Install OpenCL on Debian, Ubuntu and Mint orderly

Libraries - can't have enough

If you read different types of manuals how to compile OpenCL software on Linux, then you can get dizzy of all the LD-parameters. Also when installing the SDKs from AMD, Intel and NVIDIA, you get different locations for libraries, header-files, etc. Now GPGPU is old-fashioned and we go for heterogeneous programming, the chances get higher you will have more SDKs on your machine. Also if you want to keep it the way you have, reading this article gives you insight in what the design is after it all. Note that Intel's drivers don't give OpenCL support for their GPUs, but CPUs only.

As my mother said when I was young: "actually cleaning up is very simple". I'm busy creating a PPA for this, but that will take some more time.

First the idea. For developers OpenCL consists of 5 parts:

- GPUs-only: drivers with OpenCL-support
- The OpenCL header-files

Vendor specific libraries (needed when using -lOpenCL)
libOpenCL.so -> a special driver

- An installable client driver

Currently GPU-drivers are always OpenCL-capable, so you only need to secure 4 steps. These are discussed below.

Please note that in certain 64-bit distributions there is not lib64, but only 'lib' and 'lib32'. If that is the case for you, you can use the commands that are mentioned with 32-bit.
Header-files
update: A new package "opencl-headers" installs exactly these files for you. Even better: ocl-icd-opencl-dev installs everything for OpenCL 1.2 (problems with OpenCL 2.0 still)

No more export CPPFLAGS="-I/some_directory/opencl_sdk/include" at last! All SDKs provide the OpenCL 1.1 header-files originated from Khronos (or should).

We only need to put all headers found from the Khronos-webpage in **/usr/include/CL/**:
cd /usr/include

sudo mkdir CL

cd CL

sudo wget http://www.khronos.org/registry/cl/api/1.2/**cl_d3d10.h**
http://www.khronos.org/registry/cl/api/1.2/**cl_d3d11.h**
http://www.khronos.org/registry/cl/api/1.2/**cl_dx9_media_sharing.h**
http://www.khronos.org/registry/cl/api/1.2/**cl_ext.h**

http://www.khronos.org/registry/cl/api/1.2/**cl_gl_ext.h**
http://www.khronos.org/registry/cl/api/1.2/**cl_gl.h**
http://www.khronos.org/registry/cl/api/1.2/**cl.h**
http://www.khronos.org/registry/cl/api/1.2/**cl_platform.h**
http://www.khronos.org/registry/cl/api/1.2/**opencl.h**
http://www.khronos.org/registry/cl/api/1.2/cl.hpp ;

If you are on mobile, also get EGL:
sudo wget http://www.khronos.org/registry/cl/api/1.2/**cl_egl.h**
If you want **1.1** headers, do the following:
cd /usr/include

sudo mkdir CL

cd CL

sudo wget http://www.khronos.org/registry/cl/api/1.1/**cl_d3d10.h**
http://www.khronos.org/registry/cl/api/1.1/**cl_ext.h**
http://www.khronos.org/registry/cl/api/1.1/**cl_gl_ext.h**
http://www.khronos.org/registry/cl/api/1.1/**cl_gl.h**
http://www.khronos.org/registry/cl/api/1.1/**cl.h**
http://www.khronos.org/registry/cl/api/1.1/**cl_platform.h**
http://www.khronos.org/registry/cl/api/1.1/**opencl.h**
http://www.khronos.org/registry/cl/api/1.1/cl.hpp ;

sudo wget http://www.khronos.org/registry/cl/api/1.1/**cl_egl.h**
Now you can be sure you have the correct header-files.

Libaries

All vendors have their favourite spot to put their libraries; but actually a "just put your coat where you find a spot" is not the best to do. According to the best answer on stackoverflow, the libraries should be in /usr/local/lib, but since these are shared libraries, Intel has found a good location: /usr/lib/OpenCL/vendors/. There was some discussion about "vendors", but think of various wrapper-libaries, IBM's OpenCL Common Runtime, and such. So I agree with their choice.

Intel

update: Intel recently has completely changed the drivers for Linux. They now install in /opt/intel. Best is to copy all files from /opt/intel/opencl-1.2-x.x.xxxxxx to /usr/lib/OpenCL/vendors/intel to keep it orderly. If you choose not to, replace /usr/lib/OpenCL/vendors/intel with **/opt/intel/opencl-1.2-x.x.xxxxxx.**

The provided rpm can be converted to deb and then works if libnuma1 is installed:

apt-get install libnuma1
alien *.rpm
dpkg -i *.deb

Though they've put their libraries at a nice spot, they made a little mistake. They put their libOpenCL.so in /usr/lib or /usr/lib64, instead of using a symbolic link. Below I discuss separately all around libOpenCL.so, since this is an important library. You need to

copy it to the right directory. For 64 bit:

sudo cp /usr/lib64/libOpenCL.so /usr/lib64/OpenCL/vendors/intel/

For 32 bit systems:

sudo cp /usr/lib/libOpenCL.so /usr/lib/OpenCL/vendors/intel

It is very possible that if you installed another OpenCL SDK later, the library is lost. Not a real problem as explained later, but then you know.

To make the libraries available, I created opencl-vendor-intel.conf in /etc/ld.so.conf.d with the content (64 bit):
[raw]
echo "/usr/lib64/OpenCL/vendors/intel" > /etc/ld.so.conf.d/opencl-vendor-intel.conf

In case you need to have 32-bit libraries too, you can add the location at the end of that file. And for 32 bit systems:

echo "/usr/lib/OpenCL/vendors/intel" > /etc/ld.so.conf.d/opencl-vendor-intel.conf
[/raw]
Then run

ldconfig

to start to using the new LD-location.

AMD

Edit: as suggested by Steffen Moeller in the comments, installing the deb-files in http://wiki.debian.org/ATIStream is easier. Just check if the files are at the right place.

The AMD APP Installer let's you choose where you want to put the SDK. Just put it somewhere you want the SDK to be. Go to the root of the AMD-APP-SDK and move the lib-directory to /usr/lib(64)/OpenCL/vendors/, for 64 bit systems:

mkdir -p /usr/lib64/OpenCL/vendors/amd/
mv lib/x86_64/* /usr/lib64/OpenCL/vendors/amd/

And for 32 bit systems:

mkdir -p /usr/lib/OpenCL/vendors/amd/
mv lib/x86/* /usr/lib/OpenCL/vendors/amd/

Then we need to add them to ld-config. For 64 bit:
[raw]
echo "/usr/lib64/OpenCL/vendors/amd" > /etc/ld.so.conf.d/opencl-vendor-amd.conf

And for 32 bit systems:

echo "/usr/lib/OpenCL/vendors/amd" > /etc/ld.so.conf.d/opencl-vendor-amd.conf
[/raw]
Then run

ldconfig

NVIDIA

This is somewhat hard. You probably want to use CUDA too, so for that reason we leave the libraries in /usr/local/cuda/lib/ to avoid breaking software. Of course I prefer them to be tidied up under /usr/lib(64)/OpenCL/vendors/ but it is no use to make a symbolic link. Installer can be found here.

Then we need to add them to ld-config, if you haven't done that. For 64 bit:
[raw]
echo "/usr/local/cuda/lib64" > /etc/ld.so.conf.d/opencl-vendor-nvidia.conf
echo "/usr/local/cuda/lib" >> /etc/ld.so.conf.d/opencl-vendor-nvidia.conf

For 32 bit:

echo "/usr/local/cuda/lib" > /etc/ld.so.conf.d/opencl-vendor-nvidia.conf
[/raw]
Then run

ldconfig

LibOpenCL.so

This library handles the selecting of the platforms (the vendors) and providing the correct libraries to the software needing the functionality. It is located under /usr/lib or /usr/lib64. You need to select which vendor you want to use. I personally think this driver should be open sourced and not from a specific vendor. Pick **one** (first line 64, second 32) out of these 6. But... from my own experience both AMD and Intel give you versions that work best with all 3 platforms, so I suggest you go for one of these.

Khronos open source

Get the "OpenCL 1.2 Installable Client Driver (ICD) Loader" from http://www.khronos.org/registry/cl/ and make the project (needs cmake). In the bin-directory there will be libOpenCL.so.1.2. Remove all files startting with libOpenCL.so* and copy libOpenCL.so.1.2 to /usr/lib/.

sudo ln -s /usr/lib64/libOpenCL.so /usr/lib64/libOpenCL.so.1.2
sudo ln -s /usr/lib/libOpenCL.so /usr/lib/libOpenCL.so.1.2

I use this myself. Will add the binaries later.

AMD

sudo ln -s /usr/lib64/OpenCL/vendors/amd/libOpenCL.so /usr/lib64/libOpenCL.so.1.2
sudo ln -s /usr/lib/OpenCL/vendors/amd/libOpenCL.so /usr/lib/libOpenCL.so.1.2

NVIDIA

Strongly discouraged to use this libOpenCL-library!

sudo ln -s /usr/local/cuda/lib64/libOpenCL.so /usr/lib64/libOpenCL.so.1.1
sudo ln -s /usr/local/cuda/lib/libOpenCL.so /usr/lib/libOpenCL.so.1.1

Intel

sudo ln -s /usr/lib64/OpenCL/vendors/intel/libOpenCL.so /usr/lib64/libOpenCL.so.1.2
sudo ln -s /usr/lib/OpenCL/vendors/intel/libOpenCL.so /usr/lib/libOpenCL.so.1.2

Then we add libOpenCL.so.1, libOpenCL.so.1.0 and libOpenCL.so:

sudo ln -s /usr/lib64/libOpenCL.so.1.2 /usr/lib64/libOpenCL.so.1
sudo ln -s /usr/lib64/libOpenCL.so.1 /usr/lib64/libOpenCL.so
sudo ln -s /usr/lib/libOpenCL.so.1.2 /usr/lib/libOpenCL.so.1
sudo ln -s /usr/lib/libOpenCL.so.1 /usr/lib/libOpenCL.so

As libOpenCL.so.1.2 is/should be backwards compatible with libOpenCL.so.1.0 and libOpenCL.so.1.1, you can choose to make those symbolic links too. Only do this when you have wrongly linked software - link to libOpenCL.so.1 in your own software.

Be sure to link to libOpenCL.so.1.1 if you chose to use NVidia's library.

Installable Client Drivers

important: If you have chosen to leave the files in the original locations and skipped most of this tutorial, be sure to put the whole path and not only the filename in the icd-files.

If you list the platforms available, you actually list the ICDs. If you have written your own compiler, then you can easily add it without interfering with others. Like you can access an Intel CPU via both the AMD-ICD and Intel-ICD.

In /etc/OpenCL/vendor/ all ICDs need to be put. You'll find them already here, or you have to create them. This is how they are provided now, but I omitted the library-location (which was in nvidia.icd), since it still gives errors if the ldconfig-steps where not done correctly.
[raw]
sudo echo "libatiocl64.so" > /etc/OpenCL/vendors/atiocl64.icd
sudo echo "libatiocl32.so" > /etc/OpenCL/vendors/atiocl32.icd

sudo echo "libintelocl.so" > /etc/OpenCL/vendors/intelocl.icd

sudo echo "libcuda.so" > /etc/OpenCL/vendors/nvidia.icd
[/raw]
You can pick any name for the icd-files. AMD might replace 'ati' by 'amd' in their libraries, so if it stops working when updating, you know where to look.

Back to programming

When compiling a C or C++ program, you can keep your makefile simple. When the PPA with all this gets available, I'll let you know via Twitter, Facebook. Tweet or like, if you support this blog!