Add Druid as Storage backend in Timeline

Bingxue Qiu

Motivation

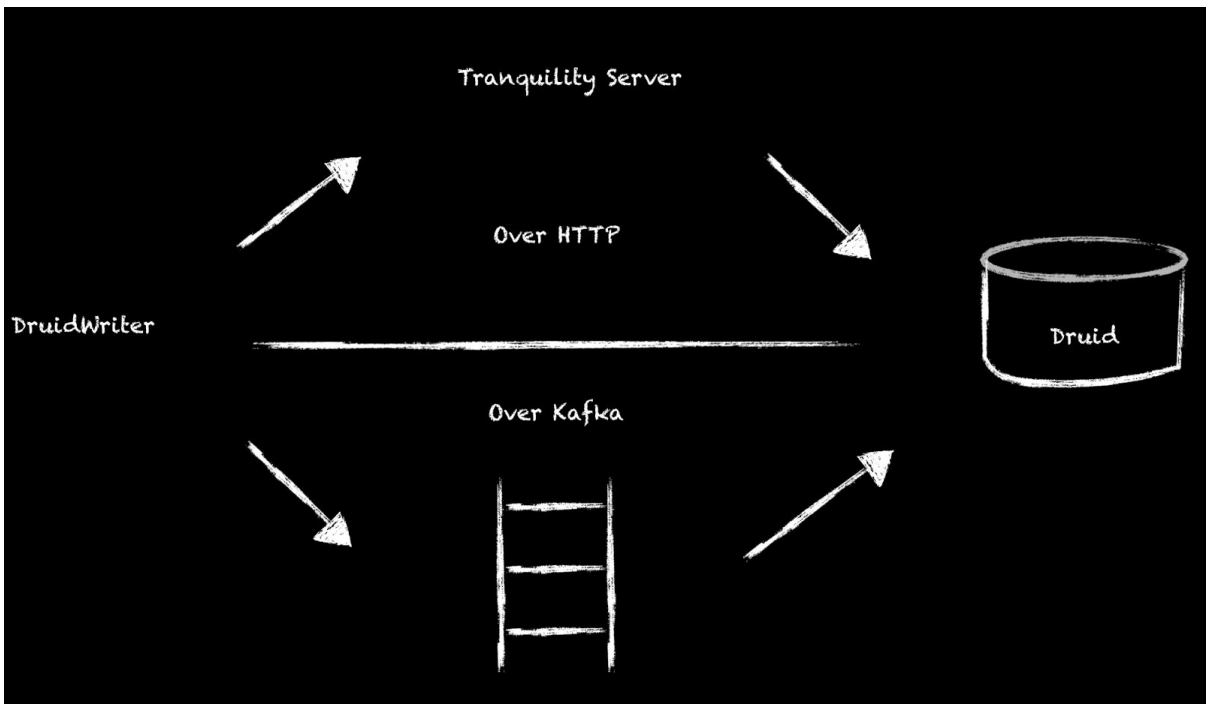
To add druid as storage backend in YARN Timeline service, we should consider about two things:

- Write to Druid
- Druid schema

Design

Druid Writer

We can push a data stream into Druid in real-time using Tranquility over HTTP. For more Scalability and Reliability, we can load data from stream system like kafka by Realtime nodes.The following diagram illustrates the design at a high level:



Druid Schema

Currently, TimelineEntity objects have the following primary fields to hold timeline data:

- metrics

A set of metrics related to this entity. Each metric item contains metric name (id), value.

- events

A set of TimelineEvents, each event contains one id and a map to store related information.

- configs

A map from a string (config name) to a string (config value) representing all configs associated with the entity.

- info

A map from a string (info key name) to an object (info value) to hold up related information for this entity.

We hold the metrics/events/config/info by tables of timelineservice.metric/ timelineservice.event/timelineservice.config/timelineservice.info. With these tables, we can query/agg the data at low cost.

Furthermore, We focus on the info of YARN Application and Container, so we use timelineservice.application/timelineservice.container to hold these. With the application/ container tables, we can ad-hoc/agg applications with cluster/queue/flow/name/state /time/type/tags... and join the result from other tables . Such as: Show the cpu/mem utilization of cluster/queue/flow/user/app/container at any timerange or time series in milliseconds. Display the reports of Top-N/Bottom-N applications up to the standard of resource utilization.

To achieve better (sub-second) OLAP performance, we add the daily_summary table for monthly/yearly summary by daily aggregating app-level data from other table.

The Druid schema is given in bellow:

- timelineservice.metric

Item	Data Type	Description
timestamp	string	Store timestamp
clusterId	string	Identifies the cluster
userId	string	Identifies the user

entityType	string	YARN_APPLICATION_ATTEMPT/YARN_CONTAINER/MAPREDUCE_JOB/MAPREDUCE_TASK/MAPREDUCE_TASK_ATTEMPT ...
entityId	string	Identifies the entity id
metricId	string	YARN_APPLICATION_CPU_METRIC/YARN_APPLICATION_MEM_METRIC/BYTES_READ/BYTES_WRITE...
metricValue	double	Identifies the single value of metric

- timelineservice.event

Item	Data Type	Description
timestamp	string	store timestamp
clusterId	string	Identifies the cluster
entityType	string	YARN_APPLICATION_ATTEMPT/YARN_CONTAINER/MAPREDUCE_JOB/MAPREDUCE_TASK/MAPREDUCE_TASK_ATTEMPT ...
entityId	string	Identifies the entity id
eventId	string	JOB_SUBMITTED/JOB_INITED/JOB_FINISHED...
eventInfoKey	string	FINISH_TIME/STATUS...
eventValue	string	Identifies the event value

- timelineservice.config

Item	Data Type	Description
timestamp	string	Store timestamp
clusterId	string	Identifies the cluster
entityType	string	YARN_APPLICATION_ATTEMPT/YARN_CONTAINER/MAPREDUCE_JOB/MAPREDUCE_TASK/MAPREDUCE_TASK_ATTEMPT ...
entityId	string	Identifies the entity id
configKey	string	YARN_APP_NODE_LABEL_EXPRESSION/.....
configValue	string	Identifies the config value

- timelineservice.info

Item	Data Type	Description
timestamp	string	Store timestamp
clusterId	string	Identifies the cluster
appId	string	Identifies the app id
entityType	string	YARN_APPLICATION_ATTEMPT/YARN_CONTAINER/MAPREDUCE_JOB/MAPREDUCE_TASK/MAPREDUCE_TASK_ATTEMPT ...
entityId	string	Identifies the entity id

infoKey	string	Identifies the info key
infoValue	string	Identifies the info value

- timelineservice.app

Item	Data Type	Description
timestamp	string	Store timestamp
clusterId	string	Identifies the cluster
queue	string	Identifies the queue
userId	string	Identifies the user
appId	string	The app id
flowName	string	Identifies the flow name
flowRunId	long	Identifies the flow run id
flowVersion	string	Identifies the flow version
appName	string	YARN_APPLICATION_NAME
appPriority	long	YARN_APPLICATION_PRIORITY
appState	string	YARN_APPLICATION_STATE
startTime	long	Identifies the start time

finishTime	long	Identifies the finish time
tags	array of string	YARN_APPLICATION_TAGS
appType	string	YARN_APPLICATION_TYPE
callerContext	string	YARN_APPLICATION_CALLER_CONTEXT
diagnosticsInfo	string	YARN_APPLICATION_DIAGNOSTICS_INFO

- timelineservice.container

Item	Data Type	Description
timestamp	string	Store timestamp
clusterId	string	Identifies the cluster
appId	string	Identifies the app id
containerId	string	Identifies the container id
containerState	string	YARN_CONTAINER_STATE
startTime	long	Identifies the start time
finishTime	long	Identifies the finish time
containerType	string	YARN_APPLICATION_ATTEMPT_MASTER_CONTAINER

host	string	YARN_CONTAINER_ALLOCATED_HOST
httpAddress	string	YARN_CONTAINER_ALLOCATED_HOST_HTTP_ADDRESS
port	long	YARN_CONTAINER_ALLOCATED_PORT
containerPriority	long	YARN_CONTAINER_ALLOCATED_PRIORITY
diagnosticsInfo	string	YARN_CONTAINER_DIAGNOSTICS_INFO
exitStatus	double	YARN_CONTAINER_EXIT_STATUS

- timelineservice.daily_summary

Item	Data Type	Description
timestamp	string	Store timestamp
clusterId	string	Identifies the cluster
queue	string	Identifies the queue
userId	string	Identifies the user
appId	string	Identifies the the app id
flowName	string	Identifies the flow name
flowRunId	long	Identifies the the flow run id
flowVersion	string	Identifies the flow version

appName	string	YARN_APPLICATION_NAME
appPriority	long	YARN_APPLICATION_PRIORITY
appState	long	YARN_APPLICATION_STATE
startTime	long	Identifies the start time
finishTime	long	Identifies the finish time
tags	string	YARN_APPLICATION_TAGS
appType	string	YARN_APPLICATION_TYPE
cpuAllocated	double	YARN_CONTAINER_ALLOCATED_VCORE
cpuUsage	double	YARN_APPLICATION_CPU_METRIC
memAllocated	double	YARN_CONTAINER_ALLOCATED_MEMORY
memUsage	double	YARN_APPLICATION_MEM_METRIC

Druid Reader

- Support REST API already exists in ATSv2.
- Support Cluster Resource API

With Cluster Resource API, you can get cluster cpu/mem (waterline/allocation/usage) timeline with specific timerange or conditions.

- Support Cluster Report REST API

With Cluster Report API, you can get the reports of Top-N/Bottom-N applications up to the standard of resource utilization. For example: the top applications of cpu oversold.

- Other custom REST API

Please feel free to give your suggestions