

## 2.

学生：张帅豪 18030100101

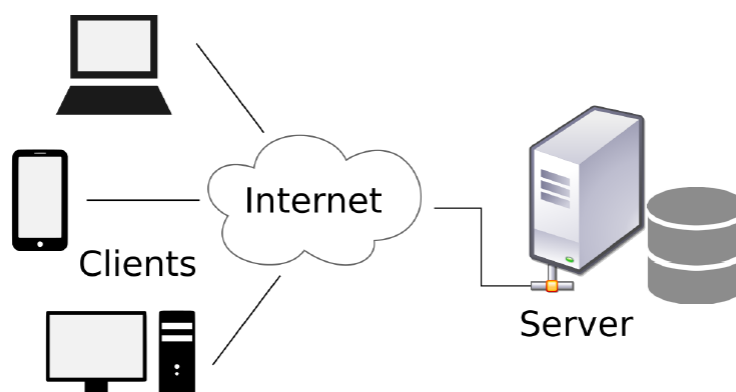
老师：李龙海

### 分布式计算分类

- 实时处理任务，也叫OLTP(Online Transaction Processing)任务，往往有高并发要求  
在线购物、在线交易等
- 准实时处理任务，流处理模型  
Web搜索时的联想、广告推送、商品推荐等
- 批处理任务，也叫OLAPOLTP(Online Analysis Processing)任务，往往只有单一用户。

### 架构模式

- 客户端-服务器(Client-Server)模式

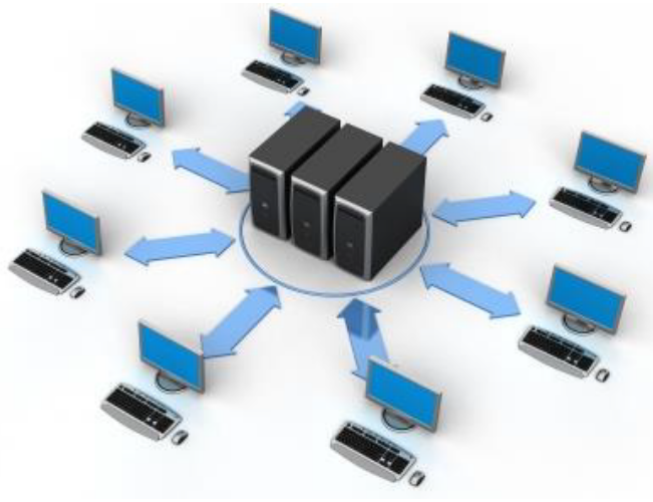


- 客户端发出服务请求，服务器端根据客户端请求参数完成实际运算，并将运算结果返回给客户端。
- 客户端运算任务轻，服务器端运算任务重。
- 客户端生命周期短，服务器端生命周期长。
- 服务器端一般要应对并发问题。
- 客户端一般负责和用户进行交互。
- 瘦客户端/胖客户端

Client-Cluster模式是Client-Server模式的变种。

服务器端由多个服务器构成，共同分担计算任务。

在宏观逻辑上，多个服务器构成的集群可以视为单一的功能更强大的计算节点。客户端感觉不到服务器端的实际构成。



负载均衡构架	常用的负载均衡策略
	<ul style="list-style-type: none"> <li><input type="checkbox"/> 随机</li> <li><input type="checkbox"/> 轮询</li> <li><input type="checkbox"/> 固定权重值</li> <li><input type="checkbox"/> IP哈希 (基于一致性随机散列函数)</li> <li><input type="checkbox"/> 最少TCP连接数</li> <li><input type="checkbox"/> 最小响应时间</li> <li><input type="checkbox"/> 基于各服务器实际负载的动态负载均衡算法</li> </ul>

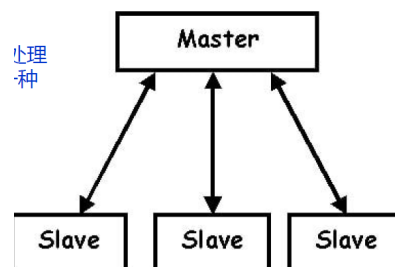
### • 主-从(Master-Slave)模式

高性能分布式计算/处理系统中经常采用的一种构架模式

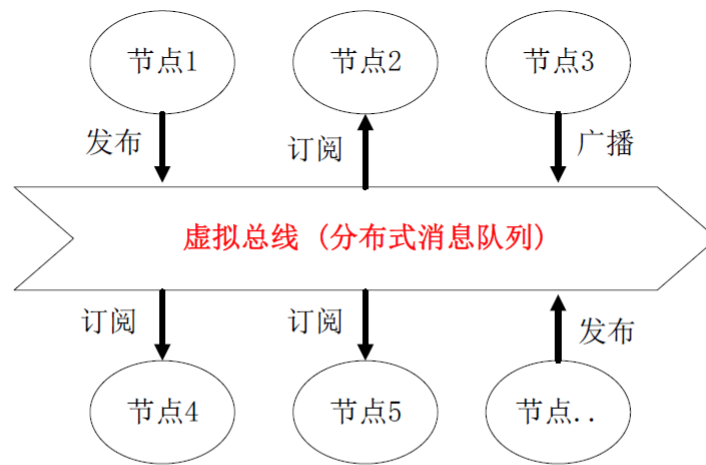
主节点 (Master) 负责将总计算任务分解为多个子任务分发给各个从节点 (Slave, 也叫Worker节点) 完成

主节点监视各个从节点的任务执行情况, 将执行失败的任务调度给其它的从节点完成

主节点在分配任务是会参考各个从节点的当前负载情况。

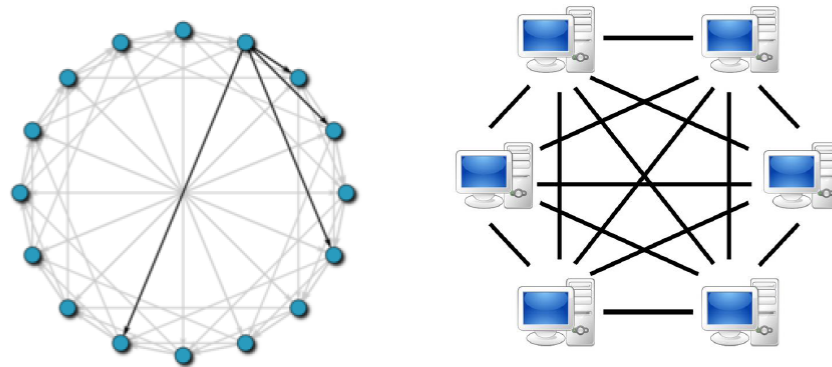


### • 总线模式



- 不同节点之间通过虚拟总线相连
- 消息发送者不必知道接收者是谁，接收者也不知道发送者是谁
- 发送者和接收者之间用异步方式通信
- 一种松耦合架构
- 不同节点完成不同功能，分工协作

• 对等(Peer-to-Peer)模式



- 系统中每个计算节点在任务分工上是完全对等的。
- 完全相同的软件在不同的计算机上运行，只是初始化参数不同
- 结构化P2P：不同节点之间的交互模式遵循固定规律
- 非结构化P2P：不同节点之间的交互模式没有固定规律

• 混合模式

## 中间件

	中间件的作用
<div style="border: 1px solid #ccc; padding: 10px;"> <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 5px;"> <span>Applications, services</span> <span>应用层</span> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 5px;"> <span style="border: 2px solid red; border-radius: 50%; padding: 2px;">Middleware</span> <span>中间件</span> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 5px;"> <span>Operating system</span> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>Computer and network hardware</span> </div> <div style="position: absolute; right: -20px; top: 50%; transform: translateY(-50%); font-size: 20px;">}</div> <div style="position: absolute; right: -20px; bottom: 0;">Platform</div> </div>	<p>为开发者提供高层的编程抽象，屏蔽分布式系统底层的异构性和复杂性</p> <p>提高互操作性和可移植性</p> <p>提供分布式系统的基础设施服务</p>

常用的中间件	中间件的表现形式
远程过程调用中间件	作为独立运行的后台进程存在
分布式对象中间件	作为运行时函数库/类库存在( LIB静态库、DLL动态库、jar包等)
分布式组件中间件	作为源代码级的函数库/类库存在
消息队列中间件	作为高级语言解释器的一部分存在
Web服务中间件	作为辅助编译工具存在
P2P中间件	作为高级语言编译器的一部分存在