存储中的副本分级存储调度策略

杨冬菊 李 青

(北方工业大学云计算研究中心 北京 100144)

(大规模流数据集成与分析技术北京市重点实验室 北京 100144)

摘 要 当集群中的部分节点是廉价主机时,采用HDFS的随机存储策略可能使访问频率高的数据存储在廉价节点上,受到廉价节点的性能影响,访问时间过长,降低了集群效率。为改善以上问题,提出一种改进的副本分级存储调度策略。为减少副本调度的次数,先根据节点的CPU、内存、网络、存储负载以及网络距离来评价节点的性能,再从中选取高性能节点进行存储。副本调度以节点中副本的访问频率为依据,结合硬件配置,把访问频率高的副本尽可能存储在高性能、高配置的节点中,以加快集群响应速度。实验结果表明,改进后的策略可以在异构集群中提高副本的访问效率,优化负载均衡。

关键词 云存储,HDFS,分级存储,副本调度

中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.04.019

Scheduling Strategy of Hierarchical Storage about Replication in Cloud Storage

YANG Dong-ju LI Qing

(Research Center for Cloud Computing, North China University of Technology, Beijing 100144, China)

(Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data, Beijing 100144, China)

Abstract HDFS takes random storage strategy, if cluster has some cheap nodes, it is possible to make high frequency data store in the low processing performance nodes, causing a long time access and poor efficiency. To solve these problems, an improved scheduling strategy of hierarchical storage about replication was proposed. In order to reduce the number of replication scheduling, firstly, the information of data node from CPU load, memory load, network load, storage load and network distance are used to evaluate node availability. Secondly, the optimal one is selected. Accessing frequency and hardware configuration are used to realize the replication scheduling. The response rate of cluster is improved by making high frequency data store on the high processing performance and high configuration node. The experimental results show that the strategy can improve access efficiency of replicas and local balancing for data storage in the heterogeneous clusters.

Keywords Cloud storage, HDFS, Hierarchical storage, Replication scheduling

1 引言

云计算技术[1]作为一种新兴的技术受到了各行各业的强烈关注,而云存储作为从云计算概念延伸和发展出来的一个新概念,其改变了传统存储提供的模式。其中,最著名的开源分布式系统之一就是Hadoop[2]。

HDFS[3](Hadoop Distributed File System)是Hadoop体系中数据存储的基础。它是一个高度容错、健壮的、低成本的分布式文件系统。HDFS简化了文件的一致性模型,通过流式方法提高了数据访问的吞吐量,尤其适合带有大量数据集的应用程序[4]。

由于HDFS的容错性允许集群中既存在廉价主机作为数据节点,又存在高配置主机作为数据节点,因此会导致集群中的节点性能存在较大差异性,这些差异主要体现在CPU和内存性能、网卡和磁盘I/O速度等方面。

HDFS使用随机策略存储数据,虽然有速度快和保障数据永久性的优点,但是在性能差异较大的集群中会引起一些问题。一方面,随机性可能会使某个节点负载过重,特别是当集群内节点性能相差较大时,如果性能较次的节点负载过重,则会降低集群运行的速度;集群中节点的性能是由硬件配置、CPU和内存使用率、网络带宽、存储负载和网络距离等因素共同决定的,其中硬件配置包含CPU、内存、网卡、磁盘容量

到稿日期:2015-11-30 返修日期:2016-03-05 本文受北京市教育委员会科技计划重点项目:支持数据资源联动的云服务社区研究(KZ201310009009),北京市属高等学校创新团队建设与教师职业发展计划基金资助项目(IDHT20130502)资助。

杨冬菊(1975-),女,博士,副研究员,主要研究方向为服务计算、行业资源中心关键技术、创建模式及其在领域中的示范应用,E-mail:yangdongju@ncut.edu.cn;李 青(1991-),男,硕士生,主要研究方向为云存储。

等参数。另一方面，由于以上因素的限制，当节点差异性较大时，随机策略不能充分发挥高性能、高配置节点的作用。假设访问频率低的数据过多地存储在这样的节点上，导致节点空间被访问频率低的数据占用。在随机策略下，可能迫使高访问率的数据存储到低性能节点上。当访问压力增大时，低性能节点便会超负荷，集群就无法快速响应客户端的访问。这种情况致使高效资源被浪费，无法充分发挥高性能、高配置节点的优势。

因此，为解决由于集群节点性能不同而产生的无法高效利用存储设备的问题，本文在“基于云存储的二阶段动态优化调度机制”[5]的基础上，充分考虑 CPU 性能、存储负载、网络距离和访问效率，提出一种副本调度策略对数据进行分级存储。

本文第 2 节阐述了国内外的一些副本调度策略，并简析其特点；第 3—4 节提出了一种以节点性能和运行状态为依据的副本调度策略，并详述了具体算法和方案；第 5 节对该策略进行了验证；最后总结了此种方案的优缺点。

2 相关工作

HDFS 由于自身的特点而允许被部署在廉价的集群上，其健壮性能够应对由低成本硬件带来的设备故障，因此如何在廉价集群中提高性能成为了一个亟待解决的问题，而副本策略的研究能够极大地提升集群性能。针对副本调度的研究主要包含两个方面：存储优化和副本调度优化。

对于存储优化的方案，文献[6]提出了一种副本策略，该策略存放数据时会根据节点性能来动态选择最优存放节点，其基于灰色预测技术的放置策略来动态调整副本的数目。但是，集群中的高性能节点会因算法最优解导致其负载非常严重，而性能相对较低的节点不能被充分利用，从而降低了集群存储资源的利用率。文献[7]提出了一种能适应节点数量动态变化的一致性 Hash 算法，但是这种算法只适用于数据节点同构的情况，当每个节点的存储空间和处理能力相差较大时，数据就不能均匀地分布到集群中，导致一些节点的负载过大，集群会呈现负载不均的状况。

对于副本调度优化的方案，文献[8]根据计算节点的处理能力并按照比例来存放数据，这种策略虽然考虑到了节点的异构性，改进了异构集群的性能，但是在副本拷贝时，可能会由于网络距离或各个数据节点的负载而导致传输过慢或者集群负载不均衡。文献[9]提出了一种适用于分布式系统的副本放置机制，它根据数据块的访问频率和存储容量选择相应的副本放置机制，可以降低数据的平均访问时间；其缺点在于无法在数据上传时选择一种方案来减小后期副本调度的代价。除此之外，文献[5]提出了一种二阶段动态优化调度机制，通过动态调度监控器获取集群存储情况来计算存储比率（存储比率＝剩余存储空间 / 已存数据占用空间），从每个机架上选取最优节点排序，根据副本数选择节点数；其缺点在于只考虑了节点存储情况，未考虑集群中节点的异构性。

综上所述，为了最大限度地利用集群中高、低配置的节点，使高访问率的数据集中在高配置节点，而访问率低的节点尽可能地存储在性能相对较次的节点上，同时在上传数据时能兼顾到后期的副本调度，减小调度开销，并且考虑到集群中节点性能、网络距离、负载、数据访问频率等因素，本文给出了一种基于评价结果的分级存储改进策略。

3 分级存储副本策略

前文提到最好将访问频率高的数据块放到高性能、高配置节点上，但在上传数据前不能确定哪些数据块是访问率高的。因此，本文设计的策略由两部分组成：节点初选方案和副本调整方案，类似于文献[5]中的节点优化选择和副本动态调度。

文献[5]中的方法只考虑了存储负载和副本访问频率，本文在此基础上加入了由 CPU、内存使用率、网络带宽、存储负载以及网络距离组成的评价模型，作为选择节点的判断依据[10]。

节点初选方案通过评价模型和节点配置的高低共同决定哪些节点可用于存储本次上传的数据。节点配置的高低由 0 和 1 代表，通过人工写在配置文件中。当客户端发出上传请求时，名称节点在每个机架上随机选取一定数量的节点，利用评价模型计算结果并排序，选取排序靠前的几个节点用作存储上传数据，并保证这些节点不在同一个机架。当客户端是数据节点时，选择本地节点作为存储的第一个数据节点，如果它是高（低）配置，再选取（$R-1$）个低（高）配置节点，R 为副本数；当客户端不是数据节点时，选取 1 个高配置和（$R-1$）个低配置节点。利用节点初选方案，尽可能减少副本调整方案变动的副本数，减轻集群承受的压力。

副本调整方案是在集群处于闲置状态时由管理员开启，它比较了人工设定的冷热阈值与节点中副本的访问频率，并根据节点的配置情况判断是否移动此副本。当移动副本时，仍然以评价模型的计算结果为依据。其中，副本的访问频率由动态调度监控器[5]负责收集。该调整方案的目的是将访问频率高的数据调度到高配置的节点上，加快副本的访问速度，将访问频率低的数据移出高配置节点，节省存储空间。利用这种分级存储的方法，充分发挥集群中高性能、高配置节点的作用。

3.1 评价模型

在此方案中，对集群节点的性能评价由以下 5 个元素组成：$NEval=\{N_{CPU}, N_{MEM}, N_{NET}, N_{STO}, N_{DIS}\}$。它们分别对应节点 CPU、内存、网络带宽、存储负载和网络距离的评价结果。

定义 1 设剩余存储空间为 $ReSto$，已存储数据量为 $UsedSto$，则存储负载 $RateSto$ 的定义为：

$$RateSto=\frac{ReSto}{UsedSto}$$

定义 2 网络距离用于说明两个节点在网络拓扑中的距离（延用 HDFS 中给出的定义）。在树状网络拓扑结构中，设子节点到父节点的距离为 1，任意两节点间的网络距离是它

们到最近的公共祖先节点的距离之和。

定义 3　设数据节点的 CPU 使用率为 Cpu，内存使用率为 Mem，带宽使用率为 Net，网络距离为 Dis，则节点的评价结果 $NEval$ 为：

$$NEval=W_{Cpu}\times Cpu+W_{Mem}\times Mem+W_{Net}\times Net+W_{Sto}\times RateSto+W_{Dis}\times Dis$$

其中，W_{Cpu}，W_{Mem}，W_{Net}，W_{Sto}，W_{Dis} 变量对应 CPU、内存和带宽使用率、存储负载和网络距离在评价中的权重值。

3.2　节点初选方案

在 HDFS 中，名称节点通过数据节点发送的心跳包来检测其是否正常运行。

在系统运行时，数据节点会把当前的性能信息随 HDFS 心跳包一同发送给名称节点，包括 CPU 及内存使用率、带宽使用率、存储负载和网络距离信息。当名称节点收到客户端的写请求时，会通过定义 3 中的评价模型对这些信息进行分析、排序，得出较优存储节点；同时，考虑到响应时间的因素，算法中仍然使用了一定的随机性，但是把随机选择节点的范围从集群缩小到机架。

3.3　副本调整方案

副本调度方案是在集群中已经存有数据的情况下进行的，名称节点在收集各数据节点中数据的访问频率后根据两个预先设定的热、冷阈值判断数据的访问频率是高还是低。

热(冷)阈值有如下的定义：当数据的访问频率比热(冷)阈值高(低)时，则称其为高(低)访问频率数据。

该方案的目的在于利用两种方法提高集群的性能：1)将高访问频率的副本移动到高配置、高性能的节点上；2)将低访问频率的副本移出高配置、高性能节点。通过这两种手段优化副本响应速度和集群空间利用率，提高集群的性能。在移动副本的同时还要兼顾节点的评价结果，避免负载不均的现象出现。

4　算法描述

4.1　节点初选算法

根据前文的描述，节点初选的流程如图 1 所示。

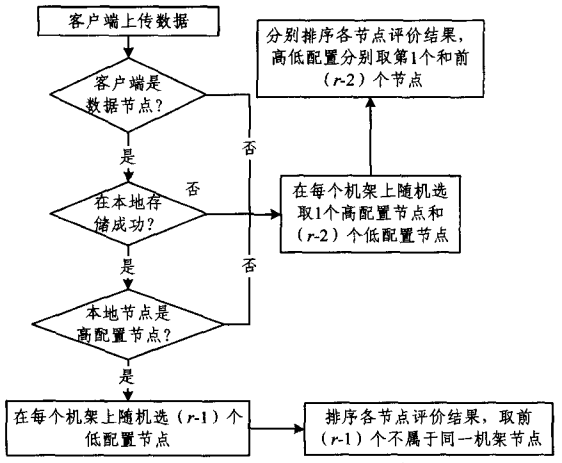

图 1　节点初选流程图

具体的算法描述如算法 1 所示，其中表 1 是对该算法参数的详细描述。

表 1　节点初选算法参数说明

符号	描述
ResultList	结果节点集合
NodeList	候选节点集合
ProList	候选节点评价集合
R	备份因子(副本数)
N	机架数
HighList	高性能节点候选集合
LowList	低性能节点候选集合
HProList	高性能节点候选评价集合
LProList	低性能节点候选评价集合

节点初选算法如算法 1 所示。

算法 1　节点初选算法

输入：R 备份因子，N 机架数
输出：ResultList 被选中存储副本的节点集合
IF 客户端是机架上的节点 node
　node→ResultList；
　IF node 是高性能节点
　　WHILE N 个机架
　　随机选取(R－1)个低配置节点→NodeList；
　　END WHILE
　　WHILE NodeList
　　通过心跳包获取 5 个评价因素；
　　通过 NEval 评价 NodeList 中的节点→ProList；
　　END WHILE
　　正序排序 ProList；
　　取前(R－1)个不属于同一个机架的节点→ResultList；
　　RETURN ResultList；
　END IF
　IF node 是低性能节点
　　WHILE N 个机架
　　随机选取 1 个高性能节点→HighList；
　　随机选取(R－2)个低性能节点→LowList；
　　END WHILE
　　WHILE HighList
　　通过心跳包获取 5 个评价因素；
　　通过 NEval 评价 HighList 中的节点→HProList；
　　END WHILE
　　WHILE LowList
　　通过心跳包获取 5 个评价因素；
　　通过 NEval 评价 LowList 中的节点→LProList；
　　END WHILE
　　正序排序 HProList 和 LProList；
　　取 HProList 中第 1 个节点→ResultList；
　　取 LProList 中前(R－2)个不属于同一机架的节点→ResultList；
　　RETURN ResultList；
　END IF
END IF
IF 客户端不是机架上的节点
　　执行“node 是低性能节点”中的语句；
END IF

4.2　副本调整算法

根据 3.3 节的描述，副本调整算法分为两部分：高配置节点调整和低配置节点调整，其流程分别如图 2 和图 3 所示。

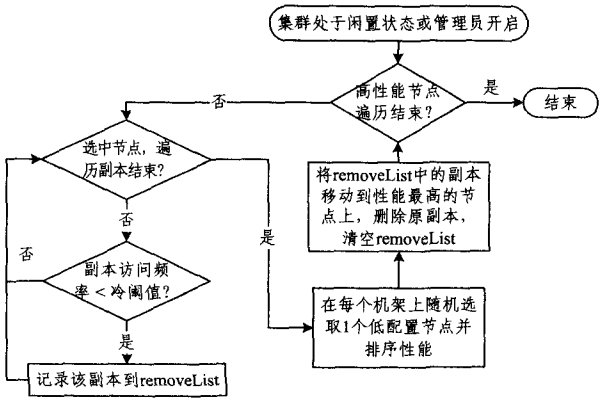

图 2 高配置节点调整

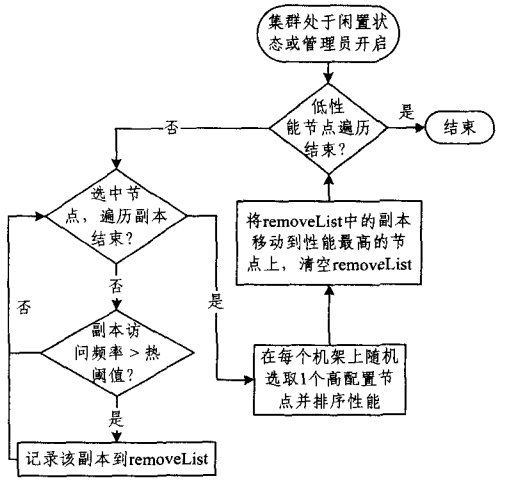

图 3 低配置节点调整

图 2 和图 3 示出了副本调整算法的流程，具体步骤如算法 2 所示，表 2 详细列出了算法中的参数。

表 2 副本调整算法参数说明

符号	描述
F	副本访问频率
lowValue	冷阈值
highValue	热阈值
RemoveList	需要移动的副本
List	候选节点集
ProList	评价结果集
node	选中节点

算法 2 副本调整算法

输入：lowValue 冷阈值，highValue 热阈值

输出：无

```
IF 集群处于闲置状态或管理员开启副本调整
  WHILE 高配置节点 //调整高配置节点
    WHILE 节点副本
      IF F<lowValue
        记录该副本→RemoveList;
      END IF
      END WHILE
      WHILE 所有机架
        随机选取 1 个低配置节点→List;
      END WHILE
      通过 NEval 评价 List 中的节点→ProList;
      取第 1 个节点 node;
      将 RemoveList 中的副本移动到 node;
      删除本地 RemoveList 中的副本;
      清空 RemoveList;
    END WHILE
    WHILE 低配置节点 //调整低配置节点
      WHILE 节点副本
        IF F>highValue
          记录该副本→RemoveList;
        END IF
      END WHILE
      WHILE 所有机架
        随机选取(R-1)个高配置节点→List;
      END WHILE
      通过 NEval 评价 List 中的节点→ProList;
      取第 1 个节点 node;
      将 RemoveList 中的副本移动到 node;
      清空 RemoveList;
    END WHILE
END IF
```

5 实验设计及结果

5.1 实验环境

实验集群中有 4 个机架，每个机架含有数量不同的节点，将高、低性能节点平均分配在每个机架上。表 3 列出了各节点的配置情况，表 4 列出了各机架之间的网络距离。

表 3 节点配置情况

机架	高配置节点	低配置节点	编号
A	1	2	2～4
B	2	2	5～8
C	1	2	9～11
D	1	2	12～14

表 4 机架间网络距离

机架	A	B	C	D
A	2	4	5	5
B	4	2	5	5
C	5	5	2	4
D	5	5	4	2

在此集群上，验证基于评价结果的 HDFS 改进策略的性能，并将其与 HDFS 随机策略进行比较。

5.2 实验内容和结果分析

在本次实验中，图 4 示出了在 HDFS 随机策略和改进策略下上传数据副本的存储对比情况。

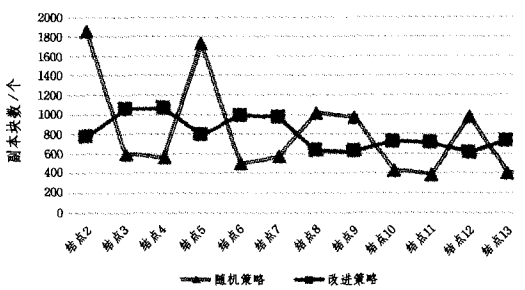

图 4 两种策略上传副本时间的对比图

图 4 显示的是 HDFS 随机策略与改进策略对上传文件

副本存储位置的影响，可以看出，改进策略使低性能节点分担了高性能节点的副本存储数量，但是由于参数限制的原因，网络距离较远的节点存储的副本会少一些。随机策略主要以高性能节点存储文件，改进策略主要利用低配置节点存储访问率较低的副本，而高配置节点存储高访问率的副本。

系统会根据初期设定的冷热阈值在高低性能节点之间调动副本，从高配置节点中移出访问率低于冷阈值的副本到低配置节点，反之同理，从而达到根据设备级别来存储不同访问级别的数据的效果。

经过一段时间的访问，节点上的副本数的变动如图 5 所示。

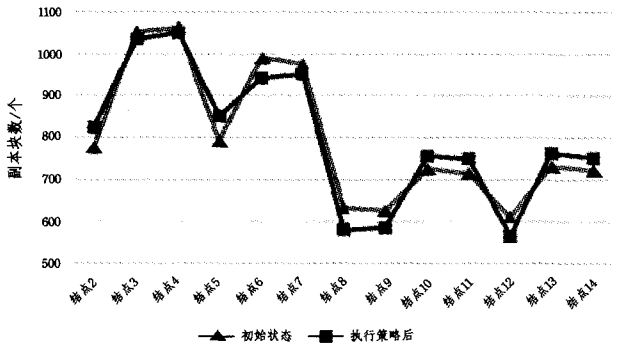

图 5　副本数变化图

结束语　本文针对集群的异构性提出了一种改进方法，根据节点性能、内存负载、网络负载、存储负载和网络距离给出节点评价，基于评价结果优先选择存储节点。由于集群中节点性能的限制，文中提出一种方案来调度访问率超过一定阈值的副本数据，通过调度方案保证高访问量数据的使用效率，但是评价模型中的参数需要经过不断地选取调整才能促使模型达到预期的效果。

参 考 文 献

[1] CHEN K, ZHENG W M. Clouding Computing: System Instances and Current Research[J]. Journal of Software, 2009, 20(5): 1337-1348. (in Chinese)
陈康，郑纬民. 云计算：系统实例与研究现状[J]. 软件学报，2009，20(5)：1337-1348.

[2] Apache Hadoop [OL]. [2013-07-10]. http://hadoop.apache.org.

[3] MILOJICIC D, WOLSKI R. Eucalyptus: delivering a private cloud [J]. Computer, 2011, 44(4): 102-104.

[4] 蔡斌，陈湘萍. Hadoop 技术内幕：深入解析 Hadoop Common 和 HDFS 架构设计与实现原理[M]. 北京：机械工业出版社，2013.

[5] REN C, YANG D J. The two-stage Dynamic Optimized Scheduling Mechanism Based on Cloud Storage[J]. Computer & Digital Engineering, 2014, 42(9): 1553-1557, 1716. (in Chinese)
任川，杨冬菊. 基于云存储的二阶段动态优化调度机制[J]. 计算机与数字工程，2014，42(9)：1553-1557，1716.

[6] TAO Y C, SHI L. Research on Dynamic Management of Data Replicas of Cloud Computing in Heterogeneous Environments [J]. Journal of Chinese Computer Systems, 2013, 34(2): 97-102. (in Chinese)
陶永才，石磊. 异构资源环境下的 MapReduce 性能优化[J]. 小型微型计算机系统，2013，34(2)：97-102.

[7] KARGER D, LEHMAN E, LEIGHTON T, et al. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web[C]// ACM Symposium on Theory of Computing. CA, USA, 1997: 654-663.

[8] XIE J, YIN S, RUAN X J, et al. Improving mapreduce performance through data placement in heterogeneous Hadoop clusters[C]// IPDPS Workshops. Atlanta: IEEE Computer Society Press, 2010: 1-9.

[9] ZAMAN S, GROSU D. A distributed algorithm for the replica placement problem[C]// Proc. of IEEE Transaction on Parallel and Distributed System. 2011: 1455-1468.

[10] LUO P, GONG X. Research and Improvement of Data Placement Strategy for HDFS [J]. Computer Engineering and Design, 2014, 35(4): 1127-1131. (in Chinese)
罗鹏，龚勋. HDFS 数据存放策略的研究与改进[J]. 计算机工程与设计，2014，35(4)：1127-1131.

(上接第 65 页)

[4] GUO P J, KIM J, RUBIN R. How video production affects student engagement: an empirical study of MOOC videos[C]// Proceedings of the First ACM Conference on Learning @ Scale Conference. 2014: 41-50.

[5] CHORIANOPOULOS K, GIANNAKOS M N, CHRISOCHOIDES N. Open system for video learning analytics[C]// Proceedings of the First ACM Conference on Learning @ Scale Conference. 2014: 153-154.

[6] GAŠEVIČ D, MIRRIAHI N, DAWSON S. Analytics of the effects of video use and instruction to support reflective learning [C]// Proceedings of the Fourth International Conference on Learning Analytics and Knowledge. 2014: 123-132.

[7] LI Y, LU J J, ZHANG Y F, LI R, et al. A Novel Video Annotation Framework Based on Video Object[C]// Proceedings of the 2009 International Joint Conference on Artificial Intelligence. 2009: 572-575.

[8] YADAV K, SHRIVASTAVA K, DESHNUKH O. Towards Supporting Non-linear Navigation in Educational Videos[C]// Proceedings of the 16th International Conference on Multimodal Interaction. 2014: 82-83.

[9] BURGE J. Insights into Teaching and Learning: Reflections on MOOC Experience[C]// Proceedings of the 46th ACM Technical Symposium on Computer Science Education. 2015: 600-603.

[10] FIDALGO-BLANCO Á, SEIN-ECHALUCE M L, GARCÍA-PEÑALVO F J, et al. Improving the MOOC learning outcomes throughout informal learning activities[C]// Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality. 2014: 611-617.

[11] VideoNot. es[EB/OL]. http://www.videonot.es.

[12] 网易云课堂[EB/OL]. http://study.163.com.

[13] 朴灵. 深入浅出 Node.js[M]. 北京：人民邮电出版社，2013.