Multicore Programming
Handout 1: Installing GCC Cilk Plus

Leo Ferres
Department of Computer Science
Universidad de Concepción
Email: lferres@inf.udec.cl
February 19, 2013

1 Introduction

For our lab work, we will need, among other tools, Intel Cilk Plus. Cilk Plus is an extension to the C and C++ programming languages that supports data and task multi-threaded parallel computing. We will see the theory behind Cilk Plus in class. This handout only tells you how to install and test Cilk Plus on your own computer.

Cilk Plus comes in the Intel Compiler suite (icc) and in the GNU Compiler Collection (included in gcc 4.7 as a patch, and in the release branch for gcc $ \geq $ 4.8). This guide is about installing gcc 4.8. gcc $ \geq $ 4.7 also comes with some useful GNU Extensions, particularly the atomic builtins that we will surely find occasion to use.

Our labs will also involve the use of a 64-bit computer for some of the problem sets. Although this document will guide you through the installation of Cilk Plus for both 32- and 64-bit computers, the emphasis and support will be for the latter.

Finally, we will target Linux systems. We have not tested Cilk Plus on Windows. And we probably just won’t bother.

2 64-Bit Computer

Obviously, to install a 64-bit version of the software, you need both: 64-bit compatible hardware, and a 64-bit instance of the Linux operating system. If executing the command

$ grep -m1 -e " lm " /proc/cpuinfo

returns non-empty (that is, it returns some string where lm is present), that means you have compliant 64-bit hardware (by the way, lm means “long mode”). Finally, if

$ uname -m

returns something like x86_64, then that means that you’re also running a 64-bit Linux OS and that you’re all set regarding OS and hardware. In what remains, we will be assuming you have a working gcc installation to bootstrap the new gcc. If not, just sudo apt-get install build-essential.

---

2 Cilk Plus is also installed on the computers in the Software Engineering lab
3 http://gcc.gnu.org/onlinedocs/gcc/_005f_005fatomic-Builtins.html
2.1 Required libraries

The next thing you need to do is install the requirements for your new version of gcc. These are gmp (The GNU Multiple Precision Arithmetic Library), mpfr (GNU Multiple Precision Floating-Point Reliably) and mpc (GNU Multiple Precision Complex Library). Although it is possible to install this from source, it is suggested that you simply install from the repositories (if you have a relatively new version of your OS distribution). In the case of Ubuntu, just do

$ sudo apt-get install libgmp-dev libmpfr-dev libmpc-dev

in that order. Otherwise, you can still install from source. Notice that the installation has to be performed in the given order: gmp → mpfr → mpc. Download the latest versions of gmp from http://gmplib.org/, mpfr from http://www.mpfr.org/ and of mpc from http://multiprecision.org/. Configure, build and install mpc by executing the following commands:

[mpc]
$ sudo mkdir -p /opt/gmp-ver
$ tar -jxvf gmp-ver.tar.bz2
$ cd gmp-ver
$ ./configure --prefix=/opt/gmp-ver
$ make
$ make check
$ sudo make install

Then configure, build and install mpfr following similar commands:

[mpfr]
$ sudo mkdir -p /opt/mpfr-ver
$ tar -jxvf mpfr-ver.tar.bz2
$ cd mpfr-ver
$ ./configure --prefix=/opt/mpfr-ver --with-gmp=/opt/gmp-ver
$ make
$ make check
$ sudo make install

and finally do the same for mpc

[mpc]
$ sudo mkdir -p /opt/mpc-ver
$ tar -zxvf mpc-ver.tar.gz
$ cd mpc-ver
$ ./configure --prefix=/opt/mpc-ver --with-gmp=/opt/gmp-ver --with-mpfr=/opt/mpfr-ver
$ make
$ make check
$ sudo make install

where ver corresponds to the latest versions of the libraries that you have downloaded.

Notice that if you need to install everything from source, you may need to provide the following set of dependencies:

$ sudo apt-get install build-essential gcc-multilib flex bison libgmp-dev libmpfr-dev mpc libmpc-dev libbpl10.11-dev libcloog-ppl-dev zlib1g zlib1g-dev libc6-dev* m4 libgmp3-dev

---

4This has been borrowed from http://software.intel.com/en-us/articles/how-to-build-gcc-cilkplus-branch-in-64bit-ubuntu-1204
5Thanks to Luis Palomo for bringing this to my attention.
2.2 GCC

Check out the latest gcc version from the svn repositories. All GCC development is done in SVN. The Git repo is a read-only mirror of the SVN repo. Current work happens on the subversion trunk. Certain larger projects happen in a branch (Cilk Plus is a branch) and are then merged to trunk.

```
$ svn checkout svn://gcc.gnu.org/svn/gcc/branches/cilkplus gcc-cilk-src
```

Next, Create two new directories, gcc-cilk-obj and gcc-cilk, in the same directory containing gcc-cilk-src.

```
$ mkdir gcc-cilk-obj gcc-cilk
```

Now configure, build and install gcc:

```
[gcc]
$ cd gcc-cilk-obj
$ ../gcc-cilk-src/configure --prefix=/absolute/path/to/gcc-cilk --disable-multilib --enable-languages=c
$ make
$ sudo make install
```

**IMPORTANT NOTE**: If you have installed the required libraries from source, then you will have to let the configure script know where they reside. Thus, the configure command would look like

```
$ configure --prefix=/absolute/path/to/gcc-cilk --with-gmp=/opt/gmp-ver --with-mpfr=/opt/mpfr-ver --with-mpc=/opt/mpc-ver --disable-multilib --enable-languages=c
```

The new GCC binary is /path/to/gcc-cilk/bin/gcc. Now you have to set up the environment to use your new installation:

```
[environment script]
export PATH=path/to/gcc-cilk/bin:$PATH
export LD_LIBRARY_PATH=path/to/gcc-cilk/lib
export LIBRARY_PATH=path/to/gcc-cilk/lib
```

Save this in, say cilk.sh, and then whenever you want to run in this environment, first do

```
$ source cilk.sh
```

2.3 Cilkutils

There are two interesting utilities that Intel makes available for use with Cilk Plus: cilkscreen (a data race detector) and cilkview a parallel/scalability measuring tool. To install these tools as well download the latest version from:

```
```

```
[cilkutil]
$ tar vxzf cilkutils.tgz
```

And that’s all there is to the installation. The only thing that remains to be done is to add the path to the cilkutil/bin directory

```
export PATH=/path/to/cilkutil/bin:$PATH
```

to your cilk.sh environment script. Now you should be able to run both utilities.
2.4 Test the installation

To test your installation, you may copy the following code into a text editor:

```c
#include <stdio.h>
#include <cilk/cilk.h>
#include <cilk/cilk_api.h>

int fib(int n) {
    int a,b;
    if (n < 2) return n;
    a = cilk_spawn fib(n-1);
    b = fib(n-2);
    cilk_sync;
    return a+b;
}

int main(int argc, char *argv[]) {
    printf "%d\n",fib(40));
    return 0;
}
```

To compile this program:

```
$ gcc fib.c -o fib -fcilkplus -lcilkrt
```

To set maximum number of worker threads for the Cilk Plus runtime:

```
export CILK_NWORKERS=N
```

To fix error using cilkview: “The Operating System configuration prevents Pin from using the default (parent) injection mode”:

```
$ sudo bash
$ echo 0 > /proc/sys/kernel/yama/ptrace_scope
```

Run cilkview:

```
$ cilkview -trials all 4 ~/fib
```

3 32-Bit Computer

*NOT IMPLEMENTED YET*

*I DON'T HAVE A 32-BIT COMPUTER TO TEST THIS*

4 Troubleshooting
## Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Comment</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-12-29-17:20</td>
<td>Initial version of doc</td>
<td>Leo</td>
</tr>
<tr>
<td>2013-01-17-15:00</td>
<td>Running experiments on Keira</td>
<td>José</td>
</tr>
</tbody>
</table>