Streaming Analytics Manager 3

Managing Stream Analytics Manager

Date of Publish: 2020-12-15



https://docs.cloudera.com/

Contents

Stream Operations	3
My Applications View	
Application Performance Monitoring	3
Exporting and Importing Stream Applications	4
Troubleshooting and Debugging a Stream Application	5
Monitoring SAM Apps and Identifying Performance Issues	5
Identifying Processor Performance Bottlenecks	10
Debugging an Application through Distributed Log Search	14
Debugging an Application through Sampling	16

Stream Operations

The Stream Operation view provides management of the stream applications, including the following:

- Application life cycle management: start, stop, edit, delete
- Application performance metrics
- Troubleshooting, debugging
- Exporting and importing applications

My Applications View

Once a stream application has been deployed, the Stream Operations displays operational views of the application.

One of these views is called My Application dashboard.

To access the application dashboard in SAM, click **My Application** tab (the hierarchy icon). The dashboard displays all applications built using Streaming Analytics Manager.

Each stream application is represented by an application tile. Hovering over the application tile displays status, metrics, and actions you can perform on the stream application.



Application Performance Monitoring

To view application performance metrics (APM) for the application, click the application name on the application tile.

The following diagram describes elements of the APM view.



Exporting and Importing Stream Applications

Service pool and environment abstractions combined with import and export capabilities allow you to move a stream application from one environment to another. This task provides instructions about importing a stream application that was exported in JSON format.

About this task

To export a stream application, click the Export icon on the **My Application** dashboard. This downloads a JSON file that represents your streaming application.

	cking-I0	Г-St	DEV	:
		0	Clone Export	
emitted 0	TRANSFERRED	errors 0	workers 0	executors 0

Procedure

1. Click on the + icon in My Applications View and select import application:



2. Select the JSON file that you want to import, provide a unique name for the application, and specify which environment to use.

Import Stream		~~~
SELECT JSON FILE *		
Choose File Trucking-IOT-Streaming-Analtyics.json		
TOPOLOGY NAME		
Trucking-IOT-Streaming-Analtics-App-Import		
INVIRONMENT *		
Dev		~
	Cancel	Ok

Troubleshooting and Debugging a Stream Application

Once we have deployed the streaming app, common actions performed by users such as DevOps, Developers, and Operations teams are the following:

- Monitoring the Application and troubleshooting and identifying performance issues
- Troubleshooting an application through Log Search
- Troubleshooting an application through Sampling

SAM makes performing these tasks easier by using the same visual approach as users have when developing the application. We will walk through these common use cases in the below sections.

Monitoring SAM Apps and Identifying Performance Issues

After deploying SAM and running the test generator for about 30 mins, your Storm Operation Mode of the app renders important metrics within each component on the canvas like below.

My Applications / View: streaming-ref-app-advanced_AUTOCREATED		
All Components Log: None Sampling: 0% -	Mode: OVERVIEW METRICS SAMPLE	🤌 🛛 0 10 minutes 🕶 💋 🖉
See Overview Metrics Directly on the SAM App within each Component	Diversion 01 Driversion 01 Control Program Strate (100) 0000 (10	Q Q
Size 242.0 Open 401b Size 242.0 Open 401b Size 242.0 Open 000 Size 240.0 Size 240.0 Size 240.0 Size 240.0 Size 240.0 Size 240.0	Signal Subjective Call Disolation Call Disolation Call Disolation Signal Disolation Call Disolation Call Disolation Call Disolation Call Disolation Signal Disolation Call Disolation Call Disolation Call Disolation Call Disolation Call Disolation Signal Disolation Call Disola	Predict (0) Prediction (0)
	EMI ENRICH-WEA(01) inter from Loady field Add 6.6. 97-9 Gen 9 23. Lay Nov Strategy (N	
All Components - Emitted an Acked an Latency A Failed Workers 94k -0.0k 99k -0.0m 34.0sec +15.4sec 0 0 3	Executors 19	

You can click on **Show Metrics** to get more details on the metrics and drill down on individual metrics. Note the detailed level metrics for **All Components**, **TruckGeoEvent Kafka** source, and **Dashboard-Predictions** Druid Sink.

All Components 🔺	Emitted 🎽 94k -0.0k	Acked ≌ 99k -0.0m	Latency 7 34.0sec +15.4se	Failed ec 0 0	Workers 3	Executors 19	Hide Metrics 🗸
Input/Output							
Acked Tuples							
Failed Tuples							
Queue							
Latency	- T						

All Components -	Emitted 🎽 94k -0.0k	Acked ≌ 99k -0.0m	Latency ⊅ 34.0sec +15.4sec	Failed 00	Workers 3	Executors 19	Hide Metrics 🗸
Input/Output							
Acked Tuples							
Failed Tuples							
Queue							
Latency	- -					_	
Dashboard-Prec Druid	lictions +	Emitted ¥ 730.0 -29 Workers 3	Acked ¥ 0.0 350.0 -160.0 Executors 19	Process I 9.4ms	.atency 凶 -508.7ms	Execute Latency 3 1.5ms -1.4ms	■ Failed O 0 Hide Metrics ✔
Input/Output						_	
Acked Tuples							
Failed Tuples							
Queue							
Process Latency							
Execute Latency						_	

Key metrics include the following:

Metric Name	Description
Execute Latency	The average time it takes an event to be processed by a given component

Metric Name	Description
Process Latency	The average time it takes an event to be acked. Bolts that join, aggregate or batch may not Ack a tuple until a number of other Tuples have been received
Complete Latency	How much time an event from source takes to be fully processed and acked by the topology. This metrics is only available for sources (e.g.: Kafka Source)
Emitted	The number of events emitted for the given time period. For example, for a Kafka Source, it is the number of events consumed for the given time period
Acked	The number of events acked for the given time period. For example, for a Kafka Source, it is the number of events consumed and then acked.

Identifying Throughput Bottlenecks

Looking through the metrics the Source and Sink metrics, we want to increase the throughput such that we emit/ consume more events from the Kafka Topic and send more events to Druid sink over time. We make some changes to the app to increase throughput.

Increase the parallelism of TruckGeoEvent (kafka topic: truck_events_avro) and TruckSpeedEvent (kafka topic: truck_speed_events_avro) from 1 to 3. Note that each of these kafka topics have three partitions.

800	TruckGeoEv KAFKA	4 03 ▶
		_
8	TruckSpeed KAFKA	€ 03) •

Increase the parallelism of the Join from 1 to 3. Since the join is grouped by driverId, we can configure the connection to use fields grouping to send all events with driverId to the same instance of the Join.

TruckGeoEv 4 03)•	Configure each connection to do a gro by driverld so that al events with the same driverld go to the sam instance of the Join	up I s GROUPING*	
		FIELDS	•
		SELECT FIELDS*	
		× driverId	× 👻
KAFKA			

8

Increase the parallelism of the DriverAvgSpeed aggregate window from 1 to 3. Since the window groups by driverId, driverName and route, we can configure the connection to use fields grouping to send all events with those field values to the same instance of the window.

	- *• \sum_{A}^{D}	GGREGATE		
			GROUPING*	
			FIELDS	•
			SELECT FIELDS*	
->• SeventType RULE	4 01 ▶●		× driverId × driverName × route	× •

Increase the parallelism of the Dashboard-Predictions Druid sink from 1 to 3 so we can have multiple JVM instances of Druid writing to the cube.



After making these changes, we re-deploy the app using SAM and run the data generator for about 15 minutes and view seeing the following metrics.

SAM's overview and detailed metrics makes it very easy to verify if the performance changes we made had the desired effect.



Throughput Improvements for the Kafka Source

The below is the before and after metrics for the TruckGeoEvent Kafka Sink:

BEFORE		After
ge TruckGeoEV 4 01] Emitted Complete Failed Acked Latinity 5.1i) 24.2 me 0 2.5ii) Log Nore Sampling: UK	By increasing pa from 1 to 3, w substantial incr throughput with r events consumed and acke	arallelism we see crease in respect to d (emitted) red tag hove Latency Latency Latency Sampling (%
TruckGeoEvent - Emitted M Acked M Complete Latency 77 Failed Workers Executor Acades 5.1k -1.1k 2.5k 500.0 24.2sec +2.2sec 0 0 3 19	Hide Metrics 🗸	TruckGeoEvent - Emitted 7 Added 7 Complete Latency 7 Failed Workers Executors kana kana 15k +ik 7.4k +i.4k 23.6sec +i.8sec 0 0 6 33 Hide Metrics ↓
Acked Tuples		Acked Toples
Queue Complete Latency		Complete Latency

The below is the before and after metrics for the Dashboard-Predictions Druid Sink:

BEFORE				After
Dashboard € 01.3 DRUDD Formated Process Exercise Failed Actions T300.019.4-ms 0.3550.0 Larger Score Sampargung Letter	By increa from 1 substan throughp writt	sing pa to 3, w tial incr ut of th en to D	rallelism e see ease in e events ruid	Dashboard € cs > DRUD € cs > Prested Prested 2.6. 9.5 2.8 0 1.3. segments to
Dashboard-Predictions - Emitted M Acted M Process takeny M Encode Latency M Fail Dual 720.0 2200 250.0 100.0 9.4ms 500.7ms 10.ms 10 m Workers Executors 19 Input/Output Acked Tuples Failed Tuples Duave	ied 0 Metrics 🛩		Dashbu Dreid Input/Output Acked Tuples Failed Tuples Queue	oard-Predictions - Emited 7 Acked 7 Process Latency 3 Excess Latency 7 Field 2.6K +400 (k 1.3K +200) 9.5ms +63.3ms 2.3ms -63.3ms 0 0 Hide Metrics ~ Hide Metrics ~
Process Latency Execute Latency			Process Laten	sy sy

Identifying Processor Performance Bottlenecks

In this scenario, we identify a custom processor that has high latency. After running the data simulator for 30 mins, we view the Overview Metrics of the topology.

My Applications / View: streaming-ref-app-advanced_AUTOCREATED-normalize-delay

II Components Log: None Sampling: 0% -	Mode: OVERVIEW METRICS SAMPLE	Ø 30 minutes Ø
TuckGooEx. 4003	DriverAvgS 43.3 **** BiblinetSp 401 **** Control **** **** **** **** **** **** **** **** **** **** **** **** **** **** ***** **** **** **** **** **** ***** ***** ***** ***** ***** ***** ***** ***** ***** ****** ****** ****** ******* ******* ************************************	Q
Link 433.0 0.000 433.0 500.0 <td< td=""><td>Split Solution Control Cont</td><td>Id: (01) Production (03) torus former family data bits 11:0 3400 Torus former family data torus former family data t</td></td<>	Split Solution Control Cont	Id: (01) Production (03) torus former family data bits 11:0 3400 Torus former family data torus former family data t

Scanning over the metrics, we see that the NORMALIZE-MODEL-FEATURES custom processor has high execute latency of 2 seconds. This means that over the last 30 minutes the average time an event spends in this component is 2 seconds.

		ORMA DRMALIZ	LIZE		4 01 ▶
•	Emitted	Process	Execute	Failed	Acked
	800.0	2.0sec	2.0sec	0	240.0
	Log: None			Sam	npling: <mark>0%</mark>

After making changes to the custom processor to address the latench, we re-deploy the app via SAM and run the data generator for about 15 minutes and view seeing the following metrics.



SAM's overview and detailed metrics makes it very easy to verify if the performance changes we made had the desired effect.

Latency Improvements

The below is the before and after metrics for the NORMALIZE-MODEL-FEATURES custom processor.

BEFORE	AFTER
NORMALIZE Image: Constraint of the state of th	storing the custom e see execute latency substantially which increased throughput d acked increases)
Emitted Acked Process Latency Custom 800.0 240.0 2.0.986 2.0986 Evecute Latency Souch 240.0 2.0986 2.0986 Custom 2.0986 2.0986 33 33 Hide Metrics V Hide Metrics V Hide Metrics V	INORMALIZE-MODEL-FEATURES-DELAY Emitted ೫ Added ೫ Process Latency № Control 28k +23k 92 k +34k 112.5ms Executors Executor 114.7ms +388.5ms 0 6 33 Hide Metrics ∨
Input/Output Acked Tuples	Input/Output
Failed Tuples	Faled Tiples
Queue	Queue
Process Latency Execute Latency	Exercise Latency

In the metric details view, the graphs provides an easy way to compare metrics before and after the code change.



		Emitted A Acked Proces 28k +26k 9.2k +8.4k 112. Execute Latency S Failed	ss Latency 뇌 5ms -1892.5ms Workers Executors
		114.7ms -1888.3ms 0 0	6 33 Hide Metrics ✔
The time after the app was re-deployed with updated custom processor: 1/23/2018 12:38	Input/Output		
	Acked Tuples		
	Failed Tuples		
Event queue	Queue		
dropped from must 1024 to 1	Process Latency		
Execute time	Execute Latency		

You can also select the Metrics tab to validate the performance improvement.

My Applications / View: streaming-ref-app-advanced_AUTOCREATED-normalize-delay



If you zoom in on the NORMALIZE-MODEL-FEATURES component, you will see that after the code change is made, throughput increases and the wait drops to 0.

	NORMALIZE ◀ 01 ►
	Emitted Process Execute Failed Latency Latency 28k 112.5ms 114.7ms 0 Acked
The time when app was re-reployed with changes	

Debugging an Application through Distributed Log Search

In a distributed system, searching for logs on different hosts for different components can be extremely tedious and time consuming. With SAM, all the application logs are indexed via the Ambari Log Search Server via Solr. SAM makes it easy to drill into and search for logs for specific components directly from the DAG view. Follow the below steps to use distributed log search:

Procedure

- 1. To enable Log Search in SAM, perform the following actions in Ambari.
 - a. In Ambari, select the Log Search service and select 'Log Search UI' from Quick Links.
 - **b.** Select the filter icon on the top right menu.
 - c. For the storm_worker component, configure the filter like the following and click Save.

Log Feeder Log Levels Filter

Components	Override	SATAL	ERROR	VARN WARN	INFO	DEBUG	
storm_worker							

2. In SAM, you can dynamically change the logging level. For example, in SAM view mode of an application, click on the Log link, select the log level and the duration you want that log level.

#	All C	ompone	Select	et Log Lin og: Info	1k Samp	oling: (0% -		
LOG LEV	EL							Coloct Log Lo	ual
TRACE	E D	EBUG	INFO	WARN	ER	ROR		Select Log Le	vei
DURATIC	N								
5s	10s	15s	30s	1m	10m	1h		 Select Log Du	iration
SAMPL	ING P	ERCENT	AGE BE	TWEEN 0 TO 1	00 ONLY				
0									
						J. J			

3. Then click on the component you want to search logs for and under Actions select Logs.

			Selec wan	t the c It to se	ompo arch	onent logs i	i you for										
		L															
→ J	OIN-ENRIC	∉01 ▶			ZE	∢ 0	01 🕨	:	Predict			€01 ▶	80	Predicti	on		€03
Emitted	Process Execute Latency Latency 5.9sec 0.0ms	Failed Acked •	Emitted 22k	Process E Latency L 1.4ms 1	atency .0ms	Failed Act	ked 4 k	Emitte	d Process Latency 0.2 ms	Execute Latency 0.2 ms	Failed 0	Acked (Emitte 16k	d Process Latency 0.0 ms	Execute Latency 0.0ms	Failed 0	7.4k
Log: INFO		Sampling: 0%	Log: INFO			Samplin	ng: 0%	Log: INF	0		Sam	pling: 0%	Log: INF	0		Sar	mpling: 0
		SAM 0	PLING PI	ERCENTA	(GE BET	TWEEN O TO	0 100 0M	ILY		Actions View Lo	gs				_	_	
							_		Click	View L	.ogs		_				

4. This brings you to the Log Search page where you can search by component (s), log level(s) and search for strings using wildcard notation.

COMPONENT			LOG LEVEL	
× NORMAL	IZE-MODEL-F	EATURES-DELAY	× 🔻 Select Log Level	
SEARCH				
Search				Ø 3 hours ◄
Date/Time	Log Level	Component Name	Log Message	
3 hours ago	INFO	NORMALIZE- MODEL- FEATURES-DELAY	Preparing bolt 52-NORMALIZE-MODEL-FEATURES-DELAY:(31)	
3 hours ago	INFO	NORMALIZE- MODEL- FEATURES-DELAY	Initialzing FeatureNormalization processor	
3 hours ago	INFO	NORMALIZE- MODEL- FEATURES-DELAY	Configured Delay timeout is (new): 2	
3 hours ago	INFO	NORMALIZE- MODEL- FEATURES-DELAY	Finished Initialzing FeatureNormalization processor	
3 hours ago	INFO	NORMALIZE- MODEL- FEATURES-DELAY	Prepared bolt 52-NORMALIZE-MODEL-FEATURES-DELAY:(31)	
3 hours ago	INFO	NORMALIZE- MODEL- FEATURES-DELAY	About to do feature normalization event: StreamlineEvent{"dataSourceld":"multiple entTime":"2018-01-23 18:11:11.179","eventSource":"truck_geo_event","truckld":84,"d emiee","routeld":6,"route":"Memphis to Little Rock","eventType":"Normal","latitude":3 onld":1,"geoAddress ":No Address Available","speed":67,"split.JoinValue": 15167310" n":"Y";"driverWagePlant:"hours","driverFatigueByHours": 51","driverFatigueByMiles"." ather":0.0,"ModeL_Feature_RinyWeather":0.0,"ModeL_Feature_WindyWeather":1.0;" 9),"auxiliaryFieldsAndValues":0,"header": ("sourceComponentName":"JOIN-ENRICHI 1-4666-a4e+e046ab3bb2f8","7feed3d0-6b40-4e68-ac3a-cec94e040a9b"],"parentId 2772c675219",c9abc1e7-17f4-4ae3-ba99-6180405d7806","318ffe99-00a5-4bf4-93 cb0-1524-46de-8993-14fdf230abe5", "sourceStream":"default")	sources","fieldsAndValu Iriverld":15,"driverName" 5.19,"longitude":-90.04,"c 1179,"week":4,"driverCer 2701","Model_Feature_F eventTimeLong":151673 MENTS","rootlds":["4a14 s":["688aa81-2375-4f30 16b-b2f91e661375"],"Id"
3 hours ago	INFO	NORMALIZE- MODEL- FEATURES-DELAY	Normalized Feautres are: {Model_Feature_FatigueByHours=0.51, Model_Feature_F ature_Certification=1, Model_Feature_WagePlan=0}	atigueByMiles=2.701, M

My Applications / View: streaming-ref-app-advanced_AUTOCREATED-normalize-delay / Log Search

Debugging an Application through Sampling

For troubleshooting, a convenient tool is to turn on sampling for a component or for the entire topology based on a sample percentage. Sampling allows you to log the emitted values of a component in the SAM App.

Procedure

- 1. To enable Log Search in SAM, perform the following actions in Ambari.
 - a. In Ambari, select the Log Search service and select 'Log Search UI' from Quick Links.
 - **b.** Select the filter icon on the top right menu.
 - c. For the storm_worker_event component, configure the filter like the following and click Save.

Log Feeder Log Levels Filter

Components	Override	S FATAL	ERROR	VARN	INFO	DEBUG		
storm_worker_event							v	

2. In SAM view mode of the App, click on the component you want to turn on sampling for and enter a sampling percentage.

	KAFKA	€03 ►			
	Emitted Complete Failed Latency 77k 33.6sec 0 Log: None Samp	Acked 8.1k			
	PLING PERCENTAGE	ETWEEN 0 TO 100 ON	ILY	Actions	
SAM 10		٢	Disable	View Logs	

3. Click the 'SAMPLE' Tab .



4. Use the Sample Search UI to search for different events that were logged.

SELECT COMPONENT :					DATE / TIME :			
× TruckGeoEvent			× •		2018-01-23 14:54:08 - 2018-01-23 15:24:08	Ø 30 n	ninute	łS▼
SEARCH BY KEY:				8	SEARCH BY ID :			
Search by Key Values, Headers, Aux Key Values			Q		Search by Event Id, Root Id, Parent Id			Q
Date/Time	Component	Key Values					ħ	- 0 -
8 minutes ago	TruckGeoEvent	*{eventTime=2018-01-23 21:21:13.616, eventTimeLong=1516742473616, eventSource=truck_geo_event, truckId=14, driverId=13, driverName=Suresh Srinivas, routeId=2, route=Memphis to Little Rock, eventType=Lane Departure, latitude=34.8, longitude=-92.09, correlationId=1, geoAddress=No Address Available}*						
8 minutes ago	TruckGeoEvent	"{eventTime=2018-01-23 21:21:20.486, eventTimeLong=1516742480486, eventSource=truck_geo_event, truckId=106, driverId=11, driverName=Jamie Engesse r, routeId=12, route=Springfield to KC Via Hanibal, eventType=Normal, latitude=39.78, longitude=-93.13, correlationId=1, geoAddress=No Address Available}"						
8 minutes ago	TruckGeoEvent	"{eventTime=2018-01-23 21:21:30.056, eventTimeLong=1516742490056, eventSource=truck_geo_event, truckId=56, driverId=10, driverName=George Vetticad en, routeId=0, route=Peoria to Ceder Rapids Route 2, eventType=Normal, latitude=42.23, longitude=-91.78, correlationId=1, geoAddress=No Address Availabl e}"						
8 minutes ago	TruckGeoEvent	"(eventTime=2018-01-23 21:21:31.546, eventTimeLong=1516742491546, eventSource=truck_geo_event, truckId=101, driverId=21, driverName=Ajay Singh, rou teld=5, route=Memphis to Little Rock Route 2, eventType=Normal, latitude=34.78, longitude=-92.31, correlationId=1, geoAddress=No Address Available)"						
7 minutes ago	TruckGeoEvent	"{eventTime=2018-01-23 21:21:42.586, eventTimeLong=1516742502586, eventSource=truck_geo_event, truckId=104, driverId=14, driverName=Paul Codding, r outeId=3, route=Joplin to Kansas City Route 2, eventType=Normal, latitude=37.31, longitude=-94.31, correlationId=1, geoAddress=No Address Available}"						
7 minutes ago	TruckGeoEvent	"{eventTime=2018-01-23 21:21:45.086, eventTimeLong=1516742505086, eventSource=truck_geo_event, truckId=38, driverId=26, driverName=Don Hilborn, rou teld=1, route=Saint Louis to Memphis, eventType=Normal, latitude=38.43, longitude=-90.35, correlationId=1, geoAddress=No Address Available}"						
7 minutes ago	TruckGeoEvent	*(eventTime=2018-01-23 21:21:48.166, eventTimeLong=1516742508166, eventSource=truck_geo_event, truckId=64, driverId=28, driverName=Michael Aube, r outeId=10, route=Joplin to Kansas City, eventType=Normal, latitude=37.66, longitude=-94.3, correlationId=1, geoAddress=No Address Available)*						
7 minutes ago	TruckGeoEvent	"(eventTime=2018-01-23 21:21:57.636, eventTimeLo teld=7, route=Saint Louis to Chicago, eventType=No	ong=1516 rmal, latit	674 tud	2517636, eventSource=truck_geo_event, truckld=92, driverld=22 e=38.65, longitude=-90.2, correlationId=1, geoAddress=No Addr	, driverName=Chr ess Available}"	is Harr	is, rou
7 minutes ago	TruckGeoEvent	" (eventTime=2018-01-23 21:21:58.666, eventTimeLong=1516742518666, eventSource=truck_geo_event, truckId=17, driverId=29, driverName=Mark Lochbihler (outeld=10, route=Springfield to KC Via Hanibal Route 2, eventType=Normal, latitude=39.71, longitude=-92.07, correlationId=1, geoAddress=No Address Availa ble)*						