HCLSoftware

Unica Journey V12.1.6 Tuning Guide

Contents

Chapter 1. About tuning Unica Journey for best performance

An installation of Unica Journey consists of several components including third-party tools (such as web application servers, databases, and load balancers) and components such as Unica Platform. All of these components have several properties, features, and settings you can configure to improve performance.

Unica Journey itself has several configuration properties which you can use to tune your installation for best performance.

Defining 'best performance' is difficult. Every environment, every implementation has different requirements.

Unica Journey runtime performance can be affected by many factors, including hardware configuration, network configuration, and Unica Journey configuration. The following guidelines and recommendations can have different results in your environment.

Journey Master Configuration Properties

List of engine properties for configuring Journey Master

1. spring.kafka.retries.config = number of time Kafka Producer Client should try to submit data to broker in case of failure
2. connections.max.idle.ms=-1, specified the idle time to keep Producers and Consumer Alive by default is 540 seconds, changing it to -1 to keep the connections alive.
3. *.topic=DATA_CLEAN--specifies name of topic from where data is read by service(DO NOT CHANGE)
4. *.topic.group.id=data-clean-service-consumer-group , -group to which consumer of this service belongs, this is unique for each service (DO NOT CHANGE)
5. *.topic.max.poll.records = number of message service reads ,each time it communicates with broker
6. *.topic.partitions = number of partition per topic (recommended:5)
7. *.topic.replications = Ref :Configuration for replication of Kafka
8. *.topic.min.threshold = values used for scaling service, based on number of messages polled, number of IO operations performed (default:2), should always be less than max threshold
9. *.topic.max.threshold = values used for scaling service, based on number of messages polled, number of IO operations performed (default:10), should always be greater than min threshold

   (Service threshold ping should be between min and max threshold, if ping is below min then service is considered for scaling-in (Instance is stopped), if threshold ping is more than max then service is considered for scale-out (Instance is increased)
10. *.topic.max.instance = (Number of max Instance this service can have , when threshold value increases, value of this parameter should always be equal to value of topic.partitions (recommended:5)
11. *.topic.default.instance = default number of instance for this service (recommended:1)
12. *.topic.sleep.time = sleep time(in milli seconds) between processing of two batches (recommended:500)
13. *.topic.max.isalive.threshold = threshold value(in milli seconds) to compare heartbeat of each service instance, if threshold value increase this assigned value , instance is restarted (recommended:200)
14. journey_control.topic=JOURNEY_CONTROL, used for communication between web/engine/platform (DO NOT CHANGE)
15. journey.engine.monitoring.topic=JOURNEY_ENGINE_MONITORING, used for monitoring service (DO NOT CHANGE)

16. journey.engine.errors.topic=JOURNEY_ENGINE_ERRORS -all engine errors are posted to this topic (DO NOT CHANGE)
17. pause_audience_exec.topic.max.isalive.threshold = 18000 (in seconds). This value is set based on the following data - `Total audiences: 5 million | Audiences paused: 1 million`. This parameter needs to be set to higher values if audience is higher than 5 million.
18. pause_audience_exec.topic.max.poll.interval = 18000000 (in milliseconds). This value is set based on the following data - `Total audiences: 5 million | Audiences paused: 1 million`. This parameter needs to be set to higher values if audience is higher than 5 million.
19. hip.request.topic=OUTGOING_MESSAGES, topic used for communicating with HIP service for sending (Email, SMS) (DO NOT CHANGE)
20. journey.kafka.consumer.fetch.min.bytes=2097152 and journey.kafka.consumer.max.partition.fetch.bytes=2097152 increasing value for these parameter may give you performance improvements based on the OS and network socket buffer size.
21. decision-split.retry.max-attempts = 3 and decision-split.retry.back-off.delay-ms = 2000 [Fixed delay (in milliseconds) between two retrial attempts for decision split query execution] This parameter needs to be set to higher values in case An Error - Transaction system exception occurs in the logs and JDBC query fails.
22. Oracle DB may need to tune for redo log size based on the Audience data inserted in the Journey tables.

*All Properties are part of journey_master_config.properties file

Table 1. Configuration for replication of Kafka

Number of Kafka Broker	Replication Factor
1	1
2	1
3	2
5	3
7	5

Journey Engine Application Properties

List of engine application properties

spring.ignite.ipFinder.List - This property is used to set pre-configured list of IP addresses specified By default, this IP finder is not shared, which means that all grid nodes have to be configured with the same list of IP addresses. For example (Server A and Server B are in cluster then for:

1. Server A – (ServerA_IP:port,ServerB_IP:Port)
2. Server B - – (ServerB_IP:port,ServerA_IP:Port)

spring.ignite.defaultDataRegion.max.size - This property is used for memory allocation to Ignite; the total size should not be less than 10 MB (Recommended 2GB on 16GB RAM in Linux)/(1GB on 16GB RAM in Windows).

journey.audience.next.state.on.data.error - This property is to decide whether audiences will move to 'no flow' or 'error state'. Values can be 'error' or 'no'

engine.logging.cron - This property is used to set run time of logging interaction scheduler job

journey_configuration.topic - specifies name of topic from where data is read by service(DO NOT CHNAGE)

journey_configuration.topic.group.id Group to which consumer of this service belongs, this is unique for each service (DO NOT CHNAGE)

journey_configuration.short_sleep sleep time(in milli seconds) between processing of two batches

restRequest.topic.max.poll.interval - This parameter need to set to higher values restRequest.topic.max.poll.interval = <3600000> in case commit failed exception is seen in the logs for a specific service then following parameters for that service need to set higher values:

- topic.max.isalive.threshold = 3600 [Sec]
- topic.max.poll.interval = 3600000 [Milli Seconds]

web.client.connections.pendingAcquireTimeout=90 - Specifies the maximum time after which a pending acquire must complete at the TCP level else the TimeoutException will be thrown (The value is in Second)

Journey Engine Service Thread Configuration

Journey engine is using all thread pool configs from service_config.properties

1. **Synchronous thread pool configuration**- This thread pool is managing all journey engine service threads.
    a. **sync.thread.pool.max.size=500** (ThreadPoolExecutor's maximum pool size).
    b. **sync.thread.pool.core.size=200** (ThreadPoolExecutor's core pool size).
    c. **sync.thread.pool.queue.capacity=10** (Capacity for the ThreadPoolExecutor's BlockingQueue).
    d. **sync.thread.pool.max.size=80** (ThreadPoolExecutor's maximum pool size).
    e. **sync.thread.pool.core.size=60** (ThreadPoolExecutor's core pool size).
    f. **sync.thread.pool.queue.capacity=500** (Capacity for the ThreadPoolExecutor's BlockingQueue).
2. **Asynchronous thread pool configuration:** This thread pool is managing all operation which can be done asynchronously. e.g.: Storing discarded data & reporting data into database, delay touchpoint processing.
    a. **async.thread.pool.max.size=20** (ThreadPoolExecutor's maximum pool size).
    b. **async.thread.pool.core.size=5** (ThreadPoolExecutor's core pool size).
    c. **async.thread.pool.queue.capacity=200000** (Capacity for the ThreadPoolExecutor's).
    d. **async.thread.pool.max.size=30** (ThreadPoolExecutor's maximum pool size).
    e. **async.thread.pool.core.size=20** (ThreadPoolExecutor's core pool size).
    f. **async.thread.pool.queue.capacity=100000** (Capacity for the ThreadPoolExecutor's).
3. **Data batching:** Large audience data will be processed in chunk(Batch). Size of batch can be configured as below:
    a. **implicit.service.databatch.timeout = 15** (Waiting time in sec for batch. If batch size is not full till 15 sec, then batch will timeout and whatever records in batch will be processed).
    b. **implicit.service.databatch.batchSize = 100** (Max size of batch).

4. **journey.audience.batch.size** = 500 (Number of audience fetched from database and passed in batches of 500 to first node of journey for processing).
5. **batch.data.import.batch.size=5** (Number of audience records passed in batches of 5 when entrysource is of file type). Max value for this property is 5.
6. **engine.monitoring.time=900000** (Time period in ms after which engine will monitor the components required for engine i.e. Database, kafka and Ignite).
7. **hip.batch.size = 100** (Number of audience is passed to HIP for processing in batches of 2).