

git_comments:

1. * * A timer for a specified time domain that can be set to register the desire for further processing * at particular time in its specified time domain. * * <p>See {@link TimeDomain} for details on the time domains available. * * <p>In a {@link DoFn}, a {@link Timer} is specified by a {@link TimerSpec} annotated with {@link * DoFn.TimerId}. * * <p>An implementation of {@link Timer} is implicitly scoped - it may be scoped to a key and * window, or a key, window, and trigger, etc. * * <p>A timer exists in one of two states: set or unset. A timer can be set only for a single time * per scope. * * <p>Timer callbacks are not guaranteed to be called immediately according to the local view of the * {@link TimeDomain}, but will be called at some time after the requested time, in timestamp * order.
2. * * Sets or resets the time relative to the current time in the timer's {@link TimeDomain} at which * this it should fire. If the timer was already set, resets it to the new requested time.
3. * * Unsets this timer. It is permitted to {@code cancel()} whether or not the timer was actually * set.
4. * 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.
5. * * A specification for a {@link Timer}. This includes its {@link TimeDomain}.
6. * 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.
7. * * A straightforward POJO {@link TimerSpec}. Package-level access for AutoValue.
8. * * Static methods for working with {@link TimerSpec}.
9. * 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.
10. * The timer ID.
11. * * Annotation for registering a callback for a timer. * * <p><i>Not currently supported by any runner. When ready, the feature will work as described * here.</i> * * <p>See the javadoc for {@link TimerId} for use in a full example. * * <p>The method annotated with {@code @OnTimer} may have parameters according to the same logic * as {@link ProcessElement}, but limited to the {@link BoundedWindow}, {@link State} subclasses, * and {@link Timer}. State and timer parameters must be annotated with their {@link StateId} and * {@link TimerId} respectively.
12. * * Annotation for declaring and dereferencing timers. * * <p><i>Not currently supported by any runner. When ready, the feature will work as described * here.</i> * * <p>To declare a timer, create a field of type {@link TimerSpec} annotated with a {@link * TimerId}. To use the cell during processing, add a parameter of the type {@link Timer} to your * {@link ProcessElement @ProcessElement} or {@link OnTimer @OnTimer} method, and annotate it with * {@link TimerId}. See the following code for an example: * * <pre>{@code * new DoFn<KV<Key, Foo>, Baz>() { * @TimerId("my-timer-id") * private final TimerSpec myTimer = TimerSpecs.timerForDomain(TimeDomain.EVENT_TIME); * * @ProcessElement * public void processElement(* ProcessContext c, * @TimerId("my-timer-id") Timer myTimer) { * myTimer.setForNowPlus(Duration.standardSeconds(...)); * } * * @OnTimer("my-timer-id") * public void onMyTimer() { * ... * } * } }</pre> * * <p>Timers are subject to the following

validity conditions: * * * Each timer must have a distinct id. * Any timer referenced in a parameter must be declared. * Timer declarations must be final. * All declared timers must have a corresponding callback annotated with {@link * OnTimer @OnTimer}. *

13. Timers are semantically incompatible with splitting
14. To be removed when the features are complete and runners have their own adequate rejection logic
15. * Describes a {@link DoFn.OnTimer} method.
16. * Types of optional parameters of the annotated method, in the order they appear.
17. * Timer declarations present on the {@link DoFn} class. Immutable.
18. * The id on the method's {@link DoFn.TimerId} annotation.
19. * Details about this {@link DoFn}'s {@link DoFn.OnTimer} methods.
20. * The annotated method itself.
21. * * Describes a timer declaration; a field of type {@link TimerSpec} annotated with * {@link DoFn.TimerId}.
22. Check the converse - that all timers have a callback. This could be relaxed to only those timers used in methods, once method parameter lists support timers.
23. * * Returns successfully if the field is valid, otherwise throws an exception via * its {@link ErrorReporter} parameter describing validation failures for the * timer declaration.
24. Find the state and timer declarations in advance of validating method parameter lists
25. Test classes at the bottom of the file
26. The method we pull out is the superclass method; this is what allows validation to remain simple. The later invokeDynamic instruction causes it to invoke the actual implementation.
27. * * In this particular test, the super class annotated both the timer and the callback, * and the subclass overrides an abstract method. This is allowed.

git_commits:

1. **summary:** This closes #1102
message: This closes #1102

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [BEAM-27] Add initial bits for DoFn Timer API
body: Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `https://www.apache.org/licenses/icla.txt). --- R: @jkff CC: @bjchambers who gave lots of good feedback on the `StateSpec` version of this PR. Time permitting, see if you have opinions about this simpler one. This adds the interfaces and annotations from the design for the user-facing timer API, with basic reflection and validation and some tests. (while putting this together, I noticed that there were state examples that were invalid in ways other than the way they were testing, so that commit is in this PR too)

github_pulls_comments:

1. There are no syntactic conflicts with base, but this needs a rebase for Jenkins to be able to pass. Going to go ahead and do it now, since it seems to be a review lull.
2. Jenkins failure is a failure to connect to local JDBC, not related to this PR.
3. Thanks a ton @jkff and @bjchambers for all the feedback. I'll squash fixups and merge.

github_pulls_reviews:

1. One thing not addressed in the design or this PR is how to come up with this `Instant`. For processing time, presumably it is easy, aka `Instant.now().plus(...)`. For event time, I think we'll need to plumb the current event time in. I actually think it is better for processing time to be accessed through a parameter as well, making it fully abstract. I am hinting at something like the `currentXYZTime()` methods from

- `Timers` (note the plural) being available as an extra parameter to `DoFn`. But hopefully in the least obtrusive way possible. It is somewhat sane to have the "current time according to the time domain that this timer is for" on the interface for `Timer` (not plural) as well.
2. This is out of sync with the code below.
 3. Fixed.
 4. One option would be to require setting a timer at some point relative now (similar to what we do in triggers). So: `myTimer.setForNowPlus(Duration.standardSeconds(30));` // For more general access, we could just have a library of modifier functions: `myTimer.setForNow(plus(Duration.standardSeconds(30)), alignedTo(Duration.standardMinutes(1)));`
 5. `@{@code TimerId}`
 6. Also we'll need an `@OnTimer` method with the same ID, correct?
 7. `thrown.expectMessage("bizzle");` so that we verify we also point at the problem?
 8. +1 to @bjchambers' suggestion - relative timers make sense, like `SplittableDoFn's ProcessContinuation.withResumeDelay()`.
 9. Add a note that the API for actually reacting to timers does not exist yet?
 10. Is the reverse true?
 11. Indentation is off
 12. This seems to duplicate the condition above, and also not test that it's splittable.
 13. This condition doesn't test that it's splittable :-/
 14. Is there a guarantee that timers in a particular time domain are delivered in order of their specified firing time?
 15. timestamp
 16. Say that it's specified by `TimerSpec`. Also say that a timer with a given id is either set or unset, but can't be scheduled for several instants at the same time (this makes `cancel()` possible).
 17. Also clarify whether it's legal to call `cancel()` when the timer has not been set, or has already been cancelled.
 18. What is an "instantiated" timer?
 19. This is a good point. Only relative timers are really useful. I'll start with that and then we can consider that library of modifier functions, which we basically already have in triggers.
 20. Rephrased. I want to keep the distinction between the annotation and the identifier.
 21. Nope. Declared and unused timers are useless, but no need to fail them. It would probably cause needless code churn if a user gradually added and removed timer usage while debugging or evolving.
 22. Yes. Added.
 23. Instead, I decided to add that annotation and its parsing to this PR.
 24. Done.
 25. Done.
 26. The condition is duplicate, but the error message is not. Nesting conditionals is slightly less readable IMO, and the particular set of things checked in each may evolve.
 27. Done.
 28. I think this is useful and implementable for the scope we currently talk about (per key [and window]). There are conceivable designs where timers are not adequately centralized for that, but I don't think any that we are considering. Added this requirement.
 29. Done.
 30. I just went ahead and shrunk it to the basics.
 31. Done.
 32. For testing I would suggest/prefer a `LinkedHashMap` be created here, to ensure deterministic ordering.
 33. I assume you're planning on validating the callbacks (and existence of them) in a future PR. If not, it should be done here.
 34. Update Javadoc.
 35. If we're adding, this should take a duration not an instant.
 36. interfaces should not have `public abstract` -- both of those are implied
 37. I prefer to make anything depending on order explode. Is there something in particular you have in mind? I could imagine it being useful to be able to give error message about code in the order the user wrote it in the file, but that's a different argument that doesn't apply to today's logic.
 38. This PR. I mentioned in my top level comment about it not being ready just yet, but I've just pushed up rudimentary validation.
 39. Done.
 40. Done.

41. Explicit is better than implicit, and context-free code is better than context-sensitive code, but done because it is standard to do so.
42. `... annotated with `@{@link OnTimer}``
43. I wonder if we should add a `DoFnField`` class to parallel `DoFnMethod``. Also, for parallelism the `field`` should be the first argument?
44. should this mention that it can only be set for a single time within a given key and window?
45. nit: I don't know this needs to be called out, but if so I'd suggest "immediately" rather than "synchronously" -- when setting the timer we've already finished our code, so it isn't clear what it would be synchronous with.
46. I'd move this up, and then say "A timer can only be set for a single time per scope."
47. Can this paragraph be merged with the "timers are not guaranteed to immediately fire" (seems to be two common themes here -- how to set and scope timers and then what guarantees there are for timer delivery rather than 4 separate paragraphs).
48. Should also have a test where the `StateSpec / TimerSpec` are defined in a parent class. As a follow-up, should we require they be protected to access them in a child class? Technically we don't care, since they're accessed via the `@TimerId`` annotation and reflection, but it may be good to have similar scoping?
49. My stylistic preference would be to disallow this. I think the declaration and use should be in the same lexical scope. I wonder if there are important use cases here?
50. Done
51. TBH I'm not really sure what the goal of `DoFnMethod`` is. Restored parallel structure (with `StateDeclaration`` too) by moving the id first in `OnTimerMethod``.
52. Done (see code block above at line ~128)
53. Done.
54. Well, "immediately" just doesn't have a meaning, because there is no such thing as absolute time (in life, but in particular in this sort of system). So "synchronously" indicates that it does not necessarily fire as soon as "someone somewhere decides that time has arrived". I've tried to come up with a more down-to-earth way of describing it.
55. Done.
56. Done.
57. I did not actually change the code to reject this situation.
58. Just a thought: do we actually need the field? We already repeat the timer name `"my-timer-id"` twice - in the `ProcessElement`` parameter and in the `onMyTimer`` callback, why have it a third time? And we have both the field and the parameter. Can we have something like: `void processElement(..., @Timer("my-timer-id", EVENT_TIME) Timer myTimer)`` or even `myTimer.setForNowPlus(duration, EVENT_TIME)``?
59. Do we want to be able to set timers from `StartBundle`` or `FinishBundle``?
60. Join these two into one big `if (isSplittable)``?
61. Nit: probably better to have state-related things first, and timer-related things second ("state and timers" are part of the same proposal, but I wouldn't say that they are so tightly coupled conceptually that they have to always go together).
62. Wait, don't they already support them? According to the javadoc, `ProcessElement`` does.
63. Indentation is off
64. May be worth putting the body of this loop in a method for potential easier testing.
65. Doesn't `field.toString()` already print the type of the field?
66. relative to the current time in the requested `TimeDomain``?
67. Actually, upon reading the rest of the PR, seems like this syntax is not supported, and we only have the field.
68. The error messages are different. Nesting the secondary condition within `if (isSplittable)`` increases the cyclomatic complexity for a reader without saving anything. Did I understand what you mean?
69. Agree. Done.
70. Thanks for catching it. Something has changed in one of my IntelliJ plugin updates...
71. I'd like to submit the annotations in pieces and enable them later, to keep code reviews ~500 LOC or certainly <1000. What I started with was javadoc only for the annotations that were currently allowed to be written, which naturally bothered reviewers since it was necessarily dead code. So I fleshed it out to describe the full API, which perhaps was a mistake. The alternative I see is to make it very terse and clearly unsupported. Would you prefer that? I am quite happy to do that until they can all be written down together. (one PR from now, in progress)
72. Done.

73. Yea, and actually it is fairly unreadable. I'll work refining the error messages, but certainly remove this redundancy.
74. Done.
75. This comment seems still out of sync with the DoFnSignatures code - DoFnSignatures doesn't look for timers in parameters of `ProcessElement`, only in fields.
76. Yeah, I'd prefer if the javadoc on what syntax is allowed matched what syntax is actually understood by DoFnSignatures.
77. `AFAIR ErrorReporter.throwIllegalArgument` actually throws an exception, so this method will never return false in practice.
78. This is unclear - it reads as if you can set a timer for a time in a different domain than the one the timer belongs to.
79. I think it is OK to document how the annotation will be used and add the functionality in a future PR. We'll revisit this when removing the "Not currently supported by any runner" statement.
80. "." (all bullets should end with "." or none of the should)
81. I think it was to allow treating all methods similarly (as a thing that has the underlying method + some stuff). Fine to not have a `DoFnField`, but keep it in mind in case it becomes useful.
82. I would say it is OK for the javadoc on the annotations to explain what we're going to support, and for the tests to demonstrate what is currently supported. I wouldn't want us to iterate a lot on Javadoc that is going to be thrown away.
83. You answered in your next comment. I will wipe out this javadoc and save it for later, pending discussion with @bjchambers that he is also fine with that.
84. Consolidating these threads - see below.
85. I talked about this with @bjchambers and @dhalperi and we came to a consensus that it is useful for reviewers to have some documentation - could just be Java comments - and mostly harmless at this stage in Beam to have some Javadoc that describes a prospective feature, as long as it has the needed caveats. So rather than convert it to a plain comment, I added caveats.
86. Oh, I had assumed the point of the `ErrorReporter` was to gather errors. Without that, it seems like just a needless parameter. And upon inspection, why does it exist?
87. Revised.
88. Now I have done so.
89. `ErrorReporter` exists in order to provide additional context to the exception being thrown. The context is accumulated as you nest `ErrorReporter`'s - typically it'll contain the DoFn class name and sometimes the method name being analyzed.
90. Got it. Changed it to `void`. It isn't ideal, because we really could gather errors, but that is a larger change.
91. `OnTimer` methods also support `Timer` arguments, right?
92. At line 409 - `OnTimer` methods will also support state.
93. Done.
94. Done.
95. Done.
96. Acknowledged.

jira_issues:

1. **summary:** Add user-ready API for interacting with timers
description: Pipeline authors will benefit from a different factorization of interaction with underlying timers. The current APIs are targeted at runner implementers.

jira_issues_comments:

1. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1102> [BEAM-27] Add initial bits for DoFn Timer API Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `<Jira issue #>` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement](<https://www.apache.org/licenses/icla.txt>). --- R: @jkff CC: @bjchambers who gave lots of good feedback on the `StateSpec` version of this PR. Time permitting, see if you have opinions about this simpler one. This adds the interfaces and annotations from the design for

the user-facing timer API, with basic reflection and validation and some tests. (while putting this together, I noticed that there were state examples that were invalid in ways other than the way they were testing, so that commit is in this PR too) You can merge this pull request into a Git repository by running: `$ git pull https://github.com/kennknowles/incubator-beam TimerSpec` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1102.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 #1102 ---- commit 238b7eaa9425247ce7152b268d488581032e5411 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T06:18:40Z Tighten up tests for StateId validation commit b1aa85d764c70ce648e4d5c068ce1b7f1e36a2ac Author: Kenneth Knowles <klk@google.com> Date: 2016-10-13T20:41:38Z Add Timer, TimerId, TimerSpec and TimerSpecs These are the simple specifications that describe a timer without instantiating it. Consists only of the time domain. commit 0cd37b50ee0d92886060fe9cb8e28273e24e0c13 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T02:53:31Z Add DoFnSignatures analysis of timer declarations commit fe2cf1688068ac0731b67724c1a8f591f03ab159 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T06:17:36Z Reject all timers in ParDo, for now ----

2. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1130> [BEAM-27] Add parsing and validation of @StateId params to @ProcessElement Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `https://www.apache.org/licenses/icla.txt). --- R: @bjchambers @jkff This is an initial draft to discuss some design decisions along the way. You can merge this pull request into a Git repository by running: `$ git pull https://github.com/kennknowles/incubator-beam State-params` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1130.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 #1130 ---- commit 33c884118ed04c0a008ef6018a06fa336a866de8 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T17:51:57Z Use the correct State class in DoFnSignatures commit 81b6994bf8162bfc3aaba40a708ce8385be76d80 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T18:07:55Z Add analysis and validation of State parameters to DoFn.ProcessElement -- --

3. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1136> [BEAM-27] Add analysis of Timer parameters for DoFn.ProcessElement Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `https://www.apache.org/licenses/icla.txt). --- This PR adds `Timer` parameters. It includes #1102 and #1130. The final commit is the only one that is new, and can be reviewed independently. R: @jkff I will pull in a committer later. You can merge this pull request into a Git repository by running: `$ git pull https://github.com/kennknowles/incubator-beam Timer-params` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1136.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 #1136 ---- commit 507cba8f0df44b1c11c7621f00ff492243bdd1f1 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T06:18:40Z Tighten up tests for StateId validation commit 08fdb304a777d981fa3860fcd0270aa40f99da75 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T21:19:59Z Fix condition on state/splittable ParDo validation commit 65eadebfb628f6eb42d17c16220e973b064608d7 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-13T20:41:38Z Add Timer, TimerId, TimerSpec and TimerSpecs These are the simple specifications that describe a timer without instantiating it. Consists only of the time domain. commit 0898999f34270a11beddb68e5068c5258ec9ebec Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T02:53:31Z Add DoFnSignatures analysis of timer declarations commit e44e56da62e48f1d763e2b2d46d55a04fbe3596c Author: Kenneth Knowles <klk@google.com> Date: 2016-10-18T20:09:57Z Reject all timers in ParDo, for now commit bcb07b534ffff475637d06a14695f38b1e151912 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-18T21:15:34Z fixup! More caveats on Timer javadoc commit 97aaf9730915d426fd865eeb67793988a8dbdefb Author: Kenneth Knowles <klk@google.com> Date:

- 2016-10-18T21:16:09Z fixup! state javadoc commit ac419d245a7bd3dc9263e61567676e3d92133f9a Author: Kenneth Knowles <klk@google.com> Date: 2016-10-18T21:21:41Z fixup! refine Timer javadoc commit 28fbe1890b54e26f7c0d0bc547440af061d1e907 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-18T22:52:28Z fixup! validate is now void commit dcce856a1fa3d8bde6b47cf200b4444d212de9d2 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T17:51:57Z Use the correct State class in DoFnSignatures commit 70ec03e5cadc272b194589ab3d7ce89e39be7b13 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-14T18:07:55Z Add analysis and validation of State parameters to DoFn.ProcessElement commit 5883e0c087f1b5642f9349ab099f2eaa82caf8fa Author: Kenneth Knowles <klk@google.com> Date: 2016-10-19T22:00:30Z Merge remote-tracking branch 'github/pr/1102' into Timer-params commit bda251e05ffe48873ccaf06f005c3847acd6085b Author: Kenneth Knowles <klk@google.com> Date: 2016-10-19T22:02:36Z Add analysis of Timer parameters to DoFn.ProcessElement ----
4. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1102>
 5. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1136>
 6. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1217>
 [BEAM-27] Add setTimer and deleteTimer by ID in TimerInternals Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-27] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement] (<https://www.apache.org/licenses/icla.txt>). --- I have pulled this commit out of my ongoing work to add timers to `DoFn`, specifically my branch where I add `DoFnInvoker.invokeOnTimer(<id>, ...)` . For users, timers will have IDs. This is a step towards that API. I think it is also just better in most ways, especially for being able to delete timers without having to know exactly when it was set for. Before this change, a timer's ID is its `TimeDomain` + `Instant`. Inlining `TimerData` for this description, I think `setTimer(TimeDomain, Instant)` is OK for setting an anonymous "wake up" as we use it today, while `deleteTimer(TimeDomain, Instant)` should be removed. R: @tgroh and @aljoscha and @amitsela I am curious about runner-specific thoughts if you have any. You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> TimerInternals Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1217.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 #1217 ---- commit 932cb20e64199942369376748faabae548f72df4 Author: Kenneth Knowles <klk@google.com> Date: 2016-10-25T19:28:14Z Add setTimer and deleteTimer by ID in TimerInternals For users, timers will have IDs. This is a step towards that API; our current API treats the timer's timestamp as its ID, more-or-less. We can leave that API or not. This change adds the more general API. ----
 7. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1217>
 8. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1252>
 [BEAM-27] Factor of parameter analysis in DoFnSignatures; use it for @OnTimer methods Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-27] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement] (<https://www.apache.org/licenses/icla.txt>). --- R: @jkff This might be easiest to review as individual commits, though the last one is the goal of this PR. This PR is a prefix of ongoing work to invoke `@OnTimer` methods. I've peeled it off and fixed it up and added a smidge of testing. Since the code is entirely shared, just making sure the right path is taken provides moderate coverage. (a further refactor could make this clearer, by isolating the shared code into a class with its own unit tests, absolving its callers of proving they exercise the code path, but I thought this baby step was better) You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> DoFnSignatures Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1252.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 #1252 ---- commit 5946ec175d55f9164f9b1b34a06539da3c24f5ae Author: Kenneth Knowles <klk@google.com> Date: 2016-11-01T04:30:40Z Refactor and reuse parameter analysis in DoFnSignatures commit 15a8901ade07c1649378a0b72238eb514877c61c Author: Kenneth Knowles <klk@google.com> Date: 2016-11-01T21:48:54Z Switch DoFnSignature, etc, from TokenType to TypeDescriptor commit

6de216695c05e4d9b283f0842aed8713075be7aa Author: Kenneth Knowles <klk@google.com> Date: 2016-11-01T21:50:24Z DoFnSignature: Make TypeDescriptor-returning methods public commit 40bbfff3dba803951607ae15d5d2cf99fbe64a32 Author: Kenneth Knowles <klk@google.com> Date: 2016-11-01T22:38:01Z Add TypeDescriptor#getTypes ----

9. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1252>
10. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1256>
11. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1307>
[BEAM-27] Connect generated DoFnInvoker's invokerOnTimer to generated OnTimerInvoker Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `<Jira issue #>` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement] (<https://www.apache.org/licenses/icla.txt>). --- You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> DoFnInvokers-OnTimerInvokers Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1307.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 #1307 ---- commit 3e759d3be013fe38cea2601fe4f7cfe95b41d42b Author: Kenneth Knowles <klk@google.com> Date: 2016-11-02T04:23:48Z Connect generated DoFnInvoker.invokerOnTimer to OnTimerInvoker ----
12. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1307>
13. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1477>
[BEAM-27] Add timerId to TimerData Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `<Jira issue #>` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement] (<https://www.apache.org/licenses/icla.txt>). --- This `timerId` is generated to be identical to historical behavior, and to be unique per time domain and timestamp. You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> timerId Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1477.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 #1477 ---- commit d53275d3075d1491d8718837b00093be09f93354 Author: Kenneth Knowles <klk@google.com> Date: 2016-11-23T22:30:57Z Add timerId to TimerData This timerId is generated to be identical to historical behavior, and to be unique per time domain and timestamp. ----
14. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1477>
15. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1528>
[BEAM-27] Add DoFn.OnTimerContext and support as a DoFn parameter Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `<Jira issue #>` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement] (<https://www.apache.org/licenses/icla.txt>). --- You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> OnTimerContext Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1528.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 #1528 ---- commit 44a7b915d502c318ffabaa6fb808207bb3ea15e8 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:10:06Z Add DoFn.OnTimerContext commit 1934b704411fed76a58cbc657c7dc8be666c3885 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:01:18Z Add support for OnTimerContext parameter in DoFnSignature commit 14d9a3794c4f8c68626075343a1dec1d4c017686 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:10:21Z Access to OnTimerContext via DoFnInvokers.ArgumentProvider ----
16. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1528>
17. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1550>
[BEAM-27] Reject timers for ParDo in each runner separately and exclude timer tests Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is

- formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `https://www.apache.org/licenses/icla.txt). --- There are many trivial commits spread across the runners, in two phases. 1. Add a JUnit category for timers and exclude it from all runners. I've included one trivial test just to see that it is working. (FWIW I have hacked the direct runner enough in later commits to get this test to pass "for the wrong reasons") 2. Remove the timer rejection code from `ParDo` and add it to every runner. R: @tgroh @aljoscha @amitsela @twiese I'll wait for all four to LGTM. You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> UsesTimers Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1550.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 #1550 ---- commit 5c99d2c3a573c3963fd6efa9e11ed860d26f0365 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:49:15Z Add JUnit category UsesTimersInParDo With this, various runners can disable tests for this capability until they support it. commit 604695f1b7cbd6dfeda94d82532fde535b3e3448 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:52:24Z Disable tests for timers in ParDo for Apex runner commit cccc58cac2f4960044e646249eb8a3db27b345e1 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:52:49Z Disable tests for timers in ParDo for Flink runner commit f27bdf9b3c2e609c54615db60f0682b34f6575c5 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:53:05Z Disables tests for timers in ParDo for Spark runner commit be9c04bf0f2af7b428a61d3b74b68ea18c4f0e03 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:53:16Z Disable tests for timers in ParDo for Dataflow runner commit b66c5cb4f92e2c0d834f44ba246b8186cc9d0291 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-08T04:24:34Z Disable tests for timers in ParDo for direct runner commit e4f062495878328359234da4e586d12cb42965f9 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-07T04:49:40Z Add basic test for timers in ParDoTest commit ad4cbbc18ffbbbedaaa17415220d7ff2908931797 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-08T04:04:51Z No longer reject timers in ParDo commit 0703093767db2776965384b09b322f01f4b79b39 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-08T04:34:34Z Reject timers for ParDo in ApexRunner commit 4b312e8d01e324db77e5f2de62eae35efc153adb Author: Kenneth Knowles <klk@google.com> Date: 2016-12-08T04:34:59Z Reject timers for ParDo in FlinkRunner commit 2e2bc885deb5000a269800bdebad8f17dff5a3a9 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-08T04:35:08Z Reject timers for ParDo in SparkRunner commit 2d2d83f721c21f681f431974edc4e044d44828c5 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-08T04:37:33Z Reject timers for ParDo in DirectRunner ----
18. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1550>
19. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1555> [BEAM-27] Fix exclusion of UsesTimersInParDo for FlinkRunner streaming tests Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [x] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `https://www.apache.org/licenses/icla.txt). --- R: anyone (fixing postcommit) ping @aljoscha @mxm You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> FlinkRunner Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1555.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 #1555 ---- commit a11eb637f6f1c7bdc41b24154cb3893bb194b096 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-08T20:42:03Z Fix exclusion of UsesTimersInParDo for FlinkRunner streaming tests ----
20. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1555>
21. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1612> [BEAM-27] Support timer setting and receiving in SimpleDoFnRunner Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [] Make sure tests pass via `mvn clean verify`.

- (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace ``<Jira issue #>`` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement] (<https://www.apache.org/licenses/icla.txt>). --- These are a couple commits peeled off that should enable each runner to begin fulfilling the timer API. You can merge this pull request into a Git repository by running: `$ git pull https://github.com/kennknowles/incubator-beam SimpleDoFnRunner-timers` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1612.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 #1612 ---- commit 95f954de00d20b27e127e41a0118a305e2cd8a94 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-08T04:09:06Z Make TimerSpec and StateSpec fields accessible commit 19ccebe6170b24af0067c7a04483d8e9156a0ef5 Author: Kenneth Knowles <klk@google.com> Date: 2016-11-23T22:21:40Z Add timer support to DoFnRunner(s) ----
22. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1652> [BEAM-27, BEAM-362] Move InMemoryTimerInternals to runners-core; restore a deprecated copy in its place Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-`<Jira issue #>`] Description of pull request` - [] Make sure tests pass via ``mvn clean verify``. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace ``<Jira issue #>`` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement] (<https://www.apache.org/licenses/icla.txt>). --- This is a replay of ec0bf7b4023ff75f4ec6723d2e77ed507eb57c51 with an additional adapter commit that should enable integration tests to succeed. Note that this configuration of commits works will with the current capabilities of ``git`` and GitHub. This behavior is not baked into the history, but is calculated dynamically by both, so it may improve over time. Notes on what would happen if this were done in one commit: - ``git log --follow`` finds the history - ``git diff --find-copies`` finds the copy - ``git blame --find-copies-harder`` finds the copy, but ``--find-copies`` does not I don't recall which level GitHub's UI works with, but the most important two are ``git log`` and ``git blame``. To make sure it works without any of the above, two commits is better. You can merge this pull request into a Git repository by running: `$ git pull https://github.com/kennknowles/incubator-beam InMemoryTimerInternals` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1652.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 #1652 ---- commit 6effd6c93587d1c0a02ba9285b47d165ae5c517d Author: Kenneth Knowles <klk@google.com> Date: 2016-12-16T04:45:56Z Move InMemoryTimerInternals to runners-core commit 557d2d724c53233ad9d34c9239ff5cf77b754d73 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-17T04:22:59Z Restore SDK's InMemoryTimerInternals, deprecated -- --
23. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1612>
24. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1660> [BEAM-27] Support setting and deleting timers by ID in InMemoryTimerInternals Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-`<Jira issue #>`] Description of pull request` - [x] Make sure tests pass via ``mvn clean verify``. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace ``<Jira issue #>`` in the title with the actual Jira issue number, if there is one. - [x] If this contribution is large, please file an Apache [Individual Contributor License Agreement] (<https://www.apache.org/licenses/icla.txt>). --- This is build on top of the move to runners-core in #1652. Only the second commit contains nontrivial changes. You can merge this pull request into a Git repository by running: `$ git pull https://github.com/kennknowles/incubator-beam InMemoryTimerInternals-dedup` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1660.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 #1660 ---- commit 6effd6c93587d1c0a02ba9285b47d165ae5c517d Author: Kenneth Knowles <klk@google.com> Date: 2016-12-16T04:45:56Z Move InMemoryTimerInternals to runners-core commit 557d2d724c53233ad9d34c9239ff5cf77b754d73 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-17T04:22:59Z Restore SDK's InMemoryTimerInternals, deprecated commit 0ef0e3a3ecabafdd934f7d03f47f0a220187fb22 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-19T22:01:36Z Support set and delete of timer by ID in InMemoryTimerInternals ----
25. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1652>

26. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1668> [BEAM-27, BEAM-362] Remove deprecated InMemoryTimerInternals from SDK Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `https://www.apache.org/licenses/icla.txt). --- You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> InMemoryTimerInternals Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1668.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 #1668 ---- commit bbea8469912b23383aeeee9ae5cf084b5801706e Author: Kenneth Knowles <klk@google.com> Date: 2016-12-20T22:07:00Z Remove deprecated InMemoryTimerInternals from SDK ----
27. GitHub user kennknowles opened a pull request: <https://github.com/apache/incubator-beam/pull/1673> [BEAM-27] Require TimeDomain to delete a timer Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [x] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `https://www.apache.org/licenses/icla.txt). --- R: @aljoscha A bit of an oversight, I neglected the fact that runners generally store different sorts of timers in rather different ways. When a user sets a timer, the `DoFnSignature` is available, so this will be for free. And when system code deletes a timer, the domain will always be known. This will require a Dataflow update, so don't worry if Dataflow-specific integration tests don't pass. You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/incubator-beam> delete-by-domain Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-beam/pull/1673.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 #1673 ---- commit 46dfd0fb4d2a1533d3ed053983faee6537d3ccf0 Author: Kenneth Knowles <klk@google.com> Date: 2016-12-21T04:09:25Z Require TimeDomain to delete a timer ----
28. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1668>
29. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1673>
30. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-beam/pull/1660>
31. GitHub user kennknowles opened a pull request: <https://github.com/apache/beam/pull/1827> [BEAM-27] Allow absolute timers Be sure to do all of the following to help us incorporate your contribution quickly and easily: - [] Make sure the PR title is formatted like: `[BEAM-<Jira issue #>] Description of pull request` - [] Make sure tests pass via `mvn clean verify`. (Even better, enable Travis-CI on your fork and ensure the whole test matrix passes). - [x] Replace `https://www.apache.org/licenses/icla.txt). --- You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/kennknowles/beam> absolute-timers Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/beam/pull/1827.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 #1827 ---- commit 352e8411612fa1915878428746aa21b618e39218 Author: Kenneth Knowles <klk@google.com> Date: 2017-01-24T04:50:50Z Allow absolute timers ----
32. Github user asfgit closed the pull request at: <https://github.com/apache/beam/pull/1827>
33. This is supported in enough runners that we should track only limitations and bugs going forward.