Sistemas Distribuídos

Algoritmos de Eleição

8

Universidade Estácio de Sá
Professor Welsing M. Pereira
www.professorwelsing.webnode.com

Algoritmos de Eleição

- Introdução
  - Um algoritmo para escolher um único processo para desempenhar uma função em particular é chamado de algoritmo de ***eleição***.
  - Exemplo:
    - Em uma variante do algoritmo do servidor central para exclusão mútua, o servidor é escolhido dentre os processos p_i, i = 1, 2, ..., N que precisam usar a seção crítica. Um algoritmo de eleição é necessário para escolher qual dos processos desempenhará a função de servidor.
  - Dizemos que um processo convoca a eleição se ele faz uma ação que inicia uma execução em particular do algoritmo de eleição.
  - Um processo individual não convoca mais do que uma eleição por vez, mas, em princípio, N processos poderiam convocar N eleições concorrentes.

Algoritmos de Eleição

- Introdução
  - A qualquer momento, um processo p_i, é participante ou não participante significando que correntemente ele está ou não está envolvido com uma eleição.
  - A escolha do processo eleito deve ser única entre todos os participantes.
  - O processo com o maior identificador deve ser o eleito.
  - O identificador deve ser qualquer valor útil, desde que os identificadores sejam exclusivos e totalmente ordenados.
  - Cada processo p_i (i=1,2,...,N) tem uma variável *elected*$_i$, que conterá o identificador do processo eleito.
  - Quando o processo se torna participante de uma eleição pela primeira vez, ele configura essa variável com o valor especial '⊥', para denotar que ela ainda não está definida.

Algoritmos de Eleição

- Um algoritmo de eleição baseado em anel

  Algoritmo de Chang e Roberts
  - Nesse algoritmo, cada processo p_i tem um canal de comunicação para o processo seguinte no anel, $p_{(i+1) \bmod N}$, e todas as mensagens são enviadas no sentido horário em torno do anel.
  - Suponhamos que não ocorram falhas e que o sistema é assíncrono.
  - O objetivo desse algoritmo é eleger um único processo, chamado de coordenador, que é aquele com o maior identificador.

Algoritmos de Eleição

- Um algoritmo de eleição baseado em anel

  Algoritmo de Chang e Roberts
  - Inicialmente, cada processo é marcado como não participante de uma eleição.
  - Qualquer processo pode iniciar a eleição.
  - Ele prossegue marcando a si mesmo como participante, colocando seu identificador em uma *mensagem de eleição* e enviando-a para seu vizinho no sentido horário.

Algoritmos de Eleição

- Um algoritmo de eleição baseado em anel
  - Quando um processo recebe uma mensagem de eleição, ele compara o identificador presente na mensagem com o seu próprio.
  - Se o identificador que chegou é maior, então ele encaminha a mensagem para seu vizinho.
  - Se o identificador que chegou é menor e o receptor não é participante, então ele substitui por seu próprio identificador na mensagem e a encaminha.
  - Mas, se o identificador que chegou é menor e o receptor é participante, a mensagem não é encaminhada.
  - Em qualquer caso, no encaminhamento de uma mensagem de eleição, o processo marca a si mesmo como participante.

Algoritmos de Eleição

- Um algoritmo de eleição baseado em anel

  Algoritmo de Chang e Roberts
  - Entretanto, se o identificador recebido for o do próprio receptor, então o identificador desse processo deve ser o maior e se tornará o coordenador.
  - Mais uma vez, o coordenador marca a si mesmo, agora, como não participante e envia uma ***mensagem elected*** para seu vizinho, anunciando sua eleição e incluindo sua identidade.

Algoritmos de Eleição

- Um algoritmo de eleição baseado em anel

  Algoritmo de Chang e Roberts
  - Quando um processo pi recebe uma mensagem elected, ele marca a si mesmo como não participante, configura sua variável $elected_i$, com o identificador presente na mensagem e, a não ser que seja o novo coordenador, encaminha a mensagem para seu vizinho.

Algoritmos de Eleição

- Um algoritmo de eleição baseado em anel

  Algoritmo de Chang e Roberts

  – Embora o algoritmo baseado em anel seja útil para se entender as propriedades dos algoritmos de eleição em geral, o fato de ele não tolerar falhas o torna limitado quanto ao seu valor prático.

  – Entretanto, com um detector de falha confiável é possível, em princípio, reconstituir o anel quando um processo falha.

Algoritmos de Eleição

- O algoritmo valentão (bully)
  – Esse algoritmo permite que os processos falhem durante uma eleição, embora presuma que a distribuição de mensagens entre os processos seja confiável.
  – Este algoritmo presume que o sistema é síncrono: ele usa tempos limites para detectar uma falha de processo.
  – Além disso, cada processo sabe quais processos têm identificadores mais altos e que pode se comunicar com todos esses processos.

Algoritmos de Eleição

- O algoritmo valentão (bully)
  – Existem três tipos de mensagens nesse algoritmo:
    - eleição, resposta e coordenador.
  – Uma mensagem de eleição é enviada para anunciar uma eleição.
  – Uma mensagem de resposta é enviada em resposta a uma mensagem de eleição.
  – Uma mensagem de coordenador é enviada para anunciar a identidade do processo eleito.
  – Um processo inicia uma eleição quando observa, por meio dos tempos limites, que o coordenador falhou. Vários processos podem descobrir isso simultaneamente.

Algoritmos de Eleição

- O algoritmo valentão (bully)
  – Como o sistema é síncrono, podemos construir um detector de falha confiável.
  – Há um atraso de transmissão de mensagem máximo de T_{trans} e um atraso máximo $T_{process}$ para processar uma mensagem.
  – Portanto, podemos calcular um tempo $T = 2.T_{trans} + T_{process}$, que é um limite superior para o tempo total decorrido desde o envio de uma mensagem até o processo receber uma resposta.
  – Se nenhuma resposta chegar dentro do tempo T, o detector de falha local poderá relatar que o destinatário do pedido pretendido falhou.

Algoritmos de Eleição

- O algoritmo valentão (bully)
  - O processo que sabe que possui o identificador mais alto pode eleger a si mesmo como coordenador simplesmente enviando uma mensagem de coordenador para todos os processos com identificadores mais baixos.
  - Por outro lado, uma processo com um identificador mais baixo inicia uma eleição enviando uma mensagem de eleição para os processos que têm identificador mais alto e esperando uma mensagem de resposta em retorno.
  - Se nenhuma resposta chegar dentro de um tempo T, o processo se considerará o coordenador (ele entenderá que os processos com identificadores mais altos falharam) e enviará uma mensagem de coordenador para todos os processos com identificadores mais baixos, anunciando isso.
  - Caso contrário, o processo esperará por mais um período T', que uma mensagem de coordenador chegue do novo coordenador.
  - Se nenhuma resposta chegar, ele iniciará outra eleição.

Algoritmos de Eleição

- O algoritmo valentão (bully)
  - Se um processo recebe uma mensagem de coordenador, ele configura a variável $elected_i$ com o identificador do coordenador contido dentro dela e trata esse processo como coordenador.
  - Se um processo recebe uma mensagem de eleição, ele envia de volta uma mensagem de resposta e inicia outra eleição – a não ser que já tenha iniciado uma.

Algoritmos de Eleição

- O algoritmo valentão (bully)
  - Quando é iniciado um processo para substituir um processo falho, ele inicia uma eleição.
  - Se tiver o identificador de processo mais alto, decidirá que é o coordenador e anunciará isso para os outros processos.
  - Assim, ele se tornará o coordenador, mesmo que o coordenador corrente esteja funcionando.
  - É por isso que o algoritmo é chamado de "valentão".