J2EE与中间件技术

李 会 格

E-mail: 1034434100@qq.com

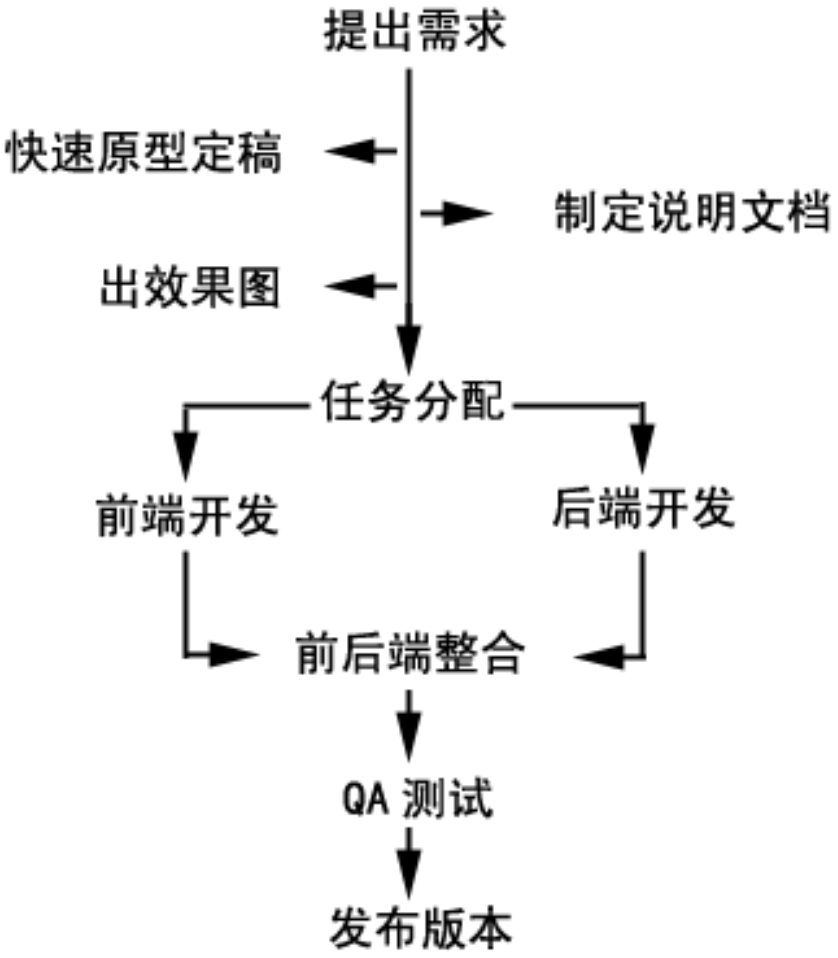
- 班级群号：
- 资源共享
- 签到打卡
- 提交作业

- **成绩构成：　　平时40%　+　期末60%**

推荐学习网站

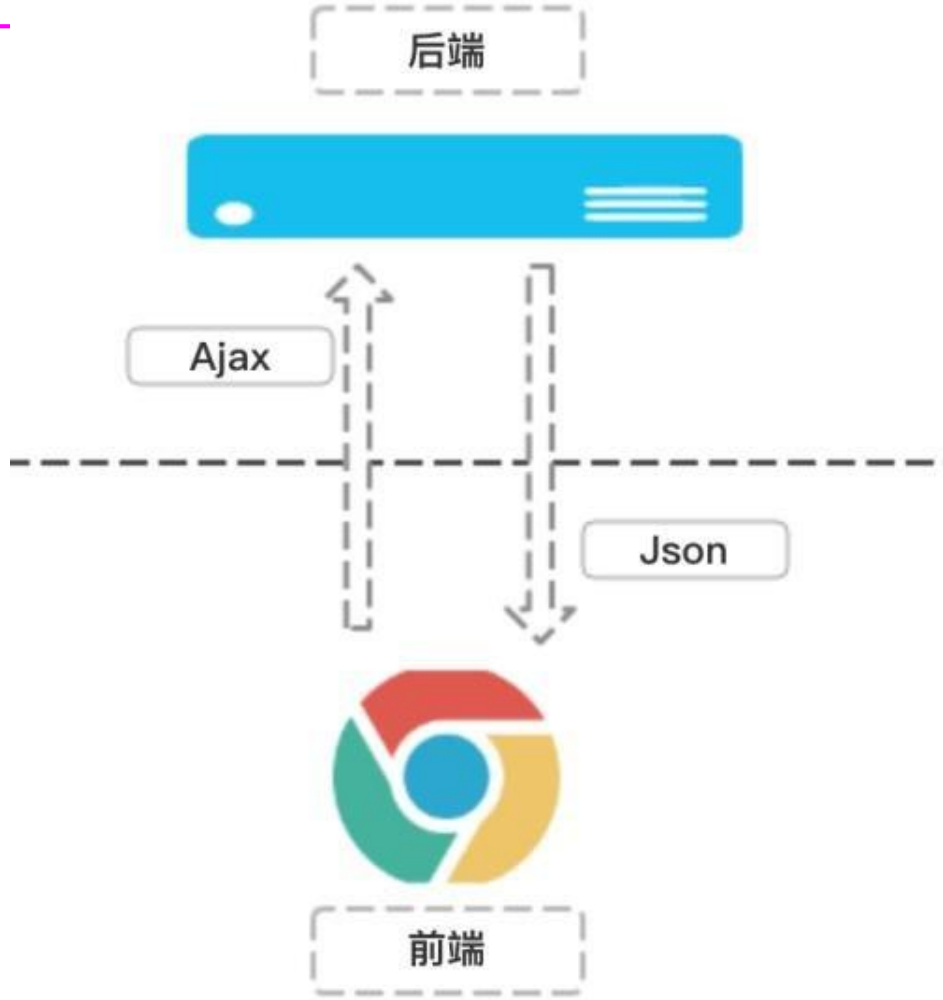
- 博客园https://www.cnblogs.com

软件的开发流程



团队组织结构及职责

组织结构及职责	
需求分析小组	与需求提出方进行需求确认，可采用快速原型或画线框图等方式进行，并给出最终说明文档。
设计小组	根据需求分析时的辅助手段（快速原型或线框图）以及用户体验设计，制作出项目各页面最终效果图。
前端小组	将设计小组给出的最终效果图转换为网页格式，实现说明文档中指定的交互功能。
后端小组	根据说明文档进行数据库设计开发、数据API开发以及对前端交付的网页进行套页（将假数据替换为真实数据）。
测试小组	为项目提供测试。


后端
Ajax
Json
前端

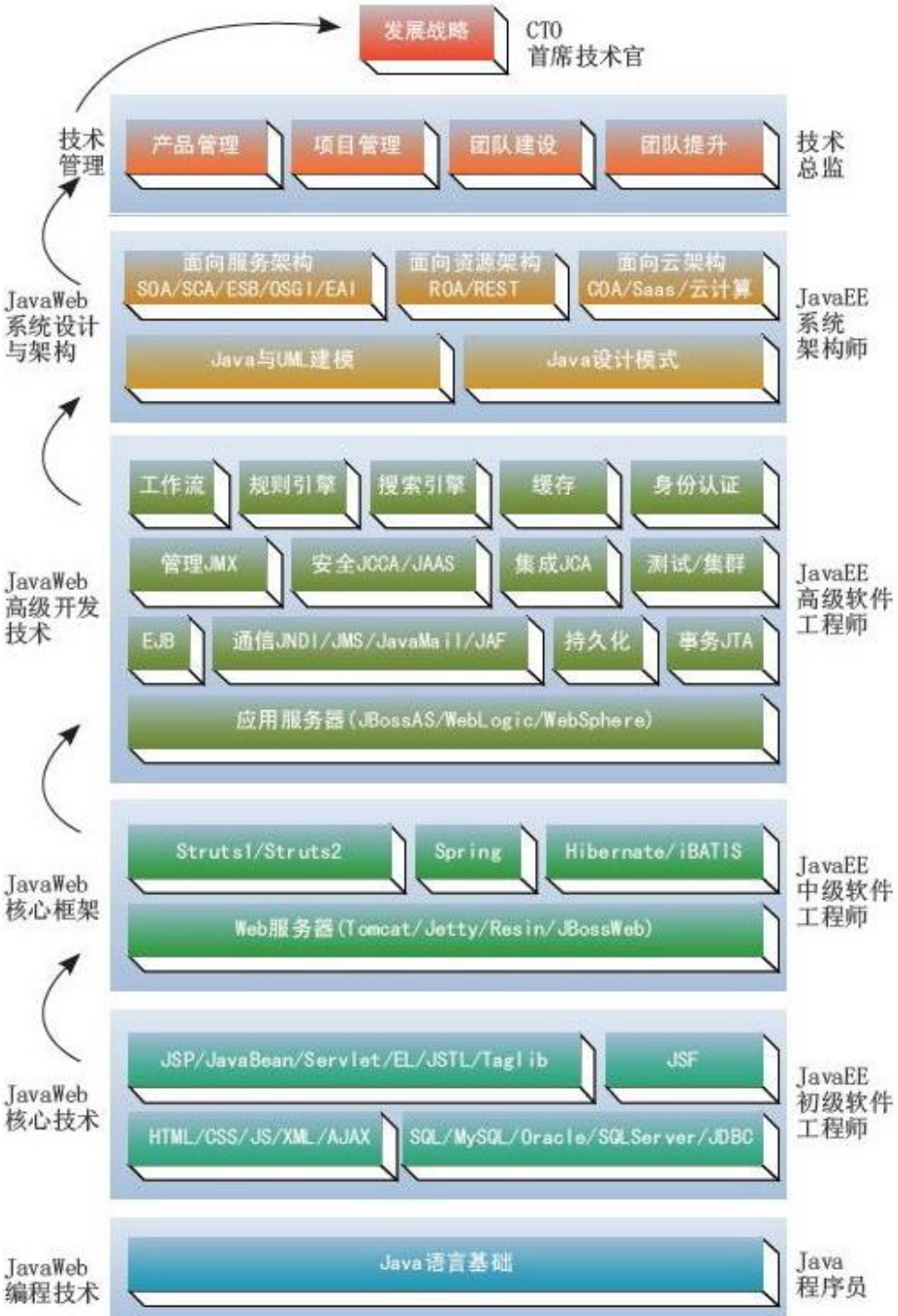
前端

- 负责显示给用户。
- 技术：html , css , js、JSP等。
- 框架：Bootstrap、Node.js、VUE等

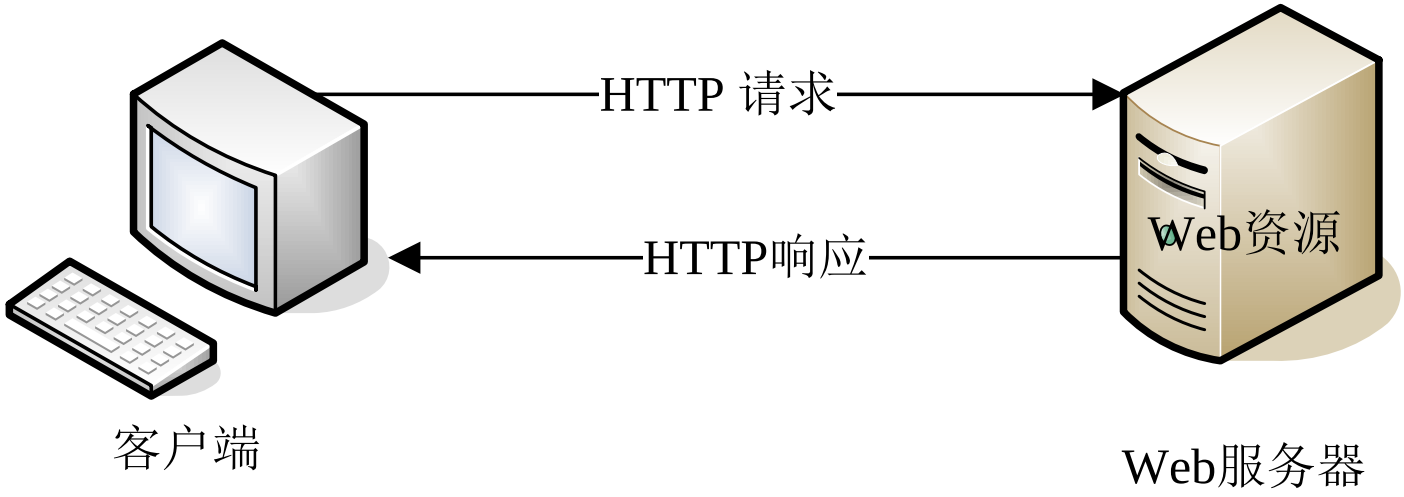
后台

- 负责对业务的控制和处理
- 负责访问和更新数据库。
- 技术：servlet+javabean+JDBC
- 框架：SSH2，SSM等

Java工程师成长路径



- Web工作原理



HTTP协议特点

- 1、简单快速
- 2、无连接
- 3、无状态

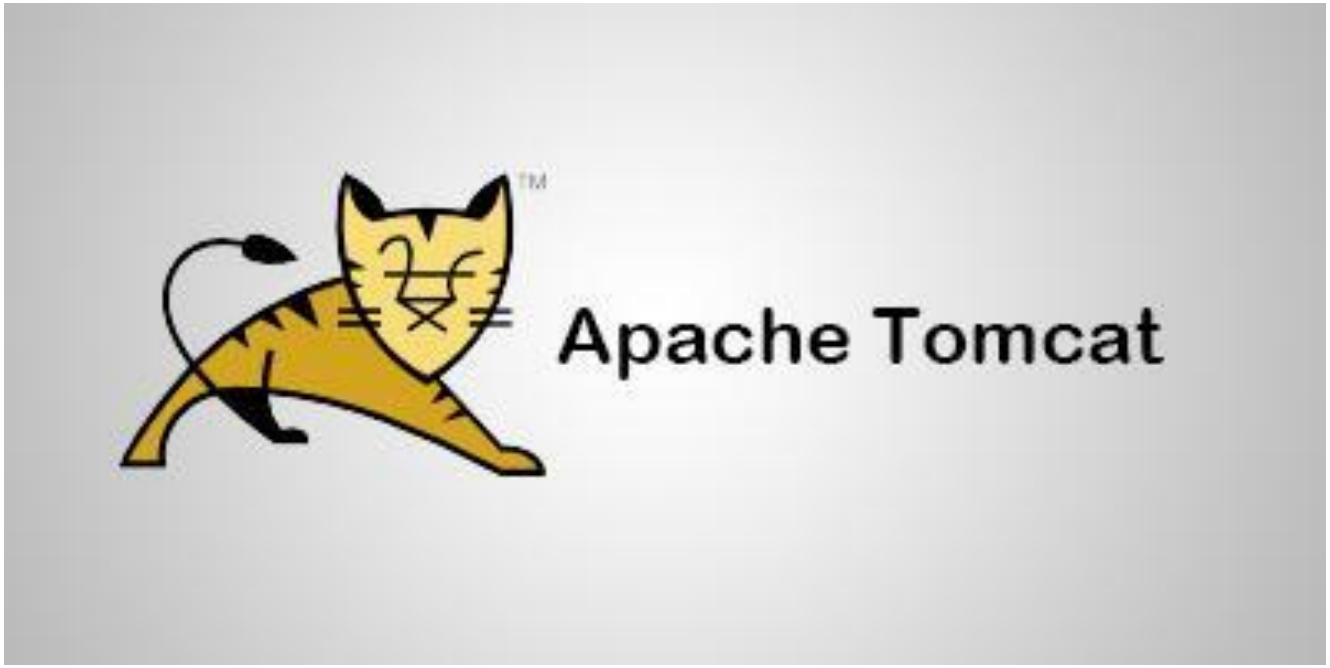
HTTPS与HTTP的区别

- 1、https协议需要到ca申请证书，一般免费证书较少，因而需要一定费用。
- 2、http是超文本传输协议，信息是明文传输，HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，比http协议安全。
- 3、http和https使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。

- 几个重要概念：
  - ✓超文本（Hyper Text）
  - ✓超媒体（Hyper Media）
  - ✓万维网（Wide World Web，WWW）
  - ✓超文本传输协议（ Hyper Text Transfer Protocol，HTTP ）
  - ✓超文本标记语言（Hyper Text Markup Language，HTML）

常用软件

Tomcat



Eclipse

eclipse

eSoftner

其他IDE

IntelliJ IDEA

MySQL

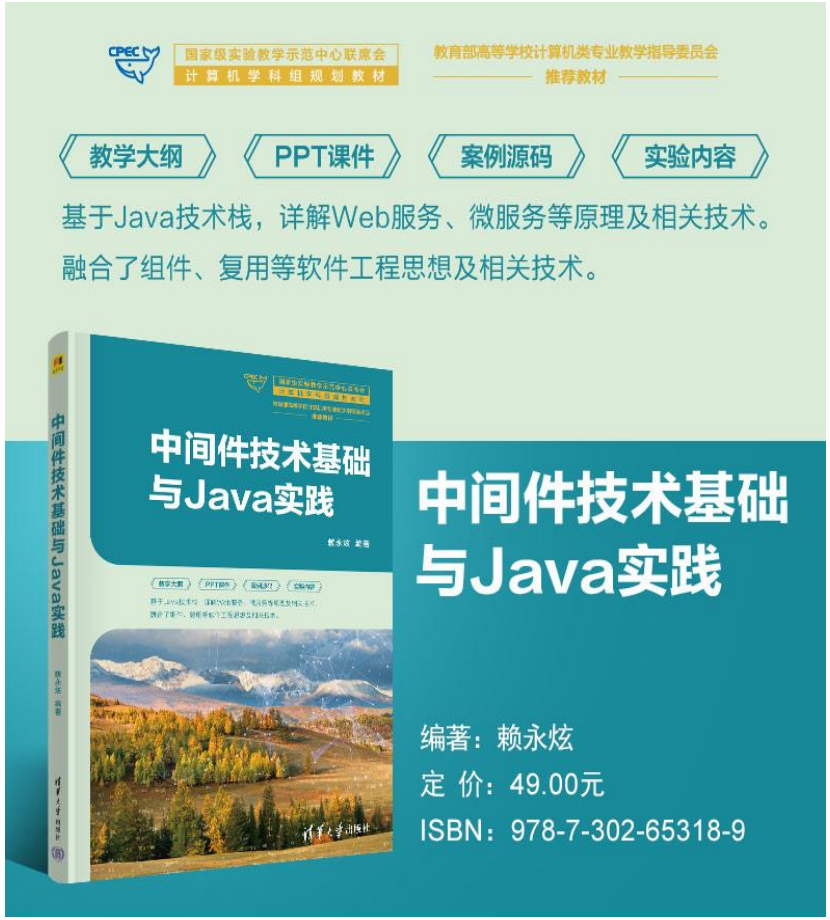
Navicat For Mysql

《中间件技术基础与Java实践》

ISBN:9787302653189

定价：49元

2024年3月出版



其它参考书目

- java EE(SSM)企业应用实战
- 分布式中间件技术实战
- Java分布式中间件开发实战

第一章　分布式系统概述

（PPT版本号：2023年12月版本）

李 会 格

E-mail：1034434100@qq.com

大纲

- 计算机系统的演化
  - ✓ 单机系统
  - ✓ 单机分布式系统
  - ✓ 中心集群系统
  - ✓ 分布式集群系统
- 分布式系统的概念
- 分布式系统的应用和意义
- 分布式系统的难点和框架
- 分布式计算和大数据技术

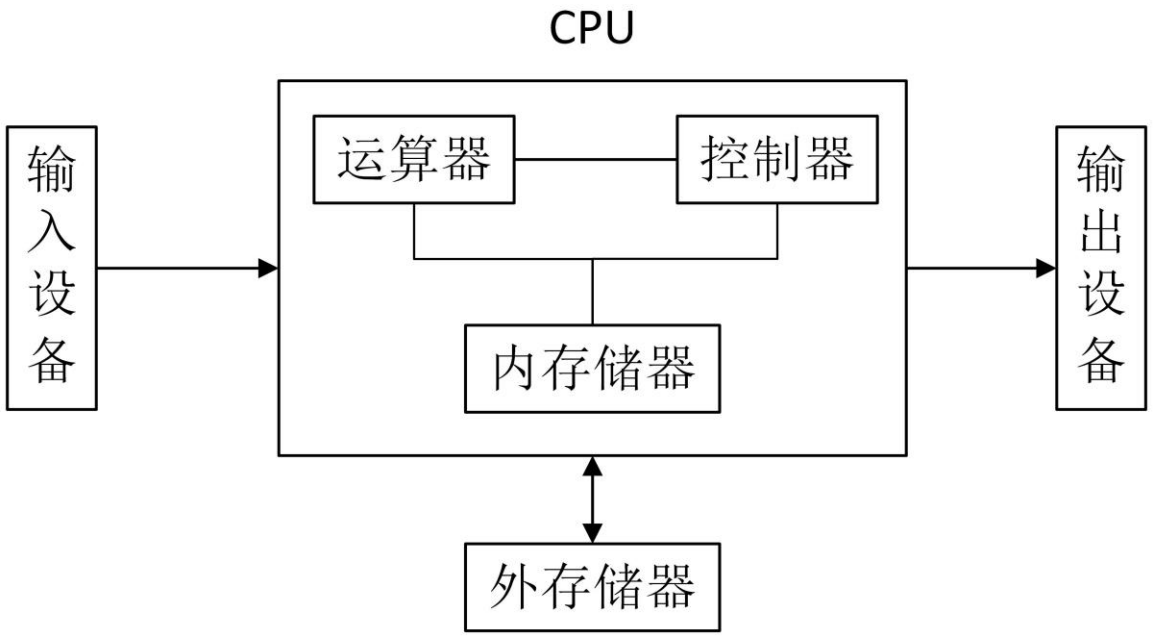
计算机系统的演化

- 从处理能力在不同计算机的分配上来划分，可将计算机系统可分为：

计算机系统
- 单机系统
- 单机分布式系统
- 中心集群系统
- 分布式集群系统

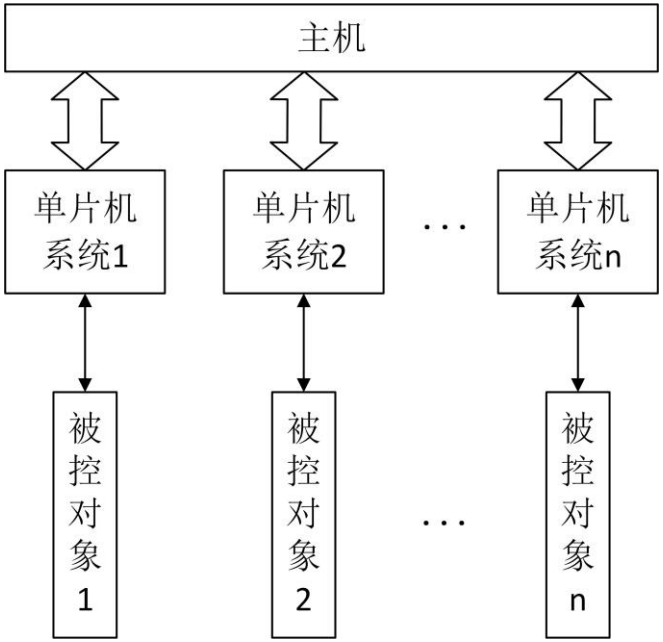
计算机系统的演化

- 单机系统：在单片机应用系统中只有一个单片机，适合于小规模的单片机应用系统



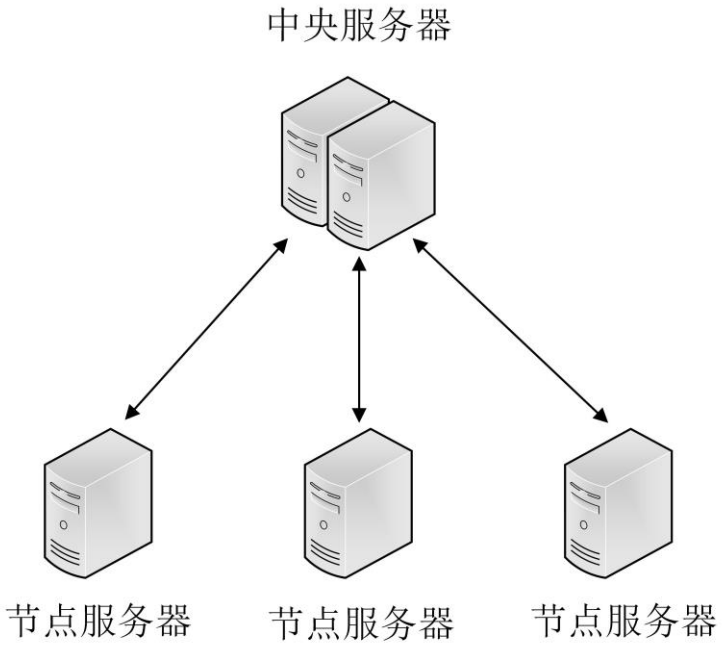
计算机系统的演化

- 单机分布式系统：由多个单机系统组成，应用系统的任务被分散到各微机和单片机上。多机结构又分为多级多机分散控制结构与局部网络结构，多级多机分散控制结构的典型代表是两级分散控制系统



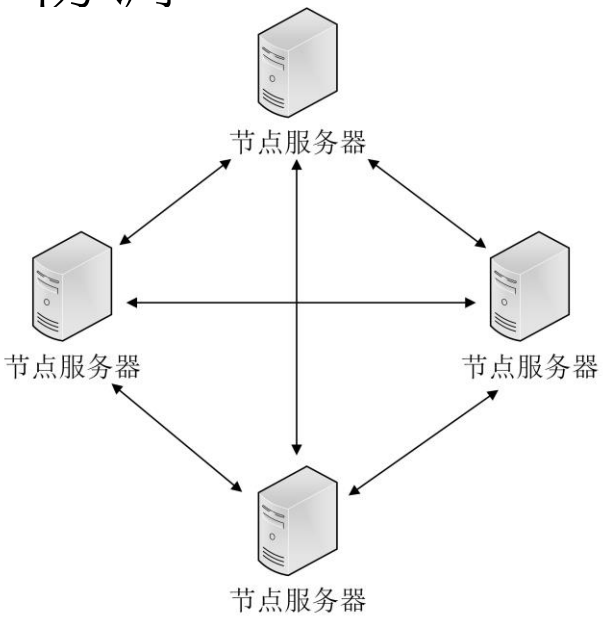
计算机系统的演化

- 中心集群系统：由一台中央服务器和多台节点服务器组成。中央服务器统一进行资源和任务调度，将任务下达给节点服务器；节点服务器执行任务并反馈结果。



计算机系统的演化

- 分布式集群系统：没有中央服务器和节点服务器之分，所有服务器平等。服务的执行和数据的存储被分散到不同的服务器集群，服务器集群间通过消息传递进行通信和协调。



大纲

- 计算机系统的演化
  - ✓ 单机系统
  - ✓ 单机分布式系统
  - ✓ 中心集群系统
  - ✓ 分布式集群系统
- 分布式系统的概念
- 分布式系统的应用和意义
- 分布式系统的难点和框架
- 分布式计算和大数据技术

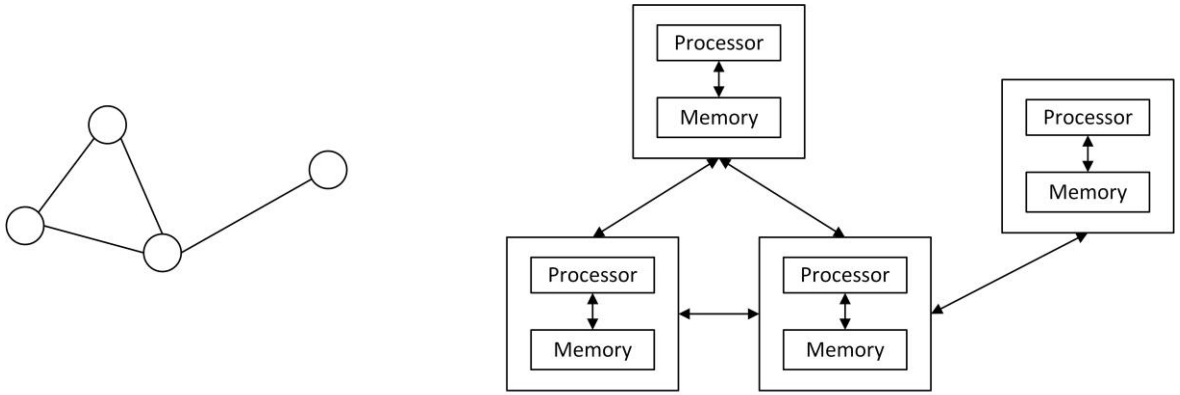
分布式系统的概念

- 分布式系统（Distributed System）是独立的计算机的集合，但是对用户来说，系统就像一台计算机一样。
  - ✓ 从硬件角度来讲，各个计算机都是自治的；
  - ✓ 从软件角度来讲，用户将整个系统看作是一整台计算机。

- 一个分布式系统中的所有的计算实体都有一个共同目标，例如解决一个需要大量计算的问题。分布式系统的目的就是合理调度分享的资源或给用户提供交流服务。

分布式系统的概念

- 分布式系统有以下典型特性：
  - ✓ 系统能够容纳个体计算机返回的错误。
  - ✓ 系统的结构（网络拓扑结构、网络延迟、计算机编号等）不能预测，会在运行过程中出现变化。
  - ✓ 每台计算机都只能受限制地观察到不完整的系统，有可能每台计算机都只知道受信息的一部分。



（a）　　（b）

分布式系统的概念

- 从软件角度来讲，分布式系统是建立在网络之上的，具有高度的内聚性和透明性。

- 内聚性体现为每一个数据库分布节点高度自治，有本地的数据库管理系统。

- 透明性是指每一个数据库分布节点对用户的应用来说都是透明的，用户感觉不到是本地还是远程，也感觉不到数据是分布存储的。

大纲

- 计算机系统的演化
  - ✓ 单机系统
  - ✓ 单机分布式系统
  - ✓ 中心集群系统
  - ✓ 分布式集群系统
- 分布式系统的概念
- 分布式系统的应用和意义
- 分布式系统的难点和框架
- 分布式计算和大数据技术

分布式系统的主要应用

- 并行和高性能应用

  原则上，并行应用也可以在共享存储器多处理机上运行，但共享存储器系统不能很好地扩大规模以包括大量的处理机。高性能计算和通信（HPCC）应用一般需要一个可伸缩的设计，这种设计取决于分布式处理。

- 容错应用

  因为每个计算机节点是自治的，所以分布式系统更加可靠。一个节点中的软件或硬件故障不影响其他节点的正常功能。

分布式系统的主要应用

- 固有的分布式应用

  许多应用是固有分布式的。这些应用是突发模式（Burst Mode）而非批量模式（Bulk Mode）。这方面的实例有事务处理和Internet Java程序。这些应用的性能取决于吞吐量（事务响应时间或每秒完成的事务数），而不是一般多处理机所用的执行时间。

分布式系统的意义

分布式系统的意义

- 解决单机性能瓶颈的问题
- 降低系统整体的成本
- 提供稳定性和可用性解决方案
- 加速网络时代的应用开发

分布式系统的意义

- 解决单机性能瓶颈的问题

  由于技术工艺的原因，计算机性能在某一时间段内必然存在一个极限和瓶颈。因此，要获得更高的系统性能，分布式系统是一个有效且唯一的解决方案。分布式的机器集群系统是目前大规模系统和软件平台的根本解决方案。

- 降低系统整体的成本

  实践证明，通过在一个系统中集中使用大量的廉价CPU和小型机，可以得到比单个的大型集中式系统高得多的性价比。此外，渐增式的增长方式也能减低系统的整体成本。

分布式系统的意义

- 提供稳定性和可用性解决方案
  - ✓ 对于集中式系统，如果主机出现故障，整个系统将无法使用。如果提供的服务质量随时间呈现较大的波动性，那么在进行上层应用的设计时会遇上系统不协调，结果不一致等多种问题。
  - ✓ 分布式系统把工作负载分散到众多的机器上，单个故障最多只会使一台机器停机，而其它机器不会受任何影响。对于关键性的应用，如核反应堆或飞机控制系统，采用分布式系统来实现主要是考虑到它可以获得高可靠性。

分布式系统的意义

- 加速网络时代的应用开发
  - ✓ 在网络时代，分布式应用的需求客观存在，且层出不穷。比如连锁超市需要建立商业分布式系统，大型的网络游戏需要处理成千上万的请求。可利用“分而治之”原则将大量服务请求进行适当的划分。
  - ✓ 分布式系统的分布性特征对用户来说是看不见的，开发者可以聚焦于系统的应用逻辑和核心流程，降低了网络大型系统的开发难度，加速网络时代的应用开发。

分布式系统的意义

- 综合起来，同集中式系统相比较，分布式系统具有以下的优点：

项目	描述
经济	微处理机提供了比大型主机更好的性能价格比
速度	分布式系统总的计算能力比单个大型主机更强
固有的分布性	一些应用涉及到空间上分散的机器
可靠性	如果一个机器崩溃，整个系统还可以运转
渐增	计算能力可以逐渐有所增加

大纲

- 计算机系统的演化
  - ✓ 单机系统
  - ✓ 单机分布式系统
  - ✓ 中心集群系统
  - ✓ 分布式集群系统
- 分布式系统的概念
- 分布式系统的应用和意义
- 分布式系统的难点
- 经典的分布式系统框架
- 分布式计算和大数据技术

分布式系统技术难点

分布式系统技术难点

- CAP不可能三角原则
- 网络延迟
- 异构系统的“不标准”问题
- 服务依赖性问题

分布式系统技术难点

- 分布式系统的CAP不可能三角原则
- CAP原则是由Eric Brewer提出的分布式系统中最为重要的理论之一，指在一个分布式系统中，Consistency（一致性）、 Availability（可用性）、Partition tolerance（分区容错性），三者不可兼得。包括：
  - 一致性（C）：在分布式系统中的所有数据备份，在同一时刻是否有相同的值。
  - 可用性（A）：在集群中一部分节点故障后，集群整体是否还能响应客户端的读写请求。
  - 分区容错性（P）：以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在C和A之间做出选择。

分布式系统技术难点

- 网络延迟
- 由于服务和数据分布在不同的机器上，每次交互都需要跨机器运行，这带来网络的延迟问题，使得系统整体性能的降低。过度的延迟就会带来系统RPC调用超时。系统超时几乎是所有分布式系统复杂性的根源。
- 主要的解决方案包括异步化，失败重试等。

分布式系统技术难点

- 异构系统的“不标准”问题
- 分布式系统涉及多台计算机，有时难免会遇到计算机之间的异构系统的不标准问题。
- 主要体现在：软件、应用不标准，通讯协议、数据格式不标准，开发、运维的过程和方法不标准。

分布式系统技术难点

- 服务依赖性问题
- 分布式架构下，服务是有依赖的。如果服务依赖链上的某个非关键业务崩溃，那么会导致整个系统不可用。
- 分布式系统通常会把系统分为四层：基础层，硬件、网络和存储设备等；平台层，也就是中间件层；应用层，也就是业务软件；接入层，接入用户请求网关、负载均衡、CDN、DNS等等。任何一层的问题都会导致整体问题。因为熟悉整个架构的人占少数，几乎是每层各自管理，所以运维被割裂开来，运维难度更高。

大纲

- 计算机系统的演化
  - ✓ 单机系统
  - ✓ 单机分布式系统
  - ✓ 中心集群系统
  - ✓ 分布式集群系统
- 分布式系统的概念
- 分布式系统的应用和意义
- 分布式系统的难点
- 经典的分布式系统框架
- 分布式计算和大数据技术

经典的分布式系统框架

- 远程调用RPC
- 在过去开发人员可以通过加载类库，内存共享和其他机制来调用本地的过程或函数。但这在分布式计算的环境中，无法调用网络环境中的另一台机器上的程序。
- 通常这个问题可通过传统的套接字编程来解决。但缺点是必须首先了解传输协议，如TCP或UDP，然后基于此方法开发。

- 因此，还有另一种解决方案：提供透明的调用机制，使得开发人员不必显式区分本地调用和远程调用，也不需要关注和理解基础网络技术协议。这就是远程过程调用协议（RPC, Remote Procedure Call）。

经典的分布式系统框架

- 分布式计算环境DCE
- 在网络计算中，分布式计算环境（DCE, Distributed Computing Environment）是由开放软件基金会（OSF）提出的分布式计算技术的工业标准集，用于提供保护和控制对数据访问的安全服务、寻找分布式资源的名字服务以及高度可伸缩的模型。
- 分布式计算环境通常用于较大的计算系统网络中，其中包括在地理位置上分散的不同大小的服务器。分布式计算环境使用客户端/服务器模型。另外，分布式计算环境也包括安全支持。

经典的分布式系统框架

- 群件和分布式开发模型
- 分布式系统有一个特别的应用称为计算机支持的协同工作（CSCW）或群件（Groupware）， 支持用户协同工作。另一个应用是分布式会议，即通过物理的分布式网络进行电子会议。
- 还有一些基于其它标准开发的模型，比如CORBA（公共对象请求代理体系结构）。CORBA使用面向对象模型实现分布式系统中的透明服务请求。除此之外还有比如微软的分布式构件对象模型（DCOM）和Sun Microsystem公司的Java Beans等。

大纲

- 计算机系统的演化
  - ✓ 单机系统
  - ✓ 单机分布式系统
  - ✓ 中心集群系统
  - ✓ 分布式集群系统
- 分布式系统的概念
- 分布式系统的应用和意义
- 分布式系统的难点
- 经典的分布式系统框架
- 分布式计算和大数据技术

大数据及其处理技术

- 大数据是指那些数据量特别大、数据类别特别复杂的数据集，这种数据集无法用传统的数据库进行存储、管理和处理。
- 大数据虽然量大且复杂，但是有价值的信息往往深藏其中，这就决定了大数据处理的效率要高。

大数据的主要特点 “4V”

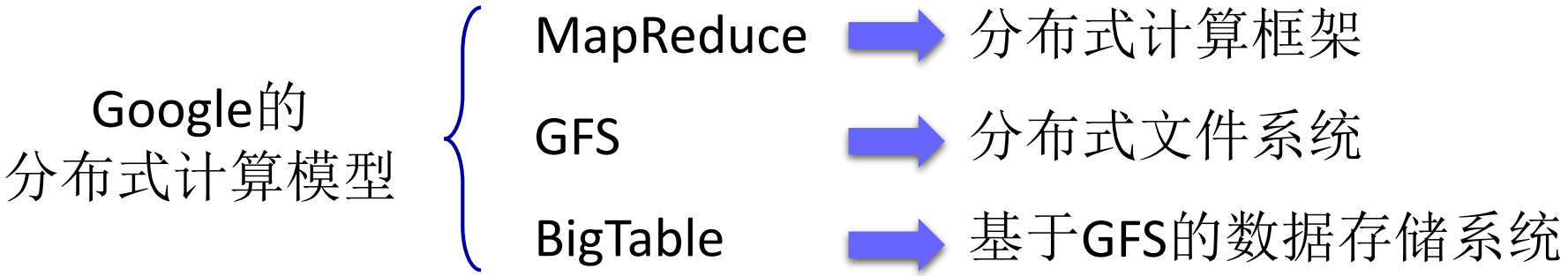
- 数据量大（Volume）
- 数据类别复杂（Variety）
- 数据处理速度快（Velocity）
- 数据真实性高（Veracity）

大数据及其处理技术

- 对于如何处理大数据，计算机科学界有两大方向：
  - 第一个方向是集中式计算，就是通过不断增加处理器的数量来增强单个计算机的计算能力，从而提高处理数据的速度。
  - 第二个方向是分布式计算，就是把一组计算机通过网络相互连接组成分散系统，然后将需要处理的大量数据分散成多个部分，交由分散系统内的计算机组同时计算。
- 由于对于互联网公司而言大型机的价格过于昂贵，如今分布式的计算和存储平台几乎成为互联网公司大规模业务的唯一解决方案。

大数据及其处理技术

- 2003年到2004年间，Google发表了MapReduce、GFS（Google File System）和BigTable三篇技术论文，提出了一套全新的分布式计算理论。



分布式的大数据处理平台

- MapReduce
- MapReduce是Google开发的Java、Python、C++编程工具，用于大规模数据集（大于1TB）的并行运算，也是云计算的核心技术，一种分布式运算技术，也是简化的分布式编程模式，适合用来处理大量数据的分布式运算，用于解决问题的程序开发模型，也是开发人员拆解问题的方法。

- MapReduce模式的思想是将要执行的问题拆解成Map（映射）和Reduce（化简）的方式，先通过Map程序将数据切割成不相关的区块，分配给大量计算机处理达到分布运算的效果，再通过Reduce程序将结果汇整，输出开发者需要的结果。

分布式的大数据处理平台

- MapReduce
- MapReduce的软件实现是指定一个Map（映射）函数，把原本的键值对（key/value）重新映射，形成一系列中间键值对，然后把它们传给Reduce函数，把具有相同中间形式key的value合并在一起。

- map和reduce函数具有一定的关联性。Map：(k1,v1)->list(k2,v2)，Reduce：(k2,list(v2))->list(v2)。其中v1、v2可以是简单数据，也可以是一组数据，对应不同的映射函数规则。

- 在Map过程中将数据并行，即把数据用映射函数规则分开，而Reduce则把分开的数据用化简函数规则合在一起，也就是说Map是一个分的过程，Reduce则对应着合。

分布式的大数据处理平台

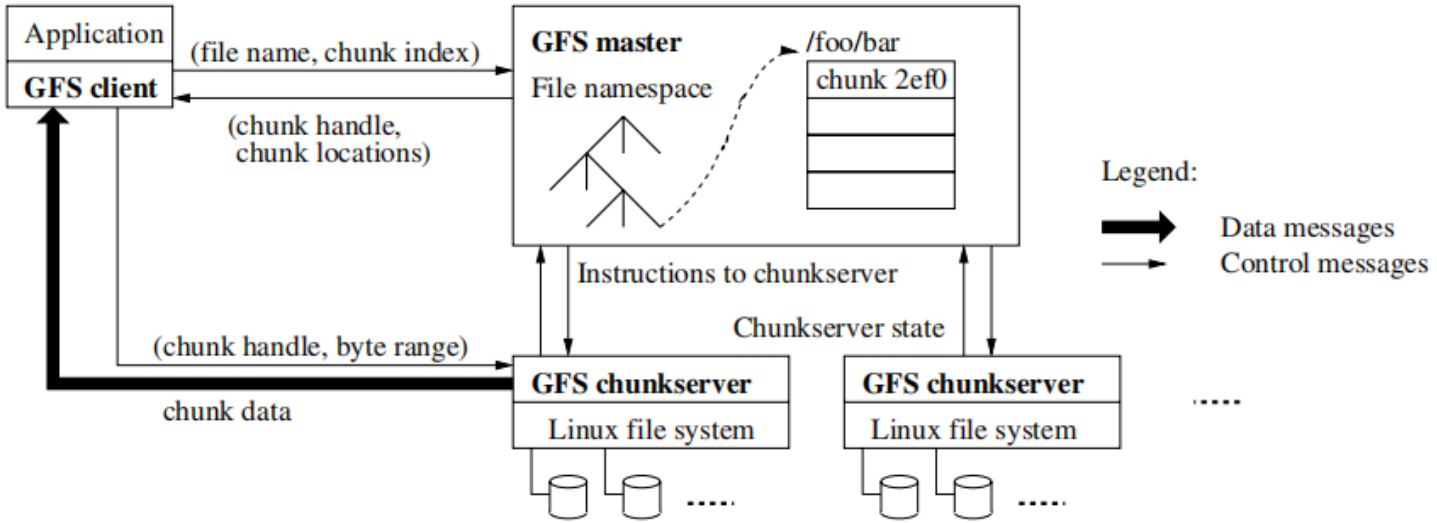
- MapReduce
- MapReduce的应用广泛，包括简单计算任务、海量输入数据、集群计算环境等，如分布grep、分布排序、单词计数、Web连接图反转、每台机器的词矢量、Web访问日志分析、反向索引构建、文档聚类、机器学习、基于统计的机器翻译等。

分布式的大数据处理平台

- GFS
- GFS是一个可扩展的分布式文件系统，用于大型的、分布式的、对大量数据进行访问的应用。它运行于廉价的普通硬件上，并提供容错功能。它可以给大量的用户提供总体性能较高的服务。

- GFS期望的应用场景是大文件，连续读，不修改，高并发。

分布式的大数据处理平台



- 一个GFS包括一个主服务器（Master）和多个块服务器（Chunk Server），这样一个GFS能够同时为多个客户端应用程序（Application）提供文件服务。文件被划分为固定的块，由主服务器安排存放到块服务器的本地硬盘上。主服务器会记录存放位置等数据，并负责维护和管理文件系统，包括块的租用、垃圾块的回收以及块在不同块服务器之间的迁移。此外，主服务器还周期性地与每个块服务器通过消息交互。

分布式的大数据处理平台

- BigTable
- BigTable是Google设计的分布式数据存储系统，用来处理海量的数据的一种非关系型的数据库。其设计目的是快速且可靠地处理PB级别的数据，并且能够部署到上千台机器上。Bigtable为客户提供了简单的数据模型，利用这个模型，客户可以动态控制数据的分布和格式，用户也可以自己推测底层存储数据的位置相关性。

- Bigtable不是关系型数据库，但是却沿用了很多关系型数据库的术语，像table（表）、row（行）、column（列）等。但本质上说，Bigtable是一个键值（key-value）映射。按作者的说法，Bigtable是一个稀疏的，分布式的，持久化的，多维的排序映射。

分布式的大数据处理平台

- BigTable
- BigTable具有如下几个特点:
  - ✓ 适合大规模海量数据，PB级数据；
  - ✓ 分布式、并发数据处理，效率极高；
  - ✓ 易于扩展，支持动态伸缩；
  - ✓ 适用于廉价设备；
  - ✓ 适合于读操作，不适合写操作；
  - ✓ 不适用于传统关系型数据库；

分布式的大数据处理平台

- BigTable
- Bigtable已经实现了以下的几个目标：适用性广泛、可扩展、高性能和高可用性。已经在超过60个Google的产品和项目上得到了应用，包括Google Analytics、GoogleFinance、Orkut、Personalized Search、Writely和GoogleEarth。

- GFS和Bigtable两层的设计是一个几乎完美的组合。GFS本质上是一个弱一致性系统，可能出现重复记录、记录乱序等各种问题。Bigtable是GFS之上的一个索引层，为了服务百PB级别的应用，采用两级的B+树索引结构。GFS保证成功的记录至少写入一次并由Bigtable记录索引，由于Bigtable是一个强一致性系统，整个系统对外表现为强一致性系统。

主流的分布式计算系统

主流的分布式计算系统
- Hadoop
- Spark
- Flink
- Storm
- Spanner

主流的分布式计算系统

- Hadoop
- Yahoo的工程师Doug Cutting和Mike Cafarella在2005年合作开发了分布式计算系统Hadoop。
- Hadoop采用MapReduce分布式计算框架，并根据GFS开发了HDFS分布式文件系统，根据BigTable开发了HBase数据存储系统。
- 尽管和Google内部使用的分布式计算系统原理相同，但是Hadoop在运算速度上依然达不到Google论文中的标准。不过，Hadoop的开源特性使其成为分布式计算系统事实上的国际标准。

主流的分布式计算系统

- Spark
- Spark是Apache基金会的开源项目，它由加州大学伯克利分校的实验室开发，是另外一种重要的分布式计算系统。
- Spark在Hadoop的基础上进行了一些架构上的改良，是专为大规模数据处理而设计的快速通用的计算引擎。
- Spark拥有Hadoop MapReduce所具有的优点；但不同于MapReduce的是其任务的中间输出结果可以保存在内存中，从而不再需要读写HDFS。因此Spark能更好地适用于数据挖掘与机器学习等需要迭代的MapReduce的算法。

主流的分布式计算系统

- Flink
- Flink是由Apache软件基金会开发的开源流数据处理框架，其核心是用Java和Scala编写的分布式流数据流引擎。
- Flink设计为在所有常见的集群环境中运行，以数据并行和流水线方式执行任意流数据程序，以内存速度和任何规模执行计算。此外，Flink的运行时本身也支持迭代算法的执行。

主流的分布式计算系统

- Storm
- Storm是Twitter主推的分布式计算系统，它由BackType团队开发，是Apache基金会的孵化项目。
- 它在Hadoop的基础上提供了实时运算的特性，可以实时地处理大数据流。不同于Hadoop和Spark，Storm不进行数据的收集和存储工作，它直接通过网络实时的接受数据并且实时地处理数据，然后直接通过网络实时的传回结果。

主流的分布式计算系统

- Spanner
- Spanner是由谷歌公司研发的、可扩展的、多版本的、全球分布式的、同步复制式数据库。在最高抽象层面，Spanner就是一个数据库，把数据分片存储在许多Paxos状态机上，位于遍布全球的数据中心内。
- Spanner被设计成可以扩展到几百万个机器节点，跨越成百上千个数据中心，具备几万亿数据库行的规模。应用可以借助于Spanner来实现高可用性，通过在一个洲的内部和跨越不同的洲之间复制数据，保证即使面对大范围的自然灾害时数据依然可用。

本章小结

- 本章首先主要介绍了分布式系统的基础概念，以及计算机从单机系统一路到分布式系统的发展历程；分析了分布式系统在某些应用场景的优势和技术难点。简要介绍了分布式系统的常用框架，介绍了分布式系统和大数据之间的联系，以及分布式的大数据计算平台。