*, as permissões do arquivo foram alteradas, conforme a seguir:

```
# ls -l arquivo.txt
-rwxrw-r-- 1 root root 23 Jan  3 18:04 arquivo.txt
```

Para que o *arquivo.txt* recebesse a letra "x" para o dono, permitindo a execução do arquivo, o comando executado foi:

- a) *chmod 7+x arquivo.txt*
- b) *chmod 764 arquivo.txt*
- c) *chmod u=g arquivo.txt*
- d) *chmod g+x arquivo.txt*



25|

Processos de automatização de tarefas mais complexas pode exigir a verificação de uma ação para a execução de uma segunda ação somente se a primeira apresentar erro.

Dentre as opções relacionadas abaixo, a que representa uma linha de *script* de *bash* que executará “comando02” somente se “comando01” apresentar erro é:

- a) comando01 ; comando02
- b) comando01 || comando02
- c) comando01 && comando02
- d) comando01 ; else comando02

26|

O daemon “cron” possui a função de gerenciar:

- a) as conexões de redes
- b) as mensagens de erros
- c) o agendamento de tarefas
- d) o compartilhamento de arquivos em uma rede local

27|

Os arquivos ficam organizados em pastas distintas chamadas diretórios. Observe os diretórios abaixo:

/bin – arquivos de configuração e inicialização críticos
/boot – Kernel e arquivos necessários para carregar o kernel
/dev – entradas de dispositivos para terminais, discos, modems, etc.
/etc – comandos necessários para mínima operação do sistemas

Assinale a opção que contém aqueles que estão com o conteúdo trocado.

- a) /bin e /etc
- b) /dev e /etc
- c) /bin e /boot
- d) /dev e /boot

28|

O comando que permite executar outros comandos como administrador do sistema é:

- a) ps
- b) root
- c) sudo
- d) super



29|

Observe a estrutura abaixo:

wheel:*:10:root, evi, garth, trent

csstaff:*:100:lloyd, evi

student:*:200:dotty

De acordo com a estrutura do arquivo `/etc/group`, o conteúdo das colunas deve ser, respectivamente:

- nome do grupo : comentário : número GID : membros do grupo
- nome do grupo : senha criptografada : número GID : membros do grupo
- nome do usuário : senha criptografada : número UID : usuários do grupo
- membro do grupo : senha criptografada : número UID : usuários do grupo

30|

O *iptables*, como um *firewall*, pode manipular um pacote dentro do kernel, utilizando a tabela nat, que traduz endereços para que redes distintas possam se comunicar.

Nessa tabela, as *chains* que podem ser utilizadas são:

- INPUT, OUTPUT e FORWARD
- FORWARD, POSTROUTING e INPUT
- INPUT, OUTPUT, PREROUTING e POSTROUTING
- PREROUTING, FORWARD, OUTPUT e POSTROUTING

31|

Deseja-se modificar a permissão do arquivo `/home/uerj/teste.txt` para que:

- O dono do arquivo tenha permissão de leitura escrita e execução;
- O grupo ao qual pertence ao usuário tenha permissão de leitura e execução;
- Os demais usuários tenham apenas permissão de leitura.

O comando a ser usado é:

- `chmod 167 /home/uerj/teste.txt`
- `chmod 457 /home/uerj/teste.txt`
- `chmod 754 /home/uerj/teste.txt`
- `chmod 764 /home/uerj/teste.txt`

32|

Em um Servidor de DNS, no seu arquivo de configuração de zona, o parâmetro que permite que o servidor secundário busque, de tempos em tempos, a comunicação com o servidor primário é:

- TTL
- retry
- expire
- refresh

33|

No servidor DNS, considerando os parâmetros do arquivo *named.hosts* do aplicativo Bind, a opção que **não** contém um registro de recurso é:

- MAIL
- NS
- MX
- A



34|

Suponha que no diretório /home/uerj/teste existam os subdiretórios d1, d2, d3, d4 e d5 e que, em cada um deles, exista um arquivo com o nome a.txt. Deseja-se fazer um *script* que mova o arquivo a.txt desses subdiretórios para o diretório /home/uerj/teste. Como todos os arquivos possuem o mesmo nome, deve-se renomeá-lo e adicionar o número do diretório ao nome do arquivo. Por exemplo, o arquivo /home/uerj/teste/d1/a.txt será movido para /home/uerj/teste/a1.txt enquanto que o arquivo /home/uerj/teste/d3/a.txt será movido para /home/uerj/teste/a3.txt.

O código do *script* que realiza esse procedimento é:

- a) for ((i=1;i<5;i++)) mv /home/uerj/teste/d\${i}/a.txt /home/uerj/teste/a\${i}.txt
- b) for ((i=1;i<6;i++)) mv /home/uerj/teste/d\${i}/a.txt /home/uerj/teste/a\${i}.txt
- c) for ((i=1;i<6;i++)); do mv /home/uerj/teste/d\${i}/a.txt /home/uerj/teste/a\${i}.txt ; done
- d) for ((i=1;i<5;i++)); do mv /home/uerj/teste/d\${i}/a.txt /home/uerj/teste/a\${i}.txt ; done

35|

Em um servidor Ldap, pode-se utilizar a porta padrão 389 para suportar sessões criptografadas e não criptografadas, dependendo da solicitação do cliente, através do seguinte mecanismo:

- a) SSL
- b) SASL
- c) LDAPS
- d) STARTTLS

36|

O interpretador de comandos *bash* permite a verificação do *status* do encerramento do último comando ou função executado no *script*. Um exemplo de trecho de código fonte que verifica o valor desse *status* é:

- a) if [\$? -ne 0]; then exit; fi
- b) if [\$# -ne 0]; then exit; fi
- c) if [\$0 -eq 1]; then exit; fi
- d) if [\$* -eq 1]; then exit; fi

37|

Observe o comando abaixo:

PowerEdge-2850:~\$ ps l

F	UID	PID	PPID	PRI	NI	VSZ	RSS	WCHAN	STAT	TTY	TIME	COMMAND
0	1000	3870	3860	20	0	8568	3236	wait	Ss	pts/1	0:00	bash
0	1000	4024	3870	20	0	2272	556	wait	S	pts/1	0:00	sh
0	1000	4025	4024	20	0	8568	3232	wait	S	pts/1	0:00	bash
0	1000	4043	4025	20	0	6500	768	-	R+	pts/1	0:00	ps l

PowerEdge-2850:~\$

Considere as afirmativas sobre a saída do comando e assinale a opção que contém as corretas.

- 1) todos os processos estão sendo rodados por um mesmo usuário
- 2) um dos processos ocupa há mais tempo a CPU
- 3) os processos “bash” possuem processos pai diferentes
- 4) todos têm a mesma prioridade
- 5) todos os processos estão esperando execução de outros processo

- a) 1, 2 e 4
- b) 1, 3 e 4
- c) 2, 3 e 5
- d) 2, 4 e 5



38|

No postfix, para melhor segurança, pode-se implementar o método de autenticação SMTP, o qual requer que clientes de e-mail enviem um nome de usuário e uma senha ao servidor SMTP durante o diálogo SMTP, e, se a autenticação foi bem-sucedida, eles poderão começar a retransmitir.

No arquivo de configuração que controla o daemon do *postfix*, o parâmetro que deve ser alterado para que o mecanismo de autenticação seja habilitado é:

- a) `smtpd_tls = yes`
- b) `smtpd_sasl_auth_enable = yes`
- c) `smtpd_client_auth_connect = yes`
- d) `smtpd_sasl_security_options = yes`

39|

No gerenciamento de processos, operações ligadas ao I/O aguardam que uma determinada operação de I/O seja finalizada, com isso o processo entra em um estado em que não pode ser morto, para que não haja inconsistências. Pode-se ainda identificá-lo através do comando *top* na coluna *status*.

Este estado é:

- a) `TASK_ZOMBIE`
- b) `TASK_STOPPED`
- c) `TASK_INTERRUPTIBLE`
- d) `TASK_UNINTERRUPTIBLE`

40|

Em servidor Web, Apache, para melhorar o desempenho e a performance em que o servidor não precise verificar em cada diretório a existência de um arquivo `.htaccess`, a diretiva que podemos usar é:

- a) `AllowOverride all`
- b) `AllowOverride Deny`
- c) `AllowOverride none`
- d) `AllowOverride Allow`

Inglês Técnico

Responda às questões de 41 a 44 considerando o texto a seguir.

Here's the first line of code ever written by a US president

Barack Obama just became the first US president to write a line of computer code (assuming George W. Bush never secretly indulged in PHP). At the White House yesterday, Obama sat down with students who were learning the fundamentals of JavaScript, the popular programming language used to create most web pages.

The line he wrote was:

`moveForward(100);`

“So I make the F in higher case?” Obama asked, correctly observing that JavaScript is case sensitive. “Semicolon?” (That semicolon is optional, but Obama apparently has a knack for recognizing JavaScript best practices.)



Obama was playing with a Code.org tutorial based on the popular Disney movie Frozen. In his line of code, the President called a function—`moveForward`—pre-defined by Code.org for the exercise.

Calling a function in JavaScript is simple: write its name exactly as it has been defined, followed by parentheses that contain its “arguments.” In this case, a single argument tells the program how many pixels to move a *Frozen* character forward. Because it’s measured in pixels, the argument has to be a number. If Obama had written `moveForward(“three steps”)`, the program would have failed, offering only a cryptic error message and exposing the president to the near-perpetual state of frustration most software developers live in.

“This is Elsa?” Obama asked, referring to the movie’s main character.

Obama was promoting Computer Science Education Week and Code.org’s Hour of Code campaign, which encourages kids to try programming for at least one hour. “It turns out the concepts are not that complicated,” Obama told the students at the White House, though his attempt to explain it suggested otherwise:

“The basic concept behind coding is that you take zeros and ones, you take two numbers, yes or no, and those can be translated into electrical messages that then run through the computer.... So all it’s doing is it’s saying yes or no over and over again, and the computer’s powerful enough that it can read a really long set of instructions really quickly.”

Something like that.

Disponível em: <http://qz.com/308904/heres-the-first-line-of-code-ever-written-by-a-us-president/>
Quartz (9 de Dezembro de 2014) - Texto de Zachary M. Seward

41|

A atividade desempenhada pelo Presidente Obama na Casa Branca foi:

- a) estudar PHP com George W. Bush
- b) estudar PHP com um grupo de estudantes, através do site Code.org
- c) ensinar estudantes a programar em JavaScript, usando um tutorial do site Code.org.
- d) promover a campanha *Hour of Code*, do site Code.org, que encoraja crianças a tentarem programar por pelo menos uma hora

42|

Enquanto programava, Obama observou e questionou, respectivamente, os seguintes aspectos:

- a) que em código PHP há distinção entre maiúscula e minúscula / sobre o uso da “vírgula” no programa
- b) que em código JavaScript não há distinção entre maiúscula e minúscula / sobre o uso da “vírgula” no programa
- c) que em código JavaScript há distinção entre maiúscula e minúscula / sobre o uso do “ponto e vírgula” no programa
- d) que em código PHP não há distinção entre maiúscula e minúscula / sobre o uso do “ponto e vírgula” no programa



43|

A explicação do Presidente Obama sobre os conceitos de programação e a percepção do autor do texto sobre essa explicação são, respectivamente:

- a) os conceitos de programação não são tão complicados / a explicação dada pelo Presidente sugeriu o contrário
- b) os conceitos de programação não são tão complicados / o Presidente deu uma explicação sobre o assunto que mostrou que ele realmente entendeu o conceito
- c) os conceitos de programação são muito complicados / o Presidente deu uma explicação sobre o assunto que mostrou que ele realmente entendeu o conceito
- d) os conceitos de programação são muito complicados / a explicação dada pelo Presidente mostra que ele realmente não entendeu como funciona um computador

44|

O exercício de programação que o Presidente Obama estava fazendo tinha o objetivo de mover um personagem da Disney para frente. Para isso, a função `moveForward` deveria ser usada.

A distância que o personagem deveria ser movido para frente e os parâmetros da função são, respectivamente:

- a) 100 pixels / o número de pixels (100)
- b) 3 passos / o número de passos ("three steps")
- c) 3 passos / o número de pixels (100) e o número de passos ("three steps")
- d) 100 pixels ou 3 / o número de pixels (100) ou o número de passos ("three steps")

Responda às questões de 45 a 50 considerando o texto a seguir.

Virtual network appliances: Benefits and drawbacks

There's lots of talk about network virtualization benefits, but are virtual network appliances all they're cracked up to be? Only in some scenarios.

Network virtualization benefits can be plentiful, but only in certain scenarios. Learn where virtual network appliances can work -- and where they can't.

If virtualization enables servers to be spun up and down on demand for cost efficiency and agility, wouldn't it make sense to implement virtual network components too? After all, virtual servers need to communicate inbound and outbound and still be firewall-protected and load balanced. That would seem to be best addressed by virtual network appliances that can be spun-up on demand, right? Only in some scenarios.

Many networking vendors have already begun to minimize development cost by using Intel-based platforms and commodity hardware. Examples of this range from the Cisco ASA firewall to F5 load balancers and Vyatta routers. The obvious next step for some of these vendors has been to offer their products in virtual appliance packaging. F5 took a small step forward with the Local Traffic Manager – Virtual Edition (LTM VE), while Vyatta claims to offer a full range of virtual appliance solutions. VMware was somewhat late to the game, but it also offers virtualized firewalls (vShield Zones and vShield App) and routers/load balancers (vShield Edge).

Virtual network appliances: What's the catch?

The problem is that unlike servers, networking appliances commonly perform I/O-intensive tasks, moving large amounts of data between network interfaces with minimal additional processing, relying heavily on dedicated hardware. All high-speed routing and packet forwarding, as well as encryption (both IPsec and SSL) and load balancing, rely on dedicated silicon. When a networking appliance is repackaged into a virtual machine format, the dedicated hardware is gone, and all these tasks must now be performed by the general-purpose CPU, sometimes resulting in extreme reduction in performance.

Implementing routers, switches or firewalls in a virtual appliance would just burn the CPU cycles that could be better used elsewhere -- unless, of course, you've over-provisioned your servers and have plenty of idle CPU cycles, in which case something has gone seriously wrong with your planning.

To make matters worse, the hypervisor software used in server virtualization solutions also virtualizes the network interfaces. That means that every I/O access path to virtualized hardware from the networking appliance results in a context switch to higher privilege software (the hypervisor), which uses numerous CPU cycles to decode what needs to be done and emulate the desired action. Also, data passed between virtual machines must be copied between their address spaces, adding further latency to the process.

There is some help in that the VMware hypervisor has the DVFilter API, which allows a loadable kernel module to inspect and modify network traffic either within the hypervisor (vNetwork Data Path Agent) or in combination with a virtual machine (vNetwork Control Path Agent). The loadable kernel module significantly reduces the VM context switching overhead.

Where virtual network appliances can work?

There are some use cases in which virtual network appliances make perfect sense. For instance, you could virtualize an appliance that performs lots of CPU-intensive processing with no reliance on dedicated hardware. Web application firewalls (WAFs) and complex load balancers are perfect examples (no wonder they're commonly implemented as loadable modules in Apache Web servers or as Squid reverse proxy servers).

Also, if you're planning to roll out multi-tenant cloud, the flexibility gained by treating networking appliances as click-to-deploy Lego bricks might more than justify the subpar performance. This is especially so if you charge your users by their actual VM/CPU usage, in which case you don't really care how much CPU they're using.

Virtualized networking also makes sense when firewall and routing functions are implemented as part of the virtual switch in each hypervisor. This could result in optimal traffic flow between virtual machines (regardless of whether they belong to the same IP subnet or not) and solve the problem of traffic trombones. Unfortunately, it seems that Cisco is still the only vendor that extends the VMware hypervisor switch using the Virtual Ethernet Module (VEM) functionality. While numerous security solutions already deploy the VMsafe APIs, the networking appliances I've seen so far (including the vShield Edge from VMware) rely on virtual machines to forward traffic between virtual (or physical) LANs.

Obviously the networking vendors have a very long way to go before reaching the true potential of virtualized networking.

Disponível em: <http://searchnetworking.techtarget.com/tip/Virtual-network-appliances-Benefits-and-drawbacks>

Search Networking – Tech Target – Texto de Ivan Pepelnjak (Março de 2011)



45|

Os tipos de aplicação de rede que fazem sentido com *appliances* de rede virtuais são:

- a) as que fazem uso intensivo de CPU e que dependem de *hardware* dedicado
- b) as que fazem uso intensivo de CPU e que não dependem de *hardware* dedicado
- c) as que não fazem uso intensivo de CPU e que dependem de *hardware* dedicado
- d) as que não fazem uso intensivo de CPU e que não dependem de *hardware* dedicado

46|

O *software* hipervisor usado em soluções de virtualização de servidores também virtualiza as interfaces de rede. Uma consequência que isso tem para o *appliance* de rede virtual é:

- a) futuramente, a latência de processamento das operações de rede podem ser diminuídas
- b) a troca de dados entre máquinas virtuais é facilitada pelo acesso direto aos seus espaços de endereçamento
- c) *softwares* importantes, como os de decodificação e análise de contexto, que usam muitos ciclos de CPU serão beneficiados
- d) operações de entrada e saída feitas pelos *appliances* de rede virtuais resultam em ciclos extras de CPU associados às trocas de contexto

47|

Os exemplos de aplicações que fazem sentido com *appliances* de rede virtuais são:

- a) *firewalls* básicos e mecanismos de balanceamento de carga complexos
- b) *firewalls* básicos e mecanismos de balanceamento de carga tradicionais
- c) *firewalls* de aplicações *web* e mecanismos de balanceamento de carga complexos
- d) *firewalls* de aplicações *web* e mecanismos de balanceamento de carga tradicionais

48|

Existem requisitos de *hardware* ao virtualizar aplicações de roteamento e encaminhamento de pacotes e criptação.

O requisito de *hardware* necessário e o desempenho esperado das aplicações são, respectivamente:

- a) *hardware* dedicado / desempenho ruim
- b) *hardware* dedicado / bom desempenho
- c) processadores de propósito geral / desempenho ruim
- d) processadores de propósito geral / bom desempenho

49|

O texto apresenta vantagem do uso de plataforma baseada em intel para dispositivos de rede. A vantagem e o próximo passo adotado pelas empresas são, respectivamente:

- a) maior comodidade / investir em embalagens melhores
- b) maior comodidade / oferecer *appliances* de rede virtuais
- c) minimizar custos de desenvolvimento / investir em embalagens melhores
- d) minimizar custos de desenvolvimento / oferecer *appliances* de rede virtuais

50|

O texto II faz um comentário sobre o uso de *appliances* de rede virtuais em “*multi-tenant cloud*”. Com base nesse comentário, é possível concluir que:

- a) se os usuários são tarifados pelo uso de CPU, o desempenho passa a ser irrelevante
- b) se usuários especiais exigem CPUs mais atuais, não podemos nos importar com o quanto de CPU eles usam
- c) *appliances* de rede virtuais são como poderosos blocos de Lego, quando comparados aos CPUs atuais
- d) a facilidade de implantar *appliances* virtuais nesses ambientes é justamente o que resulta em um bom desempenho

Cópia Internet - <http://concursos.srh.uerj.br/>