

Unica Interact

12.1 - Guide d'optimisation



Table des matières

Chapitre 1. A propos de l'optimisation de Unica Interact pour optimiser les performances.....	1
Gestion du cache.....	2
Utilisation d'Ehcache.....	2
Utilisation avec Ignite.....	7
Interface de programme d'application Unica Interact.....	10
Installation et configuration du réseau.....	10
Gestion de diagramme temps réel.....	11
Optimisation des services.....	12
Optimisation du serveur d'applications Web.....	12
Arguments JVM.....	12
Pool de connexions.....	14
Terminologie d'optimisation.....	15
Optimisation de la base de données.....	16
Optimisation d'ETL.....	16
Journalisation.....	20

Chapitre 1. A propos de l'optimisation de Unica Interact pour optimiser les performances

Une installation de Unica Interact 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 et Unica Campaign. Tous ces composants ont plusieurs propriétés, fonctions et paramètres que vous pouvez définir pour régler les performances.

Unica Interact 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. Par exemple, une mise en œuvre de Unica Interact où toutes les données des diagrammes temps réel sont collectées depuis les données temps réel, sera optimisée différemment comparée à une mise en œuvre nécessitant des informations lues dans plusieurs tables de base de données.

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

Les instructions suivantes sont organisées par rapport aux composants associés. L'ordre de modification des paramètres n'a pas d'importance.

Pour Simulator

- Simulator doit être exécuté dans un environnement de test / de transfert, car le simulateur exécute un nombre d'unités d'exécution plus élevé et peut consommer beaucoup de CPU sur le serveur d'exécution.

Gestion du cache

Dans un environnement Unica Interact volumineux où vous utilisez un grand nombre de serveurs d'exécution, vous pouvez utiliser un logiciel de gestion de cache pour répartir la charge d'exécution entre les serveurs et améliorer les performances en temps réel du groupe de serveurs d'exécution dans son ensemble. Lorsque vous activez la solution de gestion de cache avec Unica Interact, vous pouvez utiliser plusieurs serveurs d'exécution, renforcés par un équilibreur de charge. L'équilibreur de charge répartit la charge de travail sur les serveurs d'exécution du groupe et aide à maintenir l'affinité de session, ce qui signifie que lorsqu'une session entrante est gérée par un serveur d'exécution A, alors lorsque le même utilisateur émet des requêtes supplémentaires, ces requêtes sont remplies par la session sur le serveur A. Cette méthode d'équilibrage des connexions parmi les serveurs d'exécution dans le groupe peut considérablement améliorer les performances, mais présente des limitations car toutes les informations de session sont gérées en mémoire et les limites de mémoire de chaque machine virtuelle Java™ restreignent la quantité d'informations de session pouvant être gérées. Unica Interact prend en charge deux solutions de gestion de cache par défaut :

- **Ignite** : solution de mise en cache répartie centrée sur la mémoire fournie avec Interact. Pour plus d'informations sur Ignite, veuillez vous reporter à <https://apacheignite.readme.io/docs/what-is-ignite>.
- **Ehcache**, solution de mise en cache open source, incluse avec chaque installation d'Interact. Pour plus d'informations sur le logiciel Ehcache, voir <http://www.ehcache.org/documentation/>

Utilisation d'Ehcache

Pour améliorer les performances de votre groupe de serveurs d'exécution Unica Interact, vous pouvez configurer Ehcache en modifiant une série de propriétés de configuration dans Unica Platform.

Pour activer Ehcache en tant que gestionnaire de cache pour chaque serveur d'exécution Unica Interact, définissez la propriété de configuration suivante dans Unica Platform sur EHCACHE :

Unica Interact cacheManagement > caches > Interact cache > cacheManagerName

Pour utiliser Ehcache en tant que gestionnaire de cache pour stocker des états de modèle d'événement pour un groupe de serveurs d'exécution, définissez également le paramètre suivant sur EHCACHE :

Unica Interact > cacheManagement > caches > PatternStateCache >
cacheManagerName

Vous devez répéter ce processus sur chaque serveur d'exécution Unica Interact dans le groupe de serveurs pour activer Ehcache en tant que gestionnaire de cache pour le groupe de serveurs.

Un fois que vous avez activé Ehcache en tant que gestionnaire de cache, vous pouvez configurer les paramètres pour optimiser la mise en cache pour votre installation.

Modification des paramètres de configuration Ehcache

Lorsque vous spécifiez qu'un serveur d'exécution Unica Interact doit utiliser le gestionnaire de cache intégré appelé Ehcache pour améliorer les performances, vous pouvez configurer les paramètres utilisés par Ehcache pour optimiser sa valeur dans votre groupe de serveurs d'exécution.

Pour configurer le gestionnaire de cache Ehcache, vous pouvez ouvrir les propriétés de configuration suivantes dans Unica Platform :

Unica Interact > cacheManagement > Cache Managers > EHCACHE > Parameter Data

Cette catégorie de configuration contient un ensemble de propriétés de configuration par défaut pour Ehcache qui correspondent aux paramètres que vous pouvez spécifier dans un fichier de configuration Ehcache. Vous pouvez également créer des paramètres supplémentaires dans cette catégorie en cliquant sur **(Parameter)** et en le nommant pour correspondre au paramètre Ehcache que vous souhaitez modifier.

Pour des informations sur les propriétés de configuration pour Ehcache, voir Unica Interact - Guide d'administration, Annexe B : Propriétés de configuration de l'environnement d'exécution d'Unica Interact. Vous pouvez également consulter la documentation Ehcache se trouvant dans <http://www.ehcache.org/documentation/>.

Notez que pour des performances optimales avec Ehcache, dans les paramètres de configuration du serveur d'exécution Unica Interact dans Unica Platform, définissez la plus petite valeur possible pour le délai d'attente de la session (`Interact > cacheManagement > caches > InteractCache > TimeoutInSecs`).

Chaque session Unica Interact contient une certaine quantité de données de session en mémoire. Plus les sessions sont longues, plus les besoins en mémoire simultanés sont élevés. Par exemple, si vous attendez 50 sessions par seconde et que chaque session reste active pendant 20 minutes, la mémoire doit pouvoir prendre en charge 60 000 sessions à la fois, si chaque session dure 20 minutes.

La valeur doit être logique avec le scénario. Par exemple, une session de système d'appel doit pouvoir rester active pendant une minute, alors qu'une session de site Web doit rester active pendant 10 minutes.

Prise en charge d'un plus grand nombre de sessions simultanées dans Ehcache

Lorsque vous utilisez Ehcache en tant que gestionnaire de cache dans certains environnements Unica Interact, un grand nombre de sessions simultanées peut amener l'environnement d'exécution Unica Interact à dépasser sa mémoire disponible et provoquer un arrêt du système ou une erreur de manque de mémoire. Une erreur de manque de mémoire est plus susceptible de se produire si vous avez augmenté le paramètre de configuration `maxEntriesInCache` (`Unica Interact > cacheManagement > Caches > InteractCache > maxEntriesInCache`) en lui affectant une valeur supérieure à la valeur par défaut, bien que cette situation puisse exister, même avec la valeur par défaut maximale de 100 000 sessions. Pour éviter ce problème, vous pouvez réduire la valeur **maxEntriesInCache** ou suivre les instructions fournies ici pour modifier le cache système pour enregistrer la mémoire cache sur disque. Cette modification permet d'utiliser un beaucoup plus grand nombre de sessions simultanées qu'il ne serait possible autrement.

Pour empêcher l'environnement d'exécution Unica Interact de dépasser la mémoire disponible dans le segment de mémoire JVM (Java™ virtual machine), vous pouvez modifier le mécanisme de mise en cache pour utiliser le stockage sur disque pour mettre en cache les données qui dépassent la mémoire disponible.

Les administrateurs système peuvent ajuster la quantité de mémoire disponible pour les systèmes de déploiement à l'aide des paramètres JVM suivants :

```
-Xms####m -Xmx####m -XX:MaxPermSize=256m
```

Où les caractères #### correspondent à 2 048 ou à une valeur plus élevée (selon la charge du système). Notez qu'un serveur d'applications 64 bits et qu'une machine virtuelle Java sont généralement nécessaires pour les valeurs supérieures à 2 048.

Unica Interact utilise le système Ehcache de mise en cache distribué Open Source pour mettre en cache les données. Par défaut, Unica Interact utilise les paramètres spécifiés par Unica Platform pour gérer la mise en cache Ehcache. Cependant, vous pouvez remplacer ces paramètres pour Unica Interact en créant votre propre fichier de configuration Ehcache qui est automatiquement chargé lorsque Unica Interact démarre.

Pour charger un fichier de configuration Ehcache personnalisé au démarrage, les conditions suivantes doivent exister :

- La JVM doit contenir la propriété `interact.ehcache.config` de paramètre, comme dans l'exemple suivant :

```
-Dinteract.ehcache.config=/temp/abc.xml
```

Vous pouvez définir une propriété JVM pour le serveur d'applications Web dans le script de commande de démarrage Oracle WebLogic) ou Admin Console (IBM® WebSphere®). Les informations dans `/temp/abc.xml` sont le chemin d'accès au fichier XML qui contient la configuration Ehcache à charger lors du démarrage.

- Un fichier de configuration qui contient les paramètres de configuration Ehcache valides dans le format XML doit exister dans l'emplacement défini par la propriété JVM.

Si vous ne définissez pas cette propriété ou que vous la définissez et qu'il n'existe pas de fichier de configuration dans l'emplacement défini, Unica Interact utilise sa configuration de cache par défaut.

Si les deux conditions sont réunies, le fichier de configuration Ehcache est chargé au démarrage et ses paramètres remplacent les paramètres de configuration par défaut Unica Interact de mise en cache des données de sessions.

L'exemple suivant montre un fichier de configuration (de format XML) que vous pouvez utiliser pour personnaliser Ehcache :

```
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../../main/config/ehcache.xsd">

<defaultCache
maxElementsInMemory="10"
eternal="false"
timeToIdleSeconds="5"
timeToLiveSeconds="10"
overflowToDisk="true"
/>

<cache name="InteractCache"
maxElementsInMemory="5"
eternal="false"
timeToIdleSeconds="600"
timeToLiveSeconds="600"
overflowToDisk="true">

<cacheEventListenerFactory
class="com.unicacorp.interact.cache.ehcache.EHCacheEventListenerFactory"
properties=""/>
<!--For InteractCache, use the following to use the
EHCacheEventListenerFactory:-->
<cacheEventListenerFactory
class="com.unicacorp.interact.cache.EHCacheEventListenerFactory"
properties=""/>
<!--For PatternStateCache, use the following to use
the EHCacheEventListenerFactory:-->
<cacheEventListenerFactory
class="com.unicacorp.interact.cache.PatternStateCacheEventListenerFactory"
```

```

    properties=" " />
  </cache>

</ehcache>

```

Si vous avez enregistré ce fichier sous `/HCL/Unica Interact/conf/Ehcache.xml`, définissez la propriété JVM de l'application Web, comme dans l'exemple suivant :

```
-Dinteract.ehcache.config=/HCL/Unica Interact/conf/Ehcache.xml
```

Pour la liste complète des options disponibles afin de modifier le logiciel Ehcache, voir <http://www.ehcache.org/documentation/>

Utilisation avec Ignite

Pour améliorer les performances de votre groupe de serveurs d'exécution Unica Interact, vous pouvez configurer Ignite en modifiant une série de propriétés de configuration dans Unica Platform.

Pour activer Ignite en tant que gestionnaire de cache pour chaque serveur d'exécution Unica Interact, définissez la propriété de configuration suivante dans Unica Platform sur Ignite :

```
Unica Interact cacheManagement > caches > Interact cache > cacheManagerName
```

Pour utiliser Ignite en tant que gestionnaire de cache pour stocker des états de modèle d'événement pour un groupe de serveurs d'exécution, définissez également le paramètre suivant sur Ignite :

```
Unica Interact > cacheManagement > caches > PatternStateCache >
cacheManagerName
```

Vous devez répéter ce processus sur chaque serveur d'exécution Unica Interact dans le groupe de serveurs pour activer Ignite en tant que gestionnaire de cache pour le groupe de serveurs.

Une fois que vous avez activé Ignite en tant que gestionnaire de cache, vous pouvez configurer les paramètres pour optimiser la mise en cache pour votre installation.

Modification des paramètres de configuration Ignite

Lorsque vous spécifiez qu'un serveur d'exécution Unica Interact doit utiliser le gestionnaire de cache intégré appelé Ignite pour améliorer les performances, vous pouvez configurer les paramètres utilisés par Ignite pour optimiser sa valeur dans votre groupe de serveurs d'exécution.

Pour configurer le gestionnaire de cache Ignite, vous pouvez ouvrir les propriétés de configuration suivantes dans Unica Platform :

`Unica Interact > cacheManagement > Cache Managers > Ignite > Parameter Data`

Cette catégorie de configuration contient un ensemble de propriétés de configuration par défaut pour Ignite qui correspondent aux paramètres que vous pouvez spécifier dans un fichier de configuration Ignite. Vous pouvez également créer des paramètres supplémentaires dans cette catégorie en cliquant sur **(Parameter)** et en le nommant pour correspondre au paramètre Ignite que vous souhaitez modifier.

Pour des informations sur les propriétés de configuration pour Ignite, voir Unica Interact - Guide d'administration, Annexe B : Propriétés de configuration de l'environnement d'exécution d'Unica Interact. Vous pouvez également consulter la documentation Ignite se trouvant dans <https://apacheignite.readme.io/docs>.

Notez que pour des performances optimales avec Ignite, dans les paramètres de configuration du serveur d'exécution Unica Interact dans Unica Platform, définissez la plus petite valeur possible pour le délai d'attente de la session (`Interact > cacheManagement > caches > InteractCache > TimeoutInSecs`).

Chaque session Unica Interact contient une certaine quantité de données de session en mémoire. Plus les sessions sont longues, plus les besoins en mémoire simultanés sont élevés. Par exemple, si vous attendez 50 sessions par seconde et que chaque session reste active pendant 20 minutes, la mémoire doit pouvoir prendre en charge 60 000 sessions à la fois, si chaque session dure 20 minutes.

La valeur doit être logique avec le scénario. Par exemple, une session de système d'appel doit pouvoir rester active pendant une minute, alors qu'une session de site Web doit rester active pendant 10 minutes.

Prise en charge d'un plus grand nombre de sessions simultanées dans Ignite

Lorsque vous utilisez Ignite en tant que gestionnaire de cache dans certains environnements Unica Interact, un grand nombre de sessions simultanées peut amener l'environnement d'exécution Unica Interact à dépasser sa mémoire disponible et provoquer un arrêt du système ou une erreur de manque de mémoire. Une erreur de manque de mémoire est plus susceptible de se produire si vous avez augmenté le paramètre de configuration `maxEntriesInCache` (Unica Interact > cacheManagement > Caches > InteractCache > `maxEntriesInCache`) en lui affectant une valeur supérieure à la valeur par défaut, bien que cette situation puisse exister, même avec la valeur par défaut maximale de 100 000 sessions. Pour éviter ce problème, vous pouvez réduire la valeur **maxEntriesInCache** ou suivre les instructions fournies ici pour modifier le cache système pour enregistrer la mémoire cache sur disque. Cette modification permet d'utiliser un beaucoup plus grand nombre de sessions simultanées qu'il ne serait possible autrement.

Pour empêcher l'environnement d'exécution Unica Interact de dépasser la mémoire disponible dans le segment de mémoire JVM (Java virtual machine), vous pouvez modifier le mécanisme de mise en cache pour utiliser le stockage sur disque pour mettre en cache les données qui dépassent la mémoire disponible.

Cependant, activer le stockage sur disque entraînera une dégradation des performances. Les administrateurs système peuvent ajuster la quantité de mémoire disponible pour les systèmes de déploiement à l'aide des paramètres JVM suivants :

```
-Xms####m -Xmx####m -XX:MaxPermSize=256m
```

où les caractères `####` correspondent à 2 048 ou à une valeur plus élevée (selon la charge du système).

 **Remarque** : un serveur d'applications 64 bits et qu'une machine virtuelle Java sont généralement nécessaires pour les valeurs supérieures à 2048.

Basculer

Lorsque Distributed Ignite est utilisé, les données sont enregistrées sur l'ensemble de la grille. Si l'instance qui héberge une session est interrompue, les données de cette session seront perdues et toutes les requêtes ultérieures de cette session échoueront. Pour parvenir

à un basculement, vous devez configurer le paramètre `numberOfBackups` sous `Affinium|Interact|cacheManagement|Gestionnaires de cache|Ignite|Données de paramètre`. Sa valeur représente le nombre de copies de sauvegarde des données dans le cache. Une valeur supérieure entraînera une protection du basculement et des performances de lecture améliorées avec le même coût que des performances d'écriture inférieures. Par conséquent, le choix doit être fait en fonction des scénarios d'utilisation.

Interface de programme d'application Unica Interact

Utilisez l'API Java Serialization à la place de l'API SOAP. L'API Serialization offre un traitement plus efficace (5 à 10 fois supérieur en fonction de la configuration de l'application) et des temps de réponse plus courts.

Pour des informations sur l'implémentation d'un délai d'expiration personnalisé de l'API Java, consultez la documentation de l'API installée avec Unica Interact à l'adresse <Interact_home>/docs/apiJavaDoc ou consultez l'échantillon de code et les commentaires dans <Interact_home>/samples/api/SampleCustomizedInteractAPI.java.

Installation et configuration du réseau

Si le serveur Unica Interact doit gérer les données de session dans plusieurs appels d'API Unica Interact et que vous utilisez le gestionnaire de cache Ehcache pour améliorer les performances, utilisez l'équilibrage de charge "sticky load" et la gestion de session locale à la place de la gestion de session répartie.

A l'aide de Ehcache, le mode réparti subit le coût des communications entre les serveurs d'exécution Unica Interact pour maintenir la cohérence des sessions. La gestion de session locale peut éviter ce coût.

Dans les paramètres de configuration Unica Platform du serveur d'exécution Unica Interact, affectez la valeur `local` à la propriété `Interact > cacheManagement > EHCACHE > Parameter Data > cacheType`.

Gestion de diagramme temps réel

Chaque diagramme temps réel nécessite au moins une unité d'exécution pour pouvoir fonctionner. Vous pouvez surveiller un système dynamique pour déterminer s'il existe un nombre suffisant d'unités d'exécution pour tous les diagrammes temps réel.

En utilisant la console JMX, vous surveillez les statistiques JMX

`CurrentJobsInProgressBoxQueue` et `CurrentJobsInSchedulerQueue` sous `com.unicacorp.interact.flowchart`. Dans l'absolu, il ne doit en exécuter aucune, même en période de charge maximale, ce qui implique qu'il existe un nombre suffisant d'unités d'exécution pour gérer les demandes pour les exécutions de diagrammes.

 **Remarque** : L'exécution de la console JMX affecte les performances. N'exécutez pas la console JMX dans un environnement de production, sauf pour diagnostiquer un problème.

Vous pouvez surveiller ces files d'attente avec le nombre d'unités d'exécution utilisées par les diagrammes temps réel. Vous définissez les tailles de pool d'unités d'exécution des diagrammes temps réel dans Unica for Unica Interact Runtime sous `Interact > diagramme`.

- Affectez à `maxNumberOfFlowchartThreads` au minimum le nombre maximal d'utilisateurs simultanés dans le client Unica Interact. Par exemple, si le nombre maximal d'utilisateurs simultanés est égal à 50 et que chaque appel de segmentation exécute un diagramme, affectez à `maxNumberOfFlowchartThreads` la valeur 50.
- Définissez `maxNumberOfProcessBoxThreads` en fonction du nombre moyen de chemins simultanés dans les diagrammes temps réel et selon que les diagrammes sont liés au processeur ou aux E-S. La valeur doit être au moins égale à `maxNumberOfFlowchartThreads`. Par exemple, si le nombre moyen de chemins simultanés est égal à 2 et que toutes les zones de processus sont liées au processeur, définissez `maxNumberOfProcessBoxThreads` pour que la propriété soit égale à $2 * \text{maxNumberOfFlowchartThreads}$. Si les zones de processus sont liées aux E-S (par exemple, si elles exécutent des recherches et des écritures, telles que Sélection ou Instantané), il peut être nécessaire d'augmenter cette valeur.

- Affectez à `minNumberOfFlowchartThreads` la même valeur que `maxNumberOfFlowchartThreads`. De même, affectez à `minNumberOfProcessBoxThreads` la même valeur que `maxNumberOfProcessBoxThreads`.

Optimisation des services

Unica Interact dispose de plusieurs services qui gèrent les lectures et écritures de divers composants de Unica Interact, tels que le module d'auto-apprentissage et le module d'historique des réponses et des contacts.

Définissez le seuil de chaque service (sous `Interact > services > service name > cache > threshold`) en utilisant les valeurs appropriées en fonction du nombre d'opérations par seconde et du délai de chaque insertion dans la base de données. Par exemple, si le système traite 500 transactions par seconde et que chaque transaction a deux appels de contact de journal, le seuil `contactHist` doit correspondre à une valeur basée sur le délai moyen d'écriture d'un lot et 1000 contacts de journal par seconde.

Optimisation du serveur d'applications Web

Dans Unica Interact, vous optimisez l'application Web en modifiant les connexions et les arguments de machine virtuelle Java. Ces arguments affectent le débit et le délai de démarrage. Le nombre de connexions que vous utilisez est déterminé par les fonctions que vous avez activées.

Vous devez également consulter la documentation des serveurs d'applications Web et des systèmes d'exploitation pour plus d'informations sur les meilleures pratiques d'optimisation des performances.

Arguments JVM

Les arguments JVM (Java virtual machine) doivent être définis dans le script de commandes de démarrage ou Admin Console du serveur d'applications Web.

- Vérifiez que vous avez installé les derniers modules de mise à jour et correctifs du système d'exploitation, du serveur d'applications Web et JVM.

- Pour optimiser les performances, lors de l'utilisation de HotSpot VM, utilisez l'argument `-server`.
- Définissez la taille de segment de mémoire maximale pour la machine JVM en fonction de la disponibilité de mémoire dans le serveur. (Unica Interact n'est pas une application qui consomme beaucoup de mémoire). Définissez des valeurs de taille maximale et minimale de segments de mémoire identiques (en utilisant les arguments `-Xmx` et `-Xms`), ce qui allonge le temps de démarrage, mais améliore le traitement.
- Si l'application ne répond pas périodiquement, par exemple, longs temps de réponse ramenés à quelques secondes, il peut être nécessaire d'optimiser les règles de collection de place. Monitor Garbage Collection s'exécute en utilisant la console JMX et en étudiant la sortie de récupération de place après avoir activé les arguments suivants.

```
-verbosegc -XX:+PrintGCDetails
```

- Dans nos tests, Low Pause Collector élimine les lenteurs associées à la récupération de place sans affecter le traitement. Voici un groupe d'options jugées utiles pour un segment de mémoire JVM de 2 Go.

```
-XX:+UseConcMarkSweepGC -Xmn512m -XX:SurvivorRatio=6
```

En général, la collection récente doit correspondre au quart ou à la moitié du segment de mémoire total. L'espace Survivor peut être défini pour correspondre à 1/8ème de la taille de la collection récente.

- Si vous utilisez une année à deux chiffres (par exemple, 01-01-20) ou des dates à partir du 01/01/2020 inclus lorsque vous utilisez une macro Date, vous devez ajouter le paramètre JVM ci-après au démarrage de l'application pour que l'année à deux chiffres soit convertie en année à quatre chiffres, comme le requiert l'application.

```
-DInteract.enableTwoDigitYearFix=true
```

- Dans certains cas, le déploiement d'anciens canaux interactifs existants ou de canaux interactifs possédant des historiques de déploiement volumineux peut contraindre le système et requérir 2 048 Mo ou plus d'espace de segment de mémoire Java pour la phase de conception Campaign et/ou la phase d'exécution Interact.

Les administrateurs système peuvent ajuster la quantité de mémoire disponible pour les systèmes de déploiement à l'aide des paramètres JVM suivants :

`-Xms####m -Xmx####m -XX:MaxPermSize=256m`

Où les caractères #### correspondent à 2 048 ou à une valeur plus élevée (selon la charge du système). Notez qu'un serveur d'applications 64 bits et qu'une machine virtuelle Java sont généralement nécessaires pour les valeurs supérieures à 2 048.

Références

- Tuning Garbage Collection with the 5.0 Java virtual machine (http://java.sun.com/docs/hotspot/gc5.0/gc_tuning_5.html)
- Livre blanc sur l'optimisation de Java (<http://java.sun.com/performance/reference/whitepapers/tuning.html>)

Pool de connexions

Définissez la taille du pool de connexions de la source de données de l'environnement d'exécution Unica Interact en utilisant le console du serveur d'applications. Tenez compte du nombre d'utilisateurs simultanés et des connexions établies pendant la durée de vie de la session, ce qui inclut le chargement du profil et de la suppression d'offres, les lectures et les écritures de diagramme et les lectures d'apprentissage.

Fonctionnalité/Option	Connexions nécessaires si activée
<p>Au moins une des fonctions suivantes est activée</p> <ul style="list-style-type: none"> • Chargement de la table de profil • Chargement de la table de suppression d'offres • Chargement de la table de remplacement de score 	<p>1 connexion par appel client concurrent à <code>startSession</code> ou <code>setAudience</code></p> <p>Le fait qu'un seul chargement de table ou que les trois chargements de table soient activés n'a pas d'importance.</p>
Apprentissage	2 connexions

Fonctionnalité/Option	Connexions nécessaires si activée
Au moins un service de journalisation ou de suivi est activé	La valeur de <code>Interact > services > threadManagement > flushCacheToDB > maxPoolSize</code> . La valeur par défaut est 5.
Diagrammes exécutant au moins 1 appel de base de données	La valeur de <code>Interact > flowchart > maxNumberOfFlowchartThreads</code> . La valeur par défaut est 25.

Par exemple, les besoins sont les suivants.

- Nécessité de 30 appels simultanés à `startSession` sans attente lors de l'obtention d'une connexion à la base de données (30)
- Apprentissage activé (2)
- Tous les services activés (5)
- Au moins un diagramme déployé établissant une connexion de base de données (25)
- Utilisation des valeurs par défaut actuelles (0)

Ensuite, vous devez définir une taille de pool de connexions de base de données avec la valeur minimale 62 (30+2+5+25) pour optimiser les performances lorsqu'aucun client de la connexion n'attend.

Terminologie d'optimisation

L'optimisation système possède des définitions spécifiques pour les termes courants.

Temps de réponse

Délai de réponse de l'environnement d'exécution Unica Interact à une demande d'API, mesuré sur le client.

Débit

Nombre de transactions par seconde.

Transaction

Appel au serveur d'exécution Unica Interact par l'API Unica Interact, y compris les appels définis par la classe `InteractAPI`, tels que `startSession` et `setAudience`. L'appel `executeBatch` est une transaction, même s'il peut contenir plusieurs commandes. Elles n'incluent pas les méthodes qui fonctionnent avec des objets de réponse, tels que la classe `Offer`.

Optimisation de la base de données

L'optimisation de la base de données implique l'ajout d'index à des tables spécifiques et la mise à jour des statistiques.

Ajoutez les index appropriés dans les tables de profil, de suppression d'offres et de remplacement de score.

- Tables de profil. Créez un index à entrées unique dans les zones de niveau d'audience.
- Tables de suppression d'offres. Créez un index dans les zones de niveau d'audience.
- Tables de remplacement de score. Créez un index dans les zones de niveau d'audience.

En outre, vérifiez que les statistiques de ces index sont à jour. Par exemple, si l'ID audience est une combinaison de deux colonnes `CustomerId` et `HouseholdId`, créez un index sur ces colonnes dans toutes les tables et mettez à jour les statistiques.

Optimisation d'ETL

Lorsque vous configurez le module d'historique des réponses et des contacts, le module utilise un processus ETL (Extract, Transform, Load) en arrière-plan pour transférer les données des tables de transfert d'exécution vers les tables d'historique des réponses et des contacts Unica Campaign.

Cette section décrit les paramètres de configuration facultatifs que vous pouvez changer dans Unica Interact pour améliorer les performances de l'outil ETL. Il se peut que vous n'ayez pas à modifier les valeurs par défaut de ces paramètres, mais si vous le faites, suivez les instructions fournies ici et dans le document `Unica Interact - Guide d'administration` pour modifier les performances de l'outil.

Toutes les propriétés décrites ici se trouvent dans les propriétés de configuration Unica Campaign dans `Campaign | partitions | partition[n] | Interact | contactAndResponseHistTracking`.

Propriété de configuration	Valeur et description
<code>processSleepIntervalInMinutes</code>	Le nombre de minutes pendant lesquelles le module de l'historique des réponses et des contacts d'Unica Interact attend entre les copies des données des tables de transfert d'exécution d'Unica Interact dans les tables de l'historique des réponses et des contacts d'Unica Campaign. La valeur par défaut est 60.
<code>purgeOrphanResponseThresholdInMinutes</code>	Cette propriété détermine le délai qu'attend Unica Interact avant de purger les réponses qui n'ont pas de contacts correspondants (appelées également réponses orphelines). La valeur par défaut est 180. Toutefois, pour traiter un grand nombre d'enregistrements, le retard entre les traitements des contacts et des réponses peut être plus long, et vous augmentez donc cette valeur pour que les réponses ne soient pas purgées trop rapidement.
<code>maxJDBCInsertBatchSize</code>	Définit le nombre total d'enregistrements que le module d'historique des réponses et des contacts traitent dans une seule itération. Il s'agit du nombre maximal d'enregistrements d'un lot JDBC à traiter (et à assembler dans un lot), car avant de valider la requête dans les tables du système Unica Campaign, la valeur par défaut est 1 000.

Propriété de configuration

Valeur et description

Comme cette valeur fonctionne conjointement avec la propriété `maxJDBCFetchBatchSize`, il peut être nécessaire d'augmenter cette valeur si cette propriété a également augmenté de manière significative. Par exemple, si vous affectez à `maxJDBCFetchBatchSize` la valeur 2 500 000, vous pouvez faire passer cette valeur à 10 000 pour traiter l'augmentation des enregistrements.

Notez que les besoins en mémoire augmentent lorsque vous augmentez cette valeur. La valeur 10 000 affectée à cette propriété définit une bonne limite maximale du fait des demandes de mémoire.

`maxJDBCFetchBatchSize`

Détermine le nombre maximal d'enregistrements à extraire de la base de données de transfert pour une opération de traitement d'un lot ETL. La valeur par défaut est 1000, mais pour régler les performances du module d'historique des réponses et des contacts, veillez à définir une valeur supérieure au nombre d'enregistrements d'historique des réponses et des contacts généralement traités tous les jours.

Cette propriété est utilisée avec `maxJDBCFetchChunkSize` et `maxJDBCInsertBatchSize` pour déterminer le mode de traitement des enregistrements. Supposons que les valeurs aient été définies comme suit :

Propriété de configuration

Valeur et description

- `maxJDBCFetchBatchSize` : 30000
- `maxJDBCFetchChunkSize` : 1000
- `maxJDBCInsertBatchSize` : 1000

Dans cet exemple, 30 000 enregistrements sont extraits (ou le nombre total d'enregistrements s'ils sont inférieurs à 30 000). Ensuite, le module d'historique des réponses et des contacts s'exécute en boucle dans les 30 000 enregistrements, en traitant 1 000 à la fois de manière que 1 000 enregistrements soient marqués dans les tables de transfert et 1 000 soient insérés dans la table d'historique des contacts détaillé.

`maxJDBCFetchChunkSize`

Détermine le nombre maximal d'enregistrements dans un bloc JDBC (depuis un total maximal de `maxJDBCFetchBatchSize` enregistrements) à traiter avec chaque passe. La valeur par défaut est 1000. Dans certains cas, vous pouvez améliorer les performances en augmentant cette valeur pour qu'elle soit supérieure à la valeur de la propriété `MaxJDBCInsertBatchSize`.

`deleteProcessedRecords`

Cette propriété, qui indique si les enregistrements d'historique de réponses et de contacts doivent être conservés après leur traitement, est affectée par défaut de la valeur `YES`. La modification de cette valeur permet de mieux contrôler le flux de données dans le processus ETL et affecte les performances (en reportant la purge de

Propriété de configuration`fetchSize`**Valeur et description**

ces enregistrements, selon votre décision). Toutefois, vous devez savoir exécuter la maintenance de ces enregistrements manuellement pour qu'ils soient supprimés au moment approprié. Modifiez cette propriété avec précaution.

La définition d'une valeur pour `fetchSize` JDBC peut améliorer les performances des lots volumineux d'enregistrements, mais l'amélioration des performances réseau a une conséquence dans la mesure où les grandes tailles de recherche affectent l'utilisation de la mémoire. Voir la description de cette propriété de configuration dans le Unica Interactguide d'administration pour plus d'informations sur l'ajustement de cette propriété.

Pour les descriptions détaillées de chacune des propriétés de configuration, consultez l'aide en ligne de cette page de configuration ou le document Unica Interact - Guide d'administration.

Journalisation

Vérifiez que le niveau de consignation défini est INFO ou ERROR. N'utilisez jamais un paramètre de consignation prolixe, tel que DEBUG ou TRACE dans un environnement de production.

Vous pouvez définir la journalisation dans trois emplacements :

- Fichier `interact_log4j.properties`. Par défaut, ce fichier se trouve dans le répertoire `<install_dir>/Interact/conf`, où `<install_dir>` est le répertoire parent contenant les produits installés.

- Vérifiez que l'API Unica Interact n'effectue aucune journalisation. La journalisation est déterminée par le paramètre booléen disponible dans les méthodes `startSession` et `setDebug`.
- Vérifiez que la surveillance JMX est définie sur Info avec l'opération JMX `activateInfo`.