MC714

Sistemas Distribuídos

2° semestre, 2014

Nomeação

Nomeação

- Nomeação simples
  - Soluções simples (broadcast e multicast)
  - Soluções baseadas em localização nativa (home)
  - Tabelas hash distribuídas (DHT)
  - Abordagens hierárquicas
- Nomeação estruturada
- Nomeação baseada em atributos

Tabelas Hash Distribuídas (DHTs)

- Abordagem 2: tabela de derivação (finger table).
  - Cada nó mantém *m* entradas de tal forma que a i-ésima entrada aponta para o primeiro nó que sucede *p* por no mínimo 2^{i-1}.

$$FT_p[i] = succ(p + 2^{i-1})$$

  - Para consultar chave k, repassa para nó *q com índice j*:

$$q = FT_p[j] \leq k \leq FT_p[j+1]$$

  - Finger tables devem ser mantidas atualizadas
    - q == pred(succ(q+1)).
    - para todo i, achar succ(q+2^{i-1}).

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

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

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

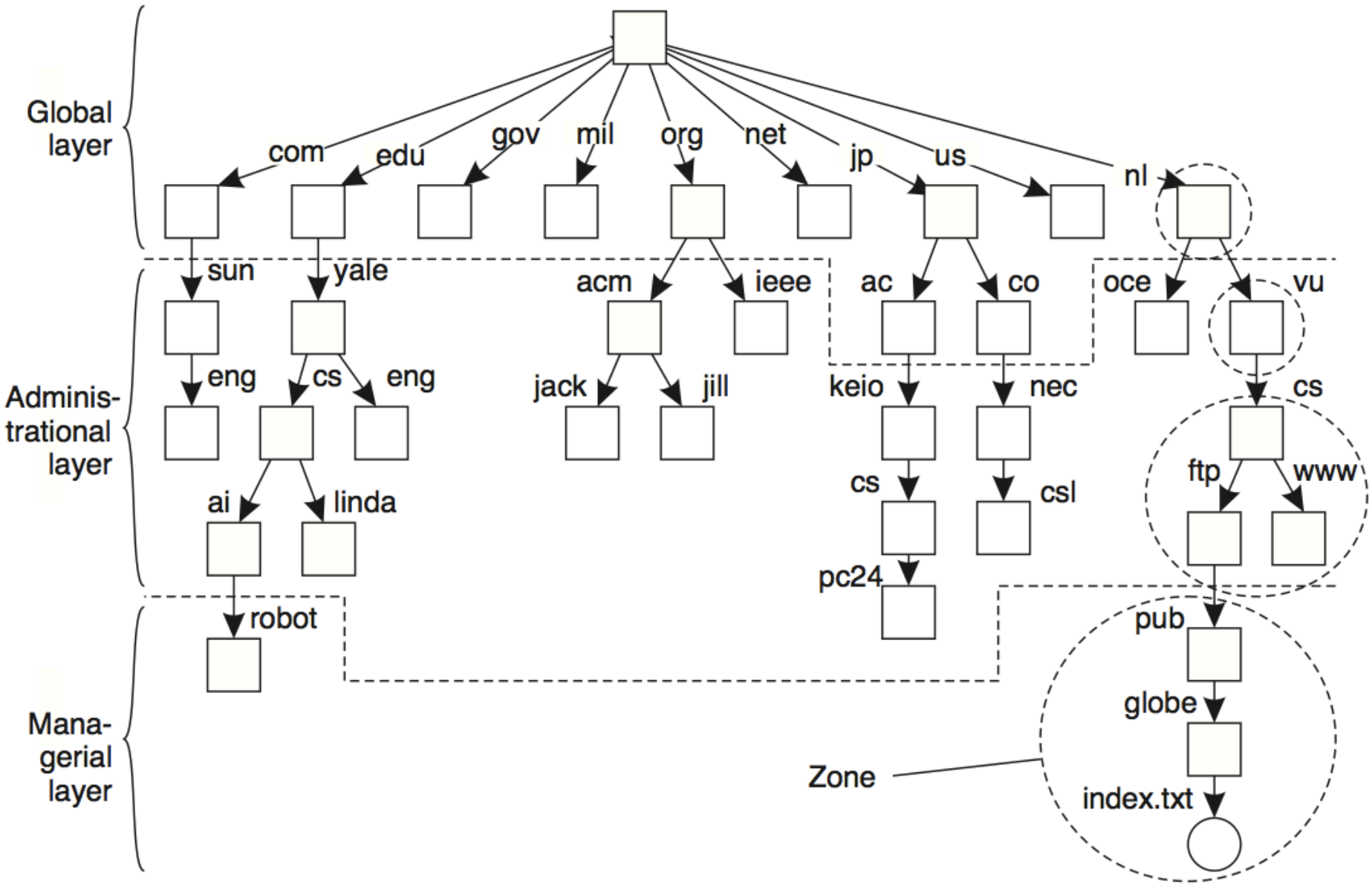
Espaço de nomes - implementação

- 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 local
- 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
  - 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.

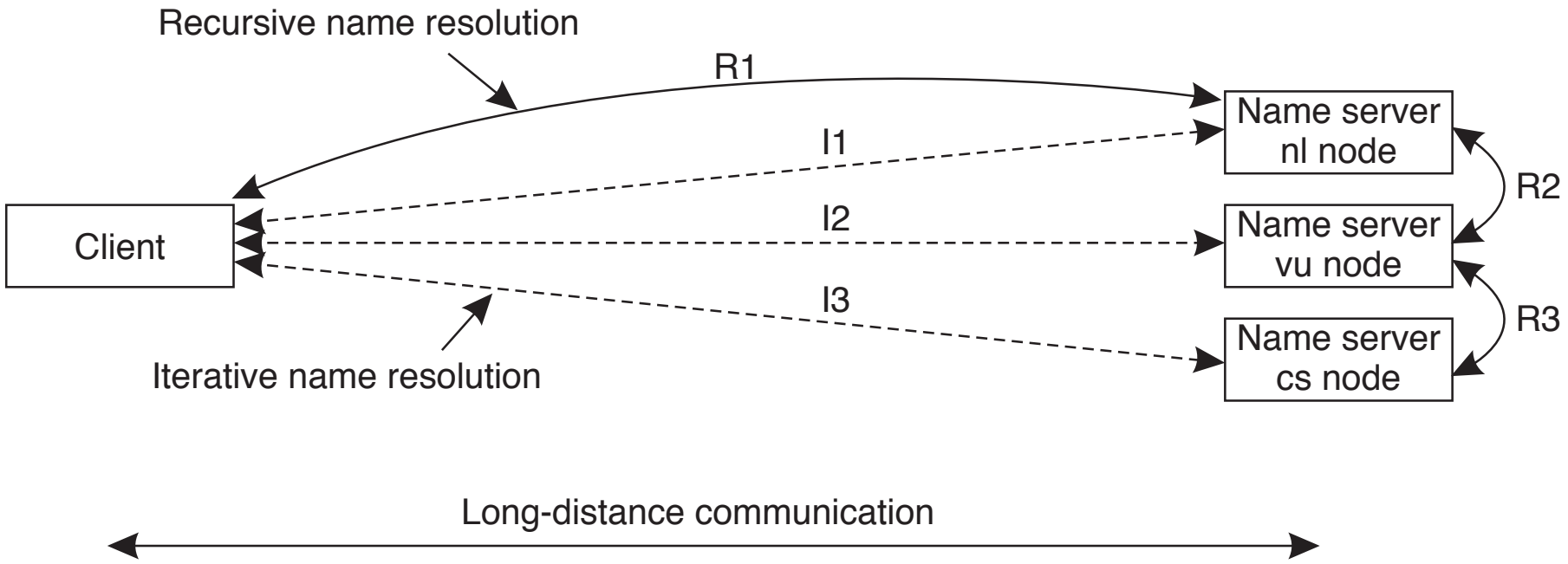
Resolução de nomes - implementação

- Resolução iterativa
  - Resolvedor entrega nome completo ao servidor-raiz
  - Servidor resolve nome até onde conhece e retorna endereço do servidor de nomes 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 atributo

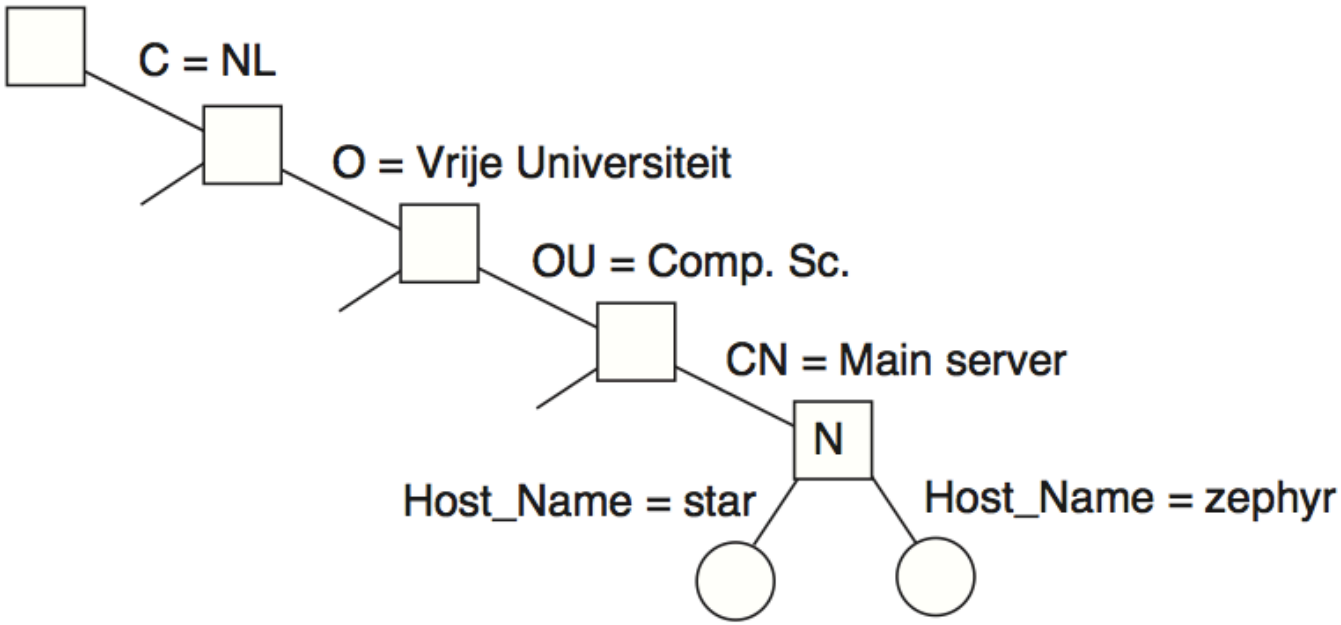
- Entidade tem conjunto associado de atributos
- Podem ser usados para buscar entidades
- “Páginas amarelas”
- Também chamados “serviços de diretório”
- Atributos podem ser descritos de forma diferente por pessoas diferentes
  - Padronização de descrição de atributos.
  - Estrutura de descrição de recurso – 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
- RDN – relative distinguished name / nome relativo distinguido
- 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 hierárquica - LDAP

- Floresta de domínios LDAP
  - Servidor global de índices
- Outros serviços de diretório, como UDDI – universal directory and discovery integration – serviços web

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

Sincronização

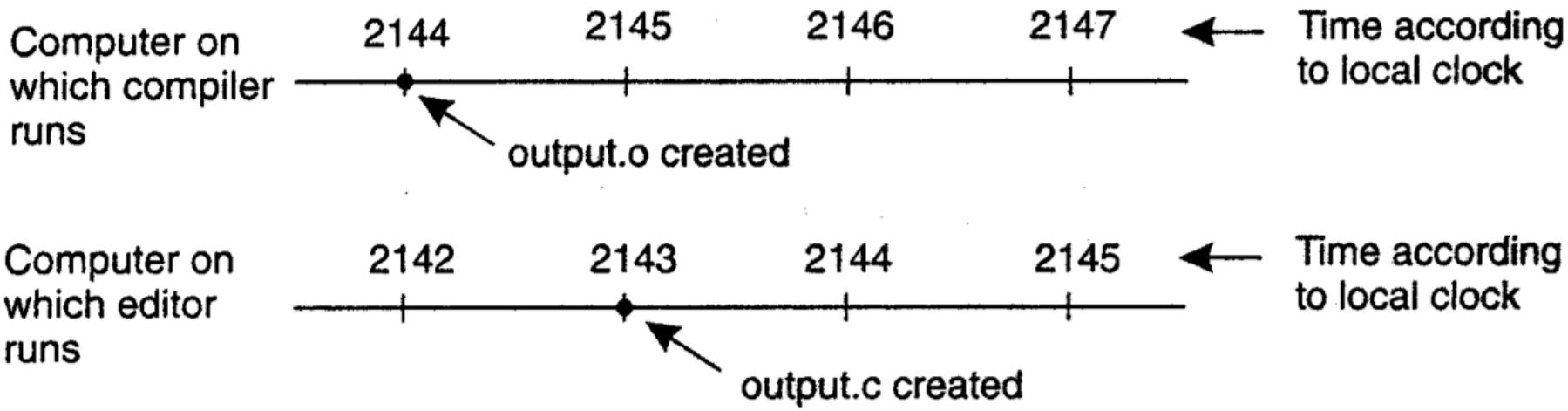
Sincronização

- 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 (32) 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

- Ex. 2D
- Supondo relógios sincronizados
- Eixo y = altura
- Eixo x = linha reta na superfície da Terra (nível do mar)
- Fig. 92

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