

JMeter のマスター: パフォーマンステスト

Apache JMeter は、主にパフォーマンステストのために設計された強力なオープンソースツールですが、機能テストにも使用できます。ウェブアプリケーション、API、データベース、その他のサービスのパフォーマンスをテストするために、複数のユーザーをシミュレートし、システムが負荷下でどのように応答するかを測定するために広く使用されています。以下は、インストールから基本的なテストの実行と結果の分析まで、JMeter を効果的に使用するためのステップバイステップガイドです。

ステップ 1: JMeter のインストール

JMeter を使用する前に、まずマシンにインストールする必要があります。以下のようにします：

- 前提条件:** JMeter は Java ベースのアプリケーションなので、Java Development Kit (JDK) または Java Runtime Environment (JRE) がインストールされている必要があります。バージョン 8 以上が推奨されます。ターミナルまたはコマンドプロンプトで `java -version` を実行して確認できます。Java がインストールされていない場合は、公式 Java ウェブサイトからダウンロードしてインストールしてください。
 - JMeter のダウンロード:** Apache JMeter ウェブサイトにアクセスし、最新のバイナリリリース（通常は `.zip` または `.tgz` ファイル）をダウンロードします。
 - ファイルの展開:** ダウンロードしたファイルを好きなディレクトリに展開します（例：Windows では `C:\JMeter`、Unix ベースのシステムでは `~/JMeter`）。展開されたフォルダには、JMeter を実行するために必要なすべてのファイルが含まれています。
 - JMeter の起動:**
 - 展開されたディレクトリ内の `bin` フォルダに移動します（例：`C:\JMeter\apache-jmeter-5.x\bin`）。
 - 適切な実行ファイルを実行します：
 - Windows:** `jmeter.bat` をダブルクリックするか、コマンドプロンプトから実行します。
 - Unix/Linux/macOS:** ターミナルで `./jmeter.sh` を実行します。
 - これにより、JMeter のグラフィカルユーザーインターフェース (GUI) が開き、テストプランを作成および管理できます。
-

ステップ 2: テストプランの作成

JMeter の**テストプラン**は、テストの内容と方法を定義します。これは、パフォーマンステストの設計図です。以下のように基本的なテストプランを設定します：

スレッドグループの追加

1. JMeter GUI で、左側のペインの **Test Plan** ノードを右クリックし、**Add > Threads (Users) > Thread Group** を選択します。
2. スレッドグループを設定します：
 - **Number of Threads (Users)**: シミュレートする仮想ユーザーの数（例：10）。
 - **Ramp-Up Period (seconds)**: JMeter がすべてのスレッドを開始する時間（例：10 秒は 1 秒ごとに 1 スレッドが開始され、10 スレッドが開始されます）。
 - **Loop Count**: 各スレッドがテストを繰り返す回数（例：1 回の実行、または「Forever」をチェックして連続ループ）。

スレッドグループはユーザー トラフィックをシミュレートします。例えば、10 スレッドで 10 秒のランプアップと 1 回のループは、10 ユーザーが 10 秒間にアプリケーションにアクセスし、各ユーザーが 1 回テストを実行することを意味します。

サンプラーの追加

サンプラーは、JMeter がターゲットシステムに送信するリクエストを定義します。ウェブテストでは、最も一般的なものが HTTP リクエストサンプラーです：1. スレッドグループを右クリックし、**Add > Sampler > HTTP Request** を選択します。2. HTTP リクエストを設定します：- **Protocol**: http または https。- **Server Name or IP**: ターゲットシステムのドメインまたは IP（例：example.com）。- **Port Number**: 通常は HTTP の場合は 80、HTTPS の場合は 443（標準の場合は空白にしておく）。- **Method**: GET、POST など、リクエストの種類に応じて。- **Path**: 特定のエンドポイントまたはページ（例：ホームページの場合は /）。- 必要に応じてパラメータやリクエストボディを追加します（例：POST リクエスト）。

このサンプラーは、各仮想ユーザーが実行するアクションを JMeter に指示します。

リスナーの追加

リスナーはテスト結果を収集し表示します：1. スレッドグループを右クリックし、**Add > Listener > View Results Tree**（または他のリスナーである **Summary Report**）を選択します。2. **View Results Tree** は、各リクエストの詳細な結果を表示します。応答時間、ステータスコード、応答データなどが含まれます。

リスナーは、アプリケーションのテスト中のパフォーマンスを分析するために不可欠です。

テストプランの保存

File > Save Test Plan As をクリックし、.jmx ファイル（例：mytest.jmx）として保存します。これにより、後で再利用または変更できます。

ステップ 3: テストの実行

テストを実行するには：1. JMeter GUI で、ツールバーの緑の **Play** ボタン (►) をクリックするか、**Run > Start** を選択します。2. JMeter は、スレッドグループで定義されたユーザーをシミュレートし、設定された HTTP リクエストを送信します。3. テストが実行中に、リスナー（例：View Results Tree）に結果が表示されます。

小規模なテストでは、GUI を使用して実行するのが良いです。大規模なテストについては、以下の「高度な使用法」セクションで非 GUI モードについて説明します。

ステップ 4: 結果の分析

テストが完了したら、リスナーを使用して結果を確認します：- **View Results Tree**: 各リクエストの成功/失敗、応答時間、応答データを表示します。- **Summary Report**: 平均応答時間、スループット（リクエスト/秒）、エラー率などの集計メトリクスを提供します。

これらのメトリクスを使用して、アプリケーションのパフォーマンスを評価できます（例：負荷下での応答速度や特定のユーザー数での失敗）。

例: 簡単なウェブページのテスト

example.com を 10 ユーザーでテストします：1. JMeter を起動します。2. スレッドグループを追加します：- スレッド: 10 - ランプアップ: 10 秒 - ループカウント: 13. HTTP リクエストサンプラーを追加します：- プロトコル: http - サーバー名: example.com - メソッド: GET - パス: / 4. View Results Tree リスナーを追加します。5. 保存してテストを実行します。6. View Results Tree で応答時間とステータスコード（例：200 OK）を確認します。

この簡単なテストは、example.com が 10 人の同時ユーザーでどのように動作するかを測定します。

高度な使用法

より複雑なシナリオには、JMeter は追加の機能を提供します：

パラメータ化

CSV Data Set Config を使用して、CSV ファイルから異なるデータ（例：ユーザー名、パスワード）をテストに供給します：1. スレッドグループに **Config Element > CSV Data Set Config** を追加します。2. ファイルパスと変数名（例：\${username}）を指定し、HTTP リクエストでこれらの変数を使用します。

コレレーション

動的な値（例：セッション ID、トークン）を処理します：1. リクエストの後に **Post-Processor > Regular Expression Extractor** または **JSON Extractor** を追加します。2. 値（例：トークン）を変数（例： `${token}` ）に抽出します。3. 後続のリクエストで `${token}` を使用します（例：ヘッダーまたはパラメータ）。

アサーション

応答を検証します：1. サンプラーに **Assertion > Response Assertion** を追加します。2. 特定のテキスト、応答コード（例：200）、または API テストに JSONPath を使用してチェックします。

非 GUI モード

大規模なテストや自動化のため：- コマンドラインから JMeter を実行します：`jmeter -n -t mytest.jmx -l results.jtl --n: 非 GUI モード。--t: テストプランファイル。--l: 結果ログファイル（例： .jtl ）。-HTML レポートを生成します：jmeter -n -t mytest.jmx -l results.jtl -e -o report_folder --e: レポートを生成。--o: HTML レポートの出力フォルダ。`

分散テスト

高負荷のため：1. 異なるマシンに複数の JMeter インスタンスを設定します。2. マスターマシンの `jmeter.properties` ファイルにスレーブ IP を含めるように設定します。3. マスターからテストを実行して負荷生成を調整します。

追加のヒント

- **サポートされるプロトコル**: HTTP 以外にも、FTP、JDBC、JMSなどをサポートしており、さまざまなシステムに対応できます。
 - **サーバーモニタリング**: JMeter をサーバーサイドツール（例：PerfMon）と組み合わせて、テスト中の CPU、メモリ、データベースのパフォーマンスを監視します。
 - **リアルなシミュレーション**: **HTTP Cookie Manager**、**HTTP Cache Manager**、**Timers** を追加して、ブラウザの動作やユーザーの思考時間を模倣します。
 - **機能テスト**: アサーションを使用して応答の正確性を検証し、パフォーマンスだけでなく機能も確認します。
-

結論

JMeter を使用するには：1. Java と GUI をインストールして起動します。2. スレッドグループ、サンプラー、リスナーを含むテストプランを作成します。3. テスト要素（ユーザー、リクエストなど）を設定します。4. テストを実行し、結果を分析します。

簡単なスタートとして、上記のように `example.com` のウェブページをテストします。経験を積むにつれて、パラメータ化、コレレーション、分散テストなどの高度な機能を探索して、複雑なリアルなシナリオに対応します。JMeter の柔軟性と豊富なドキュメントは、初心者から高度なテスターまでにとって価値のあるツールです。