

SSTable Tools

SSTable Tools

This section describes the functionality of the various **sstable** tools.

Cassandra must be stopped before these tools are executed, or unexpected results will occur. Note: the scripts do not verify that Cassandra is stopped.

- **sstabledump**
- **sstableexpiredblockers**
- **sstablelevelreset**
- **sstableloader**
- **sstablemetadata**
- **sstableofflinerelevel**
- **sstablepartitions**
- **sstablerepairedset**
- **sstablescrub**
- **sstablesplit**
- **sstableupgrade**
- **sstableutil**
- **sstableverify**

sstabledump

sstabledump

Dump contents of a given SSTable to standard output in JSON format.

You must supply exactly one sstable.

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstabledump <options> <sstable file path>

-d	CQL row per line internal representation
-e	Enumerate partition keys only
-k <arg>	Partition key
-x <arg>	Excluded partition key(s)
-t	Print raw timestamps instead of iso8601 date strings
-l	Output each row as a separate JSON object

If necessary, use sstableutil first to find out the sstables used by a table.

Dump entire table

Dump the entire table without any options.

Example:

```
sstabledump /var/lib/cassandra/data/keyspace/eventlog-
65c429e08c5a11e8939edf4f403979ef/mc-1-big-Data.db > eventlog_dump_2018Jul26

cat eventlog_dump_2018Jul26
[
  {
    "partition" : {
      "key" : [ "3578d7de-c60d-4599-ae fb-3f22a07b2bc6" ],
      "position" : 0
    },
    "rows" : [
      {
```

```

    "type" : "row",
    "position" : 61,
    "liveness_info" : { "tstamp" : "2018-07-20T20:23:08.378711Z" },
    "cells" : [
      { "name" : "event", "value" : "party" },
      { "name" : "insertedtimestamp", "value" : "2018-07-20 20:23:08.384Z" },
      { "name" : "source", "value" : "asdf" }
    ]
  }
]
},
{
  "partition" : {
    "key" : [ "d18250c0-84fc-4d40-b957-4248dc9d790e" ],
    "position" : 62
  },
  "rows" : [
    {
      "type" : "row",
      "position" : 123,
      "liveness_info" : { "tstamp" : "2018-07-20T20:23:07.783522Z" },
      "cells" : [
        { "name" : "event", "value" : "party" },
        { "name" : "insertedtimestamp", "value" : "2018-07-20 20:23:07.789Z" },
        { "name" : "source", "value" : "asdf" }
      ]
    }
  ]
},
{
  "partition" : {
    "key" : [ "cf188983-d85b-48d6-9365-25005289beb2" ],
    "position" : 124
  },
  "rows" : [
    {
      "type" : "row",
      "position" : 182,
      "liveness_info" : { "tstamp" : "2018-07-20T20:22:27.028809Z" },
      "cells" : [
        { "name" : "event", "value" : "party" },
        { "name" : "insertedtimestamp", "value" : "2018-07-20 20:22:27.055Z" },
        { "name" : "source", "value" : "asdf" }
      ]
    }
  ]
}
}

```

Dump table in a more manageable format

Use the `-l` option to dump each row as a separate JSON object. This will make the output easier to manipulate for large data sets. ref: <https://issues.apache.org/jira/browse/CASSANDRA-13848>

Example:

```
sstabledump /var/lib/cassandra/data/keyspace/eventlog-
65c429e08c5a11e8939edf4f403979ef/mc-1-big-Data.db -l > eventlog_dump_2018Jul26_justlines

cat eventlog_dump_2018Jul26_justlines
[
  {
    "partition" : {
      "key" : [ "3578d7de-c60d-4599-ae6b-3f22a07b2bc6" ],
      "position" : 0
    },
    "rows" : [
      {
        "type" : "row",
        "position" : 61,
        "liveness_info" : { "timestamp" : "2018-07-20T20:23:08.378711Z" },
        "cells" : [
          { "name" : "event", "value" : "party" },
          { "name" : "insertedtimestamp", "value" : "2018-07-20 20:23:08.384Z" },
          { "name" : "source", "value" : "asdf" }
        ]
      }
    ]
  },
  {
    "partition" : {
      "key" : [ "d18250c0-84fc-4d40-b957-4248dc9d790e" ],
      "position" : 62
    },
    "rows" : [
      {
        "type" : "row",
        "position" : 123,
        "liveness_info" : { "timestamp" : "2018-07-20T20:23:07.783522Z" },
        "cells" : [
          { "name" : "event", "value" : "party" },
          { "name" : "insertedtimestamp", "value" : "2018-07-20 20:23:07.789Z" },
```

```

        { "name" : "source", "value" : "asdf" }
      ]
    }
  ]
},
{
  "partition" : {
    "key" : [ "cf188983-d85b-48d6-9365-25005289beb2" ],
    "position" : 124
  },
  "rows" : [
    {
      "type" : "row",
      "position" : 182,
      "liveness_info" : { "tstamp" : "2018-07-20T20:22:27.028809Z" },
      "cells" : [
        { "name" : "event", "value" : "party" },
        { "name" : "insertedtimestamp", "value" : "2018-07-20 20:22:27.055Z" },
        { "name" : "source", "value" : "asdf" }
      ]
    }
  ]
}
]
}

```

Dump only keys

Dump only the keys by using the `-e` option.

Example:

```

sstabledump /var/lib/cassandra/data/keyspace/eventlog-
65c429e08c5a11e8939edf4f403979ef/mc-1-big-Data.db -e > eventlog_dump_2018Jul26_justkeys

cat eventlog_dump_2018Jul26b
[ [ "3578d7de-c60d-4599-aefb-3f22a07b2bc6" ], [ "d18250c0-84fc-4d40-b957-4248dc9d790e" ],
[ "cf188983-d85b-48d6-9365-25005289beb2" ] ]

```

Dump row for a single key

Dump a single key using the `-k` option.

Example:

```
sstabledump /var/lib/cassandra/data/keyspace/eventlog-
65c429e08c5a11e8939edf4f403979ef/mc-1-big-Data.db -k 3578d7de-c60d-4599-aefb-3f22a07b2bc6
> eventlog_dump_2018Jul26_singlekey

cat eventlog_dump_2018Jul26_singlekey
[
  {
    "partition" : {
      "key" : [ "3578d7de-c60d-4599-aefb-3f22a07b2bc6" ],
      "position" : 0
    },
    "rows" : [
      {
        "type" : "row",
        "position" : 61,
        "liveness_info" : { "tstamp" : "2018-07-20T20:23:08.378711Z" },
        "cells" : [
          { "name" : "event", "value" : "party" },
          { "name" : "insertedtimestamp", "value" : "2018-07-20 20:23:08.384Z" },
          { "name" : "source", "value" : "asdf" }
        ]
      }
    ]
  }
]
```

Exclude a key or keys in dump of rows

Dump a table except for the rows excluded with the -x option. Multiple keys can be used.

Example:

```
sstabledump /var/lib/cassandra/data/keyspace/eventlog-
65c429e08c5a11e8939edf4f403979ef/mc-1-big-Data.db -x 3578d7de-c60d-4599-aefb-3f22a07b2bc6
d18250c0-84fc-4d40-b957-4248dc9d790e > eventlog_dump_2018Jul26_excludekeys

cat eventlog_dump_2018Jul26_excludekeys
[
  {
    "partition" : {
      "key" : [ "cf188983-d85b-48d6-9365-25005289beb2" ],
      "position" : 0
    },
    "rows" : [
      {
        "type" : "row",
```



```

    "position" : 182,
    "liveness_info" : { "tstamp" : "2018-07-20T20:22:27.028809Z" },
    "cells" : [
      { "name" : "event", "value" : "party" },
      { "name" : "insertedtimestamp", "value" : "2018-07-20 20:22:27.055Z" },
      { "name" : "source", "value" : "asdf" }
    ]
  }
]
}

```

Display raw timestamps

By default, dates are displayed in iso8601 date format. Using the `-t` option will dump the data with the raw timestamp.

Example:

```

sstabledump /var/lib/cassandra/data/keyspace/eventlog-
65c429e08c5a11e8939edf4f403979ef/mc-1-big-Data.db -t -k cf188983-d85b-48d6-9365-
25005289beb2 > eventlog_dump_2018Jul26_times

cat eventlog_dump_2018Jul26_times
[
  {
    "partition" : {
      "key" : [ "cf188983-d85b-48d6-9365-25005289beb2" ],
      "position" : 124
    },
    "rows" : [
      {
        "type" : "row",
        "position" : 182,
        "liveness_info" : { "tstamp" : "1532118147028809" },
        "cells" : [
          { "name" : "event", "value" : "party" },
          { "name" : "insertedtimestamp", "value" : "2018-07-20 20:22:27.055Z" },
          { "name" : "source", "value" : "asdf" }
        ]
      }
    ]
  }
]
}

```

Display internal structure in output

Dump the table in a format that reflects the internal structure.

Example:

```
sstabledump /var/lib/cassandra/data/keyspace/eventlog-
65c429e08c5a11e8939edf4f403979ef/mc-1-big-Data.db -d > eventlog_dump_2018Jul26_d

cat eventlog_dump_2018Jul26_d
[3578d7de-c60d-4599-ae6b-3f22a07b2bc6]@0 Row[info=[ts=1532118188378711] ]: |
[event=party ts=1532118188378711], [insertedtimestamp=2018-07-20 20:23Z
ts=1532118188378711], [source=asdf ts=1532118188378711]
[d18250c0-84fc-4d40-b957-4248dc9d790e]@62 Row[info=[ts=1532118187783522] ]: |
[event=party ts=1532118187783522], [insertedtimestamp=2018-07-20 20:23Z
ts=1532118187783522], [source=asdf ts=1532118187783522]
[cf188983-d85b-48d6-9365-25005289beb2]@124 Row[info=[ts=1532118147028809] ]: |
[event=party ts=1532118147028809], [insertedtimestamp=2018-07-20 20:22Z
ts=1532118147028809], [source=asdf ts=1532118147028809]
```

sstableexpiredblockers

sstableexpiredblockers

During compaction, entire sstables can be dropped if they contain only expired tombstones, and if it is guaranteed that the data is not newer than the data in other sstables. An expired sstable can be blocked from getting dropped if its newest timestamp is newer than the oldest data in another sstable.

This tool is used to list all sstables that are blocking other sstables from getting dropped (by having older data than the newest tombstone in an expired sstable) so a user can figure out why certain sstables are still on disk.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-10015>

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstableexpiredblockers <keyspace> <table>

Output blocked sstables

If the sstables exist for the table, but no tables have older data than the newest tombstone in an expired sstable, the script will return nothing.

Otherwise, the script will return <sstable> blocks <> expired sstables from getting dropped# followed by a list of the blocked sstables.

Example:

```
sstableexpiredblockers keyspace1 standard1
```

```
[BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-0665ae80b2d711e886c66d2c86545d91/mc-2-big-Data.db') (minTS = 5, maxTS = 5, maxLDT = 2147483647)], blocks 1 expired sstables from getting dropped:
[BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-0665ae80b2d711e886c66d2c86545d91/mc-3-big-Data.db') (minTS = 1536349775157606, maxTS = 1536349780311159, maxLDT = 1536349780)],

[BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-0665ae80b2d711e886c66d2c86545d91/mc-1-big-Data.db') (minTS = 1, maxTS = 10, maxLDT = 2147483647)], blocks 1 expired sstables from getting dropped:
[BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-
```

```
0665ae80b2d711e886c66d2c86545d91/mc-3-big-Data.db') (minTS = 1536349775157606, maxTS =  
1536349780311159, maxLDT = 1536349780)],
```

sstablelevelreset

sstablelevelreset

If LeveledCompactionStrategy is set, this script can be used to reset level to 0 on a given set of sstables. This is useful if you want to, for example, change the minimum sstable size, and therefore restart the compaction process using this new configuration.

See <https://cassandra.apache.org/doc/latest/operating/compaction/lcs.html#lcs> for information on how levels are used in this compaction strategy.

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-5271>

Usage

```
sstablelevelreset --really-reset <keyspace> <table>
```

The really-reset flag is required, to ensure this intrusive command is not run accidentally.

Table not found

If the keyspace and/or table is not in the schema (e.g., if you misspelled the table name), the script will return an error.

Example:

```
ColumnFamily not found: keyspace/eventlog.
```

Table has no sstables

Example:

```
Found no sstables, did you give the correct keyspace/table?
```

Table already at level 0

The script will not set the level if it is already set to 0.

Example:

```
Skipped /var/lib/cassandra/data/keyspace/eventlog-65c429e08c5a11e8939edf4f403979ef/mc-1-big-Data.db since it is already on level 0
```

Table levels reduced to 0

If the level is not already 0, then this will reset it to 0.

Example:

```
sstablemetadata /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-8-big-Data.db | grep -i level
SSTable Level: 1

sstablelevelreset --really-reset keyspace eventlog
Changing level from 1 to 0 on /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-8-big-Data.db

sstablemetadata /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-8-big-Data.db | grep -i level
SSTable Level: 0
```


sstableloader

sstableloader

Bulk-load the sstables found in the directory <dir_path> to the configured cluster. The parent directories of <dir_path> are used as the target keyspace/table name. For example, to load an sstable named ma-1-big-Data.db into keyspace1/standard1, you will need to have the files ma-1-big-Data.db and ma-1-big-Index.db in a directory /path/to/keyspace1/standard1/. The tool will create new sstables, and does not clean up your copied files.

Several of the options listed below don't work quite as intended, and in those cases, workarounds are mentioned for specific use cases.

To avoid having the sstable files to be loaded compacted while reading them, place the files in an alternate keyspace/table path than the data directory.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-1278>

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstableloader <options> <dir_path>

-d, --nodes <initial hosts>	Required. Try to connect to these hosts (comma-separated) initially for ring information
-u, --username <username>	username for Cassandra authentication
-pw, --password <password>	password for Cassandra authentication
-p, --port <native transport port>	port used for native connection (default 9042)
-sp, --storage-port <storage port>	port used for internode communication (default 7000)
-ssp, --ssl-storage-port <ssl storage port>	port used for TLS internode communication (default 7001)
--no-progress	don't display progress
-t, --throttle <throttle>	(deprecated) throttle speed in Mbits (default 0 for unlimited) Use --throttle-mib instead
--throttle-mib <throttle-mib>	throttle speed in MiB/s (default 0 for unlimited)

-idct, --inter-dc-throttle <inter-dc-throttle>	(deprecated) inter-datacenter throttle speed in Mbits (default 0 for unlimited) Use --inter-dc-throttle-mib instead
--inter-dc-throttle-mib <inter-dc-throttle-mib>	inter-datacenter throttle speed in MiB/s (default 0 for unlimited)
--entire-sstable-throttle-mib <throttle-mib>	entire SSTable throttle speed in MiB/s (default 0 for unlimited)
--entire-sstable-inter-dc-throttle-mib <inter-dc-throttle-mib>	entire SSTable inter-datacenter throttle speed in MiB/s (default 0 for unlimited)
-cph, --connections-per-host <connectionsPerHost>	number of concurrent connections-per-host
-i, --ignore <NODES>	don't stream to this (comma separated) list of nodes
-alg, --ssl-alg <ALGORITHM>	Client SSL: algorithm (default: SunX509)
-ciphers, --ssl-ciphers <CIPHER-SUITES>	Client SSL: comma-separated list of encryption suites to use
-ks, --keystore <KEYSTORE>	Client SSL: full path to keystore
-kspw, --keystore-password <KEYSTORE-PASSWORD>	Client SSL: password of the keystore
-st, --store-type <STORE-TYPE>	Client SSL: type of store
-ts, --truststore <TRUSTSTORE>	Client SSL: full path to truststore
-tspw, --truststore-password <TRUSTSTORE-PASSWORD>	Client SSL: password of the truststore
-prtcl, --ssl-protocol <PROTOCOL>	Client SSL: connections protocol to use (default: TLS)
-ap, --auth-provider <auth provider>	custom AuthProvider class name for cassandra authentication
-f, --conf-path <path to config file>	cassandra.yaml file path for streaming throughput and client/server SSL
-v, --verbose	verbose output
-h, --help	display this help message

You can provide a `cassandra.yaml` file with the `-f` command line option to set up streaming throughput, and client and server encryption options. Only `stream_throughput_outbound`, `server_encryption_options`, and `client_encryption_options` are read from `yaml`. You can override options read from `cassandra.yaml` with corresponding command line options.

Load sstables from a Snapshot

Copy the snapshot sstables into an accessible directory and use sstableloader to restore them.

Example:

```
cp snapshots/1535397029191/* /path/to/keyspace1/standard1/

sstableloader --nodes 172.17.0.2 /var/lib/cassandra/loadme/keyspace1/standard1-
f8a4fa30aa2a11e8af27091830ac5256/
Established connection to initial hosts
Opening sstables and calculating sections to stream
Streaming relevant part of /var/lib/cassandra/loadme/keyspace1/standard1-
f8a4fa30aa2a11e8af27091830ac5256/ma-3-big-Data.db to [/172.17.0.2]
progress: [/172.17.0.2]0:1/1 100% total: 100% 0 MB/s(avg: 1 MB/s)
Summary statistics:
  Connections per host:      : 1
  Total files transferred:   : 1
  Total bytes transferred:   : 4700000
  Total duration (ms):      : 4390
  Average transfer rate (MB/s): : 1
  Peak transfer rate (MB/s): : 1
```

The -d or --nodes option is required, or the script will not run.

Example:

```
sstableloader /var/lib/cassandra/loadme/keyspace1/standard1-
f8a4fa30aa2a11e8af27091830ac5256/
Initial hosts must be specified (-d)
```

Use a Config File for SSL Clusters

If SSL encryption is enabled in the cluster, use the --conf-path option with sstableloader to point the tool to the cassandra.yaml with the relevant server_encryption_options (e.g., truststore location, algorithm). This will work better than passing individual ssl options shown above to sstableloader on the command line.

Example:

```
sstableloader --nodes 172.17.0.2 --conf-path /etc/cassandra/cassandra.yaml
/var/lib/cassandra/loadme/keyspace1/standard1-0974e5a0aa5811e8a0a06d2c86545d91/snapshots/
Established connection to initial hosts
```

```
Opening sstables and calculating sections to stream
Streaming relevant part of /var/lib/cassandra/loadme/keyspace1/standard1-
0974e5a0aa5811e8a0a06d2c86545d91/mc-1-big-Data.db to [/172.17.0.2]
progress: [/172.17.0.2]0:0/1 1 % total: 1% 9.165KiB/s (avg: 9.165KiB/s)
progress: [/172.17.0.2]0:0/1 2 % total: 2% 5.147MiB/s (avg: 18.299KiB/s)
progress: [/172.17.0.2]0:0/1 4 % total: 4% 9.751MiB/s (avg: 27.423KiB/s)
progress: [/172.17.0.2]0:0/1 5 % total: 5% 8.203MiB/s (avg: 36.524KiB/s)
...
progress: [/172.17.0.2]0:1/1 100% total: 100% 0.000KiB/s (avg: 480.513KiB/s)
```

Summary statistics:

```
Connections per host      : 1
Total files transferred   : 1
Total bytes transferred   : 4.387MiB
Total duration            : 9356 ms
Average transfer rate     : 480.105KiB/s
Peak transfer rate        : 586.410KiB/s
```

Hide Progress Output

To hide the output of progress and the summary statistics (e.g., if you wanted to use this tool in a script), use the `--no-progress` option.

Example:

```
sstableloader --nodes 172.17.0.2 --no-progress
/var/lib/cassandra/loadme/keyspace1/standard1-f8a4fa30aa2a11e8af27091830ac5256/
Established connection to initial hosts
Opening sstables and calculating sections to stream
Streaming relevant part of /var/lib/cassandra/loadme/keyspace1/standard1-
f8a4fa30aa2a11e8af27091830ac5256/ma-4-big-Data.db to [/172.17.0.2]
```

Get More Detail

Using the `--verbose` option will provide much more progress output.

Example:

```
sstableloader --nodes 172.17.0.2 --verbose /var/lib/cassandra/loadme/keyspace1/standard1-
0974e5a0aa5811e8a0a06d2c86545d91/
Established connection to initial hosts
Opening sstables and calculating sections to stream
Streaming relevant part of /var/lib/cassandra/loadme/keyspace1/standard1-
```

```
0974e5a0aa5811e8a0a06d2c86545d91/mc-1-big-Data.db to [/172.17.0.2]
progress: [/172.17.0.2]0:0/1 1 % total: 1% 12.056KiB/s (avg: 12.056KiB/s)
progress: [/172.17.0.2]0:0/1 2 % total: 2% 9.092MiB/s (avg: 24.081KiB/s)
progress: [/172.17.0.2]0:0/1 4 % total: 4% 18.832MiB/s (avg: 36.099KiB/s)
progress: [/172.17.0.2]0:0/1 5 % total: 5% 2.253MiB/s (avg: 47.882KiB/s)
progress: [/172.17.0.2]0:0/1 7 % total: 7% 6.388MiB/s (avg: 59.743KiB/s)
progress: [/172.17.0.2]0:0/1 8 % total: 8% 14.606MiB/s (avg: 71.635KiB/s)
progress: [/172.17.0.2]0:0/1 9 % total: 9% 8.880MiB/s (avg: 83.465KiB/s)
progress: [/172.17.0.2]0:0/1 11 % total: 11% 5.217MiB/s (avg: 95.176KiB/s)
progress: [/172.17.0.2]0:0/1 12 % total: 12% 12.563MiB/s (avg: 106.975KiB/s)
progress: [/172.17.0.2]0:0/1 14 % total: 14% 2.550MiB/s (avg: 118.322KiB/s)
progress: [/172.17.0.2]0:0/1 15 % total: 15% 16.638MiB/s (avg: 130.063KiB/s)
progress: [/172.17.0.2]0:0/1 17 % total: 17% 17.270MiB/s (avg: 141.793KiB/s)
progress: [/172.17.0.2]0:0/1 18 % total: 18% 11.280MiB/s (avg: 153.452KiB/s)
progress: [/172.17.0.2]0:0/1 19 % total: 19% 2.903MiB/s (avg: 164.603KiB/s)
progress: [/172.17.0.2]0:0/1 21 % total: 21% 6.744MiB/s (avg: 176.061KiB/s)
progress: [/172.17.0.2]0:0/1 22 % total: 22% 6.011MiB/s (avg: 187.440KiB/s)
progress: [/172.17.0.2]0:0/1 24 % total: 24% 9.690MiB/s (avg: 198.920KiB/s)
progress: [/172.17.0.2]0:0/1 25 % total: 25% 11.481MiB/s (avg: 210.412KiB/s)
progress: [/172.17.0.2]0:0/1 27 % total: 27% 9.957MiB/s (avg: 221.848KiB/s)
progress: [/172.17.0.2]0:0/1 28 % total: 28% 10.270MiB/s (avg: 233.265KiB/s)
progress: [/172.17.0.2]0:0/1 29 % total: 29% 7.812MiB/s (avg: 244.571KiB/s)
progress: [/172.17.0.2]0:0/1 31 % total: 31% 14.843MiB/s (avg: 256.021KiB/s)
progress: [/172.17.0.2]0:0/1 32 % total: 32% 11.457MiB/s (avg: 267.394KiB/s)
progress: [/172.17.0.2]0:0/1 34 % total: 34% 6.550MiB/s (avg: 278.536KiB/s)
progress: [/172.17.0.2]0:0/1 35 % total: 35% 9.115MiB/s (avg: 289.782KiB/s)
progress: [/172.17.0.2]0:0/1 37 % total: 37% 11.054MiB/s (avg: 301.064KiB/s)
progress: [/172.17.0.2]0:0/1 38 % total: 38% 10.449MiB/s (avg: 312.307KiB/s)
progress: [/172.17.0.2]0:0/1 39 % total: 39% 1.646MiB/s (avg: 321.665KiB/s)
progress: [/172.17.0.2]0:0/1 41 % total: 41% 13.300MiB/s (avg: 332.872KiB/s)
progress: [/172.17.0.2]0:0/1 42 % total: 42% 14.370MiB/s (avg: 344.082KiB/s)
progress: [/172.17.0.2]0:0/1 44 % total: 44% 16.734MiB/s (avg: 355.314KiB/s)
progress: [/172.17.0.2]0:0/1 45 % total: 45% 22.245MiB/s (avg: 366.592KiB/s)
progress: [/172.17.0.2]0:0/1 47 % total: 47% 25.561MiB/s (avg: 377.882KiB/s)
progress: [/172.17.0.2]0:0/1 48 % total: 48% 24.543MiB/s (avg: 389.155KiB/s)
progress: [/172.17.0.2]0:0/1 49 % total: 49% 4.894MiB/s (avg: 399.688KiB/s)
progress: [/172.17.0.2]0:0/1 51 % total: 51% 8.331MiB/s (avg: 410.559KiB/s)
progress: [/172.17.0.2]0:0/1 52 % total: 52% 5.771MiB/s (avg: 421.150KiB/s)
progress: [/172.17.0.2]0:0/1 54 % total: 54% 8.738MiB/s (avg: 431.983KiB/s)
progress: [/172.17.0.2]0:0/1 55 % total: 55% 3.406MiB/s (avg: 441.911KiB/s)
progress: [/172.17.0.2]0:0/1 56 % total: 56% 9.791MiB/s (avg: 452.730KiB/s)
progress: [/172.17.0.2]0:0/1 58 % total: 58% 3.401MiB/s (avg: 462.545KiB/s)
progress: [/172.17.0.2]0:0/1 59 % total: 59% 5.280MiB/s (avg: 472.840KiB/s)
progress: [/172.17.0.2]0:0/1 61 % total: 61% 12.232MiB/s (avg: 483.663KiB/s)
progress: [/172.17.0.2]0:0/1 62 % total: 62% 9.258MiB/s (avg: 494.325KiB/s)
progress: [/172.17.0.2]0:0/1 64 % total: 64% 2.877MiB/s (avg: 503.640KiB/s)
progress: [/172.17.0.2]0:0/1 65 % total: 65% 7.461MiB/s (avg: 514.078KiB/s)
progress: [/172.17.0.2]0:0/1 66 % total: 66% 24.247MiB/s (avg: 525.018KiB/s)
```

```

progress: [/172.17.0.2]0:0/1 68 % total: 68% 9.348MiB/s (avg: 535.563KiB/s)
progress: [/172.17.0.2]0:0/1 69 % total: 69% 5.130MiB/s (avg: 545.563KiB/s)
progress: [/172.17.0.2]0:0/1 71 % total: 71% 19.861MiB/s (avg: 556.392KiB/s)
progress: [/172.17.0.2]0:0/1 72 % total: 72% 15.501MiB/s (avg: 567.122KiB/s)
progress: [/172.17.0.2]0:0/1 74 % total: 74% 5.031MiB/s (avg: 576.996KiB/s)
progress: [/172.17.0.2]0:0/1 75 % total: 75% 22.771MiB/s (avg: 587.813KiB/s)
progress: [/172.17.0.2]0:0/1 76 % total: 76% 22.780MiB/s (avg: 598.619KiB/s)
progress: [/172.17.0.2]0:0/1 78 % total: 78% 20.684MiB/s (avg: 609.386KiB/s)
progress: [/172.17.0.2]0:0/1 79 % total: 79% 22.920MiB/s (avg: 620.173KiB/s)
progress: [/172.17.0.2]0:0/1 81 % total: 81% 7.458MiB/s (avg: 630.333KiB/s)
progress: [/172.17.0.2]0:0/1 82 % total: 82% 22.993MiB/s (avg: 641.090KiB/s)
progress: [/172.17.0.2]0:0/1 84 % total: 84% 21.392MiB/s (avg: 651.814KiB/s)
progress: [/172.17.0.2]0:0/1 85 % total: 85% 7.732MiB/s (avg: 661.938KiB/s)
progress: [/172.17.0.2]0:0/1 86 % total: 86% 3.476MiB/s (avg: 670.892KiB/s)
progress: [/172.17.0.2]0:0/1 88 % total: 88% 19.889MiB/s (avg: 681.521KiB/s)
progress: [/172.17.0.2]0:0/1 89 % total: 89% 21.077MiB/s (avg: 692.162KiB/s)
progress: [/172.17.0.2]0:0/1 91 % total: 91% 24.062MiB/s (avg: 702.835KiB/s)
progress: [/172.17.0.2]0:0/1 92 % total: 92% 19.798MiB/s (avg: 713.431KiB/s)
progress: [/172.17.0.2]0:0/1 94 % total: 94% 17.591MiB/s (avg: 723.965KiB/s)
progress: [/172.17.0.2]0:0/1 95 % total: 95% 13.725MiB/s (avg: 734.361KiB/s)
progress: [/172.17.0.2]0:0/1 96 % total: 96% 16.737MiB/s (avg: 744.846KiB/s)
progress: [/172.17.0.2]0:0/1 98 % total: 98% 22.701MiB/s (avg: 755.443KiB/s)
progress: [/172.17.0.2]0:0/1 99 % total: 99% 18.718MiB/s (avg: 765.954KiB/s)
progress: [/172.17.0.2]0:1/1 100% total: 100% 6.613MiB/s (avg: 767.802KiB/s)
progress: [/172.17.0.2]0:1/1 100% total: 100% 0.000KiB/s (avg: 670.295KiB/s)

```

Summary statistics:

```

Connections per host    : 1
Total files transferred : 1
Total bytes transferred : 4.387MiB
Total duration          : 6706 ms
Average transfer rate   : 669.835KiB/s
Peak transfer rate      : 767.802KiB/s

```

Throttling Load

To prevent the table loader from overloading the system resources, you can throttle the process with the `--throttle` option. The default is unlimited (no throttling). Throttle units are in megabits. Note that the total duration is increased in the example below.

Example:

```

sstableloader --nodes 172.17.0.2 --throttle 1
/var/lib/cassandra/loadme/keyspace1/standard1-f8a4fa30aa2a11e8af27091830ac5256/
Established connection to initial hosts

```

```
Opening sstables and calculating sections to stream
Streaming relevant part of /var/lib/cassandra/loadme/keyspace1/standard1-
f8a4fa30aa2a11e8af27091830ac5256/ma-6-big-Data.db to [/172.17.0.2]
progress: [/172.17.0.2]0:1/1 100% total: 100% 0 MB/s(avg: 0 MB/s)
Summary statistics:
  Connections per host:      : 1
  Total files transferred:   : 1
  Total bytes transferred:   : 4595705
  Total duration (ms):      : 37634
  Average transfer rate (MB/s): : 0
  Peak transfer rate (MB/s): : 0
```

Speeding up Load

To speed up the load process, the number of connections per host can be increased.

Example:

```
sstableloader --nodes 172.17.0.2 --connections-per-host 100
/var/lib/cassandra/loadme/keyspace1/standard1-f8a4fa30aa2a11e8af27091830ac5256/
Established connection to initial hosts
Opening sstables and calculating sections to stream
Streaming relevant part of /var/lib/cassandra/loadme/keyspace1/standard1-
f8a4fa30aa2a11e8af27091830ac5256/ma-9-big-Data.db to [/172.17.0.2]
progress: [/172.17.0.2]0:1/1 100% total: 100% 0 MB/s(avg: 1 MB/s)
Summary statistics:
  Connections per host:      : 100
  Total files transferred:   : 1
  Total bytes transferred:   : 4595705
  Total duration (ms):      : 3486
  Average transfer rate (MB/s): : 1
  Peak transfer rate (MB/s): : 1
```

This small data set doesn't benefit much from the increase in connections per host, but note that the total duration has decreased in this example.

sstablemetadata

sstablemetadata

Print information about an sstable from the related Statistics.db and Summary.db files to standard output.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-7159> and <https://issues.apache.org/jira/browse/CASSANDRA-10838>

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstablemetadata <options> <sstable filename(s)>

--colors	Use ANSI color sequences
--gc_grace_seconds <arg>	The gc_grace_seconds to use when calculating
--help	Help
--scan	Full sstable scan for additional details. Only available in 3.0+ sstables. Defaults: false
--timestamp_unit <arg>	Time unit that cell timestamps are written with
--unicode	Use unicode to draw histograms and progress bars droppable tombstones

Print all the metadata

Run sstablemetadata against the *Data.db* file(s) related to a table. If necessary, find the *Data.db* file(s) using sstableutil.

Example:

```
$ sstableutil keyspace1 standard1 | grep Data
/var/lib/cassandra/data/keyspace1/standard1-f6845640a6cb11e8b6836d2c86545d91/mc-1-big-Data.db

$ sstablemetadata /var/lib/cassandra/data/keyspace1/standard1-f6845640a6cb11e8b6836d2c86545d91/mc-1-big-Data.db

SSTable: /var/lib/cassandra/data/keyspace1/standard1-f6845640a6cb11e8b6836d2c86545d91/mc-1-big
Partitioner: org.apache.cassandra.dht.Murmur3Partitioner
```

```

Bloom Filter FP chance: 0.010000
Minimum timestamp: 07/08/2023 09:11:52 (1688800312685792)
Maximum timestamp: 07/08/2023 09:12:03 (1688800323542149)
Duration: Days: 0 Hours: 0 Minutes: 0 Seconds: 10
SSTable min local deletion time: 07/08/2023 09:11:52 (1688800312)
SSTable max local deletion time: 07/08/2023 09:12:03 (1688800323)
Compressor: org.apache.cassandra.io.compress.LZ4Compressor
Compression ratio: 0.553999337227618
TTL min: 86400
TTL max: 86400
First token: -9223004712949498654 (key=39373333373831303130)
Last token: 9222554117157811897 (key=4f3438394e39374d3730)
Covered clusterings: ALL
Estimated droppable tombstones: 0.9188263888888889
SSTable Level: 0
Repaired at: 0
Originating host id: badfcb83-a283-402b-818e-78ede00aa3c4
Pending repair: --
Replay positions covered: {CommitLogPosition(segmentId=1535025390651,
position=226400)=CommitLogPosition(segmentId=1535025390651, position=6849139)}
Total Column Cells: 72918
Total Rows: 24306
Estimated tombstone drop times:
1535039100:      80390
1535039160:       5645
1535039220:     13965

```

Count	Row Size	Cell Count
1	0	0
2	0	0
3	0	0
4	0	0
5	0	20000
6	0	0
7	0	0
8	0	0
10	0	0
12	0	0
14	0	0
17	0	0
20	0	0
24	0	0
29	0	0
35	0	0
42	0	0
50	0	0
60	0	0
72	0	0
86	0	0

103	0	0
124	0	0
149	0	0
179	0	0
215	0	0
258	20000	0
310	0	0
372	0	0
446	0	0
535	0	0
642	0	0
770	0	0
924	0	0
1109	0	0
1331	0	0
1597	0	0
1916	0	0
2299	0	0
2759	0	0
3311	0	0
3973	0	0
4768	0	0
5722	0	0
6866	0	0
8239	0	0
9887	0	0
11864	0	0
14237	0	0
17084	0	0
20501	0	0
24601	0	0
29521	0	0
35425	0	0
42510	0	0
51012	0	0
61214	0	0
73457	0	0
88148	0	0
105778	0	0
126934	0	0
152321	0	0
182785	0	0
219342	0	0
263210	0	0
315852	0	0
379022	0	0
454826	0	0
545791	0	0

654949	0	0
785939	0	0
943127	0	0
1131752	0	0
1358102	0	0
1629722	0	0
1955666	0	0
2346799	0	0
2816159	0	0
3379391	0	0
4055269	0	0
4866323	0	0
5839588	0	0
7007506	0	0
8409007	0	0
10090808	0	0
12108970	0	0
14530764	0	0
17436917	0	0
20924300	0	0
25109160	0	0
30130992	0	0
36157190	0	0
43388628	0	0
52066354	0	0
62479625	0	0
74975550	0	0
89970660	0	0
107964792	0	0
129557750	0	0
155469300	0	0
186563160	0	0
223875792	0	0
268650950	0	0
322381140	0	0
386857368	0	0
464228842	0	0
557074610	0	0
668489532	0	0
802187438	0	0
962624926	0	0
1155149911	0	0
1386179893	0	0
1663415872	0	0
1996099046	0	0
2395318855	0	0
2874382626	0	
3449259151	0	

4139110981	0
4966933177	0
5960319812	0
7152383774	0
8582860529	0
10299432635	0
12359319162	0
14831182994	0
17797419593	0
21356903512	0
25628284214	0
30753941057	0
36904729268	0
44285675122	0
53142810146	0
63771372175	0
76525646610	0
91830775932	0
110196931118	0
132236317342	0
158683580810	0
190420296972	0
228504356366	0
274205227639	0
329046273167	0
394855527800	0
473826633360	0
568591960032	0
682310352038	0
818772422446	0
982526906935	0
1179032288322	0
1414838745986	0

Estimated cardinality: 20196

EncodingStats minTTL: 0

EncodingStats minLocalDeletionTime: 1442880000

EncodingStats minTimestamp: 1535025565275000

KeyType: org.apache.cassandra.db.marshall.BytesType

ClusteringTypes: [org.apache.cassandra.db.marshall.UTF8Type]

StaticColumns: {C3:org.apache.cassandra.db.marshall.BytesType,

C4:org.apache.cassandra.db.marshall.BytesType,

C0:org.apache.cassandra.db.marshall.BytesType,

C1:org.apache.cassandra.db.marshall.BytesType,

C2:org.apache.cassandra.db.marshall.BytesType}

RegularColumns: {}

IsTransient: false

Specify gc grace seconds

To see the ratio of droppable tombstones given a configured gc grace seconds, use the `gc_grace_seconds` option. Because the `sstablemetadata` tool doesn't access the schema directly, this is a way to more accurately estimate droppable tombstones — for example, if you pass in `gc_grace_seconds` matching what is configured in the schema. The `gc_grace_seconds` value provided is subtracted from the current machine time (in seconds).

ref: <https://issues.apache.org/jira/browse/CASSANDRA-12208>

Example:

```
$ sstablemetadata /var/lib/cassandra/data/keyspace1/standard1-41b52700b4ed11e896476d2c86545d91/mc-12-big-Data.db | grep "Estimated tombstone drop times" -A4
Estimated tombstone drop times:
1536599100:      1
1536599640:      1
1536599700:      2

$ echo $(date +%s)
1536602005

# if gc_grace_seconds was configured at 100, all of the tombstones would be currently droppable
$ sstablemetadata --gc_grace_seconds 100 /var/lib/cassandra/data/keyspace1/standard1-41b52700b4ed11e896476d2c86545d91/mc-12-big-Data.db | grep "Estimated droppable tombstones"
Estimated droppable tombstones: 4.0E-5

# if gc_grace_seconds was configured at 4700, some of the tombstones would be currently droppable
$ sstablemetadata --gc_grace_seconds 4700 /var/lib/cassandra/data/keyspace1/standard1-41b52700b4ed11e896476d2c86545d91/mc-12-big-Data.db | grep "Estimated droppable tombstones"
Estimated droppable tombstones: 9.6111111111111E-6

# if gc_grace_seconds was configured at 100, none of the tombstones would be currently droppable
$ sstablemetadata --gc_grace_seconds 5000 /var/lib/cassandra/data/keyspace1/standard1-41b52700b4ed11e896476d2c86545d91/mc-12-big-Data.db | grep "Estimated droppable tombstones"
Estimated droppable tombstones: 0.0
```

Scanning of a table

It is possible to specify `--scan` option to get additional statistics about a table which might be time-consuming.

Example

```
sstablemetadata -s /var/lib/cassandra/data/data/ks/tb-  
d8ead8501c9511ee805bb3deafc4499b/oa-1-big-Data.db
```

This gives you statistics such as:

```
Size: 120  
Partitions: 2  
Rows: 5  
Tombstones: 0  
Cells: 5  
Widest Partitions:  
  [2] 3  
  [1] 2  
Largest Partitions:  
  [2] 147 (147 B)  
  [1] 98 (98 B)
```

Explanation of each value printed above

Value	Explanation
SSTable	prefix of the sstable filenames related to this sstable
Partitioner	partitioner type used to distribute data across nodes; defined in cassandra.yaml
Bloom Filter FP	precision of Bloom filter used in reads; defined in the table definition
Minimum timestamp	minimum timestamp of any entry in this sstable, in epoch microseconds
Maximum timestamp	maximum timestamp of any entry in this sstable, in epoch microseconds
Duration	Difference between Maximum timestamp and Minimum timestamp

Value	Explanation
SSTable min local deletion time	minimum timestamp of deletion date, based on TTL, in epoch seconds
SSTable max local deletion time	maximum timestamp of deletion date, based on TTL, in epoch seconds
Compressor	blank (-) by default; if not blank, indicates type of compression enabled on the table
TTL min	time-to-live in seconds; default 0 unless defined in the table definition
TTL max	time-to-live in seconds; default 0 unless defined in the table definition
First token	lowest token and related key found in the sstable summary
Last token	highest token and related key found in the sstable summary
Estimated droppable tombstones	ratio of tombstones to columns, using configured gc grace seconds if relevant
SSTable level	compaction level of this sstable, if leveled compaction (LCS) is used
Repaired at	the timestamp this sstable was marked as repaired via sstablerepairedset, in epoch milliseconds
Replay positions covered	the interval of time and commitlog positions related to this sstable
totalColumnsSet	number of cells in the table
totalRows	number of rows in the table
Estimated tombstone drop times	approximate number of rows that will expire, ordered by epoch seconds
Count Row Size Cell Count	two histograms in two columns; one represents distribution of Row Size and the other represents distribution of Cell Count
Estimated cardinality an estimate of unique values, used for compaction	EncodingStats* minTTL
in epoch milliseconds	EncodingStats* minLocalDeletionTime
in epoch seconds	EncodingStats* minTimestamp
in epoch microseconds	KeyType

Value	Explanation
the type of partition key, useful in reading and writing data from/to storage; defined in the table definition	ClusteringTypes
the type of clustering key, useful in reading and writing data from/to storage; defined in the table definition	StaticColumns
a list of the shared columns in the table	RegularColumns

* For the encoding stats values, the delta of this and the current epoch time is used when encoding and storing data in the most optimal way.

sstableofflinerelevel

sstableofflinerelevel

When using LeveledCompactionStrategy, sstables can get stuck at L0 on a recently bootstrapped node, and compactions may never catch up. This tool is used to bump sstables into the highest level possible.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-8301>

The way this is done is: sstables are sorted by their last token. Given an original leveling like this (note that [] indicates token boundaries, not sstable size on disk; all sstables are the same size):

```
L3 [][][][][][][][][][][]
L2 [   ][   ][   ][   ]
L1 [       ][       ]
L0 [           ]
```

Will look like this after being dropped to L0 and sorted by last token (and, to illustrate overlap, the overlapping ones are put on a new line):

```
[][][]
[   ][][][]
  [   ]
[       ]
...
```

Then, we start iterating from the smallest last-token and adding all sstables that do not cause an overlap to a level. We will reconstruct the original leveling top-down. Whenever we add an sstable to the level, we remove it from the sorted list. Once we reach the end of the sorted list, we have a full level, and can start over with the level below.

If we end up with more levels than expected, we put all levels exceeding the expected in L0, for example, original L0 files will most likely be put in a level of its own since they most often overlap many other sstables.

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstableofflinerelevel [--dry-run] <keyspace> <table>

Doing a dry run

Use the `--dry-run` option to see the current level distribution and predicted level after the change.

Example:

```
sstableofflinerelevel --dry-run keyspace eventlog
For sstables in /var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753:
Current leveling:
L0=2
Potential leveling:
L0=1
L1=1
```

Running a relevel

Example:

```
sstableofflinerelevel keyspace eventlog
For sstables in /var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753:
Current leveling:
L0=2
New leveling:
L0=1
L1=1
```

Keyspace or table not found

If an invalid keyspace and/or table is provided, an exception will be thrown.

Example:

```
sstableofflinerelevel --dry-run keyspace evenlog

Exception in thread "main" java.lang.IllegalArgumentException: Unknown
keyspace/columnFamily keyspace1.eventlog
    at
org.apache.cassandra.tools.SSTableOfflineRelevel.main(SSTableOfflineRelevel.java:96)
```

sstablepartitions

sstablepartitions

Identifies large partitions of SSTables and outputs the partition size in bytes, row count, cell count, and tombstone count.

You can supply any number of sstables file paths, or directories containing sstables. Each sstable will be analyzed separately.

If a metrics threshold such as `--min-size`, `--min-rows`, `--min-cells` or `--min-tombstones` is provided, then the partition keys exceeding of the threshold will be printed in the output. It also prints a summary of metrics for the table. The percentiles in the metrics are estimates, while the min/max/count metrics are accurate.

The default output of this tool is meant to be read by human eyes. Future versions might include small formatting changes or present new data that can fool scripts reading it. Scripts or other automatic tools should use the `--csv` flag to produce machine-readable output. Future versions will not change the format of the CSV output except for maybe adding new columns, so a proper CSV parser consuming the output should keep working.

Cassandra doesn't need to be running before this tool is executed.

Usage

sstablepartitions <options> <sstable files or directories>

<code>-t, --min-size <arg></code>	Partition size threshold, expressed as either the number of bytes or a size with unit of the form 10KiB, 20MiB, 30GiB, etc.
<code>-w, --min-rows <arg></code>	Partition row count threshold.
<code>-c, --min-cells <arg></code>	Partition cell count threshold
<code>-o, --min-tombstones <arg></code>	Partition tombstone count threshold.
<code>-k, --key <arg></code>	Partition keys to include, instead of scanning all partitions.
<code>-x, --exclude-key <arg></code>	Partition keys to exclude.
<code>-r, --recursive</code>	Scan for sstables recursively
<code>-b, --backups</code>	Include backups present in data directories when scanning directories
<code>-s, --snaphsots</code>	Include snapshots present in data directories when scanning directories

-u, --current-timestamp <arg>	Timestamp (seconds since epoch, unit time) for TTL expired calculation.
-y, --partitions-only	Only brief partition information. Exclude per-partition detailed row/cell/tombstone information from process and output.
-m, --csv	Produced CSV output (machine readable)

Examples

Analyze partition statistics for a single SSTable

Use the path to the SSTable file as the only argument.

Example:

```
sstablepartitions data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-Data.db

Processing k.t-d7be5e90e90111ed8b54efe3c39cb0bb #8 (big-oa) (1.368 GiB uncompressed,
534.979 MiB on disk)
count      Partition size      Row count      Cell count      Tombstone
~p50      767.519 KiB      770      1916
0
~p75      2.238 MiB      2299      5722
0
~p90      3.867 MiB      3311      9887
50
~p95      16.629 MiB      14237      42510
446
~p99      148.267 MiB      126934      379022
1331
~p999      368.936 MiB      315852      943127
2759
min      56.854 KiB      100      150
0
max      356.067 MiB      310706      932118
2450
count      210
```


Analyze partition statistics for all SSTables in a directory

Use the path to the SSTables directory as the only argument.

Example:

```
sstablepartitions data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb

Processing k.t-d7be5e90e90111ed8b54efe3c39cb0bb #8 (big-oa) (1.368 GiB uncompressed,
534.979 MiB on disk)
count      Partition size      Row count      Cell count      Tombstone
~p50      767.519 KiB      770      1916
0
~p75      2.238 MiB      2299      5722
0
~p90      3.867 MiB      3311      9887
50
~p95      16.629 MiB      14237      42510
446
~p99      148.267 MiB      126934      379022
1331
~p999      368.936 MiB      315852      943127
2759
min      56.854 KiB      100      150
0
max      356.067 MiB      310706      932118
2450
count      210

Processing k.t-d7be5e90e90111ed8b54efe3c39cb0bb #9 (big-oa) (457.540 MiB uncompressed,
174.880 MiB on disk)
count      Partition size      Row count      Cell count      Tombstone
~p50      1.865 MiB      1597      4768
0
~p75      13.858 MiB      14237      42510
0
~p90      28.735 MiB      29521      73457
50
~p95      34.482 MiB      29521      88148
8239
~p99      49.654 MiB      42510      126934
14237
~p999      49.654 MiB      42510      126934
```

14237			
min	47.272 KiB	100	150
0			
max	45.133 MiB	39429	118287
13030			
count	57		

Output only partitions over 100MiB in size

Use the `--min-size` option to specify the minimum size a partition must have to be included in the output.

Example:

```
sstablepartitions data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-Data.db --min-size 100MiB
```

Processing k.t-d7be5e90e90111ed8b54efe3c39cb0bb #8 (big-oa) (1.368 GiB uncompressed, 534.979 MiB on disk)

Partition: '13' (0000000d) live, size: 105.056 MiB, rows: 91490, cells: 274470, tombstones: 50 (row:50, range:0, complex:0, cell:0, row-TTLd:0, cell-TTLd:0)

Partition: '1' (00000001) live, size: 127.241 MiB, rows: 111065, cells: 333195, tombstones: 50 (row:50, range:0, complex:0, cell:0, row-TTLd:0, cell-TTLd:0)

Partition: '8' (00000008) live, size: 356.067 MiB, rows: 310706, cells: 932118, tombstones: 0 (row:0, range:0, complex:0, cell:0, row-TTLd:0, cell-TTLd:0)

Partition: '2' (00000002) live, size: 213.341 MiB, rows: 186582, cells: 559125, tombstones: 978 (row:978, range:0, complex:0, cell:0, row-TTLd:0, cell-TTLd:0)

Summary of k.t-d7be5e90e90111ed8b54efe3c39cb0bb #8 (big-oa):

File: /Users/adela pena/src/cassandra/trunk/data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-Data.db

4 partitions match

Keys: 13 1 8 2

	Partition size	Row count	Cell count	Tombstone
count				
~p50	767.519 KiB	770	1916	
0				
~p75	2.238 MiB	2299	5722	
0				
~p90	3.867 MiB	3311	9887	
50				
~p95	16.629 MiB	14237	42510	
446				
~p99	148.267 MiB	126934	379022	
1331				
~p999	368.936 MiB	315852	943127	
2759				

min	56.854 KiB	100	150
0			
max	356.067 MiB	310706	932118
2450			
count	210		

Output only partitions with more than 1000 tombstones

Use the `--min-tombstones` option to specify the minimum number of tombstones a partition must have to be included in the output.

Example:

```
sstablepartitions data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-Data.db --min-tombstones 1000

Processing k.t-d7be5e90e90111ed8b54efe3c39cb0bb #8 (big-oa) (1.368 GiB uncompressed, 534.979 MiB on disk)
  Partition: '55' (00000037) live, size: 1.290 MiB, rows: 2317, cells: 3474, tombstones: 1159 (row:1159, range:0, complex:0, cell:0, row-TTLd:0, cell-TTLd:0)
  Partition: '28' (0000001c) live, size: 1.198 MiB, rows: 2099, cells: 3147, tombstones: 1050 (row:1050, range:0, complex:0, cell:0, row-TTLd:0, cell-TTLd:0)
  Partition: '89' (00000059) live, size: 1.346 MiB, rows: 2226, cells: 3339, tombstones: 1113 (row:1113, range:0, complex:0, cell:0, row-TTLd:0, cell-TTLd:0)
  Partition: '21' (00000015) live, size: 3.853 MiB, rows: 4900, cells: 9927, tombstones: 2450 (row:2450, range:0, complex:0, cell:0, row-TTLd:0, cell-TTLd:0)
Summary of k.t-d7be5e90e90111ed8b54efe3c39cb0bb #8 (big-oa):
  File: /Users/adelpena/src/cassandra/trunk/data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-Data.db
  4 partitions match
  Keys: 55 28 89 21
```

	Partition size	Row count	Cell count	Tombstone
count				
~p50	767.519 KiB	770	1916	
0				
~p75	2.238 MiB	2299	5722	
0				
~p90	3.867 MiB	3311	9887	
50				
~p95	16.629 MiB	14237	42510	
446				
~p99	148.267 MiB	126934	379022	
1331				
~p999	368.936 MiB	315852	943127	
2759				
min	56.854 KiB	100	150	

0			
max	356.067 MiB	310706	932118
2450			
count	210		

Output CSV machine-readable output

Use the `--csv` option to output a CSV machine-readable output, combined with any threshold value.

Example:

```
sstablepartitions data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-Data.db --min
-size 100MiB --csv
key,keyBinary,live,offset,size,rowCount,cellCount,tombstoneCount,rowTombstoneCount,rangeT
ombstoneCount,complexTombstoneCount,cellTombstoneCount,rowTtlExpired,cellTtlExpired,direc
tory,keyspace,table,index,snapshot,backup,generation,format,version
"13",0000000d,true,186403543,110158965,91490,274470,50,50,0,0,0,0,0,/Users/adelapena/src/
cassandra/trunk/data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-
Data.db,k,t,,,8,big,oa
"1",00000001,true,325141542,133422183,111065,333195,50,50,0,0,0,0,0,/Users/adelapena/src/
cassandra/trunk/data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-
Data.db,k,t,,,8,big,oa
"8",00000008,true,477133752,373362819,310706,932118,0,0,0,0,0,0,0,/Users/adelapena/src/ca
ssandra/trunk/data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-
Data.db,k,t,,,8,big,oa
"2",00000002,true,851841363,223704192,186582,559125,978,978,0,0,0,0,0,/Users/adelapena/sr
c/cassandra/trunk/data/data/k/t-d7be5e90e90111ed8b54efe3c39cb0bb/oa-8-big-
Data.db,k,t,,,8,big,oa
```

sstablerepairedset

sstablerepairedset

Repairs can take a very long time in some environments, for large sizes of data. Use this tool to set the repairedAt status on a given set of sstables, so that repairs can be run on only un-repaired sstables if desired.

Note that running a repair (e.g., via `nodetool repair`) doesn't set the status of this metadata. Only setting the status of this metadata via this tool does.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-5351>

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

`sstablerepairedset --really-set <options> [-f <sstable-list> | <sstables>]`

<code>--really-set</code>	required if you want to really set the status
<code>--is-repaired</code>	set the repairedAt status to the last modified time
<code>--is-unrepaired</code>	set the repairedAt status to 0
<code>-f</code>	use a file containing a list of sstables as the input

Set a lot of sstables to unrepaired status

There are many ways to do this programmatically. This way would likely include variables for the keyspace and table.

Example:

```
find /var/lib/cassandra/data/keyspace1/standard1-d936bd20a17c11e8bc92a55ed562cd82/* -name
"*Data.db" -print0 | xargs -0 -I % sstablerepairedset --really-set --is-unrepaired %
```

Set one to many sstables to repaired status

Set the repairedAt status after a repair to mark the sstables as repaired. Again, using variables for the keyspace and table names is a good choice.

Example:

```
nodetool repair keyspace1 standard1
find /var/lib/cassandra/data/keyspace1/standard1-d936bd20a17c11e8bc92a55ed562cd82/* -name
"*Data.db" -print0 | xargs -0 -I % sstablerepairedset --really-set --is-repaired %
```

Print metadata showing repaired status

sstablemetadata can be used to view the status set or unset using this command.

Example:

```
sstablerepairedset --really-set --is-repaired
/var/lib/cassandra/data/keyspace1/standard1-
d936bd20a17c11e8bc92a55ed562cd82/mc-1-big-Data.db sstablemetadata
/var/lib/cassandra/data/keyspace1/standard1-
d936bd20a17c11e8bc92a55ed562cd82/mc-1-big-Data.db | grep "Repaired at"
Repaired at: 1534443974000
```

```
sstablerepairedset --really-set --is-unrepaired
/var/lib/cassandra/data/keyspace1/standard1-
d936bd20a17c11e8bc92a55ed562cd82/mc-1-big-Data.db sstablemetadata
/var/lib/cassandra/data/keyspace1/standard1-
d936bd20a17c11e8bc92a55ed562cd82/mc-1-big-Data.db | grep "Repaired at"
Repaired at: 0
```

Using command in a script

If you know you ran repair 2 weeks ago, you can do something like the following:

```
sstablerepairset --is-repaired -f <(find /var/lib/cassandra/data/.../ -iname "*Data.db*"
-mtime +14)
```

sstablescrub

sstablescrib

Fix a broken sstable. The scrub process rewrites the sstable, skipping any corrupted rows. Because these rows are lost, follow this process with a repair.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-4321>

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstablescrib <options> <keyspace> <table>

--debug	display stack traces
-h,--help	display this help message
-m,--manifest-check	only check and repair the leveled manifest, without actually scrubbing the sstables
-n,--no-validate	do not validate columns using column validator
-r,--reinsert-overflowed-ttl	Rewrites rows with overflowed expiration date affected by CASSANDRA-14092 with the maximum supported expiration date of 2038-01-19T03:14:06+00:00. The rows are rewritten with the original timestamp incremented by one millisecond to override/supersede any potential tombstone that may have been generated during compaction of the affected rows.
-s,--skip-corrupted	skip corrupt rows in counter tables
-v,--verbose	verbose output

Basic Scrub

The scrub without options will do a snapshot first, then write all non-corrupted files to a new sstable.

Example:

```
sstablescrib keyspace1 standard1
Pre-scrub sstables snapshotted into snapshot pre-scrub-1534424070883
Scrubbing BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-6365332094dd11e88f324f9c503e4753/mc-5-big-Data.db') (17.142MiB)
```

```
Scrub of BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-6365332094dd11e88f324f9c503e4753/mc-5-big-Data.db') complete: 73367 rows in new sstable and 0 empty (tombstoned) rows dropped
Checking leveled manifest
```

Scrub without Validation

ref: <https://issues.apache.org/jira/browse/CASSANDRA-9406>

Use the `--no-validate` option to retain data that may be misrepresented (e.g., an integer stored in a long field) but not corrupt. This data usually doesn't present any errors to the client.

Example:

```
sstablescrib --no-validate keyspace1 standard1
Pre-scrib sstables snapshot into snapshot pre-scrib-1536243158517
Scrubbing BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-bc9cf530b1da11e886c66d2c86545d91/mc-2-big-Data.db') (4.482MiB)
Scrub of BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-bc9cf530b1da11e886c66d2c86545d91/mc-2-big-Data.db') complete; looks like all 0 rows were tombstoned
```

Skip Corrupted Counter Tables

ref: <https://issues.apache.org/jira/browse/CASSANDRA-5930>

If counter tables are corrupted in a way that prevents `sstablescrib` from completing, you can use the `--skip-corrupted` option to skip scrubbing those counter tables. This workaround is not necessary in versions 2.0+.

Example:

```
sstablescrib --skip-corrupted keyspace1 counter1
```

Dealing with Overflow Dates

ref: <https://issues.apache.org/jira/browse/CASSANDRA-14092>

Using the option `--reinsert-overflowed-ttl` allows a rewriting of rows that had a max TTL going over the maximum (causing an overflow).

Example:

```
sstablescrub --reinsert-overflowed-ttl keyspace1 counter1
```

Manifest Check

As of Cassandra version 2.0, this option is no longer relevant, since level data was moved from a separate manifest into the sstable metadata.

sstablesplit

sstablesplit

Big sstable files can take up a lot of disk space. The sstablesplit tool can be used to split those large files into smaller files. It can be thought of as a type of anticompaaction.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-4766>

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstablesplit <options> <filename>

--debug	display stack traces
-h, --help	display this help message
--no-snapshot	don't snapshot the sstables before splitting
-s, --size <size>	maximum size in MB for the output sstables (default: 50)

This command should be run with Cassandra stopped. Note: the script does not verify that Cassandra is stopped.

Split a File

Split a large sstable into smaller sstables. By default, unless the option --no-snapshot is added, a snapshot will be done of the original sstable and placed in the snapshots folder.

Example:

```
sstablesplit /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-8-big-Data.db
```

```
Pre-split sstables snapshotted into snapshot pre-split-1533144514795
```

Split Multiple Files

Wildcards can be used in the filename portion of the command to split multiple files.

Example:

```
sstablesplit --size 1 /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-1*
```

Attempt to Split a Small File

If the file is already smaller than the split size provided, the sstable will not be split.

Example:

```
sstablesplit /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-8-big-Data.db
Skipping /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-8-big-Data.db: it's size (1.442 MB) is less than the split size (50 MB)
No sstables needed splitting.
```

Split a File into Specified Size

The default size used for splitting is 50MB. Specify another size with the `--size` option. The size is in megabytes (MB). Specify only the number, not the units. For example `--size 50` is correct, but `--size 50MB` is not.

Example:

```
sstablesplit --size 1 /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-9-big-Data.db
Pre-split sstables snapshotted into snapshot pre-split-1533144996008
```

Split Without Snapshot

By default, `sstablesplit` will create a snapshot before splitting. If a snapshot is not needed, use the `--no-snapshot` option to skip it.

Example:

```
sstablesplit --size 1 --no-snapshot /var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-11-big-Data.db
```

Note: There is no output, but you can see the results in your file system.

sstableupgrade

sstableupgrade

Upgrade the sstables in the given table (or snapshot) to the current version of Cassandra. This process is typically done after a Cassandra version upgrade. This operation will rewrite the sstables in the specified table to match the currently installed version of Cassandra. The sstableupgrade command can also be used to downgrade sstables to a previous version.

The snapshot option will only upgrade the specified snapshot. Upgrading snapshots is required before attempting to restore a snapshot taken in a major version older than the major version Cassandra is currently running. This will replace the files in the given snapshot as well as break any hard links to live sstables.

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstableupgrade <options> <keyspace> <table> [snapshot_name]

--debug	display stack traces
-h,--help	display this help message
-k,--keep-source	do not delete the source sstables

Rewrite tables to the current Cassandra version

Start with a set of sstables in one version of Cassandra:

```
ls -al /tmp/cassandra/data/keyspace1/standard1-9695b790a63211e8a6fb091830ac5256/
...
-rw-r--r-- 1 user wheel 348 Aug 22 13:45 keyspace1-standard1-ka-1-CRC.db
-rw-r--r-- 1 user wheel 562000 Aug 22 13:45 keyspace1-standard1-ka-1-Data.db
-rw-r--r-- 1 user wheel 10 Aug 22 13:45 keyspace1-standard1-ka-1-Digest.sha1
-rw-r--r-- 1 user wheel 25016 Aug 22 13:45 keyspace1-standard1-ka-1-Filter.db
-rw-r--r-- 1 user wheel 480000 Aug 22 13:45 keyspace1-standard1-ka-1-Index.db
-rw-r--r-- 1 user wheel 9895 Aug 22 13:45 keyspace1-standard1-ka-1-Statistics.db
-rw-r--r-- 1 user wheel 3562 Aug 22 13:45 keyspace1-standard1-ka-1-Summary.db
-rw-r--r-- 1 user wheel 79 Aug 22 13:45 keyspace1-standard1-ka-1-TOC.txt
```


After upgrading the Cassandra version, upgrade the sstables:

```
sstableupgrade keyspace1 standard1
Found 1 sstables that need upgrading.
Upgrading BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-
9695b790a63211e8a6fb091830ac5256/keyspace1-standard1-ka-1-Data.db')
Upgrade of BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-
9695b790a63211e8a6fb091830ac5256/keyspace1-standard1-ka-1-Data.db') complete.

ls -al /tmp/cassandra/data/keyspace1/standard1-9695b790a63211e8a6fb091830ac5256/
...
drwxr-xr-x  2 user  wheel      64 Aug 22 13:48 backups
-rw-r--r--  1 user  wheel     292 Aug 22 13:48 mc-2-big-CRC.db
-rw-r--r--  1 user  wheel  4599475 Aug 22 13:48 mc-2-big-Data.db
-rw-r--r--  1 user  wheel      10 Aug 22 13:48 mc-2-big-Digest.crc32
-rw-r--r--  1 user  wheel   25256 Aug 22 13:48 mc-2-big-Filter.db
-rw-r--r--  1 user  wheel  330807 Aug 22 13:48 mc-2-big-Index.db
-rw-r--r--  1 user  wheel   10312 Aug 22 13:48 mc-2-big-Statistics.db
-rw-r--r--  1 user  wheel    3506 Aug 22 13:48 mc-2-big-Summary.db
-rw-r--r--  1 user  wheel      80 Aug 22 13:48 mc-2-big-TOC.txt
```

Rewrite tables to the current Cassandra version, and keep tables in old version

Again, starting with a set of sstables in one version:

```
ls -al /tmp/cassandra/data/keyspace1/standard1-db532690a63411e8b4ae091830ac5256/
...
-rw-r--r--  1 user  wheel     348 Aug 22 13:58 keyspace1-standard1-ka-1-CRC.db
-rw-r--r--  1 user  wheel  5620000 Aug 22 13:58 keyspace1-standard1-ka-1-Data.db
-rw-r--r--  1 user  wheel      10 Aug 22 13:58 keyspace1-standard1-ka-1-Digest.sha1
-rw-r--r--  1 user  wheel   25016 Aug 22 13:58 keyspace1-standard1-ka-1-Filter.db
-rw-r--r--  1 user  wheel  4800000 Aug 22 13:58 keyspace1-standard1-ka-1-Index.db
-rw-r--r--  1 user  wheel    9895 Aug 22 13:58 keyspace1-standard1-ka-1-Statistics.db
-rw-r--r--  1 user  wheel    3562 Aug 22 13:58 keyspace1-standard1-ka-1-Summary.db
-rw-r--r--  1 user  wheel      79 Aug 22 13:58 keyspace1-standard1-ka-1-TOC.txt
```

After upgrading the Cassandra version, upgrade the sstables, retaining the original sstables:

```
sstableupgrade keyspace1 standard1 -k
Found 1 sstables that need upgrading.
Upgrading BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-
```

```
db532690a63411e8b4ae091830ac5256/keyspace1-standard1-ka-1-Data.db')
Upgrade of BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-
db532690a63411e8b4ae091830ac5256/keyspace1-standard1-ka-1-Data.db') complete.

ls -al /tmp/cassandra/data/keyspace1/standard1-db532690a63411e8b4ae091830ac5256/
...
drwxr-xr-x  2 user  wheel      64 Aug 22 14:00 backups
-rw-r--r--@ 1 user  wheel     348 Aug 22 13:58 keyspace1-standard1-ka-1-CRC.db
-rw-r--r--@ 1 user  wheel  562000 Aug 22 13:58 keyspace1-standard1-ka-1-Data.db
-rw-r--r--@ 1 user  wheel     10 Aug 22 13:58 keyspace1-standard1-ka-1-Digest.sha1
-rw-r--r--@ 1 user  wheel   25016 Aug 22 13:58 keyspace1-standard1-ka-1-Filter.db
-rw-r--r--@ 1 user  wheel  480000 Aug 22 13:58 keyspace1-standard1-ka-1-Index.db
-rw-r--r--@ 1 user  wheel   9895 Aug 22 13:58 keyspace1-standard1-ka-1-Statistics.db
-rw-r--r--@ 1 user  wheel   3562 Aug 22 13:58 keyspace1-standard1-ka-1-Summary.db
-rw-r--r--@ 1 user  wheel    79 Aug 22 13:58 keyspace1-standard1-ka-1-TOC.txt
-rw-r--r--  1 user  wheel    292 Aug 22 14:01 mc-2-big-CRC.db
-rw-r--r--  1 user  wheel  4596370 Aug 22 14:01 mc-2-big-Data.db
-rw-r--r--  1 user  wheel    10 Aug 22 14:01 mc-2-big-Digest.crc32
-rw-r--r--  1 user  wheel   25256 Aug 22 14:01 mc-2-big-Filter.db
-rw-r--r--  1 user  wheel  330801 Aug 22 14:01 mc-2-big-Index.db
-rw-r--r--  1 user  wheel   10312 Aug 22 14:01 mc-2-big-Statistics.db
-rw-r--r--  1 user  wheel   3506 Aug 22 14:01 mc-2-big-Summary.db
-rw-r--r--  1 user  wheel    80 Aug 22 14:01 mc-2-big-TOC.txt
```

Rewrite a snapshot to the current Cassandra version

Find the snapshot name:

```
nodetool listsnapshots

Snapshot Details:
Snapshot name      Keyspace name      Column family name      True size
Size on disk
...
1534962986979      keyspace1          standard1                5.85 MB
5.85 MB
```

Then rewrite the snapshot:

```
sstableupgrade keyspace1 standard1 1534962986979
Found 1 sstables that need upgrading.
Upgrading BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-
```

```
5850e9f0a63711e8a5c5091830ac5256/snapshots/1534962986979/keyspace1-standard1-ka-1-Data.db')  
Upgrade of BigTableReader(path='/var/lib/cassandra/data/keyspace1/standard1-5850e9f0a63711e8a5c5091830ac5256/snapshots/1534962986979/keyspace1-standard1-ka-1-Data.db') complete.
```

sstableutil

sstableutil

List sstable files for the provided table.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-7066>

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

Usage

sstableutil <options> <keyspace> <table>

-c, --cleanup	clean up any outstanding transactions
-d, --debug	display stack traces
-h, --help	display this help message
-o, --oplog	include operation logs
-t, --type <arg>	all (list all files, final or temporary), tmp (list temporary files only), final (list final files only),
-v, --verbose	verbose output

List all sstables

The basic command lists the sstables associated with a given keyspace/table.

Example:

```
sstableutil keyspace eventlog
Listing files...
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-32-big-CRC.db
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-32-big-Data.db
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-32-big-Digest.crc32
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-32-big-Filter.db
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-32-big-Index.db
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-32-big-Statistics.db
```

```
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-32-big-  
Summary.db  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-32-big-  
TOC.txt  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-37-big-  
CRC.db  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-37-big-  
Data.db  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-37-big-  
Digest.crc32  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-37-big-  
Filter.db  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-37-big-  
Index.db  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-37-big-  
Statistics.db  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-37-big-  
Summary.db  
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-37-big-  
TOC.txt
```

List only temporary sstables

Using the `-t` option followed by `tmp` will list all temporary sstables, in the format above. Temporary sstables were used in pre-3.0 versions of Cassandra.

List only final sstables

Using the `-t` option followed by `final` will list all final sstables, in the format above. In recent versions of Cassandra, this is the same output as not using the `-t` option.

Include transaction logs

Using the `-o` option will include transaction logs in the listing, in the format above.

Clean up sstables

Using the `-c` option removes any transactions left over from incomplete writes or compactions.

From the 3.0 upgrade notes:

New transaction log files have been introduced to replace the `compactions_in_progress` system table, temporary file markers (`tmp` and `tmplink`) and sstable ancestors. Therefore, compaction metadata no

longer contains ancestors. Transaction log files list sstable descriptors involved in compactions and other operations such as flushing and streaming. Use the `sstableutil` tool to list any sstable files currently involved in operations not yet completed, which previously would have been marked as temporary. A transaction log file contains one sstable per line, with the prefix "add:" or "remove:". They also contain a special line "commit", only inserted at the end when the transaction is committed. On startup we use these files to cleanup any partial transactions that were in progress when the process exited. If the commit line is found, we keep new sstables (those with the "add" prefix) and delete the old sstables (those with the "remove" prefix), vice-versa if the commit line is missing. Should you lose or delete these log files, both old and new sstable files will be kept as live files, which will result in duplicated sstables. These files are protected by incremental checksums so you should not manually edit them. When restoring a full backup or moving sstable files, you should clean-up any left over transactions and their temporary files first.

sstableverify

sstableverify

Check sstable(s) for errors or corruption, for the provided table.

ref: <https://issues.apache.org/jira/browse/CASSANDRA-5791>

Cassandra must be stopped before this tool is executed, or unexpected results will occur. Note: the script does not verify that Cassandra is stopped.

WARNING

See CASSANDRA-9947 and CASSANDRA-17017 for discussion around risks with this tool. Specifically: "We mark sstables that fail verification as unrepaired, but that's not going to do what you think. What it means is that the local node will use that sstable in the next repair, but other nodes will not. So all we'll end up doing is streaming whatever data we can read from it, to the other replicas. If we could magically mark whatever sstables correspond on the remote nodes, to the data in the local sstable, that would work, but we can't."

This tool requires the use of a `-f` or `--force` flag to indicate that the user understands the risks and would like to attempt its usage anyway.

Usage

sstableverify <options> <keyspace> <table>

<code>--debug</code>	display stack traces
<code>-e, --extended</code>	extended verification
<code>-h, --help</code>	display this help message
<code>-v, --verbose</code>	verbose output
<code>-f, --force</code>	allow use of tool (see CASSANDRA-17017 for risks)

Basic Verification

This is the basic verification. It is not a very quick process, and uses memory. You might need to increase your memory settings if you have many sstables.

Example:

```
sstableverify keyspace eventlog
Verifying BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
```

```
6365332094dd11e88f324f9c503e4753/mc-32-big-Data.db') (7.353MiB)
Deserializing sstable metadata for
BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-32-big-Data.db')
Checking computed hash of BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-32-big-Data.db')
Verifying BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-37-big-Data.db') (3.775MiB)
Deserializing sstable metadata for
BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-37-big-Data.db')
Checking computed hash of BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-37-big-Data.db')
```

Extended Verification

During an extended verification, the individual values will be validated for errors or corruption. This of course takes more time.

Example:

```
root@DC1C1:/# sstableverify -e keyspace eventlog
WARN 14:08:06,255 Only 33.096GiB free across all data volumes. Consider adding more
capacity to your cluster or removing obsolete snapshots
Verifying BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-32-big-Data.db') (7.353MiB)
Deserializing sstable metadata for
BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-32-big-Data.db')
Checking computed hash of BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-32-big-Data.db')
Extended Verify requested, proceeding to inspect values
Verify of BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-32-big-Data.db') succeeded. All 33211 rows read
successfully
Verifying BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-37-big-Data.db') (3.775MiB)
Deserializing sstable metadata for
BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-37-big-Data.db')
Checking computed hash of BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-37-big-Data.db')
Extended Verify requested, proceeding to inspect values
Verify of BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-37-big-Data.db') succeeded. All 17068 rows read
```

successfully

Corrupted File

Corrupted files are listed if they are detected by the script.

Example:

```
sstableverify keyspace eventlog
Verifying BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-40-big-Data.db') (7.416MiB)
Deserializing sstable metadata for
BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-40-big-Data.db')
Checking computed hash of BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-40-big-Data.db')
Error verifying BigTableReader(path='/var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-40-big-Data.db'): Corrupted:
/var/lib/cassandra/data/keyspace/eventlog-6365332094dd11e88f324f9c503e4753/mc-40-big-
Data.db
```

A similar (but less verbose) tool will show the suggested actions:

```
nodetool verify keyspace eventlog
error: Invalid SSTable /var/lib/cassandra/data/keyspace/eventlog-
6365332094dd11e88f324f9c503e4753/mc-40-big-Data.db, please force repair
```