layout	date	title	categories	tags	avatarimg	a
post	Mon Jan 16 2017 08:00:00 GMT+0800 (中国标准时间)	JMH架构与源码分析	architecture	architecture jmh	/img/head.jpg	wa

什么是JMH

简单示例

```
import org.openjdk.jmh.annotations.Benchmark;
import org.openjdk.jmh.annotations.BenchmarkMode;
import org.openjdk.jmh.annotations.Mode;
import org.openjdk.jmh.profile.ClassloaderProfiler;
import org.openjdk.jmh.results.format.ResultFormatType;
import org.openjdk.jmh.runner.Runner;
import org.openjdk.jmh.runner.RunnerException;
import org.openjdk.jmh.runner.options.Options;
import org.openjdk.jmh.runner.options.OptionsBuilder;

public class MyBenchmark {

    @Benchmark
    @BenchmarkMode(Mode.AverageTime)
    public void testMethod() {
        int a = 1;
        int b = 2;
        int sum = a + b;
    }

    public static void main(String[] args) throws RunnerException {
        Options opt = new OptionsBuilder()
                .include(MyBenchmark.class.getSimpleName())
                .forks(0)
                .resultFormat(ResultFormatType.JSON)
                .result("d:/temp.txt")
                .addProfiler(ClassloaderProfiler.class)
                .build();
```

```
        new Runner(opt, new JMHOutputFormat()).run();
    }
}
```

执行流程

- OptionsBuilder构建Options的辅助类，通过流式接口来构建Options
- Runner为执行基准测试的类
  - 接收Options作为参数
  - 可以通过自定义OutputFormat来自定义需要的输出结果，比如将结果写到网络
  - 调用run方法进行执行
- run方法中：
  - 首先通过文件锁(锁定System.getProperty("java.io.tmpdir") + "/jmh.lock")来锁定资源执行，**此处如果想同时执行多个基准测试就存在问题，需要考量！考虑使用JCGroup隔离运行环境**
  - 如果锁定成功，执行internalRun方法。
  - 如果锁定失败，且必须要锁定(判断Boolean.getBoolean("jmh.ignoreLock"))，则抛出异常；
  - 如果不必须锁定，打印日志，继续执行internalRun方法
- internalRun中：
  - 验证在Options中配置的Profiles是否有效，无效则抛ProfilersFailedException
  - 如果在Options中设置了result，则创建对应的文件
  - 根据includes和excludes配置，从BenchmarkList中查找匹配的Benchmark，正则表达式匹配，过滤出当前需要执行的测试用例
  - 针对编写的基准测试用例，JMH会自动生成一个/META-INF/BenchmarkList文件，里面记录了基准测试用例相关信息，BenchmarkList类就是从此文件中加载基准测试用例信息
  - 如果设置了Mode，则将Mode设置到过滤出来的测试用例中，注意此处使用的是clone，通过原来的测试用例的参数构建的新的测试用例，防止污染从文件里加载的测试用例信息。下面又针对Mode.All的用例clone了一遍，因为上面的clone没有涉及到Mode.All的情况
  - 针对有parameters的情况，再clone一遍测试用例

- 执行这些测试用例runBenchmarks()：
  - 根据ActionPlan类型来确定是执行runBenchmarksEmbedded还是runSeparate
  - 而ActionPlan的类型主要通过Options中的fork参数来判断,<=0为EMBEDED，否则为FORKED
  - P1:runBenchmarksEmbedded()：
    - 遍历ActionPlan中的Action进行执行，Action中包含了执行的参数和模式，最终执行单元是通过参数、模式以及Acceptor（一个回调类）构建的BenchMarkHandler
    - 在BenchMarkHandler的runIteration里，执行了具体的测试，包括各种Profile的执行和测试的最终执行
  - P2:runSeparate()：
    - 通过ProcessorBuilder来构建进程进行执行,相当于构建了一个独立的运行环境
    - 将测试用例丢到运行环境内进行运行
- 通过设置的ResultFormat格式化结果，并输出

JMH整体架构

JMH组件

- Runner：执行JMH测试
- Result：JMH执行结果
- ResultFormat：结果格式
- OutputFormat：输出格式
- Options：执行JMH的参数

JMH包结构

jmh-core中包含了7个包：

- annotations：没什么可说的了，JMH中的注解

- generators.core：用于生成JMH相关代码
- infra：对被测试的方法的资源消耗的一些模拟类(暂未搞清楚)
- profile：各种自定义监控类，类似Filter，通过OptionsBuilder.addProfiler构建profile链，增加感兴趣的监控点
- results：JMH执行结果
- runner：执行JMH的相关类
- util：工具类

相关类

注解类

AuxCounters Benchmark BenchmarkMode CompilerControl Fork Group GroupThreads Level Measurement Mode OperationsPerInvocation OutputTimeUnit Param Scope Setup State TearDown Threads Timeout Warmup