大规模软件系统日志汇集服务平台设计与实现

汤网祥 王金华 赫凌俊 李敏敬
(中国电子科技集团公司第三十二研究所 上海 201808)

摘 要 针对大规模软件系统中日志的采集、汇集与存储提出一种实现方案,可以为软件研发、系统运维、数据分析提供支撑。该方案对各类源日志进行整合,并通过本地采集、网络汇集、集中管控、异步分级处理等方式,解决了系统大规模部署下日志采集配置复杂,难以集中控制的问题。针对性地设计一种集群调度算法,提高系统的负载能力和稳定性。

关键词 日志服务平台 日志采集 集群调度

中图分类号 TP3 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2018.11.029

DESIGN AND IMPLEMENTATION OF LOG COLLECTION SERVICE PLATFORM FOR LARGE-SCALE SOFTWARE SYSTEM

Tang Wangxiang Wang Jinhua He Lingjun Li Minjing
(*The 32nd Research Institute of China Electronics Technology Group Corporation, Shanghai 201808, China*)

Abstract In this paper, an implementation scheme was proposed for the collection, gathering and storage of logs in large-scale software system, which could provide support for software research and development, system maintenance, and data analysis. This scheme integrated various kinds of source logs. By means of local collection, network gathering, centralized control and asynchronous hierarchical processing, the problem of complex configuration and difficult centralized control of log collection was solved under large-scale deployment of the system. A cluster scheduling algorithm was designed to improve the load capacity and stability of the system.

Keywords Log service platform Log collection Cluster scheduling

0 引 言

随着大数据技术发展、高速互联网以及 DevOps[1] 的兴起,大规模软件[2]运行日志收集及维护系统得到快速的发展。互联网界兴起了一些专门针对日志提供存储和分析服务的公司。这类公司搭建日志收集及分析平台,为用户提供各类数据采集的接口,让用户自行配置日志源或提交日志文件,提供针对日志内容的分析和查询能力,可以给软件运维带来便利,也可以基于日志中获得更深的业务价值[3-4]。

这类系统是需要大量应用人员参与的系统,大家各自管理维护自己的数据。但对一些专用的大规模复杂业务软件系统来说,这类日志收集分析系统的数据采集方式难以提供采集和分析服务。应用软件量大,部署范围广,软件交互非常复杂,机器软件组合成千上万级。除了服务端以外,还需要采集终端应用的日志,现有的日志采集模式下需要大量的人力和时间去维护采集源,同时也需要相当多的技术人员来进行数据分析和系统维护。在一些封闭性要求比较高、使用人员技术能力较弱、维护人员少的环境下,这种数据接入方式难以使用。另外,很多软件对自己的业务操作都有审计的需要,都需要针对这些信息设计并实现一套业务日志系统以供事后审计,在一个大型系统中造成重复设计和实现。同时,在这类环境下,对重要日志的存储时间是较长的,但日志数据的存储空间和计算资源是受限的,无法做到互联网界商业模式下的存储与计算资源的大量供应。

收稿日期:2018-06-04。**汤网祥**,工程师,主研领域:系统架构,数据挖掘,大数据分析。**王金华**,高工。**赫凌俊,工程师。李敏敬**,高工。

本文针对这类大规模复杂软件系统的日志采集和维护需求,结合当前的大数据存储分析技术和分布式索引技术,设计了一套日志汇集服务平台,提供基于集中控制模式的全系统日志数据采集与汇集能力,提供数据分析与查询展现能力,为软件研发过程和运维过程提供全过程支撑。

1 相关技术

本文的日志汇集服务平台着眼于全局控制体系,在此基础上集成优秀的开源产品,主要包括 Flume Agent、Kafka、HBase、Elastic Search 等。Flume Agent 是 Flume[5] 的一个数据采集组件,可以提供丰富的数据资源采集能力;Kafka[6] 是一个分布式的消息系统,便于横向扩展,吞吐量高,可以保证消息的有序性和可靠性,可以为高速流数据的处理提供支持;HBase[7] 是一个高性能、可扩展的分布式列式数据库存储系统;Elastic Search 是一个准实时的分布式搜索引擎,可以支持所有类型文档的搜索。通过 HBase 与 Elastic Search 的整合[8],构建二级索引,可以处理 PB 级的数据,同时保持很高的处理性能。

2 软件设计方案

2.1 整体框架

本文的大规模软件系统日志汇集服务平台,后续简称为日志服务平台,其整体框架如图 1 所示。

图 1 日志服务整体框架图

日志服务平台包含日志采集、日志汇集与存储两部分。基于分布式存储、分布式列式数据库、分布式索引服务、分布式消息服务等提供日志的采集服务和日志汇集与存储服务,日志采集提供各类终端和服务器日志数据的采集能力;日志汇集与存储提供汇集的集中控制功能,以及控制规则的数据定向汇集能力,涵盖日志开发、收集、存储、应用、销毁的全生命周期服务,为基于日志的各类数据分析提供数据支撑。

2.2 日志采集

本文日志服务平台日志采集包含两类方式:专用日志采集、通用日志采集,数据处理流程如图 2 所示。专用日志采集,通过扩展开发包进行采集,多用于终端日志采集和新建系统的日志采集,支持采集普通日志和业务日志,采集内容更加丰富;通用日志采集,是对现有软件系统输出日志数据的直接采集,不影响现有软件系统运行,支持现有系统的多种数据源,包括文件、系统事件等[9-10]。两种方式采集出的数据基于统一的标准格化式后进行缓存,也可以基于软件选择全解析,以支持本地信息复杂搜索、展现,方便开发过程问题快速定位。

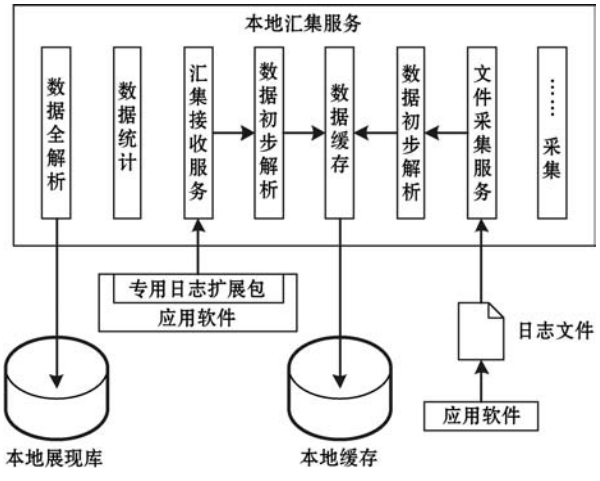

图 2 日志采集数据处理流程图

专用日志扩展包针对现有常用日志开发包进行扩展,保留现有开发接口,增加业务类日志接口,记录业务操作信息,统一内部日志级别定义,标准化内部数据格式,提供本地端口可靠提交能力。采用异步传输机制,降低日志网络传输对属主应用性能的影响,升级配置文件,通过配置文件的方式注入软件日志信息。例如对于 Java 类应用程序,通过替换开发包,简单修改配置文件,就可以实现日志的统一接入。

日志数据直接采集针对不便于进行开发包替换的应用或系统的现有日志数据提供专用的日志数据采集能力,可以监控这些应用或系统的日志输出,自动采集新产生的日志数据,进行解析处理,并使用标准化格式

存入本地缓存系统。这一部分是目前商用日志系统进行采集的主要方式,可以有选择地进行集成,融合进本文日志采集体系。该部分采集参数的配置提供本地可视化手段,并将配置控制纳入远程集中控制体系,开源软件 Flume,基于 Java 开发,具有很好的平台兼容性,是不错的整合选择。

本地汇集服务提供本地默认采集端口服务,收集本地开发包提交的日志对收集到的数据进行初步解析处理,使用标准化格式存入本地缓存系统,并提供基于日志级别和特征值的数据统计,支撑软件性能数据的监控和软件运行问题的快速发现。同时可以减少大量日志内容的提交,降低网络和服务端资源的开销。在软件研发过程中,结合选择性的全解析能力,通过缓存查询工具查看应用输出的日志,进行日志过滤查看,快速获取日志内容,进行日志分析,定位软件问题。既可以辅助规范化日志的输出,也可以辅助研发人员确定需要输出哪些信息,有助于软件运维过程中问题的定位,支撑实际运行过程中基于日志的软件问题排查定位和操作审计。

日志采集需要有广泛的兼容性,本地汇集服务会面对各类复杂的使用场景,需要有很高的兼容能力:操作系统的兼容,物理平台的兼容,独立应用输出与插件输出的兼容,扩展开发包输出信息和现有数据源直接采集信息的兼容,以及内部定义有差异的不同日志开发包进行扩展时的兼容。本日志服务平台为各类接入的日志源定义了统一的日志级别,统一的数据格式,对不同的原始定义进行了调整和整合。

日志级别相关规范信息如表 1 所示。

表 1　日志级别定义表

序号	日志级别定义			是否有接口
	原始	规范	统一定义值	
1	—	all	0	无
2	trace	debug	100	有
3	debug	debug	100	有
4	info	info	200	有
5	warn	warn	300	有
6	error	error	400	有
7	fatal	fatal	500	有
8	—	undef	1 000	无

用于集中存储的统一规范的日志信息如表 2 所示。

表 2　主要日志属性定义表

序号	标识	描述
1	Id	唯一标识(UUID)
2	IPAddr	Ip 地址
3	SoftSymbol	统一软件标识
4	Level	日志级别
5	Content	日志内容
6	TraceInfo	异常信息
7	LogTime	日志时间
8	InputTime	入库时间
9	BusinessSymbol	操作日志标识
10	Operator	操作者
11	BusinessObject	业务对象
12	OperationType	操作类型
13	⋮	⋮

很多软件的日志输出速度非常高,本地汇集要能对这类数据有相应的承载能力或应对能力。本文的汇集服务对数据的接收和处理过程使用内存处理模式,通过管道过滤器样式进行设计,结合多线程并行模式,充分利用 CPU 的处理能力,大大提高数据处理速度。设计图如图 3 所示。

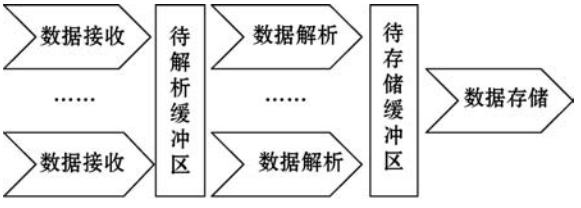

图 3　本地数据接收与处理设计图

本地汇集服务需要具备磁盘使用控制能力,数据采集持续运转,同时又要在本地保留一定量的存储,支持本地的日志快速查询,缓存数据的存储和管控模式是关键。本文通过循环数据表的方式进行数据存储,支持快速复杂查询,通过循环存储模式结合表定量模式,进行本地缓存数据量的磁盘空间控制。控制图如图 4 所示。

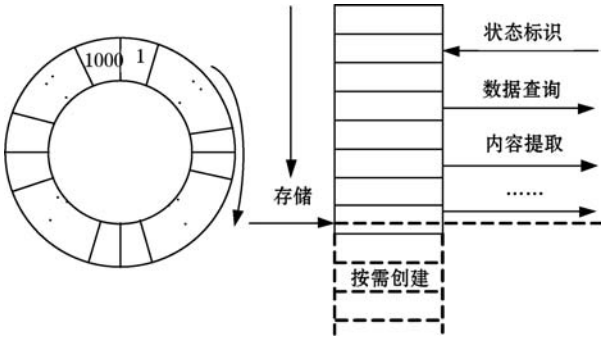

图 4　日志本地缓存控制图

2.3 日志汇集与存储

本文日志服务平台的日志汇集提供日志汇集控制系统,可对汇集进行全方位控制;采用分级汇集机制,降低系统资源最低需求,适应各类应用场景;采用异步处理模式,应对数据潮涌现象,同时可以提高业务处理速度;提供自主集群机制,分摊网络带宽和数据处理压力,降低系统资源需求,同时提高系统的可用性;根据时间段分库存储,控制常规运行态所接触的数据空间,保持全系统持续运行下的系统性能的稳定性,不会因为数据持续积累导致系统的运转速度的降低,使系统永远保持年轻。功能设计如图 5 所示。

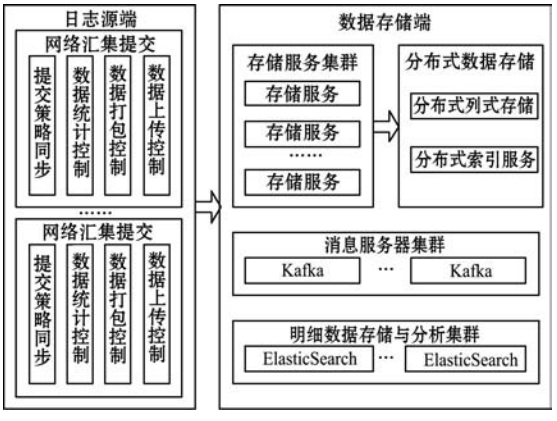

图 5 日志网络汇集与存储图

1)汇集管控:本文日志服务提供日志内容汇集控制系统,能够对提交系统进行精确控制,可以控制是否要提交,全部提交还是指定某些计算机提交;可以指定提交的时间范围,全时段提交,还是指定时间段提交,支持指定跨天时间段提交;可以指定提交内容,能够指定汇集信息的计算机、软件、时间段、日志级别,种类等要素,筛选或提取指定的日志内容;可以指定提交格式和提交目的地,除了处理全局数据的数据存储服务集群之外,还可以为消息服务器集群、明细数据存储与分析集群提交数据,为准实时数据分析用、交互数据分析等各类数据分析提供全局管理支撑。另外,提供终端统计数据汇集控制机制,可以设置统计内容模式、统计间隔等,为无日志内容汇集下的软件性能分析提供支撑。

2)数据调度与载入:通过集中控制的汇集管控管理功能构建分析用数据调度策略,确定数据传输通道和缓存空间构建机制,可以获得分析任务所需的数据内容。将历史数据和实时数据按照适当的频率调度到指定的数据缓存空间,为各类日志分析系统[11]提供数据支撑,包括准实时数据分析[12-13]、交互式数据分析[14]、批量数据分析[15]。目前常用的分析用数据缓存技术有 hive、Kafka、Elastic Search 等。这些技术可以映射现有数据,或者是将数据进行分类且时序化存储,或者提供数据索引,针对不同的数据分析需求采用不同的技术手段。调度与载入数据流图如图 6 所示。

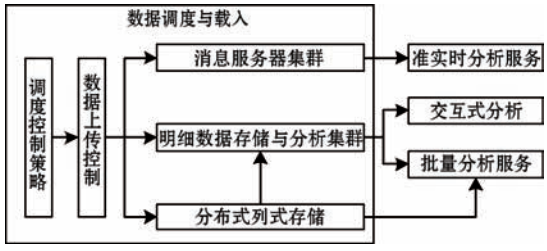

图 6 调度与载入数据流图

3)异步分级汇集机制:日志服务是属于辅助型系统,不应该对业务系统有太大的影响,在必要时还需要能够自动暂停部分能力为业务让步。因此,日志数据在本地汇集和网络集中汇集过程需要使用异步缓冲机制,降低系统前端的异常状态对后续处理的影响,以及系统后端流程对前端业务的影响,例如经常会出现的日志数据潮涌现象以及网络断连现象。本文日志服务平台中比较典型的需要使用异步机制的地方有:开发包提交、本地汇集缓存、网络汇集存储等,通过异步机制,不但可以化解异常的传递,保证系统的平稳运行,还便于引入更多计算资源,提高系统的整体性能。日志服务平台会有保存较长时间日志的能力,但持续运行情况下中往往会面临配给的集中存储资源不足的问题,难以提供全部汇集所需相应的存储空间,数据量庞大,后续处理系统无法及时应对。因此需要将日志数据分级处理,近期的全量日志保留在客户端,只提交统计量和部分需要的数据提交到服务端,需要更详细的日志数据时通过构建专用提取策略进行提取。比如:为了发现软件问题,默认只提交警告级以上日志,详细的日志后续再通过专用策略进行提取;为了业务分析,只提交操作类日志数据等。异步分级汇集数据流程如图 7 所示。

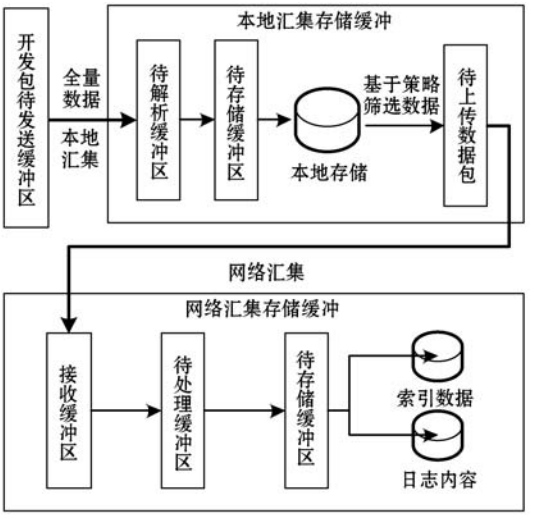

图 7 调度与载入数据流图

4）集群扩展能力:为处理全局数据存储的存储服务提供集群化服务能力,多个服务处理节点能够自行组合为服务集群,为全部数据接入点提供汇集服务,分摊数据处理压力。基于自主集群机制,可以通过横向扩展的方式增加数据处理能力,提高系统的整体性能;提供集群自动调度能力,能够根据各服务节点的负载量、可用计算资源、处理速度、汇集速度,集中控制等因素自动调整数据汇集流向;充分利用计算资源,降低数据积压,保障数据处理的及时性;能自动处理服务节点的接入和断开,基于最小变化原则重新调整数据汇集方向,进行保障系统的平稳运行。

集群调度算法主流程如图 8 所示。

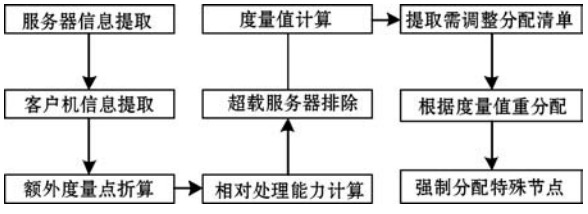

图 8　集群调度主要处理流程图

集群调度处理流程算法明细如下:

（1）提取活跃服务器清单数据、服务器处理能力数据、服务器积压量数据,以及服务器承担功能模块清单;

（2）提取活跃客户机清单以及历史平均提交速度数据;

（3）将积压量折算为处理能力数据,将功能模块折算为处理能力数据;

（4）通过原始处理能力数据减去积压量、功能模块等折算的处理能力,获得最终用于计算的各服务器的相对处理能力;

（5）个别服务器因承担过重,被排除出待分配的服务器列表,最终形成用于调度的服务器清单;

（6）根据总相对处理能力和总提交量进行对比换算,形成各服务器的处理能力度量值,总度量值的和略超出提交量,各服务器的度量值与相对处理能力等比;

（7）通过各服务器根据度量值进行对当前服务器分配进行调整,优先保留已有的提交量较大的终端,如果终端提交量超出了度量值,多出的终端放入统一调度清单,如果服务器当前负载不足,会形成负载空缺,留待统一分配;

（8）将统一调度清单中的终端根据提交量和服务器负载空缺量进行分配,也是优先将提交量大的向负载空缺量大的服务器进行分配;

（9）最后剩余的个别无法分配出去的终端根据提交量大小和空缺大小进行强制分配,大的配大的,不考虑度量值超量问题。

该算法虽然无法每次调度都做到绝对的负载均衡,但它的分配结果要比单纯的数据包个数、机器个数、数据量等的简单调度模式要合理很多,而且计算逻辑简单,处理速度快,各类参考数据都是在随着系统动态形成、持续变化,系统在持续运行中会形成一个动态的负载均衡。

5）分片存储:日志数据具有天然的时效性,数据的使用也是基于时间的维度。日志的数据量相当庞大,即便是大数据平台,超量的数据放在一张表中,数据查询也会占用掉较多的时间,而且数据越旧,使用的机率就越低。为保证系统的持续高效运行,不会因为时间的关系变得缓慢,在使用分布式列式数据库提高大数量数据存储和管理性能的基础之上,采用按月为粒度的分表存储。各条日志数据都存入对应时间段的库中,永远保持系统常规处理数据量的有限态。日志数据内容具有较大的重复性,使用数据压缩存储的方式可以有效减少存储空间;结合分布式列式数据库数据分块存储方式和数据的使用模式,优化 rowkey 的构建模型,提高数据使用效率。采用分布式索引系统对数据内容进行准实时索引,提高数据即时查询能力,通过索引和数据内容的映射关系,索引采用同样时间粒度的分库机制,提高近期数据的处理效率,保持系统性能的稳定型。

6）系统运行监控:对系统各部分的运行状态数据进行采集,并通过内部汇集机制进行汇总,提供可视化的图表展现方式对内部运行状态数据进行展现。被监控数据主要有终端接收速度、处理速度、存储速度、上存速度、缓存量、内部各服务状态、服务端接收速度、积压量、处理速度、存储速度、内部各服务运行状态等。运行监控系统可以对系统运行状态进行监控,故障排查,并辅助系统性能调优。

3　结　语

本文针对大规模软件系统中日志的采集、汇集存储和分析提出了全套的设计方案。覆盖日志的全生命周期,可以对全系统的日志采集、调度、存储、查看、分析进行统一管控,解决系统大规模部署下日志采集配置复杂,难以集中控制的问题,解决有限资源条件下数据处理和长时间存储资源不足的问题。并针对众多软件日志集中管控的特点,提出了几点日志分析方向,发掘日志的深入价值。

本文设计的日志服务平台目前已经完成了日志数据的采集、调度和存储体系。支持基于集中控制的策

略将所需的日志数据调度到指定的位置进行存储,提供日志统计分析能力和数据内容查看能力。可以纵观系统运行状态,发现哪些软件什么时间段出现问题,可以提取和查看详细日志内容,辅助研发人员进行故障诊断。图 9 所示是日志统计分析纵览图,可以查看指定月、天、时的运行分析结果,结合折线点的明细数据,确定异常日志来源于哪个软件,分析软件运行趋势。

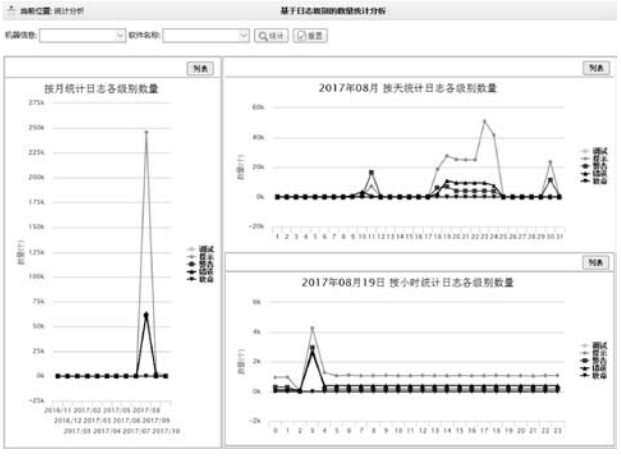

图 9 统计分析结果展示图

本文下一步的工作是丰富日志数据分析能力。结合二次解析、深度学习、语义分析等手段从日志数据中发现更多的价值,为全系统运维提供强大且智能化的支撑。

参 考 文 献

[1] 牛晓玲,吴蕾. DevOps 发展现状研究[J]. 电信网技术,2017(10):48 - 51.

[2] 廖湘科, 李姗姗, 董威,等. 大规模软件系统日志研究综述[J]. 软件学报, 2016, 27(8):1934 - 1947.

[3] 薛亚鹏. 基于实时日志系统的海量日志服务平台设计与实现[D]. 北京:中国科学院大学(中国科学院工程管理与信息技术学院),2017.

[4] 阮厦城. 分布式环境下通用日志系统的设计与实现[D]. 哈尔滨:哈尔滨工业大学,2015.

[5] 周敏菲. 基于 Kafka 和 Storm 的实时日志流处理系统的设计与实现[D]. 贵阳:贵州大学,2017.

[6] 周波. 一种基于 Flume 的海量数据分流方案[J]. 电信科学,2016,32(S1):220 - 225.

[7] 康毅. HBase 大对象存储方案的设计与实现[D]. 南京:南京大学,2013.

[8] 杜忠晖. 非结构化文档数据一体化存储检索技术研究[D]. 哈尔滨:哈尔滨工业大学,2015.

[9] 陈健峰,寇从芝. 一种日志采集统计系统的设计与实现[J]. 电脑编程技巧与维,2017(12):21 - 23.

[10] 刘必雄. 自适应日志采集间隔时间动态调整算法研究[J]. 重庆科技学院学报(自然科学版),2017,19(2):92 - 95.

[11] 许长福. 日志数据分析系统的设计与实现[D]. 北京:北京交通大学,2017.

[12] 陆世鹏. 基于 Spark Streaming 的海量日志实时处理系统的设计[J]. 电子产品可靠性与环境试验,2017,35(5):71 - 76.

[13] 胡聪, 刘翠玲, 吴尚. 基于大数据日志的预警技术分析[J]. 电气技术, 2017, 18(6):95 - 98.

[14] 刘昕林,张华兵,张海涛. 日志搜索分析管理系统的研究与应用[J]. 信息与电脑(理论版),2017(9):81 - 82.

[15] 顾兆军,王帅卿,张礼哲. 多源日志聚合分析方法[J]. 计算机工程与设,2017,38(7):1702 - 1708.

(上接第 121 页)

[14] Tan Y, Wu J, Deng H. Rapid identifying high-influence nodes in complex networks[J]. Syst. Eng. Theory, 2006, 11:79 - 85.

[15] Freeman L C. Centrality in social networks conceptual clarification[J]. Social Networks, 1978,1(3):215 - 239.

[16] Kitsak M, Gallos L K, Havlin S, et al. Identification of influential spreaders in complex networks[J]. Nature Physics, 2010,6(11):888 - 893.

[17] Altmann M. Reinterpreting network measures for models of disease transmission[J]. Social Networks, 1993,15(1):1 - 17.

[18] Estrada E, Rodríguez-Velázquez J A. Subgraph centrality in complex networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2005, 71(5 Pt 2):056103.

[19] 任晓龙, 吕琳媛. 网络重要节点排序方法综述[J]. 科学通报, 2014, 59(13):1175 - 1197.

[20] Lü L, Zhang Y C, Chi H Y, et al. Leaders in Social Networks, the Delicious Case[J]. Plos One, 2011, 6(6): e21202.

[21] Ma N, Guan J, Zhao Y. Bringing PageRank to the citation analysis[J]. Information Processing & Management, 2008, 44(2):800 - 810.

[22] Peng X L, Xu X J, Fu X, et al. Vaccination intervention on epidemic dynamics in networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2013, 87 (2): 022813.

[23] Brummitt C D, D'Souza R M, Leicht E A. Suppressing cascades of load in interdependent networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2012,109(12):4345 - 4346.

[24] Lü L, Chen D B, Zhou T. Small world yields the most effective information spreading[J]. New Journal of Physics, 2011,13(12):825 - 834.