

Sistemas Distribuídos

Aula de Hoje: Fofoca!

Prof. Elias P. Duarte Jr.
Universidade Federal do Paraná (UFPR)
Departamento de Informática
www.inf.ufpr.br/elias/sisdis



Sumário

- Revisando a difusão confiável: desempenho dos algoritmos
- *Gossip* ou Fofoca: difusão probabilística
- Definição
- Propriedades
- Algoritmo

Broadcasts: Tipos e Algoritmos

- Vimos que 4 tipos clássicos de broadcast (FIFO, Confiável, Causal e Atômico) além do Best-Effort
- Vimos 3 algoritmos de Difusão Confiável:
 - Difusão Confiável “básica”
 - Difusão Confiável com Detector de Falhas
 - Difusão Confiável Uniforme,
- Foco hoje é a Difusão Confiável

Difusão Confiável: Desempenho

- O algoritmo básico de difusão confiável utiliza N^2 mensagens (sempre)
- O algoritmo com Detector de Falhas utiliza N mensagens no melhor caso: sem-falhas
 - Falhas “devem” ser eventos “raros” ;-)
- Mas, no pior caso, o algoritmo com Detector de Falhas utiliza N^2 mensagens

Como escapar do número quadrático de mensagens da Difusão Confiável?

- Veja que em sistemas de grande porte (milhares de processos) a estratégia quadrática é inviável
- Uma estratégia para melhorar o desempenho é escapar para uma solução probabilística

Gossip ou Fofoca

- O algoritmo de *Gossip* permite justamente a difusão probabilística de informações
- Sinônimo: Difusão Epidêmica
- A intuição:
 - Um processo “conta a fofoca” ou “contamina” um subconjunto de seus vizinhos, escolhidos aleatoriamente
 - Cada um dos seus vizinhos repete o processo
 - Depois de um tempo, há uma alta probabilidade que todos os processos do sistema sabem a fofoca ou estão contaminados

Lembre-se: É Probabilístico!

- Isso significa que há uma probabilidade de um processo nunca receber – e portanto nunca entregar – uma mensagem
- A probabilidade pode ser pequena em uma execução, mas existe!
- Desta forma há situações em que não deve ser usado

Propriedades do Gossip

- Validade Probabilística: considere dois processos corretos de um sistema distribuído: x & z
- Existe uma probabilidade de que uma mensagem transmitida por x usando a primitiva *gossip(msg)* seja entregue por z
- Outras propriedades: não há mensagens espúrias, não há entrega duplicada de mensagens

O Algoritmo do Gossip

Algoritmo Difusão Probabilística: Gossip

Delivered: conjunto de mensagens entregues

MaxRetrans: parâmetro, número máximo de transmissões de 1 mensagem

fanout: parâmetro, número de vizinhos selecionados para receber mensagem

Init: Delivered $\leftarrow \Phi$;

Trans $\leftarrow 0$; // número de transmissões de uma mensagem msg

Upon gossip(msg): selecione um conjunto aleatório Fo de vizinhos

for all $p_i \in F_o$ do: send(msg) to p_i ;

Trans $\leftarrow 1$;

Upon receive(msg): if msg **not** in Delivered

then Delivered \leftarrow Delivered \cup {msg};

deliver(msg);

selecione um conjunto aleatório Fo de vizinhos

if Trans \leq MaxRetrans

then for all $p_i \in F_o$ do: send(msg) to p_i ;

Trans \leftarrow Trans + 1;

Na Difusão Probabilística

- O conjunto F_o de vizinhos selecionados aleatoriamente: tem *fanout* vizinhos
- Este número de vizinhos é parâmetro do algoritmo
- Bem como o número máximo de vezes que um processo pode retransmitir uma determinada mensagem
- Observe que quanto maior o *fanout*, menor pode ser o MaxRetrans
- Observe que o algoritmo emprega mensagens redundantes

Referências:

Uma boa referência para Gossip é:

P. Eugster, R. Guerraoui, A. Kermarrec, L. Massoulié:,
“Epidemic Information Dissemination in Distributed
Systems,” *Computer*, Vol. 37, No. 5, 2004.

O algoritmo é também descrito no livro texto.

Conclusão

- Iniciamos com uma revisão dos algoritmos de difusão confiável e seu desempenho
- Em seguida motivamos e definimos a difusão probabilística
 - Definimos o problema
 - Especificamos as propriedades
 - Vimos o algoritmo

Obrigado!
Página da Disciplina
Sistemas Distribuídos:
www.inf.ufpr.br/elias/sisdis