在不同的FIB表规模下，进行二分查询与线性查询的性能对比。统计百万次名字查找的总时间，计算出平均查找时间（单位：us）。

FIB表中name prefix的平均component数目为4.2，查找的name prefix的平均component数目为11.2时的实验结果如下所示：

全hit情况下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIB规模 | 百万 | 千万 | 一亿 | 十亿 |
| 线性查询 | 3.484 | 3.342 | 3.859 | 11.037 |
| 二分查询 | 1.862 | 2.024 | 2.374 | 5.408 |
| ratio = linear/binary | 1.87/0.53 | 1.65/0.60 | 1.63/0.61 | 2.04/0.48 |

8.00vs3.89 8.00vs3.89 8.00vs3.92 8.00vs3.93

FIB表规模变大时，hash冲突导致查询速度降低。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIB规模 | 百万 | 千万 | 一亿 | 十亿 |
| 线性查询 | 4.718 | 4.637 | 6.681 | 13.650 |
| 二分查询 | 1.588 | 1.661 | 2.327 | 4.698 |
| ratio = linear/binary | 2.97 | 2.79 | 2.87 | 2.91 |

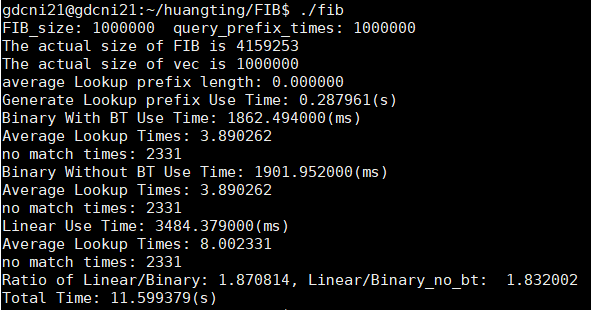
全miss情况下（理论上最优于线性查找）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIB规模 | 百万 | 千万 | 一亿 | 十亿 |
| 线性查询 | 0.466 | 0.525 | 0.463 | 0.501 |
| 二分查询 | 0.131 | 0.143 | 0.144 | 0.150 |
| ratio = linear/binary | 3.55 | 3.67 | 3.21 | 3.34 |

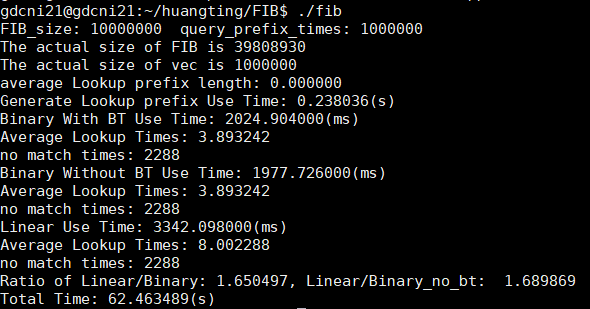
全miss情况下

全hit情况下：

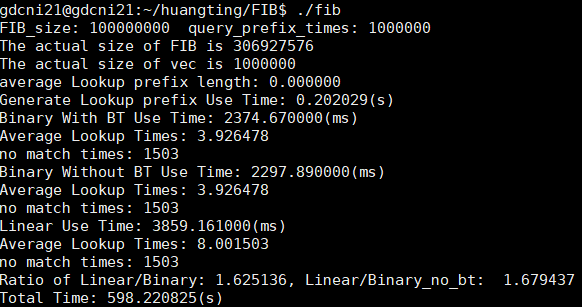
百万规模下



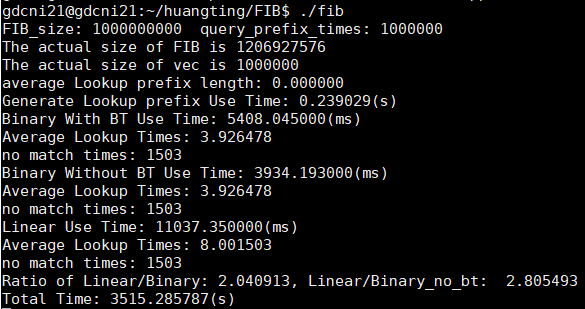
千万规模下



一亿规模下：

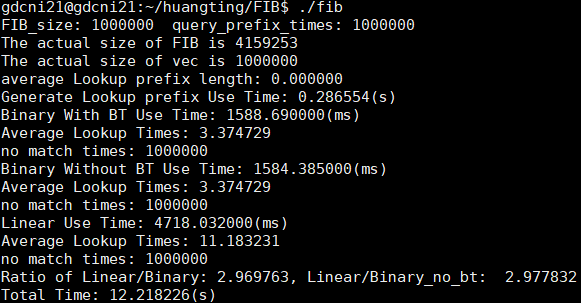


十亿规模下：

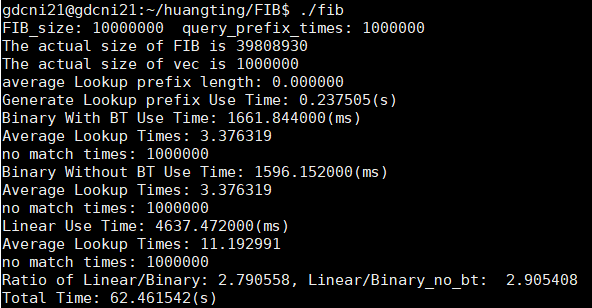


全miss情况下：

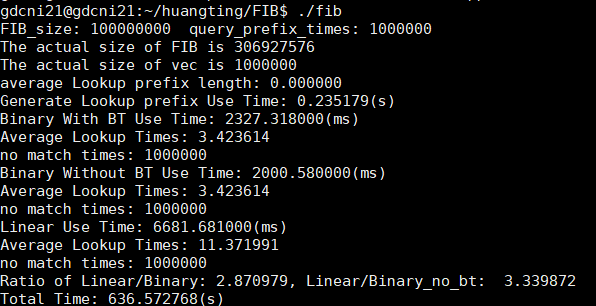
百万规模下



千万规模下



一亿规模下：



十亿规模下：

