日报格式:

主 题：日报

汇报人：左超

日 期：2018年7月23日 星期二

内 容：

* 今日学习了的内容：
* 初步了解了Elasticsearch和Lucence

Elasticsearch的功能：

1、分布式的搜索引擎和数据分析引擎

2、全文搜索，结构化检索，数据分析

3、对海量数据进行实时性的处理

特点：

可以作为一个大型的分布式集群技术处理PB级数据

主要是将全文搜索，数据分析，以及分布式技术合并在了一起

数据库的功能面对很多领域时不够用的，比如全局搜索同义词词处理复杂数据的实时性处理。

Lucene：

非结构化数据查询方法：就是一个文档一个文档的看，对于每一个文档，从头看到尾，如果此文档包含此字符串，则此文档为我们要找的文件，接着看下一个文件，直到扫描完所有的文件

顺序扫描法：

全文检索：将非结构化数据中的一部分信息提取出来，重新组织，使其变得有一定结构，然后对此有一定结构的数据进行搜索，从而达到搜索相对较快的目的

Elasticsearch的安装。

* Tcp/Ip基础知识

TCP/IP参考模型：

应用：控制应用程序；

表示：格式化数据；

会话：控制会话，建立管理终止应用程序会话；

传输：提供可靠和尽力而为的传输,(TCP UDP)

网络：互联网层,定义一个逻辑的寻址，选择最佳路径传输，路由数据包,路由器。

数据链路层：网络接口层,定义如何格式化数据，支持错误检测；

物理层：硬件,定义一些电器，机械，过程和规范，如集线器；

* 没有解决的问题：

1）

2）

* 明天的学习计划：

1）Tcp/ip数据链路层

2）深入深入了解Elasticsearch

4.学习任务完成度： （8）%[注：0% ~ 100%]

5.对自己今天学习的满意度：（8）分 [注：1-10分]

6.对今天的学习做一句话的总结：

附录：

算法1：根据前序遍历和中序遍历重构二叉树

**package** BinaryTree;

**public** **class** TreeNode {

Integer val;

TreeNode left;

TreeNode right;

**public** TreeNode(Integer val){

**this**.val = val;

left = right = **null**;

}

**public** Integer getVal() {

**return** val;

}

**public** **void** setVal(Integer val) {

**this**.val = val;

}

**public** TreeNode getLeft() {

**return** left;

}

**public** **void** setLeft(TreeNode left) {

**this**.left = left;

}

**public** TreeNode getRight() {

**return** right;

}

**public** **void** setRight(TreeNode right) {

**this**.right = right;

}

}**package** BinaryTree;

**import** java.util.Vector;

**public** **class** BuildBinaryTree {

**public** TreeNode BuildTree(Vector<Integer> pre,Vector<Integer> vin) {

Vector<Integer> Preleft = **new** Vector<Integer>();

Vector<Integer> Preright = **new** Vector<Integer>();

Vector<Integer> Vinleft = **new** Vector<Integer>();

Vector<Integer> Vinright = **new** Vector<Integer>();

**if** (!pre.isEmpty()||vin.isEmpty()) {

**return** **null**;

}**else** {

TreeNode T = **new** TreeNode(pre.get(0));

**int** i = 0;

**while**(T.getVal()!= vin.get(i)) {

Vinleft.add(vin.get(i));

Preleft.add(pre.get(i+1));

i++;

}

i++;

**while**(i<vin.size()) {

Vinright.add(vin.get(i));

Preright.add(pre.get(i));

i++;

}

T.left = BuildTree(Preleft,Vinleft);

T.right = BuildTree(Preright,Vinright);

**return** T;

}

}

}

【学习内容真实可靠，给自己的评分和完成度客观真实，不谦虚，不夸大】