基于 HDFS 的分布式长期保存系统实现研究

师洪波　吴振新

【摘　要】随着信息社会发展，海量数字信息资源存储的需求变得越来越普遍，使用分布式文件存储是一种有效的解决方案。文章通过分析 HDFS 本身信息存储交互的特点，给出了使用 HDFS 的长期保存分布式存储实现方案，为今后使用 HDFS 及 Fedora 进行分布式长期保存及管理提供了借鉴参考。

【关键词】长期保存　Fedora　分布式存储　HDFS　Hadoop

Abstract：With the development of information society，the needs for mass information storage become more and more common，and distributed file system is an effective solution. Based on analysis of the features of HDFS，this paper gives out a distributed long term preservation system solution based on HDFS，aiming at providing reference for future study of distributed long term preservation with HDFS and Fedora.

Key words：long-term preservation　Fedora　Distributed File System　HDFS　Hadoop

随着信息社会的发展，海量信息存储及分析处理在今天变得越来越普遍。在数字资源长期保存领域，需要保存处理的数字信息也急剧增长，使用传统的保存处理方式，不论是存储容量上，还是存储效率存储安全上，都难以满足保存海量数据的要求。分布式存储是解决海量信息存储及处理的有效方式，是解决长期保存海量数据存储的重要技术手段。Apache 开源基金支持的顶级项目 Hadoop[1] 项目提供了开源分布式存储解决方案 HDFS（Hadoop Distributed File System）[2]，提供了高效、安全的海量数据分布式存储平台。HDFS 不仅提供了一个分布式存储环境，同时结合 Hadoop 的 Map-Reduce[3] 编程框架还可以提供分布式海量数据处理方案，解决海量信息处理效率等问题。研究基于 HDFS 的长期保存分布式存储解决方案，对于数字资源长期保存具有重要的研究和实践意义。

1　长期保存对于存储的需求

数字信息有产生速度快、易流失的特点，因此需要长期保存系统存储这些不断增长的海量信息，同时保证资源的安全性，保障这些资源在很长一段时间后仍然可用。国内外很多研究项目都对长期保存系统进行了研究，例如 HathiTrust[4] 项目认为长期保存系统需要满足以下几个方面的要求[5]：可靠性（保证存储内容的完整性）、冗余性（在单点或多点上的副本性）、可扩展性（包括大规模数据的易管理性）及可获取性（内容可以为保存系统所用）；同时存储的方式能够方便不同存储平台的迁移，不能对存储形成过度依赖；又如欧洲的荷兰图书馆、英国图书馆等在对长期保存服务的联合研究中认为[6]：长期保存要保证保存资源的真实性，对于资源的一些偶然的或者蓄意的损坏，存储应该能对他们进行检测并修复。通过这些研究及实践，对长期保存系统需求可以总结如下：

（1）海量性。长期保存系统的存储容量应具有海量性。以国家科学图书馆的数字资源长期保存系统为例，它目前所保存的数字文献资源就已经达到了几千万条，且仅仅是一些期刊资源，如果包括图书、视频、音频资源，数量就更为巨大。同时这些资源还在以非常高的速度增长，保存这些资源使用常规的存储系统难以满足其需求，必须要具有海量容量的存储系统。

（2）可靠性。存储系统要保证保存的内容完整可靠。由于保存内容的海量性，内容出错的概率大大增加。存储本身应该具有防止这些错误（不论是硬件性还是软件性的）发生的功能，当错误发生时候应该能够恢复，以保证内容完整。

（3）可扩展性。保存内容是不断增加的，保存系统的容量也要具有良好的扩展性。首先保存介质的价格随着

时间不断降低，没有必要一次性投入购买大量存储介质；其次保存的资源的类型、范围也会不断扩大；再次信息随着时间推移也会增加。从多方面考虑，存储系统的可扩展性都非常重要。

（4）易获取性。首先要保证存储的内容可以被获取，保证保存的资源是可利用的；其次要保证能够比较高效地存取内容，不能随着资源增多而增加寻找这些资源的时间。

（5）冗余性。资源应该存储多个副本，保证存储出错情况下的安全运行。即使设计再好的运行系统往往也会面临很多不可抗因素，如机房停电、地震等不可抗因素的影响，多副本或多点存储能够提高系统应对这些故障的能力。

除了以上要求外，长期保存系统还要对所存储的数据定期进行一些数据的操作以保障保存内容的长期可用性，例如定期对所有保存文件进行校验，以确保安全性；随着文件格式的更新换代，对保存内容进行格式迁移等等文件操作。由于这些操作所涉及文件众多，具有海量性，文件的存储也应该尽可能满足这方面的需求。

2 HDFS 分布式存储特点

2.1 HDFS 的架构特点

HDFS 是设计运行在通用硬件（commodity hardware）上的分布式文件系统，在设计上具有很强的容错性。HDFS 最初是由 Nutch[7] 网络搜索引擎项目发展而来（在 Nutch 中仍在使用），目前属于 Hadoop 的子项目[8]。

HDFS 采用了类似图 1 的架构设计图，是一种 master/slave 型架构。一个 HDFS 集群是由一个 Namenode 和数个 Datanode 组成。Namenode 是中心服务器，负责管理文件系统的 Namespace 以及客户端对文件的访问，比如打开、关闭、重命名文件或目录，也负责确定数据块到具体 Datanode 节点的映射，对应图 1 中的文件目录服务。集群中的 Datanode 一般是一个节点一个，负责管理它所在节点上的存储，在 Namenode 的统一调度下进行数据块的创建、删除和复制，对应图 1 中的扁平目录服务。在 HDFS 内部，一个文件其实被分成一个或多个 Block，这些块存储在一组 Datanode 上。HDFS 的详细结构图如图 1 所示。

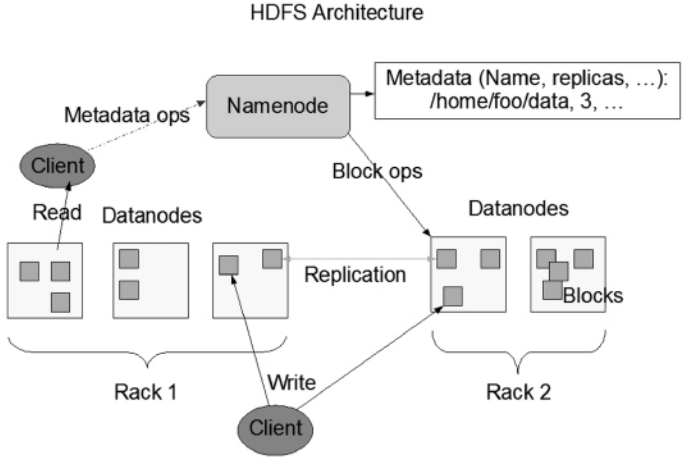

图 1 HDFS 系统架构

2.2 HDFS 的优势

HDFS 除了提供通用分布式文件系统功能之外，还具有以下优势：

（1）存储设计具有海量性。HDFS 的一个重要设计目标就是存储海量文件，所以其本身就具有海量存储特性。

（2）设计应用在通用硬件机器上，具有较强的兼容性和经济性。HDFS 采用 Java 语言开发，保证了系统有较强的可移植性。在系统设计时，对机器的软件系统、硬件系统均没有非常强的约束，给开发及应用 HDFS 提供了便捷。

（3）HDFS 系统有比较强的健壮性。首先，HDFS 的一个主要设计目标就是在存在错误的情况下保证数据的安全性。对于三种常见的错误类型：Namenode 出错、Datanode 出错和网络出错（network partitions）等，在设计时就给予了设计规避。其次，在数据存储上，所有的数据块都会有副本，既保证了数据的高并发访问性，又保障了数据的安全性和冗余性。

（4）可访问性。首先 HDFS 提供了多种访问方式，包括 Java API[9] 接口、C 语言封装的 API 接口、浏览器访问等方式，而且目前正在开发 WebDAV 协议访问的方式。其次采用文件元数据与内容分开存储，访问文件的速度

不会随着文件增多而变慢。

（5）可扩展性。HDFS 能够动态增加 DataNode 节点来增强其存储能力。这一点可以保证所提供的存储服务不会中间停机，增加了其作为底层存储的适应能力。

（6）HDFS 提供了并行文件处理基础。该特性是 HDFS 独有的特性。HDFS 是 Hadoop 分布式计算框架的基础部分，数据文件存储只有存储在 HDFS 中才能应用 Map-Reduce 应用框架进行并行的文件处理。文件存储在 HDFS 后，为进一步并行文件处理提供了可能。

通过以上分析不难发现，HDFS 的多方面特性可以满足长期保存存储的需求。

3 基于 HDFS 的层次化分布式长期保存系统的构建及实现

3.1 基于 HDFS 的层次化分布式长期保存系统的构建

参考国际上长期保存系统项目以及作者所参与的数字文献资源长期保存系统的建设，作者构建了如图 2 的基于 HDFS 的层次化分布式长期保存系统。

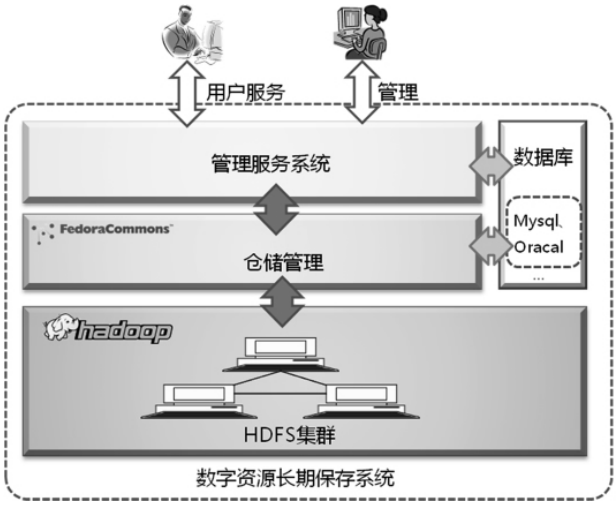

图 2 基于 HDFS 的层次化长期保存系统

该层次结构模型主要由三层组成，上层是管理服务层，中层使用 Fedora[10] 仓储管理软件层；下层是 HDFS 分布式存储层。

管理服务系统是保存系统与外界交互并对保存内容进行管理的子系统。主要功能或服务包括：负责将需要保存的内容摄入到保存系统中，对保存内容尽心管理；为用户提供检索、分发服务；对保存内容进行管理，如格式翻新、完整性校验等；各个功能以模块化形式存在，如摄入模块、索引、检索、完整性校验、翻新等。

Fedora 仓储软件层实现了对摄入内容的逻辑管理。其功能主要有：为上层管理服务系统提供各个保存内容的逻辑单元的各部分信息，如元数据、原始文件等；对保存内容统一管理；与保存内容的物理存储进行交互，存取保存媒体上的信息。在 Fedora 中，数字资源以数字对象的形式进行管理，数字对象中保存了资源的各种元数据及相互关系等信息。

底层存储系统层提供保存内容的主要媒介，提供了保存内容存储管理的主要功能。HDFS 系统提供了保存内容的物理层面的管理，实现了数据复制、数据读写等保存内容管理。

3.2 Fedora 架构特点

Fedora 是一个开源软件仓储系统，该软件由 Fedora Commons[11] 提供开发支持。Fedora 具有良好的设计架构及灵活扩展能力，越来越多的机构或项目使用 Fedora 进行数字内容存储研究及实际应用，如 eSciDoc[12]、DANS[13] 等等。

Fedora 内部对数字资源采用数字对象的形式进行管理[14]，通过定义多种内容模型[15]，对数字对象实现有效地管理。数字对象以文档形式存放在文件系统中，一部分管理及访问用的元数据存储在外部的关系型数据库中，如 Mysql 中，Fedora 对底层存储管理采用灵活的模块管理方式，通过不同的存储模块实现与不同存储介质交互的目的。如图 3 所示，Fedora 通过不同的存储管理模块实现与不同的底层存储（本地存储、云存储、网格等方式）实现交互。Fedora 默认提供的底层存储模式是本地存储模式，其他存储模式需要研究人员依据需要进行自行开发。

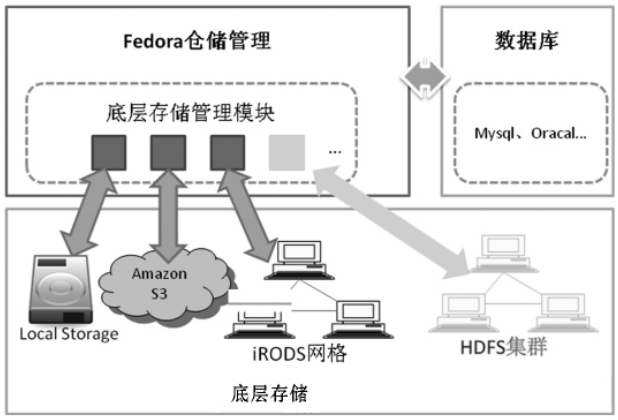

图 3 Fedora 管理架构图

如 DuraSpace 提供了与云进行交互的模块[16]，iRODS 社区提供了 iRODS 与 Fedora 交互的模块[17]。

3.3 以 HDFS 作为长期保存系统分布式存储的关键部分实现

3.3.1 关键部分

以 HDFS 作为底层分布式存储，关键是开发 HDFS 与 Fedora 交互部分。通过上面的分析可知，Fedora 主要以底层管理模块的形式与底层存储交互，所以主要开发任务也就是开发 Fedora 的 HDFS 底层存储模块。Fedora 社区本身并没有提供底层存储模块开发的相关指导或文档，不过得益于 Fedora 项目的开源性，通过对 Fedora 项目底层本地存储模块的源码进行分析，并结合 HDFS 本身交互特点，确定了如下实现思路：

（1）参考 Fedora 存储模块与本地存储交互的过程，将与本地存储交互替换为与 HDFS 存储交互。

（2）参考 HDFS API，确定需要使用的相关函数（如文件写入、读取、删改等操作函数），并将这些函数融入到新的存储模块中。

3.3.2 试验环境搭建

（1）配置 HDFS

HDFS 的安装方式有三种，即单机模式、伪分布模式、完全分布模式。由于研究主要测试 HDFS 的可用性，因此采用伪分布模式调试程序。

按照 Hadoop 官方指导文档方式分别配置机器的 SSH、配置 Hadoop 所在文件夹下的 conf 中的配置文件，重点配置 hadoop-env.sh、core-site.xml、hdfs-site.xml、mapred-site.xml 等文件。

其中 hdfs-site.xml 的配置与官网稍有不同，按照如下方式配置好 namedir 和 datadir 的文件夹。运行 Hadoop 的账户应有该文件夹的读写权限。

```
<configuration>
<property> <name>dfs.replication</name> <value>1</value> </property> <property>
<name>dfs.permissions</name>
<value>false</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>/hadoop/namespace</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>/hadoop/dataspace</value>
</property>
</configuration>
```

（2）配置 Fedora 运行环境

选择另外一台机器，下载 Fedora 发布的稳定版安装程序进行安装，配置采用官方安装文档建议的方式，这里不需额外的配置。

3.3.3 Fedora 存储模块编写

（1）编写 HDFS 底层存储管理模块

下载 Fedora 安装版本对应的源码，复制 org.fcrepo.server.storage.lowlevel 包中的类文件，到新的包中 org.fcrepo.server.storage.lowlevel.hadoop 中，分别修改以下类中的方法。

参考类 GenericFileSystem 中的 writeIntoExistingDirectory（File file，InputStream content）及 rewrite（File file，InputStream content）方法，将原有的写本地文件的方法，修改为写 HDFS 集群的方法。修改 inputStream read（File file）方法，将读取本地文件的方法修改为读取 HDFS 集群的方法。修改 delete（File file）中删除本地文件的方法，修改为删除 HDFS 集群中文件的方法。以上对 HDFS 读、写、删除等操作均参考 Hadoop 的客户端配置。

参考类 DefaultLowlevelStorageModule 中方法 postInitModule（），对方法中获取数字对象及数字对象组件存储位置的过程进行修改，将原有获取相对路径的方式转换为获取绝对路径的方式；增加初始化 HDFS 客户端过程。具体修改如下：

```
//配置数字对象和对象组件的存储位置
String objectStoreBase = getModuleParameter (DefaultLowlevelStorage.OBJECT_STORE_BASE, false);
String datastreamStoreBase = getModuleParameter (DefaultLowlevelStorage.DATASTREAM_STORE_BASE, false);
//配置客户端
org.apache.hadoop.conf.Configuration hdConfiguration = new Configuration ( ); hdConfiguration.addResource (new Path (FEDORA_HOME + File.separator +" server" +File.separator+" config" +File.separator+" core-site.xml"));
hdConfiguration.addResource (new Path (FEDORA_HOME + File.separator +" server" +File.separator +" config" +File.separator+" hdfs-site.xml"));
hdConfiguration.addResource (new Path (FEDORA_HOME + File.separator +" server" +File.separator +" config" +File.separator+" mapred-site.xml"));
try {
GenericFileSystem.hadoopFS = org.apache.hadoop.fs.FileSystem.get (hdConfiguration);
GenericFileSystem.filterHadoopFS = new org.apache.hadoop.fs.FilterFileSystem (GenericFileSystem.hadoopFS); } catch (IOException e) {
System.out.println (" Hadoop clent 构造失败 (Create Failed)!");
e.printStackTrace ();
}
```

将以上包 org.fcrepo.server.storage.lowlevel.hadoop 进行编译并打包成 hadoop-storage.jar。

（2）配置 Fedora

修改 Fedora 应用中的配置文件，将配置文件中有关底层存储的部分相关参数修改为使用 HDFS 集群存储数字资源。具体参数修改结果如下：

* 修改 fedora.fcfg 文件中的以下部分：

```
<module role=" org.fcrepo.server.storage.lowlevel.ILowlevelStorage" class=" org.fcrepo.server.storage.lowlevel.hadoop.DefaultLowlevelStorageModule" >
<param name=" object_store_base" value=" /data/objects" isFilePath=" true" >
<comment>The root directory for the internal storage of Fedora
objects. This value should be adjusted based on your installation
environment. This value should not point to the same location as
datastream_store_base.</comment>
</param>
```

<param name="datastream_store_base" value="/data/datastreams" isFilePath="true" >
<comment>The root directory for the internal storage of Managed
Content datastreams. This value should be adjusted based on your
installation environment. This value should not point to the same
location as object_store_base.</comment>
</param>
<param name="path_registry" value="org.fcrepo.server.storage.lowlevel.hadoop.DBPathRegistry" >
<comment>The java class used to determine the path registry; default
is org.fcrepo.server.storage.lowlevel.DBPathRegistry.</comment>
</param>
<param name="path_algorithm" value="org.fcrepo.server.storage.lowlevel.hadoop.TimestampPathAlgorithm" >
<comment>The java class used to determine the path algorithm;
default is org.fcrepo.server.storage.lowlevel.hadoop.TimestampPathAlgorithm.</comment>
</param>
<param name="file_system" value="org.fcrepo.server.storage.lowlevel.hadoop.GenericFileSystem" >
<comment>The java class that determines the implementation class;
default is org.fcrepo.server.storage.lowlevel.hadoop.GenericFileSystem.</comment>
</param>
<param name="backslash_is_escape" value="true" >
<comment>Whether the escape character (i.e. (the token beginning an
escape sequence) for the backing database (which includes
registry tables) is the backslash character. This is needed to
correctly store and retrieve filepaths from the registry
tables, if running under Windows/DOS. (Set to true for MySQL and
Postgresql, false for Derby, Oracle and McKoi.</comment>
</param>
</module>

* 复制 HDFS 集群中的配置文件 core-site.xml、hdfs-site.xml、mapred-site.xml 到 FEDORA_HOME1/server/config 文件夹中。

* 将与集群版本相同的 hadoop-*-core.jar 及刚才打包生成的 hadoop-storage.jar，复制到 FEDORA_HOME\tomcat\Webapps\fedora34lolevel\Web-INF\lib 中。

* 清空 Fedora（包括数据库中）的所有数据。

3.3.4 试验结果

按照以上方法进行编写及配置后，启动 Fedora 程序，该 Fedora 应用就是使用 HDFS 作为底层存储了。

首先启动 HDFS 集群，然后使用 fedora-admin.bat 或者 fedora-admin.sh 启动 Fedora 管理客户端。进行数字对象的创建（或者导入）、修改或者删除等操作，验证 Fedora 能够正常访问 HDFS 集群，证明了 HDFS 作为长期保存系统分布式存储的可行性。

4 总结

海量信息资源的存储是当前长期保存项目中不得不面对的问题，寻找到优质、高效、稳定的存储解决方案对信息资源的保存利用有着重要价值。使用层次化的系统结构使得各部分之间逻辑功能划分清楚，管理简便高效，开发方便。使用 Fedora 作为数字对象逻辑管理层，并且将层次存储交由 Fedora 交互，降低了系统的复杂度。HDFS 还提供了高效、稳定、经济的分布式文件存储解决方案，满足了长期保存系统的需求。使用过程中，Fedora 仓储软件本身并没有提供本地文件存储之外的接口，但是其层次化管理结构，使得 Fedora 集成 HDFS 成为

可能。通过修改 Fedora 原有的文件存储方式，结合 Hadoop 客户端 API，可以相对简单地完成 HDFS 集群存储管理模块的编写，从而实现 Fedora 存取 HDFS 无缝集成。本文介绍了 Fedora 的 HDFS 的存储模块的编写的主要过程，但是在使用 HDFS 的时候还有很多需要注意的问题，下面列举一些比较关键的问题：

（1）HDFS 主要针对比较大的文件进行设计，对于存取数量众多的小文件，可能存在一定的效率问题，需要额外的解决方案辅助解决。具体可以参阅文献[18]。

（2）HDFS 集群的性能调优。在实际使用 HDFS 集群作为存储环境时，需要对集群的相关参数进行调整，以达到不同应用的目的。

注释

[1]FEDORA_HOME 指 Fedora 程序安装的位置. Hadoop. http://hadoop.apache.org/, 2011-10-11
[2]HDFS. http://hadoop.apache.org/hdfs/, 2011-10-11
[3]MapReduce. http://hadoop.apache.org/mapreduce/, 2011-10-11
[4]HathiTrust. http://www.hathitrust.org/, 2011-10-11
[5]Jeremy York. Building A Future By Preserving Our Past: The Preservation Infrastructureof HathiTrust Digital Library. In: WORLD LIBRARY AND INFORMATION CONGRESS: 76TH IFLA GENERAL CONFERENCE AND ASSEMBLY, 2010: 5-6
[6]Sean Martin, David Golding, Paul Noakes et al. Long Term Preservation Service. http://www.kb.nl/hrd/dd/dd_links_en_publicaties/publicaties/KB_Long_Term_Preservation_Services_2010-08-05.pdf, 2011-10-11
[7]Nutch. http://nutch.apache.org/, 2011-10-11
[8]HDFS Architecture Guide. http://hadoop.apache.org/common/docs/current/hdfs_design.html, 2011-10-11
[9]Hadoop API. http://hadoop.apache.org/common/docs/current/api/, 2011-10-11
[10]About Fedora Commons. http://www.fedora-commons.org/about, 2011-10-11
[11]Fedora Commons. http://www.fedora-commons.org/, 2011-10-11
[12]eSciDoc. https://www.escidoc.org/, 2011-10-11
[13]DANS. http://www.dans.knaw.nl/, 2011-10-11
[14]Fedora Digital Object Model. https://wiki.duraspace.org/display/FCR30/Fedora+Digital+Object+Model, 2011-10-11
[15]Content Model Architecture. https://wiki.duraspace.org/display/FCR30/Content+Model+Architecture, 2011-10-11
[16]DuraCloud+Architecture. https://wiki.duraspace.org/display/DURACLOUD/DuraCloud+Architecture, 2011-10-11
[17]iRODS Fedora. https://www.irods.org/index.php/Fedora, 2011-10-11
[18]The Small Files Problem. http://www.cloudera.com/blog/2009/02/the-small-files-problem/, 2011-10-11

师洪波　中国科学院研究生院，中国科学院国家科学图书馆。
吴振新　中国科学院国家科学图书馆。

（上接第 28 页）

3.6　提高采访人员素质，重视对采访人员的培训工作

图书馆的采购工作是一项较为专业的工作，对采访人员要求很高，需要采访人员了解各学科专业的相关知识和图书馆学的专业知识。在进行政府采购的过程中，图书馆的采访人员还需要集中进行多个批次的采购工作，不仅需要扎实的专业功底，较强的实务能力，还需要有很强的责任心。因此，图书馆要重视提高采访人员的素质，对采访人员进行更全面的专业培训。只有这样才能提高图书馆各类信息资源的采选质量。

注释

[1]林大静．政府采购制度的研究．商业研究，2002（6）：62-65
[2]政府采购．http://baike.baidu.com/view/229817.htm，2011-10-22
[3]吴军．政府采购制度与高校图书馆的图书采购．大学图书情报学刊，2000（4）：8-10
[4]周春阳．浅论地方高校图书馆中文图书的政府采购．湖北广播电视大学学报，2009（2）：150-151
[5]肖莉．图书资料政府采购利弊谈——以江西省图书馆为例．江西图书馆学刊，2006（4）：34-35
[6]熊琪．对中文图书实施政府采购的思考．江西图书馆学刊，2007（2）：42-43

李　薇　辽源市图书馆。