大数据系统 Benchmark 综述

闫义博[1] 朱文强[2] 杨仝[3] 李晓明[3]

（[1]北京大学深圳研究生院 深圳 518055 [2]对外经济贸易大学信息学院 北京 100029 [3]北京大学计算机系 北京 100871）

摘要：Benchmark 是目前最主要的计算机系统性能评测技术，其评测的内容主要包括软件、硬件以及系统自身这三个方面中的一个或多个。在大数据时代背景下，与传统计算机系统相比，大数据相关的计算机系统具备了更高的多样性以及复杂性，因此 benchmark 评测技术将涵盖广泛的应用领域并提供多样的数据类型和复杂的数据操作。本文对 benchmark 评测基准中的测试规范进行了归纳总结，同时还列举了在大数据时代背景下 benchmark 评测技术开发中的一些挑战以及发展趋势。

关键词：基准测试，测试方法，大数据，性能

A Survey of Benchmark in Big Data

Yan Yibo[1], Zhu Wenqiang[2], Yang Tong[3], Li Xiaoming[3]

([1] Shenzhen Graduate School, Peking University, Shenzhen, 518055, China;

[2] School of Information Management, University of International Business and Economic, Beijing, 100029, China;

[3] Department of Computer Science, Peking University, Beijing, 100871, China)

Abstract: Benchmark is currently the most important technique for evaluating a computer system. The content of assessment mainly includes one or more of the three aspects of the software, the hardware and the computer system itself. In the Big Data era, compared with traditional computer system, the diversity and complexity of big data related computer systems are higher. Therefore, benchmarking technology will cover a wide range of applications and provide a wide range of data types and complex data manipulation. This paper summarizes some testing specifications and methods in benchmark and lists several challenges to adaption to changes from big data era and development trend in the development of benchmarking.

Keywords: benchmarks, benchmarking methodology, big data, performance

通信作者：杨仝 (yangtongemail@gmail.com). 国家重点研发计划 (2016YFB1000304), 973 项目 (2014CB340400), 国家自然科学基金 (61672061),中国科学院网络数据科学与技术重点实验室开放基金课题

1 引言

在计算机领域，benchmark 是一种被广泛应用于评测计算机系统的相关性能的技术。Benchmark 原指测量领域中的基准点，常用于判断不同测量对象之间的某个测量指标的差异。在计算机领域，benchmark 技术常常根据具体的应用领域建立相应的测试规范，然后依据测试规范设计测试流程，通过对该应用领域内的不同计算机系统进行测试得到测试结果，测试结果可以反映出不同计算机系统之间的性能指标的差异[错误! 未找到引用源.]。Benchmark 常用于评测计算机系统的性能测试，主要在测试响应时间、传输速度、吞吐量、资源占用率等方面，是基于性能的计算机系统设计中不可缺失的重要环节[2]。

随着计算机技术的发展，出现了越来越多的计算机系统，而如何评价某个应用领域中的计算机系统成为了学术界和工业界需要解决的首要问题。此外，在当前的大数据时代背景下，越来越多的应用领域需要使用大数据相关技术来应对数据的数量和种类的不断增加。大数据的特性使得大数据领域内的计算机系统与传统计算机系统之间存在一定的差异，例如，在对流式数据进行处理时，根据处理的时效性不同，计算机系统通常采用批量计算或流式计算，随着数据量的不断增加，人们将计算机系统开发的关注点转向低延迟、高吞吐和持续可靠的运行，这使得更加强调计算数据流和低时延的流式计算越来越受到欢迎，目前，主要的大数据处理技术包括 Hadoop[2]及其衍生技术，Hadoop 技术体系包括 HDFS[2]、MapReduce[2]和 HBase[2]等，其中还有一些用于处理流式数据的组件，例如：Tez[2]和 Spark Streaming[2]。此外，传统的 benchmark 技术还存在着样本规模较小和缺乏变量控制等问题[9]。因此用于评测大数据相关的计算机系统的 benchmark 在制定测试规范时应当充分考虑大数据特性给计算机系统带来的改变。

本文将首先介绍 benchmark 的组成及其测试规范与方法，然后列举在大数据时代背景下设计 benchmark 需要面临的一些挑战，最后，我们将介绍部分著名的常用大数据相关的 benchmark 技术。

2 Benchmark 介绍

2．1 Benchmark 的组成

Benchmark 主要由三部分组成：数据集、工作负载和度量指标。通常 benchmark 会为使用者提供两种程序，一种是将测试数据集装载程序，负责为被测试的计算机系统提供测试数据集，另外一种是测试的执行程序，负责为被测试的计算机系统提供工作负载。通过这两中程序的协同配合完成对计算机系统的评测。

Benchmark 中的数据集大体可分为三类：结构化数据、半结构化数据和非结构化数据。结构化数据也称作为行数据，指的是可以用二维表结构实现和表达其逻辑关系的数据，同时，结构化数据需要遵循对数据的格式与长度的约束，结构化数据常用关系型数据库存储和管理。具体的应用场景有企业财务系统、电子商务交易系统等。半结构化数据与结构化数据相似，但是半结构化数据并不严格遵循关系型数据库所规定的数据模型结构，其表达的对象可以具有不同的属性，但是与非结构化数据相比，它又具有一定的结构性，可以用较为宽松的数据模型进行描述，例如使用可扩展标记语言（XML）和超文本标记语言（HTML）。半结

构化数据的具体应用场景有邮件系统、成员档案系统等[10]。非结构化数据指的是那些难以用数据库的二维逻辑表示，不遵循一定的数据模型的数据，例如图片和视频等数据。非结构化数据的具体应用场景有视频监控系统、音乐网站等。在实际的应用场景中，计算机系统使用的数据类型可能是这三种其中的一种或者几种的混合。

工作负载是 Benchmark 中较为重要的一部分，它决定了测试结果的类型。工作负载可以按照不同的维度划分，按照应用领域可分为社交网络、电子商务和搜索引擎等；按照密集计算类型可分为 CPU 密集型计算和 I/O 密集型计算等；按照计算范式可分为批处理、机器学习和图计算等；按照计算延迟可分为在在线计算、离线计算和实时计算等。但是总体上工作负载一般包括处理大量数据、传输大量数据和进行高强度计算三大类。

度量指标用于直观地体现不同计算机系统某方面性能的优劣，由测试结果表示，具体的度量指标要依据测试的应用领域以及目的制定。一般情况下，为了得到某个度量指标，还需要设计一系列相关的度量指标用于约束整个测量过程来保证测量结果的准确性。

2.2 Benchmark 的测试规范

测试流程的设计需要遵守一定的测试规范。测试规范需要明确测试的目的，根据目的制定相应的度量指标。对于不同的测试对象应该有不同的测试重点，测试对象可分为组件测试和系统测试，但是无论对哪种对象进行测试，测试都需要在一个完整的计算机系统上完成。因此需要通过设置一些约束来提升测量结果的准确性，约束主要包括系统之间不同部件的相互作用、不同工作负载的占比和度量指标之间的关系等因素。此外，测试规范还应当制定相应的测试流程，包括系统环境配置、测试的步骤、每个步骤所用的方法以及不同方法中具体的参数设置等条件。最后，测试规范还应当规定评测报告的相关内容，主要是评测环境的配置以及测试结果的表现形式，方便他人重现测试结果和对比。

测试规范的来源较为广泛。有来自于工业界的一些权威组织，例如事务处理性能委员会（Transaction Processing Performance Council）的 TPC 系列测试规范和商业应用性能公司（Business Applications Performance Corporation）指定的相关测试规范；也有来自于一些开源的测试项目，例如用于评测文件系统的 IOzone 和用于评测 CPU 和内存性能的 HINT；有一些专业的评测公司会制定测试规范，例如针对于手机和其它基于 ARM 的设备进行评测的安兔兔；还有一些非评测公司会制定自己的测试规范，例如微软的 Windows System Assessment Tool 就是用于评测那些操作系统为 windows 系列的硬件的性能；此外，研究机构和一些生产商也会根据自身的业务需求执行测试规范。总的来说，虽然评测结果的权威性与测试规范的权威性相关联，但是考虑到评测目的的不同，对于测试规范的选择仍然具有一定的灵活性。

3 大数据时代下 benchmark 所面临的挑战与发展趋势

随着新的计算机系统不断出现，各种各样的 benchmark 技术也应运而生，对于 benchmark 技术的评价

也变得愈加重要。通常来说，一个好的 benchmark 需要具备五个特性，分别是：相关性、可重复性、公平性、可验证性和可使用性[11]。相关性与具体的业务环境相关联，它指的是 benchmark 提供的评测结果所蕴含的信息对使用者评判计算机系统所具备的价值高低。计算机系统的相关性能数据应当通过基于该系统的 benchmark 得到[12]。同时，该 benchmark 还应当与真实的应用领域相关联[13]。可重复性指的是按照相同的测试规范，在同一环境下，对于同一个计算机系统，benchmark 应当提供相同的评测结果，如果 benchmark 每次都提供不一样的评测结果，那么这个评测结果是不可信的。公平性指的是在不同的测试环境下，benchmark 对计算机系统的评测结果应当具有一致性，不应该出现与其它计算机系统相比一个计算机系统在某个测试环境下表现最好而在另外一个测试环境下表现较差的情况。可验证性指的是 benchmark 应当充分证明其评测结果的准确性。可使用性指的是 benchmark 应当是用户友好的，用户不但能够方便使用 benchmark 进行评测，而且也能够直观地理解评测结果。这五个特性指导了 benchmark 技术的发展方向。然而，benchmark 技术自身的发展并不是封闭的，实际上，benchmark 技术的发展应当更多的关注和依赖其测试对象的发展，也就是计算机系统的发展。

在大数据时代下，计算机系统在系统、应用和数据这三个方面发生了变化[14]。应用场景复杂度的增加以及真实数据生成速度的不断加快导致大数据相关的计算机系统与传统计算机系统所采用的数据处理技术有所差别。由于大数据相关领域内生成的真实数据往往具有较大的规模，因此，当前大数据相关的计算机系统通常采用流式处理的方法处理各种数据。此外，流式处理技术也有所改变和发展，sketch 是一种在流式处理中常用的数据结构，起初，sketch 主要用于统计数据集中不同元素的频度[2]，凭借着较低的内存占用以及较快的查询速度越来越受到欢迎，随着 sketch 相关技术的发展，sketch 也从一些经典的结构[2]衍生出许多新的类型[2]，这使得 sketch 在流式处理中有着越来越广泛的应用领域，例如：压缩感知中的稀疏逼近[2]、自然语言处理[2]和数据图[2]等[2]。这些差异和变化使得传统的 benchmark 技术难以满足对日益变化的大数据相关的计算机系统进行评测的需求，在开发新的 benchmark 技术时不但应当满足上述的五个特性，还应当兼顾大数据的特性，这就使得 benchmark 技术的开发面临着新的挑战。大数据具有四个特性：海量数据、数据类型多样、价值密度低和处理速度快[36]。大数据相关的计算机系统不但要处理海量的多样的数据，同时还要具备较快的数据处理速度。因此，在测试数据集方面，benchmark 需要为计算机系统提供更大规模的数据量和更多类型的数据；在工作负载方面，随着大数据涵盖越来越多的应用领域，benchmark 也应当丰富其工作负载的种类；在度量指标方面，benchmark 技术应当根据具体应用领域内的业务变化而作出修正。在三个方面，benchmark 都面临着挑战。

首先，由于大数据相关的计算机系统需要对大量的数据进行处理分析，benchmark 评测技术需要提供相应的测试数据，为了使测试结果更具实际意义，测试数据应当尽可能的使用实际应用场景下生成的真实数据，但是实际上为被测试系统提供大量的真实数据往往存在较大难度。一方面，真实数据的获取较为困

难。真实的测试数据通常来源于实际应用场景，因此，大量的真实数据往往被一些企业所掌控，企业会利用这些数据对运营状况作出判断或者对行业发展作出预测，这就使得这些真实数据所蕴藏的商业价值受到企业的高度重视，从而使得数据的持有者处于商业利益的考虑而拒绝分享其持有的数据。而那些愿意分享其持有的数据的企业常常会出于对用户隐私、商业机密等因素的考虑而对其分享的数据进行处理，这就使得测试用的数据和真实的数据之间仍然存在一定的差异。当然，通过搭建相应的环境来模拟真实数据的生成也是获取数据的一种手段，但是这样不但会增加测试成本，而且数据的生成速度也难以得到保证，此外，由于实际环境的复杂性较高，生成数据所具备的价值以及规模无法与真实场景下得到的数据相比。另一方面，大量测试数据的传输和存储也将增加测试成本和测试难度。用户在获取测试数据时将消耗大量的传输资源，从 benchmark 的设计上看，这不但增加了评测成本，而且也不利于对已有的 benchmark 进行扩展，有碍于测试数据的更新。同时，大量的测试数据需要庞大的存储空间，这也将导致评测成本的增加。因此，为了有效地获取测试数据以及降低评测成本，benchmark 技术的开发者应当考虑如何充分利用有限的真实有效的数据集。Benchmark 附带的测试数据集应当是规模较小，同时具备原始真实数据的各项特征。随着 benchmark 评测技术的发展，为了适应大数据所带来的改变，benchmark 可以为用户提供较小规模的数据集或者是容易获取的公开的数据集，然后通过提供相应的测试数据生成工具来生成测试数据。这样不但方便用户获取大量的测试数据，同时也有利于用户根据自己的需求来修改测试数据生成参数以满足不同的评测需求。但是这样也增加了测试数据生成工具的设计难度，一方面，生成的测试数据需要与原始真实数据具备相同的特征，这就要求开发人员对测试数据的应用领域有着充分的了解，另一方面，数据建模的方法将影响测试数据的生成速度，数据的多样性以及应用领域的多样性都为数据建模方法的选择增加了难度。因此，尽管大数据的特性使得 benchmark 中的测试数据集发生了较大的变化，数据的获取与使用成本都将增加，但是开发人员可以通过提供工具来合理地生成符合要求的测试数据，降低评测成本，从而保证 benchmark 中可使用的数据集具备越来越高的多样性和复杂性。

其次，随着大数据的应用领域越来越广泛，工作负载也具备了较强的多样性，这就使得制定测试规范变得更加复杂。Benchmark 技术应当使用公认的标准选取若干重要的大数据应用领域，并准对这些领域提供相应的工作负载[37]。由于计算机系统需要处理的数据的种类越来越多，相应地，系统工作负载的组成也将变得复杂，这时，如何设置工作负载的组成及其中不同部分在整体中所占比例将成为首先需要解决的问题。工作负载的设置应当根据实际的应用场景进行设置，通常工作负载包括数据的处理、传输和计算，根据大数据特性可知，虽然数据规模庞大，但是数据的价值密度较低，因此如何从有限的数据中尽可能的提取价值是大数据相关的计算机系统的工作重点，但是不同的大数据相关的计算机系统可能对这三种工作负载中的一种或多种有所偏重，因此在对某个应用领域内的计算机系统进行评测时开发人员需要对该领域有着较为深入的理解，从实际出发，综合考虑计算机系统的工作特点，依据实际的业务环境，合理设置工作负载的总规

模以及分配不同工作负载的比例，从而提升评测结果的价值。

最后，大数据的多样性及其技术的快速发展也为度量指标的制定增加了难度。大数据领域的不断扩大以及技术的不断发展加快了大数据相关的计算机系统的更新，随着大数据相关技术研究重点的转变，应用领域内不同业务发展方向的改变，部分度量指标有效性会逐渐降低，因此评测的度量指标也应当随之灵活变化，这就要求 benchmark 具备一定的可扩展性。这用户将在评测过程中的起到更加重要的作用，因为用户常常对其所在应用领域的发展有着更加精准的认识，当用户需要围绕一项新的技术研究时，已有的评测技术中的度量指标可能无法满足用户的评测需求，而扩展性较高的 benchmark 评测技术将提升用户在评测过程中的自主权，从而满足用户的各项需求。但是这样也带来了新的挑战，为了保证评测结果的客观性和准确性，benchmark 评测技术开发人员需要制定相应的详细的测试和规范，同时，开发人员还应密切关注不同应用领域内业务的发展以及时更新度量指标，这些都需要开发人员的持续努力和对新需求作出及时反应。

4 大数据相关的常用 benchmark 技术介绍

目前的常用 Benchmark 技术主要有以下几种：

4.1 TPC 测试集：

TPC 测试基准是由微软、英特尔、惠普等知名厂商共同建立的测试基准[38]。TCP 基准主要是针对数据库管理系统的测试系统，其测试数据库管理系统的 ACID 特性、查询速度及联机事务处理等能力，从而对数据库管理系统进行性能测试。目前为止，TPC 共发布了 TPC-A、TPC-B、TPC-C 等八个标准[40]。但是 TPC 基准只提供数据装载程序，不提供工作负载生成程序。

4.2 BigBench 基准：

BigBench[40]基准目前更多地运用于零售网站等商业系统中。BigBench 数据模型部分参考了 TPC-DS 大数据测评基准，但是 BigBench 丰富完善了 TPC-DS 数据模型中缺少的半结构化与非结构化的数据类型，同时提供了工作负载的生成程序[40]。

4.3 SPEC 基准：

SPEC[43]是一家非盈利公司，公司的主要业务是开发有效实用的 Benchmark 基准。目前其开发的基准主要应用于 CPU、图形/应用处理、高性能计算机/消息传递接口(MPI)、Java 客户机/服务器、邮件服务器、网络文件系统、Web 服务器等，以测试处理器的运算速度及吞吐量等相关性能[43]。

4.4 HiBench 基准：

目前，分布式系统基础框架 Hadoop 在云计算的大规模数据分析中的表现变得越来越突出，因此针对于 Hadoop 的大数据测试基准 HiBench[44]也被更多地应用。HiBench 基准由一组 Hadoop 程序组成，包括测试地微基准及实际 Hadoop 应用程序。HiBench 评价和表征 Hadoop 框架的速度、吞吐量、HDFS 的带宽、系统资源利用率和数据访问模式，从而对 Hadoop 系统做出性能评价。

4.5 BigDataBench 基准：

目前虽然有着广泛多样的 benchmark，但这些大数据测评基准多为针对某一单独领域的专用基准，难以覆盖大数据的多样性与复杂性。BigDataBench[45]被开发出来即是主要为了解决这一问题。BigDataBench 覆盖 5 个典型应用领域，可生成非结构化、半结构化与结构化数据 3 种数据类型及离线分析、交互式分析、在线服务、NoSQL4 种不同负载类型，以达到可广泛适用于多种不同的大数据系统的目的。

4.6 YCSB：

YCSB 的全称是：Yahoo! Cloud Serving Benchmark (YCSB)[46]。YCSB 是 Yahoo 公司开发的用来对云服务进行基础测试的工具，用以比较不同的云服务系统的性能。YCSB 的主要针对对象是在云服务平台的 NoSQL 系统，YCSB 可以在同一硬件配置下对多个系统同时进行工作流创建和运行，并记录不同系统的处理速度，从而实现比较不同的云服务系统的性能的目的。

4.7 常用 benchmark 技术对比：

本文介绍了 6 种不同的 benchmark 技术，每一种 benchmark 均具有不同的特点。针对于不同的测试环境及测试需求，不同的 benchmark 技术有着自身独特的优势与缺陷。TPC 作为最早诞生的 benchmark 之一，有着许多的拓展与延伸版本；Hibench 在 Hadoop 系统的测试上被广泛地应用；BigDataBench 相较于其他的 benchmark 算法可以生成更多种类的数据类型与工作负载；YCSB 在网络云系统的测试中一直受到高度的认可。现将本文列举的 benchmark 的特点总结对比，如下图所示：

	生成数据类型	工作负载种类	测试对象	开源性
TPC	结构化数据	离线分析	Hadoop 及 DBMS	开源
BigBench	结构化数据、半结构化数据、非结构化数据	离线分析	Hadoop 及 DBMS	不开源
SPEC	N/A	离线分析	系统及结构	开源
Hibench	非结构化数据	离线分析、实时分析	Hadoop 及 Hive	开源
BigDataBench	结构化数据、半结构化数据、非结构化数据	在线服务、离线分析、实时分析	系统及结构、NoSQL 系统	开源

YCSB	N/A	在线服务	NoSQL 系统	开源

表 1：常用 benchmark 技术对比

5 总结

本文通过对多种不同的 benchmark 进行比较研究，对 benchmark 的组成和规范进行了概括性介绍，对 benchmark 的特性进行了描述。同时，在此基础上，本文指出，在新的大数据时代背景下，大数据特性导致计算机系统发生的变化使得 benchmark 技术的开发也不断受到重视与面对挑战，数据的多样性、复杂的运行环境与庞大的负载量对 benchmark 技术的开发提出了更多的要求，为此，更多的大数据测试基准也在不断产生，以满足对大数据相关的计算机系统进行评测的需求。

参考文献：

[1] Fleming, P. J., & Wallace, J. J. (1986). How not to lie with statistics: the correct way to summarize benchmark results[J]. Communications of the ACM, 29(3), 218-221.

[2] J.L. Hennessy and D.A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann[D], San Francisco, CA, 1996.

[3] Apache. Hadoop. http://hadoop.apache.org/.

[4] Shvachko K, Kuang H, Radia S, et al. The Hadoop Distributed File System[C]// IEEE, Symposium on MASS Storage Systems and Technologies. IEEE Computer Society, 2010:1-10.

[5] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Commun. ACM, 2008, 51(1):10-10.

[6] “HBase: Bigtable-like structured storage for Hadoop HDFS,” 2010, http://hadoop.apache.org/hbase/.

[7] Apache tez. http://incubator.apache.org/projects/tez.html.

[8] M. Zaharia et al. Discretized Streams: Fault-Tolerant Streaming Computation at Scale. In Proc. of the 24th ACM Symp. on Operating Systems Principles, 2013.

[9] Castor, Kevin (2006). "Hardware Testing and Benchmarking Methodology"[OL]. Archived from the original on 2008-02-05. Retrieved 2008-02-24.

[10] Abiteboul, S. (1997). Querying semi-structured data[J]. Database Theory—ICDT'97, 1-18.

[11] Gray J. Benchmark handbook: for database and transaction processing systems[M]. Morgan Kaufmann Publishers Inc., 1992.

[12] Seltzer M, Krinsky D, Smith K, et al. The case for application-specific benchmarking[C]//Hot Topics in

Operating Systems, 1999. Proceedings of the Seventh Workshop on. IEEE, 1999: 102-107.

[13] Chen Y, Raab F, Katz R. From tpc-c to big data benchmarks: A functional workload model[M]//Specifying Big Data Benchmarks. Springer, Berlin, Heidelberg, 2014: 28-43.

[14] 金澈清，钱卫宁，周敏奇等. 数据管理系统评测基准：从传统数据库到新兴大数据[J]. 计算机学报，2015，38(1)：18-34

[15] Aggarwal, C. C., & Yu, P. S. (2010, April). On classification of high-cardinality data streams. In Proceedings of the 2010 SIAM International Conference on Data Mining (pp. 802-813). Society for Industrial and Applied Mathematics.

[16] Chen, A., Jin, Y., Cao, J., & Li, L. E. (2010, March). Tracking long duration flows in network traffic. In Infocom, 2010 proceedings ieee (pp. 1-5). IEEE.

[17] Cormode, G., & Garofalakis, M. (2005, August). Sketching streams through the net: Distributed approximate query tracking. In Proceedings of the 31st international conference on Very large data bases (pp. 13-24). VLDB Endowment.

[18] Charikar, M., Chen, K., & Farach-Colton, M. (2002, July). Finding frequent items in data streams. In International Colloquium on Automata, Languages, and Programming (pp. 693-703). Springer, Berlin, Heidelberg.

[19] Liu, Z., Manousis, A., Vorsanger, G., Sekar, V., & Braverman, V. (2016, August). One sketch to rule them all: Rethinking network flow monitoring with univmon. In Proceedings of the 2016 ACM SIGCOMM Conference (pp. 101-114). ACM.

[20] Thomas, D., Bordawekar, R., Aggarwal, C. C., & Philip, S. Y. (2009, March). On efficient query processing of stream counts on the cell processor. In Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on (pp. 748-759). IEEE.

[21] Cormode G. Count-Min Sketch[J]. Encyclopedia of Algorithms, 2009, 29(1):64-69.

[22] Cormode G, Muthukrishnan S. An Improved Data Stream Summary: The Count-Min Sketch and Its Applications[C]// Latin American Symposium on Theoretical Informatics. Springer, Berlin, Heidelberg, 2004:29-38.

[23] Goyal A, Iii H D. Lossy Conservative Update (LCU) Sketch: Succinct Approximate Count Storage[C]// AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, Usa, August. DBLP, 2012.

[24] Roy P, Khan A, Alonso G. Augmented Sketch:Faster and More Accurate Stream Processing[J].

2016:1449-1463.

[25] Yang T, Liu A X, Shahzad M, et al. A shifting framework for set queries[J]. IEEE/ACM Transactions on Networking, 2017, 25(5): 3116-3131.

[26] Yang T, Zhou Y, Jin H, et al. Pyramid sketch: A sketch framework for frequency estimation of data streams[J]. Proceedings of the VLDB Endowment, 2017, 10(11): 1442-1453.

[27] Liu P, Wang H, Gao S, et al. ID Bloom Filter: Achieving Faster Multi-Set Membership Query in Network Applications[J].

[28] Zhou Y, Liu P, Jin H, et al. One memory access sketch: a more accurate and faster sketch for per-flow measurement[C]//IEEE Globecom. 2017.

[29] Gong J, Yang T, Zhou Y, et al. Abc: a practicable sketch framework for non-uniform multisets[J]. IEEE Bigdata, 2017.

[30] Gilbert, A. C., Strauss, M. J., Tropp, J. A., & Vershynin, R. (2007, June). One sketch for all: fast algorithms for compressed sensing. In Proceedings of the thirty-ninth annual ACM symposium on Theory of computing (pp. 237-246). ACM.

[31] Talbot, D., & Osborne, M. (2007). Smoothed Bloom filter language models: Tera-scale LMs on the cheap. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).

[32] Van Durme, B., & Lall, A. (2009, July). Probabilistic Counting with Randomized Storage. In IJCAI (pp. 1574-1579).

[33] Polyzotis, N., Garofalakis, M., & Ioannidis, Y. (2004, June). Approximate XML query answers. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data (pp. 263-274). ACM.

[34] Spiegel, J., & Polyzotis, N. (2006, June). Graph-based synopses for relational selectivity estimation. In Proceedings of the 2006 ACM SIGMOD international conference on Management of data (pp. 205-216). ACM.

[35] Pietracaprina, A., Riondato, M., Upfal, E., & Vandin, F. (2010). Mining top-K frequent itemsets through progressive sampling. Data Mining and Knowledge Discovery, 21(2), 310-326.

[36] 马建光，姜巍. (2013). 大数据的概念、特征及其应用. 国防科技[J], 34(2), 10-17.

[37] Burby J, Atchison S. Actionable web analytics: using data to make smart business decisions[M]. John

Wiley & Sons, 2007.

[38] 王良. Benchmark 性能测试综述[J]. 计算机工程与应用, 2006, 42(15): 45-48.

[39] Subramanian S, Raab F, Livingtree L, et al. Tpc Benchmark[J]. Journal of Marital & Family Therapy, 2003, 18(1):71–81.

[40] Ghazal A, Raab F, Raab F, et al. BigBench: towards an industry standard benchmark for big data analytics[C]// ACM SIGMOD International Conference on Management of Data. ACM, 2013:1197-1208.

[41] Chowdhury B, Rabl T, Saadatpanah P, et al. A BigBench Implementation in the Hadoop Ecosystem[M]// Advancing Big Data Benchmarks. Springer International Publishing, 2014:3-18.

[42] Henning J L. SPEC CPU2000: measuring CPU performance in the New Millennium[J]. Computer, 2000, 33(7):28-35.

[43] KAIVALYA DIXIT, TOM SKORNIA. Standard Performance Evaluation Corporation (SPEC)[OL]. http://www.spec.org/osg/web99/:2005.

[44] Huang S, Huang J, Dai J, et al. The HiBench benchmark suite: Characterization of the MapReduce-based data analysis[C]// IEEE, International Conference on Data Engineering Workshops. IEEE, 2010:41-51.

[45] Wang L, Zhang S, Zheng C, et al. BigDataBench: A big data benchmark suite from internet services[C]// IEEE, International Symposium on High PERFORMANCE Computer Architecture. IEEE, 2014:488-499.

[46] Dey A, Fekete A, Nambiar R, et al. YCSB+T: Benchmarking web-scale transactional databases[C]// IEEE, International Conference on Data Engineering Workshops. IEEE, 2014:223-230.

作者简介

闫义博，(1995.5 -)，男，硕士研究生，研究方向：网络信息工程。