Lab3 dol实例分析与编程

example1

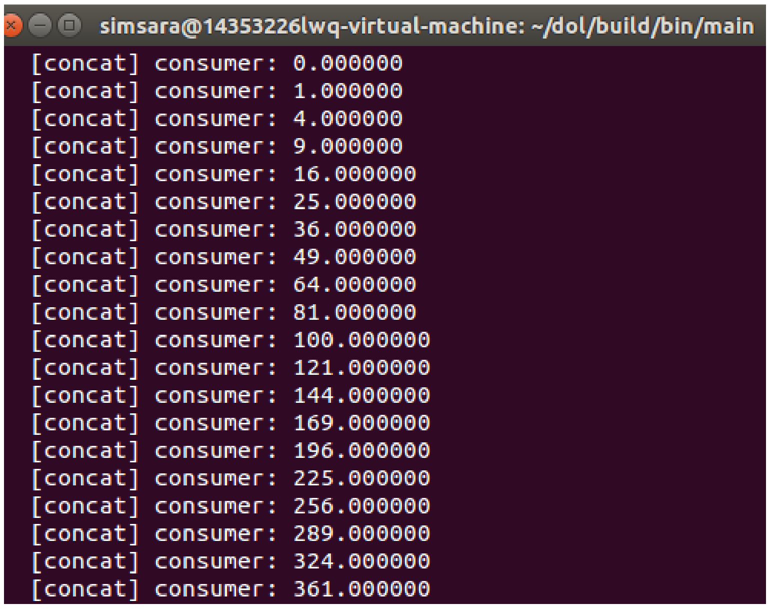
- 要求：修改example1，使其输出3次方数，tips:修改square.c
- 实验步骤：

1. 修改square.c，将“i * i”改成“i * i * i”:

```
#include <stdio.h>

#include "square.h"

void square_init(DOLProcess *p) {
    p->local->index = 0;
    p->local->len = LENGTH;
}

int square_fire(DOLProcess *p) {
    float i;

    if (p->local->index < p->local->len) {
        DOL_read((void*)PORT_IN, &i, sizeof(float), p);
        i = i*i*i;
        DOL_write((void*)PORT_OUT, &i, sizeof(float), p);
        p->local->index++;
    }

    if (p->local->index >= p->local->len) {
        DOL_detach(p);
        return -1;
    }

    return 0;
}
```

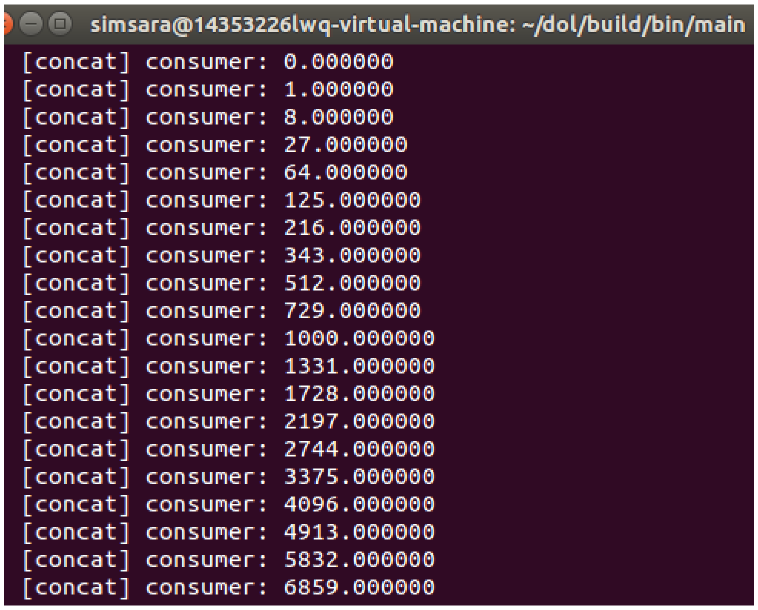
1. 运行：

   - 修改前：



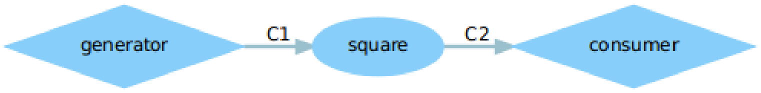
```
simsara@14353226lwq-virtual-machine: ~/dol/build/bin/main
[concat] consumer: 0.000000
[concat] consumer: 1.000000
[concat] consumer: 4.000000
[concat] consumer: 9.000000
[concat] consumer: 16.000000
[concat] consumer: 25.000000
[concat] consumer: 36.000000
[concat] consumer: 49.000000
[concat] consumer: 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
```

   - 修改后：



```
[concat] consumer: 0.000000
[concat] consumer: 1.000000
[concat] consumer: 8.000000
[concat] consumer: 27.000000
[concat] consumer: 64.000000
[concat] consumer: 125.000000
[concat] consumer: 216.000000
[concat] consumer: 343.000000
[concat] consumer: 512.000000
[concat] consumer: 729.000000
[concat] consumer: 1000.000000
[concat] consumer: 1331.000000
[concat] consumer: 1728.000000
[concat] consumer: 2197.000000
[concat] consumer: 2744.000000
[concat] consumer: 3375.000000
[concat] consumer: 4096.000000
[concat] consumer: 4913.000000
[concat] consumer: 5832.000000
[concat] consumer: 6859.000000
```

- dot截图：



example2

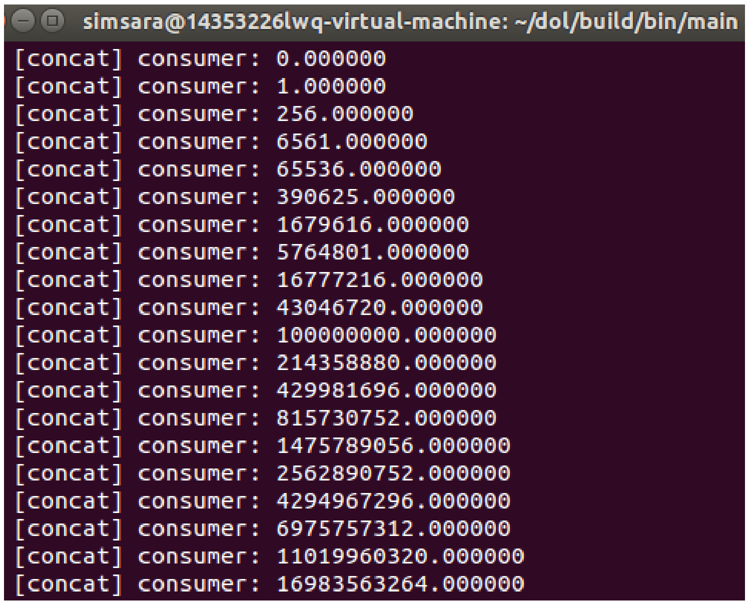
- 要求：修改example2，让3个square模块变成2个，tips:修改xml的iterator
- 实验步骤：

1. 修改example2.xml，将value从原来的“2”改成“3”：

```
<?xml version="1.0" encoding="UTF-8"?>
<processnetwork xmlns="http://www.tik.ee.ethz.ch/~shapes/schema/PROCESSNETWORK"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tik.ee.ethz.ch/~shapes/schema/PROCESSNETWORK
http://www.tik.ee.ethz.ch/~shapes/schema/processnetwork.xsd" name="example2">

  <variable value="2" name="N"/>

  <!-- instantiate resources -->
  <process name="generator">
    <port type="output" name="10"/>
    <source type="c" location="generator.c"/>
  </process>

  <iterator variable="i" range="N">
    <process name="square">
      <append function="i"/>
      <port type="input" name="0"/>
      <port type="output" name="1"/>
      <source type="c" location="square.c"/>
    </process>
  </iterator>
```

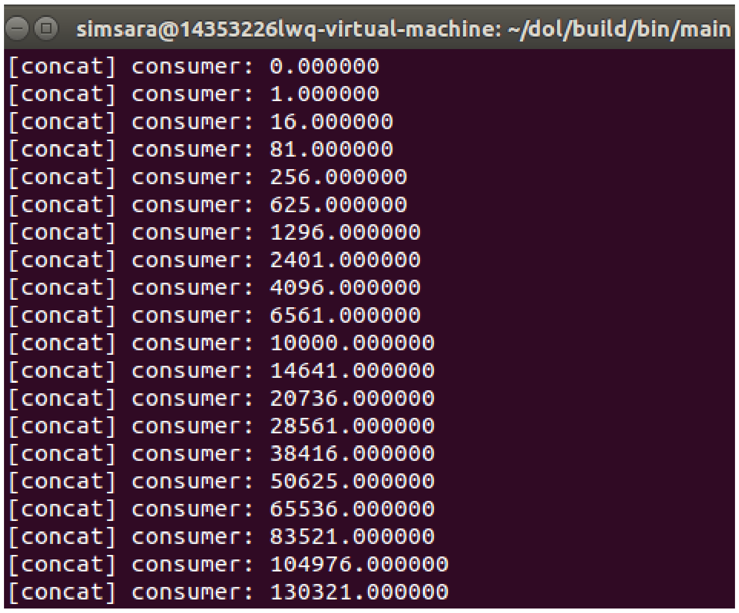
1. 运行：

   - 修改前（3个square模块，8次方）：



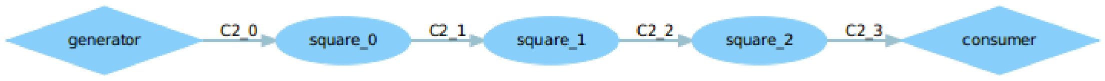
```
simsara@14353226lwq-virtual-machine: ~/dol/build/bin/main
[concat] consumer: 0.000000
[concat] consumer: 1.000000
[concat] consumer: 256.000000
[concat] consumer: 6561.000000
[concat] consumer: 65536.000000
[concat] consumer: 390625.000000
[concat] consumer: 1679616.000000
[concat] consumer: 5764801.000000
[concat] consumer: 16777216.000000
[concat] consumer: 43046720.000000
[concat] consumer: 100000000.000000
[concat] consumer: 214358880.000000
[concat] consumer: 429981696.000000
[concat] consumer: 815730752.000000
[concat] consumer: 1475789056.000000
[concat] consumer: 2562890752.000000
[concat] consumer: 4294967296.000000
[concat] consumer: 6975757312.000000
[concat] consumer: 11019960320.000000
[concat] consumer: 16983563264.000000
```

- 修改后（2个square模块，4次方）：

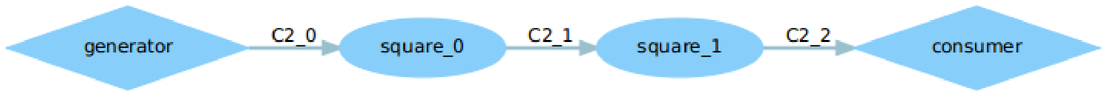


```
simsara@14353226lwq-virtual-machine: ~/dol/build/bin/main
[concat] consumer: 0.000000
[concat] consumer: 1.000000
[concat] consumer: 16.000000
[concat] consumer: 81.000000
[concat] consumer: 256.000000
[concat] consumer: 625.000000
[concat] consumer: 1296.000000
[concat] consumer: 2401.000000
[concat] consumer: 4096.000000
[concat] consumer: 6561.000000
[concat] consumer: 10000.000000
[concat] consumer: 14641.000000
[concat] consumer: 20736.000000
[concat] consumer: 28561.000000
[concat] consumer: 38416.000000
[concat] consumer: 50625.000000
[concat] consumer: 65536.000000
[concat] consumer: 83521.000000
[concat] consumer: 104976.000000
[concat] consumer: 130321.000000
```

- dot截图：
  - 修改前(3个square模块)：



  - 修改后(2个square模块)：



实验感想

- 在修改完代码之后，重新运行编译之前，一定要删除整个之前已经build好的文件夹，不然运行出来的结果会是之前的结果；

- 重新编译的具体命令行是：
  1. cd col
  2. sudo ant -f build_zip.xml all
  3. cd build/bin/main
  4. 一定要在这里把之前编译好的example文件夹给删掉！
  5. sudo ant -f runexample.xml -Dnumber=X(X为运行的example数)
- 实验具体要做的操作还是不难的，第一个就是修改square.c文件里面的"i * i"为"i * i * i"，第二个就是修改xml文件里面的value值，从3改为2；
- xml文件里面的value值就是调用模块几次，一个平方的模块被调用3次，就是计算8次方。