

基于一致性哈希算法和 Ckafka 技术的 IMS 电话实时录音系统^①



江 淞¹, 王宝海², 赵金城¹, 宋 江¹, 段佳秀³

¹(国网江苏省电力有限公司 信息通信分公司, 南京 210024)

²(南京南瑞信息通信科技有限公司, 南京 210003)

³(南京邮电大学, 南京 210003)

通讯作者: 段佳秀, E-mail: 1151323916@qq.com

摘 要: 基于电路交换的传统电话录音系统因其结构复杂, 存储的不便捷和非实时性的特点, 已经无法满足新时代行政办公对高效, 即时的通话录音需求, 因此, 提出一种基于电力 IMS 的电话实时录音系统. 首先分析了电力 IMS 交换网中电话终端实时录音的业务需求; 其次介绍了系统的实现流程, 阐明了系统的关键技术: 利用录音服务器对其镜像端口的 SIP 报文进行解析获得媒体流并解码、采用一致性哈希算法的内存数据库作为解码数据的缓存机制、利用 Ckafka 技术在两者之间构建实时数据通道; 最后就响应时间、吞吐量、容错能力和推送的最大时延这四项指标对录音服务器进行性能分析, 结果表明该系统的实时性强, 吞吐量大, 具备一定的容错能力, 并能实现多服务器之间的负载均衡.

关键词: IMS; SIP; 镜像端口; 一致性哈希算法; Ckafka

引用格式: 江淞, 王宝海, 赵金城, 宋江, 段佳秀. 基于一致性哈希算法和 Ckafka 技术的 IMS 电话实时录音系统. 计算机系统应用, 2020, 29(5): 88-93. <http://www.c-s-a.org.cn/1003-3254/7354.html>

IMS Telephone Real-Time Recording System Based on Consistent Hash Algorithm and Ckafka

JIANG Song¹, WANG Bao-Hai², ZHAO Jin-Cheng¹, SONG Jiang¹, DUAN Jia-Xiu³

¹(Information and Communication Branch, State Grid Jiangsu Electric Power Co. Ltd., Nanjing 210024, China)

²(NARI Group Corporation Information & Communication Technology Co. Ltd., Nanjing 210003, China)

³(Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: The traditional circuit-switched telephone recording system has been unable to meet the needs of efficient and instant call recording in the new era administrative office due to its complicated structure, inconvenient storage, and non-real-time characteristics. Therefore, this paper presents a real-time recording system for power IMS telephone terminal to solve these problems. First of all, the article analyzes the business needs of real-time recording of telephone terminals in the power IMS exchange network. Secondly, the article introduces the implementation process of the system and clarifies the key technologies of the system: the system uses the recording server to parse the SIP message of its mirrored port for obtaining the media stream and decoding, and consistent Hash algorithm memory database is used as decoding data caching mechanism, and the message queue between the both is Ckafka. Finally, the performance of the recording server is analyzed based on the four indicators of response time, throughput, fault tolerance, and maximum delay. The analysis shows that the system has strong real-time performance, large throughput, certain fault tolerance, and load balancing.

Key words: IMS; SIP; mirror port; consistent Hash algorithm; Ckafka

① 收稿时间: 2019-08-22; 修改时间: 2019-09-10; 采用时间: 2019-10-21; csa 在线出版时间: 2020-05-07

基于电力 IMS^[1]交换网的电话业务是国家电网公司内部重要的通信业务之一,为国家电网员工提供了高效便捷的语音通话服务. 在一些重要的部门、岗位以及重要的电话会议等场景中,通常要求对通话进行录音,以记录来电的内容. 通话录音的内容既可以用于追踪业务的进度,也可用于事后回溯以避免业务纠纷. 因此,在电力 IMS 交换网实施电话录音的行为可以完善电网内部的电话管理制度,提高业务的监管水平.

随着电话交换技术由电路交换向分组交换演进,三网融合需求的呼声越来越高,而 IP 多媒体子系统 (IMS) 技术可以支持语音,电视媒体,数据这三种业务的传输,还具有统一的平台和接口,分层的架构设计,集中的业务控制与管理,可保证的 QOS 等优点. 此外,传统的基于电路交换的电话录音技术因其系统复杂,存储的不便捷和非实时性的特点,已经无法满足新时代行政办公对高效,统一,即时的通话录音需求. 因此,本文设计出一种基于电力 IMS 交换网的电话实时录音系统,该系统可为电力 IMS 交换网中的各类终端 (IP, POTS, PC 等) 提供即时、高动态、高质量的录音服务,并通过系统后台将录音记录即时推送至软件终端,以便 IMS 终端用户随时查阅录音.

1 基于 IMS 的电话录音系统设计

该系统借助基于 IMS 核心交换机的端口镜像技术实现对整个网络的监控管理,并能够及时抓取所需要的数据,借助一致性哈希算法来查找缓存服务器,利用 Ckafka 高性能消息队列来保证消息消费的有序性和负载均衡. 最终,设计与实现一种基于会话初始 (SIP) 协议^[2]的高动态即时录音系统,其系统框图如图 1 所示.

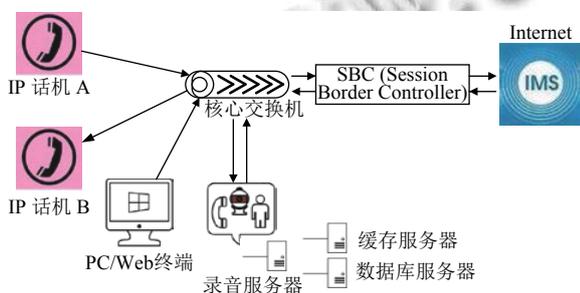


图 1 系统框图

系统的实现流程如下:

步骤 1. 主叫话机 A 向被叫话机 B 发起呼叫请求,主叫话机 A 经由核心交换机、SBC(边界会话控制

器)、电力 IMS 交换网、再回到核心交换机,向被叫话机 B 发送 SIP 协议中的 invite 信令^[3].

步骤 2. 在被叫话机 B 接收到 invite 信令后,给主叫话机 A 发送应答消息,主叫话机 A 确认应答消息后,开启与被叫话机 B 的通话.

步骤 3. IMS 软件终端与被叫话机 B 联动,当主叫话机 A 与被叫话机 B 的通话开启时,IMS 软件终端通知录音服务器开启录音.

步骤 4. 录音服务器从核心交换机的镜像端口获取信令包,再根据 from、to 等标签,对信令包进行 SIP 包解析,获取主被叫话机号码、IP、媒体端口及语音编码. 解析 RTP(媒体流)报文后,针对 g.711a、g.711u、g.729、g.726、iLBC 等不同编码方式^[4,5],将录音文件解码,通过 Ckafka 消息队列将解码的数据缓存至高性能缓存服务器.

步骤 5. 通话需要结束时,主叫话机 A 经由核心交换机、SBC(边界会话控制器)、电力 IMS 交换网、再回到核心交换机,向被叫话机 B 发送 SIP 协议中的 bye 信令.

步骤 6. 被叫话机 B 接收到 bye 信令后,给主叫话机 A 发送应答消息,主叫话机 A 确认应答消息后,结束与被叫话机 B 的通话.

步骤 7. 录音服务器取时间或集将主被叫话机的录音进行混音并存储为 PCM 格式. 录音服务器将通过系统后台将录音记录即时推送至 PC/Web 终端,以使用户随时查阅通话录音.

2 基于一致性哈希算法和 Ckafka 技术的录音服务器设计

通过对 SIP 报文中信令的解析,并从媒体端口得到通话双方的音频流,采用 Ckafka 消息队列构建实时数据通道,同时提供对数据流的处理,然后将得到的音频流传送至高性能内存数据库以保存数据. 在数据库和录音服务器之间插入分布式缓存^[6],使录音服务器不必频繁访问数据库,而是直接从缓存存取数据,从而缩短了数据的访问时间,提高了数据库的访问性能. 对于分布式缓存服务器的查找采用的是一致性哈希算法,它极大的提高了缓存命中率,以及减少在增删节点时数据迁移的成本.

2.1 Ckafka 技术

Ckafka 技术^[7]是一种消息队列的技术,它基于

Apache Kafka 消息队列引擎, 具有强大的吞吐量和可扩展性能强的特点, 主要用于高性能的流式处理、消息传输等场景, 并且完全兼容 Kafka. 其详细特性如下:

- (1) 可扩展性: 该框架可轻松扩展, 无需停机.
- (2) 高容量: 旨在处理大量数据, 并能对数据进行压缩.
- (3) 可靠性: 具有可恢复能力和一定的容错能力.
- (4) 数据转换: 可以把从信息源获取的数据流转化成要求的数据流格式.
- (5) 低延迟: 专注于传统消息传递以实现低时延.
- (6) 顺序读写: 和大部分的消息队列一致, Ckafka 可以保证数据按照顺序进行处理, 极大提升磁盘效率.
- (7) 异步通信: 在无需立即处理消息的场景下, 当访问量高时 Ckafka 仅将消息放入队列中, 等访问量降低后再对消息进行处理, 以降低系统负荷^[8].

本系统为集中录音系统, 要求并行处理的能力强, 由上面的介绍可知 Ckafka 消息队列具有强大的吞吐量, 多线程的流式处理, 具备数据压缩的特性以节约存储空间, 对于高并发数据流, 支持在线的水平扩展和消息的自平衡, 在最佳的情况下, 插入和删除数据的时间复杂度能降为 $O(1)$, 这极大的降低系统的复杂度, 提高系统运行的稳定性, 同时结合一致性哈希算法可以实现对服务器集群请求的均衡处理, 所以本文采用该技术来对录音系统的信令流和音频流进行处理, 可轻松实现毫秒级的消息处理, 极大的降低系统的时延, 提高系统的实时性.

2.2 一致性哈希算法

一致性哈希算法最初是由 Karger D 等提出的一种散列算法. 其最初是为了解决服务器集群中的热点问题, 整个算法的过程如图 2 所示^[9,10], 但是该算法具有一定的局限性, 当服务器的节点数较少, 那么对于多个请求可能会导致散列环上的数据倾斜以及服务器分配不均的问题, 因此在散列空间和物理节点之间加入虚拟节点^[11], 通过对虚拟节点的分配, 把多个虚拟节点与相对应的物理节点进行映射, 并在散列环上分布恰当的虚拟节点数目, 从而每台服务器都能达到均衡处理请求的目的.

本文提出的一致性哈希算法是在普通哈希算法上进行改进, 因为环形散列环大小为 2^{32} , 因此对目标利用哈希函数运算之后, 再进行模 2^{32} 运算, 即可将目标的 Key 分布在散列环上, 如式 (1) 所示.

$$Key = Hash(object) \% 2^{32} \tag{1}$$

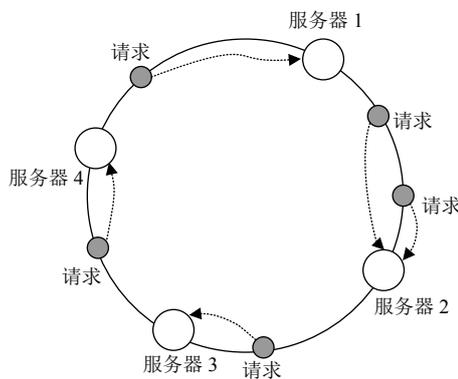


图 2 一致性哈希算法

在 SIP 报文中, Call-ID 可以作为这通会话的全局识别号, 因此可以作为该通电话的唯一标识符; 而对于不同的缓存服务器, 其 IP 地址也都不同, 因此可以利用缓存服务器的 IP 地址作为它的唯一标识符. 在不考虑虚拟节点的情况下, 本文提出的一致性哈希算法的步骤如下:

步骤 1. 利用每一台缓存服务器的 IP 地址作为哈希运算的对象并进行哈希运算, 然后把得到的结果进行模 2^{32} 运算, 得到相应的键值 $K_{s1}, K_{s2}, \dots, K_{sm}$, 至此, 已经将每一台缓存服务器都映射到散列环^[12]上, 对于 3 台服务器在散列环上映射情况如图 3 所示.

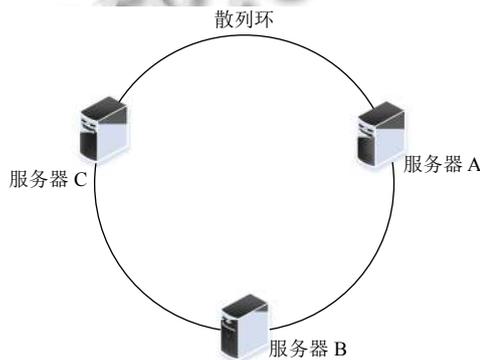


图 3 3 台服务器映射到散列环

步骤 2. 对当前打进来的某一通电话的 Call-ID 进行哈希运算, 然后再对运算结果进行模 2^{32} 运算, 得到对应的键值 K_c , 并映射到步骤 1 所示的散列环上, 现将缓存服务器和某一通电话都映射到散列环上, 其映射情况如图 4 所示.

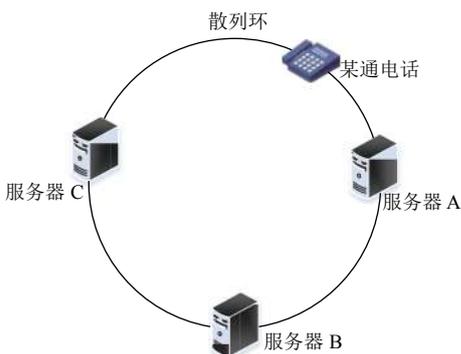


图4 某通电话与缓存服务器映射到散列环

步骤3. 从该通电话在散列环上映射的位置开始, 按顺时针方向, 找到最近的那一台服务器 A, 那么该通电话的媒体流就缓存到服务器 A, 如图5所示.

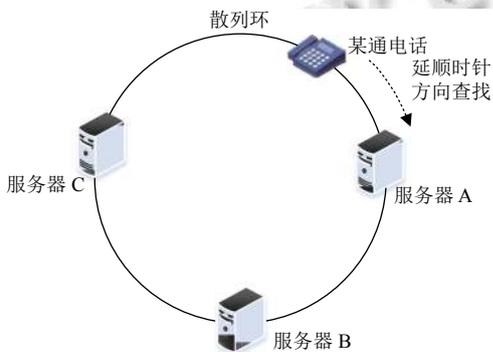


图5 寻找缓存服务器过程

对于多路通话, 亦可采用上述方法快速寻找到对应的缓存服务器. 若多路通话均映射到散列环相对固定的位置, 那么最终他们都会按顺时针寻找到最近的那台服务器缓存数据, 这将导致另外两个服务器空闲, 而当前服务器负载量过大甚至宕机. 对此, 采用虚拟节点技术^[13], 根据通话映射在散列表的位置关系, 利用物理节点复制出多个虚拟节点, 然后在散列环中安放多个虚拟节点. 随着虚拟节点的增多, 散列环上总的节点数就越多, 节点均匀分布的可能性就越大, 当多个请求同时到来时, 缓存服务器的负载就能更加均衡^[14-16], 如图6所示, 深色的表示物理节点, 浅色的表示复制出的虚拟节点. 由图分析可得: 若不引入虚拟节点技术, 将会有五通电话的请求将直接流向服务器 A, 一通电话的请求流向服务器 B, 服务器 C 处于空闲状态, 这样的分配方式极不平衡, 这将导致服务器 A 的负荷过大, 系统处理能力降低, 而引入虚拟节点技术, 每台服务器都

分到2个请求, 这样可以充分发挥每台服务器的性能, 提高系统的处理能力.

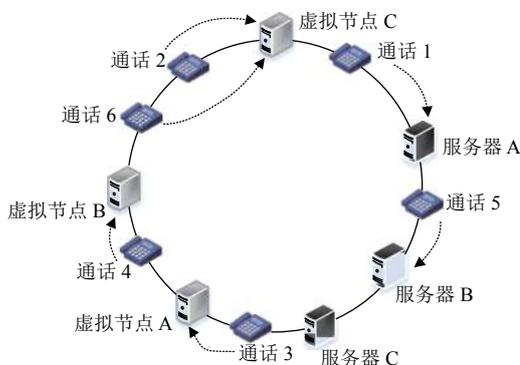


图6 虚拟节点

3 录音服务器的性能分析与评价

为了验证录音服务器的性能, 本文通过运行多台虚拟机来模拟录音服务器、缓存服务器集群以及数据库服务器集群, 并将采用一致性哈希算法和 Kafka 消息队列技术的服务器与采用普通的哈希算法的服务器进行比较, 利用 JMeter 工具来模拟用户的通话录音请求, 并根据服务器的响应时间、吞吐量、容错能力以及录音推送至客户端的最大时延这4项指标作为其评估性能的依据, 最后根据测试结果, 使用 Matlab 绘制相应的图表进行分析, 结果如图7所示.

图7显示出录音服务器中同时录音的请求数与其响应时间的关系, 由图可得请求数在900以下, 采用一致性哈希算法和 Kafka 技术的响应时间与采用普通哈希算法的响应时间基本一致, 但随着同时录音请求数的增多, 这两种方案响应时间的差距开始变大, 当录音请求数达到1300时, 采用一致性哈希算法和 Kafka 技术的录音服务器明显要好于另一种方案, 说明本设计的录音服务器对于高并发的处理能力要强于一般服务器, 其负载调度也更为合理.

图8显示出分布式录音服务器系统中服务器个数对整个系统吞吐量的关系, 由图可得服务器个数在4个以内, 两种方案的吞吐量基本一致, 当服务器个数大于4时, 随着服务器个数增多两者性能差距越来越明显, 服务器个数为8时, 两者差距最大, 约为14.29%, 说明本设计的录音服务器适合做成大规模的分布式系统, 其系统吞吐量的优势越显著.

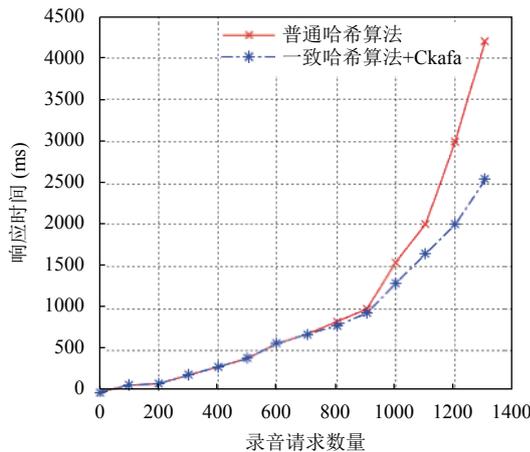


图7 请求数与响应时间关系

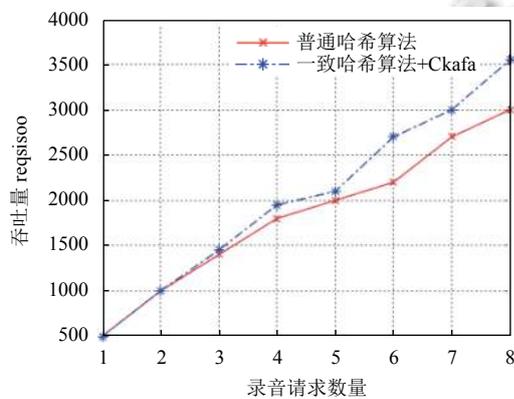


图8 服务器个数与吞吐量的关系

对于服务器集群停机情况, 本文构建了一个包含6个服务器的集群. 当群集正常接收模拟请求时, 突然关闭一个录音服务器, 并观察群集无法正常响应的错误率趋势. 实验结果如图9所示. 模拟错误发生的时间在1s时刻, 从图中可以看出, 两个方案在1s到3s的错误率处于上升阶段, 基本上没有差别. 然而, 在3s到5s内, 采用普通哈希算法的服务器集群错误率几乎不变, 但采用一致性哈希算法+Ckafa技术的录音服务器的错误率显著降低. 5s后, 两个方案的错误率开始下降, 直到错误率为0%. 因此, 可以得出结论: 当集群系统有部分服务器停机时, 采用一致性哈希算法和Ckafa技术的录音服务器可以在一定程度上降低集群错误率.

当通话结束, 录音服务器取时间或集将主叫和被叫通话的语音进行混音, 并保存为PCM格式, 随后立即将语音消息推送至PC/Web终端. 利用通用定时器,

记录从通话结束开始到PC/Web终端收到语音消息所用的最大时延, 并控制录音时间相同的通话个数, 绘制服务器同时处理通话数与终端收到推送的最大时延的关系图, 如图10所示. 由图分析可得: 当通话数小于150时, 两种方案基本保持一致, 当通话数大于150时, 差距开始拉大, 并且随着通话数的增多, 差距越来越大. 当通话数达到350时, 基于一致性哈希算法和Ckafa技术的录音服务器的最大时延为600多毫秒, 而另一种方案的最大时延已经远超过1s, 实时性得不到保证. 通过以上分析, 再结合图7录音请求数和响应时间的关系, 可以得出结论: 相比于传统电话终端录音的推送慢, 查找难, 不便捷的缺点, 本系统在一定通话数下的响应、处理、推送时间可以控制在1s以内, 突显出其实时性强的特点, 又因为是集中式录音, 不存在终端修改录音的情况, 其安全性也得到了保障.

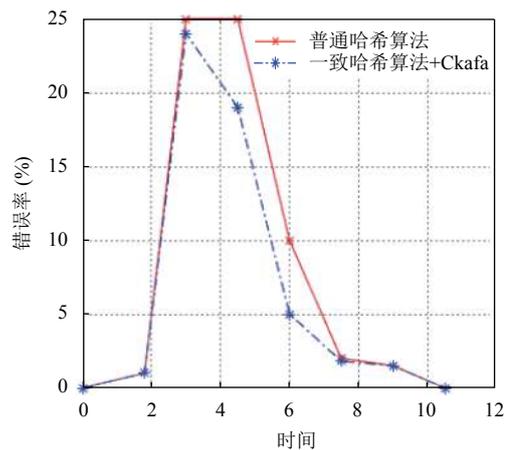


图9 服务器发生错误持续时间与错误率关系

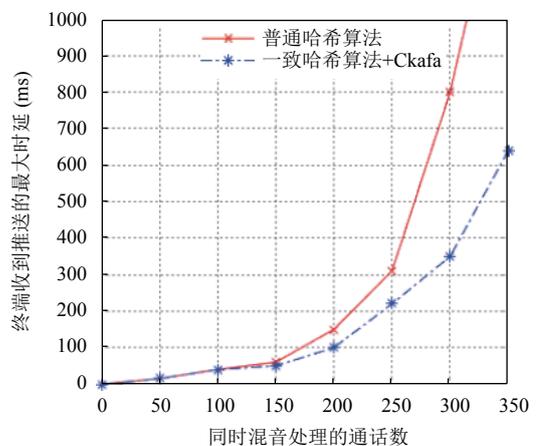


图10 同时处理通话数与最大时延的关系

4 结论与展望

本文通过分析电力IMS交换网中电话终端的实时录音业务需求,借助基于IMS核心交换机的端口镜像技术、一致性哈希算法和Ckafka高性能数据缓存技术,设计出一种基于电力IMS交换网的即时录音系统.该即时录音系统为旁路系统,实现静默录音,不会对生产系统产生影响,可以为电力IMS交换网的IP终端、POTS终端及软件终端(PC, Web)等各类终端提供即时、无差异的录音服务^[17],并通过系统后台将录音记录即时推送至软件终端,以便IMS终端用户随时查阅录音.文章最后,将基于一致性哈希算法和Ckafka技术的服务器与基于普通哈希算法的服务器就响应时间、吞吐量、容错能力和推送的最大时延这四项指标对其进行比较,分析得出,本文设计的录音服务器具有高并发的数据处理能力,高吞吐量,响应时间短,容错力和实时性强并能实现负载均衡.此外本文的算法还可用于大规模分布式服务器的查找,分布式存储等领域^[18].

参考文献

- 1 马文学,王龙龙,戎烁,等. IMS网络业务触发体系架构的研究. 计算机与网络, 2019, 45(15): 69–71. [doi: 10.3969/j.issn.1008-1739.2019.15.054]
- 2 许苏明,王忠民. SIP协议及其应用. 世界电信, 2002, (10): 45–48.
- 3 梁东杰. 计算机通信网安全协议的分析. 通讯世界, 2019, 26(5): 133–134. [doi: 10.3969/j.issn.1006-4222.2019.05.086]
- 4 王计艳,李赞,董勋,等. 核心网未来网络架构演进. 电信科学, 2015, 31(S1): 140–147.
- 5 陈子怡. 基于C语言的计算机编程技术研究. 电脑编程技巧与维护, 2018, 401(11): 63–64, 83. [doi: 10.3969/j.issn.1006-4052.2018.11.024]
- 6 Ji ZL, Ganchev I, O'Droma M, *et al.* A distributed Redis framework for use in the UCWW. Proceedings of 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. Shanghai, China. 2014. 241–244.
- 7 Bhole RH, Chapté VM, Karve AC. A study of apache Kafka in big data stream processing. Proceedings of 2018 International Conference on Information, Communication, Engineering and Technology. Pune, India. 2018. 110–113.
- 8 李浩杰,杜军威,朱桂新. 基于分布式搜索引擎的消息中间件设计. 青岛科技大学学报(自然科学版), 2016, 37(1): 102–107.
- 9 Sanchez VAB, Kim W, Eom Y, *et al.* EclipseMR: Distributed and parallel task processing with consistent hashing. Proceedings of 2007 IEEE International Conference on Cluster Computing. Honolulu, HI, USA. 2017. 322–332.
- 10 Thar K, Ullah S, Hong CS. Consistent hashing based cooperative caching and forwarding in content centric network. Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium. Hsinchu, China. 2014. 176–179.
- 11 Hong T, Wu YT, Cao BY, *et al.* A dynamic data allocation method with improved load-balancing for cloud storage system. IET International Conference on Smart and Sustainable City 2013 (ICSSC 2013). Shanghai, China. 2013. 220–225.
- 12 缪其勇. 分布式radius系统高可用负载均衡算法的设计与实现. 电子测试, 2018, (15): 68–69. [doi: 10.3969/j.issn.1000-8519.2018.15.028]
- 13 Hsiao HC, Chang CW. A symmetric load balancing algorithm with performance guarantees for distributed hash tables. IEEE Transactions on Computers, 2013, 62(4): 662–675. [doi: 10.1109/TC.2012.13]
- 14 Rieche S, Petrak L, Wehrle K. A thermal-dissipation-based approach for balancing data load in distributed hash tables. Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks. Tampa, FL, USA. 2004. 742–751.
- 15 Wang XM, Loguinov D. Load-balancing performance of consistent hashing: Asymptotic analysis of random node join. IEEE/ACM Transactions on Networking (TON), 2007, 15(4): 892–905. [doi: 10.1109/TNET.2007.893881]
- 16 Liu Q, Cai WD, Shen J, *et al.* VPCH: A consistent hashing algorithm for better load balancing in a Hadoop environment. Proceedings of the 3rd International Conference on Advanced Cloud and Big Data. Yangzhou, China. 2015. 69–72.
- 17 陈小惠,彭世新,卜宪德. IMS行政交换网集中录音系统的设计与实现. 电力信息与通信技术, 2016, 14(11): 78–82.
- 18 Zhang GX, Zhai CJ, Wang XY. Research of distributed data optimization storage and statistical method in the environment of big data. Proceedings of 2017 International Conference on Smart Grid and Electrical Automation (ICSGEA). Changsha, China. 2017. 612–617.