

掌握 JMeter：性能测试

Apache JMeter 是一个功能强大的开源工具，主要用于性能测试，但也可以用于功能测试。它广泛用于测试网络应用、API、数据库和其他服务的性能，通过模拟多个用户并测量系统在负载下的响应情况。以下是如何有效使用 JMeter 的分步指南，从安装到运行基本测试并分析结果。

第 1 步：安装 JMeter

在使用 JMeter 之前，你需要在你的机器上安装它。以下是如何操作的：

1. **先决条件：** JMeter 是一个基于 Java 的应用程序，因此你必须安装 Java 开发工具包（JDK）或 Java 运行时环境（JRE）。推荐版本 8 或更高。你可以通过在终端或命令提示符中运行 `java -version` 来验证。如果没有安装 Java，请从 [官方 Java 网站](#) 下载并安装。
 2. **下载 JMeter：** 访问 Apache JMeter 网站 并下载最新的二进制版本（通常是 .zip 或 .tgz 文件）。
 3. **解压文件：** 将下载的文件解压到你选择的目录（例如，Windows 上的 `C:\JMeter` 或 Unix 系统上的 `~/JMeter`）。解压后的文件夹包含运行 JMeter 所需的所有文件。
 4. **启动 JMeter：**
 - 导航到解压目录中的 `bin` 文件夹（例如，`C:\JMeter\apache-jmeter-5.x\bin`）。
 - 运行相应的可执行文件：
 - 在 **Windows** 上：双击 `jmeter.bat` 或通过命令提示符运行。
 - 在 **Unix/Linux/macOS** 上：在终端中执行 `./jmeter.sh`。
 - 这将打开 JMeter 图形用户界面（GUI），你将在其中创建和管理你的测试计划。
-

第 2 步：创建测试计划

测试计划 在 JMeter 中定义了你测试的内容及其方式。它是性能测试的蓝图。以下是如何设置基本测试计划的方法：

添加线程组

1. 在 JMeter GUI 中，右键点击左侧面板中的 **测试计划** 节点，然后选择 **添加 > 线程（用户） > 线程组**。
2. 配置线程组：
 - **线程数（用户）：** 要模拟的虚拟用户数（例如，10）。

- **启动时间 (秒)**: JMeter 启动所有线程所需的时间 (例如, 10 秒表示每秒启动 1 个线程, 共 10 个线程)。
- **循环次数**: 每个线程重复测试的次数 (例如, 1 表示单次运行, 或选择“永远”以进行连续循环)。

线程组模拟用户流量。例如, 10 个线程、10 秒启动时间和 1 个循环意味着 10 个用户将在 10 秒内访问应用程序, 每个用户执行一次测试。

添加采样器

采样器定义 JMeter 发送到目标系统的请求。对于网页测试, 最常见的是 HTTP 请求采样器: 1. 右键点击线程组, 然后选择 **添加 > 采样器 > HTTP 请求**。2. 配置 HTTP 请求: - **协议**: http 或 https。- **服务器名称或 IP**: 目标系统的域名或 IP (例如, example.com)。- **端口号**: 通常是 HTTP 的 80 或 HTTPS 的 443 (如果是标准端口则留空)。- **方法**: GET、POST 等, 具体取决于请求类型。- **路径**: 特定的端点或页面 (例如, / 表示主页)。- 如果需要, 添加参数或请求体 (例如, 用于 POST 请求)。

这个采样器告诉 JMeter 每个虚拟用户应该执行的操作。

添加监听器

监听器收集并显示测试结果: 1. 右键点击线程组, 然后选择 **添加 > 监听器 > 查看结果树** (或其他监听器, 如 **摘要报告**)。2. **查看结果树** 显示每个请求的详细结果, 包括响应时间、状态代码和响应数据。

监听器对于分析应用程序在测试期间的性能至关重要。

保存测试计划

点击 **文件 > 另存为测试计划**, 并保存你的 .jmx 文件 (例如, mytest.jmx)。这允许你以后重用或修改它。

第 3 步: 运行测试

要执行你的测试: 1. 在 JMeter GUI 中, 点击工具栏上的绿色 **播放** 按钮 (▶), 或者选择 **运行 > 启动**。2. JMeter 将模拟线程组中定义的用户, 发送配置的 HTTP 请求。3. 观察结果在监听器 (例如, 查看结果树) 中填充, 测试运行时。

对于小型测试, 通过 GUI 运行是可以的。对于大型测试, 请参考下面的“高级使用”部分, 了解非 GUI 模式。

第 4 步：分析结果

测试完成后，使用监听器查看结果：- **查看结果树**：显示每个请求的成功/失败、响应时间和响应数据。- **摘要报告**：提供聚合指标，如平均响应时间、吞吐量（每秒请求数）和错误率。

这些指标帮助你评估应用程序的性能（例如，在负载下的响应速度或在特定用户数下的失败情况）。

示例：测试一个简单的网页

让我们测试 `example.com` 使用 10 个用户：1. 启动 JMeter。2. 添加一个线程组：- 线程：10 - 启动时间：10 秒 - 循环次数：13. 添加一个 HTTP 请求采样器：- 协议：http - 服务器名称：example.com - 方法：GET - 路径：/ 4. 添加一个查看结果树监听器。5. 保存并运行测试。6. 在查看结果树中检查结果，查看响应时间和状态代码（例如，200 OK）。

这个简单的测试测量了 `example.com` 在 10 个同时用户下的性能。

高级使用

对于更复杂的场景，JMeter 提供了额外的功能：

参数化

使用 **CSV 数据集配置** 从 CSV 文件中将不同的数据（例如，用户名、密码）输入到你的测试中：1. 将 **配置元素 > CSV 数据集配置** 添加到线程组。2. 指定文件路径和变量名（例如，`${username}`），然后在你的 HTTP 请求中使用这些变量。

关联

处理动态值（例如，会话 ID、令牌）：1. 在请求后添加 **后处理器 > 正则表达式提取器** 或 **JSON 提取器**。2. 将值（例如，令牌）提取到变量（例如，`${token}`）。3. 在后续请求中使用 `${token}`（例如，在头部或参数中）。

断言

验证响应：1. 将 **断言 > 响应断言** 添加到采样器。2. 检查特定文本、响应代码（例如，200）或使用 JSONPath 进行 API 测试。

非 GUI 模式

对于大型测试或自动化：- 从命令行运行 JMeter：jmeter -n -t mytest.jmx -l results.jtl -n：非 GUI 模式。--t：测试计划文件。--l：结果日志文件（例如，.jtl）。- 生成 HTML 报告：jmeter -n -t mytest.jmx -l results.jtl -e -o report_folder --e：生成报告。--o：HTML 报告的输出文件夹。

分布式测试

对于高负载：1. 在不同的机器上设置多个 JMeter 实例。2. 配置主机的 jmeter.properties 文件以包含从属 IP。3. 从主机运行测试以协调负载生成。

附加提示

- **支持的协议**：除了 HTTP，JMeter 还支持 FTP、JDBC、JMS 等，使其适用于各种系统。
- **服务器监控**：将 JMeter 与服务器端工具（例如，PerfMon）配对，以在测试期间监控 CPU、内存和数据库性能。
- **现实模拟**：添加 **HTTP Cookie 管理器**、**HTTP 缓存管理器** 或 **计时器** 以模拟浏览器行为或用户思考时间。
- **功能测试**：使用断言验证响应的正确性，而不仅仅是性能。

结论

要使用 JMeter：1. 使用 Java 安装并启动 GUI。2. 创建一个包含线程组、采样器和监听器的测试计划。3. 配置测试元素（用户、请求等）。4. 运行测试并分析结果。

对于简单的开始，可以像上述那样测试一个网页，例如 example.com。随着经验的积累，探索高级功能，如参数化、关联和分布式测试，以处理复杂的、现实世界的场景。JMeter 的灵活性和广泛的文档使其成为初学者和高级测试人员都非常有价值的工具。