

冯凯 2023 年 11 月 27 日

1 项目介绍

在一期工程中，我们引导用户输入参数并仿真得到了微处理器的性能数据。接下来，我们还需要完成以下工作：

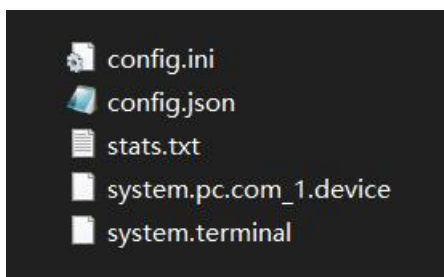
1. 处理性能数据，使它可以被功耗仿真模块调用
2. 调用功耗仿真模块，将处理过的性能数据转为功耗数据
3. 处理功耗数据，使它可以被温度仿真模块调用
4. 调用温度仿真模块，将处理过的功耗数据转为温度数据
5. 根据温度数据，画出温度图

接下来，我对上述步骤做解释。这些步骤都可以调用项目文件中的脚本完成（见第 2 节）。黄同学不需要知道怎么实现这些步骤，只需要知道这些步骤的意义以及如何调用项目文件。

1.1 处理性能数据

如果之前的工作顺利，那么每一个基准都会生成 5 个文件（下图）

- **stats.txt**：记录了工作信息；
- **config.json** 文件：记录了仿真处理器的各部分硬件信息；
- **config.ini**：与 **config.json** 内容一致；
- **system.pc.com_1.device**：记录了仿真的各个端口信息；
- **system.terminal**：记录了终端输出信息。



其中，我们最要关注的是 **stats.txt** 与 **config.json**。如果仿真正常的话，**stats.txt** 都会有四段，其中第三段信息为 ROI 区，是处理器仿真的重点关注信息，也最能代表每一个基准的特征。

但是，功耗仿真模块不能直接读取 `stats.txt` 与 `config.json`。我们需要对其进行处理，具体为：

- 对于仿真不正常的 `stats.txt`（段数小于四段），舍弃；
- 对于仿真正常的 `stats.txt`（段数为四段），分离出 ROI 区的数据，并按照一定格式保存为另一个 `txt` 文件；
- 读取 `config.json` 文件，按照芯片的布局图重写，并按照一定格式保存为另一个 `xml` 文件。

1.2 功耗仿真模块

我们调用 McPAT 功耗仿真模块，输入 1.1 节中的 `xml` 文件，输出包含了芯片功耗数据的 `txt` 文件。

1.3 处理功耗数据

温度仿真模块不能直接读取 1.2 中的功耗数据。我们需要对其进行处理，具体为：

- 读取功耗数据 `txt` 文件，按照芯片版图重写，并按照一定格式保存为另一个 `ptrace` 文件。

1.4 温度仿真模块

我们调用 HotSpot 温度仿真模块，输入 1.3 节中的 `ptrace` 文件，输出包含了芯片温度信息的 `steady` 文件。

如果仿真正常，`steady` 文件的内容会有四段：`layer 0`、`layer 1`、`layer 2` 和 `layer 3`。它们的含义为：

- **layer 0:** 芯片层（`chip specs`），这代表芯片本身的温度分布情况。包括芯片中的各个功能块（如 CPU 核心，缓存，模拟电路等）的温度映射。这是热传导的源头，能直接反映芯片在工作时的热点分布。
- **layer 1:** 散热器层（`heat sink specs`），该层表示芯片背面的温度分布，是贴合在散热器上的芯片与散热器间界面的温度。它体现了从芯片到散热器的热传播情况，是连接芯片层和散热器层的重要桥梁。

- **layer 2:** 散热片层 (heat spreader specs)，即散热器本身的温度分布。可以是散热片，也可以是整个散热装置。它显示了从芯片到散热器的热量在散热器内部是如何传导和扩散的。
 - **layer 3:** 接口材料层 (interface material specs)，最外面的一层，表示接触到环境空气/液冷的散热器外表面上的温度分布情况。它与周围环境的温度交换，反映了散热效率的好坏。
- 一般来说，我们对 **layer 0** 感兴趣。

1.5 温度图

根据 1.4 节中的 **steady** 文件，提取出 **layer 0** 的数据，画出温度图 **png** 文件。

2 项目文件说明

将项目文件解压，可以看到以下文件夹。



本项目只需要用到编号为 1-6 的文件夹。接下来，我对这些文件夹的使用方法做解释。

2.1 gem5 输入-gem5 输出

打开文件夹“**1 gem5 输入-gem5 输出**”，可以看到文件夹“**样例: gem5 输出**”。该文件夹下有每个基准程序仿真得到的性能数据。

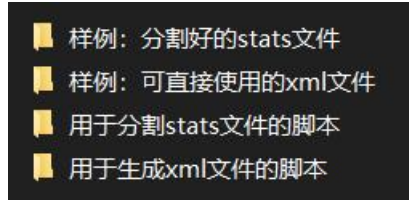
该文件的用途主要有两个：

- 作为 **gem5** 输出的性能数据的样例，供参考
- 在 **gem5** 无法运行的情况下，可暂时把样例中的文件当作 **gem5** 的输出，

不耽误后续工程

2.2 gem5 输出-McPAT 输入

打开文件夹“2 gem5 输出-McPAT 输入”，可以看到以下文件夹。



打开“用于分割 stats 文件的脚本”，可以看到 `split.py` 脚本。该脚本的功能是判断 `stats.txt` 文件是否仿真成功，如果不成功就打印 `false`，如果成功就打印 `true` 并提取出其中的四段信息，依次保存为 `1.txt`、`2.txt`、`3.txt`、`4.txt`。

具体使用方法如下：

`split.py {stats.txt 文件路径} {1.txt、2.txt、3.txt、4.txt 文件所在文件夹的路径}`

例如，如果我的 `stats.txt` 文件路径是 `./bodytrack/stats.txt`，我想直接把 `1.txt`、`2.txt`、`3.txt`、`4.txt` 保存在 `./output` 文件夹下，那么我可以命令：

`split.py ./bodytrack/stats.txt ./output`

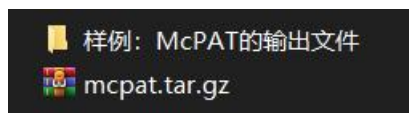
打开“用于生成 xml 文件的脚本”，可以看到 `McPATgenerator.py` 脚本。该脚本的功能是读取 `config.json` 文件，按照芯片的布局图重写，并按照一定格式保存为另一个 `xml` 文件。但是，由于目前支持的基准程序很少，我已经事先制作好每个基准程序的 `xml` 文件了，所以你暂不需要使用它。

“样例：分割好的 stats 文件”中有每个基准程序性能数据的 4 段 `txt` 文件，供参考。

“样例：可直接使用的 xml 文件”中有每个基准程序对应的 `xml` 文件。在后续的步骤中会用到，请知悉。

2.3 McPAT 输入-McPAT 输出

打开文件夹“3 McPAT 输入-McPAT 输出”，可以看到以下文件。



mcpat.tar.gz 是微处理器的功耗仿真模块。将其放到 experiment 目录下，解压即可。

“样例：McPAT 的输出文件”中有每个基准程序的 4 段性能数据输入 McPAT 后得到的功耗数据，供参考。请注意甄别上一节中 1.txt 和本节中 1.txt 的区别，之前的 1.txt 指的是性能数据的第 1 段，本节的 1.txt 指的是将性能数据第 1 段输入 McPAT 后得到的功耗数据。它们的文件名相同，但文件内容不同。

2.4 McPAT 输出-HotSpot 输入

打开文件夹“4 McPAT 输出-HotSpot 输入”，可以看到以下文件夹。



“用于生成 ptrace 文件的脚本”中包含了 writeptrace.py 脚本和 ev6.flp 文件。writeptrace.py 的功能是读取功耗数据 txt 文件，按照芯片版图重写，并按照一定格式保存为另一个 ptrace 文件。ev6.flp 是包含芯片版图信息的文件。它们的使用方法是：

writeptrace.py {flp 文件路径} {txt 文件路径，该文件为 McPAT 输出文件}
{ptrace 文件路径，该文件为要生成的 hotspot 输入文件}

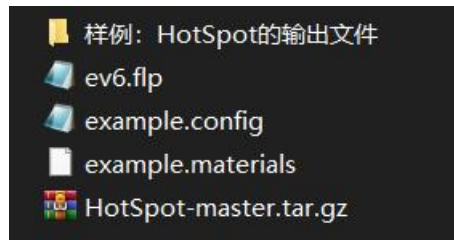
例如，如果我的 ev6.flp 就在当前文件夹下，McPAT 输出的 3.txt 在 ./bodytrack 文件夹中，我 ./output 文件夹中生成 3.ptrace，那么我的命令为：

```
writeptrace.py ./ev6.flp ./bodytrack/3.txt ./output/3.ptrace
```

“样例：HotSpot 输入”中有一些预先生成好的 ptrace 文件，供参考。

2.5 HotSpot 输入-HotSpot 输出

打开文件夹“5 HotSpot 输入-HotSpot 输出”，可以看到以下文件。



HotSpot-master.tar.gz 是微处理器温度仿真模块。将其放到 experiment 目录下，解压即可。

ev6.flp 是包含了芯片版图信息的文件，和上一节中的 ev6.flp 相同。

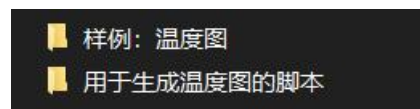
example.config 为全局配置文件，包含了边界条件，比热等物理信息，以及仿真的精度。

example.materials 为包含了芯片材料信息的文件。

“样例: HotSpot 的输出文件”中有一些预先生成好的温度数据 steady 文件，供参考。

2.6 HotSpot 输出-温度图

打开文件夹“6 Hotspot 输出-温度图”，可以看到以下文件。



“用于生成温度图的脚本”中，包含了 flpdraw.py 脚本。该脚本的功能是根据 steady 文件，提取出 layer 0 的数据，画出温度图 png 文件。它的使用方法是：

flpdraw.py {.flp 文件的路径} {.steady 文件的路径} {要画第几层} {热图的名字} {要输出的温度图的路径}

其中，.flp 文件指的是 ev6.flp，在之前小节中已经解释过。.steady 文件指的是 HotSpot 输出的温度数据.steady 文件。要画的层数可以填 0、1、2、3 中的一个数字。热图的名字指的是写在热图上的标题。

例如，我的 ev6.flp 在当前目录下。我想要使用的 bodytrack.grid.steady 文件在 ./bodytrack 目录下。我想要画出芯片的 layer 0。热图的标题取《bodytrack_0》。热图的文件名为 bodytrack_map，放到当前目录下。那么，我的命令为：

```
flpdraw.py      ./ev6.flp      ./bodytrack/bodytrack.grid.steady      0
bodytrack_0 ./bodytrack_map
```

3 向导软件制作

以下步骤自动完成，不需要引导用户手动点击确认。

在一期工程中，我们在 `gem5_output` 文件夹下创建了{基准程序名}文件夹，在每个子文件夹内有对应基准程序的性能数据。

3.1 处理性能数据

在实验目录下创建 `McPAT_input` 文件夹，如果未创建就创建一个，如果已创建就清空其内容。

调用项目文件“2 gem5 输出-McPAT 输入”内的 `split.py`，检测 `gem5_output` 中每一个性能数据的合法性。

- 如果结果是 `true`，则在 `McPAT_input` 中创建{基准程序名}文件夹，切割性能数据，将切割后的 `1.txt`、`2.txt`、`3.txt`、`4.txt` 保存到该文件夹下。
- 结果是 `false`，则删除该性能数据对应的基准程序和所在文件夹，并通知用户这个基准程序的性能仿真结果非法。

将“样例：可直接使用的 xml 文件”中的 xml 文件复制到 `McPAT_input` 下的{基准程序名}文件夹中。

3.2 调用功耗仿真模块

在实验目录下创建 `McPAT_output` 文件夹，如果未创建就创建一个，如果已创建就清空其内容。

进入 `mcpat` 文件夹。该文件夹下有 `mcpat` 功耗仿真模块，使用方法为：

```
./mcpat-infile {xml 文件路径} -print_level {等级} > {功耗数据文件路径}
```

其中，xml 文件指的是 `McPAT_input` 下{基准程序名}文件夹中的 xml 文件。等级取 5。功耗数据文件路径指的是 `McPAT` 的输出文件的路径。

例如，我的 xml 文件保存在 `McPAT_input/Blacksholes` 文件夹下，文件名为 `DerivO3CPU_Blackscholes_3.xml`。我希望将 `McPAT` 的输出文件保存在当前路径下，文件名为 `3.txt`，那么我的命令为：

```
./mcpat -infile ../McPAT_input/Blacksholes/DerivO3CPU_Blackscholes_3.xml  
-print_level 5 > ./3.txt
```

接下来，调用 McPAT，对 McPAT_input/{基准程序名}文件夹下的第三段 xml 文件进行功耗仿真。第三段 xml 文件形如 “DerivO3CPU_Blackscholes_3.xml”，带有下列划线 3 的后缀。将仿真结果命名为 3.txt，存放到 McPAT_output/{基准程序名}文件夹中。

3.3 处理功耗数据

在实验目录下创建 HotSpot_input 文件夹，如果未创建就创建一个，如果已创建就清空其内容。

调用项目文件 “4 McPAT 输出-HotSpot 输入” 中的 writeptrace.py，对 McPAT_output/{基准程序名}文件夹中的 3.txt 进行处理，输出{基准程序名}.ptrace，保存在 HotSpot_input/{基准程序名}文件夹下。

3.4 调用温度仿真模块

在实验目录下创建 HotSpot_output 文件夹，如果未创建就创建一个，如果已创建就清空其内容。

进入 HotSpot-master 文件夹。该文件夹下有 HotSpot 功耗仿真模块，使用方法为：

```
./hotspot \
-c {example.config 的路径，项目文件提供} \
-f {ev6.flp 的路径，项目文件提供} \
-p {ptrace 文件的路径，上一步骤获取} \
-materials_file {example.materials 的路径，项目文件提供} \
-o {ttrace 文件，就填{基准程序名}.ttrace} \
-steady_file {就填{基准程序名}.steady} \
-model_type {仿真模式，就填 grid} \
-grid_steady_file {.grid.steady 文件，就填{基准程序名}.grid.steady}
```

例如，如果我的 example.config、ev6.flp、example.materials 都在上一级目录下，我想要仿真 blackscholes.ptrace，那么我可以命令：

```
../hotspot \
-c ../example.config \
```



```
-f ../ev6.flp \  
-p blackscholes.ptrace \  
-materials_file ../example.materials \  
-o blackscholes.ttrace \  
-steady_file blackscholes.steady \  
-model_type grid \  
-grid_steady_file blackscholes.grid.steady
```

接下来，调用 HotSpot，对 HotSpot_input/{基准程序名}文件夹下的 ptrace 文件进行温度仿真。将仿真结果保存到 McPAT_output/{基准程序名}文件夹中。

3.5 画温度图

在实验目录下创建 HeatMap 文件夹，如果未创建就创建一个，如果已创建就清空其内容。

调用项目文件“6 Hotspot 输出-温度图”中的 flpdraw.py，对 HotSpot_output 中的每一个 steady 文件进行绘制。绘制 layer 0。温度图的标题为{基准程序名}。温度图的文件名为{基准程序名}。温度图保存到 HeatMap 文件夹中。

4 其他

本文为微处理器热特性仿真的向导软件二期工程提供了一个方案。方案目前仅包括功耗仿真、温度仿真与绘制温度图等步骤。我们先将这一部分做好，然后视情况安排后续工作。