Elasticsearch: Fast & Slow

查超 基础平台部

2015 年 10月

HanSight 瀚思

Who am I

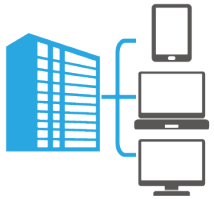
- **查超**@瀚思
- 加入一年，成都研发中心基础平台部
- 技术背景：Hadoop、Elasticsearch、Spring Cloud
- 目前做SaaS产品线

Elasticsearch：Fast & Slow

1. 背景
   - Hansight
   - 系统架构
   - 客户需求分析
   - 分而治之
2. Searching Fast
   - 场景分析
   - 设计方案
   - 内存快查
   - 修改源码
   - 实测总结
3. Searching Slow
   - 场景分析
   - Hadoop并行计算
   - 代码实现
   - 限制及解决思路
4. 尾声
   - Q&A
   - 感谢
   - 欢迎加入

企业背景

HanSight 瀚思

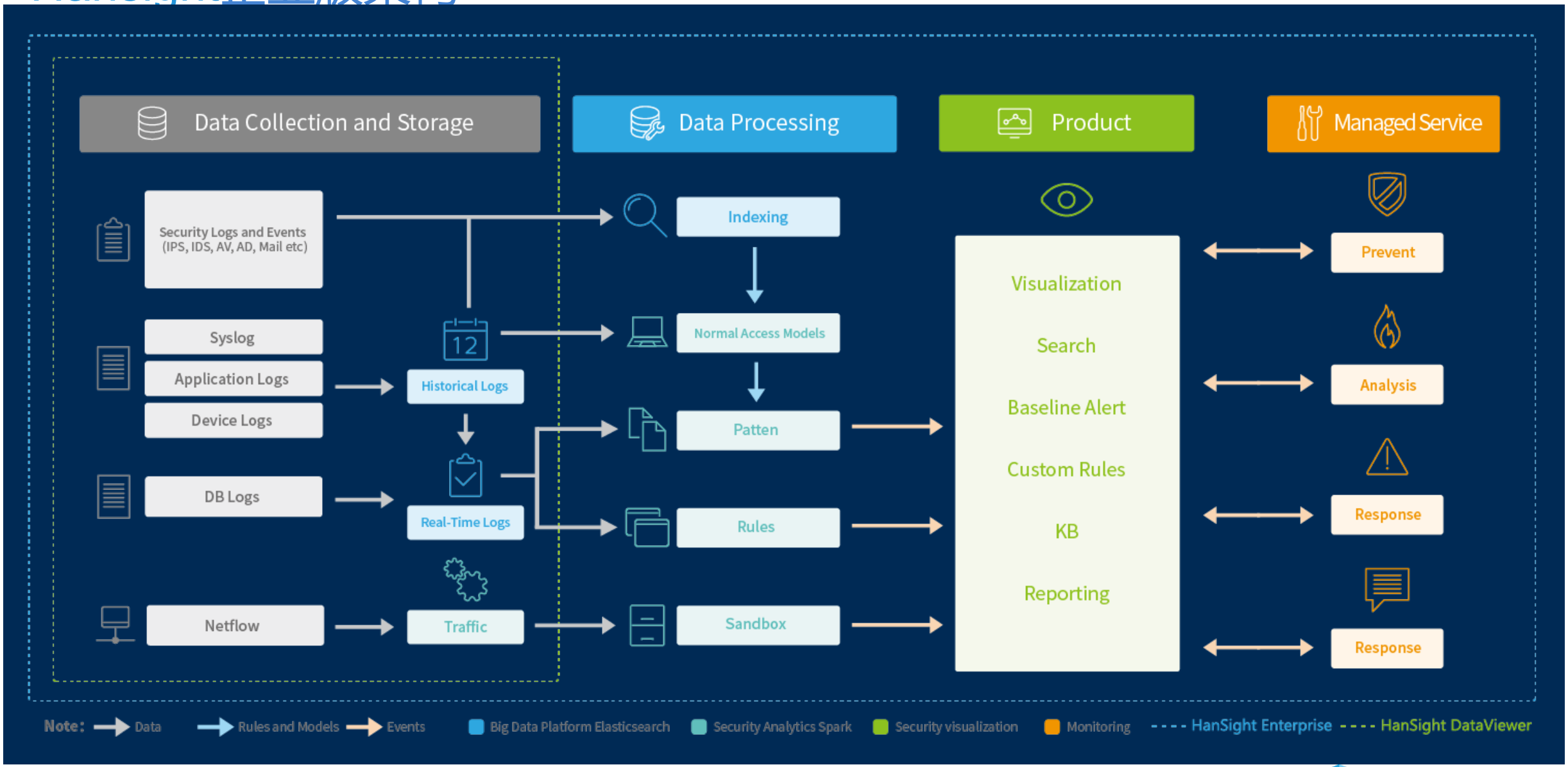


- 成立于2014年的创业公司
- **数据驱动安全** - 以大数据分析的方式解决安全问题
- 创始人来自趋势科技、云天、微软、Oracle

- 两大类产品线：**企业版**和**SaaS**版
- 企业版销售对象是各种大型企业总部、事业单位等
- 基于Elasticsearch 1.5
- 重点在性能、安全、算法

- SaaS版 - **安全易** - 面向中小企业
- 云端的多租户SIEM
- 基于Elasticsearch 1.7 + Kibana 4.1 改编
- 重点在可视化

Hansight企业版架构

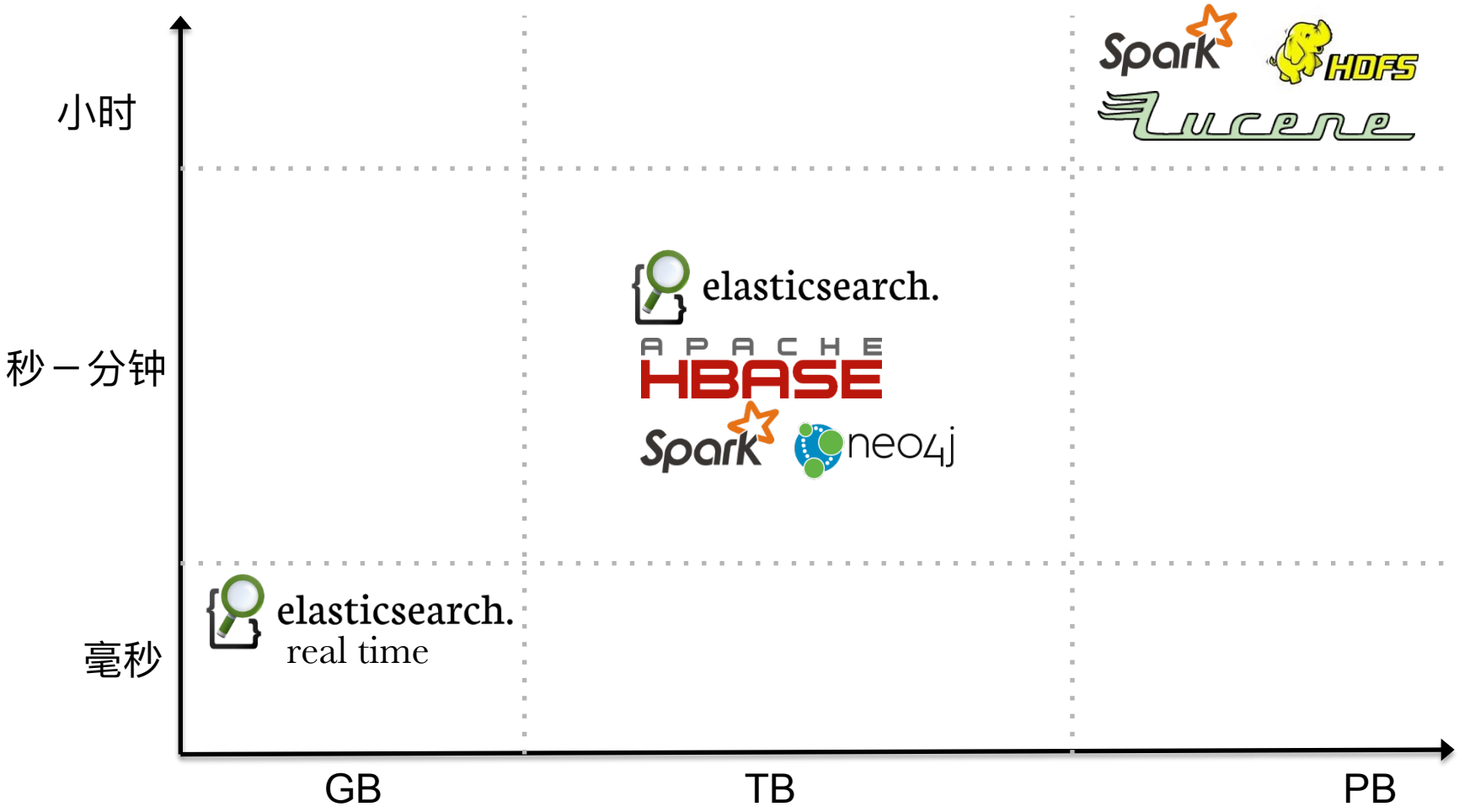


大企业用户特殊处理能力需求

- 一般用户需求：
  - 一天一个Elasticsearch Index
  - 数据量100G - 10T之间
  - 延迟<10秒，一般保留30天Index
- 特殊用户需求：
  - **毫秒级**别实时数据：实时账号欺诈检测
  - **PB级**别历史数据：几个月前的安全事件回溯

混合Elasticsearch方案

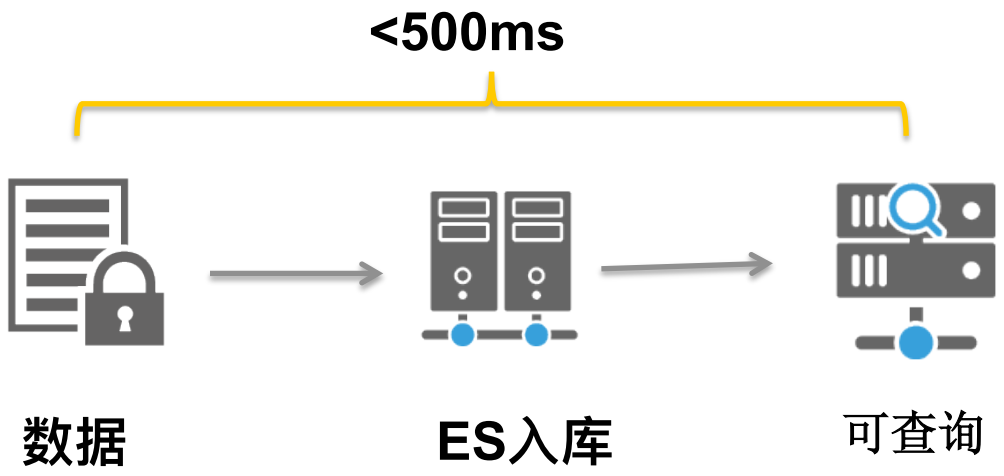
- 划分不同的容量和性能场景，分而治之



Searching Fast

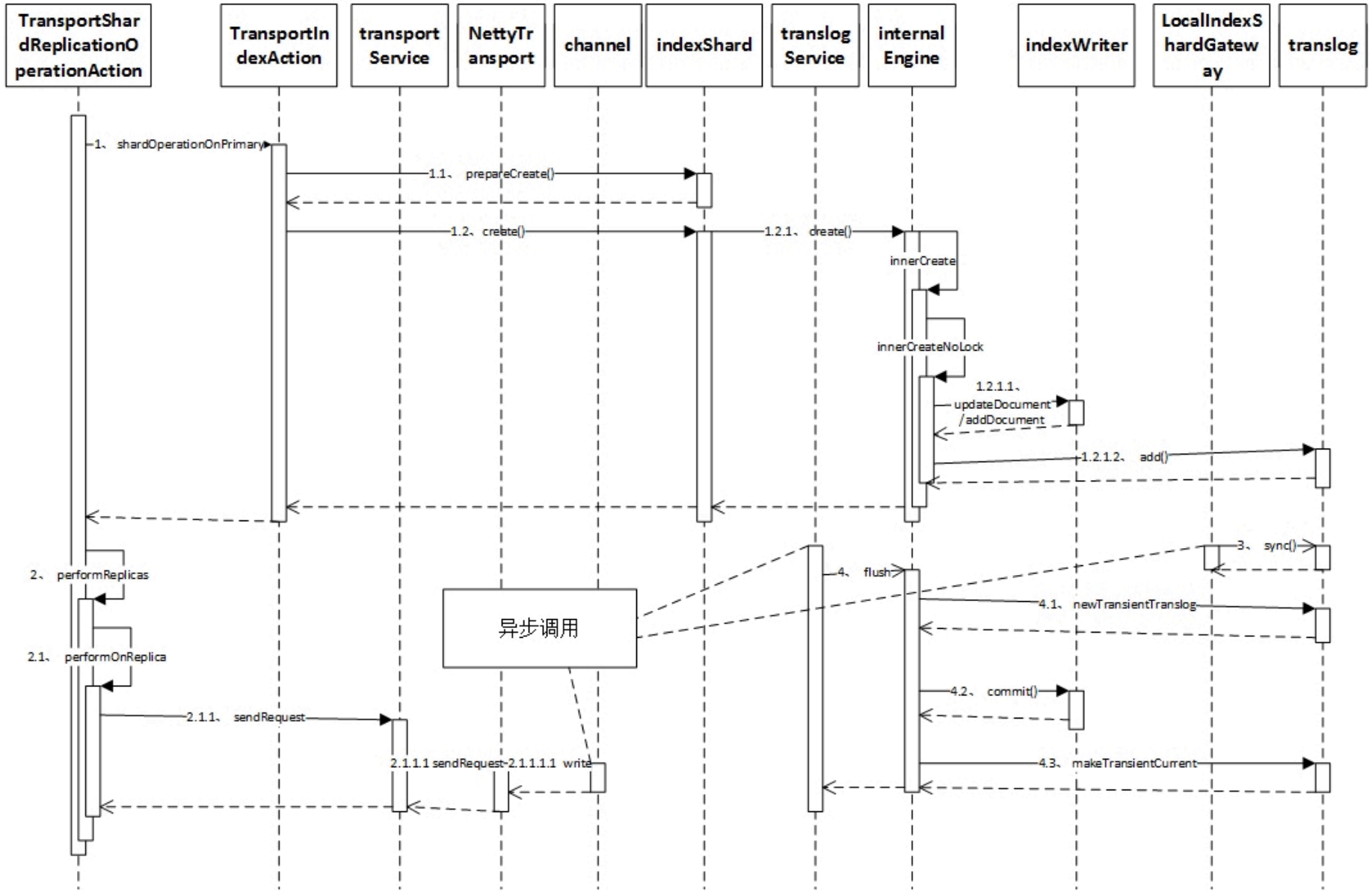
Searching Fast

- 场景分析
  - 快速处理数据
  - 不改变数据入库速度
  - 入库到查询之间的延迟降到毫秒级
  - 如何做到？



改flush_interval配置?

- index.refresh_interval: 500ms
- Index速度无法提升
- 从分析Elasticsearch Index流程入手

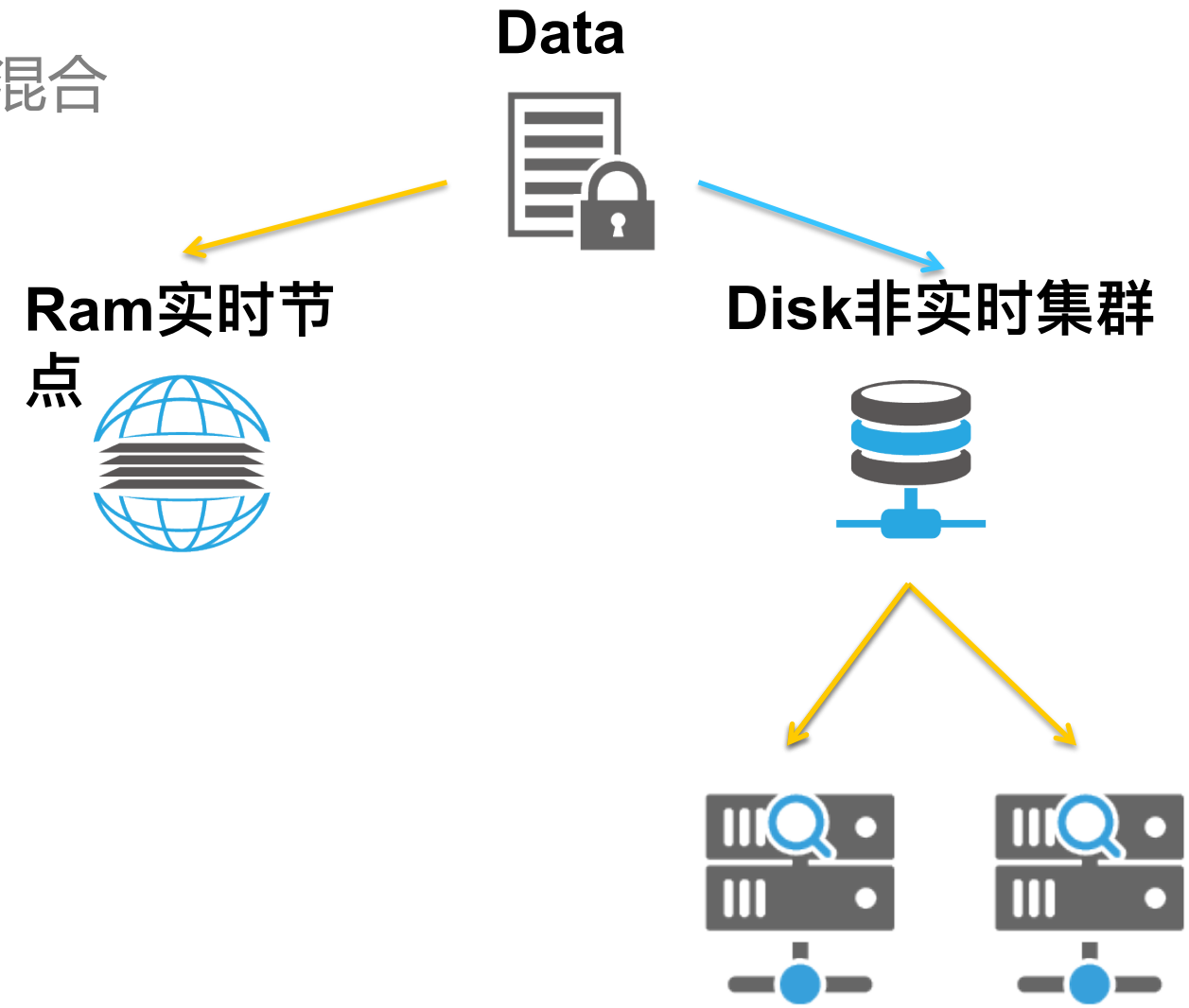


Searching Fast

- 设计方案
  - 数据放内存中查询
  - 修改Elasticsearch代码，去掉translog，减少额外数据操作
  - Index吞吐量上不去，但是CPU使用率不高 - 改Elasticsearch配置

方案部署

- 非实时查询集群 + 实时查询节点混合
  - 数据全部放在内存减少磁盘IO
  - 允许少量数据丢失
  - 数据一式两份会进入延迟集群



Searching Fast

- 优点
  - 高Index吞吐量 + 低延迟
  - 接口仍然是标准的Elasticsearch
- 实测总结
  - 4T/Day，2.8G/分钟，保留10分钟
  - 每分钟一个Index，构成先进先出队列，32G = 10 Index
  - 延迟<200ms
  - EPS > 24,000

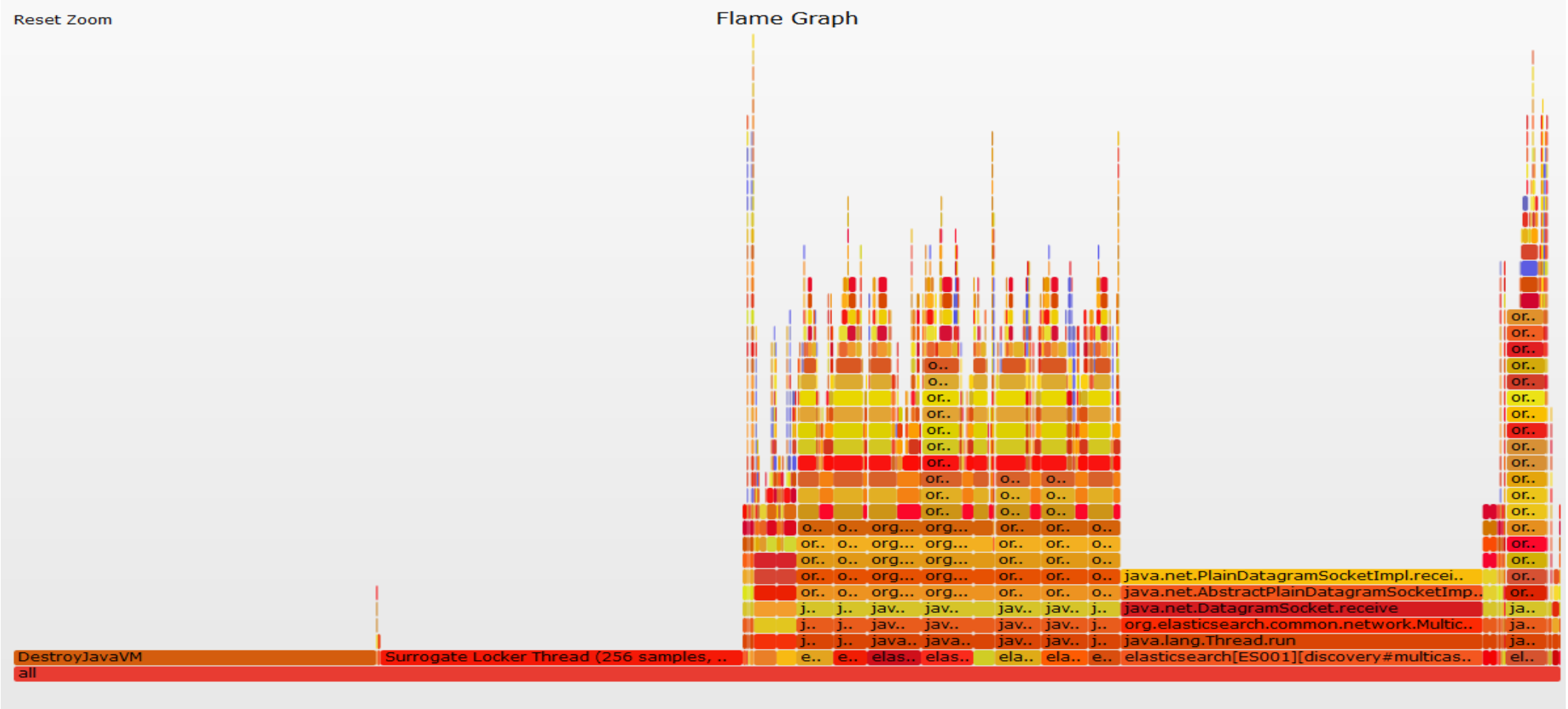
Searching Slow

Searching Slow

- PB级别历史数据查询
- 为何不直接配置一个ES集群有上千个Index？
  - 业内最大ES集群不超过200
  - ES对机器资源有较高需求
  - 用户偶尔需要查询老数据
  - 适量牺牲速度换资源需求

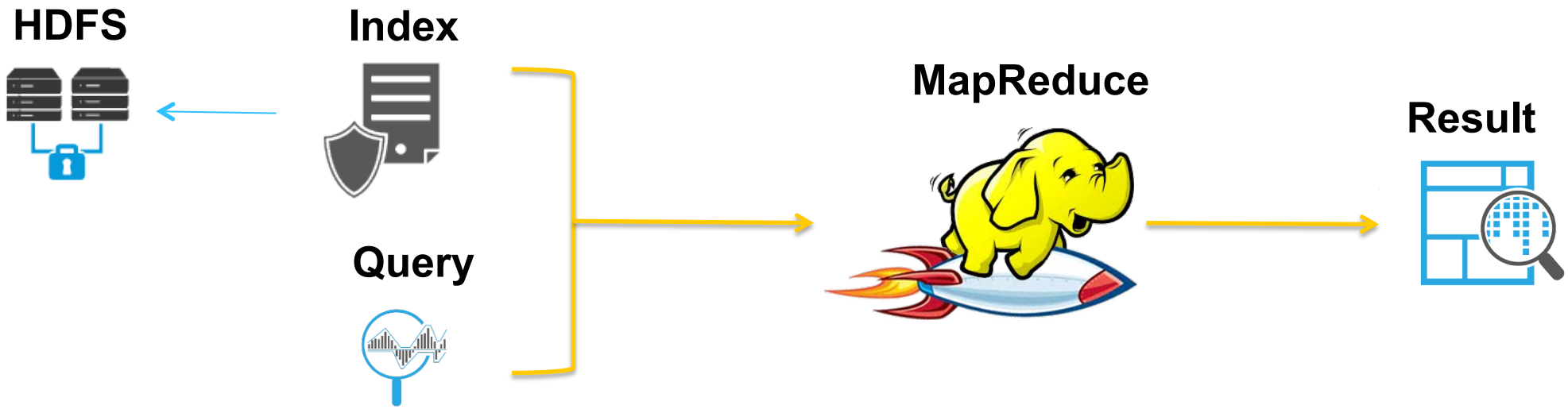
Elasticsearch的资源开销

- 缺省配置下，**shard**数据很少的时候的火焰图



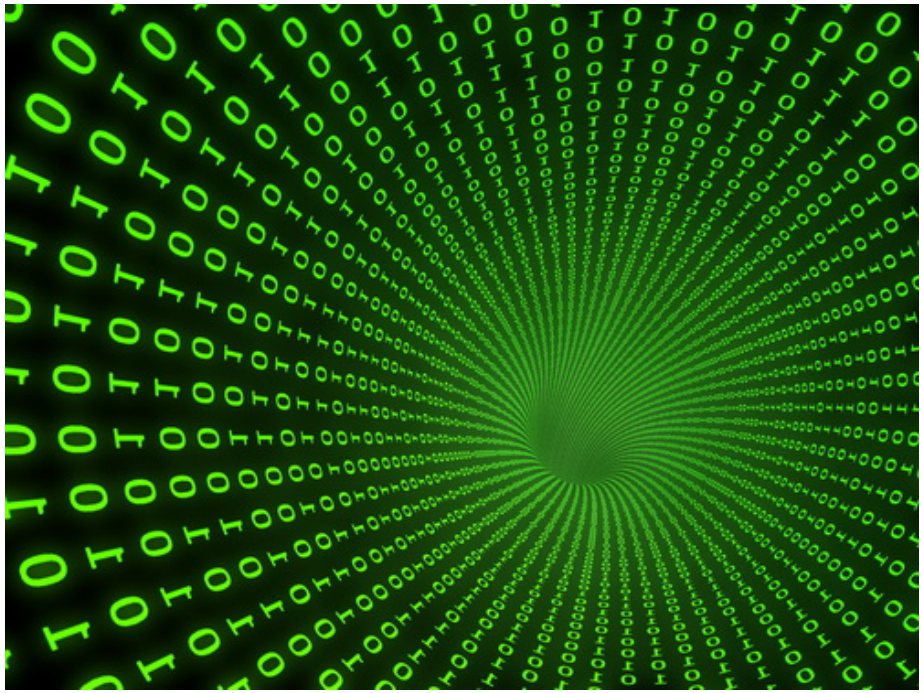
Searching Slow

- Hadoop并行计算
  - 建立好的Index保存在HDFS上
  - 调用Lucene标准的IndexSearcher接口
  - 用Hadoop并行计算查询



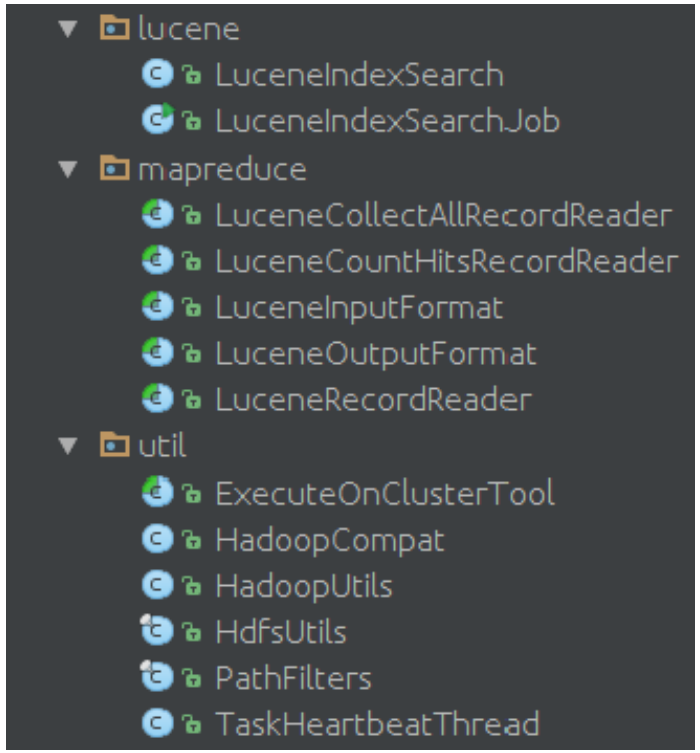
Searching Slow

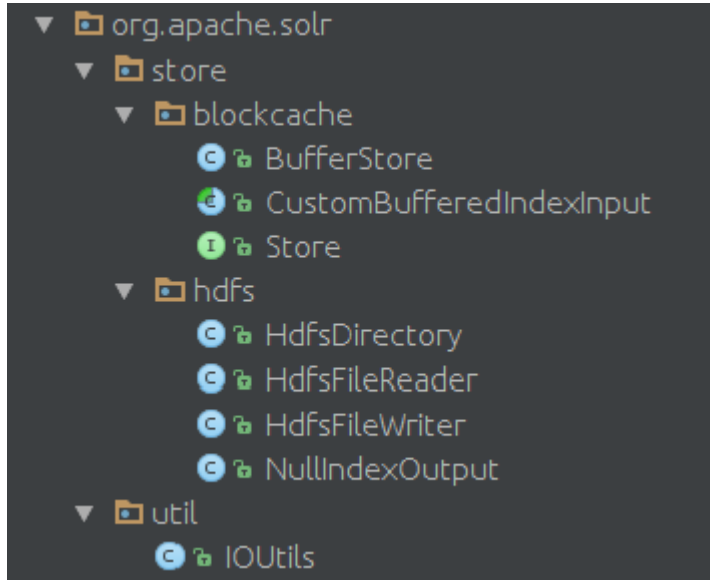
- 代码实现
  - 4.10的Lucene Index
  - Index存储在HDFS上
  - 对query过程MapReduce
  - 调用LuceneSearcher接口



Searching Slow

- 代码结构





Searching Slow

- LuceneIndexSearch

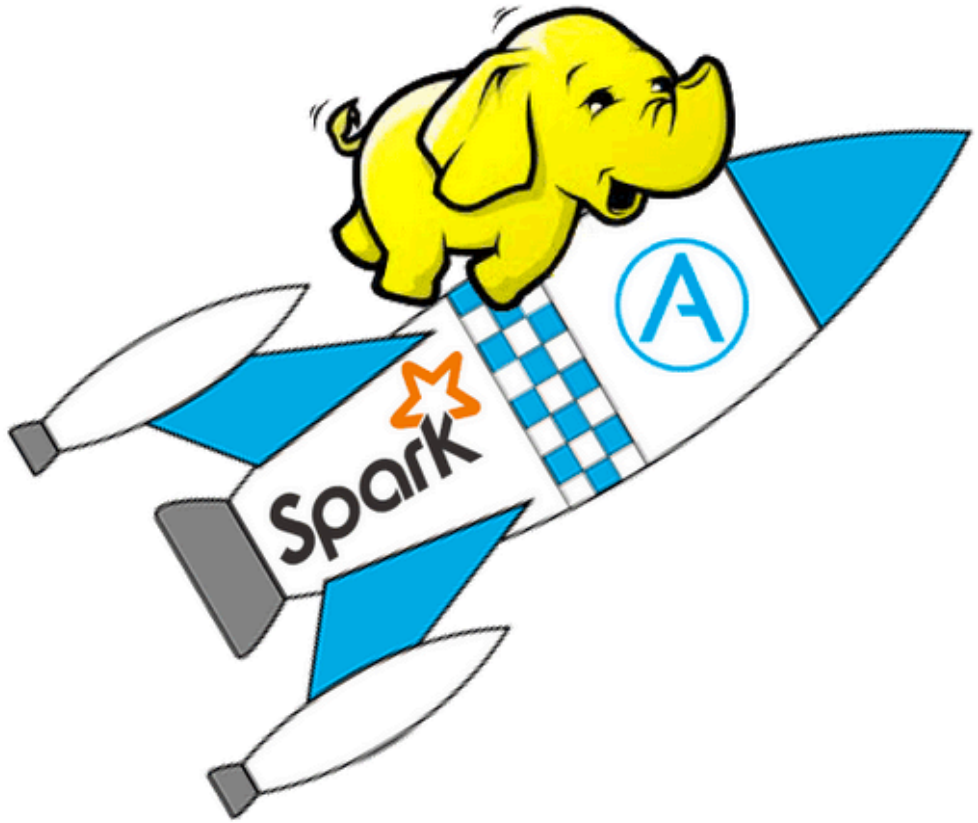
```
public int doSearch(List<Path> inputPaths, Path outputPath, String failureMessage) throws Exception{
    Job job = new Job(super.getConf());
    job.setJarByClass(LuceneIndexSearch.class);


    job.setInputFormatClass(IndexInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    job.setMapperClass(SearchMapper.class);
    job.setReducerClass(SearchReducer.class);

    job.setOutputKeyClass(IntWritable.class);
    job.setOutputValueClass(Text.class);

    IndexInputFormat.setInputPaths(inputPaths, HadoopCompat.getConfiguration(job));

    IndexInputFormat.setQueries(QUERIES, job.getConfiguration());

    TextOutputFormat.setOutputPath(job, outputPath);

    boolean result = job.waitForCompletion(true);
    System.out.println(result);

    new File(new File(outputPath.toString()), "lucene-search-results");

    return result? 0 : -1;

}
```

Searching Slow

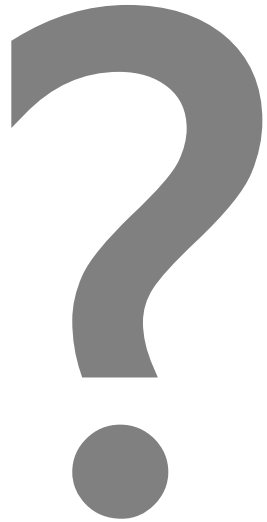
- LuceneInputFormat<T extends Writable> extends InputFormat<IntWritable, T>
- Indexes和query一起，拆分成许多的split
- 每个split里面单独运行查询程序
- 分split按照最大200，最大10GB，为什么？
- 一个Index大小超过10GB怎么办？

Searching Slow

- 限制及解决思路
  - 速度问题：Hadoop换Spark？
  - Aggregation
  - SparkSearch



Q & A



查超 chao_zha@hansight.com

广告时间 - 欢迎加入瀚思

- 瀚思是国内最早专注从事海量日志管理、大数据安全分析的厂商。
- 北京、成都和南京三地都有研发中心
- 技术驱动，Blog http://hansight.com/blog-list-1.html
- 鼓励员工对开源项目做贡献
- 欢迎对算法、安全、大数据平台方面人才加入
- 简历投 hr@hansight.com

- 瀚思获得2015年度中国大数据安全解决方案奖、2015红鲱鱼Red Herring全球100强等多个重量级奖项。

谢谢 | HanSight 瀚思

www.**HanSight**.com

微信公众号：瀚思安信

北京市海淀区中关村软件园9号楼2区306A

成都市天晖路360号晶科1号801室