

بسمه تعالی

پروژه‌ی مطالعاتی

جرم‌شناسی پایگاه داده

رویدادننگاری، ممیزی و جرم‌شناسی در

پایگاه داده‌ی غیررابطه‌ای کاساندر

فهرست مطالب

۱	مقدمه	۳
۲	نحوه‌ی ثبت وقایع در پایگاه‌داده‌ی کاساندر	۳
۲-۱	انواع فایل‌های رویدادنگاری	۳
۲-۱-۱	فایل‌های رویدادنگاری تنظیم‌شده در logback.xml	۴
۲-۱-۲	Commit log	۷
۳	ممیزی در پایگاه‌داده‌ی کاساندر	۸
۳-۱	ساختار رویدادهای ثبت شده در DataStax Enterprise	۱۱
۳-۲	ذخیره رویدادها در جدولی در کاساندر	۱۲
۴	جمع‌بندی	۱۵
۵	منابع	۱۵

۱ مقدمه

رویدادنگاری در پایگاه‌داده موجب بهبود و بازیابی^۱ از شرایط شکست^۲ و همگام‌سازی پایگاه‌های داده‌ی اولیه^۳ و ثانویه^۴ و در نتیجه، دسترس‌پذیری بالاتری^۵ پایگاه‌داده می‌شود. تمامی پایگاه‌های داده شامل رویدادهایی در طول حیات خود هستند که ثبت آن‌ها، تغییرات در پایگاه‌های داده را قابل ردگیری می‌کند. همچنین ممیزی در پایگاه‌های داده شامل مشاهده‌ی پایگاه‌داده به منظور شناسایی فعالیت‌های کاربران پایگاه‌داده است. مدیران پایگاه‌های داده ممیزی را با اهداف امنیتی بر روی پایگاه‌های داده فعال می‌کنند. با توجه به اهمیت رویدادنگاری و ممیزی در پایگاه‌های داده و همچنین اهمیت پایگاه‌های داده‌ی NoSQL، در این گزارش رویدادنگاری و ممیزی در پایگاه‌داده کاساندر مورد بررسی قرار گرفته است.

۲ نحوه‌ی ثبت وقایع^۶ در پایگاه‌داده‌ی کاساندر

در این بخش انواع فایل‌های رویدادنگاری مورد بررسی قرار گرفته است. همچنین گزینه‌های قابل تنظیم برای نحوه‌ی نظارت و ثبت وقایع مشخص شده‌اند. این بخش از گزارش بر روی DataStax Cassandra Community نسخه‌ی ۲.۲.۳ بر روی سیستم عامل ویندوز ۷ تهیه شده است.

۲-۱ انواع فایل‌های رویدادنگاری

در

جدول ۱ انواع فایل‌های رویدادنگاری در کاساندر به اختصار توضیح داده شده‌اند.

Recover^۱

Failure^۲

Primary^۳

Secondary^۴

High availability^۵

Logging^۶

جدول ۱ انواع فایل‌های رویدادنگاری در کاساندر

اطلاعات ثبت شده در فایل	نوع فایل رویدادنگاری
ثبت رویدادها در سطوح مختلف	فایل‌های رویدادنگاری تنظیم شده در logback.xml
اطلاعاتی به منظور بازیابی داده‌ها در شرایط شکست ^۷ سخت‌افزاری	Commit Log

در ادامه هر یک از این فایل‌ها مورد بررسی قرار گرفته‌است.

۲-۱-۱ فایل‌های رویدادنگاری تنظیم شده در **logback.xml**

با تغییر فایل **logback.xml** می‌توان خواص رویدادنگاری را در کاساندر تغییر داد. به صورت پیش فرض رویدادهایی در سطح **INFO** در فایلی با نام **system.log** و رویدادهایی در سطح **Debug** در فایلی با نام **debug.log** ثبت می‌شوند (انواع سطوح رویدادنگاری در بخش ۲-۱-۱-۱ توضیح داده شده‌اند). همچنین در صورتی که کاساندر در پیش‌زمینه^۸ اجرا شود، رویدادهای سطح **INFO** در میز فرمان^۹ نیز نمایش داده می‌شوند [۱]. فایل **logback.xml** بر روی ویندوز^۷ در مسیر زیر واقع شده‌است.

```
C:\Program Files\DataStax Community\apache-cassandra\conf
```

تنظیمات پیش فرض در این فایل به صورت زیر است.

```
<!-- SYSTEMLOG rolling file appender to system.log (INFO level) -->
<appender name="SYSTEMLOG"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
    <level>INFO</level>
```

^۷ Failure

^۸ Foreground

^۹ Console

```
</filter>
<file>${cassandra.logdir}/system.log</file>
<rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
  <fileNamePattern>${cassandra.logdir}/system.log.%i.zip</fileNamePattern>
  <minIndex>1</minIndex>
  <maxIndex>20</maxIndex>
</rollingPolicy>
<triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
  <maxFileSize>20MB</maxFileSize>
</triggeringPolicy>
<encoder>
  <pattern>%-5level [%thread] %date{ISO8601} %F:%L - %msg%n</pattern>
</encoder>
</appender>

<!-- DEBUGLOG rolling file appender to debug.log (all levels) -->
<appender name="DEBUGLOG" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${cassandra.logdir}/debug.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>${cassandra.logdir}/debug.log.%i.zip</fileNamePattern>
    <minIndex>1</minIndex>
    <maxIndex>20</maxIndex>
  </rollingPolicy>
  <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
    <maxFileSize>20MB</maxFileSize>
  </triggeringPolicy>
  <encoder>
    <pattern>%-5level [%thread] %date{ISO8601} %F:%L - %msg%n</pattern>
  </encoder>
</appender>

<appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
  <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
```

```
<level>INFO</level>
</filter>
<encoder>
  <pattern>%-5level %date{HH:mm:ss,SSS} %msg%n</pattern>
</encoder>
</appender>
```

همانطور که در تنظیمات فوق نشان داده شده است، appender به منظور تنظیم محل رویدادنگاری و سایر پیکربندی‌های مرتبط با رویدادنگاری استفاده می‌شود. به‌عنوان نمونه، در تنظیمات زیر، ساختار رویدادهای ثبت‌شده مشخص شده است.

```
<pattern>%-5level %date{HH:mm:ss,SSS} %msg%n</pattern>
```

همچنین تنظیمات مربوط به خط‌مشی^{۱۰} آرشیو فایل‌های رویدادنگاری در قسمت‌هایی مشابه قسمت زیر در برچسب^{۱۱} rollingPolicy تنظیم می‌شود. با استفاده از این تنظیمات می‌توان محل و نام فایل‌های آرشیو را مشخص کرد. همچنین می‌توان حداقل و حداکثر اندازه‌ی فایل‌های رویدادنگاری را تنظیم کرد. طبق تنظیمات زیر، در صورتی که سائز فایل رویدادنگاری حداکثر به ۲۰MB برسد، فایل debug.log.<number>.zip تولید و آرشیو می‌شود.

```
<rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
  <fileNamePattern>${cassandra.logdir}/debug.log.%i.zip</fileNamePattern>
  <minIndex>1</minIndex>
  <maxIndex>20</maxIndex>
</rollingPolicy>
```

۲-۱-۱-۱ سطوح رویدادنگاری

مقادیر مجاز برای سطوح رویدادنگاری شامل موارد زیر است [۲, ۵]:

- ALL: ثبت اطلاعات مربوط به رویدادها در تمامی سطوح.

^{۱۰} Policy

^{۱۱} Tag

- **TRACE**: ثبت حداکثر اطلاعات مربوط به رویدادها.
- **DEBUG**: ثبت اطلاعات با جزئیات در مورد روند سیستم.
- **INFO**: ثبت شرایطی که جنبه‌ی اطلاعاتی دارند.
- **WARN**: ثبت شرایط نزدیک به خطا و موقعیت‌های غیرمنتظره یا ناخواسته در زمان اجرا
- **ERROR**: ثبت حداقل اطلاعات مربوط به رویدادها، ثبت خطاهای زمان اجرا یا شرایط غیرمنتظره
- **OFF**: عدم ثبت رویدادها

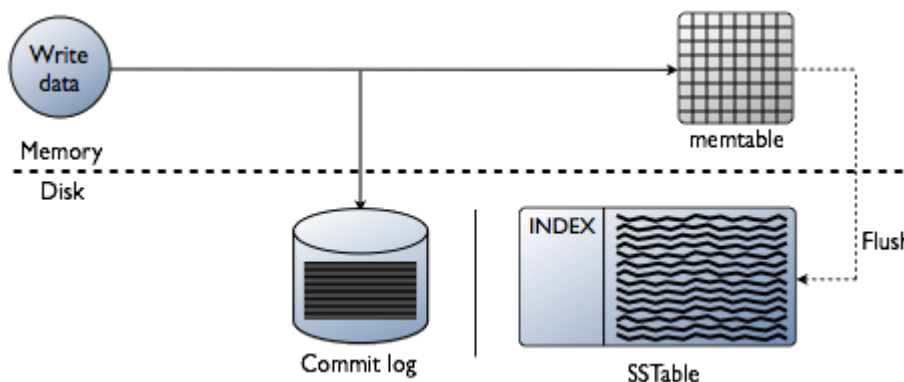
نمونه‌ای از رویدادهای ثبت‌شده در فایل debug.log در شکل ۱ نشان داده شده است.

```
DEBUG [SharedPool-Worker-3] 2017-11-26 20:52:48,226 FileCacheService.java:102 - Evicting cold readers
DEBUG [SharedPool-Worker-1] 2017-11-26 20:52:56,593 AbstractQueryPager.java:95 - Fetched 2 live rows
DEBUG [SharedPool-Worker-1] 2017-11-26 20:52:56,593 AbstractQueryPager.java:112 - Got result (2) smal
DEBUG [SharedPool-Worker-1] 2017-11-26 20:52:56,594 AbstractQueryPager.java:133 - Remaining rows to r
DEBUG [SharedPool-Worker-3] 2017-11-26 20:53:18,227 FileCacheService.java:102 - Evicting cold readers
DEBUG [SharedPool-Worker-3] 2017-11-26 20:53:18,228 FileCacheService.java:102 - Evicting cold readers
DEBUG [SharedPool-Worker-1] 2017-11-26 20:53:19,748 AbstractQueryPager.java:95 - Fetched 3 live rows
DEBUG [SharedPool-Worker-1] 2017-11-26 20:53:19,748 AbstractQueryPager.java:112 - Got result (3) smal
DEBUG [SharedPool-Worker-1] 2017-11-26 20:53:19,748 AbstractQueryPager.java:133 - Remaining rows to r
```

شکل ۱ اطلاعات ثبت‌شده در فایل debug.log

۲-۱-۲ Commit log

زمانی که عمل نوشتن بر روی دیسک انجام می‌شود، کاساندر داده‌ها را در ساختاری در حافظه به نام memTable ذخیره می‌کند و در عین حال اطلاعات را در commit log بر روی دیسک می‌نویسد (شکل ۲). [۳]



شکل ۲ مراحل نوشتن اطلاعات بر روی دیسک

زمانی که محتوای موجود در memTable از حد آستانه‌ی تنظیم‌شده فراتر برود، داده‌های موجود در آن در صفی قرار می‌گیرند تا بر روی دیسک نوشته شوند. داده‌های موجود در commit log پس از تخلیه‌ی داده‌های memTable بر روی دیسک، پاک می‌شوند. در حقیقت commit log برای بازیابی داده‌های موجود در memTable در شرایط شکست در سخت‌افزار کاربرد دارد [۳].

۳ ممیزی^{۱۲} در پایگاه‌داده‌ی کاساندر

در این بخش از گزارش، ممیزی در DataStax Enterprise (DSE) نسخه‌ی ۵،۱،۵ مورد بررسی قرار گرفته‌است. به دلیل آن‌که DataStax Enterprise نسخه‌ی ۵،۱،۵، از ویندوز پشتیبانی نمی‌کند [۶]، این بخش بر روی سیستم عامل Mac OS X نسخه‌ی ۱۰،۱۱،۵ تهیه شده‌است.

رویدادهای ممیزی می‌توانند در فایل‌های رویدادنگاری یا در جدولی در کاساندر ذخیره شوند. هنگامی که ممیزی فعال شود، به صورت پیش‌فرض رویدادها در فایل‌هایی بر روی سیستم عامل ذخیره می‌شوند، در صورتی که سائز رویدادهای ثبت‌شده افزایش پیدا کند، ذخیره‌ی رویدادها در جدولی در کاساندر بهتر خواهد بود [۴].

فایل پیکربندی dse.yaml به منظور فعال‌سازی و تنظیم ممیزی در کاساندر استفاده می‌شود. برای فعال‌سازی ممیزی، در قسمت audit_logging_options مقدار پارامتر enabled را باید true کرد. همچنین سایر تنظیمات مرتبط با ممیزی در فایل dse.yaml در ادامه توضیح داده شده‌است (شکل ۳) [۴]:

- تنظیم گزینه‌ی logger به منظور ثبت رویدادهای ممیزی در جدولی از کاساندر (مقدار CassandraAuditWriter) یا در فایل (مقدار SLF4JAuditWriter)

- تعیین انواع رویدادهایی که باید از آن‌ها ممیزی تهیه شود، از جمله:

- AUTH: ثبت رویدادهای مربوط به ورود به سیستم
- DML: ثبت رویدادهای درج، به‌روزرسانی، حذف و سایر عملیات‌های مرتبط با تغییر داده‌ها

- **DDL**: ثبت ایجاد شیء و نقش، تغییر و حذف آن و سایر رویدادهای مرتبط با تعریف داده‌ها
- **DCL**: ثبت رویدادهای اعطا و حذف نقش، ایجاد نقش و سایر رویدادهای مرتبط با نقش‌ها
- **QUERY**: ثبت تمامی پرسمان‌ها

این موارد با پارامترهای `include_categories` یا `exclude_categories` تنظیم می‌شوند.

- تنظیم فضای کلیدهایی^{۱۳} که بایستی از آن‌ها ممیزی تهیه شود (پارامترهای `included_keyspaces` و `excluded_keyspaces`)

- در صورتی که رویدادها در جدولی در کاساندر ثبت می‌شوند، تنظیم زمان نگهداری رویدادهای ثبت‌شده در پارامتر `retention_time` انجام می‌شود. مقدار پیش‌فرض این پارامتر، صفر به معنی حفظ تمامی رویدادها به طور نامحدود است.

```
audit_logging_options:
  enabled: true

# The logger used for logging audit information
# Available loggers are:
#   CassandraAuditWriter - logs audit info to a cassandra table. This logger can be run either synchronously
#                           asynchronously. Audit logs are stored in the dse_audit.audit_log table.
#                           When run synchronously, a query will not execute until it has been written
#                           to the audit log table successfully. If there is a failure between when an audit
#                           written, and it's query is executed, the audit logs may contain queries that were
#                           executed.
#   SLF4JAuditWriter -    logs audit info to an slf4j logger. The logger name is `SLF4JAuditWriter`,
#                           and can be configured in the logback.xml file.
logger: SLF4JAuditWriter

# # Comma separated list of audit event categories to be included or excluded from the audit log.
# # Defaults to including all categories and keyspaces.
# # Categories are: QUERY, DML, DDL, DCL, AUTH, ADMIN, ERROR
# # Specify either included or excluded categories. Specifying both is an error
# included_categories:
# excluded_categories:

# # Comma separated list of keyspaces to be included or excluded from the audit log.
# # Specify either included or excluded keyspaces. Specifying both is an error
# included_keyspaces:
# excluded_keyspaces:

# The amount of time, in hours, audit events are retained by supporting loggers
# Currently, only the CassandraAuditWriter supports retention time
# values of 0 or less retain events forever
retention_time: 0
```

شکل ۳ تنظیمات مرتبط با ممیزی در فایل dse.yaml

در صورتی که تنظیمات مطابق شکل ۳ تنظیمات مرتبط با ممیزی در فایل dse.yaml اعمال شوند و رویدادها در فایل ثبت شوند، می‌توان ادامه‌ی تنظیمات را در فایل logback.xml انجام داد. بدین منظور تنظیمات پیش‌فرضی در این فایل قرار دارد که می‌توان آن‌ها را پذیرفت و یا تغییر داد. تنظیمات پیش‌فرض مربوط به ممیزی در فایل، در ادامه آورده شده است [۴].

```
<!--audit log-->
<appender name="SLF4JAuditWriterAppender"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${cassandra.logdir}/audit/audit.log</file>
  <encoder>
    <pattern>%-5level [%thread] %date{ISO8601} %X{service} %F:%L -
%msg%n</pattern>
    <immediateFlush>true</immediateFlush>
  </encoder>
  <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>${cassandra.logdir}/audit/audit.log.%i.zip</fileNamePattern>
    <minIndex>1</minIndex>
    <maxIndex>5</maxIndex>
```

```
</rollingPolicy>
<triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
  <maxFileSize>200MB</maxFileSize>
</triggeringPolicy>
</appender>

<logger name="SLF4JAuditWriter" level="INFO" additivity="false">
  <appender-ref ref="SLF4JAuditWriterAppender"/>
</logger>
```

مطابق با این تنظیمات رویدادهای ممیزی در فایل `audit.log` در مسیر `audit.log` در `audit.log` می‌شوند. نمونه‌ای از فایل `audit.log` در شکل ۴ نشان داده شده است.

```
INFO [Native-Transport-Requests-2] 2017-11-27 14:55:51,854 SLF4JAuditWriter.java:96 - host:/192.168.1.8|source:/192.168.1.8
authenticated:anonymous|timestamp:1511781951854|category:DML|type:SET_KS|ks:test1|operation:use test1|consistency level:ONE
INFO [Native-Transport-Requests-1] 2017-11-27 14:55:51,856 SLF4JAuditWriter.java:96 - host:/192.168.1.8|source:/192.168.1.8
authenticated:anonymous|timestamp:1511781951856|category:DML|type:SET_KS|ks:test1|operation:USE "test1"|consistency level:ONE
INFO [Native-Transport-Requests-1] 2017-11-27 14:55:58,650 SLF4JAuditWriter.java:96 - host:/192.168.1.8|source:/192.168.1.8
authenticated:anonymous|timestamp:1511781958650|category:QUERY|type:CQL_SELECT|ks:test1|cf:test|operation:select * from test;
INFO [Native-Transport-Requests-1] 2017-11-27 14:59:43,146 SLF4JAuditWriter.java:96 - host:/192.168.1.8|source:/192.168.1.8
authenticated:anonymous|timestamp:1511782183146|category:DML|type:CQL_UPDATE|ks:test1|cf:test|operation:insert into test (a)
INFO [Native-Transport-Requests-1] 2017-11-27 15:00:05,739 SLF4JAuditWriter.java:96 - host:/192.168.1.8|source:/192.168.1.8
authenticated:anonymous|timestamp:1511782205739|category:DML|type:CQL_UPDATE|ks:test1|cf:test|operation:insert into test (a)
INFO [Native-Transport-Requests-1] 2017-11-27 15:00:07,180 SLF4JAuditWriter.java:96 - host:/192.168.1.8|source:/192.168.1.8
authenticated:anonymous|timestamp:1511782207180|category:QUERY|type:CQL_SELECT|ks:test1|cf:test|operation:select * from test;
```

شکل ۴ فایل `audit.log`

۳-۱ ساختار رویدادهای ثبت شده در `DataStax Enterprise`

رویدادها شامل زوج‌های نام/مقدار است که با (|) از هم جدا شده‌اند.

```
host:/192.168.1.8|source:/192.168.1.8|user:anonymous|authenticated:anonymous|timestamp:15
11781951854|category:DML|type:SET_KS|ks:test1|operation:use test1|consistency
level:ONE
```

این بیغام‌ها شامل قسمت‌های مختلفی هستند که برخی از این قسمت‌ها اختیاری و برخی دیگر اجباری

هستند. در جدول ۲ قسمت‌های اجباری بیان شده‌اند [۴].

جدول ۲ فیلدهای اجباری در ساختار رویدادها

مقدار فیلد	نام فیلد
آدرس گره ^{۱۴} ی (DSE) DataStax Enterprise	Host
آدرس کارخواه ^{۱۵}	Source
کاربر احراز اصالت شده	User
زمان سیستمی رویداد	Timestamp
به‌عنوان مثال DML/DDL/QUERY	Category
عملیات سطح API	Type

۳-۲ ذخیره رویدادها در جدولی در کاساندرا

برای ذخیره‌ی داده‌ها در جدولی در کاساندرا، باید در فایل `dse.yaml` مقدار پارامتر `logger` به `CassandraAuditWriter` تغییر داده شود و `DataStax Enterprise` مجدداً راه‌اندازی شود. رویدادهای ممیزی در جدول `dse_audit.audit_log` نوشته می‌شوند. نمونه‌ای از رویدادهای ثبت‌شده در این جدول در شکل ۵ نشان داده شده‌است.

^{۱۴} Node

^{۱۵} Client

<code>insert into test (a) values (40)\n;</code>	<code>/192.168.1.8</code>	<code>test</code>
<code>CQL_UPDATE</code>	<code>anonymous</code>	
<code>2017-11-27 00:00:00.000000+0000</code>	<code>192.168.1.8</code>	<code>50400 1a620490-d382-1e7-8878-1fe7c4babc51</code>
<code>anonymous</code>	<code>null</code>	<code>QUERY ONE</code>
<code>test1</code>		
<code>select * from test;</code>	<code>/192.168.1.8</code>	<code>test</code>
<code>CQL_SELECT</code>	<code>anonymous</code>	
<code>2017-11-27 00:00:00.000000+0000</code>	<code>192.168.1.8</code>	<code>50400 1d0b0480-d382-1e7-8878-1fe7c4babc51</code>
<code>anonymous</code>	<code>null</code>	<code>DML ONE</code>
<code>test1</code>		

شکل ۵ محتوای جدول `dse_audit.audit_log`

همچنین تنظیمات دیگری مرتبط با ثبت رویدادها در جدول در فایل `dse.yaml` وجود دارد که می‌توان در صورت نیاز آن‌ها را تغییر داد یا بدون تغییر از آن‌ها بهره برد.

`cassandra_audit_writer_options:`

`mode: sync`

`batch_size: 50`

`flush_time: 500`

`num_writers: 10`

`queue_size: 10000`

`write_consistency: QUORUM`

`dropped_event_log: /var/log/cassandra/dropped_audit_events.log`

در ادامه پارامترهای فوق به‌طور مختصر توضیح داده شده‌اند.

- mode: تنظیم روشی که نویسنده‌ی ممیزی^{۱۶} با آن روش، رویدادها را در جدول ممیزی می‌نویسد.
 - sync: در این روش، یک پرسمان تا زمانی که رویداد ممیزی با موفقیت نوشته نشود، اجرا نخواهد شد.
 - async: رویدادهای ممیزی برای نوشته شدن در جدول ممیزی، در صف قرار می‌گیرند و لزومی برای رویدادنگاری پیش از اجرای پرسمان وجود ندارد. یک استخر از نخ^{۱۷}‌های نویسنده، رویدادهای ممیزی را از صف برداشته و آن‌ها را در قالب پرسمان‌های دسته‌ای^{۱۸}، در جدول ممیزی می‌نویسد. درحالی‌که این روش کارایی را در شرایطی که بار^{۱۹} زیاد است بهبود می‌دهد، در صورتی که شکستی بین زمان اجرای پرسمان و زمان نوشته شدن رویداد ممیزی متناظر با پرسمان در جدول اتفاق بیفتد، جدول ممیزی ممکن است سطرهای مرتبط با پرسمان‌های اجراشده را از دست بدهد.
- batch_size: در صورتی که روش async استفاده شود، این پارامتر کاربرد خواهد داشت. این پارامتر باید بزرگتر از صفر باشد. حداکثر تعداد رویدادهایی که نویسنده آن‌ها را از صف برداشته تا آن‌ها را در جدول بنویسد.
- flush_time: در صورتی که روش async استفاده شود، این پارامتر کاربرد خواهد داشت. حداکثر میزان زمان برحسب میلی‌ثانیه، پیش از آنکه رویداد توسط نویسنده از صف خارج شود. این گزینه باعث می‌شود، در صورت وجود تعداد کمی پرسمان در سیستم، رویدادها مدت زمان بسیار زیادی در صف برای نوشته شدن در جدول قرار نگیرند.
- num_writers: در صورتی که روش async استفاده شود، این پارامتر کاربرد خواهد داشت. تعداد نخ‌های نویسنده که رویدادها را در CassandraAuditWriter به صورت ناهمزمان^{۲۰} می‌نویسند.

^{۱۶} Audit writer

^{۱۷} Thread

^{۱۸} Batch queries

^{۱۹} Load

^{۲۰} Asynchronously

- `queue_size`: اندازه‌ی صفی از رویدادهای ممیزی که به نخ‌های نویسنده تزریق می‌شود. زمانی که رویدادهای بیشتری نسبت به سرعت نوشتن نویسنده تولید شود، صف پر می‌شود. در این صورت، پرسرمان‌های جدیدتر موقتاً متوقف می‌شوند تا زمانی که دومرتبه در صف فضای خالی ایجاد شود. در صورتی که مقدار این پارامتر صفر باشد، اندازه‌ی صف نامحدود خواهد بود که می‌تواند منجر به استفاده‌ی بیش از حد از منابع در شرایطی شود که بار پرسرمان‌های اجرایی خیلی زیاد است.
- `write_consistency`: سطح سازگاری که برای نوشتن رویدادهای ممیزی استفاده می‌شود. منظور از سازگاری، چگونگی به‌روزرسانی و همگام‌سازی یک سطر از داده‌های کاساندر در تمامی کارگزارهای کپی شده، است. مقدار پیش‌فرض این پارامتر QUORUM است که به معنی نوشتن در `commit log` و `memtable` در حد نصاب گره‌های تکرار^{۲۱} است.
- `dropped_event_log`: دایرکتوری به منظور ذخیره‌ی فایل رویدادننگاری که رویدادهای حذف شده^{۲۲} را گزارش می‌دهند.

۴ جمع‌بندی

در این گزارش رویدادننگاری و ممیزی در پایگاه‌داده‌ی کاساندر مورد بررسی قرار گرفته‌است. رویدادهای ممیزی می‌توانند در فایل‌های رویدادننگاری یا در جدولی در کاساندر ذخیره شوند. هنگامی که ممیزی فعال شود، به صورت پیش‌فرض رویدادها در فایل‌هایی بر روی سیستم‌عامل ذخیره می‌شوند، در صورتی که سائز رویدادهای ثبت‌شده افزایش پیدا کند، ذخیره‌ی رویدادها در جدولی در کاساندر بهتر خواهد بود. لازم به ذکر است که در رابطه با جرم‌شناسی در کاساندر منبعی یافت نشد.

۵ منابع

- [۱]. http://cassandra.apache.org/doc/latest/getting_started/configuring.html?highlight=loggin

^{۲۱} Replica nodes

^{۲۲} Dropped events

- [۲]. https://docs.datastax.com/en/cassandra/2.1/cassandra/configuration/configLoggingLevels_r.html
- [۳]. https://teddyma.gitbooks.io/learncassandra/content/model/where_is_data_stored.html
- [۴]. https://docs.datastax.com/en/dse/5.1/dse-admin/datastax_enterprise/security/secAuditAbout.html
- [۵]. https://docs.datastax.com/en/dse/5.1/dse-dev/datastax_enterprise/search/loggingSearch.html
- [۶]. https://docs.datastax.com/en/landing_page/doc/landing_page/supportedPlatforms.html#supportedPlatforms__dse-supported-platforms-51