抽象

今天的数据中心为数万台机器的集群提供了巨大的聚合带宽。 然而，由于即使是最高端交换机，端口也是有限的，数据中心拓扑结构通常由多根树组成，在任意一对主机之间具有许多等价路径。 现有的IP多路径协议通常依赖于每个流的静态散列，并且由于长时间的冲突而导致相当大的带宽损失。

在本文中，我们介绍了Hedera，一种可扩展的动态流调度系统，可以自适应地调度多级交换结构，以有效利用聚合网络资源。 我们使用商品交换机和未修改的主机来描述我们的实现，并且显示对于一个模拟的8,192主机数据中心，Hedera提供了比静态负载平衡方法优于96％的高达113％的二等分带宽。

1.简介

以几年前还无法预见的速度和规模，大型组织正在建设庞大的数据中心，支持数万台机器;其他公司正在将其计算，存储和运营等服务都转移到云计算托管服务提供商。许多应用程序（从商品应用程序托管到科学计算，网页搜索和MapReduce）都需要大量的集群内带宽。随着数据中心及其应用程序规模的不断扩大，扩展网络结构的容量以实现潜在的全面通信成为一项特殊的挑战。

基于云的应用程序有几个属性使得数据中心网络设计难以解决。首先，数据中心工作负载对于网络设计师来说是先验未知的，并且在时间和空间上可能会变化。结果，静态资源分配不足。其次，客户希望在商品操作系统上运行他们的软件;因此，网络必须提供高带宽而不需要软件或协议的改变。第三，虚拟化技术 - 基于云的托管服务提供商通常使用虚拟化技术来高效地将客户跨多台物理机器复用 - 使客户很难保证虚拟化应用程序实例在同一物理机架上运行。没有这种物理位置，应用程序将面临传统数据中心拓扑中的机架间网络瓶颈[2]。

不是应用本身的问题。数据中心使用的路由和转发协议是为特定的部署设置而设计的。传统上，在普通的企业/内联网环境中，通信模式相对可预测，并且数量不多的流行通信目标。主机之间通常只有少数路径，次路径主要用于容错。相比之下，最近的数据中心设计依赖路径多重性来实现主机的水平缩放[3,16,17,19,18]。由于这些原因，数据中心拓扑与典型的企业网络大不相同。

一些数据中心应用程序通常会启动多种主机之间的连接，并需要大量的总带宽。由于最高端商用交换机的端口数量有限，数据中心拓扑结构通常采用多根链路的形式，链路速度更高，但总体带宽越来越低[2]。这些多根树在所有主机对之间有许多路径。一个关键的挑战是同时并且沿着这些路径动态地转发流，以最小化/减少链路超额预订并且提供可接受的总带宽。

不幸的是，现有的网络转发协议经过优化，在没有故障的情况下，为每个源/目的地址对选择一条路径。 这样的静态单路径转发可以显着地利用具有任何扇出的多根树。 在企业和数据中心环境中最先进的转发使用ECMP [21]（等价多路径）使用流散列法在可用路径上静态分散流。 流到路径的这种静态映射不考虑当前的网络利用率或流量大小，导致交换缓冲器的冲突，以及整体交换机利用率的下降。

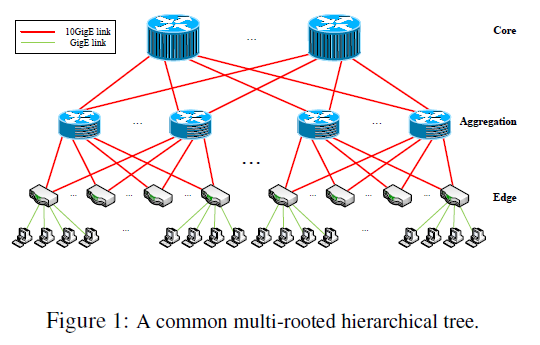
本文介绍了Hedera，这是一种用于数据中心中多级交换拓扑的动态流量调度系统。 Hedera收集组成交换机的流量信息，计算流量的非冲突路径，并指示交换机相应地重新路由流量。我们的目标是最大限度地提高聚合网络利用率 - 分割带宽，并以最小的调度程序开销或对活动流量的影响来实现。通过全面了解路由和流量需求，我们使调度系统达到本地调度器无法企及的瓶颈。

我们已经在PortLand测试平台上完成了Hedera的全面实施[29]。对于我们的实现和大规模的模拟，我们的算法都能提供最佳的几个百分点的性能 - 一个假想的无阻塞交换机 - 用于许多有趣和现实的通信模式，并且在我们的测试中提供比状态多4倍的带宽的艺术ECMP技术。 Hedera通过适度的控制和计算开销来提供这些带宽改进。

我们的放置算法的一个要求是在理想条件下个别流量需求的准确视图。 不幸的是，由于终端主机或网络中其他地方的限制，测量当前TCP流量带宽可能与流量可以通过适当调度实现的带宽无关。 因此，我们提出了一种有效的算法来估计每个流量在最大公平资源分配下实现的理想带宽份额，并描述了该算法在设计调度技术中的帮助。

2背景

最近MapReduce [8]，Hadoop [1]和Dryad [22]等强大的分布式计算框架以及搜索，电子商务和社交网络等Web服务的发展导致了由大宗商品 级个人电脑。 同时，数以万计的机器上存储的数据集的大小和复杂性也达到了数PB，数量空前增长[14]。



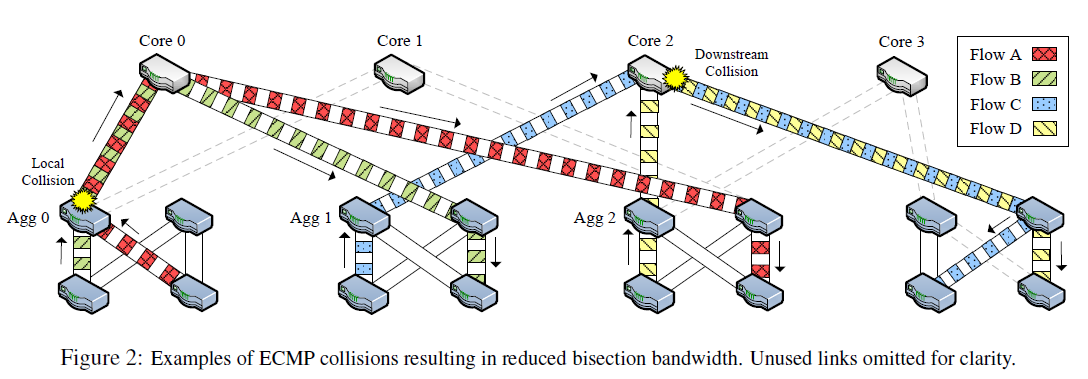
这些集群应用的瓶颈通常是网络，而不是本地资源[4,7,9,14,16]。 因此，提高应用性能可能取决于提高网络性能。 大多数传统的数据中心网络拓扑结构都是层次结构树，连接到端主机的小型廉价边缘交换机[2]。 这种网络通过两层或三层交换机相互连接，以克服商用交换机端口密度的限制。 随着数以万计的机器建立更大的数据中心，最近的研究主张数据中心网络的横向扩展而不是纵向扩展[3,16,17]。 而不是使用具有更高速度和端口密度的昂贵的核心路由器，网络将在任何给定的源和目的地边缘交换机（所谓的多根树形拓扑（例如图1））之间利用更多数量的并行路径。

因此，我们发现自己处于一个僵局 - 使用多根拓扑的网络设计，有可能在所有通信主机之间提供完全的二等分带宽，但是没有一个有效的协议在网络内转发数据，或者调度器将流量合理地分配给路径利用这种高度的并行性。 为了解决这些问题，我们介绍了Hedera的架构，这是一个利用数据中心拓扑中路径分集的系统，可以为一系列流量模式提供近似的二等分带宽

2.1数据中心流量模式

目前，由于隐私和安全问题没有公开可用的数据中心流量跟踪，因此我们在已发布的工作中沿着流量分布的方向生成模式，以模拟典型的数据中心工作负载以评估我们的技术。我们还创建了可能强调数据中心网络的综合通信模式。最近的数据中心交通研究[4,16,24]显示，在空间和时间上的交流矩阵有巨大的变化;一个典型的服务器展现了许多小的事务型RPC流（例如搜索结果）以及少量的大型传输（例如备份，诸如MapReduce作业的后端操作）。我们相信，网络结构应该对各种通信模式具有鲁棒性，并且不应该强迫应用程序开发人员将其通信模式与在特定网络设置中可以实现良好性能的通信模式相匹配，以便最小化开发和调试时间，轻松地从一个网络环境移植到另一个。

因此，本文将重点放在生成压力和网络饱和的流量模式，并将Hedera的性能与当前基于散列的多路径转发方案进行比较。



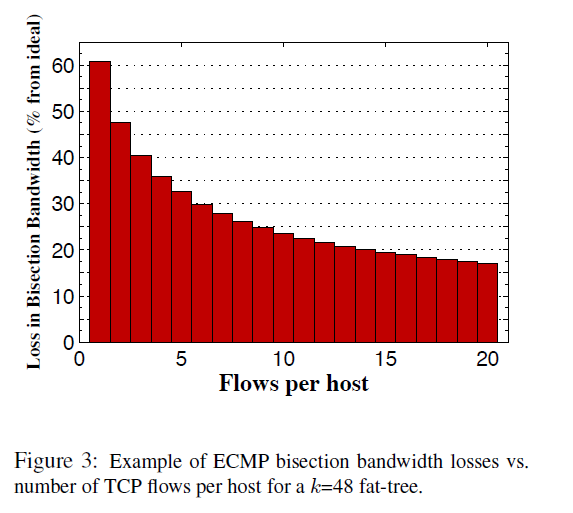
2.2当前数据中心多路径

为了充分利用数据中心拓扑中的多条路径，目前的技术状态是使用等价多路径转发（ECMP）[2]。 启用ECMP的交换机为给定的子网配置了多个可能的转发路径。 当具有多个候选路径的分组到达时，它被转发到对应于该分组头部的选定字段的哈希模数的路径[21]，将负载分裂到多个路径上的每个子网。 这样，一个流的数据包都采取相同的路径，并保持它们的到达顺序（当发生数据包重新排序时TCP的性能显着降低，因为它将这解释为由于网络拥塞造成的数据包丢失的征兆）。

一个紧密相关的方法是Valiant负载平衡（VLB）[16,17,34]，它实质上保证了在网状网络中通过将来自网状网中的源交换机的单个数据包从随机选择的中间“核心” 交换机，最终将这些数据包转发到目标交换机。 最近VLB [16]的实现在每个流上执行随机转发，而不是基于每个数据包，以保持数据包排序。 请注意，每个流量VLB实际上等同于ECMP

ECMP的一个关键限制是两个或更多的大流可能在它们的哈希路径上出现冲突，并最终在相同交换机的输出端口出引发碰撞，产生这种本来可以避免的瓶颈，如图2所示。这里，我们考虑一个示例 多进制1 Gbps网络拓扑中的主机子集。 我们确定由哈希引起的两种类型的碰撞。 首先，由于散列冲突，TCP流A和B在交换机Agg0本地干扰，并受到出口链路的1Gbps容量限制到Core0。 其次，对于下行干扰，Agg1和Agg2独立转发数据包，无法预见Core2对于流量C和D的冲突。

在这个例子中，所有四个TCP流量都可以达到1Gbps的容量，转发得到改进; 流A可能已经被转发到Core1，流D可能已经被转发到Core3。 但是由于这些冲突，所有四个流量都以每个500Mbps的速率瓶颈，50％的二等分带宽损失。



请注意，ECMP和基于流的VLB的性能本质上取决于流量大小和每个主机的流量数量。 基于散列的转发在网络中的主机同时进行全互通的情况下，或者只持续几个RTT的单个流的情况下表现良好。 非均匀的通信模式，特别是涉及大块数据传输的模式，需要更仔细的流量调度以避免网络瓶颈。

我们将这些权衡的全面评估推迟到第6部分，然而我们可以通过简单的蒙特卡洛模拟捕捉散列性能降低的直觉。考虑由1GigE 48端口交换机组成的3阶段胖树，其中27k个主机执行数据混洗。流量被散列到路径上，每个链路被封装在1GigE上。如果每个主机一次只向一个远程主机传输相同数量的数据，则散列冲突将使网络的平分带宽平均减少60.8％（图3）。但是，如果每台主机并行地通过1000个并发流向远程主机进行通信，则散列冲突只会将总分叉带宽减少2.5％。这里的直觉是，如果每个主机同时有很多流量，它们的单个速率将会很小，并且冲突不会很明显：​​每个链路有1000个时隙要填充，只有大于1000个数据流散列到同样的链接。总体而言，Hedera补充了ECMP，补充了导致ECMP问题的通信模式的默认ECMP行为

2.3动态流量需求估计

图2说明了任何动态网络调度机制的另一个重要要求。找到一个良好的全网调度的直接方法是测量网络中所有链路的利用率，并将流量从利用率高的链路转移到利用率较低的链路。关键的问题是流量的调度。同样，直接的方法是测量受限链路上每个流所需要的带宽，并将流移动到具有足够带宽容纳这条流的备用路径。不幸的是，一个流量的当前带宽可能不能反映实际需求。我们定义一个TCP流量的自然需求，意味着它将在一个完全无阻塞的网络中增长的速率，也就是没有考虑到他会和别的流冲突，从而阻塞的情况。例如，在图2中，所有流量都以500Mbps的速度通信，尽管所有的流量都可以以1Gbps的速度进行通信，并且转发性能更好。在4.2节中，我们展示了如何有效地估计流的自然需求，以更好地告知Hedera的布局算法。

3架构

在高层描述，Hedera有三个基本步骤的控制循环。 首先，它检测边缘交换机处的大流量。 接下来，它估计大流量的自然需求，并使用布局算法来计算它们的良好路径。 最后，这些路径安装在交换机上。 我们设计了Hedera以支持任何通用的多根树形拓扑结构，如图1所示，在第5节中我们使用胖树拓扑展示了我们的物理实现。

3.1交换机初始化

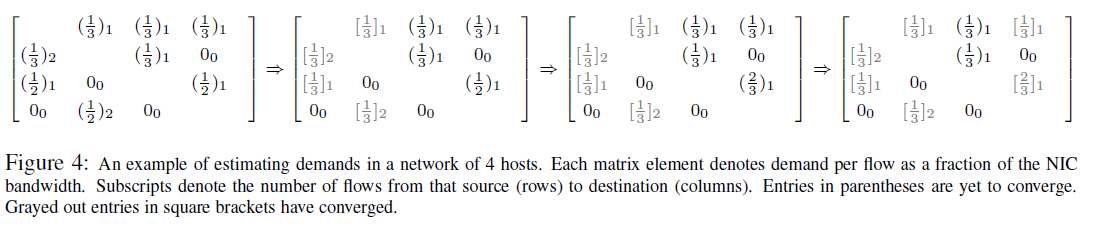
为了利用多根树中的路径多样性，我们必须在所有核心交换机之间尽可能均匀地传出去往或来自任何主机的外出流量。因此，在我们的系统中，上行路径时是非确定性的，但是下行是确定性的。具体而言，对于多根拓扑，当给定核心交换机之后，从源到目的主机的路径就被唯一确定下来了。

为了在下行路径上执行这个确定性，我们使用前缀为目标pod的IP地址范围初始化核心交换机。一个pod是从核心交换机开始的任何子组（在我们的胖树测试平台中，它是一个完整的聚合和边缘交换的二分图，见图8）。类似地，我们初始化汇聚交换机，该汇总交换机具有该接口中边缘交换机的下行端口的前缀。最后，边缘交换机将数据包直接转发给其连接的主机。

当一个新的流程开始时，默认的交换机行为是基于流量的10元组的一个散列，沿着它的一个等价路径（类似于ECMP）转发它。 这条路径被使用，直到流量增长超过阈值，此时Hedera会动态计算一个适当的位置。 因此，假定所有流量都很小，直到超过阈值，在我们的实现中达到100 Mbps（每个主机1GigE链路的10％）。 流是包含<src MAC，dst MAC，src IP，dst IP，EtherType，IP协议，TCP src端口，dst端口，VLAN标记，输入端口>的相同10元组的数据包流。

3.2调度程序设计

中央调度器（可能被复制用于故障转移和可伸缩性）基于对当前网络通信需求的定期更新来动态地操纵边缘和汇聚交换机的转发表。



调度程序旨在将流分配给非冲突路径;更具体地说，它试图不将多个流放置在不能满足其自然带宽需求的链路上。在这个模型中，每当一个流持续一段时间，其带宽需求增长超过一个定义的限制时，我们使用第4节中描述的调度算法之一给它分配一个路径。根据这个选择的路径，调度器将流入口插入该流的源端的边缘和汇聚交换机;这些条目在其新选择的路径上重定向流。一旦流终止， 流条目在超时后过期。请注意，由调度程序维护的状态只是软状态，不需要与任何副本同步以处理故障。调度程序状态不需要正确性（连接性）;相反，它有助于性能优化。

当然，具体调度算法的选择是开放的。 在本文中，我们比较两个算法，全球首次适应和模拟退火，ECMP。 两种算法搜索流到核映射，目的是增加当前通信模式的聚合二等分带宽，补充大流量的默认ECMP转发。

4估计和调度

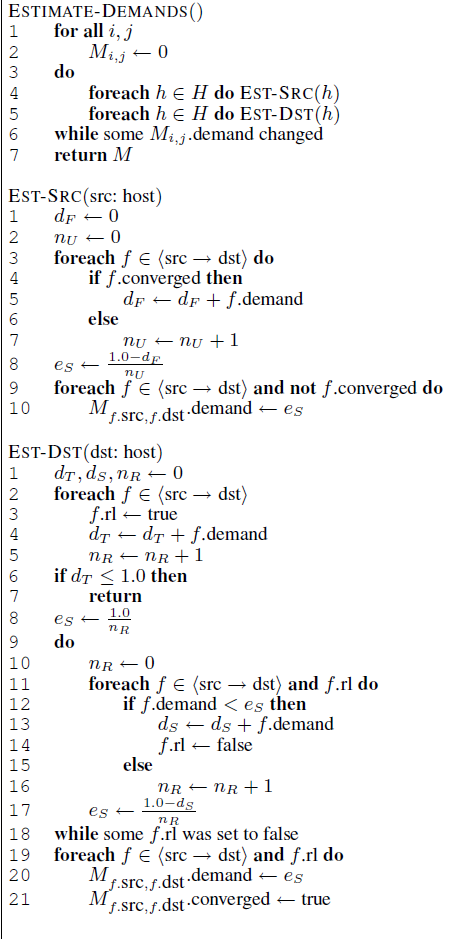
在不超过任何链路的容量的情况下，在一般网络中查找流量路由称为MULTICOMMODITY FLOW问题，对于整数流程是NP完全的[11]。 而对于三级克洛斯网络，同时流路由可以在多项式时间内解决，对于五级克洛斯网络（即三级胖树）[20]，没有已知的多项式时间算法。 由于我们的目的不是为了优化Hedera的特定拓扑结构，因此本文提出了可应用于一系列实际数据中心拓扑的实用启发式方法。

4.1主机与网络限制流量

流可以分为两类：网络限制的（例如，来自RAM的数据传输）和主机限制的（例如受到主机磁盘访问，处理等的限制）。 受网络限制的流将沿着其分配的路径使用可用的所有带宽。 这种流量受到网络拥塞的限制，而不是主机NIC。 主机限制流量在理论上可以实现由源主机和目标主机的“较慢”限制的最大吞吐量。 在非最优调度的情况下，网络限制流可能实现的带宽小于底层拓扑可用的最大可能带宽。 在本文中，我们关注网络流量限制，因为限制主机流量是机器内部瓶颈的一个症状，超出了本文的范围。

4.2需求估计

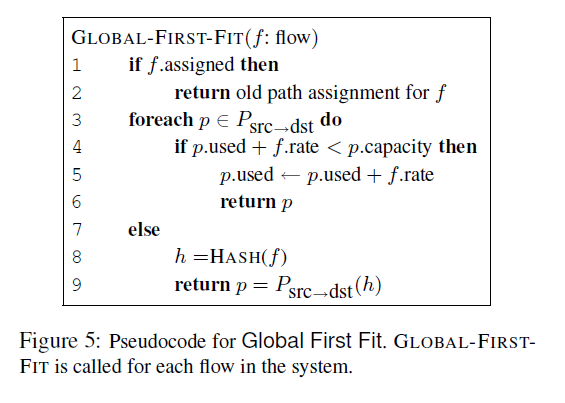
在理想的无阻塞网络中，TCP流量的当前发送速率很少说明其自然带宽需求（2.3节）。 因此，为了做出智能流量放置决策，我们需要知道流量的最大公平带宽分配，就好像它们仅受到发送器或接收器NIC的限制。 当网络受限时，发送方将尝试在其所有流量之间公平分配其可用带宽。 TCP的AIMD行为与网络中的公平排队相结合，试图达到最大最小公平。 注意，当从主机A到另一个主机B有多个流时，每个流将具有相同的稳态需求。 我们现在描述如何在假设的平衡状态下查找TCP需求。



需求估计器的输入是所有活动大流量的源和目的地对的集合F. 估计器保持N×N矩阵M; N是主机的数量。 第i行第j列的元素包含3个值：（1）从主机i到主机j的流量，（2）从主机i到主机j的每个流量的估计需求，以及（3） 标志着需求汇聚的流量的“汇聚”旗帜。 需求估计器重复执行以下步骤：增加源的流量，减小接收器的容量，直到流量收敛; 图7显示了伪代码。 请注意，在每次减少接收器流量的迭代中，一个或多个流量会收敛，直到最终所有流量汇聚到自然需求。 估计时间复杂度为O（| F |）。

图4用一个简单的例子说明了估计流量需求的过程。考虑通过非阻塞拓扑连接的4个主机（H0，H1，H2和H3）。假设

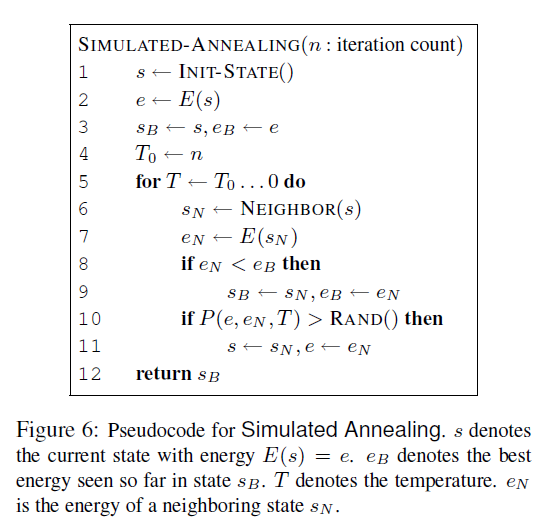
H0分别向H1，H2和H3发送1个流; H1向H0发送2个流量，H1向H2发送1个流量; H2分别向H0和H3发送1个流; H3发送2个流量给H1。该图显示了需求估计器的迭代。矩阵表示在算法的连续阶段期间的流量需求，从发送者的流量增加开始，接着在接收器处的流量减少等等。最后一个矩阵表示流量的最终估计自然需求。对于实际的通信模式，当前活动流的需求矩阵是一个稀疏矩阵，因为大多数主机一次将与一小部分远程主机进行通信。需求估计器在很大程度上也是可并行化的，有利于可扩展性。实际上，我们的实现使用并行和稀疏矩阵数据结构来提高算法的性能和内存占用。



4.3全局首次适应

在多根树型拓扑中，任何一对源主机和目标主机之间都有几个可能的等成本路径。当检测到新的大流量（例如主机链路容量的10％）时，调度器线性地搜索所有可能的路径以找到其链路组件可以适应该流量的路径。如果找到了这样的路径，那么该流被“放置”在该路径上：首先，在对应于该路径的链路上对该流进行容量预留。其次，调度器在相应的边缘和汇聚交换机中创建转发条目。为此，调度程序维护网络中每条链路的保留容量，并使用该容量确定哪些路径可用来承载新的流量。流量到期时，保留被清除。

请注意，这对应于第一个拟合算法;一个流被贪婪地分配了可以容纳它的第一条路径。当网络负载较轻时，在许多可能的路径中找到这样的路径可能是容易的;然而，随着网络负载的增加和链路饱和，这种选择变得更加困难。全局优先并不能保证所有的流都能被满足，但是这个算法在实践中表现得相当好，如第6节所示。我们在图5中给出了全局优先的伪代码。



4.4模拟退火

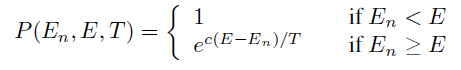
接下来我们描述模拟退火调度器，它执行概率搜索来有效地计算流的路径。 我们的方法的关键思想是为每个目的主机分配单个核心交换机，而不是为每个流进行分配，这大大减少了搜索空间。 模拟退火通过给主机A指定核心交换机来转发到主机A的所有流。

算法的输入是所有将要被放置的大流的集合，以及由需求估计器估计出来的流量需求。 模拟退火搜索状态解空间来找到近乎最优的解决方法（图6）。 函数E定义当前状态下的能量。 在每次迭代中，根据当前和邻近状态的能量E以及当前温度T计算出概率P，然后以一定的概率P移动到邻近状态。随着模拟退火算法的每次迭代，温度都会降低，当温度为零时，迭代就会停止。允许解决方案转移到更高的能量状态以避免局部最优。

1. 状态：从目标主机到核心交换机的一组映射。 Pod中的每台主机都分配有一个特定的用于接收流量的核心交换机。

2. 能量函数E：当前状态下，超出总链路容量的流量总和。 每个状态会为每个流量分配一条特定的路径。 我们使用这些信息来查找超过容量的链路，对于这些流量需求超过链路容量的链路，我们计算出超出需求的总和。

3. 温度T：终止前的剩余迭代次数。

4. 从状态s到邻居状态sn的能量E和En的接受概率P. 其中c是可以改变的参数。 我们根据实验确定c = 0.5×T0，得到了16个主机集群的最佳结果，c = 1000×T0对于大型的数据中心来说是最好的。

5. 邻居生成器功能NEIGHBOR（）：交换当前状态的任何一个pod中的两个主机对应的核心交换机。（表示并非同一个pod中分配的主机就是一样的）

虽然模拟退火是一种众所周知的技术，但是我们的贡献在于进行了优化，对于显著减少搜索空间，选择合适的能量函数以及邻居选择功能以确保快速收敛到接近最优的调度。一个简单的方法是分别为每个流分配一个核心交换机并执行模拟退火。然而，这导致巨大的搜索空间限制了模拟退火的有效性。搜索空间的直径（在任何两个状态之间的最大邻居跳数）采用这种方法等于系统中的流量总数。我们将核心交换机分配给目标主机的技术将搜索空间的直径减小到数据中心中的流量总数和主机数量的最小值。这种启发式算法显着缩小了搜索空间：在一个拥有27K台主机的大型数据中心，等价于拥有27k个大型流量下，搜索空间大小减少了1012000倍。当搜索空间的大小和直径减小时，模拟退火效果更好[12] 。使用简单的方法，算法的运行时间与流的数量和迭代次数成正比，而我们的技术的运行时间仅取决于迭代次数。

我们实现了模拟退火的基础和优化版本。 我们的仿真显示，对于具有16k流量的8,192个主机的数据中心中的随机通信模式，与基线相比，我们的技术可使分叉带宽提高20％，计算时间比基础版本缩短10倍。 这些收益随着数据中心的规模以及流量的增加而增加。

初始状态：每个pod都有一些来自核心交换机的固定的下行链路容量，这仅对于发往该pod的流量有用。 所以这里一个重要的想法就是我们应该把多个不同的核心交换机分布在同一个pod中的多个不同的主机。 对于胖树来说，在同一个pod中的主机数量等于核心交换机的数量，所以建议一对一的映射。 我们将状态解空间限制在这样的分配下。即，我们不将核心交换机分配给单个流，而是分配给单个目标主机。 请注意，初始状态的选择仅在模拟退火调度程序第一次运行时使用。 我们使用一个优化方法来处理系统的动态变化，这样随着时间的推移会降低初始状态的重要性。

邻居生成器：精心设计的邻居生成器函数可以从本质上避免进入局部最优。我们的实现了三种不同的邻居生成器函数：

（1）随机选择一个pod，然后随机选择这个pod里面的两台主机，然后交换着两台主机对应的核心交换机。

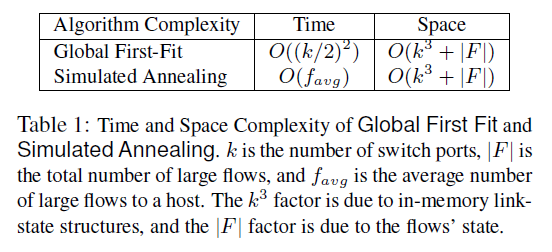
（2）随机选择一个边缘层交换机，随机选择的两台主机，交换这两台主机对应的核心交换机。

（3）以相等的概率随机选择一个边缘交换机或汇聚层交换机，选中了边缘交换机或汇聚层交换机之后，随机选定该核心交换机或汇聚层交换机下的两台主机，交换这两台主机对应的核心交换机。我们的邻居生成器函数在每次迭代运行时，以相同的概率在所描述的3种技术之间进行选择。使用多个邻居生成器函数可以帮助我们避免在单个邻居生成器函数时搜索解空间中的可能陷入的局部最优。

能量函数的计算：邻居的能量函数可以根据当前状态的能量和交换了核心交换机之后的能量的增量来计算。 我们不需要重新计算所有链接的超出容量。 交换一对主机对应的核心交换机只会影响那些发往这两个主机的流。所以我们只需要针对那些在当前状态的能量下被更新了能量值的特定的链路重新计算能量函数的差值。 因此，计算能量的时间仅取决于流向两个受影响主机的大流的数量。

动态变化的流量：随着流动模式的动态变化，在每个调度阶段，少数流量将被重新分类为大流量，而一些较旧的流量将会完成其传输。 我们已经实现了一个优化，我们将初始状态设置为前一个调度阶段的最佳状态。 这使得现有的持续流的路由布局尽可能少地被破坏，如果它们的当前路径仍然能够支持它们的带宽需求的话。 此外，由于该优化，模拟退火调度器首次启动时使用的初始状态随着时间的推移变得不太重要。

搜索空间：模拟退火的关键特点是根据目标主机分配唯一的核心交换机，这对缩小搜索空间的大小至关重要。 但是，有一些通信模式，最佳解决方案必须要求单个目标主机通过多个核心交换机接收传入流量。 尽管我们省略了细节，但是我们发现至少对于胖树拓扑来说，在下列情况下可以处理所有通信模式：i）往返于主机的大流量的最大数目最多为k / 2，其中k 是网络交换机中端口的数量，或者ii）将每个大流量的最小阈值设置为链路容量的2 / k。 鉴于在实践中数据中心可能是由相对较高基数的交换机构成的，例如k？ 如图32所示，我们的搜索空间优化不太可能消除在实践中定位最佳流量分配的可能性。



4.5全局算法的比较

采用全局首次优先适应，大流量可以在检测到时立即重新路由，并基本上固定在其保留链路上。 尽管模拟退火等待下一个调度，使用先前计算的流程安排来优化当前的放置，并且由于其概率搜索，平均提供更好的网络利用率。

我们选择全局优先和模拟退火算法，因为它们简单;我们认为更复杂的算法会阻碍调度器的可扩展性和效率，同时仅获得增量带宽回报。我们相信，它们在计算复杂性上达到了正确的平衡并实现了性能提升。表1给出了这两种算法的时间和空间复杂性。注意，全局优先适应的时间复杂度与| F |无关，网络中大流量的数量是独立的，并且模拟退火的时间复杂度与k无关。

更重要的是，我们算法的简单性使得它们都非常适合硬件实现，比如在FPGA中，因为它们主要由简单的算术组成。这样的实现将显着降低跨越独立调度器机器的网络堆栈的通信开销。

总的来说，虽然模拟退火在概念上涉及更多，但我们在第二部分展示。 6几乎总是胜过Global First Fit，为我们的测试平台和更大的仿真提供了接近最佳二等分带宽。 我们相信，模拟退火的额外概念复杂性是由现代数据中心的网络基础设施的带宽收益和巨大的投资来证明的。

4.6容错

任何调度程序必须在执行流分配时考虑交换机和链路故障。 虽然我们省略了细节，为了简洁起见，我们的Hedera实现增加了PortLand路由和容错协议[29]。 因此，Hedera调度程序知道使用标准PortLand机制的故障，并可以重新路由映射到故障组件的流。

5实施

为了在真实的物理多根网络上测试我们的调度技术，我们构建了在之前的工作中抽象描述的fat-tree网络的例子[3]。 此外，为了理解我们的算法如何随着网络规模的扩大而变化，我们实现了一个模拟器，在一个调度算法的控制下，模拟具有多个流的大型网络的行为。

5.1拓扑

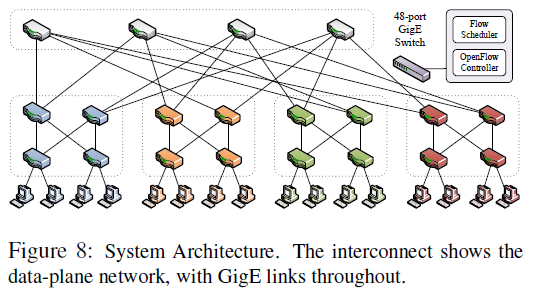
对于本文的其余部分，我们采用以下术语：对于由k端口交换机构建的胖树网络，存在k个pod，每个pod包括两个层：较低的pod交换机（边缘交换机）和较高的pod交换机 （汇聚交换机）。 每个边缘交换机管理（k / 2）个主机。 k个pod通过（k / 2）的平方个核心交换机相互连接。

这种拓扑的主要优点之一是可用路径的高度多样性; 在任何给定的源和目标主机对之间，都有（k / 2）的平方个开销相同的路径，每个对应一个核心交换机。但请注意，这些路径不是链接不相交的。为了利用这个路径分集（为了最大化可实现的二等分带宽），我们必须分配流量非冲突路径。我们的工作的一个关键要求是执行此类调度，而不对端主机网络堆栈或操作系统进行修改。我们的测试平台由16个主机组成，使用20个4端口交换机的胖树互连，如图8所示。

我们部署了一个将所有交换机连接到一个48端口非阻塞GigE交换机的并行控制平面。我们强调这个控制网络不是Hedera架构所必需的，而是在我们的测试平台中用作调试和比较工具。该网络仅向交换机传输交通监控和管理信息;但是，这些消息也可以使用数据平面进行传输。当然，对于数千个主机的大型网络，控制网络可以被组织为传统的树形结构，因为控制流量应该只是数据流量的一小部分。在我们的部署中，流量调度程序在连接到48端口交换机的单独机器上运行。

5.2硬件描述

测试平台中的交换机是1U双核3.2 GHz Intel Xeon机器，具有3GB RAM和NetFPGA 4端口GigE PCI卡交换机[26]。 16台主机是1U四核2.13 GHz Intel Xeon机器，配备3GB内存。 这些主机有两个GigE端口，第一个连接到控制网络进行测试和调试，另一个连接到NetFPGA边缘交换机。 控制网络被组织为简单的星形拓扑结构。 中央交换机是Quanta LB4G 48端口GigE交换机。 调度程序机器具有双核2.4 GHz Intel Pentium CPU和2GB RAM。



5.3 OpenFlow控制

树中的交换机都运行OpenFlow [27]，允许访问所有交换机的转发表。 OpenFlow实施已经被移植到各种商用交换机上，包括瞻博网络，惠普和思科的交换机。 OpenFlow交换机将传入的数据包与流条目进行匹配，流条目指定特定的操作，如复制，在特定端口上转发，丢弃和广播。 NetFPGA OpenFlow交换机有两个硬件表：32个条目的TCAM（接受可变长度前缀）和一个32K条目SRAM，只接受具有完全合格10元组的流条目。

当OpenFlow交换机启动时，它们试图打开一个安全通道到中央控制器。控制器可以查询，插入，修改流条目或执行许多其他操作。交换机维护每个流量和每个端口的统计信息，例如总字节数和流量持续时间。交换机的缺省行为如下：如果传入的数据包与TCAM或SRAM表中的任何流条目不匹配，则交换机将插入一个具有适当输出端口（基于ECMP）的新流表项，允许任何后续数据包以硬件线速直接转发。一旦流量增长超过指定的阈值，Hedera调度器可以修改该流的流入口，以沿着新选择的路径重定向它。

5.4调度频率

我们的调度程序实现轮询边缘交换机进行流量统计（以检测大流量），并每5秒执行一次需求估计和调度。 这个时期完全是由于OpenFlow NetFPGA实现的寄存器读取限制。 然而，我们在第6节中的可扩展性测量表明，适度配置的机器可以在几毫秒内调度数万个流量，即使在5秒的轮询速率下，Hedera也显着优于当前ECMP方法的二等分带宽。 一般来说，我们认为应该使用简单的技术来实现亚秒和潜在的低于100ms的调度间隔

5.5模拟器

由于我们的物理测试台仅限于16台主机，我们还开发了一个模拟器，对TCP流的网络行为进行粗略模拟。 仿真器负责流量到达和离开，以显示具有动态通信模式的大型网络的系统可扩展性。 我们使用多达8,192个主机的网络流量模拟器来检查我们的不同调度算法。 现有的分组级模拟器，如ns-2，不适用于此目的。 每个以1Gbps发送的8192个主机的仿真将需要处理2.5×1011个数据包，用于60秒的运行。 如果使用每个数据包模拟器来模拟每秒使用TCP传输一百万个数据包，则需要71个小时才能模拟一个测试用例。

我们的模拟器将数据中心拓扑建模为具有有向边的网络图。 每条边都有一个固定的容量。 模拟器接受主机之间的通信模式，并使用它，以及平均流量规格和到达率，来生成模拟的流量。 模拟器产生具有指数分布长度的新流，开始时间基于具有给定平均值的泊松到达过程。 目的地在第6部分的基础上。

仿真以离散时间滴答进行。 在每个时间点，模拟会更新网络中所有流量的速率，并在需要时生成新的流量。它还定期调用调度程序将（新）路线分配给流。 当调用Simulated Annealing和Global First Fit调度程序时，模拟器首先调用需求估计器并传递其结果。

在更新流量时，模拟器建模TCP慢启动和AIMD，但不执行每个数据包的计算。 每一个滴答，模拟器洗牌流量的顺序，并计算每个流量的预期增长率，受流量路径上可用带宽的约束。 如果一个流量开始慢，它的速度翻倍。 如果处于拥塞避免状态，则其速率会相加增加（使用15 MB / s的附加增加因子来模拟RTT为100μs的网络）。 如果流量路径饱和，则流量速率减半，带宽沿路径释放。 每个滴答，我们还计算流发送的字节数和已完成发送所有字节的清除流。

由于我们的模拟器没有对单个数据包进行建模，因此不能捕获不同数据包大小的性能差异。这个决定的另一个后果是我们的模拟不能捕获流动间的动态或缓冲行为。因此，TCP Reno / New Reno可能会比我们的模拟器预测的要差一些。另外，我们将TCP流模型化为单向，尽管实际的TCP流涉及到反向的ACK。然而，对于1500字节的以太网帧和延迟的ACK，由ACK引起的带宽约为2％。我们认为这些权衡是研究本文所述规模网络所必需的。

我们运行每次模拟60秒，测量中间40秒的平均二等分带宽。由于模拟器不捕获流动间和流量突发，所以我们的结果是乐观的（模拟器带宽超过测试台测量值），因为所产生的散列冲突有时会导致整个数据窗口丢失，造成超时在测试台上（见第6节）。对于控制网络，我们观察到模拟器的性能与测试平台的性能更接近。同样，对于尽量优化最小争用的全局优先和模拟退火，我们观察到模拟器和测试平台的性能非常好。在所有的结果中，当流量之间存在竞争时，仿真器表现出比测试台更好的性能。

6评估

本节介绍使用我们的测试平台和模拟器对Hedera进行的评估。 这些测试的目标是确定实现的总分等分带宽

各种交通模式。

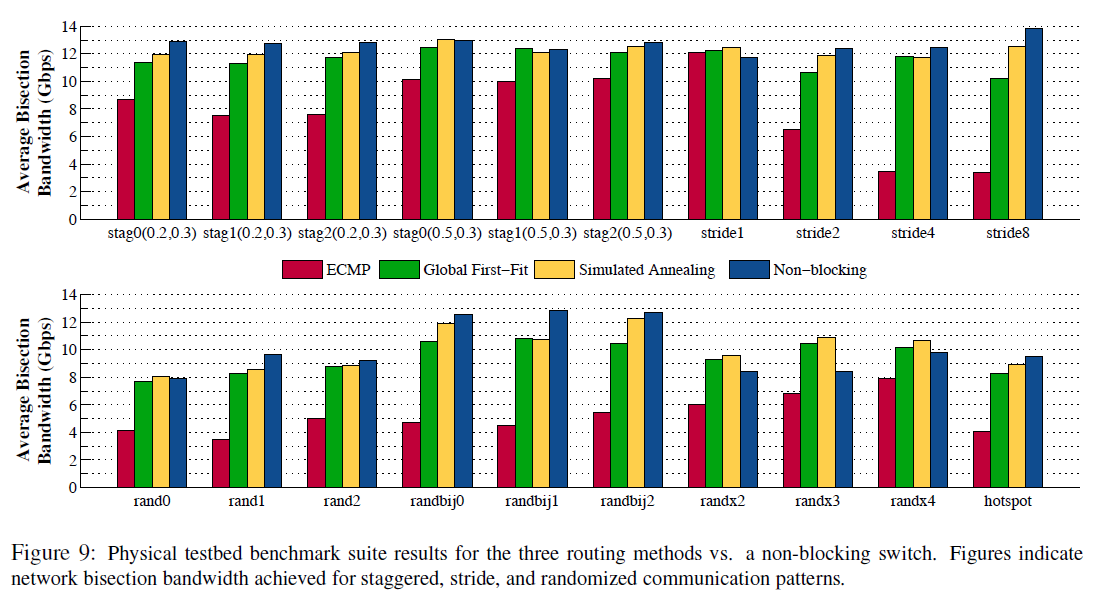
6.1基准通信套件

在没有商业数据中心网络的情况下，对于测试平台和模拟器评估，我们首先根据以下样式创建一组类似于[3]的通信模式：

（1）Stride（i）：索引为x的主机通过索引（x + i）mod（num个主机）发送给主机。

（2）交错问题（EdgeP，PodP）：主机以概率EdgeP发送给同一边缘交换机中的另一个主机，以概率PodP发送给同一个主机，以概率1-EdgeP-PodP发送给网络的其余部分。

（3）随机：主机以一致的概率发送给网络中的任何其他主机。 我们包括双映射和热点地区。



我们考虑这些不同大小的网络的映射：16个主机，1024个主机和8192个主机，对应于k = {4,16,32}。

6.2测试平台的基准测试结果

我们运行基准测试，如下所示：16台主机为传入流量打开套接字接收器并不断测量传入带宽。 接下来，主机根据上述大小和目的地开始流动。 每个实验持续60秒，并使用TCP流量; 我们观察了中间40秒的平均二等分带宽。

我们将胖树网络上调度器的性能与控制网络上相同实验的性能进行比较。 控制网络使用非阻塞的48端口千兆以太网交换机连接所有16台主机，代表一个理想的网络。 另外，我们还包括一个基于静态散列的ECMP方案，其中转发路径由目标主机IP地址的散列值决定。

图9显示了各种随机，交错，跨步和热点通信模式的二等分带宽;我们的实验使用TCP饱和链接。在几乎所有探索的通信模式中，全局优先和模拟退火显着优于静态散列（ECMP），并达到网络的最佳二等分带宽（15.4Gb / s吞吐量）。自然地，这些方案的表现随交流地点的增加而提高，如交错概率数字所示。请注意，对于跨步模式（HPC计算应用程序通用），启发式一致地计算正确的流到核映射以有效利用胖树网络，而静态散列的性能随着步长的增加而迅速恶化。此外，对于某些模式，这些启发式也稍微超过用于我们的控制网络的商用48端口交换机。我们怀疑这是由于Net-FPGA与Quanta交换机的不同缓冲器/算法造成的。

通过使用测试平台上的数据包捕获进一步检查性能，我们发现当流之间存在争用时，整个TCP数据包窗口常常丢失。 因此，TCP连接处于空闲状态，直到重新启动计时器（RTOmin = 200ms）。 与其他方案相比，基于ECMP哈希的流布置经历了5倍的重传超时数。 这解释了模拟器中ECMP的过分乐观的性能，如第5节所述，因为我们的模拟器不模拟重传超时和单个数据包丢失。

6.3数据随机播放

我们还在我们的测试平台上执行了全部到内存的数据混洗。 对于许多MapReduce / Hadoop操作来说，数据洗牌是一项昂贵而必要的操作，在这种操作中，每台主机都将大量数据传输给参与洗牌的其他每台主机。 在这个实验中，每台主机使用TCP（120GB shuffle）顺序传输500MB到每个其他主机。

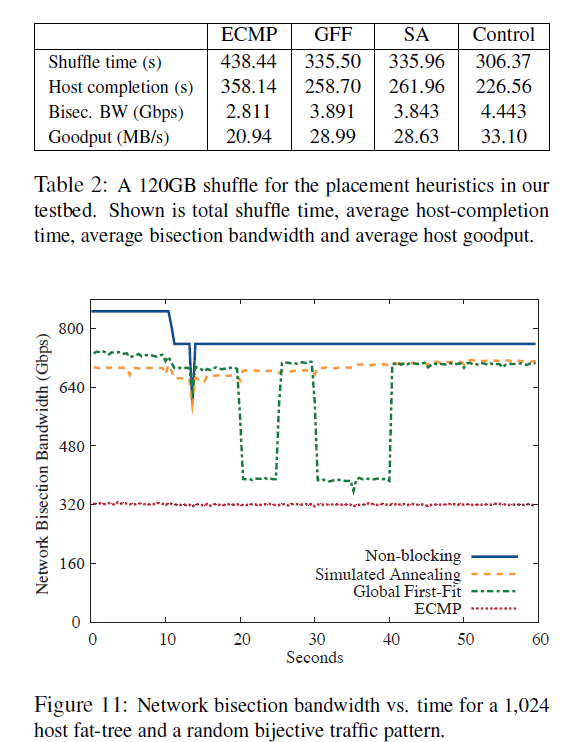
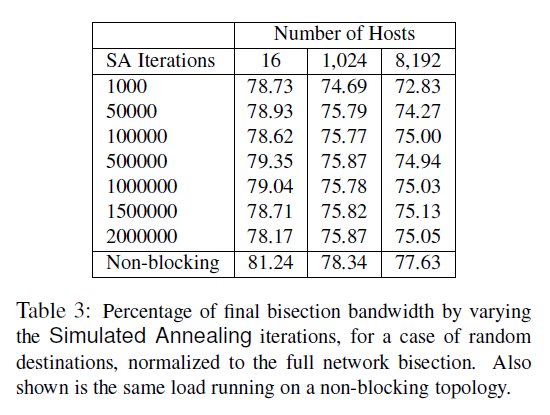


表2中的混洗结果表明，集中式流程调度比基于静态ECMP基于散列的路由执行效果好得多（平分带宽提高了39％）。将这与VL2 [16]中执行的数据混洗进行比较，其中涉及所有主机同时传输到所有其他主机（相对于我们工作中的顺序传输），我们看到静态哈希表现更好，当流量显着大于路径的数量;直观地说，当大量流量上的任何不平衡被平均化时，散列冲突不太可能引起显着的降级。因此，除了Hedera观察/路由计算控制环路的延迟之外，我们还认为，具有许多短小的类似RPC的流量的流量工作负载将从动态调度中获益有限，并且Hedera的默认ECMP转发能够有效地执行负载均衡这个案例。因此，通过对我们的调度程序进行阈值处理，使其只在更大的流量上运行，Hedera对于这两种通信模式都表现出色。



6.4仿真结果

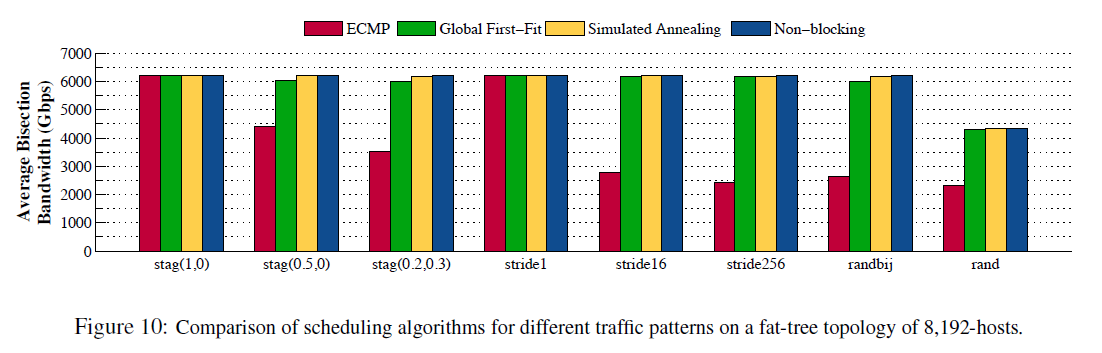
6.4.1通信模式

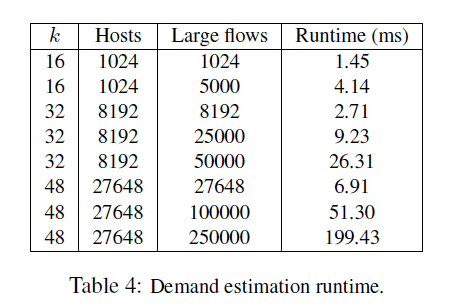
在图10中，我们显示了运行8192个主机（当k = 32时）的模拟胖树网络的基准测试套件时所实现的汇总二等分带宽。我们将我们的算法与整个数据中心的假想非阻塞交换机以及静态ECMP散列进行比较。 ECMP的性能随着本地通信的可能性下降而恶化。这是因为即使对于一个完全公平和完全统一的散列函数，路径分配中的冲突确实发生在相同的交换机中，或者在下游交换机处发生流，浪费了一部分可用带宽。全局调度程序使得按设计选择的离散流程布局可以减少重叠。在大多数这些不同的通信模式中，我们的动态布局算法明显优于静态ECMP哈希。图11显示了1,024主胖树网络的二等分带宽随时间的变化。全球第一次适应和模拟退火执行相当接近最佳的大部分的实验。

6.4.2模拟退火的质量

为了探索模拟退火的参数空间，我们在表3中显示了随机，非生物通信模式在每个调度周期中改变迭代次数的效果。这张表证实了我们关于分配质量和迭代次数的最初直觉，因为大部分的改进都是在前几次迭代中进行的。我们观察到模拟退火的性能在最初的几次迭代之后逐渐逼近模拟退火的最佳结果。

该表还显示随着主机和流量的增加，随机通信模式的最终二等分带宽的百分比。这支持了我们的信念：模拟退火可以在每个调度周期内以相对较少的迭代运行，并且仍然可以实现随时间推移的可比较的性能。这是通过记忆各个时期的核心任务，以及每个时间间隔只有几个新的大流量来帮助的。



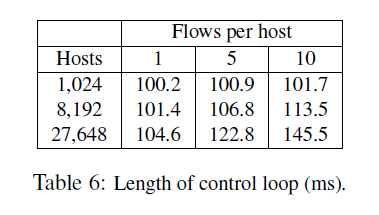


6.4.3需求估计的复杂性

由于需求估计是在每个调度周期执行一次，因此其运行时间必须相当小，以使控制回路的长度尽可能小。 我们研究了不同规模的数据中心不同流量矩阵的需求估计的运行时间。

表4显示了不同输入大小的需求估计器的运行时间。 所报告的运行时间用于需求估计器的运行，在适中的四核2.13GHz机器上使用4个并行线程执行。 即使是拥有27,648个主机和25万个大流量（每个主机近10个大流量的平均值）的大型数据中心，需求估计算法的运行时间也只有200毫秒。 对于更常见的情况，在我们的设置中运行时间约为50-100ms。 我们期望调度器机器是具有更多内核的相当高性能的机器，因此即使在极端情况下仍然保持运行时间在100ms以下。

即使对于具有27k主机的数据中心中的近250,000个大流量的极端情况，使用稀疏矩阵表示的需求估计器的存储器需求也小于20MB。 在更常见的情况下，在数据中心有大量合理数量的大数据流的情况下，整个数据结构将适合现代CPU的L2高速缓存。 考虑到涉及的操作的简单性和数量，FPGA实现可以将稀疏矩阵存储在片外SRAM中。 像Xil-inx Virtex-5这样的FPGA可以实现多达200个并行处理内核来处理这个矩阵。 我们估计，即使在250,000个大流量的情况下，这样的配置也将具有大约5ms的计算等待时间来执行需求估计。



6.4.4模拟退火的复杂性

在表5中，我们显示了不同实验场景的模拟退火的运行时间。模拟退火的运行时间渐进地独立于主机的数量，仅依赖于流量的数量。这里主要介绍的是我们的模拟退火实施的可扩展性及其实际应用的潜力;对于数千个主机的网络和每台主机的合理数量的流量，模拟退火运行时间大约为几十毫秒，即使是10,000次迭代也是如此。

6.4.5控制开销

为了评估集中调度设计的总控制开销，我们分析了调度的总体通信和计算需求。控制回路包括3个组件 - 网络中的所有交换机将大流量的细节发送到调度器，调度器估计流量的需求并计算它们的路由，调度器将新的流量发送到交换机。

我们做了一些假设来分析控制回路的长度。 （1）控制平面由48端口GigE交换机组成，每个交换机的平均延迟为10μs。 （2）交换机和控制器之间的消息格式基于OpenFlow协议（每个流条目72B）[27]。 （3）流量需求估计和调度的总计算时间保守地假定为100ms。 （4）到调度器的最后一跳链路被假定为10GigE链路。这个更高速的最后一跳链路允许大量的交换机同时与调度器通信。我们假定可以充分利用到控制器的10 GigE链路来传输调度更新。

表6显示了每个主机不同数量的大流量的控制回路的长度。这些值表示控制回路的长度由计算时间决定，估计为100 ms。这些结果显示了大型数据中心集中调度方法的可扩展性。

7相关工作

数据中心网络最近出现了大量新的研究建议; 然而，没有一个满意地解决网络的二等分带宽问题。 VL2 [16]和季风[17]建议使用每流量Valiant负载平衡，这可能会导致带宽损失由于长期的碰撞，如在这项工作中所展示的。 西雅图[25]提出了一个单一的第二层域与一跳交换DHT MAC地址解析，但不解决多路径。 DCell [19]和BCube [18]建议使用递归定义的数据中心网络拓扑结构，这涉及多NIC服务器，并可能导致更深层次的超额订阅链接。 再一次，多路径没有明确的解决。

研究人员还从广域的角度探索了多路径环境下的调度流程。 TeXCP [23]和MATE [10]通过使用显式拥塞通知数据包在广域内的多个路径上执行动态流量工程，这些数据包还需要不可用的交换机支持。他们采用分布式流量工程，而我们则利用紧密耦合的中央调度器来利用数据中心环境。 FLARE [31]提出了在流程粒度（TCP数据包突发）的广域中的多路径转发;然而，目前还不清楚数据中心内的低时延是否符合流突发的时间要求，以防止数据包重新排序，并仍然取得良好的性能Miura等人通过使用标记VLAN和商用PC多路径利用胖树网络[28]。集中式路由器控制强制路由或访问控制策略已经在4D架构[15]之前提出过了，像Tesseract [35]，Ethane [6]和RCP [5]这样的项目，与Hedera的集中式流程调度。

虚拟交换结构和各个Clos网络的工作主要集中在抽象层面，但没有涉及使用现有商品组件构建可操作的多层交换机体系结构。 Turner提出了一种优化的无阻塞虚电路交换机[33]，Smiljanic改进了Turner的负载均衡器，并着重于在广义的Clos分组交换网络中提供的算法保证[32]。 Oki等人 设计改进的单个三级Clos交换机的调度算法[30]，并且Holmburg提供了用于多级Clos网络的同时和增量调度的模型[20]。 Geoffray和Hoefler在多级互连网络中描述了一些提高二等分带宽的策略，特别关注源路由，每包分散方法，这些方法打破了以太网上TCP / IP的排序要求[13]。

8结论

我们工作最重要的发现是，为了有效利用可用的网络资源，具有主动流程全局知识的中央调度器可以显着优于基于散列的ECMP负载平衡。 我们通过将调度决策集中在可能负责通过网络发送的大部分字节的大流量来限制我们方法的开销。 我们发现Hedera的性能增益取决于网络流量的速率和持续时间; 当网络受到许多大容量数据传输的影响时，网络内部和网络直径都会带来更多的好处。

在本文中，我们已经展示了构建多根树木调度系统的工作原型的可行性，并且已经表明，模拟退火几乎总是胜过全局第一匹配，并且能够在很宽的范围内提供接近最佳的二等分带宽 的通信模式都在我们的物理测试平台和由数千个节点组成的模拟数据中心网络中。 鉴于我们的流量布局算法的低计算和延迟开销，与数据中心相关的网络基础设施的巨大投资（数百万美元）以及Hedera部署（例如，一个或两个服务器）的增量成本，我们展示动态 流量调度有可能以适度的额外成本提供大量的带宽收益。

致谢

我们感谢UCSD的Nathan Farrington和DCSwitch团队对本项目的宝贵帮助和支持。 我们还要感谢我们的牧羊人Ant Rowstron和匿名审稿人对本文早期草稿的有益反馈。