IBM Unica Interact
Version 8 Release 6
May 25, 2012

Tuning Guide

Note

Before using this information and the product it supports, read the information in “Notices” on page 11.

This edition applies to version 8, release 6, modification 0 of IBM Unica Interact and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About tuning Interact for best performance

An installation of Interact consists of several components including third-party tools (such as web application servers, databases, and load balancers) and IBM® Unica® components such as Marketing Platform and Campaign. All of these components have several properties, features, and settings you can configure to improve performance. Interact 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. For example, an implementation of Interact where all data for interactive flowcharts is gathered from real-time data, would be tuned differently than an implementation requiring information read from several database tables.

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

The following guidelines are organized by related components. The order in which you modify any settings does not matter.

Interact API

Use the Java Serialization API instead of SOAP API. The Serialization API provides better throughput (can be 5-10 times more depending on the application configuration) and a shorter response time.

Installation and network configuration

If the Interact server is required to maintain session data across multiple Interact API calls, use sticky load balancing and local session management instead of distributed session management. Distributed mode incurs the cost of communication between the Interact runtime servers to keep the sessions consistent.

InIBM Unica Marketing Platform configuration settings for the Interact runtime server, set the `Interact > sessionManagement > cacheType` property to `local`.

Session management

In the IBM Unica Marketing Platform configuration settings for the Interact runtime server, set the session timeout (`Interact > sessionManagement > sessionTimeoutInSecs`) to the smallest acceptable value.

Each Interact session contains some amount of session data in memory. The longer you maintain sessions, the more concurrent memory requirements you have. For example, if you are expecting 50 sessions per second, and each session can remain active for 20 minutes, you might require the memory to support 60,000 sessions at a time, if every session lasted the full 20 minutes.

The value must be logical for your scenario. For example, a call system session might need to remain active for a minute, but a website session should remain active for 10 minutes.

Supporting a greater number of concurrent sessions

In some Interact environments, a high number of concurrent sessions may cause the Interact runtime to exceed its available memory, causing a system slowdown or out-of-memory error. An out-of-memory situation is more likely if you have increased the `maxNumberOfSessions` configuration parameter (**Interact | sessionManagement | maxNumberOfSessions**) to a number higher than the default setting, although it can occur even with the default setting of a maximum of 100,000 sessions. To avoid this issue, you can reduce the **maxNumberOfSessions** value, or you can follow the instructions here to modify the system memory cache to roll the cached memory over to disk storage. This modification allows many more concurrent sessions than would otherwise be possible.

To prevent the Interact runtime from exceeding the available memory in the Java virtual machine (JVM) memory heap, you can modify the memory caching mechanism to use disk storage for caching the data that exceeds the available memory.

Interact uses an open source distributed caching system called Ehcache for caching data. By default, Interact uses the settings specified by the IBM Unica Marketing Platform to manage the Ehcache caching. However, you can override those settings for Interact by creating your own Ehcache configuration file that is automatically loaded whenever Interact starts up.

To load a custom Ehcache configuration file on startup, the following must be true:

- Your JVM must include the parameter `interact.ehcache.config` property, as in the following example:

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

  You can set a JVM property for your web application server in the startup command script (Oracle WebLogic) or Admin Console (IBM WebSphere®). The information in */temp/abc.xml* is the actual path to the XML file containing the Ehcache configuration you want to load at startup.
- A configuration file containing valid Ehcache configuration settings in XML format must exist at the location specified by the JVM property.

  If you do not set this property, or if you set this property and there is no configuration file at the specified location, Interact uses its default cache configuration.

If both conditions are true, the Ehcache configuration file is loaded on startup, and its settings override any default Interact configuration parameters for caching session data.

The following example shows a sample configuration file (in XML format) that you might use to customize 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.EHCacheEventListenerFactory"
 properties=""/>
</cache>

</ehcache>
```

If you saved this file as /IBM/Interact/conf/Ehcache.xml, you would then set the JVM property for the web application as in the following example:

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

For a complete list of the options available for modifying the Ehcache software, see http://ehcache.org/files/documentation/EhcacheUserGuide-2.0-2.3.pdf

Interactive flowchart management

Every interactive flowchart requires at least one thread to run. You can monitor a live system to see whether there are enough threads for all the interactive flowcharts. Using JMX console, monitor the JMX statistics CurrentJobsInProcessBoxQueue and CurrentJobsInSchedulerQueue under com.unicacorp.interact.flowchart. Ideally, they should be zero even under peak load, which implies there are enough threads to handle the requests for flowchart executions.

Note: Running the JMX console does impact performance. You should not run the JMX console in a production environment except when diagnosing an issue.

You can control these queues with the number of threads used by interactive flowcharts. You set flowchart thread pool sizes in IBM Unica Marketing for Interact runtime under Interact > flowchart.

- Set maxNumberOfFlowchartThreads to be at least the maximum number of concurrent users expected on the Interact client. For example, if the maximum number of concurrent users is 50 and each call to segmentation runs one flowchart, set maxNumberOfFlowchartThreads to 50.
- Set maxNumberOfProcessBoxThreads based on the average number of concurrent paths in the flowcharts and whether the flowcharts are CPU bound or I/O bound. It should be at least equal to maxNumberOfFlowchartThreads. For example, if the average number of concurrent paths in the flowcharts is 2 and all the process boxes are CPU bound, set maxNumberOfProcessBoxThreads to be 2*maxNumberOfFlowchartThreads. If the process boxes are I/O bound (for example if they perform database lookups or writes, such as a Select or Snapshot process), then that number might need to be set to a larger value.
- Set minNumberOfFlowchartThreads to be the same as maxNumberOfFlowchartThreads. Likewise, set minNumberOfProcessBoxThreads to be the same as maxNumberOfProcessBoxThreads.

Service tuning

Interact has several services that manage database reads and writes by various components of Interact, for example, the built-in learning module and the contact and response history module.

Set threshold for each of the services (under `Interact > services > service name > cache > threshold`) to appropriate values based on the number of operations per second and the time for each insert to database. For example, if the system throughput requirements are 500 transactions per second and each transaction has 2 log contact calls, then the `contactHist` threshold should be set to a value based on the average time to write a batch and 1000 log contacts per second.

Web application server tuning

Besides the following sections, also see the documentation for your web application servers and operating systems for information about best practices for performance tuning.

JVM arguments

Java virtual machine (JVM) arguments are defined in the startup command script (Oracle WebLogic) or Admin Console (IBM WebSphere) for your web application server.

- Confirm you have the latest service packs and patches installed for your operating system, web application server, and JVM.
- For best performance, when using Sun HotSpot VM, use the `–server` argument.
- Decide on a maximum heap size for the JVM, based on the memory availability in the server. (Interact is not a memory intensive application). Set the max and min sizes of the heap to be the same (using `–Xmx` and `-Xms` arguments), which increases the startup time, but gives better throughput.
- If the application is unresponsive periodically, for example, long response times running to few seconds, the Garbage Collection policy might need to be tuned. Monitor Garbage Collection runs using JMX console and by studying the Garbage Collection output after enabling the following arguments.

  ```
  -verbosegc -XX:+PrintGCDetails
  ```
- In our tests, the Low Pause Collector was found to eliminate Garbage Collection-related slowness without sacrificing throughput. The following is one set of options was found to be useful for a 2 GB JVM heap.

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

  In general, the young collection should be about 1/4 to 1/2 of the total heap. The Survivor Space can be set to 1/8th the size of young collection.

References

- Tuning Garbage Collection with the 5.0 Java virtual machine (http://java.sun.com/docs/hotspot/gc5.0/gc_tuning_5.html)
- Java Tuning white paper (http://java.sun.com/performance/reference/whitepapers/tuning.html)

Connection pool

Set the size of the connection pool of the Interact runtime data source using the application server console. Take into account the number of concurrent users and the connections made during the lifetime of a session, which includes loading profile, loading offer suppression, reads and writes from flowcharts, and reads from learning.

Feature/Option	Connections Required If Enabled
At least one of the following features is enabled • Load profile table • Load Offer Suppression table • Load Score Override table	1 connection per concurrent client call to startSession or setAudience It does not matter if only one table load or all three table loads are enabled.
Learning	2 connections
At least one logging or tracking service enabled	The value of `Interact > services > threadManagement > flushCacheToDB > maxPoolSize`. The default is 5.
Flowcharts that make at least 1 database call	The value of `Interact > flowchart > maxNumberOfFlowchartThreads`. The default is 25.

For example, if you have the following requirements.

- Require that 30 concurrent calls to `startSession` not wait when obtaining a database connection (30)
- Have learning turned on (2)
- All services turned on (5)
- Have at least one deployed flowchart that makes a db connection (25)
- Rely on current defaults (0)

Then you should set up a database connection pool size with a minimum of 62 (30+2+5+25) for optimal performance where no single consumer of the connection will wait.

Tuning terminology

The following tuning guidelines were determined based on the following terms.

Response time: The amount of time it takes for the Interact runtime server to respond to an API request as measured from the client side.

Throughput: The number of transactions per second.

Transaction: A call to the Interact runtime server by the Interact API, including calls defined by the InteractAPI class such as `startSession` and `setAudience`. The `executeBatch` call is one transaction, even though it can contain several commands. These do not include methods that work with response objects, for example, the Offer class.

Database tuning

Add appropriate indexes in Profile, Offer suppression, and Score override tables.

- Profile tables. Create a unique index on the audience level fields.
- Offer suppression tables. Create an index on the audience level fields.
- Score override tables. Create an index on the audience level fields.

Also, make sure that the statistics on these indexes are up to date. For example, if the Audience ID is a combination of two columns `CustomerId` and `HouseholdId`, create an index on these columns in all the tables and update the statistics.

ETL Tuning

When you configure the contact and response history module, the module uses a background Extract, Transform, Load (ETL) process to move data from the runtime staging tables to the Campaign contact and response history tables.

This section describes the optional configuration settings you might want to change in IBM Unica Interact to improve the performance of the ETL tool. You might not need to modify any of these configuration parameters from their default settings; however, if you do, follow the guidelines here and in the *IBM Unica Interact Administrator's Guide* to modify the tool's performance.

All of the properties described here are found in Campaign configuration properties, in `Campaign | partitions | partition`*`[n]`*` | Interact | contactAndResponseHistTracking`.

Configuration Property	Value and Description
`processSleepIntervalInMinutes`	The number of minutes the Interact contact and response history module waits between copying data from the Interact runtime staging tables to the Campaign contact and response history tables. The default value is 60.
`purgeOrphanResponseThresholdInMinutes`	This property determines how long Interact waits before purging responses that have no corresponding contact (also known as "orphaned responses"). The default is 180; however, for processing many records, the delay between processing contacts and responses may be greater, and you would therefore increase this value to prevent responses from being purged too quickly.

Configuration Property	Value and Description
maxJDBCInsertBatchSize	Out of the total number of records that the contact and response history module processes in one iteration, this is the maximum number of records of a JDBC batch to process (and assemble together in a batch) before committing the query into the Campaign system tables The default value is 1000. Because this value works together with the maxJDBCFetchBatchSize property, you might need to increase this value if that property also increased significantly. For example, if you set maxJDBCFetchBatchSize to 2,500,000, you might increase this value to 10,000 to handle the increase in records. Note that memory requirements increase as you increase this value; a setting of 10,000 for this property is a good upper-limit due to memory demands.
maxJDBCFetchBatchSize	Determines the maximum number of records to fetch from the staging database for an ETL batch processing operation. The default value is 1000, but to tune the performance of the contact and response history module, make sure to set this value to a number greater than the number of contact history records generally processed each day. This property is used together with maxJDBCFetchChunkSize and maxJDBCInsertBatchSize to determine how the records are processed. For example, suppose the values were set as shown here: • maxJDBCFetchBatchSize: 30000 • maxJDBCFetchChunkSize: 1000 • maxJDBCInsertBatchSize: 1000 In this example, 30,000 records are fetched (or the total number of records if there are under 30,000). Then, the contact and response history module loops through that 30,000 records, processing 1,000 at a time, so that 1,000 records are marked in the staging tables, and 1,000 are inserted into the detail contact history table.
maxJDBCFetchChunkSize	Determines the maximum number of records in a JDBC chunk (from a total of up to maxJDBCFetchBatchSize records) to process with each pass The default value is 1000. In some cases, you might be able to improve performance by increasing this value above the MaxJDBCInsertBatchSize property value.

Configuration Property	**Value and Description**
`deleteProcessedRecords`	This property, which specifies whether to retain contact and response history records after they are processed, is set to `YES` by default. Changing this value can give you more control over the data flow within the ETL process and affect performance (by delaying the purging of these records until a later time of your determination); however, you must be knowledgeable to handle the maintenance of these records manually to be sure that they are removed at appropriate times. Use caution when modifying this setting.
`fetchSize`	Providing a value for the JDBC `fetchSize` can improve performance for large batches of records, but the tradeoff for improved networking performance is the impact of larger fetch sizes on memory usage. See the description of this configuration property in the *IBM Unica Interact Administrator's Guide* for more information about adjusting this setting.

For detailed descriptions of each of the configuration properties described here, see the online help for that configuration page, or see the *IBM Unica Interact Administrator's Guide.*

Logging

Make sure the log level is set to INFO or ERROR. Never use a verbose log setting like DEBUG or TRACE in a production environment.

There are three places where you can configure logging:

- Set the logging level in the `interact_log4j.properties` file. By default, this file is installed in `<`*`install_dir`*`>/Interact/conf` directory, where <*install_dir*> is the parent directory where your IBM Unica products are installed.
- Confirm that Interact API is not logging. Logging is determined by Boolean setting available in the `startSession` and `setDebug` methods.
- Confirm that JMX monitoring is set to Info with the `activateInfo` JMX operation.

Contacting IBM Unica technical support

If you encounter a problem that you cannot resolve by consulting the documentation, your company's designated support contact can log a call with IBM Unica technical support. Use the information in this section to ensure that your problem is resolved efficiently and successfully.

If you are not a designated support contact at your company, contact your IBM Unica administrator for information.

Information to gather

Before you contact IBM Unica technical support, gather the following information:

- A brief description of the nature of your issue.
- Detailed error messages you see when the issue occurs.
- Detailed steps to reproduce the issue.
- Related log files, session files, configuration files, and data files.
- Information about your product and system environment, which you can obtain as described in "System information."

System information

When you call IBM Unica technical support, you might be asked to provide information about your environment.

If your problem does not prevent you from logging in, much of this information is available on the About page, which provides information about your installed IBM Unica applications.

You can access the About page by selecting **Help > About**. If the About page is not accessible, you can obtain the version number of any IBM Unica application by viewing the `version.txt` file located under the installation directory for each application.

Contact information for IBM Unica technical support

For ways to contact IBM Unica technical support, see the IBM Unica Product Technical Support website: (http://www.unica.com/about/product-technical-support.htm).

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
170 Tracer Lane
Waltham, MA 02451
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Printed in USA