

Document Classification

CPD

Computação Paralela e Distribuída

Serial+OpenMP - 2012-13

Tue 11h

Grupo1

1. Introdução

O objectivo deste projecto é dado um conjunto de D documentos, cada documento classificado de acordo com S temas e com um número C de gabinetes, atribuir os documentos aos gabinetes baseando-se na relevância dos temas.

2. Descrição do algoritmo e abordagem utilizada para paralelização

Seguindo a metodologia de Foster decidimos partilhar os dados do problema pelos vários processos, isto é, dividirmos o total de documentos pelos processos, esta partição é possível pois as operações de cálculo de distâncias e de mudança de gabinete do documento podem ser realizadas independentes dos outros documentos.

O cálculo da soma dos temas de um gabinete e do total de documentos pertencente a esse gabinete também pode ser realizada independentemente dos outros documentos.

No carregamento do ficheiro para memória, os vários documentos são lidos pelo processo principal, e este envia os documentos lidos para os processos secundários.

Para o cálculo das distâncias dos documentos a cada gabinete é necessário cada processo saber as médias dos temas de todos os gabinetes, para este efeito os processos comunicam após cada cálculo da soma dos temas dos gabinetes dos seus documentos de modo a adicionar estas somas parciais e cada processo poder obter o valor de cada gabinete. Esta comunicação é global e sincronizada entre todos os processos.

Vamos então aglomerar a tarefa de cálculo de distâncias e mudança de gabinete e a tarefa de cálculo de somas de temas parciais dos gabinetes pois o cálculo anterior precisa de saber o gabinete a que cada documento pertence, esta aglomeração deve-se ao facto que

separar estas duas operações por processos diferentes iria levar à comunicação da posição de cada documento de um processo para o outro.

Quanto ao mapeamento do trabalho a realizar os documentos são separados em partes iguais pelos vários processos, excepto numa situação em que o número de documentos não é perfeitamente divisível pelo número de processos neste caso damos o pedaço mais pequeno ao processo principal para este se encontrar mais disponível para outras operações caso necessário.

No algoritmo descrito na Figure 1, começamos por carregar o ficheiro em memória colocando os totais de documentos, temas e gabinetes em variáveis globais estas variáveis são enviadas no momento para os outros processos e usando dois vectores de doubles declarados *static volatile*, um representa os documentos carregados e outro o estado dos gabinetes.

A distribuição dos documentos é feita pelo processo principal à medida que este carrega o ficheiro em memória enviando os documentos a serem processados por cada processo, depois de carregar o ficheiro e construir as estruturas necessárias começamos com o algoritmo principal:

1. Calcular para cada gabinete a soma de cada tema baseado nas classificações dos temas dos documentos atribuídos ao processo, assim como o número de documentos em cada gabinete
2. Todos os processos somam as suas somas de temas dos gabinetes e o número de documentos contados e calculam a média de cada gabinete
3. Calcular as distâncias entre cada documento atribuído ao processo e todos os gabinetes e mover o documento para o gabinete com a menor distância;
4. Se algum documento foi mudado de gabinete ir para o passo 1, se não ir para o passo 2

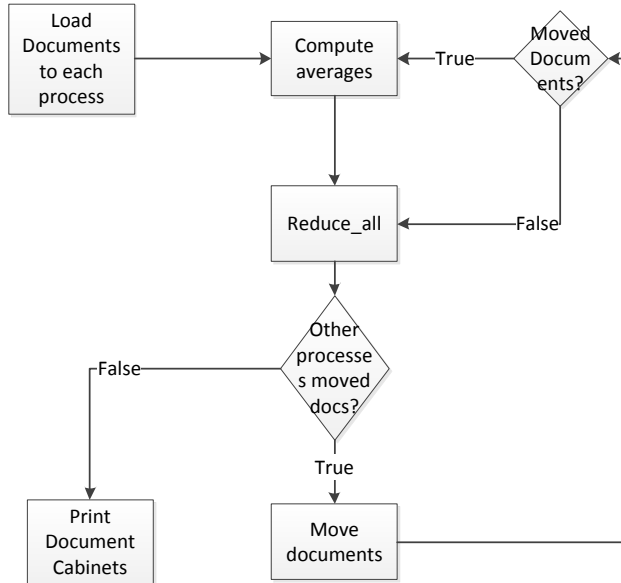


Figure 1. Algoritmo usado

5. Todos os processos comunicam para verificar se alguém moveu documentos, se ninguém moveu enviam as posições dos documentos para o processo principal e este imprime os resultados

3. Decomposição, Sincronização e Balanceamento de Carga

3.1. Loading Data

Na leitura do ficheiro e carregamento dos dados o processo principal vai lendo o ficheiro e primeiro realiza um Broadcast para distribuir um vector de inteiros que inclui os totais lidos das primeiras linhas.

E depois divide os documentos igualmente pelos processos, quando o processo principal tiver lido os documentos sobre os quais um processo secundário vai trabalhar, o processo principal realiza um Send com vector parcial com os documentos, e continua a ler o ficheiro após o envio, quando acabar de distribuir os pedaços pelos vários processos secundários, o processo principal fica com o ultimo pedaço do vector dos documentos.

A decisão de enviar os pedaços de documentos à medida que são lidos permite que os processos que recebem logo os seus documentos comecem a trabalhar enquanto o processo principal ainda está a carregar o ficheiro.

Quanto ao processo principal ficar com o ultimo pedaço foi decidido de modo ao processo principal ter em certos casos menos documentos que os processos

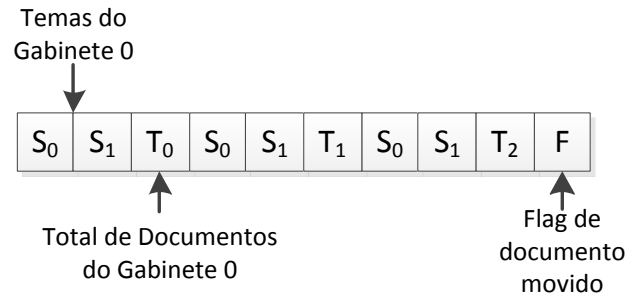


Figure 2. Estrutura do Vector de Gabinetes

secundários e assim se encontrar mais disponível para operações extraordinárias, estas no entanto não se verificam pois as nossas zonas de comunicação entre os processos são poucas e simples.

3.2. Cálculo de Médias

Cada processo tem aproximadamente documentos/processos documentos guardados, para calcular as médias dos gabinetes cada processo soma os vários temas de cada documento pertencente a um gabinete e conta o número de documentos que cada gabinete tem.

Estas somas são guardadas num vector de doubles que, por exemplo, se tivéssemos 3 gabinetes com 2 temas cada teria a estrutura na Figure 2.

Neste vector de doubles são guardados os temas de cada gabinete separados pelo contador dos documentos que cada gabinete tem atribuídos a ele no final do vector tem uma flag que é usada para saber se o processo moveu documentos de gabinetes no passo anterior, na 1ª execução do compute averages esta flag é colocada a 1 e alterada mais tarde pela função que move os documentos entre gabinetes.

Depois destas somas estarem calculadas para todos os gabinetes, é feito um AllReduce do vector dos gabinetes de cada processo de modo a todos os processos terem os totais globais, a estrutura do vector permite que isto seja feito apenas com um ALLReduce sem a necessidade de realizar operações separadas para as diferentes somas, cada processo então realiza a divisão final entre os temas e o número de documentos para calcular a média dos gabinetes.

3.3. Calcular Distâncias

Quando o vector de Gabinetes se encontrar actualizado os vários processos vão calcular as distâncias aos gabinetes de cada um dos seus documentos e movê-los se tal for o caso. De seguida o processo altera a

sua Flag no vector de Gabinetes se este processo tiver movido algum documento

Se a Flag de documento movido estiver a 0 o processo não recalcula novas somas para os Gabinetes pois tem as somas guardadas da iteração anterior, e estas matêm-se iguais pois nenhum dos seus documentos mudou de posição, o processo então salta para o passo de AllReduce.

No passo a seguir ao cálculo das médias é que é testada a condição de paragem do algoritmo, após o AllReduce se um ou mais processos tiverem movido documentos, a Flag irá se encontrar maior que zero em todos os processos então estes sabem que o algoritmo ainda não parou e devem continuar.

4. Resultados Obtidos

Devido à problemas imprevistos no desenvolvimento da nossa solução e devido à enorme afluência de processos ao nível do cluster não tivemos tempo suficiente para testar e analisar a fundo a performance da nossa solução.

Esta solução que entregamos junto com este relatório foi capaz de correr a maiorias dos nossos testes realizados.

Também entregamos uma versão do docs-omp-mpi.c que no momento desta entrega se encontrava a funcionar correctamente em todos os nossos testes, excepto eventuais problemas com as ocupações das máquinas. Esta versão funciona distribuída nas várias máquinas fornecidas com o comando availableNode e é possível usar a variável global para definir o número de threads a correr em cada processo.