# Avaliação

Este capítulo é dedicado à experimentação da implementação realizada do algoritmo Invitation (Garcia-Molina, 1982).

## Ambiente onde não há perda de mensagens

Nesta seção, foi descrito os testes do algoritmo Invitation, em um ambiente onde não há perda de mensagens. O intuito é entender melhor, qual o comportamento esperado do algoritmo em um ambiente “ideal”, apensar que como explanado em sala de aula, ambientes “ideais” são raros e falhas são inerentes a sistemas distribuídos.

### Convergência para diferentes números de nós

Realizamos aqui um teste para avaliar a quantidade de mensagens trocadas, e em quantos passos o algoritmo consegue sair de seu estado inicial (onde todos os nós são coordenadores apenas de si mesmos). A quantidade de nós testadas cresceu geometricamente iniciando-se em 2 até 512 nós.

#### Quantidade de mensagens trocadas

Durante os testes para avaliar a convergência, foi avaliada a quantidade de mensagens necessárias para a convergência do algoritmo, e a quantidade de passos necessários para alcançar o fim da eleição de líder. O teste mostrou os seguintes resultados:

Gráfico - Números de mensagens trocadas durante a fase de eleição inicial

A quantidade de mensagens trocadas para esta implementação e exibidas no Gráfico 1, foram extraídas da execução da ferramenta. Através dela, do algoritmo e da implementação, pode-se calcular uma função que prevê a quantidade exata de mensagens trocadas para cada fase do algoritmo, onde N é a quantidade de nós em que todos podem se comunicar com todos (grafo completo):

|  |  |  |  |
| --- | --- | --- | --- |
| Fase | Tipo de Mensagem | Quantidade de mensagens | Descrição |
| Check Members | AYCoord | O(N²-N) | Considerando o estado inicial como pior caso, onde todos os nós são coordenadores de si mesmos e não coordenam nenhum outro nó. |
| Check Members | AYC\_answer | O(N²-N) | Também considerando o pior caso, todas as mensagens AYCoord serão respondidas com uma AYC\_answer. |
| Merge | Invitation | O(N-1) | O coordenador que tiver maior prioridade, irá iniciar o convite para “*merge*” e irá enviar um convite para todos os outros coordenadores, que no pior caso, será todos os outros nós. |
| Merge | Accept | O(N-1) | Para cada mensagem Invitation, será enviada uma mensagem de Accept para o coordenador que iniciou o *merge*. |
| Merge | Accept\_answer | O(N-1) | Para cada mensagem Accept, o coordenador que iniciou o merge irá responder um *Accept\_answer.* |
| Reorganize | Ready | O(N-1) | Para iniciar a fase de Reorganize, o coordenador candidato irá enviar uma mensagem de *Ready* para todos os demais que irá coordenar. |
| Reorganize | Ready\_answer | O(N-1) | Todos os nós que receberem a mensagem de Ready irão responder com uma mensagem de Ready\_answer |

Tabela - Cálculo de mensagens trocadas durante a primeira eleição do algoritmo Invitation

Dessa forma, durante a primeira eleição, considerando o pior caso, serão trocadas:

(N²-N) + (N²-N) + (N-1) + (N-1) + (N-1) + (N-1) + (N-1) = **2N² + 3N – 5**

Exemplo (4 nós): 2(4)² + 3(4) – 5 = 39 mensagens, que corresponde às mensagens trocadas pelo teste com 4 nós (Gráfico 1) pela implementação que acompanha este trabalho. Este cálculo nos mostra, que durante a eleição inicial, considerando que todos os nós estão interligados com todos os outros nós, a quantidade de mensagens trocadas possui complexidade exponencial de O(N²) para este algoritmo.

#### Quantidade de passos necessários para a eleição

Como não há perda de mensagens, todos os testes tiveram a mesma quantidade de passos:

Todas as quantidades de nós testadas necessitaram de 66 passos para realizar a eleição inicial.

Segundo o algoritmo de Garcia-Molina, ao ser instanciado cada nó é o coordenador de si mesmo. Também é definido o passo onde cada coordenador verifica se há outros coordenadores acessíveis (*Check\_members*()). Essa checagem é realizada através do envio de uma mensagem AYCoord para todos os nós alcançáveis. Essa checagem foi implementada para acontecer a cada 50 passos, logo, no passo 50 todos os coordenadores enviam uma mensagem AYCoord (como podemos ver na linha 1 e 2 do log da Tabela 1). Seguindo isso, ocorrem as trocas de mensagens para a eleição do coordenador entre os passos 51 até 66 (Tabela 1). Logo, como não há perda de mensagens, e todos podem enviar mensagens dentro de um mesmo passo, independente da quantidade de nós, a quantidade de passos para a eleição inicial será de 66 passos para esta implementação. Vale ressaltar, que este número está diretamente ligado ao intervalo de checagem (*Check\_members*()) de outros coordenadores.

|  |  |
| --- | --- |
|  |  |
| 1 | STEP 51- Node 1: Received MSG AYCoord by 2 |
| 2 | STEP 51- Node 2: Received MSG AYCoord by 1 |
| 3 | STEP 52- Node 1: Received MSG AYC\_answer by 2 said your coord is Node(ID=2) |
| 4 | STEP 52- Node 1: Setup timerToMerge = STEP 72 |
| 5 | STEP 52- Node 2: Received MSG AYC\_answer by 1 said your is Node(ID=1) |
| 6 | STEP 52- Node 2: SetUP timerToMerge 62 |
| 7 | STEP 62- Node 2: Start merge |
| 8 | STEP 63- Node 1: Received MSG Invitation by 2 |
| 9 | STEP 64- Node 2: Received MSG Accept by 1 |
| 10 | STEP 64- Node 2: Start Reorganizing |
| 11 | STEP 65- Node 1: Received MSG Accept\_answer by 2 |
| 12 | STEP 65- Node 1: Received MSG Ready by 2 |
| 13 | STEP 66- Node 2: Received MSG Ready\_answer by 1 |

Tabela - Log dos primeiros passos do algoritmo Invitation (Garcia-Molina)

Após a eleição inicial, como não há perda de mensagens neste teste nem nenhum outro distúrbio nos nós, ele ficará indefinidamente no estado ótimo (um coordenador ativo e aceito por todos os outros nós ativos).