Instituto Federal de Educação, Ciência e Tecnologia da Bahia

Campus: Santo Antônio de Jesus **Data**: 06/08/2025

Curso: Analise e Desenvolvimento de Sistemas

Disciplina: Sistemas Distribuídos **Docente**: Felipe Silva

Discentes: Anderson Guilherme Santana Moraes e Carlos de Jesus Conceição Junior

Relatório Técnico: Simulação e Análise de Algoritmos de Eleição em Sistemas Distribuídos: Bully e Anel

Santo Antônio de Jesus
06/08/2025

Introdução

Este relatório detalha a implementação e análise de um sistema distribuído simulado, projetado para comparar o comportamento de dois algoritmos de eleição clássicos: **Bully** e **Anel**. A aplicação, desenvolvida em Java com comunicação via RMI, modela um ambiente com cinco processos autônomos que precisam eleger um novo líder (coordenador) em caso de falhas.

O objetivo principal é observar e analisar como cada algoritmo lida com a ausência do coordenador em dois cenários distintos: uma falha simples com recuperação e uma falha múltipla. Através da análise de logs, este documento compara a eficiência, a robustez e as características de cada abordagem, determinando suas vantagens, desvantagens e casos de uso ideais.

Explicação Conceitual dos Algoritmos

Algoritmos de eleição são fundamentais em sistemas distribuídos para garantir que, na ausência de um processo coordenador, os processos restantes possam concordar sobre um novo líder de forma autônoma.

Algoritmo Bully

O Algoritmo Bully baseia-se na premissa de que o processo com o maior identificador (ID) entre os ativos deve ser o líder. Seu funcionamento é competitivo e direto:

1. Quando um processo *P* deteta a falha do coordenador, ele inicia uma eleição enviando uma mensagem *ELEICAO* para todos os processos com ID maior que o seu.
2. Se um processo com ID maior recebe a mensagem *ELEICAO*, ele responde com uma mensagem *OK* (essencialmente dizendo "eu assumo a partir daqui") e inicia sua própria eleição. O processo *P* que recebeu o *OK* sabe que não será o líder e apenas aguarda.
3. Se um processo que iniciou uma eleição não recebe nenhuma resposta *OK* dentro de um tempo limite, ele se considera o processo ativo com o maior ID, vence a eleição e anuncia a todos que é o novo *COORDENADOR*.

Algoritmo de Anel

O Algoritmo de Anel opera de forma cooperativa, assumindo que os processos formam um anel lógico.

1. Quando um processo *P* deteta a falha do coordenador, ele cria uma mensagem *ELEICAO* contendo uma lista com o seu próprio ID e a envia para seu sucessor no anel.

2. Cada processo que recebe a mensagem *ELEICAO* adiciona seu próprio ID à lista e a repassa para seu sucessor.
3. A mensagem circula pelo anel até retornar ao processo *P* que a iniciou. Neste momento, a lista contém os IDs de todos os processos ativos que participaram.
4. O processo *P* então elege o processo com o maior ID da lista como o novo coordenador e circula uma nova mensagem *COORDENADOR* para informar a todos sobre o resultado.

Análise das Simulações e Logs de Testes

Foram executados dois cenários de falha para cada algoritmo. A análise a seguir baseia-se nos logs gerados, que demonstram o comportamento de cada algoritmo em detalhe.

Cenário A: Falha e Recuperação do Coordenador

Neste cenário, o coordenador inicial (P5) falha, um novo líder (P4) é eleito, e em seguida, P5 se recupera e retoma a liderança.

Execução com Algoritmo Bully - Cenário A

A natureza competitiva do Bully é claramente visível. Múltiplos processos detectam a falha e iniciam eleições concorrentes, gerando um alto volume de mensagens.

LOG - Início da Eleição Concorrente:

```
>>> AÇÃO: Coordenador P5 irá falhar agora. <<<
[P5] Simulação de FALHA.
[Simulador] Processo P5 DESREGISTRADO do RMI.
[P1] Pingando coordenador P5...
[P1] Falha ao contatar P5. Considerado como FALHO.
[P1] Iniciou uma ELEIÇÃO (Bully).
[P2] Recebeu: Msg[Tipo=ELEICAO, Remetente=1, PIDs=[]]
[P1] Recebeu: Msg[Tipo=OK, Remetente=2, PIDs=[]]
[P2] Iniciou uma ELEIÇÃO (Bully).
[P3] Recebeu: Msg[Tipo=ELEICAO, Remetente=2, PIDs=[]]
[P2] Recebeu: Msg[Tipo=OK, Remetente=3, PIDs=[]]
[P3] Iniciou uma ELEIÇÃO (Bully).
[P4] Recebeu: Msg[Tipo=ELEICAO, Remetente=3, PIDs=[]]
[P3] Recebeu: Msg[Tipo=OK, Remetente=4, PIDs=[]]
[P4] Iniciou uma ELEIÇÃO (Bully).
```

Análise: O log mostra uma "cascata de intimidação". P1 inicia, mas é silenciado por P2, que por sua vez inicia e é silenciado por P3. Este caos converge para o resultado correto.

LOG - Eleição e Recuperação:

```
[P4] Se declarou o novo COORDENADOR (Bully).
[P1] Recebeu: Msg[Tipo=COORDENADOR, Remetente=4, PIDs=[]]
[P1] Novo coordenador definido: P4
[P2] Recebeu: Msg[Tipo=COORDENADOR, Remetente=4, PIDs=[]]
[P2] Novo coordenador definido: P4
[P3] Eleição Bully encerrada. Aguardando anúncio.
[P3] Recebeu: Msg[Tipo=COORDENADOR, Remetente=4, PIDs=[]]
[P3] Novo coordenador definido: P4
[P4] Falha ao contatar P5. Considerado como FALHO.
[P2] Eleição Bully encerrada. Aguardando anúncio.
[P1] Eleição Bully encerrada. Aguardando anúncio.
[P1] Pingando coordenador P4...
[P4] Recebeu: Msg[Tipo=PING, Remetente=1, PIDs=[]]

>>> AÇÃO: Antigo coordenador P5 irá se recuperar. <<<
[P5] Simulação de RECUPERAÇÃO.
[P5] Iniciou uma ELEIÇÃO (Bully).
[P5] Se declarou o novo COORDENADOR (Bully).
[P1] Recebeu: Msg[Tipo=COORDENADOR, Remetente=5, PIDs=[]]
[P1] Novo coordenador definido: P5
[P2] Recebeu: Msg[Tipo=COORDENADOR, Remetente=5, PIDs=[]]
[P2] Novo coordenador definido: P5
[P3] Recebeu: Msg[Tipo=COORDENADOR, Remetente=5, PIDs=[]]
[P3] Novo coordenador definido: P5
[P4] Recebeu: Msg[Tipo=COORDENADOR, Remetente=5, PIDs=[]]
[P4] Novo coordenador definido: P5
```

Análise: P4, sendo o processo ativo de maior ID, vence a eleição. Quando P5 retorna, sua primeira ação é iniciar uma nova eleição. Tendo o maior ID de todos, ele vence imediatamente e retoma a liderança.

Execução com Algoritmo de Anel - Cenário A

O comportamento do Anel é mais ordenado e previsível, com um número fixo de mensagens.

LOG - Circulação da Mensagem de Eleição:

```
>>> AÇÃO: Coordenador P5 irá falhar agora. <<<
[P5] Simulação de FALHA.
[Simulador] Processo P5 DESREGISTRADO do RMI.
[P1] Pingando coordenador P5...
[P1] Falha ao contatar P5. Considerado como FALHO.
[P1] Iniciou uma ELEIÇÃO (Anel).
[P2] Recebeu: Msg[Tipo=ELEICAO, Remetente=1, PIDs=[1]]
[P2] Repassando mensagem de eleição. PIDs na lista: [1, 2]
[P3] Recebeu: Msg[Tipo=ELEICAO, Remetente=1, PIDs=[1, 2]]
[P3] Repassando mensagem de eleição. PIDs na lista: [1, 2, 3]
[P4] Recebeu: Msg[Tipo=ELEICAO, Remetente=1, PIDs=[1, 2, 3]]
[P4] Repassando mensagem de eleição. PIDs na lista: [1, 2, 3, 4]
```

Análise: Ao contrário do Bully, apenas um processo (P1) inicia a eleição. A mensagem de eleição circula de forma cooperativa, com cada processo a adicionar o seu ID à lista.

LOG - Conclusão e Recuperação:

```
[P3] Eleição em Anel concluída. Novo coordenador será P4
[P4] Recebeu: Msg[Tipo=COORDENADOR, Remetente=4, PIDs=[]]
[P4] Novo coordenador definido: P4
[P1] Pingando coordenador P4...
[P4] Recebeu: Msg[Tipo=PING, Remetente=1, PIDs=[]]
[P2] Pingando coordenador P4...
[P4] Recebeu: Msg[Tipo=PING, Remetente=2, PIDs=[]]

>>> AÇÃO: Antigo coordenador P5 irá se recuperar. <<<
[P5] Simulação de RECUPERAÇÃO.
[P5] Iniciou uma ELEIÇÃO (Anel).
[P1] Recebeu: Msg[Tipo=ELEICAO, Remetente=5, PIDs=[5]]
[P1] Repassando mensagem de eleição. PIDs na lista: [5, 1]
[P2] Recebeu: Msg[Tipo=ELEICAO, Remetente=5, PIDs=[5, 1]]
[P2] Repassando mensagem de eleição. PIDs na lista: [5, 1, 2]
[P3] Recebeu: Msg[Tipo=ELEICAO, Remetente=5, PIDs=[5, 1, 2]]
[P3] Repassando mensagem de eleição. PIDs na lista: [5, 1, 2, 3]
[P4] Recebeu: Msg[Tipo=ELEICAO, Remetente=5, PIDs=[5, 1, 2, 3]]
[P4] Repassando mensagem de eleição. PIDs na lista: [5, 1, 2, 3, 4]
[P5] Recebeu: Msg[Tipo=ELEICAO, Remetente=5, PIDs=[5, 1, 2, 3, 4]]
[P5] Eleição em Anel concluída. Novo coordenador será P5
[P1] Recebeu: Msg[Tipo=COORDENADOR, Remetente=5, PIDs=[]]
[P1] Novo coordenador definido: P5
```

Análise: A mensagem retorna ao iniciador (P1), que elege o maior ID da lista *[4, 1, 2, 3].* O resultado é anunciado. Quando P5 se recupera, ele inicia uma nova eleição em anel, que circula por todos os 5 processos, e P5 é corretamente eleito.

Cenário B: Falha de Múltiplos Processos

Neste cenário, os processos P5 e P4 falham, testando a robustez dos algoritmos em eleger um líder entre os processos de menor ID.

Execução com Algoritmo Bully - Cenário B

LOG - Eleição com Múltiplas Falhas:

```
>>> AÇÃO: Coordenador P5 irá falhar. <<<
[P5] Simulação de FALHA.
[Simulador] Processo P5 DESREGISTRADO do RMI.

>>> AÇÃO: Processo P4 também irá falhar. <<<
[P4] Simulação de FALHA.
[Simulador] Processo P4 DESREGISTRADO do RMI.

>>> Aguardando eleição entre os processos restantes (P1, P2, P3)... <<<
[P1] Pingando coordenador P5...
[P1] Falha ao contatar P5. Considerado como FALHO.
[P1] Iniciou uma ELEIÇÃO (Bully).
[P2] Recebeu: Msg[Tipo=ELEICAO, Remetente=1, PIDs=[]]
[P1] Recebeu: Msg[Tipo=OK, Remetente=2, PIDs=[]]
[P2] Iniciou uma ELEIÇÃO (Bully).
[P3] Recebeu: Msg[Tipo=ELEICAO, Remetente=2, PIDs=[]]
[P2] Recebeu: Msg[Tipo=OK, Remetente=3, PIDs=[]]
[P3] Iniciou uma ELEIÇÃO (Bully).
[P3] Falha ao contatar P4. Considerado como FALHO.
[P3] Falha ao contatar P5. Considerado como FALHO.
[P3] Já está em um processo de eleição (Bully).
[P2] Falha ao contatar P4. Considerado como FALHO.
[P2] Falha ao contatar P5. Considerado como FALHO.
```

[

Análise: P1 inicia a eleição e tenta contatar os processos com IDs maiores (P4 e P5). Como ambos falharam, P2 não recebe nenhuma resposta *OK*.

LOG - Convergência para P3:

```
[P3] Se declarou o novo COORDENADOR (Bully).
[P1] Recebeu: Msg[Tipo=COORDENADOR, Remetente=3, PIDs=[]]
[P1] Novo coordenador definido: P3
[P2] Recebeu: Msg[Tipo=COORDENADOR, Remetente=3, PIDs=[]]
[P2] Novo coordenador definido: P3
```

Análise: Após o seu tempo de espera, e sem receber nenhuma resposta, P3 conclui corretamente que é o processo ativo com o maior ID e se declara o novo coordenador. O sistema estabiliza com P3 como líder, provando a sua eficácia.

Execução com Algoritmo de Anel - Cenário B

LOG - Anel se Reconfigura Automaticamente:

```
>>> AÇÃO: Coordenador P5 irá falhar. <<<
[P5] Simulação de FALHA.
[Simulador] Processo P5 DESREGISTRADO do RMI.

>>> AÇÃO: Processo P4 também irá falhar. <<<
[P4] Simulação de FALHA.
[Simulador] Processo P4 DESREGISTRADO do RMI.

>>> Aguardando eleição entre os processos restantes (P1, P2, P3)... <<<
[P1] Pingando coordenador P5...
[P1] Falha ao contatar P5. Considerado como FALHO.
[P1] Iniciou uma ELEIÇÃO (Anel).
[P2] Recebeu: Msg[Tipo=ELEICAO, Remetente=1, PIDs=[1]]
[P2] Repassando mensagem de eleição. PIDs na lista: [1, 2]
[P3] Recebeu: Msg[Tipo=ELEICAO, Remetente=1, PIDs=[1, 2]]
[P3] Repassando mensagem de eleição. PIDs na lista: [1, 2, 3]
```

Análise: O processo getProximoNoAnel() deteta que P4 e P5 estão inativos e salta diretamente para o próximo processo vivo. O log mostra a mensagem a passar de P3 diretamente de volta para P1, reconfigurando o anel para *P1 -> P2 -> P3 -> P1*.

LOG - Eleição do Novo Líder (P3):

```
[P1] Recebeu: Msg[Tipo=ELEICAO, Remetente=1, PIDs=[1, 2, 3]]
[P1] Eleição em Anel concluída. Novo coordenador será P3
[P2] Recebeu: Msg[Tipo=COORDENADOR, Remetente=3, PIDs=[]]
[P2] Novo coordenador definido: P3
```

Análise: Ao receber a mensagem de volta, P1 analisa a lista de participantes [1, 2, 3] e elege 3 como o maior ID. A mensagem de COORDENADOR é circulada, e o sistema estabiliza com P3 como o novo líder, demonstrando a resiliência do algoritmo.

Análise Comparativa

Característica	Algoritmo Bully	Algoritmo de Anel
Nº de Mensagens	**Pior caso:** O(n^2) - Muitas eleições concorrentes podem gerar um alto tráfego de rede, como visto nos logs.	**Fixo:** 2(n-k) - Onde k é o número de falhas. Mais previsível e menos sobrecarregado.
Tempo de Eleição	**Rápido.** A eleição pode terminar em duas trocas de mensagens se um processo de baixo ID iniciar e o de maior ID estiver ativo.	**Lento.** A mensagem precisa dar a volta completa no anel, o que pode levar mais tempo.
Robustez	Robusto. Lida bem com múltiplas falhas e recuperações.	Robusto, mas requer que o anel lógico possa ser reconstruído.

Vantagens e Desvantagens Observadas

A simulação em ambos os cenários permitiu observar na prática as seguintes características de cada algoritmo:

Algoritmo Bully

- **Vantagem:** A sua principal vantagem é a velocidade de convergência. Como visto no Cenário A, um processo que se recupera (P5) consegue iniciar uma

eleição e se declarar coordenador quase imediatamente, sem precisar de esperar por confirmações de outros processos.

- **Desvantagem:** A desvantagem mais clara, observada nos logs, é a alta sobrecarga de mensagens. Em ambos os cenários, a falha do coordenador levou a múltiplas eleições concorrentes, gerando um tráfego de rede "caótico" e ineficiente, com muitas mensagens *ELEICAO* e *OK*.

Algoritmo de Anel

- **Vantagem:** A sua maior força é a eficiência e previsibilidade da rede. Os logs mostraram um fluxo de mensagens muito mais ordenado, com um número fixo de mensagens a circular pelo anel. Isto evita o risco de congestionamento que o Bully poderia causar.

- **Desvantagem:** O algoritmo é inerentemente mais lento. A eleição só termina depois de a mensagem de eleição dar uma volta completa no anel, o que, dependendo do número de processos e da latência, pode levar consideravelmente mais tempo do que o Bully para eleger um novo líder.

Casos de Uso Adequados

Com base nas vantagens e desvantagens observadas, a escolha entre os algoritmos depende diretamente dos requisitos do sistema:

- **Algoritmo Bully:** É mais adequado para sistemas de pequena ou média escala, onde a velocidade de recuperação é a prioridade máxima e a rede pode tolerar picos de mensagens. A sua simplicidade e rapidez tornam-no ideal para ambientes onde um líder precisa de ser estabelecido o mais rápido possível.

- **Algoritmo de Anel:** É ideal para sistemas de grande escala, onde a eficiência e a economia de recursos de rede são cruciais. O seu comportamento previsível e o baixo número de mensagens garantem que o processo de eleição não irá sobrecarregar o sistema, mesmo com um grande número de nós.

Conclusão

A implementação e simulação dos algoritmos Bully e Anel permitiram uma análise prática de duas abordagens distintas para o problema de eleição de líder. A simulação validou que ambos os algoritmos são robustos e capazes de eleger um novo coordenador em cenários de falha. O Algoritmo Bully demonstrou ser mais rápido, porém com um custo de mensagens potencialmente alto, enquanto o Algoritmo de Anel se mostrou mais económico em termos de mensagens, mas com uma latência maior. A escolha entre os dois, portanto, depende diretamente dos requisitos específicos do sistema.

Santo António de Jesus
06/08/2025