Protocolos em Redes de Dados

Aula 13

Tabelas de dispersão distribuídas e redes sobrepostas

Luís Rodrigues

FCUL

2005-2006

Sumário

- Introdução.
- DHTs.
- Chord.
- Pastry.

Tabelas de dispersão distribuída

- Como concretizar um directório escalável?
  - Distribuindo o directório por todos os nós.
  - cada nó mantém uma porção do directório.

Filiação

- Se o número de máquinas é pequeno, todos os membros do grupo conhecem o número e identificação de todos os outros membros.
- Uma regra determinista permite decidir quem fica responsável por cada entrada.
- Problema:
  - E se o número de participantes é tão grande que não se torna viável conhecer todos os outros participantes?

Usar uma tabela de dispersão

- Define-se um espaço de endereçamento vasto cujo tamanho é conhecido por todos.
  - Por exemplo, 32 ou 64 dígitos.
- Define-se uma função de dispersão que é também conhecida à partida por todos.

Tabela de dispersão

- As entradas do dicionário projectam-se no espaço de endereçamento através da função de dispersão.
  - Por exemplo, se:
    $H($ "ze cabra" $) = 102566$
    a informação referente a este famoso artista será guardada na posição 102566 do dicionário.

Rede sobreposta

- Desafio: como é que os nós se organizam de modo a
  - saberem quais as entradas que cada um tem de guardar
  - "encontrarem" o nó responsável por uma posição
  - isto sem serem obrigados a conhecerem o número e identificação de todos os participante

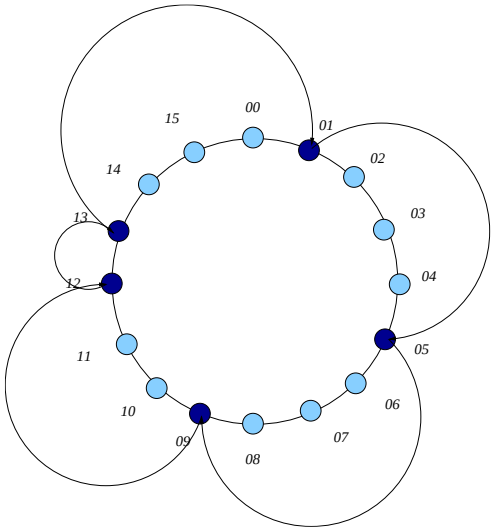
Rede sobreposta

- Os nós devem organizar-se numa rede, em que cada nó só é obrigado a conhecer um número reduzido de vizinhos.
- Apesar disso, deve ser possível encaminhar uma mensagem para qualquer vizinho, sabendo o seu identificador.
- Rede sobreposta (overlay network)
  - Assume-se que qualquer par de nós pode sempre estabelecer uma relação de vizinhança pois existem um sistema de encaminhamento por baixo (tipicamente o IP).

Identificação nos nós

- Cada nó assume uma posição aleatória no espaço de nomes.
  - Usando um gerador de números aleatórios ou aplicando uma função de dispersão a um identificador único local (ou seu endereço IP, por exemplo).
- Os nós organizam-se num anel lógico:
  - No mínimo, cada nó só tem de conhecer dois vizinhos: ou nó com identificador anterior e seguinte.

Rede em anel



Atribuição de entradas aos nós

- Um nó fica responsável por todas as entradas entre o identificador do seu antecessor no anel e seu identificador (inclusive).
  - No exemplo anterior, o nó de identificador 05 fica responsável pelas entradas $02, 03, 04$ e 05.

Encaminhamento no anel

- Para encontrar uma entrada basta ir rodando no anel até encontrar o nó responsável por essa entrada.
  - Quando se faz uma pergunta, pode-se indicar logo o endereço IP para onde deve ser enviada a resposta.

Entrada no anel.

- Imagine que o nó 07 quer entrar no anel.
- Só precisa de conhecer um membro qualquer.
- Usando entes membro, envia uma mensagem para si próprio.
  - A mensagem irá percorre o anel até chegar ao nó 05 que sabe que o próximo membro é o nó 09.
- Uma vez que o nó conhece o seu sucessor e antecessor, pode enviar mensagens de controlo para reconfigurar o anel.

re-distribuição das entradas

- Depois de entrar no anel, um nó pode iniciar a redistribuição da entradas.
  - Ficando responsável por algumas entradas que anteriormente estavam ao cuidado do seu sucessor.

Escalabilidade

- Cada nó só precisa de conhecer dois vizinhos.
- Problema:
  - O encaminhamento pode ser muito lento (no pior caso, percorrer todo o anel).

Optimizando o encaminhamento

- Mantendo atalhos.
  - Não necessitam de estar sempre coerentes: em último caso utiliza-se o anel para encaminhar as mensagens.
- Cada nó pode manter uma "cache" com endereço IP de alguns nós em diversos pontos do anel, a diferentes distâncias.
  - Por exemplo, o nó 00 pode manter o endereço IP do nó mais perto da posição 02, 04, 08, 16, 32, 64, etc.
  - Mais precisamente, o atalho de nível i do nó n mantém a identificação de *sucessor* $(n + 2^{i-1})$.
- Mantendo um número de entradas logaritmico em relação ao número de nós, é possível encaminhar com um número de saltos também logaritmico.

Entradas concorrentes no anel

- Não existe um mecanismo de exclusão mútua na entrada do anel.
- devido a este facto, a informação referente ao sucessor e antecessor pode ficar incoerente.
- Os nós executam periodicamente um procedimento de estabilização:
  - O nó p pergunta ao seu sucessor qual o seu antecessor.
  - Caso existe uma incoerência, repara-a corrigindo as ligações erradas.

Rede sobreposta e rede real

- Dois vizinhos na rede sobreposta podem estar muito afastados na rede física.
  - Isto pode levar a que o encaminhamento na rede subjacente seja sub-óptimo.
- Várias alternativas:
  - Distorcer a função de dispersão (pode ser difícil).
  - Optimizar os atalhos.

Optimização dos atalhos

- Damos só a intuição do algoritmo.
- Considere que o nó 01 quer encaminhar uma mensagem para o nó 15.
- Para optimizar o encaminhamento pode tentar manter um atalho para um nó "perto" (na rede sobreposta) do nó 15.
  - Por exemplo o nó 12 ou o nó 13.
- Existindo várias alternativas, pode escolher um atalho para um nó que esteja também"perto" na rede subjacente:
  - Por exemplo, estimando o "round-trip time" para cada um desses atalhos.

Pastry

- Tabela de dispersão distribuída que tenta assegurar que existe uma correlação entre o encaminhamento na rede sobreposta e o encaminhamento na rede subjacente.

Propriedades

- Identificadores de 128 dígitos.
- Encaminha para o nó mais próximo de uma chave em $log_{2^b} N$ passos.
  - Por exemplo $b = 2$ ou $b = 4$
- O encaminhamento é assegurado a não ser que falhem $L/2$ nós com identificadores adjacentes falhem simultaneamente.

Estado em cada nó

- Uma tabela de encaminhamento.
- Um conjunto de vizinhos.
  - Os M nós mais perto na rede subjacente.
- Um conjunto de folhas.
  - Os L nós numericamente mais próximos

Estado no nó pastry

Identificador do no **10233102**

Folhas

Menor		Maior	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Tabela de encaminhamento

10233102	02212102	**10233102**	22301203	31203203
10233102	**1–0–233102**	1–1–301233	1–2–230203	1–3–021022
10233102	10–0–31203	10–1–32102	**10–2–33102**	10–3–23302
10233102	102–0–0230	102–1–1302	102–2–2302	**102–3–3102**
10233102	1023–0–322	1023–1–000	1023–2–121	**1023–3–102**
10233102	10233–0–01	**10233–1–02**	10233–2–32	
10233102	**102331–0–2**		102331–2–0	
10233102			**1023310–2**	

Vizinhos

13021022	10200230	11301233	10233102
02212102	22301203	31203203	33213321

Encaminhamento

- Se o nó de destino for uma "folha" de uma das do nó p, enviar para a folha mais próxima do nó de destino.
- Caso contrário usar a tabela de encaminhamento.

Entrada de nós

- Assuma-se que o nó X pretende entrar na rede.
- Escolhe um nó A que esteja próximo de si na rede subjacente.
- Pede a A para encaminhar uma mensagem de "entrada" para X.
  - Esta mensagem vai até ao nó Z que possui o identificador mais próximo de X e que já está na rede.
- Cada nó no caminho envia as suas tabelas para o novo nó X.

Entrada de nós

- A tabela de vizinhos do nó X é obtida através da A.
- A tabela de folhas do nó X é obtida através de Z.
- A tabela de encaminhamento é obtida através da linha relevante de cada nó por onde a mensagem de "entrada" passou.
  - A primeira linha vem de A.
  - A segunda linha, do segundo hop da mensagem de "entrada".
  - etc.

Entrada de nós

- Por fim, o nó X envia o seu estado a todos os nós cujos identificadores conhece:
  - Ou seja aos seus nós folha, vizinhos e membros da tabela de encaminhamento.
  - Estes actualizam o seu estado em função desta informação.

Gestão da concorrência

- Quando um nó C envia o seu estado para um novo nó X, envia uma estampilha temporal da última alteração.
- Quando o nó X envia de volta o seu estado para C,volta a enviar esta estampilha.
- Se C vê que o seu estado se alterou entretanto, re-inicia o processo de actualização do novo membro.

Proximidade

- Numa segunda fase, um nó obtém a tabela de encaminhamento e a tabela de vizinhos de cada nó.
  - Usa os nós que conhece deste modo para tentar melhorar a proximidade das suas tabelas de encaminhamento.

Aplicações

- Serviços de nomes escaláveis para sistemas peer-to-peer
  - Substituição do Naspter
- Sistemas de ficheiros distribuídos
- Sistemas de armazenamento eterno
- Sistemas de subscrição-publicação
  - Baseados em pontos de rendez-vous

Resumo

- DHTs
- Chord
- Pastry.
- Aplicações