1.

**并行和并发区别**

1、并行是指两者同时执行一件事，比如赛跑，两个人都在不停的往前跑；  
2、并发是指资源有限的情况下，两者交替轮流使用资源，比如一段路(单核CPU资源)同时只能过一个人，A走一段后，让给B，B用完继续给A ，交替使用，目的是提高效率

内存泄漏

<http://blog.csdn.net/gzh0222/article/details/8538727>

<http://blog.csdn.net/yhyr_ycy/article/details/52566105>

多级bolt之间传递数据，因数据类型多样，所以将数据序列化

<http://blog.chinaunix.net/uid-28379365-id-5015198.html>

spark

<http://blog.csdn.net/u010330043/article/details/52293212>

yarn

<http://dongxicheng.org/mapreduce-nextgen/understand-yarn-container-concept/>

yarn on spark

<http://blog.csdn.net/fansy1990/article/details/54314249>

**ConcurrentHashMap的锁分段技术**

HashTable容器在竞争激烈的并发环境下表现出效率低下的原因，是因为所有访问HashTable的线程都必须竞争同一把锁，那假如容器里有多把锁，每一把锁用于锁容器其中一部分数据，那么当多线程访问容器里不同数据段的数据时，线程间就不会存在锁竞争，从而可以有效的提高并发访问效率，这就是ConcurrentHashMap所使用的锁分段技术，首先将数据分成一段一段的存储，然后给每一段数据配一把锁，当一个线程占用锁访问其中一个段数据的时候，其他段的数据也能被其他线程访问。

<http://ifeve.com/concurrenthashmap/>

linux启动jar

<http://www.cnblogs.com/ahudyan-forever/p/5772016.html>

spark优势:

提供了一套支持DAG图的分布式并行计算的编程框架，减少多次计算之间中间结果写到Hdfs的开销

提供Cache机制来支持需要反复迭代计算或者多次数据共享，减少数据读取的IO开销

使用多线程池模型来减少task启动开稍，shuffle过程中避免不必要的sort操作以及减少磁盘IO操作

广泛的数据集操作类型

1.从 Application 启动的角度来观察BlockManager

在 Application 启动的时候会在 spark-env.sh 中注册 BlockMangerMaster 以及 MapOutputTracker，其中：

BlockManagerMaster：对整集群的 Block 数据进行管理；

MapOutputTracker：跟踪所有的 Mapper 的输出；

BlockManagerMasterEndpoint 本身是一个消息体，会负责通过远程消息通信的方式去管理所有节点的 BlockManager；

每个启动一个 ExecutorBackend 都会实例化 BlockManager 并通过远程通信的方式注册给 BlockMangerMaster；实际上是 Executor 中的 BlockManager 注册给 Driver 上的 BlockMangerMasterEndpoiont；(BlockManger 是 Driver 中的一个普通的对象而己，所以无法直接对一个对象做HA)

MemoryStore 是 BlockManager 中专门负责内存数据存储和读写的类，MemoryStore 是以 一个又一个 Block 为单位的

DiskStore 是 BlockManager 中专门负责磁盘数据存储和读写的类；

DiskBlockManager：管理 LogicalBlock 与 Disk 上的 PhysicalBlock 之间的映射关联并负责磁盘的文件的创建，读写等;