



Instituto Politécnico Nacional.
Escuela Superior de Cómputo.



U.A. Analysis and design of parallel algorithms.

Práctica 0. “Instalando MPI en Ubuntu.”

Moreno González Gabriela.

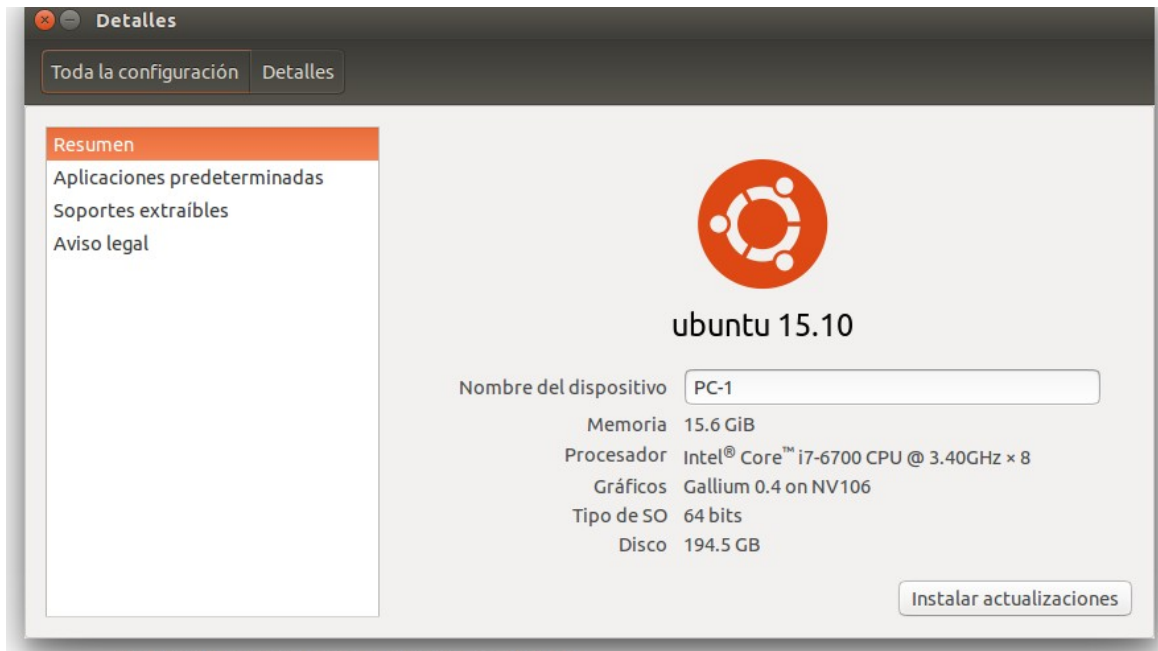
Profa. Morales Guitrón Sandra Luz.

3CV9.

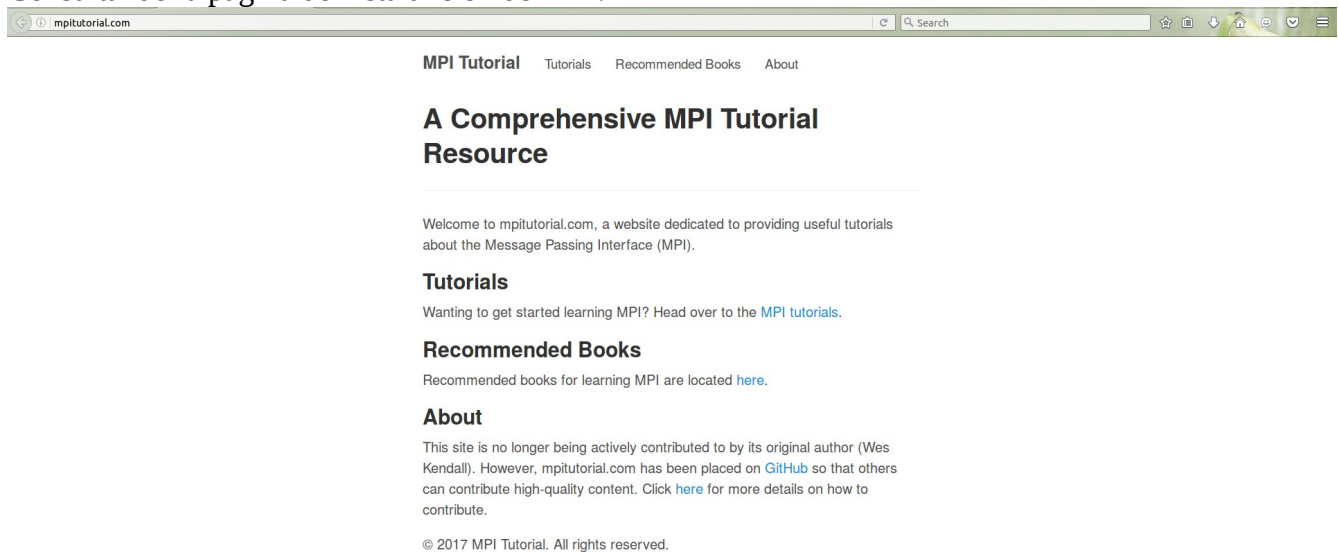
Fecha: 04/09/2017

Instalando MPI en Ubuntu.

Primeramente vamos a checar las características de nuestro Sistema Operativo:



Tenemos Ubuntu superior a 15.0 y un procesador Intel Core i7, por lo que usar MPI será fácil. Consultando la página de instalación de MPI.



Y consultando los tutoriales:

mpitutorial.com/tutorials/

Search

MPI Tutorial

Tutorials

Recommended Books

About

Tutorials

Welcome to the MPI tutorials! In these tutorials, you will learn a wide array of concepts about MPI. Below are the available lessons, each of which contain example code.

The tutorials assume that the reader has a basic knowledge of C, some C++, and Linux.

Introduction and MPI installation

- [MPI tutorial introduction](#)
- [Installing MPICH2 on a single machine](#)
- [Launching an Amazon EC2 MPI cluster](#)
- [Running an MPI cluster within a LAN](#)
- [Running an MPI hello world application](#)

Blocking point-to-point communication

- [Sending and receiving with MPI_Send and MPI_Recv](#)
- [Dynamic receiving with MPI_Probe and MPI_Status](#)
- [Point-to-point communication application - Random walking](#)

Basic collective communication

- [Collective communication introduction with MPI_Bcast](#)
- [Common collectives - MPI_Scatter, MPI_Gather, and MPI_Allgather](#)
- [Application example - Performing parallel rank computation with basic collectives](#)

Advanced collective communication

mpitutorial.com/tutorials/installing-mpich2/

Search

MPI Tutorial

Tutorials

Recommended Books

About

Installing MPICH2 on a Single Machine

Author: Wes Kendall

MPI is simply a standard which others follow in their implementation. Because of this, there are a wide variety of MPI implementations out there. One of the most popular implementations, MPICH2, will be used for all of the examples provided through this site. Users are free to use any implementation they wish, but only instructions for installing MPICH2 will be provided. Furthermore, the scripts and code provided for the lessons are only guaranteed to execute and run with the latest version of MPICH2.

MPICH2 is a widely-used implementation of MPI that is developed primarily by Argonne National Laboratory in the United States. The main reason for choosing MPICH2 over other implementations is simply because of my familiarity with the interface and because of my close relationship with Argonne National Laboratory. I also encourage others to check out OpenMPI, which is also a widely-used implementation.

Installing MPICH2

The latest version of MPICH2 is available [here](#). The version that I will be using for all of the examples on the site is 1.4, which was released June 16, 2011. Go ahead and download the source code, uncompress the folder, and change into the MPICH2 directory.

```
>>> tar -xzf mpich2-1.4.tar.gz
>>> cd mpich2-1.4
```

mpitutorial.com/tutorials/installing-mpich2/

```
>>> tar -xzf mpich2-1.4.tar.gz
>>> cd mpich2-1.4
```

Once doing this, you should be able to configure your installation by performing `./configure`. I added a couple of parameters to my configuration to avoid building the MPI Fortran library. If you need to install MPICH2 to a local directory (for example, if you don't have root access to your machine), type `./configure --prefix=/installation/directory/path` For more information about possible configuration parameters, type `./configure --help`

```
>>> ./configure --disable-fortran
Configuring MPICH2 version 1.4 with '--disable-f77' '--disable-fc'
Running on system: Darwin Wes-Kendalls-Macbook-Pro.local 10.7.0 Darw
in Kernel Version 10.7.0: Sat Jan 29 15:17:16 PST 2011; root:xnu1504
.9.37~1/RELEASE_I386 i386
checking for gcc... gcc
```

When configuration is done, it should say "Configuration completed." Once this is through, it is time to build and install MPICH2 with `make`; `sudo make install`.

```
>>> make; sudo make install
Beginning make
Using variables CC='gcc' CFLAGS=' -O2' LDFLAGS=' ' F77=' ' FFLAGS='
' FC=' ' FCFLAGS=' ' CXX='c++' CXXFLAGS=' -O2' AR='ar' CPP='gcc-E'
CPP
```

If your build was successful, you should be able to type `mpiexec --version` and see something similar to this.

```
>>> mpiexec --version
HYDRA build details:
  Version:                3.1.4
  Release Date:            Fri Feb 20 15:02:56 CST 2015
  CC:                      gcc
  CXX:                     g++
```

Así, siguiendo las instrucciones:

```
alumno@PC-1: ~/Escritorio
alumno@PC-1:~$ dpkg --get-architecture | grep compiler
ii  g++                                amd64        GNU C++ compiler 4:5.2.1-3ubuntu1
ii  g++-5                              amd64        GNU C++ compiler 5.2.1-22ubuntu2
ii  gcc                                amd64        GNU C compiler 4:5.2.1-3ubuntu1
ii  gcc-5                              amd64        GNU C compiler 5.2.1-22ubuntu2
ii  hardening-includes                 amd64        GNU C compiler 2.7ubuntu1
ening                               all          Makefile for enabling compiler flags for security hard
ii  liblvm3.6v5:amd64                 amd64        Modular compiler and toolchain technologies, runtime l
library
ii  libxkbcommon0:amd64               amd64        library interface to the XKB compiler - shared library
alumno@PC-1:~$ sudo apt-get install libcr-dev mpich2 mpich2-doc
[sudo] password for alumno:
Lo sentimos, vuelva a intentarlo.
[sudo] password for alumno:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Y terminando de instalar nos muestra esto:

```
alumno@PC-1: ~/Escritorio
) en modo automático
Configurando libcr0 (0.8.5-2.2) ...
Configurando libcr-dev (0.8.5-2.2) ...
Configurando mpich-doc (3.1-6) ...
Configurando mpich2-doc (3.1-6) ...
Configurando hwloc-nox (1.11.0-1) ...
Configurando ocl-icd-libopencl1:amd64 (2.2.7-2) ...
Configurando libhwloc-plugins (1.11.0-1) ...
Configurando libmpl1:amd64 (3.1-6) ...
Configurando libmpich12:amd64 (3.1-6) ...
Configurando libmpl-dev (3.1-6) ...
Configurando libopa1:amd64 (3.1-6) ...
Configurando libopa-dev (3.1-6) ...
Configurando libmpich-dev (3.1-6) ...
DEBUG: postinst in libmpich-dev
update-alternatives: utilizando /usr/include/mpich para proveer /usr/include/mpi
(mpi) en modo automático
Configurando mpich (3.1-6) ...
update-alternatives: utilizando /usr/bin/mpirun.mpich para proveer /usr/bin/mpir
un (mpirun) en modo automático
Configurando mpich2 (3.1-6) ...
Procesando disparadores para libc-bin (2.21-0ubuntu4.3) ...
alumno@PC-1:~$ cd Descargas/
alumno@PC-1:~/Descargas$ cd ..
```

Entonces, para probar la configuración, hagamos un hola mundo con MPI. Nuestro código fuente es:

```
mpi_hello.c (~/Escritorio) - gedit
/* C Example */
#include <mpi.h>
#include <stdio.h>

int main (int argc, char* argv[])
{
    int rank, size;

    MPI_Init (&argc, &argv); /* starts MPI */
    MPI_Comm_rank (MPI_COMM_WORLD, &rank); /* get current process id */
    MPI_Comm_size (MPI_COMM_WORLD, &size); /* get number of processes */
    printf( "Hello world from process %d of %d\n", rank, size );
    MPI_Finalize();
    return 0;
}
```

Veamos como se compila y ejecuta:

```
alumno@PC-1:~/Escritorio$ mpicc mpi_hello.c -o hello
alumno@PC-1:~/Escritorio$ mpirun -np 2 ./hello
Hello world from process 0 of 2
Hello world from process 1 of 2
alumno@PC-1:~/Escritorio$
```

Así, comprobamos que MPI ha sido instalado de forma correcta en la máquina.

Conclusiones:

Linux es de los sistemas operativos que comienza a controlar toda la forma en la que se realizan códigos, permitiendo a sus usuarios alterar el código del sistema operativo y dejando que hagan las modificaciones a su gusto. El hecho de que MPI sea más fácil de instalar en linux que en Windows tiene que ver con la forma en la que Windows maneja su sistema operativo, permitiendo que linux tome ventaja de su open source, mientras que windows si detecta algo que no tiene licencia o no es de su fabricante es complicado estar instalando el software. Habrá que probar si en MAC se puede realizar de forma fácil como en linux.

Bibliografía:

<http://mpitutorial.com/tutorials/installing-mpich2/>

<https://jetcracker.wordpress.com/2012/03/01/how-to-install-mpi-in-ubuntu/>