

**Unica Journey 12.1.1 -  
Guide d'optimisation**



# Contents

<b>Chapter 1. A propos de l'optimisation de Unica Journey pour optimiser les performances.....</b>	<b>1</b>
Journey Master - Propriétés de configuration.....	1
Journey Propriétés de l'application Engine.....	3
Journey Configuration des unités d'exécution du service Engine.....	4

# Chapter 1. A propos de l'optimisation de Unica Journey pour optimiser les performances

Une installation d'Unica Journey est constituée de plusieurs composants comprenant des outils tiers (tels que des serveurs d'applications Web, des bases de données et des équilibreurs de charge) et de composants tels que Unica Platform. Tous ces composants ont plusieurs propriétés, fonctions et paramètres que vous pouvez définir pour régler les performances.

Unica Journey dispose de plusieurs propriétés de configuration que vous pouvez utiliser pour régler les performances de l'installation.

La définition des performances optimales est difficile. Chaque environnement, chaque mise en œuvre a des besoins différents.

Les performances de l'environnement d'exécution Unica Journey peuvent être affectés par divers facteurs, notamment, la configuration du matériel, la configuration du réseau et la configuration Unica Journey. Les instructions et recommandations suivantes peuvent avoir un impact différent dans votre environnement.

## Journey Master - Propriétés de configuration

### Liste des propriétés Engine pour la configuration de Journey Master

1. `spring.kafka.retries.config` = nombre de fois où le client Kafka Producer doit essayer d'envoyer des données au courtier en cas de défaillance
2. `connections.max.idle.ms=-1`, par défaut, le temps d'inactivité spécifié pour conserver les connexions des producteurs et du consommateur actives est de 540 secondes. Valeur remplacée par -1 pour maintenir les connexions actives.
3. `*.topic=DATA_CLEAN` - Indique le nom de la rubrique à partir de laquelle les données sont lues par le service (NE PAS MODIFIER)
4. `*.topic.group.id=data-clean-service-consumer-group` - Groupe auquel le consommateur de ce service appartient, il est unique pour chaque service (NE PAS MODIFIER)

5. \*.topic.max.poll.records = nombre de lectures de service de messages, chaque fois qu'il communique avec le courtier
6. \*.topic.partitions = nombre de partitions par rubrique (recommandé : 5)
7. \*.topic.replications = : configuration pour la réplication de Kafka
8. \*.topic.min.threshold = valeurs utilisées pour la mise à l'échelle du service, en fonction du nombre de messages interrogés, le nombre d'opérations d'E/S effectuées (par défaut : 2). Doit toujours être inférieur au seuil maximal
9. \*.topic.max.threshold = valeurs utilisées pour la mise à l'échelle du service, en fonction du nombre de messages interrogés, le nombre d'opérations d'E/S effectuées (par défaut : 10). Doit toujours être supérieur au seuil minimal.  
  
(La valeur de ping du service doit être comprise entre le seuil minimal et le seuil maximal. Si la valeur de ping est inférieure à la valeur minimale, alors une baisse de capacité est envisagée pour le service (arrêt de l'instance). Si la valeur de ping est supérieure à la valeur maximale, une augmentation de capacité est envisagée pour le service (l'instance est augmentée))
10. \*.topic.max.instance = (nombre d'instances maximal que ce service peut avoir, lorsque la valeur de seuil augmente, la valeur de ce paramètre doit toujours être égale à la valeur de topic.partitions (recommandé : 5)
11. \*.topic.default.instance = nombre par défaut d'instances pour ce service (recommandé : 1)
12. \*.topic.sleep.time = durée de veille (en millisecondes) entre le traitement de deux lots (recommandé : 500)
13. \*.topic.max.isalive.threshold = valeur de seuil (en millisecondes) pour comparer les signaux de présence de chaque instance de service. Si la valeur de seuil dépasse la valeur affectée, l'instance est redémarrée (recommandé : 200)
14. journey\_control.topic=JOURNEY\_CONTROL, utilisé pour la communication entre Web/Engine/Platform (NE PAS MODIFIER)
15. journey.engine.monitoring.topic=JOURNEY\_ENGINE\_MONITORING, utilisé pour le service de surveillance (NE PAS MODIFIER)

16. `journey.engine.errors.topic=JOURNEY_ENGINE_ERRORS` - Toutes les erreurs d'Engine sont publiées dans cette rubrique (NE PAS MODIFIER)

17. `hip.request.topic=OUTGOING_MESSAGES`, rubrique utilisée pour la communication avec le service HIP pour l'envoi (E-mail, SMS) (NE PAS MODIFIER)

**\* Toutes les propriétés font partie du fichier  
`journey_master_config.properties`**

**Table 1. Configuration de la réplication de Kafka**

Nombre de courtiers Kafka	Facteur de réplication
1	1
2	1
3	2
5	3
7	5

## Journey Propriétés de l'application Engine

### Liste des propriétés d'application Engine

**`spring.ignite.ipFinder.List`** - Cette propriété est utilisée pour définir la liste pré-configurée des adresses IP spécifiées par défaut, cette propriété n'est pas partagée, ce qui signifie que tous les nœuds de grille doivent être configurés avec la même liste d'adresses IP. Par exemple (le serveur A et le serveur B sont en cluster, donc pour :

1. Serveur A – (ServerA\_IP: port, ServerB\_IP: port)
2. Serveur B--(ServerB\_IP: port, ServerA\_IP: port)

**`spring.ignite.defaultDataRegion.max.size`** - Cette propriété est utilisée pour l'allocation de mémoire à Ignite ; la taille totale ne doit pas être inférieure à 10 Mo (2 Go sur 16 Go de RAM sous Linux)/(1 Go sur 16 Go de RAM sous Windows).

**journey.audience.next.state.on.data.error** - Cette propriété permet de décider si les audiences seront déplacées vers 'no Flow' ou 'error State'. Les valeurs peuvent être 'error' ou 'no'

**engine.logging.cron** - Cette propriété est utilisée pour déterminer le moment d'exécution du travail de planificateur d'interaction de journalisation

**journey\_configuration.topic** - Indique le nom de la rubrique à partir de laquelle les données sont lues par le service (NE PAS MODIFIER)

**journey\_configuration.topic.group.id** - Groupe auquel le consommateur de ce service appartient, il est unique pour chaque service (NE PAS MODIFIER)

**journey\_configuration.short\_sleep** - Temps de veille (en millisecondes) entre le traitement de deux lots

## Journey Configuration des unités d'exécution du service Engine

### Journey Engine utilise toutes les configurations de pools d'unités d'exécution du fichier service\_config.properties

1. **Configuration du pool d'unités d'exécution synchrones** - Ce pool d'unités d'exécution gère toutes les unités d'exécution du service Journey Engine
  - a. **sync.thread.pool.max.size=80** (taille maximale du pool de l'exécuteur de pool d'unités d'exécution).
  - b. **sync.thread.pool.core.size=60** (taille du pool principal de l'exécuteur de pool d'unités d'exécution).
  - c. **sync.thread.pool.queue.capacity=500** (capacité de la file d'attente de blocage de l'exécuteur de pool d'unités d'exécution).
2. **Configuration du pool d'unités d'exécution asynchrones** : Ce pool d'unités d'exécution gère toutes les opérations qui peuvent être exécutées de façon asynchrone. Exemple : le stockage de données rejetées et des données de génération de rapports dans la base de données, le traitement de point de contact de retard.

- a. **async.thread.pool.max.size=30** (taille maximale du pool de l'exécuteur de pool d'unités d'exécution).
  - b. **async.thread.pool.core.size=20** (taille du pool principal de l'exécuteur de pool d'unités d'exécution).
  - c. **async.thread.pool.queue.capacity=100000 = 100000** (capacité de l'exécuteur de pool d'unités d'exécution).
3. **Traitement par lots de données** : les données d'audience volumineuses seront traitées par bloc (lot). La taille du lot peut être configurée comme suit :
  - a. **implicit.service.databatch.timeout = 15** (Temps d'attente en secondes pour le lot. Si la taille des lots n'est pas atteinte au bout de 15 secondes, le délai d'attente du traitement par lots sera dépassé et les différents enregistrements du lot seront traités.
  - b. **implicit.service.databatch.batchSize = 100** (Taille maximale du lot).
4. **journey.audience.batch.size = 500** (Nombre d'audiences extraites de la base de données et transmises par lots de 500 au premier nœud de Journey pour traitement).
5. **batch.data.import.batch.size=5** (Nombre d'enregistrements d'audience transmis par lots de 5 lorsqu'entrysource est de type Fichier). La valeur maximale de cette propriété est 5.
6. **engine.monitoring.time=900000** (période de temps en ms après laquelle Engine surveillera les composants dont il a besoin, c.-à-d., base de données, Kafka et Ignite).
7. **hip.batch.size = 100** (le nombre d'audiences est transmis à HIP pour traitement par lots de 2).