

01 | 分布式缘何而起：从单兵，到游击队，到集团军

2019-09-23 聂鹏程 来自北京

《分布式技术原理与算法解析》



你好，我是聂鹏程。这是专栏的第一篇文章，我们先来聊聊什么是分布式。

与其直接用些抽象、晦涩的技术名词去给分布式下一个定义，还不如从理解分布式的发展驱动因素开始，我们一起去探寻它的本质，自然而然地也就清楚它的定义了。

在今天这篇文章中，我将带你了解分布式的起源，是如何从单台计算机发展到分布式的，进而帮助你深入理解什么是分布式。为了方便你更好地理解这个演进过程，我将不考虑多核、多处理器的情况，假定每台计算机都是单核、单处理器的。

分布式起源

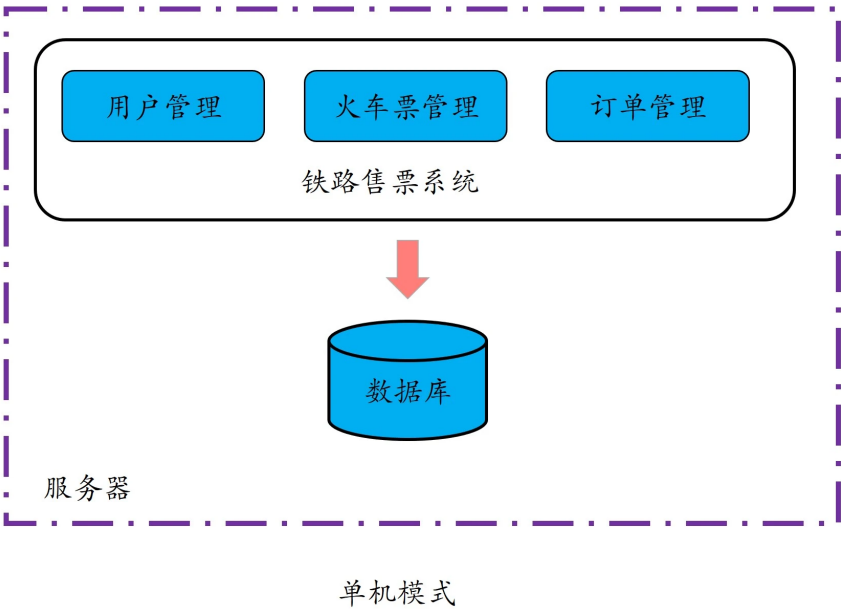
单兵模式：单机模式

1946 年情人节发布的 ENIAC 是世界上的第一台通用计算机，它占地 170 平米重达 30 吨，每秒可进行 5000 次加法或者 400 次乘法运算，标志着单机模式的开始。

所谓**单机模式**是指，所有应用程序和数据均部署在一台电脑或服务器上，由一台计算机完成所有的处理。

以铁路售票系统为例，铁路售票系统包括用户管理、火车票管理和订单管理等模块，数据包括用户数据、火车票数据和订单数据等，如果使用单机模式，那么所有的模块和数据均会部署在同一台计算机上，也就是说数据存储、请求处理均由该计算机完成。这种模式的好处是功能、代码和数据集中，便于维护、管理和执行。

单机模式的示意图，如下所示：



这里需要注意的是，本文的所有示意图中，紫色虚线表示在一台计算机内。

事物均有两面性，我们再来看看单机模式的缺点。单个计算机的处理能力取决于 CPU 和内存等，但硬件的发展速度和性能是有限的，而且升级硬件的性价比也是我们要考虑的，由此决定了 CPU 和内存等硬件的性能将成为单机模式的瓶颈。

你有没有发现，单机模式和单兵作战模式非常相似，单台计算机能力再强，就好比特种兵以一敌百，但终归能力有限。此外，将所有任务都交给一台计算机，也会存在将所有鸡蛋放到一个篮子里的风险，也就是单点失效问题。

归纳一下，单机模式的主要问题是：**性能受限、存在单点失效问题。**

游击队模式：数据并行或数据分布式

既然单机模式存在性能和可用性的问题。那么，有没有什么更好的计算模式呢？答案是肯定的。

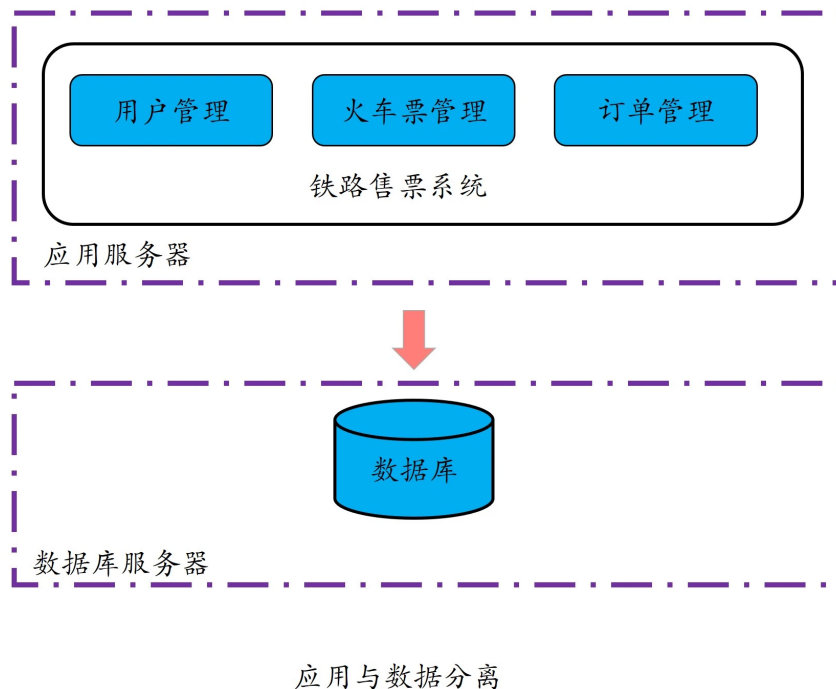
为解决单机模式的问题，并行计算得到了发展，进而出现了数据并行（也叫作数据分布式）模式。**并行计算**采用消息共享模式使用多台计算机并行运行或执行多项任务，核心原理是每台计算机上执行相同的程序，将数据进行拆分放到不同的计算机上进行计算。

请注意，并行计算强调的是对数据进行拆分，任务程序在每台机器上运行。要达到这个目的，我们必须首先把单机模式中的应用和数据分离，才可能实现对数据的拆分。这里的应用就是执行任务的程序，任务就是提交的请求。以铁路售票系统为例，运行在服务器上的用户管理、火车票管理和订单管理等程序就是应用，用户提交的查询火车票、购买火车票的请求就是任务。

在单机模式中，应用和数据均在一台计算机或服务器上，要实现数据的并行，首先必须将应用和数据分离以便将应用部署到不同的计算机或服务器上；然后，对同类型的数据进行拆分，比方说，不同计算机或服务器上的应用可以到不同的数据库上获取数据执行任务。

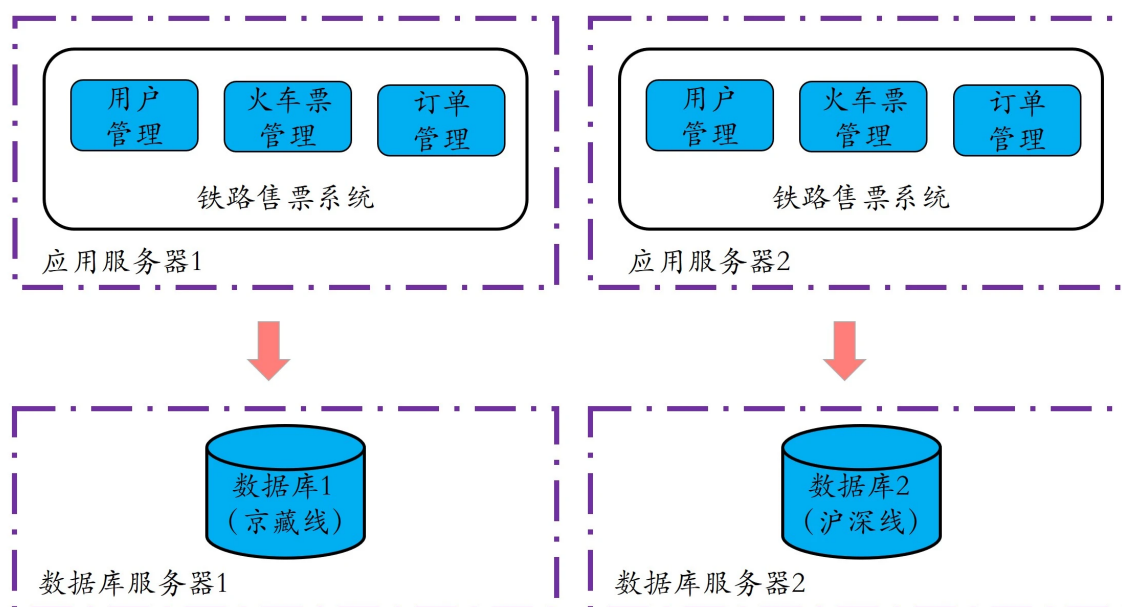
以铁路售票系统的数据并行为例，主要包括两个步骤，如下所示：

第一步，将应用与数据分离，分别部署到不同的服务器上：



第二步，对数据进行拆分，比如把同一类型的数据拆分到两个甚至更多的数据库中，这样应用服务器上的任务就可以针对不同数据并行执行了。

对于铁路售票系统来说，根据线路将用户、火车票和订单数据拆分到不同的数据库中，部署到不同的服务器上，比如京藏线的数据放在数据库服务器 1 上的数据库中，沪深线的数据放在数据库服务器 2 上的数据库中。



需要注意的是，为了更好地帮助你理解这个数据拆分的过程，我在这里选择拆分数据库的方式进行讲解。由于数据库服务器本身的并发特性，因此你也可以根据你的业务情况进行选择，比方说所有业务服务器共用一个数据库服务器，而不一定真的需要去进行数据库拆分。

可以看出，在数据并行或数据分布式模式中，每台计算机都是全量地从头到尾一条龙地执行一个程序，就像一个全能的铁道游击队战士。所以，你也可以将这种模式形象地理解成游击队模式，就和铁道游击队插曲的歌词有点类似：“我们扒飞车那个搞机枪，撞火车那个炸桥梁.....”

这种模式的好处是，可以利用多台计算机并行处理多个请求，使得我们可以在相同的时间内完成更多的请求处理，解决了单机模式的计算效率瓶颈问题。但这种模式仍然存在如下几个问题，在实际应用中，我们需要对其进行相应的优化：

相同的应用部署到不同的服务器上，当大量用户请求过来时，如何能比较均衡地转发到不同的应用服务器上呢？解决这个问题的方法是设计一个负载均衡器，我会在“分布式高可靠”模块与你讲述负载均衡的相关原理。

当请求量较大时，对数据库的频繁读写操作，使得数据库的 IO 访问成为瓶颈。解决这个问题的方式是读写分离，读数据库只接收读请求，写数据库只接收写请求，当然读写数据库之间要进行数据同步，以保证数据一致性。

当有些数据成为热点数据时，会导致数据库访问频繁，压力增大。解决这个问题的方法是引入缓存机制，将热点数据加载到缓存中，一方面可以减轻数据库的压力，另一方面也可以提升查询效率。

从上面介绍可以看出，数据并行模式实现了多请求并行处理，但如果单个请求特别复杂，比方说需要几天甚至一周时间的时候，数据并行模式的整体计算效率还是不够高。

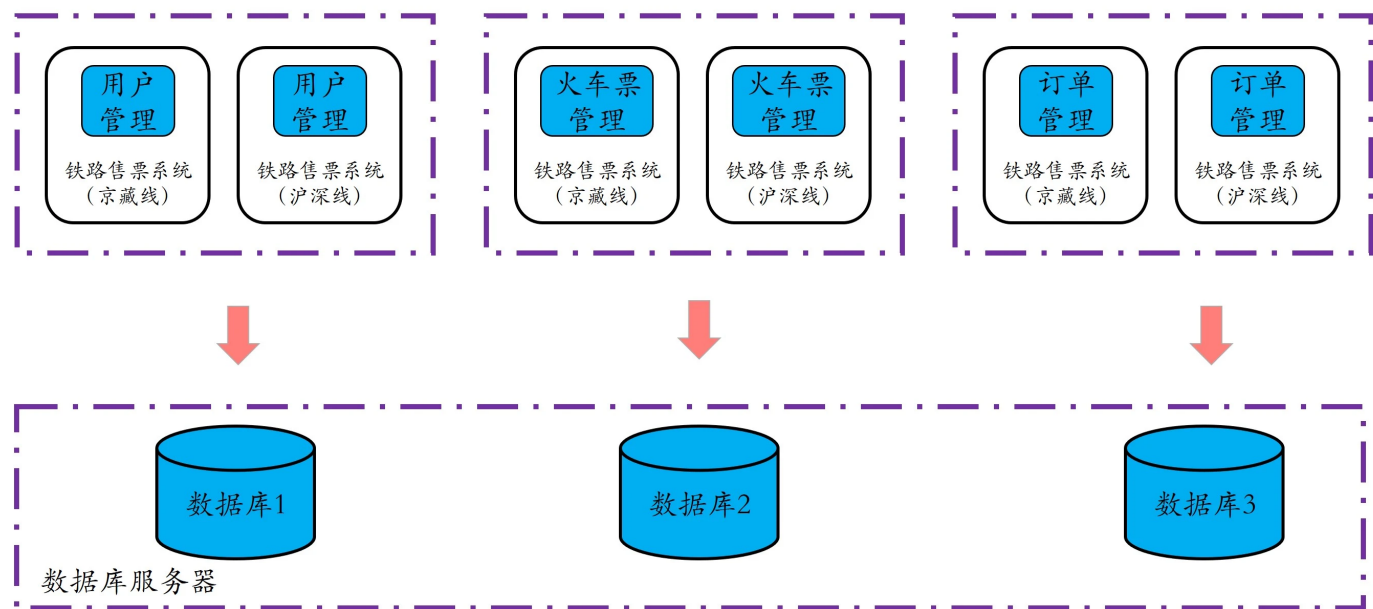
由此可见，数据并行模式的主要问题是：**对提升单个任务的执行性能及降低时延无效。**

集团军模式：任务并行或任务分布式

那么，有没有办法可以提高单个任务的执行性能，或者缩短单个任务的执行时间呢？答案是肯定的。任务并行（也叫作任务分布式）就是为了解决这个问题而生的。那什么是任务并行呢？

任务并行指的是，将单个复杂的任务拆分为多个子任务，从而使得多个子任务可以在不同的计算机上并行执行。

我们仍以铁路售票系统为例，任务并行首先是对应用进行拆分，比如按照领域模型将用户管理、火车票管理、订单管理拆分成多个子系统分别运行在不同的计算机或服务器上。换句话说，原本包括用户管理、火车票管理和订单管理的一个复杂任务，被拆分成了多个子任务在不同计算机或服务器上执行，如下图所示：



任务并行（任务分布式）模式

可以看出，任务并行模式完成一项复杂任务主要有两个核心步骤：首先将单任务拆分成多个子任务，然后让多个子任务并行执行。这种模式和集团军模式很像，任务拆分者对应领导者，不同子系统对应不同兵种，不同子程序执行不同任务就像不同的兵种执行不同的命令一样，并且运行相同子系统或子任务的计算机又可以组成一个兵团。

在集团军模式中，由于多个子任务可以在多台计算机上运行，因此通过将同一任务待处理的数据分散到多个计算机上，在这些计算机上同时进行处理，就可以加快任务执行的速度。因为，只要一个复杂任务拆分出的任意子任务执行时间变短了，那么这个任务的整体执行时间就变短了。

当然，nothing is perfect。集团军模式在提供了更好的性能、扩展性、可维护性的同时，也带来了设计上的复杂性问题，毕竟对一个大型业务的拆分也是一个难题。不过，对于大型业务来讲，从长远收益来看，这个短期的设计阵痛是值得的。这也是许多大型互联网公司、高性能计算机构等竞相对业务进行拆分以及重构的一个重要原因。

分布式是什么？

讲了半天，那到底什么是分布式呢？

总结一下，分布式其实就是将相同或相关的程序运行在多台计算机上，从而实现特定目标的一种计算方式。

从这个定义来看，数据并行、任务并行其实都可以算是分布式的一种形态。从这些计算方式的演变中不难看出，**产生分布式的最主要驱动力量，是我们对于性能、可用性及可扩展性的不懈追求。**

总结

在今天这篇文章中，我和你分享了分布式的起源，即从单机模式到数据并行（也叫作数据分布式）模式，再到任务并行（也叫作任务分布式）模式。

单机模式指的是，所有业务和数据均部署到同一台机器上。这种模式的好处是功能、代码和数据集中，便于维护、管理和执行，但计算效率是瓶颈。也就是说单机模式性能受限，也存在单点失效的问题。

数据并行（也叫作数据分布式）模式指的是，对数据进行拆分，利用多台计算机并行执行多个相同任务，通过在相同的时间内完成多个相同任务，从而缩短所有任务的总体执行时间，但对提升单个任务的执行性能及降低时延无效。

任务并行（也叫作任务分布式）模式指的是，单任务按照执行流程，拆分成多个子任务，多个子任务分别并行执行，只要一个复杂任务中的任意子任务的执行时间变短了，那么这个业务的整体执行时间也就变短了。该模式在提高性能、扩展性、可维护性等的同时，也带来了设计上的复杂性问题，比如复杂任务的拆分。

在数据并行和任务并行这两个模式的使用上，用户通常会比较疑惑，到底是采用数据并行还是任务并行呢？一个简单的原则就是：任务执行时间短，数据规模大、类型相同且无依赖，则可采用数据并行；如果任务复杂、执行时间长，且任务可拆分为多个子任务，则考虑任务并行。在实际业务中，通常是这两种模式并用。赶紧行动起来，去分析一下你的业务到底应该采用哪种分布式模式吧，加油！

课后思考

你觉得分布式与传统的并行计算的区别是什么？你可以结合多核、多处理器的情况进行思考。

我是聂鹏程，感谢你的收听，欢迎你在评论区给我留言分享你的观点，也欢迎你把这篇文章分享给更多的朋友一起阅读。我们下期再见！

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (76)



吾皇万岁万万岁

2019-09-23

之前去会所，一口气叫上十个嫩模，到房间之后，我就给她们讲Quorum协议怎么实现节点选举，蒙特卡洛方法的实际应用，MVCC怎么实现事务的并发，tcp怎么做到传输可靠...她们很感兴趣，我也很愉快的就回来了。

后来我朋友听了，他们也想去，于是去了五个人（五人同行，两人免单），这次可以服务50个嫩模，每人10个，在不同的房间，我们又给她们讲了Quorum协议...蒙特卡洛...MVCC...tcp传输等等，她们很开心，我们也开心的回来了。

再下次的时候，也去了5个人，但这次我们想服务更多的嫩模，于是我们进行了现场分组，想听Quorum的和第一个朋友走，想听蒙特卡洛的和第二个朋友走，想听MVCC的和第三个朋友走.....于是我们每个人只讲一个知识点就回来了，很轻松也很开心，所以我们经常去会所。

上面三部分，依次单机模式、数据并行、计算并行。

后来，会所里面的姑娘都有了正经的职业，叫程序员鼓励师！

作者回复: 你确定你开的不是时光机吗? 为什么读着你的文字总有种移步换景, 有种穿越的错觉? :)

共 27 条评论 >

👍 219



1024

2019-09-24

分布式起源

单兵模式: 单机

将服务和数据都部署到同一个节点(服务器)上

存在的问题: 单点故障, 单点性能瓶颈

游击队模式: 数据分布式

实际上是水平分层

1. 数据分层

2. 将数据分离

就是针对不同的请求, 去路由到不同的数据节点上

存在问题: 对提升单个任务的执行性能及降低时延无效

集团军模式: 任务分布式

实际上是垂直分层

拆分服务到不同节点, 一个任务可能要请求多个服务, 多个服务并行执行

类似于不同兵种, 各司其职, 协同作战

存在问题: 数据的全局一致性(各节点通信), 数据的可用性保证(副本), 各节点的调度(哪个节点可以响应服务)

总结

我理解的分布式: 将一个任务, 拆分成多个子任务, 每个任务都对应一个服务, 每个服务又都部署到不同的节点上, 实现一个任务, 多服务并行响应, 最后通过调度、统筹返回请求的结果
为什么会有分布式: 是我们对性能、可用性、及可扩展性的不懈追求

课后问题

传统的并行计算是时空上复用多处理器

分布式是空间上给任务分配多处理器

作者回复: 积跬步, 而终至千里! 加油!

共 3 条评论 >

👍 30



臧卞嘤
2019-09-23

"分布式其实就是将相同或相关的程序运行在多台计算机上，从而实现特定目标的一种计算方式。"

这就相当于说 "分布式是一种计算方式"，感觉有点别扭，套用到分布式协调、分布式计算、分布式存储上，感觉更别扭。

作者回复: 从你的留言，能看出你有勤于思考的习惯！

我想你如果理解下面两个问题，你的困惑也就消除了：

1. 为什么个人电脑、服务器、平板、手机等都可以叫做计算机，而不是叫做存储机或其它机？
2. 为什么大学里一级学科一般都只有计算机科学与技术，而没有什么存储科学与技术或协调科学与技术？

共 3 条评论 >

👍 18



longhaiqwe

2019-10-07

首先，单核是无法实现真正的并行计算的，更多的是使用时间分片的方式，在各个“计算”之间跳动执行。

然后，多核是可以实现真正的并行计算的。那我们为何不在机器上增加CPU和内存，而是去研究分布式这样的技术呢？

我的思考哈，因为一台机器做好了放那儿，硬件是已经确定了的。假如我们需要更好的性能，也就是更多的CPU和内存，我们当然可以把机器拆了，然后现场改板子加CPU和内存，不过明显我只是为了强调不可行才说的“当然可以”。分布式的作用之一就在于我们不需要改变原来的机器，而是增加一台机器来达到增加CPU和内存的目的。这就是我理解的“可扩展性”，分布式能带来“高性能”无疑，不过不考虑在原来机器上增加CPU和内存的难度，我觉得不使用分布式技术也是可以做到的。

最后，假设我们在一台机器上增加CPU和内存是不需要任何成本的（服务不断），也存在假设这台机器挂了，我们的服务也就会中断的问题。而使用分布式，也就是多台机器，即使其中一台挂了，还有其他机器可以对外提供服务。这就是我理解的“可用性”。

共 1 条评论 >

👍 12



c-xyy

2019-10-13

分布式强调拆分，如果面对的问题的瓶颈在数据量，则讲焦点集中在数据的拆分上，也就是数据分布式；如果问题的瓶颈在任务执行逻辑的复杂性上，则将任务进行拆分，也就是计算的分布式；讲相同的逻辑部署在多个节点上，一般称之为集群，平时我们经常说分布式集群系统，实际的包含了底层任务的拆分和多机部署的含义。



11



zhaozp

2019-09-29

打卡文章学习：

1、分布式是系统架构不断演进的结果，是对系统性能不断追求的结果。单机模式的缺点是性能瓶颈并且有单点故障。这时系统中往往最先出现的数据瓶颈所以先出现数据并行（分库分表，读写分离）等，任务也会水平扩展增加节点通过负载均衡路由，缺点就是对提升单个任务的性能和时延无效。随着系统的复杂性提升，任务的执行成为系统瓶颈，这时需要对任务进行拆分，拆分成小粒度的子任务，容器技术也实现了子任务单独部署，当然缺点是引入了系统的复杂度。

2、分布式其实就是将相同或相关的程序运行在多台计算机上，从而实现特定目标的计算方式。结合"四横"：数据并行就是对数据存储和管理的分布式化、任务并行就是对计算方式的分布式化，也会涉及资源池的分布式和通信的分布式。结合"四纵"：数据和任务拆分后怎么解决分布式协同、分布式调度、分布式高可用的问题，容器化技术也致力于解决分布式部署问题。

课后思考：

我理解传统的并行计算还是运行于单服务器，其核心在于充分利用服务器的CPU、内存等资源，可以通过多线程并发技术实现。而分布式是把任务拆分到不同的服务器上，是业务解耦，提高可用性和可扩展性。但是也引入了系统设计复杂度。

作者回复：积跬步，而终至千里！加油！



9



Dale

2019-09-25

分布式核心在于将任务拆解分布在不同的服务器上，业务解耦，提高性能和可用性。并行计算核心在于充分利用服务器cpu，内存，网络资源，可以将单个任务以多进程多线程方式来运行。分布式中采用并行计算可以提高性能。



8



老男孩

2019-09-24

传统的并行计算，拿单核来说就算通过CPU的指令在多个进程或者线程间切换来实现宏观上的并行，微观上还是串行的。多核是可以真正实现并行的。传统的并行计算是同一个主机的不同进程或者线程，分布式是不同物理主机中的进程或者线程。



6



景梦园

2020-05-11

世界上的任何事物都是在做Trade-Off，根据奥卡姆剃刀原则，能用简单的就不用复杂的。虽然在学分布式，但是也要看到三种模式各有优劣的。另外最近也在学并发编程，感觉并发编程是“微观”的分布式，系统级是“宏观”的，触类旁通。

作者回复: 🍊🍊🍊



4



djfhchdh

2019-09-24

并行计算强调运算的同时进行，对于多核处理器，就是每个处理器都同时进行运算；而分布式更多是从数据、任务的并行上来考虑，通常是利用多台机器，较少考虑多处理器并行的问题



3



向往的生活

2019-09-23

厉害

作者回复: 宝剑锋自磨砺出，加油! :)



3



seker

2021-07-08

单机模式是指所有应用和数据均部署在一台服务器上，由这一台服务器来完成所有处理。
个人理解：应用和数据库在不拆分的情况下，同时部署在一台服务器上，或分别部署在一台服务器上(应用一台服务器、数据库一台服务器)，均可视为单机模式。
优势：应用和数据集中，好维护、好管理、好执行。
劣势：从硬件层面说，CPU、内存、磁盘均有瓶颈，不可能无限制的升级，升级还需要考虑性价比，同时硬件也存在损耗。从软件层面说单应用单数据库如果因异常挂掉，那么将导致服务不可用，对用户使用造成影响，对公司业务造成损失。综上劣势归纳为一句话就是会存在单点失效问题。

数据并行/数据分布式

数据不一定是指数据库，应用中是否包含数据？答案是包含的，比如从页面传递给服务端接口的数据、服务端接口返回给页面用于展示的数据、接口中计算产生的数据、接口为具体业务封装的可以装载数据的数据结构等。因此数据不仅存在于数据库中，也存在于应用中。只不过数据库中数据是做了持久化到磁盘的。

这种模式的一个关键点是利用多台计算机并行处理多个请求，且每台计算机都完整的执行一个请求。

优势：解决单机模式计算效率瓶颈问题，相同时间内可以完成更多的请求处理。

劣势：无法提升单个任务的执行性能及降低时延。

任务并行/任务分布式

将单个复杂的任务拆分为多个子任务，从而使得多个子任务可以在不同的计算机上并行执行。多个子任务中独立的子任务(没有对其它子任务强依赖的任务)，可以在同一时刻被执行，这样就使得原本单个复杂的任务执行时间变短了。

优势：提高了任务执行性能、降低了延时

劣势：带来了设计上的复杂性，即业务拆分难题



fomy

2019-11-16

如果系统的瓶颈在数据量太大，则对数据进行分片，如MySQL的数据分片和Redis Cluster。

如果系统的瓶颈在于业务复杂，耦合度太高，则将任务拆分到各个微服务，分别部署到不同服务器上，这就是常说的分布式微服务。

如果系统的瓶颈在于单个微服务负载过高，难以处理高并发请求时，则使用集群部署，同一个微服务有多个实例，应该也可以叫做并行计算吧。



阿亮

2019-10-02

任务并行不是特别理解。如果有一个任务有多个任务组成，且是顺序执行的，前后有依赖，在这种情况下，无论怎么拆分也是不可能提高效率的吧，如果放到不同机器上，是否还会降低执行效率？

共 2 条评论 >



流云

2020-03-03

传统的并行计算通常是在一个计算机内，共享内存等设备，多个CPU直接的通讯走的是总线，这种通讯时延低可靠性也强；对于分布式系统，多个计算机是需要依赖网络来通讯，而计算机

网络通常是不太可靠的，在物理距离内有不可逾越的延迟时间。



👍 1



钱

2020-02-12

阅过留痕

哈哈，和自己的理解一致，假设单机的容量无限，性能极佳，永不宕机，那就没有分布式啥事了啦！并且整个宇宙只需要一台计算机就行了，可惜假设不成立，但人类又希望有这样一台神一样的计算机，所以，分布式就出现了。

侠义的容量无限，性能极佳，永不宕机也就实现了，一台容量不够来两台，一台不够快来两台（要想快，必须并行着来，大任务或数据必须是可拆分的，且拆分后可以并行着来）一台挂了另一台顶上。

其他的一切都是围绕怎么拆？怎么合？怎么协调来做文章的啦！



👍 1



张理查

2019-12-20

分布式技术原理与算法#Day2

今天打卡了第一节和第二节课，解决了两个议题，第一个是究竟什么算分布式，第二个是如何评价分布式系统。

首先我们将系统抽象成代码和数据。所谓单机模式就是部署在一台机器上，但第一是性能有瓶颈，第二是挂了系统就没了，存在单点问题。我们解决这个问题的思路是分数据和分代码：分数据，又叫并行计算，就是将数据分多台机器，跑同样的代码，这样好处是解决了性能瓶颈，但有三个问题，一是分配均匀的问题，可以引入负载均衡器，二是分布后的数据如果单一入口访问，会有IO瓶颈，可以通过读写分离解决，其实就是增加入口，三是热点数据的问题，我们通过缓存机制可以解决。分数据相当于通过增加员工数量解决项目过多的问题，但无法解决单个项目的完成时间，这里假设一个人干一个项目。分代码，就是分解项目分工，通过增加工种数量来提升单个项目的方式，因此分代码包括业务拆分和子业务并行两个步骤，好处就是可以找到代码瓶颈，有的放矢，但这么做的代价就是引入了复杂性，毕竟工种多了，多个人干一个项目就有协调的问题。

那么什么是分布式呢？其实就是我们不满足单个机器的性能，可用性，因而将代码用多台机器来跑，如果任务简单就是数据量大，那我们就拆数据，代码不变，一起跑。如果任务复杂，那我们就拆代码，有性能瓶颈的任务多分点机器干。

当然，你肯定会想又切数据又切代码。这在生产也很常见。

作者回复：赞，举一反三，触类旁通！



旅途

2019-10-13

数据并行例子中，上面两个服务访问下面的数据库是不是应该是交叉的，每个服务都可以访问下面两个数据库,否则就是服务和地点有绑顶关系，按你接下来说的请求转发，那么就是根据查询的地点来指定特定的服务器，而不是负载均衡了



leslie

2019-09-23

关于分布式和并行计算的区别：

1) 从虚拟化这块来解释：分布式其实就是我们现在所最常用的Kubernetes或者docker,传统的并行就是Vmware

2) 从数据系统这块来解释就是：RMDB和no sql DB的区别；RMDB严格遵守ACID特性，nosql 做不到只能在CAP中选择其中的2项。

不知道这样的解释是否算是一种解释吧。



亦知码蚁

2019-09-23

老师举的数据并行的例子 不是任务并行吗 让他们处理各自的地区的订单 只不过没把任务划分成更小的子任务

共 1 条评论 >

