基于事件驱动的 MapReduce 类流量产生方法与网络评测

邵 恩[1),2)] 孙凝晖[1)] 郭嘉梁[1),2)] 元国军[1),2)] 王 展[1)] 曹 政[1)]

[1)](中国科学院计算技术研究所计算机体系结构国家重点实验室 北京 100190)

[2)](中国科学院大学 北京 100049)

摘 要 大规模网络结构设计是构建大规模分布式系统和 E 级高性能计算集群的核心技术之一,底层网络设计者需要结合顶层应用通信流量特征,进行网络结构选型与优化.不当的应用通信模型会引起网络结构设计与实际需求的背离,进而导致系统通信和整体性能的下降.传统基于"黑盒"数据分析的流量建模方法存在业务建模粒度粗和应用数据规模扩展性差等缺陷.该研究引入模拟业务内部逻辑的"事件驱动"思想,提出一种针对主流计算模式 MapReduce 进行流量建模与流量产生方法.与真实应用流量的对比评测显示,该方法能够准确体现 MapReduce 计算业务所产生网络流量的特征.基于正确的流量模型,该文对四种主流数据中心网络进行了性能模拟分析.结果表明:相较负载随机均匀分布流量,同一种网络在负载 MapReduce 特性流量时性能将下降超过 30%,因此特性流量能更加明显地展现网络拥塞与瓶颈问题.仿真实验所得到的有关网络性能瓶颈、拓扑可扩展性以及网络性价比的结论,为大规模数据中心网络选型和性能优化提供了新的依据.

关键词 分布式系统;MapReduce;数据中心网络;事件驱动;大规模网络模拟

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2018.02265

Event-Driven Method for MapReduce Traffic Generation and Network Evaluation

SHAO En[1),2)] SUN Nin-Hui[1)] GUO Jia-Liang[1),2)] YUAN Guo-Jun[1),2)] WANG Zhan[1)] CAO Zheng[1)]

[1)](*State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

[2)](*University of Chinese Academy of Sciences, Beijing* 100049)

Abstract Interconnection network design is one of the core technologies in the constructions of exascale clusters and large-scale distributed systems. Such large-scale computing system is expected to be achieved in the near future due to the rapid innovations of semiconductor logic and memory, architectures, interconnections and other industry technologies. Among these, due to performance and cost factors, interconnection network plays a critical role in such a large-scale computing system. In large-scale clusters or datacenter, the design of interconnection network is facing greater challenges. Firstly, the increasing computing capacity of a single node requires the network providing higher bandwidth and lower latency. Secondly, the increasing number of nodes requires the network has extremely better scalability. Thirdly, the increasing scale of system leads to worse performance of collective communication, which is harmful to the performance and scalability

收稿日期:2016-11-29;在线出版日期:2017-09-30.本课题得到国家重点研发计划项目(2016YFB0200300,2016YFGX030148,2016YFB0200205,2016GZKF0JT006)、国家自然科学基金项目(61572464,61402444)、中国科学院战略性先导科技专项(XDB24060600)资助.**邵 恩**,男,1988 年生,博士研究生,工程师,中国计算机学会(CCF)会员,主要研究方向为高性能计算、计算机系统结构、SDN、Big Data、光电混合网络.E-mail:enshao@163.com.**孙凝晖**,男,1968 年生,博士,研究员,博士生导师,中国计算机学会(CCF)会员,主要研究领域为计算机系统结构、高性能计算.**郭嘉梁**,男,1992 年生,硕士研究生,主要研究方向为分布式系统、高性能计算.**元国军**,男,1983 年生,博士研究生,高级工程师,中国计算机学会(CCF)会员,主要研究方向为计算机系统结构、光互连网络.**王 展**,男,1986 年生,博士,助理研究员,中国计算机学会(CCF)会员,主要研究方向为虚拟化技术与计算机系统结构.**曹 政**,男,1982 年生,博士,副研究员,中国计算机学会(CCF)会员,主要研究方向为高性能计算和互连网络.

of applications. Fourthly, the increasing number of devices requires the network has better reliability. As the performance of compute nodes keep increasing, interconnection network has gradually become the bottleneck of large-scale computing system. However, switch chip, the core component of interconnection network, can offer limited aggregate bandwidth because of the constraint of physical processes and packaging technologies. The underlying network designers should consider the processing characteristics of the network traffic when selecting and optimizing the network architecture. Improper traffic model will cause the departure between network architecture and characteristics of communication, which will reduce the overall performance of data centers and clusters. Big data platform has the cost-effective advantage of data processing with the feature of simplified programming and parallel computing, which has being more and more recognized by the industry. In recent years, the community of high-performance computing is also increasingly using Big data platform for HPC data processing, which has become a powerful means for scientific data analysis gradually. Scientific data traffic generated by the application of HPC tends to have many requirements, including high quality processing, compute in communication link and huge date size, which is called as "high-throughput" traffic. The scale of data processing and the port cost of network need to be considered during the design of datacenter for distributed computer system. The most widely used model for computing and communication in distributed system is MapReduce. The traditional traffic generation method for "Black-Box" is coarse granularity with poor scalability. Therefore, this paper presents a methodology for MapReduce traffic modeling and generation based on the idea of "event-driven". The accuracy evaluation, which compared our methodology with the real application traffic, indicates that the traffic generated by our method can accurately reflect the characteristics of the network traffic generated by MapReduce in distributed computing system. Our performance simulation analysis and bottleneck analysis of four major data center networks, which is conducted by using the characteristic flow in network simulator, shows that the difference of network performance between the one loaded with MapReduce traffic and the one loaded with uniform random traffic is more than 30%, indicating that characteristic traffic could more obviously reveal the issues of network congestion and bottleneck. The results of our simulation, related to the bottleneck of network performance, topology scalability and network cost-effectiveness, provide a new way for large-scale data center network selection and network performance optimization.

Keywords distributed system; MapReduce; data center network; event-driven; large-scale network simulation

1 引 言

大数据平台具有简化编程以及分布式并行化处理等特点,其数据处理的高性价比优势逐渐被越来也多的行业所认可.近年来高性能计算社区也越来越多地使用大数据应用来进行高性能计算的数据处理[1],利用大数据平台对高性能计算数据进行分析处理已经逐渐成为了科学数据分析的强有力手段[2].高性能应用所产生的科学数据往往具有高质量处理需求、通信与处理并行执行、网络总数据负载量大的特点,因此被称为"高通量"数据.使用分布式系统处理"高通量"数据,需要考虑数据处理总量规模与网络建设成本的平衡关系[3].

现有大部分数据中心的网络结构相对稳定,从表面上看网络结构可能无需调整和优化.但是数据中心的建设并不是一成不变的,数据中心承载的业务以及对计算的需求会随着时间的推移发生变化.在进行数据中心扩容建设和下一代数据中心设计时,就需要结合其所承载的业务特征的网络流量进行有针对性地网络结构设计.

目前网络模拟设计所使用的网络流量,往往属

于人造流量，以“黑盒”流量建模方法为主，已经有较为充分的研究. 文献[4]通过函数拟合方法，提出“ECHO”流量模型. 文献[5]设计了“TIVC”流量模型实现对网络流量的预测并指导任务的部署. 传统的以“黑盒”方式进行流量建模，对采集到的流量不加区分地进行函数拟合，没有分析流量在分布式系统内的成因. 这种传统流量建模方法所产生的网络流量往往不具有真实计算业务特征；在不同线程数量规模下表现出可扩展性差和适配性弱的劣势；基于真实流量采样所产生的流量，对数据中心内节点任务的部署具有相关性，不利于方法的移植和迁移.

目前网络流量建模方法的研究，还没有像本文研究中一样，基于对应用业务特性的分析，完成对业务“事件”的建模. 同时也没有如本文中基于“事件”间相互制约关系而开展的“事件驱动”流量建模方法. 本文研究意义体现在以下两点：(1) 是对 MapReduce 业务执行特性与流量产生模式的新颖探索；(2) 通过对目前主流数据中心网络拓扑结构进行仿真评测，得出网络结构对系统性能影响的结论.

MapReduce 这种计算模式目前被广泛应用于数据挖掘、人工智能、机器学习、搜索引擎索引编制、推荐算法等方面，已经逐渐成为图计算的代表性模式，在工业界应用比较广泛. 面向这种计算模式的流量建模研究，对目前和未来的数据中心设计和优化工作，具有较高的借鉴和参考价值.

本文最大贡献是：借鉴“白盒”测试理论，通过事件驱动的流量建模和实现方法，产生能够体现 MapReduce 计算模式中业务特征的网络流量，如图 1 中模型建模的核心方法所示.

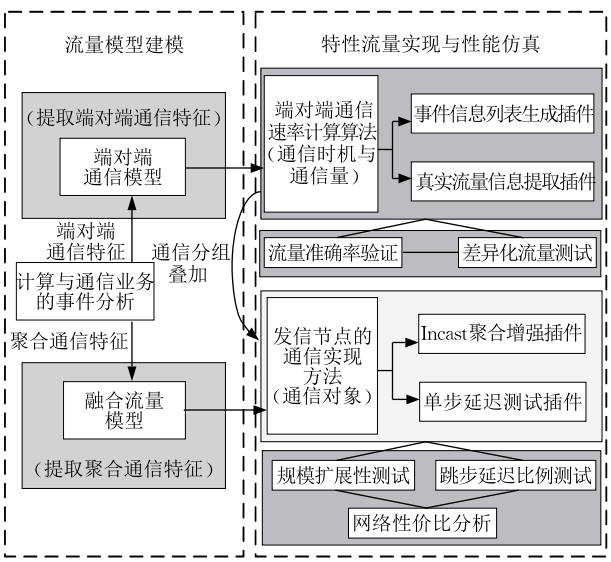


图 1　基于事件驱动的流量研究流程

目前相关研究领域对流量产生方法已有很多工作，但是往往受限于应用自身执行流程的复杂度和流量统计在部署实现时的复杂性，都会面临多方面问题，本文的流量建模也针对这些问题进行了网络流量的建模设计. 上述问题和本文流量建模的重点集中在以下三部分：

(1) 可扩展性. 目前业界方法仅能针对固定任务线程规模(Task 数量固定). 如果应用运行不同规模的线程数量时，需要重新调整和设计流量产生方法.

(2) 正确性. 受限于应用业务执行流程的复杂性，单纯通过流量抓取和统计的方法，很难正确地表现由业务执行特征而导致的流量特征.

(3) 部署相关性. 目前业界方法是针对数据中心内任务部署条件都固定的情况，不能适应任务部署方式改变的情况.

2　特征事件抽取与流量建模

本节首先基于 MapReduce 计算框架的实现原理，针对本研究在流量建模方面所面临的三个问题，对影响网络流量的业务工作原理进行分析. 针对可扩展性建模设计需求，结合业务间相互牵制与相互作用特性，对体现网络流量特性的“事件”进行定义和建模. 为保证流量建模的正确性，通过对通信事件横向分析和纵向统计的方式，对单节点流量进行建模. 最后面向建模部署无关性，对融合流量建模方法进行阐述.

2.1　MapReduce 计算模式特征分析

(1) 面向建模正确性问题. 为了保证流量建模能够正确地反映应用的业务特征，需要针对 MapReduce 业务流程的各个阶段进行分析. 对 Hadoop 类应用建模的核心工作就是对 MapReduce 计算模式的建模，图 2 就是根据 Hadoop 源码对 MapReduce 过程的解构，主要分为 Input、Map、Shuffle、Reduce 和 Output 五个阶段.

① Input 阶段. 输入数据按配置的 Block 大小切分成多个 Block，每个 Block 按配置存储多个复本，Hadoop 尽可能保证不同复本存储在不同结点上.

② Map 阶段. 每个 Mapper 子任务读取一个 Split. 每个 Split 包含一个或多个 Block，作为一个逻辑单元. Split 被分割成记录，框架对每条记录调用 Map 函数. Map 的输出将分别切割成多个 Spill. Spill 中的每条记录经过 Reducer 指派、排序分组和预 Reduce 处理，最终被存储到文件. 如果一台 Mapper

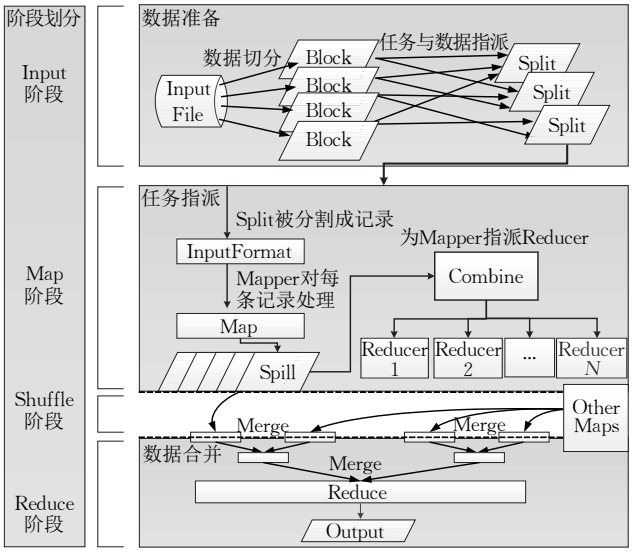


图 2 MapReduce 业务特征分析

机上对某个 Reducer 产生了多个上述处理所得的 Spill 文件,则进行合并,合并时同样执行排序分组和 Combine 流程.

③ Shuffle 阶段. 每个 Mapper 产生的 Spill 文件再次被分派给每个 Reducer. 每个 Reducer 从每个 Mapper 接收给它的数据,如果能在内存中合并就在内存中合并,否则接收后先存储,等全部完成后再来合并. 最终为每个 Reducer 准备好一个待处理的文件.

④ Reduce 与 Output 阶段. 每个 Reducer 的输入文件先同样执行排序、分组和 Combine 流程,然后根据 Reduce 函数得到最终结果,最终结果写到 HDFS 中.

各阶段通信特征分析:Map Task 运行阶段主要分为五个阶段,即 Read、Map、Collect、Spill 以及 Combine. 在 Map 执行的五个阶段中的 Read 阶段是通过 InputSplit 从 HDFS 中解析并读取多个 key/value 对,如果该数据是存储到本地节点物理存储介质上的,则不会产生额外的网络通信流量;反之则会受到远程数据调用影响,产生网络流量. 而这部分流量是否产生将跟分布式系统中 HDFS 缓存数据的备份充分性相关. 其他四个阶段仅使用本地节点的 CPU 和缓存资源,数据通信均在节点内部,且均不会产生网络流量. Reduce Task 整体计算流程包括五个阶段,即 Shuffle、Merge、Sort、Reduce 以及 Write. 其中产生网络流量的阶段发生在 Shuffle 阶段上,该阶段由运行 Reduce Task 的节点发起,从同任务组内运行 Map Task 的节点进行远程数据片拷贝.

(2) 面向建模可扩展性问题. 当应用运行不同规模的线程数量时,会改变流量发生的通信对象,由此影响流量通信特征. 而在 MapReduce 这种计算框架中,线程的分配与数据处理任务"Task"分配紧密相关. 因此需要对 MapReduce 计算框架的任务分配工作原理进行分析. 在基于 MapReduce 计算模型的分布式系统中,每一次对数据的独立处理过程被称为"Job". 大规模分布式系统在处理 Job 的流程中:一方面将整体的数据处理任务进行任务分割,另一方面会将需要处理的数据也进行分割,同时分布式地存储到集群中各个任务的各个节点上. 通过任务分割,任务单个 Job 处理被划分为了多个"Group"任务组,每个 Group 组对应了需要处理的数据和计算任务;结合 MapReduce 计算模型,每个 Group 组将会映射为 Map Task 和 Reduce Task 两种类型的"Task". 任务分割的原则也是遵循数据处理规模的可扩展性,如图 3 所示,即根据需要处理的数据总量进行任务组的分割. 数据处理的量越多,被分割得出的 Task 个数也就越多. Task 占用每个物理计算节点的物理资源主要包括内存区块和 CPU core;而在 Hadoop 系统中运行 MapReduce 计算任务前,JobTracker 会将任务分割并将其调度到合适的物理节点上运行,如图 3 所示.

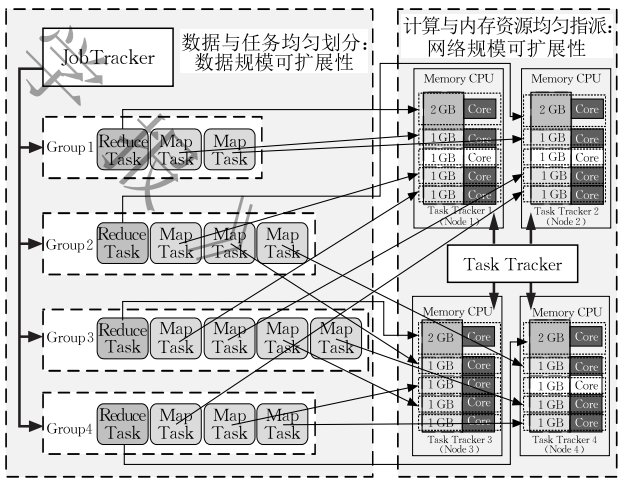


图 3 基于可扩展性的 Task 逻辑划分与物理指派

任务线程个数对网络流量的影响:应用启动线程个数将直接影响 Map 和 Reduce 两类 Task 的数量. Map Task 的个数与数据规模和通过在应用中指定 Split 的大小有关. 在 Input 和 Map 阶段,在 Split 一定的情况下,Map 的个数越多,说明数据规模越大,占用的计算资源越多. 因此,随着 Map 个数的增加,Input 和 Map 阶段的时间会增加. 同时,在数据规模一定的情况下,Map 的个数越多,说明 Split 越小,每个 Map Task 产生的结果会相应的减少,导

致 Shuffle 阶段需要向 Reduce Task 传输的数据量减少. 因此，随着 Map 个数的增加，Shuffle 阶段的时间会相应的减少. Reduce Task 的个数和用户在配置中指定的 partition 个数有关. Reduce 的个数越多，则在 Shuffle 阶段产生的数据流越多，导致 Shuffle 阶段的时间和 Reduce 的个数成正比. 在 Reduce 阶段需要进行数据的计算，Reduce 个数的增加导致了程序的并行程度的增加，因此 Reduce 阶段的时间随着 Reduce 个数的增加而减少.

(3) 面向建模部署相关性问题. 在 MapReduce 计算框架内，节点部署和任务调度策略也会影响网络流量特征的变化，下面就此进行分析. Hadoop 这种分布式数据处理框架，在进行 Task 到物理节点部署时都会遵循多方面的优化策略，如任务全局负载均衡(Task Balance)、本地数据优先指派(Local Prefer)等. 在现实的大规模分布式系统中，往往无法像图 3 中的理想情况一样，做到完美的任务分配. 实际系统中可能会出现多个 Map Task 被分配到了同一个 CPU 上进行处理，因此会出现同一 Job 的不同 Map Task 在 CPU 上启动和结束时间不同的现象. Reduce Task 需要等待同一个任务组内的各个 Map Task 进行完毕后，才能进行执行，因此任务组内的两种 Task 间，也将存在执行先后顺序上的依赖关系.

2.2　特征事件抽取与正确性建模

本文基于对 MapReduce 通信特征的提取，结合 Task 逻辑划分与物理指派，将计算流程进行抽象，完成对计算模式中“事件”的建模. 这样的建模方法既将 MapReduce 业务执行流程的特点通过“事件”进行提炼，同时建模的方法兼顾流量特征的正确性问题.

(1) “事件”的定义. 下面对影响网络流量特征的业务事件进行定义，并对网络流量产生的影响进行分析. 本文所提出的基于事件驱动的网络流量产生方法，借鉴“白盒”测试理论，通过对 MapReduce 类大数据平台的行为分析，得出不同运行阶段计算与通信业务之间的依赖关系. “事件驱动”中的“事件”指：在分布系统中具有不同的网络通信特征，并且实际执行了 MapReduce 计算业务的子过程；所有“事件”按时间顺序的组合将构成 MapReduce 执行的全过程. 网络通信的数据流量由各个数据收发节点间的“单节点”通信流量组成，而事件间的相互依赖关系将直接影响“通信对”的流量特征. 通过分布式系统中所有“通信对”流量的叠加，就能得出体现应用通信特性的网络流量. 传统建模方法是将大数据平台视为整体“黑盒”，进行特征提取与流量生成. 相比于传统方法，本研究中所提出的流量采集和建模方法能够更加准确和细粒度地表现出大数据平台的通信特征.

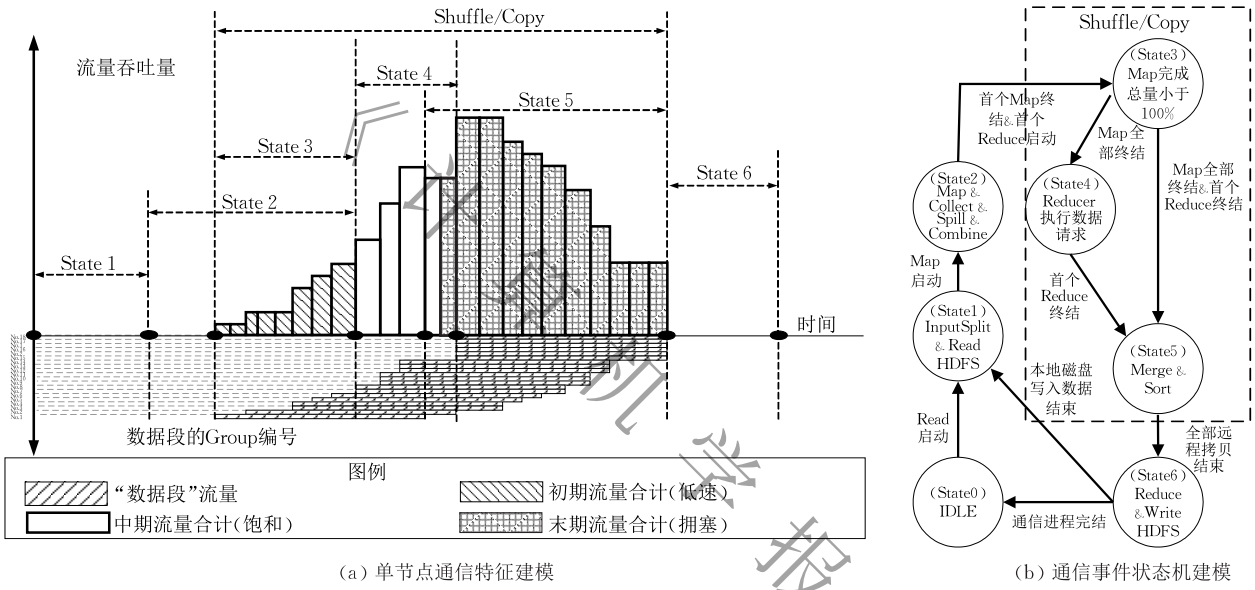
(2) “事件”的抽取与建模. 通过对业务事件与业务处理流程的分析，可以看出“拷贝”过程的执行是以“事件”的完成作为前提条件，即计算与通信业务执行流程中的具体步骤，下面将对远程数据拷贝前所执行的事件(执行步骤)进行特征抽取，并完成对事件的建模. 由于 MapReduce 作业包含 Map 和 Reduce 两类 Task，“事件”就要从这两类 Task 中抽取. Map Task 计算任务在执行期间，JobTracker 将履行作业监控功能，监控 Map 和 Reduce 两类 Task 执行完成的情况. JobTracker 监控 Map 进程的数量有限，该数值与 Shuffle 执行方法中的参数“parallelcopies”相关，而在 Map Task 的全集中，具体哪些 Map Task 被检测到，存在选择随机性. 在 JobTracker 所监控的 Map Task 都执行完毕后，JobTracker 会通过 RPC 向该任务组内运行 Reduce Task 的计算节点发出这几个 Map Task 的数据完备通知，同时将所完成 Map Task 的主机号更新到 Reduce 节点的“pendingHost”集合中. 而运行 Reduce Task 的计算节点将会收到多个数据完备通知，每个节点将维护一个目前 Map Task 完备的集合“pendingHost”，即在该集合中的 Map Task 均已运行完毕后，可以响应 Reduce 节点对该 Map 节点进行远程数据拷贝请求. Reduce Task 将随机地在“pendingHost”集合中选取固定个数的 Map Task，进行并行数据请求. Reduce 节点的数据请求并行度也由 Shuffle 执行方法中的参数“parallelcopies”决定. 而运行 Map Task 的计算节点在远程数据拷贝后，将根据自己计算节点的 CPU core 数量，启动最多两倍数量的线程来并行地处理拷贝请求. 当接收到的请求数量大于所启动的线程时，将随机选择线程数量的请求进行响应，而没有被选中的线程将当线程空闲后再进行处理. 远程数据请求在被响应后，将出现 server-client 之间的固定数据量的单节点通信.

2.3　单节点流量与可扩展性建模

随任务线程数量的扩展，网络流量基本单位(单节点通信流量)的通信特征将随之变化. 为满足流量建模的可扩展性，本小节将结合上文中对事件间相互牵制与相互作用的事件驱动特点，通过对通信事件横向分析和纵向统计，完成单节点通信流量建模与整合.

(1) 单节点流量的事件驱动特征. 本文通过事件

模拟单元对事件的通信行为进行模拟仿真，如图 4 所示. 建模的过程就是根据 Map-Map 和 Map-Copy 之间的事件依赖关系，模拟 Map Task 和 Shuffle 事件的执行顺序. 具体方法如下：将具有通信流量产生的 Shuffle 事件，按照所处任务组 Group 序号顺序，根据启动事件先后顺序进行排列，可以生成分布式系统中通信对象间，单节点数据流的微观通信模型，如图 4(a)所示. 该图表示 Hadoop 系统中，基于 server-client 模式的单节点通信，通信事件随时间变化的情况，即作为应用流量特征信息传递给“事件映射单元”. 事件模拟单元产生流量特征信息的方法如下：首先需要产生 Map Task 事件执行的顺序表. 由于计算节点的 CPU core 有限，无法同时启动所有计算线程. 能够并行运行的 Map Task 数量总是小于或等于该计算节点启动的线程数量，根据该原则即可得到图 6 中“Map 事件执行序列”. “Map 事件执行序列”是指在运行 Map 任务的计算节点中，处理 Map Task 的执行顺序. 而影响通信流量产生 Shuffle 事件的 Map Task，指代归属于同一 Group 任务组内的 Map Task，而这些 Task 也是 JobTracker 同一监控的对象. 由于 JobTracker 能够检测的 Map Task 数量有限，无法同时对所有 Map Task 线程进行检测. Map Task 在执行完毕后，需要等待 JobTracker 检测到该进程，才能继续启动后续通信流程.



(a) 单节点通信特征建模　　(b) 通信事件状态机建模

图 4　基于事件的“单节点”流量模型建模

(2)任务线程对单节点通信的影响. 下面针对事件模拟单元中，任务线程产生行为单节点通信的模拟内容进行说明. Map 节点将会收到的来自各个 Reduce 线程发送的数据请求，这些任务线程间的数据请求将统一存入“Map 数据请求应答池”中. Map 节点将启动固定个数的数据应答线程. 在 Hadoop 源码中，此参数的系统默认值是处理器个数的两倍. 数据应答线程将在请求应答池中，随机选择应答任务进行应答，这些线程将并行地处理来自不同 Reduce 节点的数据请求. 一个 Map 计算节点将会运行属于多个任务组的 Map Task，如图 4(a)纵轴所示. 而每一个任务组都会有至少一个 Reduce Task 运行在分布式系统的一个计算节点上. 因此，Map 线程对单个 Reduce 线程生成流量并执行通信的过程，就产生了“单节点”通信流量. 图 4(a)体现了组成一个 Map 节点进行单节点通信的各个数据段随时间的分布情况.

(3)数据段叠加与建模可扩展性. 流量模型中，每组成对的 Map 与 Reduce 任务线程，将生成“数据段”流量. “数据段”流量经过叠加，将组成“单节点”流量. 而这种对单节点流量的建模方法，可根据应用线程数扩展规模的不同，对组成单节点流量的数据段进行合理配置，由此满足对建模可扩展性的需求.

单节点流量整合方法如下：当 Map 节点上的第一个数据请求被应答时，将启动远程数据传输，即图 4(a)中的“首个 Map 终结&首个 Reduce 启动”事件. 而该请求导致的数据传输时间和数据速率需要通过图 6 中的“事件信息提取实现单元”，结合对小规模下的分布式系统进行数据捕获而得到. 该方法实现途径较多，不具有唯一性，因此不在本文所提出的流量产生方法内说明. 在图 4(a)所表示的分布式系统中由单一进程所导致的单节点通信，每一个横向左下斜线方块表示一条由远程进程数据请求所导致的数据传输，称为“数据段流量”，即发生通信的

事件.为了方便流量的产生,本文所提出的流量产生方法将每个数据段通信期间的流量整形为“均等传输”流量,即数据段通信期间保持相同数据传输速率进行通信,且各节点的各个数据段传输时间和通信数据总量相同.每一对 Map 节点与 Reduce 节点间的“单节点”通信流量将由多个数据段组成,如图 4(a)中各个数据段将会在时间维度上进行重叠.数据块在时间维度上重叠得越严重,将导致总数据传输速率的上升.根据本节前文中各个事件的产生方法,可以得到分布式系统内,组成任意两节点进行通信的各个“数据段”的起始和终止时间.若分布式系统中拥有 n 个进程,当该算法产生系统中各个节点通信流量的所有数据段,其最大时间复杂度将是 $O(n^2)$.

2.4　融合流量与部署无关性建模

任务部署方式发生改变不会影响单节点流量的特征,但是会直接影响基于单节点通信叠加的融合流量.针对于此,本文通过对单节点通信叠加后,所形成“融合流量”的建模,保证了对网络中整体流量对各种网络部署情况的适应性,满足建模的部署无关性需求.下面对基于单节点通信叠加的“融合流量”的建模方法进行说明.

(1)通过任务分组表示部署特征.在前文中所提到的 Group 任务组中,每一个 Group 包含了分布在系统中各个计算节点的 Map Task 和 Reduce Task.而不同的部署策略对融合流量的影响将体现在 Group 内计算节点成员的配置方式上.运行 Reduce Task 的计算节点将作为收信节点,而运行 Map Task 的节点将作为发信节点.因此在分布式系统中,单独一个 Group 内发生的通信,可视为如图 5 所示的“多对一”集合通信.但是根据 Group 的 Task 分布仅能以接收节点的视角,明确哪些节点会对其执行发信行为,如图 5 左侧的虚线框所示内容.为了实现流量的发信行为,需要以发信节点的视角,明确各个发信节点都需要对哪些节点进行发信,如图 5 右侧的实线框所示部分.

(2)包含部署描述特征的融合流量.根据上文中对任务组部署特征信息的映射,可见“多对一”集合通信的进程间通信关系,是融合流量建模中代表任务部署特征的重要部分.在流量建模时,可根据不同的部署情况,通过配置进程间通信关系,来改变网络流量部署相关性特征.下面将针对进程间通信关系的生成方法进行描述.Map 和 Reduce 两类任务进程双方间的通信关系可以理解为:从各个节点的“收信对象列表”向各节点的“发信对象列表”映射的过程,如图 5 所示过程.该映射方法通过在遍历收信对象的过程中,对具有相同发信节点的收信对象表项进行分组.遍历后所得的各个分组,即为各个节点发信对象列表.分布式系统的各个发信节点,将通过“等概率轮询”的访问方式,对发信对象列表内的各个节点进行发信.当每个 Group 内只有一个节点运行 Reduce 节点时,发信节点对各个收信节点的轮询发信过程,可以视为该发信节点对其所参与的各个 Group 的轮询.

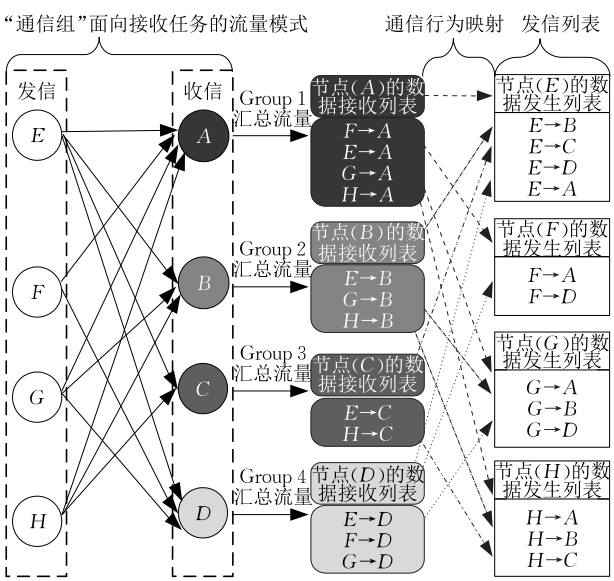


图 5　发信节点数据通信映射算法示意

3　面向大规模网络模拟的流量实现

为了在网络模拟器中实现特性流量并以此进行网络评测,需要为流量产生方法提供“单节点流量实现单元”和“融合流量实现单元”,如图 6 所示,本节将对完成流量实现的这两个单元进行阐述.

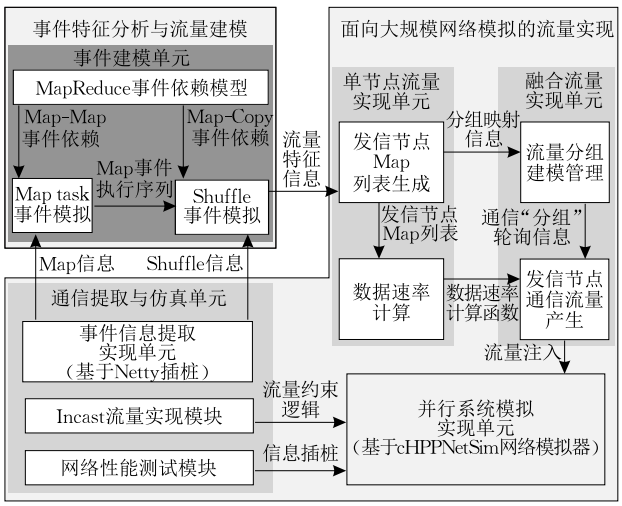


图 6　流量产生方法及实现方法的结构示意

3.1 单节点流量实现方法

(1)面向可扩展性的流量特征信息存储方法.流量实现的本质,就是确定在任意时刻流量数据传输速率的大小.在计算单节点流量速率前,首先需要对图 6 中的流量特征信息进行存储;再通过"发信节点 Map 列表生成"模块为数据速率计算提供计算条件.下面将对流量特征信息的存储方法进行说明.为了计算分布式系统中,任意两计算节点间发生网络通信时,在任何时间点的通信速率,首先需要将所有"数据段"的相关信息进行存储,如图 7 所示,这些数据段组成全系统通信流量.在图 7 所示信息列表的数据结构包括:运行 Map Task 节点的序号作为 Node ID;以及该节点作为发信节点时,所有需要发出数据段信息的结构体数组索引地址.数据段信息结构体的成员变量如图 7 所示,包括:数据段隶属任务组序号、通信目的节点序号、开始时间、结束时间、数据段的独立数据传输速率.通过前文对数据段相关信息的遍历方法,可以完成该信息列表对各个信息位的填充,完成该列表的生成工作.这种对流量特征信息的存储方法,可根据应用任务线程不同的规模情况,对流量特征信息进行组织,便于数据速率的计算.

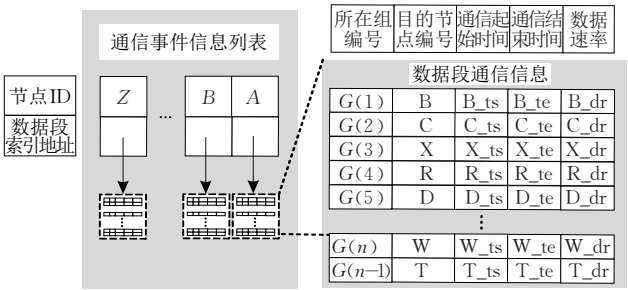


图 7 发信节点通信事件信息列表

(2)面向正确性的数据速率计算原理.为了使建模流量能够正确地表现计算业务特征,需明确流量计算速率的计算原理与上文中对流量建模的关系,下面对单节点通信模型的横向组合进行说明.如图 4(a)的微观分析方式,可以明确组成单节点通信流量的各个子流量(数据段),在时间维度上的分布情况.为了计算分布式系统中任意时刻的单节点通信数据传输速率,需要将通信流量内的数据段进行纵向叠加,以宏观的视角分析通信事件各个阶段内的通信流量特征,如图 4(a)左下斜线所示数据段流量所示.图 4(b)是图 4(a)对各个横向数据段进行纵向叠加,得出单节点通信事件的状态图,即状态机建模结果.结合图 4(a)的流量叠加汇总结果和图 4(b)的状态转移情况来看,在 Input 和单纯运行 Map Task 的阶段,通信量很少,几乎可以忽略.在图 4(b)中具有通信流量发生的阶段是 Shuffle/Copy 阶段,在通信发生的最早时间点并不是所有 Map Task 都完成的时刻,而是发生在图 4(a)中"首个 Map 终结"时间点出现的时间.

(3)面向可扩展性的数据速率计算方法.针对图 6 中"数据速率计算"方法,本文提出一种单节点通信的数据速率计算算法.该算法可根据应用不同的线程规模,结合数据段的流量特征信息,得出流量传输速率.速率计算的功能是在给定发信时间的"Tsend"条件下,计算从以图 7 中"Node ID"为发送数据节点,到以"Root Sequence"接收数据节点之间,两节点间的任意时刻单节点通信数据传输速率"Data_rate_total".具体操作步骤如下:

步骤 1. 根据发信节点通信事件信息列表(符合当前应用任务线程规模),找到图 7 中"Node ID"节点的数据段信息表(数据段表)地址,并索引到该节点所需要发出数据的全部数据段.

步骤 2. 对发信节点的数据段表,遍历 Root number 与目标收信节点相同的表项.对所选出的表项进行发包时间条件判断,判断该表项的通信启动时间(Time_start)和结束时间(Time_end)是否满足:Time_start<Tsend<Time_end.

步骤 3. 如果步骤 2 判断结果为真,则将该表项的"Data_rate"数据,累加到初值为 0 的单点时刻数据传输速率"Data_rate_total".判断当前是否完成了发信节点的遍历,如果未完成遍历则返回步骤 2,对下一条数据段表项进行判断累加操作.

步骤 4. 如果步骤 3 完成了对所求发信节点的各个数据段表项的遍历操作,则将经过累加计算的"Data_rate_total"数值,即作为:所求的发信节点对目的节点在 Tsend 时刻发信的数据传输速率.

3.2 融合流量实现方法

(1)发信节点流量产生实现思路.下面针对融合流量的具体实现方法,按照图 6 中实现该功能的各模块,按其分工分别进行说明.针对图 6 中"融合流量单元"的实现方法是为实现图 5 流量,利用已生成的单节点通信流量(图 4 建模流量),面向发信节点实施对各个目的节点通信(如图 5),生成分布式系统的全局性"融合流量".随发信动作的执行,发信节点将在不同时刻对收信节点进行发信,结合前文

所述的单节点通信数据速率计算方法，可以实现各个 Group 通信的发信行为. 融合流量的产生包含“发信节点通信流量产生”和“流量分组建模与管理”两个方法. 如图 6 所示，发信节点负载实施发包行为. 相应的流量产生方法，需要结合单节点流量单元所提供的数据速率计算函数信息，产生符合运行 MapReduce 类应用特征的数据流量. 该方法需要从“数据速率计算”方法获得“数据速率计算函数”. 基于此函数计算任意两通信节点在不同时刻的通信速率. 图 6 中流量实现单元的实现方法需要结合具体仿真软件环境进行设计. 根据文献[6-7]中对数据包长度的统计结论，发信节点所发出数据包的包长保持在 100 Byte～400 Byte 之间，即生成的仿真流量是由短小的数据包组成“低负载”流量.

(2) 面向部署无关性的流量分组管理. 由于流量分组已经包含了任务的部署特征，因此对流量分组的管理和使用，具有部署无关性. 如图 6 所示，流量分组体现了分布式系统中通信行为的规则. 面向“多对一发信分组模型”的建模方法和分组轮询选择管理方法，实现形如“多对一发信分组”的聚合通信行为策略. 该策略需要从“单节点”流量单元的“Map 列表生成”方法获得“流量模型映射分布信息”，包括“Reducer 部署节点信息”和“对应各个 Reducer 的发信节点信息”；这部分信息将帮助本方法完成所述“分组”的通信模型建模. 所述“分组轮询选择”管理方法是指发信节点轮询式地对其所服务的分组进行发信的行为，得到“分组轮询选择结果”. 该结果为当前发信节点对其所在的多个 Group 分组的轮询选择结果，即选择其中一个分组的 Reduce 收信节点进行发信.

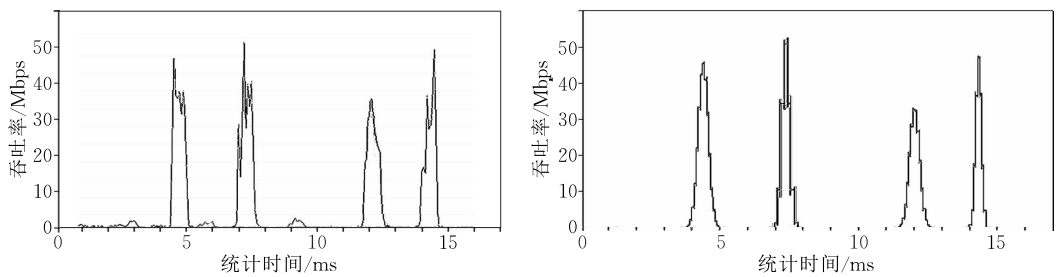
4　流量模型验证与网络性能评测

针对本文提出的基于事件模拟的 MapReduce 类特性流量产生方法，为了说明方法的可行性，仅仅从微观的角度对 MapReduce 的计算特征与模拟行为进行分析是不完整的. 还要从宏观角度，通过比对真实流量和模拟流量的差异，体现出模拟流量对数据处理特征的匹配度，表现流量是否真正能够反应出分布式系统的流量特征.

目前 RDMA 被广泛关注，所以针对未来的需要，本节所进行的仿真实验均是基于 PFC 的无丢包网络进行评测和验证. 本节中所进行的网络[8-9]性能仿真分别基于胖树网络 Fattree、缩减型胖树网络 TorFT、HyperX 网络和 CLOS 网络，分别负载 MapReduce 通信特征的数据流量，旨在反应四种网络在承载特性流量时，各种网络通信的差异化表现以及其成因. 本文网络模拟基于文献[10]由中国科学院计算技术研究所设计的“cHPPNetSim”多功能可配置并行网络模拟器进行仿真(configurable HPP network simulator). 该模拟平台主要功能是对大规模并行网络进行细粒度的模拟，模拟结果可以得到网络整体性能、局部性能，并获取每个网络部件运行状态.

4.1　流量模拟准确率评测

通过前文中对 MapReduce 计算模型中各个事件以及其相关影响关系的分析，已经从微观角度进行了较为全面的分析. 本节通过 tcpdump 与 wireshark 相结合的方法进行流量分析与过滤，使用 tcpdump 针对特定网卡监视网络数据包的传递方向，并输出数据包的报文头信息. 再通过 wireshark 对 tcpdump 抓取的包进行进一步的筛选，通过将报文头中的目的 IP、源 IP 作为过滤条件，提取并计算单位时间内分布式系统单节点的数据传输总量，即为单节点数据传输速率，如图 8(a)所示. 结合 cHPPNetSim 模拟器中对数据速率计算函数，可以计算得到本文提出的通信仿真流量速率随时间的分布情况，如图 8(b)所示. 通过图 8，可以看到点对点通信流量在时间间隔与速率轻度方面都比较接近.



(a) 数据中心的实时流量　　(b) 事件驱动的仿真流量

图 8　点对点通信流量模拟验证测试

为了以量化角度分析流量产生方法的准确性，本研究对含5、10、15个计算节点的实际分布式系统进行流量采集，如表1中Net Scale.通过与本文流量产生方法进行数据速率做差值，得出误差值占真实流量的比例，取其反比例可得单时刻流量准确率.通过计算准确率的平均值，可得表1中的数值.用本文方法，可在不同规模网络下进行数据插桩采集，如图6中的“事件信息提取实现单元”.分别对5、10、15规模下网络进行事件信息提取，并分别计算在不同网络规模下的提取方法的准确率，如表1中DAQ (Data Acquisition) Scale.测试分别包括PageRank、K-Means和NutchIndexing三种应用，进行准确率评测，如表1中“APP”.

表1 模拟流量准确率比较表

DAQ Scale	5-DAQ			10-DAQ		15-DAQ
APP \ NNet SScale	5 Node	10 Node	15 Node	10 Node	15 Node	15 Node
PageRank	92.3%	93.9%	91.7%	86.1%	85.5%	86.8%
K-Means	93.3%	87.9%	91.6%	91.1%	85.7%	95.0%
NutchIndexing	90.1%	90.9%	87.7%	91.6%	91.4%	93.2%
(average)	91.9%	90.9%	90.3%	89.6%	87.5%	91.7%

本节所进行的流量模拟准确率评测实验，目的是为了体现本文所提出的特性流量生成方法的特点.在反映流量特征方面的性能上，最直观的评判方法就是将流量产生的方法与实际分布式系统中的流量进行比较.目前受限于实验环境，最大规模可以供作者进行数据流量提取的实验环境，就是所提到的由15个计算节点所搭建的分布式系统.

为了证明本文所提出流量产生方法的特征准确性是否与分布式集群规模相关(即流量产生方法在对不同规模网络的代表性问题)，本小节的流量准确率评测实验中，作者针对5、10、15三种不同节点规模的分布式系统进行关键参数提取，分别得到了三种DAQ Scale特征提取的流量，再分别针对不同的网络规矩产生特性流量.通过与同等规模的真实分布式系统的流量比较，可以从表1中发现：5-DAG产生流量在网络规模为15节点的情况下，对三种应用的比较结果均在90%左右.实验结果证明，针对三种不同规模分布式系统提取特征参数所产生的特性流量，与实际系统流量相比均保持较高的准确率，并没有因为关键参数采样的规模的不同出现比较明显的准确率的偏差.因此本文采用15个节点进行仿真实验验证.

测试结果表明，本文所提出的基于事件的网络流量产生方法，准确率基本保持在90%及以上.针对不同应用，准确率没有明显差异.在方法扩展性方面，通过对小规模网络进行必要信息采集后，应用于更大规模的网络流量的模拟，所得准确率基本与使用大规模网络采集信息的模拟方法持平.因此，说明本文的方法可适用于大规模网络中特性流量的产生，且无需依赖于对大规模网络信息的采集.从图10可以看出，影响流量准确率的主要因素集中在Shuffle通信流量之外的流量.而根据文献[11]影响分布式系统网络性能的流量是Shuffle阶段内的流量，因此本文所提出的流量产生方法能够体现MapReduce计算模式下网络流量的特征.

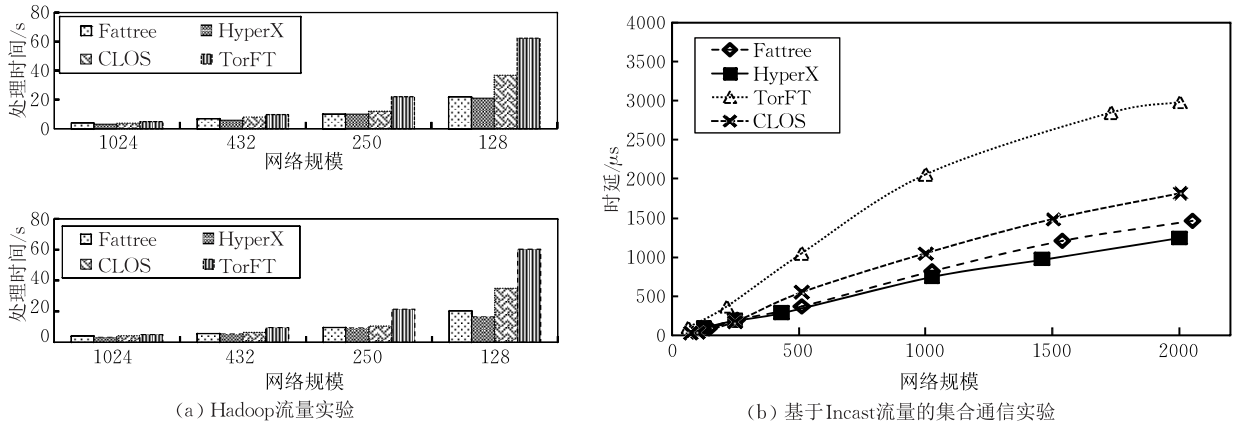
4.2 应用负载完成时间测试

Hadoop平台流量负载的应用完成时间测试，如图9(a)所示.应用数据总量保持不变条件下，网络传输的总数据量为100 GB，通信流量将被均匀分散到网络中各个节点上.从四种网络的独立通信能力角度，可以看出：Fattree与HyperX网络只需更少的时间就可完成100 GByte的数据传输，即性能要优于CLOS网络.而TorFT网络需要的传输时间最长，网络性能最差.

Incast应用通信特性测试如图9(b)所示，设置Incast发信端Group轮询切换策略，单次完成对一个Group的Reduce节点发信，需要连续发送10个256 Byte的数据包.与图9(a)的测试不同，在图9(b)的应用完成时间测试中，随网络规模增长网络传输的总数据量正比增长，且网络中流量分布仍保持均匀分布.从图9(b)仿真数据可以看到，在2000节点规模下，TorFT耗时最长，相对TorFT：CLOS耗时减少30%，Fattree耗时减少50%，HyperX耗时减少58%.随网络规模增长，四种网络特性差距逐渐增大，但是，除了TorFT网络外，CLOS、Fattree和HyperX三种应用完成时间虽然在增长，但增长趋势逐渐趋于平缓，即增长速度逐渐降低.在HyperX网络在“Incast局部热点性通信”的应用流量下，具有性能优势，但随着网络规模增长，优势的增长速度将降低.

本节实验可以从以下3个方面进行分析：(1)同种网络的扩展性分析，即同种网络的应用完成时间随网络规模的不同而出现差异；(2)网络通信总量的影响力分析；(3)延迟敏感与带宽敏感阶段的流量影响力分析.

同种网络的扩展性分析.在图9(a)的实验中，



(a) Hadoop流量实验　　(b) 基于Incast流量的集合通信实验

图 9　应用完成时间性能测试

同种网络随规模扩展，应用完成时间随之下降. 而在图 9(b)的实验中，随规模扩展，应用完成时间随之上升，且上升的速度逐渐下降. 造成两类实验随规模扩展而出现的结果变化相反的原因是：两类实验分别在保证网络通信总量不变条件和随网络规模正比增加网络规模条件下进行的.

网络通信总量的影响力分析. 在图 9(a)实验中，测试结果反映出通信性能随同种网络拓扑规模上升，应用完成时间成“逆对数特性”下降. 随着网络规模增大，在图 9(a)实验的四种网络的通信性能逐渐趋同. 该现象出现的原因是由于网络流量被均匀分布在网络中，网络通信的发生局部性通信依然不明显. 在像图 9(a)中这种总数据传输总量保持不变的条件下，导致的局部拥塞性的局部通信流量随流量的均匀分布而减少. 因此，网络中流量也随着网络规模增加而变得越来越稀少，网络拥塞也变得越来越不明显. 而图 9(b)的实验随网络规模增长网络传输的总数据量正比增长，因此随规模扩展不同网络的通信性能出现分化，性能差异逐渐明显.

延迟敏感与带宽敏感阶段的流量影响力分析. 流量特征的变化不能简单根据 Hadoop 分布式系统所运行的应用而区分，而是在 MapReduce 执行的不同阶段产生不同类型的流量；不同运行阶段，延迟和带宽的影响程度都会不同. 业务在产生管理和资源调度任务时，会产生数据量较小的数据流量，而这部分运行阶段属于“延迟敏感性运行阶段”. 但该部分所产生的网络流量较小，不是引起网络拥塞的主要成因，这部分流量对网络结构选型与优化影响程度比较低. 从图 9(a)上看出图中同时包含“带宽敏感阶段”，即 Shuffle 阶段流量和“延迟敏感运行阶段”，即 Map 阶段流量的测试结果，与图 9(a)下图仅有“带宽敏感运行阶段”的测试结果相近. 该结果反映出由“延迟敏感性运行阶段”产生的 Map 阶段流量，对由网络拥塞导致的应用整体运行性能的影响并不明显.

4.3　网络规模的可扩展性测试

在符合 MapReduce 通信特征的数据流量下，在多种网络规模情况下，通过等规模不同网络的可扩展性延迟与带宽性能实验，体现各个网络随规模扩展后，负载特性流量时通信能力的变化趋势.

Incast 发信端 Group 轮询切换策略，单次完成对一个 Group 的 Reduce 节点发信，需要连续发送 10 个 256 Bytes 的数据包. 如图 10 所示，进行在 MapReduce 特性流量下的大规模网络扩展性通信性能测试. 在测试中，应用数据总量随网络规模扩大成正比增长条件下，10 个 Hadoop 应用数量的满负荷流量作为网络注入流量，且不使用 Incast 热点性增强插件条件下进行测试仿真. 从仿真结果可以看到，随网络规模增长，最大平均数据接收带宽逐渐下降. 随扩展性上升，四种网络通信差异化特性依然存在，且随着网络注入比例的逐渐扩大，四种网络在节点的平均接收带宽的差异逐渐固定.

TorFT 在各种规模下，保持较低的数据接收能力，网络拥塞现象明显. 随网络规模增加，CLOS 通信性能下降速度快于 Fattree 和 HyperX. Fattree 网络的节点的接收带宽通信性能保持最高，但随着网络规模扩展其性能逐渐与 HyperX 网络趋近，100 k 规模下两种网络性能最为接近. 可见，在规模超过 100 k 后，相对于 HyperX 网络，Fattree 网络通信性能不再有优势.

通过本节实验可以从以下两个方面进行分析. (1) 等规模各类网络性能差异化分析，即在相同网

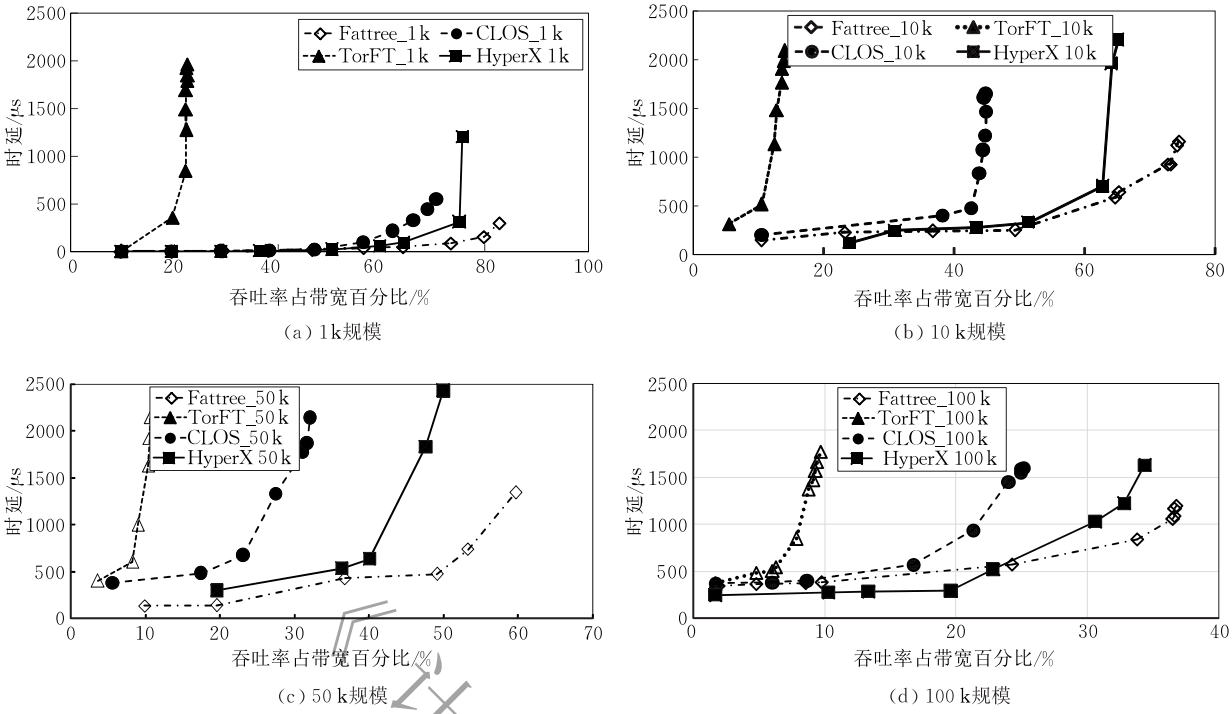


图 10 网络可扩展性通信测试

络规模的前提下不同网络反映的性能不同的原因；(2)同种网络的不等规模性能分析，即各种网络的“延迟-带宽”性能，随网络规模的不同而出现不同程度的性能下降的原因.

等规模各类网络性能差异化分析. 图 10 的实验反映出，性能最好的网络结构是 Fattree 和 HyperX. 前者作为无阻塞网络的代表，流量在网络中负载最为平衡. 后者网络中节点间在横向与纵向方向有较好的连通性，同时网络平均跳步数最低，由此带来了网络通信性能的优势. 而 TorFT 网络叶子节点处的聚合缩减设计，导致网络整体通信性能出现最差的性能表现.

同种网络的不等规模性能分析. 四种网络随规模增长，交换节点的最大接收带宽逐渐下降. 这种现象的原因是随网络规模以及数据传输总量的正比增长，网络中出现局部拥塞的概率也会正比增长，即会更容易发生网络局部拥塞. 而一旦网络中越容易出现局部拥塞，就会更加提前发生拥塞连锁效应，导致网络最大吞吐率降低.

4.4 网络跳步延迟比例测试

任何网络的结构都会存在结构上的短板. 在网络跳步延迟性能测试中，分析各个网络中各级交换机的延迟比重，可一目了然地明确各个网络通信瓶颈的位置. 网络结构设计者可通过如增加网络带宽、网络链路“多轨化”等方法对网络瓶颈位置进行局部优化，由此提高网络整体性能. 结合之前实验中 TorFT 网络性能相较其他网络，出现明显偏低的现象. 下面将对 TorFT 与其他三种网络，实施跳步延迟比例测试.

Fattree 跳步延迟分布如图 11(a)所示，可以看到 Fattree 主要延迟集中在最后三跳交换机上，出现终端拥塞现象. HyperX 跳步延迟分布如图 11(b)所示. 从仿真结果可以看出，Fattree 与 HyperX 网络具有相似的通信特性. TorFT 跳步延迟分布如图 11(c)所示，TorFT 在通信初期拥塞分布比较均匀. 随时间推进，初级交换机拥塞逐渐明显，主要拥塞集中在第二级交换机上. CLOS 网络跳步延迟分布如图 11(d)所示，该网络在拓扑连接上不存在 TOR 缩减. 仿真结果表明，延迟主要集中在第 2、3、5 跳交换机上.

通过对比两种网络的跳步延迟分布，寻找造成网络性能急剧下降的形成原因，由此便于为网络设计者优化网络结构. 通过本节实验可以从以下两个方面进行分析：(1)跳步延迟分布均匀原因分析，即结合网络结构特点与网络跳步均匀分布的现象进行原因分析；(2)延迟分布不均匀原因分析.

跳步延迟分布均匀原因分析. 除初级交换机延迟较低外，Fattree 与 HyperX 网络拥塞分布较为均匀，即网络中后部网络拥塞集中度较高. 这两种网络出现均匀跳步拥塞的分布现象，是因为网络瓶颈位

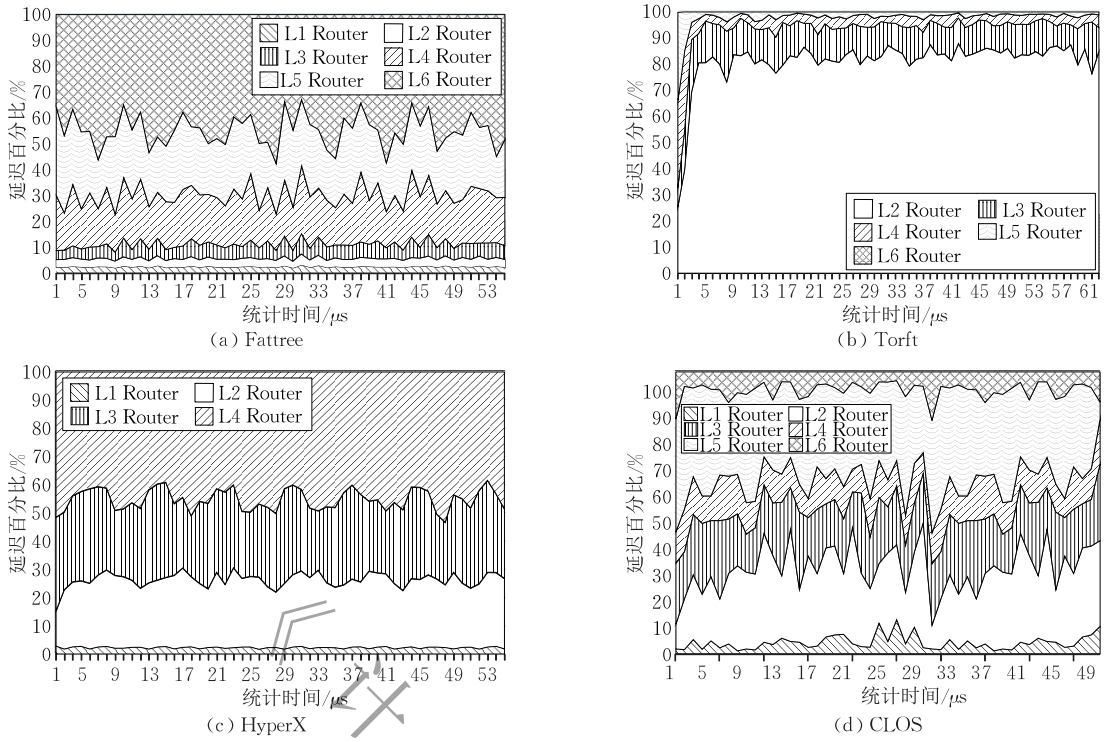


图 11　网络跳步延迟比例分布图

于数据在网络链路的后半部分，即接收数据的计算节点的数据接收能力成为了网络瓶颈. 也正是由于均匀分布的网络拥塞，这两种网络通信性能的优势才得以体现.

延迟分布不均匀原因分析. TorFT 出现延迟分布随时间推移会发生明显变化. 出现这样的现象是因为拥塞集中在网络传输路径的起始阶段. 从缩减胖树的网络结构上分析，由于叶子节点的链路缩减形成网络通信性能瓶颈，导致网络拥塞随流量注入的增加更明显地得以体现. 相较 Fattree 与 HyperX，CLOS 网络拥塞位于网络中部，出现这种现象是因为网络拥塞瓶颈位于网络中部交换节点的通信能力(即这种网络结构的链路交换能力)分布不够均匀.

4.5　“差异化”流量测试与网络性价比分析

对同一种网络结构而言，本文所提出的特性流量与均匀随机流量对网络通信性能将带来不同的影响. 同时，负载应用数量不同的通信特性流量也将造成网络性能的差异化. 因为不同网络在负载特性流量和均匀随机流量上，分别具有不同的通信特征. 本节实验中，四种网络将分别负载不同应用数量的特性流量以及均匀随机流量，由此体现不同网络在运行“差异化”流量下所表现的通信特性.

为体现流量的局部拥塞差异化，将使用“Incast”流量作为 MapReduce 特性流量进行仿真实验. 不同于前文中在相同特征负载流量下对网络的仿真测试，本节测试在 100 GB 为总数据量的同等应用数据总量的条件下，对四种网络拓扑的 150 节点规模，分别进行 Uniform Random(均匀随机流量)、单应用 Incast 流量、多应用 Incast 流量负载下的网络带宽-延迟性能测试，如图 12 所示.

Fattree 测试现象如图 12(a)所示，均匀随机流量，低延迟和高带宽特性优势明显. Incast 流量随 app 数量上升，网络整体填充度上升，接收带宽逐渐靠近均匀随机的情况，且 10app 基本处于饱和状态，最大吞吐率占带宽比重为 84.7%. HyperX 测试现象如图 12(b)所示，均匀随机流量虽具有最佳通信性能，节点接收带宽优势已不存在. 在饱和性 Incast 流量注入下，性能趋近于均匀随机流量性能，网络最大平均数据接收带宽性能优于 Fattree 和 CLOS. Incast 流量最大吞吐率占带宽比重为 90.2%. CLOS 测试现象如图 12(c)所示，均匀随机流量具有最佳通信性能，延迟特性优势存在低于 5%的带宽性能优势. 在饱和性 Incast 流量注入下，网络最大平均数据接收带宽性能低于 Fattree 和 HyperX. Incast 流量最大吞吐率占带宽比重可达 81.3%. TorFT 测试现象如图 12(d)所示，均匀随机流量与 Incast 流量各种 app 注入量性能相似. Incast 流量随 app 数量上升，因网络拥塞限制，网络整体填充度不再上

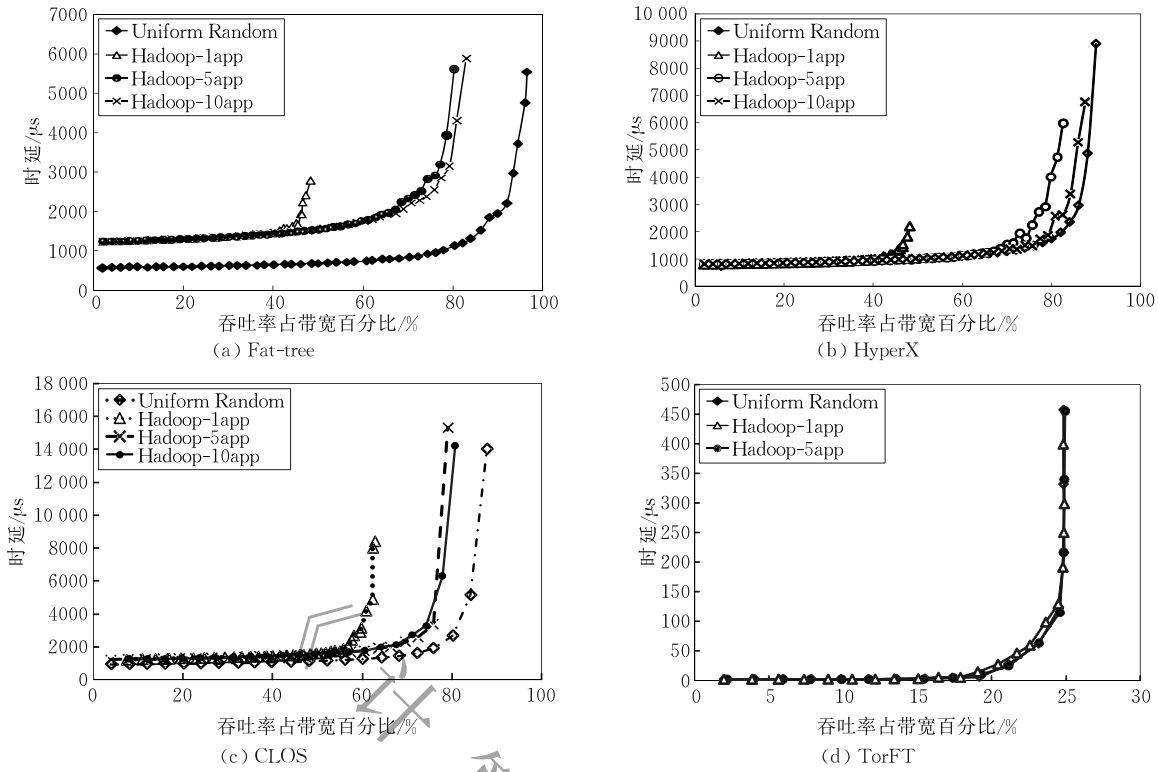


图 12 “差异化”流量通信性能对比测试

升,网络性能最差. TorFT 网络拥塞状态最为明显. Incast 流量最大吞吐率占带宽比重为 24.6%.

下面对不同网络负载不同特征流量情况下,所表现出的不同网络性能进行分析. 在“差异化”流量负载实验中,除网络性能最差的 TorFT 网络,其他三种网络负载单应用(1app)特性流量条件下,相较均匀随机流量均具有超过 30%以上的通信性能差异;而特性流量由于更具有局部通信密集特点,网络负载该流量时性能出现劣势现象更加明显. 结合多应用特性流量,即饱和特性流量下四种网络在 MapReduce 特性流量下的接收带宽性能,可以体现不同网络在运用到实际系统中的通信性能差异.

下面对各种网络在性价比方面的表现进行分析. 网络选型和设计时,除了要考虑网络通信性能外,还需要考虑到数据中心设计的成本问题,即“性能价格比”,即网络通信性能与网络结构对交换设备的端口数量的需求的比值. 本节对四种网络通信性能价格比进行分析,计算方法是网络端口总数($NumP_{all}$)去除全部连接 TOR 的交换机端口($NumP_{TOR}\times NetScale$)后,通过式(1)计算性价比.

$$G=\frac{\mathrm{Max}(ThroughPut)}{(NumP_{all}-NumP_{TOR}\times NetScale)\times G_{TorFT}} \quad (1)$$

计算结果如图 13 所示,式(1)中“G”表示目标网络的归一化性价比,“Max($ThroughPut$)”表示该网络在负载 MapReduce 特性流量时可达到的最大接收带宽,该参数可通过图 10 实验中获得. “G_{TorFT}”表示 TorFT 网络的性价比.

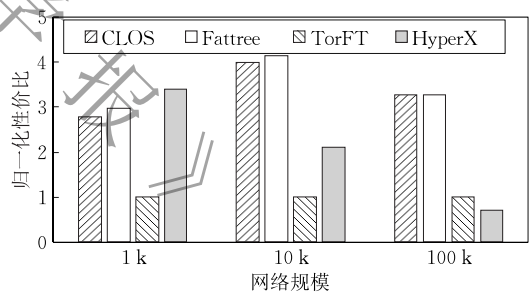


图 13 通量带宽与端口消耗的归一化性价比

以 TorFT 作为被除数(如式(1)中的 G_{TorFT}),对其它网络数据进行归一化计算,可得归一化结果,如图 13 所示. 在 1k 规模下,CLOS,Fattree 和 HyperX 三种网络的性价比接近,均高于 TorFT 网络,其中 HyperX 网络性价比最高. 但是在 10k 规模下,相对 HyperX 与 TorFT 网络,CLOS 与 Fattree 性价比优势明显;而 HyperX 网络随着规模扩展,交换机端口的需求增长明显,进而严重影响网络性价比. 在 100k 规模下,CLOS 与 Fattree 性价比基本持平. 而前文中 HyperX 网络所表现出的高吞吐量优势,已经追赶不上网络端口需求增长导致成本激增带来的劣势,且出现了 HyperX 网络性价比低于 TorFT 网络的结果.

5　相关研究

传统的“黑盒”流量产生方法，已经有较为充分的研究. 文献[4]提出一种高精度可扩展的建模机制，利用分层的“马尔可夫链”分析方法，对时间维度和空间维度的大数据应用网络行为进行建模. 文献[5]提出空间维度的模型以及 4 种可随时间进行动态流量调整的模型. 但是该方法产生的数据流量与真实的通信流量在通信速率数值上具有明显的差异. 文献[12]针对数据中心虚拟机间的虚拟链路进行建模，提供从虚拟链路到实际物理链路的映射，能有效地描述数据中心流量的空间特征. 文献[13]在“Oktopus”流量模型的设计方法中，提出了原始的虚拟网络抽象模型以及改进型虚拟抽象模型，充分考虑到了流量的空间局部性.

如何基于数据流采取合理的抽样方法，是对流量测量研究的关键. 文献[14-15]均对网络流量分析中较常用的流量属性特征进行了统计计算. 文献[16]中对数据流缩减了用于流量分析的数据量，同时保留了流量分析可控的重要信息. 针对网络中流量的抓取方法，文献[17]通过对应用层交互数据进行抓取与分析，设计并实现了面向高层次信息分析的流量提取工具. 针对网络流量分类与特性计算的相关研究，文献[18]利用单机散列聚合的方法进行网络级分流和特征计算. 针对网络中的流量分析，文献[19]基于深度学习提出了基于时间域流量预测与流量回归预测两种网络流量预测与估计模型. 针对流量识别与加工处理方面的研究，文献[20]通过对网络协议数据包设置“网络指纹”的方法，使用人工神经对网络设备进行分类判决. 针对网络流控与带宽分配策略对网络性能的影响，文献[21]研究集群中带宽分配管理方式的不同，对 K-Means 在不同数量集群中运行情况的差异化影响.

传统流量产生机制与方法虽然都具有较高的应用特征复现，但所使用的流量产生方法往往不具有在数据量和网络规模方面的通用性. 目前所提出的流量产生方法的贡献仅以“黑盒”或是结合通信协议的通信实现细节，集中在对所采集流量数据的分析上，没有针对不同的网络拓扑结构以及通信流量产生的本质原因进行分析. 同时目前文献也没有直接对运行分布式系统的数据中心，提出网络结构优化方面的建议.

6　总结及下一步工作

针对 MapReduce 计算模式的分布式系统，本文所提出的基于事件的流量产生方法，能够更加细粒度地反映 MapReduce 业务计算的实际运行特征，且满足应用种类、数据量规模和网络规模的多样性需求. 单一面向流量“黑盒”分析的流量产生方法，均从总体数据流量的宏观量化分析出发，业务特征分析粗粒度，往往不具备应用方法的普适性. 模拟准确率评测结果表明，本文的事件驱动的流量产生方法能够有效产生反应分布式系统内计算业务所产生的流量特征.

本文研究中的路由和数据流调度方法均是按照路由与数据流调度性能最优情况，对大规模网络交换进行模拟. 实际数据中心承载的应用种类差异性较大，任何流量模型都很难产生与任何一个数据中心某段时间内，完全一致的多租户、高并发混杂流量. 但是，本文基于事件驱动的对流量进行建模的思想，以最优化的方式对网络流量产生的机理和其特征进行了研究，产生符合计算模式特性的流量.

本研究中关于特性分析与提取方面的工作，可作为对不同应用或计算模式特征分析的基础性可扩展研究工作. 同时，基于事件驱动的流量模型，对流量建模方法也具有借鉴意义. 最终产生的流量是否能够体现具体应用的计算模式特征，还与是否对具体计算模式处理细节的理解和分析紧密相关.

本研究工作后续还会结合不同数据中心中承载应用的差异化特点，以便更有针对性地产生符合实际数据中心网络运行情况的流量；同时在今后的工作中，将结合更大规模的系统进行实体测试. 面向 MapReduce 特性流量产生方法与网络评测，目前值得进一步研究的问题还有很多，会逐步成为高性能计算和大数据网络体系结构的重要研究热点.

致　谢　感谢中国科学院国有资产经营有限责任公司对本论文的支持！感谢各位编辑和审稿人为本文的完善所提出的有益建议！

参　考　文　献

[1] Grodowitz M L, Sreepathi S. Hierarchical clustering and k-means analysis of HPC application kernels performance characteristics//Proceedings of the High PERFORMANCE Extreme Computing Conference. Waltham, USA, 2015: 1-6

[2] Yang X, Liu N, Feng B, et al. PortHadoop: Support direct HPC data processing in Hadoop//Proceedings of the IEEE International Conference on Big Data. Santa Clara, USA, 2015: 223-232

[3] Islam N S, Wasi-Ur-Rahman M, Lu X, et al. Performance characterization and acceleration of in-memory file systems for Hadoop and Spark applications on HPC clusters// Proceedings of the IEEE International Conference on Big Data. Santa Clara, USA, 2015: 243-252

[4] Delimitrou C, Sankar S, Kansal A, et al. ECHO: Recreating network traffic maps for datacenters with tens of thousands of servers//Proceedings of the IEEE International Symposium on Workload Characterization. San Diego, USA, 2012: 14-24

[5] Xie D, Ding N, Hu Y C, et al. The only constant is change: Incorporating time-varying network reservations in data centers. ACM SIGCOMM Computer Communication Review, 2012, 42(4): 199-210

[6] Roy A, Zeng H, Bagga J, et al. Inside the social network's (Datacenter) network. ACM SIGCOMM Computer Communication, 2015, 45(5): 123-137

[7] Benson T, Anand A, Akella A, et al. Understanding data center traffic characteristics. ACM SIGCOMM Computer Communication Review, 2009, 40(1): 65-72

[8] Dally W J, Towles B. Principles and Practices of Interconnection Networks. Oxford, UK: Morgan Kaufmann Publishers, 2004

[9] Duato J, Yalamanchili S, Ni L. Interconnection Networks: An Engineering Approach. California, USA: IEEE Computer Society Press, 1997: 17

[10] Fan Z, Cao Z, Su Y, et al. HiNetSim: A Parallel Simulator for Large-Scale Hierarchical Direct Networks//Network and Parallel Computing. Berlin Heidelberg, Germany: Springer, 2014: 120-131

[11] Guo De-Ke, Luo Lai-Long, Li Yan, et al. Aggregating Incast transfers in data centers. Journal of Computer Research and Development, 2016, 53(1): 53-67(in Chinese)
(郭得科, 罗来龙, 李妍等. 数据中心内 Incast 流量的网内聚合研究. 计算机研究与发展, 2016, 53(1): 53-67)

[12] Guo C, Lu G, Wang H J, et al. SecondNet: A data center network virtualization architecture with bandwidth guarantees //Proceedings of the ACM Conference on Emerging NETWORKING Experiments and Technology. Philadelphia, USA, 2010: 620-622

[13] Ballani H, Costa P, Karagiannis T, et al. Towards predictable datacenter networks. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 242-253

[14] He Gao-Feng, Yang Ming, Luo Jun-Zhou, et al. Online identification of Tor anonymous communication traffic. Journal of Software, 2013, 24(3): 540-556(in Chinese)
(何高峰, 杨明, 罗军舟等. Tor 匿名通信流量在线识别方法. 软件学报, 2013, 24(3): 540-556)

[15] Korczyński M. Classifying Application Flows and Intrusion Detection in the Internet Traffic[Ph. D. dissertation]. École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (EDMSTII), University of Grenoble, France, 2012

[16] Zhou Ai-Ping, Cheng Guang, Guo Xiao-Jun, et al. High-speed network traffic measurement method. Journal of Software, 2014, 25(1): 135-153(in Chinese)
(周爱平, 程光, 郭晓军等. 高速网络流量测量方法. 软件学报, 2014, 25(1): 135-153)

[17] Amro A, Almuhammadi S, Zhioua S. NetInfoMiner: High-level information extraction from network traffic//Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp). Jeju Island, South Korea, 2017: 143-150

[18] Li Ming, Ye Ji-Hua, Li Jian-Lian, et al. The research on network traffic management based on hash aggregation. Journal of Jiangxi Normal University, 2011, 35(2): 174-177

[19] Nie L, Jiang D, Guo L, et al, Traffic matrix prediction and estimation based on deep learning for data center networks// Proceedings of the GLOBECOM Workshops. Washington, USA, 2017: 1-6

[20] Reece T, Sathyanarayana S, Robinson W H, Beyah R A. On the outside looking in: Towards detecting counterfeit devices using network traffic analysis. IEEE Transactions on Multi-Scale Computing Systems, 2017, 3(1): 50-61

[21] Purnawansyah, Haviluddin. *K*-means clustering implementation in network traffic activities//Proceedings of the 2016 International Conference on Computational Intelligence and Cybernetics. Makassar, Indonesia, 2016: 51-54

SHAO En, born in 1988, Ph. D. candidate, engineer. His main research interests focus on high performance computing, computer architecture, SDN, big data and optical interconnection.

SUN Ning-Hui, born in 1968, Ph. D., professor, Ph. D. supervisor. His main research interests include computer architecture, high performance computing.

GUO Jia-Liang, born in 1992, M. S. candidate. His main research interests include distribution system and high performance computing.

YUAN Guo-Jun, born in 1983, Ph. D. candidate, senior engineer. His main research interests include computer architecture, optical interconnection and VLSI design.

WANG Zhan, born in 1986, Ph. D., assistant professor. His main research interests include virtualization technology and computer architecture.

CAO Zheng, born in 1982, Ph. D., associate professor. His main research interests include high performance computing and interconnection networks.

Background

Nowadays, "IT3.0" has come into our eyes which includes AI, big data, and architecture technology and so on. Currently big data is increasingly attracting the attention which has become the most importing role in research of computer science. However, the traffic modeling of big data is a kind of difficult problem, because the method of traffic generation rely on huge computation ability and data storage.

Big data platform has the cost-effective advantage of data processing with the feature of simplified programming and parallel computing, which has been more and more recognized by the industry. In recent years, the community of high-performance computing is also increasingly using Big Data platform for HPC data processing, which has become a powerful means for scientific data analysis gradually. Scientific data traffic generated by the application of HPC tends to have many requirements, including high quality processing, compute in communication link and huge date size, which is called as "high-throughput" traffic. The scale of data processing and the port cost of network need to be considered during the design of DC for distributed computer system. The main method of architecture design in DC includes two ways: (1) The design of private network according to application-specific. (2) The optimized implementation for the general network topology according to the traffic character. The application by customized network need to be processed in an effective way based on distributed data processing system.

At present, MapReduce is the most widely used computing and communication model in distributed system, and "Black-Box" is the traditional method for traffic generating. Since "Black-Box" is coarse-grained with poor scalability, this paper introduces an event-driven method by fine modeling of MapReduce traffic and further implements corresponding traffic generator in a well-tested parallel network simulator. The contrast experiment with real application traffic shows our method can correctly reflect the applications' characteristics when system scales.

Thanks for long-standing strong support of the CAS Holdings. This research was supported by the National Program on Key Research Project (2016YFB0200300, 2016YFGX030148, 2016YFB0200205, 2016GZKF0JT006), the National Natural Science Foundation of China (61572464, 61402444), and the Strategic Priority Research Program (XDB24060600).