High performance computing systems

Lab 4

Dept. of Computer Architecture
Faculty of ETI
Gdansk University of Technology

Robert Kałaska

The following example presents a basic functions in OpenMP. OpenMP is another framework for parallel programming. It is mostly used within one machine. First part represent different ways of synchronization and second part presents how to use parallel for loop and take advantage of reduction operation.

Compilation instructions:

gcc -fopenmp sample.c -o sample -lm

```
#include <stdio.h>

#include <stdlib.h>

#include <omp.h>

#include <math.h>


int threadnum=4;


int main(int argc,char **argv) {


  //set number of threads

  omp_set_num_threads(threadnum);


  //lock object - usable for synchronization

  omp_lock_t writelock;
```

```
  omp_init_lock(&writelock);

  int value=0;

  printf("Value at the begining is %d \n",value);



  //parallel - all thread execute below code in parallel
  #pragma omp parallel shared(value)
  {
    //synchronization with atomic operation
    #pragma omp atomic update
    value++;
  }

 printf("Value after parallel (4 thread) is %d \n",value);

//parallel -  all thread execute below code in parallel - synchronization with lock
#pragma omp parallel shared(value)
{
  omp_set_lock(&writelock);
  value++;
  omp_unset_lock(&writelock);
}

 printf("Value after parallel (4 thread) is %d \n",value);
```

```
//parallel - all thread execute below code in parallel but critical section will be executed only by one thread at the same time

 #pragma omp parallel shared(value)

{

  #pragma omp critical

  {

    value++;

  }

}



  long orders = 1000000000;

  long i=0;

  double beer;

  double total_beers = 0.0;



  //parallel for loop - all threads execute in parallel different parts of iterations

  #pragma omp parallel for private(beer) reduction(+:total_beers) // reduction allow to execute operation on chosen variable in all threads

  for(i=0;i<orders;i++) {

    beer = 1.0/pow((double)2,(double)i);

    total_beers = total_beers + beer;

  }



  printf("Your order is %f beers",total_beers);
```

```
}
```