



負荷テストの極意 第三回

株式会社アシスト
矢野英也

イントロダクション

今回は、負荷テストの種類、および計画フェーズにおいて考えるべき点について、以下 3 点お話をさせていただきます。

- ・ 負荷テストの目的と種類
- ・ 負荷テスト環境(構成要素)について
- ・ 負荷テストにツールは必要か

今回は準備フェーズについてです。準備フェーズでは負荷テスト計画に沿って、テストスクリプト作成、テストシナリオの作成、テストデータの準備、監視項目の設定等の作業を行います。今回は以下の 4 点についてお話ししたいと思います。

- ・ テストスクリプトの作成
 - ・ テストシナリオの作成
 - ・ テストデータの準備
 - ・ 監視項目の設定
- ※本項目は負荷テストツールを使用することを前提としています。

テストスクリプト

テストスクリプトは計画フェーズで決められたテストケース(テスト対象のユーザ操作)をツールが自動化用に記録したものです。テストスクリプト作成時によくスクリプト作成開始のタイミングについて話題にあがります。開発からテストチームへアプリケーションがリリースされた後に開始するのがベストですが、スケジュールを考慮して、リリース前にスクリプト作成を開始する場合があります。その場合、アプリケーションの変更はリスクとして捉え、開発部門と変更箇所について綿密に共有した方がいいでしょう。(実際、画面タイトルが変更になって、画面タイトルをチェックする関数がエラーになったことがあります。)

またテストスクリプトを作成する際、以下の点を考慮します。

- ・ 思考遅延時間

思考遅延時間とは、人が考える時間をシミュレートするようテストスクリプトに遅延を発生させる機能です。思考遅延時間の有無によって、単位時間あたりの処理数が大きく変わります。

例)

思考遅延時間有り ⇒単位時間あたりの処理数:小

思考遅延時間無し ⇒単位時間あたりの処理数:大

- ・ 繰り返し対象の操作

ログイン、ログアウトを繰り返しの対象に含む、含まない等があります。例えば、業務アプリケーションのような特定ユーザを対象にした負荷テストにおいては、ユーザ数、実行処理数を正確に実行する必要があり、各ユーザのログイン、ログアウトは1回のみにして、業務処理は繰り返し実行します。コンシューマ向けアプリケーションのような不特定多数を対象にした負荷テストにおいては、ログイン、ログアウトも繰り返しの対象に含め、より多くのユーザが実行したかのような負荷テストを実施します。

- ・ エラー検出方法

HTTP ステータスコードだけだと、正常動作の範囲でない画面(例:ただいま、ページが込み合っています。)が表示された場合でも正常と見なしてしまいます。したがって HTTP のステータスコードだけでなく、画面タイトルなどで正常、異常を判断する場合があります。

- ・ パラメータ対象箇所

パラメータとは、同一スクリプトの入力データにあたる部分を変数にすることでスクリプトを繰り返し使いやすくするツールの機能のことで、(JSTQB の用語で言うと「データ駆動テスト」を実行する際に使用する機能です。)ログインアカウント、検索条件、登録内容等、同一データではなく値を変えて、より実利用に近いテストスクリプトを作成します。

補足:

テスト計画フェーズでの話になりますが、テストケースの範囲について話題になることがあります。機能的に重要なものを優先する場合がありますが、既存アプリケーションのバージョンアップ等であれば、運用中のアクセスデータを元にテスト範囲を決めるのもひとつの手段です。全てのユーザ操作をテスト範囲にするのは不可能ですので、ビジネスインパクト、運用データをもとに決めます。

テストシナリオ準備

テストシナリオは、ユーザ数、実行のタイミング、監視項目を設定等を設定したファイルです。テストシナリオを作成する際は、負荷テストの要件(例:秒あたり 100PV、分あたり 100 ログイン)を満たすように設計する必要があります。実際はリハーサル等でユーザ数、繰り返し回数、思考遅延時間を微調整しながら、目標をクリアするように進めますが、ある程度は見通しをもって設定しておいた方がいいでしょう。

監視項目の設定については、「監視項目の準備」にてお話いたします。

テストデータの準備

まずDB側のテストデータ量については、本番を想定したデータを準備します。テスト要件によっては、数年後を想定してデータを用意する場合があります。データの準備にあたっては、本番環境から抽出した実データをもとに、擬似データ変換、マスキング変換、ランダム変換機能を持ったツールを使ってテストデータを作成する、といったことも手段として考えられます。

テストスクリプトのパラメータデータについては、ユーザ数、繰り返し件数、テスト実施回数を考慮して、あらかじめデータを準備しておいた方がいいでしょう。

監視項目の準備

これも計画フェーズでの話になりますが、事前に監視項目、および判断基準を事前に決めておく必要があります。監視項目が多ければいいというものではなく、負荷テストの目的にあわせて、監視項目を決めた方がいいでしょう。監視項目は大きく分けて以下の4つに大別できます。

- ・ エンドユーザ視点
応答時間、PV、スループット、エラー件等
- ・ サーバ視点
CPU、メモリ、DISKIO、ネットワーク
- ・ アプリケーション視点
接続数、プロセス数、ヒープサイズ、GC回数等
- ・ アプリケーション内部視点
メモリーリーク、ロック競合等

私達が負荷テスト支援をさせていただく場合、基本的には「エンドユーザ視点」「サーバ視点」の監視を設定し、問題が発生した場合、さらに詳細情報を取得する、というアプローチをとっています。

以下、Linuxを例にしたサーバ視点の監視項目です。

カテゴリ	監視項目	判断基準	説明
CPU	CPU 使用率	平均 70%以内	CPU 時間あたりの占有比率。この値のみでは判断せず、実行可能キューの結果とあわせて評価する。
	実行可能キュー(実行されているものも含む)	システムの CPU 個数以下であること	システムの CPU 個数の 2~3 倍以上になった場合、CPU 過負荷と判断する。vmstat の r 値を確認する。
メモリ	Memory(free/buffers/cached)	各項目が枯渇しないこと	それぞれ vmstat の free 値、buffer 値、cache 値を確認する。
	Memory(swap 使用率)	頻繁に変動しないこと	Linux/UNIX の場合、メモリに swap 領域を確保しており、その使用率が初期値から変動しないことを確認する。Free コマンドを利用する。
ディスク	DISK I/O	I/O 待ちに費やした CPU 使用率時間の比率が 10%を超えないこと	10%を超える場合には、I/O 処理の制約でシステムに応答遅延が発生している可能性がある。vmstat の wa 値を確認する。また iostat コマンドを使ってボトルネックを特定する場合もある。

また監視データの収集方法については、さまざまな方法が考えられますが、後の分析のことを考えると、負荷テストツールでまとめて収集できた方がいいでしょう。

今回は準備フェーズについてお話させていただきました。準備フェーズは、利用する負荷テストツールによって、スクリプトの編集作業等の作業工数が大きく変わります。ツールの特性を理解した上で、スケジュールを設定されるかと思いますが。

次回は準備フェーズ第二弾です。ツールの重要性についてお話する予定です。