Gem5 文档

## 1.编译：

clone两个库(gem5-pard/PARDg5-V)，设置PARDg5-V Makefile内的路径

（1）gem5-pard文件编译

**Command**：**scons build/X86/gem5.opt build/X86/gem5.debug**

（2）pardg5-v文件编译

修改Makefile文件，确保相应路径名的正确性。

**Command：make all**

（3）镜像文件

**目录**：pardg5-v/system/

为了缩减文件大小，部分文件需要去网站下载，未压缩到本文档中，这里给出下载地址。

X86root.img：http://www.m5sim.org/dist/current/x86/x86-system.tar.bz2

解压文件后，将disks下linux-x86.img拷贝到/system/disks/X86root.img

## 2.启动命令：

**Makefile文件在configs/RunConfig.mk中**

**其中 --guest=[0-4]表示在启动gem5时自动启动Dom数**

**运行**：**make run EXTRAS="--caches --l2cache --l2\_assoc=16 --l2\_size=2MB" CPU\_TYPE=atomic**

**Tip**：

* 3种CPU类型：timing，atomic，detailed
  + Atomic：最简单规模的模型，一个cycle完成一条指令的执行，memory 模型比较理想化，访存操作为原子性操作。适用于快速功能模拟。
  + Timing：无流水线的模拟，但是使用了存储器访问时序模型，用以统计存储器访问延迟。
  + Detailed：O3模拟器模拟了乱序执行和超标量执行的指令间依赖，以及运行在多CPU上的并发执行的多线程。默认7级流水：取值、译码、重命名、发射、执行、写回、提交。模拟了物理寄存器文件、IO、LSQ、ROB功能部件池等。主要参数为流水管道间延迟、硬件线程数、IQ/LSQ/ROB项数、FU延迟、物理寄存器重命名、分支预测、访存依赖预测等。
* 必须记得加 –caches –l2cache，不然默认make run为uncached模式

运行后的界面为gem5执行界面。并会打印如下信息：

Listening for com\_1 connection on port 3456 **// PRM端口号**

Listening for tap connection on port 3500

… ….

Listening for com\_4 connection on port 3457 **//Dom4端口号**

Listening for com\_3 connection on port 3458 **//Dom3端口号**

Listening for com\_2 connection on port 3459 **//Dom2端口号**

Listening for com\_1 connection on port 3460 **//Dom1端口号**

## 3.PRM测试：

**登陆PRM**：**m5term localhost 3456**

**Tip – m5term install**：http://gem5.org/M5term

**CP模块加载**：**insmod /lib/modules/2.6.28.4-gc395911-dirty/cpa.ko**

**查看CP属性**：**cat sys/cpa/cp[0-4]/ident**

cp0：L2TAG\_CP

cp1：MEMBUS\_CP

cp2：MEMCTRL\_CP

cp3：MEMCTRL\_CP

cp4：IOBRG\_CP

**三张表的信息**：

sys/cpa/cp[0-4]/ldoms/ldom[0-3]/[parameters | statistics | triggers]/

通过cat 查看，echo修改

## 4.启动其它LDOM:

### 方法1 ---- Gem5 内部控制

**gem5窗口Ctrl+C中断，进入命令行**

**查看command**：**help**

**创建VM**：**create** // 每启动一个dom前都需要create

**启动VM**：**startup [0-3]**

**登陆VM**：gem5外通过 **m5term localhost 3460** //第二节显示的端口号

### 方法2 ---- Gem5 外部控制

**脚本文件：utils/command.sh**

**创建VM**：**utils/command.sh create**

**启动VM**：**utils/command.sh startup [0-3]**

**登陆VM**：**m5term localhost 3460** //第二节显示的端口号

## 5.监控网页/状态：

<http://172.18.11.209/dashboard/#HW-Demo>

### （1）设置checkpoint

按照2中描述启动PRM，并在PRM中输入：**m5 checkpoint**

### （2）恢复数据

退出gem5，然后重新启动，command如下：

**make run EXTRAS="--caches --l2cache --l2\_assoc=16 --l2\_size=2MB --dumpreset-stats=100000,100000 -r 1" CPU\_TYPE=detailed**

### （3）启动VM

按照3的方法登陆PRM，然后再通过4的任意一种方法启动VM。

结果可以在两个地方看到。

1. 在m5out/stats.txt文件中查看。
2. 在本机地址2003端口查看。

方法b实现文件路径：gem5-pard/src/base/stats/[carbon.hh | .cc]

其中carbon.cc文件中，

noOutput函数进行正则表达式匹配，筛选出需要的数据；

Generate函数会将结果发送到目标sockfd中。

正则表达式的代码在：configs/RunConfig.mk的--carbon-regex中设置

Some corrections:

1. src/base/stats/carbon.cc: 191

write(……) -> if(write(……)) {};

1. src/sim/console.cc: 12

system(“clear”) -> if(system(“clear”)){};

1. hyper/arch/PARDg5VLDomain.cc: 105

std::string kernelName = “…..”

1. configs/common/FSConfig.py:133

disk1.childImage(disk('cpu2006.img')) ->disk1.childImage(disk('x86root.img'))

kernel and disks:

host kernel: system/binaries/linux-2.6.28-GM/vmlinux

host disk: system/disks/PARD5GM.img

guest kernel: system/binaries/x86\_64-vmlinux-2.6.28.4.smp

guest disk: system/disks/x86root.img

./command.sh shutdown [0-3] has problems?