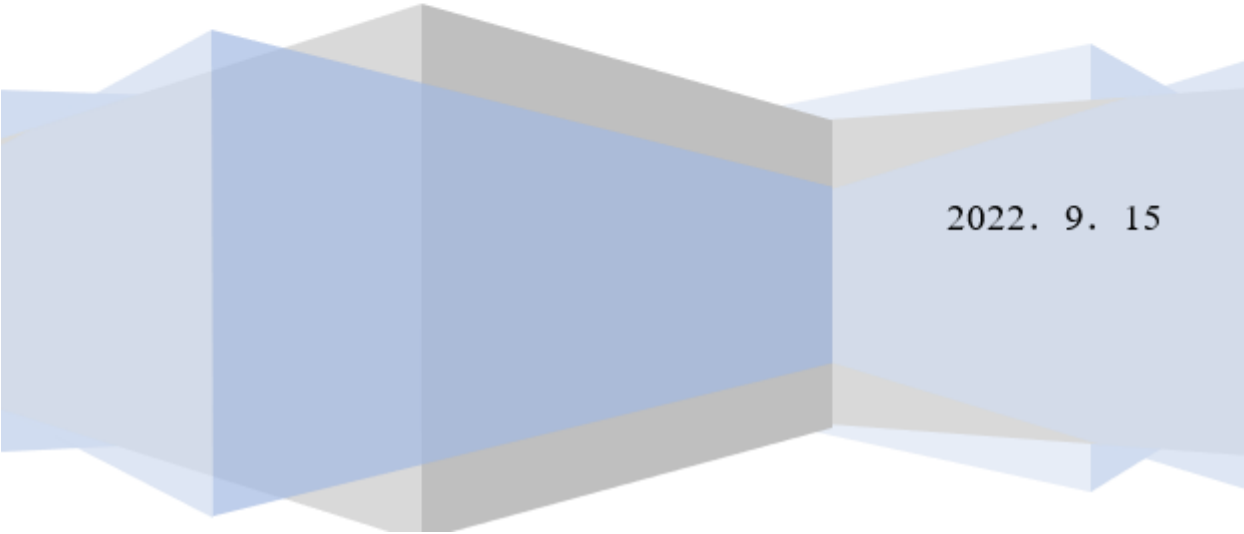


Mica_Profile_Manual

Version 1.2

Peng Zhan



2022. 9. 15

修订记录

日期	版本	描述	作者	审查
2022/06/13	v1.0	文档初稿	彭展	凌明
2022/08/09	v1.1	更改了代码保存编码格式	彭展	凌明
2022/09/15	v1.2	将ROB_SIZE作为输入参数, 并修正了关键路径长度分布的错误	彭展	凌明

Mica_Profile_Manual v1.2

一、运行环境

本项目使用Microsoft Visual Studio 2022编译，操作系统为windows 10，编程语言为C++。

二、Mica_Profile使用方法

1, 在<mica_profile_manual>文件夹下，打开Windows PowerShell

2,输入以下命令：

```
./mica_profile 参数1 参数2 参数3 参数4 参数5 参数6
```

其中：

参数1为输入文件的绝对路径,如:

C:\mica_profile_manual\mica_profile\instrace.new.log

参数2为输出文件夹的绝对路径,如:

C:\mica_profile_manual\mica_profile

注：该路径下需要创建两个子文件夹Trace_Summaryfea和Trace_Summarymica，分别用于存放带解释的输出文件和纯数字的输出文件

参数3为interval size的大小，如：

1000000

参数4为输出文件名（不含路径），如：

instrace.test

参数5用于说明输入trace的运行状态是AARCH64还是AARCH32，该参数只能是 AARCH64或 AARCH32中的一个，如：

AARCH64

参数6为ROB的大小，如：

64

以下为一运行mica_profile的示例(注：两行之间为空格连接，而非换行符)：

```
./mica_profile C:\mica_profile_manual\mica_profile\instrace.new.log  
C:\mica_profile_manual\mica_profile 1000000 instrace.test AARCH64 64
```

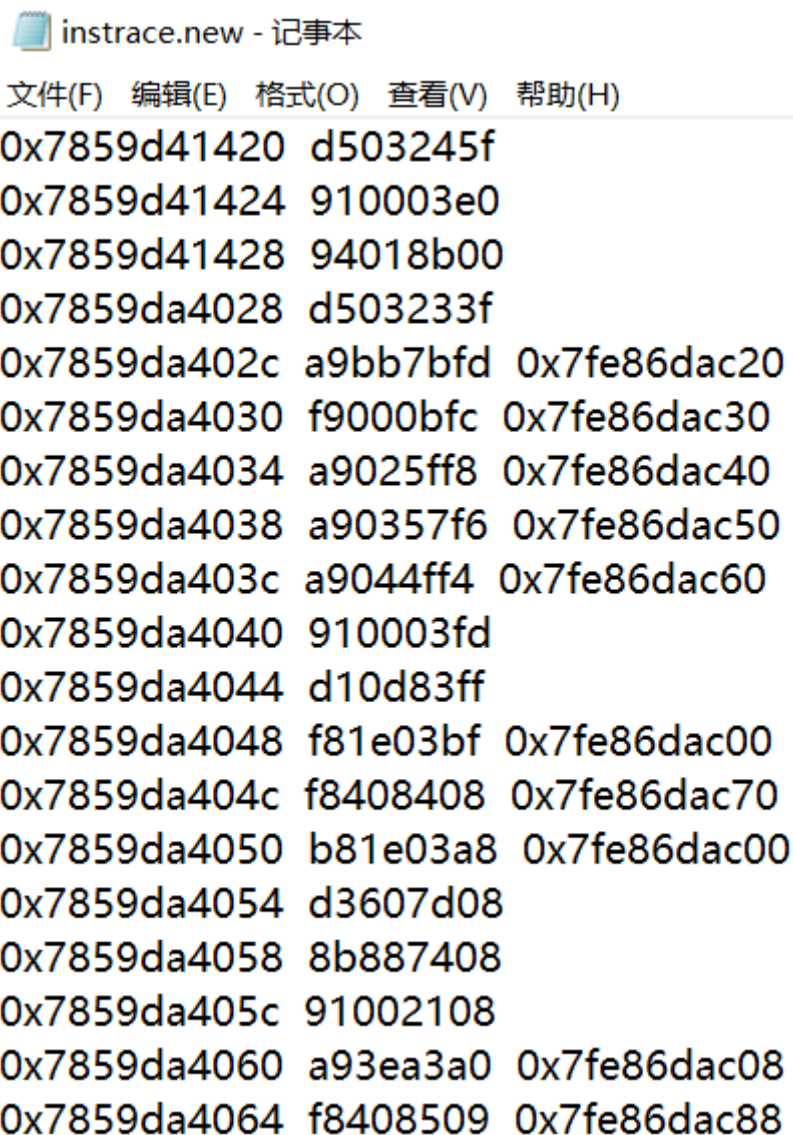
3,正确配置参数并运行后，Windows PowerShell中将出现如下字样：

```
running: interval_1.....
```

表示程序开始运行第一个interval。待出现“All intervals completed”字样时，代表所有interval运行完毕，输出文件将位于路径：“<输出文件夹>\Trace_Summaryfea”以及“<输出文件夹>\Trace_Summarymica”下，分别为带解释的分析文件和纯数字的分析文件。

三，输入文件格式

输入文件的格式与“DynamorioInstrace”文档中说明的输出文件相匹配，其格式的范例见“mica_profile_manual\mica_profile\instrace.new.log”文件，下图为该文件的部分截取：



```
instrace.new - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
0x7859d41420 d503245f
0x7859d41424 910003e0
0x7859d41428 94018b00
0x7859da4028 d503233f
0x7859da402c a9bb7bfd 0x7fe86dac20
0x7859da4030 f9000bfc 0x7fe86dac30
0x7859da4034 a9025ff8 0x7fe86dac40
0x7859da4038 a90357f6 0x7fe86dac50
0x7859da403c a9044ff4 0x7fe86dac60
0x7859da4040 910003fd
0x7859da4044 d10d83ff
0x7859da4048 f81e03bf 0x7fe86dac00
0x7859da404c f8408408 0x7fe86dac70
0x7859da4050 b81e03a8 0x7fe86dac00
0x7859da4054 d3607d08
0x7859da4058 8b887408
0x7859da405c 91002108
0x7859da4060 a93ea3a0 0x7fe86dac08
0x7859da4064 f8408509 0x7fe86dac88
```

其中，每行代表一条指令trace，换行符为“0x0a”。每条trace第一部分为pc值，第二部分为机器码。如果该指令为访存指令，则将访存地址记为第三部分，每部分之间用空格分隔。

四，输出文件解释

位于“<输出文件夹>\Trace_Summaryfea”下的输出文件为带解释的分析文件，位于“<输出文件夹>\Trace_Summarymica”下的输出文件为纯数字的分析文件，二者在顺序上是相互对应的，下面将对输出的带解释的分析文件进行更详细的解释。

1，位于-----InstMix-----部分的数据为指令混合比特征，它指的是特定类型的指令在程序执行过程中存在的数量。

2, 位于-----CriticalPath-----部分的数据为关键路径长度特征, 它指的是指令窗口中最长依赖链路的长度。

3, 位于-----RegDependent-----部分的数据中, regDepLen为寄存器依赖距离, 是指在动态指令流中, 存在真数据依赖的两条指令间间隔的动态指令数; intRegWriteNum、fpRegWriteNum、ccRegWriteNum之和表示对整数寄存器写的指令条数; intRegReadNum、fpRegReadNum之和表示寄存器读的指令条数。

4, 位于-----InstLocality-----部分的数据代表指令局部性, 其中fetchReuseDist是指令重用距离, 指在动态指令流中, 一条指令再次执行时中间间隔的动态指令数, 取2的对数; fetchAddrDist是指令地址差, 指在固定长度的动态指令队列中, 每条指令与首条指令的PC值之差, 取2的对数。

5, 位于-----DataLocality-----部分的数据代表数据局部性, 也称为访存局部性, 其中Globaltemp为全局重用距离, 表示队列中两条访存地址相同的指令间间隔的指令数, 取2的对数; Localtemp为局部重用距离, 表示两条访存地址和PC值均相同的指令间间隔的指令数, 取2的对数; Globalspac为全局地址跨度, 表示相邻两条访存指令, 其访问的内存地址之差, 取8的对数; Localspac为局部地址跨度, 表示相邻两条访存指令PC值相同时, 其访问的内存地址之差, 取8的对数。

6, 位于-----BranchBehavior-----部分的数据中, branchBasicBlockSize表示相邻分支指令间间隔的指令数, 取2的对数; fwBranchs表示向前跳转的分支指令数, bwBranchs表示向后跳转的分支指令数, notTakenBranchs表示未跳转的分支指令数, takenChanged表示跳转发生变化的次数; branchJumpDistance表示分支跳转时的跳转距离, 跳转距离=pc差值/100。

7, 位于-----SerialInstDist-----部分的数据serialBlockSize为串行指令块大小, 表示相邻两条串行指令之间间隔的指令数量, 取2的对数。

8, 位于-----InstSet-----部分的数据中, armInstNum代表该interval中A32指令的数量; thumbInstNum代表该interval中T32指令的数量; jazelleInstNum代表该interval中jazelle指令的数量; aarch64InstNum代表该interval中A64指令的数量。

每个interval的分析数据由该8个部分组成。若输入文件按interval大小可分为多个部分, 则InstSet部分之后的数据是下一个interval的数据。纯数字的分析文件是该带解释的分析文件的数字部分, 每个interval占一行。

五, 测试指定指令

使用Microsoft Visual Studio 2022打开“mica_profile_manual\mica_profile\mica_profile.vcxproj”文件以打开本项目, 在main.cpp文件中, #define TEST_SINGLE_INST以开启测试模式。本源码使用结构体数组instTrace[]来存放指令的信息, 如机器码、PC值等, 每个instTrace结构体代表一条指令。在测试模式下, 将不再通过输入文件提取指令的信息至instTrace中, 而是在216行#ifdef TEST_SINGLE_INST语句后改写代码以手动输入instTrace的信息, 并通过在visual studio顶部的“调试->mica_profile调试属性->调试->命令参数”中键入配置参数(同上文)以对该手动输入的指令进行分析测试。以下为一示例:

1. 打开项目, 在main.cpp中, 去掉第15行 //define TEST_SINGLE_INST中的//注释符。

2. 在#define TEST_SINGLE_INST后的代码处(221行), 添加一条指令的信息:

```
instTrace[0].pc = 0xfeef7d8;
instTrace[0].opcode = 0xf943aa80;
```

```
instTrace[0].inst_paddr = instTrace[0].pc;  
instTrace[0].inst_iset = ISET_THUMB;  
instTrace[0].load_paddr = 1;
```

3,在visual studio顶部的“调试->mica_profile调试属性->调试->命令参数”中键入如下参数(注：两行之间为空格连接，而非换行符)：

```
C:\mica_profile_manual\mica_profile\instrace.new.log  
C:\mica_profile_manual\mica_profile 1 instrace.test AARCH32
```

4,接下来可设置断点或单步调试以对该条指令进行分析,建议将断点设于statistic_trace.cpp源文件下的各函数入口处进行调试。