PPAR Tutorial 4 **Master 1 in CS**

Getting started with MPI

1 Installing MPI

Please follow the steps below:

1. Download MPICH-3.2 from "`http://www.mpich.org`".
2. Untar the downloaded archive and quickly explore the content.
3. Decide where to install this MPICH distribution. We will suppose the directory you will choose is stored in the environment variable "`MPIHOME`".
4. Configure the installation (in Bash):
   ```
   $ ./configure --disable-fortran --disable-f77 --disable-fc
                 --prefix=$MPIHOME 2>&1 | tee c.txt
   ```
5. Build the library: `$ make 2>&1 | tee m.txt`
6. Install the library: `$ make install 2>&1 | tee mi.txt`
7. Include the path to the binaries to the environment variable `PATH`
   (add `export PATH=<HOME>/local/mpich-install/bin:$PATH` to your `.bashrc` file).
8. Check whether `mpicc` and `mpirun` are now available.

2 Mnemonics and π

A short mnemonic for remembering the first seven decimal digits of π is "How I wish I could calculate pi". Much longer texts have however been conceived for π, which have the particular property to correspond to its decimal digits.

For every word in the text, we need to consider, as a digit, its size in terms of characters, and to keep the digits in the same order as for the words in the text. If the size of some words is equal or greater than 10, then the two digits need to be added to the sequence. The special characters do not contribute to the computation of the word size, but they are rather exploited to add the digit "0" in the sequence (exception made for the point ".", the blank character and the new-line character). Finally, if a number is included in the text, it needs to be copied in the digit sequence as is.

Exercise A

Download the code available at the address:

`/share/m1info/PPAR/TP/TP3`

and complete the missing parts in order to perform the conversion of texts in digit sequences by following the rules above. Execute your program by considering the texts that you can find at the same address and verify whether it is able to correctly generate the first digits of π.

Exercise B

Write an MPI parallel version of your program. The text can be read by all processors, while the conversion of the text needs to be performed in parallel: every processor can work on part of the text. Pay particular attention to not cut words when splitting up the text. At the end of the computations, the processors do not need to communicate, they can just print their partial results on the screen.

Exercise C

You may have noticed that the random access to the standard input by the processors does not allow us to have a clear visualization of the digit sequence. Instead to print the digits as soon as they are generated, we can keep them in an array of integers, and print them all at the end of the computations. Moreover, by using the function `MPI_Barrier`, we can force the processors to print the content of their local arrays in the right order.