

Get 'em before they get you

Václav Pech

Software Developer, Technology Evangelist
JetBrains, s.r.o.

Agenda

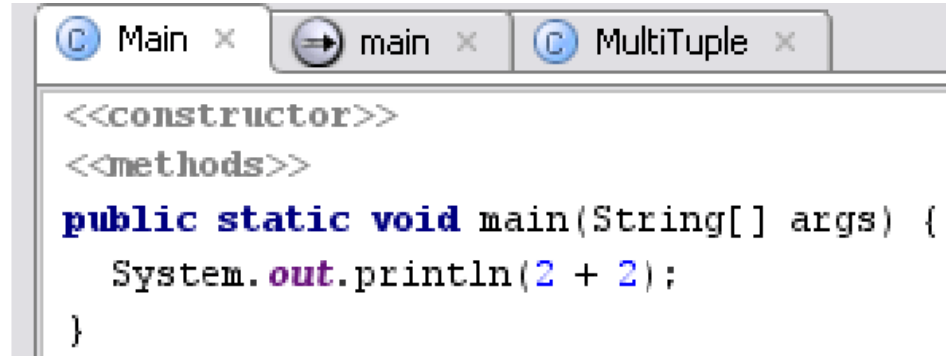
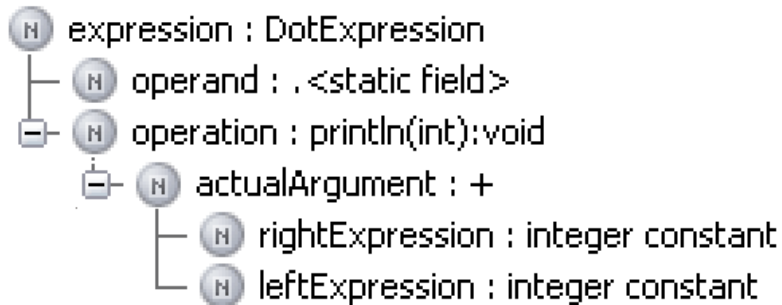
- Static Code Analysis
- Duplicate Detection
- Stacktrace Analysis
- Dataflow Analysis
- Dependency Analysis
- DSM



Static Code Analysis

- Searches AST for bug patterns
 - On demand
 - On the fly

Abstract Syntax Tree



```
Main x main x MultiTuple x
<<constructor>>
<<methods>>
public static void main(String[] args) {
    System.out.println(2 + 2);
}
```

The screenshot shows an IDE window with three tabs: `Main`, `main`, and `MultiTuple`. The `main` tab is active and shows the following Java code:

Advanced capabilities

- Quick fixes
- Profiles
- Suppressing false positives
- Spell-checker
- Command-line and CI integration

There's more than Java out there

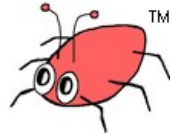
- JavaScript
- CSS, HTML, XML
- GWT, Struts
- EE, ME

```
! .prodname span {  
    display: block;  
    margin: -3px 0 0 0;  
    padding: 0;  
    font-size: 0.8em;  
    font-weight: normal;  
    font-size: 70%;  
}
```

Property font-size is overwritten

```
var selectedQuantityIndex = volumeQuantities.length - 1;  
if ((quantity == null) || (isNaN(quantity))) {  
    // Comparison quantity == null may cause unexpected type coercion
```

722



Create your own bug patterns

Define

an AST sub-tree to search for

an AST sub-tree to replace it with

Helper annotations - JSR 305 & 308

- @Nullable, @NotNull, @Nls, @NonNls
- @PropertyKey, @Pattern, @Language
- @ThreadSafe, @GuardedBy
- @Immutable

@Pattern

```
abstract class Tool {  
    @Pattern("[a-zA-Z_0-9]+")  
    abstract String getId();  
}
```

```
class MyTool extends Tool {  
    String getId() {  
        return "ID (Not Valid)";  
    }  
}
```

Expression 'ID (Not Valid)' doesn't match pattern: [a-zA-Z_0-9]+ [more...](#) (Ctrl+F1)

Language injection

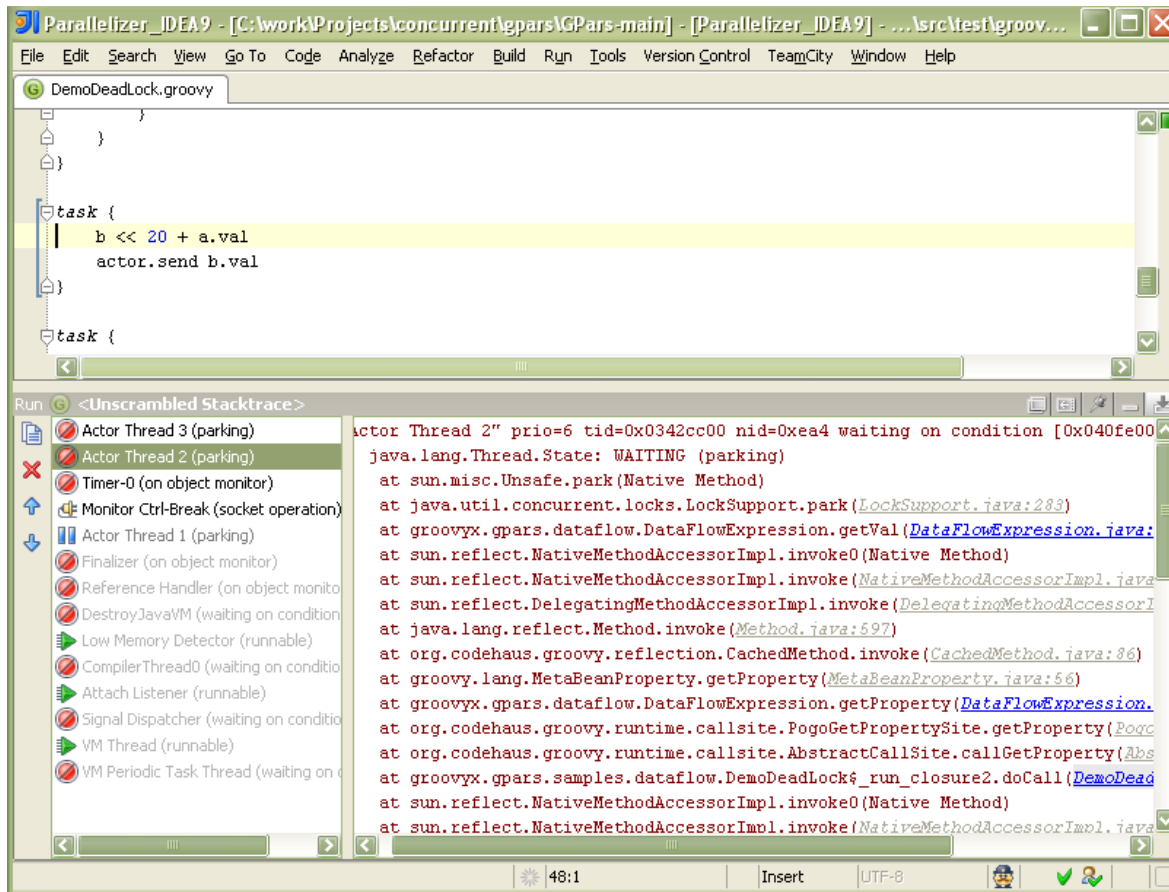
```
@Language("CSS")  
String cssDemo = "h2 { font-style: italic; }";
```

```
@Language("RegExp")  
String regexpDemo = "\\p{Alpha}(abc)\\1\\2";
```

Unresolved backreference

```
@Language("HTML")  
String xmlDemo = "<head><title>Test</title></hed>";
```

Stacktrace Analysis



The screenshot shows the IntelliJ IDEA IDE interface. The top window displays the source code for `DemoDeadLock.groovy`. A `task` block is highlighted, containing the following code:

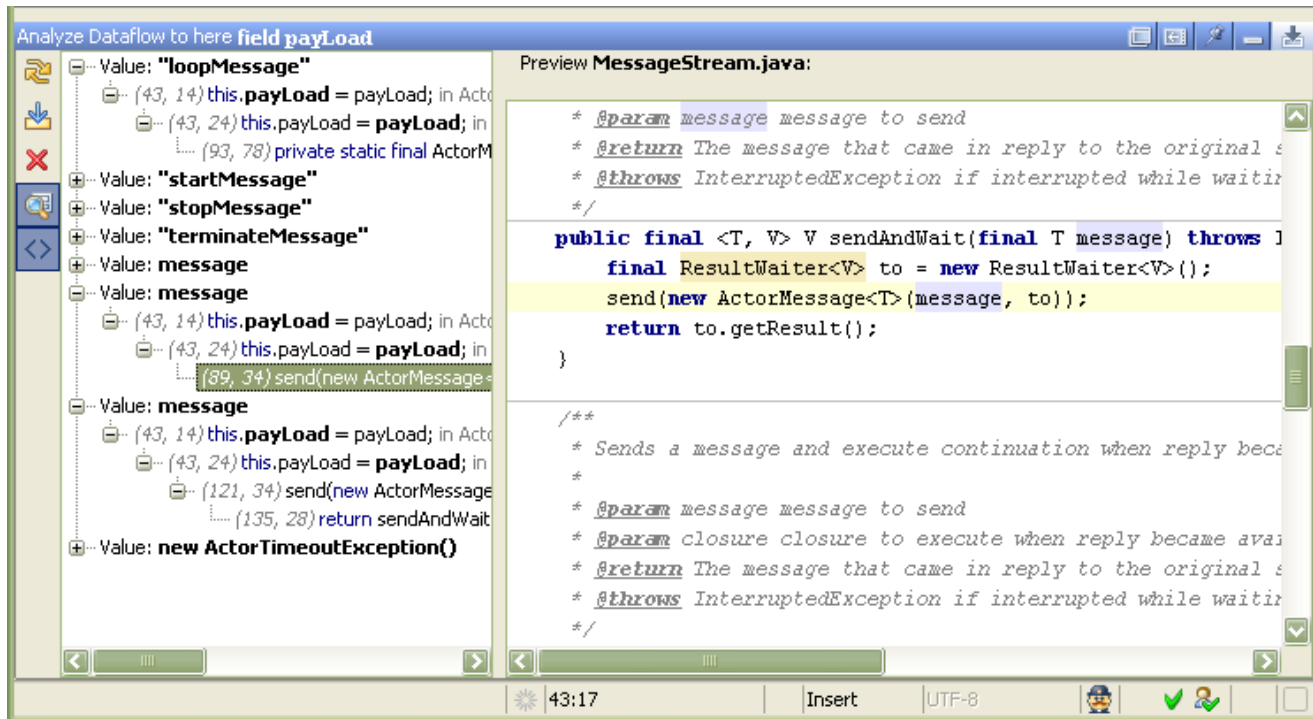
```
task {  
    b << 20 + a.val  
    actor.send b.val  
}
```

The bottom window shows the "Run" console with an "Unscrambled Stacktrace". The stacktrace details the state of "Actor Thread 2" and the sequence of method calls leading to the deadlock:

```
Actor Thread 2" prio=6 tid=0x0342cc00 nid=0xea4 waiting on condition [0x040fe00  
java.lang.Thread.State: WAITING (parking)  
    at sun.misc.Unsafe.park(Native Method)  
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:283)  
    at groovyx.gpars.dataflow.DataFlowExpression.getVal(DataFlowExpression.java:  
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:  
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorI  
    at java.lang.reflect.Method.invoke(Method.java:597)  
    at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:86)  
    at groovy.lang.MetaBeanProperty.getProperty(MetaBeanProperty.java:56)  
    at groovyx.gpars.dataflow.DataFlowExpression.getProperty(DataFlowExpression.  
    at org.codehaus.groovy.runtime.callsite.PogoGetPropertySite.getProperty(Pogo  
    at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callGetProperty(Abs  
    at groovyx.gpars.samples.dataflow.DemoDeadLock$_run_closure2.doCall(DemoDead  
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java
```

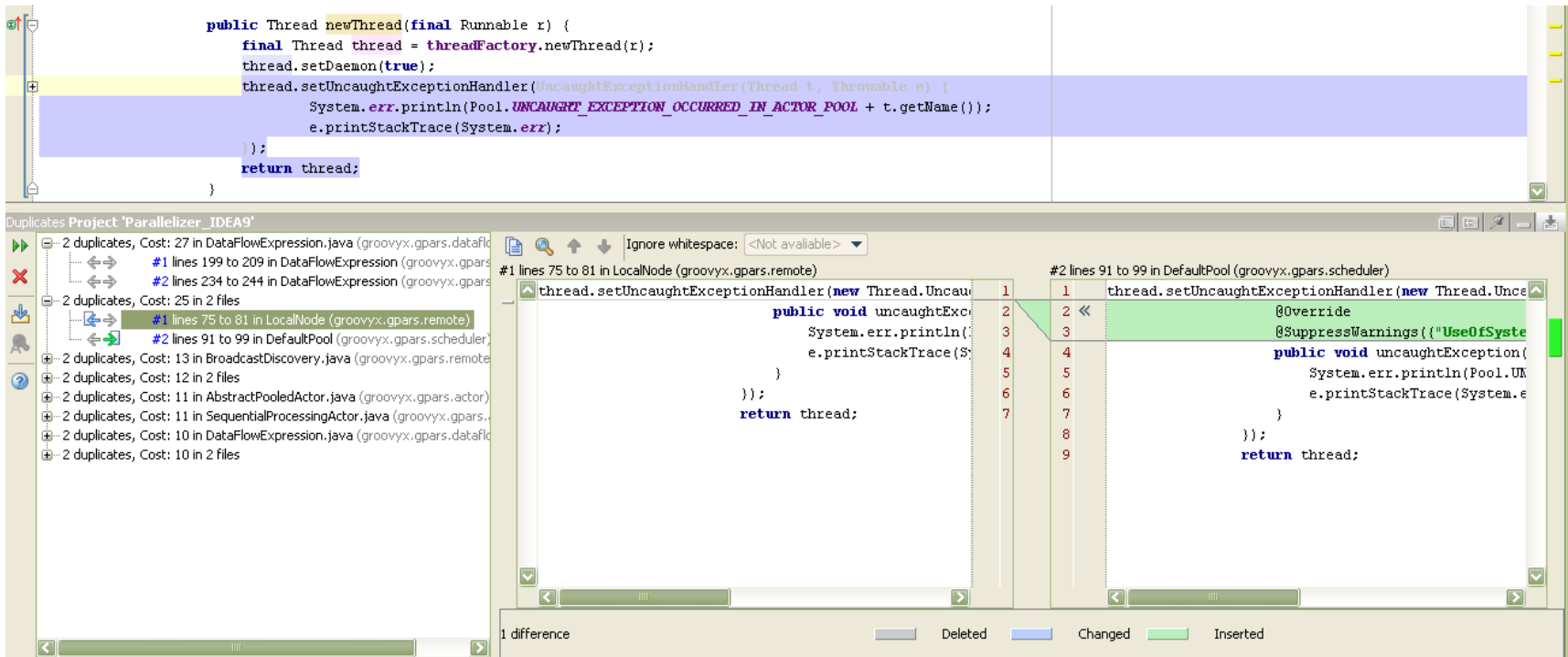
Dataflow Analysis

- From / To here



The screenshot displays the IntelliJ IDEA interface with the Dataflow Analysis tool open. The left pane, titled "Analyze Dataflow to here field payload", shows a vertical flow of data. It starts with a "loopMessage" value, followed by several assignments to "this.payload" at various line numbers (e.g., (43, 14), (43, 24), (93, 78)). It then shows "startMessage" and "terminateMessage" values, followed by "message" values and a "new ActorTimeoutException()" value. The right pane, titled "Preview MessageStream.java:", shows the source code for the "sendAndWait" method. The code includes Javadoc comments and the method signature: `public final <T, V> V sendAndWait(final T message) throws InterruptedException`. The method body shows the creation of a `ResultWaiter` and the sending of an `ActorMessage` to it, followed by a `return` statement. The status bar at the bottom indicates the time is 43:17, the mode is Insert, and the encoding is UTF-8.

Duplicate detection



```

public Thread newThread(final Runnable r) {
    final Thread thread = threadFactory.newThread(r);
    thread.setDaemon(true);
    thread.setUncaughtExceptionHandler(new UncaughtExceptionHandler(Thread t, Throwable e) {
        System.err.println(Pool.UNCAUGHT_EXCEPTION_OCCURRED_IN_ACTOR_POOL + t.getName());
        e.printStackTrace(System.err);
    });
    return thread;
}
    
```

Duplicates Project 'Parallelizer_IDEA9'
 - 2 duplicates, Cost: 27 in DataFlowExpression.java (groovyx.gpars.dataflow)

- #1 lines 199 to 209 in DataFlowExpression (groovyx.gpars.dataflow)
- #2 lines 234 to 244 in DataFlowExpression (groovyx.gpars.dataflow)

 - 2 duplicates, Cost: 25 in 2 files

- #1 lines 75 to 81 in LocalNode (groovyx.gpars.remote)
- #2 lines 91 to 99 in DefaultPool (groovyx.gpars.scheduler)

 - 2 duplicates, Cost: 13 in BroadcastDiscovery.java (groovyx.gpars.scheduler)

- #1 lines 199 to 209 in BroadcastDiscovery (groovyx.gpars.scheduler)
- #2 lines 234 to 244 in BroadcastDiscovery (groovyx.gpars.scheduler)

 - 2 duplicates, Cost: 12 in 2 files

- #1 lines 75 to 81 in LocalNode (groovyx.gpars.remote)
- #2 lines 91 to 99 in DefaultPool (groovyx.gpars.scheduler)

 - 2 duplicates, Cost: 11 in AbstractPooledActor.java (groovyx.gpars.actor)

- #1 lines 199 to 209 in AbstractPooledActor (groovyx.gpars.actor)
- #2 lines 234 to 244 in AbstractPooledActor (groovyx.gpars.actor)

 - 2 duplicates, Cost: 11 in SequentialProcessingActor.java (groovyx.gpars.actor)

- #1 lines 199 to 209 in SequentialProcessingActor (groovyx.gpars.actor)
- #2 lines 234 to 244 in SequentialProcessingActor (groovyx.gpars.actor)

 - 2 duplicates, Cost: 10 in DataFlowExpression.java (groovyx.gpars.dataflow)

- #1 lines 199 to 209 in DataFlowExpression (groovyx.gpars.dataflow)
- #2 lines 234 to 244 in DataFlowExpression (groovyx.gpars.dataflow)

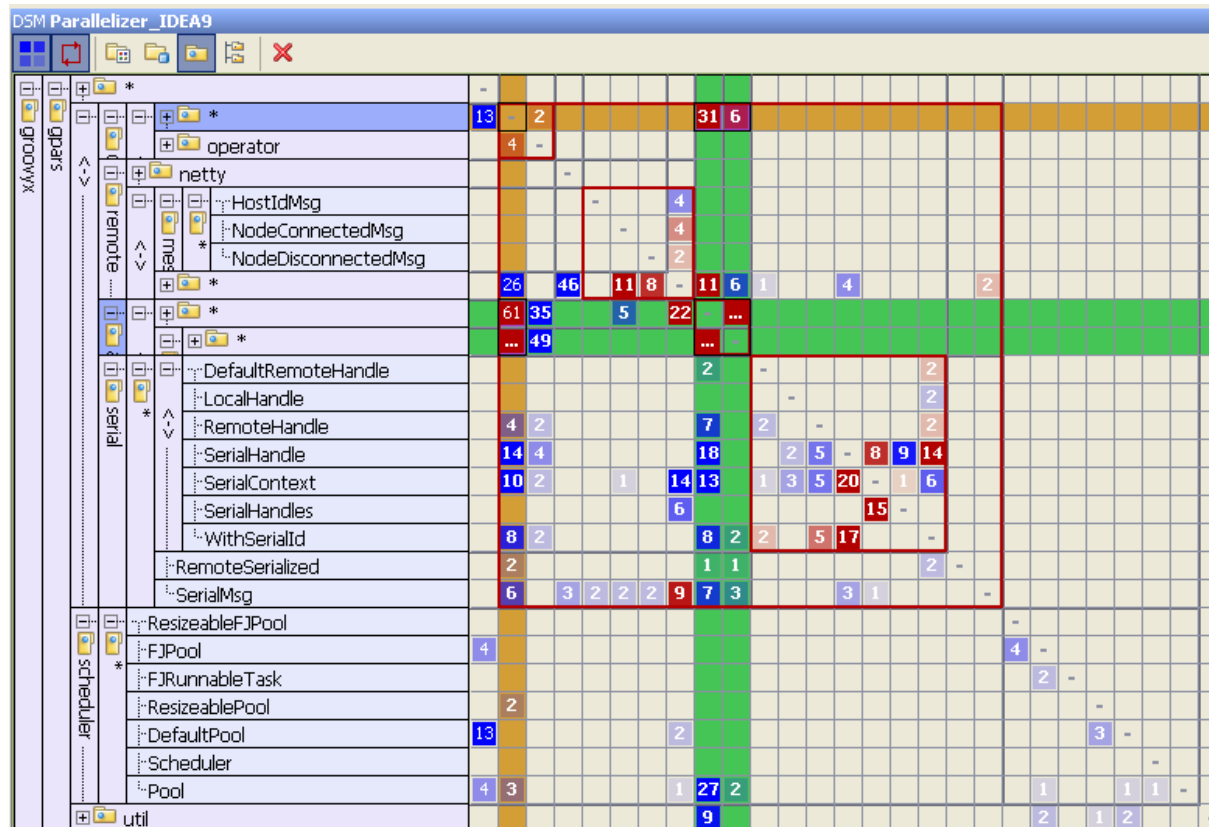
 - 2 duplicates, Cost: 10 in 2 files

- #1 lines 75 to 81 in LocalNode (groovyx.gpars.remote)
- #2 lines 91 to 99 in DefaultPool (groovyx.gpars.scheduler)

#1 lines 75 to 81 in LocalNode (groovyx.gpars.remote)	#2 lines 91 to 99 in DefaultPool (groovyx.gpars.scheduler)
1 thread.setUncaughtExceptionHandler(new Thread.UncaughtExceptionHandler() {	1 thread.setUncaughtExceptionHandler(new Thread.UncaughtExceptionHandler() {
2 public void uncaughtException(Thread t, Throwable e) {	2 @Override
3 System.err.println(Pool.UNCAUGHT_EXCEPTION_OCCURRED_IN_ACTOR_POOL + t.getName());	3 @SuppressWarnings({"UseOfSystemOutOrSystemErr"})
4 e.printStackTrace(System.err);	4 public void uncaughtException(Thread t, Throwable e) {
5 }	5 System.err.println(Pool.UNCAUGHT_EXCEPTION_OCCURRED_IN_ACTOR_POOL + t.getName());
6 }}	6 e.printStackTrace(System.err);
7 return thread;	7 }
	8 }}
	9 return thread;

1 difference
 Deleted Changed Inserted

Dependency Structure Matrix



Summary

- Customize
- Use permanently
- Run continuously

So that you can sleep better!

A large, stylized question mark graphic. The top curve is orange, the middle stem is yellow, and the bottom dot is orange. The word 'Questions' is centered within the upper curve of the question mark.

Questions