可扩展的商品数据中心网络架构

穆罕默德·票价 malfares@cs.ucsd.edu
亚历山大Loukissas aloukiss@cs.ucsd.edu
Amin Vahdat vahdat@cs.ucsd.edu
计算机科学与工程系
加州大学圣地亚哥分校
La Jolla，CA 92093-0404

抽象

今天的数据中心可能包含数万台计算机，具有显着的总带宽要求。 网络架构通常由一系列路由和交换元件组成，逐渐更专业且昂贵的设备向上移动网络层次结构。 不幸的是，即使部署最高端的IP交换机/路由器，由此产生的拓扑结构只能支持网络边缘可用的总带宽的50％，同时仍然产生巨大的成本。 数据中心节点之间的不均匀带宽使应用程序设计复杂化，并限制了系统的整体性能。

在本文中，我们展示了如何利用大部分商用以太网交换机来支持由数万个元素组成的集群的全集合带宽。 类似于商业计算机的群集已经在很大程度上取代了更专业的SMP和MPP，我们认为，适当的架构和互连的商品交换机可以以比当今高端解决方案可用的更低的成本提供更多的性能。 我们的方法不需要修改终端主机网络接口，操作系统或应用程序; 批判性地，它完全向后兼容以太网，IP和TCP。

类别和主题描述符

C.2.1 [**网络体系结构和设计**]：网络拓扑结构;
C.2.2 [**网络协议**]：路由协议

一般条款

设计，性能，管理，可靠性

关键词

数据中心拓扑，等价路由

1. 介绍

日益增长的专业知识集群的商用电脑使许多机构能够以高性价比的方式利用petaflops的计算能力和PB级的存储空间。 数以万计的个人电脑的群集并不是最大的

*SIGCOMM'08，8*月17日至22日，2008年，西雅图，华盛顿，美国。

机构和千节点集群在大学，研究实验室和公司中越来越普遍。 重要的应用类包括科学计算，财务分析，数据分析和仓储，以及大型网络服务。

今天，大规模集群的主要瓶颈往往是节点间通信带宽。 许多应用程序必须与远程节点交换信息以继续进行本地计算。 例如，MapReduce [12]必须执行重要的数据混洗，以便在继续执行缩减阶段之前传输映射阶段的输出。 在基于群集的文件系统[18,28,13,26]上运行的应用程序在继续进行I / O操作之前通常需要远程节点访问。 对网络搜索引擎的查询通常需要与主机反向索引的群集中的每个节点进行并行通信，以返回最相关的结果[7]。 即使在逻辑上不同的集群之间，通常还需要有重要的通信要求，例如，当从负责构建索引的站点更新执行搜索的各个集群的反向索引时。 互联网服务越来越多地采用面向服务的架构[13]，其中单个网页的检索可能需要与数百个在远程节点上运行的单个子服务的协调和通信。 最后，并行科学应用的重要通信要求是众所周知的[27,8]。

构建大规模集群的通信结构有两个高级选择。 一个选项利用专门的硬件和通信协议，如InfiniBand [2]或Myrinet [6]。 虽然这些解决方案可以扩展到具有高带宽的成千上万个节点的集群，但它们不会利用商品部件（因此更昂贵），并且不能与TCP / IP应用程序本地兼容。 第二选择利用商用以太网交换机和路由器来互连集群机。 这种方法支持熟悉的管理基础设施以及未修改的应用程序，操作系统和硬件。 不幸的是，聚合簇带宽随着群集大小而变差，并且达到最高带宽水平会带来群集大小上的非线性成本增加。

出于兼容性和成本原因，大多数集群通信系统遵循第二种方法。 然而，根据通信模式，大群集中的通信带宽可能会因为重要因素而被超额订购。 也就是说，连接到相同物理交换机的两个节点可能能够以全带宽（例如，1Gbps）进行通信，但是可能在层次结构中的多个级别之间的交换机之间移动，可能严重地限制可用带宽。 解决这些瓶颈需要非商品解决方案，例如大型10Gbps交换机和路由器。 此外，沿着互连交换机树的典型单路径路由意味着整个群集带宽受通信层次结构根部可用带宽的限制。
即使我们处于10Gbps技术变得具有成本竞争力的转变点，最大的10Gbps交换机仍然会产生巨大的成本，并且仍然限制最大集群的总体可用带宽。

在这种情况下，本文的目的是设计出满足以下目标数据中心通信架构：•可伸缩的互连带宽：它应该为在数据中心中的任意主机在网络中的任何其他主机进行通信是可能的其本地网络接口的全部带宽。

- 规模经济：正如商品个人计算机成为大规模计算环境的基础，我们希望利用相同的规模经济使廉价的现成以太网交换机成为大型数据中心网络的基础。
- 向后兼容性：整个系统应与运行以太网和IP的主机向后兼容。 也就是说，几乎普遍利用商用以太网和运行IP的现有数据中心应该能够利用新的互连架构，而无需修改。

我们可以看出，通过在胖胖架构中互连商品交换机，我们可以实现由数万个节点组成的集群的完全二等分带宽。 具体来说，我们的一个架构采用了48端口以太网交换机，可以为27,648台主机提供全部带宽。 通过利用严格的商品交换机，我们实现了比现有解决方案更低的成本，同时提供更多带宽。 我们的解决方案不需要对终端主机进行任何更改，完全与TCP / IP兼容，并且只对交换机本身的转发功能进行适度的修改。 我们也希望我们的方法将提供全带宽的大型集群一旦万兆以太网交换机成为边缘商品，鉴于目前没有任何更高速度的以太网的替代品（不惜任何代价）的*唯*一途径。 即使更高速的以太网解决方案可用，它们最初将以极大的成本具有小的端口密度。

2. 背景

1. 当前数据中心网络拓扑

我们进行了一项研究，以确定目前数据中心通信网络的最佳实践。 我们专注于利用以太网和IP的商品设计; 我们在第7节讨论我们的工作与替代技术的关系。

1. 拓扑

当今的典型架构包括交换机或路由器的两层或三层树。 三层设计（见图1）在树中，在中间*汇聚*层，并在树的叶子*边缘*层的根目录的*核心*一级。 双层设计只有核心和边缘层。 通常，两层设计可以支持5K至8K的主机。 由于我们针对大约25,000个主机，我们将注意力限于三层设计。

开关[1] 在树的树叶上有一些GigE端口（48-288）以及一些10GeE上行链路到一个或多个网络元件，在叶片交换机之间聚合和传输数据包。 在较高级别的层次结构中，具有10个GigE端口（通常为32-128）的交换机和用于在边缘之间聚合流量的显着交换容量。

我们假设使用两种类型的交换机，它们表示端口密度和带宽中当前的高端端口。 第一个在树的边缘使用，是一个48端口GigE交换机，具有四个10 GigE上行链路。 对于更高级别的通信层次结构，我们考虑128端口10 GigE交换机。 两种类型的交换机允许所有直接连接的主机以其网络接口的全速相互通信。

2. 超订购

许多数据中心设计引入超额认购，作为降低设计总成本的手段。 我们定义术语*超额预订*是终端主机到特定通信拓扑的总二等分带宽之间的最坏情况下的可实现的总带宽的比值。 1：1的超额预订表明，所有主机可能在其网络接口的全部带宽（例如，用于商业以太网设计的1 Gb / s）上可能与任意其他主机通信。 5：1的超额认购值意味着只有20％的可用主机带宽可用于某些通信模式。 典型设计超过2.5：1（400 Mbps）到8：1（125 Mbps）[1]。 尽管对于1 Gb / s以太网来说，尽管对1 Gb以太网进行了超量订购的数据中心是可能的，但我们在2.1.4节中讨论过，尽管适用于大型数据中心的设计成本通常也是令人望而却步的。 当超越单个交换机时，实现10 Gb / s以太网的完全二等分带宽是不可能的。

3. 多路径路由

在较大群集中的任意主机之间传送全部带宽需要具有多个核心交换机的“多根”树（见图1）。 这又需要一种多路径路由技术，如ECMP [19]。 目前，大多数企业核心交换机支持ECMP。 没有使用ECMP，可以使用1：1超额订阅的单根核心支持的最大的集群将限制在1,280个节点（对应于单个128端口10 GigE交换机的可用带宽）。

要充分利用多条路径，ECMP执行流之间的静态*负载分担*。 这在分配决策中不考虑流量带宽，即使对于简单的通信模式也可能导致超额认购。 此外，目前的ECMP实现将路径的多重性限制为8-16，这通常比为较大数据中心提供高二分带宽所需的分集少。 此外，路由表条目的数量与考虑的路径数量相乘增加，这增加了成本并且还可以增加查询延迟。

4. 成本

为大型集群构建网络互连的成本大大影响设计决策。 如上所述，通常会引入超额认购来降低总成本。 在这里，我们用不同数量的主机和使用当前最佳实践超额认购的各种配置的大概成本。 我们假设边缘的每个48端口GigE交换机的成本为7,000美元，聚合和核心层中的128端口10 GigE交换机的成本为700,000美元。 在这些计算中我们不考虑布线成本。 图2绘出了成本在数百万美元作为终端主机上的x轴的总数的函数。 每条曲线表示目标超额预订比例。 例如，在所有主机之间互连20,000台主机与全带宽的交换硬件达到约3,700万美元。 对应于3：1超额订购的曲线绘制了互连终端主机的成本，其中任意端主机通信的最大可用带宽将被限制为大约330 Mbps。

图1：通用数据中心互连拓扑。 主机到交换机链路是GigE，交换机之间的链路是10 GigE。

图2：不同超额认购率的当前成本估算与主机的最大可能数量。

	分层设计			胖树		
年	**10 GigE**	**主持人**	**成本/ GigE**	**GigE**	**主持人**	**成本/ GigE**
2002年	28端口	4,480	$ 25.3K	28端口	5,488	$ 4.5K
2004年	32端口	7,680	$ 4.4K	48端口	27,648	$ 1.6K
2006年	64端口	10,240	$ 2.1K	48端口	27,648	$ 1.2K
2008年	128端口	20,480	$ 1.8K	48端口	27,648	$ 0.3K

表1：最大可能的簇大小为1的收敛比：1对不同年份。

我们还包括使用我们提出的脂肪树结构进行比较的1：1的超额预订费用。

总的来说，我们发现现有的在大型集群中提供高带宽的技术会带来巨大的成本，而基于胖树的集群互连对于以适中的成本提供可扩展带宽具有重要的前景。 然而，从某种意义上说，图2低估了构建数据中心架构中采用最高端组件的难度和费用。 2008年，10个GigE交换机正在成为商品部件的边缘; 将GigE与10 GigE交换机进行比较时，每个端口的每个端口的价格差异大致为5个差异，并且该差异继续缩小。 为了探索历史趋势，我们在表1中显示了可以使用特定年份中可用的最高端交换机支持的最大集群配置的成本。 我们将这些价值观纳入2002年，2004年，2006年和2008年的高端10 GigE交换机的各种供应商的产品公告的历史研究。

我们使用我们的调查结果，构建当年技术支持最大的集群配置，同时保持1：1的超额认购。 表1显示了特定年份最大的10 GigE交换机; 我们在层次结构设计的核心层和聚合层中采用这些交换机。 表1还显示了当年最大的商品GigE交换机 我们EM

在脂肪树的所有层和边缘层处采用这些交换机进行分层设计。

采用高端交换机的传统技术所支持的最大集群尺寸一直受到可用端口密度的限制。 此外，高端交换机最初可用时，会带来极高的成本。 请注意，我们对传统层次结构的计算有些慷慨，因为聚合层上的商品GigE交换机直到最近才具有必要的10 GigE上行链路。 另一方面，基于胖树拓扑的群集规模较大，总成本下降得更快，更早（由于以下商品定价趋势较早）。 此外，在胖树拓扑中不需要更高速的上行链路。

最后，值得注意的是，今天，在所有节点之间构建一个具有10 Gbps带宽的27,648节点集群，在技术上是不可行的。 另一方面，fattree交换架构将利用近商品48端口10 GigE交换机，并承担超过6.9亿美元的成本。 尽管在大多数设置中可能是成本高昂的，但最重要的是，甚至不可能使用高端交换机的传统聚合来构建这样的配置，因为今天的交换机没有甚至超过10GeE的产品甚至以太网标准。

2. Clos网络/胖树

今天，商品和非商品交换机之间的价格差异提供了强大的动力，从许多小商品交易所建立大规模的通信网络，而不是更大，更昂贵的交换网络。 五十多年前，电话交换机的类似趋势导致Charles Clos设计了一种网络拓扑，通过适当地互连较小的商品交换机，为许多终端设备提供高水平的带宽[11]。

我们采用克洛斯拓扑的一个特殊的实例称为*fattree* [23]互连商品以太网交换机。 我们组织*勢k*进制脂肪树如图3有K吊舱，每个包含$K / 2$个开关的两层。 每K个-port在较低层交换机直接相连到$k / 2$的主机。 剩余的$K / 2$个端口的每一个连接到k第k端口/ 2在层次结构的聚合层。

有$（K / 2）^2$ *k-port*核心交换机。 每个核心开关具有连接到每K个吊舱的一个端口。 该*属*任何核心交换机的端口*我*连接到*我*英使得在每个吊舱交换机的汇聚层连续的端口连接到核心交换机上$（K / 2）$的进展。 一般情况下，用K -port开关内置一个胖树支持k *四分之三*的主机。 在本文中，我们专注于设计到K = 48。 我们的方法推广到对于k任意值。

胖树拓扑的一个优点是所有交换元件是相同的，使我们能够利用通信架构中所有交换机的便宜商品部分。 [2]此外，脂肪树*重排无阻塞的*，这意味着任意通信模式，有一些一套将饱和所有可用拓扑终端主机带宽路径。 由于需要防止TCP流的分组重新排序，因此在实践中实现1：1的超额认购率可能是困难的。

图3显示了脂肪树其中k = 4的最简单的非平凡实例。 连接到同一边缘交换机的所有主机都构成自己的子网。 因此，连接到相同下层交换机的主机的所有流量都被切换，而所有其他流量都被路由。

作为此拓扑的示例实例，从48端口GigE交换机构建的胖树将由48个pod组成，每个pod包含边缘层和每个具有24个交换机的聚合层。 每个pod中的边缘交换机每个分配24个主机。 该网络支持27,648个主机，由1,152个子网组成，每个子网共有24个主机。 在不同荚中的任何一对主机之间有576条相同的路径。 部署这种网络架构的成本将是$ *8*， 64 *男*，相较于$ 37 *男*前面所述的传统技术。

3. 概要

鉴于我们的目标网络架构，在本文的其余部分，我们解决了以太网部署中采用此拓扑结构的两个主要问题。 首先，IP /以太网通常在每个源和目的地之间构建单个路由路径。 对于甚至简单的通信模式，这种单路径路由将很快导致胖树上的瓶颈，从而显着限制了整体性能。 我们描述了IP转发的简单扩展，以有效利用胖树可用的高扇区。 第二，胖树拓扑可以在大型网络中施加明显的布线复杂性。 在某种程度上，这种开销在胖树拓扑中是固有的，但在第6节中，我们提供了包装和放置技术来改善这种开销。 最后，我们已经在第3节所述的Click [21]中构建了我们的架构的原型。第5节中介绍的初始性能评估证实了我们在小规模部署中的方法的潜在性能优势。

3. 建筑

在本节中，我们将描述一种以胖树拓扑互连商品交换机的架构。 我们首先激励在路由表结构中进行轻微修改的需要。 然后，我们将介绍如何为集群中的主机分配IP地址。 接下来，我们介绍两级路由查找的概念，以帮助跨胖胖树的多路径路由。 然后我们介绍我们采用的算法来填充每个交换机中的转发表。 我们还将流分类和流调度技术描述为备用多路径路由方法。 最后，我们提出一个简单的容错方案，并描述我们的方法的热和功率特性。

1. 动机

在该网络中实现最大二等分带宽需要在核心交换机之间尽可能均匀地扩展来自任何给定端口的输出流量。 路由协议，如OSPF 2 [25]通常采取的跳数作为度量“shortestpath”，并在*K*进制胖树拓扑结构（参见2.2节），有 *（K／2）2*这样的最短路径在不同的pod上的任何两个主机之间，但是只选择一个。 因此，交换机将集中到给定子网的流量集中到单个端口，即使存在提供相同成本的其他选择。 此外，根据OSPF消息的到达时间的交织，可以选择核心交换机的一小部分（也许只有一个）作为荚之间的中间链路。 这将导致这些点的严重拥塞，并且不利用胖树中的路径冗余。

扩展如OSPF-ECMP [30]除了在所考虑的交换机类中不可用，导致所需前缀数量的爆炸。 下级吊舱开关需要为*每一个*子网的其他 *（K／2）*的前缀; 合计*k* *的 *（K／2）2*前缀。

因此，我们需要一个简单，细粒度的pod之间的流量扩散方法，利用拓扑的结构。 交换机必须能够识别和特别处理需要均匀分布的流量类别。 为了实现这一点，我们提出使用基于目的地IP地址的低位比特扩展输出流量的两级路由表（见第3.3节）。

2. 解决

我们分配专用10中的网络中的所有IP地址。 *0。 0。 0/8*块。 我们按照熟悉四点缀形式具有下列条件：该吊舱开关被给予形式10 .*pod.switch*的地址。 1，其中*荚果*表示荚数目（在*[0，K* - 1]），以及*开关*表示开关的位置中的料盒（在*[0，K-1]*，开始从左至右，从下到上）。 我们给核心交换机的形式10 .*kji*，其中*j*的地址和*I*表示在第 *（k／2）2*核心交换机网格，开关的坐标（每个在*[1，（K／2）]*，从左上角开始）。

主机的地址是从连接到的pod开关中的; 主机具有形式的地址：10 .*pod.switch.ID*，其中*ID*是在该子网中的主机的位置（在[2 *中，k／2* + 1]，从左至右开始）。 因此，每一个低级别的交换机负责对于*k／2*的主机/ 24的子网（对于*k <256)*。 图3示出了用于对应于*k* = 4脂肪树本寻址方案的例子。 尽管使用可用的地址空间相对较为浪费，但它简化了构建路由表，如下所示。 尽管如此，该方案可扩展到4.2M主机。

3. 两级路由表

为了提供第3.1节中动机的均匀分配机制，我们修改路由表以允许两级前缀查找。 在主路由表中的每个条目将可能有额外的指针的*（后缀，口）*项小型二次表。 第一级前缀*终止*，如果它不含有任何secondlevel后缀和辅助表可以由更多**目的地** 10被指向。 *2。 0。* 3 **将采取虚线路径。**

图3：简单胖树拓扑。 使用3.3节中描述的两级路由表，从源头 10个 **包** 。 *0。 1。* **2**

图4：两级表示例。 这是开关的桌子

10。 2。 2。 **1。 目的IP传入数据包地址** *10。 2。 1。* 2 **被转发端口1，而目的IP地址为** 10 **的数据包** 。 *3。 0。* 3 **转发端口3。**

比一级的一级前缀。 而在主表条目都是左撇子（即/形式为1 *M* 0 *32 米m*的*前缀*口罩），辅助表中条目的形式0 *32 米*右手（即/ *M后缀*口罩1 *米*）。 如果最长匹配的前缀搜索产生非终止前缀，则找到并使用辅助表中最长匹配的后缀。

这种两级结构将稍微增加路由表查找延迟，但硬件前缀搜索的并行性质应确保只有边际惩罚（见下文）。 这是因为这些表格非常小。 如下图所示，任何荚交换机的路由表将包含不超过*克/* 2前缀和*k／2*后缀。

4. 两级查找实现

我们现在描述如何使用内容可寻址内存（CAM）[9]在硬件中实现两级查找。 CAM在搜索密集型应用中使用，并且比用于找到与位模式匹配的算法方法[15,29]更快。 CAM可以在单个时钟周期内在其所有条目之间执行并行搜索。 查找引擎使用特殊类型的CAM，称为三元CAM（TCAM）。 一个TCAM可以存储*不*匹配除0和1的在特定位置的*关心*的位，使它适于存储可变长度前缀，例如在路由表中找到的那些。 在缺点方面，CAM具有相当低的存储密度，它们非常耗电，而且

图5：TCAM两级路由表的实现。

每一点都昂贵 然而，在我们的体系结构，路由表可以在相对适度的大小的TCAM实现*（K*条目每32比特宽）。

图5显示了我们提出的两级查找引擎的实现。 TCAM存储地址前缀和后缀，后缀依次索引存储下一跳的IP地址和输出端口的RAM。 我们将左手（前缀）条目存储在较大地址中的数字较小的地址和右手（后缀）条目中。 我们对CAM的输出进行编码，以便输出具有数字最小匹配地址的条目。 这满足了我们两级查找的具体应用的语义：当一个数据包的目标IP地址与左撇子和右撇子相匹配时，选择左撇子。 例如，使用在图5中，与目的地IP地址10的数据包的路由表。 *2。 0。* 3相匹配的左手入口*10。 2。* 0 .*X*和右手入门*XXX* 3。 该数据包的端口0正确转发然而，一个数据包目的IP地址为*10。 3。 1。* 2场比赛只有右手入门*XXX* 2和转发端口2。

5. 路由算法

胖树中的前两级开关作为过滤流量扩散器; 任何给定的pod中的下层和上层交换机都具有到该pod中的子网的终止前缀。 因此，如果主机向同一个pod中的另一个主机发送数据包，但在不同的子网上，则该pod中的所有上层交换机将具有指向目标子网交换机的终止前缀。

对于所有其他传出吊舱间流量，荚交换机有一个默认/ 0前缀与辅助表匹配主机ID（目标IP地址的最低显著字节）。 我们使用主机ID作为确定性熵的来源; 否则将导致流量均匀地传出链接到核心交换机之间扶摇[3]。 这也将导致相同主机的后续数据包遵循相同的路径，因此避免数据包重新排序。

在核心交换机中，我们为所有网络ID分配终止的第一级前缀，每个前缀指向包含该网络的适当的端口。 一旦一个数据包到达核心交换机，恰好有一个链接到其目标吊舱，并且该交换机将包括该数据包的POD（10 .*POD。0 0/16端口*）的终端/ 16前缀。 一旦分组到达其目的地荚，接收上层吊舱开关也将包括一个（10 .*pod.switch 0/24，端口*）前缀该分组直接到它的目的地的子网开关，它是最后切换到其目的地主机。 因此，交通扩散仅在分组旅程的前半部分发生。

可以设计分布式协议，在每个交换机中逐步建立必要的转发状态。 然而，为简单起见，我们假设拥有集群互连拓扑的全面知识的中心实体。 该中央路由控制负责静态生成所有路由表，并在网络设置阶段将表加载到交换机中。 动态路由协议还将负责检测各个交换机的故障并执行路径故障切换（参见第3.8节）。 下面我们总结了在pod和core交换机上生成转发表的步骤。

荚开关

在每个pod开关中，我们为包含在同一个pod中的子网分配终止前缀。 对于吊舱间流量，我们添加一个/ 0前缀与辅助表匹配的主机标识。 算法1显示用于生成上部pod开关的路由表的伪代码。 出端口模数移位的原因是为了避免来自寻址到具有相同主机ID的主机的不同下层交换机的流量进入相同的上层交换机。

对于较低荚开关，我们简单地忽略/ 24子网前缀的步骤，在第3行，因为该子网本身的流量切换，以及区域内和跨荚通信应该上层交换机之间进行平分。

核心交换机

由于每个核心交换机被连接到每荚（端口*i*被连接到荚果*i* ）*中*，核心交换机仅包含终止/ 16前缀指向目的地豆荚，如图算法2这种算法生成的表的大小是在*k个*线性。 网络中没有开关包含具有大于*k*第一级前缀或*K／2*的第二级后缀的更多的表。

路由示例

使用双电平表示的网络操作中，我们给出了由源10取为一个包路由决策的实例。 *0。 1。* 2到目的地*10。 2。 0。* 3，如示于图3首先，源主机的网关交换机 *（10。0 1 1）*将只包带/ 0第一级前缀基于主机匹配，并因此将转发分组根据该前缀的辅助表的ID字节。 在该表中，分组0相匹配。 *0。 0。 八分之三*后缀，它指向端口2和交换机*10。 0。 2。* 1。 开关*10。 0。 2。* 1也遵循端口3相同的步骤和前锋，连接到核心交换机*10。 4。 1。* 1。核心交换机的分组匹配到一个终止10 。 *2。* 0 。0／16前缀，它指向目的地荚2 **1分 的foreach** *荚X 在* [0 ，*K* - 1] **做**

2. **的foreach** *开关ž 在* [（ *K／2*） *中，k* - 1] **做**
3. **的foreach** *子网 我 在* [0 ，（ *克/* 2）- 1] **做**

```
4. addPrefix（10 X。Ž 0.1，10 X。我 0.0 / 24，我）；
5. 结束
6. （10 addPrefix X。Ž 0.1，0.0.0.0/0，0）；
7. 的foreach 主机ID 我 在 [2 ，（K / 2）+ 1] 做
8. addSuffix（10。X。Ž 0.1，0.0.0。我 / 8，

                                        （我 - 2 + Ž）MOD（K / 2）+（K / 2））；

9. 结束
10. 结束
11. 结束
```

算法1：生成聚合交换机的路由表。假设函数签名*addPrefix（开关，前缀，端口）*，*addSuffix（开关，后缀，端口）*和*addSuffix*增加了一个第二级后缀到最后添加的第一级前缀。

```
1. 的foreach Ĵ 在 [1 ，（克 / 2）] 做
2. FOREACH 我 在 [1 ，（克 / 2）] 做
3. 的foreach 目的地荚 X 在 [0 ，（克 / 2）- 1] 做
4. addPrefix（10 k。Ĵ。我，10。X .0.0 / 16，X）；
5. 结束
6. 结束
7. 结束
```

算法2：生成核心交换机的路由表。

端口2上，以及开关10。2。2。1。此开关属于相同群作为目的地的子网，并因此具有端接前缀，10。2。0。0/24，它指向负责该子网，开关10。2。0。1上端口0从那里，标准开关技术将分组传送到目标主机10。2。0。3。

请注意，从同时通信10。0。1。3到另一个主机10。2。0。2，传统的单路径IP路由将遵循如上因为两个目的地是在同一个子网流量相同的路径。不幸的是，这将消除所有的胖树拓扑的扇出效益。相反，我们的双级表查找允许转10。0。1。1到第二流转发到10。0。3。1基于在两电平表右手匹配。

6. 流分类

除了上述两种电平的路由技术，我们还考虑两个可选动态路由技术，因为它们是目前在几种商业可用的路由器[10,3]。我们的目标是要量化这些技术的潜在好处，但承认他们将承担额外的每个数据包的开销。重要的是，在这些方案中的任何维护的状态是软的，个别开关能回落到两电平的路由的情况下的状态被丢失。

随着流量扩散到核心交换机的一个替代方法中，我们在荚交换机执行与动态端口重新分配流分类，以克服可避免的本地拥塞（例如情况下，当两个流竞争同一输出端口时，即使具有另一个端口到目的地相同的成本未充分利用）。我们定义一个*流*与用于该数据包报头（通常源和目的地IP地址，目的地传输端口）的字段的子集的相同的条目的分组序列。特别是，荚果开关：

1. 识别相同流的后续分组，并且forwardthem同一输出端口上。
2. 周期性地重新分配流量输出portsto最小数量的最小化不同的端口的总流量之间的任何差异。

步骤1是对分组重排序的量度，而第2步的目的是确保在流上向上指向的端口动态地改变流尺寸的面部公平分配。4.2节更详细地介绍了流分类的我们的实现和流量分布启发。

7. 流调度

几项研究已经表明，转移时间的分布和因特网业务的脉冲串长度是长尾[14]，并且其特征在于几个大长时间流量（负责大部分带宽）和许多小短命合的[16]。我们认为，路由大流量在确定网络的实现分切带宽，因此应该特殊处理最重要的作用。在流管理此替代方法中，我们安排大流量到彼此最小化重叠。中央调度使这一选择，与网络中的所有活动的大量流动的全球知识。在这最初的设计中，我们只考虑单一的大流量从每次打开一个主机发起的情况。

1. 边缘交换机

和以前一样，边缘交换机本地最初分配一个新的流向leastloaded端口。然而，边缘交换机另外检测到任何外出流动，其大小增长高于预定阈值，并周期性地发送通知到一个中央调度器指定源和目的地的所有活性大流量。这代表了在一个争用的路径由用于该流的放置边缘交换机的请求。

请注意，与第3.6节，此方案不允许边缘交换机独立重新分配流量的端口，无论大小。中央调度是命令重新分配权力的唯一实体。

2. 中央调度

中央调度器，可能复制的，跟踪所有活性大流量，并尝试分配如果可能的话它们非冲突路径。调度程序维护在网络中足见其可用性进行大流量的各个环节布尔状态。

对于吊舱间流量，回想有（K / 2）^{2}的网络中的任何给定的一对主机之间的可能路径，并且每个这些路径的对应于核心交换机。当调度器接收到新的流的通知时，它线性地搜索整个核心交换机找到一个其相应的路径组件不包括一个保留的链路。[4] 当找到这样的路径，调度标记这些链接为保留，并在源与荚对应于该流的所选择的道路正确的传出端口通知有关大写和层交换机。类似的搜索用于帧内荚大流量进行;此时，用于通过上层荚开关的无竞争路径。调度垃圾收集，其上次更新超过给定的时间早流动，清除其保留。需要注意的是边缘交换机不阻塞等待调度程序来执行此计算，但最初处理大流量像任何其他。

8. 容错

任何一对主机之间的可用路径的冗余，使用于容错的胖树拓扑吸引力。我们提出了一个简单的故障广播协议，它允许交换机路线周围链路或开关失效的一个或两个跳下游。

在这个方案中，网络中的每个交换机维护一个*双向转发检测*会话（BFD [20]）与每个其邻居的，以确定何时一个链路或邻近开关失败。从容错的角度来看，失败的两个类中可被风化：（A）芯和上级开关之间的荚，和（b）内较低和上层交换机之间。显然，一个较低级别的开关的故障将会导致对直接连接的主机断开; 在叶子冗余开关元件是容忍这样的故障的唯一方法。我们在这里描述的链路故障，因为开关故障触发相同BFD警报，并引起同样的反应。

1. 较低到上部层交换机

较低和上层交换机之间的链路故障影响三类业务：

1. 传出之间和内部荚流量thelower层交换机始发。在这种情况下，本地流分类器设置该链接到无穷大的"成本"和不分配它的任何新流，并且选择另一个可用的上层开关。
2. 使用上层交换机作为中介帧内荚流量。作为响应，该开关广播一个标签通知所有其它较低层交换机的链路故障的相同吊舱。分配新流预期的输出端口是否对应于这些代码中的一个时，这些开关将检查，如果可能避免它。[五]

3. 间荚流量进入上层开关。连接到上部层交换机核心交换机有它作为其唯一到荚访问，因此上层切换广播该标签将其所有核心交换机标志着它不能携带业务到下层交换机的子网。这些核心反过来镜此标记来它们被连接到在其它荚所有上层交换机进行切换。最后，上层交换机分配新流至该子网时避免单个受影响的核心交换机。

2. 上层至核心交换机

从上层交换机到核心的链路的故障影响两个类的通信：

1. 即将离任的荚间流量，在这种情况下，本地路由表，标志着受影响的链接不可用，并在本地选择另一核心交换机。
2. 传入荚际交通。在这种情况下，核心交换机广播一个标签到所有其他上层交换机它被直接连接到其标志着无法携带信息流，使其整个*吊舱*。如前，这些上层开关将分配发往该荚流动时避免核心交换机。

当然，当链路故障和开关回来并重新建立BFD会话，前面的步骤是相反的，取消其作用。此外，适应第3.7节的方案以适应链路和交换机故障是相对简单的。调度标记报道为向下繁忙或不可用，从而不合格，其包括它从考虑任何路径，实际上路由周围故障大流量的任何链接。

9. 电源和散热问题

除了性能和成本，中出现的数据中心设计中的另一个主要问题是功耗。组成互连的更高层级的数据中心交换机通常需要消耗数千瓦特，并且在大规模数据中心互连的功率要求，可以数百千瓦。几乎同样重要的是从交换机散热的问题。企业级开关产生大量的热，因此需要专用的冷却系统。

图6：电源和散热的比较。

在这一节中，我们分析了我们的架构在功耗要求和散热，并与其他典型的方法进行比较。我们立足于在交换机数据表中报告的数字我们的分析，虽然我们承认，这些报道的值由不同的供应商以不同的方式测量的，因此可能并不总是反映在部署系统的特性。

图7：总功率消耗和散热的比较。

比较每类开关电源需求，我们通过开关在一个开关可以支持Gbps的总总带宽正常化的总功耗和散热。图6地块均在三个不同的交换机型号。正如我们所看到的，10个吉比特以太网交换机（最后三个在x轴）大致消耗每双Gbps的瓦特和消散商品的大致三倍的热的GigE开关时归一化带宽。

最后，我们还计算了可支持大约27K主机互连估计总功耗和散热。对于分层设计，我们采用576个的ProCurve 2900边缘交换机和54的BigIron RX-32交换机（36在聚集和18在核心层）。胖树架构采用2,880美国网件GSM 7252S交换机。我们可以用更便宜的NetGear的开关，因为我们并不需要在胖树互连10个千兆以太网上行链路（存在于的ProCurve）。图7示出的是，虽然我们的体系结构采用多个单独的开关，功率消耗和散热优于那些由当前数据中心设计发生，以较少的56.6％的功耗和少56.5％的热耗散。当然，实际的功率消耗和散热必须部署来测量;我们留下这样的研究，我们正在进行的工作。

4. 实施

为了验证在本文中描述的通信架构中，我们构建的前面部分中描述的转发算法的简单原型。我们已经完成了使用NetFPGAs [24]的原型。该NetFPGA包含利用TCAMs IPv4路由器的实现。我们适当修改路由表查找程序，如第3.4节。我们修改总计不到100行额外的代码，并没有引入可测量的额外的查询等待时间，支持我们的信念，我们提出的修改可纳入现有的交换机。

开展较大规模的评估，我们还内置采用点击的原型，对我们的评价，本文的重点。点击[21]是支持实施的实验路由器设计的模块化软件路由器架构。A单击路由器是称为分组处理模块的图*元件*执行的任务，例如路由表查找或递减一个数据包的TTL。当链接在一起，点击元素可以进行复杂的路由器功能和协议软件。

1. TwoLevelTable

我们建立了一个新的点击元素，*TwoLevelTable*，它实现了3.3节所述两级路由表的想法。该元素具有一个输入和两个或多个输出。路由表中的内容使用的输入文件，让所有的前缀和后缀初始化。对于每一个数据包时，TwoLevelTable元素查找最长匹配的一级前缀。如果前缀终止，它会立即转发前缀的端口上的数据包。否则，将执行在二次表右手最长的匹配后缀搜索和相应的端口上转发。

此元件可以代替在[21]中提供的符合标准的IP路由器的配置示例的中央路由表元素。我们生成的IP路由器上的所有端口的带宽限制元件来模拟链路饱和容量的增加修改的类似4端口版本。

2. FlowClassifier

为了提供3.6节中的流分类功能，我们描述了我们实现点击元素的*FlowClassifier*有一个输入和两个或多个输出。它执行基于所述传入的数据包，使得后续的具有相同的源和目的地出口相同的端口的数据包（以避免分组重排序）的源和目的地IP地址的简单流分类。该元素具有最小化其highest-和加载的最低输出端口的总流量之间的差的增加目标。

即使个别流量大小事先已知的，该问题是NP-hard的装箱优化问题[17]的变体。然而，流量的大小其实不知道*先验*，使问题变得更加困难。我们按照算法3.每隔几秒钟列出的启发式贪婪，启发式尝试切换，如果需要的话，最多三个流的输出端口，以尽量减少其输出端口的总流量之间的差异。

```
  //调用上的每个输入数据包 1
IncomingPacket( 分组 )

  2. 开始
  3. 分组的散列源和目的地IP领域;

    //我们以前见过这个流程？

  4. 如果 看到（散列） 然后
  5. 查找先前分配的端口 X;
  6. 发送端口包 X;
  7. 其他
  8. 记录新流 'F;
  9. 分配 'F到加载至少向上端口 X;
 10. 发送端口的分组 X;
 11. 结束
 12. 结束

  //调用每 吨 秒 13个
RearrangeFlows()

 14. 开始
 15. 对于 I = 0至 2 做
```

16. 向上查找端口 $p_{最大}$ 和 $p_{最小}$ 与最大

最小的聚集体传出的通信，分别; **17** 计算 D ，之间的差 $p_{最大}$ 和 $p_{分钟}$ ；

18. 找到最大的流 $^{\prime\prime}F$ 分配给端口 $p_{最大}$ ，其尺寸小于 δ ；
19. **如果** *这样的流存在* **然后**
20. 切换流的输出端口 $^{\prime\prime}F$ 到 $p_{分钟}$ ；
21. **结束**
22. **结束**
23. **结束**

算法3：流分类启发。对于在第5节的实验中，*吨*为1秒。

回想一下，FlowClassifier元件是用于交通扩散到两电平表的替代方案。网络使用这些要素将采用普通的路由表。例如，上部荚开关的路由表包含了所有分配给该吊舱像之前的子网前缀。然而，除此之外，我们增加了一个/ 0前缀匹配需要向上均匀地扩散到核心层的所有剩余荚际交通。仅匹配前缀的所有数据包被定向到FlowClassifier的输入。分类器试图均匀地分配其输出中传出帧间荚流根据所描述的启发式的，其输出被直接连接到核心交换机。核心交换机不需要分类，以及它们的路由表不变。

请注意，此解决方案具有不需要正确性软状态，但只作为一个性能优化。此分类是偶尔破坏性的，作为流的最小数目也可以周期性地重新设置，从而可能导致分组重排序。但是，它也适应于动态改变的流的大小和

在长期的"公平"。[6]

3. FlowScheduler

如在第3.7节所描述的，我们实施了元件*FlowReporter*，它驻留在所有的边缘交换机，并且检测输出流，其尺寸大于给定的阈值。它定期发送通知至约这些活性大流量的中央调度器。

所述*FlowScheduler*元件接收关于从边缘交换机活性大流量通知，并试图找到他们无竞争路径。为此，它使网络中的所有链接的二进制状态，以及先前放置流的列表。对于任何新的大流量，调度器执行的源主机和目标主机之间的所有等开销的路径中的线性搜索以找到一个其路径部件都是未保留。如果找到这样的路径，流痕调度所有组件环节预留和发送关于该流程对有关荚开关路径的通知。我们还修改吊舱的交换机处理来自调度这些端口重新分配的消息。

调度器维护两个主要的数据结构：在网络中所有链路的二进制数组（总共$4 * k * (K/2)^2$个链接），以及先前放置流动及其分配的路径的哈希表。平均为新流放置线性搜索需要$2 * (K/2)^2$的存储器访问，使得调度器的计算复杂度为$\emptyset(k^3)$用于空间和$\emptyset(k^2)$对时间。对于一个典型的值k（每个交换机端口的数目）是48，使得这两个值可管理的，如在第5.3节定量。

5. 评价

衡量我们设计的总分带宽，我们生成通信映射的基准套件来评估4端口胖树的使用TwoLevelTable交换机，FlowClassifier和FlowScheduler性能。我们比较这些方法的标准分层树有3 。6：1收敛比，类似于当前的数据中心设计中发现的。

1. 实验说明

在4端口胖树，有16台主机，四个荚（每个具有四个开关），和四个核心交换机。因此，存在总共20个开关和16台端主机的是（对于较大的簇，开关的数量会比主机的数目小）。我们这些复用元件36到10台的物理机器，由一个48端口的ProCurve 2900交换机1个千兆以太网链路互连。这些机器在2.33GHz的双核英特尔至强处理器的CPU，具有4096KB缓存和4GB的RAM，运行Debian GNU / Linux的2.6.17.3。开关中的每荚托管一台机器上;每个吊舱的主机托管一台机器上;和剩下的两个机器上运行的两个核心开关，每个。无论是交换机和主机是点击配置，在用户级别运行。在网络中的Click元件之间的所有虚拟链路是带宽受限的，以96Mbit / s的以确保该结构没有CPU限制。

对于分层树网络的情况相比，我们有四台机器运行的每个四台主机和四台机器分别运行一个额外的上行4个荚开关。四个荚开关被连接到专用机器上运行的4端口的核心交换机。强制执行3.6：从荚切换到核心交换机上行链路1的超售，这些链路是带宽受限于106.67Mbit / s，并且所有其他链路被限制为96Mbit /秒。

每个主机生成传出通信的恒定96Mbit /秒。我们衡量它的传入流量的速率。所有主机的所有双射通信映射的最小总计传入流量是该网络的有效分带宽。

2. 基准测试套件

我们根据以下策略生成通信对，与任何主机接收到恰好一个主机流量增加的限制（即，映射是1对1）：•随机：主机发送到任何其他主机的网络与统一的概率。

- 步幅（*我*）：具有索引主机*x*将发送到所述主机具有索引（ $X + 我$ ）*模*16。

交错习题（*SubnetP，初步发展大纲图*）：当一个主机就会发送到它的概率子网的另一台主机*SubnetP*，并与概率荚*初步发展大纲图*，以及其他人的概率为1 - *SubnetP - 初步发展大纲图*。

- 跨荚来电：多荚发送到不同的主机在同一个吊舱，以及所有碰巧选择了相同的核心交换机。该核心交换机到目标吊舱链接将被超额认购。最坏情况下的*本地*对于这种情况下收敛比为（ $k - 1$ ）：1。
- 同一ID传出：主机在同一个子网发送到不同的网络中的主机，以使目标主机具有相同的主机ID字节别处英寸 静态路由技术迫使他们采取同样传出向上端口。对于这种情况下的最坏情况下的比率（ $K/2$ ）：1。这就是FlowClassifier有望最提高性能的情况下。

测试	树	两级表	流分类	流调度
随机	53.4%	75.0%	76.3%	93.5%
步幅（1）	100.0%	100.0%	100.0%	100.0%
步幅（2）	78.1%	100.0%	100.0%	99.5%
步幅（4）	27.9%	100.0%	100.0%	100.0%
步幅（8）	28.0%	100.0%	100.0%	99.9%
交错习题（1.0，0.0）	100.0%	100.0%	100.0%	100.0%
交错习题（0.5，0.3）	83.6%	82.0%	86.2%	93.4%
交错习题（0.2，0.3）	64.9%	75.6%	80.2%	88.5%
最坏的情况：				
跨荚传入	28.0%	50.6%	75.1%	99.9%
同一ID传出	27.8%	38.5%	75.4%	87.4%

表2：网络的集合带宽，作为理想的二分带宽为树，两级表，流分类和流调度方法的百分比。对于胖树网络理想的二分带宽是1.536Gbps。

3. 结果

表2示出了上述实验的结果。这些结果是在5个运行的基准测试的/置换，平均在每1分钟。如所预期的，对于任何全间荚通信模式，传统的树浸透链接到核心交换机，并从而达到周围28％的带宽理想的为在这种情况下，所有主机。树好显著进行沟通对是彼此接近。

两电平表开关达到约75％的理想的分带宽用于随机通信模式。这可以通过表的静态性质来解释:在任何给定的子网两个主机有一个50％的发送到主机与同一主机ID的机会，在这种情况下，因为它们是同一个输出端口上转发它们的组合吞吐量减半。这使得双方的预期是75％。我们预计两电平表的性能改善与提高随机通信*k*一样会有多个流碰撞具有较高的单一链路上的可能性较小*k*。所述吊舱间传入情况下，用于两级表给出了50％的二分带宽;然而，相同-ID传出效果是通过在核心路由器拥塞进一步加剧。

由于其动态流量分配和重新分配，流分类性能优于传统的树，在所有情况下twolevel表，约为最坏情况分带宽的75％。然而，由于拥堵它避免的类型是完全地方仍然不完善:有可能在因为由一种或两跳上游路由决定的一个核心交换机造成堵塞。这种类型的次优路由的发生是因为开关只有可用的本地知识。

该FlowScheduler，在另一方面，作用于全球知识，并尝试大流量分配到不相交的路径，从而实现93％的理想分带宽随机通信映射，并超越所有的基准测试的所有其它方法。中心调度程序的所有活动的大流量的知识和所有链接的状态的使用可能是不可行的任意大的网络，但胖树拓扑的规律性大大简化了非竞争路径的搜索。

在一个单独的试验中，当适度置备2.33GHz的商品的PC上运行的表3示出了用于中央调度程序的时间和空间要求。用于改变*k*，我们产生假放置请求（每个主机一个）来测量平均的时间来处理一个放置请求，以及用于维持链路状态和流动状态的数据结构所需的总存储器。对于27K主机的网络，调度需要的内存适度5.6MB，并可能将在为0.8ms下一个流程。

k	主机	平均时间/REQ（*微秒*）	链路状态记忆	流状态记忆
4	16	50.9	64乙	4 KB
16	1024	55.3	4 KB	205 KB
24	3456	116.8	14 KB	691 KB
32	8,192	237.6	33 KB	1.64 MB
48	27,648	754.43	111 KB	5.53 MB

表3：流量调度程序的时间和内存需求。

6. 打包

胖树拓扑结构集群互联的一个缺点是连接所有机器所需要的线缆数量。用10个的GigE开关进行聚合的一个微不足道的好处是在传输带宽高达层次结构的相同量的所需的电缆的数量减少10的因子。在我们提出的胖树拓扑，我们不充分利用万兆以太网链路或交换机都因为非商品件会膨胀的成本，更重要的是，因为胖树拓扑需要依靠大扇出到多个交换机在每个层次结构中的层，以实现其缩放属性。

承认增加布线费用是固有的，胖树拓扑，在本节中，我们考虑一些封装技术来减轻这种开销。总之，我们所提出的封装技术消除了大部分所要求的外部布线的，减少了所需的布线，这又简化了集群管理并降低总成本的总长度。此外，该方法允许对网络的增量部署。

图8：建议的包装解决方案。唯一的外部电缆豆荚和核心节点之间。

我们提出我们的做法，最大容量27,648个节点的集群利用48端口以太网交换机作为胖树的构建块的上下文。这种设计推广到不同大小的簇。我们开始与个别豆荚弥补较大的聚类复制单元的设计，参见图8。每个吊舱包括576个机和48个单独的48端口的GigE开关。为[illegible]容纳48吨的机器。因此，每个吊舱包括12个机架与每个48吨的机器。

我们将48个开关组成每个吊舱胖树的前两层在一个集中的机架。然而，我们假设到48个开关打包[illegible]些端口中的，576直接连接到在盒的机器，在对应边缘连接。另一个576个端口扇出到在每个开关57[illegible]2304个端口（48 * 48）。其他1152个端口在荚果开关内部接线，以考虑吊舱的边缘层和汇聚层之间[illegible]

我们进一步蔓延形成横跨各个容器胖树的顶部576个所需的核心交换机。假定总共48个吊舱，每[illegible]的，12将直接连接到附近放置在同一荚核心交换机。剩余的电缆将散开，在套12的，以容纳在远稍[illegible]到荚开关的事实打开更多的机会为适当的“电缆包装”，以减少布线的复杂性。

最后，最大限度地减少电缆总长度是另一个重要的考虑因素。为了这样做，我们周围放置吊舱开[illegible]将减少相对于在一个荚单个机架的更“水平”的布局的电缆长度。类似地，我们躺在荚出一个7×7网[illegible]减少到适当的核心交换机角架间布线距离，并将支持电缆长度和包装，以支持吊舱间连通的一些标准化。

我们也认为是交替的设计，并没有收集切换到中央机架。在这种方法中，两个48端口交换机将被分配到每个齿条。主机将互连到所述开关在套24这种方法具有需要更短的电缆将主机连接到其第一一跳交换机和消除这些电缆一起如果机架得到适当内部包装的优点。我们放弃了这个，因为我们将失去以消除互连边缘和汇聚层，每荚内的576线的机会。这些电缆需要纵横交错12个机架中的每个吊舱，增加显著的复杂性。

7. 相关工作

我们在数据中心网络架构的工作一定是建立在工作中的一些相关领域。也许最密切相关，我们的努力是构建可扩展互连多方努力，在很大程度上出来的超级计算机和大规模并行处理（MPP）的社区。许多MPP互连已经被组织成脂肪的树木，包括思维机器[31，22]和SGI [33]系统。思维机器采用伪随机转发决定胖树链路之间进行负载均衡。虽然这种方法取得了良好的负载均衡，很容易出现数据包重新排序。 Myrinet的开关[6]也采用胖树拓扑和已经流行了基于集群的超级计算机。 Myrinet的采用基于预定的拓扑知识源路由，使直通低延迟开关的实施方式。主机还负责通过测量往返延迟可用路由进行负载分担。相对于所有这些努力，我们专注于利用商品以太网交换机互连大规模集群，表现出适当的路由和封装技术。

InfiniBand的[2]是用于高性能计算环境一个受欢迎的互连，目前迁移到数据中心环境。 InfiniBand的也能达到使用的Clos拓扑的变体可扩展的带宽。例如，Sun最近宣布从720布置在5级胖树24端口InfiniBand交换机建有3456端口InfiniBand开关[4]。然而，InfiniBand的强加自己的1-4层协议，使以太网/ IP / TCP在某些设置更吸引人，特别是为10Gbps以太网的价格继续下降。

另一种流行的MPP互连拓扑是一个圆环，例如在蓝色基因/ L [5]，克雷XT3 [32]。环面直接互连处理器在一些数量的其邻居的*k*维点阵。维数确定源和目标之间的跳的预期数量。在MPP环境中，环面具有不具有电简单点对点链接沿着任何专用开关元件的益处。在集群环境中，环面的布线复杂性很快变得过高和卸载所有路由和转发功能，以商品主机/操作系统通常是不切实际的。

我们提出的转发技术涉及现有的路由技术，例如OSPF 2和等价多路径（ECMP）[25，30，19]。我们的多路径建议利用胖树拓扑结构的具体性能取得良好的业绩。相对于我们的工作，ECMP提出了三种类别的无国籍转发算法：（I）循环赛和随机化;（ii）区域分割，其中一个特定的前缀被分成两个具有较大的掩码长度;和（iii）一个分割散列技术的一组基于所述源和目的地址的输出端口之间流动。第一种方法免受潜在的包重新排序问题，对于TCP尤其是有问题的困扰。第二种方法会导致路由前缀的数量爆破。在25000台主机的网络，这将需要约600,000路由表条目。除了成本的增加，在这个规模的查表将产生显著的等待时间。出于这个原因，目前的企业级路由器允许最多16路ECMP路由。最后的方法不作出分配决策，这将很快导致超额即使是简单的通信模式占流量的带宽。

8. 结论

带宽是越来越多地在大规模集群的扩展性瓶颈。在层次结构的顶部处理了围绕开关的层次结构，以价格昂贵，非商用交换机这个瓶颈中心现有的解决方案。在任何给定时间点，高端开关的端口密度，而在同一时间招致成本高限制整体的簇大小。在本文中，我们提出一个利用商品以太网交换机用于大规模集群提供可扩展带宽的数据中心通信的体系结构。我们立足我们的拓扑周围的胖树，然后存在技术，同时保持与以太网，IP和TCP向后兼容执行可扩展路由。

总体而言，我们发现，我们能够以显著成本低于现有技术，提供可扩展的带宽。虽然需要额外的工作来充分验证我们的做法，我们认为更大的商用交换机的数量有故址在数据中心高端交换机在商品PC集群已经取代了高端计算环境的超级计算机同样的方式的可能性。

致谢

我们希望对本文初稿感谢乔治·巴尔加斯以及匿名裁判的宝贵意见。

9. 参考

1. 思科数据中心2.5架构设计指南。

http://www.cisco.com/univercd/cc/td/doc/溶液/ dcidg21.pdf 。

2. InfiniBand的体系结构规范第1卷，版本1.0。 http://www.infinibandta.org/specs 。
3. 瞻博网络J-流量。 http://www.juniper.net/techpubs/软件/ ERX / junose61 / swconfig的路由，VOL1 / HTM L / IP-jflow -统计- config2.html 。
4. 孙数据中心交换机3456架构白皮书。 http://www.sun.com/products/networking/数据中心/ ds3456 / ds3456_wp.pdf 。
5. M·布卢姆里奇，D·切，P·科特斯，A.加拉，M·吉帕帕，
   P. 海德堡，S.辛格B. Steinmacher-Burow ，T·塔克肯，和P。Vranas。设计与蓝色基因/ L环面互连网络的分析。 *IBM研究报告RC23025（W0312-022）*，3，2003。
6. N·博登，D·科昂，R·费尔德曼，A. Kulawik，C·塞茨，和J·塞佐维奇。Myrinet的：一个千兆每秒LOCAL的 REA网络。 *微，IEEE* ，15（1），1995年。
7. S·布林和L.页。大规模超文本网络搜索引擎剖析。 *计算机网络和ISDN系统* ，30（1-7），1998年。
8. R·谢弗里萨恩，M·拉姆齐，C·费赫特和I.沙拉波夫。工作负载的特性ü 在高性能和技术计算的sed。在 *对超级计算国际会议* ，2007年。
9. L·奇斯文和杰·达克沃斯。内容寻址和联想记忆：替代了无所不在的RAM。 *计算机* ，22（7）：51 - 64，1989年。
10. B·克莱兹。词 SCO系统NetFlow服务出口版本9. RFC 3954，互联网工程任务组，2004年。
11. C·克洛斯。的研究无阻塞交换网络。 *贝尔系统技术杂志* ，32（2），1953年。
12. J·迪恩和S·马沃特。MapReduce的：简化数据处理 大型集群。 *USENIX研讨会操作系统设计与实现* ，2004年。
13. G·德坎迪亚，D·哈斯托兰，M·贾帕尼，G. Kakulapati，

A.拉克什曼，A. Pilchin，S·锡瓦萨布拉马尼安，P. Vosshall和W·博赫尔斯。迪纳摩：亚马逊的高度可用的key-value存储。*ACM研讨会作业系统原理*，2007年。

14. 阿布·道尼。证据在互联网长尾分布。 *ACM SIGCOMM研讨会互联网测量* ，2001年。
15. W·瑟顿，G·巴尔加斯，和Z·迪舍。树位图：

硬件/增量式更新软件IP查找。
SIGCOMM计算机通信评论 ，34（2）：97-122，2004年。

16. SB佛瑞德，T·博纳德，A. Proutie 重，G.ř é GNI é ，和JW罗伯茨。

统计带宽共享：拥塞的流量水平的研究。*SIGCOMM计算机通信评论*，2001年。

17. MR Garey和兹·约翰逊。 *计算机和顽固性：指南NP完全性理论* 。惠·弗里曼 ，1979年。
18. S·马沃特，H·戈壁夫和S.-T. 梁。该谷歌文件系统。 *ACM SIGOPS操作系统审查* ，37（5），2003年。
19. C·霍普斯。分析的等价多路径算法。RFC 2992，互联网工程任务组，2000。
20. D·卡茨，D·沃德。 BFD用于IPv4和IPv6（单跳）（草案）。技术报告，互联网工程任务组，2008年。
21. E·科赫莱尔，R·莫里斯，B·切，J·詹诺蒂，和M·卡肖克。点击模块化路由器。 *计算机系统ACM交易* ，18（3），2000。
22. C·莱泽森，Z·阿汉德 ，D·道格拉斯，C·费曼，M·甘慕克，J·希尔，D·希利斯，B·库斯茨莫尔，M·皮尔，D·韦尔斯，等。连接机CM-5的网络架构（扩展摘要）。 *ACM研讨会并行算法和架构* ，1992年。
23. CE乐 iserson。发树：对硬件的高效的超级计算通用网络。 *IEEE TRANSACTIONS ON计算机* ，34（10）：892 - 901，1985。
24. J.洛克伍德N·麦基敦，G·沃森，G·吉布，P·哈特克，J·纳斯，R·拉赫拉曼和J.洛。NetFPGA - 地理标志的开放式平台 gabit速率的网络交换和路由。在 *微电子系统教育IEEE国际会议* ，2007年。
25. J·莫伊。OSPF版本2，RFC 2328，互联网工程任务组，1998年。
26. F·施米克和R·哈斯金。GPFS：大型计算群集共享磁盘文件系统。在 *USENIX会议上的文件和存储技术* ，2002年。
27. LR斯科特T·克拉克和B·巴格里。 *科学并行计算* 。普林斯顿大学出版社，2005年。
28. SGI 开发中心开放源码的Linux XFS。XFS：高性能的日志文件系统。 http://oss.sgi.com/projects/xfs/ 。
29. V·斯里尼瓦萨恩和G·巴尔加斯。更快的IP查找使用受控前缀扩展。 *在计算机系统ACM交易（TOCS）* ，17（1）：1 - 40，1999年。
30. D·塞勒和C·霍普斯。多径问题的单播和组播下一跳选择。RFC 2991，互联网工程任务组，2000。
31. L·塔克和G·罗伯逊。建筑和连线机中的应用。 *计算机* ，21（8），1988。
32. J·韦特，S·阿拉姆，J·达尼根，TH，M·法希，P·罗特和
   P. 沃利。克雷XT3的早期评估。在 *IEEE国际并行与分布处理研讨会* ，2006年。
33. M·伍德克，D·罗布，D·罗和K. Feind。了SGI Altix 3000全球Shared- 内存架构。 *SGI白皮书* ，2003。

•

[1]我们使用术语*开关*在整个本文的其余部分，以指的是执行两个2层交换和第三层路由设备。

[2]。注意，开关的均匀性不是必需的，因为更大的开关可以在核心（例如，用于复用）被使用。虽然这些可能有更长的平均无故障时间（MTTF），这违背了成本优势，并保持相同的布线成本。

[3]由于表是静态的，有可能达不到完美的分布。我们研究在最坏情况下的通信模式
第5节

[4]发现对所有大流量的最佳位置要求要么知道源和所有的目的地流的时间提前或现有流的路径重新分配; 然而，这贪心试探法给出了一个很好的近似，并在模拟中实现94％的效率为27K主机之间随机注定流动。

[5]，我们依靠终端到终端的机制，以重新开始中断的流

[6]在这个意义上，初始安置决定正在不断因为所有流尺寸校正公平不断跟踪来近似流和端口的最佳分配。