实验报告

姓名： 袁峥 学号： 2015K8009929008

一、实验题目：高效IP路由查找实验

二、实验内容

1、实现最基本的前缀树查找

2、实现多bit前缀树及优化

叶推+压缩指针+压缩向量

3、基于forwarding-table.txt数据集进行结果测试

①以最基本的前缀树查找结果为基准，检查多bit前缀树是否正确

可以将forwarding-table.txt中的IP地址作为查找的输入

②对比两种不同方法的性能内存开销

平均单次查找时间

三、实验流程

1、最基本的前缀树查找

建树时从最高位开始，每一位代表一个节点，如果当前路径对应分支为空，则增加相应节点。在整棵树构建完毕后，可以进行前缀树压缩，对于只有一个子节点的分支可以进行如图的压缩，以减少节点数目和匹配次数。



2、多bit前缀树及优化

前缀树中每次不只匹配1比特，而是多比特一起匹配，以减少访问节点的数目，在具体实现中，采用了4bit前缀树。

优化①：叶推：将包含匹配前缀的中间节点下推到叶子节点（可能不止一个叶子节点）



优化②：压缩指针：将指向子节点的指针，改进为指向子节点向量的指针，可以减少内存空间使用



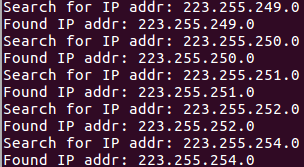
优化③：压缩向量



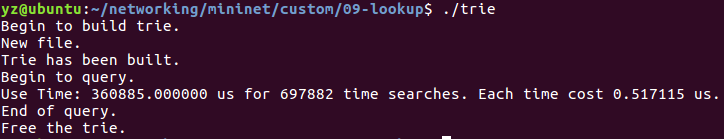
四、实验结果

1、最基本的前缀树查找

①正确性

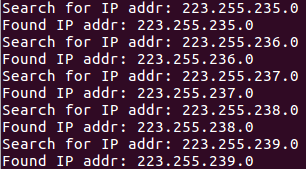


②平均单次查找时间

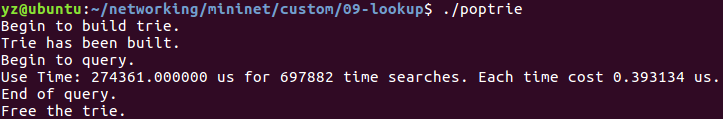


2、多bit前缀树及优化

①正确性



②平均单次查找时间



五、结果分析

从查找结果可以看出，最基本的前缀树查找和多bit前缀优化树的查找结果都是正确的。从平均单词查找时间来看，使用叶推、压缩指针及压缩向量优化后的多bit前缀树的查找时间明显要比最基本的前缀树查找时间短，大约快了四分之一，可以看出多bit前缀树还是十分有效的。

六、实验总结

此次实验是难得的一次代码全部需要自己写的实验。相比于以前的实验，在实验前阅读老师提供的源代码，通过分析可以理清函数之间的关系，此次实验没有提供任何的初始代码，前缀树的数据结构也需要自己设计，这也是此次实验的难点之一。

此外，此次实验碰到的最大的困难是在调试的时候，无法清晰的获得建成树的结构，因此在调试过程中只能通过大量输出，并加上静态代码分析的判断，来不断分析猜测出错的原因，再不断的改正尝试。这样的办法十分浪费时间，但好像没有找到效率更高的调试办法。最后在反复尝试后，终于查找出了正确结果。

另外，在我原来的理解中，此次优化实验实现的是边建树边优化，在压缩向量的实现上，每次如果新加一个节点，需要重新分配内存并进行内存拷贝，实现上效率低下，因此对此产生了疑惑。在做完实验后与老师交流的过程中，得知此次优化实验的目的是在先建完多bit前缀树后，整棵树一起进行优化，而非在边建树的过程中边进行优化，这样思考起来，进行的三步优化就比较科学。