The Datacenter as a Computer

——An Introduction to the Design of Warehouse-Scale Machines

Chracter1 介绍

ARPANET 已经有 40 多年的历史了，the World Wide Web 最近也庆祝了 20 周年。然而，互联网技术在很大程度上是由这两个显著的里程碑所激发的，如今仍在继续改变着行业和我们的文化，没有任何放缓的迹象。基于 web 的电子邮件、搜索和社交网络等流行互联网服务的出现，加上全球范围内高速连接的可用性的提高，加速了服务器端或云计算的发展趋势。

WSCs 目前为谷歌、亚马逊、Facebook 和微软在线服务部门等公司提供的服务提供动力。它们与传统的数据中心有很大的不同:它们属于单个组织，使用相对同类的硬件和系统软件平台，并且共享一个公共的系统管理层。通常，与在传统数据中心中运行的第三方软件的优势相比，大部分应用程序、中间件和系统软件都是内部构建的。最重要的是，WSCs 运行的非常大的应用程序(或 Internet 服务)数量较少，而公共资源管理基础结构允许显著的部署灵活性。基于同质化、单组织控制的要求，以及对成本效率的关注，促使设计师在构建和操作这些系统时采取新的方法。

Chracter3 硬件构建块

如前所述，仓库级计算机(WSCs)的体系结构在很大程度上是由构建块的选择决定的。这个过程类似于选择实现微处理器的逻辑元素，或者在构建服务器平台时选择正确的芯片组和组件。在本例中，主要构建块是服务器硬件、网络结构和存储层次结构组件。在本章中，我们将重点讨论服务器硬件的选择，目的是为如何做出这样的选择建立直觉。

3.1 性价比高的服务器硬件

低端服务器集群是目前 WSCs 的首选构建块。出现这种情况的原因有很多，其中最主要的一个原因是低端服务器的潜在成本效率，与高端共享内存系统相比，高端共享内存系统早先

是高性能和技术计算空间的首选构件。低端服务器平台与非常庞大的个人电脑市场共享许多关键组件，因此更能从规模经济中获益。

3.1.1 大规模 SMP 通信效率的影响

图 3.1 展示了随着 SMP 节点数量的增加，并行任务的执行时间为三个级别的通信强度。

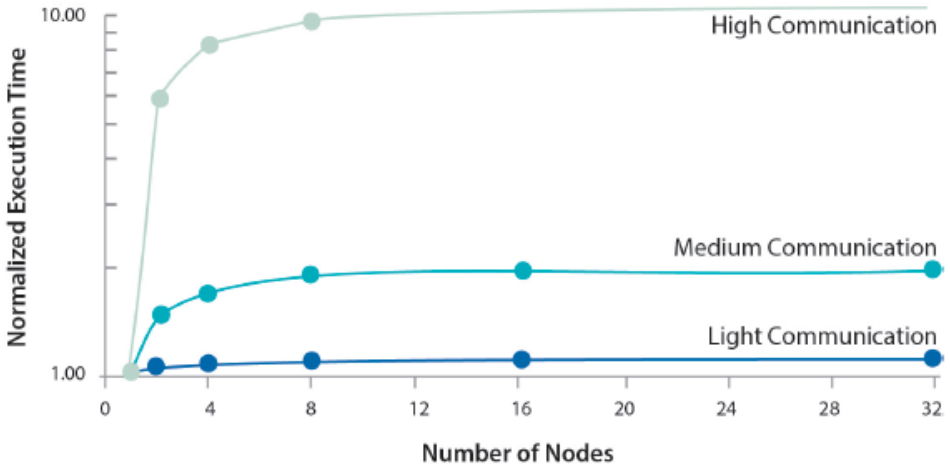

Figure 3.1: Execution time of parallel tasks as the number of SMP nodes increases for three levels of communication intensity. Execution time is normalized to the single node case and plotted in logarithmic scale.

图 3.2 展示了对于大小不同的集群，使用大型 SMP 服务器节点(128 核 SMP)构建的集群比使用低端服务器节点(4 核 SMP)构建的处理器核数量相同的集群具有性能优势。

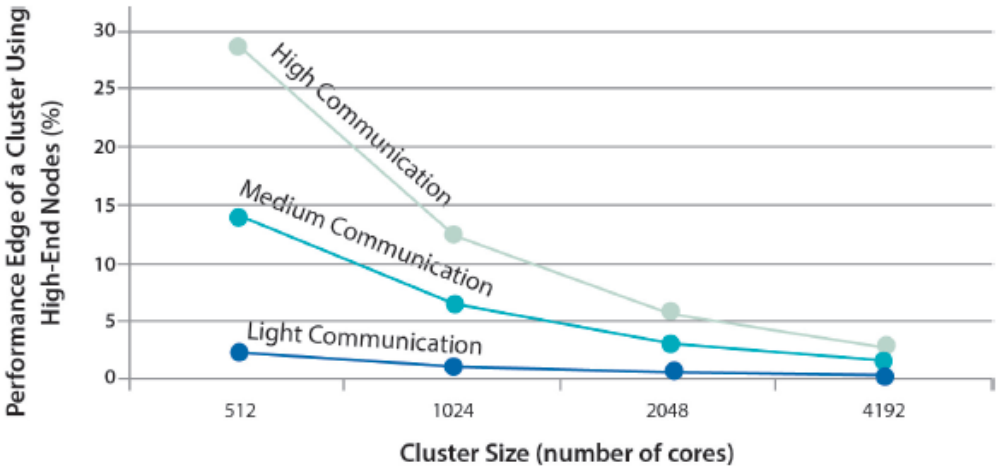

Figure 3.2: Performance advantage of a cluster built with large SMP server nodes (128-core SMP) over a cluster with the same number of processor cores built with low-end server nodes (four-core SMP), for clusters of varying size.

3.1.2 健壮的 VS 弱的服务器

使用更小、更慢的 cpu 的优点与使用中档商用服务器而不是高端 smp 的优点非常相似:

- 中档服务器中的多核 cpu 通常比低端处理器具有更高的价格/性能优势，因此同样数量的吞吐量可以用多个更小的 cpu 以更低的价格买到两到五倍的价格
- 许多应用程序是内存或 I/ o 绑定的，因此对于大型应用程序，速度更快的 cpu 不能很好地进行扩展，从而进一步增强了简单 cpu 的价格优势
- 较慢的 cpu 效率更高;通常情况下，CPU 功耗降低 O(k2)时，CPU 频率降低了 k 倍

3.1.3 平衡的设计

因为数据的大小和服务的普及也起着重要的作用。例如，具有庞大数据集但请求流量相对较小的服务可能能够直接从磁盘驱动器中提供其大部分内容，而磁盘驱动器的存储成本较低(以 s/GB 计算)，但吞吐量较低。非常流行的服务，要么具有较小的数据集大小，要么具有较大的数据局部性，可以从内存服务中获益。最后，这个领域的工作负载波动也是 WSC 架构师面临的挑战。软件基础可能发展得如此之快，以至于服务器设计选择在其生命周期(通常为 3 - 4 年)中变得不太理想。对于整个 WSC 来说，这个问题更加重要，因为数据中心设施的生命周期通常跨越几个服务器生命周期，或者超过 10 年左右。在这些情况下，尝试设想在 WSC 系统的生命周期内可能需要的各种机械或设施升级，并在设施的设计阶段考虑到这一点，是很有用的。

3.2 WSC 存储

WSC 工作负载操作的数据通常分为两类:对单个运行任务私有的数据和分布式工作负载共享状态的数据。私有数据倾向于驻留在本地 DRAM 或磁盘中，很少被复制，并且通过其单个用户语义简化了管理。与此相反，共享数据必须更加持久，并由大量客户端访问，因此需要更复杂的分布式存储系统。接下来我们将讨论这些 WSC 存储系统的主要特性。

3.2.1 无结构的 WSC 存储

谷歌的 GFS 是一个具有简单的类文件抽象的存储系统示例。GFS 的设计目的是支持 Web 搜索索引系统(将爬行的 Web 页面转换为索引文件用于 Web 搜索的系统)，因此它关注数千个并发阅读器/写入器的高吞吐量以及在高硬件故障率下的健壮性能。GFS 用户通常操作大量的数据，因此，GFS 对大型操作进行了进一步的优化。系统体系结构由一个主进程(处理元数据操作)和数千个从进程组成，这些进程在每个具有磁盘驱动器的服务器上运行，以管理这些驱动器上的数据块。在 GFS 中，容错功能是通过跨机器复制而不是在机器内部提供的，就像 RAID 系统中的情况一样。跨机器复制允许系统容忍机器和网络故障，并允许快速恢复，因为给定磁盘或机器的副本可以分布在数千台其他机器上。

3.2.2 有结构的 WSC 存储

对于操作大量数据的系统来说，GFS 和 Colossus 的简单文件抽象可能足够了，但是应用程序开发人员还需要类似于数据库的 WSC 功能，在这种功能中，数据集可以结构化并建立索引，以便进行简单的小更新或复杂查询。像谷歌的 BigTable 和 Amazon 的 Dynamo 这样的结构化分布式存储系统就是为了满足这些需求而设计的。与传统的数据库系统相比，BigTable 和 Dynamo 牺牲了一些特性，比如丰富的模式表示和强一致性，以获得更大的性能和可用性。例如，BigTable 提供了一个简单的多维排序映射，它由行键(字符串)组成，与列中组织的多个值相关联，形成一个分布式稀疏表空间。列值与时间戳关联，以支持版本控制和时间序列。

3.2.3 存储和网络技术的相互作用

WSC 分布式存储系统的成功部分归功于数据中心网络结构的发展。Ananthanarayanan 等人通过观察到网络和磁盘性能之间的差距已经扩大到磁盘局部性不再与数据中心计算相关的程度。这种观察使基于分布式磁盘的存储系统的设计大大简化，并提高了利用率，因为 WSC 设施中的任何磁盘字节原则上都可以被任何任务使用，而不管它们的相对位置如何。Flash 作为一种可行的分布式存储系统企业存储设备技术的出现，对数据中心网络结构提出了新的挑战。单个企业闪存设备可以实现磁盘驱动器操作吞吐量的 100 倍以上。这样的性

能水平不仅会延长数据中心 fabric 平分带宽，而且还需要存储节点中的更多 CPU 资源来以如此高的速率处理存储操作。因此，与今天的 flash 局部仍然相当相关。

3.3 WSC 网络

图 3.3 展示了我们可以通过级联这些交换机芯片来构建更大的交换机。

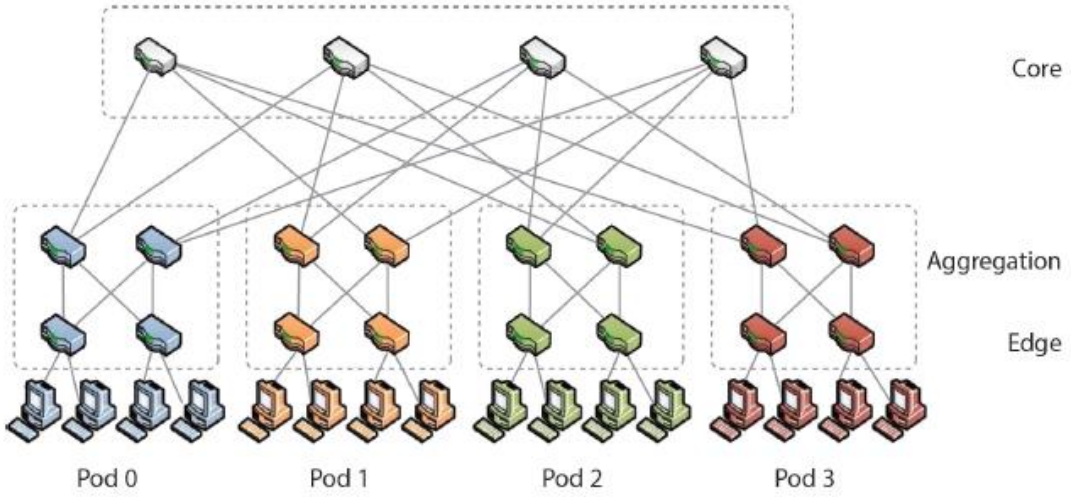

Figure 3.3: Sample three-stage fat tree topology. With appropriate scheduling, this tree can deliver the same throughput as a single-stage crossbar switch. (Image courtesy of Amin Vahdat.)