

Multi-thread Lab Report

Machine Used: Legion Y7000P IRX9

Operating System: Ununtu 20.04

测试说明

在src文件夹中有test_validation.sh和auto_run.sh脚本。

其中test_validation.sh会自动在images里面的所有图片上运行histo-private, histo-lockfree, histo-lock1, histo-lock2, 并将结果与histogram比较以检测正确性。

而auto_run.sh可以选择二进制文件、线程数、ppm文件、运行次数来运行, 并计算出平均值和标准差。

具体使用方式是:

```
chmod +x test_validation.sh
chmod +x auto_run.sh

./test_validation.sh run_times
./auto_run.sh binary threads ppm run_times
```

其中run_times是运行次数, binary是二进制文件名, threads是线程数, ppm是图片文件名。

auto_run也可以不加参数直接运行, 它会在命令行中引导你输入参数。

histogram

此文件是lab自带的, 只支持单线程。均值和标准差是100次运行后的结果计算得到的, 单位ns。下面表格中, 括号外的是均值, 括号内的是标准差。

Image	Mean(Std Dev)
300px.ppm	48233.92(9957.39)
earth.ppm	4083411.67(157468.37)
flood.ppm	162382.43(14136.76)
moon-small.ppm	259306.24(25280.60)
moon-large.ppm	19774670.50(350273.75)
phobos.ppm	961417.67(161075.34)
university.ppm	105547.93(11682.05)

Exercise 1: histo-private

每个线程计算一个私有直方图。在所有线程完成之后，主线程会合并所有的私有直方图。本文件主要使用pthread_create/pthread_join来实现并行。

表格中图片的顺序与histogram中一致

1	2	4	8
45560(4636)	196675(115054)	209527(115490)	286948(68355)
4081634(166223)	2265167(143010)	1327493(223972)	1292480(138234)
196963(29638)	222253(35485)	258291(51800)	362508(121951)
254102(36426)	257638(64830)	251905(51463)	373736(129118)
20280518(685551)	9761579(421789)	8091081(1174497)	5680276(978890)
993879(48451)	727589(66971)	574917(67503)	549028(167353)
99086(11323)	207841(54257)	226176(71958)	294808(51192)

Exercise 2: histo-lockfree

所有线程共享一个直方图。数据更新是通过 C++11 的 `std::atomic<int>` 的无锁原子加法实现的。

1	2	4	8
742202(56747)	1365212(136780)	1397970(181193)	1518508(238933)
49983945(1009220)	79002572(2701724)	87551536(2132508)	91306186(2669040)
2515646(91740)	3827164(321341)	3708478(110675)	3703725(462332)
3285086(169857)	8368163(574298)	8165681(247125)	8290183(793009)
252052057(2348579)	504881909(13579137)	502516220(12733783)	497617162(14558568)
8509192(214206)	21292201(890690)	19031565(475470)	19160567(617057)
1505425(50030)	2580959(211750)	3101116(500061)	3285977(690250)

Exercise 3: histo-lock1 & histo-lock2

histo-lock1中利用 `__sync_lock_test_and_set` 实现了TAS自旋锁，并且所有线程仍是共享一个直方图。

1	2	4	8
46215(7801)	2987497(279866)	3706763(296572)	4526174(268667)
4047473(123997)	187589458(4640509)	263778285(5146946)	306067055(9663060)
201814(38609)	8101149(388431)	10974920(397256)	12497169(535814)
252063(20479)	19193021(888097)	26351300(1002337)	27439688(1018841)
20247822(504387)	1196545182(53298805)	1616975618(30221961)	1673374116(7878529)
988609(41749)	53586671(2472517)	68020539(1805515)	103617345(9463584)
98732(6604)	5288749(412242)	8274293(277527)	9758336(749314)

histo-lock2中利用 `__sync_fetch_and_add` 实现了基于Ticket Lock的自旋锁。

1	2	4	8
45742(3256)	2988188(322128)	3612000(376649)	4214473(478656)
4026657(150005)	193806609(3228142)	271898159(5432633)	299944004(5868463)
189862(28852)	8864153(674542)	10722067(620670)	11924175(904801)
251898(16792)	20989827(728369)	25679246(715807)	25572976(2069411)
20415032(563066)	1220164626(20801929)	1480937262(26353183)	1490958639(2240311)
983448(44491)	54059522(1164221)	76438793(2086052)	111512791(3370362)
101878(16689)	5575970(528802)	8342481(413282)	9374277(341267)

参考文献

1. <https://zhuanlan.zhihu.com/p/125742057> (<https://zhuanlan.zhihu.com/p/125742057>)
2. <https://www.csgraduates.com/code/spinlock/> (<https://www.csgraduates.com/code/spinlock/>)
3. <https://zhuanlan.zhihu.com/p/80727111> (<https://zhuanlan.zhihu.com/p/80727111>)