Metacat - Bug #6662

Metacat fails large-file upload

02/05/2015 10:53 PM - Matt Jones

Status:	Closed	**Start date:**	02/05/2015
Priority:	Normal	**Due date:**	
Assignee:	Matt Jones	**% Done:**	100%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	2.4.3	**Spent time:**	0.00 hour
Bugzilla-Id:			

Description

Metacat seems to have a hard limit set on file upload size, at least for the DataONE MN.create() API. I attempted to call create() on a 4GiB file, which produced the error below in the logs.

Looking into the code, for Metacat 2.4.2, it appears the size limit is hardcoded on line 677 of D1ResourceHandler.java:

MultipartRequestResolver mrr =
new MultipartRequestResolver(tmpDir.getAbsolutePath(), 1000000000, 0);

To fix this, we should set a reasonable size that allows individual files to include typical multi-gigabyte-sized files. At a minimum this should be configurable and not hard-coded.

The produced error was:
org.dataone.service.exceptions.ServiceFailure: Could not resolve multipart files: the request was rejected because its size (1000001678) exceeds the configured maximum (1000000000)
at edu.ucsb.nceas.metacat.restservice.D1ResourceHandler.collectMultipartFiles(D1ResourceHandler.java:683)
at edu.ucsb.nceas.metacat.restservice.MNResourceHandler.putObject(MNResourceHandler.java:1381)
at edu.ucsb.nceas.metacat.restservice.MNResourceHandler.handle(MNResourceHandler.java:255)
at edu.ucsb.nceas.metacat.restservice.D1RestServlet.doPost(D1RestServlet.java:84)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:646)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:303)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:208)
at edu.ucsb.nceas.metacat.restservice.D1URLFilter.doFilter(D1URLFilter.java:48)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:241)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:208)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:220)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:122)
at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:501)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:170)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:98)
at org.apache.catalina.valves.AccessLogValve.invoke(AccessLogValve.java:950)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:116)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:408)
at org.apache.coyote.ajp.AjpProcessor.process(AjpProcessor.java:193)
at org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(AbstractProtocol.java:607)
at org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run(JIoEndpoint.java:313)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
at java.lang.Thread.run(Thread.java:745)

History

#1 - 02/06/2015 01:30 AM - Matt Jones

- Status changed from New to In Progress

- % Done changed from 0 to 90

Removed the hardcoding by adding a config parameter in commit r9094. Needs testing.

Should also discuss the alternative strategy, which would be to not have a hard limit at all, and instead use the HTTP Content-Length header to determine how large the max upload needs to be on a request. This might be an opening for a DOS though.

#2 - 02/06/2015 11:35 AM - Matt Jones

Although the value is no longer hardcoded in the source, it still is limited to 2GB because MultipartRequestResolver takes an int value for the max size, which in java has Integer.MAX_VALUE = 2147483647. Setting a value higher than this in the config file results in a parsing error. Changes to MultipartRequestResolver will be needed to allow a larger upload.

#3 - 02/10/2015 12:50 AM - Matt Jones

- Target version changed from 2.5.0 to 2.4.3

#4 - 02/10/2015 12:52 AM - Matt Jones

- Status changed from In Progress to Closed

- % Done changed from 90 to 100

Turns out that setting the value to -1 allows unlimited sized uploads (only limited by memory/disk). I refactored Metacat to move the size limit into the configuration file (metacat.properties), and tested it with a setting of -1 and was able to successfully upload a 4GB file from the R client. See https://dev.nceas.ucsb.edu/knb/d1/mn/v1/meta/urn:uuid:f61059ea-3dfb-486d-867e-7d9f59940a15