Hadoop实战

徐冬
frankxus@gmail.com

Agenda

- Hadoop简介
- hdfs
  - 设计原则
  - 特性
  - 系统结构
  - 命令行接口/API
- mapreduce
  - 系统结构
  - 特性
  - APIs

HADOOP 简介

Hadoop

- 开源分布式系统实现
- 完整的mapreduce计算框架实现
- 山寨Google计算系统中的强者

系统	Hadoop组件	Mimic of?
文件系统（DFS）	HDFS	GFS
MapReduce计算框架	MapReduce	MapReduce
锁服务	ZooKeeper	Chubby
RPC	Avro	ProtocolBuffer
高级语言/工作流支持	Hive/Pig/Cascading	Sawzaw*
实时（KV）存储	HBase/HyperTable	BigTable

HDFS简介

- HDFS is designed for
  - 大容量分布式文件系统
  - 支持MapReduce操作（locality）
- HDFS is not designed for
  - 小文件存储
  - 小数据量传输
  - 随机读写

MapReduce简介

- 成熟高效的MapReduce框架
- 众多实用特性
  - Distributed cache
  - 压缩
  - …

设计原则

- 硬件错误是常态而不是异常
- 流式数据访问
- 大规模数据集
- 简单的一致性模型
- “移动计算比移动数据更划算”
- 异构软硬件平台间的可移植性

特性

- 容灾
- 大容量/大吞吐量（水平扩展能力）
- 为mapreduce计算设计的数据本地化能力

实战

- 命令行操作
  - hadoop fs
  - 练习1：命令行操作
- 其他接口
  - Java API
    - FileSystem/DistributedFileSystem
  - 练习2：Java API操作HDFS
  - libhdfs
  - fuse-fs

特性

- 简单一致操作模型
- 容灾
- 并行能力
- Data Locality

MapReduce编程

- 练习1：wordcount – a glance
- APIs
  - 程序逻辑
    - Mapper/Reducer
    - Combiner
    - Partitioner
    - KeyComparator/GroupingComparator
  - 全局数据分发
    - Configuration
    - Distributed Cache

MapReduce编程

- APIs
  - 文件格式/序列化
    - InputFormat/OutputFormat
    - Compression
    - Writable classes
  - 监视
    - Counters
    - Reporter
  - 其它
    - OutputCommitter

程序逻辑

- Mapper

```
/**
 * Counts the words in each line.
 * For each line of input, break the line into words
 * and emit them as(<b>word</b>, <b>1</b>).
 */
public static class MapClass extends MapReduceBase
  implements Mapper<LongWritable, Text, Text, IntWritable> {

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(LongWritable key, Text value,
                  OutputCollector<Text, IntWritable> output,
                  Reporter reporter) throws IOException {
    String line = value.toString();
    StringTokenizer itr = new StringTokenizer(line);
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      output.collect(word, one);
    }
  }
}
```

程序逻辑

- Reducer

```
/**
 * A reducer class that just emits the sum of the input values.
 */
public static class Reduce extends MapReduceBase
  implements Reducer<Text, IntWritable, Text, IntWritable> {

  public void reduce(Text key, Iterator<IntWritable> values,
                     OutputCollector<Text, IntWritable> output,
                     Reporter reporter) throws IOException {
    int sum = 0;
    while (values.hasNext()) {
      sum += values.next().get();
    }
    output.collect(key, new IntWritable(sum));
  }
}
```

程序逻辑

- Combiner
  - IO优化
  - 尽可能的合并
  - 练习2：观察打开/关闭combiner的效果
- Partitioner
  - 分区路由规则

```
/** Partition keys by their {@link Object#hashCode()}. */
public class HashPartitioner<K2, V2> implements Partitioner<K2, V2> {
  public void configure(JobConf job) {}
  /** Use {@link Object#hashCode()} to partition. */
  public int getPartition(K2 key, V2 value,
                          int numReduceTasks) {
    return (key.hashCode() & Integer.MAX_VALUE) % numReduceTasks;
  }
}
```

程序逻辑

- KeyComparator
- GroupingComparator
- 练习3：二次排序

全局数据分发

- Configuration
  - 适合小数据量（配置）的分发
- Distributed Cache
  - 分发配置文件
  - 为重复分发优化（Cache）
- 没有全局变量

文件格式/序列化

- InputFormat/OutputFormat
  - TextInputFormat
  - SequenceFileInputFormat
  - 练习4：设置输出文件格式为sequencefile并查看结果
- Writable Classes

文件格式/序列化

- Compression
  - 配置（依赖具体版本）
    - mapred.compress.output
    - mapred.compress.map.output
  - Codecs （依赖具体版本）
    - io.compression.codecs
    - mapred.output.compression.codec
    - …

监视

- Counters
  - 全局动态更新计数器
  - 精确性？
  - 控制数量
- Reporter

其他

- OutputCommiter
  - FileOutputCommiter
  - DBOutputCommiter

其他接口

- Streaming
  - I/O重定向的原理支持其他语言编写mapreduce逻辑
  - 练习5：用脚本语言重写wordcount
  - 练习6：使用-reduce aggregate选项简化编程
- 高级抽象
  - Pig
  - Hive
  - Cascading
  - …

Back up slides…

高级议题

- Hadoop Master瓶颈
  - NameNode
    - Namespace & Block map内存上限：支持的文件数不是无限的
    - 请求响应：lsr/fsck/…
  - JobTracker
    - 调度器
      - 大任务的影响
    - 密集计算请求
      - Job History?
      - Counters

高级议题

- MR任务优化
  - Writable object reuse
  - IO优化
    - 使用Combiner
    - HashMap combiner
    - 压缩
  - 合理配置参数
    - Map/reduce number
    - 排序消耗内存
    - Shuffle进程数？
    - 运行后参数调优：vaidya
  - 使用高级抽象 - I mean hive
  - 优化业务逻辑

高级议题

- MR任务调试
  - Mapper/Reducer
    - Task log
    - Counters
    - Jdb through JPDA（运行在Local模式下）
  - Streaming
    - 本地调试
  - InputFormat/RecordReader/…