**Cheng, Loi**

October 17, 2019

Harvard Extension - Big Data Principles e88

Homework 7: Flume

* **Make sure you submit your solution document as a separate file in Canvas**
* **submit all your source code/ config files (if any) in a separate archive, named <LastName>\_<FirstName>\_HW7.zip**

Please identify which problems were completed. If any were incomplete, please identify where you encountered problems.

|  |
| --- |
| Problem 1: 100%  Problem 2: 100%  Problem 3: 100%  Problem 4: 100%  Problem 5 Bonus:  Homework was done on EC2 |

**Problem 1: Simple Flume Setup [25 points]**

Paste your Flume config file below [5 points]

|  |
| --- |
| # Licensed to the Apache Software Foundation (ASF) under one  # or more contributor license agreements.  See the NOTICE file  # distributed with this work for additional information  # regarding copyright ownership.  The ASF licenses this file  # to you under the Apache License, Version 2.0 (the  # "License"); you may not use this file except in compliance  # with the License.  You may obtain a copy of the License at  #  #  http://www.apache.org/licenses/LICENSE-2.0  #  # Unless required by applicable law or agreed to in writing,  # software distributed under the License is distributed on an  # "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY  # KIND, either express or implied.  See the License for the  # specific language governing permissions and limitations  # under the License.  # The configuration file needs to define the sources,  # the channels and the sinks.  # Sources, channels and sinks are defined per agent,  # in this case called 'a1’  a1.sources = r1  #a1.sinks = loggerSink  a1.sinks = filesink  a1.channels = memoryChannel  a1.sources.r1.type = spooldir  a1.sources.r1.spoolDir = /home/centos/hw7/p1\_source  a1.sources.r1.fileHeader = true  a1.sources.r1.channels = memoryChannel    # channel configuration  # Each channel's type must be defined.  a1.channels.memoryChannel.type = memory  a1.channels.memoryChannel.capacity = 1000  a1.channels.memoryChannel.transactionCapacity = 100  # Each sink's type must be defined  a1.sinks.loggerSink.type = logger  #Specify the channel the sink should use  a1.sinks.loggerSink.channel = memoryChannel  a1.sinks.filesink.type = file\_roll  a1.sinks.filesink.channel = memoryChannel  a1.sinks.filesink.sink.directory = /home/centos/hw7/p1\_sink  a1.sinks.filesink.sink.pathManager.extension = out  a1.sinks.filesink.sink.pathManager.prefix = assignment7\_ |

Show content of your source and sink directories before and after the Flume run. Compare the content by showing the line count for all files ('wc -l') These can be screen shots (ideally), or a copy/paste of console text. [5 points]

|  |
| --- |
| --- BEFORE RUN ---  Source    Sink    --- AFTER RUN ---  Source    Sink |

Paste the last 10 lines of the flume log file [5 points]

|  |
| --- |
|  |

Demo results of Experiment 1 - content of Flume's source and sink dirs, and relevant lines from its log file [5 points]

|  |
| --- |
| It looks like flume copied the contents of one of the files over from source to sink, but it crashed when it tried to rename the file from **file-input1.csv** to **file-input1.csv.COMPLETED** because a file with the same name already exists in the directory.  Source    Sink    Log |

Demo results of Experiment 2 - content of Flume's source and sink dirs, and relevant lines from its log file [5 points]

|  |
| --- |
| The new files were copied to the sink  --- FLUME STOPPED, NEW FILES COPIED ---  Source    Sink    Log    --- FLUME STARTED ---  Source    Sink    Log |

**Problem 2: Memory Channel Analysis [25 points]**

Paste your web server's index.html files below; demo access to this page - screenshot [1 point]

|  |
| --- |
|  |

Show corresponding entry in the web server log file [1 point]

|  |
| --- |
|  |

Paste content of your script/program that generates requests to your web server via 'curl' at a specified rate [1 point]

|  |
| --- |
| # Write a program that will issue the curl command  # at a rate of x times per second where x is a configurable parameter. The program should run in an endless loop.  import sys  import os  import time  from subprocess import call  rate = int(sys.argv[1])  # Time to sleep between requests  wait\_time = 1.0 / rate  # Write the output to the /dev/null as we are not interested in the content  # Count number of requests  count = 0  with open(os.devnull,'w') as DEVNULL:      while True:          call(['curl','-s','http://localhost:80'], stdout=DEVNULL)          count += 1          if count % rate == 0:              print(count)          time.sleep(wait\_time) |

Paste your Flume config file below [2 points]

|  |
| --- |
| # Licensed to the Apache Software Foundation (ASF) under one  # or more contributor license agreements.  See the NOTICE file  # distributed with this work for additional information  # regarding copyright ownership.  The ASF licenses this file  # to you under the Apache License, Version 2.0 (the  # "License"); you may not use this file except in compliance  # with the License.  You may obtain a copy of the License at  #  #  http://www.apache.org/licenses/LICENSE-2.0  #  # Unless required by applicable law or agreed to in writing,  # software distributed under the License is distributed on an  # "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY  # KIND, either express or implied.  See the License for the  # specific language governing permissions and limitations  # under the License.  # The configuration file needs to define the sources,  # the channels and the sinks.  # Sources, channels and sinks are defined per agent,  # in this case called 'a1’  a1.sources = r1  a1.sinks = filesink  a1.channels = memoryChannel  a1.sources.r1.type = exec  a1.sources.r1.channels = memoryChannel  a1.sources.r1.command = tail -F /var/log/httpd/access\_log  # channel configuration  # Each channel's type must be defined.  a1.channels.memoryChannel.type = memory  a1.channels.memoryChannel.capacity = 100  a1.channels.memoryChannel.transactionCapacity = 100  # Each sink's type must be defined  a1.sinks.loggerSink.type = logger  # Specify the channel the sink should use  a1.sinks.loggerSink.channel = memoryChannel  a1.sinks.filesink.type = file\_roll  a1.sinks.filesink.channel = memoryChannel  a1.sinks.filesink.sink.directory = /home/centos/hw7/p2\_sink  a1.sinks.filesink.sink.pathManager.extension = out  a1.sinks.filesink.sink.pathManager.prefix = assignment7\_ |

Demo your Flume topology working - compare content of the web server log with the resulting files in the sink directory [5 points]

|  |
| --- |
| Contents in the webserver log and the sink are the same  Web server log    Flume sink |

Demo results of the Experiment 1 - content of the web server logs (number of lines), Flume sink dir, relevant lines from Flume's log file [5 points]

|  |
| --- |
| ----- BEFORE chmod 444 -----  All the lines generated in access\_log are copied to the flume sink  access\_log    p2\_sink    ----- AFTER chmod 444 p2\_sink -----  Flume cannot write any more lines from access\_log to p2\_sink. The Flume log indicates an “Unable to deliver event” error.  access\_log    p2\_sink    Flume log    ----- AFTER chmod 777 p2\_sink -----  New \*.out file are being created in p2\_sink, but Flume still cannot write any more lines from access\_log to p2\_sink. The Flume log still indicates an “Unable to deliver event” error.  access\_log    p2\_sink    Flume log |

Demo results of the Experiment 2 - content of the web server logs (number of lines), Flume sink dir, relevant lines from Flume's log file; explain the results [5 points]

|  |
| --- |
| ----- BEFORE STOPPING FLUME -----  All the lines generated in access\_log are copied to the flume sink  access\_log    p2\_sink    Flume log    ---- FLUME STOPPED -----  New lines generated in access\_log are no longer copied to the flume sink  access\_log    p2\_sink    ----- FLUME RESTARTED -----  Flume resumes copying new lines from access\_log to the flume sink. The lines that were generated in the access\_log when flume was stopped did not get copied to the flume sink.  access\_log    p2\_sink |

Demo results of the Experiment 3 - content of the web server logs (number of lines), Flume sink dir, relevant lines from Flume's log file - at different event generation rates; explain the results [5 points]

|  |
| --- |
| ----------  Rate = 20 per second, capacities set at 10  Flume is not copying any lines from access\_log  a1.channels.memoryChannel.capacity = 10  a1.channels.memoryChannel.transactionCapacity = 10  a1.sinks.filesink.sink.batchSize = 10  access\_log    p2\_sink    Flume log    ----------  Rate = 20 per second, capacities are set at 100  Flume is operational, and copied about 1000 lines to the sink  a1.channels.memoryChannel.capacity = 100  a1.channels.memoryChannel.transactionCapacity = 100  a1.sinks.filesink.sink.batchSize = 100  p2\_sink    access\_log    Flume log    ----------  Rate = 50 per second, Flume crashes after copying about 181 lines to the sink  a1.channels.memoryChannel.capacity = 100  a1.channels.memoryChannel.transactionCapacity = 100  a1.sinks.filesink.sink.batchSize = 100  access\_log    p2\_sink    flume log    ----------  Rate = 40 per second, Flume is operational and copied 2000 lines to the sink  a1.channels.memoryChannel.capacity = 100  a1.channels.memoryChannel.transactionCapacity = 100  a1.sinks.filesink.sink.batchSize = 100  access\_log    p2\_sink    ----------  Rate = 1000 per second, Flume crashes after copying about 181 lines to the sink  a1.channels.memoryChannel.capacity = 100  a1.channels.memoryChannel.transactionCapacity = 100  a1.sinks.filesink.sink.batchSize = 100  access\_log    p2\_sink only copied a small portion of the access\_log    Flume log reports an error |

**Problem 3: File Channel Analysis [25 points]**

Paste your Flume config file below [2 points]

|  |
| --- |
| # Licensed to the Apache Software Foundation (ASF) under one  # or more contributor license agreements.  See the NOTICE file  # distributed with this work for additional information  # regarding copyright ownership.  The ASF licenses this file  # to you under the Apache License, Version 2.0 (the  # "License"); you may not use this file except in compliance  # with the License.  You may obtain a copy of the License at  #  #  http://www.apache.org/licenses/LICENSE-2.0  #  # Unless required by applicable law or agreed to in writing,  # software distributed under the License is distributed on an  # "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY  # KIND, either express or implied.  See the License for the  # specific language governing permissions and limitations  # under the License.  # The configuration file needs to define the sources,  # the channels and the sinks.  # Sources, channels and sinks are defined per agent,  # in this case called 'a1’  a1.sources = r1  #  a1.sinks = filesink  a1.channels = fc  a1.sources.r1.type = exec  a1.sources.r1.channels = fc  a1.sources.r1.command = tail -F /var/log/httpd/access\_log  # Each sink's type must be defined  a1.sinks.loggerSink.type = logger  #Specify the channel the sink should use  a1.sinks.loggerSink.channel = fc  a1.sinks.filesink.type = file\_roll  a1.sinks.filesink.channel = fc  a1.sinks.filesink.sink.directory = /home/centos/hw7/p3\_sink  a1.sinks.filesink.sink.pathManager.extension = out  a1.sinks.filesink.sink.pathManager.prefix = assignment7  a1.sinks.filesink.sink.rollInterval = 10  # channel configuration  a1.channels.fc.type = file  a1.channels.fc.capacity = 100  a1.channels.fc.transactionCapacity = 100  a1.channels.fc.checkpointDir = /home/centos/hw7/p3\_checkpoint  a1.channels.fc.dataDirs = /home/centos/hw7/p3\_data |

Demo your Flume topology working - compare content of the web server log with the resulting files in the sink directory [3 points]

|  |
| --- |
| All the events from the web server log are copied to the file sink  Web server log    File sink |

Demo results of the Experiment 1 - content of the web server logs (number of lines), Flume sink dir, relevant lines from Flume's log file and content of the 'checkpointDir' and 'dataDir'; explain the results; [5 points]

|  |
| --- |
| Before chmod 444 of file sink, events from the web server log are copied to the sink as expected    After chmod 444 of file sink, Flume logs an error    Checkpoint and data folder show 1 checkpoint was created    After chmod 777 of sink, flume resumes from checkpoint, and adds 100 events from the checkpoint to file sink      Most of the lines from the access log were not copied to the sink because flume did not have access to the sink folder |

What do you need to do to save more events? [2 points]

|  |
| --- |
| Because we set the file capacity to 100, the checkpoint was able to only save at most 100 events from the access\_log that were created after flume lost access to the sink folder. We can increase the capacity to save more events  a1.channels.fc.capacity = 100  After restarting the flume agent, lines are being copied to the sink again |

Demo results of the Experiment 2 - content of the web server logs (number of lines), Flume sink dir, relevant lines from Flume's log file and content of the 'checkpointDir' and 'dataDir'; explain any difference in behavior you may see with Experiment 1 [5 points]

|  |
| --- |
| About 2300 events from the web log were not copied from to the sink during the time flume was stopped. Once flume was started again, events were copied to the sink again. The behavior is very similar to Problem 2 Experiment 2.  Flume agent stopped  Flume log, web log    File sink , checkpoint, data dir    Flume agent started again  Flume log, web log    File sink , checkpoint, data dir    Flume agent stopped and curl stopped  Flume log, web log    File sink , checkpoint, data dir |

Demo results of the Experiment 3 - with different event generation rates and 'capacity' settings of the channel; show relevant changes in the content of the 'checkpointDir' and 'dataDir', Flume sink dir, relevant lines from Flume's log file; explain the results [5 points]

|  |
| --- |
| -----------  Capacity = 100 events  a1.channels.fc.capacity = 100  20 events per second – Flume is operational, all events are copied    50 events per second – Flume is operational, all events are copied    70 events per second – Flume crashed, events after crash were not copied to sink    100 events per second – Flume crashed, events after crash were not copied to sink    -----------  Capacity = 10000 events  a1.channels.fc.capacity = 10000  20 per second - OK    50 per second - OK    100 per second - OK    1000 per second – OK, but the machine cannot actually generate 1000 events per seconds    5000 per second – OK, but the machine cannot actually generate 5000 events per seconds    Console log shows it took about 48 seconds to generate 5000 events, which means the machine can at most generate about 100 events per second |

At what point do you start seeing lost data, if any? [3 points]

|  |
| --- |
| With channel capacity at 100, flume crashes and loses data at rates around 70-100 per second. It seems event rate should be ideally about half of the capacity or lower for the sink to be able to take the events out of the channel.  With channel capacity at 10000, flume does not crash and lose data. This is probably because the machine can generate about 100 events per second at most, and the channel capacity is more than sufficient at this rate. |

**Problem 4: Flume with UUID interceptor [25 points]**

Paste your Flume config file below [10 points]

|  |
| --- |
| # Licensed to the Apache Software Foundation (ASF) under one  # or more contributor license agreements.  See the NOTICE file  # distributed with this work for additional information  # regarding copyright ownership.  The ASF licenses this file  # to you under the Apache License, Version 2.0 (the  # "License"); you may not use this file except in compliance  # with the License.  You may obtain a copy of the License at  #  #  http://www.apache.org/licenses/LICENSE-2.0  #  # Unless required by applicable law or agreed to in writing,  # software distributed under the License is distributed on an  # "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY  # KIND, either express or implied.  See the License for the  # specific language governing permissions and limitations  # under the License.  # The configuration file needs to define the sources,  # the channels and the sinks.  # Sources, channels and sinks are defined per agent,  # in this case called 'a1’  a1.sources = r1  #  a1.sinks = filesink  a1.channels = fc  a1.sources.r1.type = exec  a1.sources.r1.channels = fc  a1.sources.r1.command = tail -F /var/log/httpd/access\_log  # Interceptor  a1.sources.r1.interceptors = i1  a1.sources.r1.interceptors.i1.type = org.apache.flume.sink.solr.morphline.UUIDInterceptor$Builder  a1.sources.r1.interceptors.i1.headerName = UUID  # Each sink's type must be defined  a1.sinks.loggerSink.type = logger  # Specify the channel the sink should use  a1.sinks.loggerSink.channel = fc  a1.sinks.filesink.type = file\_roll  a1.sinks.filesink.channel = fc  a1.sinks.filesink.sink.directory = /home/centos/hw7/p4\_sink  a1.sinks.filesink.sink.pathManager.extension = out  a1.sinks.filesink.sink.pathManager.prefix = hw7p4\_  # set roll to 10 seceonds  a1.sinks.filesink.sink.rollInterval = 10  # add header and text to sink  a1.sinks.filesink.sink.serializer = header\_and\_text  a1.sinks.filesink.sink.serializer.appendNewline = true  # channel configuration  a1.channels.fc.type = file  a1.channels.fc.capacity = 100  a1.channels.fc.transactionCapacity = 100  a1.channels.fc.checkpointDir = /home/centos/hw7/p4\_checkpoint  a1.channels.fc.dataDirs = /home/centos/hw7/p4\_data |

Demo that the resulting events in the sink now have UUIDs [15 points]

|  |
| --- |
|  |

**Problem 5: Bonus: Flume with HDFS sink [+15 points]**

Paste your Flume config file below

|  |
| --- |
|  |

Demo your Flume topology working at a rate of 100 requests/events per second - compare content of the web server log with the resulting files in the sink HDFS directories

|  |
| --- |
|  |

Demo that correct events are stored into the correct corresponding HDFS folders/files

|  |
| --- |
|  |