

Simpoint 实验记录

采用不同聚类数对结果的影响

聚类数为4（MAX=5）

- 实验配置如下：

```
# 这里normal的配置和使用O3重载checkpoint时的配置一样
build/ARM/gem5.fast --outdir=${SE_OUT_DIR_NORMAL} \
configs/example/se.py \
--cpu-type O3_ARM_v7a_3 \
--cpu-clock 2.5GHz \
--num-cpu 1 \
--caches --l2cache --l1i_size 64kB --l1d_size 32kB --l2_size 1MB \
--l1i_assoc 8 --l1d_assoc 8 --l2_assoc 16 --cacheline_size 128 \
--mem-type DDR3_2133_8x8 --mem-size 16GB
```

- 此时的权重为：

聚类簇序号	权重（已归一化）
0	0.345946
1	0.0378378
2	0.318919
3	0.297297

可以发现，此时存在三个权重较大的聚类簇，而剩下一个聚类簇的权重相比可以忽略不计。

- 对此分类结果进行模拟，结果如下：

参数	正常运行的结果	Simpoint的结果	误差(%)
IPC	1.676	1.336	20.24
Dcache miss rate (%)	6.462	5.881	9.00
Mispredict rate (%)	1.664	3.266	96.27
ROB stall rate (%)	0.0029	0.0019	34.36

发现当聚类簇总数过小时会存在相当大的误差，可能的原因如下：

- 设置的 interval length 不够长，之前设置的 interval length = 1e7 可能无法较好的模拟一个较长的代码段
- 可能有不三个主要的代码段，设置聚类数过少就较为片面

聚类数为28（MAX=30）

- 实验配置如下：

```
# 这里normal的配置和使用O3重载checkpoint时的配置一样
build/ARM/gem5.fast --outdir=${SE_OUT_DIR_NORMAL} \
configs/example/se.py \
--cpu-type O3_ARM_v7a_3 \
--cpu-clock 2.5GHz \
--num-cpu 1 \
--caches --l2cache --l1i_size 64kB --l1d_size 32kB --l2_size 1MB \
--l1i_assoc 8 --l1d_assoc 8 --l2_assoc 16 --cacheline_size 128 \
--mem-type DDR3_2133_8x8 --mem-size 16GB
```

- 此时的权重为（只标出了权重大于0.05的项）：

聚类簇序号	权重（已归一化）
-------	----------

聚类簇序号	权重（已归一化）
...	...
3	0.108106
4	0.0972973
...	...
11	0.145946
...	...
17	0.0648649
...	...
24	0.0594595
...	...

可以发现，此时不存在权重特别大的聚类簇，但相比而言较大的有三个聚类簇（近似大于等于0.1），其他结果都比较平均。

- 对此分类结果进行模拟，结果如下：

参数	正常运行的结果	Simpoint的结果	误差(%)
IPC	1.676	1.687	0.71
Dcache miss rate (%)	6.462	6.452	0.16
Mispredict rate (%)	1.664	1.694	1.80
ROB stall rate (%)	0.0029	0.0030	4.52

发现当聚类簇较多时，误差被控制在一个合理的范围内。

- 接下来考虑**只算权重大于0.05的聚类簇(共五个)**，结果如下：

参数	正常运行的结果	Simpoint的结果	误差(%)
IPC	1.676	1.712	2.17
Dcache miss rate (%)	6.462	6.405	0.88
Mispredict rate (%)	1.664	1.436	13.69
ROB stall rate (%)	0.0029	0.0042	43.58

发现结果相较于之前全部都算的误差变大，部分误差甚至超过 10%，所以还是要全部 checkpoint 都算为好。

结论

- 综上所述，在 GEM5 里使用 Simpoint 是具有相当合理性的，但是如果取典型的片段进行模拟，合理的措施还是尝试减小聚类个数以及增大 interval length.

在重载checkpoint时更改参数对实验结果的影响

更改CPU类型以及主频

- 将CPU类型从 O3_ARM_v7a_3 更改为 DerivO3CPU，将主频从 2.5GHz 调整为 1GHz.
- 实验结果如下：

参数	正常运行的结果	Simpoint的结果	误差(%)
IPC	1.906	1.924	0.94
Dcache miss rate (%)	7.170	7.192	0.32

参数	正常运行的结果	Simpoint的结果	误差(%)
Mispredict rate (%)	1.844	1.798	2.51
ROB stall rate (%)	0.0007	0.0005	20.27

更改cache参数

- 将 cache 参数从
-caches --l2cache --l1i_size 64kB --l1d_size 32kB --l2_size 1MB --l1i_assoc 8 --l1d_assoc 8 --l2_assoc 16 --cacheline_size 128
调整到
--caches --l2cache --l1i_size 32kB --l1d_size 16kB --l2_size 512kB --l1i_assoc 1 --l1d_assoc 1 --l2_assoc 1 --cacheline_size 128
- 实验结果如下：

参数	正常运行的结果	Simpoint的结果	误差(%)
IPC	1.535	1.493	2.74
Dcache miss rate (%)	9.759	9.790	0.31
Mispredict rate (%)	1.685	1.717	1.89
ROB stall rate (%)	0.0057	0.0061	8.07

更改mem类型

- 将 mem 参数从 DDR3_2133_8x8 调整为 DDR4_2400_16x4
- 实验结果如下：

参数	正常运行的结果	Simpoint的结果	误差(%)
IPC	1.668	1.677	0.58
Dcache miss rate (%)	6.463	6.453	0.16
Mispredict rate (%)	1.664	1.694	1.78
ROB stall rate (%)	0.0028	0.0031	9.67

结论

- 后来又尝试了减少聚类总数但增加 interval length 的策略，但是准确率依然不够，详见下表：

```
Simpoint results are:
IPC = 1.6541297322928858
Dcache miss rate = 0.05404827342566031
Predict incorrect rate = 0.020100357727433814
ROB stall rate = 2.463187811511429e-05

Normal results are:
IPC = 1.667545945165815
Dcache miss rate = 0.06463431515940303
Predict incorrect rate = 0.01663909859660678
ROB stall rate = 2.86616936490782e-05

Errors are:
IPC : 0.8045483191526154 %
Dcache miss : 16.37836141318294 %
Predict incorrect : 20.801962983336676 %
ROB stall: 14.059935129107476 %
```
- 在重载时更改 CPU,cache 以及 mem 的参数对除了 ROB stall 外的实验结果都没有特别大的影响，ROB stall 受影响大主要是因为 Stall 次数太少了，所以较少的浮动在误差上会体现的比较大，但是从总的 ROB stall rate 来看绝对误差都在 0.001% 内，还是可以接受的。