

HP LoadRunner

向け Windows® オペレーティングシステム

ソフトウェアバージョン: Service Pack 11.52

チュートリアル

ドキュメントリリース日: 2013 年 3 月 (英語版)

ソフトウェアリリース日: 2013 年 3 月 (英語版)



目次

チュートリアル	1
目次	2
HP LoadRunner チュートリアルへようこそ	8
LoadRunner チュートリアルへようこそ	8
LoadRunner コンポーネントについて	8
LoadRunner の用語について	9
LoadRunner 負荷テスト・プロセスについて	9
HP Web Tours の起動	10
アプリケーションのパフォーマンス要件の定義	11
次の練習について	11
練習 1: 仮想ユーザ・スクリプトの作成	12
Virtual User Generator (VuGen) の紹介	12
空の仮想ユーザ・スクリプトの作成	12
仮想ユーザ・スクリプトへの記録	14
仮想ユーザ・スクリプトの表示	16
次の練習について	19
練習 2: スクリプトの再生	20
仮想ユーザの実行時の動作の設定方法	20
仮想ユーザ・スクリプトの実行方法	23
再生に関する情報の表示方法	24
スクリプト再生成功の確認方法	26
再生結果の検索またはフィルタ方法	27
次の練習について	29
練習 3: 一般的な再生の問題の解決	30
再生エラーのための HP Web Tours の準備	30
一意のサーバ値の使用方法	31
次の練習について	33

練習 4: 負荷テスト用仮想ユーザ・スクリプトの準備	34
ビジネス・プロセスの実行時間を測定する方法	34
複数のユーザをエミュレートする方法	36
Web ページの内容の確認方法	39
デバッグ情報の生成方法	41
テストの再生成功の確認方法	42
次の練習について	43
練習 5: 負荷テスト・シナリオの作成	44
LoadRunner Controller の紹介	44
Controller の開始方法	44
Controller の概要	46
スクリプトの詳細の変更方法	47
システムで負荷を生成する方法	48
実際の負荷動作のエミュレート方法	50
さまざまなタイプのユーザのエミュレート方法	53
負荷下のシステムの監視方法	56
次の練習について	58
練習 6: 負荷テストの実行	59
Controller の実行ビューの概要	59
負荷テスト・シナリオの実行方法	60
負荷下のアプリケーションの監視方法	61
リアルタイムで実行しているユーザの監視方法	62
仮想ユーザ・アクションのサマリの表示場所	63
テスト中に負荷を増やす方法	63
負荷下のアプリケーションの動作方法	64
アプリケーションでエラーが発生したかどうか	65
シナリオの実行が完了したことを知る方法	66
システムが負荷のかかった状態で良好に実行されたかどうか	67
次の練習について	68
練習 7: シナリオの分析	69
Analysis セッションの処理方法	69
Analysis セッションの開始方法	70

Analysis のウィンドウの概要	70
ゴールの達成の確認(サービス・レベル・アグリーメント)	72
SLA の定義方法	72
パフォーマンス・サマリの表示	78
パフォーマンスのグラフによる表示方法	81
サーバのパフォーマンスの確認	83
テンプレートの保存	85
問題の原因の特定方法	86
シナリオ実行で収集可能なその他の情報	89
結果の発行方法	91
結論	93

ご注意

保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載で追加保証を意図するものは一切ありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR 12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

© Copyright 1993-2013 Hewlett-Packard Development Company, L.P.

商標について

Adobe™は、Adobe Systems Incorporated (アドビシステムズ社)の商標です。

Microsoft®およびWindows®は、米国におけるMicrosoft Corporationの登録商標です。

UNIX®は、The Open Groupの登録商標です。

謝辞

本製品には、OpenSSL ツールキットを利用するために OpenSSL プロジェクト (<http://www.openssl.org/>(英語 サイト)) によって開発されたソフトウェアが含まれます

本製品には、Eric Young (eay@cryptsoft.com) によって作成された暗号化ソフトウェアが含まれます

本製品には、Tim Hudson (tjh@cryptsoft.com) によって作成されたソフトウェアが含まれます

ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ソフトウェアバージョンの番号は、ソフトウェアのバージョンを示します。
- ドキュメントリリース日は、ドキュメントが更新されるたびに更新されます。
- ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

更新状況、およびご使用のドキュメントが最新版かどうかは、次のサイトで確認できます。

<http://support.openview.hp.com/selfsolve/manuals>

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの登録は、次のWebサイトから行なうことができます。

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

または、HP Passport のログインページの [**New users - please register**] リンクをクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

サポート

HPソフトウェアサポートオンラインWebサイトを参照してください。

<http://support.openview.hp.com>

このサイトでは、HPのお客様窓口のほか、HPソフトウェアが提供する製品、サービス、およびサポートに関する詳細情報をご覧いただけます。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様のビジネスを管理するのに必要な対話型の技術サポートツールに、素早く効率的にアクセスできます。HPソフトウェアサポートのWebサイトでは、次のようなことができます。

- 関心のあるナレッジドキュメントの検索
- サポートケースの登録とエンハンスメント要求のトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HPサポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

一部のサポートを除き、サポートのご利用には、HP Passportユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDを登録するには、次のWebサイトにアクセスしてください。

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

アクセスレベルの詳細については、次のWebサイトをご覧ください。

http://support.openview.hp.com/access_level.jsp

HP LoadRunner チュートリアルへようこそ

LoadRunner チュートリアルへようこそ

LoadRunner チュートリアルへようこそ。このチュートリアルは、自分のペースで学習できる、印刷可能なガイドです。負荷テストのすべてのプロセスについて説明しており、LoadRunner テスト環境について理解を深めることができます。

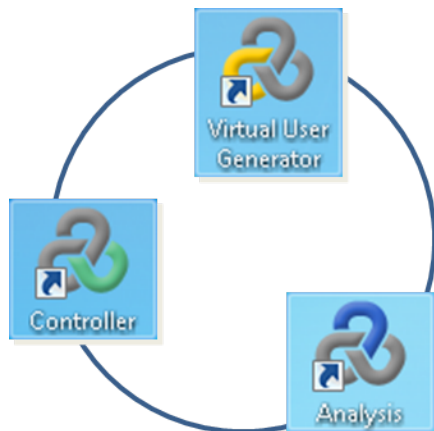
- この練習では、LoadRunner とテスト概念について説明します。
- 練習 1 から 4 では、VuGen(Virtual User Generator) を使用して仮想ユーザ・スクリプトを作成する方法について説明します。
- 練習 5 と 6 では、LoadRunner Controller を使用して負荷テストを設計し、実行する方法について説明します。
- 練習 7 では、分析ツールである LoadRunner Analysis について説明します。負荷テストの分析に役立つグラフやレポートの作成方法について解説します。

このチュートリアルを終えると、ご自分のシステムを対象に簡単な負荷テストを設計、実行、監視できるようになります。このチュートリアルは、構成されている順序どおりに進めることをお勧めします。

LoadRunner コンポーネントについて

LoadRunner には、次のコンポーネントがあります。

- **Virtual User Generator(VuGen)** は、エンドユーザのビジネス・プロセスをキャプチャし、仮想ユーザ・スクリプトとも呼ばれる自動パフォーマンス・テスト・スクリプトを作成します。
- **Controller** は、負荷テストの編成、実行、管理、監視を行います。
- **Analysis** は、負荷テストの結果の表示、分析、比較を行うのに役立ちます。



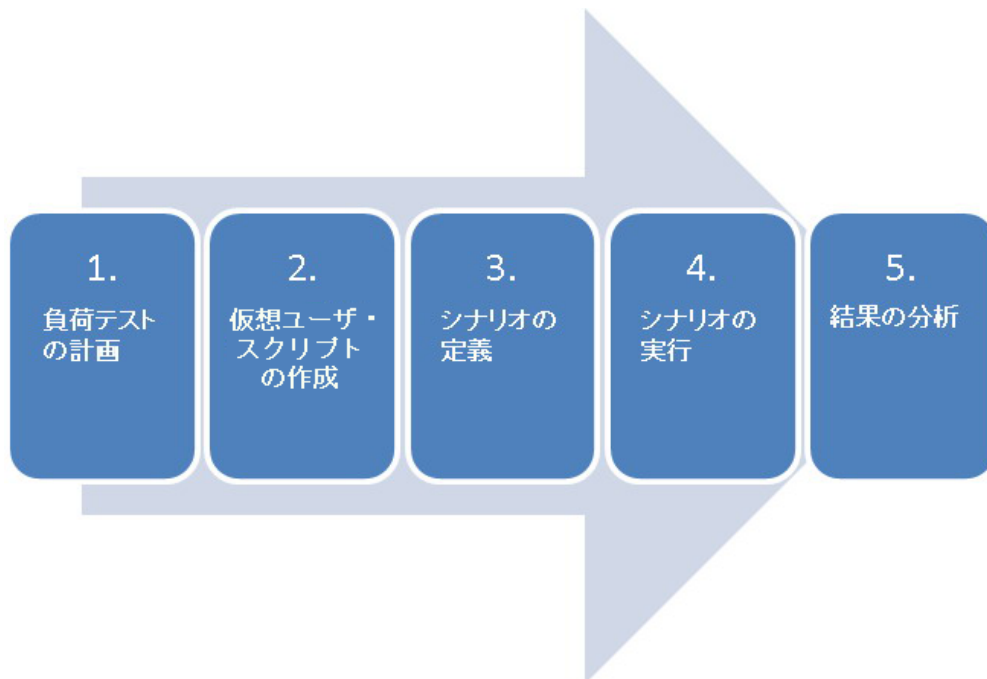
- **Load Generator** は、仮想ユーザを実行してシステムで負荷を生成するコンピュータです。

LoadRunner の用語について

- シナリオは、パフォーマンス要件に基づいて、テスト・セッション中に発生するイベントを定義します。
- シナリオの中で、LoadRunner は実際のユーザを**仮想ユーザ (Vuser)** で置き換えます。仮想ユーザは、システムを操作する実際のユーザのアクションをエミュレートします。シナリオには、数十、数百、あるいは数千もの仮想ユーザを含めることができます。
- **仮想ユーザ・スクリプト**には、シナリオの中で仮想ユーザが実行するアクションが記述されます。
- システムのパフォーマンスを測定するには、**トランザクション**を定義します。トランザクションは、1つ以上のエンドユーザのビジネス・プロセスを表しています。トランザクションによって、こうしたビジネス・プロセスを実行するのにかかる時間を測定できます。
- 各仮想ユーザ・スクリプトは**プロトコル**に基づきます。プロトコルは、クライアントとサーバ間の通信方法です。
- 仮想ユーザ・スクリプトの**スクリプト領域**は、Load Generator が仮想ユーザ・スクリプトを実行するために必要とするさまざまなリソースの量によって定義されます。一般的なリソースには、メモリ、CPU パワー、ディスク領域などが含まれます。

LoadRunner 負荷テスト・プロセスについて

LoadRunner での負荷テストは、一般に5つの段階(計画、スクリプト作成、シナリオ定義、シナリオ実行、結果分析)で構成されます。



1. **負荷テストの計画**:同時ユーザ数、一般的なビジネス・プロセス、必要な応答時間など、パフォーマンス・テスト要件を定義します。
2. **仮想ユーザ・スクリプトの作成**:VuGenを使用してエンドユーザの操作を自動スクリプトにキャプチャします。

3. **シナリオの定義**: Controller を使用して、負荷テスト環境をセットアップします。
4. **シナリオの実行**: Controller を使用して負荷テストの実行、管理、監視を行います。
5. **結果の分析**: LoadRunner Analysis を使用して、グラフおよびレポートを作成し、システムパフォーマンスを評価します。

HP Web Tours の起動

負荷テストのソリューションとして LoadRunner を説明するために、本チュートリアルでは、サンプル・アプリケーション用のパフォーマンス要件を使用します。サンプル・アプリケーション *HP Web Tours* は、Web ベースの旅行代理店システムです。*HP Web Tours* のユーザは、Web サーバに接続し、フライトを検索して予約し、フライト日程を確認します。

LoadRunner では 50 種類以上のアプリケーションをサポートしていますが、本チュートリアルでは、Web ベース・アプリケーションの負荷テストを行う方法について説明します。Web ベースではないアプリケーションの負荷テストを行う場合は、HP にご相談ください。

チュートリアルの本項では、*HP Web Tours* を起動してログオンする方法を学習します。

1. サンプル Web サーバを起動します。

[スタート]>[すべてのプログラム]>[HP ソフトウェア]>[HP LoadRunner]>[Samples]>[Web]>[Start Web Server]を選択します。

[Start Web Server]ダイアログ・ボックスが開きます。このダイアログ・ボックスを開いたまま、HP Web Tours アプリケーションにアクセスします。

2. HP Web Tours を起動します。

[スタート]>[すべてのプログラム]>[HP ソフトウェア]>[HP LoadRunner]>[Samples]>[Web]>[HP Web Tours Application]を選択します。ブラウザが開き、*HP Web Tours* のホーム・ページが表示されます。

注: LoadRunner がコンピュータの標準フォルダにインストールされていることを確認します。LoadRunner が標準でないフォルダにインストールされている場合、HP Web Tours アプリケーションは起動しません。

注: HP Web Tours アプリケーションには、Java がインストールされているブラウザが必要です。詳細については、関連する Java のドキュメントを参照してください。

3. HP Web Tours にログインします。
 - a. 次の情報を入力します。
 - ユーザ名: **jojo**
 - パスワード: **bean**
 - b. 左側のペインで **[Login]** をクリックします。HP Web Tours のようこそページが表示されます。
4. 航空券を予約します。
 - a. 左側のペインで **[Flights]** をクリックします。[Find Flight] ページが開きます。
 - b. **[Arrival city]** を **[Los Angeles]** に変更します。
 - c. **[Continue]** をクリックします。

- d. [Find Flight] ページで、標準の航空券の選択をそのまま受け入れて[Continue]をクリックします。
 - e. [Payment Details] ページで、[Continue]をクリックします。[Invoice] ページが表示されます。このページでは、航空券の予約が要約されています。
5. HP Web Tours セッションを終了します。
- 左側のペインで[Sign Off]をクリックしてログオフします。

アプリケーションのパフォーマンス要件の定義

これで、HP Web Tours に慣れたところで、HP Web Tours がビジネス・ニーズを満たすかどうかの承認を担当するパフォーマンス・エンジニアになったと想定してください。プロジェクト・マネージャは、リリースのための4つの条件をあなたに伝えてきました。

1. HP Web Tours では、10社の同時旅行代理店を正常に処理する必要があります。
2. HP Web Tours では、90秒を超えない応答時間で、同時に10件のフライト予約を処理できなければなりません。
3. HP Web Tours では、120秒を超えない応答時間で、日程の確認を同時に実行している10社の旅行代理店を処理できなければなりません。
4. HP Web Tours では、10秒を超えない応答時間で、システムのサイン・インおよびサイン・アウトを行っている10社の旅行代理店を処理できなければなりません。

リリース前に成功か失敗か判断できるように、本チュートリアルでは、各ビジネス要件について確認する負荷テストの作成プロセスについて説明します。

次の練習について

この練習で、LoadRunnerの基本を説明しました。次に、「練習 1: 仮想ユーザ・スクリプトの作成」(12ページ)に進みます。

練習 1: 仮想ユーザ・スクリプトの作成

システムで負荷を生成するには、最初に実際のユーザの振る舞いをエミュレートするために実行できる仮想ユーザ・スクリプトを作成します。仮想ユーザ・スクリプトの構築および作成には、VuGenを使用します。

この練習は、次の項目で構成されています。

- Virtual User Generator(VuGen) の紹介
- 空の仮想ユーザ・スクリプトの作成
- 仮想ユーザ・スクリプトへの記録
- 仮想ユーザ・スクリプトの表示

Virtual User Generator(VuGen) の紹介

パフォーマンス・テスト環境において、LoadRunnerは、実際のユーザを仮想ユーザ(**Vuser**とも呼ばれます)で置き換えます。仮想ユーザは、再現可能かつ予測可能な方法で一般的なユーザ・アクションをエミュレートして、システムに負荷を生成します。

仮想ユーザ・スクリプトを作成するには、VuGen(LoadRunner の Virtual User Generator)を使用します。VuGenでは、記録して再生するという方法が使用されます。アプリケーションでビジネス・プロセスを実行すると、VuGenはアクションを記録し、こうしたアクションを仮想ユーザ・スクリプトのステップに変換します。これらの仮想ユーザ・スクリプトは、負荷テストの基礎となります。

空の仮想ユーザ・スクリプトの作成

仮想ユーザ・スクリプトを作成するには、最初にVuGenを起動し、空のスクリプトを作成します。その後、イベントを記録し、手動拡張をスクリプトに追加することによって、空のスクリプトを拡張できます。

本項では、VuGenを起動して、**Web - HTTP/HTML** プロトコルに基づく空の仮想ユーザ・スクリプトを作成します。

定義: プロトコルは、クライアントとサーバ間の通信方法です。

空の仮想ユーザ・スクリプトを作成するには、次の手順を実行します。

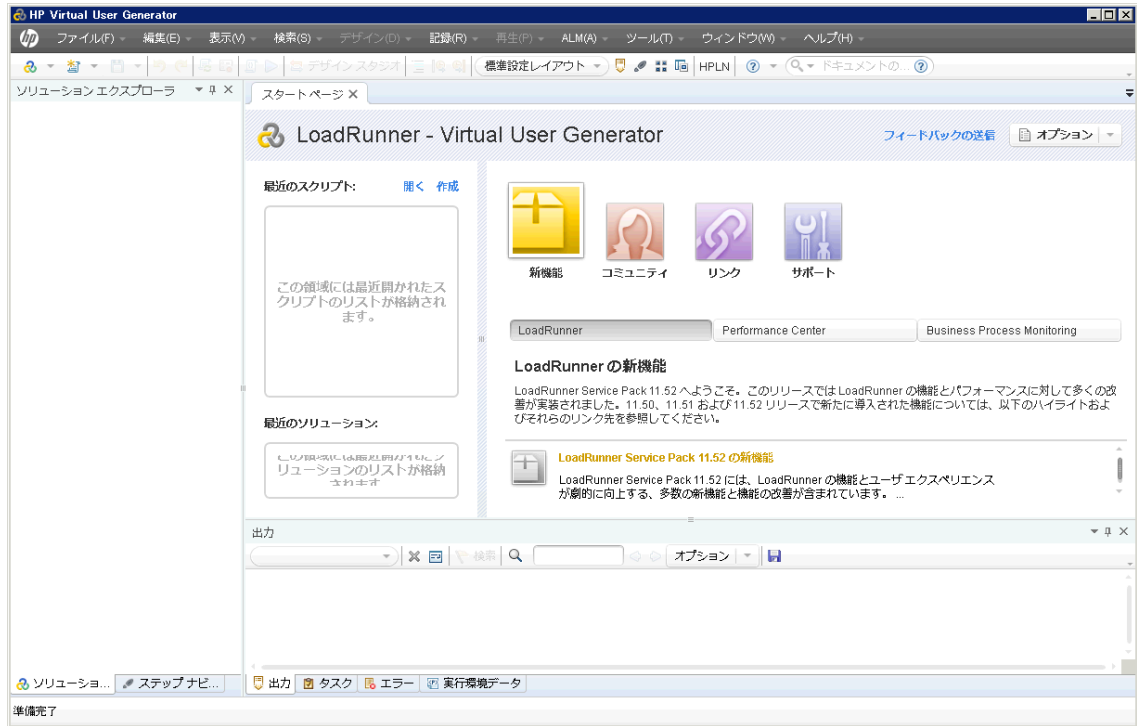
1. [スタート]>[すべてのプログラム]>[HP ソフトウェア]>[HP LoadRunner]>[VuGen]をクリックするか、デスクトップで[Virtual User Generator]ショートカット・アイコンをダブルクリックします。




HP Virtual User Generator(VuGen) が起動します。

チュートリアル

練習 1: 仮想ユーザ・スクリプトの作成



2. [ファイル]>[新規スクリプトおよびソリューション]をクリックするか、VuGen ツールバーの[新規スクリプトの追加]ボタン  をクリックします。[新規スクリプトの作成]ダイアログ・ボックスが開きます。




3. [カテゴリ]が[シングルプロトコル]であることを確認します。VuGen では、シングルプロトコル・スクリプトで利用可能なプロトコルのリストが表示されます。
4. 利用可能なプロトコルのリストから、[Web - HTTP/HTML]を選択して[作成]をクリックします。VuGen が空の仮想ユーザ・スクリプトを作成し、そのスクリプトを VuGen エディタに表示します。

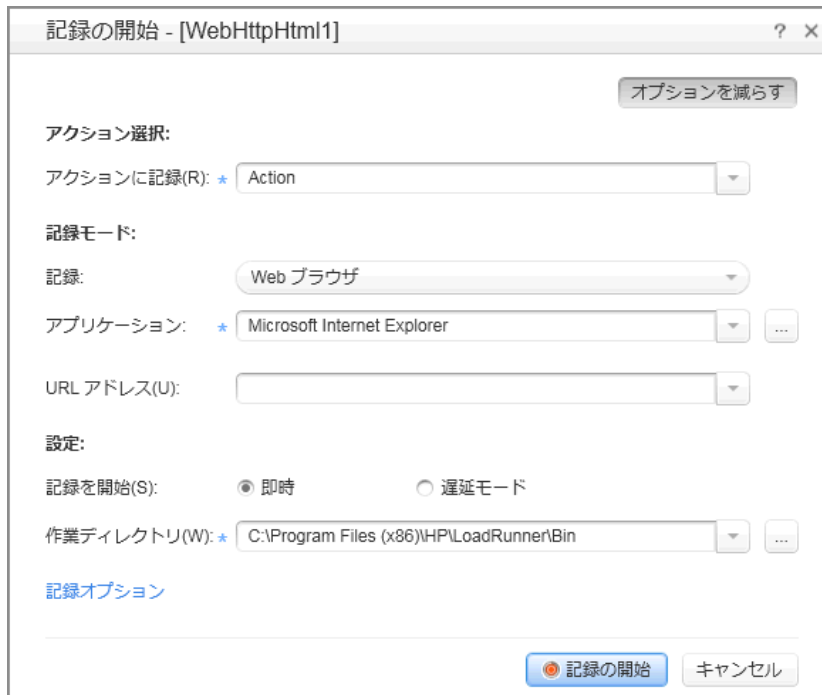
仮想ユーザ・スクリプトへの記録

仮想ユーザ・スクリプトの開発における次の手順は、実際のユーザによって実行されるアクションの記録です。前の項では、空の Web - HTTP/HTML 仮想ユーザ・スクリプトを作成しました。これで、アクションを直接スクリプトに記録できるようになりました。本項では、デジタールからロサンゼルスへのフライトを予約し、フライト日程を確認する旅行代理店のアクションを追跡します。

注: HP Web Tours アプリケーションで実行するアクションを VuGen で記録できるようにするには、[記録]>[記録オプション]をクリックします。[記録オプション]ダイアログ・ボックスで、[一般]>[スクリプト]を選択し、[スクリプト編集オプション]で[COM ローカル サーバとして作成されたプロセスを追跡する]チェック・ボックスをクリアされていることを確認します。

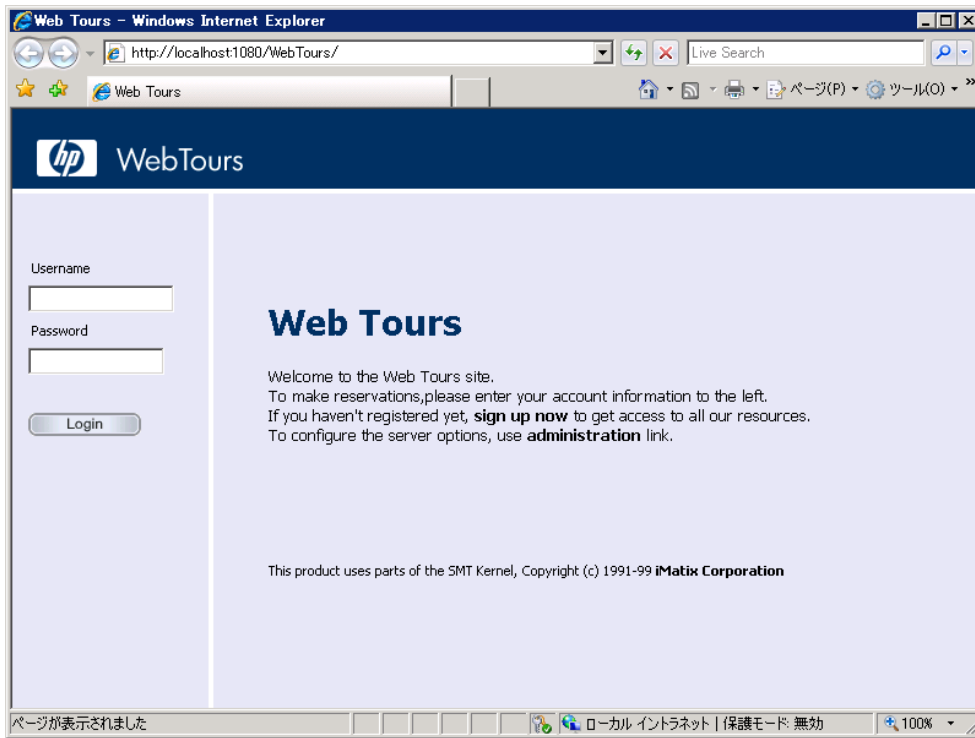
仮想ユーザ・スクリプトを記録するには、次の手順を実行します。

1. HP Web Tours Web サイトで記録を開始します。
 - a. [記録]>[記録]をクリックするか、VuGen ツールバーの[記録]ボタン  をクリックします。[記録の開始]ダイアログ・ボックスが開きます。



- b. [アクションに記録]ボックスで、[Action]を選択します。
- c. [記録]の一覧で、[Web ブラウザ]を選択します。

- d. [アプリケーション] ボックスに、**Microsoft Internet Explorer** が表示されていることを確認します。
- e. [URL アドレス] ボックスに「**http://localhost:1080/WebTours**」と入力します。
- f. [記録の開始] をクリックします。新しい Web ブラウザが開き、HP Web Tours ホーム・ページが表示されます。




注: HP Web Tours を開く際にエラーが発生した場合は、Web Tours Server が動作しているか確認します。[Start Web Server] ダイアログ・ボックスが表示された場合、サーバは動作しています。サーバを起動するには、[スタート]>[すべてのプログラム]>[HP ソフトウェア]>[HP LoadRunner]>[Samples]>[Web]>[Start Web Server] を選択します。

VuGen のフローティング記録ツールバーが開きます。



2. HP Web Tours にログインします。
 - a. 次の情報を入力します。
 - ユーザ名 : **jojo**
 - パスワード : **bean**
 - b. 左側のペインで[Login] をクリックします。HP Web Tours のようこそページが表示されます。
3. フライトの詳細を入力します。

- a. [Flights] をクリックします。[Find Flight] ページが開きます。
 - b. [Departure City] で [Denver] (標準設定) を選択します。
 - c. [Departure Date]: 標準設定のままとします。明日の日付になります。
 - d. [Arrival City] で [Los Angeles] を選択します。
 - e. [Return Date]: 標準設定のままとします。明後日の日付になります。
 - f. [Seating Preference] で [Aisle] を選択します。
 - g. その他の設定については標準の設定をそのまま受け入れて [Continue] をクリックします。
[Find Flight] ページが開きます。
4. フライトを選択します。
- 標準のフライト設定をそのまま受け入れて [Continue] をクリックします。[Payment Details] ページが開きます。
5. 支払い情報を入力し、フライトを予約します。
- a. [Credit Card] ボックスで「12345678」と入力します。
 - b. [Exp Date] ボックスで「06/14」と入力します。
 - c. [Continue] をクリックします。[Invoice] ページが開き、インボイスが表示されます。
6. 左側のペインで [Itinerary] をクリックします。[Itinerary] ページが開きます。
7. 左側のペインで [Sign Off] をクリックします。
8. ブラウザを閉じてから、VuGen フローティング・ツールバーの [記録停止] ボタン  をクリックして、記録プロセスを停止します。
- VuGen は必要なコードを生成し、そのコードを仮想ユーザ・スクリプトに挿入します。
- デザイン・スタジオが起動した場合、[閉じる] をクリックしてデザイン・スタジオを閉じます。
9. 仮想ユーザ・スクリプトを保存します。
- a. [ファイル] > [名前を付けてスクリプトを保存] を選択します。
 - b. <LoadRunner のインストール先フォルダ>\tutorial に移動し、「Scripts」という名前の新しいフォルダを作成します。その新しい「Scripts」フォルダに移動します。
 - c. [ファイル名] ボックスに「basic_tutorial」と入力します。
 - d. [保存] をクリックします。VuGen によってスクリプトが保存され、VuGen タイトル・バーにスクリプト名が表示されます。

仮想ユーザ・スクリプトの表示

前の項では、ログイン、フライト予約、日程の確認、ログオフといった旅行代理店によるアクションを記録しました。VuGen には、[記録開始] ボタンをクリックしてから [記録停止] ボタンをクリックするまでのステップが記録されました。


これで、VuGen を使用してスクリプトを表示できます。VuGen では、仮想ユーザ・スクリプトをさまざまな形式で表示できます。

- ソリューション・エクスプローラでは、仮想ユーザ・スクリプトのさまざまな部分だけでなく、仮想ユーザ・スクリプトに関連付けられた多数のファイルへの構造化アクセスが可能です。




- ステップ・ナビゲータでは、スクリプトのアイコン・ベースのビューが表示され、仮想ユーザのアクションがステップとして一覧表示されます。記録時に実行したアクションごとに、VuGenは該当するステップをステップ・ナビゲータで生成します。



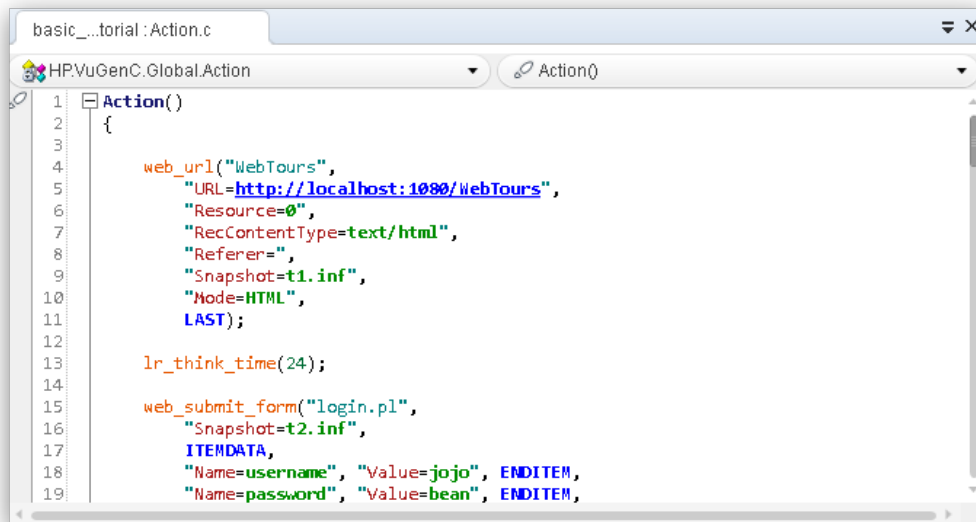
ステップ・ナビゲータではスナップショット・アイコン  が表示され、特定のステップにスナップショットが含まれることを示します。



スクリプトをステップ・ナビゲータで表示するには、[表示]>[ステップ ナビゲータ]を選択する

か、VuGen ツールバーの[ステップ ナビゲータ]ボタン  をクリックします。ステップ・ナビゲータで任意のステップをダブルクリックすると、該当する関数がエディタで表示されます。

- VuGen **エディタ**には、スクリプトのテキスト・ベースのビューが表示されます。エディタには、仮想ユーザのアクションがAPI 関数として一覧表示されます。エディタでは、VuGen によってスクリプトの関数とその引数値が色分けして表示されます。C 言語、LoadRunner API 関数、または制御フロー・ステートメントを直接スクリプトに入力できます。



```
basic_tutorial : Action.c
HP:VuGenC.Global.Action
Action()
1 Action()
2 {
3
4     web_url("WebTours",
5         "URL=http://localhost:1080/webTours",
6         "Resource=0",
7         "RecContentType=text/html",
8         "Referer=",
9         "Snapshot=t1.inf",
10        "Mode=HTML",
11        LAST);
12
13    lr_think_time(24);
14
15    web_submit_form("login.pl",
16        "Snapshot=t2.inf",
17        ITENDATA,
18        "Name=username", "Value=jojo", ENDITEM,
19        "Name=password", "Value=bean", ENDITEM,
```

次の練習について

これで、基本となる仮想ユーザ・スクリプトの記録および表示の学習が終了し、「練習 2: スクリプトの再生」(20ページ)に進む準備ができました。

練習 2: スクリプトの再生

前の練習では、一連の一般的なユーザ・アクションを記録して、VuGen がこうしたアクションを仮想ユーザ・スクリプトに記録しました。これで、記録されたスクリプトを再生し、スクリプトを負荷テスト・シナリオに組み込む前に、スクリプトが正しく実行されるか検証できます。

スクリプトを再生する前に、仮想ユーザの動作を定義するスクリプトの実行環境を設定する必要があります。

この練習は、次の項目で構成されています。

- 実行時の動作の設定方法
- 仮想ユーザ・スクリプトの実行方法
- 再生に関する情報の表示方法
- スクリプト再生成功の確認方法
- 結果の検索またはフィルタ方法

仮想ユーザの実行時の動作の設定方法

LoadRunner 実行環境設定では、さまざまな種類にわたるユーザの操作や動作をエミュレートできます。たとえば、サーバの出力にすぐに対応するユーザをエミュレートすることも、作業を停止して考えてから対応するユーザをエミュレートすることもできます。また実行環境設定では、仮想ユーザが仮想ユーザ・スクリプトの中でアクションを反復する回数も指定できます。

一般的な実行環境設定と、特定の仮想ユーザ・プロトコルのみ固有の設定があります。たとえば、Web エミュレーションの場合、Firefox、Chrome、または Internet Explorer でスクリプトを再生するように仮想ユーザを設定できます。特定のプロトコルの実行環境設定については、「[練習 4: 負荷テスト用仮想ユーザ・スクリプトの準備](#)」(34ページ)を参照してください。この練習では、すべての仮想ユーザ・プロトコルに適用される一般的な実行環境設定についていくつか説明します。一般的な実行環境設定には、以下のようなものがあります。

- **[実行論理]**: 仮想ユーザが仮想ユーザ・スクリプトのさまざまなセクションを反復する回数。
- **[ペースの設定]**: 反復の間隔。
- **[思考遅延時間]**: スクリプトのステップ間で仮想ユーザが考えるために動作を停止する時間。
- **[ログ]**: 再生中に収集する情報のレベル。

注: この練習では、**VuGen** を使用して実行環境設定を変更する方法について説明します。以降の練習で、**LoadRunner Controller** を使用して実行環境設定を変更する方法について説明します。

実行環境を設定するには、次の手順を実行します。

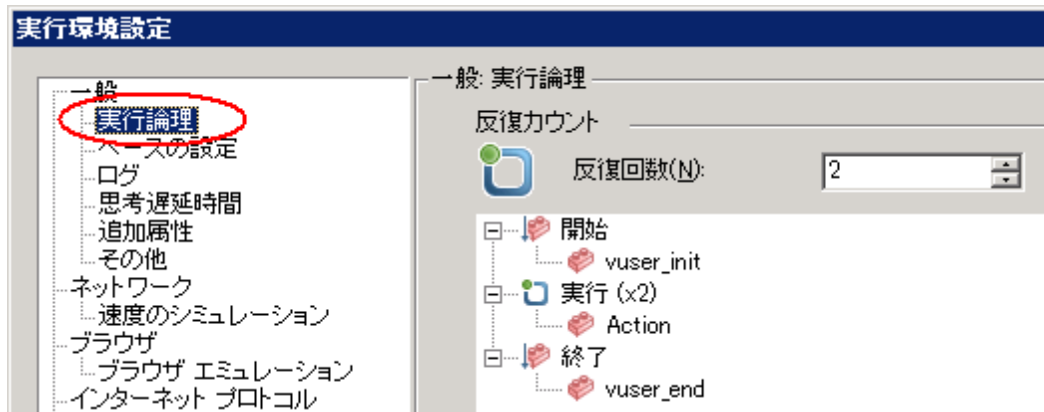
1. 前の練習で保存した仮想ユーザ・スクリプトを開きます。
2. **[実行環境設定]** ダイアログ・ボックスを開きます。

[再生] > **[実行環境設定]** をクリックします。**[実行環境設定]** ダイアログ・ボックスが開きます。



3. 実行論理を設定します。

左側のペインで、[一般]の下に[実行論理]をクリックします。

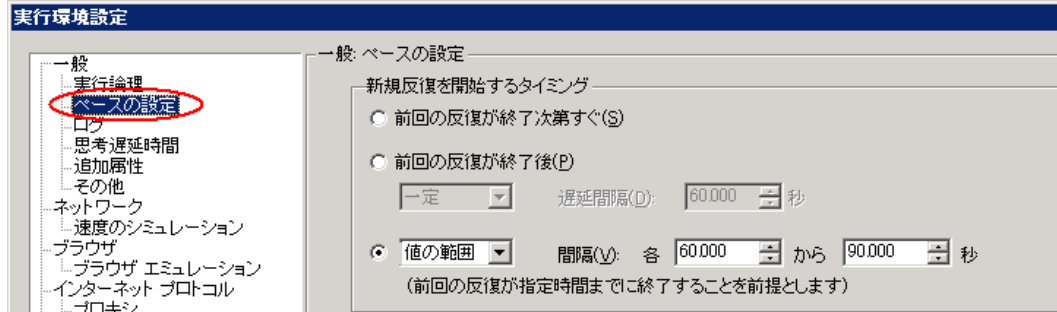


[実行論理]設定では、仮想ユーザ・スクリプトの反復回数を設定できます。これは、スクリプトを再生したときの仮想ユーザ・スクリプトのActionセクションの反復回数です。

[反復回数]を2に設定します。

4. ペースを設定します。

左側のペインで、[一般]の下に[ペースの設定]をクリックします。



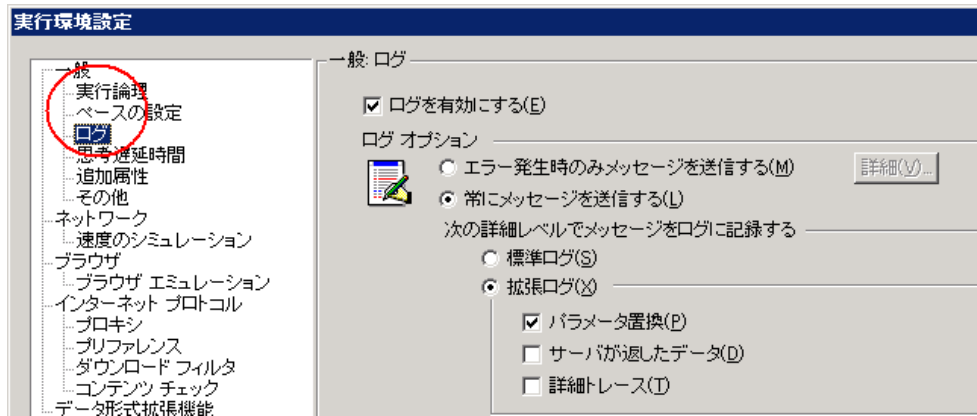
[ペースの設定]では、反復の間隔を制御できます。ランダムな間隔を指定します。これにより、動作間でユーザが待機する現実の設定が正確にエミュレートされます。たとえば、反復アクション間で実際のユーザが正確に60秒待機するとは限りません。

3番目のラジオボタンを選択し、次のように選択します。

ランダムな間隔、60.000秒から90.000秒ごと。

5. ログを設定します。

左側のペインで、[一般]の下の[ログ]をクリックします。

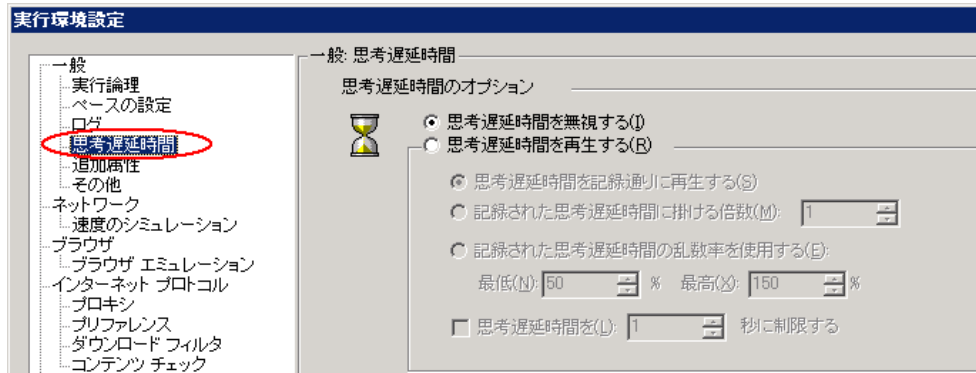


[ログ]設定では、仮想ユーザ・スクリプトの実行時にログに記録する情報量を指定します。仮想ユーザ・スクリプトの作成中には、デバッグのためにいくつかのログを有効にできます。ただし、スクリプトが正しく動作することを確認したら、エラー・ログのみを有効にするか、ログを無効することもできます。

[拡張ログ]を選択し、[パラメータ置換]を有効にします。このオプションは次の練習に関係します。詳細については、次の練習で説明します。

6. 思考遅延時間の設定を開きます。

左側のペインで、[一般]の下の[思考遅延時間]をクリックします。



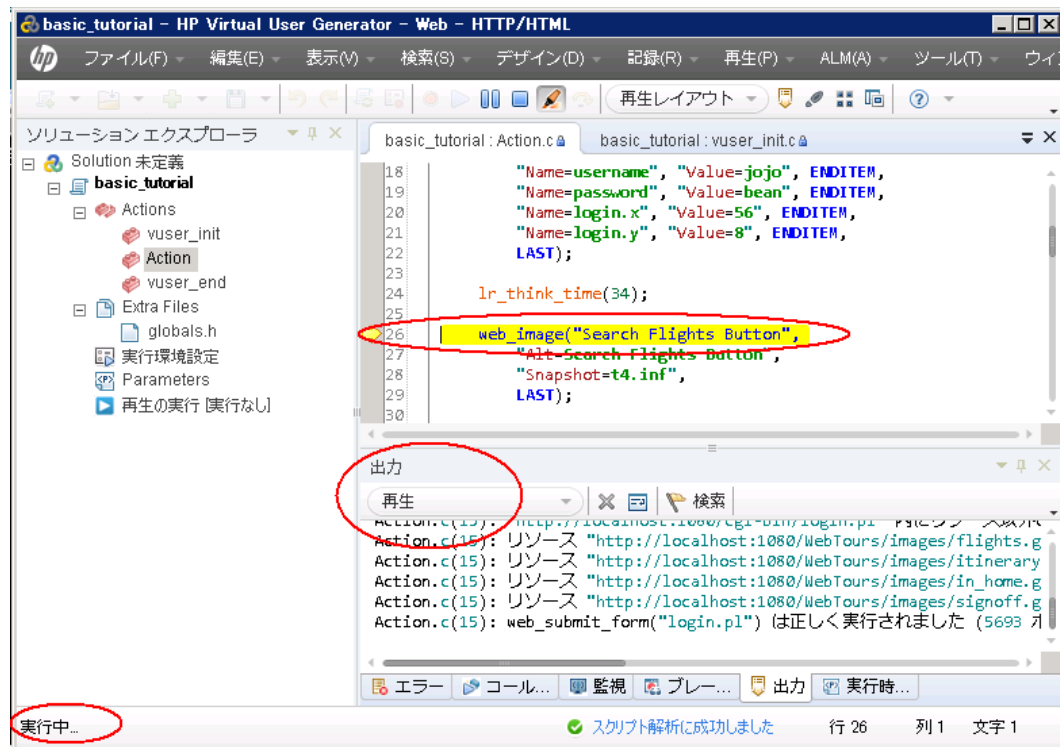
思考遅延の設定は標準の[思考遅延時間を無視する]のままにします。思考遅延時間は、Controllerで設定します。VuGenでスクリプトを実行する場合は、思考遅延時間が含まれないため迅速に実行される点に注意してください。

7. [OK]をクリックして、[実行環境設定]ダイアログ・ボックスを閉じます。


仮想ユーザ・スクリプトの実行方法

仮想ユーザ・スクリプトを記録して実行環境を設定すると、スクリプトを実行できるようになります。VuGenには、スクリプトが実行していることを示すインジケータが多数用意されています。

1. メッセージ「実行中」がVuGenの左下に表示されます。
2. 黄色いマーカがエディタに表示され、特定の行が再生されていることを示します。
3. 再生の進行中、[出力]ペインにメッセージが表示されます。



仮想ユーザ・スクリプトを実行するには、次の手順を実行します。

1. 仮想ユーザ・スクリプトを VuGen で開きます。
2. [再生]>[実行]をクリックするか、VuGen ツールバーの[再生]ボタン  をクリックします。
再生の完了後、相関を検索するように求めるメッセージ・ボックスが表示される場合があります。
[いいえ]をクリックします。

再生に関する情報の表示方法

仮想ユーザ・スクリプトの実行が停止したら、再生のサマリを表示できます。再生のサマリは[再生のサマリ]タブに表示されます。



WebHttpHtml1 : 再生のサマリ × WebHttpHtml1 : vuser_end.c WebHttpHtml1 : vuser_init.c

スクリプト名: WebHttpHtml1

 スクリプト成功

経過時間: 00:45.096 mm:ss.ms 開始日時: 2013/05/24 13:26:06 思考遅延時間: 00:00.000 mm:ss.ms
終了日時: 2013/05/24 13:26:51 浪費時間: 00:00.288 mm:ss.ms

結果範囲の選択:

再生統計サマリ: 平均

名前	値
合計ヒット数	14
合計スループット(バイト数)	19659
新規接続	3
接続のシャットダウン	3


詳細の参照先:
[テスト結果](#)
[再生ログ](#)

[再生のサマリ]タブでは、再生の間隔、再生の開始時間と終了時間など、スクリプト実行に関する基本的な情報が一覧表示されます。さらに、[再生のサマリ]タブにはスクリプト・イベントのログを表示するリンクと、スクリプト実行の詳細な結果を表示するもう1つのリンクがあります。

再生ログとは、スクリプトの再生中に起こったイベントのログであり、VuGen の[出力]ペインに表示されます。[出力]ペインでは再生中に起こったイベントのテキスト・サマリが色分けして表示されます。

チュートリアルの本項では、再生ログを開き、ログ内の特定のイベントおよび通知を確認します。

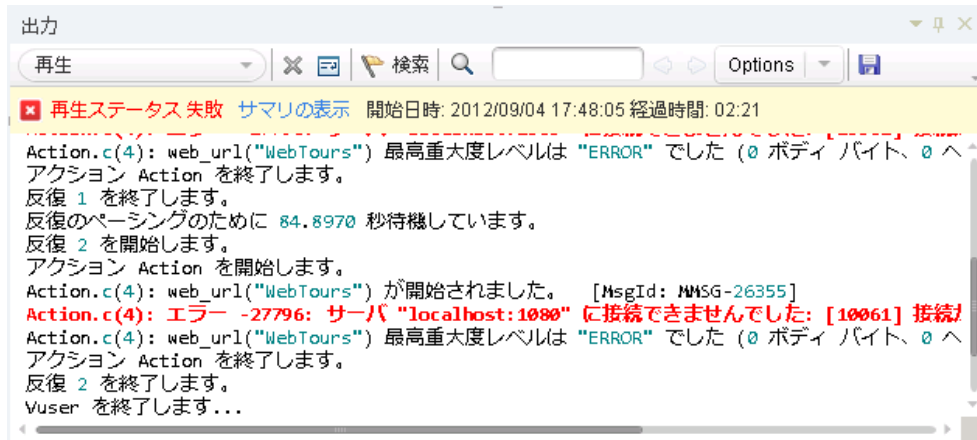
再生ログを表示するには、次の手順を実行します。

1. 仮想ユーザ・スクリプトを再生した後で、[表示]>[出力]をクリックするか、VuGen ツールバーの [出力] ボタン  をクリックします。または、[再生のサマリ] タブにある [再生ログ] リンクをクリックします。
2. [出力] ペインで [再生] が選択されていることを確認してください。



3. VuGen のメニューから、[検索]>[簡易検索]をクリックして[検索]ダイアログ・ボックスを開きます。
4. [範囲]から[現在のスクリプト]を選択します。
5. [検索に含める]をクリックしてから[ログ]チェック・ボックスを選択します。
6. [検索]ダイアログ・ボックスを使用して、再生ログで以下の項目を確認します。
 - a. 「仮想ユーザ・スクリプトが開始されました」-スクリプト実行の開始。
 - b. 「仮想ユーザが終了しました」-スクリプト実行の終了。
 - c. 「反復」-各反復の開始および終了と、反復の回数(このテキストはオレンジ色の文字で表示されます)。

注: [出力] ペインには、成功したステップは緑で、エラーは赤で表示されます。たとえば、仮想ユーザがサーバに接続できなかった場合、[出力] ペインにはエラー・テキストが赤で表示され、エラーが発生したスクリプトの行番号が表示されます。

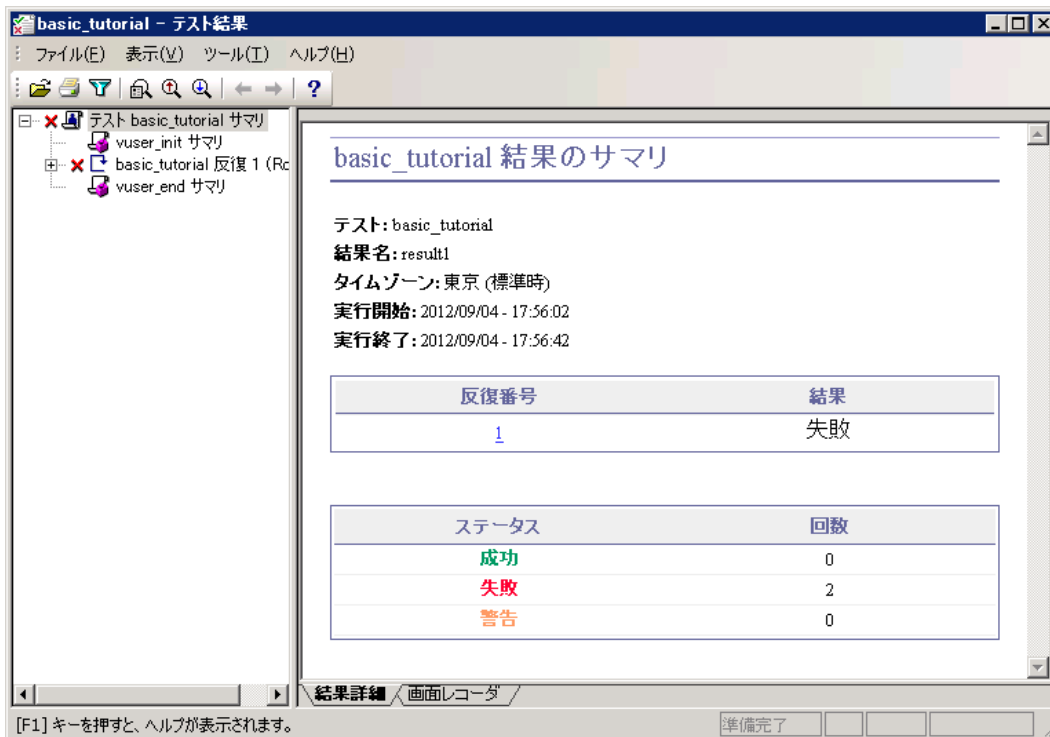


注: [出力] ペインで行をダブルクリックすると、VuGen によって該当するステップが VuGen エディタのスクリプトで示されます。

スクリプト再生成功の確認方法

記録したイベントを再生したら、再生結果を調べて、スクリプトの再生が成功したか確認する必要があります。何か失敗した場合は、失敗した理由と時間を確認する必要があります。

本項では、スクリプト実行の結果を表示し、分析します。VuGen は、再生の結果を [テスト結果] ウィンドウにまとめます。



[テスト結果] ウィンドウを初めて開くと、ペインが2つ表示されます(左側にツリー・ペイン、右側に結果サマリ・ペイン)。

- [ツリー]ペインには、結果ツリーが表示されます。ツリーでの各反復には番号が付けられます。
- [結果サマリ]ペインには、スクリプト再生の詳細や画面レコーダのムービー(存在する場合)が表示されます。最上部の表には、成功した反復と失敗した反復が表示されます。仮想ユーザが最初の記録に従って HP Web Tours サイトを正常にナビゲートした場合、テストは成功したとみなされます。最下部の表には、トランザクションおよびチェックポイントが成功したか失敗したかが表示されます。これらの機能は、本チュートリアルの後の部分でテストに追加します。

再生結果を表示するには、次の手順を実行します。

1. [再生のサマリ]タブをクリックします。
2. [再生]>[テスト結果]をクリックします。[テスト結果]ウィンドウが開き、再生結果のサマリが表示されます。

次のセクションでは、意図した Web ページに再生時にスクリプトが到達したかどうか判断するため、再生結果をドリルダウンします。

再生結果の検索またはフィルタ方法

何かが失敗したことを再生結果が示している場合は、ドリルダウンを行って、失敗した箇所を特定できます。

[テスト結果]ウィンドウの[ツリー]ペインで、テスト・ツリーを展開して、各ステップの結果を個別に表示できます。[サマリ]ペインには、その反復時の再生のスナップショットが表示されます。

1. ツリーで反復ノードを展開します。
 - a. 結果ツリーで[**basic_tutorial 反復 1**]ノードを展開します。
 - b. [**Action Summary**]ノードを展開します。展開されたノードには、その反復で実行されたステップのリストが表示されます。

2. 結果のスナップショットを表示します。

[**Submit Form:login.pl**]ノードをクリックします。[サマリ]ペインには、そのステップに関連付けられている再生スナップショットが表示されます。


3. ステップのサマリを表示します。

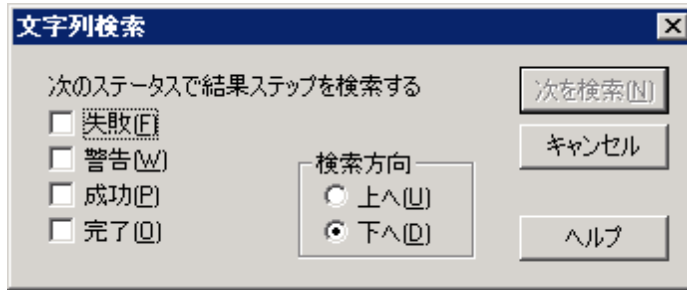
[サマリ]ペインには、オブジェクトまたはステップの名前、ページが正常に読み込まれたかどうかに関する詳細、結果(成功、失敗、完了、警告)、ステップが実行された時間といったステップのサマリ情報が表示されます。

4. 結果ステータスを検索します。

成功または失敗という単語がないか再生結果を検索できます。

全体的な結果サマリが再生が失敗したことを示している場合に、失敗した箇所を特定するのに役立ちます。

- a. 再生結果を検索するには、[ツール]>[検索]を選択するか、[テスト結果]ツールバーの[検索]ボタン  をクリックします。[文字列検索]ダイアログ・ボックスが開きます。




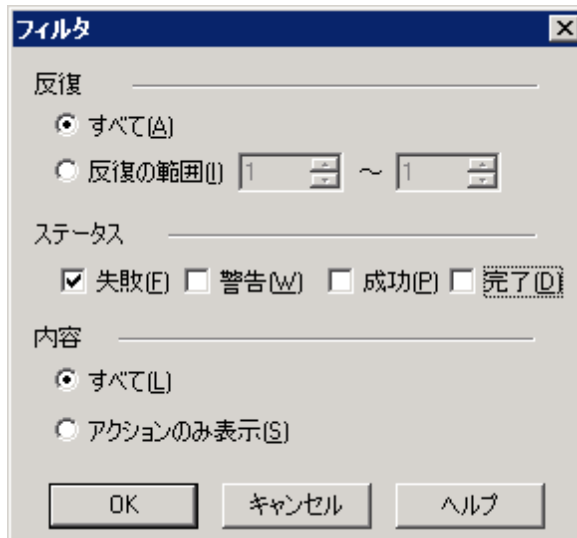
- b. [成功]チェック・ボックスを選択し、ほかのオプションが選択されていないことを確認して、[次を検索]をクリックします。[テスト ツリー]ペインでは、ステータスが[成功]である最初のステップが強調表示されます。

注: 選択したステータスのステップが見つからない場合、ステップは強調表示されません。

5. 結果にフィルタを適用します。

[テスト ツリー]ペインにフィルタを適用して、特定の反復またはステータスを表示できます。たとえば、[失敗]ステータスのみ表示されるようにフィルタを適用できます。

- a. [表示]>[フィルタ]を選択するか、[テスト結果]ツールバーの[フィルタ]ボタン  をクリックします。[フィルタ]ダイアログ・ボックスが開きます。



- b. [ステータス]の下で[失敗]を選択し、その他のすべてのオプションをクリアします。
- c. [内容]の下で[すべて]を選択し、[OK]をクリックします。失敗がなかったため、左側のペインは空になります。
6. [テスト結果]ウィンドウを閉じます。
- [ファイル]>[終了]をクリックします。

次の練習について

HP Web Tours アプリケーションを操作するユーザをエミュレートした仮想ユーザ・スクリプトの再生に成功しました。「練習 3: 一般的な再生の問題の解決」(30ページ)に進んでください。

練習 3: 一般的な再生の問題の解決

仮想ユーザ・スクリプトを作成したら、VuGen で実行して検証します。場合によっては、同じアクションの記録が成功していても、再生が失敗することがあります。

多くのアプリケーションでは、ユーザがアプリケーションを使用するたびに変わる動的な値が使用されます。たとえば、あるサーバは、すべての新しいセッションに対して一意のセッション ID を割り当てます。記録されたセッションを再生しようとする、アプリケーションは、記録されたセッションの ID とは異なる新しいセッション ID を作成します。こうした一意のセッション ID のような動的な値は、特定の種類の仮想ユーザ・スクリプトを再生したときに問題となる場合があります。たとえば、動的なセッション ID は、Web-HTTP/HTML スクリプトを再生したときには問題が起こることが多くありますが、Web-TruClient スクリプトを再生したときには問題が起こりません。

LoadRunner は相関を使用して動的な値の問題に対処します。相関により、変化する値(ここではセッション ID) がパラメータに保存されます。仮想ユーザは、仮想ユーザ・スクリプトの実行時、記録された値を使用しません。代わりに、サーバによって割り当てられた新しいセッション ID を使用します。

この練習では、Web-HTTP/HTML 仮想ユーザ・スクリプトを実行するときに LoadRunner が動的な値に関する問題を解決する方法について観察します。

この練習は、次の項目で構成されています。

- 再生エラーのための HP Web Tours の準備
- 一意のサーバ値の使用方法

再生エラーのための HP Web Tours の準備

再生に関する一般的な失敗を説明するために、HP Web Tours アプリケーションの設定を変更する必要があります。この設定により、HP Web Tours サーバに対して一意のセッション ID の発行と要求を行うように指定します。

1. HP Web Tours を起動します。

[スタート]>[すべてのプログラム]>[HP ソフトウェア]>[HP LoadRunner]>[Samples]>[Web]>[HP Web Tours Application]を選択します。ブラウザが開き、HP Web Tours のホーム・ページが表示されます。


2. サーバのオプションを変更します。
 - a. HP Web Tours のホーム・ページの[administration]リンクをクリックします。管理ページが開きます。
 - b. [Set LOGIN form's action tag to an error page]チェックボックスを選択します。この設定により、サーバは、重複するセッション ID を許可しないようになります。
 - c. ページの最下部までスクロールし、[Update]をクリックします。
 - d. ページの最下部までスクロールし、[Return to the Web Tours Homepage]リンクをクリックします。

一意のサーバ値の使用方法

変更後の HP Web Tours の設定では、サーバは、各仮想ユーザに一意のセッション ID を割り当てます。「練習 1: 仮想ユーザ・スクリプトの作成」(12ページ)で記録した変更なしの仮想ユーザ・スクリプトを再生しようとすると、再生は失敗します。

この問題に対処するために、VuGen を使用して、セッション ID の相関の必要性を検出します。元のセッション ID をパラメータに保存するステップを挿入するように、VuGen を指定します。VuGen は、以降の再生セッションごとに、新しい一意のセッション ID をパラメータに保存します。仮想ユーザが仮想ユーザ・スクリプトのステップを実行すると、仮想ユーザは初めに記録された値の代わりに、保存されたセッション ID 値を使用します。

1. 動的な値を持つ新しい仮想ユーザ・スクリプトを記録します。
 - a. 「練習 1: 仮想ユーザ・スクリプトの作成」(12ページ)で前に記録した同じステップで新しい仮想ユーザ・スクリプトを記録します。
 - b. スクリプトを `basic_tutorial_Cor` として保存します。
2. 新しいスクリプトを再生します。

[再生]>[実行]をクリックするか、VuGen ツールバーの[再生]ボタン  をクリックします。

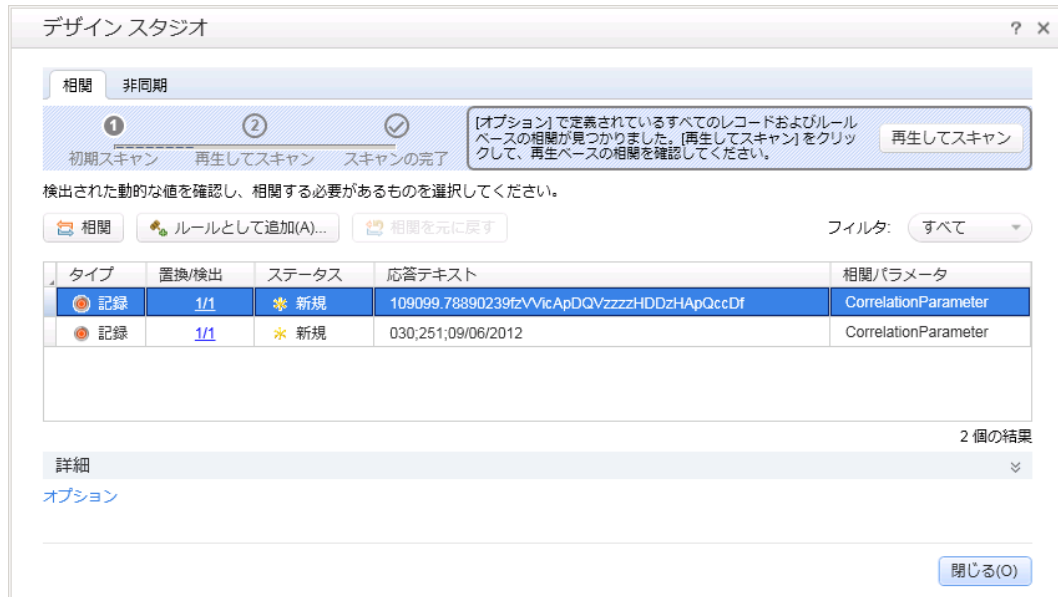
VuGen が新しい仮想ユーザ・スクリプトを実行します。[出力]ペインの[再生ログ]に、赤で色分けされたエラー・メッセージがいくつか表示されます。

再生の完了後、相関を検索するように求めるメッセージ・ボックスが表示される場合があります。
[いいえ]をクリックします。

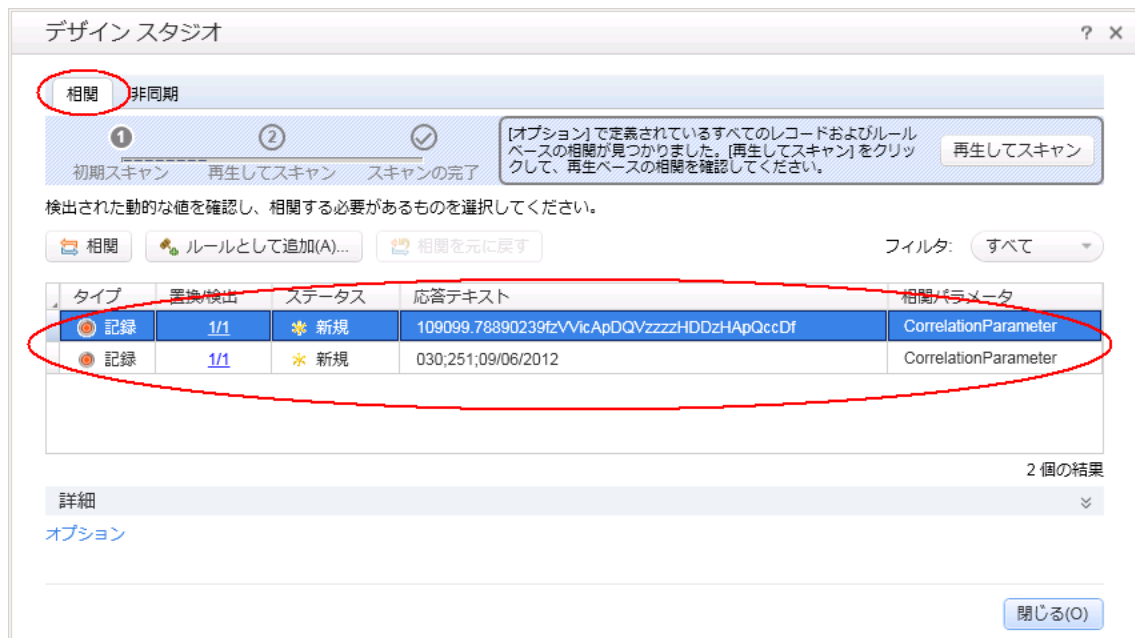
3. 再生のサマリを表示します。

[再生のサマリ]タブを参照します。このサマリは、スクリプトの再生が失敗したことを示しています。
4. 相関のためにスクリプトをスキャンします。

[再生のサマリ]タブで[相関]リンクをクリックします。デザイン・スタジオが起動します。



VuGen はスクリプトとその関連データをスキャンし、動的な値を検索します。デザイン・スタジオの [相関] タブに、相関を必要とする 3 つの動的な値がリスト表示されます。これらの 3 つの値のうち最も長い値はセッション ID です。



5. セッション ID を相関させます。
 - a. [相関] タブでセッション ID エントリを選択し、[相関] をクリックします。VuGen は、セッション ID のステータスを適用済みに変更し、新しい関数を仮想ユーザー・スクリプトの一番上に挿入します。この新しい関数は初めのセッション ID をパラメータに保存します。

VuGen は、以降の再生セッションごとに、新しい一意のセッション ID をパラメータに保存します。仮想ユーザーの実行時に、仮想ユーザーは初めに記録された値の代わりに保存された ID 値を使用します。

- b. [閉じる] をクリックしてデザイン・スタジオを閉じます。


6. 関連ステートメントの構文を調べます。

VuGen エディタで、VuGen がスクリプトに追加したステートメントを探します。この新しいステートメントは次のようになります。

```
web_reg_save_param_regexp(  
  "ParamName=CorrelationParameter",  
  "RegExp=userSession\\ value=(.*?)>\\n<table\\ border",  
  "SEARCH_FILTERS",  
  "Scope=Body",  
  "IgnoreRedirections=No",  
  "RequestUrl=*/nav.pl*",  
  LAST);
```

ステートメントにより、VuGen は正規表現 (一意のセッション ID) に含まれるこのデータの最初の出現を **CorrelationParameter** というパラメータに保存します。

7. 再度スクリプトを再生します。

- a. [再生]>[実行]をクリックするか、VuGen ツールバーの[再生]ボタン  をクリックして、スクリプトを再び再生します。再生が終了したら、[出力]ペインの[再生ログ]を確認します。赤で色分けされたエラー・メッセージが発行されていない点に注意してください。
- b. スクリプトの **web_reg_save_param_regexp** を右クリックし、[再生ログのステップに移動]を選択します。VuGen によって、再生ログの対応する行にカーソルが挿入されます。関数 **web_reg_save_param_regexp** が成功し、相関が正常に機能したことがログからわかります。

8. 一意のセッション ID を無視するように、HP Web Tours サーバをリセットします。

- a. [スタート]>[すべてのプログラム]>[HP LoadRunner]>[Samples]>[Web]>[HP Web Tours Application]をクリックして、HP Web Tours を開きます。
- b. HP Web Tours ホーム・ページの[administration]リンクをクリックします。
- c. 管理ページで、[Set LOGIN form's action tag to an error page]チェック・ボックスをクリアします。
- d. ページの最下部までスクロールし、[Update]をクリックします。

次の練習について

これで、再生に関する一般的な問題の学習が終了し、「練習 4: 負荷テスト用仮想ユーザ・スクリプトの準備」(34ページ)に進む準備ができました。

練習 4: 負荷テスト用仮想ユーザ・スクリプトの準備

前の練習では、仮想ユーザ・スクリプトの応答が実際のユーザを正確にエミュレートしていることを確認しました。次のステップとして、負荷テスト用のスクリプトを準備します。多数のユーザが同時に作業している場合、システムはどのように動作するでしょうか。システムの処理速度が許容できないレベルにまで低下しないでしょうか。

この練習では、スクリプトを拡張し、より効果的に負荷テストのプロセスを進めるためのさまざまな方法について学びます。

この練習は、次の項目で構成されています。

- ビジネス・プロセスの実行時間を測定する方法
- 複数のユーザをエミュレートする方法
- Web ページの内容の確認方法
- デバッグ情報の生成方法
- テストの再生成功の確認方法

ビジネス・プロセスの実行時間を測定する方法

アプリケーションをデプロイするために準備するときは、ログオンに要した時間やフライトの予約に要した時間など、特定のビジネス・プロセスの実行時間を測定する必要があります。通常、各ビジネス・プロセスはスクリプトの1つまたは複数のステップやアクションで構成されます。仮想ユーザ・スクリプトでは、これらのアクションを**トランザクション**に含めることで、測定する必要のある一連のアクションを指定します。

トランザクションが含まれる仮想ユーザ・スクリプトを実行すると、LoadRunner は、トランザクションの実行に要する時間に関する情報を収集し、その結果を色付きのグラフやレポートで表示します。この情報を元に、アプリケーションがパフォーマンスの要件を満たしているかどうか判断できます。

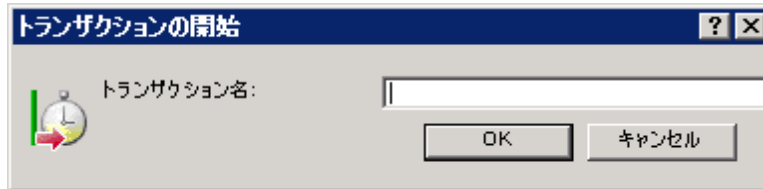
トランザクションは仮想ユーザ・スクリプト内のどこにでも手動で挿入できます。一連のステップをトランザクションとしてマークするには、最初のステップの前に **start_transaction** マーカを挿入し、最後のステップの後に **end_transaction** マーカを挿入します。

本項では、ユーザがフライトを検索して確定するまでに要する時間を計測するためのトランザクションを、スクリプトに挿入します。

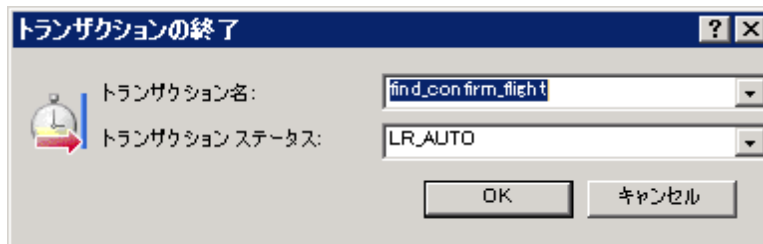
仮想ユーザ・スクリプトにトランザクションを挿入するには、次の手順を実行します。

1. VuGen で、「練習 1: 仮想ユーザ・スクリプトの作成」(12ページ)で作成した **Basic_Tutorial** スクリプトを開きます。まだ開いている場合は、この名前が表示されているタブを選択できます。開いていない場合は、[ファイル]メニューから開くことができます。
2. [ステップ ナビゲータ] タブをクリックしてステップ・ナビゲータを表示します。

3. トランザクション開始 マーカを挿入します。
 - a. ステップ・ナビゲータで、ステップ[Image: Search Flights Button]を探します。
 - b. [Image: Search Flights Button]ステップをダブルクリックして、該当する[web_image]ステップをエディタで表示します。
 - c. [表示]>[ステップ ツールボックス]をクリックします。ステップ・ツールボックスが右側のペインに表示されます。
 - d. ステップ・ツールボックスの[共通]の下で[Ir_start_transaction]を選択し、エディタまでドラッグして、[web_image]ステップの前でドロップします。[トランザクションの開始]ダイアログボックスが開きます。



- e. [トランザクション名]ボックスで「find_confirm_flight」と入力し、[OK]をクリックします。VuGen は[Ir_start_transaction]ステップをステップ・ナビゲータに挿入し、該当する Ir_start_transaction 関数をエディタに挿入します。
4. トランザクション終了 マーカを挿入します。
 - a. ステップ・ナビゲータで、ステップ[Submit Data: reservations.pl_2]を探します。
 - b. [Submit Data: reservations.pl_2]ステップをダブルクリックして、該当する[web_submit_data]ステップをエディタで表示します。
 - c. ステップ・ツールボックスの[共通]の下で[Ir_end_transaction]を選択し、エディタまでドラッグして、[web_submit_data]ステップの後ろでドロップします。[トランザクションの終了]ダイアログボックスが開きます。



- d. [find_confirm_flight]が[トランザクション名]ボックスに表示されることを確認し、[OK]をクリックします。VuGen は[Ir_end_transaction]ステップをステップ・ナビゲータに挿入し、該当する Ir_end_transaction 関数をエディタに挿入します。

行	名前	ステップ	スク
4	web_url	Url: WebTours	Actio
13	lr_think_time	思考遅延時間 - 24	Actio
15	web_submit_fo	Submit Form: log	Actio
24	lr_think_time	思考遅延時間 - 34	Actio
25	lr_start_transaction	トランザクション	Actio
27	web_image	Image: Search Fl	Actio
32	lr_think_time	思考遅延時間 - 51	Actio
34	web_submit_fo	Submit Form: res	Actio
49	lr_think_time	思考遅延時間 - 17	Actio
51	web_submit_fo	Submit Form: res	Actio
57	lr_end_transaction	トランザクション	Actio
60	lr_think_time	思考遅延時間 - 26	Actio
62	web_submit_fo	Submit Form: res	Actio
77	lr_think_time	思考遅延時間 - 17	Actio
79	web_image	Image: Itinerary E	Actio
84	lr_think_time	思考遅延時間 - 6	Actio
86	web_image	Image: SignOff B	Actio

これで、**find_confirm_flight** トランザクションが定義できました。

複数のユーザをエミュレートする方法

このエミュレーションでは、ユーザによるフライトの予約と通路側席の選択を追跡しました。ただし、実際の設定はユーザの好みによって異なります。したがって、テストを改善するために、ユーザが別の座席設定（通路側、窓側、または指定なし）を選択した場合でも予約が実行されるかどうかを確かめる必要があります。


これを実現するには、スクリプトを「パラメータ化」します。つまり、記録された値 **Aisle** (通路側) を受け取り、それをパラメータに置き換えます。パラメータの値はパラメータ・ファイルに記述します。スクリプトを実行すると、仮想ユーザは値（通路側、窓側、または指定なし）をパラメータ・ファイルから使用します。それによって、実際の旅行会社における環境がエミュレートされます。


スクリプトをパラメータ化するには、次の手順を実行します。

1. データを変更するセクションを見つけます。
 - a. [表示]>[ステップ ナビゲータ]を選択して、ステップ・ナビゲータを左側のペインに表示します。
 - b. ステップ・ナビゲータで、[Submit Data: reservations.pl]ステップを探します。
 - c. [Submit Form: reservations.pl]ステップを右クリックし、[引数の表示]を選択します。



[フォームを送信ステップのプロパティ]ダイアログ・ボックスが開きます。

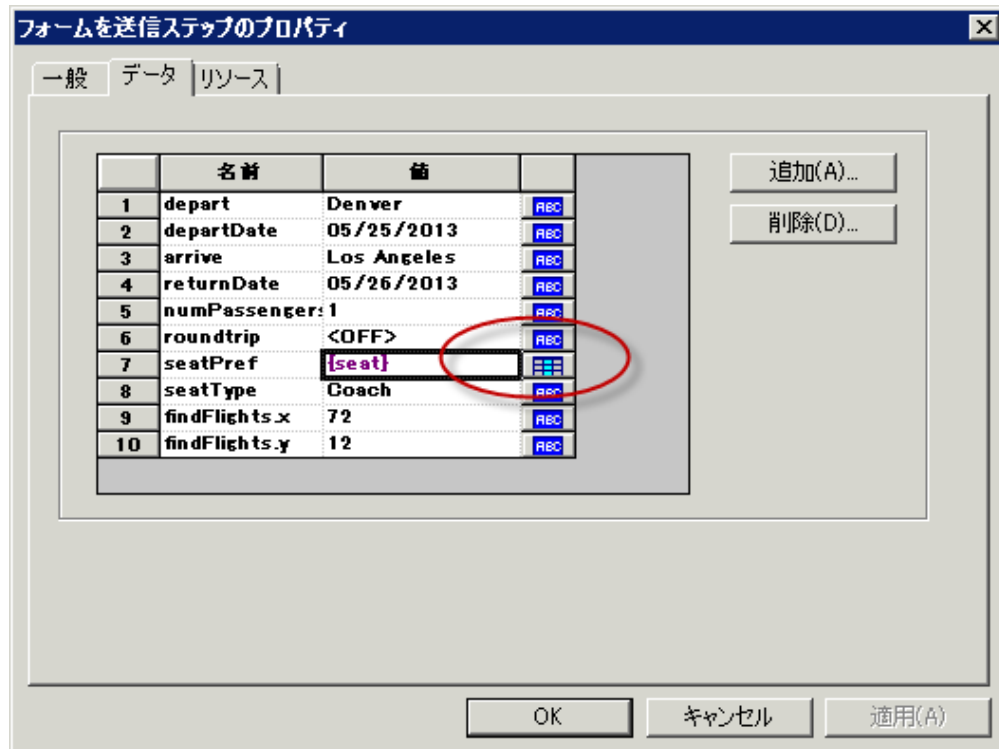



グリッドの各引数の右側にある[ABC]アイコン  は、その引数が固定値であることを示しています。

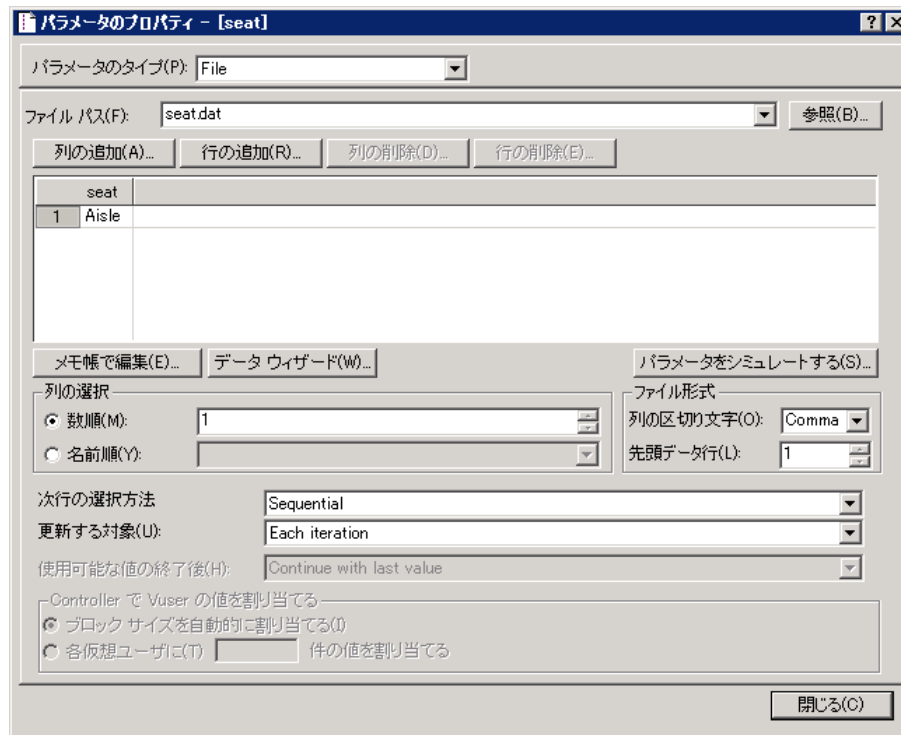
2. 固定値を可変値に変更します。
 - a. [フォームを送信ステップのプロパティ]ダイアログ・ボックスで、グリッドの7番目の行にある **seatPref** を選択します。
 - b. **seatPref** 引数の横にある[ABC]アイコン  をクリックします。[パラメータの選択または作成]ダイアログ・ボックスが開きます。



3. パラメータを作成します。
 - a. [パラメータ名]ボックスに「**seat**」と入力します。
 - b. [OK]をクリックします。[フォームを送信ステップのプロパティ]ダイアログ・ボックスで、VuGen は [ABC]アイコン  を [パラメータ]アイコン  に置き換えます。



- c. {seat} の横にある[パラメータ]アイコン  をクリックし、[パラメータのプロパティ]を選択します。[パラメータのプロパティ]ダイアログ・ボックスが開きます。



4. パラメータの値を指定します。
 - a. [行の追加]をクリックします。行がテーブルに追加されます。
 - b. 単語「値」を「Window」に置き換えます。
 - c. [行の追加]をクリックします。別の行がテーブルに追加されます。
 - d. 単語「値」を「None」に置き換えます。



注: 値の大文字と小文字は区別されます。

- e. ダイアログ・ボックスの[列の選択]および[ファイル形式]のセクションでは標準設定を維持します。
5. テストでのデータの変更方法を定義します。
 - a. 標準設定 [Update value on: Each iteration]を維持し、反復ごとに新しい値をとるようVuGenに指示します。
 - b. [閉じる]をクリックし、[パラメータのプロパティ]ダイアログ・ボックスを閉じます。
 - c. [OK]をクリックし、[フォームを送信ステップのプロパティ]ダイアログ・ボックスを閉じます。

これで、座席設定用のパラメータが作成されました。負荷テストを実行すると、仮想ユーザは記録された値 **Aisle** ではなくパラメータ値を使用します。

スクリプトを実行すると、反復ごとに発生するパラメータ置換が再生ログに表示されます。仮想ユーザは最初の反復に「**Aisle**」を使用し、2番目の反復に「**Window**」を使用し、3番目の反復に「**None**」を使用します。


Web ページの内容の確認方法


























テストを実行するとき、返されるページ上で特定の内容が見つかったかどうかを確認する必要があります。コンテンツ・チェックは、スクリプトの実行中に期待どおりの情報がWebページ上に表示されるかどうかを検証するものです。2種類の内容チェックを挿入できます。

- [テキスト チェック]: テキスト文字列がWebページ上に表示されているかどうかを検証します。
- [画像 チェック]: 画像がWebページ上に表示されているかどうかを検証します。

本項では、HP Web Tours の予約ページに「**Find Flight**」という語句が表示されるかを検証するテキスト・チェックを挿入します。

テキスト・チェックの挿入

1. [ステップ ナビゲータ] タブをクリックしてステップ・ナビゲータを表示します。
2. ステップ・ナビゲータで, [Submit Form: reservations.pl] ステップを探します。
3. VuGen ツールバーの[スナップショット] ボタン  をクリックして[スナップショット] ペインを表示します。
4. ステップ・ナビゲータで, [Submit Form:reservations.pl] ステップのスナップショット・アイコンをダブルクリックします。該当するスナップショットが[スナップショット] ペインに表示されます。
5. VuGen メニューで[表示]>[ステップ ツールボックス]をクリックします。[ステップ ツールボックス] ペインが表示されます。
6. [ステップ ツールボックス - 検索] ボックスに web_reg と入力し, [フィルタ結果] で[web_reg_find] ステップを探します。
7. [web_reg_find] ステップをツールボックスからエディタ内の web_submit_form 関数の前にドラッグします。[文字列の検索] ダイアログ・ボックスが開きます。
8. [文字列の検索] ダイアログ・ボックスで, [特定の文字列の検索] ボックスに Find Flight と入力し, [OK] をクリックします。
9. VuGen は[web_reg_find] ステップをステップ・ナビゲータに挿入し, 該当する web_reg_find 関数をエディタに挿入します。

ステップ ナビゲータ		
Search text in すべて  		
行	名前	ステップ
4	 web_url 	Url: WebTours
13	 lr_think_time	思考遅延時間 - 246 (秒)
15	 web_submit_form 	Submit Form: login.pl
24	 lr_think_time	思考遅延時間 - 34 (秒)
25	 lr_start_transaction	トランザクションの開始
28	 web_reg_save_param	Web Reg Save Param F
37	 web_reg_save_param	Web Reg Save Param F
45	 web_image 	Image: Search Flights B
50	 lr_think_time	思考遅延時間 - 51 (秒)
53	 web_reg_save_param	Web Reg Save Param F
60	 web_reg_find	Service: Reg Find
62	 web_submit_form 	Submit Form: reservatio
77	 lr_think_time	思考遅延時間 - 17 (秒)
79	 web_submit_form 	Submit Form: reservatio
85	 lr_end_transaction	トランザクションの終了
88	 lr_think_time	思考遅延時間 - 26 (秒)
90	 web_submit_form 	Submit Form: reservatio

スクリプトを再生すると、テキスト「Find Flight」が検索され、見つかったかどうか再生ログに示されます。

デバッグ情報の生成方法

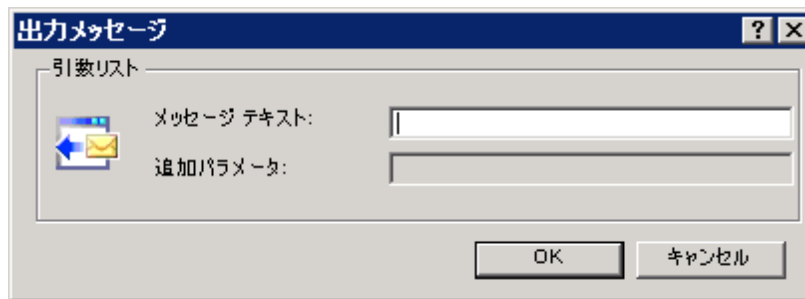
テスト実行中の特定の時点で、スクリプト実行に関する情報を含むメッセージを生成して送信する必要があります。これらの出力メッセージは[出力]ペインの再生ログと Controller の[出力]ウィンドウの両方に表示されます。標準の出力メッセージ、またはエラーが発生したことを示すメッセージを送信できます。

エラー・メッセージで作業を行う場合は、ステータスが**失敗**かどうかを検証する方法をお勧めします。ステータスが**失敗**の場合は、エラー・メッセージを発行するよう VuGen を設定します。詳細については、『HP LoadRunner Function Reference』の例を参照してください。

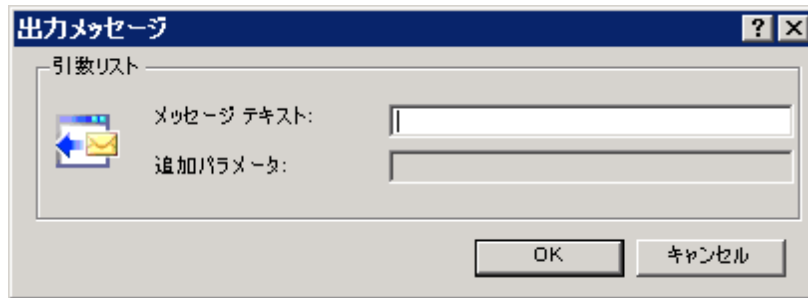
チュートリアルの本項では、アプリケーションによる予約が完全に済んだ後に出力メッセージを挿入するよう VuGen を設定します。


出力メッセージを挿入するには、次の手順を実行します。

1. [ステップ ナビゲータ] タブをクリックしてステップ・ナビゲータを表示します。
2. ステップ・ナビゲータで、最後のステップ [Image: SignOff Button] を探します。
3. [Image: SignOff Button] ステップをダブルクリックします。エディタで該当する `web_image` 関数が表示されます。
4. ステップ・ツールボックスの[共通]の下で `lr_output_message` 関数を探します。
5. `lr_output_message` 関数を選択し、エディタまでドラッグして、`web_image` 関数の後ろでドロップします。[出力メッセージ] ダイアログ・ボックスが開きます。



6. [メッセージ テキスト] ボックスに「The flight was booked」(フライトが予約されました) と入力します。
7. [OK] をクリックします。 `lr_output_message` がスクリプトに追加され、該当するステップがステップ・ナビゲータに追加されます。



8. VuGen ツールバーの[保存]ボタン  をクリックしてスクリプトを保存します。


エラー・メッセージを挿入する場合も同様の手順を繰り返します。ただし、ステップ・ツールボックスで、`lr_output_message` 関数の代わりに `lr_error_message` 関数を選択します。

テストの再生成功の確認方法

本項では、拡張スクリプトを実行して、テキスト・チェックを実行するために再生ログを検索します。テキスト・チェックの結果だけでなく、トランザクションおよびパラメータ化の詳細も表示されます。

画像およびテキストのチェックは必要なメモリ量が増えるため、標準では再生中は無効となっています。画像またはテキストのチェックを実行する場合は、実行環境の設定でチェックを有効にする必要があります。

1. 画像とテキストのチェックを有効にします。
 - a. [再生]>[実行環境設定]をクリックし、[実行環境設定]ダイアログ・ボックスを開きます。
 - b. [インターネット プロトコル]>[プリファレンス]をクリックします。
 - c. [画像とテキスト チェックを有効にする]を選択します。
 - d. [OK]をクリックして、[実行環境設定]ダイアログ・ボックスを閉じます。
2. スクリプトを実行します。

VuGen ツールバーの[再生]ボタン  をクリックする。VuGen によってスクリプトの実行が開始され、再生ログのエントリが[出力]ペインに生成されます。

スクリプトの実行が完了するまで待ちます。

3. テキスト・チェックを探します。
 - a. [出力]ペインをクリックし、[再生]を選択します。
 - b. 再生ログの中をクリックし、Ctrl + F キーを押して[検索]ダイアログ・ボックスを開きます。
 - c. `web_reg_find` を検索します。

最初のインスタンスは次のようになっています。

```
web_reg_find が開始されました
```

[次を検索]をクリックして、`web_reg_find` の次のインスタンスを表示します。2 番目のインスタンスは次のようになっています。


web_reg_find の登録に成功しました。

これは実際のテキスト・チェックではなく、フォームの送信後にテキスト・チェックを行うように VuGen を準備しているだけです。

[**次を検索**] をクリックして、**web_reg_find** の次のインスタンスを表示します。このインスタンスは次のようになっています。

"Text=Find Flight" を対象とした web_reg_find の登録に成功しました(カウント = 1)。

これは、テキストが見つかったことを示します。だれかが Web ページを変更して語句「**Find Flight**」を削除した場合、以降の実行では、そのテキストが見つからなかったことが出力に示されます。

4. トランザクションの開始を探します。
 - a. 再生ログで、Ctrl + F キーを押して[検索]ダイアログ・ボックスを開きます。
 - b. 単語「**トランザクション**」を探します。この通知は青で表示されます。
5. パラメータ置換を表示します。
 - a. 再生ログで、Ctrl + F キーを押して[検索]ダイアログ・ボックスを開きます。
 - b. 単語「**パラメータ**」を探します。ログに「**seat** = "**Aisle**"」という通知が含まれます。
 - c. もう一度検索を実行し(F3 キー)、次の置換を探します。VuGen が反復ごとに異なる値を受け取っていることがわかります。
6. [ファイル]>[保存]を選択するか、VuGen ツールバーの[**上書き保存**]ボタン  をクリックします。

次の練習について

この練習で、スクリプトを作成して負荷テスト用に調整する作業が完了しました。次に、「[練習 5: 負荷テスト・シナリオの作成](#)」(44ページ)に進みます。

練習 5: 負荷テスト・シナリオの作成

前の練習では、VuGenを使用して仮想ユーザ・スクリプトを検証しました。この練習では、複数の仮想ユーザの負荷がかかった状態でシステムを評価します。10か所の旅行代理店による、フライト予約システムを使用した同時操作をエミュレートし、こうしたユーザの負荷がかかった状態でシステムの動作を観察します。このテストを設計して実行するには、LoadRunner Controllerを使用します。

シナリオの目標

この練習の目標は、10か所の旅行代理店がシステムに対して同時にログオンし、フライトの検索と購入および日程のチェックを行い、ログオフするまでの動作をエミュレートするシナリオを作成することです。

この練習は、次の項目で構成されています。

- LoadRunner Controller の紹介
- Controller の開始方法
- Controller の概要
- スクリプトの詳細の変更方法
- システムで負荷を生成する方法
- 実際の負荷動作のエミュレート方法
- さまざまなタイプのユーザのエミュレート方法
- 負荷下のシステムの監視方法

LoadRunner Controller の紹介

負荷テストとは、一般的な作業条件の下でシステムをテストすることです。たとえば、多数の旅行代理店が同じフライト予約システムで同時にフライトを予約する場合のシステムのテストがあります。

テストを設計する際は、実際の状況をエミュレートするように設計します。このためには、アプリケーションに対して負荷を生成できること、負荷をいつ適用するのかについてスケジュールを設定できること（複数のユーザがシステムに対してログオンとログオフをまったく同時に実行することはないため）が必要になります。また、さまざまな種類にわたるユーザの操作や動作をエミュレートする必要もあります。たとえば、一部のユーザがFirefoxを使用してシステムにアクセスし、一部のユーザがInternet Explorerを使用してアクセスする場合があります。また、モデム、DSL、ケーブルなど、異なるネットワーク接続を使用するユーザがいる場合もあります。これらの設定を、シナリオの中に作成して保存します。

作業環境を正確にエミュレートできるように、Controllerにはテストの作成と実行の作業に必要なツールがすべて含まれています。

Controller の開始方法

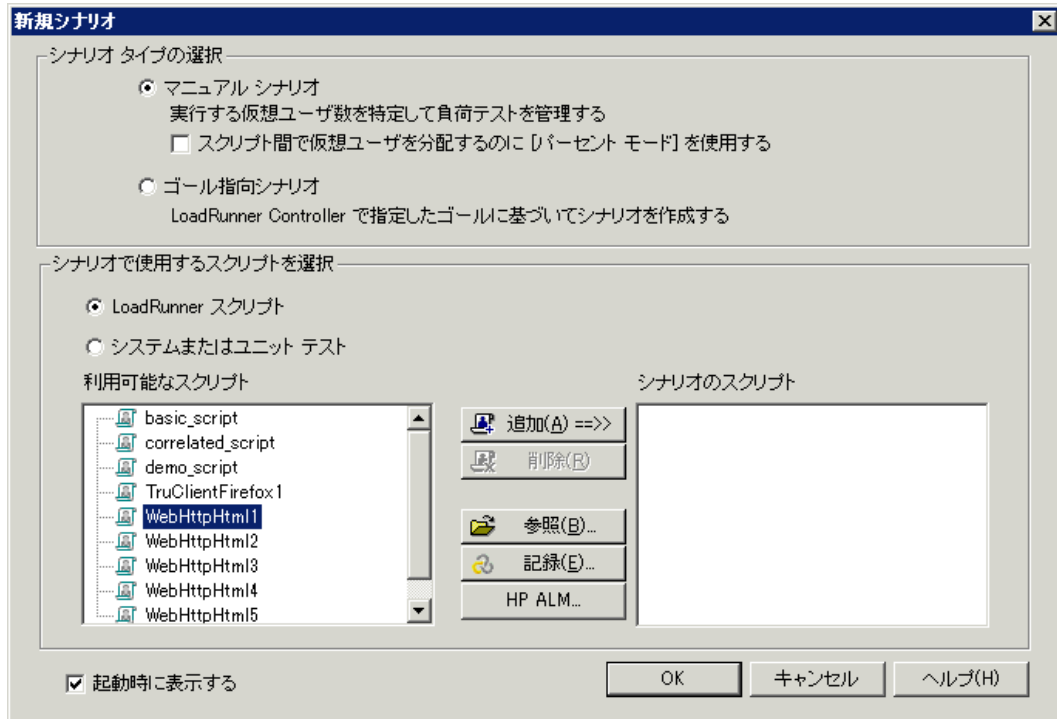
シナリオ作成を開始するには、LoadRunner Controllerを開きます。

1. HP LoadRunner Controller を開きます。

[スタート]>[すべてのプログラム]>[HP ソフトウェア]>[HP LoadRunner]>[Controller]を選択するか、デスクトップで[Controller]ショートカット・アイコンをクリックします。



HP LoadRunner Controller が起動し、[新規シナリオ]ダイアログ・ボックスが表示されます。



2. シナリオのタイプを選択します。

2 種類のシナリオ・タイプが存在します。

「**マニュアル・シナリオ**」では、実行中の仮想ユーザの数や仮想ユーザの実行回数を制御でき、いくつかの仮想ユーザをアプリケーションで同時に実行できるかをテストできます。パーセント・モードを使用すると、ビジネス・アナリストが指定するパーセントに基づいて、スクリプト間での仮想ユーザの総数を配分できます。インストール後、最初に LoadRunner を起動すると[パーセント モード]チェック・ボックスは標準で選択されています。選択されている場合は、クリアします。

「**ゴール指向シナリオ**」は、システムが特定のゴール(目標)を達成できるかどうかを調べるのに使用します。たとえば、特定のトランザクション応答時間、秒ごとのヒット数、秒ごとのトランザクション数などに基づいてゴールを決定すると、LoadRunner によって、それらのゴールに基づくシナリオが自動的に作成されます。

- [マニュアルシナリオ]をクリックします。

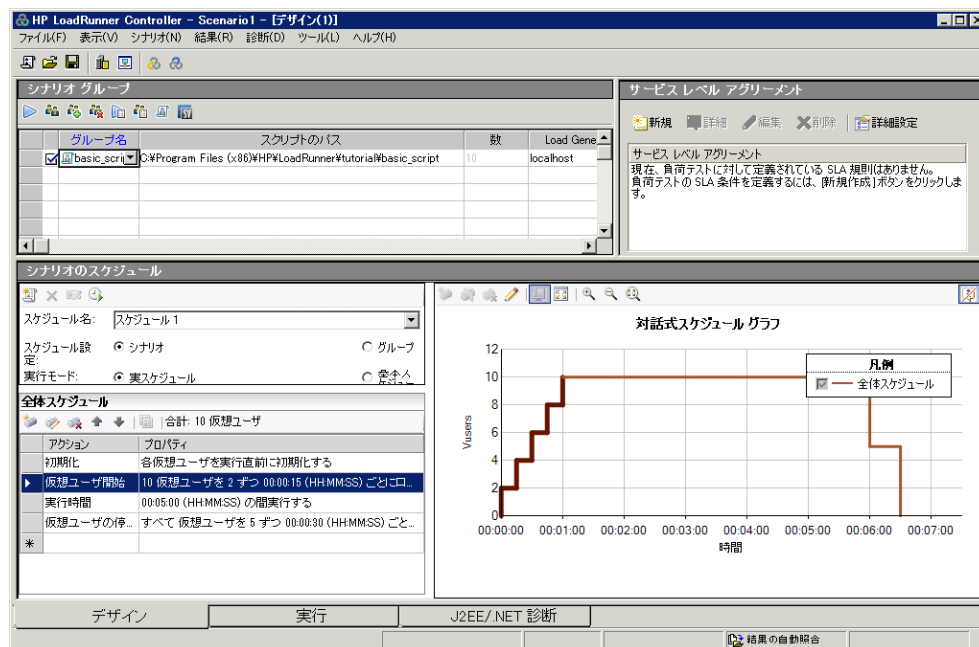
3. 仮想ユーザ・スクリプトを負荷テストに追加します。

このチュートリアルでは、仮想ユーザ・スクリプトを1つだけ使用して、同じアクションを実行する1つのユーザ・グループをモデル化します。より多様なユーザ・プロファイルを使用して現実世界のシナリオをより正確にエミュレートするには、多数の異なる仮想ユーザ・グループを作成し、各グループで異なるユーザ設定を持つ複数のスクリプトを実行します。

以前にVuGenで記録したスクリプトには、テストを必要とするビジネス・プロセスが含まれています。これらのビジネス・プロセスには、ログオン、フライトの検索、フライトの購入、フライト日程のチェック、およびサイトからのログオフまでの過程が含まれています。同様のスクリプトをシナリオに追加し、これらのアクションをフライト予約システムで同時に実行する8つの旅行代理店をエミュレートするよう、シナリオを設定します。また、テスト中にさらに2人の仮想ユーザを追加します。

このために、前の手順で作成したスクリプトとほぼ同じサンプル・スクリプトが用意されています。このサンプル・スクリプトを使用することをお勧めします。

- [新規シナリオ]ダイアログ・ボックスで、[参照]ボタンをクリックします。
- <LoadRunner インストール先フォルダ>\Tutorial フォルダで **basic_script** を探して選択します。[開く]をクリックします。
- [追加]ボタンをクリックして、[新規シナリオ]ダイアログ・ボックスの[利用できるスクリプト]ペインから[シナリオ内のスクリプト]ペインに **basic_script** を移動します。
- [OK]をクリックします。LoadRunner Controller が起動し、新しいシナリオの[デザイン]タブが表示されます。



[デザイン]タブのコントロールがControllerの左下にある点に注意してください。

Controller の概要

Controllerの[デザイン]タブは、負荷テストを設計するための主なインターフェースです。[デザイン]タブは以下の3つのペインに分かれています。

1. [シナリオグループ]ペイン
2. [サービスレベルアグリーメント]ペイン
3. [シナリオのスケジュール]ペイン

The screenshot displays the HP LoadRunner Controller interface for Scenario 1 in Design mode. The top menu bar includes options like File, View, Scenario, Results, Diagnostics, Tools, and Help. The main workspace is divided into three panels:

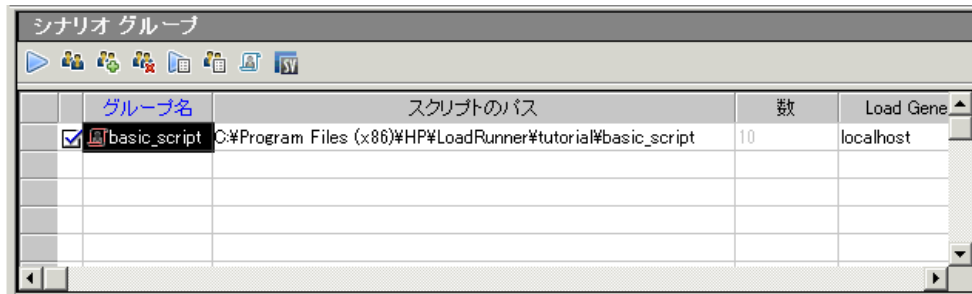
- シナリオグループ (Scenario Groups):** A table with columns: グループ名 (Group Name), スクリプトのパス (Script Path), 数 (Count), and Load Gene. One entry is visible: basic_script, C:\Program Files (x86)\HP\LoadRunner\tutorial\basic_script, 10, localhost.
- サービスレベルアグリーメント (Service Level Agreement):** A panel with buttons for New, Details, Edit, Delete, and Detailed Settings. It contains text: "サービスレベルアグリーメント 現在、負荷テストに対して定義されている SLA 規則はありません。負荷テストの SLA 条件を定義するには、[新規作成] ボタンをクリックします。"
- シナリオのスケジュール (Scenario Schedule):** A panel with a '対話式スケジュール グラフ' (Interactive Schedule Graph). The graph shows 'Users' on the y-axis (0 to 12) and '時間' (Time) on the x-axis (00:00:00 to 00:07:00). The graph shows a step-like increase in users from 0 to 10 between 00:00:00 and 00:06:00, then a drop to 5 and a final drop to 0.

1. [シナリオグループ]ペイン: 仮想ユーザグループは[シナリオのスクリプト]ペインで設定します。システムの典型的なユーザを表すさまざまなグループを作成し、実行する仮想ユーザ数、実行するマシンを指定します。
2. [サービスレベルアグリーメント]ペイン: 負荷テスト・シナリオの設計時には、パフォーマンス測定値の目標、または SLA(サービス・レベル・アグリーメント)を定義できます。シナリオを実行すると、LoadRunner はパフォーマンスに関連するデータを収集し格納します。実行を分析すると、Analysis はこのデータを SLA に照らして比較し、定義されている測定値に対して SLA のステータスを判定します。
3. [シナリオのスケジュール]ペイン: [シナリオのスケジュール]ペインでは、ユーザの動作を正確に演じる負荷動作を設定します。仮想ユーザが実行するアクション、負荷がアプリケーションに適用される割合、負荷テストの期間、および負荷の終了方法を定義します。


スクリプトの詳細の変更方法

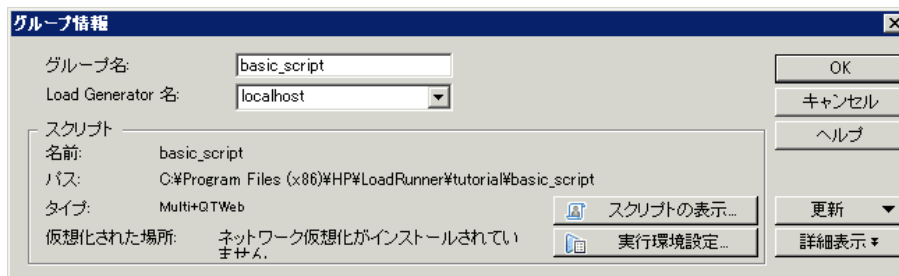
シナリオの詳細は次の方法で変更します。

1. **basic_script** が [シナリオグループ] ペインの [グループ名] カラムに表示されていることを確認します。



2. [グループ名]を変更します。

- a. [シナリオ グループ] ペインで、**basic_script** を選択して[詳細]ボタン  をクリックします。[グループ情報] ダイアログ・ボックスが表示されます。



- b. [グループ名] ボックスに、よりわかりやすい名前を入力します。たとえば、**travel_agent** などを入力します。
- c. [OK] をクリックします。この新しい名前は [デザイン] タブの [シナリオ グループ] ペインに表示されます。


システムで負荷を生成する方法

仮想ユーザ・スクリプトをシナリオに追加した後で、システムで負荷を生成するコンピュータである Load Generator を設定します。

Load Generator とは、システムで負荷を生成するために複数の仮想ユーザを実行するコンピュータです。複数の Load Generator を使用し、それぞれの Load Generator で複数の仮想ユーザをホストできます。

本項では、シナリオへの Load Generator の追加と、Load Generator への接続のテストについて学びます。

1. Load Generator を追加します。

Controller ツールバーの [Load Generators] ボタン  をクリックします。[Load Generator] ダイアログ・ボックスが開きます。



[Load Generator] ダイアログ・ボックスでは、シナリオで定義された Load Generator を表示して設定できます。[Load Generator] ダイアログ・ボックスには、**localhost** という名前の Load Generator の詳細が表示されます。**localhost** Load Generator のステータスは[**ダウン**]です。これは、Controller が **localhost** という名前の Load Generator に接続されていないことを示します。

このチュートリアルでは、ローカル・コンピュータを Load Generator として使用します。

標準的な実稼動システムでは、Load Generator が複数あり、それぞれが複数の仮想ユーザのホストになる点に注意してください。

2. Load Generator への接続をテストします。

シナリオを実行すると、Controller が自動的に Load Generator に接続します。ただし、シナリオの実行を試みる前に接続をテストできます。

a. [Load Generator] ダイアログ・ボックスで、**localhost** を選択して[**接続**]をクリックします。

Controller が Load Generator マシンへの接続を試みます。接続が確立されると、Load Generator の[**ステータス**]が「**ダウン**」から「**準備完了**」に変わります。



- b. [閉じる]をクリックします。

実際の負荷動作のエミュレート方法

Load Generator の追加が完了すると、負荷動作を設定できるようになります。

通常、複数のユーザがシステムに対してまったく同時にログオンとログオフを実行することはありません。LoadRunner では、システムに対するユーザのログオンとログオフを段階的に実行することが可能です。また、シナリオの期間やシナリオの終了方法も決めることができます。次に設定するシナリオは比較的シンプルです。ただし、現実のシナリオをより正確に反映したシナリオをデザインする場合、より現実に近い仮想ユーザの動作を定義することができます。

Controller の[シナリオのスケジュール]ペインでマニュアル・シナリオの負荷の動作を設定します。[シナリオのスケジュール]ペインは、[スケジュールの定義]領域、[アクション]グリッド、および[対話式スケジュール グラフ]の3つのセクションに分かれています。

ここで、標準の負荷設定を変更し、シナリオ・スケジュールを設定します。

1. スケジュールのタイプと実行モードを選択します。

[シナリオのスケジュール]ペインで、[スケジュール設定: シナリオ]と[実行モード: 実スケジュール]が選択されていることを確認します。



2. スケジュール・アクションの定義を設定します。

[全体スケジュール]グリッドから、または対話式スケジュールグラフを操作することで、シナリオのスケジュールに[仮想ユーザ開始]、[実行時間]、および[仮想ユーザの停止]のアクションを設定

できます。グラフから定義を設定した場合、[全体スケジュール]グリッドのプロパティは必要に応じて調整されます。

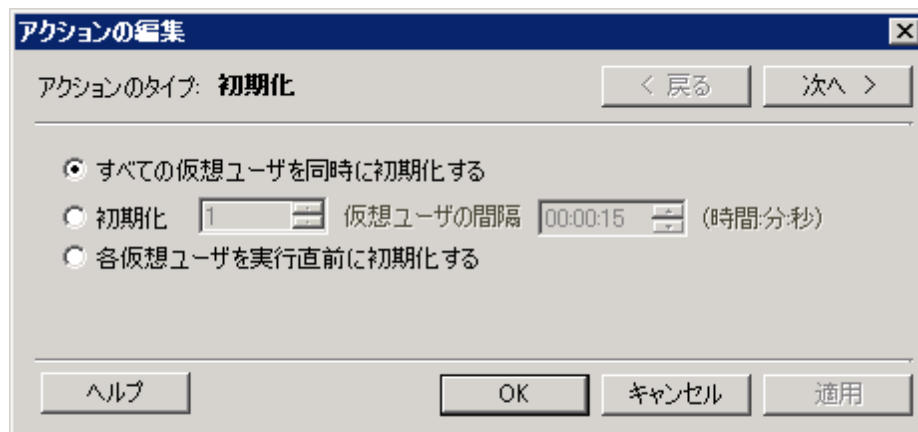
定義を設定し、[全体スケジュール]グリッドが次のようになるようにします。

全体スケジュール	
合計: 8 仮想ユーザ	
アクション	プロパティ
▶ 初期化	すべての仮想ユーザを同時に初期化する
仮想ユーザ開始	8 仮想ユーザを 2 ずつ 00:00:30 (HH:MM:SS) ごとにロードする
実行時間	00:10:00 (HH:MM:SS) の間実行する
仮想ユーザの停...	すべて 仮想ユーザを 2 ずつ 00:00:30 (HH:MM:SS) ごとに停止する
*	

- a. 仮想ユーザの初期化を設定します。

初期化とは、スクリプトで `vuser_init` アクションを実行し、負荷テストのための Vuser および Load Generator を準備することです。システム構成によっては、仮想ユーザを実行する前に初期化することで、より現実に近い結果が得られる場合があります。

- i. [全体スケジュール]グリッドの中で[初期化]をダブルクリックします。[アクションを編集]ダイアログ・ボックスが開き、[初期化]アクションが表示されます。



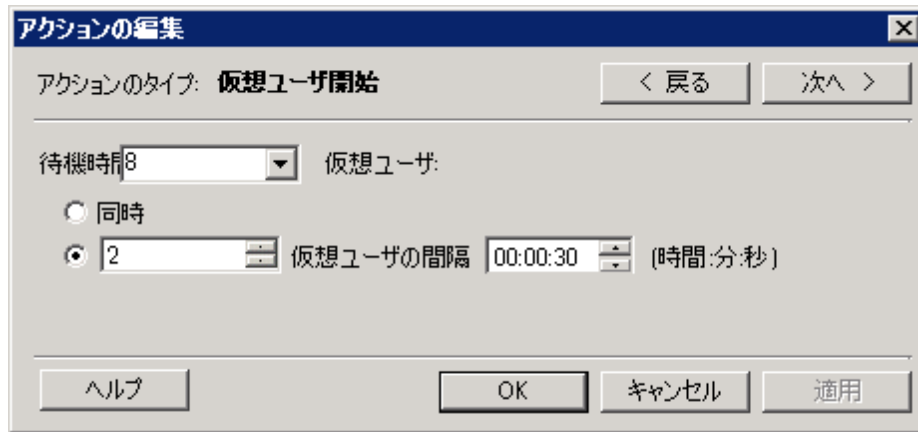
- ii. [すべての仮想ユーザを同時に初期化する]を選択します。

- iii. [OK]をクリックします。

- b. 仮想ユーザの段階的な開始を指定します。


仮想ユーザを間隔を開けて起動することにより、サイトにおける仮想ユーザの負荷が時間とともに段階的に増える状況を調べることができ、どの時点でシステムの応答時間が低下するのかを正確に特定するのに役立ちます。


- i. [全体スケジュール]グリッドの中で[仮想ユーザ開始]をダブルクリックします。[アクションを編集]ダイアログ・ボックスが開き、[仮想ユーザ開始]アクションが表示されます。



- ii. [待機期間 X 仮想ユーザ] ボックスに「8」を入力し、2 番目のオプション「2 仮想ユーザの間隔 00:00:30(30 秒)」を選択します。
 - iii. [OK] をクリックします。
- c. 期間のスケジュールを設定します。

期間を指定し、仮想ユーザがその指定の期間の間、スケジュール・アクションの実行を必ず続行するようにし、サーバにかかる連続的な負荷を測定できるようにします。期間を設定した場合には、スクリプトの実行環境の設定の中で設定した反復数に関係なく、スクリプトはその期間の間、必要な回数だけ何回でも反復して実行されます。

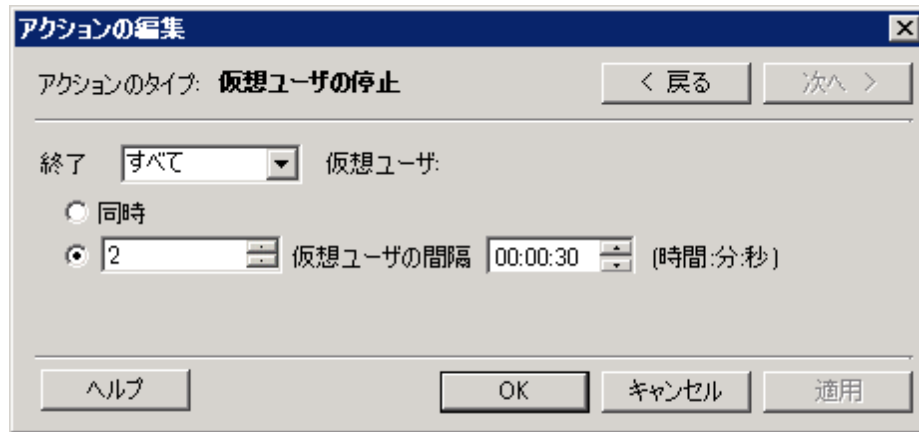
- i. 対話式スケジュール・グラフのツールバーの[編集モード]ボタン  をクリックし、対話式スケジュール・グラフが編集モードになっていることを確認します。
- ii. 対話式スケジュール・グラフで、実行時間を表す水平線をクリックします。線が強調表示され、両端に点とひし形が表示されます。

ひし形の上に凡例が表示されている場合、対話式スケジュール・グラフのツールバーの[凡例を非表示]ボタン  をクリックして、ひし形を表示します。

- iii. 括弧内の時間が **00:11:30** になるまでひし形のエンドポイントを右にドラッグします。仮想ユーザが 10 分間実行するよう設定されました。
- d. 段階的に終了するようスケジュールを設定します。

アプリケーションがしきい値に達した後にメモリ・リークの検出 やシステム復旧のチェックができるように、仮想ユーザを段階的に停止することをお勧めします。

- i. [全体スケジュール] グリッドの中で[仮想ユーザの停止]をダブルクリックします。[アクションを編集] ダイアログ・ボックスが開き、[仮想ユーザの停止] アクションが表示されます。




- ii. 2 番目のオプションを選択し、「2 仮想ユーザの間隔 00:00:30(30 秒)」と入力します。
- iii. [OK]をクリックします。

さまざまなタイプのユーザのエミュレート方法

負荷スケジュールの設定が完了したら、テスト中の仮想ユーザの動作を指定します。

実際のユーザをエミュレートするときは、ユーザの実際の動作を考慮する必要があります。動作には、ユーザがアクションの合間に一時停止する時間や、ユーザがアクションを繰り返す回数などがあります。

本項では、LoadRunner の実行環境設定について学び、思考遅延時間とログを有効にします。

1. [実行環境設定]ダイアログ・ボックスを開きます。
 - a. Controller で[デザイン]タブをクリックします。
 - b. [シナリオグループ]ペインで `travel_agents` グループを選択します。
 - c. [シナリオグループ]ツールバーの[実行環境設定]ボタン  をクリックします。[実行環境設定]ダイアログ・ボックスが開きます。



実行環境の設定では、さまざまな種類にわたるユーザの操作や動作をエミュレートできません。説明する項目は次のとおりです。

[実行論理]: 仮想ユーザが1組のアクションを繰り返す回数。

[ペースの設定]: アクションを繰り返す前の待ち時間。

[ログ]: テスト中に収集したい情報のレベル。シナリオを初めて実行するときは、最初の実行が失敗した場合にデバッグ情報が得られるように、ログ・メッセージを生成することをお勧めします。

[思考遅延時間]: ステップ間でユーザが考えるために動作を停止する時間。ユーザは各自の経験レベルや目的に応じてアプリケーションとやり取りします。そのため、技術的に習熟したユーザの方が新規ユーザよりもすばやく作業を進める可能性があります。思考遅延時間を有効にすることによって、負荷テスト中に実際のユーザをより正確にエミュレートするよう、仮想ユーザを設定できます。

[速度のシミュレーション]: モデム、DSL、ケーブルなどの異なるネットワーク接続を使用するユーザ。

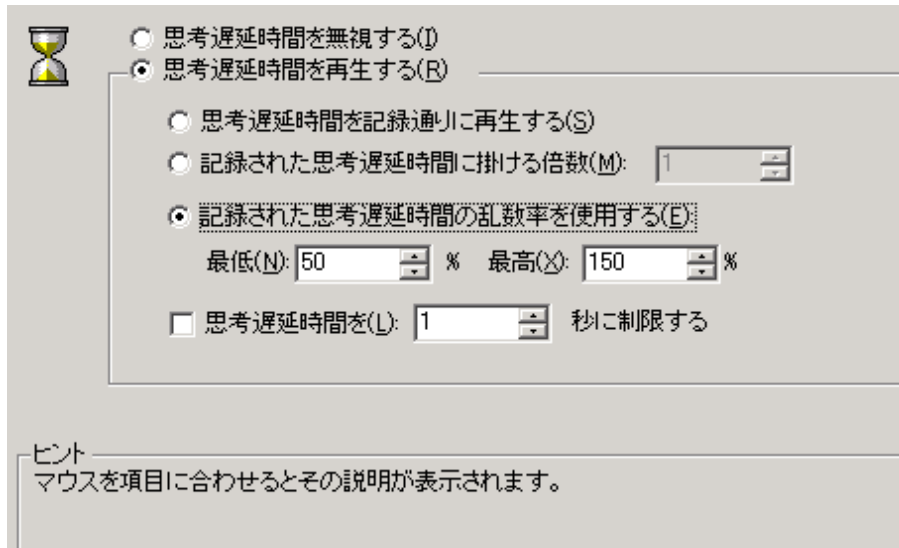
[ブラウザのエミュレーション]: さまざまなブラウザを使用してアプリケーションのパフォーマンスを確認するユーザ。

[コンテンツチェック]: ユーザ定義のエラーを自動的に検出します。

エラーが発生したときにアプリケーションからカスタム・ページを送信するとします。このカスタム・ページには必ず単語「ASP Error」が含まれており、サーバから返されるすべてのページを検索して、テキスト「ASP Error」が存在するかどうかを調べる必要があります。

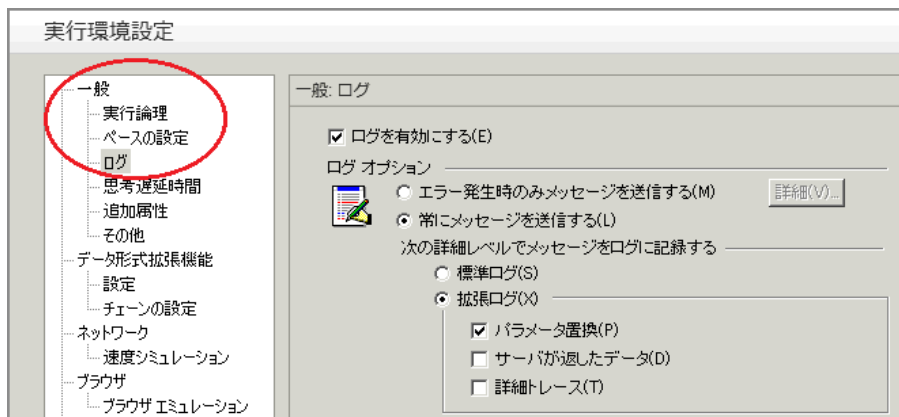
実行環境設定の[コンテンツチェック]を使用すると、テストの実行中にこのテキストを自動的に検索するようLoadRunnerを設定できます。LoadRunnerによってテキストが検索され、検出された場合はエラーが生成されます。シナリオの実行中に、コンテンツ・チェックのエラーを特定できます。

2. 思考遅延時間を有効にします。
 - a. [実行環境設定]ダイアログ・ボックスで, [一般]>[思考遅延時間]をクリックします。
 - b. [思考遅延時間を再生する]を選択し, [記録された思考遅延時間の乱数率を使用する]を選択します。
 - c. 最小値として **50%** を指定し, 最大値として **150%** を指定します。



前述の指定では, 記録された思考遅延時間の乱数率を使用して, 多様な習熟度のユーザをエミュレートします。たとえば, フライトの選択に対して記録された思考遅延時間が4秒の場合, ランダムの思考遅延時間は2秒から6秒(4の50%から150%)となります。

3. ログを有効にします。
 - a. [実行環境設定]ダイアログ・ボックスで, [一般]>[ログ]をクリックします。
 - b. [ログを有効にする]を選択します。
 - c. [ログオプション]で, [常にメッセージを送信する]を選択します。
 - d. [拡張ログ]をクリックし, [サーバが返したデータ]を選択します。



初めてデバッグを実行した後に、負荷テスト用に拡張ログを有効にすることはお勧めできません。このチュートリアルでは、仮想ユーザの出力ログの情報を提供するための目的で有効にしています。

- e. [OK]をクリックして、[実行環境設定]ダイアログ・ボックスを閉じます。
- f. シナリオを保存します。

負荷下のシステムの監視方法

テストにおける仮想ユーザの動作の設定が完了したら、モニタを設定します。

アプリケーションへの負荷の生成中に、アプリケーションがリアルタイムでどのように動作するか、また潜在的なボトルネックがどこに存在するか確認します。負荷テスト中にシステムの層、サーバ、コンポーネントごとのパフォーマンスを測定するには、LoadRunnerの統合モニタ・スイートを使用します。LoadRunnerには、Webサーバ、アプリケーション・サーバ、データベース・サーバ、ERP/CRMサーバなど、さまざまな主要バックエンド・システム・コンポーネントに対応したモニタがあります。

たとえば、稼働中のWebサーバのタイプに応じたWebサーバ・リソース・モニタを選択できます。また、たとえばIISなど関連するモニタ用のライセンスを購入し、そのモニタを使用して、IISリソースに反映された問題を特定できます。

本項では、Windowsリソース・モニタの追加方法と設定方法を学びます。このモニタを使用すると、CPU、ディスク、メモリの各リソースについて、負荷の影響を調べることができます。

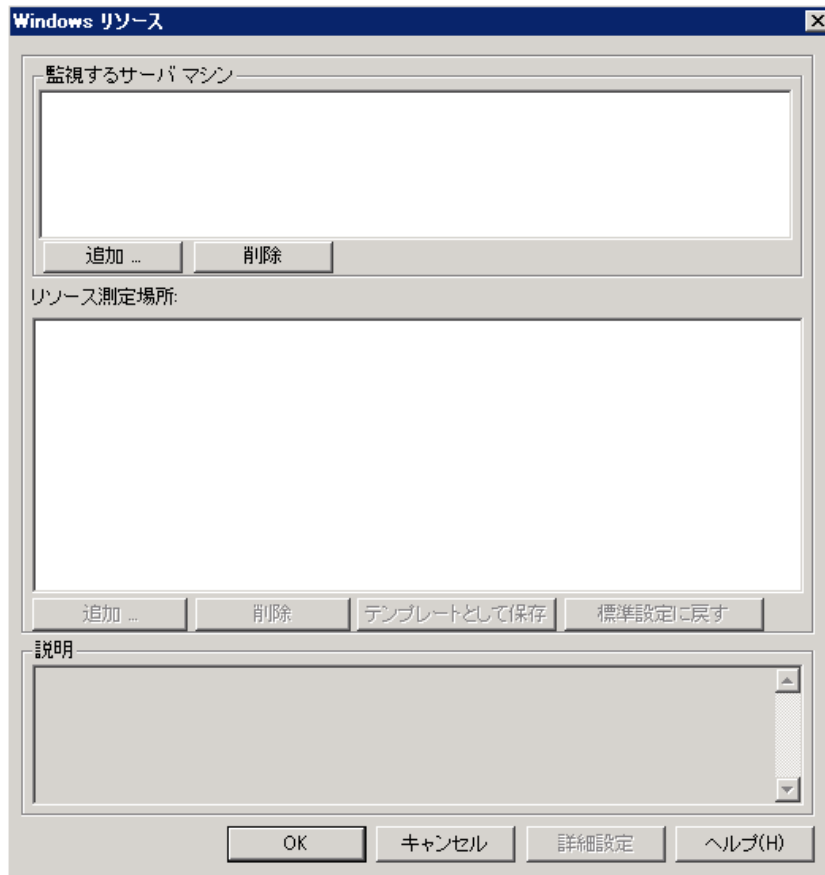
1. Windowsリソース・モニタを選択します。
 - a. Controllerの下部の[実行]タブをクリックします。

[Windowsリソース]グラフは、グラフ表示領域に表示される4つの標準グラフの1つです。ほかのグラフを開く方法については次の練習で学びます。

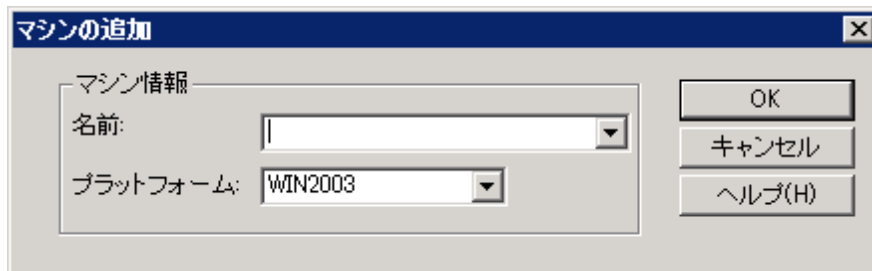


- b. [Windowsリソース]グラフ内を右クリックし、[測定値の追加]を選択します。[Windowsリ

ソース]ダイアログ・ボックスが開きます。

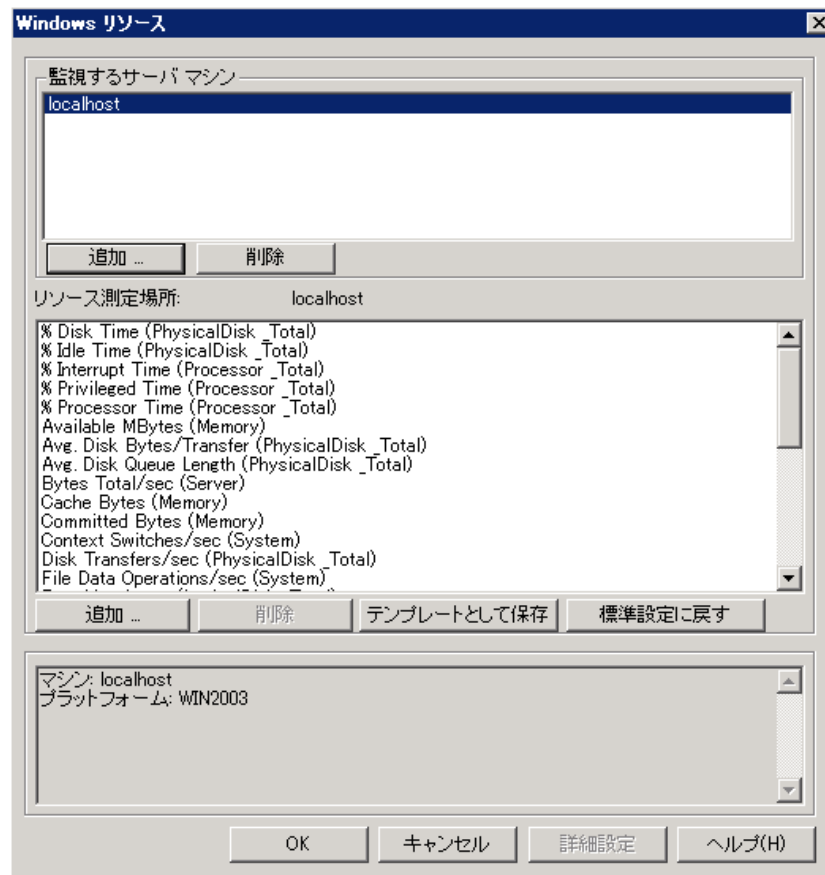


2. 監視対象のサーバを選択します。
 - a. [Windows リソース]ダイアログ・ボックスの[監視するサーバマシン]セクションで, [追加]をクリックします。[マシンの追加]ダイアログ・ボックスが開きます。



- b. [名前]ボックスに「localhost」と入力します。(Load Generator を別のマシンで実行している場合は, そのマシンのサーバ名または IP アドレスを入力します)。
 - c. [プラットフォーム]リストから, マシンを実行するプラットフォームを選択します。
 - d. [OK]をクリックします。

標準の Windows リソース測定値が[リソース測定場所 <サーバ・マシン>]に表示されます。



3. モニタを起動します。

[Windows リソース]ダイアログ・ボックスで[OK]をクリックすると、ダイアログ・ボックスが閉じてモニタが起動します。

次の練習について

この練習で、負荷テスト・シナリオをデザインする作業が完了しました。次に、「練習 6: 負荷テストの実行」(59ページ)に進みます。

練習 6: 負荷テストの実行

負荷テストを実行すると、LoadRunnerにより、システムに対して負荷が生成されます。この後、LoadRunnerのモニタとグラフを使用して、負荷がかかった場合のシステムのパフォーマンスを観察できます。

この練習は、次の項目で構成されています。

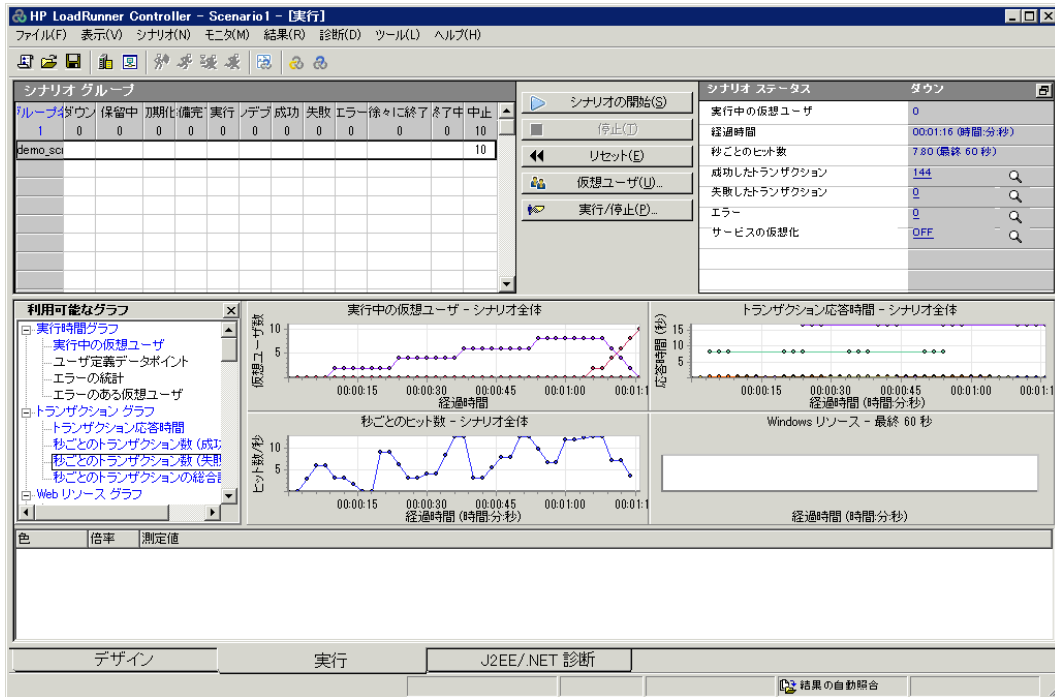
- Controllerの実行ビューの概要
- 負荷テスト・シナリオの実行方法
- リアルタイムで実行しているユーザの監視方法
- ユーザ・アクションのサマリの表示場所
- テスト中に負荷を増やす方法
- 負荷下のアプリケーションの動作方法
- アプリケーションでエラーが発生したかどうか
- テストの実行が完了したことを知る方法
- アプリケーションが負荷のかかった状態で良好に実行されたかどうか

Controllerの実行ビューの概要

Controllerの[実行]タブは、シナリオの管理および監視を行うための中心となる場所です。[実行]タブ(Controllerウィンドウの下部)は、5つのペインで構成されます。

1. **[シナリオグループ]ペイン:** 左上のペインには、シナリオ・グループの仮想ユーザのステータスが表示されます。このペインの右側にあるボタンを使用して、シナリオの開始、停止、およびリセットを行い、個々の仮想ユーザのステータスを表示します。また、仮想ユーザを手動でさらに追加してシナリオ実行中にアプリケーションに対する負荷を増やすためのボタンもあります。
2. **[シナリオステータス]ペイン:** 右上のペインで、実行中の仮想ユーザの数や、仮想ユーザの個々のアクションのステータスなど、負荷テストのサマリを表示できます。
3. **[利用可能なグラフ]ペイン:** 左側の中央のペインには、LoadRunnerグラフのリストが表示されます。グラフを開くには、ツリーでグラフを選択し、グラフ・表示領域にドラッグします。
4. **[グラフの表示]ペイン:** 右側の中央のペインでは、1～8個のグラフが表示されるようカスタマイズできます([表示]>[グラフの表示])。
5. **[グラフの凡例]ペイン:** 下部のペインには、選択したグラフのデータが表示されます。

[実行]タブのコントロールがControllerの下部にあります。



負荷テスト・シナリオの実行方法

本項では、シナリオを開始します。

1. Controller の実行ビューを開きます。

Controller の下部の[実行]タブをクリックします。

[シナリオ グループ] ペインの[ダウン] カラムに 8 個の仮想ユーザが存在することがわかります。これらはシナリオの作成時に作成した仮想ユーザです。

シナリオ グループ													
グループ名	ダウン	保留中	初期化	準備完	実行	デバッグ	成功	失敗	エラー	徐々に終了	終了中	中止	
1	10	0	0	0	0	0	0	0	0	0	0	0	
travel_agent	10												

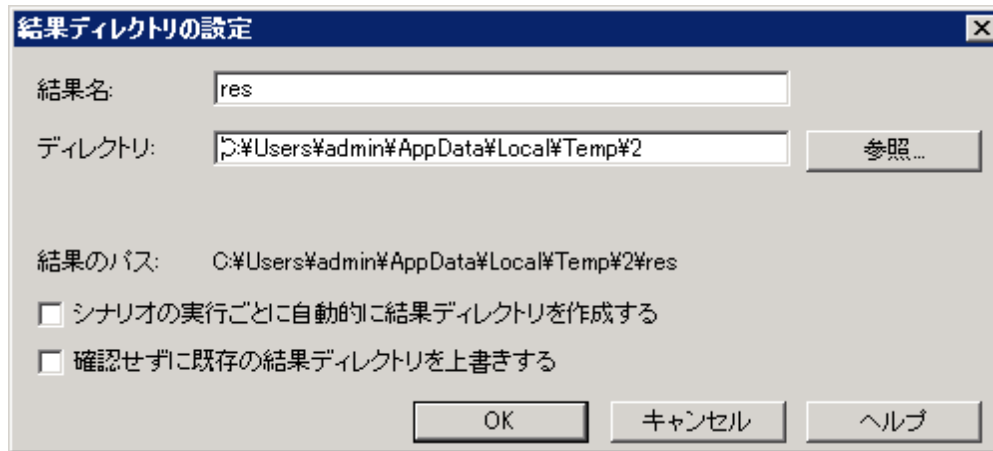
シナリオはまだ実行されていないため、ほかのカウンタはすべて 0 のままで、グラフ表示領域内のグラフは ([Windows リソース] を除き) すべて空になっています。次の手順でシナリオの実行を開始すると、グラフとカウンタは情報を表示し始めます。

2. シナリオを開始します。

[シナリオの開始] ボタンをクリックするか、[シナリオ]>[開始] を選択して、シナリオの実行を開始します。

チュートリアルを初めて実行している場合は、Controller によってシナリオが開始されます。結果のファイルは Load Generator の temp フォルダに自動的に保存されます。

テストを繰り返す場合は、既存の結果ファイルを上書きしてもよいか尋ねられます。初回の負荷テストの場合は、その結果を以降の負荷テストの結果と比較する際の基準として使用するので、[いいえ] をクリックします。[結果ディレクトリの設定] ダイアログ・ボックスが表示されます。



新しい結果フォルダを指定します。グラフを分析する際にいくつかのシナリオ実行結果を重ね合わせるように、結果セットごとに、わかりやすい一意の名前を入力します。

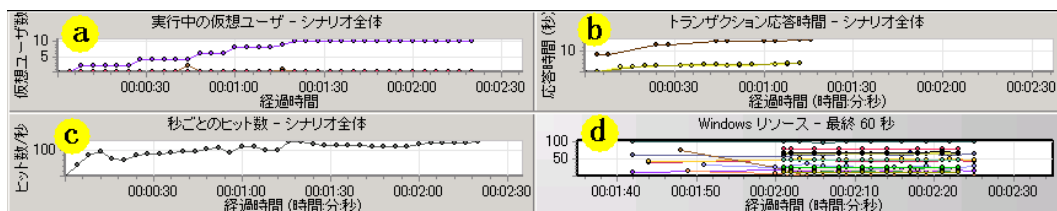
負荷下のアプリケーションの監視方法

Controller のオンライン・グラフは、モニタによって収集されたパフォーマンス・データを表示するのに使用します。この情報を使用して、システム環境において潜在的な問題のある領域を特定します。

1. パフォーマンス・グラフを検証します。

[実行] タブの [グラフの表示] ペインには、以下の標準グラフが表示されます。

- a. [実行中の仮想ユーザ - シナリオ全体] グラフ: 任意の時点において実行されている仮想ユーザの数を表示します。
- b. [トランザクション応答時間 - シナリオ全体] グラフ: 各トランザクションが完了するのに要した時間を表示します。
- c. [秒ごとのヒット数 - シナリオ全体] グラフ: シナリオ実行時の経過秒ごとの、仮想ユーザが Web サーバに対して行ったヒット (HTTP 要求) の数を表示します。
- d. [Windows リソース] グラフ: シナリオの実行中に測定された Windows のリソースが表示されます。



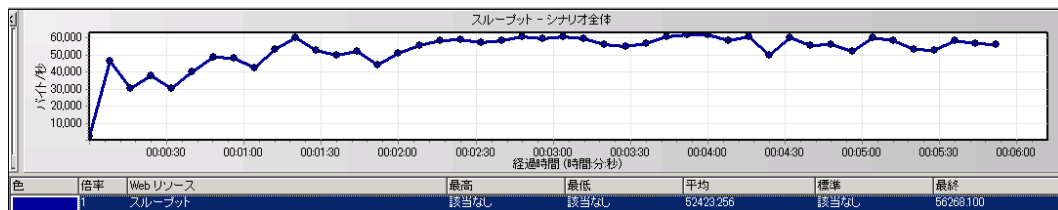
2. 個々の測定値を強調表示します。
 - a. [Windows リソース] グラフをダブルクリックすると、[グラフの表示] ペイン全体に合わせて拡大表示されます。各測定値は、[グラフの凡例] ペインの色分けされた行に表示されます。各行は、グラフ上の同じ色の線に対応しています。
 - b. 行を選択すると、対応するグラフの線が強調表示されます。また、その逆の場合も同様です。
 - c. グラフを元のサイズに縮小するには、グラフをもう一度ダブルクリックします。

3. スループット情報を表示します。

[利用可能なグラフ]ペインの[Web リソース グラフ]の下で[スループット]グラフを選択し、[グラフの表示]ペインまでドラッグします。[スループット]グラフの測定値が[グラフの表示]ペインと[グラフの凡例]ペインに表示されます。

[スループット]グラフには、任意の指定された1秒間に仮想ユーザがサーバから受け取ったデータ量(バイト単位)が示されます。このグラフと[トランザクション応答時間]グラフとを比較することで、スループットがトランザクションのパフォーマンスに与える影響を調べることができます。

時間の経過と仮想ユーザ数の増加に伴ってスループットが増えている場合は、帯域幅が十分なことを示しています。仮想ユーザ数が増加してもグラフが比較的平坦な場合は、提供されているデータの量が帯域幅によって制限されていると考えられます。

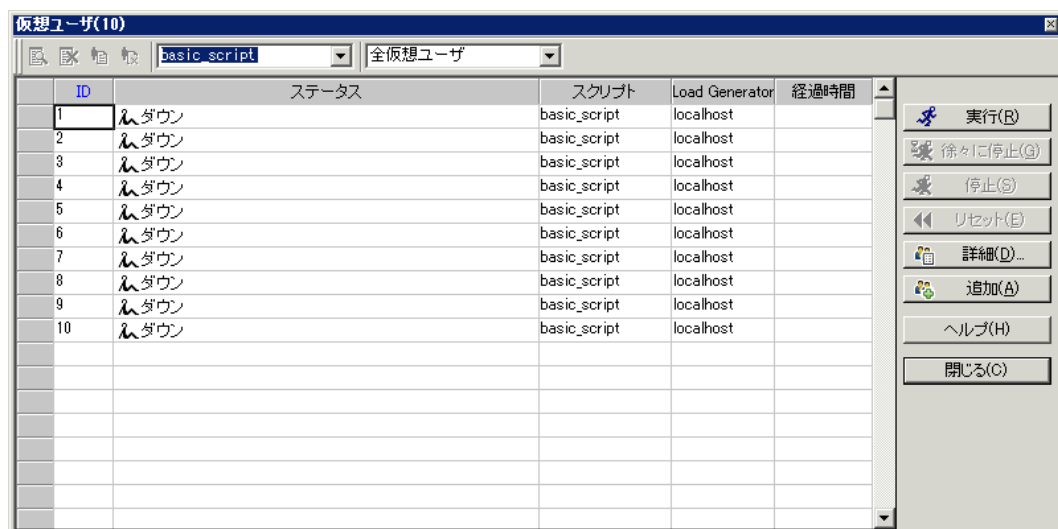


リアルタイムで実行しているユーザの監視方法

ユーザをエミュレートするときは、仮想ユーザのアクションをリアルタイムで表示し、各ユーザが正しいアクションを実行していることを確かめることができます。Controller では、実行時ビューアを使用してアクションをリアルタイムで表示できます。



仮想ユーザのアクションを視覚的に観測するには、次の手順を実行します。

1. Controller の[実行]タブで、[仮想ユーザ] ボタンをクリックします。[Vusers] ダイアログ・ボックスが開きます。



[ステータス]カラムに、各仮想ユーザのステータスが表示されます。前の例では4個の仮想ユーザのステータスが実行中で、別の4個の仮想ユーザのステータスがダウンであることがわかります。スケジューラの[仮想ユーザ開始]アクションによって、一度に2個の仮想ユーザを解放するよう


Controllerに指示されました。シナリオの実行が進むと、引き続き仮想ユーザが30秒間隔でグループごとに2個追加されます。

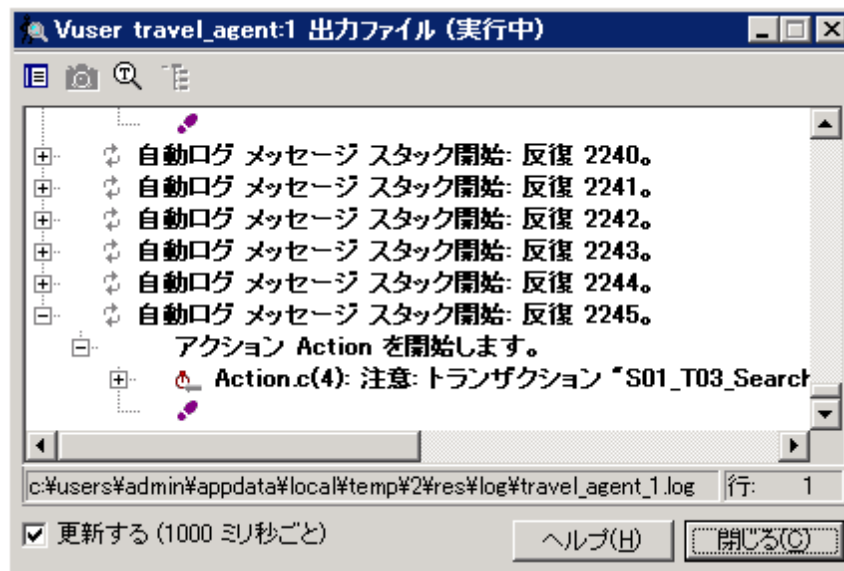
2. 仮想ユーザ・リストで、実行中の仮想ユーザを選択します。
3. 仮想ユーザ・ツールバーの[選択した仮想ユーザを表示]ボタンをクリックします。実行時ビューアが表示され、選択した仮想ユーザが実行しているアクションが表示されます。仮想ユーザ・スクリプトのステップを仮想ユーザがひとつおとり実行するにつれて、実行時ビューアが更新されます。
4. 仮想ユーザ・ツールバーの[選択した仮想ユーザを非表示]ボタンをクリックして、実行時ビューアを閉じます。

仮想ユーザ・アクションのサマリの表示場所

テスト実行中における個々の仮想ユーザの進行状況を確認するために、仮想ユーザのアクションのテキスト・サマリが格納されたログ・ファイルを表示できます。

仮想ユーザのアクションのテキスト・サマリを確認するには、次の手順を実行します。

1. [仮想ユーザ]ウィンドウで、実行中の仮想ユーザを選択します。
2. 仮想ユーザ・ツールバーの[仮想ユーザログの表示]ボタンをクリックします。[仮想ユーザログ]ダイアログ・ボックスが開きます。



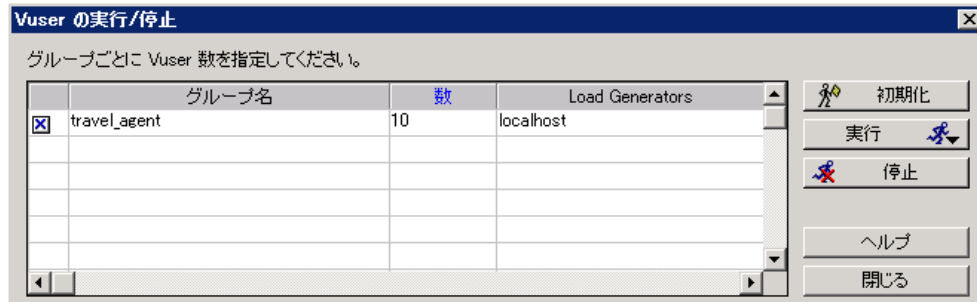
ログには、仮想ユーザのアクションに対応したメッセージがあります。たとえば、このウィンドウにあるメッセージ「仮想ユーザ・スクリプトが開始されました」は、仮想ユーザの実行の開始を示しています。ログの最後までスクロールし、仮想ユーザがアクションを実行するごとに新しいメッセージが追加されていることを確かめます。

3. [仮想ユーザログ]ダイアログ・ボックスと[仮想ユーザ]ダイアログ・ボックスを閉じます。

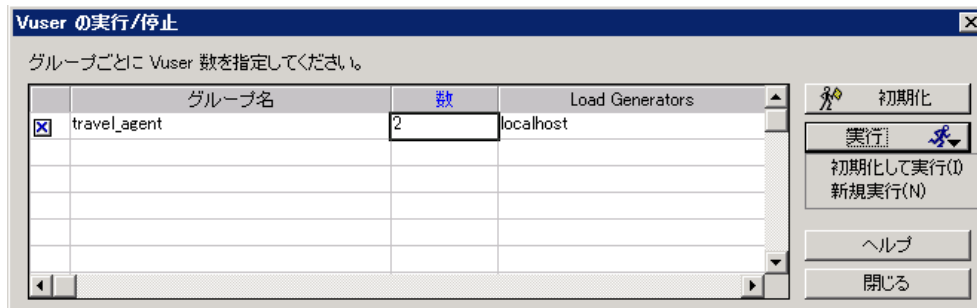
テスト中に負荷を増やす方法

より多くのユーザを手動で追加することによって、負荷テストの実行中にシステムにかかる負荷を増やすことができます。

1. 負荷テスト中に負荷を増やすには、次の手順を実行します。
 - a. Controller で[実行]タブが表示されていることを確認します。
 - b. [Vusers ユーザの実行/停止]ボタンをクリックします。[Vusers の実行/停止]ダイアログボックスが開き、シナリオの中で実行するために現在割り当てられている仮想ユーザの数が表示されます。



- c. [数]カラムには、グループに追加する仮想ユーザの数を入力します。2 個の仮想ユーザを追加して実行するには、[数]カラムの数値 8 を数値 2 に置き換えます。
- d. [実行]をクリックして仮想ユーザを追加します。
- e. 元の仮想ユーザのいくつかはまだ初期化されていない場合は、[初期化して実行]および[新規実行]オプションが表示されます。[新規実行]オプションを選択します。



これら 2 個の追加の仮想ユーザは **travel_agent** グループに配分され、**localhost** Load Generator で実行されます。これで、[シナリオステータス]ペインには、10 個の仮想ユーザが実行されていることが示されます。

追加された仮想ユーザを LoadRunner Controller が起動できないことを示す警告メッセージが表示されることがあります。これは、ローカルマシンを Load Generator として使用していて、メモリリソースが限られていることが原因で起こります。この問題を避けるため、ほとんどの場合、専用のマシンを Load Generator として使用します。

負荷下のアプリケーションの動作方法

[シナリオステータス]ペイン([実行]タブにあります)に実行中のシナリオのサマリを確認します。[シナリオステータス]ペインでは、どの仮想ユーザのアクションがアプリケーションの問題の原因になっているかをドリルダウンして調べることができます。失敗トランザクションやエラーの数が多い場合、アプリケーションが負荷のかかった状態で期待どおりに実行されていないことを示す場合があります。

1. シナリオのステータスを表示します。

[シナリオステータス]ペインのヘッダに、シナリオの全体的なステータスが表示されます。



シナリオ ステータス	
実行中の仮想ユーザ	0
経過時間	00:00:12 (時間:分:秒)
秒ごとのヒット数	0.00 (最終 60 秒)
成功したトランザクション	0
失敗したトランザクション	0
エラー	0
サービスの仮想化	OFF

2. 仮想ユーザ・アクションのブレークダウンを表示します。

- a. [シナリオステータス]ペインにある[成功したトランザクション]をクリックします。[トランザクション]ダイアログ・ボックスが開き、トランザクションの詳細のリストが表示されます。



名称	TPS	通過	失敗	停止
S01_T02_Login	0.3	20	0	0
S01_T03_SearchFlight	0.3	20	0	0
S01_T07_SignOff	0.1	20	0	0
vuser_end_Transaction	0.3	20	0	0
vuser_init_Transaction	0.1	20	0	0
vuser_end_Transaction	0.1	20	0	0

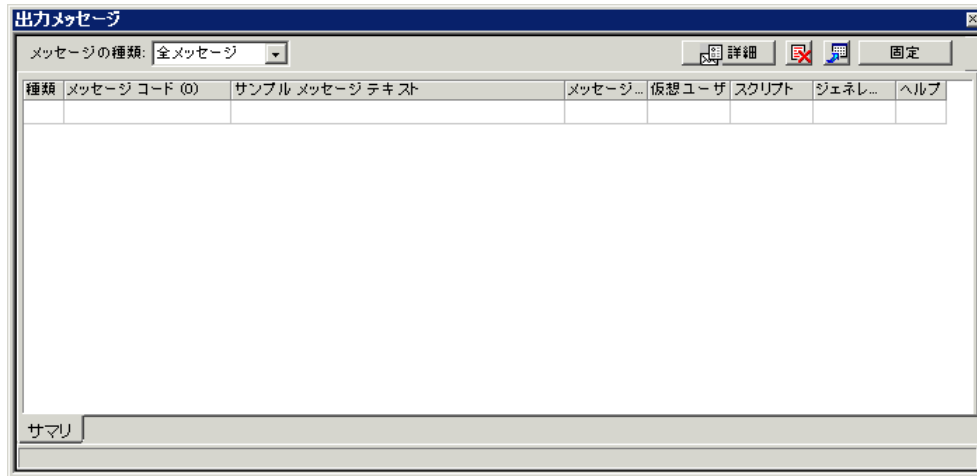
- b. [閉じる]をクリックし、[トランザクション]ダイアログ・ボックスを閉じます。

アプリケーションでエラーが発生したかどうか

大きな負荷のかかった状態でアプリケーションが失敗し始める場合は、エラーや失敗トランザクションが発生した可能性があります。Controller はこれらのエラー・メッセージを[出カウインドウ]に表示します。

1. エラーまたは警告メッセージがないか確認します。
 - a. [表示]>[出力メッセージを表示]を選択します。[警告]や[エラー]など、タイプによってメッセージをソートできます。

[出力メッセージ]ダイアログ・ボックスが開き、メッセージ・テキスト、生成されたメッセージの総数、エラーを生成した仮想ユーザとLoad Generator、およびエラーが発生したスクリプトが表示されます。



- b. メッセージの詳細を表示するには、そのメッセージを選択して、[詳細]をクリックします。[詳細メッセージテキスト]ボックスが開き、詳しいメッセージ・テキストが表示されます。
2. ログ情報の詳細を表示します。

エラー・コードに関連する、個々のメッセージ、仮想ユーザ、スクリプト、およびLoad Generatorの情報は、該当するカラムの青いリンクをクリックして表示できます。

たとえば、スクリプトの中でエラーが発生した場所を探すには、[メッセージの総数]カラムをドリルダウンします。[出力ウィンドウ]には、選択したエラー・コードのすべてのメッセージが一覧表示されます。また、時間、反復数、およびエラーが発生したスクリプト内の行も表示されます。

3. [行番号]カラムをドリルダウンします。

VuGenが開き、エラーが発生したスクリプト内の行が表示されます。これらの情報をもとに、負荷時においてアプリケーションが失敗している原因となっている、応答時間の遅いトランザクションを特定できます。

シナリオの実行が完了したことを知る方法

シナリオの実行が完了すると、[シナリオステータス]ペインのヘッダに「ダウン」ステータスが表示されます。これは、シナリオのすべての仮想ユーザの実行が完了したことを示します。

シナリオ ステータス	
実行中の仮想ユーザ	0
経過時間	00:00:12 (時間:分:秒)
秒ごとのヒット数	0.00 (最終 60 秒)
成功したトランザクション	0
失敗したトランザクション	0
エラー	0
サービスの仮想化	OFF


[Vusers]ダイアログ・ボックスを開き、個々の仮想ユーザのステータスを調べることができます。
[Vusers]ダイアログ・ボックスには、各仮想ユーザが実行を繰り返した反復回数、成功した反復の数、および経過時間が表示されます。

ID	ステータス	スクリプト	Load Generator	経過時間
1*	中止 反復試行回数: 6 成功数: 6	basic_script	localhost	00:01:29
2*	中止 反復試行回数: 6 成功数: 6	basic_script	localhost	00:01:29
3*	中止 反復試行回数: 5 成功数: 5	basic_script	localhost	00:01:15
4*	中止 反復試行回数: 5 成功数: 5	basic_script	localhost	00:01:15
5*	中止 反復試行回数: 4 成功数: 4	basic_script	localhost	00:01:00
6*	中止 反復試行回数: 4 成功数: 4	basic_script	localhost	00:01:00
7*	中止 反復試行回数: 3 成功数: 3	basic_script	localhost	00:00:45
8*	中止 反復試行回数: 3 成功数: 3	basic_script	localhost	00:00:45
9*	中止 反復試行回数: 2 成功数: 2	basic_script	localhost	00:00:30
10*	中止 反復試行回数: 2 成功数: 2	basic_script	localhost	00:00:29

システムが負荷のかかった状態で良好に実行されたかどうか

システムが負荷のかかった状態で良好に実行されたかどうかを確かめるためには、トランザクションの応答時間を調べ、その応答時間が許容し得る範囲内に収まったかどうかを確かめる必要があります。シナリオ中にトランザクションの応答時間が長くなっている場合は、ボトルネックを突き止める必要があります。これについては、「練習 7: シナリオの分析」(69ページ)でさらに詳しく学びます。

問題を切り分けたら、開発者、DBA、ネットワークその他のシステム技術者とともに、問題を修正するための共同作業が必要になります。調整作業が完了したら、負荷テストを繰り返し実行して、望ましい結果が得られることを確認します。以上の過程を繰り返すことによって、システムのパフォーマンスを最適化します。

同じ設定で再度実行できるようにシナリオを保存するには、[ファイル]>[保存]をクリックするか、Controller のツールバーの[上書き保存]ボタン  をクリックします。

次の練習について

以上で、簡単な負荷テスト・シナリオの実行と表示の方法について学びました。次に、「練習 7: シナリオの分析」(69ページ)に進みます。

練習 7: シナリオの分析

前の練習では、シナリオ実行の設計、制御、および実行の方法を学習しました。サーバをロードしたら、システムのパフォーマンスを改善するために、実行を分析して取り除く必要のある問題を特定します。

Analysis のセッション中に生成されるグラフおよびレポートには、シナリオのパフォーマンスに関する重要な情報が示されています。Analysis グラフおよびレポートを使用すれば、アプリケーションのボトルネックを簡単に特定でき、パフォーマンスを改善するにはシステムのどの部分の修正が必要かがわかります。

この練習は、次の項目で構成されています。

- Analysis セッションの処理方法
- Analysis セッションの開始方法
- Analysis のウィンドウの概要
- ゴールの達成の確認(サービス・レベル・アグリーメント)
- パフォーマンス・サマリの表示
- パフォーマンスのグラフによる表示方法
- サーバのパフォーマンスの確認
- 問題の原因の特定方法
- シナリオ実行で収集可能なその他の情報
- 結果の発行方法
- 結論

Analysis セッションの処理方法

Analysis セッションの目的は、次のようにシステムのパフォーマンスの不具合を発見し、不具合の原因を特定することです。

- テストの期待値は満たされたでしょうか。負荷のかかった状態でのユーザのトランザクション時間はどうかでしょうか。SLA はゴールを達成しましたか、それとも外してしまいましたか。トランザクションの平均トランザクション時間はどうかでしょうか。
- システムのどの部分がパフォーマンス低下の原因となったのでしょうか。ネットワークおよびサーバの時間はどうかでしょうか。
- トランザクション時間とバックエンド・モニタ・マトリックスを相関させることで、考えられる原因を発見できるでしょうか。

次の項では、LoadRunner Analysis を起動して、パフォーマンスの問題を発見し問題の原因を特定するのに役立つグラフおよびレポートを作成、表示する方法を学習します。

Analysis セッションの開始方法

1. HP LoadRunner Analysis を開きます。

デスクトップ上の Analysis アイコンをダブルクリックし、[スタート]>[すべてのプログラム]>[HP ソフトウェア]>[HP LoadRunner]>[Analysis]を選択します。HP LoadRunner Analysis が起動します。

2. Analysis セッション・ファイルを開きます。

本項ではより多様な結果を説明するため、前の練習で実行したものとほぼ同じシナリオを実行しました。ただし、今回のシナリオでは、仮想ユーザを 10 個ではなく 70 個使用しています。このシナリオの結果に基づいて作成された Analysis セッションを開きます。

- a. Analysis のウィンドウで、[ファイル]>[開く]を選択します。[Analysis セッションを開く]ダイアログ・ボックスが開きます。

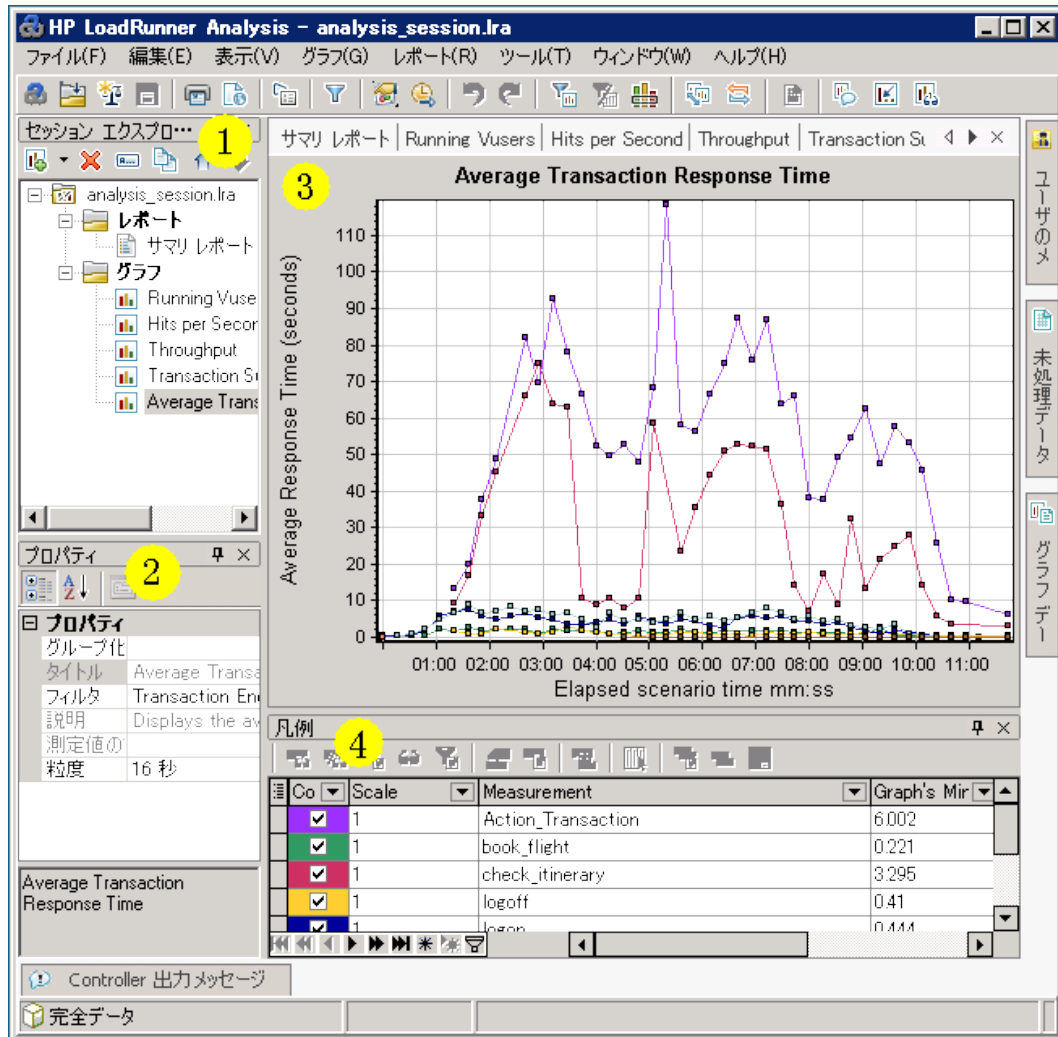


- b. <LoadRunner のインストール先>\tutorial フォルダで analysis_session を選択し、[開く]をクリックします。Analysis がセッション・ファイルを開きます。

Analysis のウィンドウの概要

Analysis には、次の主要なペインがあります。

1. セッション・エクスプローラ
2. [プロパティ]ペイン
3. [グラフの表示]ペイン
4. [凡例]ペイン



1. **セッション・エクスプローラ:**左上のペインには、閲覧できるレポートおよびグラフが表示されています。このペインで、Analysisの起動時に表示されない新しいレポートまたはグラフを表示したり、不要なものを削除したりできます。
2. **[プロパティ]ペイン:**左下のペインには、[セッションエクスプローラ]で選択したグラフやレポートの詳細が[プロパティ]ペインに表示されます。黒で表示されたフィールドは編集可能です。
3. **[グラフの表示]ペイン:**右上のペインにはグラフが表示されます。標準設定では、セッションを開くとこの領域にサマリ・レポートが表示されます。
4. **[凡例]ペイン:**右下のペインには、選択したグラフのデータが表示されます。

注: ツールバーからアクセスできる追加のペインがあります。これらのペインは、画面上の任意の場所にドラッグしてドロップできます。

ゴールの達成の確認(サービス・レベル・アグリーメント)

本項では、サービス・レベル・アグリーメント(SLA)を紹介します。

SLA とは負荷テストシナリオに対して定義した特定のゴールです。Analysis はこれらのゴールを、実行中に LoadRunner が収集、保存したパフォーマンスに関連したデータと比較し、ゴールの SLA ステータスを判断します([成功]または[失敗])。

たとえば、スクリプト内のトランザクションの平均トランザクション時間に、特定のゴール、またはしきい値を定義することができます。テスト実行後、LoadRunner は定義したゴールを実際に記録されたトランザクション時間の平均と比較します。Analysis は定義した各 SLA のステータスを、[成功]または[失敗]と表示します。たとえば、実際の平均トランザクション時間が定義したしきい値を超えなかった場合、SLA ステータスは[成功]になります。

ゴール定義の1部として、負荷条件を考慮するよう SLA を設定できます。つまり、**実行中の仮想ユーザ数**、**スループット**などの負荷のレベルに応じて、許容するしきい値が変化します。負荷が増加するに従い、より高いしきい値が許容されます。

定義されたゴールに従い、LoadRunner は次の方法で SLA ステータスを判断します。

- **SLA ステータスが経過時間中の時間範囲内に決まるもの** Analysis は実行のある時間内に、設定された時間間隔(たとえば、5 秒ごと)で SLA ステータスを表示します。
- **SLA ステータスが実行の全体によって決まるもの** Analysis はシナリオ実行全体に対して1つの SLA ステータスを表示します。

SLA はシナリオ実行前に Controller で、またはシナリオ実行後に Analysis で定義できます。

次の項では、HP Web Tours の例を使用して SLA を定義します。HP Web Tours の管理者が **book_flight** および **search_flight** トランザクションの平均トランザクション時間が一定の値を超えるたびにそれを知りたいと想定します。これを実現するには、トランザクションを選択し、次にしきい値を設定します。これらのしきい値は、許容可能な平均トランザクション時間の最大値です。

また、特定の負荷条件を考慮してこれらのしきい値を設定します。この場合は実行仮想ユーザ数です。つまり、実行仮想ユーザ数が増えるにつれ、しきい値は上昇します。

なぜなら、HP Web Tours の管理者は平均トランザクション時間を可能なかぎり低く抑えたいですが、1年の特定の時期にほかの時期に比べて高い負荷を HP Web Tours サイトが処理しなければならないと想定するのが妥当だからです。たとえば、旅行シーズンのピーク期間中、より多くの旅行エージェントがフライトの予約、日程の確認などのためにサイトにログインします。この高い負荷は妥当であるため、この時期の多少長い平均トランザクション時間は許容されます。

軽い負荷、平均負荷、重い負荷という、3つの負荷シナリオを考慮するよう SLA を設定します。各シナリオはそれぞれのしきい値を持ちます。

SLA の定義方法

SLA はシナリオの実行後、Analysis で定義します。

注: シナリオの実行前に Controller で SLA を定義することを推奨します。ただし、このチュートリ

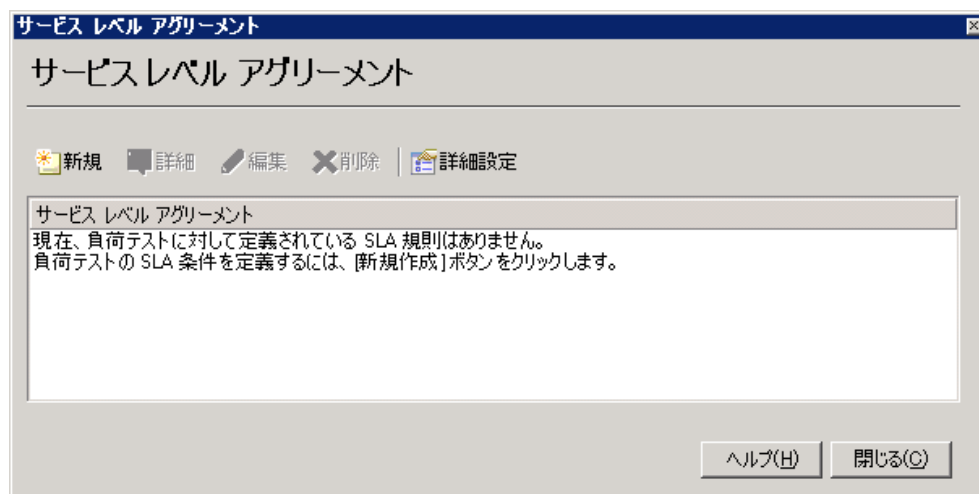
アルでは前の練習で実行したものと同一のテスト・シナリオを分析していないため、SLAを Analysis で定義します。ControllerでSLAを定義するには、[デザイン]タブの[サービスレベルアグリーメント]セクションで[新規作成]をクリックします。

サンプルのセッション・ファイル内の **book_flight** および **search_flight** トランザクションの平均トランザクション時間に特定のゴールを設定するSLAを定義します。

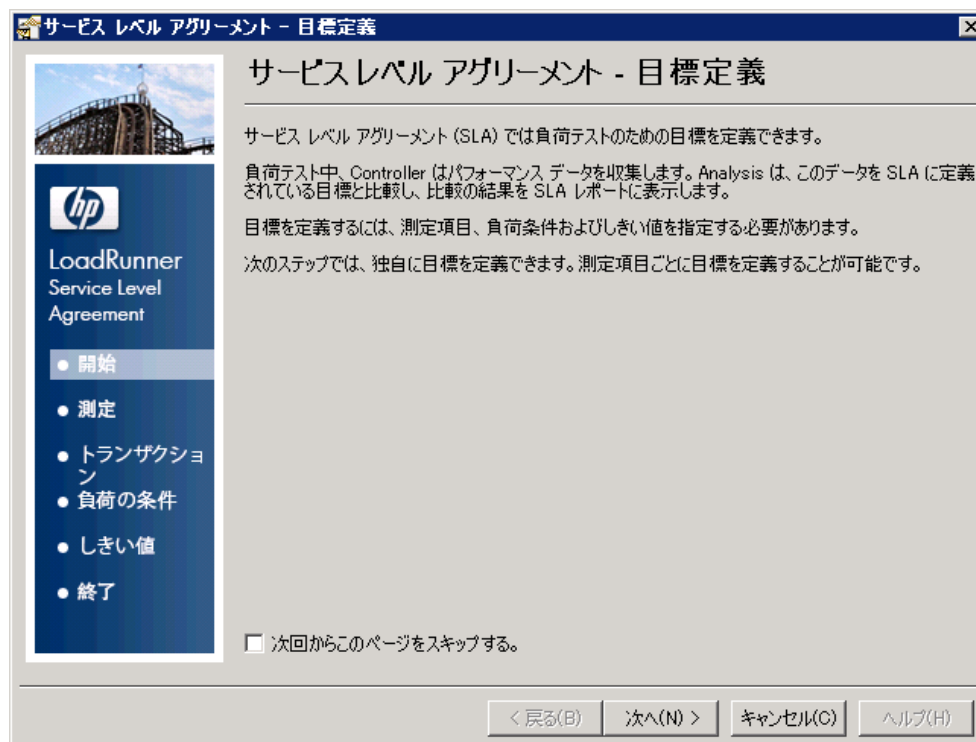
平均トランザクション時間は、設定した時間間隔で実行中に測定します。

SLAを定義するには次の手順を実行します。

1. SLA ウィザードを開きます。
 - a. LoadRunner Analysis で、[ツール]>[SLA ルールの設定]を選択します。[サービスレベルアグリーメント]ダイアログ・ボックスが開きます。

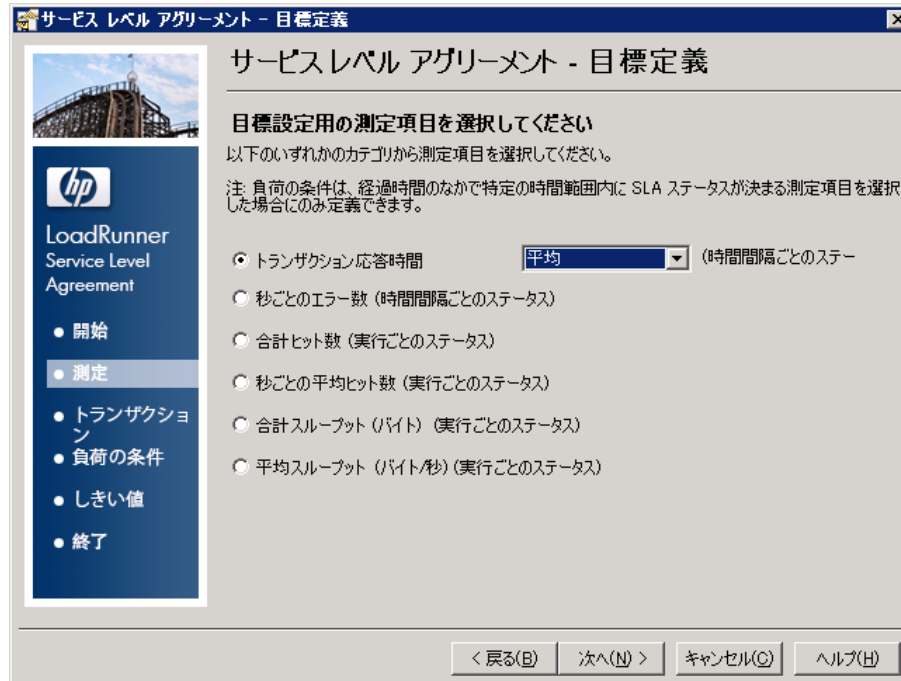


- b. [新規作成]をクリックしてサービス・レベル・アグリーメント・ウィザードを開きます。



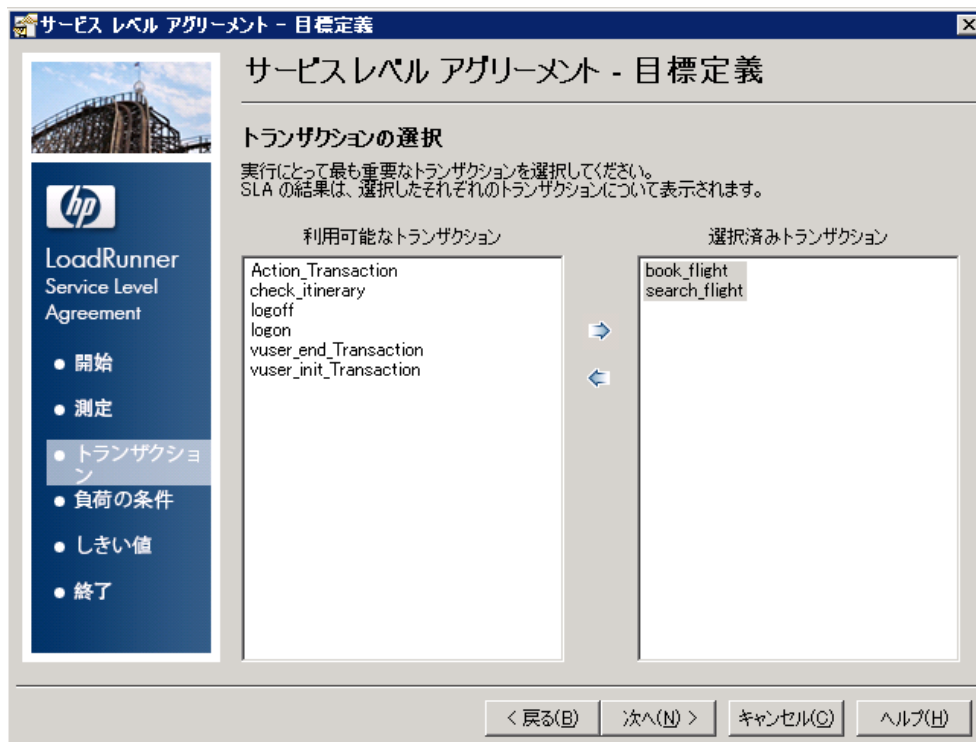
サービス・レベル・アグリーメント・ウィザードを初めて開く場合、[開始]ページが表示されます。次にウィザードを実行したときにこのページが表示されないようにするには、[次回からこのページをスキップする]チェック・ボックスを選択します。

- c. [次へ]をクリックします。
2. ゴールの測定方法を選択します。
 - a. [目標設定用の測定項目を選択してください]ページで、[トランザクション応答時間：平均]を選択します。



- b. [次へ]をクリックします。
3. 監視するトランザクションを選択します。

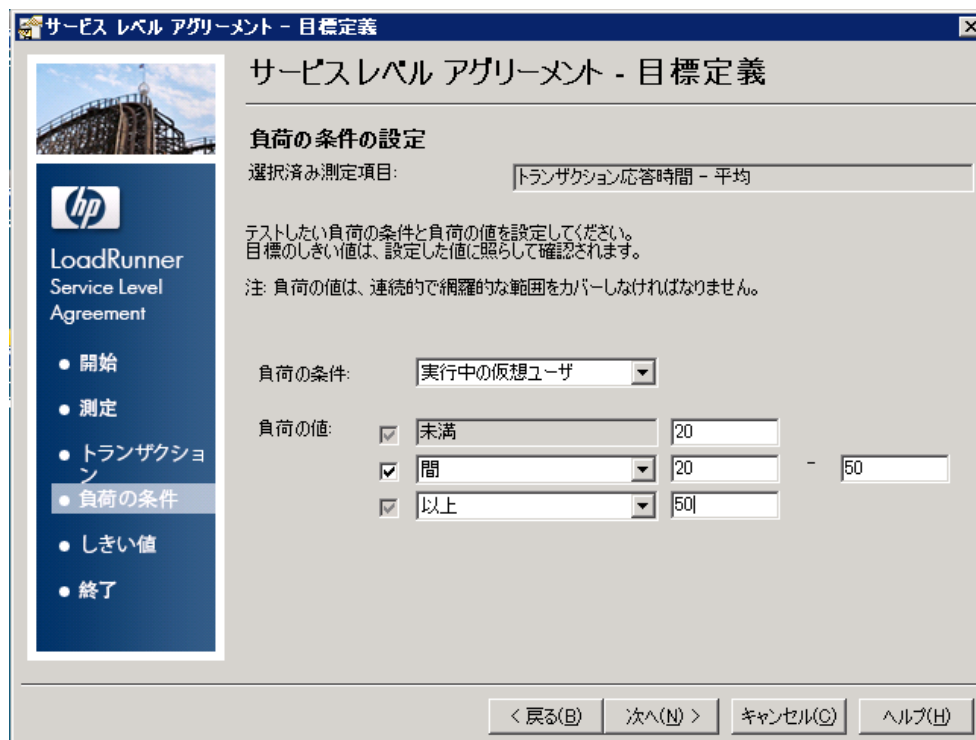
[選択済みのトランザクション] ページで、監視するトランザクションを[利用可能なトランザクション]リストから選択します。
 - a. **check_itinerary** トランザクションをダブルクリックして、[選択済みのトランザクション]リストに移動します。



- b. [次へ]をクリックします。
4. 負荷の条件を設定します。

[負荷の条件の設定]ページで、異なる負荷シナリオを考慮するようSLAを設定します。

 - a. [負荷の条件]ドロップダウン・リストから、[実行中の仮想ユーザ]を選択します。
 - b. 次の例のように[負荷の値]を設定します。



上記の画面では、3つの潜在的な負荷シナリオに対して許容する平均トランザクション時間をSLAに設定しました。

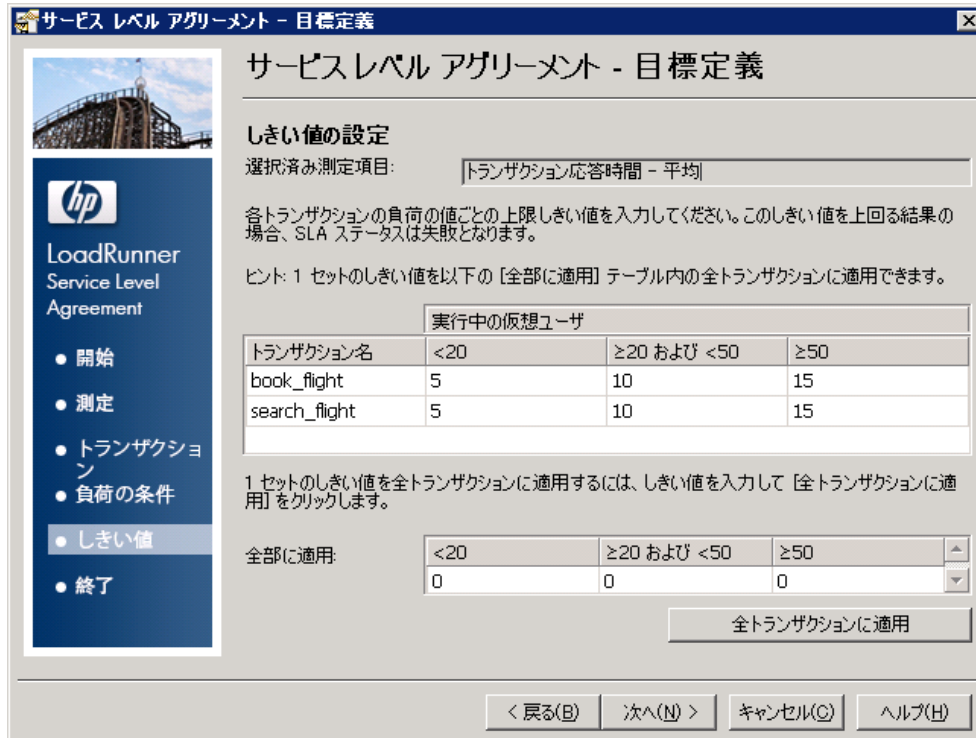
- 軽い負荷 :0 から 19 仮想ユーザの間
- 平均負荷 :20 から 49 仮想ユーザの間
- 重い負荷 :50 仮想ユーザ以上

c. [次へ]をクリックします。

5. しきい値を設定します。

[しきい値の設定]ページで、**check_itinerary**トランザクションに対する、許容できる平均トランザクション時間を定義します。

次の例のようにしきい値を設定します。



これで、次の平均トランザクション時間を許容すると設定しました。

- 軽い負荷: 5 秒以下
- 平均負荷: 10 秒以下
- 重い負荷: 15 秒以下

6. SLA を保存します。

SLA を保存してウィザードを閉じるには、続くページ上で [次へ]、[完了]、[閉じる] をクリックします。

Analysis はこの SLA 設定をサマリ・レポートに適用します。その後、レポートはすべての関連する SLA 情報を含んで更新されます。

パフォーマンス・サマリの表示

[サマリレポート] タブにはシナリオ実行に関する一般情報および統計に加え、関連するすべての SLA 情報が表示されます。たとえば、定義した SLA の観点から最もパフォーマンスの悪かったトランザクションは何か、設定した時間間隔でのトランザクションの具体的な処理内容、SLA ステータス等です。サマリ・レポートは、[セッション エクスプローラ] から開きます。

1. 全体的なシナリオの統計について

統計サマリ・セクションで、テストで最大 70 個の仮想ユーザが実行されたことがわかります。合計スループット、平均スループット、合計ヒット数、平均ヒット数などのほかの統計値も表示されています。

The screenshot shows the 'Analysis Summary' window in HP LoadRunner. The window title is 'サマリレポート' (Summary Report) and it has tabs for 'Running Users', 'Hits per Second', 'Throughput', and 'Transaction St'. The main content area is titled 'Analysis Summary' and contains the following information:

- Scenario Name:** C:\Tutorial_Scenario_gaby.lrs
- Results in:** S:\lrun6\nt\1414
- Session:** \runner_download\prd\tutorial\tutorial_result\tutorial_result.lrr
- Duration:** 11 分 および 57 秒.

Below this is the 'Statistics Summary' section, which includes a table of key performance indicators:

Maximum Running Users:	70
Total Throughput (bytes):	273,489,363
Average Throughput (bytes/second):	380,904
Total Hits:	8,789
Average Hits per Second:	12.241
Total Errors:	9,226

To the right of the 'Average Hits per Second' row, there is a link: [View HTTP Responses Summary](#).

Below the statistics is the '5 Worst Transactions' section, which contains the text: **トランザクションはすべて SLA 境界内でした。** (All transactions were within SLA boundaries).

At the bottom of the window, there is a section titled 'Scenario Behavior Over Time'.

2. 最もパフォーマンスの悪かったトランザクション

[5 Worst Transactions]の表にSLAが定義されている中で最もパフォーマンスの悪かったトランザクションが5個まで表示されます。

5 Worst Transactions

トランザクション名	失敗の割合 [%] (超過時間 / トランザクション持続時間)	失敗値 [%] (応答時間 / SLA)
- book_flight	50.806	37.324

時間区間	平均超過割合	最大超過割合
00:01:10-00:03:55	45.35	93.46
00:04:15-00:04:20	12.26	12.26
00:04:25-00:04:35	35.37	56.29
00:04:50-00:05:00	32.8	44.13
00:05:15-00:05:20	5.88	5.88
00:05:25-00:05:30	1.56	1.56
00:05:40-00:05:55	17.25	30.47
00:06:05-00:06:25	18.35	25.22
00:06:50-00:07:50	39.9	68.93
00:08:00-00:08:10	29.84	40.59
00:08:20-00:08:25	23.73	23.73
00:08:30-00:08:35	11.38	11.38

トランザクションの分析

check_itinerary トランザクションの期間中、SLA しきい値がその期間の 66.4% を超えたことがわかります。実行全体で SLA しきい値を超えた平均の割合は 200.684% でした。

3. SLA しきい値を超えた時間区間の確認

[Scenario Behavior Over Time] セクションには、異なる時間区間に各トランザクションがどのように実行されたかが表示されます。緑の四角はトランザクションが SLA しきい値内で実行された時間区間、赤の四角はトランザクションが失敗した時間区間、そして灰色は関連する SLA が定義されていない時間区間を表します。

Scenario Behavior Over Time

The SLA status of the following measurements displayed over time. You can select a specific time range for each transaction in order to analyze the time range.

トランザクション: book_flight ドリルダウンする時間範囲: 開始: 000:00:00 終了: 000:02:00

測定値の名前	時間範囲
テスト対象アプリケーションのエラー	0 0.1 24 31 18 25 32 43 38 17 0.7 0.5 0.2 0.4 0.2 0.5 0+ 0+
book_flight	
search_flight	
	00:00:00 00:00:40 00:01:20 00:02:00 00:02:40 00:03:20 00:04:00 00:04:40 00:05:20 00:06:00 00:06:40 00:07:20 00:08:00 00:08:40 00:09:20 00:10:00 00:10:40 00:11:20

トランザクションの分析

SLA を定義したトランザクションについて、ほとんどの区間で **check_itinerary** がしきい値を超えたことがわかります。

4. 全体的なトランザクションのパフォーマンスの確認

[Transaction Summary] には、各トランザクションの動作のサマリが表示されます。

Transaction Summary

Transactions: Total Passed: 1,100 Total Failed: 3,112 Total Stopped: 0

Average Response Time

Transaction Name	SLA ステータス	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
Action_Transaction		6.002	58.577	155.399	25.387	89.081	144	3,081	0
book_flight		0.219	1.555	3.949	0.817	2.666	175	0	0
check_itinerary		3.295	32.826	119.258	26.407	65.754	147	28	0
logout		0.406	1.005	12.909	1.146	1.67	144	3	0
login		0.444	3.934	9.864	2.161	6.777	175	0	0
search_flight		0.464	4.915	11.17	2.365	7.966	175	0	0
vuser_end_Transaction		0	0	0	0	0	70	0	0
vuser_init_Transaction		0	0.013	0.077	0.024	0.059	70	0	0

Service Level Agreement Legend: Pass Fail No Data

また、**check_itinerary** トランザクションが 28 回失敗したこともわかります。

各トランザクションの時間を確認します。**[90 Percent]** カラムは、ある特定のトランザクション実行の 90% の時間を示します。テスト実行中に実行された **check_itinerary** トランザクションのうち、90% の時間が 65.754 秒であったことがわかります。これは、平均時間 (32.826 秒) の倍です。つまり、大部分の **check_itinerary** トランザクションの応答時間は非常に長かったということです。

[SLA ステータス] カラムにどのように SLA のトランザクションに関連する全体のステータスが表示されているかに注意してください。**check_itinerary** が **Fail** になっています。

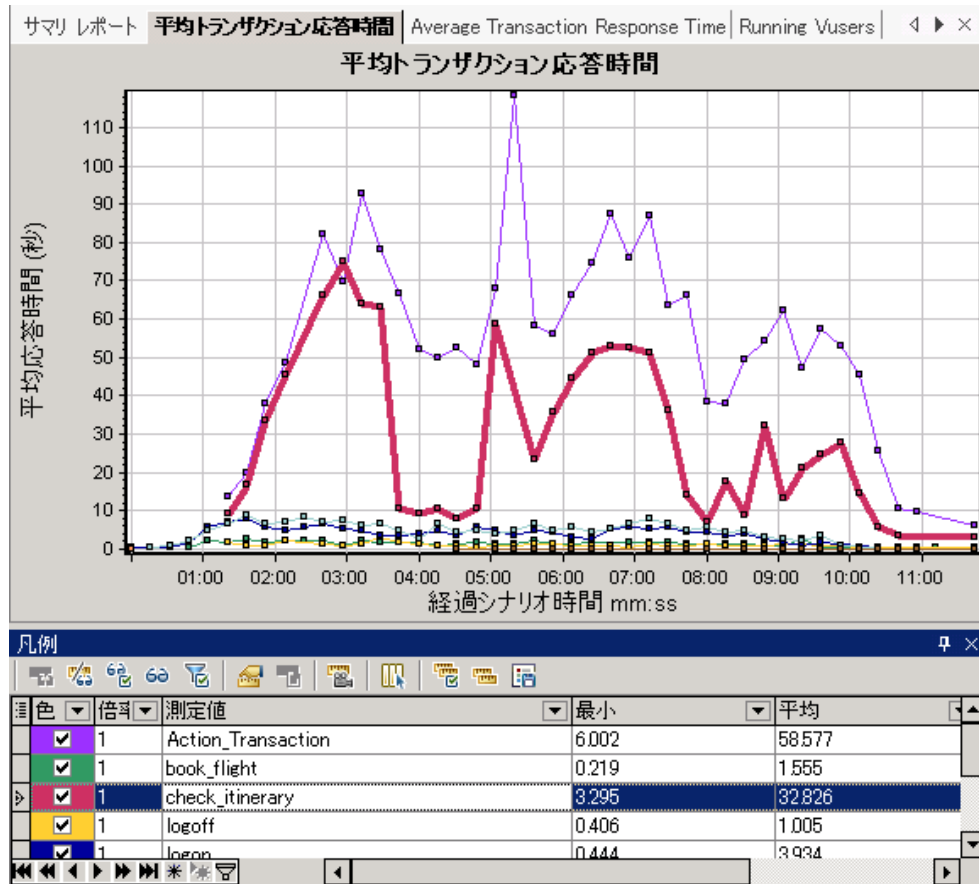
パフォーマンスのグラフによる表示方法

利用可能なグラフは [セッション エクスプローラ] ペインからアクセスできます。ここでは [平均トランザクション応答時間] グラフを表示、分析します。

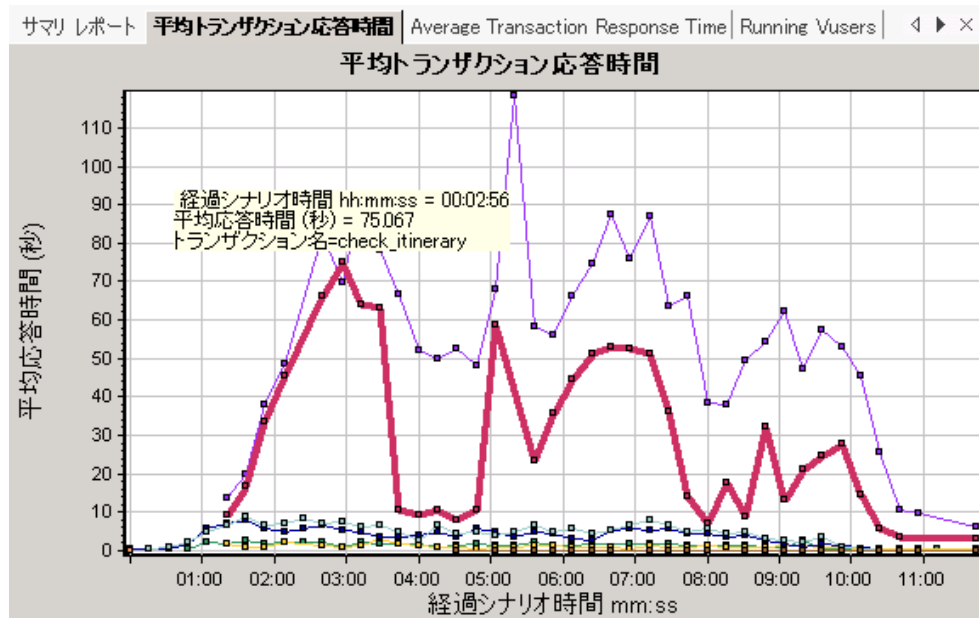
1. [平均トランザクション応答時間] グラフを開きます。
 - a. [セッション エクスプローラ] の [グラフ] の下の、[平均トランザクション応答時間] を選択します。グラフ表示領域で [平均トランザクション応答時間] グラフが開きます。

注: [セッション エクスプローラ] ペインにグラフが表示されない場合、[グラフ] ノードを右クリックし、[新規グラフを開く] ダイアログ・ボックスで [トランザクション: 平均トランザクション応答時間] ノードを選択します。[グラフを開く] をクリックして [セッション エクスプローラ] ペインにグラフを追加します。

- b. [凡例] ペインで、**check_itinerary** トランザクションをクリックします。グラフで **check_itinerary** トランザクションが強調表示されます。



グラフ上のポイントは、シナリオ実行時の特定の時間における平均トランザクション時間を示しています。グラフのポイント上にカーソルを置きます。黄色のボックス内にそのポイントの座標が表示されます。



2. 結果の分析:

check_itinerary トランザクションの平均トランザクション時間が大きく変動している点に注目してください。平均時間はシナリオ開始から2分56秒後にピークに達し、75.067秒となっています。

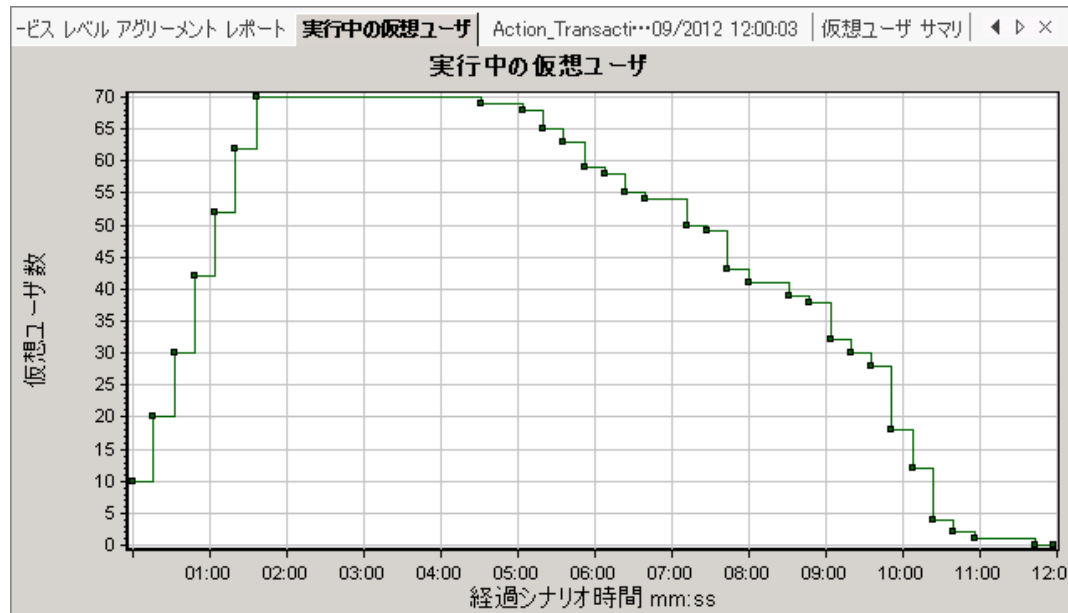
パフォーマンスの良いサーバでは、トランザクションの平均時間は比較的安定するでしょう。グラフの下部で、**logon** トランザクション、**logout** トランザクション、**book_flight** トランザクション、および **search_flight** トランザクションの平均時間がより安定している点に注目してください。

サーバのパフォーマンスの確認

前のセクションで、サーバのパフォーマンスが不安定であることを確認しました。そこで、70個の実行中の仮想ユーザがシステムのパフォーマンスに与える影響を分析します。

1. 仮想ユーザの動作を調査します。


[セッション エクスプローラ] の [グラフ] で、[実行中の仮想ユーザ] をクリックします。グラフ表示領域で [実行中の仮想ユーザ] グラフが開きます。



シナリオ実行の最初に、実行中の仮想ユーザが徐々に開始されたことがわかります。次に3分間70個の仮想ユーザが同時に実行され、その後徐々に仮想ユーザが実行を停止しています。

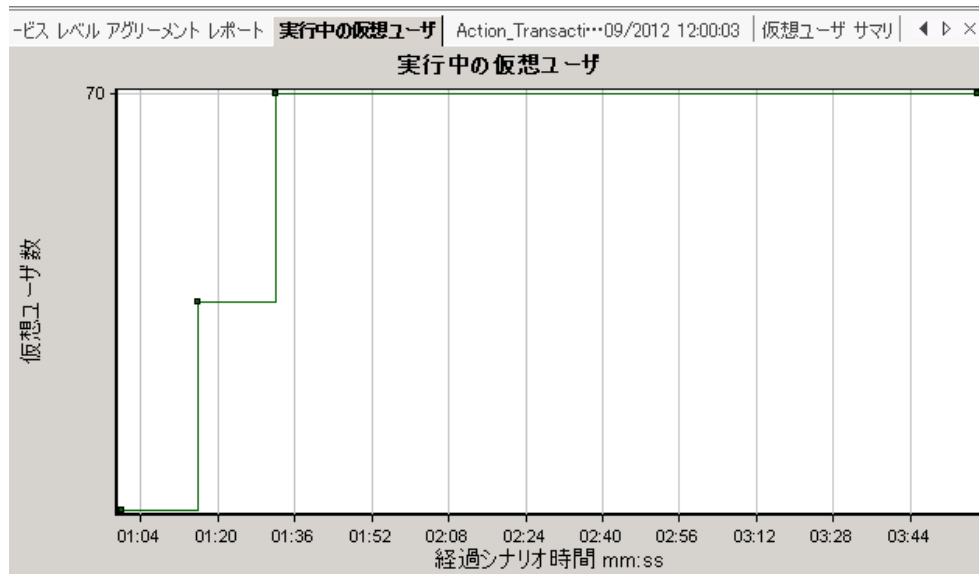
2. すべての仮想ユーザが同時に実行されたときのタイム・スライスだけが表示されるように、グラフにフィルタを適用します。


グラフにフィルタを適用すると、グラフ・データが絞り込まれ、指定された条件に合ったデータだけが表示されます。その場合、それら以外のデータはすべて非表示になります。

- a. グラフを右クリックして、[フィルタ/グループ化の設定] を選択するか、または Analysis ツールバーの [フィルタ/グループ化の設定] ボタン  をクリックします。
- b. [フィルタ条件] 領域で、[経過シナリオ時間] 行の [値] カラムを選択します。
- c. 下向き矢印をクリックして、時間範囲を **000:01:30** から **000:03:45** まで指定します。
- d. [OK] をクリックします。

- e. [グラフの設定]ダイアログ・ボックスで, [OK]をクリックします。

[実行中の仮想ユーザ]グラフには, シナリオ開始後 1分 30 秒から3分 45 秒の間に実行された仮想ユーザのみが表示されます。ほかの仮想ユーザは除外されます。



注: フィルタをクリアするには, グラフを右クリックして[フィルタとグループ化のクリア]を選択するか, または Analysis ツールバーの[フィルタ/グループ化のクリア]ボタン  をクリックします。

3. [実行中の仮想ユーザ]グラフと[平均トランザクション応答時間]グラフを相関させて, それらのデータを比較します。

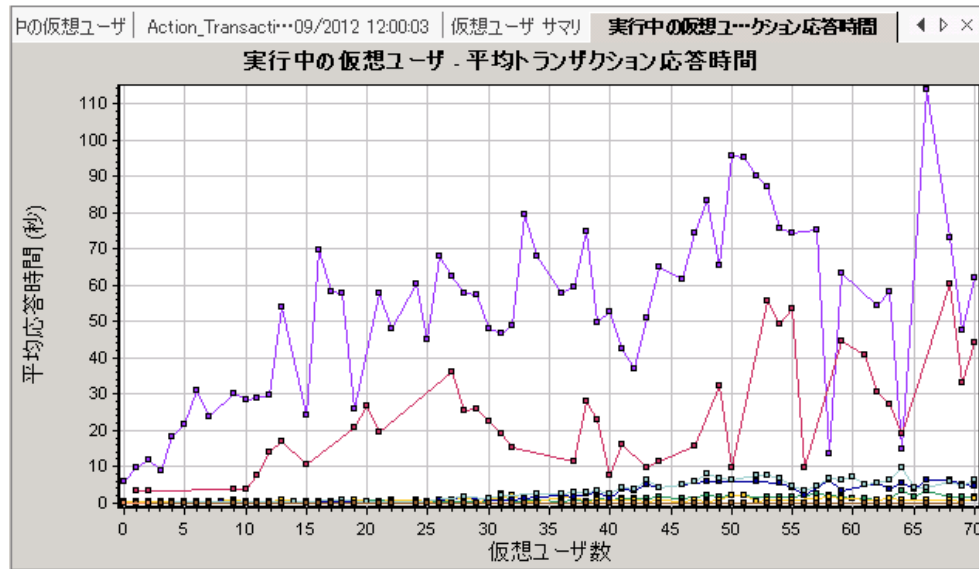
2つのグラフを結合して, 一方のグラフのデータがもう一方のグラフのデータに与える影響を確認できます。これを「2つのグラフを相関させる」といいます。

たとえば, [実行中の仮想ユーザ]グラフと[平均トランザクション応答時間]グラフを相関させて, 多数の仮想ユーザがトランザクションの平均時間に与える影響を確認できます。

- [実行中の仮想ユーザ]グラフを右クリックして, [フィルタとグループ化のクリア]を選択します。
- グラフを右クリックして, [グラフの結合]を選択します。
- [結合対象グラフの選択]リストから[平均トランザクション応答時間]を選択します。
- [結合タイプの選択]で[相関]を選択し, [OK]をクリックします。

[実行中の仮想ユーザ]グラフと[平均トランザクション応答時間]グラフが1つのグラフ, [実

行中の仮想ユーザ-平均トランザクション応答時間]グラフに表示されています。



4. 相関グラフを分析します。

このグラフで、仮想ユーザ数の増加に伴って、**check_itinerary**トランザクションの平均時間がわずかにずつ増えていることがわかります。つまり、負荷が大きくなるのに伴い、平均時間も増えています。

仮想ユーザ数が66個になった時点で、突然平均時間が急増します。これをテストによる「**サーバの限界超越**」といいます。同時実行中の仮想ユーザ数が66個を超えると、明らかに時間が遅くなり始めました。

テンプレートの保存

グラフにフィルタを適用し、2つのグラフを相関させました。次にシナリオを分析するとき、同じフィルタと結合条件を適用した同じグラフを表示する可能性があります。結合およびフィルタの設定をテンプレートに保存して、別のAnalysisセッションに適用できます。

テンプレートを保存するには、次の手順を実行します。

1. [ツール]>[テンプレート]を選択します。[テンプレートの適用/編集]ダイアログ・ボックスが開きます。
2. [テンプレート]ペインで、[新規作成]ボタンをクリックします。[新規テンプレートの追加]ダイアログ・ボックスが開きます。
3. テンプレートの適切な名前を入力し、[OK]をクリックします。
4. [OK]をクリックして、[テンプレートの適用/編集]ダイアログ・ボックスを閉じます。

次に新しいAnalysisセッションを開いたときに保存したテンプレートを使用するには、次の手順を実行します。

1. [ツール]>[テンプレート]を選択します。[テンプレートの適用/編集]ダイアログ・ボックスが開きます。
2. リストからテンプレートを選択し、[保存して閉じる]をクリックします。

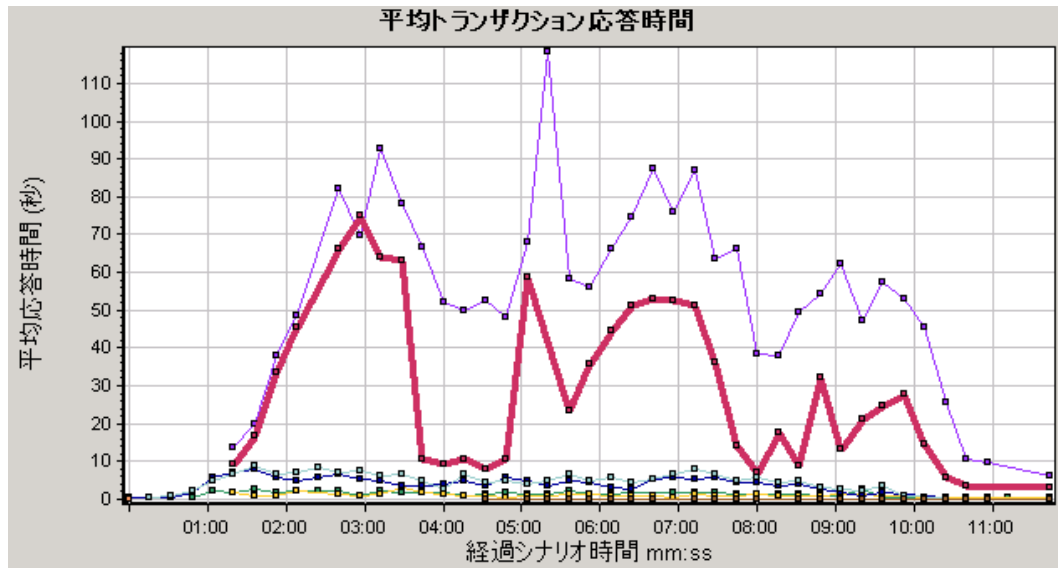
問題の原因の特定方法

サーバでの負荷の増加が、**check_itinerary** トランザクションの平均応答時間に悪影響を与えることを確認しました。

check_itinerary トランザクションをさらにドリルダウンして、パフォーマンスに悪影響を及ぼしているシステム・リソースを確認できます。

自動相関ツールにより、**check_itinerary** トランザクションの応答時間に影響を与えた可能性のあるデータを含むすべてのグラフを結合し、問題発生時に何が起こっていたかを特定できます。

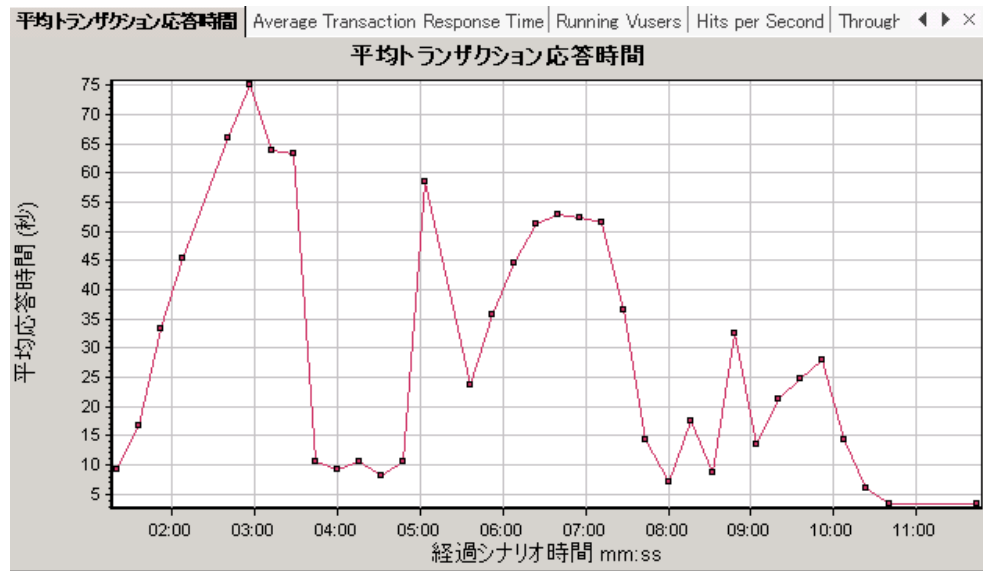
1. グラフ・ツリーから、[平均トランザクション応答時間] グラフを選択します。



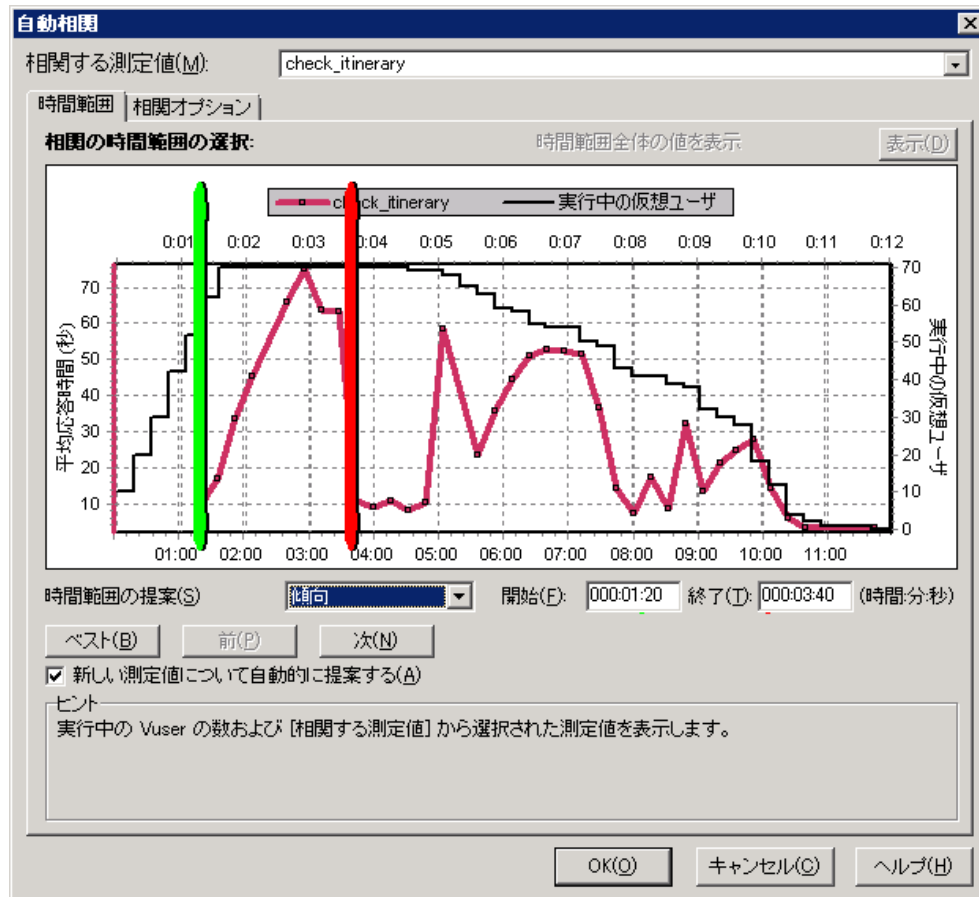
check_itinerary トランザクションの、特に経過時間が1分から4分の間を見ます。すぐに平均応答時間が増加し始め、ほぼ3分でピークに達するまで増加し続けました。

2. [平均トランザクション応答時間] グラフにフィルタを適用して、**check_itinerary** トランザクションのみを表示します。
 - a. グラフを右クリックして、[フィルタ/グループ化の設定] を選択します。
 - b. [トランザクション名] または [値] セルで **check_itinerary** を選択します。
 - c. [OK] をクリックします。

フィルタされたグラフでは、**check_itinerary** トランザクションのみが表示されます。

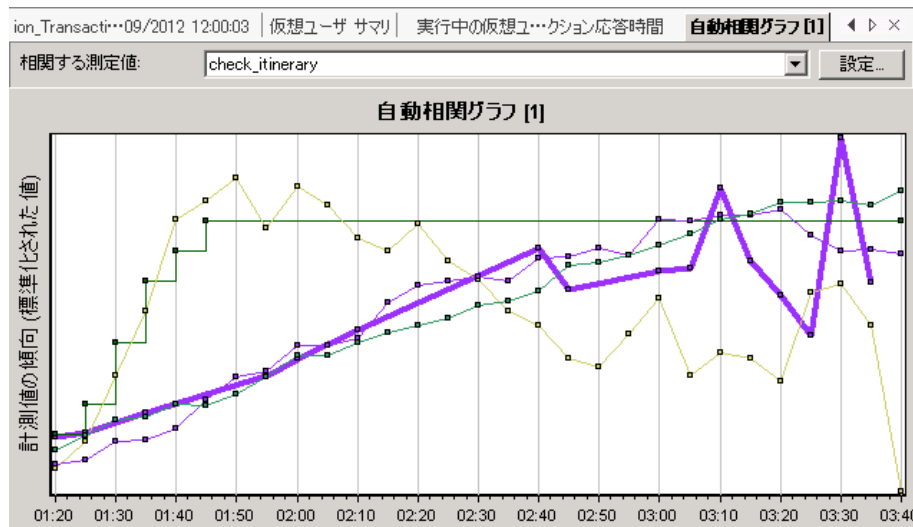


3. グラフを自動相関します。
 - a. グラフを右クリックして、[自動相関]を選択します。
 - b. [自動相関]ダイアログ・ボックスで、相関する測定値が**check_itinerary**であることを確認し、時間範囲を 1:20 から 3:40 に設定します。時間範囲は、ボックスに時間を入力するか、経過シナリオ時間軸に沿って緑と赤の棒をドラッグして設定します。



c. [OK]をクリックします。

グラフ表示領域に自動相関グラフが表示されます。**check_itinerary** トランザクションが強調表示されます。



自動相関グラフには、標準設定の名前「自動相関グラフ [1]」が付けられます。

4. グラフの名前を変更します。
 - a. セッション・エクスプローラの[グラフ]で、[自動相関グラフ [1]]を右クリックし、[項目の名前変更]を選択します。グラフ名が編集可能になります。
 - b. 「Auto Correlated - check_itinerary」と入力して、Enter キーを押すか Analysis のウィンドウのどこかをクリックします。
5. 自動相関グラフを分析します。

グラフの下にある[凡例]ペインの[グラフ]カラムから、[Windows リソース: Pool Nonpaged Bytes と Private Bytes]測定値に下方スクロールします。

色	グラフ	倍率	測定値	相関一致	相関
✓	Windows リソース	標準化	% Disk Time (PhysicalDisk_Total):alabama	8	直接関連して
✓	Windows リソース	標準化	% Processor Time (Process_xiwin32):alabama	44	逆相関
✓	Windows リソース	標準化	% Processor Time (Processor_Total):alabama	18	直接関連して
✓	Windows リソース	標準化	File Data Operations/sec (System):alabama	27	逆相関
✓	Windows リソース	標準化	Interrupts/sec (Processor_Total):alabama	42	直接関連して
✓	Windows リソース	標準化	Page Faults/sec (Memory):alabama	35	逆相関
✓	Windows リソース	標準化	Page Faults/sec (Process_Total):alabama	48	逆相関
✓	Windows リソース	標準化	Page Faults/sec (Process_xiwin32):alabama	29	逆相関
✓	Windows リソース	標準化	Pool Nonpaged Bytes (Memory):alabama	75	直接関連して
✓	Windows リソース	標準化	Private Bytes (Process_Total):alabama	76	直接関連して
✓	Windows リソース	標準化	Processor Queue Length (System):alabama	29	直接関連して
✓	Windows リソース	標準化	Threads (Objects):alabama	22	直接関連して

[測定値]カラムおよび[相関一致]カラムで、メモリに関連するこれらの測定値のcheck_itinerary トランザクションとの[相関一致]が70%以上であることがわかります。これは、指定した時間間隔で、これらの要素の動作とcheck_itinerary トランザクションの動作は密接に関連していたことを意味します。

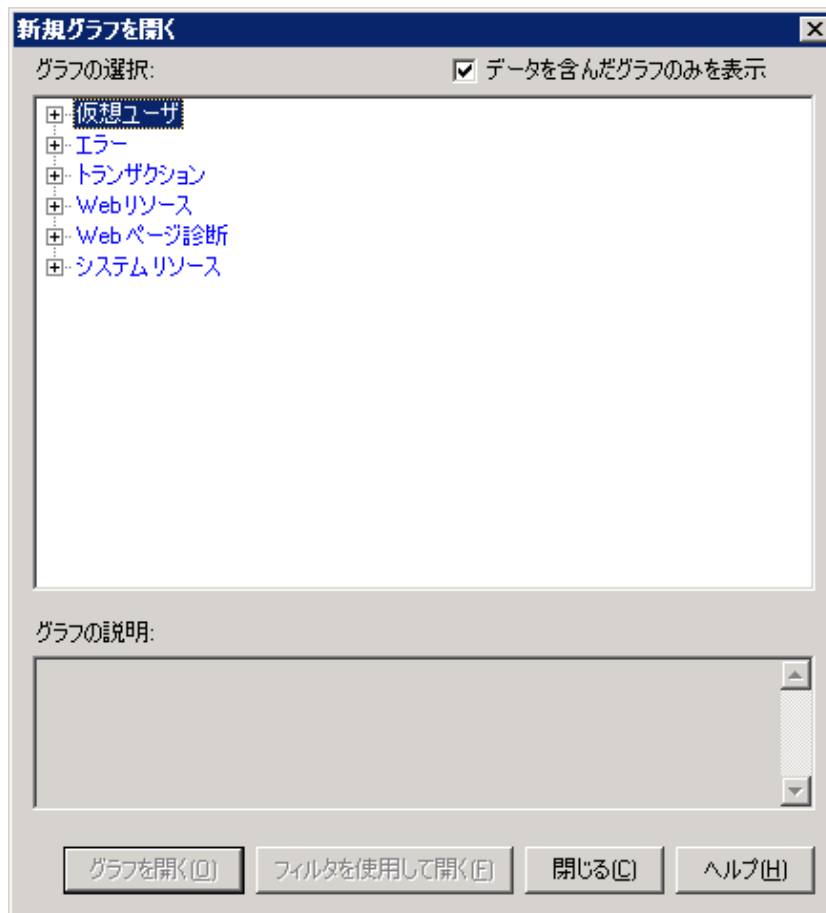
check_itinerary トランザクションの応答時間がピークに達した瞬間に、システムのメモリ・リソースが不足したと判断できます。

シナリオ実行で収集可能なその他の情報

Analysis セッションの開始時点でグラフ・ツリーに表示されるグラフに加えて、シナリオ実行に関するその他の情報を取得するためにさまざまなグラフを表示できます。

1. [グラフ]>[新規グラフの追加]をクリックします。

[新規グラフを開く]ダイアログ・ボックスが開き、データがあり表示可能なグラフのカテゴリが一覧表示されます。



- **仮想ユーザ**: 仮想ユーザと仮想ユーザのステータスに関する情報が表示されます。
 - **エラー**: エラーの統計が表示されます。
 - **トランザクション**: トランザクションとトランザクションの応答時間に関するデータが表示されます。
 - **Web リソース**: ヒット数、スループット、および接続データが表示されます。
 - **Web ページ診断**: スクリプトに含まれる各監視対象 Web ページに関するデータが表示されます。
 - **システム リソース**: システム・リソースの使用状況のデータが表示されます。
2. 新しいグラフを表示します。
- a. [新規グラフを開く]ダイアログ・ボックスで、カテゴリの横にある[+]をクリックしてカテゴリを展開します。
 - b. グラフを選択し、[グラフを開く]をクリックします。
 - c. [閉じる]をクリックして、[新規グラフを開く]ダイアログ・ボックスを閉じます。
- ほかにもいくつかのグラフを開いて、シナリオ実行についてさらに分析してください。

結果の発行方法

HTML レポートまたは Microsoft Word レポートを使用して、Analysis セッションの結果を発行できます。レポートは設計者のテンプレートを使用して作成され、提示されたグラフやデータの説明および凡例が含まれます。

HTML レポート

HTML レポートは、あらゆるブラウザで開いて表示できます。

HTML レポートを作成するには、次の手順を実行します。

1. [レポート]>[HTML レポート]をクリックします。
2. レポートのファイル名と格納場所のパスを指定します。
3. [保存]をクリックします。

Analysis はレポートを作成して、そのレポートを Web ブラウザで表示します。HTML レポートのレイアウトと Analysis セッションのレイアウトが非常に似ている点に注目してください。左のペインのリンクをクリックして、さまざまなグラフを表示できます。ページの下部に各グラフの説明が表示されます。

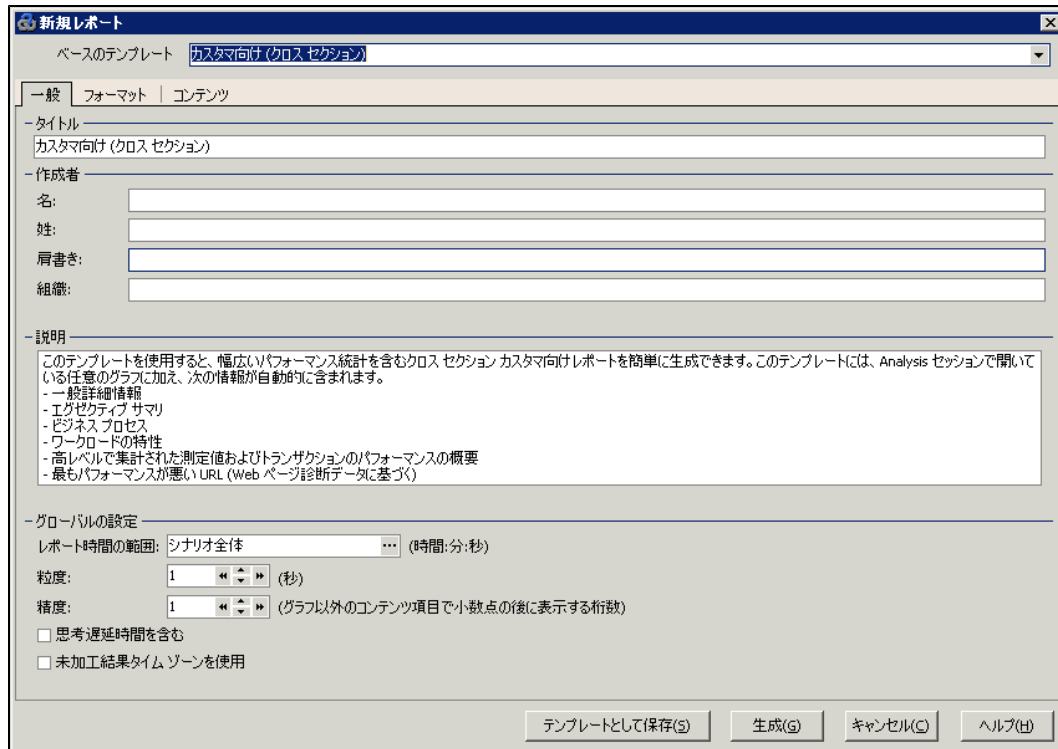
Microsoft Word レポート

Analysis セッションを Microsoft Word レポートを使用して表示できます。Word レポートにはシナリオ、測定値の説明などに関する一般情報を含むオプションがあるので、HTML レポートより包括的です。また、レポートを書式設定して、会社名、会社のロゴ、作成者情報を含めることができます。

ほかの Microsoft Word ファイルと同様、レポートは編集可能であるため、レポートを作成した後にコメントや結果を追加できます。

Microsoft Word レポートを作成するには、次の手順を実行します。

1. [レポート]>[新規レポート]をクリックします。
[新規レポート]ダイアログ・ボックスが開きます。



2. [一般]タブ:
 - a. [ベースのテンプレート]で, [詳細レポート (シングルラン)]を選択します。
 - b. レポートのタイトルを入力します。
 - c. 作成者の名前, 肩書き, 会社名を入力します。

3. [フォーマット]タブ:


標準設定では, レポートにはタイトル・ページ, 目次, グラフ詳細, グラフの説明, および測定値の説明が作成されます。スクリプトの詳細をレポートに追加するオプションを選択することもできます。その場合, ビジネス・プロセス・ステップのサムネイル・イメージが表示されます。

[会社のロゴを含める]を選択してファイルの場所を参照し, 会社のロゴを含めることができます。ロゴ・ファイルは .bmp 形式のファイルです。

4. [コンテンツ]タブ:


- a. レポートに含めるシナリオ実行および Analysis セッションのセクションを選択します。

このチュートリアルでは, [コンテンツ項目]リストにエグゼクティブ・サマリを追加します。

- b. [追加]ボタン  をクリックして, [コンテンツ項目の追加]ウィンドウを開きます。グリッドで[エグゼクティブ サマリ]を選択し, [OK]をクリックします。エグゼクティブ・サマリ項目が[コンテンツ項目]ペイン内のリストに追加されます。

次のテキストを編集ボックスに入力します。

- 目標: このテスト・シナリオ目標は...
- 結論: 到達した結論は以下のとおりです。

- c. [コンテンツ項目] ペインで, [平均キロバイトによる最大 URL] を選択し, [削除] ボタン  をクリックします。これにより, このグラフがレポートから除外されます。
- d. レポートの項目の表示順序を変更します。
 - [コンテンツ項目] ペインで, [ワークロードの特性] を選択します。[選択されたカラム] リストで [秒ごとの平均ヒット数] を選択します。
 - 項目が [合計トランザクション数] の下に表示されるまで下向き矢印をクリックします。レポートでは, [合計トランザクション数] 項目の下に [秒ごとの平均ヒット数] 項目が表示されます。
5. [生成] をクリックします。

データが集められ, レポートが Microsoft Word ファイルで作成されます。ファイルは Microsoft Word で開きます。

Analysis セッション中に生成されたグラフに加えて, レポートには目的と結論, およびレポートの作成時に含めるようにしたその他のセクションとグラフが含まれます。

結論

この練習では, サービス・レベル・アグリーメントの定義, シナリオ実行の分析, およびレポートを使用した結果の発行についての基本を学びました。

また, 過度の負荷が原因と思われるサーバのボトルネックを示すさまざまなグラフを調査してパフォーマンスの問題を特定する方法についても学びました。グラフを設定して関連データを表示することによりこれらのボトルネックの原因を特定する方法についても学びました。