**分布式系统**

**一、无状态的分布式**

对于无状态的分布式计算，不需要在线的主备来容错，如果一个计算节点宕机或者奔溃，可以重新启动一个计算节点来重新开始该计算任务。有许多分布式计算不区分对待主备。

分布式计算往往是因性能需求而设计，将庞大的计算任务分割为多个较小的计算任务，分配给多个计算节点来处理任务，最后合并中间的计算结果得到最终结果。

无状态的分布式计算有几个关键点：

1. 计算任务输入的存储和分割
2. 计算任务分割和合并
3. 如何分配计算任务到计算节点

三个典型的无状态分布式计算案例：

1. 词统计
2. 位置匹配
3. 图形运算

**词统计**

在搜索引擎中，需要统计所有文档中词出现的频率，文档是计算的输入，有大量的数据需要处理。这就是MapReduce框架要来解决的问题。

计算任务输入的数据是非常大的，如果把计算任务分配到计算节点，计算节点在读取计算任务的输入数据的话，这将在整个分布式计算网络产生大量的数据通信。

计算任务输入的数据是一个一个的文档，可以认为计算任务按照文档分割。计算任务按照文档进行分割后，单个计算任务统计单个文档的词频率，还需要计算任务的合并，将所有单个文档的词频率合并为所有文档的词频率。这需要保存单个计算任务输出的中间数据，用于后续的合并。同样面临合并计算任务输入的数据量大而产生的网络数据通信问题。

由于计算任务的输入数据量太大，在将计算任务分配大计算节点，由计算节点通过网络读取计算任务的输入数据是不可能的。这类分布式计算需要将某个计算任务分配到其输入的存储节点上执行计算任务。

计算节点和存储节点是耦合关系，计算任务的合并有大量的存储和计算要求。

**位置匹配**

在无线网络中，有一种定位算法，该算法要求事先测量一个位置的信号数据，这就是位置的测量数据，收集一个区域的所有位置的测量数据，形成测量数据库。

对于后续一个移动终端，根据其上报的测量数据，和事先收集的测量数据库中数据匹配，匹配度最高的测量数据对应的位置可以认为是当前移动终端的位置。

该分布式计算任务要求进行匹配运算。计算任务的输入是一个移动终端的测量数据和整个测量数据库，这里没有计算任务合并的需求。

如果避开具体应用来讲的话，计算任务的分割在这里是个难题。在无线网络中，有cell的概念，一个移动终端当前只属于一个cell，虽然有多个cell的信号和测量数据。

对于计算任务输入的测量数据库，按照cell将测量数据分割存储在多个存储节点上，根据移动终端所属于的cell，将该计算任务分配给该cell对应的存储节点上。这里没有计算合并需求。

如何进行计算任务的分割是关键。

**图形运算**

有一个位置数据库，记录了所有用户的当前位置，现在需要找出位于指定的圆形区域的用户。计算任务的输入是用户的位置信息和一个圆。

用户数量太大，判断所有点是否位于圆以内的计算是不现实的。

需要结合计算输入的圆R来分割位置数据库。将整个位置区域划分为多个子区域，将子区域的覆盖区定义为子区域向外扩展圆半径大小的区域，将位于该子区域的覆盖区的用户位置信息存储到该子区域对应的存储上。整个位置数据库按照子区域分割，当然这里有重复数据。

对于指定的圆形区域，可以判断其位于那个子区域，然后分配该计算任务到该子区域的存储节点上执行计算任务。

如何进行计算任务的输入的分割是关键。当然这种计算任务的输入的分割使得更新位置变得更加复杂。

**二、服务的主备分布式**

为了服务的高可靠，需要服务长时间在线，单个节点服务是不可能完成任务的，奔溃宕机等故障是时常发生的。

这需要主备的分布式服务，在主服务宕机后，备机可以启动提供服务。

三个典型的应用案例：

1. 有可靠存储保证的处理程序
2. 复杂计算程序
3. 数据同步程序

**有可靠存储保证的处理程序**

数据处理程序需要从指定的数据源读取数据，然后加工数据，写结果数据到存储。这个服务是一直在线提供服务的。

数据处理程序记录了当前加工数据的状态，如当前已经处理的数据时间位于2020:03:18 14:11:00。

当该数据处理程序宕机后，备机启动，从处理数据的位置状态继续处理数据。这要求有可靠的存储系统。

在一个时刻，只能最多有一个数据处理程序在服务，同时需要有一个数据处理程序在服务。

**复杂的计算程序**

这种应用比较简单，因为没有中间状态，在主机程序奔溃后，直接启动备机程序重新开始。

**数据同步程序**

当没有可靠的存储时，主机服务生成的数据需要有备份，需要使用备机来同步数据备份，在主机宕机不能提供服务时，有备机来提供服务，保证数据的可用和可靠。

这里有一个很重要的关键点：

1. 如何进行主备切换

我们看看有哪些实现：

1. 使用可靠的数据库来实现，基于可靠的数据库实现分布式锁，基于心跳机制实现主备以及其切换
2. 使用可靠的存储服务，如Redis来实现，基于Redis来实现分布式锁，基于心跳机制实现主备以及其切换
3. 使用可靠的集群服务，如基于Paxos算法实现的Zookeeper服务，基于Raft或者Paxos算法实现的Sentinel服务。

Redis就是基于Raft算法的sentinel服务来保证主备的。

**三、服务的主从分布式**

没有状态的主从分布式

有状态的主从分布式

**四、服务的主主分布式**

共识算法

**五、交互的的分布式**