

Monitoring Tool for Distributed Java Applications

Jakub Háva

Supervisor: RNDr. Pavel Parízek, Ph.D

Consultant: RNDr. Michal Malohlava, Ph.D

Introduction

- Tool able to collect information from all nodes in a cluster
- Keep this information related to each other
- No need to change the target application

Agenda

- Related Tools
- Requirements
- Architecture Overview
- Implementation
- Example Usage

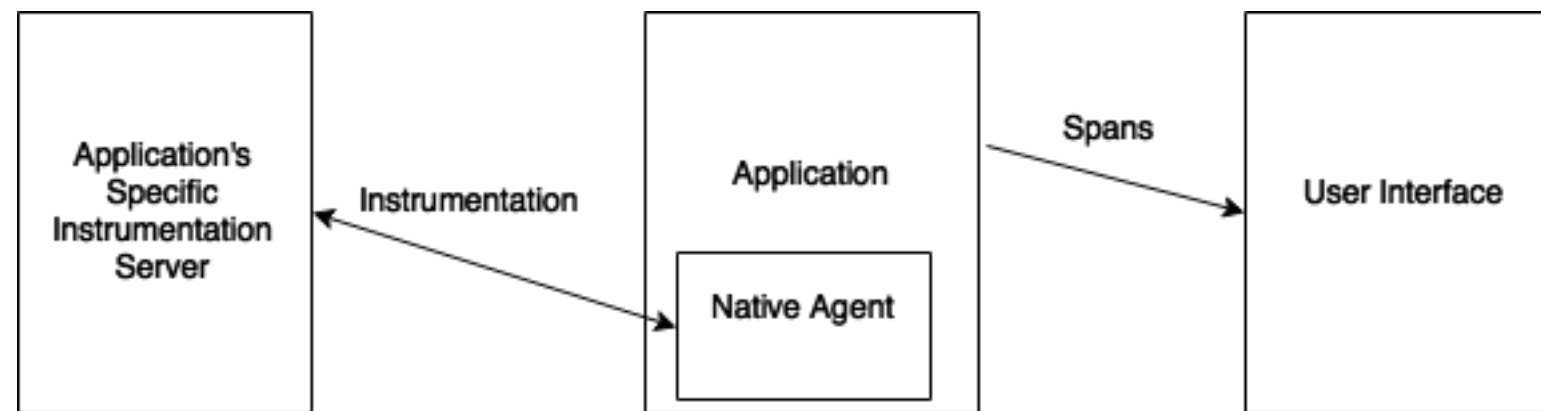
Related Tools

- Google Dapper
 - Closed-source
- Open Zipkin
 - Requires changes of the target application

Requirements

- Influence the monitored application as low as possible
- Use high-level programming language to define instrumentation points - two groups of target users
- Generally applicable solution

Architecture



- Span - encapsulates communication between neighbouring nodes

Deployment Modes

- Shared
 - Instrumentation server is shared across the nodes in the cluster
- Local
 - Each node in the cluster has its own instrumentation server

Implementation Details

- Dedicated Instrumentation Server
 - Byte Buddy Instrumentation Library
- Native Java Agent
 - JVMTI
 - JNI

Example Usage

- Demonstration on H2O machine learning platform
- Monitoring behaviour of distributed map-reduce tasks
- With collaboration with Michal Malohlava from H2O.ai.

Define the Instrumentation Points

```
/**
 * The following methods are used to open and close span denoting task mapping
 */
public static class map {
    @Advice.OnMethodEnter
    public static void enter(@Advice.This MRTask thizz) {
        if (thizz instanceof SumMRTask) {
            TraceContext tc = TraceContext.getFromObject(thizz);
            tc.openNestedSpan( name: "H2O Node" + H2O.SELF.index() + " - mapping ");
        }
    }

    @Advice.OnMethodExit
    public static void exit(@Advice.This MRTask thizz) {
        TraceContext tc = TraceContext.getFromObject(thizz);
        tc.closeCurrentSpan();
    }
}
```

Start the Application

- Start application with the Native agent attached
- Just java option -agentlib="..."
- In local mode, the instrumentation server is started automatically

See the Results

Services	649.000ms	1.298s	1.947s	2.596s
- mrtask	3.245s : h2o node0 - complete mrtask computation	-	-	-
- mrtask	3.242s : h2o node0 - setting and splitting	-	-	-
mrtask	1μ : h2o node0 - remote work - none	-	-	-
- mrtask	3.239s : h2o node0 - remote work - rpc	-	-	-
- mrtask	-	1.552s : h2o node1 - setting and splitting	-	-
mrtask	-	1μ : h2o node1 - remote work - none	-	-
- mrtask	-	1.605s : h2o node1 - remote work - rpc	-	-
- mrtask	-	-	23.000ms : h2o node2 - setting and splitting	-
- mrtask	-	-	3.000ms : h2o node2 - local work - chunks : 2 - new thread	-
mrtask	-	-	1.000ms : h2o node2 - local work - chunks : 1 - new thread	-
mrtask	-	-	1.000ms : h2o node2 - local work - chunks : 1 - same thread	-
mrtask	-	-	1μ : h2o node2 - reducing left	-
mrtask	-	-	1.000ms : h2o node2 - reducing right	-
- mrtask	-	-	-	19.000ms : h2o node2 - local work - chunks : 2 - new thread
mrtask	-	-	-	1.000ms : h2o node2 - local work - chunks : 1 - new thread
- mrtask	-	-	-	18.000ms : h2o node2 - local work - chunks : 2 - new thread
mrtask	-	-	-	1μ : h2o node2 - local work - chunks : 1 - new thread
mrtask	-	-	-	1μ : h2o node2 - local work - chunks : 1 - same thread
mrtask	-	-	-	1.000ms : h2o node2 - local work - chunks : 1 - same thread
mrtask	-	-	-	1μ : h2o node2 - reducing left
mrtask	-	-	-	1.000ms : h2o node2 - reducing right
- mrtask	-	2.258s : h2o node0 - local work - chunks : 2 - new thread	-	-
- mrtask	-	1.000ms : h2o node0 - local work - chunks : 1 - new thread	-	-
mrtask	-	1.000ms : h2o node0 - mapping	-	-
mrtask	-	1μ : h2o node0 - mapping	-	-
mrtask	-	1μ : h2o node0 - mapping	-	-
mrtask	-	1μ : h2o node0 - local work - chunks : 1 - same thread	-	-
mrtask	-	-	-	-
mrtask	-	-	-	-

Evaluation

- Benefits
 - Generic solution
 - Constant overhead
 - Target application remains unchanged
 - Extendable
- Limitations
 - Java 8+
 - Attaching agent at run-time is not possible at the moment

Thank you!

- H2O-3 - <https://github.com/h2oai/h2o-3>
- Distrace - <https://github.com/jakubhava/Distrace>



Proxy Instrumentation Server

- Hierarchical structure of the instrumentation servers
- If a class is not available, ask the parent server
- Combines local and shared approach - asks for a class locally first