

性能 VIP 课程基础知识总篇

版本	编撰	类型	作者	时间
V1.0	新增课程基础知识总篇	N-新增 M-修改	Allen	2020-2-14
V1.1		M-修改	Allen	2020-6-11

@湖南零檬信息技术有限公司
创新教育，极致服务

目录

性能 VIP 课程基础知识总篇.....	1
基础知识总篇.....	3
● Vmware.....	3
● Linux 基本技能.....	3
● CPU 工作原理.....	3
● 内存知识.....	4
● JVM 知识.....	4
● Apache 基础知识.....	6
● Tomcat 学习资料:	6
● Docker 基础知识.....	8
● K8s 学习资料.....	9
● 微服务.....	10
● Dubbo + zookeeper.....	10
● Nginx 学习资料.....	11
● 负载均衡.....	12
● 集群知识.....	13
● 关系型数据库.....	14
● Mysql 数据库基础知识.....	14
● Postgresql 数据库知识.....	15
● 非关系型数据库.....	15
● Redis 基础知识.....	16
● Memcache 学习资料.....	16
● Mongodb 基础知识.....	17
● 时序型数据库.....	17
● Influxdb 基础知识.....	17
● Prometheus 基础知识.....	18
● 程序员必须掌握的英语单词.....	18

基础知识总篇

Vmware

重要程度：★★★★★

简介：

Vmware 是一款虚拟机管理软件，允许一台真实的电脑在一个操作系统中同时打开并运行多个不同的操作系统。



01.vmware安装.
pdf

Linux 基本技能

重要程度：★★★★★

简介：

推荐学习地址：

鸟哥私房菜：<https://www.showapi.com/book/view/2089/0>
<https://linux.cn/>

CPU 工作原理

重要程度：★★★★★

简介：

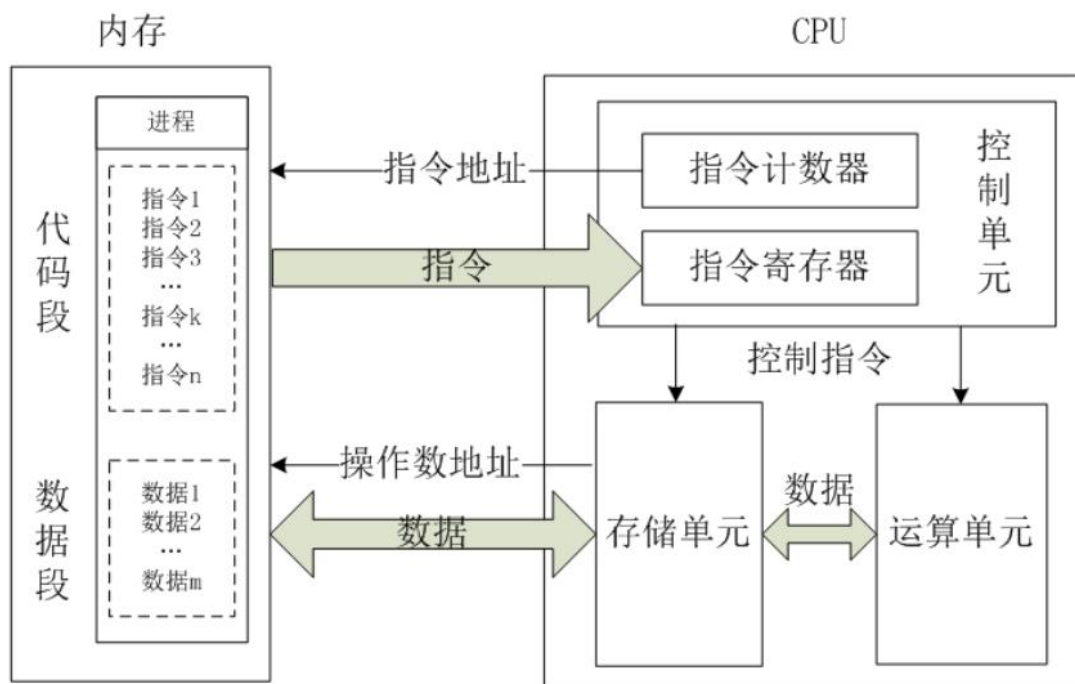
Cpu，中央处理器。影响 cpu 性能的主要因素分两大块：主频、架构；

Cpu 的结构主要由运算器、控制器、寄存器三大块组成

控制单元：是整个 cpu 的指挥控制中心，由指令寄存器 IR(Instruction Register)、指令译码器 ID(Instruction Decoder)和操作控制器 OC(Operation Controller)等

运算单元：是运算器的核心，执行运算操作

存储单元：包括 CPU 片内缓存和寄存器组



参考学习文章：

<https://baijiahao.baidu.com/s?id=1619812694399963397&wfr=spider&for=pc>

<http://blog.chinaunix.net/uid-23069658-id-3563960.html>

<https://cloud.tencent.com/developer/article/1498759>

内存知识

重要程度：★★★

简介：

内存 memory，是 cpu 与其他设备沟通的桥梁，主要用来临时存放数据，配合 cpu 工作，协调 cpu 的处理速度。

参考学习资料：https://blog.csdn.net/qq_39312683/article/details/96026884

JVM 知识

重要程度：★★★★

简介

JVM 是 **Java Virtual Machine**（Java 虚拟机）的缩写，JVM 是一种用于计算设备的规范，它是一个虚构出来的能够运行 **Java bytecode** 的虚拟计算机，是通过在实际的计算机上仿真模拟各种计算机功能来实现的。

Java 虚拟机有自己**完善的硬件架构**（如处理器、堆栈、寄存器等），还具有**相应的指令系统**。

Java 虚拟机**本质上就是一个程序**，当它在命令行上启动的时候，就开始执行保存在文件中的字节码指令。Java 语言的**可移植性**正是**建立在 Java 虚拟机**的基础上。任何平台只要装有针对性该平台的 **Java 虚拟机**，字节码文件（.class）就可以在该平台上运行。

Java 虚拟机主要分为五大模块：**类装载器子系统、运行时数据区、执行引擎、本地方法接口和垃圾收集模块**。

Java 虚拟机没有寄存器，所以指令集是使用 **Java 栈**来存储中间数据。

对于 Java 运行时涉及到的存储区域主要包括：**程序计数器、Java 虚拟机栈(调用栈)、本地方法栈、java 堆、方法区以及直接内存等等**。

程序计数器：主要是取下一条指令，在 Java 里面主要是取下一条指令的字节码文件；

Java 虚拟机栈：主要是利用**栈先进后出**的特性存储局部变量表，动态链接等，主要包括堆内存和栈内存，对于程序员内存分析而言是特别重要的。

本地方法栈：与上边的栈基本作用差不多，只不过这里是为 Java 方法而服务。

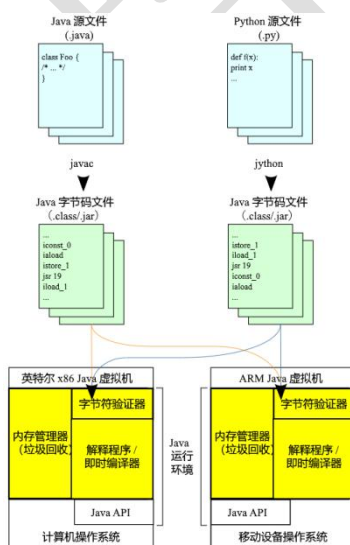
Java 堆：是内存管理中最大的一块，所有的线程共享这一块内容，同时该部分也是垃圾收集器的主要区域。

Java 虚拟机的垃圾回收，**动态分配和回收**。

内存回收首先就是**判断某一个部分是生存还是死亡**，具体的算法常见的有二种：

1、引用计数算法；

2、是否可达性分析算法，该算法核心思想是依靠判断对象是否存活，通过一系列的 GC(垃圾收集 garbage collection) ROOTS 的对象作为起始点，采用搜索的算法遍历引用链，如果搜索过程中没有发现该节点，则认为该节点是不可达的，即可回收的，在 Java 里面，一般可以使用该算法处理问题。



Gc 学习参考文章：

<https://baijiahao.baidu.com/s?id=1651319476120132392&wfr=spider&f>

[or=pc](#)

<https://www.jianshu.com/p/2f4a8e04c657>

<https://cloud.tencent.com/developer/information/JVM%E7%9F%A5%E8%AF%86>

Apache 基础知识

重要程度：★★★★★

简介：

Apache HTTP Server（简称 Apache）是 Apache 软件基金会的一个开放源码的网页服务器软件，可以在大多数电脑操作系统中运行。由于其跨平台 and 安全性，被广泛使用，是最流行的 Web 服务器软件之一。它可以运行在几乎所有广泛使用的计算机平台上。

LAMP 是一组通常一起使用来运行动态网站的自由软件的首字母缩写：
Linux+Apache+Mysql(MariaDB)+PHP

参考学习资料：

官网：<https://httpd.apache.org/>

文档：<https://httpd.apache.org/docs/2.4/>

Ubuntu 系统中 apache 安装指南：<https://wiki.ubuntu.org.cn/Apache>

Centos 系统中搭建 apache 环境：

https://blog.csdn.net/weixin_42199590/article/details/94410946

<https://www.cnblogs.com/xiangqs/p/8663947.html>

Tomcat 学习资料：

重要程度：★★★★★

简介：

是一个轻量级的 web 应用服务区，使用于并发性不是很高的系统中。Tomcat 中最顶层的是 server，代表整个服务器，一个 server 可以包含至少一个 service，每个 service 可以包含多个 connector 和一个 container。Connector 用于处理连接相关的事情，并提供 Socket 与 Request 和 Response 相关的转化；Container 用于封装和管理 Servlet，以及具体处理 Request 请求；多个 connector 就可以配置多种类型连接，如 http\https。



Tomcat 的 conf 目录中的 server.xml 是其最重要的配置文件

server - 8.5.51.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8005" shutdown="SHUTDOWN">
  1 <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
      type="org.apache.catalina.UserDatabase"
      description="User database that can be updated and saved"
      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
      pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  2 <Service name="Catalina">
    3 <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" />
    <Connector protocol="AJP/1.3"
      address=":::1"
      port="8009"
      redirectPort="8443" />
    4 <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
          resourceName="UserDatabase" />
      </Realm>
      <Host name="localhost" appBase="webapps"
        unpackWARs="true" autoDeploy="true">
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
          prefix="localhost_access_log" suffix=".txt"
          pattern="%h %l %u %t &quot;%r&quot; %s %b" />
      </Host>
    </Engine>
  </Service>
</Server>
```

tomcat 连接池配置:

connector 中

connectionTimeout:连接超时,毫秒为单位.对于高并发对实时要求不高的可以使适当增加该值

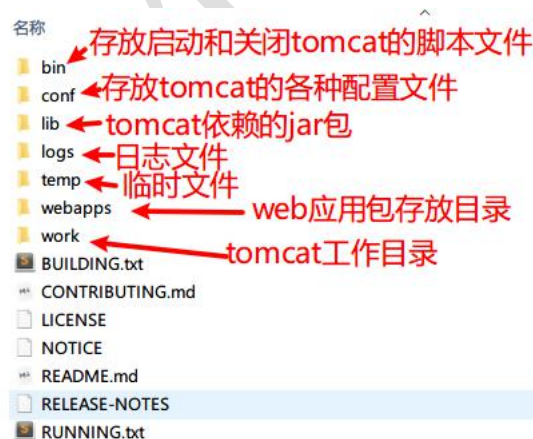
maxThreads:最大并发连接数

acceptCount:在超过最大连接数后,可以接受的排队数量

minSpareThreads:Tomcat 初始化时默认创建的线程数,也是以后线程增加时一次增加的最小数量

maxSpareThreads:这个参数标识,一旦创建的线程数量超过这个值,Tomcat 就会关闭不活动的线程

enableLookups:关闭 DNS 查询



Tomcat 的 JVM 配置

修改 bin 目录中 catalina.sh 文件

JAVA_OPTS="-Xms256m -Xmx512m -Xss1024K -XX:PermSize=128m -XX:MaxPermSize=256m"

- Xms 为 jvm 启动时分配的初始内存 比如-Xms200m，表示分配 200M
- Xmx 为 jvm 运行分配的最大内存 比如-Xmx500m，表示 jvm 进程最多只能占
用 500M 内存
- Xss 每个线程堆栈的大小 一般情况下 256K 是足够了。影响了此进程中
并发线程数大小
- XX PermSize=64M JVM 初始分配的非堆内存
- XX MaxPermSize=128M JVM 最大允许分配的非堆内存，按需分配

参考学习：

<https://how2j.cn/k/tomcat/tomcat-tutorial/541.html>

tomcat7

<https://tomcat.apache.org/tomcat-7.0-doc/index.html>

tomcat8.5

<https://tomcat.apache.org/tomcat-8.5-doc/index.html>

tomcat9

<http://tomcat.apache.org/tomcat-9.0-doc/index.html>

<https://www.cnblogs.com/chz-blogs/p/11448635.html>

https://segmentfault.com/a/1190000013122831?utm_source=tag-newest

<https://www.cnblogs.com/centos2017/p/9956432.html>

Linux+tomcat 环境安装学习参考：

<https://blog.csdn.net/fukai8350/article/details/80467224>

Tomcat 性能调优：

<https://www.cnblogs.com/kismetv/p/7806063.html>

Docker 基础知识

重要程度：★★★★★

简介：

Docker 是一个开源的**应用容器引擎**，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 Linux 机器或 Windows 机器上，也可以实现虚拟化，容器是完全使用沙箱机制，相互之间不会有任何接口。

Docker 使用客户端-服务器 (C/S) 架构模式，使用远程 API 来管理和创建 Docker 容器。Docker 容器通过 **Docker 镜像** 来创建。

docker 是基于 **LXC** 的轻量级虚拟化，相比 **KVM** 启动更快，占用资源更少。

LXC:Linux Container 容器是一种操作系统层虚拟化技术，可以提供轻量级的虚拟化，以便隔离进程和资源。

KVM:是 **Kernel-based Virtual Machine** 基于内核的虚拟机,是一种用于 **Linux** 内核中的虚拟化基础设施，可将 **Linux** 内核转化为一个虚拟机监视器。

Docker 并不是全能的,也不是 **KVM** 之类虚拟化手段的替代品。**Docker** 是基于 **Linux 64bit** 的，无法在 **32bit** 的 **linux/Windows/unix** 环境下使用。

Docker 在本质上是一个附加系统。

Docker 有两种文件格式，**Dockerfile** 和 **Compose file**。**Dockerfile** 定义了单个容器的内容和启动时候的行为。**Compose file** 定义了一个多容器应用。

Docker 可以按照 **Dockerfile** 的内容，构建镜像。

Compose file 是一个 **YAML** 文件，定义了服务（**service**）、网络、卷（**volume**）

推荐学习材料：



docker.pdf

参考学习资料：

<https://www.runoob.com/docker/docker-tutorial.html>

K8s 学习资料

重要程度：★★★★

简介：

K8s 是 **kubernetes** 的简写。是一个开源的，用于管理云平台中多个主机上的容器化的应用，**Kubernetes** 的目标是让部署容器化的应用简单并且高效（**powerful**），**Kubernetes** 提供了应用部署，规划，更新，维护的一种机制。**Kubernetes** 是 **Google** 开源的一个容器编排引擎，它支持自动化部署、大规模可伸缩、应用容器化管理。

参考学习资料：

<https://www.kubernetes.org.cn/k8s>

<https://kubernetes.io/docs/tutorials/kubernetes-basics/>

微服务

重要程度：★★★★

简介：

微服务 (Microservices) 是一种软件架构风格，它是以专注于单一责任与功能的小型功能区块 (Small Building Blocks) 为基础，利用模块化的方式组合出复杂的大型应用程序，各功能区块使用与语言无关 (Language-Independent/Language agnostic) 的 API 集相互通信。

参考学习资料：https://www.sohu.com/a/322139140_100109711

Dubbo + zookeeper

重要程度：★★★★

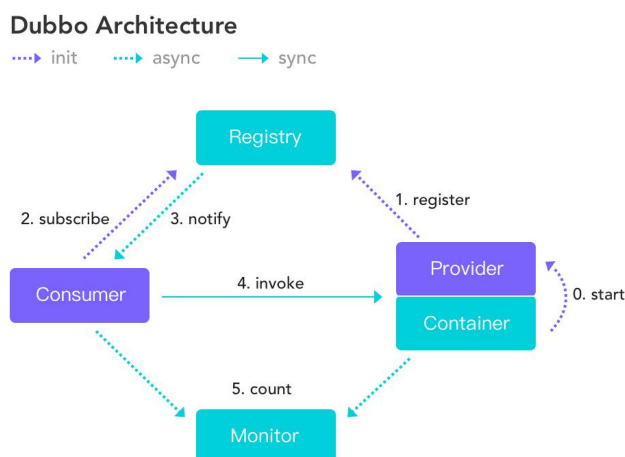
简介：

Dubbo 是一款高性能、轻量级的开源 java RPC 框架，它提供了三大核心能力：面向接口的远程方法调用，智能容错和负载均衡，以及服务自动注册和发现。

推荐学习地址：

<http://dubbo.apache.org/zh-cn/>

知识描述：



Dubbo 缺省协议采用单一长连接和 NIO 异步通讯，适合小数量大并发的服务调用，以及服务消费者机器数远大于服务提供者机器数的情况。

Dubbo 接口，手工测试时，可以使用 telnet 方式：

<http://dubbo.apache.org/zh-cn/docs/user/references/telnet.html>

Zookeeper 注册中心: zookeeper 是一个树型的目录服务，支持变更推送，适合作为 dubbo 服务的注册中心，工业强度高，可用于生产环境，并推荐使用。

参考学习材料：

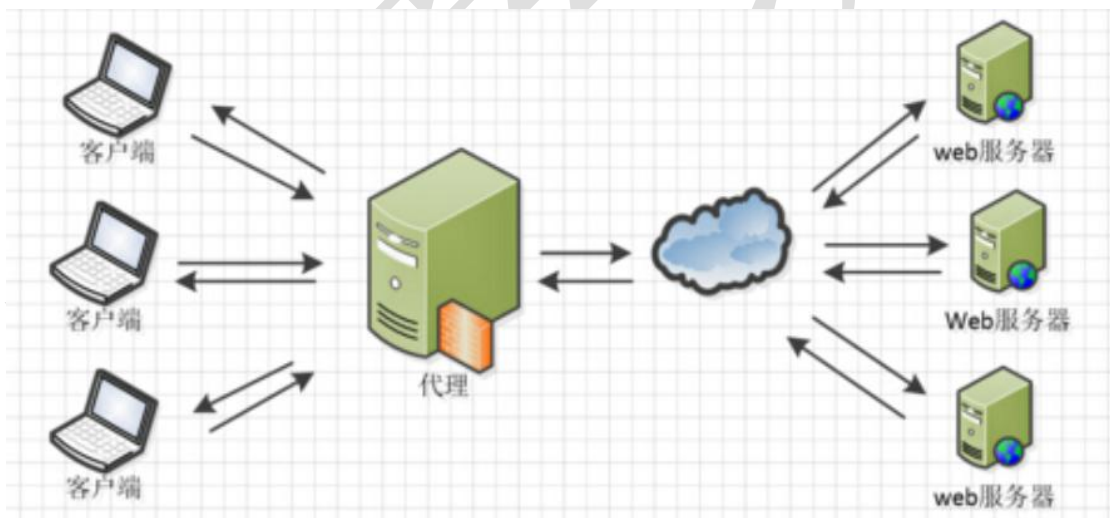
<http://dubbo.apache.org/zh-cn/docs/user/quick-start.html>

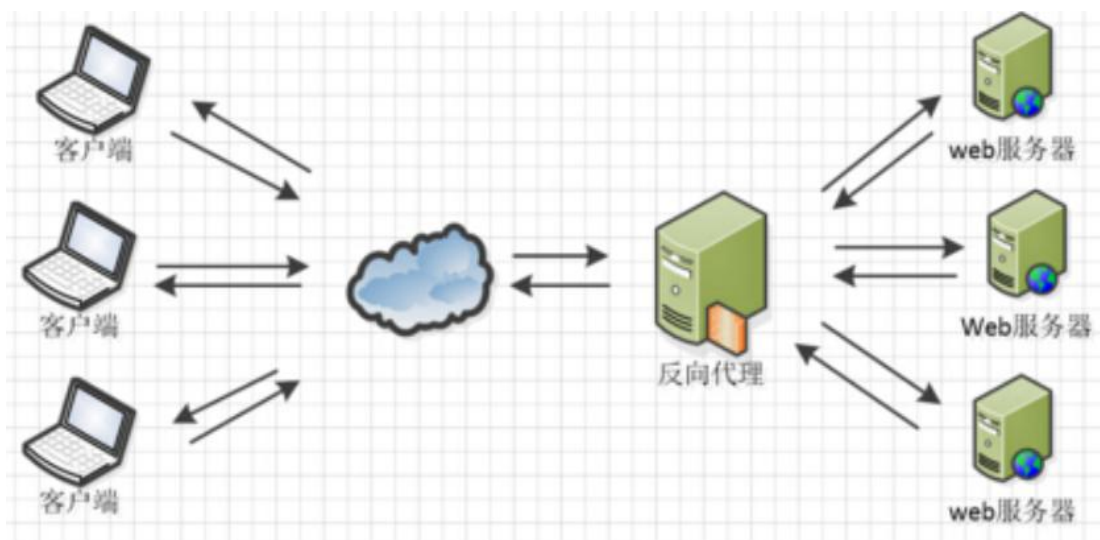
📚 Nginx 学习资料

重要程度：★★★★★

简介：

Nginx (engine x) 是一个采用 C 进行编写的高性能的 HTTP 和反向代理 web 服务器
其特点是占有内存少，并发能力强





参考学习资料：

Nginx 配置详解：

<https://www.runoob.com/w3cnote/nginx-setup-intro.html>

Nginx 服务器安装及配置文件详解：

<https://www.runoob.com/w3cnote/nginx-install-and-config.html>

Linux 下 Nginx+Tomcat 负载均衡和动静分离配置要点

<https://www.runoob.com/w3cnote/linux-nginx-tomcat.html>

负载均衡

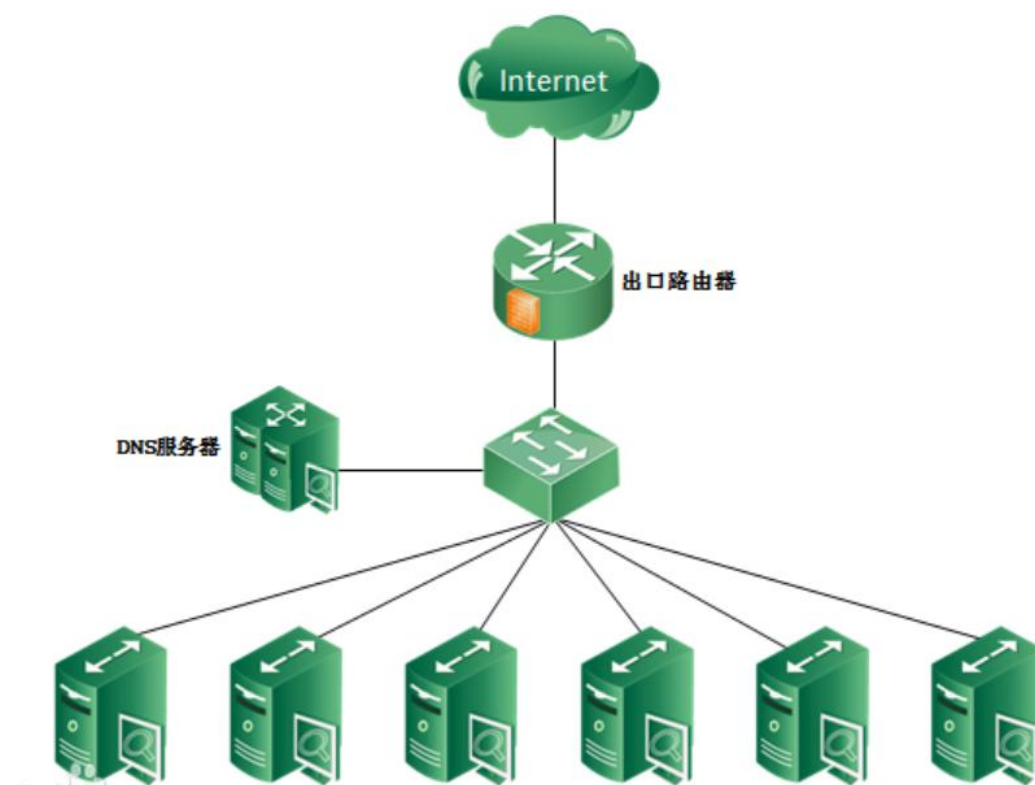
重要程度：★★★★★

简介：

负载均衡（Load balancing）是一种计算机技术，用来在多个计算机（计算机集群）、网络连接、CPU、磁盘驱动器或其他资源中分配负载，以达到最优化资源使用、最大化吞吐率、最小化响应时间、同时避免过载的目的。使用带有负载均衡的多个服务器组件，取代单一的组件，可以通过冗余提高可靠性。负载均衡服务通常是由专用软件和硬件来完成。主要作用是将大量作业合理地分摊到多个操作单元上进行执行，用于解决互联网架构中的高并发和高可用的问题。

负载均衡有软件和硬件负载均衡，硬件负载均衡 F5、软件负载均衡 nginx、阿里的 SLB。

负载均衡算法：主要分为静态和动态两类。静态负载均衡算法以固定的概率分配任务，不考虑服务器的状态信息，如轮转算法、加权轮转算法等；动态负载均衡算法以服务器的实时负载状态信息来决定任务的分配，如最小连接法、加权最小连接法等。



参考学习材料：https://www.sohu.com/a/277818965_411876
https://blog.csdn.net/bpb_cx/article/details/82771168

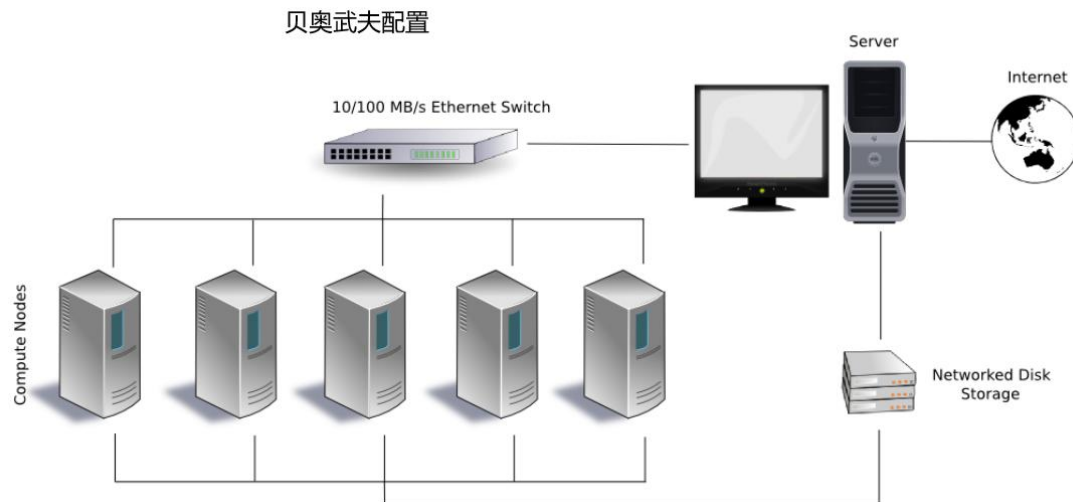
集群知识

重要程度：★★★★★

简介：

计算机集群（computer cluster）简称集群，是一种计算机系统，它通过一组松散集成的计算机软件或硬件连接起来高度紧密地协作完成计算工作。在某种意义上，他们可以被看作是一台计算机。

负载均衡集群运行时一般通过一个或者多个前端负载均衡器将工作负载分发到后端的一组服务器上，从而达到整个系统的高性能和高可用性。



参考学习资料：

https://blog.csdn.net/qq_16116549/article/details/88551090

关系型数据库

重要程度：★★★★★

简介：

关系数据库，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示。标准数据查询语言 SQL 就是一种基于关系数据库，这种语言执行对关系数据库中数据的检索和操作。

关系模型由关系数据结构、关系操作集合、关系完整性约束三部分组成。

关系型数据库是由多张能互相联接的二维行列表格组成的数据库

主流的关系型数据库有 Oracle、DB2、Microsoft SQL Server、Microsoft Access、MySQL、postgresql、 mariaDB、sqlite 等

参考学习资料： <https://www.php.cn/sql/421899.html>

Mysql 数据库基础知识

重要程度：★★★★★

简介：

MySQL 是一种开源的关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL 支持大型的数据库。可以处理拥有上千万条记录的大型数据库。

MySQL 使用标准的 SQL 数据语言形式。

MySQL 可以运行于多个系统上，并且支持多种语言。

推荐学习地址：

<https://www.showapi.com/book/view/2147/0>

<https://www.runoob.com/mysql/mysql-install.html>

Postgresql 数据库知识

重要程度：★★★★★

简介：

PostgreSQL 是一个功能非常强大的、源代码开放的客户/服务器关系型数据库管理系统。

参考学习资料：<https://www.runoob.com/postgresql/postgresql-tutorial.html>

<http://www.postgres.cn/v2/document>

非关系型数据库

重要程度：★★★★★

简介：

NoSQL(Not Only SQL)，泛指非关系型的数据库。NoSQL 数据库的产生就是为了解决大规模数据集合多重数据种类带来的挑战，尤其是大数据应用难题。

NoSQL 有如下优点：易扩展，NoSQL 数据库种类繁多，但是一个共同的特点都是去掉关系数据库的关系型特性。数据之间无关系，这样就非常容易扩展。无形之间也在架构的层面上带来了可扩展的能力。大数据量，高性能，NoSQL 数据库都具有非常高的读写性能，尤其在大数据量下，同样表现优秀。这得益于它的无关系性，数据库的结构简单。

分类：**键值(Key-Value)存储数据库**，这一类数据库主要会使用到一个哈希表，这个表中有一个特定的键和一个指针指向特定的数据。典型代表：**Redis**

列存储数据库：这部分数据库通常是用来应对分布式存储的海量数据，键仍然存在，但是它们的特点是指向了多个列。这些列是由列家族来安排的。典型代表：**HBase**

文档型数据库：该类型的数据模型是版本化的文档，半结构化的文档以特定的格式存储。典型代表：**MongoDB、SequoiaDB**

图形(Graph)数据库：它是使用灵活的图形模型，并且能够扩展到多个服务器上

Redis 基础知识

重要程度：★★★★★

简介：

Redis 是一个高性能的 key-value 数据库，redis 支持各种不同方式的排序。与 memcached 一样，为了保证效率，数据都是缓存在内存中，使用内存存储。区别的是 redis 会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了 master-slave(主从)同步。

Redis 支持主从同步。数据可以从主服务器向任意数量的从服务器上同步，从服务器可以是关联其他从服务器的主服务器。这使得 Redis 可执行单层树复制

推荐学习地址：

<https://www.runoob.com/redis/redis-tutorial.html>

Redis 设计与实现：<https://www.showapi.com/book/view/2054/0>

Memcache 学习资料

重要程度：★★★★★

简介：

memcache 是一套分布式的高速缓存系统，将数据调用到内存中，然后从内存中读取，从而大大提高读取速度。

最大 30 天的数据过期时间，设置为永久的也会在这个时间过期，常量 `REALTIME_MAXDELTA`

最大键长为 250 字节，大于该长度无法存储，常量 `KEY_MAX_LENGTH`

单个 item 最大数据是 1MB，超过 1MB 数据不予存储，常量 `POWER_BLOCK`

最大同时连接数是 200，通过 `conn_init()` 中的 `freetotal` 进行控制，最大软连接数是 1024，通过 `settings.maxconns=1024` 进行控制

参考学习资料：

<https://www.runoob.com/memcached/memcached-tutorial.html>

MongoDB 基础知识

重要程度：★★★★★

简介：

MongoDB 是一个基于分布式文件存储的数据库，介于关系数据库和非关系数据库之间的产品。是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似 json 的 bson 格式，因此可以存储比较复杂的数据类型。Mongo 最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

参考学习材料：<https://www.runoob.com/mongodb/mongodb-tutorial.html>

时序型数据库

重要程度：★★★

简介：

时序数据库全称为时间序列数据库。时间序列数据库主要用于指处理带时间标签（按照时间的顺序变化，即时间序列化）的数据，带时间标签的数据也称为时间序列数据。时序大数据解决方案通过使用特殊的存储方式，使得时序大数据可以高效存储和快速处理海量时序大数据，是解决海量数据处理的一项重要技术。该技术采用特殊数据存储方式，极大提高了时间相关数据的处理能力，相对于关系型数据库它的存储空间减半，查询速度极大的提高。

典型代表 “influxdb、prometheus”

Influxdb 基础知识

重要程度：★★★

简介：

InfluxDB 是一个由 InfluxData 开发的开源时序型数据库。它由 Go 写成，着力于高性能地查询与存储时序型数据。InfluxDB 被广泛应用于存储系统的监控数据。

InfluxDB 存储的数据从逻辑上由 Measurement, tag 组以及 field 组以及一个时间戳组成

的

Measurement: 由一个字符串表示该条记录对应的含义

tag 组: 由一组键值对组成，表示的是该条记录的一系列属性信息

field 组: 也是由一组键值对组成，表示的是该条记录具体的 **value** 信息(有名称)

时间戳: 就是该条记录的时间属性

参考学习资料: <https://www.cntofu.com/book/118/readme.html>

Prometheus 基础知识

重要程度: ★★★

简介:

Prometheus 它是由谷歌研发的一款开源的监控软件，目前已经被云计算本地基金会托管。

一个多维数据模型，其中包含通过度量标准名称和键/值对标识的时间序列数据

PromQL，一种灵活的查询语言，可利用此维度

不依赖分布式存储；单服务器节点是自治的

时间序列收集通过 **HTTP** 上的拉模型进行

通过中间网关支持推送时间序列

通过服务发现或静态配置发现目标

多种图形和仪表板支持模式

易于管理、轻易获取服务内部状态、高效灵活的查询语句、支持本地和远程存储、采用 **http** 协议，默认 **pull** 模式拉取数据，也可以通过中间网关 **push** 数据、支持自动发现、可扩展、易集成

参考学习资料:

<https://prometheus.io/docs/introduction/overview/>

<https://www.cnblogs.com/fatyao/p/11007357.html>

<https://www.cnblogs.com/zqj-blog/p/10871033.html>

程序员必须掌握的英语单词

重要程度: ★★★★★

参考学习地址: <https://www.showapi.com/book/view/2093/0>