

Flume and HDFS performance insights

Omid Vahdaty, Big Data Ninja



Test Details

- **Fan In** architecture
- **Collector** sending data, Input 100GB of pcap files, rate in ingress: 2MB/s.
- **MiniFlume** : Thrift source , mem channel, AVRO sink.
- **MegaFlume**: AVRO source, mem channel, HDFS Sink
- **Hadoop**
 - 4 DataNode
 - 1 NameNode



Mini Flume config



- i. Virtual Machine: 8 CPU core, 16GB RAM. Ubuntu 14.04.
- i. Environment: `export JAVA_OPTS="-Xms100m -Xmx12000m Dcom.sun.management.jmxremote`
- i. Config
 - a. 2X thrift sources. thread count to 8.
 - b. 2X avro sinks. roll size 10000.
 - c. `memory-channel.capacity = 100100`
 - d. `memory-channel.transactionCapacity=10010`

Mini Flume File channel config

`agent.channels.k1.type = file`

`agent.channels.k1.capacity = 82000000`

`agent.channels.k1.transactionCapacity = 2000009`

`agent.channels.k1.checkpointDir = /hadoop-data/hadoopuser/flume/tmp/checkpoint1`

`agent.channels.k1.dataDirs = /hadoop-data/hadoopuser/flume/data/dn1`

`agent.channels.k1.useDualCheckpoints=true`

`agent.channels.k1.backupCheckpointDir = /hadoop-data/hadoopuser/flume/tmp/checkpoint11`

`agent.channels.k1.use-fast-replay = true`

`agent.channels.k1.checkpointInterval=60000`



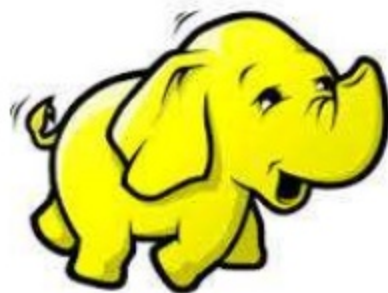
Mega Flume config

- i. Virtual Machine: **16 CPU core**, 16GB RAM.
- ii. Environment: export JAVA_OPTS="-Xms100m -Xmx12000m -Dcom.sun.management.jmxremote
- iii. mem channel
 - a. memory-channel.capacity = 6200000
 - b. memory-channel.transactionCapacity=620000
- iv. 2 avro sources 8 threads
- v. 2 HDFS sinks
 - 1. agent.sinks.hdfs-sink.hdfs.path = hdfs://master1:54310/data/EthernetCollector
 - 2. agent.sinks.hdfs-sink.hdfs.fileType = DataStream
 - 3. agent.sinks.hdfs-sink.hdfs.useLocalTimeStamp = true
 - 4. agent.sinks.hdfs-sink.hdfs.filePrefix = ethernet
 - 5. agent.sinks.hdfs-sink.hdfs.fileSuffix = .avro
 - 6. agent.sinks.hdfs-sink.hdfs.inUsePrefix = part.
 - 7. agent.sinks.hdfs-sink.serializer = avro_event
 - 8. agent.sinks.hdfs-sink.hdfs.minBlockReplicas= 1
 - 9. **agent.sinks.hdfs-sink.hdfs.threadPoolSize= 16**
 - 10. **agent.sinks.hdfs-sink.hdfs.rollCount = 250030**
 - 11. **agent.sinks.hdfs-sink.hdfs.rollSize = 0**
 - 12. **agent.sinks.hdfs-sink.hdfs.batchSize = 620000**
 - 13. **agent.sinks.hdfs-sink.hdfs.rollInterval = 0**



Hadoop Cluster

- i. 4 hadoop data nodes , **12 Disks per node**, 4 cores. 8GB RAM.
- ii. Single Namenode, not HA , no Yarn. no ZooKeeper.



Flume Performance insights

- i. thrift source has threads!!!
- ii. VERY consuming in terms of resources
- iii. **file channel** is MUCH **slower** than **mem channel**
- iv. **sink HDFS** , stabilizes after long time (~ 30 min), each change takes time.
 - 1. buffering into RAM – the RAM graphs goes up.
 - 2. Stable ingress rate - RAM consumption is parallel line to the time axis.
 - 3. small roll size - may crash the cluster
 - 4. too much "pressure" on hadoop crushes the cluster, causes data-node loss. it even cause the Name-node to enter safe mode, or even loss ALL data nodes.
 - 5. Rule of thumb- for each sink - at least 2 data nodes with at least 2 data disks.
 - 6. each batch request is divided to several threads. i.e 2MB/s is write speed on hadoop per sink per node in parallel. read a little about flume performance metrics:
<https://cwiki.apache.org/confluence/display/FLUME/Performance+Measurements+-+round+2>
 - 7. notice the above article: 20 data-node, 8 disks, 8 sinks.



Flume Performance insights

- i. each flume should be considered for tuning of ingress and egress on same node. monitor the RAM metrics via data-dog or another monitoring tool , and line should be parallel to time axis at all time.
- ii. each source/sink may have threads - significant impact on performance.
- iii. when increase batch size - all other parameters should increase with similar ratio.
- iv. be sure to understand difference of batch vs. roll.
- v. use unique numbers for debugging.
- vi. each event size changes the buffering in the flume drastically. as long as it bound by min/max values, your config is ok
- vii. Consider giving unique prefix names per sink - multiply your tmp files in parallel.



File channel insights



- Consider 2 partitions per file channel : data and checkpoint
- Don't Consider file channels per port - unless you have separate partitions per file channel.
- There is a limit in max capacity the flume can handle. - if you need to buffer a week of down time, consider scaling out VIA fan in
- Consider Max file capacity written to disk - to be below RAM size, to utilize OS caching.very efficient
- Consider increasing the transaction Capacity to 1M events for fast recovery from filechannel reply.



File channel insights

- The File Channel takes a **comma-separated list of data directories** as the value of the **dataDirs** parameter
- If the channel is stopped while it is checkpointing, the checkpoint may be incomplete or corrupt. A corrupt or incomplete checkpoint could make the restart of the channel extremely slow, as the channel would need to read and replay all data files. To avoid this problem, it is recommended that the **useDualCheckpoints** parameter be set to **true** and that **backupCheckpointDir** be set
- It will always retain two files per data directory, even if the files do not have any events to be taken and committed. The channel will also not delete the file currently being written to.
- Using NFS-mounted disks with the File Channel is not a good idea
- Read: <https://www.safaribooksonline.com/library/view/using-flume/9781491905326/ch04.html>
- Increase checkpoint interval if you use dual backup. (checkpoint time was doubled if you use dual backup)

Hadoop Performance Insights



Hadoop

- i. more data node disk increases performance
- ii. hadoop was designed for parallelism.but flume sinks - are not very powerful. Yes you can add more sinks - but - you need stronger cluster - 1 sink = 2 data nodes with at least 2 data partions.
- iii. you can add nodes dynamically easily while the cluster is running , so no need to restart cluster.
- iv. Increase IO buffer to 64KB or 128KB (assuming large block size)
- v. Increase NN handlers to 20 X number datanode


General Performance Insights

Generally speaking:

- i. very HARD to simulate the situation with 1 server. I over committed resources, causing failures in the HDFS.
- ii. the amount of thread is ENORMOUS! but very light, and short spanned. not CPU intensive.
- iii. no Egress to engine was tested in this scenario
- iv. not data correctness was tested.
- v. very hard to fine tune flume - each change on file based sinks - take about 10 min to reflect/stabilize in monitoring (unless it crushed first).



Stay in touch...

- [Omid Vahdaty](#) 
- +972-54-2384178



 **HALO**
ANALYTICS



The logo for Jajah Telefonica. It features the word 'jajah' in a stylized, pink, lowercase font, with a small grey speech bubble icon above the 'j'. Below it, the word 'Telefonica' is written in a black, cursive script font.