Royal Institute of Technology

Introduction to High-Performance Computing, DN2258

Final project, parallel search

Sindri Magnússon, sindrim@kth.se, 871209-7156
Brynjar Smári Bjarnasson, bsbj@kth.se, 840824-4690

September 23, 2012

Contents

1 Introduction

The problem of searhing

1.1 The data

The

2 OpemMP

A Code

A.1 serial code

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <math.h>
#include <stdlib.h>
long long ae_load_file_to_memory(const char *filename, char
    **result)
{
  long long size = 0;
  FILE *f = fopen(filename, "rb");
  if (f == NULL)
  {
    *result = NULL;
    return -1; // -1 means file opening fail
  }
  fseek(f, 0, SEEK_END);
  size = ftell(f);
  fseek(f, 0, SEEK_SET);
  *result = (char *)malloc(size+1);
  if (size != fread(*result, sizeof(char), size, f))
  {
    free(*result);
    return -2; // -2 means file reading fail
  }
  fclose(f);
  (*result)[size] = 0;
  return size;
}

int read_file(char* input_file, char* key, int result_size_block,
    long long nr_lines, int line_size){
  //  input_file:        is direction to a data file.
  //  key:               the search string.
  //  result_size_block: is the block size of the array that
  //                     contains the resutls which contains the id
  //                     of the mathing data.
  //  nr_lines:          nr of lines in file.
  //  line_sie:          size of line in char
  //
  // It is assumed that all lines in input_file have the same
      length.

  // open up file
  //FILE *file;
```

```
42    //file = fopen(file_name,"r");
43    char line[line_size]; // line in file
44
45
46    int pch_id; // first column in the current line, corresponds to
          line id
47    char* pch_seq; // second column in the current file, corresponds
          to the seq in this line
48
49    int result_size = result_size_block; // initualized result size
50    int *result;
51    result = malloc(sizeof(*result)*result_size);
52    int result_counter = 0;
53    long long i;
54    int key_len = strlen(key);
55    for ( i = 0; i < nr_lines; i++ ) {
56      strncpy(line,&input_file[i*line_size],line_size);
57//      pch_id = atoi(strtok(line,"\t\n"));
58//      pch_seq = strtok(NULL,"\t\n");
59//      if (pch_seq == NULL){
60//        printf("pch_id,%i", pch_id);
61//      }
62      //if ( strcmp(pch_seq, key) == 0){
63      line[line_size - 1] = '\0';
64      if ( strcmp(&line[line_size-(key_len+1)], key) == 0){
65        if (result_size == result_counter ){
66          result_size = result_size + result_size_block;
67          result = realloc(result, result_size*sizeof(*result) );
68          if (result == NULL){
69            printf("Error reallocating memory\n");
70            exit(1);
71          }
72        }
73        result[result_counter] = pch_id;
74        result_counter++;
75      }
76    }
77
78    return result_counter;
79  }
80
81
82  int main( int argc, const char* argv[] )
83  {
84    time_t start,end;
85    //int data_size = 100000000;
86  //  int string_size = 6;
87  //  int block_size = 1.5*(data_size/pow(10,string_size));
88  //  int read_count;
89    char* file_name = "../data/file.txt";
90    char* result;
91    long long nr_bytes;
92    start = time(NULL);
93    nr_bytes = ae_load_file_to_memory(file_name,&result);
94    end = time(NULL);
95    printf("load file %f\n",difftime(end,start));
96    int line_size;
97    long long i;
98    for ( i=0 ; i<nr_bytes ; i++ ){
99      if ( result[i]=='\n' ){
100       line_size = i+1;
101       break;
```

```
102     }
103   }
104
105   // assume that all the lines in the file have the same size
106   long long data_size = nr_bytes / line_size;
107
108   int string_size = 6;
109   int block_size = 1.5*(data_size/pow(10,string_size));
110   int read_count;
111   start = time(NULL);
112   read_count =
          read_file(result,"123123",block_size,data_size,line_size);
113   end = time(NULL);
114   printf("search: %f\n",difftime(end,start));
115   printf("result found: %i\n", read_count);
116
117 }
```

../src/search_serial.c

A.2 OpenMP

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <time.h>
4  #include <math.h>
5  #include <omp.h>
6  #include <stdlib.h>
7  long long ae_load_file_to_memory(const char *filename, char
       **result)
8  {
9    long long size = 0;
10   FILE *f = fopen(filename, "rb");
11   if (f == NULL)
12   {
13     *result = NULL;
14     return -1; // -1 means file opening fail
15   }
16   fseek(f, 0, SEEK_END);
17   size = ftell(f);
18   fseek(f, 0, SEEK_SET);
19   *result = (char *)malloc(size+1);
20   if (size != fread(*result, sizeof(char), size, f))
21   {
22     free(*result);
23     return -2; // -2 means file reading fail
24   }
25   fclose(f);
26   (*result)[size] = 0;
27   return size;
28 }
29
30 int read_file(char* input_file, char* key, int result_size_block,
       long long nr_lines, int line_size){
31   //  input_file:        is direction to a data file.
32   //  key:               the search string.
33   //  result_size_block: is the block size of the array that
34   //                     contains the resutls which contains the id
35   //                     of the mathing data.
36   //  nr_lines:          nr of lines in file.
37   //  line_sie:          size of line in char
```

```
38    //
39    // It is assumed that all lines in input_file have the same
          length.
40
41    // open up file
42    //FILE *file;
43    //file = fopen(file_name,"r");
44    char line[line_size]; // line in file
45
46
47    int pch_id; // first column in the current line, corresponds to
          line id
48    char* pch_seq; // second column in the current file, corresponds
          to the seq in this line
49
50    //#pragma omp parallel private(pch_id,pch_seq)
          shared(result,result_counter)
51    int result_size = result_size_block; // initualized result size
52    int *result;
53    result = malloc(sizeof(*result)*result_size);
54    int result_counter = 0;
55    long long i;
56    int key_len = strlen(key);
57    #pragma omp parallel for private(pch_id,pch_seq,i,line)
          shared(result,result_counter,result_size,key,result_size_block)
58    for ( i = 0; i < nr_lines; i++ ) {
59      strncpy(line,&input_file[i*line_size],line_size);
60//      pch_id = atoi(strtok(line,"\t\n"));
61//      pch_seq = strtok(NULL,"\t\n");
62//      if (pch_seq == NULL){
63//        printf("pch_id,%i", pch_id);
64//      }
65      //if ( strcmp(pch_seq, key) == 0){
66      line[line_size - 1] = '\0';
67      if ( strcmp(&line[line_size-(key_len+1)], key) == 0){
68        if (result_size == result_counter ){
69          result_size = result_size + result_size_block;
70          result = realloc(result, result_size*sizeof(*result) );
71          if (result == NULL){
72            printf("Error reallocating memory\n");
73            exit(1);
74          }
75        }
76        result[result_counter] = pch_id;
77        result_counter++;
78      }
79    }
80
81    return result_counter;
82  }
83
84
85  int main( int argc, const char* argv[] )
86  {
87    double start,end;
88    double dif;
89    //int data_size = 100000000;
90//    int string_size = 6;
91//    int block_size = 1.5*(data_size/pow(10,string_size));
92//    int read_count;
93    char* file_name = "../data/file.txt";
94    char* result;
```

```
95    long long nr_bytes;
96    start = omp_get_wtime();
97    nr_bytes = ae_load_file_to_memory(file_name,&result);
98    end = omp_get_wtime();
99    dif = end-start;
100   printf("LoadFile: %f\n", dif);
101   int line_size;
102   long long i;
103   for ( i=0 ; i<nr_bytes ; i++ ){
104     if ( result[i]=='\n' ){
105       line_size = i+1;
106       break;
107     }
108   }
109
110   // assume that all the lines in the file have the same size
111   long long data_size = nr_bytes / line_size;
112
113   int string_size = 6;
114   int block_size = 1.5*(data_size/pow(10,string_size));
115   int read_count;
116   start = omp_get_wtime();
117   read_count =
          read_file(result,"123123",block_size,data_size,line_size);
118   end = omp_get_wtime();
119   dif = end - start;
120   printf("Search: %f\n",dif);
121   printf("result found: %i\n", read_count);
122
123 }
```

../src/search_openmp.c