Tuning the Tomcat Container

Pre-requisites

Caveat: These are recommendations only and it is recommended that customers refer to Tomcat documentation, for the version you're using, and use a performance testing optimisation strategy to optimally configure IG according to their deployment. Only an incremental optimisation strategy will determine the optimal configuration for a given deployment and user load.

See Tomcat documentation:

- Tomcat 8.0, see http://tomcat.apache.org/tomcat-8.0-doc/config/http.html
- Tomcat 9.0, see http://tomcat.apache.org/tomcat-9.0-doc/config/http.html

See IG documentation and blogs:

- Ref: Tuning the Tomcat Container

Tuning the Tomcat Container JVM

See Tuning the JVM.

Tomcat Connectors

In order to configure IG on Tomcat, it is important to note Tomcat version and configured connector type. Below is a list of the Tomcat 8.x and 9.x `Connector` types, with some pertinent qualities of each:

Java Blocking Connector BIO (Tomcat 3.x - 8.x)

- Blocking - uses a single thread per concurrent connection.
- Thread returned to pool after response completes.
- Thread blocked while long-latency processing in progress.
- Connections may be consumed needlessly - e.g. within socket-timeout or until keepalive.
- Connections may be leaked - if client does not close properly, until keepalive timeout.
- Removed in Tomcat 9.x - see docs

Java Nio Connector NIO (Tomcat 6.x+)

- Multiplexes between requests.
- Poller threads poll connections to determine which are in use.
- Worker threads handed request data when received on the connection.
- Poller threads return response data to the client.
- Poller threads check non-busy threads and make them available again - don't wait for keepalive timeout.

Java Nio2 Connector NIO2 (Tomcat 8.x+)

- The key difference seems to be in async handling, where NIO uses polling (of the Future), whereas NIO2 uses callbacks.
- Supports finer buffer control, with NIO2-specific config options:
    - Configure direct- or nondirect-ByteBuffer use, with implications for performance but also stability (garbage-collection)
    - Tune read and write buffer sizes.
- Aligns with Servlet 3.1 API - callback handlers.

- For a comparison of these, see the following tables:
    - connectors available in 8.x
    - connectors available in 9.x - excludes BIO connector

GSA: If there is already comparison tables available, I would just point to them, and add our own added-value stuff (why choosing one versus the others)

WM: There is the above comparison table but it's not very useful really. The above is from researching multiple different sources to understand the differences more fully. I think it's interesting to us and the field, though not for our docs.

(Will remove Tuesday 30/6/2020 if no further comment)

BIO versus NIO

While there are potential performance gains to be had in selecting the NIO Connector over the BIO Connector, it is not certain. It is largely determined by the server usage. The main advantages to be gained are:

- Scalability as the NIO Connector utilises fewer, better utilised threads and so can handle more concurrent requests.

- Where the backend is doing anything IO or latency intensive then we should see improvement with NIO.

GSA: I think we should just describe NIO stuff: IG is built for async processing, that would be a non-sense to advertise blocking behaviours! So, I would move your writing on NIO/NIO2 diff here

WM: Same point really, it's useful to know for us and the field. If they encounter BIO, what does it mean? Not intended for our docs though...

(Will remove Tuesday 30/6/2020 if no further comment)

Summary

To take advantage of IG's asynchronous thread model, it is recommended that the Tomcat container is configured to use an NIO or NIO2 `Connector`. Please refer to Tomcat 8.x or Tomcat 9.x documentation for a comparison.

Connector Configuration

In order to configure IG on Tomcat, the following Tomcat connector configuration options are worthy of consideration:

Option	Description
`maxConnections`	The maximum number of connections that the server will accept and process at any given time: • default is 10,000 for BIO or 8192 for NIO/ NIO2 • For NIO/ NIO2, this value can be set to -1 and connections will not be counted.
`connectionTimeout`	The available time for connecting to a server-side socket, before timing out and abandoning the connection attempt.
`soTimeout`	The socket timeout, after which the request is deemed to have failed.
`acceptCount`	The maximum queue length to queue incoming requests when all threads are busy - default is 100.
`executor`	An `Executor` can be configured to manage the `Connector` thread-pool - rather than configuring thread-based options directly on the `Connector`. This supports the sharing of an `Executor` between `Connector`s. Note that this is specially recommended to provide finer control over server resources, notably threads. Refer to Executor config.
`maxThreads`	The maximum number of request processing threads to be created by this `Connector`, which therefore determines the maximum number of simultaneous requests that can be handled - default is 200. This can also be configured via the executor. GSA: I would just focus on `executor` options (there is a `maxThreads` here as well) WM: There is... The thing is, the `executor` is actually intended to allow multiple `Connector`s to use a single `Executor` (with common config). If a `Connector` does not have this configured then it uses its own config, which in turn actually creates :drumroll: an `Executor` for it to use (privately). Maybe I should make a note of this? GSA: that looks like to me that executor is the Tomcat "supported" way, while maxThreads are still here for backward compatibility *(Will remove Tuesday 30/6/2020 if no further comment)*
`minSpareThreads`	The minimum number of threads that should be available, idle or active, at any given time - default 10. This can also be configured via the executor. GSA: probably part of the executor config right ? WM: It's part of both - plain `Connector` and a `Connector#executor` config. *(Will remove Tuesday 30/6/2020 if no further comment)*

See Tomcat docs for 8.x and 9.x for more information.

Summary

The following Tomcat `Connector` configuration options should be considered:

- `maxConnections`
- `connectionTimeout`
- `soTimeout`
- `acceptCount`
- `executor`
- `maxThreads`
- `minSpareThreads`

Please refer to Tomcat 8.x or Tomcat 9.x documentation for a comparison.

Configuring IG on Tomcat

See Tuning IG and the ClientHandler/ ReverseProxyHandler

References

- https://techblog.bozho.net/tomcats-default-connectors/
- https://stackoverflow.com/questions/25356703/tomcat-connector-architecture-thread-pools-and-async-servlets
- https://www.datadoghq.com/blog/tomcat-architecture-and-performance/
- http://remm.blogspot.com/2014/04/nio-2-in-apache-tomcat-8.html

Related articles

- Tuning IG and the ClientHandler/ ReverseProxyHandler
- Tuning the JVM
- Tuning the Tomcat Container