Featured in Enterprise Architecture

Preparing for Your First MongoDB Deployment: Capacity Planning and Monitoring

In this article, author Mat Keep discusses the deployment best practices of MongoDB databases with focus on capacity planning and monitoring aspects. He also explains the topics like hardware selection, key metrics for monitoring and when it’s time to add shards.

All in Enterprise Architecture

San Francisco 2013
Nov 11 - Nov 15

London 2014
March 3 - Mar 7

- Mobile
- HTML5
- JavaScript
- APM
- API Design
- Big Data
- Cloud
- Agile

All topics
IBM Code Rally

You are here: InfoQ Homepage News ThreadSafe Concurrency Static Analysis Tool Announces First Public Release

ThreadSafe Concurrency Static Analysis Tool Announces First Public Release

by Vikram Gupta on Aug 28, 2013 / Discuss

Share

"Read later"
"My Reading List"
UK based Contemplate Ltd. has announced the first public release of their flagship product ThreadSafe, a static analysis tool for locating concurrency bugs and inefficiencies in Java code.

Prior to this release, which is labeled version 1.2, ThreadSafe was only available to a select group of users. In addition to public availability, this release adds a range of new features.

ThreadSafe runs on Windows, Linux, and OS X, and can be executed as an Eclipse plugin, or via an integration into the SonarQube code quality platform. It will soon be available from Grammatech as well, as a plugin to CodeSonar, the popular static analysis tool.

InfoQ loaded up ThreadSafe and took it for a test drive on a medium-sized multithreaded application. Installation into Eclipse was straightforward and required a restart of Eclipse.

For comparison purposes, we also ran the same application through FindBugs, the popular static analysis tool for locating bugs in Java from a range of categories, including "Multithreaded correctness".

Each of these tools lays claim to its own niche. Where FindBugs located concurrency bugs such as "Method does not release lock on all exception paths" (see Figure 1 below), ThreadSafe seemed better at locating incorrect applications of the Java Memory Model and other infractions more related to actual concurrency misuse.

Figure 1 - FindBugs Multithreaded Correctness bugs

- Multithreaded correctness (70)
  - Empty Synchronized blocks (1)
    - Empty synchronized block (1)
  - Lock not released on all paths (14)
    - Method does not release lock on all exception paths (14)
  - Naked notify (8)
    - Naked notify (8)
  - Possible locking on wrong object (1)
    - Synchronization on getClass rather than class literal (1)
  - Unconditional wait (16)
    - Unconditional wait (16)
  - Use of volatile (11)
    - An increment to a volatile field isn't atomic (11)
  - Wait not in loop (17)
    - Condition.await() not in loop (3)
    - Wait not in loop (14)
  - Wait with two locks held (2)
    - Wait with two locks held (2)

Figure 2 - ThreadSafe analysis on the same project.
Double-clicking on a bug in ThreadSafe opens that code in an Eclipse editor.

Clicking on any bug highlights that bug and exposes a detail pane that provides more detail about that bug. In the case of Figure 3 we selected "Field 'complete' may be synchronized inconsistently". You can see that it displays not only the incorrect accesses, but also the correct accesses, which really helps in understanding the inconsistencies in your code. You can click on the line number in the detail pane to navigate to that line in the source. You can also click on the "(more)" link to see more detailed information about the bug in your web browser.

Figure 3 - Details Pane

ThreadSafe is very aware of the Java Memory Model. Most of the code that it located in our project were related to inconsistent synchronization, which besides permitting race conditions, could lead to "out-of-thin-air" errors and might hide instance variable writes in one thread from other concurrent threads.

InfoQ purposely introduced a potential "deadly-embrace" deadlock into the codebase, and ThreadSafe did not locate it. We pointed this out to Contemplate CSO David Aspinall, who acknowledged:

We do have some deadlock analysis in ThreadSafe, but it is limited. Inside Contemplate we have more sophisticated deadlock analyses which we use for consulting projects. We plan to move these into the product in future, when we
can ensure a low enough false-positive rate in line with the rest of ThreadSafe.

InfoQ also asked Aspinall if ThreadSafe is aware of Brian Goetz's concurrency annotations. He replied:

In fact we have support for these in the analysis engine but it's not yet exposed to users. This will go into one of our next updates.

We asked Contemplate CEO Derick James about the pricing:

Our pricing model is per-project licensing. We have a scale of license costs based on a broad banding of lines of code. This approach is in line with many of the commercial code quality products we see.

A little background on the company and the product: Contemplate Ltd is a UK spin-out company from the University of Edinburgh School of Informatics, founded in 2009. ThreadSafe is its first product and builds on a 5-year academic and industrial research program studying new generation technology for advanced program analysis.

Before the public release of ThreadSafe, Contemplate says they worked closely with several international investment banks to prove the product on complex commercial multi-threaded code bases.

You can request a trial version of ThreadSafe by sending an email to contact@contemplateltd.com. In East Asia ThreadSafe is also available from the Architect Group.

- Sections
- Process & Practices
- Development
- Topics
- Concurrency
- Debugging
- Static Analysis
- Java
- Bugs and Hotfixes
- FindBugs

Related Editorial

Approachable Concurrency for the JVM with Groovy Parallel Systems

Concurrency in Clojure

Ozma: Extending Scala with Oz Concurrency

Duncan Coutts on Parallelism and Concurrency with Haskell, Distributed Programming with Cloud Haskell

Concurrency in iOS

Related Vendor Content

API Patterns for Cloud & Mobile

Top 5 Use Cases for Moving to a Hybrid Cloud

Continuous Delivery: A Maturity Assessment Model

Everything You Need To Know About Graph Databases – View Introductory Video

Start your FREE TRIAL of AppDynamics Pro

Related Sponsor

AppDynamics is the next-generation application performance management solution that simplifies the management of complex,