Sistemas Distribuídos em Larga Escala
Trabalho Prático

-

Mestrado em Engenharia Informática
Universidade do Minho

Grupo nº 1

PG41080	João Ribeiro Imperadeiro
PG41081	José Alberto Martins Boticas
PG41091	Nelson José Dias Teixeira

20 de maio de 2020

1 Introdução

O presente relatório descreve o desenvolvimento do projeto de cariz prático da unidade curricular de Sistemas Distribuídos em Larga Escala. Neste trabalho é requerida a implementação de um dos algoritmos de agregação distribuída disponíveis na referência [2] indicada no enunciado do mesmo. Após concretizar a especificação do algoritmo escolhido, é posteriormente solicitado o teste do mesmo no simulador desenvolvido ao longo do semestre do presente ano letivo.

Tal como é referido nos manuscritos [1] e [2], a agregação de dados distribuídos desempenha um papel bastante importante na concepção de diversos sistemas escaláveis uma vez que possibilita a determinação descentralizada de propriedades globais significativas, que posteriormente podem ser utilizadas para direcionar a execução de outras aplicações distribuídas. Vários algoritmos de agregação distribuída foram propostos ao longo dos últimos anos, exibindo propriedades diferentes em termos de precisão, desempenho e de comunicação. No entanto, muitas dessas abordagens não possuem caraterísticas relacionadas com a tolerância de falhas. Desta forma, no âmbito de sistemas distribuídos, este grupo de trabalho viu-se interessado em implementar um dos algoritmos associados a esta propriedade.

Relativamente à estrutura deste relatório, é exibido o algoritmo escolhido pelos elementos deste grupo, apresentando o conceito e a implementação intrínsecos ao mesmo. Para além disso, são eviden-

ciados alguns aspetos relativos ao simulador utilizado para proceder à realização de testes do algoritmo em causa, efetuando finalmente uma análise dos resultados obtidos.

2 Algoritmo

Dos algoritmos de agregação distribuída presentes no documento referenciado no enunciado deste trabalho prático [2], os elementos deste grupo optaram por escolher o algoritmo ***flow updating***. Este, ao nível de comunicação, é classificado como não estruturado, inserindo-se na categoria *gossip* que, por sua vez, diz respeito à forma como as mensagens são disseminadas pela rede de comunicação. Quanto à perspetiva computacional, este algoritmo é baseado no conceito de *averaging*, isto é, na computação iterativa de médias parciais que, ao longo do tempo, convergem para um resultado final previamente determinado. Esta última técnica permite também a derivação de outras funções de agregação (como por exemplo, *count* ou *sum*) de acordo com as combinações dos valores inicialmente instanciados.

Uma das razões que levou este grupo a escolher este algoritmo foi a capacidade do mesmo em tolerar a injeção de falhas. Esta última caraterística é bastante importante sobretudo no que diz respeito à perda de mensagens trocadas na rede de comunicação. Para além desta vantagem associada ao contexto de sistemas distribuídos, este algoritmo possui não só um melhor desempenho quando comparado com os outros da mesma classe, como também possibilita uma computação precisa de valores. Por fim, a execução do algoritmo em causa é independente da topologia do roteamento de rede.

No 4º capítulo deste documento é apresentado todos os resultados obtidos após a implementação do algoritmo escolhido, discutindo-se a veracidade das propriedades mencionadas acima.

2.1 Conceito

O algoritmo é baseado na manipulação de fluxos (no sentido teórico de grafos), que são atualizados usando mensagens idempotentes, fornecendo a ele recursos exclusivos de robustez.

2.2 Implementação

...

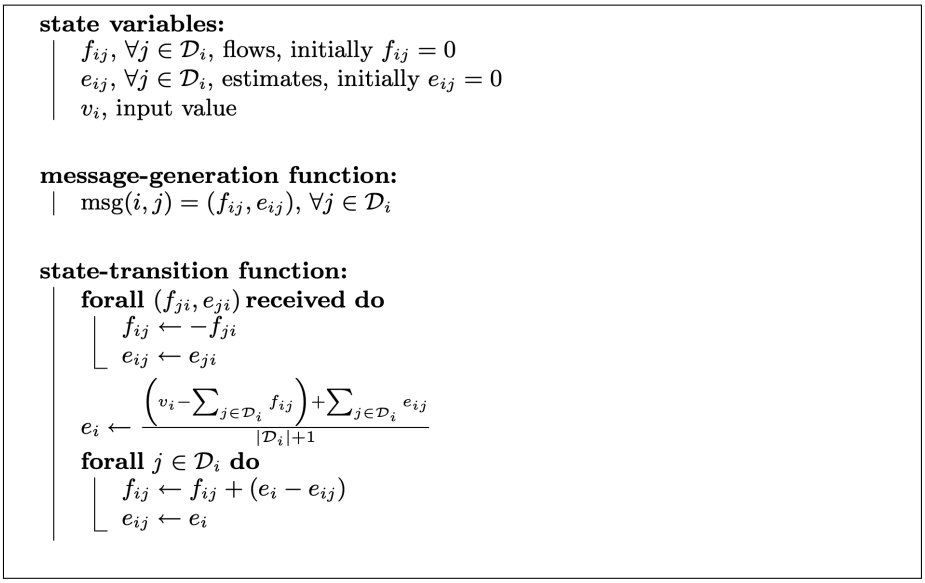

Figura 1: Pseudocódigo do algoritmo *Flow Updating*

3 Simulador

...

4 Análise de resultados obtidos

Os resultados da avaliação obtidos, quando comparados com outras abordagens de *averaging*, revelaram que os superam em termos de complexidade temporal (maior desempenho) e ao nível do número de mensagens trocadas (*overhead*).

5 Conclusão

Este algoritmo tolera a perda substancial de mensagens (*link failures*), enquanto outros algoritmos concorrentes da mesma categoria podem ser afetados por uma única mensagem perdida.

Referências

[1] Paulo Jesus, Carlos Baquero e Paulo Almeida. «Fault-Tolerant Aggregation by Flow Updating». Em: jun. de 2009, pp. 73–86. DOI: 10.1007/978-3-642-02164-0_6.

[2] Paulo Jesus, Carlos Baquero e Paulo Almeida. «A Survey of Distributed Data Aggregation Algorithms». Em: *Communications Surveys & Tutorials, IEEE* 17 (out. de 2011). DOI: 10.1109/COMST.2014.2354398. URL: https://arxiv.org/pdf/1110.0725.pdf.