

HP LoadRunner

Windows オペレーティング・システム用

ソフトウェア・バージョン : 9.00

チュートリアル

文書番号 : LRTUT9.00JP/01

文書発行日 : 2007 年 6 月 (英語版)

ソフトウェア・リリース日 : 2007 年 6 月 (英語版)



利用条件

保証

HP の製品およびサービスの保証は、かかる製品およびサービスに付属する明示的な保証の声明において定められている保証に限ります。本文書の内容は、追加の保証を構成するものではありません。HP は、本文書に技術的な間違いまたは編集上の間違い、あるいは欠落があった場合でも責任を負わないものとします。

本文書に含まれる情報は、事前の予告なく変更されることがあります。

制限事項

本コンピュータ・ソフトウェアは、機密性があります。これらを所有、使用、または複製するには、HP からの有効なライセンスが必要です。FAR 12.211 および 12.212 に従って、商用コンピュータソフトウェア、コンピュータソフトウェアのドキュメント、および商用アイテムの技術データは、HP の標準商用ライセンス条件に基づいて米国政府にライセンスされています。

サードパーティ Web サイト

HP は、補足情報の検索に役立つ外部サードパーティ Web サイトへのリンクを提供します。サイトの内容と利用の可否は予告なしに変更される場合があります。HP は、サイトの内容または利用の可否について、いかなる表明も保証も行いません。

著作権

© 1994 - 2007 Mercury Interactive Corporation, All rights reserved

商標

Java™ は、Sun Microsystems, Inc. の米国商標です。

Microsoft® および Windows® は、Microsoft Corporation の米国登録商標です。

Oracle® は、カリフォルニア州レッドウッド市の Oracle Corporation の米国登録商標です。

Unix® は、The Open Group の登録商標です。

文書の更新

本書のタイトル・ページには、次の識別情報が含まれています。

- ソフトウェアのバージョンを示すソフトウェア・バージョン番号
 - 文書が更新されるたびに更新される文書発行日
 - 本バージョンのソフトウェアをリリースした日付を示す、ソフトウェア・リリース日付
- 最新のアップデートまたは文書の最新版を使用していることを確認するには、
<http://h50146.www5.hp.com/doc/manual/openview/> を参照します。

サポート

Mercury 製品のサポート

従来 Mercury が提供していた製品のサポート情報は、次の方法で入手できます。

- HP Software Services Integrator (SVI) パートナー (www.hp.com/managementsoftware/svi_partner_list) を通している場合は、SVI 代理店にお問い合わせください。
- 有効な HP Software サポート契約をお持ちの場合は、HP Software Support サイトを参照して、セルフソルブ技術情報検索を使用して技術的な質問に対する回答を検索できます。
- 従来 Mercury が提供していた製品のサポート・プロセスおよびツールの最新情報については、HP-Mercury Support Web サイト (<http://support.mercury.com>) (英語サイト) を参照してください。
- その他の質問については、HP の営業担当にお問い合わせください。

HP ソフトウェアのサポート

HP Software Support Web サイトへは、www.hp.com/managementsoftware/services からアクセスできます。

HP Software のオンライン・サポートでは、対話型の技術支援ツールに効率的にアクセスできます。サポートをご利用のお客様は、サポート・サイトを使うことで次のような利点があります。

- 参照したいナレッジ文書の検索
- エンハンスメント要求およびサポート・ケースの登録とトラッキング
- 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 サポートの連絡先の検索
- 利用可能なサービスに関する情報の参照
- ソフトウェアの他のお客様とのディスカッションに参加
- ソフトウェアのトレーニングの調査および登録

ほとんどのサポート・エリアは、HP Passport ユーザとしての登録およびサインインが必要です。また多くは、サポート契約も必要です。アクセス・レベルの詳細情報については、www.hp.com/managementsoftware/access_level を参照してください。

HP Passport ID の登録を申請するには、
www.managementsoftware.hp.com/passport-registration.html（英語サイト）にアクセスしてくだ
さい。

目次

LoadRunner チュートリアルへようこそ	11
練習 1: LoadRunner の紹介	13
パフォーマンス・テストを自動化する理由	14
LoadRunner コンポーネントについて	14
LoadRunner の用語について	15
負荷テスト・プロセスについて	15
HP Tours について	16
アプリケーションの要件	17
練習 2: LoadRunner の威力	19
負荷テストの作成	20
負荷テストの実行	24
負荷テストの監視	25
結果の分析	28
練習 3: スクリプトの作成	29
Virtual User Generator (VuGen) について	29
ユーザ動作の記録の開始方法	30
VuGen ウィザード・モードの使用方法	32
ビジネス・プロセスの記録によるスクリプトの作成方法	33
スクリプトの表示方法	37
練習 4: スクリプトの再生	41
実行時の振る舞いの設定方法	41
スクリプト実行のリアルタイムでの監視方法	45
再生に関する情報の表示方法	47
テストの成功の確認方法	49
結果の検索またはフィルタ方法	50
練習 5: 再生に関する一般的な問題の解決方法	55
再生エラーのための HP Tours の準備	56
一意のサーバ値の使用方法	57

練習 6： 負荷テスト用スクリプトの準備	63
ビジネス・プロセスの測定方法	64
複数のユーザをエミュレートする方法.....	67
Web ページの内容の確認方法.....	71
デバッグ情報の生成方法	74
テスト成功の確認方法.....	76
練習 7： 負荷テスト・シナリオの作成	79
LoadRunner Controller の紹介.....	80
負荷テストに含めるユーザの組み合わせ	82
Controller ウィンドウの概要	84
大きな負荷を生成する方法.....	85
実際の負荷動作のエミュレート方法	86
さまざまなタイプのユーザをエミュレート方法	91
負荷下のシステムの監視方法.....	94
練習 8： 負荷テストの実行	99
Controller の [実行] ビューの概要.....	100
負荷テスト・シナリオの実行方法.....	101
負荷下のアプリケーションの監視方法.....	103
リアルタイムで実行しているユーザの監視方法	105
ユーザ・アクションのサマリの表示場所	106
テスト中に負荷を増やす方法.....	107
負荷下のアプリケーションの対処方法.....	108
アプリケーションでエラーが発生したかどうか	109
テストの実行が完了したことを知る方法	111
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.....	112
練習 9： 高度なゴール指向シナリオ	113
使用するゴール・タイプ	114
ゴール指向シナリオの作成方法	114
Controller ウィンドウの概要（ゴール指向シナリオ）.....	116
ゴールの定義方法	117
負荷の動作の確認方法.....	118
監視するオンライン・グラフ	120
ゴール指向シナリオの実行方法	121
ゴールの達成の確認	124

練習 10 : シナリオの分析	125
Analysis セッションの処理方法	126
Analysis セッションの開始方法	127
Analysis のウィンドウの概要	128
サマリ・レポート	129
サーバのパフォーマンスの確認	133
問題の原因の特定方法	137
シナリオ実行で収集可能なその他の情報	141
結果の発行方法	143
結論	146

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。負荷テストの全工程について説明しており、LoadRunner テスト環境の理解を深めることができます。

最初の練習では、LoadRunner とテスト概念について説明します。

練習 2 では、サンプル・テストを使用して、LoadRunner ツールの機能について説明します。

練習 3 から 6 では、HP Virtual User Generator を使用してスクリプトを作成する方法について説明します。

練習 7 から 9 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 10 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、ご自分のアプリケーションを対象に簡単なテストを設計、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。

チュートリアルへようこそ

練習 1

LoadRunner の紹介

この 20 年間、企業は、作業を自動化する手段としてソフトウェアに頼ってきました。ソフトウェア・アプリケーションは、効率や生産性を高め、グローバル経済におけるコラボレーションと情報共有のための新しい手段を提供するのに使用されています。事実、ソフトウェア・アプリケーションは、ビジネスに不可欠な情報共有と、あらゆる種類のトランザクション処理の両方において、主要な手段となりました。いまや、電子メールから CRM やトランザクション処理に至るまで、ソフトウェア・アプリケーションはビジネス**そのもの**です。

これまでにソフトウェア開発技術が変化し、大いに成熟する一方で、現代のアプリケーションの複雑さは大幅に増しました。アプリケーションは、かつて紙や手作業で行っていた作業をするのに、何十何百ものコンポーネントを使用する場合があります。アプリケーションの複雑さの度合いと、ビジネス・プロセスにおいて起こりうる失敗の数との間には、直接的な相関関係があります。これにより、問題の根本原因を特定するのがますます困難になっています。

その上、ソフトウェア・アプリケーションは車などとは異なります。ソフトウェア・アプリケーションには、摩耗した場合にのみ置き換えればよい耐久性のある部品がありません。競争力を高めるため、あるいはビジネスの状況変化に対応するため、ソフトウェア・アプリケーションは週、月、年単位で変わります。このような変化の連続により、企業が対処しなければならない別のリスクが生まれてきます。

驚くべき変化のスピードとソフトウェアの複雑化は、ソフトウェア開発プロセスに極めて大きなリスクをもたらします。厳しいパフォーマンス・テストを行うことは、ビジネスに対するこのリスクを定量化および軽減するための最も一般的なストラテジです。HP LoadRunner による自動負荷テストは、アプリケーション開発プロセスの重要な要素です。

パフォーマンス・テストを自動化する理由

自動パフォーマンス・テストは、アプリケーション、アップグレード、パッチ開発のリスクを軽減するために製品、人々、プロセスを活用する管理手法です。根本的には、自動パフォーマンス・テストは、並行してシステムのパフォーマンスおよびエンドユーザの経験を測定しながら、開発前のシステムに対して運用の作業負荷を適用します。きちんと作成されたパフォーマンス・テストでは、次のような疑問に対する答えが得られます。

- ▶ アプリケーションの応答は、意図したユーザにとって十分な速さか。
- ▶ アプリケーションは、予想されるユーザ負荷およびそれ以上の負荷を処理するか。
- ▶ アプリケーションは、ビジネスに必要な数のトランザクションを処理するか。
- ▶ アプリケーションは、予想される、また予期しないユーザ負荷下で安定しているか。
- ▶ ユーザは、稼働日に確実に有意義な体験を得られるか。

自動パフォーマンス・テストは、これらの疑問に答えることで、ビジネス条件の変更の影響を定量化します。その結果、開発のリスクが明確になります。効果的な自動パフォーマンス・テスト・プロセスは、十分に情報を得た上でリリースの決定を下すのに役立ち、システムのダウンタイムおよび可用性の問題の発生を防ぎます。

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。

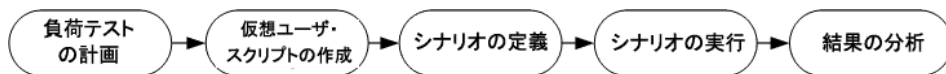
- ▶ **Virtual User Generator** は、エンドユーザのビジネス・プロセスをキャプチャし、仮想ユーザ・スクリプトとも呼ばれる自動パフォーマンス・テスト・スクリプトを作成します。
- ▶ **Controller** は、負荷テストの編成、実行、管理、監視を行います。
- ▶ **Load Generator** は、仮想ユーザを実行して負荷を生成します。
- ▶ **Analysis** は、パフォーマンス結果の表示、分析、比較を行うのに役立ちます。
- ▶ **Launcher** は、すべての LoadRunner コンポーネントへのアクセス・ポイントを提供します。

LoadRunner の用語について

- シナリオ** **シナリオ**とは、パフォーマンス要件に基づいて、各テスト・セッション中に発生するイベントを定義するファイルです。
- 仮想ユーザ** 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。仮想ユーザは実際のユーザのアクションをエミュレートし、アプリケーションを操作します。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** 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。
- トランザクション** サーバ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、測定対象のエンドユーザのビジネス・プロセスを表しています。

負荷テスト・プロセスについて

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。



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。

仮想ユーザ・スクリプトの作成：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。

シナリオの定義：LoadRunner Controller を使用して、負荷テスト環境をセットアップします。

シナリオの実行：LoadRunner Controller で負荷テストの実行、管理、監視を行います。

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、パフォーマンスを評価します。

HP Tours について

HP のソリューションを説明するために、本チュートリアルでは、サンプル・アプリケーション用のサンプル・パフォーマンス要件を使用します。アプリケーション HP Tours は、Web ベースの旅行代理店システムです。HP 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 40 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。

本項では、HP Tours アプリケーションについて確認します。

HP Tours の起動

HP Tours アプリケーションを使用して、実践的なパフォーマンス・テストを体験します。チュートリアルを進める前に、次の手順に従って、このアプリケーションのルック・アンド・フィールに慣れておきます。

1 サンプル Web サーバが動作していることを確認します。

Web サーバは、LoadRunner のインストールおよび再起動後、自動的に開始されます。再度システムを再起動してもサーバが動作しない場合は、**[スタート] > [プログラム] > [LoadRunner] > [Samples] > [Web] > [Start Web Server]** を選択します。

注： Web サーバがすでに動作しているときに Web サーバを起動しようとする
と、エラー・メッセージが表示されます。この場合、エラー・メッセージは無視して、チュートリアルを進めます。

2 HP Tours アプリケーションを開きます。

[スタート] > [プログラム] > [LoadRunner] > [Samples] > [Web] > [HP Web Tours Application] を選択します。ブラウザが開き、HP Tours の最初のページが表示されます。

3 HP Tours にログインします。

次の情報を入力します。

[ユーザ名] : jojo

[パスワード] : bean

左側の表示枠で [login] をクリックします。HP Tours の「Welcome」ページが表示されます。

4 航空券を予約します。

左側の表示枠で [Flights] をクリックします。「Find Flight」ページが開きます。目的地を Los Angeles に変更します。[Continue...] をクリックします。

5 HP Tours セッションを終了します。

左側の表示枠で [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要件

HP Tours に慣れたところで、アプリケーションがビジネス・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 Tours では、10 社の旅行代理店を同時に正常に処理する必要があります。
- 2 HP 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 Tours では、120 秒を超えない応答時間で、行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 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。

練習 1 • LoadRunner の紹介

練習 2

LoadRunner の威力

LoadRunner の威力を実証するため、最大 10 個の同時ユーザによるデータベース・アプリケーションへの負荷テストを実行、分析します。このテストでは、フライト予約システム（ログオン、フライトの検索、航空券の購入、行程の確認、ログオフなど）を同時に使用する旅行代理店をエミュレートします。

テストの間、LoadRunner のオンライン・モニタを使用して、負荷下の Web サーバの振る舞いを観察できます。特に、ユーザ・アクションに対するサーバの応答時間（トランザクション応答時間）に負荷の増大がどのように影響し、エラーを引き起こすか、確認します。

どのようにすれば、LoadRunner を使用してシステムに対して負荷を生成し、その負荷に対するシステムの反応性を測定できるかがわかったら、LoadRunner のコンポーネント（VuGen, Controller, Analysis）を使用して独自のテストを作成、実行し、テスト結果を分析する方法について学びます。

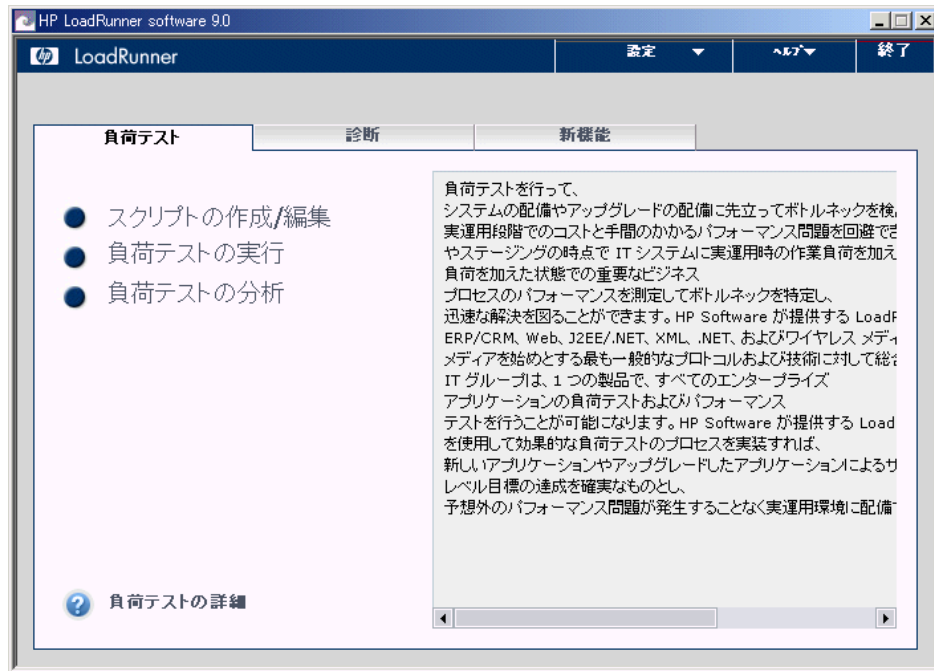
注： LoadRunner では、アプリケーションを効果的にテストするのに必要な数の仮想ユーザ（Vuser）を使用するライセンスが購入できます。この LoadRunner 体験版では、10 個の仮想ユーザを使用するライセンスが与えられています。

負荷テストの作成

Controller は、テストの構築、管理、監視を行う中央コンソールです。Controller を使用して、実際のユーザが実行するアクションをエミュレートするサンプル・スクリプトを実行し、同時に多くの仮想ユーザにアクションを実行させてシステムに負荷を生成します。

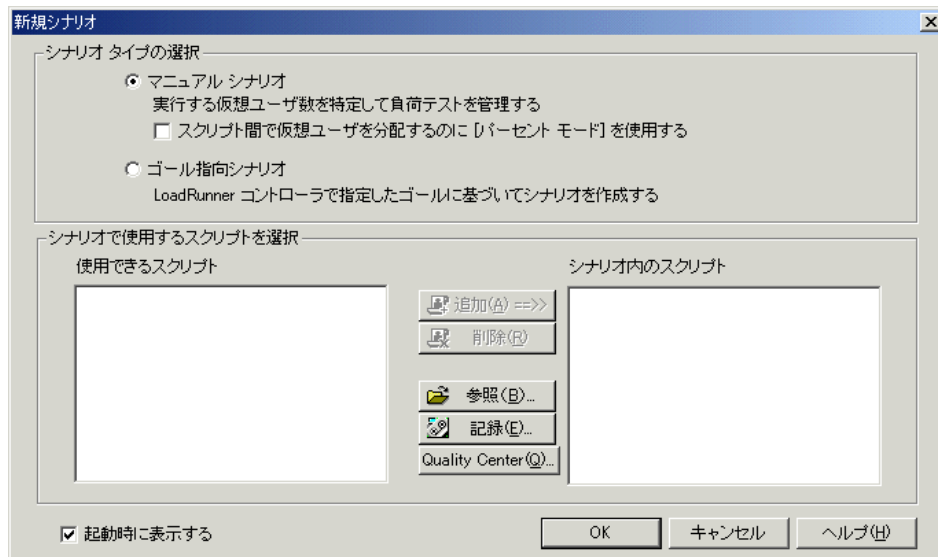
1 [HP LoadRunner] ウィンドウを開きます。

[スタート] > [プログラム] > [LoadRunner] > [LoadRunner] を選択します。HP LoadRunner Launcher ウィンドウが開きます。



2 Controller を起動します。

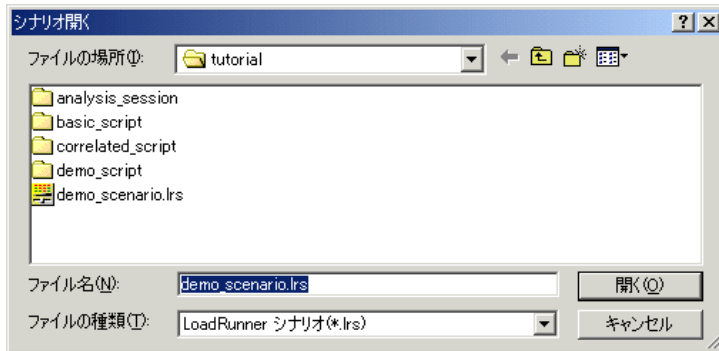
[**負荷テスト**] タブで、[**負荷テストの実行**] をクリックします。標準設定では、[LoadRunner コントローラ] ウィンドウが表示され、[新規シナリオ] ダイアログ・ボックスが表示されます。



[**キャンセル**] をクリックします。

3 サンプル・テストを開きます。

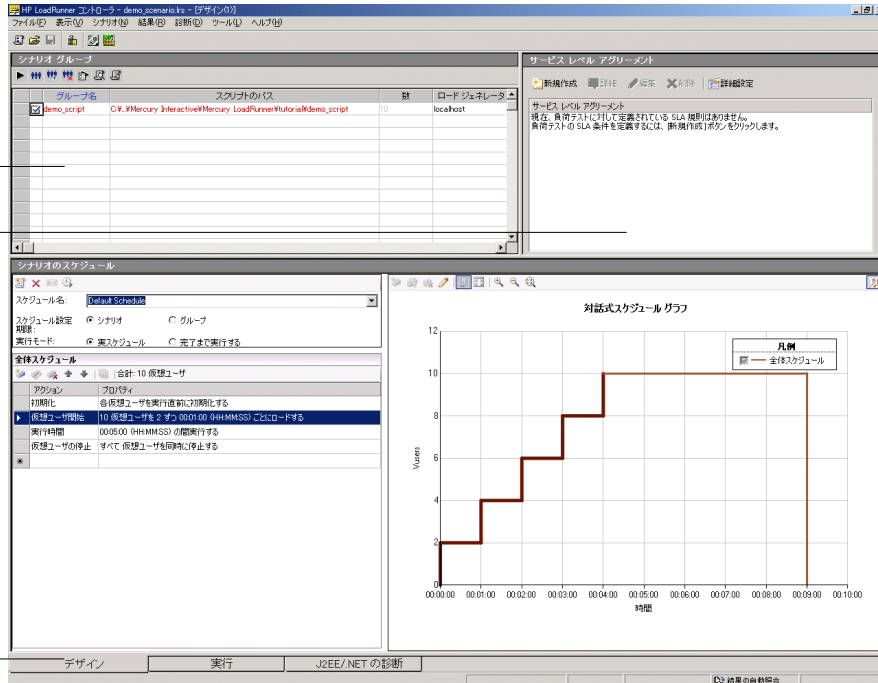
Controller のメニューから [ファイル] > [開く] を選択し、
 < LoadRunner インストール先フォルダ > ¥Tutorial ディレクトリの
demo_scenario.lrs を開きます。



LoadRunner Controller の [デザイン] タブが開き, [シナリオ グループ] 表示枠
 に **demo_script** テストが表示されます。テストの実行用に 10 個の仮想ユーザ
 が割り当てられていることがわかります。

[シナリオ
 グループ]
 セクション

 SLA



[デザイン]
 タブ

注：標準の LoadRunner インストール・ディレクトリにチュートリアルをインストールしなかった場合、このスクリプト・パスは正しくありません（スクリプト・パスが赤で表示されます）。正しいパスを入力するには、スクリプトを選択し、下向き矢印をクリックします。[参照] ボタンをクリックし、< LoadRunner インストール先フォルダ > ¥Tutorial ディレクトリの **demo_script** を指定して、[OK] をクリックします。

注：標準の LoadRunner インストール・ディレクトリにチュートリアルをインストールをした場合でも、スクリプト・パスが赤く表示されることがあります。この場合も [参照] ボタンをクリックし、正しい demo_script を指定して [OK] をクリックしてください。

これで、テスト実行の準備が整います。

負荷テストの実行



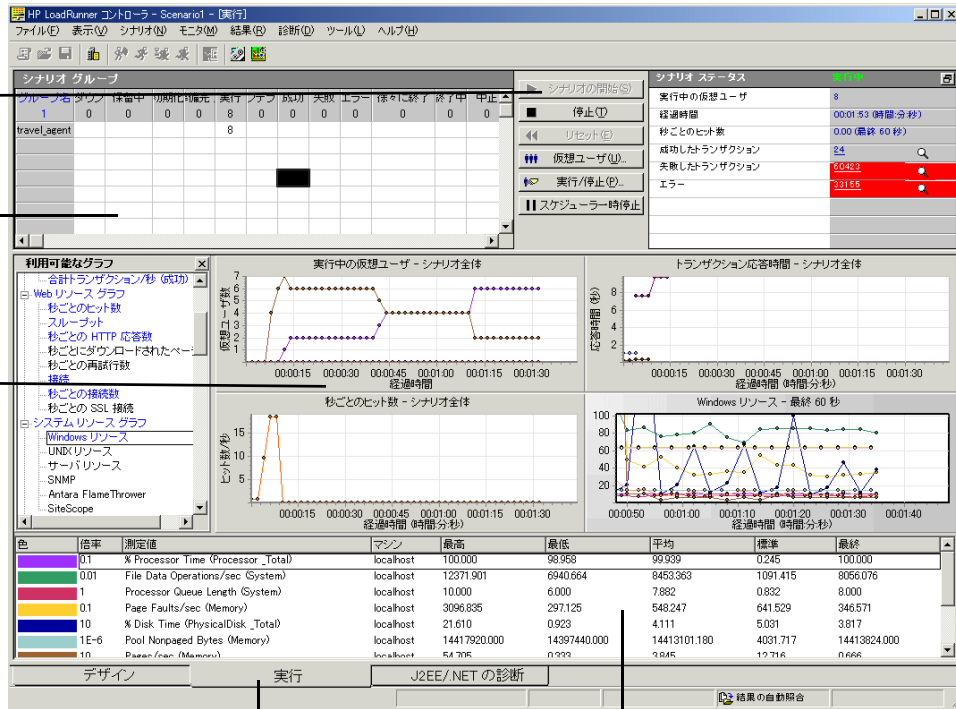
[シナリオ開始] ボタンをクリックします。Controller の実行ビューが表示され、シナリオの実行が開始されます。

[シナリオ グループ] 表示枠では、仮想ユーザが徐々に実行され、システムに対して負荷が生成されるのが確認できます。仮想ユーザのアクションに対するサーバの応答性は、オンライン・グラフで確認できます。

[シナリオの開始] :

[シナリオグループ] セクション

[オンライングラフ] 領域



[実行] タブ

グラフ測定値領域

注： このチュートリアル用シナリオは、英語版 LoadRunner で作成されています。そのため、表示される語句が英語の場合があります。シナリオを日本語版 LoadRunner で作成した場合、これらは日本語で表示されます。

負荷テストの監視

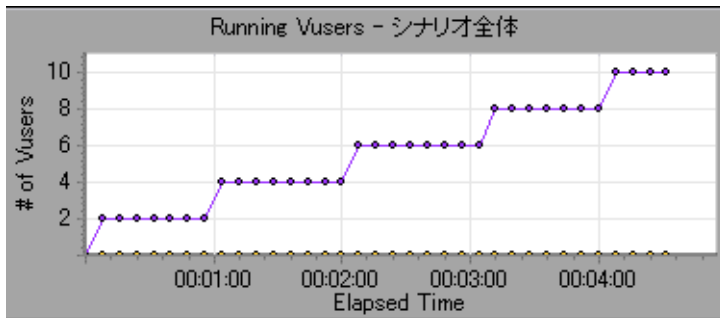
アプリケーションへの負荷の生成中に、アプリケーションがリアルタイムでどのように動作するか、また潜在的なボトルネックがどこに存在するか確認します。負荷テスト中にシステムの層、サーバ、コンポーネントごとのパフォーマンスを測定するには、LoadRunner の統合モニタ・スイートを使用します。

LoadRunner には、Web、アプリケーション、ネットワーク、データベース、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するモニタがあります。

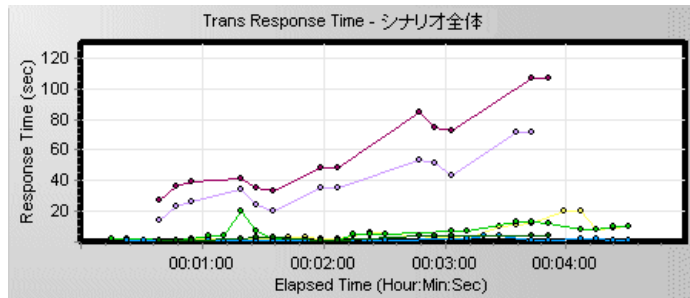
1 標準のグラフを表示します。

標準設定では、Controller には、実行中の仮想ユーザ、トランザクション応答時間、秒ごとのヒット数、Windows リソースのグラフが表示されます。最初の 3 つは設定不要です。Windows リソース・モニタは、このテストに合わせて設定されています。

- ▶ **[Running Vusers - シナリオ全体]** グラフでは、特定の時間に実行されている仮想ユーザの数を監視できます。1 分ごとに 2 個の割合で仮想ユーザが徐々に実行されていくのが確認できます。



- ▶ **[Trans Response Time - シナリオ全体]** グラフでは、各トランザクションが完了するのに要した時間が監視できます。顧客がシステムへのログオン、フライトの検索、航空券の購入、行程の確認、システムからのログオフを行うのに要する時間を確認できます。



テスト対象アプリケーションで作業する仮想ユーザ数が増えるのに従って、トランザクション応答時間が長くなり、顧客に提供するサービスのレベルが低下することがわかります。

- ▶ **[Hits per Second - シナリオ全体]** グラフでは、シナリオ実行の経過秒ごとに仮想ユーザが行った Web サーバに対するヒット (HTTP 要求) の数が監視できます。これにより、サーバに対して生成された負荷の量を確認できます。
- ▶ **[Windows Resources]** グラフでは、シナリオ実行時に測定された Windows リソースの使用状況 (CPU、ディスク、メモリの使用率など) が監視できます。Windows リソースおよびその他のモニタの設定方法については、練習 7「負荷テスト・シナリオの作成」で学びます。

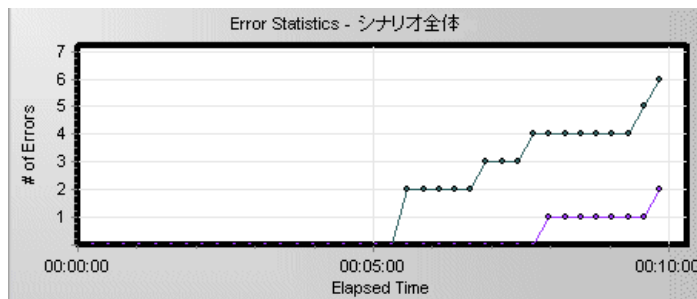
各測定値は、[グラフの凡例] の色分けされた行に表示されます。各行は、グラフ上の同じ色の線に対応しています。

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。

2 エラー情報の表示

マシンで重い負荷を処理していると、エラーが発生する可能性があります。

[利用可能なグラフ] ツリーで [**Error Statistics**] グラフを選択し、[Windows Resources] グラフ表示枠までドラッグします。[Error Statistics] グラフには、シナリオ実行時のエラーの数と、エラーが発生した時間に関する詳細が表示されます。エラーは、発生元によりグループ分けされます。たとえば、スクリプト内の場所や Load Generator 名などに分けられます。



この例では、5分後に、次第に増加する多くのエラーがシステムで発生し始めていることがわかります。これらのエラーは、応答時間の遅延によるタイムアウトが原因で発生しました。

注：シナリオの実行には数分かかります。シナリオが、続けて結果をオンラインで表示したら、グラフと仮想ユーザを切り替えることができます。

結果の分析

テストの実行後、LoadRunner に、詳細なグラフとレポートで構成される綿密な分析セクションが表示されます。複数のシナリオの結果を結合すれば、複数のグラフを比較できます。あるいは自動相関ツールを使用して、応答時間に影響を与えた可能性のあるデータが含まれる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特定できます。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切り分けることができ、パフォーマンスを改善するためにシステムのどの部分の修正が必要かが簡単に分かります。Analysis ツールの使用方法については、練習 10「シナリオの分析」で学びます。



Analysis をシナリオの結果が表示した状態で開くには、[結果] > [結果の設定] を選択するか、[結果の分析] ボタンをクリックします。結果は、
< LoadRunner インストール先フォルダ > ¥Results¥tutorial_demo_res ディレクトリに保存されます。



次の練習について

LoadRunner の自動テスト・プロセスの能力が確認できたら、これで負荷テスト・スクリプトの作成方法、独自のテストの設計および実行方法、テスト結果の分析方法を学ぶ準備ができました。次の練習では、これらの結果を得るプロセスを見ていきます。

練習 3

スクリプトの作成

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自動スクリプトを作成します。

この練習は、次の項目で構成されています。

- ▶ Virtual User Generator (VuGen) について
- ▶ ユーザ動作の記録の開始方法
- ▶ VuGen ウィザード・モードの使用方法
- ▶ ビジネス・プロセスの記録によるスクリプトの作成方法
- ▶ スクリプトの表示方法

Virtual User Generator (VuGen) について

テスト環境において、LoadRunner は、物理的なマシンの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、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。

LoadRunner Virtual User Generator (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、負荷テストの基礎となる自動スクリプトにアクションを記録します。

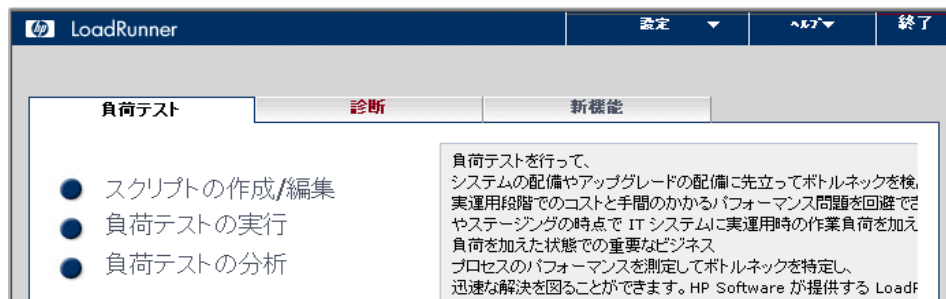
ユーザ動作の記録の開始方法

ユーザ・アクションの記録を開始するには、VuGen を起動し、空のスクリプトを作成します。そして、イベントを記録し、手動拡張を追加することによって、空のスクリプトを埋めていきます。

本項では、VuGen を起動し、空の Web スクリプトを作成します。

1 LoadRunner を起動します。

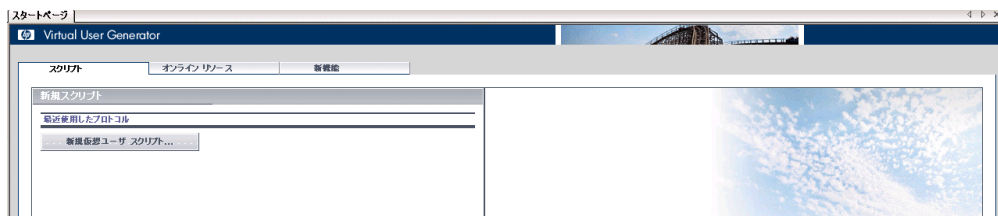
[スタート] > [プログラム] > [LoadRunner] > [LoadRunner] を選択します。HP LoadRunner Launcher ウィンドウが開きます。



2 VuGen を開きます。

Launcher ウィンドウで、[負荷テスト] タブをクリックします。

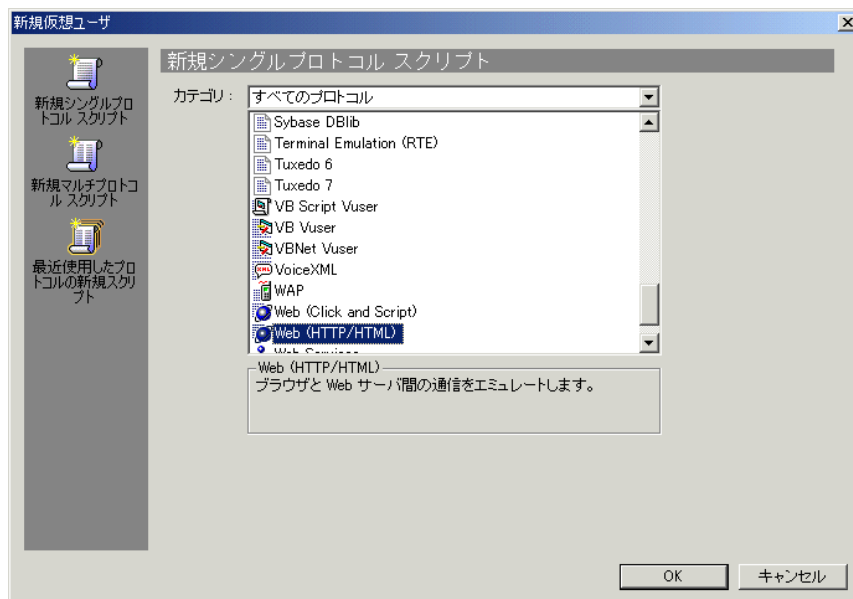
[スクリプトの作成 / 編集] をクリックします。VuGen の開始ページが開きます。



3 空の Web スクリプトを作成します。

VuGen の開始ページで [スクリプト] タブにある [新規仮想ユーザ スクリプト...] をクリックします。

[新規仮想ユーザ] ダイアログ・ボックスが開き、新規シングル・プロトコル・スクリプトのオプションが表示されます。



プロトコルとは、システムのバック・エンドとの通信にクライアントが使用する言語です。HP Tours は Web ベースのアプリケーションであるため、Web 仮想ユーザ・スクリプトを作成します。

カテゴリ・タイプが [すべてのプロトコル] になっていることを確認します。VuGen では、シングル・プロトコル・スクリプトで使用できる全プロトコルのリストが表示されます。リストをスクロール・ダウンし、[Web (HTTP/HTML)] を選択し、[OK] をクリックして、空の Web スクリプトを作成します。

注：上級ユーザは、1つの記録セッション中に複数のプロトコルをマルチ・プロトコル・スクリプトに記録できます。本チュートリアルでは、Web タイプのシングル・プロトコル・スクリプトを作成します。その他のタイプのシングル・プロトコル・スクリプトまたはマルチ・プロトコル・スクリプトの記録手順は、Web スクリプトの記録手順と似ています。

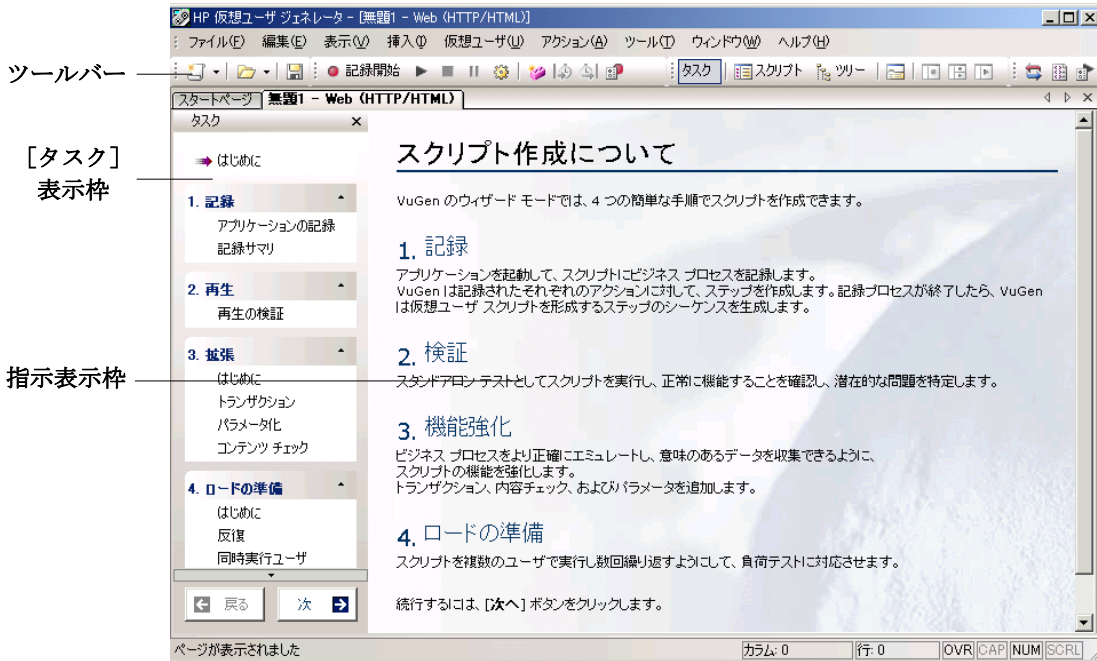
VuGen ウィザード・モードの使用法

タスク

左側に表示される**タスク表示枠**とともに、VuGen の**ウィザード・モード**で空のスクリプトが開きます（タスク表示枠が表示されない場合は、ツールバーの**[タスク]** ボタンをクリックします。[記録の開始] ダイアログ・ボックスが自動的に開いた場合は、**[キャンセル]** をクリックします。

VuGen のウィザードでは、段階的にスクリプトを作成し、そのスクリプトをテスト環境に適合させることができます。

タスク表示枠には、スクリプト作成プロセスの各手順またはタスクが表示されます。各手順の進行に従い、VuGen によってウィンドウのメイン・エリアに詳細な指示およびガイドラインが表示されます。



VuGen ウィンドウをカスタマイズして、ツールバーの表示と非表示を切り替えることができます。ツールバーの表示と非表示を切り替えるには、**[表示]** > **[ツールバー]** を選択し、目的のツールバーの横にあるチェック・マークを切り替えます。

タスク表示枠を開き、いずれかのタスク・ステップをクリックすれば、どの段階でも VuGen ウィザードに戻ります。

ビジネス・プロセスの記録によるスクリプトの作成方法

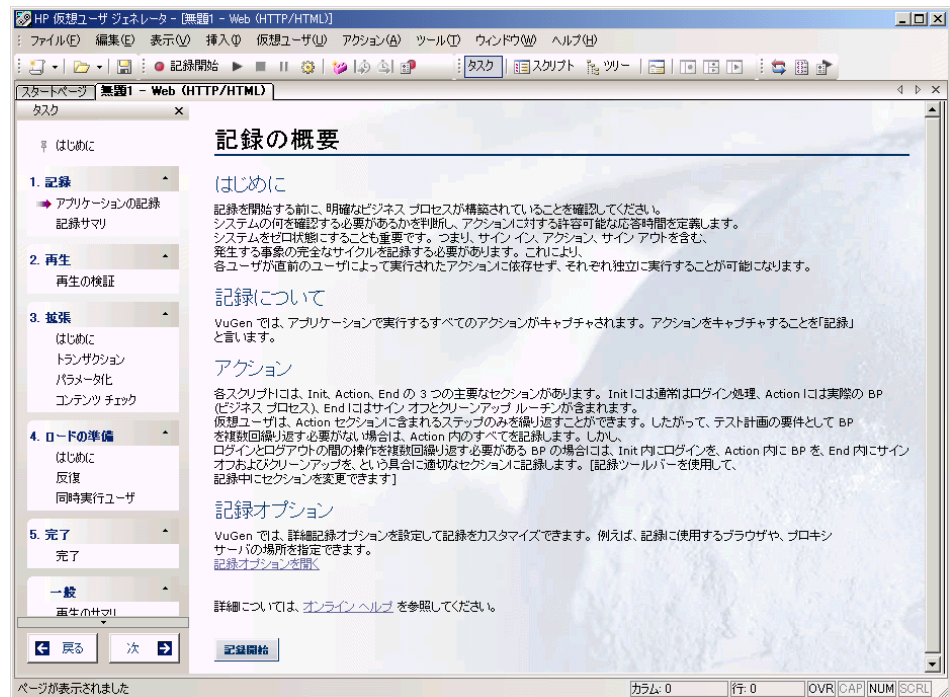
ユーザ・エミュレーションの作成における次の手順は、実際のユーザが実行するイベントの記録です。前の項では、空の Web スクリプトを作成しました。これで、イベントを直接スクリプトに記録できるようになりました。本項では、デンバーからロサンゼルスへのフライトを予約し、フライト日程を確認する 1 人の乗客のイベントを追跡します。

スクリプトを記録するには、次の手順を実行します。

1 HP Tours Web サイトで記録を開始します。

タスク表示枠の手順 1 [アプリケーションの記録] をクリックします。

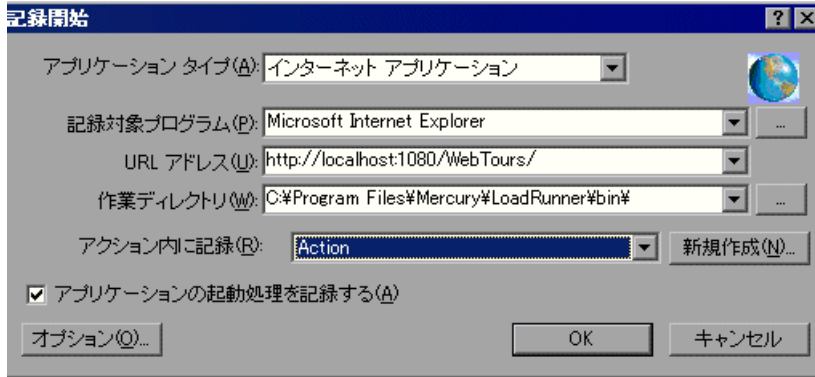
指示表示枠の最下部にある [記録開始] をクリックします。



あるいは、[Vuser] > [記録開始] を選択するか、ページ最上部のツールバーにある [記録開始] ボタンをクリックします。

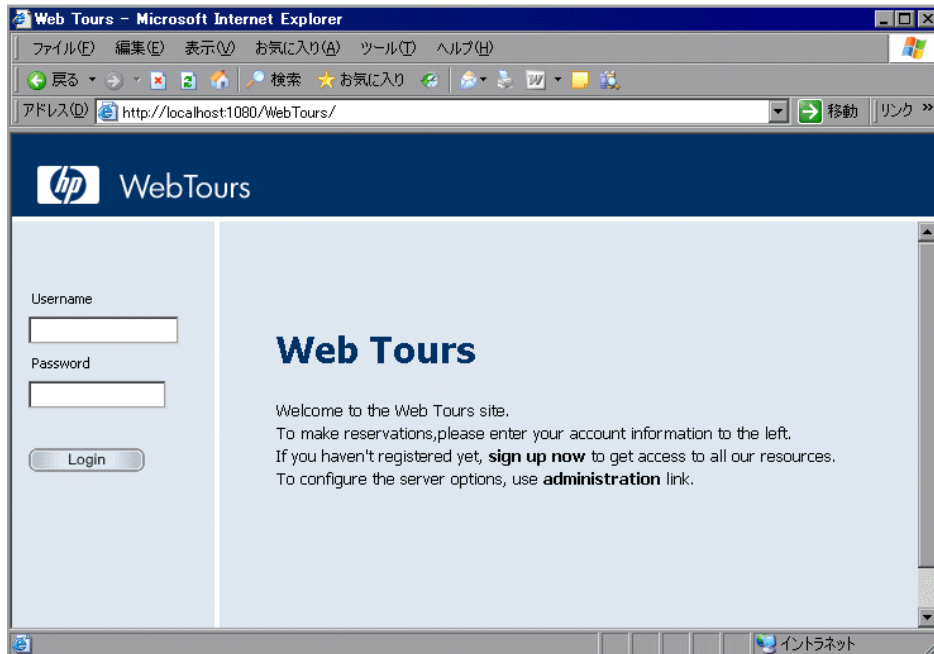
練習 3・スクリプトの作成

[記録開始] ダイアログ・ボックスが開きます。



[URL アドレス] ボックスに「http://localhost:1080/WebTours/」と入力します。
[アクション内に記録] ボックスで、[Action] を選択します。[OK] をクリックします。

新しい Web ブラウザが開き、HP Tours サイトが表示されます。



注：サイトを開く際にエラーが発生した場合は、Web サーバが動作しているか確認します。サーバを起動するには、[スタート] > [プログラム] > [LoadRunner] > [Samples] > [Web] > [Start Web Server] を選択します。

フローティング記録ツールバーが開きます。



2 HP Tours Web サイトにログオンします。

[User Name] に「jojo」と入力し、[Password] に「bean」と入力します。
[Login] をクリックします。「Welcome」ページが開きます。

3 フライトの詳細を入力します。

[Flights] をクリックします。「Find Flight」ページが開きます。

- [Departure City] : Denver (標準設定)。
- [Departure Date] : 標準設定のままとします。現在の日付になります。
- [Arrival City] : Los Angeles
- [Return Date] : 標準設定のままとします。明日の日付になります。
- [Seating Preference] : Aisle (通路側)。

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。

[Credit Card] ボックスに「12345678」と入力し、[Exp Date] ボックスに「06/06」と入力します。[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。

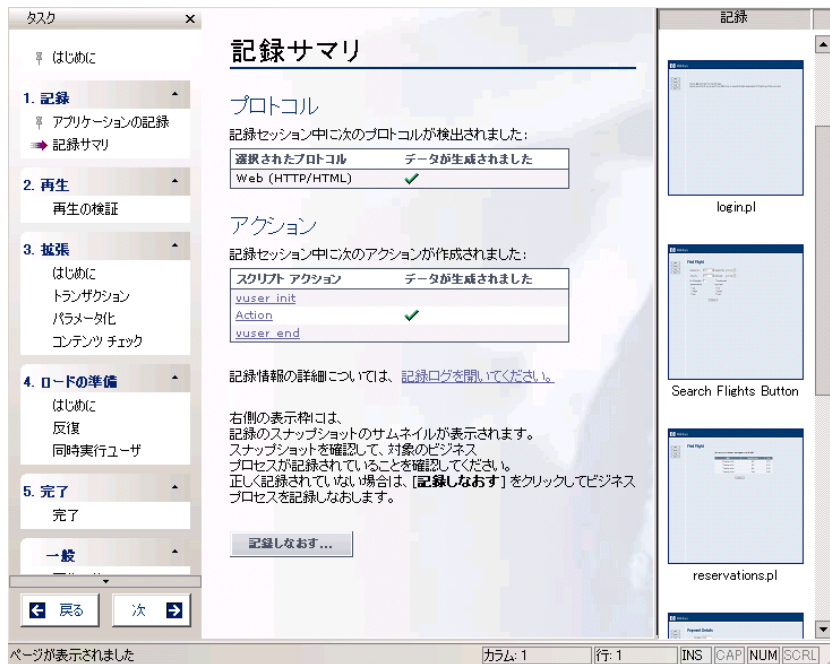
6 行程を確認します。

左側の表示枠で **[Itinerary]** をクリックします。「Itinerary」ページが開きます。

7 左側の表示枠で **[Sign Off]** をクリックします。

8 フローティング・ツールバーの **[停止]** をクリックして、記録プロセスを停止します。

仮想ユーザ・スクリプトが作成されている間、[コード生成] ポップ・アップ・ウィンドウが開いています。VuGen ウィザードは、タスク表示枠の次の手順に自動的に進み、記録サマリを表示します（サマリが表示されない場合は、タスク表示枠の **[記録サマリ]** をクリックします）。



記録サマリには、プロトコル情報、およびセッション中に作成されたアクションのリストが記載されます。記録時に実行した手順ごとに、VuGen によってスナップショット（記録時のウィンドウの画像）が生成されています。

記録されたスナップショットのサムネイルは、右側の表示枠に表示されます。理由に関係なく、スクリプトを再記録する必要がある場合は、ページの最下部にある **[記録しなおす]** ボタンをクリックします。



- 9 [ファイル] > [保存] を選択するか, [保存] ボタンをクリックします。
[ファイル名] ボックスに「basic_tutorial」と入力し, [保存] をクリックします。VuGenによってLoadRunnerスクリプト・フォルダにファイルが保存され, タイトル・バーにテスト名が表示されます。

スクリプトの表示方法

前の項では, 旅行代理店によるログイン, フライト予約, ログオフを記録しました。VuGenには, [記録開始] ボタンをクリックしてから [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内のスクリプトを参照できるようになります。スクリプトは, ツリー・ビューまたはスクリプト・ビューで確認できます。ツリー・ビューは, 仮想ユーザのアクションをステップとして表示するアイコンベースのビューであり, スクリプト・ビューは, 仮想ユーザのアクションを関数として表示するテキストベースのビューです。

ツリー・ビュー



スクリプトをツリー・ビューで表示するには, [表示] > [ツリー ビュー] を選択するか, [ツリー ビュー] ボタンをクリックします。ツリー・ビューをウィンドウ全体に表示するには, [タスク] ボタンをクリックして, タスク表示枠を非表示にします。

The screenshot shows the VuGen interface for a test named 'basic_tutorial - Web (HTTP/HTML)'. On the left, the 'Action0' tree view is expanded, showing a list of actions: 'Url: WebTours', 'Submit Form: login.pl', 'Image: Search Flights Button', 'Submit Form: reservations.pl', 'Submit Form: reservations.pl_2', 'Submit Form: reservations.pl_3', and 'Image: SignOff Button'. On the right, a '記録時のスナップショット' (Snapshot during recording) is displayed, showing a screenshot of the HP WebTours application. The application has a blue header with the HP logo and 'WebTours' text. Below the header, there are buttons for 'Flights', 'Itinerary', 'Home', and 'Sign Off'. To the right of these buttons is a 'Find Flight' section with a 'Departure City' dropdown menu set to 'Denver' and a 'Departure Date' field. The interface also includes tabs for 'ページビュー', 'クライアントの要求', and 'サーバの応答'.

テスト・ツリー

スナップショット

練習 3・スクリプトの作成

記録時に実行したステップごとに、VuGen によってテスト・ツリーにアイコンおよびタイトルが生成されています。ツリー・ビューには、ユーザのアクションがスクリプト・ステップとして表示されます。大部分のステップには、対応する記録のスナップショットが添付されています。

スナップショットにより、記録プロセス時に記録された画面が正確に確認できるため、スクリプトが理解しやすくなり、複数のエンジニアによるスクリプトの共有が容易になります。後でスナップショットを比較して、スクリプトの正確さを検証できます。また、VuGen は、再生時に各ステップのスナップショットを作成します。

テスト・ツリーのステップの横にあるプラス記号 (+) をクリックします。フライト予約時に記録された**思考遅延時間**が表示されます。思考遅延時間は、ステップ間で待機した実際の時間を表し、ユーザの速いまたは遅い振る舞いを負荷下でエミュレートするのに使用できます。思考遅延時間は、実際のユーザの振る舞いをより正確に負荷テストに反映できるメカニズムです。

スクリプト・ビュー

スクリプト・ビューは、仮想ユーザのアクションを API 関数として表示するテキストベースのビューです。スクリプト・ビューでスクリプトを表示するには、**[表示] > [スクリプト ビュー]** を選択するか、**[スクリプト ビュー]** ボタンをクリックします。



VuGen
エディタ

```
スタートページ basic_tutorial - Web (HTTP/HTML)
vuser_init
Action
vuser_end
globals.h

Action()
{
    web_url("MercuryWebTours",
        "URL=http://localhost:1080/MercuryWebTours/"
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t1.inf",
        "Mode=HTML",
        LAST);

    lr_think_time(11);

    web_submit_form("login.pl",
        "Snapshot=t2.inf",
        ITEMDATA,
        "Name=username", "Value=jojo", ENDITEM,
        "Name=password", "Value=bean", ENDITEM,
        LAST);
}
```

ヘルプを表示するには、F1 を押します。 カラム: 1 行: 24 INS CAP NUM SC

スクリプト・ビューでは、色分けされた関数とその引数値が記載されたスクリプトがエディタに表示されます。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このウィンドウに入力できます。

注： LoadRunner では ANSI C が使用されます。スクリプトがより高度化しても、C 言語を使用すれば、記録と再生以上の機能で LoadRunner を拡張できます。



次の練習について

これで、基本となるスクリプトの記録および表示の学習が終了し、練習 4「スクリプトの再生」に進む準備ができました。x

練習 3・スクリプトの作成

練習 4

スクリプトの再生

フライトの予約などの標準的なユーザ・アクション・セットを記録することによって、実際のユーザのエミュレーションを作成しました。記録されたスクリプトを負荷テスト・シナリオに組み込む前に、記録されたスクリプトが正しく実行されるか検証するには、記録されたスクリプトを再生します。再生中には、ブラウザでアクションを表示して、すべてが期待どおりか確認できます。スクリプトが正しく再生されない場合は、練習 5「再生に関する一般的な問題の解決方法」の説明に従って関連の追加が必要な場合があります。

スクリプトを再生する前に、仮想ユーザの振る舞いを設定するのに役立つ実行環境を設定できます。

この練習は、次の項目で構成されています。

- ▶ 実行時の振る舞いの設定方法
- ▶ スクリプト実行のリアルタイムでの監視方法
- ▶ 再生に関する情報の表示方法
- ▶ テストの成功の確認方法
- ▶ 結果の検索またはフィルタ方法

実行時の振る舞いの設定方法

LoadRunner の実行環境の設定では、さまざまな種類の実際のユーザの動作や振る舞い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応答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応答するユーザをエミュレートすることもできます。また実行環境の設定では、仮想ユーザがアクションを反復する回数および頻度も指定できます。

練習 4・スクリプトの再生

一般的な実行環境の設定と、特定の仮想ユーザ・タイプに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、Internet Explorer ではなく Netscape 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。固有の設定については、練習 6「負荷テスト用スクリプトの準備」で取り上げます。

この練習では、すべてのタイプのスクリプトに適用される一般的な実行環境の設定について説明します。次のような設定が含まれます。

- ▶ **[実行論理]**：反復の回数。
- ▶ **[ペースの設定]**：反復の間隔。
- ▶ **[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。
- ▶ **[ログ]**：再生中に収集する情報のレベル。

LoadRunner Controller からでも実行環境の設定を変更できます。これについては後の練習で説明します。

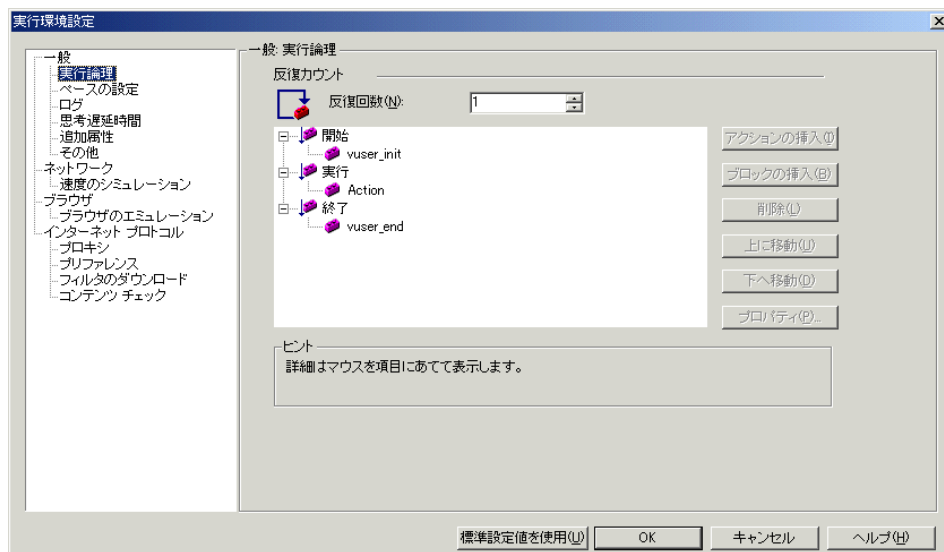
1 **[実行環境設定]** ダイアログ・ボックスを開きます。

タスク表示枠が表示されていることを確認します（表示されていない場合は、**[タスク]** ボタンをクリックします）。タスク表示枠の **[再生の検証]** をクリックします。

指示表示枠の見出し「**実行環境の設定**」の下にある **[実行環境の設定を開く]** ハイパーリンクをクリックします。

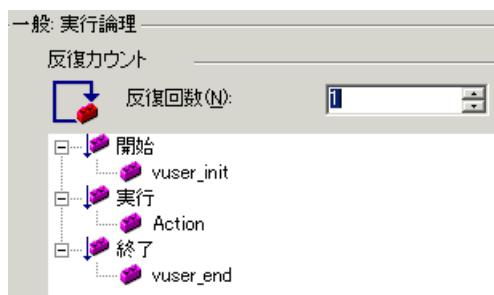


あるいは、F4 キーを押すか、ツールバーの「**実行環境設定の編集**」ボタンをクリックします。「実行環境設定」ダイアログ・ボックスが開きます。



2 実行論理の設定を開きます。

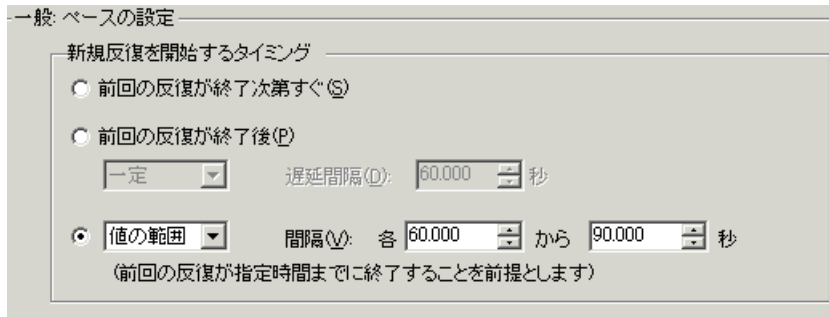
「**実行論理**」ノードを選択します。



このノードには、**反復**の回数、または動作を連続して繰り返す回数を設定します。反復回数を 2 に設定します。

3 ペースを設定します。

[ペースの設定] ノードを選択します。



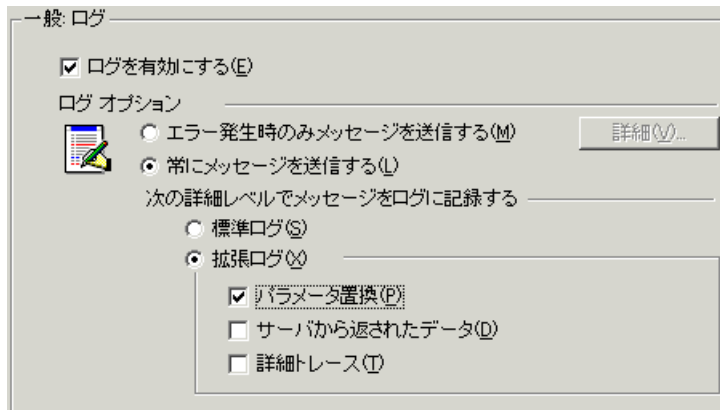
このノードでは、反復の間隔を制御できます。ランダムな間隔を指定します。これにより、動作間でユーザが待機する現実の設定が正確にエミュレートされます。ランダムな間隔では、反復間で正確に 60 秒待機する実際のユーザはありません。

3 番目のオプションを選択し、次のように選択します。

値の範囲 間隔: 各 **60.000** から **90.000** 秒

4 ログを設定します。

[ログ] ノードを選択します。

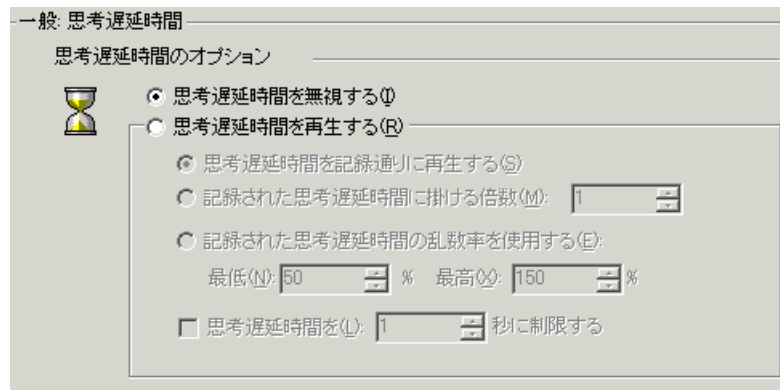


[ログ] 設定では、テストの実行時にログに記録する情報量を指定します。開発中には、デバッグのためにいくつかのログを有効にできます。しかし、スクリプトが正しく動作することを確認したら、エラー・ログのみを有効にするか、ログを無効にします。

[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。

5 思考遅延時間の設定を開きます。

[思考遅延時間] ノードを選択します。



何も変更しないでください。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。

6 [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。

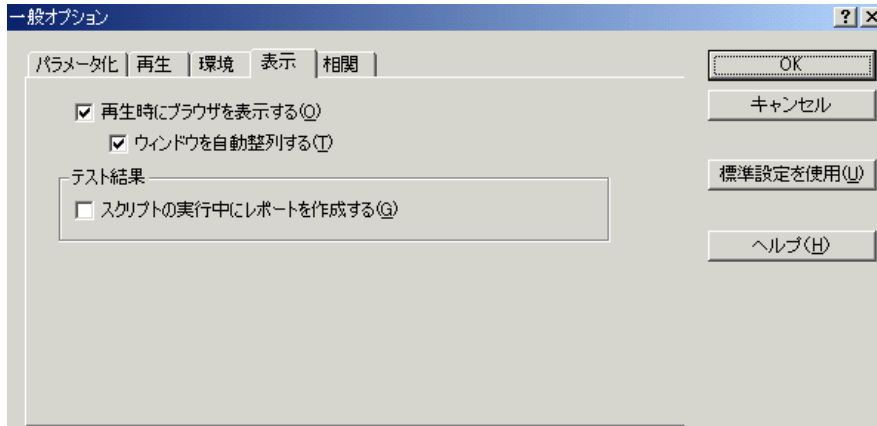
スクリプト実行のリアルタイムでの監視方法

VuGen の実行時ビューア機能により、記録されたスクリプトの再生時に仮想ユーザの動作がリアルタイムで表示されます。

標準設定では、VuGen は、スクリプトのアクションのアニメーションを表示せずにバックグラウンドでテストを実行します。しかし、本チュートリアルでは、VuGen が各ステップを実行する様子を確認できるビューアにアクションが表示されるように VuGen を設定します。ビューアは実際のブラウザではなく、仮想ユーザに返されたページのスナップショットのみが表示されます。

練習 4・スクリプトの再生

- 1 [ツール] > [一般オプション] を選択し、[表示] タブを選択します。
- 2 [再生時にブラウザを表示する] オプションおよび [ウィンドウを自動整列する] オプションを選択します。



- 3 [OK] をクリックして、[一般オプション] を閉じます。
- 4 タスク表示枠の [再生の検証] をクリックし、指示表示枠の最下部にある [再生の開始] ボタンをクリックします。あるいは、F5 キーを押すか、ツールバーの [実行] ボタンをクリックします。
- 5 [結果ディレクトリの指定] ダイアログ・ボックスが開き、結果ディレクトリの格納先を求められたら、標準の名前を受け入れ、[OK] をクリックします。

しばらくして、VuGen が実行時ビューアを開き、最後に開いたものに応じてスクリプト・ビューまたはツリー・ビューでスクリプトの実行を開始します。実行時ビューアでは、仮想ユーザのアクションを視覚的に観察できます。この再生は、記録時に実行したステップの正確な再生です。

再生に関する情報の表示方法

スクリプトの実行が停止したら、再生のサマ리를ウィザードで表示できます。タスク表示枠の「**再生の検証**」をクリックして、最新の再生のサマ리를表示します。

最新の再生のサマリ

結果ディレクトリ: result1
 再生ステータス: **正常に実行されました。**
 注意: ランタイム エラーは検出されませんでした。しかし、右側の表示枠の再生時のスナップショットと記録時のスナップショットを比較してビジネス プロセスが正しく再生されたかどうかを確認してください。

動的サーバの値 (相関):
 VuGen がスクリプト内で 1 つ以上の動的な値を検出しました。サーバが記録された値と異なる一意の値を期待している場合、再生中に問題が発生する可能性があります。これらの値を相関することを推奨します。
[相関の詳細について](#)
 VuGen が検出した動的な値の一覧を表示できます。
[動的なサーバ値の表示および解決](#)

検証:
 再生が、記録したセッションを正確にエミュレートするかどうかを確認するには、右側の表示枠の再生時と記録時のスナップショットを比較します。

再生の詳細情報は、[ビジュアル テスト結果](#)、または再生ログから取得できます。

スクリプトを表示または変更するには、[編集モードに切り替えます](#)。

[実行環境の設定](#)
 を使用して再生時に異なるユーザの動作をエミュレートします。

再生しなおす オンライン ヘルプ...

最新の再生のサマリには、検出されたエラーが一覧表示され、記録時および再生時のスナップショットのサムネイルが表示されます。スナップショットを比較して、記録と再生との間の不一致を調べることができます。

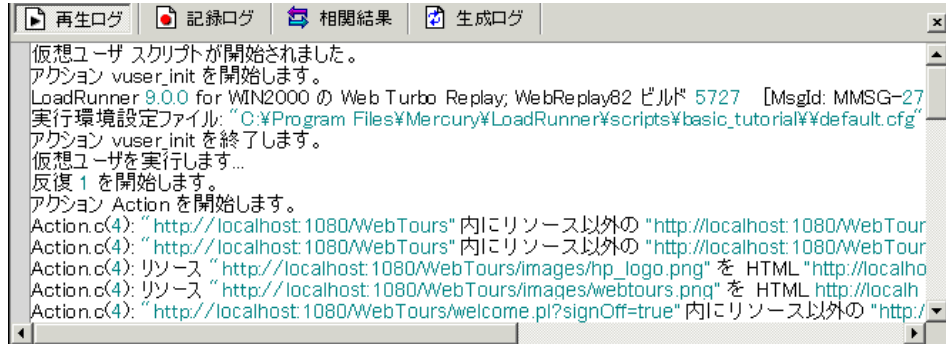
また、イベントのテキスト・サマ리를確認することによって、仮想ユーザの動作を調べることができます。VuGen の「出力」ウィンドウの「**再生ログ**」タブには、この情報が色分けされて表示されます。

本項では、再生ログを開き、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 指示表示枠の「再生ログから取得できます。」ハイパーリンクをクリックします。あるいは、ツールバーの「出力ウィンドウの表示 / 非表示」ボタンをクリックするか、メニューから「表示」>「出力ウィンドウ」を選択します。次に、「再生ログ」タブをクリックします。



- 2 再生ログで Ctrl キーを押しながら F キーを押して、「検索」ダイアログ・ボックスを開きます。次の項目を見つけます。

- ▶ **開始, 終了** : スクリプト実行の開始および終了（仮想ユーザ スクリプトが開始されました, 仮想ユーザが終了しました）。
- ▶ **反復** : 反復の開始および終了, ならびに反復の回数。

VuGen では、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、テストの途中で接続が途切れた場合、VuGen はエラーの行番号を示し、行全体を赤いテキストで表示します。

- 3 再生ログの行をダブルクリックします。スクリプト内の対応するステップに移動します。スクリプト・ビューの左側にある黒い線がそのステップを表します。

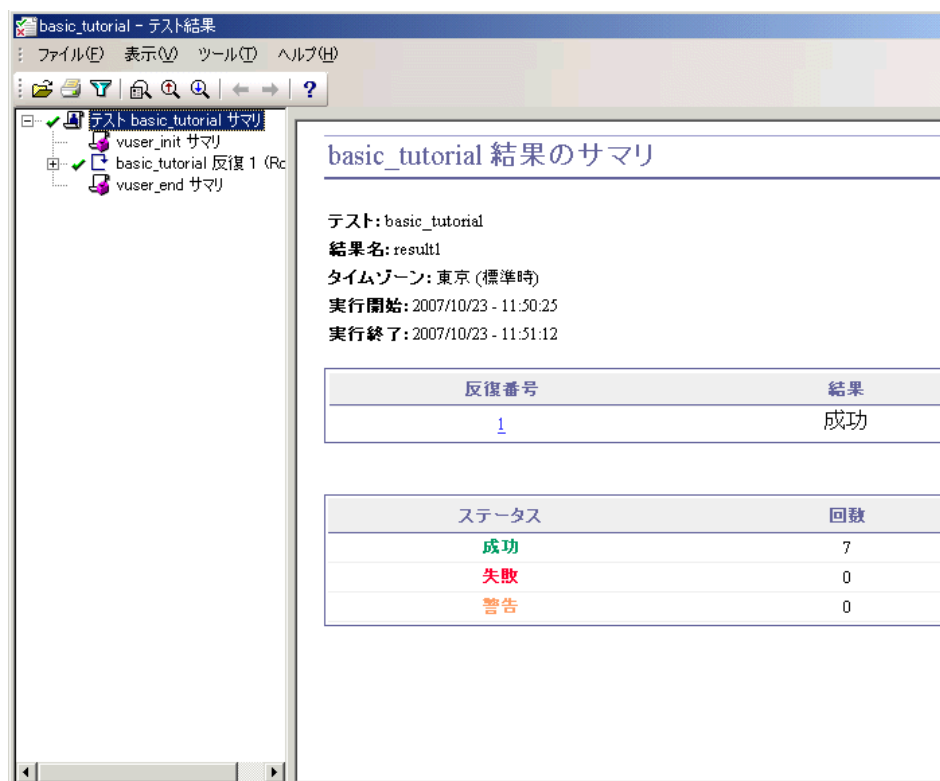
テストの成功の確認方法

記録したイベントを再生したら、結果を調べて、すべて成功したか確認する必要があります。何かが失敗した場合は、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。

テスト結果を表示するには、次の手順を実行します。

- 1 タスク表示枠の [再生の検証] をクリックして、ウィザードに戻ります。
- 2 指示表示枠の見出し「検証」の下にある [ビジュアル テスト結果] ハイパーリンクをクリックします。あるいは、[表示] > [テスト結果] を選択します。新しい結果ウィンドウが開きます。



The screenshot shows the 'basic_tutorial - テスト結果' window. The left pane shows a tree view with the following items: 'テスト basic_tutorial サマリ' (checked), 'vuser_init サマリ', 'basic_tutorial 反復 1 (Rc)' (checked), and 'vuser_end サマリ'. The right pane displays the summary for 'basic_tutorial 結果のサマリ'.

テスト: basic_tutorial
結果名: result1
タイムゾーン: 東京 (標準時)
実行開始: 2007/10/23 - 11:50:25
実行終了: 2007/10/23 - 11:51:12

反復番号	結果
1	成功

ステータス	回数
成功	7
失敗	0
警告	0

練習 4・スクリプトの再生

[テスト結果] ウィンドウは、初めて開いた場合、表示枠が 2 つ表示されます (左側にツリー表示枠、右側にサマリ表示枠)。

ツリー表示枠には、結果ツリーが表示されます。各反復には番号が付けられます。サマリ表示枠には、テストの詳細が表示されます。

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。VuGen の仮想ユーザが最初の記録に従って HP 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見なされます。

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。

次の項では、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、テスト結果をドリル・ダウンします。

結果の検索またはフィルタ方法

何かが失敗したことをテスト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。

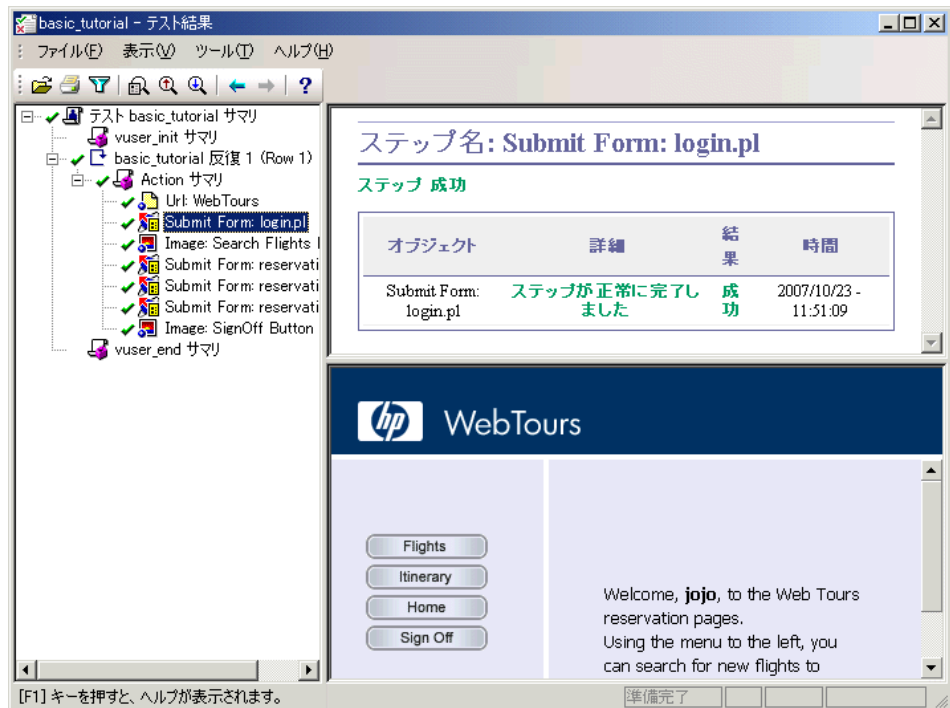
[テスト結果] ウィンドウで、テスト・ツリーを展開して、各ステップの結果を個別に表示できます。[テスト結果] ウィンドウには、その反復時の再生のスナップショットが表示されます。

1 反復の分岐を展開します。

左側の表示枠のプラス記号をクリックして [反復 1] 分岐を展開し、次に [Action サマリ] 分岐を展開します。展開された分岐には、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。

2 番目のステップ **Submit Form** を選択します。[テスト結果] ウィンドウには、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。

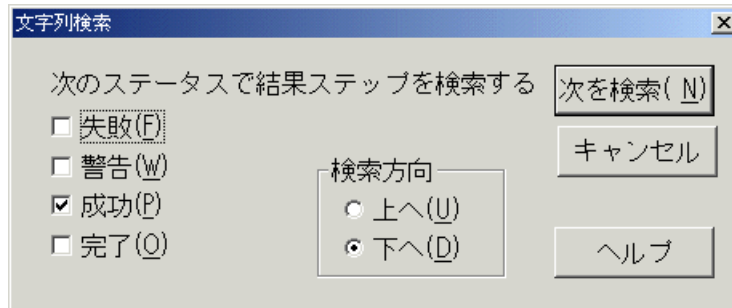
[テスト結果] ウィンドウの右上の表示枠には、オブジェクトまたはステップの名前、ページが正常に読み込まれたかどうかに関する詳細、結果 (**成功**, **失敗**, **完了**, **警告**)、ステップが実行された時間といったステップのサマリ情報が表示されます。

4 結果ステータスを検索します。

全体的な結果サマリがテストが失敗したことを示している場合は、失敗した箇所を特定する必要があります。**失敗**という単語がないかテスト結果を検索できます。



テスト結果を検索するには、[ツール] > [検索] を選択するか、[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。



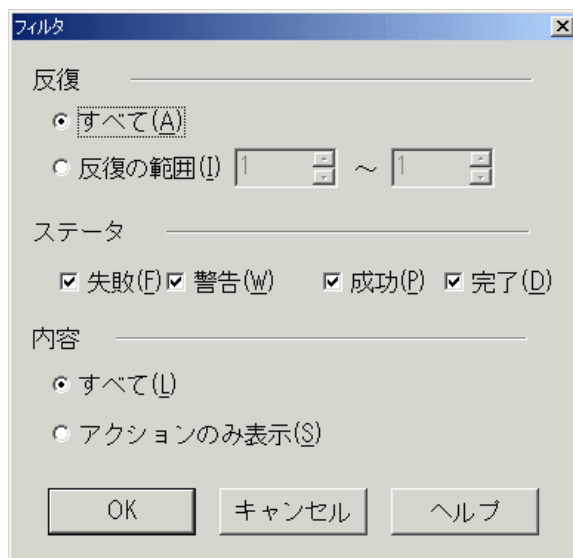
[成功] オプションを選択し、他のオプションが選択されていないことを確認して、[次を検索] をクリックします。[テスト結果] ウィンドウには、ステータスが**成功**の最初のステップが表示されます。

再度、[ツール] > [検索] を選択するか、[検索] ボタンをクリックします。[検索] ダイアログ・ボックスで、[失敗] オプションを選択し、[成功] オプションをクリアして、[次を検索] をクリックします。[テスト結果] ウィンドウでは、失敗した結果は見つかりませんでした。

5 結果にフィルタを適用します。

[テスト結果] ウィンドウ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。

結果にフィルタを適用するには、[表示] > [フィルタ] を選択するか、[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。



[ステータス] セクションで [**失敗**] オプションを選択し、その他のすべてのオプションをクリアします。[内容] セクションで [**すべて**] オプションを選択し、[OK] をクリックします。失敗がなかったため、左側の表示枠は空になります。

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 Tours アプリケーションのテストは正常に実行されました。練習 5 「再生に関する一般的な問題の解決方法」に進んでください。

練習 4 • スクリプトの再生

練習 5

再生に関する一般的な問題の解決方法

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、単純な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。

LoadRunner では、**相関**を利用してこの問題に対処しています。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、エミュレーションの実行時、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、LoadRunner が動的な値に関する問題を自動的に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 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 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 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 Tours Web 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。

1 HP Tours を起動します。

[スタート] > [プログラム] > [LoadRunner] > [Samples] > [Web] > [HP Web Tours Application] を選択します。ブラウザが開き、HP Tours の最初の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。

HP Web Tours の最初のページの [administration] リンクをクリックします。管理ページが開きます。

[Set LOGIN form's action tag to an error page] というタイトルの 3 番目のチェックボックスを選択します。ページの最下部までスクロールし、[Update] をクリックします。

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。

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。

3 ブラウザを閉じます。

一意のサーバ値の使用方法

変更後の HP Tours の設定では、サーバは、仮想ユーザに一意のセッション ID を割り当てます。スクリプトを再生しようとする、その再生は失敗します。

この問題に対処するために、VuGen を使用して、セッション ID の関連の必要性を自動的に検出します。スクリプトの実行後、VuGen は、スクリプトを対象に相関をスキャンするようにメッセージを表示します。

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、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。以降のステップでは、初めに記録された値の代わりに、保存された値が使用されます。

1 動的な値を持つ新しいスクリプトを記録します。

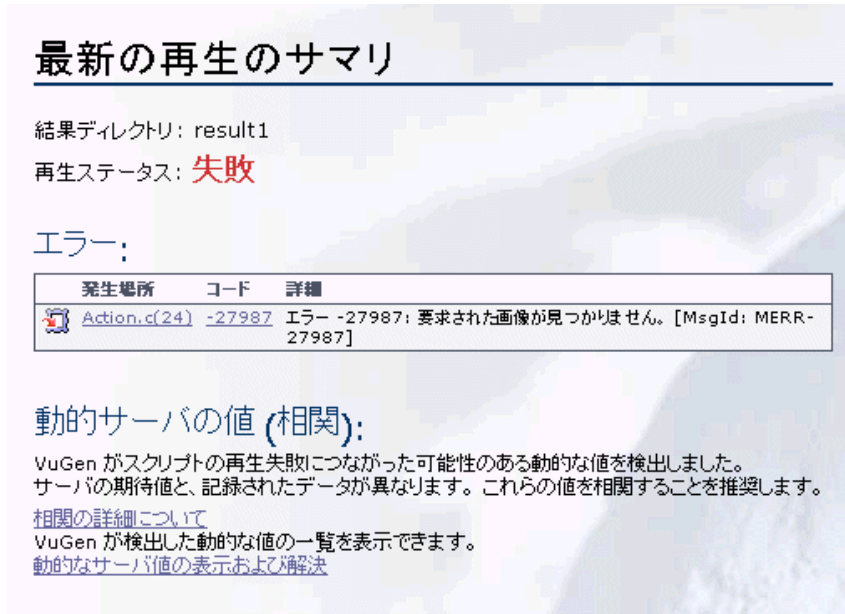
練習 3 (33 ページ「ビジネス・プロセスの記録によるスクリプトの作成方法」) で以前に記録したのと同じステップを持つ新しいスクリプトを記録し、そのスクリプトを `Basic_Tutorial_Cor` という名前で保存します。

2 スクリプトを再生します。

タスク表示枠の **[再生の検証]** をクリックし、指示表示枠の最下部にある **[再生の開始]** ボタンをクリックします。VuGen が新しいスクリプトを実行します。出力ウィンドウの **[再生ログ]** タブに、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。

3 再生のサマリを表示します。


タスク表示枠の [再生の検証] をクリックして、最新の再生のサマリを表示します。



最新の再生のサマリ

結果ディレクトリ: result1
再生ステータス: **失敗**

エラー:

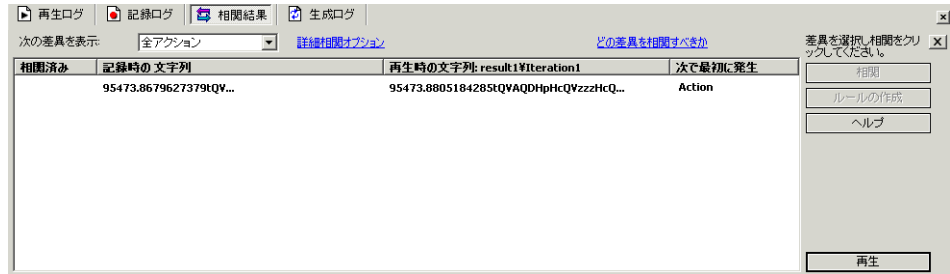
発生場所	コード	詳細
 Action.c(24)	-27987	エラー -27987: 要求された画像が見つかりません。 [MsgId: MERR-27987]

動的サーバの値 (相関):
VuGen がスクリプトの再生失敗につながった可能性のある動的な値を検出しました。サーバの期待値と、記録されたデータが異なります。これらの値を相関することを推奨します。
[相関の詳細について](#)
VuGen が検出した動的な値の一覧を表示できます。
[動的なサーバ値の表示および解決](#)

このサマリは、スクリプトが失敗したことを示しています。

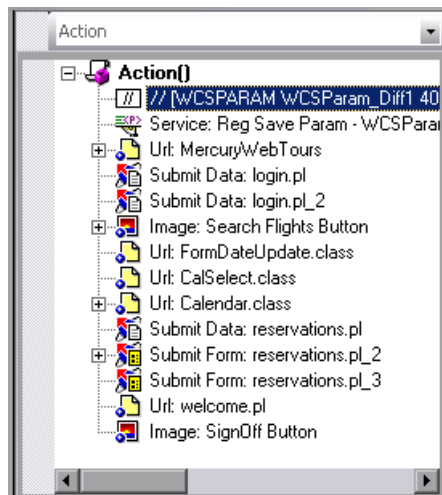
4 スクリプトを対象に相関をスキャンします。

指示表示枠の見出し「動的サーバの値（相関）」の下にある「**動的なサーバ値の表示および解決**」ハイパーリンクをクリックします。VuGen はスクリプトをスキャンし、記録された値と再生値との違いを検索します。VuGen は、出力ウィンドウの「**相関結果**」タブに、相関を必要とする可能性のある相違のリストを表示します。



5 セッション ID を相関させます。

「**相関結果**」タブの最初のエントリを選択し、「**相関**」をクリックします。VuGen は、スクリプトの先頭に、元のセッション ID をパラメータに保存する新しいステップを挿入します。VuGen は、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。以降のステップでは、初めに記録された値の代わりに、保存された値が使用されます。この新しいステップをテスト・ツリーに表示するには、「**表示**」 > 「**ツリー ビュー**」を選択します。



6 関連ステートメントの構文を調べます。

[表示] > [スクリプト ビュー] を選択して、スクリプトの関連ステートメントを表示します。VuGen がスクリプトに追加したステートメントは次のようになります。

```
web_reg_save_param ("WCSPParam_Diff1",  
    "LB=userSession value=",  
    "RB=>",  
    "Ord=1",  
    "RelFrameId=1.2.1",  
    "Search=Body",  
    LAST);
```

このステートメントの意味は、「2つの文字列（左の境界 `userSession value=` および右の境界 `>`）の間にあるデータのサーバ応答をチェックし、このデータの最初の出現を `WCSPParam_Diff1` というパラメータに保存する」ということです。

7 再度スクリプトを再生します。

[仮想ユーザ] > [実行] を選択して、再度スクリプトを再生します。再生が終了したら、[表示] > [ツリー ビュー] を選択します。[再生ログ] タブを参照します。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。

スクリプトの2番目のステップ **Service: Reg Save Param** 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` が成功し、相関が機能したことがログからわかります。

8 サーバの設定を修正します。

一意のセッション ID を無視するように、サーバをリセットします。

[スタート] > [プログラム] > [LoadRunner] > [Samples] > [Web] > [HP Web Tours Application] を選択して、HP Tours を開きます。HP Web Tours の最初のページの [administration] リンクをクリックします。管理ページで、[Set LOGIN form's action tag to an error page] というタイトルの3番目のチェックボックスをクリアします。ページの最下部までスクロールし、[Update] をクリックします。ブラウザを閉じます。

自動相関

この練習では、ユーザ・アクションの**記録後**に、スクリプトを対象に相関をスキャンしました。

VuGen には、**記録セッション中**に動的な値を処理する、設定可能な相関ルールもあります。自動相関の詳細については、『**HP Virtual User Generator User's Guide**』（英語版）を参照してください。

ほとんどのサーバには、リンクおよび参照の作成時に必ず使用される、明確な構文ルール、つまり「コンテキスト」があります。サポートされているアプリケーション・サーバでセッションを記録する場合は、VuGen の組み込み相関ルールが使用できます。VuGen は、記録中に動的な値を検出して相関させます。



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 6「負荷テスト用スクリプトの準備」に進む準備ができました。

練習 5・再生に関する一般的な問題の解決方法

練習 6

負荷テスト用スクリプトの準備

前の練習では、スクリプトがアプリケーションを正確にエミュレートしていることを確認しました。再生をリアルタイムで表示し、仮想ユーザが一般的なビジネス・プロセスを実行することを確認しました。

ただし、これはシングル・ユーザをエミュレートした場合です。多数のユーザが同時に作業している場合もアプリケーションは正常に動作するでしょうか。正常に動作する場合でも、アプリケーション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。

そこで、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、応答時間データを収集するように設定します。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・プロセスの測定方法
- ▶ 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
- ▶ テスト成功の確認方法

ビジネス・プロセスの測定方法

アプリケーションを配備するために準備するときは、ログインに要した時間や航空券の予約に要した時間など、特定のビジネス・プロセスの間隔を測定する必要があります。通常、これらのビジネス・プロセスはスクリプト内の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。LoadRunner では、測定する必要のある一連のアクションをトランザクションとして指定します。

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本来の条件を満たしているかどうかを調べます。

トランザクションはスクリプト内のどこにでも手作業で挿入できます。ユーザ・ステップをトランザクションとしてマークするには、トランザクションの最初のステップの前に「**トランザクション開始**」マーカを置き、最後のステップの後に「**トランザクション終了**」マーカを置きます。

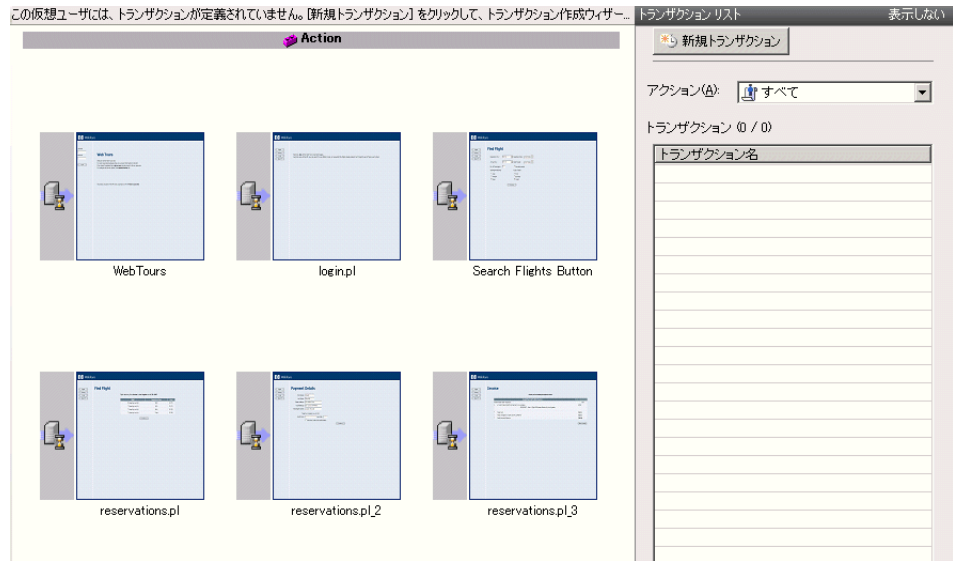
本項では、ユーザがフライトを検索して確定するまでに要する時間を計測するためのトランザクションを、スクリプトに挿入します。

練習 3 で作成したスクリプト **Basic_Tutorial** 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。

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 トランザクションの作成ウィザードを開きます。

[タスク] 表示枠が表示されていることを確認します（表示されていない場合は [タスク] ボタンをクリックします）。[タスク] 表示枠で、見出し [拡張] の下にある [トランザクション] をクリックします。トランザクションの作成ウィザードが開きます。



トランザクションの作成ウィザードには、スクリプトに含まれるさまざまなステップのサムネイルが表示されます。

[新規トランザクション] ボタンをクリックします。これで、トランザクション・マーカをドラッグしてスクリプト内の指定の位置に配置できるようになります。ここで、トランザクションの開始点を挿入するようウィザードから指示されます。

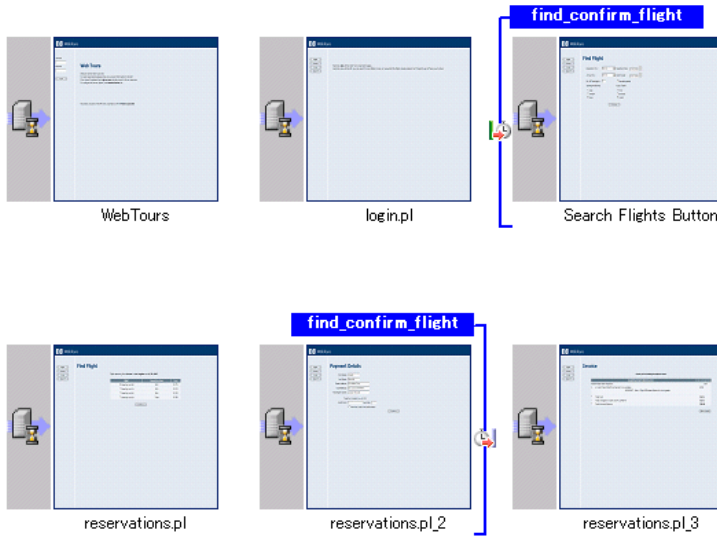
2 トランザクション開始マーカとトランザクション終了マーカを挿入します。

マウスを使用し、[Search Flights Button] というタイトルの 3 番目のサムネイルの前にマーカを置き、クリックします。次に、トランザクションの終了点を挿入するようウィザードから指示されます。

マウスを使用し、[reservations.pl_2] というタイトルの 5 番目のサムネイルの後にマーカを置き、クリックします。

3 トランザクションの名前を指定します。

トランザクションの名前を入力するようウィザードから指示されます。「find_confirm_flight」と入力し、Enter キーを押します。



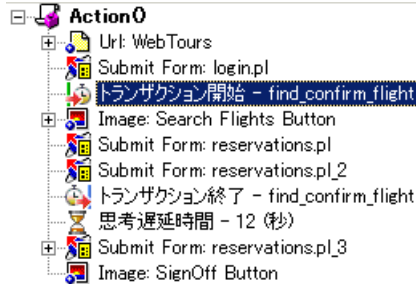
これで、新しいトランザクションが作成されました。トランザクションの開始点または終了点を調整するには、各マーカをスクリプト内の別の位置にドラッグします。トランザクションの名前を変更することもできます。これには、トランザクション開始マーカの上にある既存の名前をクリックし、新しい名前を入力します。

4 ツリー・ビューでトランザクションを観察します。



[表示] > [ツリー ビュー] を選択するか、ツールバーの [ツリー ビュー] ボタンをクリックして、ツリー・ビューに移動します。

トランザクション開始マーカとトランザクション終了マーカが、それらを挿入した正確な位置に新しいステップとして追加されたことを確かめます。



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。しかし、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。

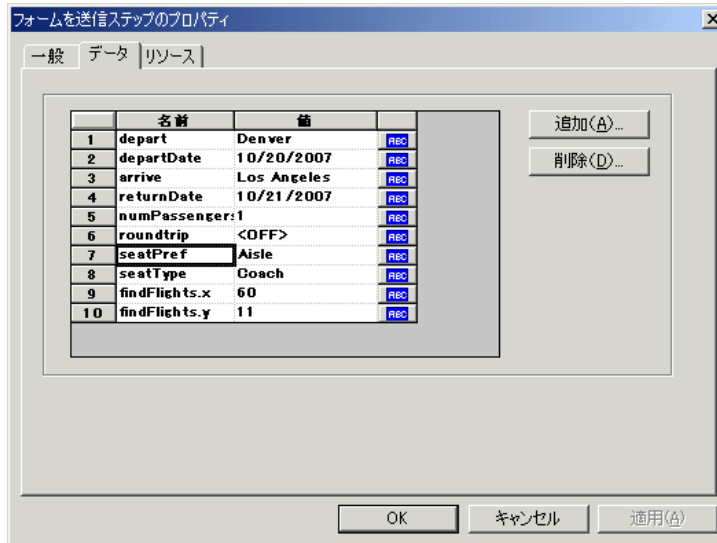
これを実現するには、スクリプトを「パラメータ化」します。つまり、記録された値 **Aisle**（通路側）を受け取り、それをパラメータに置き換えます。パラメータの値はパラメータ・ファイルに記述します。スクリプトを実行すると、仮想ユーザは値（通路側、ウィンドウ、または指定なし）をパラメータ・ファイルから受け取ります。それによって、実際の旅行会社における環境がエミュレートされます。

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。

[表示] > [ツリー ビュー] を選択してツリー・ビューに移動します。

テスト・ツリーで、[Submit Form : reservations.pl] ステップをダブルクリックします。[データ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。



右側のカラムにある [ABC] アイコンによって、引数が定数であることが示されます。

2 固定値が可変値になることを指定します。

7行目 **Aisle** にある **seatPref** の値を選択します。

Aisle 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。

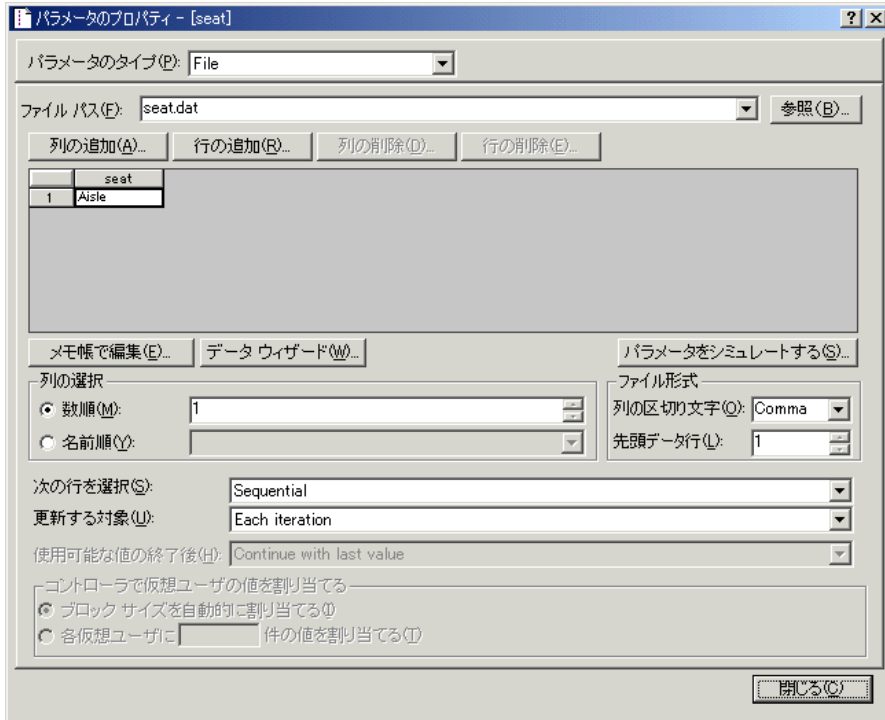


3 パラメータを作成します。

File パラメータ・タイプを使用して、パラメータ名 **seat** を指定します。[OK] をクリックします。VuGen によって、**ABC** アイコンがパラメータ・アイコンに置き換えられます。



- 4 **{seat}** の横にあるパラメータ・アイコンをクリックします。ポップアップ・メニューから **[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。



- 5 いくつかのサンプル値を指定してデータを変更します。

[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。単語「値」を「Window」に置き換えます。

[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。単語「値」を「None」に置き換えます。

値の大文字と小文字は区別されません。

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。

6 テストでのデータの変更方法を定義します。

標準設定を維持し、反復ごとに乱数値ではなく連続した値をとるよう VuGen を設定します。

[次の行を選択] : [Sequential] (連続)

[更新する対象] : [Each iteration] (反復ごと)

7 [閉じる] をクリックして [パラメータのプロパティ] ダイアログ・ボックスを閉じ、[OK] をクリックして [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。1 回目の反復では仮想ユーザは **Aisle** を選択し、2 回目の反復では仮想ユーザは **Window** を選択したことがわかります。

Web ページの内容の確認方法

テストを実行するときに、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ることがあります。「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。挿入できる内容チェックには、テキスト・チェックと画像チェックの 2 種類があります。

- ▶ 「テキスト・チェック」は、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」は、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。

テキストの検索

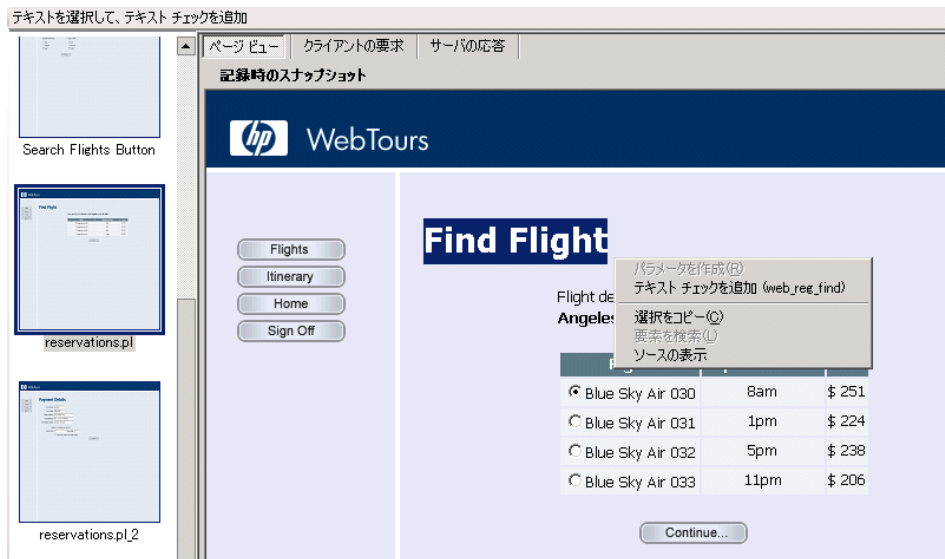
本項では、スクリプトの予約ページに表示される「Find Flight」という語句を検証するテキスト・チェックを追加します。

テキスト・チェックを挿入するには、次の手順を実行します。

1 コンテンツ・チェック・ウィザードを開きます。

[タスク] 表示枠が表示されていることを確認します（表示されていない場合は [タスク] ボタンをクリックします）。[タスク] 表示枠で、見出し [拡張] の下にある [コンテンツ チェック] をクリックします。

コンテンツ・チェック・ウィザードには、スクリプトにある各ステップのサムネイルが表示されます。



右側の表示枠で [ページ ビュー] タブを選択し、サムネイルのスナップショットを表示します。

2 検証するテキストのあるページを選択します。

[reservations.pl] というタイトルの 4 番目のサムネイルをクリックします。

3 検証するテキストを選択します。

スナップショットの中にある単語 Find Flight を強調表示します。単語を選択した状態で右クリックし、[テキスト チェックを追加 (web-reg-find)] を選択します。

[テキストの検索] ダイアログ・ボックスが開き、[特定のテキストを検索する] ボックスで選択したテキストが表示されます。[OK] をクリックします。

4 新しいステップを表示します。

ツリー・ビュー ([表示] > [ツリー ビュー]) で、VuGen によって新しいステップ **Service:Reg Find** がスクリプトに挿入されたことが示されます。このステップによってテキスト・チェックが登録されます。LoadRunner はステップの実行後にテキストを検証します。再生中、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。

画像の検索

本項では、ユーザのログオフ後に画像 **webtours.png** がページ上に表示されていることを検証する画像チェックを追加します。

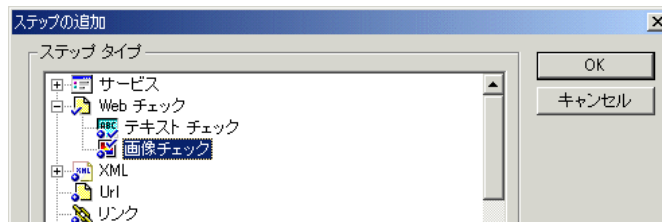
画像チェックを挿入するには、次の手順を実行します。

- 1 [表示] > [ツリー ビュー] を選択してツリー・ビューに戻ります。
- 2 検証する画像のあるページを選択します。

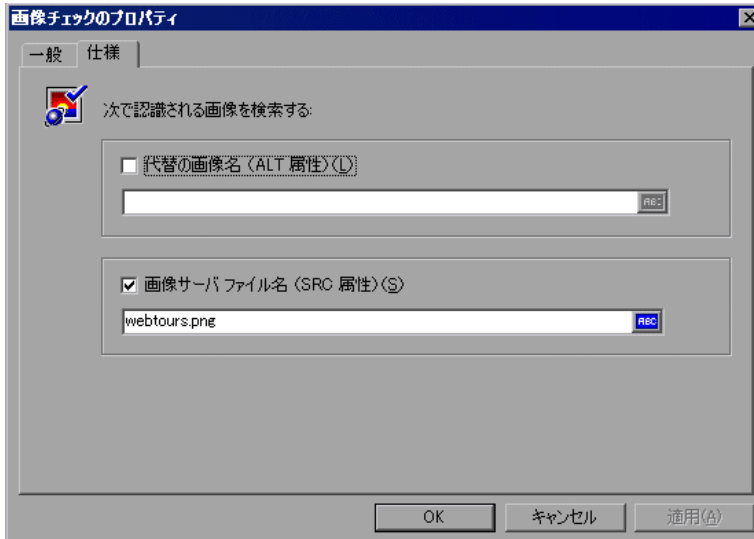
Image: SignOff Button ステップを選択します。右側の表示枠で [ページ ビュー] タブを選択し、ステップのスナップショットを表示します。

3 画像チェック・ステップを挿入します。

[挿入] > [新規ステップ] を選択します。[ステップの追加] ダイアログ・ボックスが開きます。



[Web チェック] を展開し、[画像チェック] を選択します。[OK] をクリックします。[画像チェックのプロパティ] ダイアログ・ボックスが表示されます。



4 画像を指定します。

[仕様] タブで、[画像サーバファイル名] オプションを選択し、画像の名前 `fma-gateway.jpg` をエディット・ボックスに入力します。

[OK] をクリックします。Image Check ステップが Image: SignOff Button ステップのサブステップとして挿入されます。



5 スクリプトを保存します。

再生中、画像 `webtours.png` 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、メッセージを出力に送信し、位置その他の情報を示す必要があることがあります。これらの出力メッセージは再生ログと 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。この例については、**オンライン関数リファレンス**を参照してください。

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。

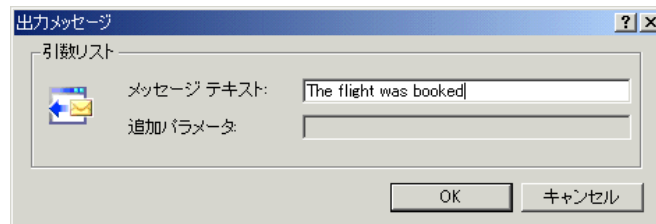
1 位置を選択します。

最後のステップ **Image: SignOff Button** を選択します。スナップショットが右側に表示されます。

2 出力メッセージを挿入します。

[**挿入**] > [**新規ステップ**] を選択します。[ステップの追加] ダイアログ・ボックスが開きます。下にスクロールして、[**出力メッセージ**] を選択し、[**OK**] をクリックします。

[出力メッセージ] ダイアログ・ボックスが開きます。



3 メッセージを入力します。

「The flight was booked」（フライトが予約されました）というメッセージを [**メッセージ テキスト**] ボックスに入力し、[**OK**] をクリックします。[出力メッセージ] がツリーに追加されます。



4 スクリプトを保存します。

注：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、[ステップの追加] ダイアログ・ボックスで、[**出力メッセージ**] ではなく [**エラー メッセージ**] を選択します。

テスト成功の確認方法

本項では、拡張スクリプトを実行して再生ログを表示し、テキスト・チェックと画像チェックを探します。テキスト・チェック、画像チェック、トランザクション、およびパラメータ化を表示します。

画像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 画像チェックを有効にします。

[実行環境設定] ダイアログ・ボックス ([**仮想ユーザ**] > [**実行環境の設定**]) を開き、[**インターネットプロトコル:プリファレンス**] ノードを選択します。[**画像とテキストチェックを有効にする**] オプションを選択します。[**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。



[実行] ボタンをクリックするか、[**仮想ユーザ**] > [**実行**] を選択します。VuGen によってスクリプトの実行が開始され、再生ログが [出力] ウィンドウに作成されます。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。

出力ウィンドウが開いていることを確認します ([**表示**] > [**出力ウィンドウ**])。[**再生ログ**] タブをクリックし、Ctrl+F キーを押して [検索] ダイアログ・ボックスを開きます。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 のレジストリ登録を完了しました

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・チェックのために VuGen を準備しているだけです。

もう一度検索を実行し (F3 キー)、web_reg_find の次のインスタンスを探します。このインスタンスは次のようになっています。

web_reg_find を "Text=Find Flight" を対象にレジストリ登録しました

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 ページを変更して単語 **Welcome** 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 画像チェックを探します。

Ctrl+F キーを押し、`web_image_check` を検索します。検索結果は次のようになります。

"web_image_check" の成功を 1 件検出しました。Alt="", Src="webtours.png"

これは、画像が見つかったことを示します。誰かが Web ページを変更して画像を削除した場合、以降の実行では、その画像が見つからなかったことが出力に示されます。

5 トランザクションの開始を探します。

再生ログをクリックし、Ctrl+F キーを押して [検索] ダイアログ・ボックスを開きます。単語「トランザクション」を探します。

6 パラメータ置換を表示します。

再生ログをクリックし、Ctrl+F キーを押して [検索] ダイアログ・ボックスを開きます。単語「パラメータ」を探します。ログに "seat" = "Aisle" という通知があります。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。



7 [ファイル] > [保存] を選択するか、[作業中のシナリオを保存] 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 7「負荷テスト・シナリオの作成」に進みます。

練習 6・負荷テスト用スクリプトの準備

練習 7

負荷テスト・シナリオの作成

前の練習では、Virtual User Generator でテストを検証することができました。この練習では、アプリケーションを負荷のかかった状態でテストします。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、負荷のかかった状態でシステムの動作を観察します。このテストを作成して実行するには、LoadRunner Controller を使用します。

この練習は、次の項目で構成されています。

- ▶ LoadRunner Controller の紹介
- ▶ 負荷テストに含めるユーザの組み合わせ
- ▶ Controller ウィンドウの概要
- ▶ 大きな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をエミュレート方法
- ▶ 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アプリケーションをテストすることです。このような作業条件の例としては、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航空券を予約する操作があります。

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大きな負荷を生成できることと、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ふるまいをエミュレートする必要もあります。たとえば、ユーザはアプリケーションのパフォーマンスを表示する際、Internet Explorer ではなく Netscape を使用することや、モデム、DSL、ケーブルなどの別のネットワーク接続を使用することが考えられます。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。

シナリオ・オブジェクト

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、システムから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。

Controller の開始

シナリオの作成を開始するには、Controller を起動し、新しいシナリオを作成します。

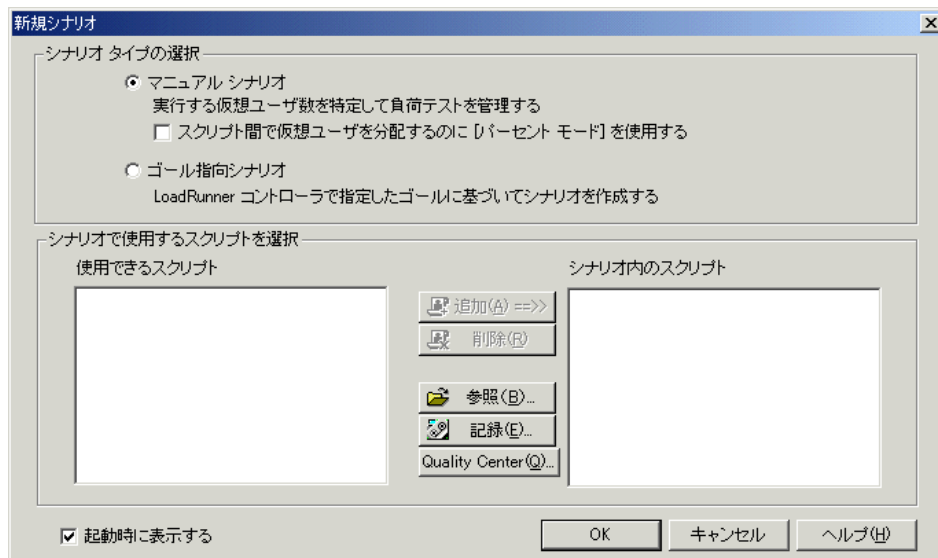
1 HP LoadRunner を起動します。

[スタート] > [プログラム] > [LoadRunner] > [LoadRunner] を選択します。HP LoadRunner Launcher ウィンドウが開きます。

2 Controller を起動します。

[**負荷テスト**] タブで、[**負荷テストの実行**] をクリックします。LoadRunner Controller が起動します。

標準設定では、[LoadRunner コントローラ] ウィンドウが表示され、[新規シナリオ] ダイアログ・ボックスが表示されます。



3 シナリオのタイプを選択します。

[**マニュアル シナリオ**] を選択します。

「**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。

「**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、ゴールを決定します。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。ゴール指向シナリオは練習 9「高度なゴール指向シナリオ」で作成します。

負荷テストに含めるユーザの組み合わせ

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・プロファイルの組み合わせを使用して実際のシナリオをエミュレートする場合は、異なるグループを作成し、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、航空券の購入、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人のユーザを追加します。

1 スクリプトを負荷テストに追加します。

本チュートリアルのために、前の手順で作成したスクリプトとほぼ同じ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。

[参照] ボタンをクリックし、
<LoadRunner のインストール先ディレクトリ>¥Tutorial ディレクトリにある **basic_script** に移動します。

[使用できるスクリプト] セクションおよび [シナリオ内のスクリプト] セクションにスクリプトが表示されます。

[OK] をクリックします。LoadRunner Controller の [デザイン] タブでシナリオが開きます。

2 負荷テスト・シナリオのデザインを開始します。

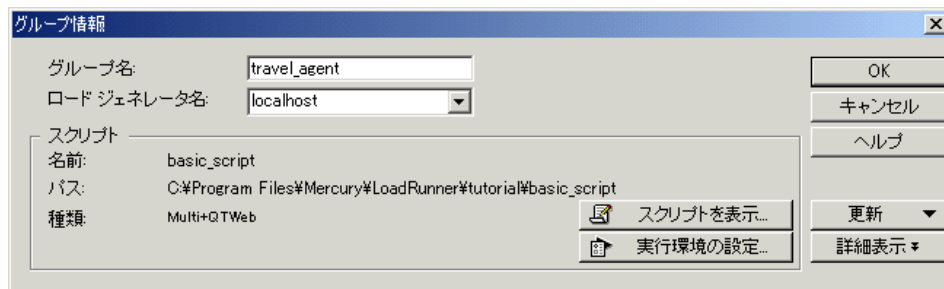
basic_script が [シナリオ グループ] ウィンドウの [グループ名] カラムに表示されていることを確認します。

3 グループ名と仮想ユーザ数を変更します。



[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。

[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` などと入力します。



[OK] をクリックします。これで、Controller ウィンドウの [シナリオ グループ] セクションに名前が表示されます。

Controller ウィンドウの概要

Controller ウィンドウの [デザイン] タブには次の 3 つの主要なセクションがあります。

- ▶ シナリオのスケジュール
- ▶ シナリオ・グループ
- ▶ サービス レベル アグリーメント

[シナリオ グループ] セクション (マニュアル・シナリオ)

SLA

[シナリオのスケジュール] セクション (マニュアル・シナリオ)

[デザイン] タブ

グループ名	スクリプトのパス	数	ロード ジェネレータ
basic_script	C:\Program Files\Mercury\LoadRunner\tutorial\basic_script	10	localhost

シナリオのスケジュール

スケジュール名: 対話式スケジュール

スケジュール設定: シナリオ グループ

期限: 完了まで実行 実行モード

全種スケジュール

- 合計: 10 仮想ユーザ
- アクション: プロパティ
- 初期化: 各仮想ユーザを実行直前に初期化する
- 仮想ユーザ開始: 10 仮想ユーザを 2 ずつ 00:00:16 (HH:MM:SS) ごとにロードする
- 実行時間: 00:06:00 (HH:MM:SS) の間実行する
- 仮想ユーザの停止: すべて 仮想ユーザを 5 ずつ 00:00:30 (HH:MM:SS) ごとに停止する

対話式スケジュールグラフ

Y軸: Users (0-12)

X軸: 時間 (00:00:00 - 00:07:00)

凡例: 全体スケジュール

[シナリオのスケジュール] : [シナリオのスケジュール] セクションでは、ユーザの振る舞いを正確に演じる負荷動作を設定します。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、および負荷の停止方法を決めます。

[シナリオ・グループ] : [シナリオ グループ] セクションでは、仮想ユーザ・グループを設定します。システムの標準的なユーザを表すさまざまなグループを作成します。実行するアクション、実行する仮想ユーザの数、実行するマシンを定義します。

[サービス レベル アグリーメント] : 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、つまり**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。

大きな負荷を生成する方法

スクリプトを追加し、シナリオの中で実行する必要がある仮想ユーザの数を決めた後、Load Generator マシンを設定できます。

Load Generator は、仮想ユーザを実行することによってアプリケーションに負荷を生成するマシンです。複数の Load Generator マシンを使用し、それぞれのマシンで多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。



[**Load Generator**] ボタンをクリックします。[ロード ジェネレータ] ダイアログ・ボックスが開き、**localhost** Load Generator マシンの詳細が表示されます。



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（標準でシナリオの中に含まれています）。**localhost** Load Generator のステータスは「**ダウン**」です。これは、Controller が Load Generator に接続されていないことを示します。

注：標準的な実稼動環境では、Load Generator マシンが複数あり、それぞれが多数の仮想ユーザのホストになります。マシンを追加するには、[ロードジェネレータ] ダイアログ・ボックスで [追加] ボタンをクリックし、マシン名とプラットフォーム・タイプを [新規ロードジェネレータの追加] ダイアログ・ボックスで入力します。

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。しかし、シナリオの実行を試みる前に接続をテストできます。

localhost Load Generator を選択し、[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、ステータスが「**ダウン**」から「**準備中**」に変わります。

[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 マシン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、負荷テストの期間やシナリオの停止方法も決めることができます。

Controller ウィンドウの [シナリオのスケジュール] 領域でマニュアル・シナリオの負荷の動作を設定できます。[シナリオのスケジュール] 領域は、スケジュール定義表示枠とアクション・スケジュール・グリッドの2つの区画に分かれています。

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。

スケジュール定義表示枠の中で、[スケジュール設定期限：シナリオ]と[実行モード：実スケジュール]を選択します。

スケジュール名: スケジュール1

スケジュール設定
期限: シナリオ グループ

実行モード: 実スケジュール 完了まで実行する

2 スケジュール・アクションの定義を設定します。

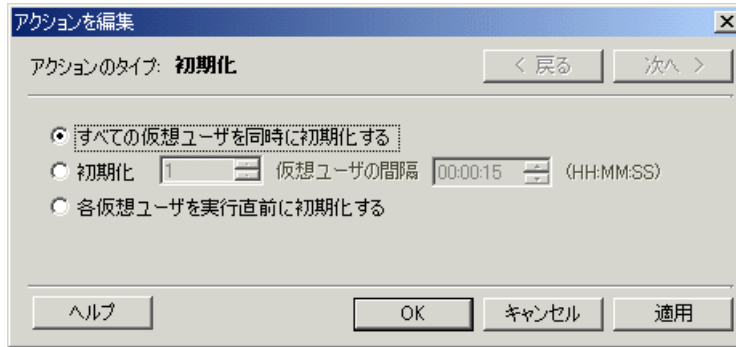
[アクション] グリッドが次のように見えるように、後述の設定を行います。

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	各仮想ユーザを実行直前に初期化する
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
仮想ユーザの停止	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、仮想ユーザと Load Generator を負荷テストの実行のために準備することを意味します。実行を開始する前に仮想ユーザを初期化することで、CPU の消費量が減り、より現実に近い結果が得られるようになります。

[アクション] 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。[すべての仮想ユーザを同時に初期化する] を選択します。

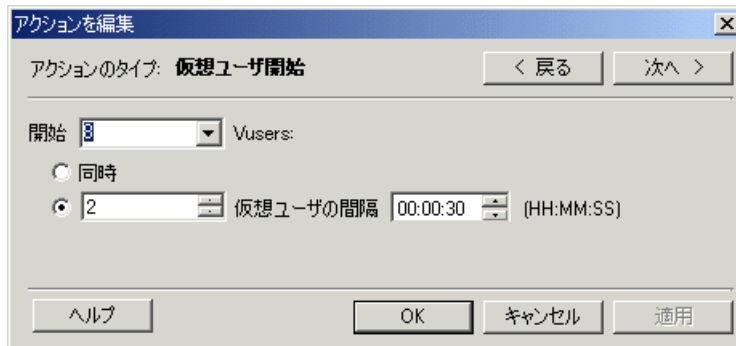


b) 段階的な起動を指定します。

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。

[アクション] 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。

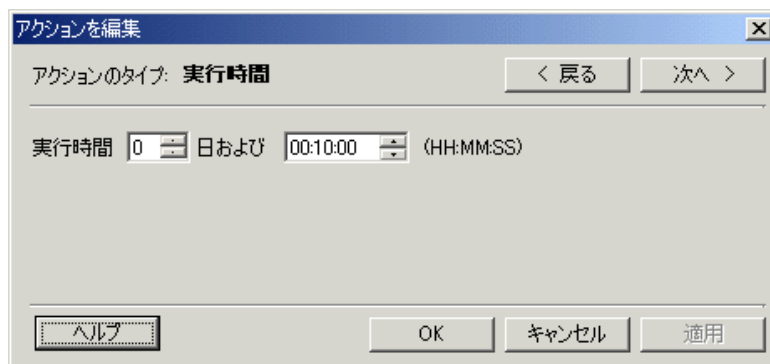
[開始 X Vusers] ボックスに、8 仮想ユーザを指定し、2 番目の選択肢「2 仮想ユーザの間隔 00:00:30」を選択します。



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。

[アクション] グリッドの中で [**実行時間**] 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、「実行時間」アクションが表示されます。10 分間実行するように設定されていることを確認します。

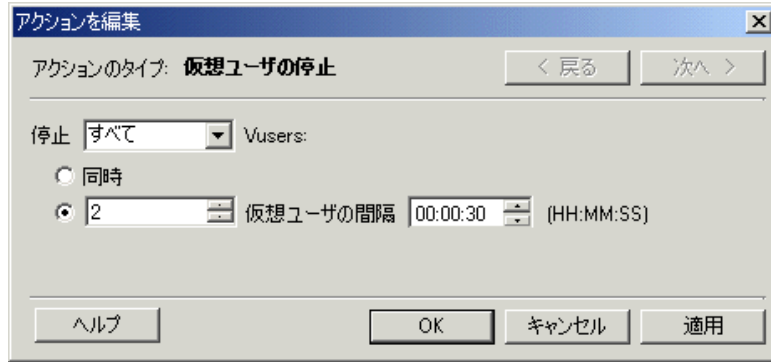


d) 段階的に終了するようスケジュールを設定します。

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。

練習 7・負荷テスト・シナリオの作成

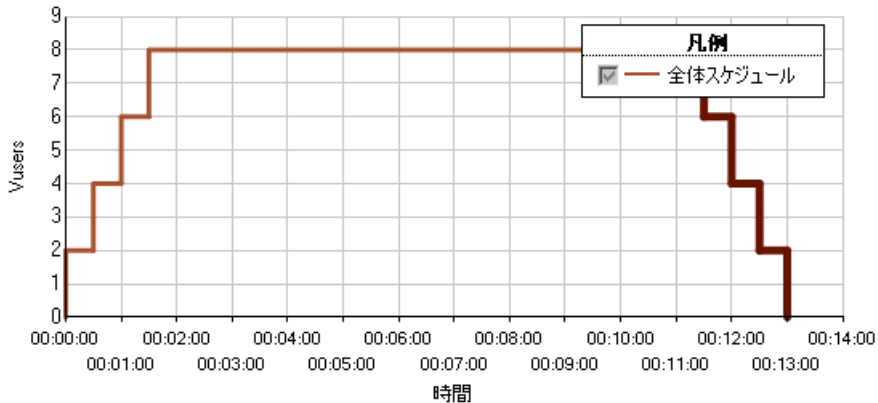
[シナリオ スケジュール] グリッドの中で [仮想ユーザの停止] をダブルクリックします。[アクションを編集] ダイアログ・ボックスが開き、「仮想ユーザの停止」アクションが表示されます。2 つめの選択肢、つまり「2 人の仮想ユーザを 30 秒ごとに停止する」を選択します。



3 スケジューラをグラフ形式で表示します。

[対話式スケジュール グラフ] のグラフは、シナリオ・スケジュールの「仮想ユーザ開始」、「継続時間」および「仮想ユーザ停止」のそれぞれのアクションを示します。このグラフの特徴の 1 つは、それが対話型であること、つまりグラフの線をドラッグすることで任意の設定を変更できることです。

対話式スケジュール グラフ



さまざまなタイプのユーザをエミュレート方法

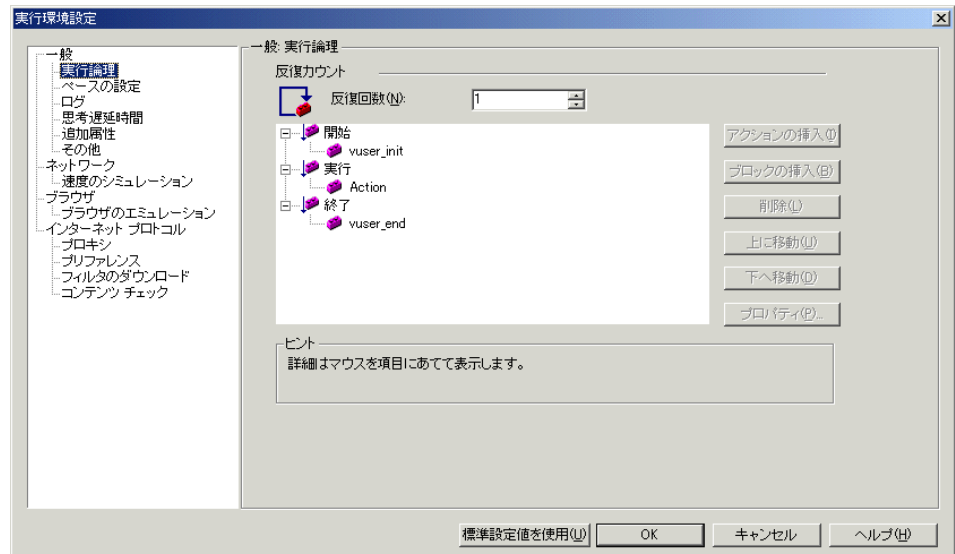
負荷スケジュールの設定が完了したら、テスト中の仮想ユーザの振る舞いを指定します。

実際のユーザをエミュレートするときは、ユーザの実際の振る舞いを考慮する必要があります。振る舞いには、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。

[デザイン] タブをクリックし、スクリプトを選択して、[実行環境の設定] ボタンをクリックします。[実行環境設定] ダイアログ・ボックスが表示されます。



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。次の設定があります。

- ▶ **[実行論理]** : 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]** : テスト中に収集する必要のある情報のレベル。

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]**：ユーザがステップの間に思考する時間。

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、ケーブルなどの別の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表示するユーザ。
- ▶ **[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明確に特定できます。

2 思考遅延時間を有効にします。

[一般：思考遅延時間] ノードを選択します。[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] オプションを選択します。最小値として 50% を指定し、最大値として 150% を指定します。

一般: 思考遅延時間

思考遅延時間のオプション

思考遅延時間を無視する(O)

思考遅延時間を再生する(R)

思考遅延時間を記録通りに再生する(S)

記録された思考遅延時間に掛ける倍数(M):

記録された思考遅延時間の乱数率を使用する(E)

最低(M): % 最高(X): %

思考遅延時間を(L): 秒に制限する

記録された思考遅延時間の乱数率は、多様な習熟度に基づいてユーザをエミュレートする場合に使用します。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、ランダムな思考遅延時間は 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。

[一般：ログ] ノードを選択し、[ログを有効にする] を選択します。[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。[拡張ログ] を選択し、[サーバから返されたデータ] を選択します。

一般: ログ

ログを有効にする(E)

ログ オプション

エラー発生時のみメッセージを送信する(M)

常にメッセージを送信する(L)

次の詳細レベルでメッセージをログに記録する

標準ログ(S)

拡張ログ(X)

パラメータ置換(P)

サーバから返されたデータ(D)

詳細トレース(T)

注：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だけの目的で有効にしています。

[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。

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振る舞いの設定が完了したら、モニタを設定します。

アプリケーションに大きい負荷をかけている間、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。

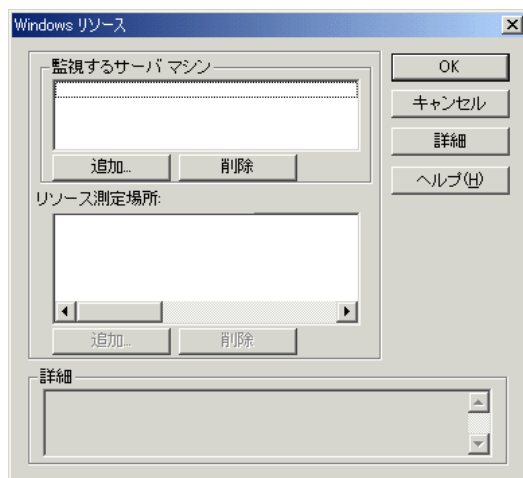
1 Windows リソース・モニタを選択します。

[コントローラ] ウィンドウで **[実行]** タブをクリックし, [実行] ビューを開きます。

[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。他のグラフを開く方法については次の練習で学びます。

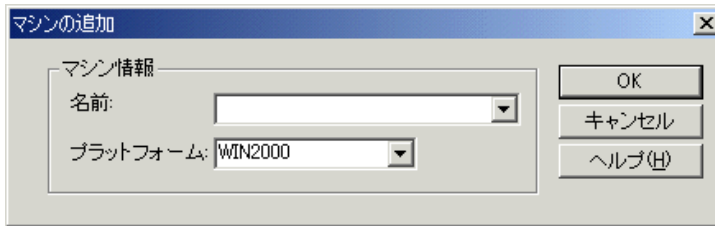


[Windows リソース] グラフを右クリックし, **[測定値の追加]** を選択します。
[Windows リソース] ダイアログ・ボックスが開きます。



2 監視対象のサーバを選択します。

[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。



[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)。[プラットフォーム] ボックスに、マシンを実行するプラットフォームを入力します。[OK] をクリックします。

標準の Windows リソース測定項目が [リソースの測定場所: <サーバマシン>] 表示枠に表示されます。



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、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8「負荷テストの実行」に進みます。

練習 7・負荷テスト・シナリオの作成

練習 8

負荷テストの実行

テストを実行すると、LoadRunner により、アプリケーション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、アプリケーションのパフォーマンスを実際の条件のもとで観察できます。

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
- ▶ 負荷下のアプリケーションの監視方法
- ▶ 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対処方法
- ▶ 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

[コントローラ] ウィンドウの [実行] タブは、テスト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] ビューには 5 つの主要セクションがあります。

- ▶ シナリオ・グループ
- ▶ シナリオ・ステータス
- ▶ 利用可能なグラフ・ツリー
- ▶ グラフ表示領域
- ▶ グラフの凡例

The screenshot displays the HP LoadRunner Controller interface in the [実行] (Execution) view. The window title is "HP LoadRunner コントローラ - Scenario1 - [実行]". The menu bar includes options like "ファイル(F)", "表示(V)", "シナリオ(S)", "モニタ(M)", "結果(R)", "診断(D)", "ツール(L)", and "ヘルプ(H)".

The interface is divided into several key sections:

- シナリオ グループ (Scenario Group):** A table showing the status of various scenarios. A box highlights this section.
- シナリオ ステータス (Scenario Status):** A panel on the right showing details for the selected scenario, such as "実行中の仮想ユーザ" (8), "経過時間" (00:01:53), and "成功したトランザクション" (24). A box highlights this section.
- 利用可能なグラフ (Available Graphs):** A tree view on the left listing various performance metrics like "合計トランザクション/秒 (成功)", "Web リソース グラフ", and "秒ごとのヒット数". A box highlights this section.
- グラフ表示領域 (Graph Display Area):** The central area contains several line graphs showing performance metrics over time. A box highlights this section.
- グラフの凡例 (Graph Legend):** A table at the bottom provides a legend for the graphs, listing metrics, their units, and the machines being monitored. A box highlights this section.

The legend table at the bottom is as follows:

色	倍率	測定値	マシン	最高	最低	平均	標準	最終
紫	0.1	% Processor Time (Processor_Total)	localhost	100.000	98.958	99.939	0.245	100.000
緑	0.01	File Data Operations/sec (System)	localhost	12371.901	6940.664	8453.363	1091.415	8056.076
赤	1	Processor Queue Length (System)	localhost	310	6.000	7.882	0.832	8.000
黄	0.1	Page Faults/sec (Memory)	localhost	4835	297.125	548.247	641.529	346.571
青	10	% Disk Time (PhysicalDisk_Total)	localhost	310	0.923	4.111	5.031	3.817
水	1E-6	Pool Nonpaged Bytes (Memory)	localhost	7920.000	14397440.000	14413101.180	4031.717	14413824.000
茶	10	Page/sec (Memory)	localhost	54.716	0.933	3.845	12.716	0.666

[シナリオ・グループ]：左上の表示枠には、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。この表示枠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、仮想ユーザを手作業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。

[シナリオ ステータス]：右上の表示枠で、負荷テストのサマリを表示できます。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を表示できます。

[利用可能なグラフ] ツリー：左側の中央の表示枠には、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、そのグラフをグラフ表示領域にドラッグします。

[グラフ表示領域]：右側の中央の表示枠では、1～8個のグラフが表示されるようにカスタマイズできます（[表示] > [グラフの表示]）。

[グラフの凡例]：下部の表示枠には、選択したグラフのデータが表示されます。

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。

画面下部の [実行] タブを選択します。

[シナリオ グループ] 領域の [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	実行	ソープ	成功	失敗	エラー	徐々に終了	終了中	中止
1	8	0	0	0	0	0	0	0	0	0	0	0
travel_agent	8											

シナリオはまだ実行されていないため、他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。

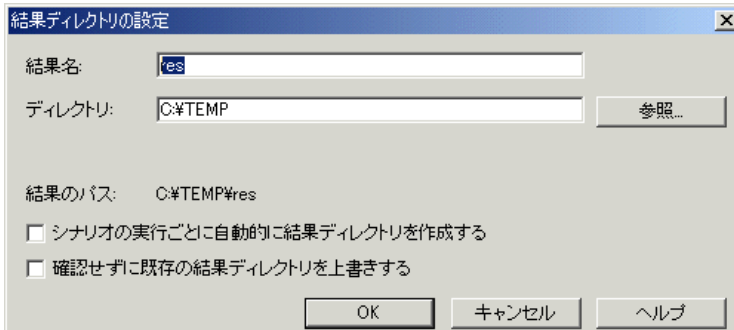


[シナリオ開始] ボタンをクリックするか, [シナリオ] > [開始] を選択して, テストの実行を開始します。

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 の一時ディレクトリ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結果は, それ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。



新しい結果ディレクトリを指定します。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ことができるように, 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。

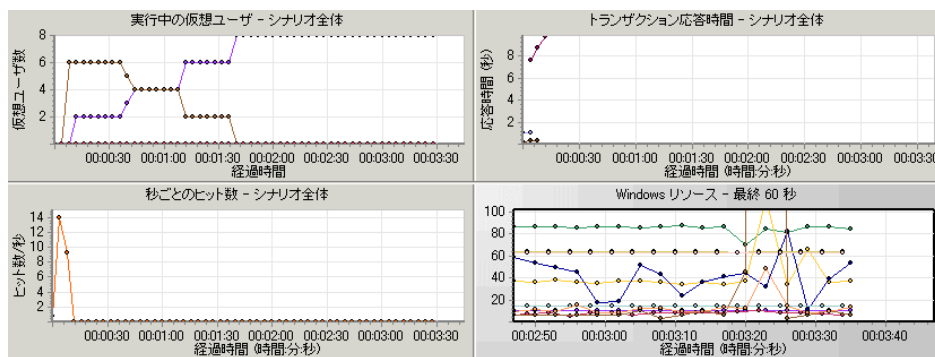
[実行] タブには次の標準のオンライン・グラフが表示されます。

[**実行中の仮想ユーザ - シナリオ全体**] グラフには、特定の時間に実行されている仮想ユーザの数が表示されます。

[**トランザクション応答時間 - シナリオ全体**] グラフには、各トランザクションが完了するのに要した時間が表示されます。

[**秒ごとのヒット数 - シナリオ全体**] グラフには、シナリオ実行の経過秒ごとに仮想ユーザが Web サーバに対して行ったヒット (HTTP 要求) の数が表示されます。

[**Windows リソース**] グラフには、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。

[Windows リソース] グラフを拡大するには、グラフ表示枠をダブルクリックします。各測定値は、[グラフの凡例] 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。

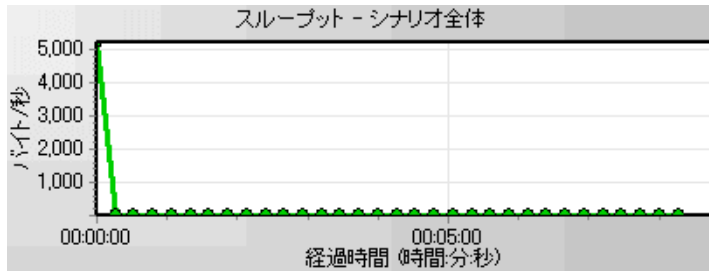
行を選択すると、対応するグラフの線が強調表示されます。また、その逆の場合も同様です。グラフを縮小するには、グラフをもう一度ダブルクリックします。

3 スループット情報を表示します。

[利用可能なグラフ] ツリー内で [スループット] グラフを選択し、グラフ表示領域までドラッグします。[スループット] グラフの測定値が表示ウィンドウとグラフの凡例に表示されます。

[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。



色	倍率	Web リソース	最高	最低	平均	標準	最終
	1	スループット	該当なし	該当なし	15346.383	該当なし	75085.800

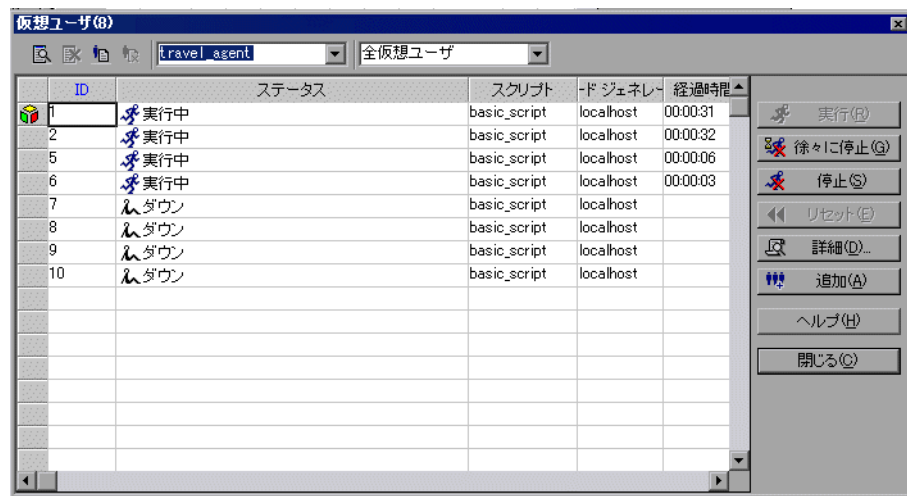
リアルタイムで実行しているユーザの監視方法

ユーザをエミュレートするときは、ユーザのアクションをリアルタイムで表示し、各ユーザが正しいステップを実行していることを確かめることができる必要があります。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 **[仮想ユーザ]** ボタンをクリックします。[仮想ユーザ] ウィンドウが開きます。



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。スケジュールの「仮想ユーザ開始」アクションによって、Controller で一度に 2 個の仮想ユーザが解放されました。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現在実行しているアクションが表示されます。記録されたシナリオのステップを仮想ユーザが一通り実行するにつれて、ウィンドウが更新されます。



- 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、実行時ビューアのログを閉じます。

ユーザ・アクションのサマリの表示場所

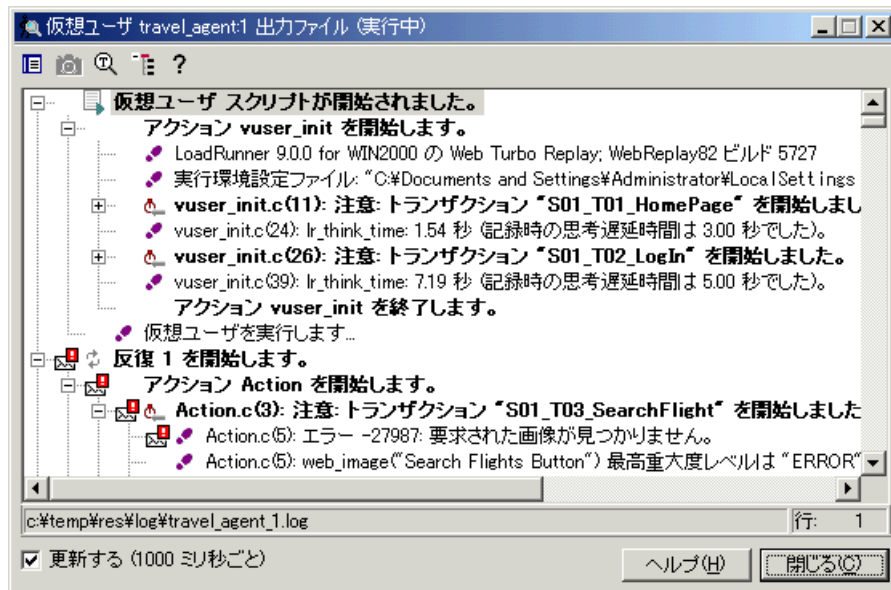
テスト実行中における個々の仮想ユーザの進行状況を検証するために、仮想ユーザのアクションのテキスト・サマリが格納されたログ・ファイルを表示できます。

イベントのテキスト・サマリを確認するには、次の手順を実行します。



[仮想ユーザ] ウィンドウで、実行中の仮想ユーザを選択し、**[仮想ユーザ ログを表示]** ボタンをクリックします。

[仮想ユーザ 出力ファイル] ウィンドウが開きます。



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、シナリオの開始を示しています。ログの最後までスクロールし、選択した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。


注： 仮想ユーザ・ログに情報が格納されるのは、[実行環境設定] ダイアログ・ボックスの [ログ] タブでログ機能を有効にした場合のみです。

[仮想ユーザ ログ] ウィンドウと [仮想ユーザ] ウィンドウを閉じます。

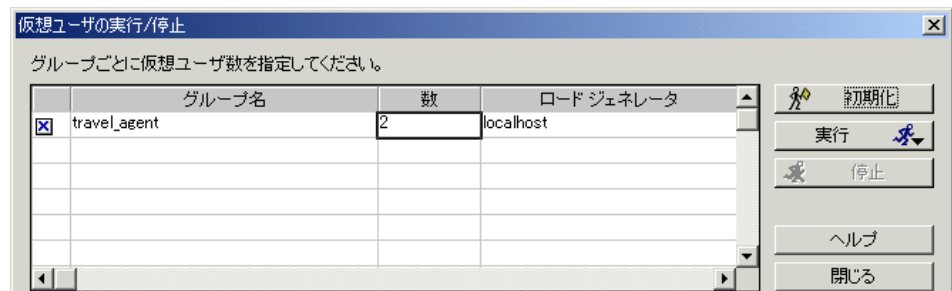
テスト中に負荷を増やす方法

より多くのユーザを手作業で追加することによって、負荷テストの実行中にアプリケーションにかかる負荷を増やすことができます。

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。

- 1  **[仮想ユーザの実行 / 停止]** ボタンをクリックします。[仮想ユーザの実行 / 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。
- 2 **[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。
- 3 **[実行]** をクリックして仮想ユーザを追加します。

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]** オプションが開きます。**[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] ウィンドウには、10 個の仮想ユーザが実行されていることが示されます。

注：LoadRunner Controller が追加された仮想ユーザを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・リソースが非常に限られていることが原因で起こります。このような問題を避けるため、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対処方法

[シナリオ ステータス] ウィンドウに実行中のシナリオの概要が表示されているか確認し、どの仮想ユーザ・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・ダウンして調べます。失敗トランザクションやエラーの多くは、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します。

1 テストのステータスを表示します。

[シナリオ ステータス] ウィンドウに、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	
実行中の仮想ユーザ	8	
経過時間	00:01:53 (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	24	🔍
失敗したトランザクション	60423	🔍
エラー	33155	🔍

2 仮想ユーザ・アクションのブレイクダウンを表示します。

[シナリオ ステータス] ウィンドウにある **[成功したトランザクション]** リンクをクリックし、トランザクションの詳細の一覧を表示します。[トランザクション] ダイアログ・ボックスが表示されます。



名前	TPS	成功	失敗	中止
S01_T02_LoggedIn	0.3	12	0	0
S01_T03_SearchFlight	0.0	0	11249	0
Action_Transaction	0.0	0	11249	0
vuser_init_Transaction	0.3	12	0	0
S01_T01_HomePage	0.3	12	0	0

閉じる(C) ヘルプ(H)

アプリケーションでエラーが発生したかどうか

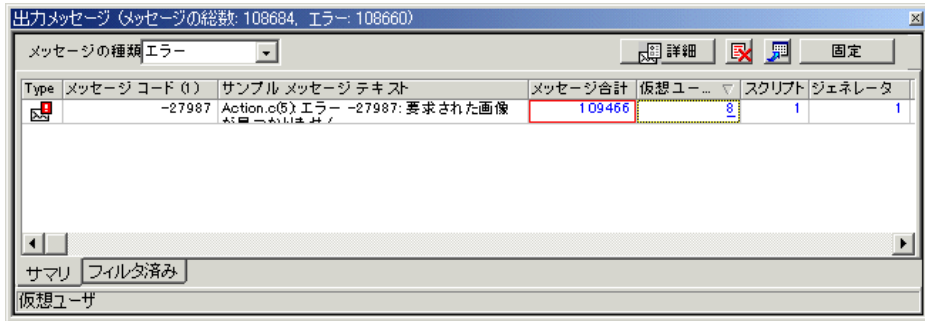
大きな負荷のかかった状態でアプリケーションが失敗し始める場合は、エラーや失敗トランザクションが発生した可能性があります。Controller はこれらのエラー・メッセージを [出力] ウィンドウに表示します。

1 エラー・メッセージがないか確認します。

[表示] > [出力メッセージを表示] をクリックするか、[シナリオ ステータス] ウィンドウにある **[エラー]** リンクをクリックします。

練習 8・負荷テストの実行

[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。

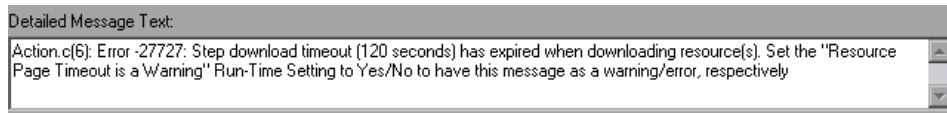


Type	メッセージコード (1)	サンプル メッセージ テキスト	メッセージ合計	仮想ユー...	スクリプト	ジェネレータ
	-27987	Action.c(6) エラー -27987: 要求された画像が見つかりません	109466		1	1



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] ボタンをクリックします。[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。

次の例はタイムアウト・エラーを示しています。Web サーバが、目標の時間間隔以内に要求に応答していません。



Detailed Message Text:
Action.c(6): Error -27727: Step download timeout (120 seconds) has expired when downloading resource(s). Set the "Resource Page Timeout is a Warning" Run-Time Setting to Yes/No to have this message as a warning/error, respectively

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、[メッセージ合計] 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。



[Line Number] 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す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。

テストの実行が完了したことを知る方法

テストの実行が完了すると、[シナリオステータス] ウィンドウに「ダウン」ステータスが表示されます。これは、仮想ユーザの実行が停止したことを示します。

[仮想ユーザ] ウィンドウで、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。LoadRunner には、仮想ユーザがタスクを繰り返した回数（反復）、成功した反復の数、および経過時間が表示されます。



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

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、トランザクションが顧客にとって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。トランザクションの応答時間が悪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 10：シナリオの分析」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、ボトルネック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、[保存] ボタン をクリックして、[ファイル名] ボックスにシナリオ名を入力します。



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 9「高度なゴール指向シナリオ」に進みます。

練習 9

高度なゴール指向シナリオ

前の2つの練習では、負荷テストを手動で作成して実行する方法を学習しました。ここでは、テストで達成するゴールを定義します。

アプリケーションを配備する前に受け入れテストを実行して、システムが予想される現実の作業負荷に耐えることを確認します。サーバのパフォーマンスの期待値は、秒ごとのヒットの数や秒ごとのトランザクション数などによって定義されています。この期待値は、アプリケーションの要件を定義するビジネス・アナリストによって決まるか、または実運用されている旧バージョンのアプリケーションまたはその他のソースから取得される可能性があります。生成する秒ごとのヒット数、秒ごとのトランザクション数、またはトランザクション応答時間のゴールを設定します。LoadRunner が、ゴール指向シナリオを使用して自動的に必要なゴールを生成します。アプリケーションが一定の負荷下にある間に、トランザクション応答時間を監視して顧客に提供されているサービス・レベルを確認できます。

この練習では、Web サーバ上での秒ごとのヒット数が3個で、最小5個、最大10個の仮想ユーザを使用してこの負荷レベルを5分間維持するゴール指向シナリオを作成します。この練習は、次の項目で構成されています。

- ▶ 使用するゴール・タイプ
- ▶ ゴール指向シナリオの作成方法
- ▶ Controller ウィンドウの概要（ゴール指向シナリオ）
- ▶ ゴールの定義方法
- ▶ 負荷の動作の確認方法
- ▶ 監視するオンライン・グラフ
- ▶ ゴール指向シナリオの実行方法
- ▶ ゴールの達成の確認

使用するゴール・タイプ

LoadRunner には、ゴール指向シナリオに次の 5 種類のゴール・タイプが用意されています。同時に実行する仮想ユーザ数、秒ごとのヒット数、秒ごとのトランザクション数、分ごとのページ数、シナリオの目標トランザクション応答時間。

- ▶ さまざまなビジネス・プロセスを実行できる仮想ユーザの総数がわかっている場合は、仮想ユーザのゴール・タイプを使用できます。
- ▶ サーバの強度がわかっている場合は、秒ごとのヒット数、分ごとのページ数、または秒ごとのトランザクション数のゴール・タイプを使用できます。
- ▶ トランザクションの完了に必要な応答時間がわかっている場合は、トランザクション応答時間のゴール・タイプを使用できます。たとえば、電子商取引サイトへのログインで顧客を 5 秒以上待たせないようにしたい場合には、最大許容トランザクション応答時間を 5 秒に指定し、実際に応答できた仮想ユーザ数を確認します。

ゴール指向シナリオの作成方法

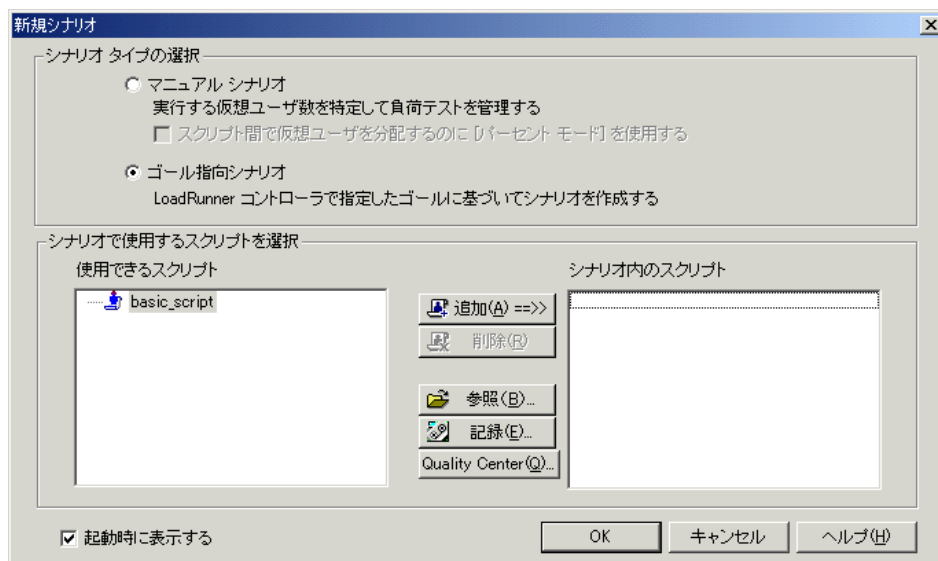
ユーザ・プロファイルの組み合わせを使用して現実のシステムをエミュレートするには、いくつかのスクリプトをシナリオに割り当てるとともに、スクリプトに負荷の割合を割り当てます。割合は期待負荷に従って設定する必要があります。

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。

1 シナリオを新規作成します。



[ファイル] > [新規作成] を選択するか、[新規作成] をクリックして、[新規シナリオ] ダイアログ・ボックスを開きます。



2 シナリオ・タイプを選択します。

[ゴール指向シナリオ] オプションを選択します。

3 スクリプトを選択します。



[使用できるスクリプト] リストから **basic_script** を選択し、[追加] ボタンをクリックします。スクリプトが [シナリオ内のスクリプト] 表示枠に表示されます。

[OK] をクリックします。LoadRunner Controller の [デザイン] ビューが開き、**basic_script** が [スクリプト名] カラムに表示されます。

Controller ウィンドウの概要（ゴール指向シナリオ）

Controller ウィンドウ（ゴール指向）の [デザイン] ビューには、次の 3 つの主要セクションがあります。

- ▶ シナリオ ゴール
- ▶ シナリオ スクリプト
- ▶ サービス レベル アグリーメント

[シナリオのスクリプト] セクション

SLA

[シナリオのゴール] セクション

[シナリオゴールの編集]

[デザイン] タブ

[シナリオのスクリプト]：表示枠の上部では、仮想ユーザ・スクリプト、各スクリプトのパス、各スクリプトに割り当てられた総目標の割合、および Load Generator マシンを確認します。シナリオの設定はここでを行います。

[シナリオのゴール]：表示枠の下部には、テスト目標、ゴールを達成するために使用するユーザ数、シナリオの継続時間、および負荷の動作が表示されます。[シナリオゴールの編集] ダイアログ・ボックスで、ゴールの設定を定義します。

[サービス レベル アグリーメント] : 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、つまり**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。

ゴールの定義方法

実行するスクリプトを選択したので、次は達成するゴールを定義する必要があります。

本項では、ゴール・プロファイルを作成し、シナリオ・ゴールを定義します。

1 [シナリオ ゴールの編集] ダイアログ・ボックスを開きます。



[シナリオ ゴールの編集] ボタンをクリックするか、[シナリオ] > [ゴールの定義] を選択します。[シナリオ ゴールの編集] ダイアログ・ボックスが表示されます。

The screenshot shows the 'シナリオ ゴールの編集' (Scenario Goal Edit) dialog box. It includes the following elements:

- Goal Profile Name:** Hits per Second 100
- Buttons:** 名前の変更(R), 削除(D), 新規作成(N), シナリオ開始時間...
- Scenario Goal Definition:**
 - Goal Type: 秒ごとのヒット数
 - Target: 100.00 秒ごとのヒット数
 - Next Use: 最小 50, 最大 150, 仮想ユーザ
- Scenario Settings / Load Action:**
 - Execution Time: 継続時間 00:03:00 (時間:分:秒) (目標の達成後)
 - Target Unreachable Cases:
 - シナリオを停止し、結果を保存する
 - ゴールに達成せずにシナリオを続行する
 - エラー・メッセージを送信する
- Load Preview Graph:** 負荷のプレビュー. The graph shows '秒ごとのヒット数' (Hits per second) on the y-axis (0 to 200) and '経過時間' (Elapsed time) on the x-axis (00:00 to 00:30). The line rises from 0 at 00:00 to 100 at 00:03 and then remains constant at 100.
- Footer:** 記録された思考遅延時間を変更しない | OK | キャンセル | ヘルプ

2 ゴール・プロファイルの論理名を指定します。

[新規作成] をクリックし、[新規ゴール プロファイル] ダイアログ・ボックスに新しいゴール・プロファイル名（たとえば「秒ごとのヒット数 3」）を入力し、[OK] をクリックします。

新しいゴール・プロファイル名がセレクトタに表示されます。

3 シナリオ・ゴールを定義します。

[ゴールのタイプ] ボックスで、[秒ごとのヒット数] を選択します。

[達成するゴール: X 秒ごとのヒット数] ボックスに、**3** を入力します。

4 実行する LoadRunner の最小および最大仮想ユーザ数の範囲を設定します。

仮想ユーザの最小数に **5**、仮想ユーザの最大数に **10** を入力します。この値は、サーバ上で同時にヒットを作成する旅行代理店の最小数と最大数の範囲に対応しています。

負荷の動作の確認方法

テストの目標を定義したので、次は Controller にいつどのように目標を達成させるかを指定する必要があります。

複数のユーザがまったく同じ時間にシステムにログオンまたはログオフすることはありません。実際のユーザをエミュレートするために、LoadRunner の [負荷の動作] タブには、システムに徐々にログオンおよびログオフする機能が用意されています。また、一定期間サーバに負荷をかけた状態にすることもできます。LoadRunner では、[シナリオの設定] タブでサーバに負荷をかける時間を指定できます。

負荷テストの動作を定義するには、次の手順を実行します。

- 1 [シナリオ ゴールの編集] ダイアログ・ボックスで、[ランプアップ] タブを選択して、[自動] を選択します。

ランプアップ

自動

次の時間をかけて秒ごとの目標ヒット数に到達: 00:10:00 (時間:分:秒)

ステップアップ: 20 秒ごとのヒット数

各 00:02:00 (時間:分:秒)

これにより、Controllerに必要な数の仮想ユーザを同時に実行するよう指示します。

- 2 [シナリオの設定] タブを選択します。

目標の達成後テストを継続する時間を5分と指定し、[ゴールを達成せずにシナリオを続行する] を選択します。

実行時間

継続時間 000:05:00 (時間:分:秒) (目標の達成後)

目標が達成できない場合

シナリオを停止し、結果を保存する

ゴールに達成せずにシナリオを続行する

エラーメッセージを送信する

秒ごとのヒット数が3個という負荷が達成すると、Controllerは必要に応じて仮想ユーザを追加または削除して、測定値を指定された目標の6%以内に維持しながら、さらに5分間シナリオの実行を継続します。これにより、サーバが一定期間負荷に耐えられることを確認します。

- 3 記録された思考遅延時間は使用しないよう設定します。

[シナリオ ゴールの編集] ダイアログ・ボックスの左下角で、[記録された思考遅延時間を変更しない] が無効であることを確認します。

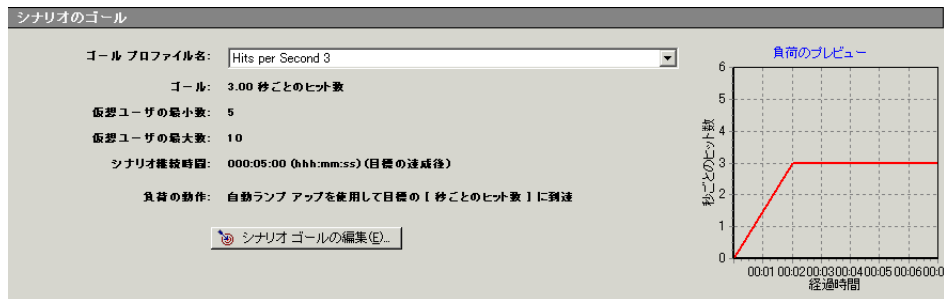
練習 9・高度なゴール指向シナリオ

このオプションを有効にする場合、LoadRunner はスクリプトで記録された思考遅延時間を使用してシナリオを実行します。また、目標達成のためにシナリオ中の仮想ユーザ数を増やさなければならないことがあります。

4 [シナリオ ゴールの編集] ダイアログ・ボックスを閉じます。

[OK] をクリックして、[シナリオ ゴールの編集] ダイアログ・ボックスを閉じます。

入力したシナリオ・ゴールの情報は、[シナリオのゴール] ウィンドウに表示されます。



監視するオンライン・グラフ

テストの目標と負荷の動作を定義したら、LoadRunner モニタを設定できます。このテストでは、[秒ごとのヒット数] グラフを監視してサーバで生成される負荷を追跡します。[トランザクション応答時間] グラフを監視して、サーバに負荷がかかっている状態で顧客への応答に要した時間を確認することもできます。また、[スループット] グラフおよび [Windows リソース] グラフで負荷の影響を監視できます。

秒ごとのヒット数、トランザクション応答時間、およびスループットの監視は、ユーザに対して設定されています。Windows リソース・モニタの設定は、練習 7「負荷テスト・シナリオの作成」の手順に従ってください。

ゴール指向シナリオの実行方法

シナリオとゴールを設定したので、負荷のかかった状態でテストを開始してアプリケーションを監視できます。本項では、ゴール指向シナリオを実行してテストの動作を調べます。

1 [コントローラ] ウィンドウの [実行] タブを開きます。

画面下部の [実行] タブを選択します。

シナリオはまだ実行されていないため、カウンタはすべて 0 でグラフもすべて空です。次の手順でシナリオを開始すると、グラフとカウンタは情報を表示し始めます。

2 結果ディレクトリの名前を指定します。

[結果] > [結果の設定] を選択して [結果ディレクトリの設定] ダイアログ・ボックスを開き、結果セットの一意の名前（たとえば `travel_agent_3hps`）を入力します。

3 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] を選択します。

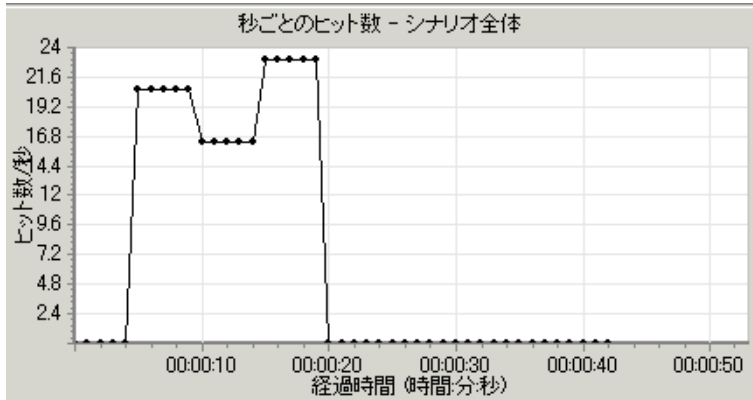
Controller がシナリオを開始します。

5 個の仮想ユーザがランプ・アップされて実行が開始されます。LoadRunner はゴールに必要な秒ごとのヒット数 3 個を生成しようとします。

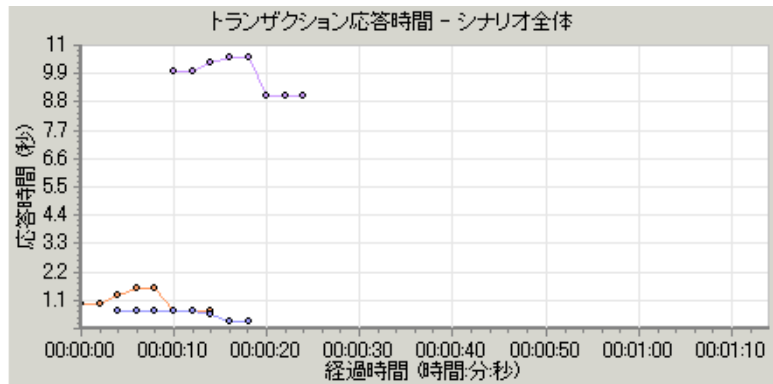
テスト中、Controller は仮想ユーザを自動的に開始、終了して、指定されたゴールを保持します。

4 オンライン・グラフを表示します。

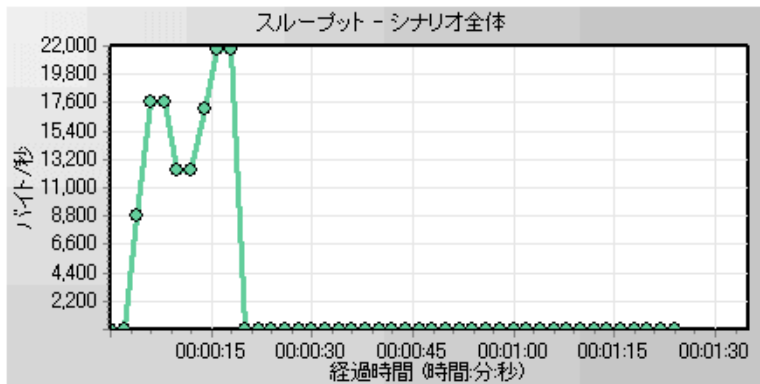
[秒ごとのヒット数] グラフは、シナリオの経過秒ごとに仮想ユーザが Web サーバに対して行ったヒット (HTTP 要求) の数を示します。すぐに必要な負荷レベルに達します。



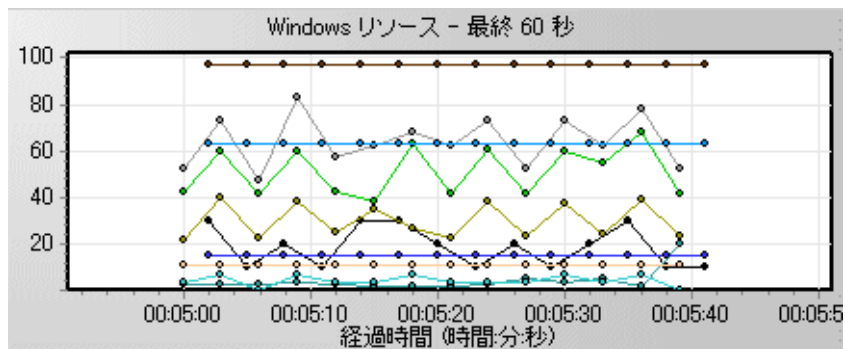
[トランザクション応答時間] グラフには、各トランザクションが完了するのに要した時間が表示されます。トランザクション応答時間を監視して、サーバに負荷がかかっている状態で顧客への応答に要した時間を確認することはとても重要です。



また、[スループット] グラフを見るには、利用可能なグラフ・ツリーで [スループット] を選択してグラフ表示領域までドラッグします。[スループット] グラフは、任意の時点で仮想ユーザが Web サーバから受信したデータ量を表示します。



プロセッサ、ディスク、またはメモリのサーバの使用率の問題に関しては、サーバの Windows リソースの使用状況を監視できます。テスト中に監視することで、パフォーマンス問題の原因を直ちに特定するのに役立ちます。



[Windows リソース] グラフの凡例で測定値のリストを確認できます。

色	倍率	測定値	マシン	最高	最低	平均	標準	最終
	10	% Processor Time (Processor _Total)	hi-dt	8.333	4.663	6.394	1.289	4.688
	1	File Data Operations/sec (System)	hi-dt	75.660	25.665	51.599	13.084	41.673
	10	Processor Queue Length (System)	hi-dt	1.000	0.000	0.471	0.499	1.000
	0.1	Page Faults/sec (Memory)	hi-dt	400.380	203.381	291.813	81.016	205.698
	10	% Disk Time (PhysicalDisk _Total)	hi-dt	0.980	0.221	0.420	0.206	0.388
	1E-6	Pool Nonpaged Bytes (Memory)	hi-dt	15118336.000	15110144.000	15114962.820	4031.717	15110144.000
	10	Pages/sec (Memory)	hi-dt	0.668	0.331	0.463	0.163	0.333
	0.01	Interrupts/sec (Processor _Total)	hi-dt	1053.162	1037.994	1047.021	4.852	1052.159
	0.1	Threads (Objects)	hi-dt	633.000	631.000	631.118	0.471	631.000

ゴールの達成の確認

この練習の目標は、予想される現実の作業負荷の元で、システムが許容できるサービス・レベルを顧客に提供することを確認することでした。そのような条件をエミュレートするために、最小 5 個、最大 10 個の仮想ユーザを実行している状態で、シナリオの実行中の秒ごとのヒット数が 3 個という負荷目標を設定しました。5 個から 10 個の仮想ユーザを実行している状態で、シナリオ実行時の経過秒ごとにサーバ上で仮想ユーザが行ったヒットが 3 個の場合、ゴール・パラメータは達成されたこととなります。秒ごとのヒット数が 3 個に到達しない場合、LoadRunner は定義した目標を達成できなかったというメッセージを表示します。

注：このライセンスでは最大の仮想ユーザ数は 10 個に制限されるので、ゴールに到達しない場合があります。

テストの実行後、今後使用できるようシナリオの設定を保存する必要があります。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、[保存] ボタンをクリックして [ファイル名] ボックスにシナリオ名を入力します。



次の練習について

この練習で、ゴール指向シナリオの設計と実行が終了しました。次に練習 10 「シナリオの分析」に進みます。



練習 10

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
- ▶ サマリ・レポート - ゴールの達成の確認
- ▶ 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、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。

- 1 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応答時間はどうだったでしょうか。トランザクションの平均トランザクション応答時間はどうだったでしょうか。
- 2 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応答時間はどうだったでしょうか。
- 3 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 を起動します。

[スタート] > [プログラム] > [LoadRunner] > [LoadRunner] を選択します。HP LoadRunner Launcher ウィンドウが開きます。

2 LoadRunner Analysis を起動します。

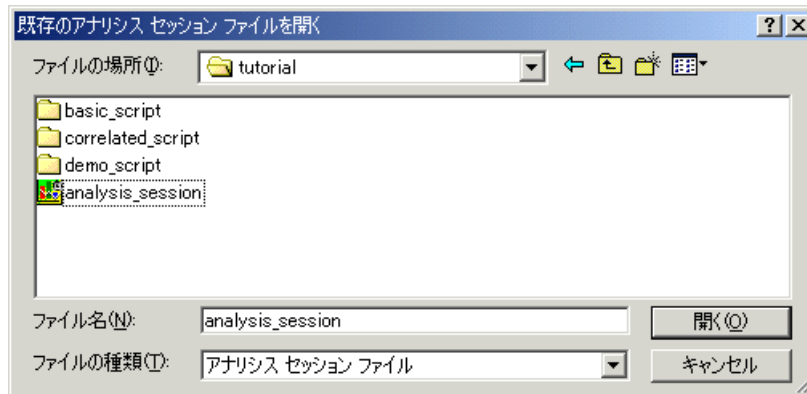
[負荷テスト] タブで、[負荷テストの分析] をクリックします。LoadRunner Analysis が起動されます。

3 Analysis セッション・ファイルを開きます。

本項ではより注目に値する結果を得るため、前の練習で実行したものとほぼ同じテスト・シナリオを実行しました。ただし、今回のテストでは、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。

Analysis のウィンドウで、[ファイル] > [開く] を選択します。[既存アナリシスセッションファイルを開く] ダイアログ・ボックスが開きます。

< 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。

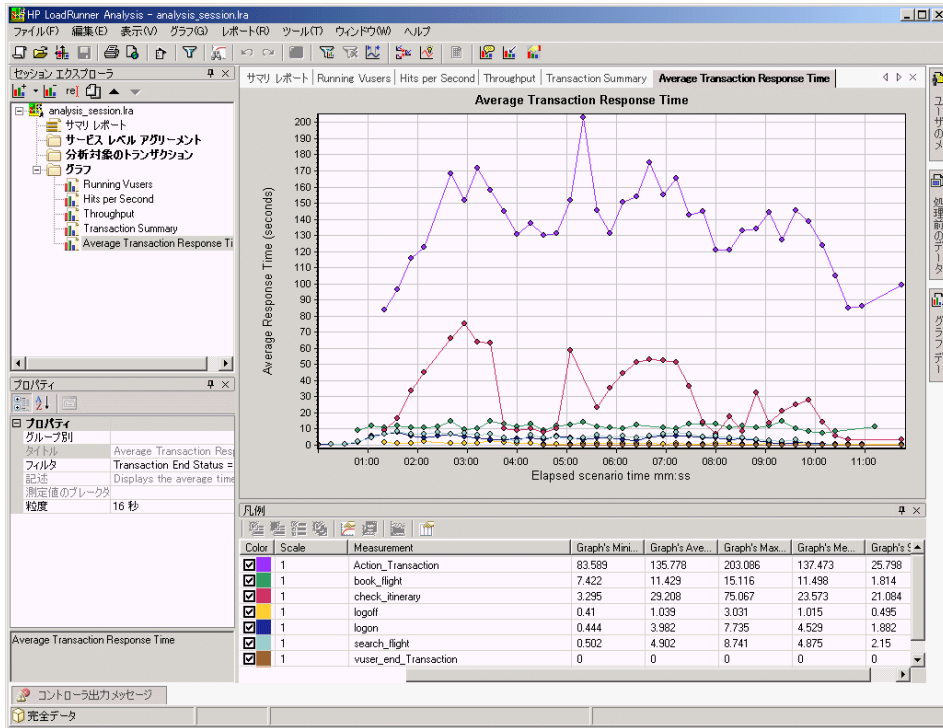


注：この Analysis セッションは英語版 LoadRunner で作成されています。したがって、表示される語句が英語の場合があります。セッションを日本語版 LoadRunner で作成した場合、これらは日本語表示となります。

Analysis のウィンドウの概要

Analysis には、次の 3 つの主要なウィンドウがあります。

- ▶ セッション エクスプローラ
- ▶ グラフ表示領域
- ▶ グラフの凡例



[セッション エクスプローラ]：左上の表示枠には、閲覧できるレポートおよびグラフが表示されています。この表示枠で、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。

[グラフ表示領域]：右上の表示枠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 Analysis のサマリ・レポートが表示されます。

[グラフの凡例]：右下の表示枠には、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、追加の情報を提供するほかのウィンドウにアクセスできます。たとえば、[プロパティ] ウィンドウがあります。これらのウィンドウは、画面上の任意の場所にドラッグしてドロップできます。

サマリ・レポート

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。このレポートには、次のセクションがあります。

統計サマリ

レポートの**統計サマリ**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。参考のために、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記録されています。

アナリシス サマリ

期間: 28/06/2004 09:59:49 - 28/06/2004 10:11:46

シナリオ名: C:\Tutorial_Scenario_gaby.lrs
 セッションの結果: S:\run6\nt\1414\runner_download\prd\tutorial\tutorial_result\tutorial_result.lrs
 継続時間: 11 分 および 57 秒。

統計サマリ

最大実行仮想ユーザ数:	70	
合計スループット (バイト):	273,489,363	
平均スループット (バイト/秒):	380,904	
合計ヒート数:	8,789	
秒ごとの平均ヒート数:	12.241	HTTP 応答サマリの表示
エラー総数:	9,226	

次を使用して SLA データを定義できます [SLA 設定ウィザード](#)

次を使用してトランザクション動作を分析できます [トランザクションメカニズムの分析](#)

トランザクション・サマリ — ゴールの達成の確認

Analysis セッションの Analysis のサマリ・レポートで一番重要なセクションは、**[トランザクション サマリ]** です。

トランザクション サマリ

トランザクション: 成功総数: 1,100 失敗総数: 3,112 停止総数: 0 平均応答時間

Transaction Name	SLA Status	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
Action Transaction		78.016	139.18	252.471	28.215	170.866	144	3,081	0
book_flight		5.375	11.399	17.541	3.015	15.407	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
user_end_Transaction		0	0	0	0	0	70	0	0
user_init_Transaction		0	0.013	0.077	0.024	0.062	70	0	0

サービスレベル アグリーメントの凡例: 成功 失敗 データなし

[トランザクション サマリ] には、各トランザクションの振る舞いのサマリが表示されます。

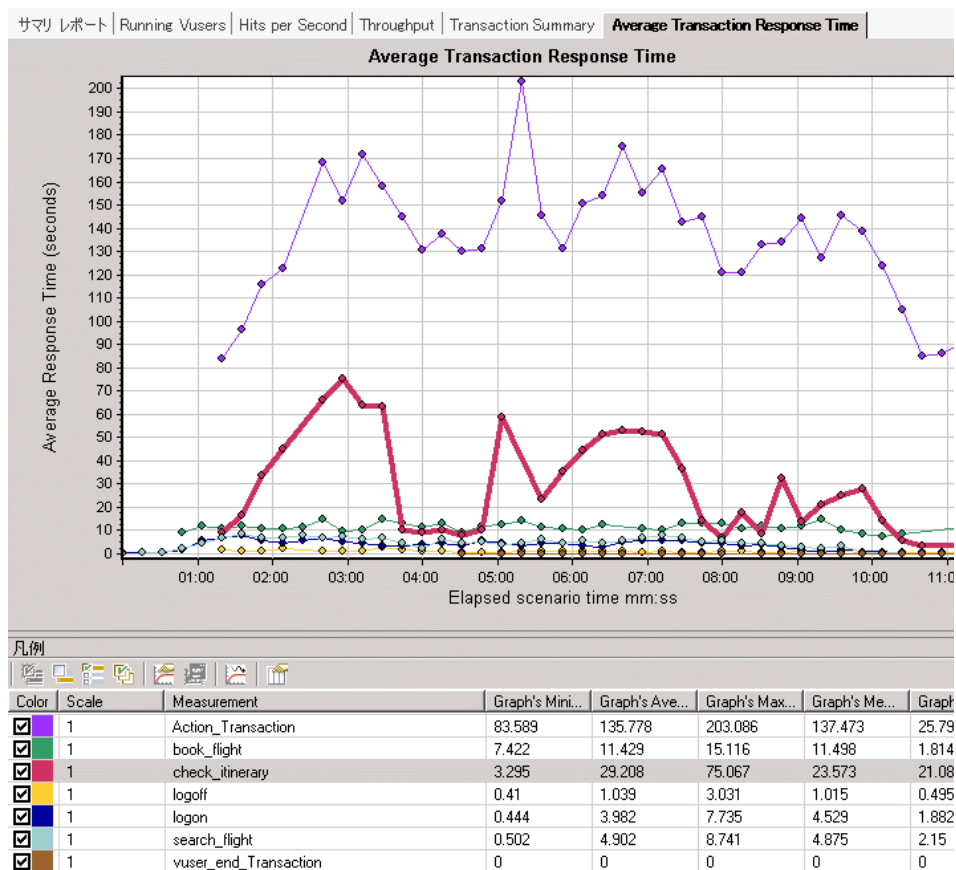
各トランザクションの応答時間を見ます。[90 パーセント] カラムは、ある特定のトランザクション実行の 90% の応答時間を示します。テスト実行中に実行された `check_itinerary` トランザクションのうち、90% の応答時間が 65.744 秒であったことがわかります。これは、このトランザクションの平均応答時間 (32.826 秒) の倍です。つまり、大部分の `check_itinerary` トランザクションの応答時間は非常に長かったということです。

また、トランザクションが 28 回失敗したこともわかります。

1 [平均トランザクション応答時間] (Average Transaction Response Time) グラフを開きます。

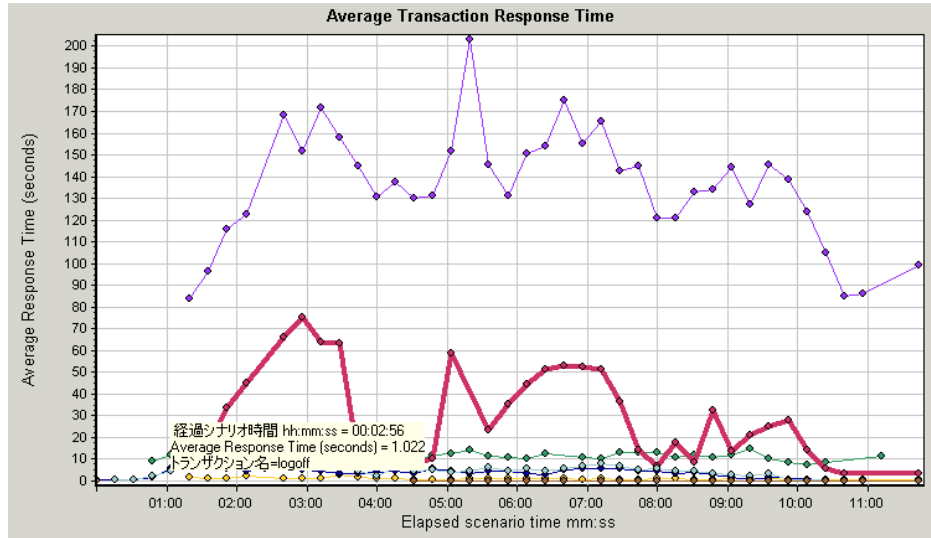
[トランザクション名] カラムで **check_itinerary** トランザクションをクリックします。

グラフ表示領域で [平均トランザクション応答時間] (Average Transaction Response Time) グラフが開きます。グラフ内とグラフの下の凡例で、**check_itinerary** トランザクションが強調表示されます。



練習 10・シナリオの分析

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応答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 結果を分析します。

check_itinerary トランザクションの平均応答時間が大きく変動している点に注目してください。平均応答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。

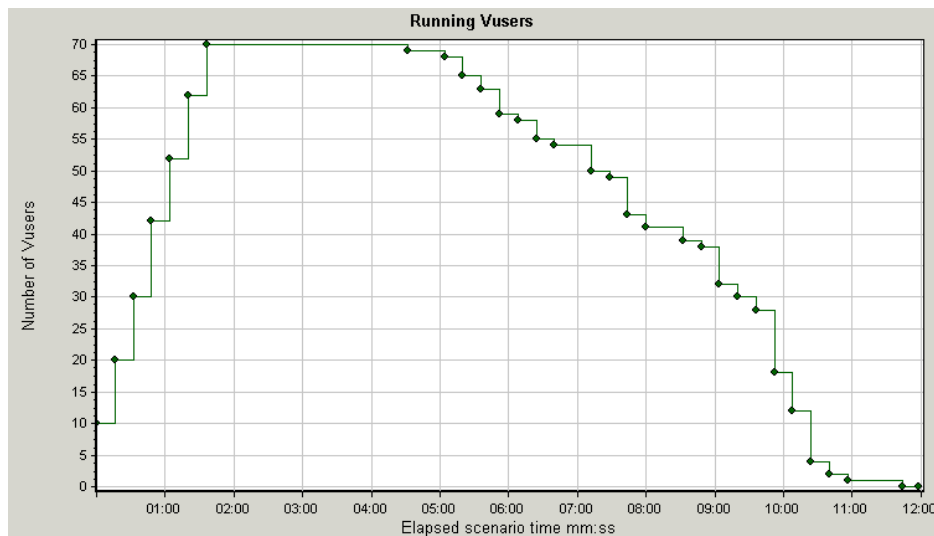
パフォーマンスの良いサーバでは、トランザクションの平均応答時間は比較的安定するでしょう。グラフの下部で、logon トランザクション、logoff トランザクション、book_flight トランザクション、および search_flight トランザクションの平均応答時間がいくらか安定している点に注目してください。

サーバのパフォーマンスの確認

前の項で、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 仮想ユーザの振る舞いを調査します。

グラフ・ツリーで **[実行中の仮想ユーザ]** (Running Vusers) 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]** (Running Vusers) 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ランプ・ダウンされ始めたのがわかります。

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。



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、またはツールバーの **[フィルタ/グループ化の設定]** アイコンをクリックします。

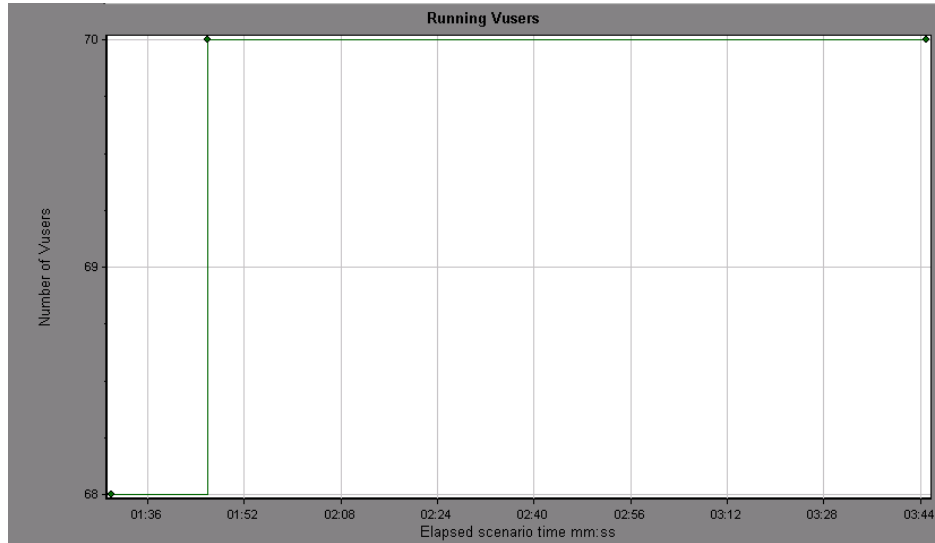
練習 10・シナリオの分析

[フィルタ条件] 領域で, [シナリオ経過時間] 行の [値] カラムを選択します。下向き矢印をクリックして, 時間範囲を 1:30 から 3:45 までにします。
[OK] をクリックします。

[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: グラフ内にデータが表示されない場合は, [実行中の仮想ユーザ] グラフを新規追加して, 同じようにフィルタを適用してください。



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, またはツールバーの [フィルタとグループ化のクリア] アイコ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応答時間に与える影響を確認できます。

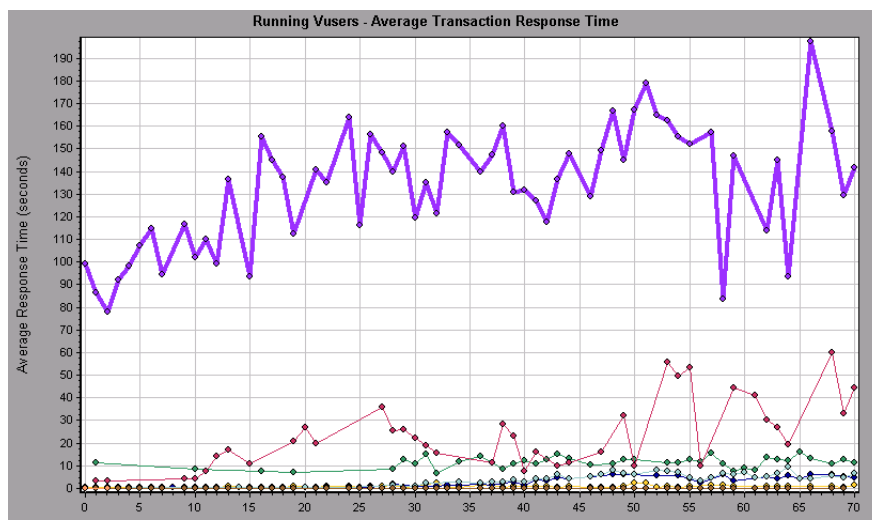
[実行中の仮想ユーザ] グラフを右クリックして、[**グラフの結合**] を選択します。

[**結合するグラフの選択**] 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。

[**結合の種類を選択**] 領域で [**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] グラフとしてグラフ表示領域に表示されます。

注： グラフ内にデータが表示されない場合は、[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] グラフを新規追加して、同じように結合させてください。



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応答時間がわずか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、平均応答時間も確実に増えています。

仮想ユーザ数が 64 個になった時点で、突然平均応答時間が急増します。これをテストによる「**サーバの限界超越**」といいます。実行中の仮想ユーザ数が 64 個を超えると同時に、明らかに応答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に、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] メニューで、[テンプレート] > [テンプレートとして保存] を選択します。
- 2 テンプレートの適切な名前を入力します。
- 3 [新規セッションにこのテンプレートを自動的に適用する] オプションをクリアします。
- 4 [OK] をクリックします。

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] メニューで、[テンプレート] > [テンプレートの編集 / 適用] を選択します。
- 2 リストからテンプレートを選択し、[テンプレートの適用] をクリックします。

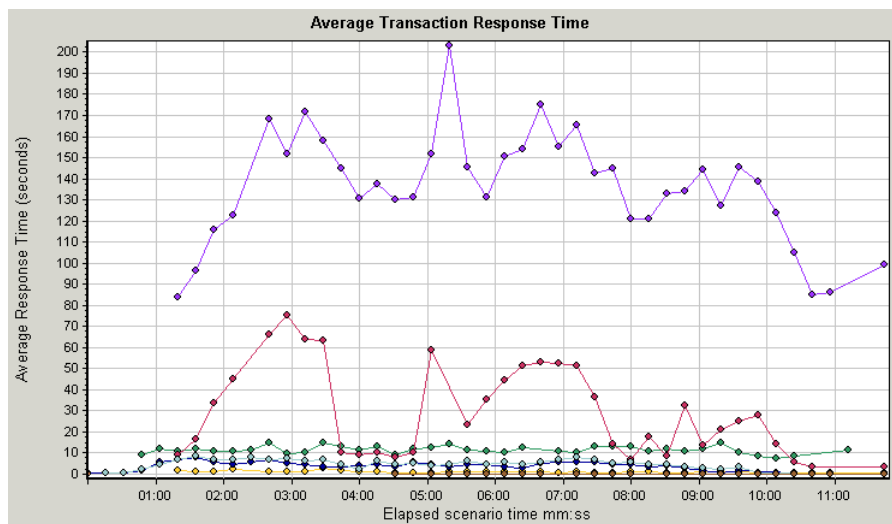
問題の原因の特定方法

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。

LoadRunner Analysis 固有の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。

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。

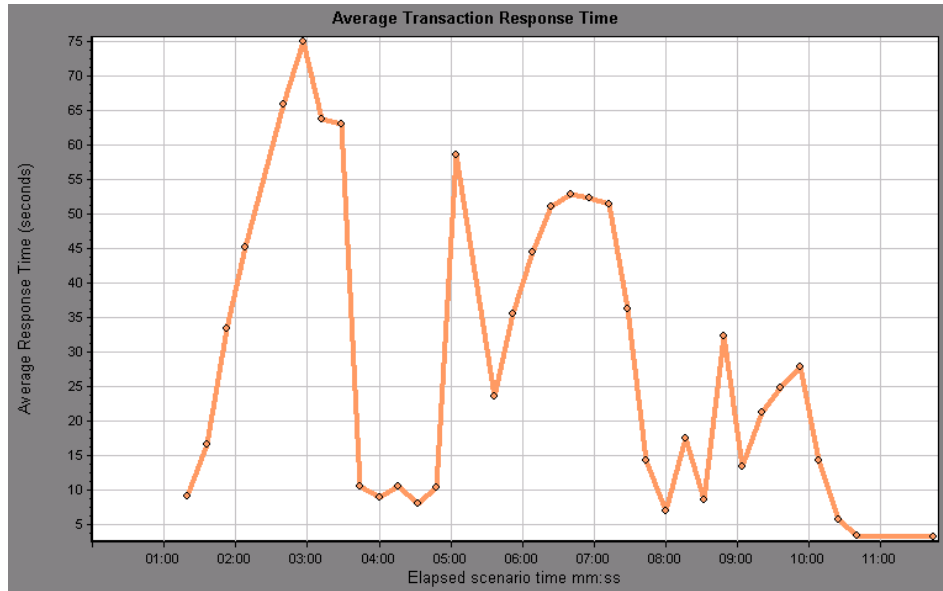
[トランザクション名] ボックスで `check_itinerary` を選択します。

[OK] をクリックします。

練習 10・シナリオの分析

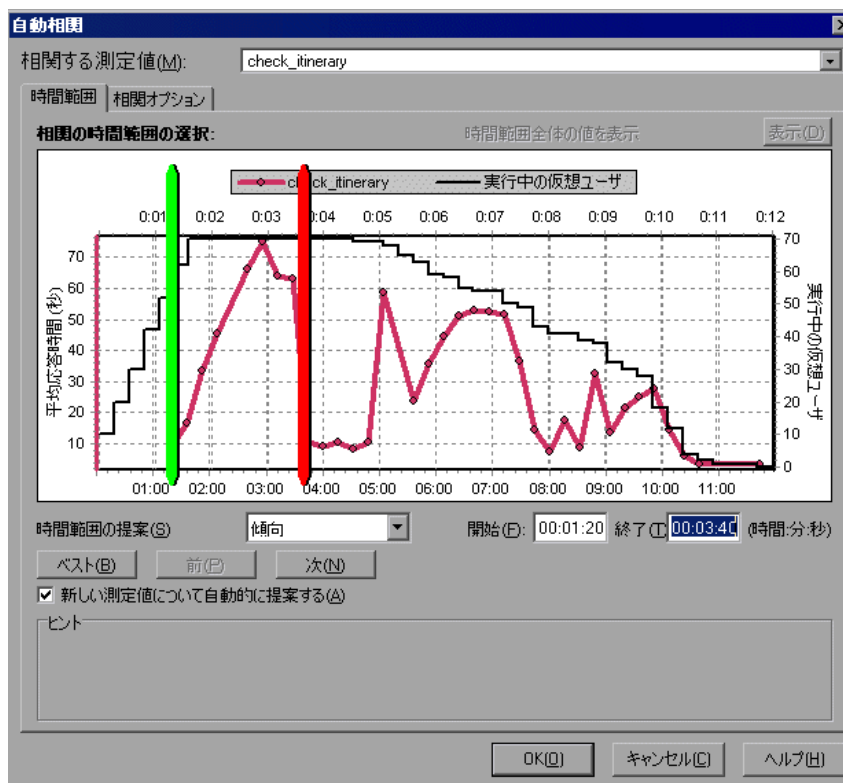
フィルタが適用されたグラフは `check_itinerary` トランザクションのみを表示し、ほかのすべてのトランザクションは表示されません。

注： グラフ内にデータが表示されない場合は、[平均トランザクション応答時間] グラフを新規追加してフィルタを実行してください。



3 グラフを自動相関します。

グラフを右クリックして，[自動相関] を選択します。

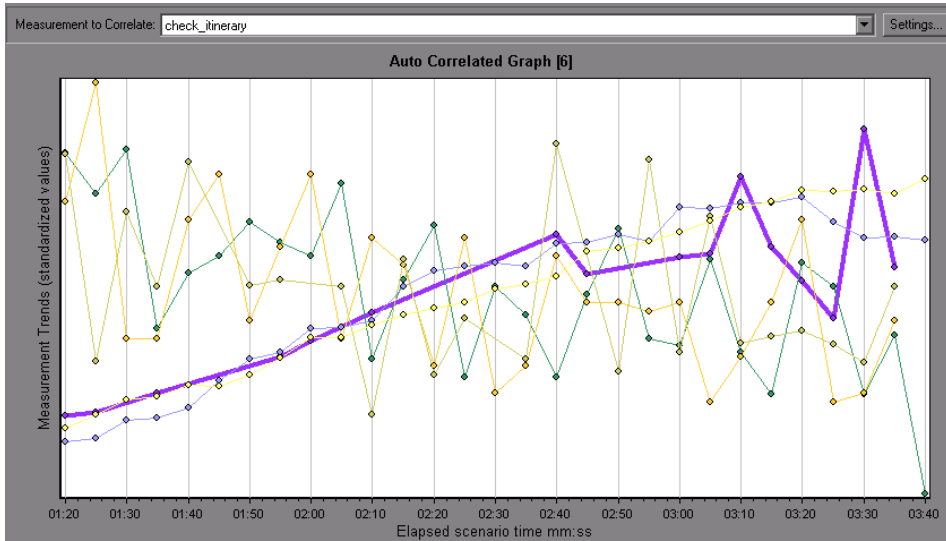


[自動相関] ダイアログ・ボックスで，相関する測定値が **check_itinerary** であることを確認し，時間範囲を **1:20** から **3:40** に設定します。時間範囲は，ボックスに時間を入力するか，経過シナリオ時間軸に沿って緑と赤の棒をドラッグして設定します。

[OK] をクリックします。

練習 10・シナリオの分析

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 [番号]**」が付けられます。

4 グラフの名前を変更します。

グラフ・ツリーで、**[自動相関グラフ [番号]]** グラフをクリックし、**[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。

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。

グラフの下の凡例を見ます。

Color	Graph	Scale	Measurement	Correlation Match	Correlation	Machine Na...	Monitor Type
<input checked="" type="checkbox"/>	Average Tra...	Standardized	check_itinerary	100	Directly Related	N/A	N/A
<input checked="" type="checkbox"/>	Windows R...	Standardized	Private Bytes (Process_Total):alabama	76	Directly Related	Alabama	
<input checked="" type="checkbox"/>	Windows R...	Standardized	Pool Nonpaged Bytes (Memory):alabama	75	Directly Related	Alabama	
<input checked="" type="checkbox"/>	Windows R...	Standardized	Page Faults/sec (Process_Total):alabama	48	Inversely Related	Alabama	
<input checked="" type="checkbox"/>	Hits per Sec...	Standardized	Hits	44	Inversely Related	N/A	N/A
<input checked="" type="checkbox"/>	Windows R...	Standardized	% Processor Time (Process_xiwin32):alabama	44	Inversely Related	Alabama	
<input type="checkbox"/>	Windows R...	Standardized	Interrupts/sec (Processor_Total):alabama	42	Directly Related	Alabama	

[測定値] カラムで、メモリに関連する測定値である **Private Bytes** と **Pool Nonpaged Bytes** 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、指定した時間帯で、これらの要素の振る舞いと `check_itinerary` トランザクションの振る舞い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推定できます。

注：日本語版 LoadRunner では、チュートリアル用 Analysis セッションで、システム・リソース・データが相関結果に反映されないことがあります。その場合、[測定値] カラムに Private Bytes と PoolNonpaged Bytes が表示されません。

シナリオ実行で収集可能なその他の情報

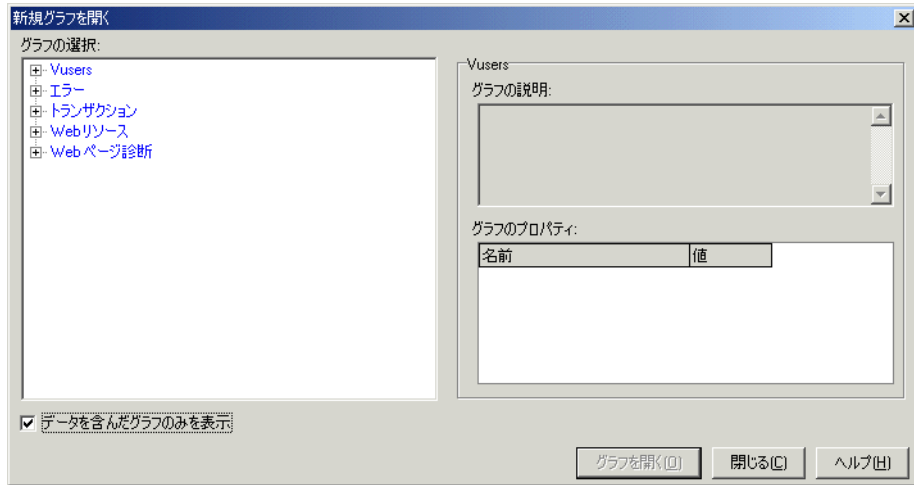
Analysis セッションの開始時点でグラフ・ツリーに表示されるグラフに加えて、シナリオ実行に関するその他の情報を取得するためにさまざまなグラフを表示できます。

1 新しいグラフを表示します。



ツールバーで [新規グラフの追加] ボタンをクリックするか、またはグラフ・ツリーで [新規項目の追加] > [新規グラフの追加] を選択します。

[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。



- ▶ **[Vusers]** グラフには、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]** グラフには、エラーの統計が表示されます。
- ▶ **[トランザクション]** グラフには、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]** グラフには、ヒット・データ、スループット・データ、および接続データが表示されます。
- ▶ **[Web ページ診断]** グラフには、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]** グラフには、システム・リソースの使用状況のデータが表示されます（日本語版 LoadRunner では、チュートリアル用 Analysis セッションで、システム・リソースが表示されない場合があります）。

[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。

グラフを選択し、**[グラフを開く]** をクリックします。

[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] メニューから [HTML レポート] を選択します。
- 2 レポートのファイル名と格納場所のパスを選択します。[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。左の表示枠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。

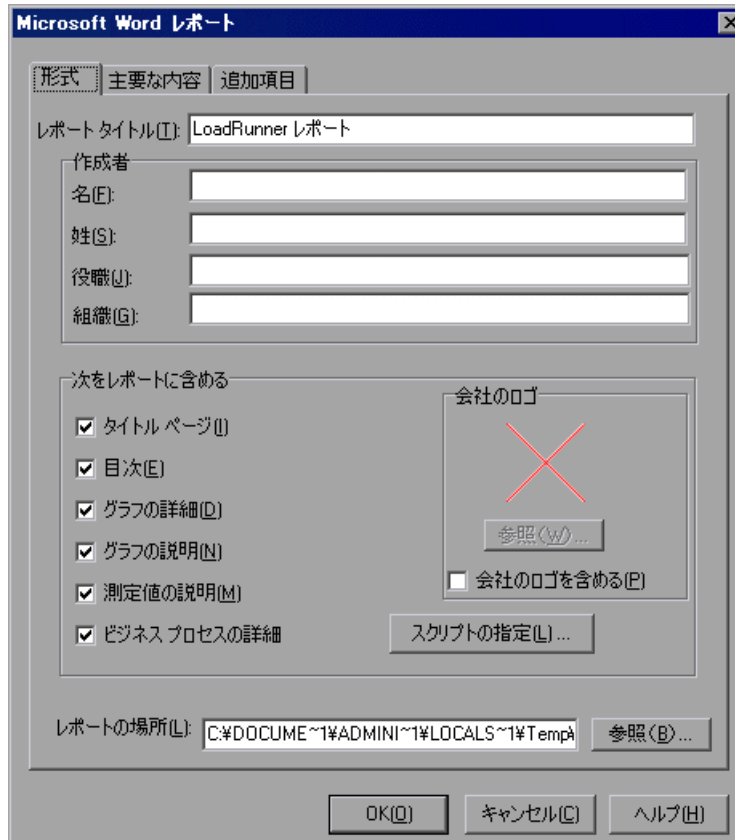
Microsoft Word レポート

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。

レポートは Word ファイルのように編集可能なので、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] メニューから、[Microsoft Word レポート] を選択します。
[Microsoft Word レポート] ダイアログ・ボックスが開きます。



- 2 [形式] タブをクリックします。
 - ▶ レポートのタイトルを入力します。
 - ▶ 作成者の名前、役職、組織名を入力します。
 - ▶ 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]** 領域で、**[会社のロゴを含める]** を選択して **[参照]** をクリックし、ロゴ・ファイルのパスを検索します。ロゴ・ファイルは **.bmp** 形式のファイルです。
- ▶ レポートの格納場所を入力します。

3 **[主要な内容]** タブをクリックします。

- ▶ レポートに含めるシナリオおよび Analysis セッションのセクションを選択します。標準設定では、**[サーバのパフォーマンス]** を除いて、リストにあるすべてのセクションが選択されます。

[サーバのパフォーマンス] を選択します。

- ▶ **[編集]** をクリックします。**[エグゼクティブ サマリ]** ダイアログ・ボックスが開きます。ここで目的と結論を入力します。

[目標] ボックスに、「**テスト・シナリオの目的は・・・**」と入力します。

[結論] ボックスに、「**結論は次のとおりです**」と入力します。

[OK] をクリックし、**[エグゼクティブ サマリ]** ダイアログ・ボックスを閉じます。

4 **[追加項目]** タブをクリックします。

- ▶ レポートに含めるグラフを指定します。標準設定では、セッションのすべてのグラフがリストに表示されて選択されます。またレポートにはグラフのメモが含まれます。
- ▶ 詳細については、Analysis セッションでまだ開いていなかったグラフを追加できます。

[追加] ボタンをクリックします。**[新規グラフを開く]** ダイアログ・ボックスが開きます。**[システム リソース]** カテゴリを展開して **[Windows リソース]** グラフを選択します。**[グラフの追加]** をクリックします。**[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。レポートに含めるグラフのリストに **[Windows リソース]** グラフが表示されます。

- ▶ レポートのグラフの表示順序を指定できます。

[平均トランザクション応答時間] をクリックして、このグラフを選択します。

グラフが **[実行中の仮想ユーザ]** の下に表示されるまで **[UP]** ボタンをクリックします。

練習 10・シナリオの分析

レポートでは、[実行中の仮想ユーザ] グラフの下に [平均トランザクション応答時間] グラフが表示されます。

- 5 [Microsoft Word レポート] ダイアログ・ボックスで、[OK] をクリックします。データが集められ、レポートが 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、シナリオ実行の分析と、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や、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。