DOL开发环境配置

一、DOL框架描述

The distributed operation layer (DOL) is a framework that enables the (semi-) automatic mapping of applications onto the multiprocessor SHAPES architecture platform. The DOL consists of basically three parts:

- DOL Application Programming Interface: The DOL defines a set of computation and communication routines that enable the programming of distributed, parallel applications for the SHAPES platform. Using these routines, application programmers can write programs without having detailed knowledge about the underlying architecture. In fact, these routines are subject to further refinement in the hardware dependent software (HdS) layer.
- DOL Functional Simulation: To provide programmers a possibility to test their applications, a functional simulation framework has been developed. Besides functional verification of applications, this framework is used to obtain performance parameters at the application level.
- DOL Mapping Optimization: The goal of the DOL mapping optimization is to compute a set of optimal mappings of an application onto the SHAPES architecture platform. In a first step, XML based specification formats have been defined that allow to describe the application and the architecture at an abstract level. Still, all the information necessary to obtain accurate performance estimates is contained.

二、DOL安装笔记

1. Ubuntu 安装必要的环境

- `$ sudo apt-get update`
- `$ sudo apt-get install ant`
- `$ sudo apt-get install openjdk-7-jdk(or openjdk-8-jdk)`
- `$ sudo apt-get install upzip`

2. 下载必要的文件

- 下载systemc-2.3.1.tgz文件 `$ sudo wget http://www.accellera.org/images/downloads/standards/systemc/systemc-2.3.1.tgz`
- 下载dol_ethz.zip文件 `sudo wget http://www.tik.ee.ethz.ch/~shapes/downloads/dol_ethz.zip`

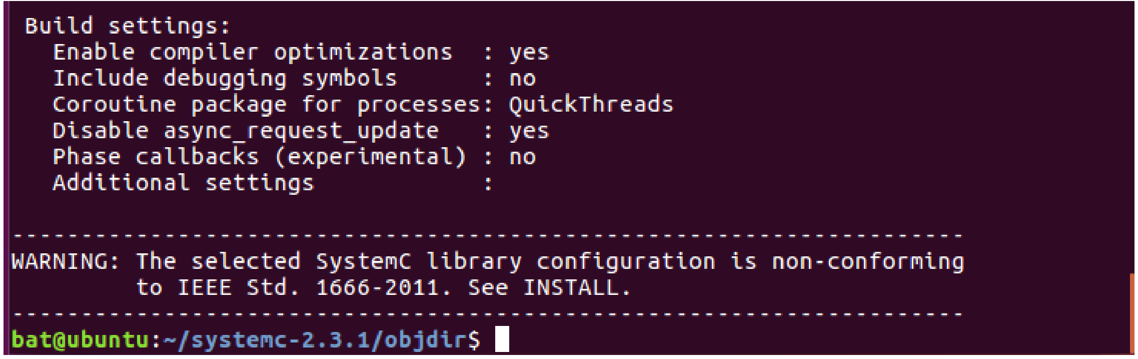
3. 解压文件

- 新建dol的文件夹
  `$ mkdir dol`
- 将dolethz.zip解压到dol文件夹中 `$ unzip dol_ethz.zip -d dol`
- 解压systemc文件
  `$ tar -zxvf systemc-2.3.1.tgz`

4. 编译Systemc

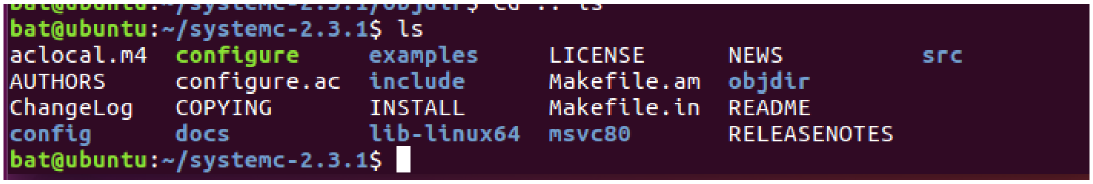
- 进入systemc-2.3.1
  `$ cd systemc-2.3.1`
- 新建临时文件夹
  `$ mkdir objdir`
- 进入临时文件夹
  `$ cd objdir`

- 运行configure
  ```
  $ ../configure CXX=g++ --disable-async-updates
  ```



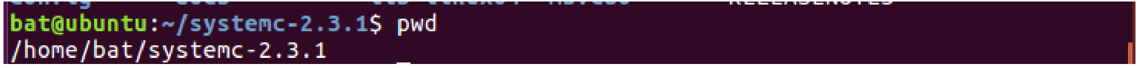
```
 Build settings:
   Enable compiler optimizations  : yes
   Include debugging symbols      : no
   Coroutine package for processes: QuickThreads
   Disable async_request_update   : yes
   Phase callbacks (experimental) : no
   Additional settings            :

---------------------------------------------------------------------
WARNING: The selected SystemC library configuration is non-conforming
         to IEEE Std. 1666-2011. See INSTALL.
---------------------------------------------------------------------
bat@ubuntu:~/systemc-2.3.1/objdir$ 
```

- 编译
  ```
  $ sudo make install
  ```



```
bat@ubuntu:~/systemc-2.3.1$ ls
aclocal.m4   configure      examples     LICENSE      NEWS           src
AUTHORS      configure.ac   include      Makefile.am  objdir
ChangeLog    COPYING        INSTALL      Makefile.in  README
config       docs           lib-linux64  msvc80       RELEASENOTES
bat@ubuntu:~/systemc-2.3.1$ 
```

- 记录当前的工作路径
  ```
  $ pwd
  ```

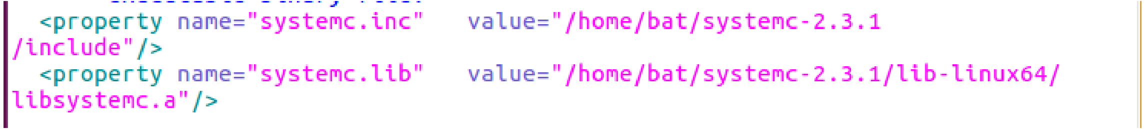


```
bat@ubuntu:~/systemc-2.3.1$ pwd
/home/bat/systemc-2.3.1
```

5. 编译Dol

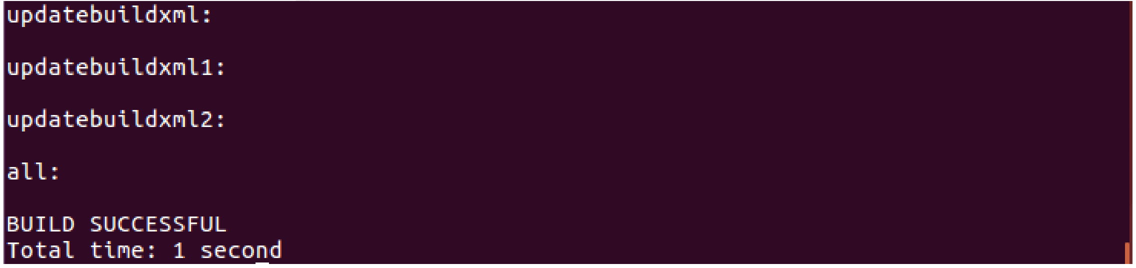
- 进入dol文件夹
  ```
  $ cd../dol
  ```

- 修改build_zip.xml文件[pwd是前面储存的工作路径]
  ```
  <property name="systemc.inc" value="pwd/include"/> <property name="systemc.lib" value="pwd/lib-linux64/libsystemc.a"/>
  ```



```
  <property name="systemc.inc"   value="/home/bat/systemc-2.3.1
/include"/>
  <property name="systemc.lib"   value="/home/bat/systemc-2.3.1/lib-linux64/
libsystemc.a"/>
```

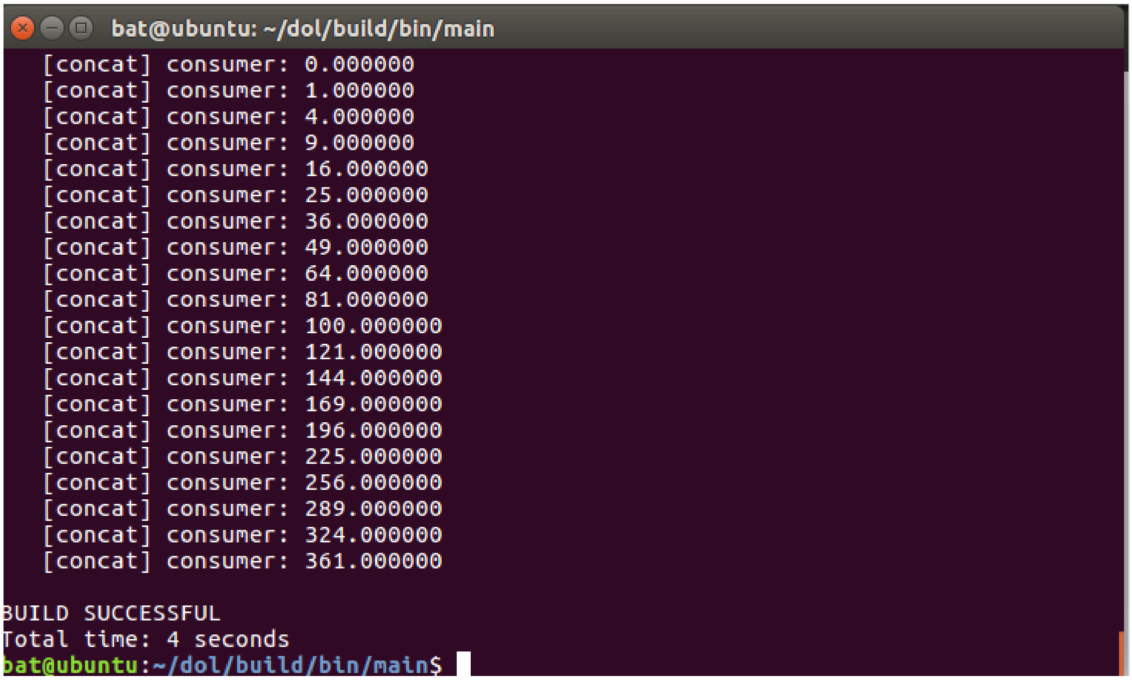
- 编译
  ```
  $ sudo ant -f build zip.xml all
  ```



```
updatebuildxml:

updatebuildxml1:

updatebuildxml2:

all:

BUILD SUCCESSFUL
Total time: 1 second
```

- 进入build/bin/main
  ```
  $ cd build/bin/main
  ```

- 运行第一个例子
  ```
  $ sudo ant -f runexample.xml -Dnumber=1
  ```



```
   [concat] consumer: 0.000000
   [concat] consumer: 1.000000
   [concat] consumer: 4.000000
   [concat] consumer: 9.000000
   [concat] consumer: 16.000000
   [concat] consumer: 25.000000
   [concat] consumer: 36.000000
   [concat] consumer: 49.000000
   [concat] consumer: 64.000000
   [concat] consumer: 81.000000
   [concat] consumer: 100.000000
   [concat] consumer: 121.000000
   [concat] consumer: 144.000000
   [concat] consumer: 169.000000
   [concat] consumer: 196.000000
   [concat] consumer: 225.000000
   [concat] consumer: 256.000000
   [concat] consumer: 289.000000
   [concat] consumer: 324.000000
   [concat] consumer: 361.000000

BUILD SUCCESSFUL
Total time: 4 seconds
bat@ubuntu:~/dol/build/bin/main$ 
```

三、实验感想

通过本次实验完成了在Ubuntu下配置DOL框架，然后使用Markdown语言写本次配置的文档。使用的软件是MarkdownPad2，然后使用的图床是使用国内的七牛云在线储存。然后学习了将文档储存到github上进行版本管理。