组会报告

徐益

2018 年 12 月 13 日

1 工作内容

1. 处理线程死锁 Bug；
2. 更换线程绑定 CPU 方案；
3. 测试系统性能；
4. 修改开题报告。

2 更换线程绑定 CPU 方案

```
pool_init(CORE_NUM, 1, 0);       // Tx Scheduling Thread Pool
pool_init(CORE_NUM * 2, 1, 1);   // Channel Thread Pool
pool_init(CORE_NUM * 3, 1, 2);   // Rx Scheduling Thread Pool
for (i = 0; i < CPU_NUM; i++)
{
    tx_pool_indx[i] = 3 + i;
    pool_init(CORE_NUM * i + 1, TX_CORE_NUM, tx_pool_indx[i]);
    rx_pool_indx[i] = 7 + i;
    pool_init(CORE_NUM * i + 4, RX_CORE_NUM, rx_pool_indx[i]);
}
```

图 1: 新的线程绑定 CPU 方案

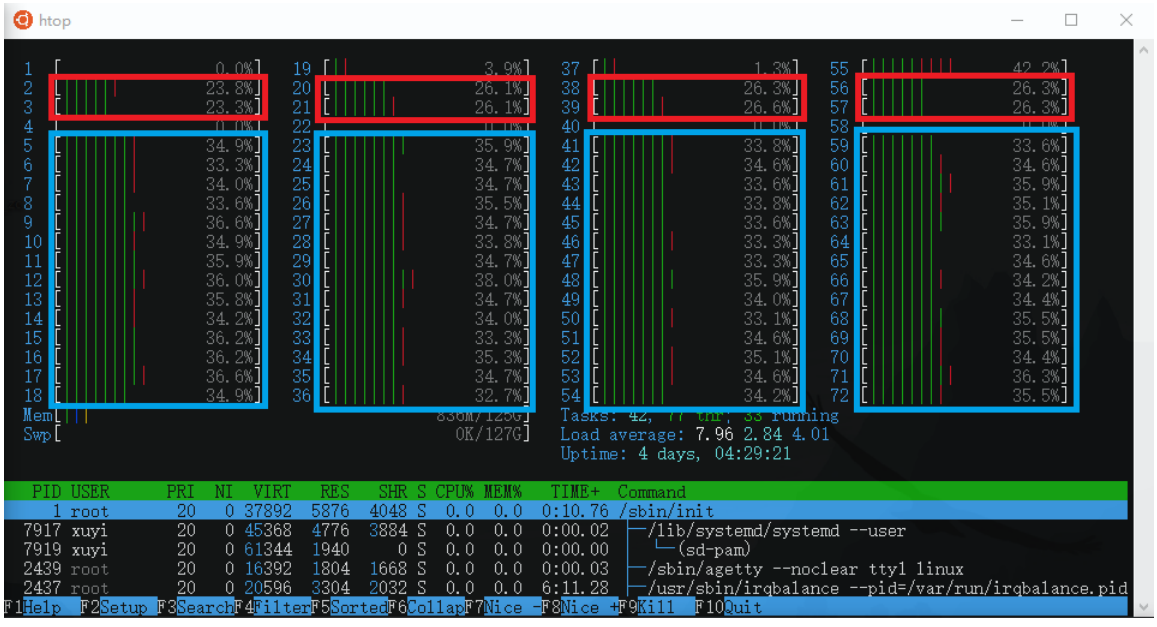

图 2: CPU 占用情况

3 测试系统性能

4 测试结果

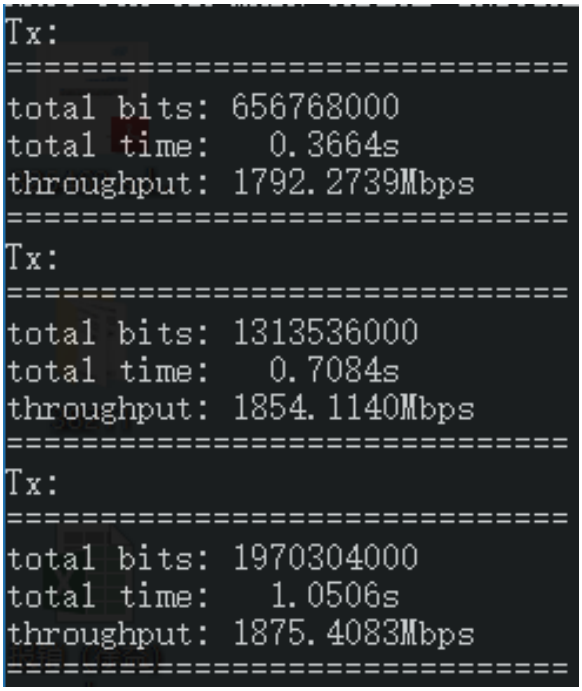

图 3: 发送端性能

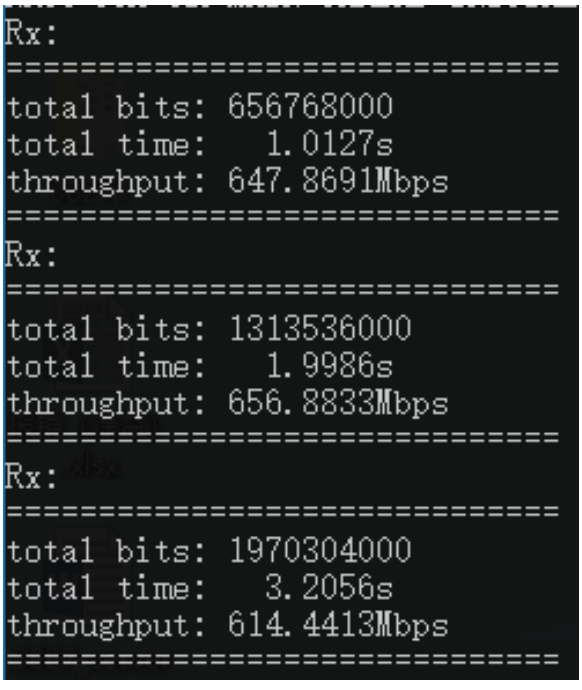

图 4: 接收端性能

5 CPU 使用情况与吞吐量的关系

表 1: 发送端 CPU 使用情况与吞吐量的关系

CPU	CORE	Throughput	Degree of Parallelism
1	1	302.28Mbps	100.00%
4	1	1038.59Mbps	85.90%
4	2	1792.27Mbps	74.11%

表 2: 接收端 CPU 使用情况与吞吐量的关系

CPU	CORE	Throughput	Degree of Parallelism
1	1	47.38Mbps	100.00%
4	1	176.62Mbps	93.19%
4	2	298.13Mbps	78.65%
4	3	388.55Mbps	68.34%
4	4	418.02Mbps	55.41%
4	5	512.31Mbps	54.06%
4	6	521.66Mbps	45.88%
4	7	520.70Mbps	39.25%
4	8	610.47Mbps	40.26%
4	9	615.41Mbps	36.08%
4	10	604.59Mbps	31.90%
4	11	609.78Mbps	29.25%
4	12	609.33Mbps	26.79%
4	13	586.91Mbps	23.82%
4	14	674.87Mbps	25.44%

5.1 其他问题

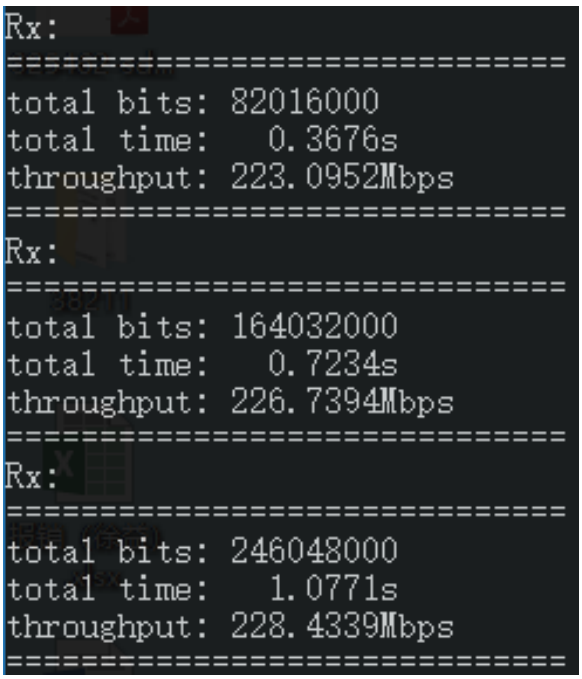

图 5: 单流 4CPU2CORE

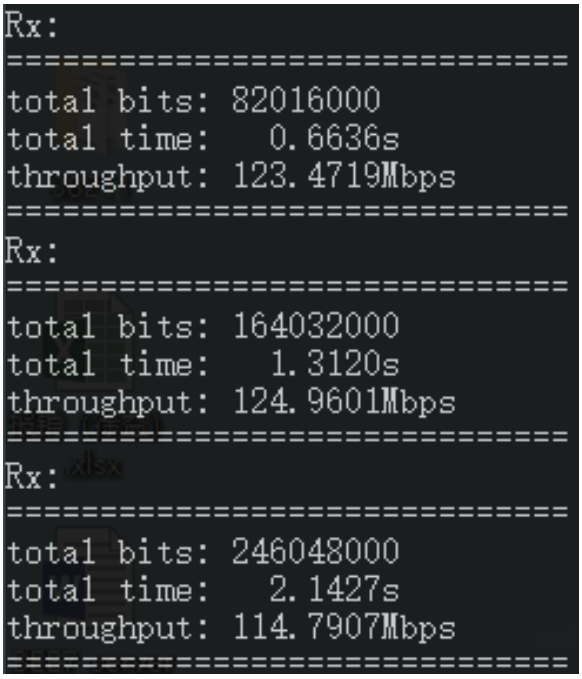

图 6: 单流 4CPU14CORE

6 已做过的尝试和待实现的尝试

6.1 已做过的尝试

1. 在按线程池分配 CPU 后，再修改线程亲和度。（性能未提升）
2. 在发送端改为轮询调度结构。（性能未提升）

6.2 待实现的尝试

1. 按 CPU 分配运算空间。
2. 修改线程池源码。

7 修改开题报告