MC714

Sistemas Distribuídos

1° semestre, 2017

Abordagens hierárquicas – visão geral

- Rede dividida em conjunto de domínios
  - Podem ser divididos em subdomínios
- Domínio mais alto: abrange a rede toda
- Domínio-folha: mais baixo na rede

- Dir(D): nó diretório que monitora entidades dentro de um domínio *D.*
- Dir(D), com D sendo domínio mais alto: nó (de diretório) raiz
- Fig. 81

Abordagens hierárquicas – visão geral

- Cada entidade tem registro de localização no nó de diretório do domínio a que pertence.
  - Registro de localização para entidade E no nó de diretório N para um domínio-folha D contém endereço corrente de E em D.
  - Em um nível mais alto: nó de diretório N', domínio D' (que contém D) contém registro de localização de E como um ponteiro para N.
  - Nó raiz: registro de localização para cada entidade com ponteiro para nó de diretório do próximo subdomínio onde entidade está localizada.

Abordagens hierárquicas – visão geral

- Entidade pode ter vários endereços
  - Entidade replicada
  - Endereços no domínio folha D1 e D2
  - Nó de diretório que contém D1 e D2: dois ponteiros para entidade
- Fig 82.

Abordagens hierárquicas – consulta

- Cliente *C* deseja localizar *E*
  - Emite requisição de consulta ao nó de diretório do domínio folha D, onde *C* reside.
  - D tem registro de E?
    - Se sim, E está no domínio de D. Pode responder.
    - Se não, repassa requisição ao seu pai D'.
    - D' tem registro de E?
      - ...
    - Chega a M, que retorna registro de E → E está em dom(M).
    - Registro do nó folha sob M tem endereço de E. Retorna.
- Fig 83.

Abordagens hierárquicas – atualização

- Entidade E cria réplica em domínio folha D.
- Insere seu endereço em dir(D).
- dir(d) repassa ao nó pai D'.
- D' repassa ao nó pai .....
  - Até chegar a nó M que já tem registro de outra réplica de E.
- Fig. 84
- Remoção: análoga à inserção.
  - Sobe até encontrar registro para E que fique não vazio após a remoção, ou até a raiz.

Nomeação estruturada

- Nomes simples são bons para máquinas, mas não são convenientes para seres humanos.
- Nomeação estruturada: composição de nomes simples.
  - Arquivos, hospedeiros na Internet
- Espaço de nomes
  - Grafo dirigido rotulado, com dois tipos de nós: nós folha e nós de diretório

Nomeação estruturada

- Nó folha: entidade nomeada, sem saídas.
- Nó de diretório: vários ramos de saída, rotulados.
  - Armazena tabela de entradas (rótulo do ramo, identificador do nó)
- Nó raiz: somente saídas
- Fig. 85
- Caminhos referenciados pela sequencia de rótulos
- N:<label-1, label-2, ..., label-n> → nome de caminho
- Caminho absoluto: 1º nó do nome é raiz
- C.C.: caminho relativo

Resolução de nomes

- Processo de busca de um nome: resolução de nome
- Mecanismo de fechamento: seleção do nó inicial de um espaço de nomes onde resolução começa.
- Apelidos (aliases)
  - Ponteiros estritos
  - Ponteiros simbólicos
- Resolução de nomes pode ocorrer em mais de um espaço de nomes
  - Sistema de arquivo montado
  - Espaço de nomes externo (ex.: NFS).

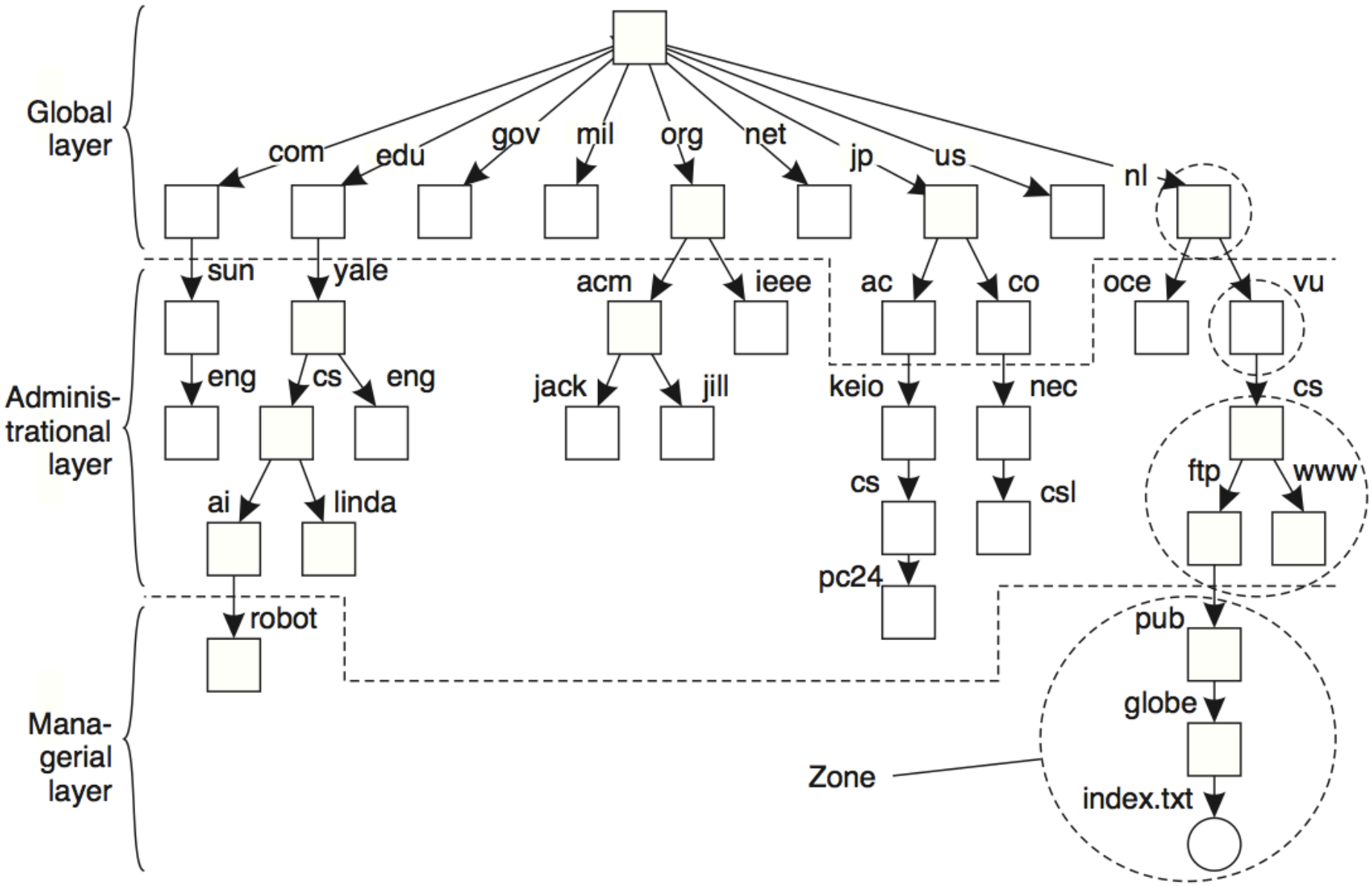
Espaço de nomes - implementação

- SDs: distribuir implementação do espaço de nomes por vários servidores de nomes.
  - Distribuir nós do grafo de nomeação.

- Grande escala → comum em hierarquia
  - Camada global
  - Camada administrativa
  - Camada gerencial

Espaço de nomes - implementação

- Camada global
  - Nós de nível mais alto (raiz e nós próximos)
  - Mais estáveis → tabelas de diretório raramente mudam
- Camada administrativa
  - Nós de diretório gerenciados por uma única organização
  - Representam grupos de entidades que pertencem à mesma organização ou unidade administrativa
  - Mudanças com maior frequência que na camada global
- Camada gerencial
  - Mudança periódica
  - Nós mantidos por administradores e usuários


Global layer
Adminis-trational layer
Mana-gerial layer
com
edu
gov
mil
org
net
jp
us
nl
sun
yale
acm
ieee
ac
co
oce
vu
eng
cs
eng
jack
jill
keio
nec
cs
ftp
www
ai
linda
cs
csl
pc24
robot
pub
globe
index.txt
Zone

Espaço de nomes - implementação

- Camada global
  - Alta disponibilidade: se falha, grande parte do espaço fica inalcançável.
  - Desempenho: Baixa taxa de mudança; Cache local é útil. Não precisam responder tão rapidamente.
  - Replicação pode ser aplicada.
- Camada administrativa
  - Se falhar, muitos recursos dentro da organização podem ficar inalcançáveis.
  - Deve responder mais rapidamente que camada global.
- Camada gerencial
  - Indisponibilidade temporária afeta poucos usuários
  - Desempenho é crucial;
  - Muda com frequencia → cache pode não funcionar muito bem.

Espaço de nomes - implementação

Item	Global	Administrational	Managerial
Geographical scale of network	Worldwide	Organization	Department
Total number of nodes	Few	Many	Vast numbers
Responsiveness to lookups	Seconds	Milliseconds	Immediate
Update propagation	Lazy	Immediate	Immediate
Number of replicas	Many	None or few	None
Is client-side caching applied?	Yes	Yes	Sometimes

Figure 5-14. A comparison between name servers for implementing nodes from a large-scale name space partitioned into a global layer, an administrational layer, and a managerial layer.

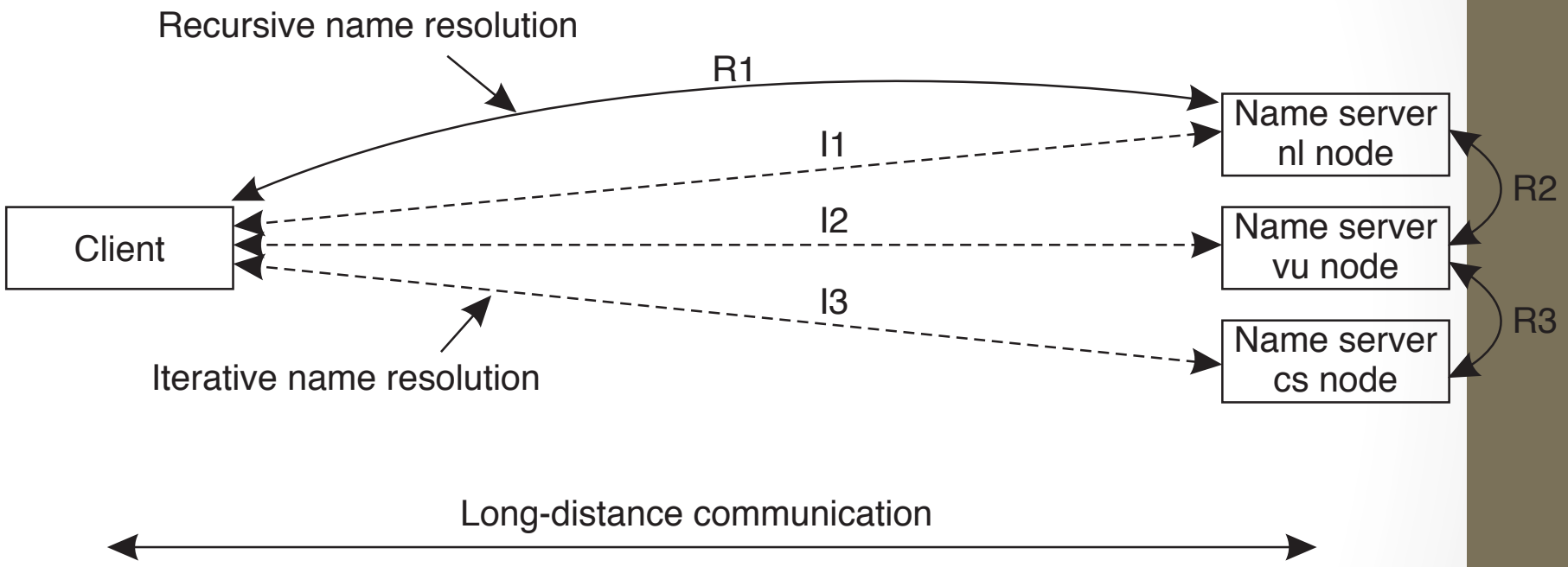
Resolução de nomes - implementação

- Resolução iterativa
  - Resolvedor entrega nome completo ao servidor-raiz
  - Servidor resolve nome até onde conhece e retorna endereço do servidor de nome associado
  - Resolvedor de nome do cliente entra em contato com servidor retornado
  - ...
  - Fig. 86

Resolução de nomes - implementação

- Resolução recursiva
  - Ao invés de retornar resultado intermediário, o próprio servidor consultado realiza consulta ao próximo nível.
  - Carga maior aos servidores de nomes.
  - Em geral servidores na camada global suportam somente resolução iterativa.
  - Cache mais eficiente que na iterativa.
  - Pode reduzir custo de comunicação.
  - Fig. 87

Resolução de nomes - implementação



Comunicação iterativa versus recursiva.

Resolução recursiva e cache

Servidor para nós	Deve resolver	Busca	Passa para filho	Recebe faz cache	Responde para cliente
cs	<ftp>	#<ftp>	—	—	#<ftp>
vu	<cs,ftp>	#<cs>	<ftp>	#<ftp>	#<cs> #<cs, ftp>
nl	<vu,cs,ftp>	#<vu>	<cs,ftp>	#<cs> #<cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>
root	<nl,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

Exemplo - DNS

- Domain Name System – DNS
- Resolve endereços IP a partir de nomes na Internet
- Espaço de nomes hierárquico: listagem de rótulos separados por pontos.

Name	Record type	Record value
cs.vu.nl	SOA	star (1999121502,7200,3600,2419200,86400)
cs.vu.nl	NS	star.cs.vu.nl
cs.vu.nl	NS	top.cs.vu.nl
cs.vu.nl	NS	solo.cs.vu.nl
cs.vu.nl	TXT	"Vrije Universiteit - Math. & Comp. Sc."
cs.vu.nl	MX	1 zephyr.cs.vu.nl
cs.vu.nl	MX	2 tornado.cs.vu.nl
cs.vu.nl	MX	3 star.cs.vu.nl
star.cs.vu.nl	HINFO	Sun Unix
star.cs.vu.nl	MX	1 star.cs.vu.nl
star.cs.vu.nl	MX	10 zephyr.cs.vu.nl
star.cs.vu.nl	A	130.37.24.6
star.cs.vu.nl	A	192.31.231.42
zephyr.cs.vu.nl	HINFO	Sun Unix
zephyr.cs.vu.nl	MX	1 zephyr.cs.vu.nl
zephyr.cs.vu.nl	MX	2 tornado.cs.vu.nl
zephyr.cs.vu.nl	A	192.31.231.66
www.cs.vu.nl	CNAME	soling.cs.vu.nl
ftp.cs.vu.nl	CNAME	soling.cs.vu.nl
soling.cs.vu.nl	HINFO	Sun Unix
soling.cs.vu.nl	MX	1 soling.cs.vu.nl
soling.cs.vu.nl	MX	10 zephyr.cs.vu.nl
soling.cs.vu.nl	A	130.37.24.11
laser.cs.vu.nl	HINFO	PC MS-DOS
laser.cs.vu.nl	A	130.37.30.32
vucs-das.cs.vu.nl	PTR	0.26.37.130.in-addr.arpa
vucs-das.cs.vu.nl	A	130.37.26.0

Nomeação baseada em atributos

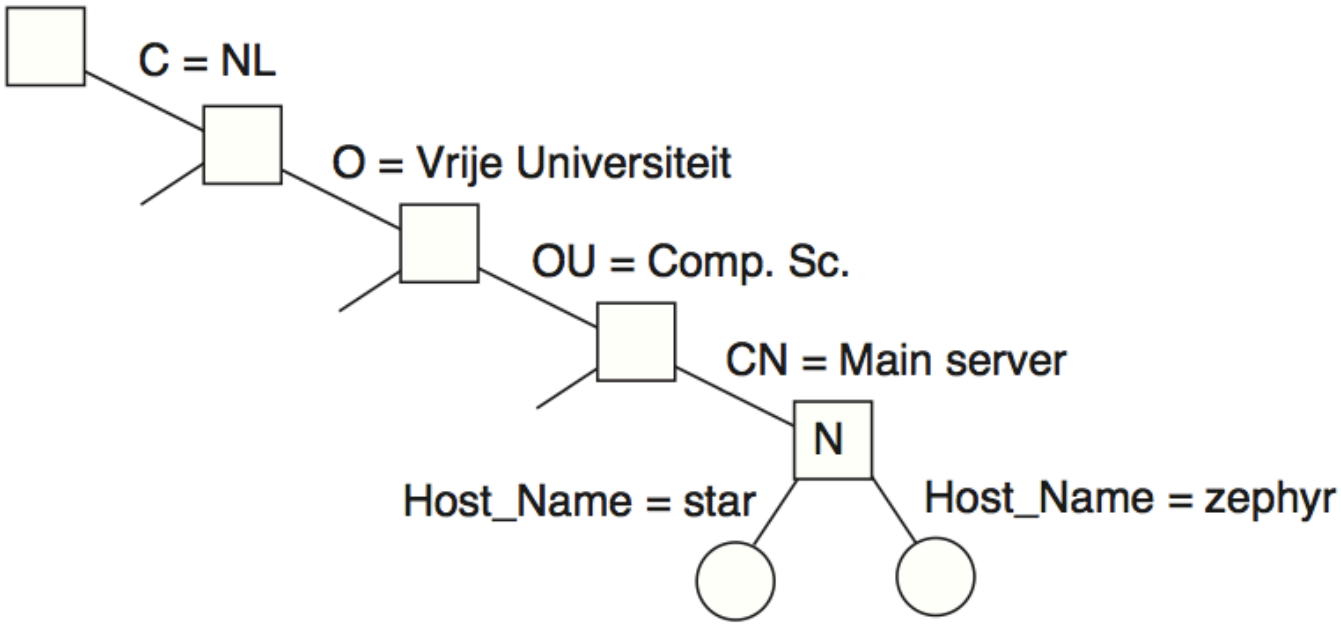
- Entidade tem conjunto associado de atributos
- Podem ser usados para buscar entidades
- “Páginas amarelas”
- Também chamados “serviços de diretório”
- Atributos podem ser descritos de forma diferente por pessoas diferentes
  - Padronização de descrição de atributos.
  - Estrutura de descrição de recurso – resource description framework – RDF
  - Sujeito, predicado, objeto – (Pessoa, nome, Alice)

Implementação hierárquica - LDAP

- Nomeação estruturada + nomeação baseada em atributos
- Protocolo leve de acesso a diretório – LDAP (lightweight directory access protocol)
- Registro: pares (atributo,valor)
- DIB – directory information base / base de informações de diretório
  - Conjunto de todas as entradas de diretório
- Fig. 88

Implementação hierárquica - LDAP

- Ex.: /C=NL/O=Vrije Universiteit/OU=Comp. Sc.
- Nome globalmente exclusivo (similar ao DNS – nl.vu.cs).
- Resulta em hierarquia
  - Árvore de informações de diretório – DIT
  - Pode ser distribuída – Agentes de serviço de diretório (DSA)
    - Cada pedaço análogo a zona em DNS
    - Cada DSA se comporta de maneira parecida com um servidor de nomes
- LDAP: recursos de busca através da DIB.
  - answer=search("&(C=NL)(O=Vrije Universiteit)(OU=*)(CN=Main server)")



Atributo	Valor
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Comp. Sc.
CommonName	Main server
Host_Name	star
Host_Address	192.31.231.42

Atributo	Valor
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Comp. Sc.
CommonName	Main server
Host_Name	zephyr
Host_Address	137.37.20.10

Implementação descentralizada

- P2P para sistemas de nomeação baseados em atributos
- Mapear (atributo,valor) de forma eficiente
  - Resulta em busca eficiente
  - Evita busca exaustiva

- Mapeamento para DHTs
- Redes de sobreposição semântica

Implementação descentralizada

- Entidade/Recurso descritos por meio de atributos possivelmente organizados em hierarquia
- AVTree – attribute-value tree
- Usada para codificação que mapeia para um sistema baseado em DHT
- Fig. 89
- Questão: transformar AVTrees em conjuntos de chaves na DHT
- Um hash para cada caminho

Implementação descentralizada

- h1: hash(tipo-livro)
- h2: hash(tipo-livro-autor)
- h3: hash(tipo-livro-autor-Tolkien)
- h4: hash(tipo-livro-título)
- h5: hash(tipo-livro-título-LOTR)
- h6: hash(gênero-fantasia)
- Fig. 90
- Nó responsável por um hash h_i mantém (referência para) o recurso.

Implementação descentralizada

- Redes de sobreposição semântica
- Premissa: consultas originadas no nó P estão relacionadas com os recursos que ele tem.
- P: conjunto de ligações com nós semanticamente próximos → visão parcial
- Rede de sobreposição semântica
- Rede de sobreposição aleatória como base
- Sobreposição semântica: k vizinhos mais semelhantes
- Fig. 91

Implementação descentralizada

- Definir similaridade: muitas formas
- Ex. 1: Nome de arquivo
  - construir rede semântica de acordo com respostas a consultas
  - Se nós vizinhos não respondem, faz broadcast (limitado)
- Ex. 2: função proximidade semântica
  - Função que conta número de arquivos em comum entre dois nós
  - Meta: otimizar função proximidade
  - Camada superior pode manter lista de vizinhos semanticamente próximos por meio de gossiping
    - Envio para um par meus vizinhos semelhantes a ele

Sincronização

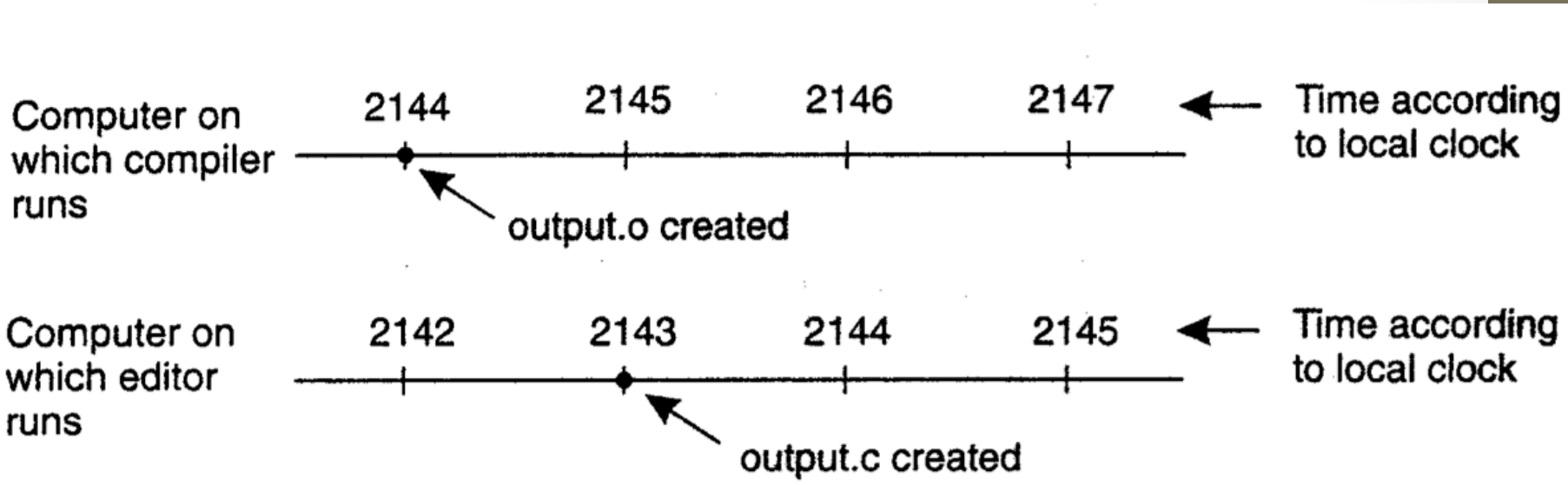
Sincronização

- Evitar acesso concorrente a recursos
- Concordar com ordem que eventos ocorreram
  - Qual mensagem foi enviada primeiro?
- Sincronização baseada em tempo real
- Sincronização relativa
- Pode necessitar de coordenador: eleição de líder

Sincronização de relógios

Sincronização de relógios

- Ex.: Make



- É possível sincronizar todos os relógios em um sistema distribuído?

Sincronização de relógios

- É possível sincronizar todos os relógios em um sistema distribuído?

Relógios físicos

Relógios físicos

- “Temporizador”
  - Número de ciclos de relógio após uma data inicial fixa no sistema.
- Sistema com N computadores
  - Defasagem de relógio

Medição do tempo

- Problema: tempo solar e tempo de relógios atômicos divergem.
- Sol: dia solar de 24h
  - 1s = 1/86.400 de um dia solar
- Relógio atômico: transições por segundo de átomo de césio 133.
  - 1s = 9.192.631.770 transições.
- Hoje, 86.400 segundos TAI (tempo atômico internacional) equivalem 3ms a menos que um dia solar médio.

Medição do tempo

- Solução: adicionar segundos à hora do relógio quando diferença > 800ms
  - Bureau International de l'Heure - BIH
- Sistema de medição baseado em segundos TAI constantes → Hora (ou tempo) coordenada universal (UTC)

- UTC: NIST broadcast por rádio WWV (precisão prática +- 10ms) e por satélite (0,5ms)

Sistema de posicionamento global

- Global positioning system – GPS
- Sistema distribuído de determinação de posição geográfica baseado em satélites e lançado em 1978.
- 29 (31) satélites a ~20.000km de altura
- Cada satélite tem até 4 relógios atômicos calibrados periodicamente
- Satélite transmite sua posição em broadcast com marcas de tempo
- Receptor na Terra calcula sua posição

Sistema de posicionamento global

- Leva um certo tempo para que os dados sobre posição de um satélite cheguem ao receptor.
- Relógio do receptor não está em sincronia com o do satélite.

- Relógios não estão perfeitamente sincronizados
  - Correção de 38 microssegundos por dia devido à relatividade (gravidade +45; dilatação do tempo -7)
  - Não leva em conta segundos extras UTC
- Velocidade de propagação não é constante
- Terra não é esfera perfeita

Algoritmos de sincronização de relógios

- Se uma máquina tiver receptor WWV, meta é manter todas as outras máquinas sincronizadas com ela.
- Se nenhuma tem receptor WWV, cada uma monitora seu próprio horário e objetivo é manter todas as máquinas com relógio o mais próximo possível.
- Cada máquina tem temporizador que provoca interrupção H vezes por segundo.
  - Manipulador de interrupção soma 1 a um relógio de software
  - Relógio mantém número de tics que ocorreram desde um instante pré-determinado.

Algoritmos de sincronização de relógios - Modelo

- C = valor do relógio
- Para UTC = t , $C_p(t)$ é valor do relógio da máquina *p*.
  - Ideal: $C_p(t)$ = t para todo p e t → $C'_p(t) = dC/dt$ seria 1
  - $C'_p(t)$ é a frequência do relógio de *p* no tempo t
  - $C'_p(t)$ - 1 é a defasagem do relógio de *p*
    - magnitude da diferença entre relógio de *p* e o relógio perfeito
  - $C_p(t)$ – t é o deslocamento em relação a uma hora específica

Algoritmos de sincronização de relógios - Modelo

- Temporizadores reais têm imprecisão
  - H = 60 → 216.000 ciclos por hora
  - Erro relativo em chips temporizadores: 10^{-5} → 215.998 a 216.002 ciclos por hora
- Se existir uma constante ρ tal que $1 - \rho \leq \frac{dC}{dt} \leq 1 + \rho$

então o temporizador está funcionando dentro da especificação.

- ρ: taxa máxima de deriva → especificado pelo fabricante
- Fig 93.

Algoritmos de sincronização de relógios - Modelo

- Defasagem Δt após sincronização: até $2\rho\Delta t$ se derivarem em direções opostas.
- Para garantir defasagem máxima δ, relógios devem ser sincronizados no mínimo a cada $\frac{\delta}{2\rho}$ segundos.

- Para esse modelo, diferença nos algoritmos de sincronização de relógios está na maneira como essa sincronização periódica é feita.

Algoritmos de sincronização de relógios

- Abordagem 1: clientes consultam servidor de tempo que possui hora precisa.
- Atrasos nas mensagens farão com que hora fique desatualizada.
- Fig. 94
- A envia req. a B com marca de tempo T1
- B marca T2 (sua hora), tempo que recebeu
- B envia T3 e T2.
- A marca T4. desloc = $\frac{(T2 - T1) + (T3 - T4)}{2}$

Algoritmos de sincronização de relógios

- Se relógio de A estiver adiantado, A precisaria atrasar seu relógio → consequências ruins.

- Pode ser feito de forma gradual, fazendo relógio andar mais devagar (ou mais rápido).

Protocolo de tempo de rede (NTP)

- Ajustado entre pares de servidores.
- B também consulta A para saber sua hora.
- Usa estimativa anterior de deslocamento, e atraso:

$$\delta = \frac{(T2 - T1) + (T4 - T3)}{2}$$

- Obtém 8 pares (deslocamento,atraso).
- Valor mínimo de atraso e respectivo deslocamento são adotados.

Protocolo de tempo de rede (NTP)

- B também poderia ajustar seu relógio em relação a A
- Se B é mais preciso, não deveria ajustar
- NTP divide servidores em estratos
  - Estrato 1: relógio de referência (WWV, relógio atômico)
  - A contacta B → A só ajusta seu relógio se estrato de A é maior
  - Após sincronização, A torna-se um nível mais alto que B
    - Se estrato de B é k, A torna-se k+1

Algoritmo de Berkeley

- NTP: servidor é passivo
- Unix/Berkeley: servidor é ativo → pergunta hora das máquinas de tempos em tempos
- Média das repostas e envia para as máquinas ajustarem seus relógios
- Adequado para sistemas sem um receptor WWV.
- Relógio do daemon ajustado manualmente de tempos em tempos.
- Fig. 95

Sincronização em redes sem fio

- Redes sem fio, em particular redes sensores: recursos restritos.
- Algoritmos diferentes para sincronização de relógios.
- Uma abordagem: sincronização em broadcast de referência – RBS.
- Não adota como premissa único nó com valor de hora real.
- Visa sincronização interna (não visa UTC).
- Permite somente que receptores sincronizem.

Sincronização em redes sem fio

- Remetente transmite mensagens de referência em broadcast.
- Obs.: tempo de propagação é aproximadamente constante para qualquer nó (sem múltiplos saltos).
- Atraso desconsidera tempo de preparação da mensagem e tempo gasto no adaptador de rede
- Protocolos como NTP adicionam marca de tempo antes de ser passada para a interface de rede
  - Redes sem fio: fatores não-determinísticos como contenção → variabilidade
  - Eliminados em RBS → usa tempo de entrega no receptor

Sincronização em redes sem fio

- Nó transmite mensagem de referência *m.*
- Cada nó *p* registra hora $T_{p,m}$ em que recebeu m.
  - Obtida do relógio local de *p*
- Dois nós *p* e *q* trocam seus respectivos horários de entrega para estimar deslocamento relativo

$$Deslocamento[p,q] = \frac{\sum_{k=1}^{M}(T_{p,k} - T_{q,k})}{M}$$

- onde M é o número de mensagem de referência enviadas

Sincronização em redes sem fio

- Alternativa: regressão linear

$$Deslocamento[p,q](t) = \alpha t + \beta$$

- α e β calculados pelos pares ($T_{p,k}$, $T_{q,k}$)