Item 206
git_comments:

1. ----------
2. * * Private default constructor to avoid instantiation.
3. ---------- Simple reading and writing of files ----------
4. ----------
5. ---------- Silent I/O cleanup / closing ----------
6. * The block size for byte operations in byte.
7. ---------- Byte copy operations ---------- ---------- Stream input skipping ----------
8. * * Private constructor to prevent instantiation.
9. * The default charset to convert strings to bytes
10. * The default charset to convert strings to bytes
11. ----------
12. * * Private default constructor to avoid instantiation.
13. ---------- Simple reading and writing of files ----------
14. ----------
15. ---------- Silent I/O cleanup / closing ----------
16. * The block size for byte operations in byte.
17. ---------- Byte copy operations ---------- ---------- Stream input skipping ----------
18. * * Private constructor to prevent instantiation.
19. now, find the sequence of zeros that should be compressed
20. first, convert bytes to 16 bit chunks
21. ---------- Port range parsing ----------
22. convert into text form
23. * * Creates a compressed URL style representation of an Inet6Address. * * <p>This method copies and adopts code from Google's Guava library. * We re-implement this here in order to reduce dependency on Guava. * The Guava library has frequently caused dependency conflicts in the past.
24. * * Checks the given boolean condition, and throws an {@code IllegalStateException} if * the condition is not met (evaluates to {@code false}). * * @param condition The condition to check * * @throws IllegalStateException Thrown, if the condition is violated.
25. * * Checks the given boolean condition, and throws an {@code IllegalStateException} if * the condition is not met (evaluates to {@code false}). * * @param condition The condition to check * @param errorMessageTemplate The message template for the {@code IllegalStateException} * that is thrown if the check fails. The template substitutes its * {@code %s} placeholders with the error message arguments. * @param errorMessageArgs The arguments for the error message, to be inserted into the * message template for the {@code %s} placeholders. * * @throws IllegalStateException Thrown, if the condition is violated.
26. Boolean Condition Checking (Argument) ---------- Boolean Condition Checking (State) ----------
27. * * Checks the given boolean condition, and throws an {@code IllegalStateException} if * the condition is not met (evaluates to {@code false}). The exception will have the * given error message. * * @param condition The condition to check * @param errorMessage The message for the {@code IllegalStateException} that is thrown if the check fails. * * @throws IllegalStateException Thrown, if the condition is violated.
28. ----------

git_commits:

1. **summary:** [FLINK-3700] [core] Remove Guava dependency from flink-core

message: [FLINK-3700] [core] Remove Guava dependency from flink-core Almost all Guava functionality used within flink-core has corresponding utils in Flink's codebase, or the JDK library. This replaces the Guava code as follows - Preconditions calls by Flink's Preconditions class - Collection utils by simple Java Collection calls - Iterator's by Flink's Union Iterator - Files by simple util methods arount java.nio.Files - InetAddresses IPv6 encoding code has been adapted into Flink's NetUtils (with attribution comments) Some util classes where moved from flink-runtime to flink-core. This closes #1854

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [FLINK-3700] [core] Remove Guava as a dependency from "flink-core"
   body: **Note: This builds on top of pull request #1853** Almost all Guava functionality used within `flink-core` has corresponding utils in Flink's codebase, or the JDK library. This replaces the Guava code as follows - Preconditions calls by Flink's Preconditions class - Collection utils by simple Java Collection calls - Iterator's by Flink's Union Iterator - Murmur Hasher calls by Flink's `MathUtil.murmurHash()` - Files by simple util methods around `java.nio.Files` - InetAddresses IPv6 encoding code has been adapted into Flink's NetUtils (with attribution comments) - `Joiner` by `org.apache.commons.lang3.StringUtils`, which we reference anyways, and which is a well-behaved dependency (might even get rid of this in the future). Some util classes where moved from `flink-runtime` to `flink-core`.

github_pulls_comments:

1. +1 to merge
2. Thanks for the refactoring @StephanEwen! Will merge this.

github_pulls_reviews:

jira_issues:

1. **summary:** Replace Guava Preconditions class with Flink Preconditions
   description: In order to reduce the dependency on Guava (which has cause us quite a bit of pain in the past with its version conflicts), I suggest to add a Flink {{Preconditions}} class.

jira_issues_comments:

1. Could we not just copy the portion of Guava that we use to the Flink codebase then? (It is apache licensed) This would be another portion of code that we maintain and test while there are already several packages out there that provide the functionality.
2. GitHub user StephanEwen opened a pull request: https://github.com/apache/flink/pull/1853 [FLINK-3700] [core] Add 'Preconditions' utility class. The functionality that Flink uses from Guava is super simple and limited. We get a big dependency that has caused a lot of pain in the past, simply to get access to some simple utility methods. This has cause us quite a bit of pain in the past, because of Guava version conflicts and the necessary dependency shading. In order to reduce the dependency on Guava, this adds a simple Flink Preconditions class. While referencing well established libraries is a good idea for standalone apps, it is a problem for frameworks like Flink, if the dependencies are not well-behaved with respect to version compatibility. Guava is not well behaved in that sense. You can merge this pull request into a Git repository by running: $ git pull https://github.com/StephanEwen/incubator-flink flink_preconditions Alternatively you can review and apply these changes as the patch at: https://github.com/apache/flink/pull/1853.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #1853 ---- ----
3. GitHub user StephanEwen opened a pull request: https://github.com/apache/flink/pull/1854 [FLINK-3700] [core] Remove Guava as a dependency from "flink-core" **Note: This builds on top of pull request #1853** Almost all Guava functionality used within `flink-core` has corresponding utils in Flink's codebase, or the JDK library. This replaces the Guava code as follows - Preconditions calls by Flink's Preconditions class - Collection utils by simple Java Collection calls - Iterator's by Flink's Union Iterator - Murmur Hasher calls by Flink's `MathUtil.murmurHash()` - Files by simple util methods around `java.nio.Files` - InetAddresses IPv6 encoding code has been adapted into Flink's NetUtils (with

attribution comments) Some util classes where moved from `flink-runtime` to `flink-core`. You can merge this pull request into a Git repository by running: $ git pull https://github.com/StephanEwen/incubator-flink guava_free_core Alternatively you can review and apply these changes as the patch at: https://github.com/apache/flink/pull/1854.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #1854 ---- commit 535e83a297d39315d9d4b36672444b44f409c81e Author: Stephan Ewen <sewen@apache.org> Date: 2016-04-05T10:37:33Z [FLINK-3700] [build] Add 'findbugs' (javax.annotation) annotations as a core dependency. commit eac0813d9f7298211bf52aec61ce346a133436f8 Author: Stephan Ewen <sewen@apache.org> Date: 2016-04-05T11:23:14Z [FLINK-3700] [core] Add 'Preconditions' utility class. commit 052db9286ad8aa8df3e122c300361dbc384a0190 Author: Stephan Ewen <sewen@apache.org> Date: 2016-04-05T13:18:32Z [FLINK-3700] [core] Removes Guava Dependency from flink-core Almost all Guava functionality used within flink-core has corresponding utils in Flink's codebase, or the JDK library. This replaces the Guava code as follows - Preconditions calls by Flink's Preconditions class - Collection utils by simple Java Collection calls - Iterator's by Flink's Union Iterator - Files by simple util methods arount java.nio.Files - InetAddresses IPv6 encoding code has been adapted into Flink's NetUtils (with attribution comments) Some util classes where moved from flink-runtime to flink-core. ----

4. Github user tillrohrmann commented on the pull request: https://github.com/apache/flink/pull/1853#issuecomment-205811166 +1 for merging :-)
5. I agree for any complex code. The code that was actually used by Guava (at least in flink-core) were all very simple convenience functions that we basically have already, or are simple to replicate (like the {{Preconditions}} class).
6. Github user uce commented on the pull request: https://github.com/apache/flink/pull/1853#issuecomment-206299884 +1. I know of some other places where I've used Guava's Hasher or Cache, which we can also remove in order to be able to remove our Guava dependency.
7. Github user uce commented on the pull request: https://github.com/apache/flink/pull/1854#issuecomment-208308210 +1 to merge
8. Github user StephanEwen commented on the pull request: https://github.com/apache/flink/pull/1853#issuecomment-208312455 Merging this...
9. Github user ZackPierce commented on a diff in the pull request: https://github.com/apache/flink/pull/1853#discussion_r59289246 --- Diff: flink-core/src/main/java/org/apache/flink/util/Preconditions.java --- @@ -0,0 +1,213 @@ +/* + * Licensed to the Apache Software Foundation (ASF) under one + * or more contributor license agreements. See the NOTICE file + * distributed with this work for additional information + * regarding copyright ownership. The ASF licenses this file + * to you under the Apache License, Version 2.0 (the + * "License"); you may not use this file except in compliance + * with the License. You may obtain a copy of the License at + * + * http://www.apache.org/licenses/LICENSE-2.0 + * + * Unless required by applicable law or agreed to in writing, software + * distributed under the License is distributed on an "AS IS" BASIS, + * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. + * See the License for the specific language governing permissions and + * limitations under the License. + */ + +// -- +// This class is largely adapted from "com.google.common.base.Preconditions", +// which is part of the "Guava" library. +// +// Because of frequent issues with dependency conflicts, this class was +// added to the Flink code base to reduce dependency on Guava. +// -- + +package org.apache.flink.util; + +import org.apache.flink.annotation.Internal; + +import javax.annotation.Nullable; + +/** + * A collection of static utility methods to validate input. + * + * <p>This class is modelled after Google Guava's Preconditions class, and partly takes code + * from that class. We add this code to the Flink code base in order to reduce external + * dependencies. + */ +@Internal +public final class Preconditions { + + // -- + // Null checks + // -- + + /** + * Ensures that the given object reference is not null. + * Upon violation, a {@code NullPointerException} with no message is thrown. + * + * @param reference The object reference + * @return The object reference itself (generically typed). + * + * @throws NullPointerException Thrown, if the passed reference was null. + */ + public static <T> T checkNotNull(T reference) { --- End diff -- For what it is worth, `Objects.requireNonNull` has overloads with the same signature (as this and the next method down), and is [present in the JDK since 1.7](https://docs.oracle.com/javase/7/docs/api/java/util/Objects.html#requireNonNull(T)). Using `requireNonNull` instead may help reduce the burden of semi-standard code that Flink reimplements.

10. Fixed via 885d543be8a8c0d1acdffa0003e394347d5376ef
11. Github user StephanEwen commented on a diff in the pull request: https://github.com/apache/flink/pull/1853#discussion_r59518788 --- Diff: flink-core/src/main/java/org/apache/flink/util/Preconditions.java --- @@ -0,0 +1,213 @@ +/* + * Licensed to the Apache Software Foundation (ASF) under one + * or more contributor license agreements. See the NOTICE file + * distributed with this work for additional information + * regarding copyright ownership. The ASF licenses this file + * to you under the Apache License, Version 2.0 (the + * "License"); you may not use this file except in compliance + * with the License. You may obtain a copy of the License at + * + * http://www.apache.org/licenses/LICENSE-2.0 + * + * Unless required by applicable law or agreed to in writing, software + * distributed under the License is distributed on an "AS IS" BASIS, + * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. + * See the License for the specific language governing permissions and + * limitations under the License. + */ + +// -- +// This class is largely adapted from "com.google.common.base.Preconditions", +// which is part of the "Guava" library. +// +// Because of frequent issues with dependency conflicts, this class was +// added to the Flink code base to reduce dependency on Guava. +// -- + +package org.apache.flink.util; + +import org.apache.flink.annotation.Internal; + +import javax.annotation.Nullable; + +/** + * A collection of static utility methods to validate input. + * + * <p>This class is modelled after Google Guava's Preconditions class, and partly takes code + * from that class. We add this code to the Flink code base in order to reduce external + * dependencies. + */ +@Internal +public final class Preconditions { + + // -- + // Null checks + // -- + + /** + * Ensures that the given object reference is not null. + * Upon violation, a {@code NullPointerException} with no message is thrown. + * + * @param reference The object reference + * @return The object reference itself (generically typed). + * + * @throws NullPointerException Thrown, if the passed reference was null. + */ + public static <T> T checkNotNull(T reference) { --- End diff -- True, I have been using that quite a bit. There are, however, a few signatures missing (like message with placeholders and var-arg parameters) as well as separate functions like `checkArgument` and `checkState`.
12. Github user StephanEwen commented on the pull request: https://github.com/apache/flink/pull/1853#issuecomment-209507866 Manually merged in 885d543be8a8c0d1acdffa0003e394347d5376ef
13. Github user StephanEwen closed the pull request at: https://github.com/apache/flink/pull/1853
14. Github user fhueske commented on the pull request: https://github.com/apache/flink/pull/1854#issuecomment-210398946 Thanks for the refactoring @StephanEwen! Will merge this.
15. Github user asfgit closed the pull request at: https://github.com/apache/flink/pull/1854
16. Commit 760a0d9e7fd9fa88e9f7408b137d78df384d764f removed Guava as dependency from {{flink-core}}.