CAPÍTULO 5 - NOMEAÇÃO

Nome: cadeia de *bits* ou caracteres usada para referenciar uma entidade, como: hospedeiros, impressoras, discos, arquivos, processos, usuários.

Endereço: ponto de acesso com o qual se pode agir sobre uma entidade.

Não se deve usar o endereço como nome, porque, assim que o endereço mudar ou for designado a outra entidade, perde-se a referência à antiga.

Identificador: independente de localização/endereço, todo identificador referencia só uma entidade; uma entidade é referenciada por só um identificador; o identificador é único, sempre referenciando a mesma entidade.

Nomeação Simples

Por broadcast: uma mensagem que contém o identificador da entidade é enviada por *broadcast* a cada máquina da rede; elas verificam se têm a entidade e, as que tiverem, retornam uma mensagem com o endereço do ponto de acesso. O ARP (Address Resolution Protocol) funciona assim, buscando o endereço de enlace quando é dado apenas o endereço de IP.

Por multicasting: *broadcast* torna-se ineficiente para redes cheias, e máquinas podem estar ocupadas para responder requisições. Multicast envia mensagens apenas a hospedeiros de um grupo restrito. Grupos possuem um endereço *multicast* e, quando uma mensagem chega a ele, a camada de rede usa de melhor esforço para entregar a mensagem a todos hospedeiros do grupo.

Ponteiros Repassadores: quando uma entidade se move de A para B, deixa em A uma referência a seu novo endereço em B. A vantagem é sua simplicidade, mas a desvantagem é que, se se tratar de uma entidade móvel, o custo para chegar a ela pode tornar-se proibitivo. Além disso, localizações intermediárias terão de manter seus repassadores por quanto tempo for

necessário, desperdiçando recursos. Outra vulnerabilidade é o rompimento de um enlace, o que faria perder a referência a todo resto da cadeia.

Abordagens baseadas na localização nativa

Localização nativa: monitora a localização corrente de uma entidade. Técnicas podem ser aplicadas para evitar falhas de rede ou processo. O IP móvel é assim. Suas desvantagens: a alta na latência de comunicação; a obrigação de sempre existir a localização nativa para contatar a entidade.

Abordagens hierárquicas

Uma rede é dividida em domínios, e há um único domínio de nível mais alto, que abrange toda extensão da rede. Cada domínio pode ser dividido em vários menores; domínios de nível mais baixo são denominados domínios-folha.

Quando uma requisição é feita a um domínio-folha e ele não consegue atendê-la, passa-a a seu nó-pai e, caso esse também não consiga, passa-a acima, e assim por diante. Quando se chega ao nó que contém o registro de localização para a entidade buscada, a solicitação passa pelos nós abaixo dos seus até que se chegue ao destino.

Nomeação Estruturada

Confusamente, a nomeação simples é composta apenas de bits; a nomeação estruturada é de caracteres, feita para ser simples de usuários entenderem, como ocorre no sistema de arquivos.

Espaço de nome: costumam ser representados como um gráfico dirigido e rotulado com dois tipos de nós; um nó-folha representa uma entidade nomeada e tem a propriedade de não ter ramos de saída. Geralmente,

representam a própria entidade, como o arquivo em "/bin/teste.txt." Por sua vez, nós-diretórios tem vários ramos de saída, rotulados, cada, por nome. Cada nó-diretório guarda uma tabela cujo ramo de saída é representado por um par (rótulo_ramo, identificador_nó.) Quando o nome de um caminho começa no nó-raiz, chama-se absoluto; caso contrário, é dito caminho relativo.

Resolução de nomes: processo de busca de um nome para que se ache a entidade por ele representada.

Mecanismo de Fechamento: serve para saber como e onde iniciar uma resolução de nomes. Trata da seleção do nó inicial em um espaço de nomes a partir do qual a resolução de nomes deve começar. As dificuldades em compreender os mecanismos de fechamento se dá porque eles são implícitos.

Apelidos (aliases): outro nome para entidade, como variáveis de ambiente. Para isso, pode-se usar *links* simbólicos ou *hard links*. Neste caso, vários nomes de caminho absoluto referenciam o mesmo nó em gráfico de nomeação. No outro, há um nó-folha que, em vez de armazenar o endereço ou estado de uma entidade, armazena o caminho absoluto para outra entidade.

Ponto de Montagem: nó raiz de um espaço de nomes.

NFS: Sistema de Arquivos de Rede, é um sistema que vem com um protocolo que descreve como um cliente pode acessar um arquivo em um servidor remoto. Para isso, usa-se URLs do NFS como nfs://xyz.br//caio/abc.txt. Essa URL nomeia um arquivo abc.txt (dentro do diretório /caio/), em um servidor NFS xyz.br. A parte do nome é resolvido por DNS; do acesso ao servidor, pelo NFS, e do diretório interno, pelo servidor de espaço de nomes.

DNS

O DNS distribui-se por três espaços de nome; dividida nas camadas:

Camada Global: formada pelo nó-raiz e outros próximos. São os de nível mais alto e, portanto, muito estáveis. Alta disponibilidade é-lhes essencial

porque, se deixam de funcionar, todos os nós abaixo de onde a falha ocorreu ficam inacessíveis. Em questões de performance, como dificilmente mudam seus registros, pode-se guardá-los em cache sem maiores perdas. Pode ter um desempenho menor para processar solicitações, uma vez que o número dessas é menor, dado que a maioria são feitas ao cache, tornando desnecessário um desempenho muito alto. Mantidas apenas por administradores de sistema.

Camada Administrativa: diretórios que, juntos, são gerenciados por só uma organização. São relativamente estáveis, embora ocorram mais mudanças que as da camada anterior. A disponibilidade aqui é de suma importância só para usuários da organização; externos a ela, pode não ser tão necessário. Em questão de desempenho, também se pode usar cache. Mas, diferente da camada global que pode demorar mais, essa camada deve ter um tempo de resposta melhor, consolidando atualizações mais rapidamente que na camada global, pois é inaceitável, p. ex., que um usuário só possa usar sua conta depois de muito tempo. Mantidas apenas por administradores de sistema.

Camada Gerencial: consiste em nós cujo comportamento é a mudança periódica. Hospedeiros na rede local, arquivos compartilhados, diretórios e arquivos de usuário pertencem a essa camada. As exigências de disponibilidade deste nível costumam ser ínfimas, bastando uma máquina dedicada para rodar servidores de nomes. Desempenho, no entanto, é crucial, pois operações devem ser aplicadas imediatamente, e cache é em geral inefetivo. São mantidas por administradores de sistemas e usuários individuais do sistema distribuído.

Zonas: espaços que não se sobrepõem no DNS. É parte do espaço de nome que é implementado por um servidor de nomes separado.

Implementação de Resolução de Nomes

Iterativa: o resolvedor de nomes entrega o nome completo ao servidor raiz. Este resolverá o nome do caminho até onde puder, e retorna ao cliente.

Depois, o cliente envia as demais partes; outro servidor resolve o quanto pode, e então retorna ao cliente. Assim segue até que se tenha resolvido todo nome. Nesse caso, manter resultados em cache não é tão efetivo, e os custos de comunicação, para o cliente, são altos.

Recursiva: em vez de retornar cada resultado intermediário de volta ao resolvedor de nomes do cliente, um servidor de nomes passa o resultado para o próximo servidor de nomes que encontrar. O problema da implementação recursiva é que ela impõe uma exigência de desempenho mais alta a cada servidor de nomes. Por esse motivo, servidores de nome da camada global somente resolvem nomes de forma iterativa. Nesse caso, manter resultados em cache é mais efetivo do que da forma iterativa. Além disso, os custos de comunicação podem ser reduzidos, porque permite que cada servidor de nomes aprenda gradativamente o endereço de servidores de nome responsáveis pela implementação de nós de nível mais baixo.

Espaço de Nome DNS

O DNS é organizado em hierarquia de árvore com raiz. Um rótulo é uma cadeia alfanumérica case insensitive, com comprimento máximo de 63 caracteres; um nome de caminho é limitado a 255 caracteres. A representação da cadeia de um nome de caminho consiste na listagem de seus rótulos, começando da extrema direita, e separando os rótulos por um ponto; a raiz é um ponto na extrema direita na cadeia, e tal ponto costuma ser omitido para fins práticos. Exemplo de endereço: abc.dce.xyz.br.

Cada nó tem um ramo de entrada. O rótulo anexado ao ramo de entrada de um nó também é usado como o nome para aquele nó. Sub-árvores são denominadas domínio; o caminho da raiz a essa sub-árvore é chamado de nome de domínio, que pode ser absoluto ou relativo.

Conteúdo de nós são compostos por conjunto de registros de recursos:

TIPO:	REFERE-SE À:	DESCRIÇÃO:
SOA	Zona	Informações da zona representada
A	Hospedeiro	IP do hospedeiro que o nó representa
MX	Domínio	Servidor de e-mail para manipular o e-mail do nó
SRV	Domínio	Servidor para manipular um serviço específico
NS	Zona	Servidor de nomes para a zona representada
CNAME	Nó	Link simbólico
PTR	Hospedeiro	Nome canônico do hospedeiro
HINFO	Hospedeiro	Informações sobre o hospedeiro do nó
TXT	Todos	Qualquer informação considerada útil

SOA: registro de início de autoridade (Start Of Authority), guarda informações como o endereço de e-mail do administrador de sistemas responsável pela zona representada, o nome do hospedeiro em que os dados sobre a zona podem ser buscados etc.

IMPLEMENTAÇÃO DO DNS

Nomes do DNS podem ser divididos em uma camada global e uma administrativa. A camada gerencial, formada por sistemas locais de arquivos, não é parte formal do DNS, e ele não gerencia essa camada.

Cada zona é implementada por um servidor de nomes, replicado por questões de disponibilidade.

Atualizações para uma zona são manipuladas pelo servidor primário de nomes. Tais atualizações são aplicadas por modificar o banco de dados do servidor primário.

Servidores de nome secundários não acessam a base diretamente. Em vez disso, pedem ao servidor primário que lhe transfiram seu conteúdo. Isso se chama **transferência de zona**.

Um banco de dados DNS é implementado como uma pequena coleção de arquivos. Destes, os mais importantes contêm todos os registros de recursos para todos os nós de uma determinada zona.

Com isso, nós podem ser identificados de forma simples, por meio de seus nomes de domínio. Assim, a noção de um identificador de nó se reduz a um index implícito de arquivo. Um arquivo é mais ou menos como abaixo:

Name	Record type	Record value
cs.vu.nl.	SOA	star.cs.vu.nl. hostmaster.cs.vu.nl. 2005092900 7200 3600 2419200 3600
cs.vu.nl.	TXT	"VU University - Computer Science"
cs.vu.nl.	MX	1 mail.few.vu.nl.
cs.vu.nl.	NS	ns.vu.nl.
cs.vu.nl.	NS	top.cs.vu.nl.
cs.vu.nl.	NS	solo.cs.vu.nl.
cs.vu.nl.	NS	star.cs.vu.nl.
star.cs.vu.nl.	A	130.37.24.6
star.cs.vu.nl.	A	192.31.231.42
star.cs.vu.nl.	MX	1 star.cs.vu.nl.
star.cs.vu.nl.	MX	666 zephyr.cs.vu.nl.
star.cs.vu.nl.	HINFO	"Sun" "Unix"
zephyr.cs.vu.nl.	A	130.37.20.10
zephyr.cs.vu.nl.	MX	1 zephyr.cs.vu.nl.
zephyr.cs.vu.nl.	MX	2 tornado.cs.vu.nl.
zephyr.cs.vu.nl.	HINFO	"Sun" "Unix"
ftp.cs.vu.nl.	CNAME	soling.cs.vu.nl.
www.cs.vu.nl.	CNAME	soling.cs.vu.nl.
soling.cs.vu.nl.	A	130.37.20.20
soling.cs.vu.nl.	MX	1 soling.cs.vu.nl.
soling.cs.vu.nl.	MX	666 zephyr.cs.vu.nl.
soling.cs.vu.nl.	HINFO	"Sun" "Unix"
vucs-das1.cs.vu.nl.	PTR	0.198.37.130.in-addr.arpa.
vucs-das1.cs.vu.nl.	A	130.37.198.0
inkt.cs.vu.nl.	HINFO	"OCE" "Proprietary"
inkt.cs.vu.nl.	A	192.168.4.3
pen.cs.vu.nl.	HINFO	"OCE" "Proprietary"
pen.cs.vu.nl.	A	192.168.4.2
localhost.cs.vu.nl.	A	127.0.0.1

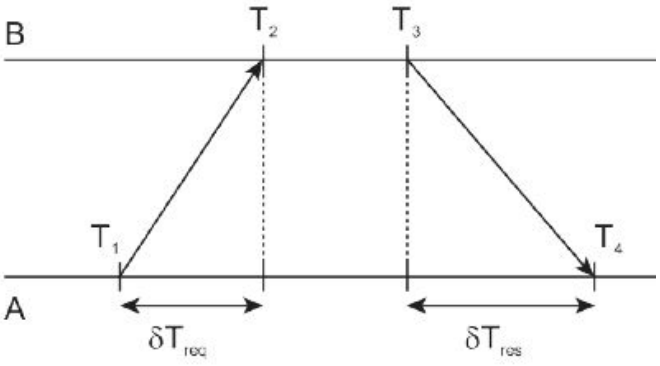
CAPÍTULO 6 - SINCRONIZAÇÃO

Sincronizações de relógio são úteis porque o tempo é ambíguo em SDs. Temporização exata é importante para vários programas que dependem de marcas de tempo, como os de controle de bolsa ou de sincronia de represas.

Defasagem de relógio: ocorre porque os cristais que mantêm os relógios funcionam em taxas ligeiramente diferentes, o que faz com que os relógios gradativamente saiam de sincronia. Por conta disso, é importante sincronizá-los com relógio do mundo real, bem como sincronizá-los entre si.

Protocolo de Tempo de Rede

Para que máquinas ajustem seus horários, elas podem consultar um servidor externo. Mas, para isso, precisam levar em conta o atraso da troca de mensagens. Para isso, o deslocamento pode ser calculado como: ((T2 - T1)+(T3-T4))/2.



Se essa fórmula der negativo, significa que A (quem mandou T1) deve atrasar seu relógio em relação a B. Mas o tempo não pode andar para trás. Assim, para realizar esse atraso, basta diminuir ligeiramente (de 10ms para 9ms, por exemplo) a frequência das interrupções. O mesmo princípio pode ser

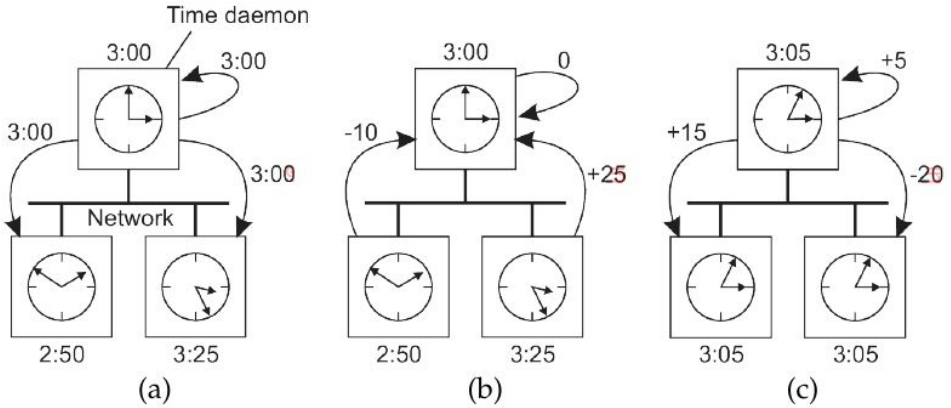
aplicado para adiantamento, onde a frequência de interrupções é aumentada (de 10ms para 11ms.) Se o resultado der igual a 0, nada precisa ser feito.

No caso dos servidores NTP, há ainda um cálculo de atraso: ((T2-T1)+(T4-T3))/2. Aí, guarda-se em *buffer* oito pares de deslocamento e de atraso. Depois, verifica-se qual valor mínimo foi encontrado para esse atraso. Esse valor passa a ser a melhor estimativa para o atraso entre os dois servidores; e o deslocamento associado a ele passa a ser a melhor estimativa confiável de deslocamento entre os servidores.

Além disso, no NTP, se o servidor consultante tiver a hora diferente do servidor consultado, e o estrato do servidor consultante for menor do que o consultado, é o consultado que precisa arrumar sua hora. O estrato de valor igual a 1 é dado àqueles servidores que tenham acesso direto a um relógio atômico ou a outros relógios com precisão científica determinada.

Algoritmo de Berkley

Nesse algoritmo, há um daemon que regularmente consulta o horário de todas as máquinas da rede. Aí, faz-se a média dos horários, e manda às outras máquinas que atrasem ou adiantem seus relógios, para que cheguem à média.



Sincronização de relógios em redes sem fio

Em redes sem fio ou de sensores, devido às limitações de recursos ou custos de salto entre redes, fica difícil realizar a comunicação para ajustar o tempo. Para resolver isso, usa-se a Sincronização em Broadcast de Referência (RBS.) Esse algoritmo é basicamente o mesmo que o de Berkley, exceto que a máquina remetente não sincroniza seu horário com as demais. O horário é passado via broadcast, daí seu nome.

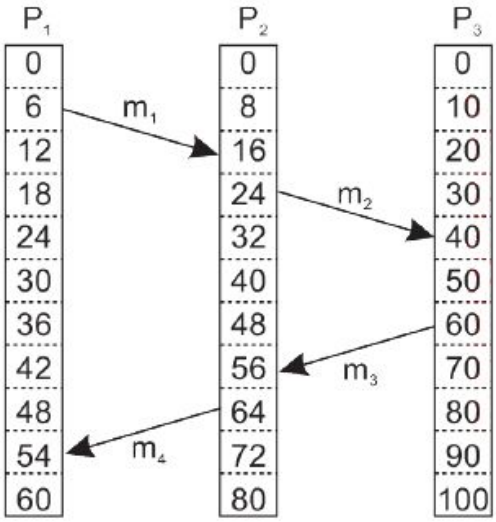
Relógios Lógicos de Lamport

Se dois processos não interagem, não é necessário que seus relógios sejam sincronizados, pois a falta de sincronia é inobservável. Além disso, a hora absoluta em que as coisas ocorrem não importa; o que importa é que os processos concordem com a ordem com que os eventos ocorreram.

Para isso, se há um evento *a* que acontece antes de um *b*, escreve-se a->b. Nessa notação, entende-se que todos os processos concordam que o evento *a* aconteceu antes de um evento *b*. Isso é transitivo, o que quer dizer que se a->b e b->c, então a->c.

Se a é um evento de um processo, e b é um evento do mesmo processo, e a ocorre antes de b, então a->b é verdadeiro. Além disso, se a é um evento de envio de mensagem, e b o evento de recebimento dessa mensagem, então a->b também é verdadeiro, porque não se pode receber mensagens antes delas terem sido escritas.

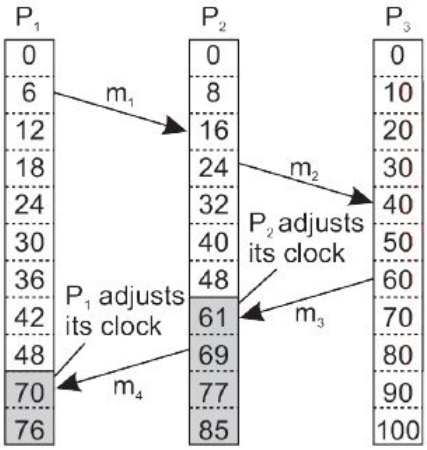
Eventos concorrentes: são eventos que não trocam mensagens entre si e que ocorrem em processos diferentes. Assim, nada pode ou nada precisa ser dito a respeito deles. Logo, nem a->b nem b->a são verdadeiras quando esses eventos a e b são concorrentes.



O algoritmo, por sua vez, funciona assim: suponha que haja 3 processos, P1, P2 e P3, e que eles estejam distribuídos entre quaisquer máquinas, em cujos relógios respectivamente pulsem de 6 em 6, de 8 em 8, e 10 em 10 vezes. Suponha que no tempo P1 de 6, envie-se uma mensagem a P2, que é recebida no tempo 16 de P2. Suponha que essa mensagem seja enviada a P3 no tempo 24 de P2, e que ela seja recebida no tempo 40 de P3. Agora, no tempo 60 de P3, deseja-se enviar essa mensagem de volta a P2. Como se realiza isso de forma a não voltar no tempo, como aconteceria na imagem ao lado?

Para resolver, bastaria que, quando P2 recebesse a mensagem de P3 (que saiu no tempo P3 de 60), P2 considerasse seu tempo como 61, isso é, o tempo enviado de P3 + 1. Agora suponha que P2, em seu próximo tempo, queira retornar a mensagem à P1. Ora, sem essa mudança, o P2 enviaria em seu tempo 64 a mensagem de P1, que a receberia no tempo 54, conforme a imagem acima. Isso seria desastroso.

Como o tempo de P2 foi corrigido para 61 após receber a mensagem de P3 que foi enviada no tempo P3 de 60, P2 passará um ciclo antes de enviar a mensagem a P1, ou seja, ele a enviará para P1 no seu tempo 61+8=69. P1, por sua vez, precisará recebê-la no tempo de P2+1, ou seja, 69+1=70. E ele seguirá seu ciclo normalmente, indo de 6 em 6, conforme a imagem ao lado.



Multicast totalmente ordenado: operação pela qual todas as mensagens são entregues na mesma ordem a cada receptor. Para isso, são usados os relógios lógicos de Lamport.

Considere que: nenhuma mensagem será perdida; todas serão recebidas na ordem em que foram enviadas; cada mensagem sempre transportará a marca de tempo correspondente ao tempo lógico de seu remetente.

Quando a mensagem é enviada ao multicast, ela é enviada conceitualmente ao remetente. Quando um processo recebe uma mensagem, ela é colocada em uma fila de cache local, ordenada por tempo. Esse receptor envia mensagens multicast de reconhecimento aos outros processos. A certa altura, todos os processos terão a mesma cópia da fila local.

O processo só pode entregar uma mensagem enfileirada à aplicação que estiver executando quando essa mensagem estiver no início da fila e tiver sido reconhecida por cada um dos outros processos. Quando isso acontece, a mensagem é retirada da fila e entregue à aplicação; reconhecimentos associados são removidos.

Como todos processos têm a mesma cópia da fila, todas as mensagens são entregues na mesma ordem em todos os lugares. Assim se estabelece o multicast ordenado usando-se relógios de Lamport.