Aluno: José Norberto Fagundes Isaias - 19202785
DEC7558 - Sistemas Distribuídos

ALGORITMO DE ACORDO NO DESTINO

O algoritmo implementado foi baseado no Algoritmo de Paxos, que é um algoritmo clássico de consenso. A linguagem de programação utilizada foi o Python devido a sua simplicidade na compilação, execução do programa e do uso de bibliotecas. Neste caso, o Algoritmo de Paxos clássico foi modificado para aceitar múltiplos proponentes e para aceitar as propostas que possuam valor que teve mais propostas.

Para o funcionamento do algoritmo, foram implementadas duas classes, o Aceitador (*Acceptor*) e o Proponente ou Propositor (*Proposer*). O Aceitador tem um dicionário para as promessas e um contador de votos por *slot.* Além disso, ele tem um método para preparar as propostas (*receive_prepare*), que recebe o *slot* para qual a proposta será feita e o identificador da proposta. Esta função verifica se já há um valor prometido para um *slot* ou se o id da proposta é maior que o id da promessa. O *slot* é um espaço na fila de mensagens. Caso a primeira condição seja falsa ou a segunda seja verdadeira, o identificador da promessa é substituído pelo identificador da proposta. O Aceitador também tem a função *receive_accept_request.* Este método recebe o *slot* para qual a proposta será feita, o identificador da proposta e o valor proposto. Esta função verifica se já há um valor prometido para um *slot* ou se o id da proposta é maior ou igual que o id da promessa. Caso a primeira condição seja falsa ou a segunda seja verdadeira, o identificador da promessa é substituído pelo identificador da proposta. Além disso, o contador de votos por *slot* e por valor da mensagem é incrementado em um. O Aceitador ainda tem um terceiro método que retorna o consenso do Aceitador. Este método recebe o *slot* verificado e o tamanho do quórum. Verifica-se então a contagem de votos por valor proposto por *slot*. Se a contagem é maior ou igual ao tamanho do quórum, a função retorna o valor correspondente do *slot.*

A segunda classe, a do Propositor, possui um identificador próprio, um identificador para a sua proposta, o tamanho do quorum e um vetor com os Aceitadores. O Propositor possui apenas um método, que é o responsável por fazer a proposta (*propose*), o qual recebe o *slot* para qual será feita a proposta e o valor proposto. Neste método, há um contador de promessas aceitas. Primeiro, para cada Aceitador do vetor passado para o Propositor, o Propositor faz a proposta para o Aceitador usando a sua função de preparar propostas. Caso a proposta passe na Fase de Preparação, o contador de promessas é incrementado em um. Depois de passar por todos os Aceitadores, o método verifica se o número de promessas é menor que o quórum. Caso esta condição for verdadeira, o método retorna falso e uma mensagem de que houve falha em atingir o quórum na Fase de Preparação. Depois disso, verifica-se se a proposta foi finalmente aceita por cada um dos Aceitadores da lista usando a função *receive_accept_request* destes. Caso tenha

sido aceita a proposta, a contagem de “votos” para o valor proposto é incrementada. Por fim, se a contagem de votos for maior ou igual ao tamanho do quórum, a função retorna verdadeiro, caso contrário, retorna falso e informa da falha ao atingir o quórum na Fase de Aceitação.

A última parte é a função principal do código. Nesta parte, é definido a quantidade de aceitadores, o tamanho do quórum e os Aceitadores são criados. Também é definido uma função que apenas cria Propositores e chama a sua função de fazer a proposta. Listas com mensagens diferentes são criadas. E então, *threads* são criadas e iniciadas para simular múltiplos Propositores tentando fazer propostas para os Aceitadores, através da função citada anteriormente. Por fim, o consenso de todos os Aceitadores é verificado para cada *slot*.

Para rodar o algoritmo desenvolvido, basta ter o Python instalado no computador. O código usa apenas bibliotecas padrão do Python, e portanto não é necessário instalar nenhuma dependência externa. Para rodar a aplicação então, deve-se usar o executável do Python, ou apenas o comando “python3” ou “python” seguido do endereço, absoluto ou relativo, do arquivo Python do código disponível no Github (https://github.com/josenorbs22/acordo_no_destino_sd.git)