

平均值试图用一个结果来表示整体分布情况，不同于仅计算平均值，我们可以按照时间间隔对分布进行采样。当用于正态分布的数据时，通常以固定间隔进行采样。不过只要稍加修改，我们就可以将这种技术应用于 JVM 统计。

修改的方法是，使用的采样要将长尾分布考虑进去，从平均值开始，然后是第 90 百分位数，并用对数校正，如下面的方法计时结果所示。这意味着我们是在按照一种更符合数据形状的模式进行采样。

```
50.0% level was 23 ns
90.0% level was 30 ns
99.0% level was 43 ns
99.9% level was 164 ns
99.99% level was 248 ns
99.999% level was 3,458 ns
99.9999% level was 17,463 ns
```

采样表明，尽管执行一个 getter 方法的平均时间是 23 纳秒，但对于 1000 个请求中的一个，时间就差了一个数量级；对于 100 000 个请求中的一个，则比平均值差了两个数量级。

长尾分布也可以称为高动态范围分布 (high dynamic range distribution)。观测值的动态范围通常定义为最大记录值除以最小值（假设它不为零）。

将对数进行百分位数是理解长尾的一个有用的简单工具。然而，要进行更复杂的分析，我们可以使用一个公共库来处理高动态范围的数据集。这个库叫作 HdrHistogram，可以从 GitHub 上获得。它最初是由 Azul Systems 公司的 Gil Tene 创建的，Mike Barker、Darach Ennis 和 Coda Hale 也做了额外的工作。



直方图通过使用一组有限的范围区间（称为“桶”）来总结数据，并显示数据落入每个桶的频率。

HdrHistogram 也可以在 Maven 中央仓库上使用。在本书撰写时，它的当前版本是 2.1.9，要将其添加到自己的项目中，可以通过在 pom.xml 中添加如下依赖信息来实现：

```
<dependency>
  <groupId>org.hdrhistogram</groupId>
  <artifactId>HdrHistogram</artifactId>
  <version>2.1.9</version>
</dependency>
```

下面来看一个使用 HdrHistogram 的简单例子，这个例子接收一个由数字组成的文件，为连续结果之间的差值计算 HdrHistogram：

```
public class BenchmarkWithHdrHistogram {
    private static final long NORMALIZER = 1_000_000;

    private static final Histogram HISTOGRAM
        = new Histogram(TimeUnit.MINUTES.toMicros(1), 2);

    public static void main(String[] args) throws Exception {
        final List<String> values = Files.readAllLines(Paths.get(args[0]));
```