

Анализ логов

Введение

DRD ведет 3 лог-файла:

- **drd.log**, содержащий информационные сообщения
- **drd_error.log**, в который пишутся ошибки самого DRD
- **drd_races.log**, содержащий отчеты обо всех обнаруженных гонках в приложении.



Если DRD ведет себя некорректно (а то и совсем падает), то крайне желательно приложить эти файлы к баг-репорту.

drd_races.log

Пример обнаруженной гонки по полю какого-то класса:

```
WRITE_READ data race occurred between current thread AWT-EventQueue-0(id = 18) and thread Thread-11(id = 27)

Race target: FIELD com/devexperts/jtt/tracker/gui/worklogs/model/HistoryWorklogsAdapter.fromMillis
Current thread 18: com/devexperts/jtt/tracker/gui/worklogs/model/HistoryWorklogsAdapter.changePeriod (line : 46)
Racing thread 27: com/devexperts/jtt/tracker/gui/worklogs/model/HistoryWorklogsAdapter.access$200 (line : 20)

-----Stack trace for racing thread (id = 27) is not available.-----

-----Current thread's stack trace (id = 18) : -----
    at com.devexperts.jtt.tracker.gui.worklogs.model.HistoryWorklogsAdapter.changePeriod(HistoryWorklogsAdapter.
java:46)
    at com.devexperts.jtt.tracker.gui.worklogs.model.WorklogAdapterWrapper.changePeriod(WorklogAdapterWrapper.
java:253)
    at com.devexperts.jtt.tracker.gui.worklogs.WorklogsHistoryPanelController.refresh
(WorklogsHistoryPanelController.java:119)
    at com.devexperts.jtt.tracker.gui.worklogs.WorklogsHistoryPanelController.refresh
(WorklogsHistoryPanelController.java:123)
    at com.devexperts.jtt.tracker.gui.worklogs.WorklogsHistoryPanelController.actionPerformed
(WorklogsHistoryPanelController.java:80)
    at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:1995)
    at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2318)
    at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:387)

-----Current thread's vc were: -----
Live: [1:2][18:68693][26:7] size = 6; Dead: [12:5]; tid = 18
-----Data vc were: -----
{W:[0:0] R:[27:13]; exclusive = true}
```

Это значит следующее: текущий поток (id=18, name=AWT-EventQueue-0) обратился к полю fromMillis класса HistoryWorklogsAdapter на чтение из метода HistoryWorklogsAdapter.changePeriod(), строка 46. Это обращение образует гонку с предшествовавшим обращением к этому же полю на запись в другом потоке (name = "Thread-11", id=27) из метода HistoryWorklogsAdapter.access\$200() в строке номер 20. Stacktrace потока id=27 недоступен, stacktrace текущего потока (id=18) указан.

Также указано состояние часов текущего потока и поля на момент обнаружения гонки (например, по ним видно, что текущий поток не располагал информацией о предыдущем обращении второго потока к этому полю - у него компонента часов, соответствующая потоку с id=27, отсутствует, то есть равна 0, что меньше, чем эта же компонента в часах поля - 13).

Пример обнаруженной гонки по какому-то объекту

```

READ_WRITE data race occurred between current thread AWT-EventQueue-0(id = 18) and thread Thread-11(id = 27)

Race target: method calls on java/util/Collection
Current thread 18 called java/util/Collection.iterator() at : com/devexperts/jtt/util/adapters
/CachingDataAdapter.addAll (line : 66)
Racing thread 27 called java/util/Collection.addAll() at : com/devexperts/jtt/tracker/core/service/impl
/CacheAwareDataProvider.getWorklogs (line : 118)

-----Stack trace for racing thread (id = 27) is not available.-----

-----Current thread's stack trace (id = 18) : -----
    at com.devexperts.jtt.util.adapters.CachingDataAdapter.addAll(CachingDataAdapter.java:66)
    at com.devexperts.jtt.tracker.gui.worklogs.model.HistoryWorklogsAdapter.access$500(HistoryWorklogsAdapter.
java:20)
    at com.devexperts.jtt.tracker.gui.worklogs.model.HistoryWorklogsAdapter$2.succeeded(HistoryWorklogsAdapter.
java:69)
    at com.devexperts.jtt.tracker.gui.worklogs.model.HistoryWorklogsAdapter$2.succeeded(HistoryWorklogsAdapter.
java:66)
    at com.devexperts.jtt.tracker.core.service.impl.DataServiceImpl$1$1.run(DataServiceImpl.java:80)
    at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:209)

-----Current thread's vc were: -----
Live: [1:2][18:68676] size = 4; Dead: [12:5]; tid = 18
-----Field vc were: -----
{W:[27:14] R:[0:0]; exclusive = true}

```

Это значит следующее: текущий поток (id=18) вызвал метод iterator() на объекте типа Collection в методе CachingDataAdapter.addAll(), что DRD трактовал как обращение к этому объекту на чтение. Этот вызов образует гонку с предшествовавшим вызовом метода "add()" того же объекта, что DRD трактовал как обращение к нему на запись, в другом потоке (id=27) из метода CacheAwareDataProvider.getWorklogs(), строка номер 118. Stacktrace потока id=27 недоступен, stacktrace текущего потока (id=18) указан. Также указано состояние часов текущего потока и поля на момент обнаружения гонки.

drd.log

В начале работы DRD в drd.log пишутся конфиги, которые прочитал DRD, и выдается сообщение об успешном запуске DRD. Если этих строк нет, значит что-то не так.

```

DRDAgent started it's work.
Sync instrumentation scope config : <description of scope, in which sync events would be intercepted>
Race detection instrumentation scope config : <description of scope, in which data races would be searched>

Accesses config:
Accesses of following fields wouldn't be instrumented : <description of accesses to which fields wouldn't be
analyzed against data races>
Foreign calls of following methods wouldn't be instrumented : <description of which foreign calls wouldn't be
analyzed against data races>

Contracts config:
Contracts matcher would try to apply following contracts sequentially. If no one matches, target method would
be treated as write.
<List of contracts>

Trace config: <trace config>

+++++
Config initialized : true.
Transformer and detector launched.
+++++
...
Happens-before config read successfully. Synchronization points are: ...

```

Во время работы DRD в drd.log пишутся информационные сообщения, которые могут быть полезны разработчикам DRD для выявления проблем и их причин. В частности, раз в минуту собирается статистика о потреблении памяти и количестве вазов различных методов DRD. Собранные данные будут невероятно важны при анализе перформанса/ошибок DRD разработчиками (авторами данного мануала).

Пример статистики:

```

+++++
Last time stats were gathered 59999 ms ago.
19 active threads.
Memory usage: init = 134185856(131040K) used = 53789304(52528K) committed = 80805888(78912K) max = 1908932608
(1864192K).
Contract clocks: 0 ( + 0 - 0 = 0. Disposal took 0 ms).
  Hits : 0.
  Approx total reserved size: 0; real size : 0. Dead threads clocks size: 0.
  Contracts ignored : 0, processed : 0.
Volatile clocks: 13 ( + 0 - 0 = 0. Disposal took 0 ms).
  Hits : 0.
  Approx total reserved size: 260; real size : 104. Dead threads clocks size: 34.
  Volatiles ignored : 0, processed : 0.
Syn clocks: 58 ( + 0 - 0 = 0). New clock for nothing : 0.
  Hits : 200.
  Approx total reserved size : 960; real size : 426. Dead threads clocks size : 180.
  Syncs ignored : 0, processed : 200.
Foreign clocks: 2388 ( + 600 - 0 = 600). New clock for nothing : 0.
  Hits : 1800
  Approx total reserved size: 10532; real size : 9988. Dead threads clocks size: 180
Vector clock counters:
  New array allocations for live clocks: 0 vs 0 reusages.
  Dead clocks during merge: 0 cached, 0 same, 0 new.
  Generations update: 0 light, 0 hard (0 of them from zero).
  Vector clock merge: 0 optimized vs 0 full acquires; 0 optimized vs 0 full releases. 0 two-way merges.
Diff was calculated 0 times.
  Shared reads in data clock occurred 0 times.
  Data races reported: 0, skipped: 0.
  VC total live resizes: 0, total dead resizes: 0.
  Yield counter : 0.
HBDynamic helper cache hits : 0; misses : 0.
Happens-before vertices returned true : 0; false : 0.
Foreign calls: 10000; field hits: 0
-----Data clock histogram (top 20): -----
24170  com/devexperts/jtt/tracker/core/time/impl/TimeSpyImpl$2 (java/util/Map$Entry)
6259  com/devexperts/jtt/util/event/impl/StrongRefEventManager (java/util/Iterator)
3637  com/devexperts/jtt/tracker/core/time/impl/TimeSpyImpl$2 (java/util/Set)
3637  com/devexperts/jtt/tracker/core/time/impl/TimeSpyImpl$2 (java/util/HashMap)
3539  com/devexperts/jtt/tracker/core/time/impl/TimeSpyImpl$2 (java/util/Iterator)
2216  com/devexperts/jtt/tracker/core/storage/impl/local/SerializableStorageImpl (java/util/Set)
2059  com/devexperts/jtt/common/impl/IssueHeaderImpl
768   com/devexperts/jtt/common/impl/IssueDetailsImpl
711   com/devexperts/jtt/tracker/core/storage/impl/local/IssueRecord (java/util/Collection)
517   com/devexperts/jtt/tracker/core/issues/IssuesManager$StorageListener$1 (java/util/Iterator)
351   com/devexperts/jtt/tracker/gui/activity/spy/impl/UserMonitorStateProvider (java/awt/PointInfo)
335   com/devexperts/jtt/tracker/core/issues/IssuesManager$StorageListener (java/util/Iterator)
280   com/devexperts/jtt/tracker/core/issues/IssuesManager$StorageListener (java/util/Set)
274   com/devexperts/jtt/common/impl/WorklogImpl
221   com/devexperts/jtt/tracker/core/jira/sync/JiraSyncService (java/util/Iterator)
218   com/devexperts/jtt/tracker/gui/log/LogTabPanelController$2 (javax/swing/event/TableModelEvent)
172   com/devexperts/jtt/util/service/AbstractService (java/util/Iterator)
165   com/devexperts/jtt/tracker/core/issues/IssuesManager$StorageListener$1
161   com/devexperts/jtt/tracker/gui/worklogs/WorklogEditorPanelController
144   com/devexperts/jtt/util/adapters/AbstractDataAdapter$3
-----Total: 53276-----
Total 5 data races reported up to this moment.
Stats gathered in 0 ms.
+++++

```