Document Classification

CPD
Computação Paralela e Distribuida
Parallel and Distributed Computing
Serial+OpenMP - 2012-13

1. Introduction

The objective of the project is taking a set of D documents, with each document classified according to S subjects, and a number C of cabinets, assign documents to cabinets according to the similarity of subjects.

2. Description

We begin with loading the file into memory placing the totals of documents, subjects and cabinets into global variables, and using 2 structures that represent a Document and a Cabinet we create two vectors, one of pointers to the Documents loaded and another of pointers to the Cabinets, these vectors are global and declared static and volatile.

The round robin assignment of Documents is performed while loading the documents, so now we are free to start the main algorithm:

1. Calculate each subject average based on the documents of the cabinet;
2. Compute the distance between a document and each cabinet and move the document to the cabinet with shorter distance
3. Go to step 1 if a document was moved

On Step 1, for each cabinet we reset the subject scores to 0 and then for each document we check if it belongs to the cabinet and add it's subject score to the cabinet's subject score and count the document towards the total of documents belonging to the cabinet, after this we divide each cabinet's total subject score by the number of documents belonging to that cabinet in order to obtain the average.

On Step 2, for each document we calculate it's distance to each cabinet by calculating the normal of the vector created by the joining of the cabinet and document subject scores and change the document to the cabinet with the smallest normal, if any document is moved to a different cabinet we change a flag to note this occurrence.

3. Parallelization

3.1. Loading Data

Samuel estás mais bem equipado para falar sobre esta parte.

3.2. Computing Averages

We calculate the averages by assigning the cabinets to the available threads, this raises some issues with load balancing when there are less cabinets then available threads and because in most steps of the main algorithm the Cabinets will have different quantities of documents assigned which leads to different work loads for each Cabinet.

In order to address the load balancing issue we first attempted to change the manner in which the averages are summed.

Changing the order of the nested for loops in order to split the Documents among threads and have each thread summing the Document's subject scores to the respective Cabinet's subject scores, this choice had synchronization delays.

Parallelizing the inner loops of the original solution, instead of splitting the cabinets among threads we split the Documents among threads, but the constant overhead of thread creation lead to slower execution times.

Problems with Nested Parallelism. Samuel estás mais bem equipado para falar sobre esta parte.

In the end the initial solution was the one chosen because it performed better then all other alternatives we tested.

3.3. Computing Distances

When Computing the Distances we split the Documents among available threads using a simple pragma omp for and placing the auxiliary variables in the private list, this proved to be the simplest and most efficient solution.

This implementation guarantees even distribution of work among threads and has no critical region between threads because the threads only write in the Documents they are assigned and not on the Documents of other threads.

One change that helped us gain some additional speed-up was the use in each thread of a auxiliary variable pointer, declared static volatile, to a Cabinet that was used as a temporary pointer to the Cabinet to which the distance is being calculated , we believe this increased performance due to preventing the system from optimizing access to Cabinet structure and preventing some cache invalidations when documents accessed the same Cabinet pointer.

4. Evaluation

Falar nas razões para termos speedups, meter uma tabela para falar de speedups e efficiência com 2/4 cores e meter um gráfico comparativo.

5. Conclusions

????