IBM Unica Interact
Version 8.5.0 Publication Date: June 7, 2011

Tuning Guide

Copyright

Table of Contents

Preface

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 you should gather

Before you contact IBM Unica technical support, you should 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" below.

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 each application's installation directory.

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).

1 About tuning Interact

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 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 should 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 may have different results in your environment.

The following guidelines are organized by related components. The order you make changes in does not matter.

Interact API

Use the Java Serialization API instead of SOAP API. The Serialization API provides better throughput (can be 5 to 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.

In IBM Unica Marketing for the Interact runtime server, set the `Interact > sessionManagement > cacheType` property to `local`.

Session management

In IBM Unica Marketing 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 could 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 may need to remain active for a minute, but a web site session should remain active for 10 minutes.

Interactive flowchart management

Every interactive flowchart requires at least one thread to run. You can monitor a live system to see if 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.

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 may 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 two log contact calls, then the `contactHist` threshold should be set to a valued 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

These arguments are defined in the startup command script (BEA 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's 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 may 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 has been found to eliminate Garbage Collection related slowness without sacrificing throughput. The following is one set of options that has been found to be useful for a 2GB 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[tm] 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 one 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.

WebSphere

When using Websphere, use version 6.1.0.17 or higher to avoid performance problems related to acquiring a database connection from connection pools, as documented by IBM (http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg1PK61944).

Tuning terminology

The following tuning guidelines have been 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. These include calls defined by the InteractAPI class, for example `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 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.

Logging

Make sure the log level is set to INFO or ERROR. A verbose log setting like DEBUG or TRACE should never be used 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. This is a Boolean setting available in the `startSession` and `setDebug` methods.
- Confirm that JMX monitoring is set to Info with the `activateInfo` JMX operation.