Hadoop Common / HADOOP-1568

NameNode Schema for HttpFileSystem

Details

Type:	New Feature	Status:	CLOSED
Priority:	Major	Resolution:	Fixed
Affects Version/s:	None	Fix Version/s:	0.14.0
Component/s:	fs		
Labels:	None		

Description

This issue will track the design and implementation of (the first pass of) a servlet on the namenode for querying its filesystem via HTTP. The proposed syntax for queries and responses is as follows.

Query

```
GET http://<nn>:<port>/ls.jsp[<?option>[&option]*] HTTP/1.1
```

Where *option* may be any of the following query parameters:
path : String (default: '/')
recursive : boolean (default: false)
filter : String (default: none)

Response

The response will be returned as an XML document in the following format:

```
<listing path="..." recursive="(yes|no)" filter="..."
         time="yyyy-MM-dd hh:mm:ss UTC" version="...">
  <directory path="..."/>
  <file path="..." modified="yyyy-MM-dd hh:mm:ss" blocksize="..."
        replication="..." size="..."
        dnurl="http://dn:port/streamFile?..."/>
</listing>
```

Attachments

1568-6.patch	20 kB	10/Aug/07 22:04
compat.patch	0.8 kB	11/Aug/07 00:35
xmlenc-0.52.jar	15 kB	17/Jul/07 21:40

Issue Links

is depended upon by

HADOOP-1563 Create FileSystem implementation to read HDFS data via http CLOSED

relates to

HADOOP-1621 Make FileStatus a concrete class CLOSED

Activity

Christopher Douglas created issue - 05/Jul/07 21:34

Christopher Douglas made changes - 05/Jul/07 21:35

Field	Original Value	New Value
Link		This issue is blocked by ~~HADOOP-1563~~ [~~HADOOP-1563~~]

Christopher Douglas made changes - 05/Jul/07 21:36

Link		This issue blocks ~~HADOOP-1563~~ [~~HADOOP-1563~~]

Christopher Douglas made changes - 05/Jul/07 21:37

Link	This issue is blocked by ~~HADOOP-1563~~ [~~HADOOP-1563~~]	

Owen O'Malley added a comment - 05/Jul/07 21:53

Instead of dnurl, I'd suggest dataurl.

Thomas White added a comment - 06/Jul/07 08:47 - edited

How about making the file path a part of the URL path? Also, why not drop the .jsp as it doesn't add anything. So:

```
http://<nn>:<port>/ls/path[<?option>[&option]*]
```

How does filter work - can you give an example, please?

For the response, I would strongly recommend using ISO 8601 for the timestamp representation. E.g.:

```
modified="yyyy-MM-ddThh:mm:ssZ"
```

Doug Cutting added a comment - 06/Jul/07 19:25

We could simply use HTML and observe the conventions described in https://issues.apache.org/jira/browse/HADOOP-1563#action_12510760...

Owen O'Malley added a comment - 06/Jul/07 20:55

Actually, we need to subclass the reader from ~~HADOOP-1563~~ to be specific for HDFS anyways, because we need the attributes to be available. In particular, we need file size, replication, block size, and modification time. If we are defining a format for tools to parse, I'd much rather use xml than http, because it is far more appropriate for that use.

Owen O'Malley added a comment - 06/Jul/07 21:00

On the namespace, I guess we should go ahead and manage it with a servlet, via:

http://<nn>:<port>/ls/<path>[?option(&option)*]

because that will play nicely with Doug's patch on ~~HADOOP-1563~~.

Doug Cutting added a comment - 06/Jul/07 21:18

> we need file size, replication, block size, and modification time

File size and modification time are already supported, via the Content-Length and Last-Modified http headers. We could support replication by adding a Replication and Block-Size by adding headers and looking for them. (An HTTP HEAD request is used to get file status.) If we need to at all. This is not intended as a primary access means for HDFS.

If we use a de-facto standard for exposing file systems, then we can use a shared tool. Alternately, we can invent a format and create a specific tool. An advantage of using the de-facto-standard is that it is a lowest-common-denominator. We're less likely to add features that are version-specific. It should always be compatible between HDFS versions. Older versions can use it to access a newer version and vice versa. If we start supporting more advanced features, then we increase the chance that we'll have incompatibilities.

Also, note that, the way our FileSystem API works, directory listing is most naturally separate from fetching file status. This can be optimized by caching file status in paths returned from directory listings, but we still need a way to fetch individual file's status information.

Using the de-facto standard also avoids having to design a feature-complete XML schema!

Christopher Douglas added a comment - 07/Jul/07 02:28

Thoughts:

1) It isn't clear how one would format a list of block locations in a reply. One could request the contents of a consecutive set of blocks by setting query parameters or using a range header field, but XML provides a more expressive format for that association that might also permit a consumer to contact multiple datanodes simultaneously, from different machines, etc. We don't need or support it now, but situations where it could be useful are not difficult to imagine.
2) Each request for file info is a separate HTTP HEAD request. A consumer needs to parse the link to acquire interesting information about the file, including its name (as in the patch). If we hope to avoid assumptions about the link syntax, a HEAD request is necessary for every file in the listing. Granted, the link syntax is probably pretty stable, but parsing URIs for file paths doesn't seem like a more future-proof solution than an XML schema.

Further, if we hope to implement (1), we would probably break the current implementation of (2) or include block location URIs in header fields.

It seems prudent to restrict this servlet to meta-information, i.e. not returning file content in response to a GET and leave that to StreamFile or some other facility. Some of the ideas in this and in ~~HADOOP-1563~~ seem very appropriate for StreamFile, though.

Thomas White added a comment - 08/Jul/07 21:05

My instinct is to avoid creating an XML schema if we can reuse another one (HTML, plus some HTTP headers). As Doug points out, we can use lots of existing tools. (And it's what the REST crowd seem to be doing these days - e.g. Chapter 9 of http://www.oreilly.com/catalog/9780596529260/).

> 1) It isn't clear how one would format a list of block locations in a reply.

Could we use a `rel="alternate"` attribute (http://www.w3.org/TR/html4/types.html#type-links) to list (alternative) block locations? I'm not sure, but thought it was worth suggesting.

Owen O'Malley added a comment - 09/Jul/07 18:18

I really don't see how scraping data out of html is better than parsing xml. In my experience, doing html scraping is fairly brittle because the dependencies aren't obvious to the maintainers of the server. By putting it into an xml format, it is very clear that formatting doesn't matter, but that attribute names do. You also get libraries to help you parse the xml, which you don't for http.

Our primary use case is distcp of a large dataset. It will kill performance to require the copy planner to do a http head for each file (or even worse each attribute). I really don't see anyway to make it perform adequately using that approach. Yes, the client will need to cache the metadata, but that isn't too hard.

James P. White added a comment - 09/Jul/07 18:36

The improvement comes by using a format that encodes a machine-readable schema into HTML.

The best of these (which you may like to think of as "microformats done right") is eRDF (Embedded RDF):

http://www.ifcx.org/wiki/EmbeddedRDF.html

In addition to easily enabling data to be extracted from the HTML, a GRDDL header can point to an extraction (or the original in the case of RDF-to-rRDF generation) that simply requires a GET.

I've been trying to identify a good RDF schema for file systems, but most folks just treat file systems as resources and use RDF collection types. I'm thinking though there may be something useful in Mozilla. Fortunately though, thanks to RDF's open approach to schema, it doesn't really matter if you start with a good commonly used one or not.

And since this is XHTML, there is no special library support needed that isn't in an XML library. As the eRDF home page illusrates, a simple XSLT script is an easy (though not necessarily the fastest) way to get data out of eRDF.

Doug Cutting added a comment - 09/Jul/07 19:46

> It isn't clear how one would format a list of block locations in a reply. [...]

I think we need to be clearer about the goal here before we will reach consensus. If the goal is to have a simple way of accessing HDFS (and potentially other file systems) over HTTP that's unlikely to change between versions, then I don't think we should be worrying about, e.g., formatting lists of block locations. Here we should be looking for a lowest-common-denominator to maximize interoperability. Mostly it just needs to make distcp and MapReduce input possible, with decent performance. It should not attempt to be full featured (as that will make it more fragile and prone to inter-version compatibility problems) and is thus unlikely to perform optimally.

If scraping HTML doesn't work, we should consider WEBDAV (again). The PROPFIND method permits directory enumeration, returned as XML.

Doug Cutting added a comment - 09/Jul/07 20:07

> It will kill performance to require the copy planner to do a http head for each file

It should be significantly faster than the actual copy, no? The copy will need to stat each file at open, which is the same number of namenode requests. So, yes, it might significantly affect copy performance, but it shouldn't dominate. Isn't that acceptable for a compatibility tool? Again, if we want to provide optimal performance, then we'll need to expose more of the internals (like block locations) and then the tool will be more fragile. If that's required, then perhaps we should instead consider making the HDFS client and servers all support multiple protocol versions.

Owen O'Malley added a comment - 09/Jul/07 21:52

The problem is that under our proposal the planner does 1 http get. Under your proposal, the planner does 10 million (or 40 million without caching) serial http head operations. That will take a long time. The actual gets will take a long time, but they are running in parallel and will be answered by the datanodes.

Christopher Douglas added a comment - 09/Jul/07 22:14

Attached a patch for an example servlet.

This patch includes a bug: NameNode::getFileInfo throws an exception with "/" as a parameter. I'm looking into what might be causing this.

Christopher Douglas made changes - 09/Jul/07 22:14

Attachment		ls-xml.patch [12361444]

Doug Cutting added a comment - 09/Jul/07 22:33

> under our proposal the planner does 1 http get. Under your proposal, the planner does 10 million (or 40 million without caching) serial http head operations

What is your proposal? What is the task? I thought it was listing a directory to be copied. That would take one HEAD per file in the directory. Are you copying a directory with 10M files? Why are you multiplying by 4? Each file only needs to be stat'd once. It actually doesn't even need that if we're willing to forgo sorting by length. So it could just use a single GET with HTML too--just list the names to be copied. Recursive listings would take caching isDir in the path, but could still be reduced to a single GET per dir.

Michael Bieniosek added a comment - 09/Jul/07 23:03

What's wrong with implementing a subset of WebDAV? Why are you inventing your own protocol?

Christopher Douglas made changes - 10/Jul/07 21:49

Attachment		ls-xml2.patch [12361531]

Christopher Douglas made changes - 11/Jul/07 01:06

Attachment	ls-xml2.patch [12361531]	

Christopher Douglas made changes - 11/Jul/07 01:07

Attachment		ls-xml2.patch [12361545]

Christopher Douglas made changes - 17/Jul/07 07:32

Attachment	ls-xml.patch [12361444]	

Christopher Douglas made changes - 17/Jul/07 07:32

Attachment	ls-xml2.patch [12361545]	

Christopher Douglas made changes - 17/Jul/07 07:33

Attachment		1568.patch [12361960]

Christopher Douglas made changes - 17/Jul/07 07:39

Link		This issue is blocked by ~~HADOOP-1621~~ [~~HADOOP-1621~~]

Christopher Douglas made changes - 17/Jul/07 20:51

Attachment	1568.patch [12361960]	

Christopher Douglas added a comment - 17/Jul/07 21:12

Note: this requires xmlenc (http://xmlenc.sourceforge.net/)

Christopher Douglas made changes - 17/Jul/07 21:12

Attachment		1568.patch [12362007]

Christopher Douglas made changes - 17/Jul/07 21:12

Comment	[Modified to use xmlenc (http://xmlenc.sourceforge.net/)]	

Christopher Douglas made changes - 17/Jul/07 21:40

Attachment		xmlenc-0.52.jar [12362011]

Doug Cutting added a comment - 17/Jul/07 21:47

You redirect file data requests to the first datanode with the first block. Might it be better to randomly select a datanode with the file's first block? Better yet might be to try to select a datanode that has the most blocks from the file, randomly picking in the case of ties (which will typically be small files that might be frequently accessed).

I'm not too fond of the servlet names BrowseFile and CatFile. Maybe something like ListPathsServlet and FileDataServlet, mounted as /listPaths/* and /data/* would be better?

Christopher Douglas made changes - 19/Jul/07 00:50

Attachment		1568-2.patch [12362100]

Christopher Douglas made changes - 19/Jul/07 22:43

Attachment	1568-2.patch [12362100]	

Christopher Douglas made changes - 19/Jul/07 22:43

Attachment		1568-2.patch [12362171]

Christopher Douglas added a comment - 30/Jul/07 21:57

Added missing static decl for initializer; added brackets for code blocks

Christopher Douglas made changes - 30/Jul/07 21:57

Attachment		1568-3.patch [12362818]

Christopher Douglas added a comment - 06/Aug/07 22:24

1568-3.patch missing change to conf/hadoop-default.xml

Christopher Douglas made changes - 06/Aug/07 22:24

Attachment		1568-4.patch [12363282]

Christopher Douglas made changes - 06/Aug/07 22:24

Attachment	1568.patch [12362007]	

Christopher Douglas made changes - 06/Aug/07 22:24

Attachment	1568-2.patch [12362171]	

Christopher Douglas added a comment - 07/Aug/07 00:52

Temporarily removed dependencies on ~~HADOOP-1621~~ so this may be committed independently.

Christopher Douglas made changes - 07/Aug/07 00:52

Attachment		1568-5.patch [12363288]

Christopher Douglas made changes - 07/Aug/07 00:53

Link	This issue is blocked by ~~HADOOP-1621~~ [~~HADOOP-1621~~]	

Christopher Douglas made changes - 07/Aug/07 00:54

Link		This issue relates to ~~HADOOP-1621~~ [~~HADOOP-1621~~]

Christopher Douglas made changes - 07/Aug/07 20:50

Fix Version/s		0.14.0 [12312474]
Status	Open [1]	Patch Available [10002]

Hadoop QA added a comment - 08/Aug/07 06:47

-1, build or testing failed

2 attempts failed to build and test the latest attachment http://issues.apache.org/jira/secure/attachment/12363288/1568-5.patch against trunk revision r563649.

Test results: http://lucene.zones.apache.org:8080/hudson/job/Hadoop-Patch/527/testReport/
Console output: http://lucene.zones.apache.org:8080/hudson/job/Hadoop-Patch/527/console

Please note that this message is automatically generated and may represent a problem with the automation system and not the patch.

Christopher Douglas made changes - 10/Aug/07 18:58

Attachment	1568-4.patch [12363282]	

Christopher Douglas made changes - 10/Aug/07 18:58

Attachment		1568-4.patch [12363607]

Christopher Douglas made changes - 10/Aug/07 22:04

Attachment	1568-3.patch [12362818]	

Christopher Douglas made changes - 10/Aug/07 22:04

Attachment	1568-4.patch [12363607]	

Christopher Douglas made changes - 10/Aug/07 22:04

Attachment	1568-5.patch [12363288]	

Christopher Douglas made changes - 10/Aug/07 22:04

Attachment		1568-6.patch [12363629]

Christopher Douglas made changes - 10/Aug/07 22:12

Status	Patch Available [10002]	Open [1]

Christopher Douglas made changes - 10/Aug/07 22:13

Fix Version/s	0.14.0 [12312474]	
Fix Version/s		0.15.0 [12312565]
Status	Open [1]	Patch Available [10002]

Owen O'Malley added a comment - 10/Aug/07 23:51

I committed this to 0.14, since it is just adding a few files and will provide forward and backward version compatability to HDFS. Thanks, Chris.

Owen O'Malley made changes - 10/Aug/07 23:51

Fix Version/s	0.15.0 [12312565]	
Fix Version/s		0.14.0 [12312474]
Resolution		Fixed [1]
Status	Patch Available [10002]	Resolved [5]

Nigel Daley added a comment - 11/Aug/07 00:32

Hudson -1'd this patch because it doesn't compile on JDK 1.5. This patch introduces JDK 1.6 dependencies which should be removed both from trunk and branch-0.14

Nigel Daley made changes - 11/Aug/07 00:32

Resolution	Fixed [1]	
Status	Resolved [5]	Reopened [4]

Christopher Douglas added a comment - 11/Aug/07 00:35

Patch against revision removing 1.6 dependency on IOException taking a cause.

Christopher Douglas made changes - 11/Aug/07 00:35

Attachment		compat.patch [12363643]

Jim Kellerman added a comment - 13/Aug/07 05:10

As we move forward, does trunk require jdk 1.6?

If not, this patch should be applied to trunk as well.

Owen O'Malley added a comment - 13/Aug/07 17:29

I just committed a fix for the java 1.5 problem. Sorry about that.

Owen O'Malley made changes - 13/Aug/07 17:29

Resolution		Fixed [1]
Status	Reopened [4]	Resolved [5]

Doug Cutting made changes - 20/Aug/07 18:12

Status	Resolved [5]	Closed [6]

Thomas White added a comment - 31/Aug/07 20:26

Sorry, just had a chance to look at this (been on vacation), and I had a few comments:

1. SimpleDateFormat is not thread safe, so df in ListPathsServlet cannot be shared. (See http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html#synchronization) It's probably simplest to create a new instance for each GET.
2. Is HftpFileSystem a typo?
3. A few unit tests would be nice at some point 🙂

Doug Cutting added a comment - 31/Aug/07 20:48

> 2. Is HftpFileSystem a typo?

No. It's not generic access to filesystems over HTTP (as I attempted in ~~HADOOP-1563~~) but rather access to the Hadoop FileSystem feature set over HTTP. We couldn't specify 'http' as the URI's scheme, since the actual URL accessed doesn't match this. The HftpFileSystem implementation translates the hftp: URIs to HTTP URLs accessing two servlets, one for directory listings and another for file content.

The purpose of all this is to provide a version-independent means to access HDFS filesystems, for back-compatiblity. So, if you're running two clusters and only upgrade one, but wish them to be able to access one another, this should let you. This protocol is simple-enough that we don't expect it to change incompatibly, unlike HDFS's internal protocols, which change regularly.

Thomas White added a comment - 31/Aug/07 21:10

Doug - thanks for the clarification.

Gavin McDonald made changes - 02/May/13 02:29

Link This issue blocks ~~HADOOP-1563~~ [~~HADOOP-1563~~]

Gavin McDonald made changes - 02/May/13 02:29

Link This issue is depended upon by ~~HADOOP-1563~~ [~~HADOOP-1563~~]

Transition	Time In Source Status	Execution Times
Christopher Douglas made transition - 10/Aug/07 22:12 PATCH AVAILABLE → OPEN	3d 1h 22m	1
Christopher Douglas made transition - 10/Aug/07 22:13 OPEN → PATCH AVAILABLE	32d 23h 16m	2
Owen O'Malley made transition - 10/Aug/07 23:51 PATCH AVAILABLE → RESOLVED	1h 38m	1
Nigel Daley made transition - 11/Aug/07 00:32 RESOLVED → REOPENED	41m 12s	1
Owen O'Malley made transition - 13/Aug/07 17:29 REOPENED → RESOLVED	2d 16h 57m	1
Doug Cutting made transition - 20/Aug/07 18:12 RESOLVED → CLOSED	7d 42m	1

People

Assignee:

Christopher Douglas

Reporter:

Christopher Douglas

Votes:

0 Vote for this issue

Watchers:

2 Start watching this issue

Dates

Created:

05/Jul/07 21:34

Updated:

02/May/13 02:29

Resolved:

13/Aug/07 17:29