## RMI (Remote method invocation)

远程方法调用 ejb基于此协议

## RPC (Remote Procedure Call)

远程过程调用 dubbo基于此协议

EJB

部署分布式应用程序 基于rmi实现

JMS (java message service)

java队列消息协议 消息由头(head) 属性(property) 主体(body) 组成支持两种消息传递模型: 点对点(ptp) 和 发布/订阅(pub/sub) 简单来说是一对一传递和一对多传递ActiveMQ基于此协议

## ActiveMQ

消息队列支持jms provider实现，支持 in-vm(管道通讯)，tcp,ssl,nio,Jgroup,jxta多种传输协议

## SOA

是一种粗粒度、开放式、松耦合的服务架构，一个面向服务的架构样式,支持异构,支持变化可理解为成型软件的oop思想 oop是开发的范式说得通俗一点就是，系统中分为三种角色：服务提供者、服务使用者和注册中心，提供者发布服务到注册中心，使用者通过注册中心发现所需服务，然后与该服务的提供者绑定，并调用服务。其实就是接口开发思想

## MAVEN

Maven是一个项目管理工具，包含项目对象模型(粗略理解为项目级的oop思想)，对于mavne来说 一个软件工程的唯一标识是由开发者(GroupId)、项目名(artifactid)、版本号(version)共同决定，每个工程都可以打包为jar、war、ear、或pom(打包成pom是给其他项目继承所用，一般将parent项目打包成pom)。

maven有聚合和继承两个概念，聚合就是我知道你们，你们不知道我，继承是你们都知道我，而我不知道你们。

maven有三个独立的生命周期 构建(build),清理(clean),建立项目发布站点(site)

maven基本命令:

install:将包安装到本地仓库,供其他mavne使用(其他maven引用的都是本地仓库的项目.jar)

clean:清除target目录下的文件(编译后的文件)

compile：编译项目中的代码

deploy：将包发布到远程的maven仓库，并提供给其他开发者使用。

package:将项目打包到项目的target目录下

## GIT

一个开源分布式版本控制系统，有工作区(日常编辑的部分),暂存区(git add之后的部分),版本库(git commit后的部分),HEAD(当前版本指针) 常用指令:git pull origin master/git push origin master git add \*,git commit,git --reset hard(放弃本地所有修改)git branch xxx(切换分支)

## SVN

项目版本控制，协同开发，资料备份的工具，每次commit前先update

SVN和GIT的区别:

git是分布式 svn不是GIT跟SVN一样有自己的集中式版本库或服务器。但，GIT更倾向于被使用于分布式模式，也就是每个开发人员从中心版本库/服务器上chect out代码后会在自己的机器上克隆一个自己的版本库。可以这样说，如果你被困在一个不能连接网络的地方时，就像在飞机上，地下室，电梯里等，你仍然能够提交文件，查看历史版本记录，创建项目分支

GIT把内容按元数据方式存储，而SVN是按文件

git分支和svn分支不同(git切换分支方便快捷)  
 GIT没有一个全局版本号，svn有

GIT的内容完整性要优于SVN，因为GIT的内容存储使用的是[SHA-1](http://en.wikipedia.org/wiki/SHA-1" \t "http://blog.jobbole.com/31444/_new)哈希算法。这能确保代码内容的完整性，确保在遇到磁盘故障和网络问题时降低对版本库的破坏

## Socket

理解为网络上的io流，负责请求接收网络信息，tcp/ip协议的实现，处于osi的表现层和传输层之间的抽象层,负责网络通信

## Mina

一个基于Socket的封装网络通信框架，对c端s端都进行了封装，用于开发高性能，高可扩展性的网络应用程序,采用java 1.7新特性NIO，能自动发送心跳包(判断连接是否断开),能自动监听客户端断开

## mysql

c和c++编写，关系型数据库 轻量级开源 没啥特别的

innodb和myisam引擎区别

innodb 支持事务功能，myisam 不支持。

Myisam 的执行速度更快，性能更好。

InnoDB 为行级锁，myisam 为表级锁。

## Hibernate和Mybatis

hibernate是对象关系映射的持久层框架，对jdbc进行封装，完全遵循JPA的实现规范，掌握较慢

mybatis 是轻量级orm持久层框架，掌握较快  
hibernate是全自动化，mybatis是半自动化:

hibernate完全可以通过对象关系模型实现对数据库的操作，拥有完整的JavaBean对象与数据库的映射结构来自动生成sql。而mybatis仅有基本的字段映射，对象数据以及对象实际关系仍然需要通过手写sql来实现和管理。

hibernate数据库移植性远大于mybatis:

hibernate加强了对象和数据库的映射，降低了耦合度而mysql的sql则耦合度较高

hibernate拥有完整的日志系统，mybatis则欠缺一些：

hibernate日志系统非常健全，涉及广泛，包括：sql记录、关系异常、优化警告、缓存提示、脏数据警告等；而mybatis则除了基本记录功能外，功能薄弱很多。

从sql优化的角度讲，mybatis要比hibernate要好

mybatis可直接控制sql，灵活性强，但同时也降低了开发的效率，对自己的sql有信心的人 mybatis的效率远高于hibernate

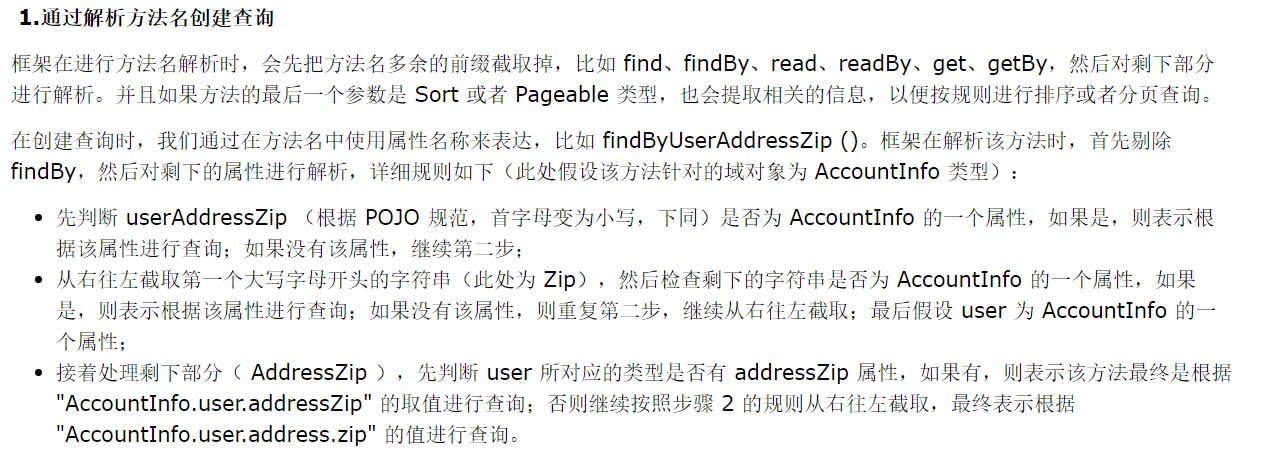
## Spring Boot

Spring web开发的一个子框架，强调约定大于配置来实现快速开发，该框架不需要以往xml配置，可达到引入jar包则用，如需定制一些配置可以配置在application.properties配置文件里。

spring boot 多用于创建独立的微服务

## Spring data jpa

spring data 下的一个jpa实现项目，严格遵循jpa标准，简化操作，实现repository<id类型,映射对象>(crudRepository)接口 在domain里面加上注解(和hibernate一样) 基于crud的操作不用再写，如需定制 ，findByUserId(String id) 意思就是 解析的时候先解析findBy后面的userid(整体) 如果该对象有这字段，则根据userid去查 参数即id ，如果里面没有userid属性，则从右往左解析到大写字母停止(Id)



## Spring mvc

轻量级web框架 没啥好说的 支持restful风格注解

## MongoDb 和 redis区别

都是非关系型数据库，不过mongodb更接近mysql 有索引，游标

mongodb新版本开始支持事务,redis也支持事务 不过比较弱，在大量的数据面前，redis打不过Mongodb 但这都不是瓶颈，都可以持久化，

mongodb 查询海量数据性能很好，可适用于大数据的存储，mongodb有空间索引，mongodb 集群比redis成熟

## SwaggerUi

为spring mvc接口提供可视化文档的框架,能方便的对接口进行调试，没有很强的依赖性，能在多种平台主机上运行

## JQuery

一个轻量级的封装js的框架,不影响页面加载速度(太多),封装了选择器查找简化了css代码 使可读性增强,jquery还有很多第三方插件，树图，日期，图片等，同时，jquery的扩展很强，提供了扩展接口JQuery.extend(object) ，jquery的所有插件都是基于这个

## Struts2

首先，这是一个view层的框架，并不是struts的升级版，吸取了struts的思想，s1使用继承抽象来编程，s2使用接口来编程，可以定制服务。s1必须要保证线程安全，因为仅有一个Action的实例来处理所有的请求，s2默认是一个action一个实例，可设置，s1是传统的jsp机制传值，s2在action里加入了valueStack 可以使用taglib直接取值

## lucene

lucene 是一个开放源代码的全文检索引擎工具包，但它不是一个完整的全文检索引擎，而是一个全文检索引擎的架构，提供了完整的查询引擎和索引引擎，部分[文本分析](http://baike.baidu.com/view/3488135.htm" \t "http://baike.baidu.com/_blank)引擎

五个核心类：

indexWriter 索引过程的核心组件，用于创建新的索引并且把文档加入已有索引中

Directory 描述索引存放位置，是一个抽象类，具体的两个子类提供了特定的存储索引方法，一个是FSDirectory，它在文件系统中的目录创建索引文件。另一个子类是RAWDirectory，它是把索引数据保存在了内存中。

Analyzer 分析器 它负责从将被索引的文本文件中提取语汇单元（tokens），并剔除剩下的无用信息。如果被索引的文件不是纯文本，那么就需要先将其转换为文本。

Document 承载数据的实体，一个索引的基本单元，一个文档代表一堆域（filed）的集合 域(filed)代表文档和文档相关的元数据(最小数据单位)

lucene 只能处理文本 因为Lucene的内核本身只处理String 和 Reader两个对象

## solr

solr是一个基于lucene开发的一个分布式搜索引擎，支持restful风格，提供http请求访问接口，通过xml文件交互

## elatecsearch

elatecsearch一个基于lucene开发的一个分布式搜索引擎，接近实时搜索，在对已有的数据搜索的时候，solr更快。当实时建立索引时, Solr会产生io阻塞，查询性能较差, Elasticsearch具有明显的优势。随着数据量的增加，so效率会降低，而es没什么变化。solr利用zookeeper进行分布式管理，es自身带有分布式管理，solr支持多格式的数据，而es只支持json格式

## OCR

[光学字符识别](http://baike.baidu.com/view/230331.htm" \t "http://baike.baidu.com/_blank)，通过扫描图片可以将其中的文字读取出来，tesseract是java的一个框架，对中文的扫描准确度不太高

## Nginx

高性能的HTTP和反向代理服务器，俄罗斯开发，正常执行的Nginx是多进程(一个master进程,多个work进程，还有缓存加载进程和缓存管理进程)，所有进程都是单线程的。当然Nginx也可以单进程实现(不要监控)

FastCGI是一个可伸缩地、高速地在HTTP server和动态脚本语言间通信的接口

Nginx不支持对外部程序的直接调用或者解析，所有的外部程序（包括PHP）必须通过FastCGI接口来调用。FastCGI接口在Linux下是socket（这个socket可以是文件socket，也可以是ip socket）。

# java基础篇

## java内存模型

jvm包括三个部分，堆内存(存放java对象)，非堆内存(存放类加载信息和其他meta-data),其他(jvm自身代码)

默认空余堆内存小于40%，jvm会增大堆直到-xmx(最大内存值)，由-xx:MinHeapFreeRatio指定

默认空余堆内存大于70%，jvm会减少堆到-xms(初始化最小内存值)，由-xx:MaxHeapFreeRatio指定

默认新生代,老年代的区域大小比是1:2



## jvm内存申请

1jvm会试图为相关java对象在Eden中初始化一块内存区域

如果Eden空间足够的时候，内存申请结束，否则进行下一步

jvm会试图释放在Eden中所有不活跃的对象(>=1级的垃圾回收)，释放后若Eden空间仍不足以放入新对象，则试图将部分Eden中活跃对象放入Suvivor区

Survivor区用来作为Eden及OLD的中间交换区域，当OLD区空间足够，Survivor区的对象会被移到OLD区，否则会保留在Survivor区

当OLD区空间不够时，jvm会在OLD进行full gc(0级)垃圾收集

完全垃圾收集后，若Survivor及OLD区仍然无法存放从Eden复制过来的部分对象,则会出现out of memory错误

## jvm内存模型规则

不允许read和load、store和write操作之一单独出现

不允许一个线程丢弃它的最近assign的操作，即变量在工作内存中改变了之后必须同步到主内存中。

不允许一个线程无原因地（没有发生过任何assign操作）把数据从工作内存同步回主内存中。

一个新的变量只能在主内存中诞生，不允许在工作内存中直接使用一个未被初始化（load或assign）的变量。即就是对一个变量实施use和store操作之前，必须先执行过了assign和load操作。

一个变量在同一时刻只允许一条线程对其进行lock操作，lock和unlock必须成对出现

如果对一个变量执行lock操作，将会清空工作内存中此变量的值，在执行引擎使用这个变量前需要重新执行load或assign操作初始化变量的值

如果一个变量事先没有被lock操作锁定，则不允许对它执行unlock操作；也不允许去unlock一个被其他线程锁定的变量。

对一个变量执行unlock操作之前，必须先把此变量同步到主内存中（执行store和write操作）。

# 连接池

当connect对象调用close的时候 代理对象会拦截到该方法请求(method.equals(“close”))然后将当前连接放入连接池中,以备后续使用

# 线程池

初始化创建多个线程,用循环判断任务队列是否为null,如果不为空(有新任务)则用调用空线程来执行

任务实现runnable接口

# NAT技术

静态转换:私有IP转公网IP 1对1

动态转换 私有IP转公网IP 多对多

端口多路复用 私有IP转公网IP 多对一

# NAT穿透

<http://www.cnblogs.com/regedit/articles/829725.html>

三次握手 户机首先发出一个SYN消息，服务器使用SYN+ACK应答表示接收到了这个消息，最后客户机再以[ACK](http://baike.baidu.com/view/204040.htm" \t "http://baike.baidu.com/_blank)消息响应。这样在客户机和服务器之间才能建立起可靠的TCP连接，数据才可以在客户机和[服务器](http://baike.baidu.com/view/899.htm" \t "http://baike.baidu.com/_blank)之间传递。

# Reactor 和 **Proactor的区别**

Reactor是同步，由处理器实际IO操作。

Proactor是异步(真异步),处理器异步调用IO操作，由系统内核实现IO(需系统支持)

# IO和NIO

同步阻塞

同步非阻塞

异步阻塞

异步非阻塞