

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 : 11.00 パッチ 02

チュートリアル

ドキュメント・リリース日 : 2011 年 2 月 (英語版)

ドキュメント・リリース日 : 2011 年 2 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。

著作権

© Copyright 1993 - 2011 Hewlett-Packard Development Company, L.P.

商標

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

文書の更新

本書の表紙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。
- 文書が更新されるごとに変更されるドキュメント・リリース日。
- 当該ソフトウェア・バージョンのリリース日を示す、ソフトウェア・リリース日。

最新の更新を確認する、あるいは使用している文書が最新版であるかどうかを確認するには、次の URL を参照してください。

<http://h20230.www2.hp.com/selfsolve/manuals>

このサイトを使用するには HP Passport に登録してサインインする必要があります。HP Passport ID を登録するには、次の URL を参照してください。

<http://h20229.www2.hp.com/passport-registration.html>

または、HP Passport のログイン・ページで [New users - please register] リンクをクリックしてください。

適切な製品のサポート・サービスに登録すれば、最新版または新版を入手できます。詳細については、HP の担当窓口にお問い合わせください。

サポート

HP Software のサポート Web サイトは次のとおりです。

<http://www.hp.com/go/hpsoftwaresupport>

この Web サイトでは、連絡先情報や、HP Software が提供する製品、サービスおよびサポートの詳細を提供しています。

HP Software のオンライン・サポートでは、セルフソルブ機能を提供しています。ビジネス管理に必要な対話型技術サポート・ツールにアクセスするための迅速かつ効率的な方法を提供します。弊社サポートの大切なお客様として、サポート Web サイトを使用して次のことが行えます。

- 興味のあるナレッジ文書の検索
- サポート事例と向上のためのリクエストの送信および追跡
- 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 サポート契約の検索
- 利用可能なサービスに関する情報の確認
- ソフトウェアを利用しているほかのお客様との討論への参加
- ソフトウェア・トレーニングの検索と登録

サポート領域のほとんどでは HP Passport ユーザとして登録しサインインする必要があります。また多くでサポート契約も必要です。HP Passport ID を登録するには、次の URL を参照してください。

<http://h20229.www2.hp.com/passport-registration.html>

アクセス・レベルの詳細については、次の URL を参照してください。

http://h20230.www2.hp.com/new_access_levels.jsp

目次

LoadRunner チュートリアルへようこそ	9
第 1 章 : LoadRunner の紹介	11
パフォーマンス・テストを自動化する理由	12
LoadRunner コンポーネントについて	12
LoadRunner の用語について	13
負荷テスト・プロセスについて	14
HP Web Tours について	14
アプリケーションの要件	16
第 2 章 : LoadRunner の威力	17
負荷テストの作成	18
負荷テストの実行	21
負荷テストの監視	22
結果の分析	25
第 3 章 : スクリプトの作成	27
Virtual User Generator (VuGen) について	27
ユーザ動作の記録の開始方法	28
VuGen のウィザード・モードの使用方法	30
ビジネス・プロセスの記録によるスクリプトの作成方法	32
スクリプトの表示方法	36
第 4 章 : スクリプトの再生	39
実行時の動作の設定方法	39
スクリプト実行のリアルタイムでの監視方法	43
再生に関する情報の表示方法	45
テストの成功の確認方法	46
結果の検索またはフィルタ方法	48
第 5 章 : 再生に関する一般的な問題の解決方法	53
再生エラーのための HP Web Tours の準備	54
一意のサーバ値の使用方法	54

第 6 章：負荷テスト用スクリプトの準備	59
ビジネス・プロセスの測定方法.....	60
複数のユーザをエミュレートする方法.....	63
Web ページの内容の確認方法.....	68
デバッグ情報の生成方法.....	69
テスト成功の確認方法.....	71
第 7 章：負荷テスト・シナリオの作成	73
LoadRunner Controller の概要.....	74
Controller の開始方法.....	74
Controller ウィンドウの概要.....	77
スクリプトの詳細の変更方法.....	78
大きな負荷を生成する方法.....	79
実際の負荷動作のエミュレート方法.....	80
さまざまなタイプのユーザのエミュレート方法.....	85
負荷下のシステムの監視方法.....	88
第 8 章：負荷テストの実行	93
Controller の実行ビューの概要.....	94
負荷テスト・シナリオの実行方法.....	96
負荷下のアプリケーションの監視方法.....	98
リアルタイムで実行しているユーザの監視方法.....	100
ユーザ・アクションのサマリの表示場所.....	101
テスト中に負荷を増やす方法.....	102
負荷下のアプリケーションの動作方法.....	103
アプリケーションでエラーが発生したかどうか.....	104
テストの実行が完了したことを知る方法.....	106
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.....	106
第 9 章：高度なゴール指向シナリオ	107
使用するゴール・タイプ.....	108
ゴール指向シナリオの作成方法.....	109
Controller ウィンドウの概要（ゴール指向シナリオ）.....	110
ゴールの定義方法.....	111
負荷の動作の確認方法.....	113
監視するオンライン・グラフ.....	115
ゴール指向シナリオの実行方法.....	116
ゴールの達成の確認.....	119

第 10 章：シナリオの分析	121
Analysis セッションの処理方法	122
Analysis セッションの開始方法	123
Analysis のウィンドウの概要	124
ゴールの達成の確認（サービス・レベル・アグリーメント）	125
パフォーマンス・サマリの表示	133
パフォーマンスのグラフによる表示方法	137
サーバのパフォーマンスの確認	140
問題の原因の特定方法	144
シナリオ実行で収集可能なその他の情報	148
結果の発行方法	150
結論	153

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。

練習 1 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 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。

はじめに

1

LoadRunner の紹介

この 30 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、かつて紙や手動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。

そのうえ、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恒久的な部品がありません。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。

驚くほどの変化の速さ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戦略です。HP LoadRunner による自動負荷テストは、アプリケーション導入プロセスの重要な要素です。

このセクションの内容

- ▶ 12 ページの「パフォーマンス・テストを自動化する理由」
- ▶ 12 ページの「LoadRunner コンポーネントについて」
- ▶ 13 ページの「LoadRunner の用語について」
- ▶ 14 ページの「負荷テスト・プロセスについて」
- ▶ 14 ページの「HP Web Tours について」
- ▶ 16 ページの「アプリケーションの要件」

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。

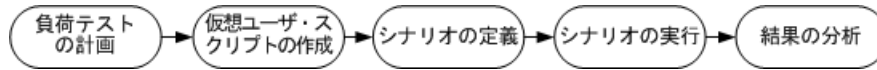
- ▶ **Virtual User Generator** は、エンドユーザのビジネス・プロセスをキャプチャし、仮想ユーザ・スクリプトとも呼ばれる自動パフォーマンス・テスト・スクリプトを作成します。
- ▶ **Controller** は、負荷テストの編成、実行、管理、監視を行います。
- ▶ **Load Generator** は、仮想ユーザを実行して負荷を生成します。
- ▶ **Analysis** は、パフォーマンス結果の表示、分析、比較を行うのに役立ちます。
- ▶ **Launcher** は、すべての LoadRunner コンポーネントへのアクセス・ポイントを提供します。

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 Web 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 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 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。

このセクションでは，HP Web Tours Application について確認します。

HP Web Tours の起動

HP Web Tours Application 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] > [すべてのプログラム] > [HP 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 Web Tours Application を開きます。

[スタート] > [すべてのプログラム] > [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ディレクトリにインストールされていることを確認します。LoadRunner が標準でないディレクトリにインストールされている場合、HP Web Tours Application は開きません。
 - ▶ HP Web Tours Application 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。

次の情報を入力します。

- ▶ [User name] : jojo

▶ [Password] : bean

左側の表示枠で [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 左側の表示枠で [Flights] をクリックします。[Find Flight] ページが開きます。[Arrival city] を [Los Angeles] に変更します。[Continue] をクリックします。
- b [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。
- c [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

これで、HP Web 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 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。

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、フライトの購入、日程の確認、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 サーバの動作を観察できます。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。

この練習は、次の項目で構成されています。

- ▶ 18 ページの「負荷テストの作成」
- ▶ 21 ページの「負荷テストの実行」
- ▶ 22 ページの「負荷テストの監視」
- ▶ 25 ページの「結果の分析」

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。

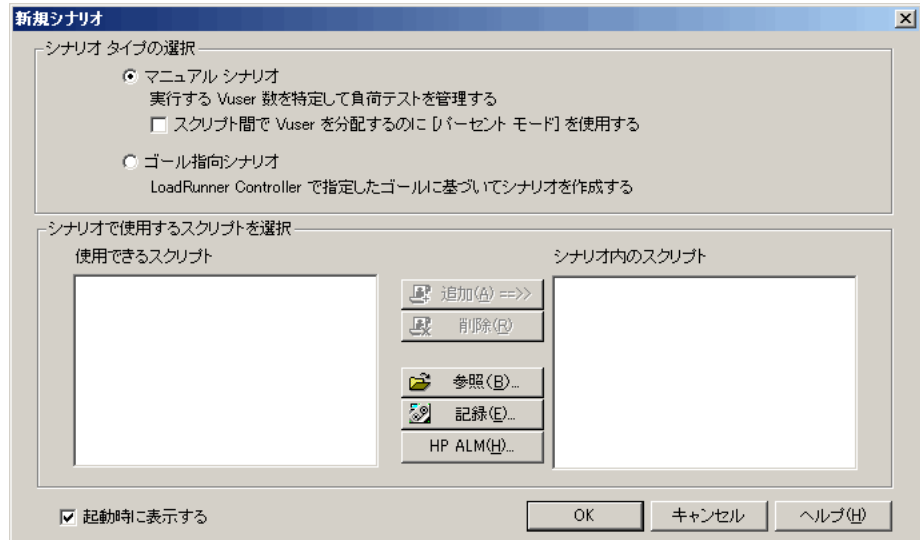
1 [HP LoadRunner] ウィンドウを開きます。

[スタート] > [すべてのプログラム] > [HP LoadRunner] > [LoadRunner] を選択します。[HP LoadRunner 11.00] ウィンドウが開きます。



2 Controller を起動します。

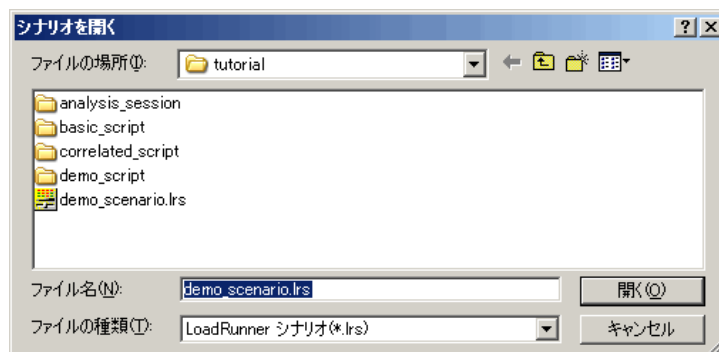
[LoadRunner ランチャー] 表示枠で、[**負荷テストの実行**] をクリックします。標準設定では、LoadRunner Controller が [新規シナリオ] ダイアログ・ボックスとともに表示されます。



[**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。



第 2 章 • LoadRunner の威力

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

グループ名	スクリプトのパス	数	Load Generators
demo_script.1	C:\Program Files\HP\LoadRunner\tutorial\demo_script	10	localhost

シナリオのスケジュール

スケジュール名: Default Schedule

スケジュール設定: シナリオ (グループ)

実行モード: 実スケジュール (基本スケジュール)

全体スケジュール

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 00:01:00 (HHMMSS) にロードす。
実行時間	00:05:00 (HHMMSS) の間実行する
仮想ユーザの停止	すべて 仮想ユーザを同時に停止する

対話式スケジュール グラフ

凡例: 全体スケジュール

Y軸: Users (0-12)

X軸: 時間 (00:00:00 - 00:10:00)

スケジュールグラフのデータ:

時間	仮想ユーザ数
00:00:00	0
00:01:00	2
00:02:00	4
00:03:00	6
00:04:00	8
00:05:00	10
00:06:00	10
00:07:00	10
00:08:00	10
00:09:00	10
00:10:00	0

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。

これで, テスト実行の準備が整います。

負荷テストの実行



[実行] タブで、[シナリオの開始] 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。

The screenshot shows the HP LoadRunner Controller interface. The top window is titled "HP LoadRunner Controller - demo_scenario.lrs - [実行]". It features a menu bar (File, View, Scenario, Monitor, Results, Diagnostics, Tools, Help) and a toolbar with icons for starting, pausing, and stopping the scenario.

On the left side, there are three labels with arrows pointing to specific parts of the interface:

- [シナリオの開始] points to the "シナリオの開始(S)" button.
- [シナリオグループ] セクション points to the "グループ" (Groups) table.
- [オンライングラフ] 領域 points to the "利用可能なグラフ" (Available Graphs) list.

The "シナリオ ステータス" (Scenario Status) panel on the right shows the current state of the scenario, including "実行中の User" (Running Users), "確認時間" (Check Time), and "秒ごとのヒット数" (Hits per Second).

The "Running Users - シナリオ全体" graph shows the number of users running over time. The "Hits per Second - シナリオ全体" graph shows the number of hits per second. The "Trans Response Time" graph shows the response time for transactions. The "Windows Resource" graph shows the resource usage of the Windows operating system.

At the bottom, there are three tabs: "デザイン" (Design), "実行" (Execution), and "J2EE/.NET の診断" (J2EE/.NET Diagnostics). The "実行" tab is currently selected.

At the bottom of the screenshot, there are two labels with arrows pointing to the tabs:

- [実行] タブ points to the "実行" tab.
- [グラフの測定値] 領域 points to the "J2EE/.NET の診断" tab.

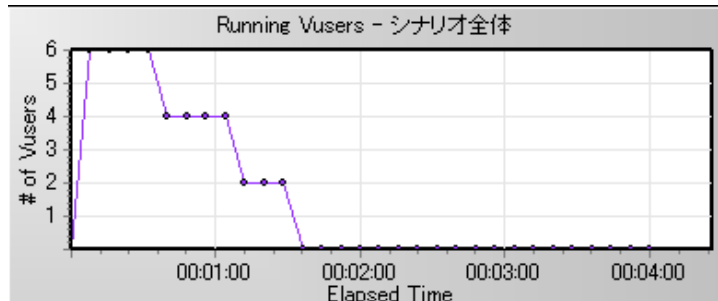
負荷テストの監視

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。

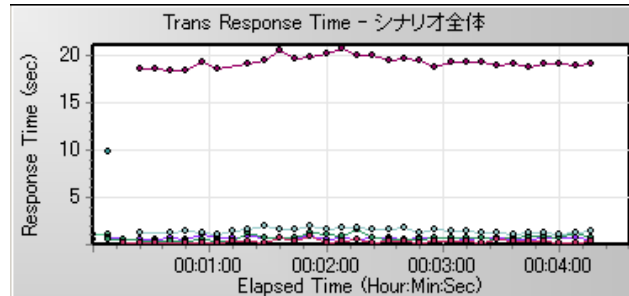
1 標準のグラフを表示します。

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。

- ▶ **[実行中の仮想ユーザ - 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。



- ▶ **[トランザクション応答時間 - シナリオ全体]** : このグラフでは、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、フライトの購入、日程の確認、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。

- ▶ **[秒ごとのヒット数 - シナリオ全体]** : このグラフでは、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 リソース]** : このグラフでは、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。

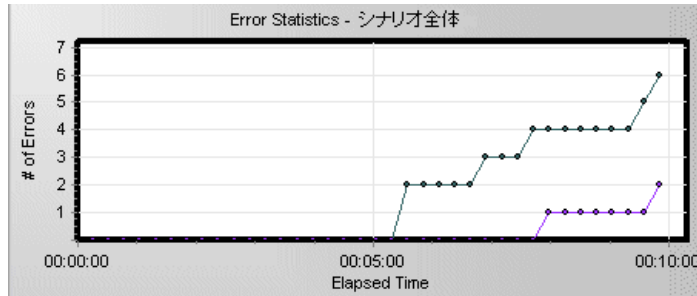
ヒント :

- ▶ 各測定値は、ウィンドウの下部にある [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 [エラーの統計] グラフを選択し、
[Windows リソース] グラフ表示枠までドラッグします。[エラーの統計]
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。あるいは自動関連ツールを使用して、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わかります。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。



シナリオの結果を表示するために Analysis を開くには、[結果] > [結果の分析] 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。

3

スクリプトの作成

負荷を生成するには、最初に、実際のユーザの動作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。

- ▶ 27 ページの「Virtual User Generator (VuGen) について」
- ▶ 28 ページの「ユーザ動作の記録の開始方法」
- ▶ 30 ページの「VuGen のウィザード・モードの使用法」
- ▶ 32 ページの「ビジネス・プロセスの記録によるスクリプトの作成方法」
- ▶ 36 ページの「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**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。

注： HP LoadRunner クイック・スタートが終了している場合は、そこで記録したスクリプト・ステップが次のセクションで記録するスクリプト・ステップとまったく同じであることに気が付くことでしょう。ここでは、全体的な記録プロセスについて、さらに詳細に説明します。

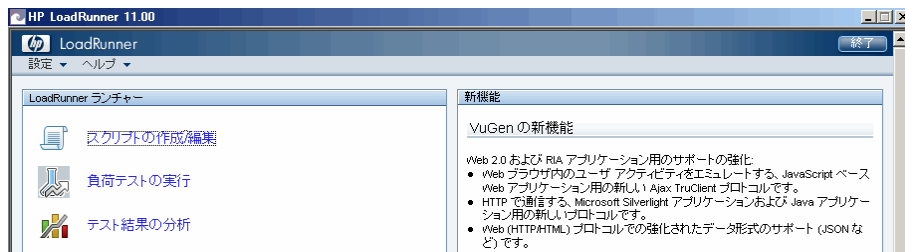
ユーザ動作の記録の開始方法

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。

このセクションでは、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] > [すべてのプログラム] > [HP LoadRunner] > [LoadRunner] を選択します。[HP LoadRunner 11.00] ウィンドウが開きます。



2 VuGen を開きます。

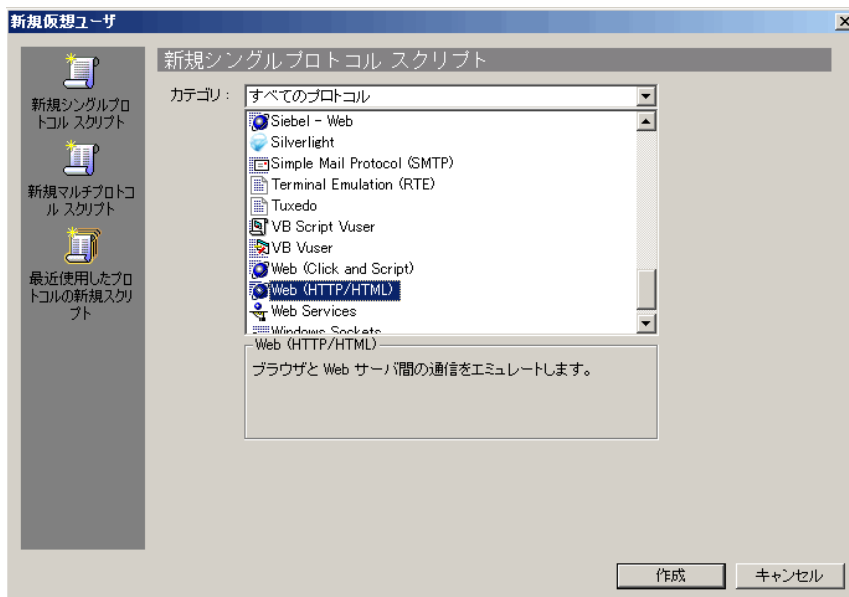
[LoadRunner ランチャー] 表示枠で、[スクリプトの作成/編集] をクリックします。VuGen の [スタートページ] が開きます。

3 空の Web スクリプトを作成します。



[Virtual User Generator へようこそ] 領域で、[新規スクリプト] ボタンをクリックします。

[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 Web 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、[作成] 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。

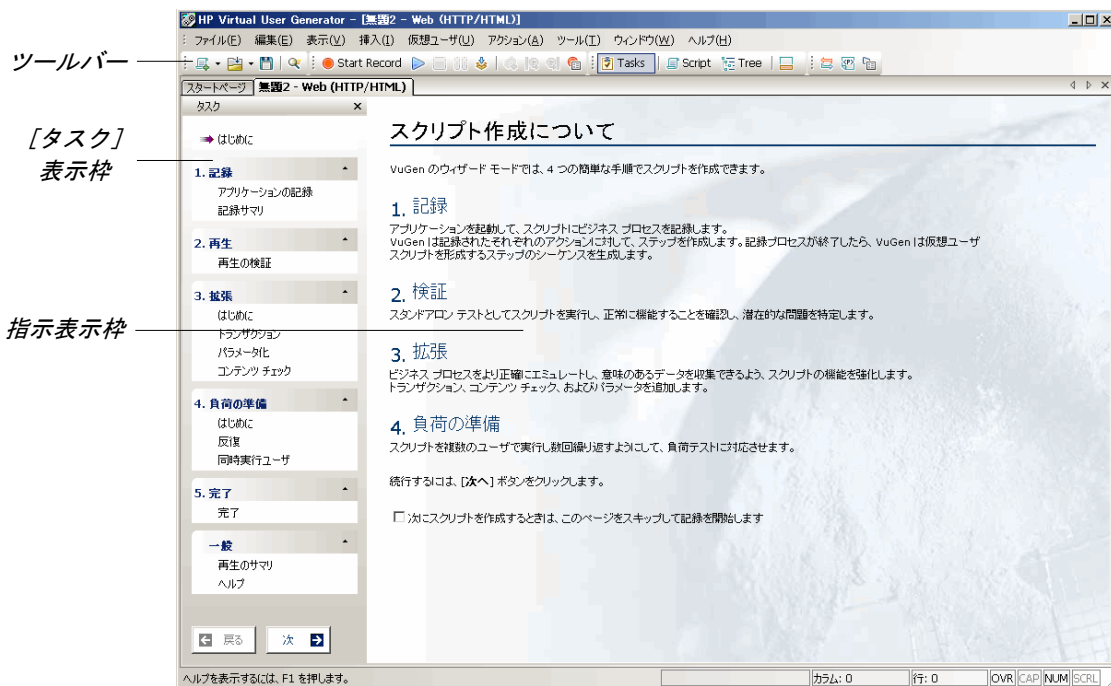
VuGen のウィザード・モードの使用方法



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 ウィザードに戻ることができます。

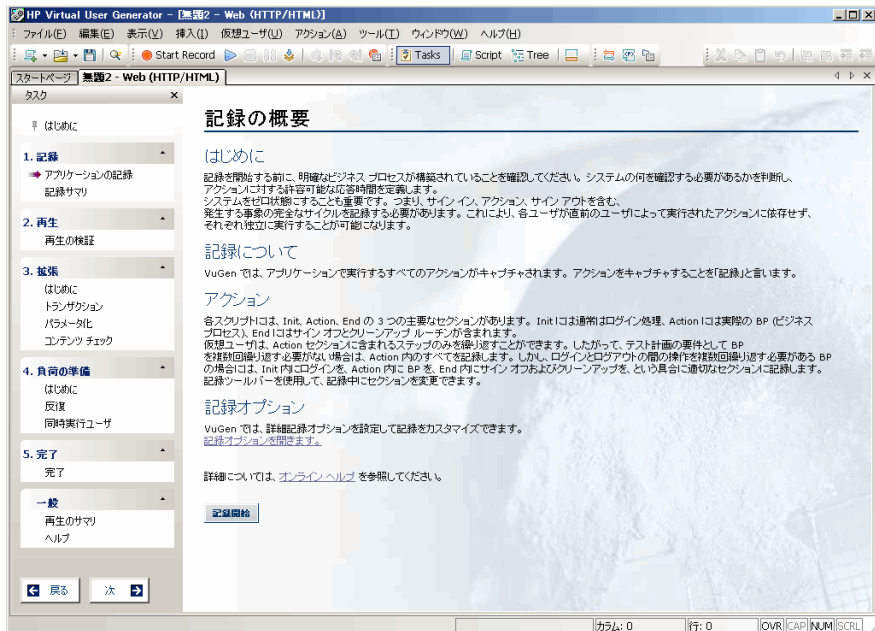
ビジネス・プロセスの記録によるスクリプトの作成方法

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。前のセクションでは、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。このセクションでは、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、次の手順で行います。

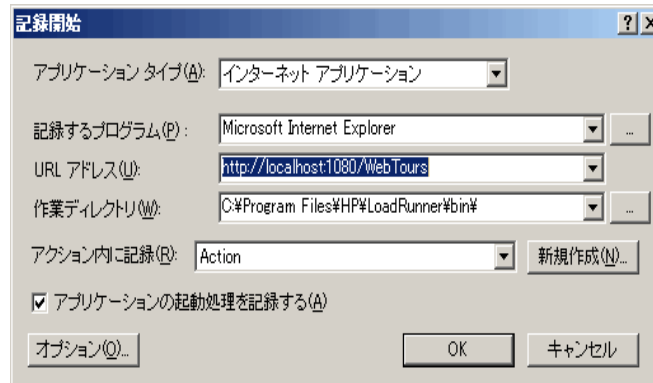
1 HP Web Tours Web サイトで記録を開始します。

- a [タスク] 表示枠で、[アプリケーションの記録] をクリックします。
- b 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、[仮想ユーザ] > [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。

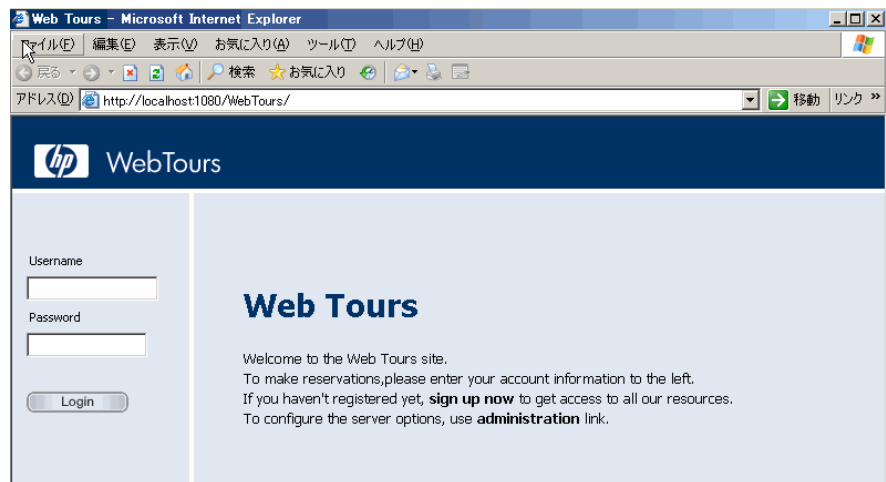
[記録開始] ダイアログ・ボックスが開きます。



c 次の操作を行います。

- ▶ [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 Web 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] > [すべてのプログラム] > [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 Web Tours Web サイトにログインします。

[ユーザー名] に「jojo」と入力し、[パスワード] に「bean」と入力します。
[ログイン] をクリックします。ようこそページが開きます。

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。次の情報を入力します。

- ▶ [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

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。[検索結果] 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「06/10」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 日程を確認します。

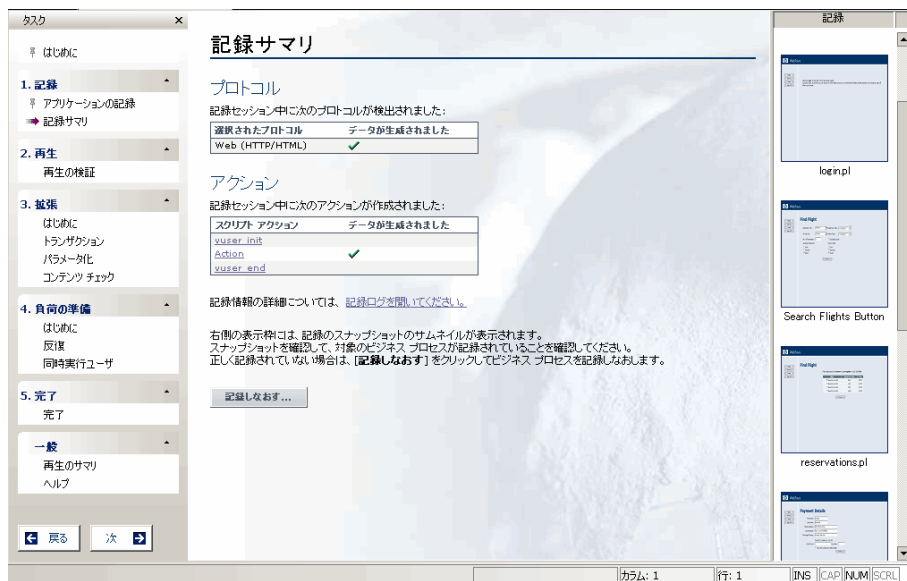
左側の表示枠で [Itinerary] をクリックします。[Itinerary] ページが開きます。

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、[コードの生成] ポップ・アップ・ウィンドウが開きます。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리를表示します（サマリが表示されていない場合は、[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、[上書き保存] ボタンをクリックし、<LoadRunner インストール先フォルダ>\scripts に移動して、Tutorial という名前の新しいフォルダを作成します。[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・バーにスクリプト名が表示されます。

スクリプトの表示方法

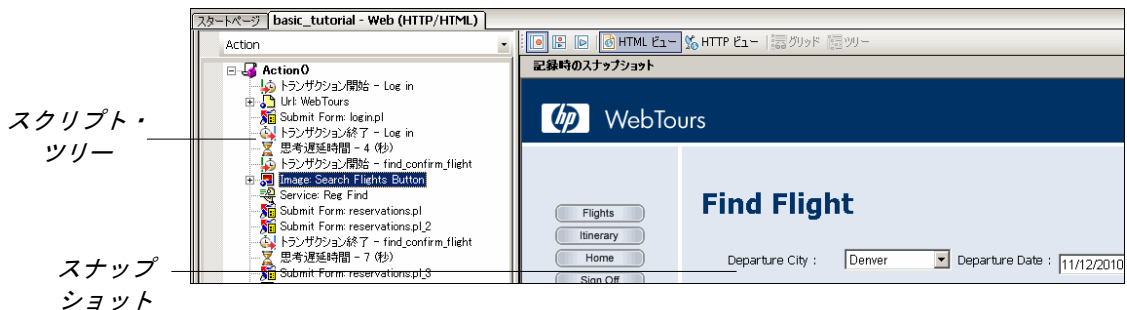
前のセクションでは、旅行代理店によるログイン、フライト予約、日程の確認、ログオフを記録しました。VuGen には、[記録開始] ボタンをクリックしてから [停止] ボタンをクリックするまでのステップが記録されました。

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] 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。



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動作を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、実際のユーザの動作をより正確に負荷テストに反映するメカニズムです。

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] 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("WebTours",
        "URL=http://localhost:1080/WebTours",
        "Resource=0",
        "ContentType=text/html",
        "Referer=",
        "Snapshot=t1.inf",
        "Mode=HTML",
        LAST);

    lr_think_time(87);

    web_submit_form("login.pl",
        "Snapshot=t2.inf",
        ITEMDATA,
        "Name=username", "Value=jjj", ENDITEM,
        "Name=password", "Value=bean", ENDITEM,
        "Name=login.x", "Value=54", ENDITEM,
        "Name=login.y", "Value=12", ENDITEM,
    );
}
    
```

カラム: 1 行: 1 INS CAP NUM

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ますが、**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。

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。これで、記録されたスクリプトを再生し、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できます。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、仮想ユーザの動作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。

- ▶ 39 ページの「実行時の動作の設定方法」
- ▶ 43 ページの「スクリプト実行のリアルタイムでの監視方法」
- ▶ 45 ページの「再生に関する情報の表示方法」
- ▶ 46 ページの「テストの成功の確認方法」
- ▶ 48 ページの「結果の検索またはフィルタ方法」

実行時の動作の設定方法

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。

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。説明する項目は次のとおりです。

- ▶ **[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。

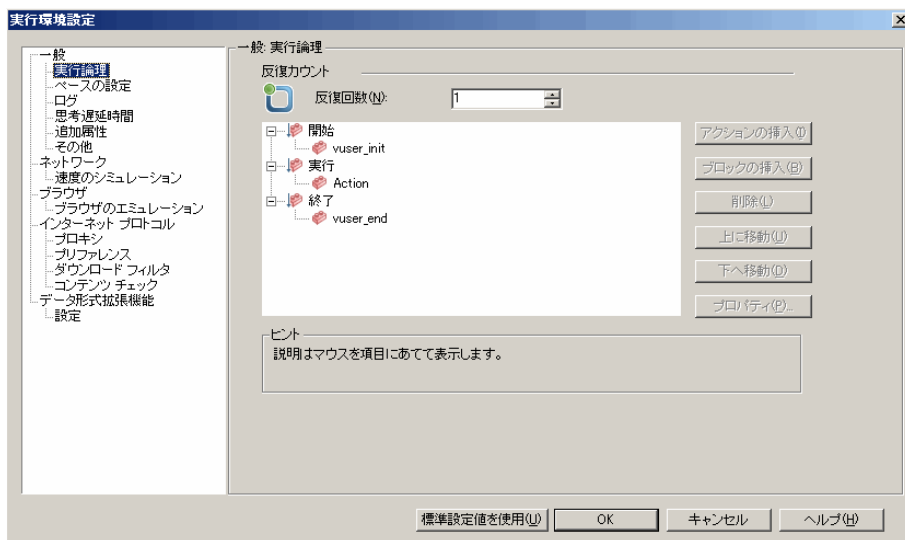
[タスク] 表示枠が表示されていることを確認します（表示されていない場合は [タスク] ボタンをクリックします）。[タスク] 表示枠の **[再生の検証]** をクリックします。

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



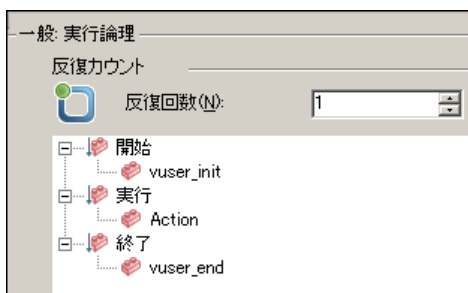
あるいは、F4 キーを押すか、ツールバーの **[実行環境設定]** ボタンをクリックします。

[実行環境設定] ダイアログ・ボックスが開きます。



2 実行論理を設定します。

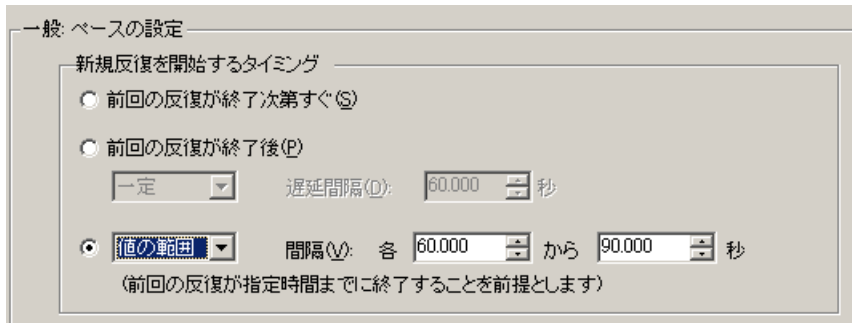
左側の表示枠で [実行論理] ノードを選択します。



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。

左側の表示枠で [ペースの設定] ノードを選択します。



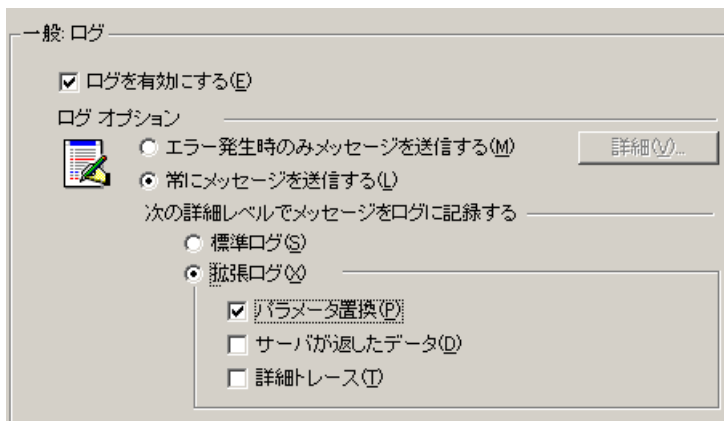
このノードでは、反復の間隔を制御できます。ランダムな間隔を指定します。これにより、動作間でユーザが待機する現実の設定が正確にエミュレートされます。ランダムな間隔では、反復間で正確に 60 秒待機する実際のユーザはありません。

3 番目のラジオ・ボタンを選択し、次のように選択します。

[値の範囲] 間隔：各 60.000 秒から 90.000 秒

4 ログを設定します。

左側の表示枠で [ログ] ノードを選択します。

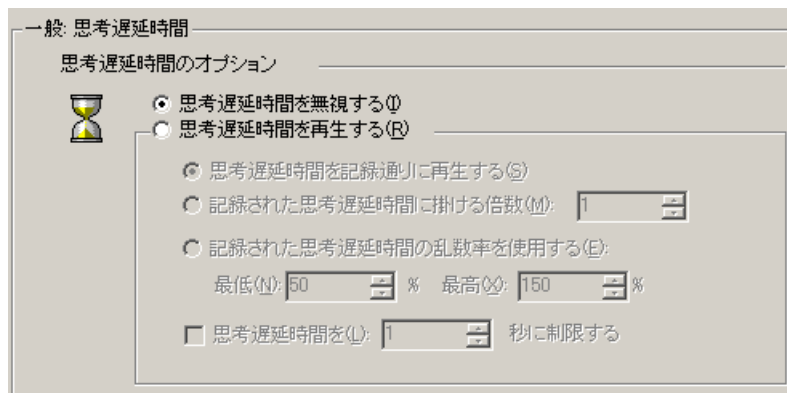


[ログ] 設定では、テストの実行時にログに記録する情報量を指定します。開発中には、デバッグのためにいくつかのログを有効にできます。ただし、スクリプトが正しく動作することを確認したら、エラー・ログのみを有効にするか、ログを無効にします。

[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。

左側の表示枠で [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。

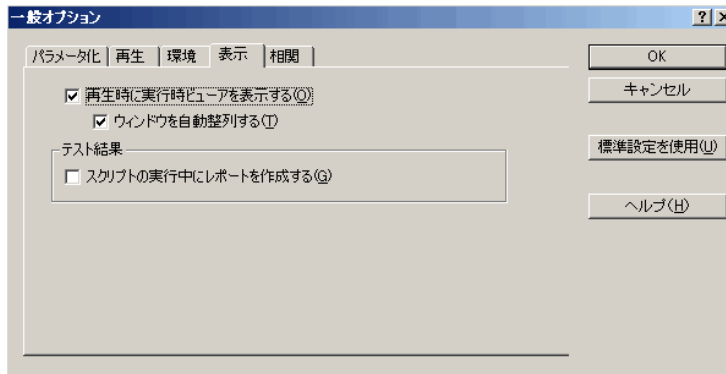
6 [OK] をクリックして、[実行環境設定] ダイアログ・ボックスを閉じます。

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。ただし、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。

- 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 [再生時に実行時ビューアを表示する] 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] ダイアログ・ボックスを閉じます。



- 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、結果ファイルの保存先の入力を求められたら、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。
- 6 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

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。
 [再生の検証] をクリックし、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
 反復 2 で使用する再生サムネイル
 注: 複数の反復の場合、VuGen では vuser_init セクションのサムネイルは表示されません。

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...

記録 再生

reservations.pl_2 reservations.pl_2

reservations.pl_3 reservations.pl_3

Itinerary Button Itinerary Button

SignOff Button SignOff Button

カラム: 1 行: 1 INS CAP NUM SCRL

[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。

このセクションでは、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、次の手順で行います。



- 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。

```
仮想ユーザ スクリプトが 2010-11-12 15:21:11 で開始されました
アクション vuser_init を開始します。
LoadRunner 11.0.0 for WIN2003 の Web Turbo 再生; ビルド 8859 (Aug 18 2010 20:14:31) [MsgId: MMSG-27143]
Run Mode: HTML [MsgId: MMSG-26000]
実行環境設定ファイル: "C:\Program Files\HP\LoadRunner\scripts\basic_tutorial\default.cfg" [MsgId: MMSG-27141]
アクション vuser_init を終了します。
Vuser を実行します...
反復 1 を開始します。
アクション Action を開始します。
Action.c(4): リダイレクト "http://localhost:1080/WebTours/" (リダイレクトの深さは 0 です) [MsgId: MMSG-26694]
Action.c(4): 場所: "http://localhost:1080/WebTours/" [MsgId: MMSG-26693]
Action.c(4): "http://localhost:1080/WebTours/" 中に URL コードがある "http://localhost:1080/WebTours/headers.htm" を検知しました
```

- 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。
 - ▶ 「開始されました」および「終了しました」: 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（オレンジ色の文字）。

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、スクリプト・ビュー内のステップが強調表示されます。

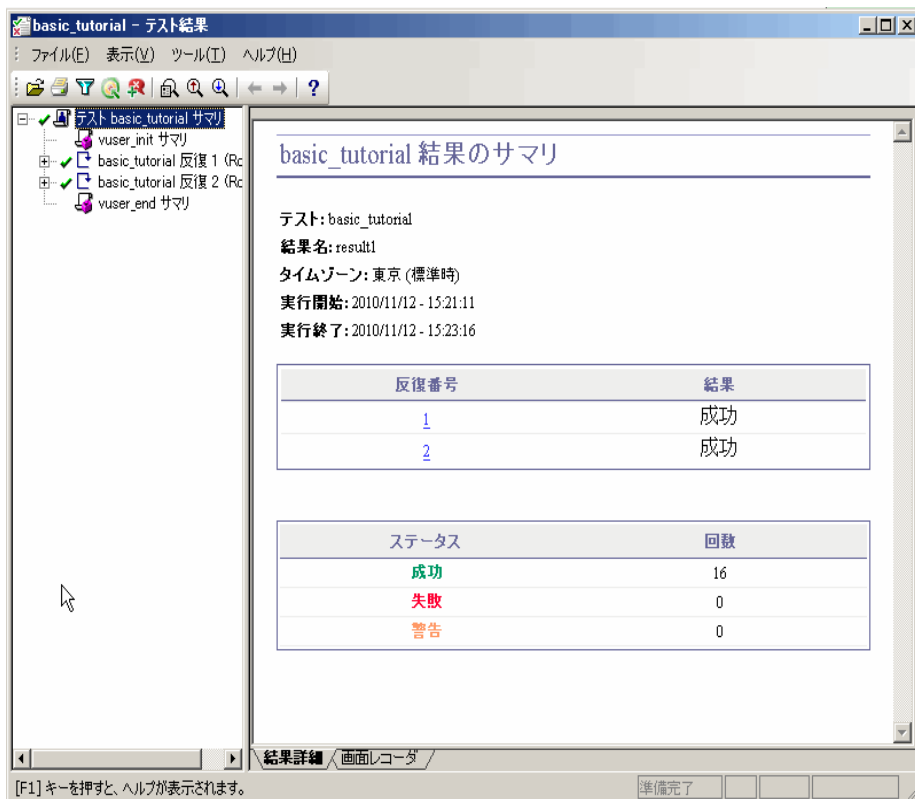
テストの成功の確認方法

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。

このセクションでは、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、次の手順で行います。

- 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。[テスト結果] ウィンドウが開きます。



[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。

[サマリ] 表示枠の最上部の表には、成功した反復と失敗した反復が示されます。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示されます。これらの機能は、本チュートリアル後の部分でテストに追加します。

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、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。

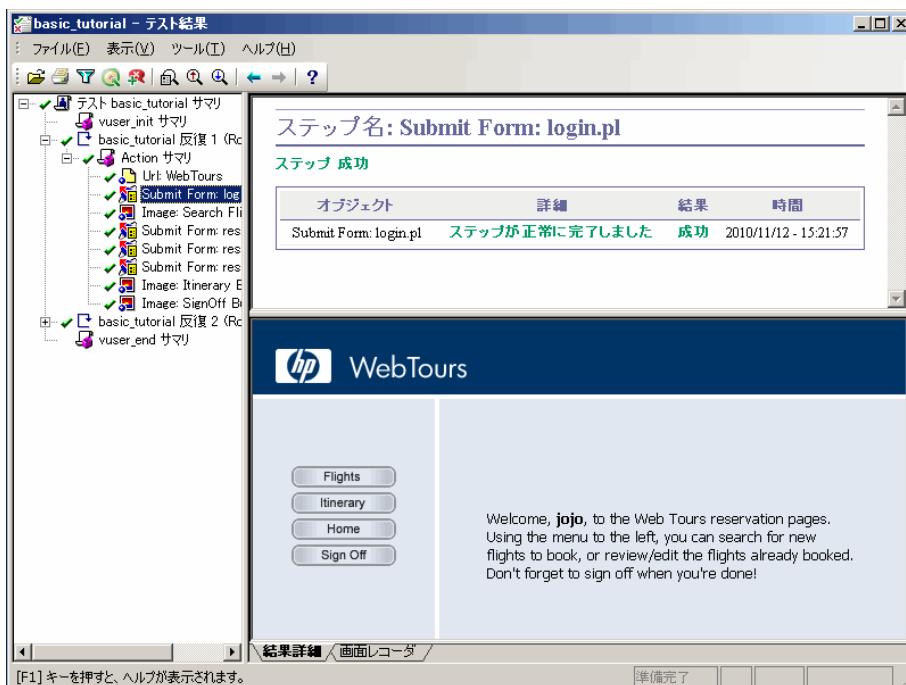
[ツリー] 表示枠で、テスト・ツリーを展開して、各ステップの結果を個別に表示できます。[サマリ] 表示枠には、その反復時の再生のスナップショットが表示されます。

1 ツリーで反復ノードを展開します。

左側の表示枠のプラス (+) 記号をクリックして [basic_tutorial 反復 1] ノードを展開し、次に [Action サマリ] 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] ノードを選択します。[サマリ] 表示枠には、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] 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。

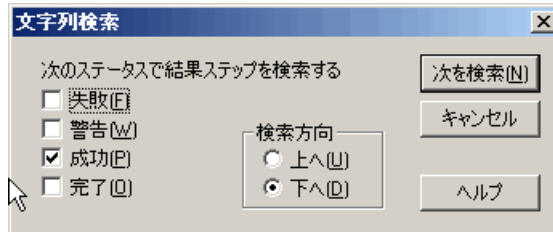
4 結果ステータスを検索します。

成功または**失敗**という単語がないかテスト結果を検索できます。

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に、失敗した箇所を特定するのに役立ちます。



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] 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] 表示枠では、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。

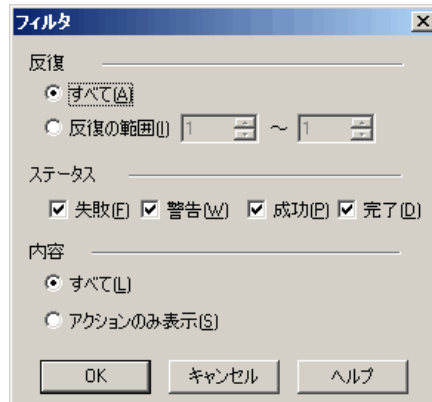
注： 選択したステータスのステップが見つからない場合、ステップは強調表示されません。

5 結果にフィルタを適用します。

[テスト ツリー] 表示枠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] を選択します。

次の練習について



HP Web Tours Application 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。

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。

- ▶ 54 ページの「再生エラーのための HP Web Tours の準備」
- ▶ 54 ページの「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 Application 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 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 Web Tours を起動します。

[スタート] > [すべてのプログラム] > [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] というタイトル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。
- c 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法

変更後の HP Web 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関連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 (32 ページの「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` という名前で作成します。

2 新しい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	説明
 Action.c(33)	-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 相関のためにスクリプトをスキャンします。

指示表示枠の見出し「動的サーバの値」の下にある「**動的なサーバ値の表示および解決**」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 ("WCSParam_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、「次の 2 つの文字列の間にあるデータのサーバ応答をチェックする」ということです。

▶ 左の境界 : **userSession value=**

▶ 右の境界 : **>**

ステートメントにより、VuGen はこのデータの最初の出現を **WCSParam_Diff1** 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。

[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 **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] > [すべてのプログラム] > [HP LoadRunner] > [Samples] > [Web] > [HP Web Tours Application] を選択して、HP Web Tours を開きます。HP Web Tours ホーム・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 ユーザーズ・ガイド』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。

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。

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。

- ▶ 60 ページの「ビジネス・プロセスの測定方法」
- ▶ 63 ページの「複数のユーザをエミュレートする方法」
- ▶ 68 ページの「Web ページの内容の確認方法」
- ▶ 69 ページの「デバッグ情報の生成方法」
- ▶ 71 ページの「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動で挿入できます。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。

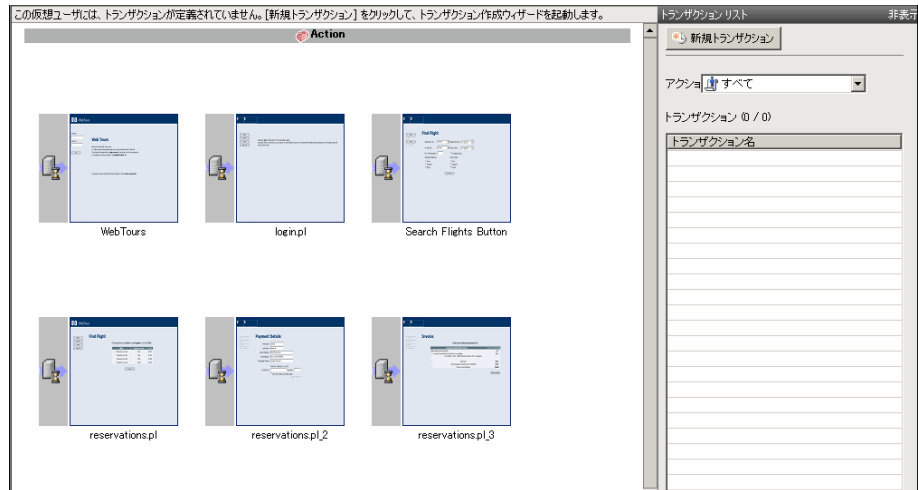
このセクションでは、ユーザがフライトを検索して確定するまでに要する時間を計測するためのトランザクションを、スクリプトに挿入します。

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、次の手順で行います。

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。[新規トランザクション] 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。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。

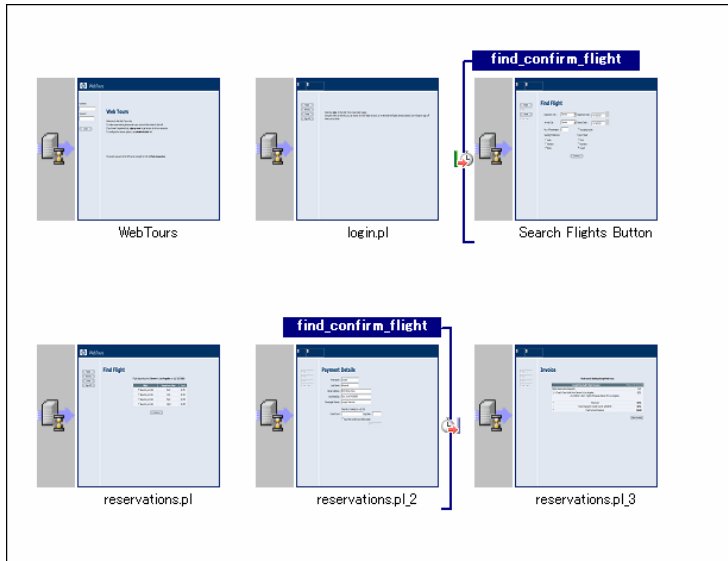
2 トランザクション開始マーカとトランザクション終了マーカを挿入します。

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トランザクション開始の括弧を持っていき、クリックして固定します。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前にトランザクション終了の括弧を持っていき、クリックして固定します。

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。



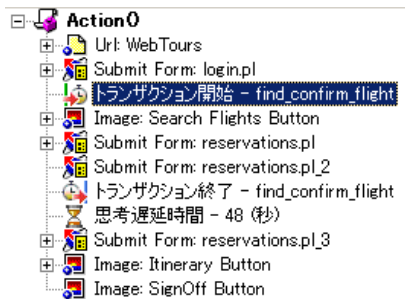
これで、新しいトランザクションが作成されました。トランザクションの開始点または終了点を調整するには、各マーカをスクリプト内の別の位置にドラッグします。トランザクションの名前を変更することもできます。これには、トランザクション開始マーカの上にある既存の名前をクリックし、新しい名前を入力します。

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] 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。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、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、次の手順で行います。

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 Data : 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 固定値を可変値に変更します。

7 行目の **seatPref** 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。

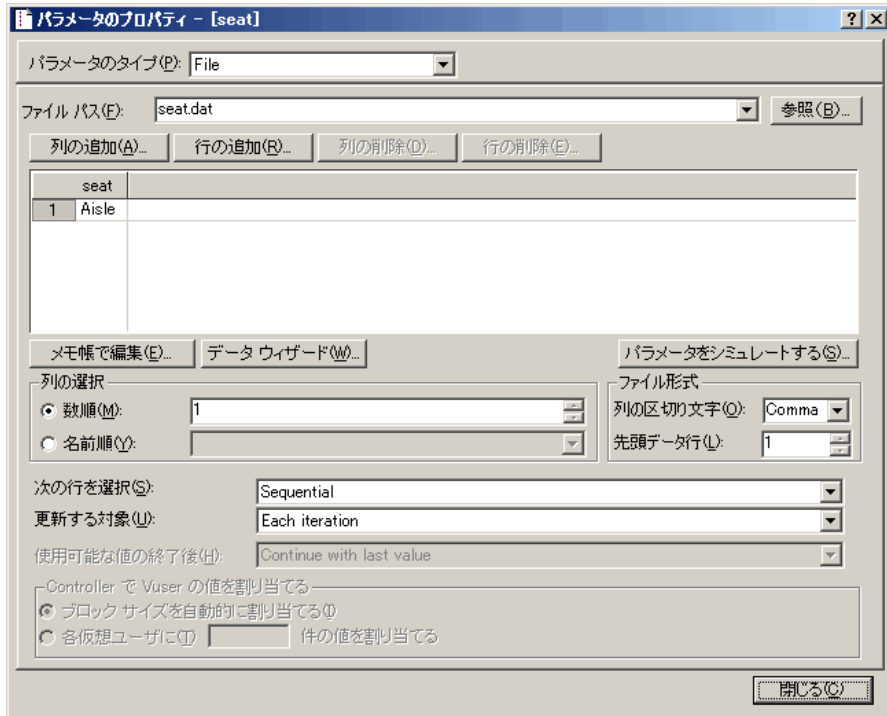


3 パラメータを作成します。



[**パラメータ名**] 編集フィールドに、「seat.」と入力します。[**ファイル**] パラメータ・タイプを受け入れます。[**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、[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 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。



注： 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 に指示します。

- ▶ [**次の行を選択**] : [Sequential] (順次)
- ▶ [**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。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。

テキスト チェック

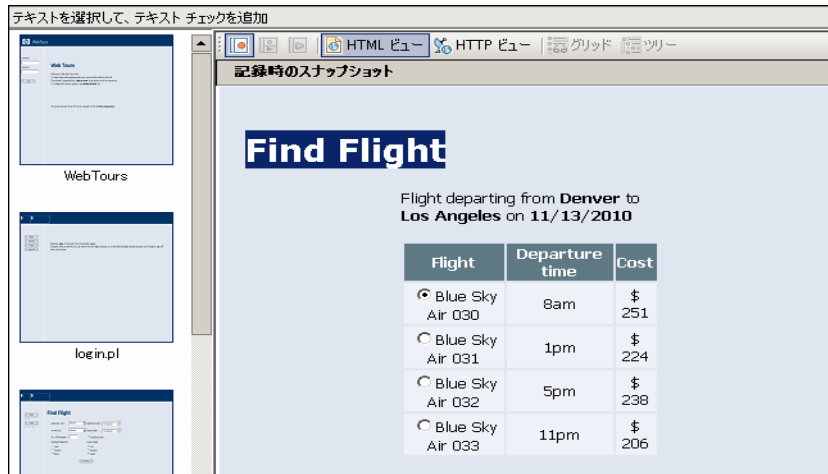
このセクションでは、スクリプトの予約ページに「Find Flight」という語句が表示されるかを検証するテキスト・チェックを追加します。

テキスト・チェックを挿入するには、次の手順で行います。

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・ウィザードが開き、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。



ツールバーの [HTML View] 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。

スナップショット内で単語「Find Flight」を強調表示し、右クリックして [テキスト チェックを追加- (web_reg_find)] を選択します。

[文字列の検索] ダイアログ・ボックスが開き、[特定の文字列を検索する] 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、テキストがあるかどうかを検証します。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。

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。例については、『HP LoadRunner オンライン関数リファレンス』を参照してください。

このセクションでは、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、次の手順で行います。

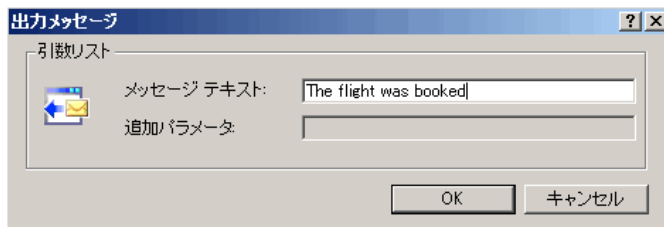
1 位置を選択します。

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

このセクションでは、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**] 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**] 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。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。

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。

```
"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。

[再生ログ] タブで、CTRL + F キーを押して [検索] ダイアログ・ボックスを開きます。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。

[再生ログ] タブで、CTRL + F キーを押して [検索] ダイアログ・ボックスを開きます。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。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、[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 7、「負荷テスト・シナリオの作成」に進みます。

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。このテストを設計して実行するには、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。

- ▶ 74 ページの「LoadRunner Controller の概要」
- ▶ 74 ページの「Controller の開始方法」
- ▶ 77 ページの「Controller ウィンドウの概要」
- ▶ 78 ページの「スクリプトの詳細の変更方法」
- ▶ 79 ページの「大きな負荷を生成する方法」
- ▶ 80 ページの「実際の負荷動作のエミュレート方法」
- ▶ 85 ページの「さまざまなタイプのユーザのエミュレート方法」
- ▶ 88 ページの「負荷下のシステムの監視方法」

LoadRunner Controller の概要

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フライト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動作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。

Controller の開始方法

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。

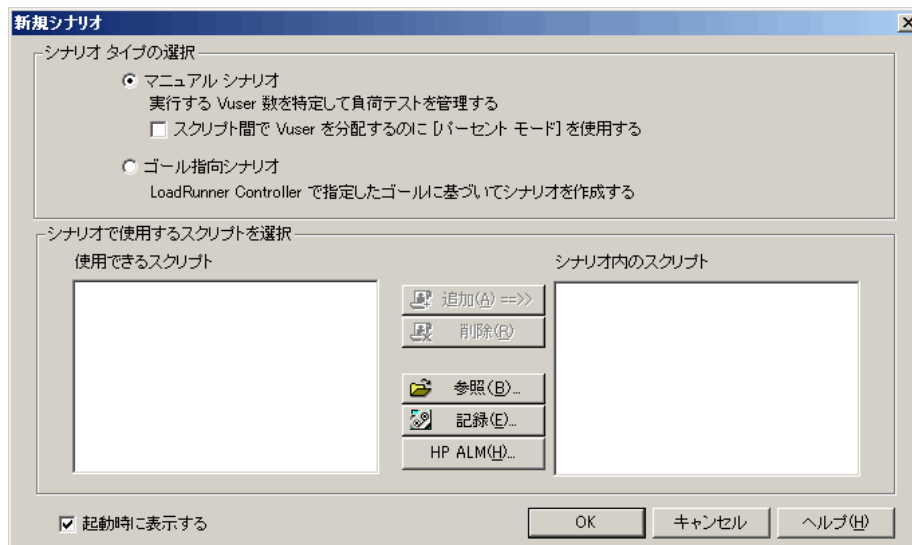
1 HP LoadRunner を起動します。

[スタート] > [すべてのプログラム] > [HP LoadRunner] > [LoadRunner]
を選択します。[HP LoadRunner 11.00] ウィンドウが開きます。

2 Controller を起動します。

[LoadRunner ランチャー] 表示枠で、[**負荷テストの実行**] をクリックします。HP LoadRunner Controller が起動します。

標準設定では、Controller が [新規シナリオ] ダイアログ・ボックスとともに表示されます。



3 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。

- ▶ 「**マニユアル・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。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。

[**マニュアル シナリオ**] を選択します。

4 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実際のシナリオをエミュレートするには、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、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。

Controller ウィンドウの概要

Controller ウィンドウの **[デザイン]** タブは次の3つの主要なセクションに分けられます。

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

[シナリオ
グループ]
表示枠 (マ
ニュアル・
シナリオ)

SLA

[シナリオの
スケジュール]
表示枠
(マニュアル・
シナリオ)

[デザイン]
タブ

- ▶ **[シナリオ グループ] 表示枠**： [シナリオ グループ] 表示枠では、仮想ユーザ・グループを設定します。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。
- ▶ **[サービス レベル アグリーメント] 表示枠**： 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**サービス・レベル・アグリーメント (SLA)**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。SLAについては練習10でさらに学びます。

- ▶ **[シナリオのスケジュール] 表示枠**：[シナリオのスケジュール] 表示枠では、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、および負荷の停止方法を定義します。

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] 表示枠** 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。

シナリオ グループ			
グループ名	スクリプトのパス	数	Load Generators
<input checked="" type="checkbox"/> basic_script	C:\Program Files\HP\LoadRunner\tutorial\basic_script	10	localhost

- 2 **[グループ名]** を変更します。



スクリプトを選択し、**[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。

グループ情報 ✕

グループ名:

Load Generator 名:

スクリプト

名前: basic_script

パス: C:\Program Files\HP\LoadRunner\tutorial\basic_script

種類: Multi+QWeb

[OK] をクリックします。[デザイン] タブの [シナリオ グループ] 表示枠に名前が表示されます。

大きな負荷を生成する方法

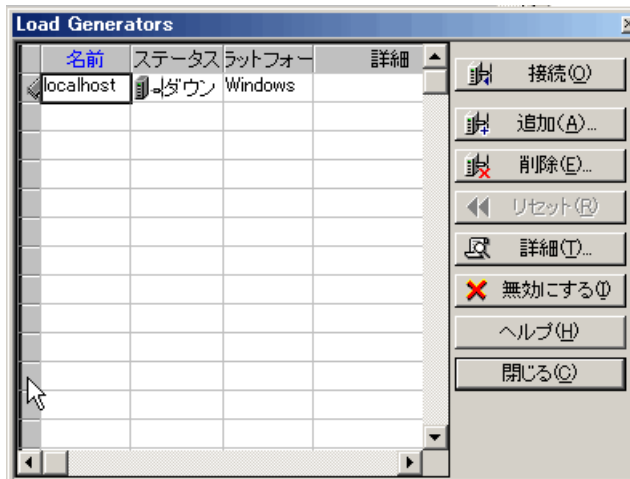
スクリプトの追加後、負荷を生成するマシンを設定します。

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 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。このセクションでは、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。



[デザイン] タブで、[Load Generators] ボタンをクリックします。[Load Generators] ダイアログ・ボックスが開き、**localhost** という名前の Load Generator 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 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。

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。

localhost 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「**準備完了**」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 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停止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 ウィンドウの **[シナリオのスケジュール]** 表示枠でマニュアル・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]** 表示枠は、**[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。

[スケジュールの定義] 領域の中で, [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] 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。

[アクション] 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, [アクション] グリッドのプロパティは必要に応じて調整されます。

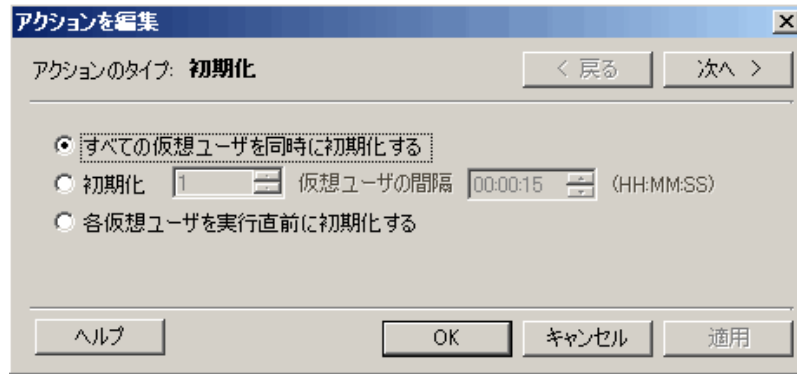
定義を設定し, [アクション] グリッドが次のようになるようにします。

全体スケジュール	
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
仮想ユーザの停止	すべて 仮想ユーザを 2 ずつ 00:00:30 (HH:MM:SS) ごとに停...
*	

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。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。

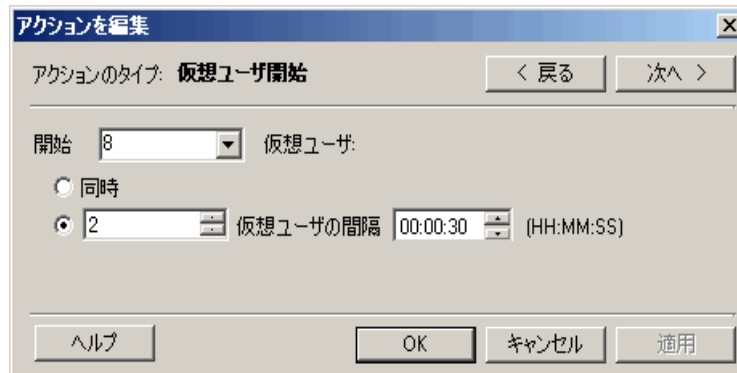


b 段階的な開始を指定します ([スケジュール アクション] グリッドから)。

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 仮想ユーザ] ボックスに 8 仮想ユーザを指定し、2 番目のオプション [開始 2 仮想ユーザの間隔 00:00:30] (30 秒) を選択します。



c 期間をスケジュールします（対話式スケジュール・グラフから）。

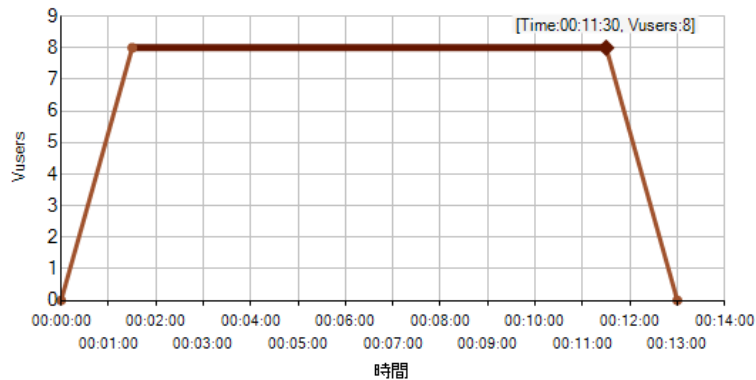
期間を指定し、仮想ユーザがその指定の期間の間、スケジュール・アクションの実行を必ず続行するようにし、サーバにかかる連続的な負荷を測定できるようにします。期間を設定した場合には、スクリプトの実行環境の設定の中で設定した反復数に関係なく、スクリプトはその期間の間、必要な回数だけ何回でも反復して実行されます。



[対話式スケジュール グラフ] 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。

[アクション] グリッドで、[実行時間] をクリックするか、グラフ内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。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。

対話式スケジュール グラフ

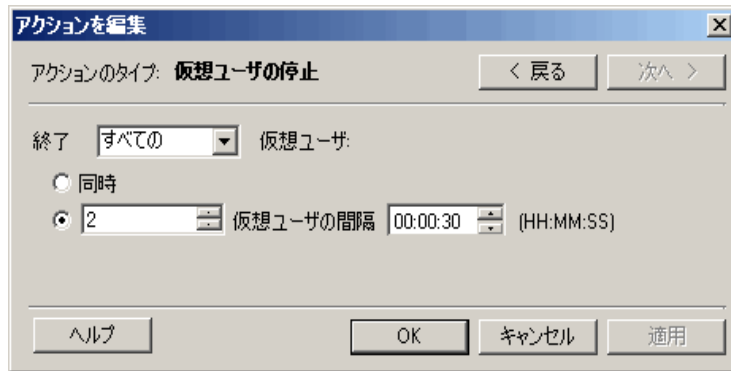


注： ひし形の前に凡例が表示されている場合、[凡例を非表示] ボタンをクリックして、ひし形を表示します。

d 段階的な終了をスケジュールします（[スケジュール アクション] グリッドから）。

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。

[アクション] 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 番目のオプションを選択し、「**終了 2 仮想ユーザの間隔 00:00:30（30 秒）**」と入力します。



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。

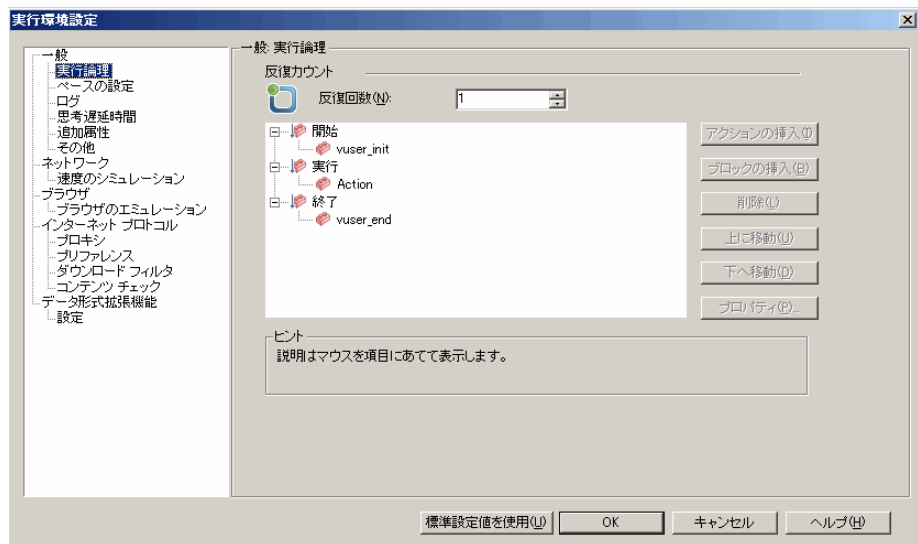
実際のユーザをエミュレートするときは、ユーザの実際の動作を考慮する必要があります。動作には、ユーザがアクションの合間に一時停止する時間や、ユーザがアクションを繰り返す回数などがあります。

このセクションでは、LoadRunner の実行環境の設定について学び、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] タブでスクリプトを選択し、[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・ボックスが表示されます。



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す。説明する項目は次のとおりです。

- ▶ **[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。

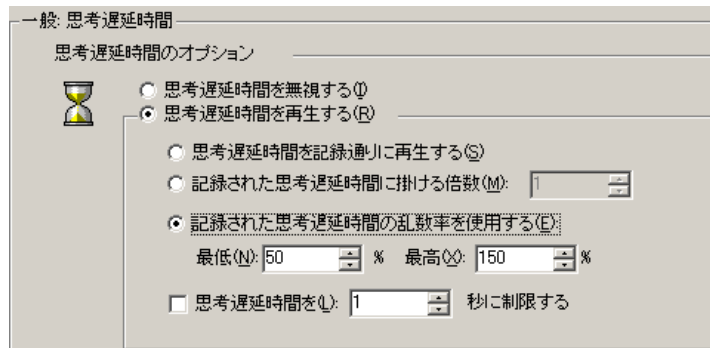
- ▶ **[コンテンツ チェック]**：ユーザ定義のエラーを自動的に検出します。

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。

実行環境の設定の「**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。

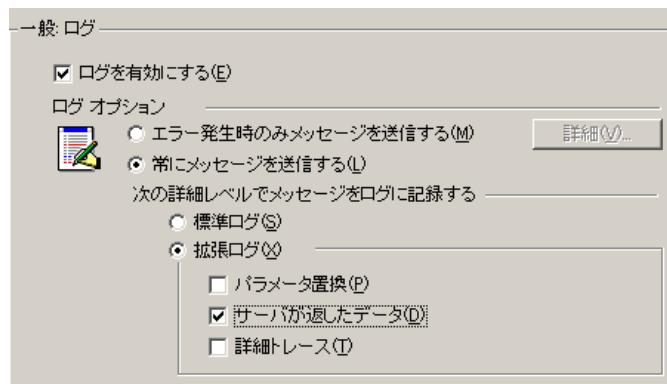
[一般：思考遅延時間] ノードを選択します。[思考遅延時間を再生する] を選択し、[記録された思考遅延時間の乱数率を使用する] オプションを選択します。最小値として 50% を指定し、最大値として 150% を指定します。



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、[サーバから返したデータ] を選択します。



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。

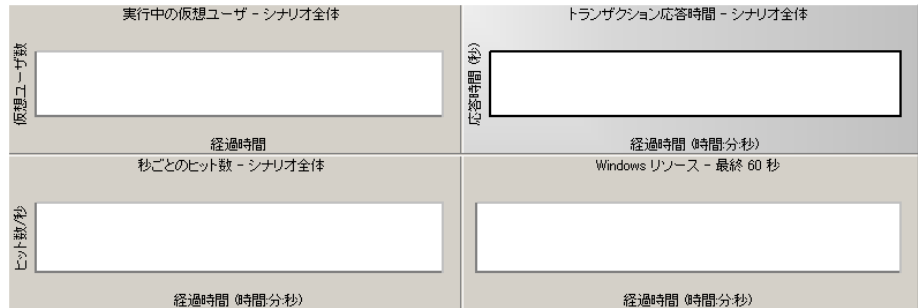
たとえば、稼働中の Web サーバのタイプに応じた Web サーバ・リソース・モニタを選択できます。また、たとえば IIS など関連するモニタ用のライセンスを購入し、そのモニタを使用して、IIS リソースに反映された問題を特定できます。

このセクションでは、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。

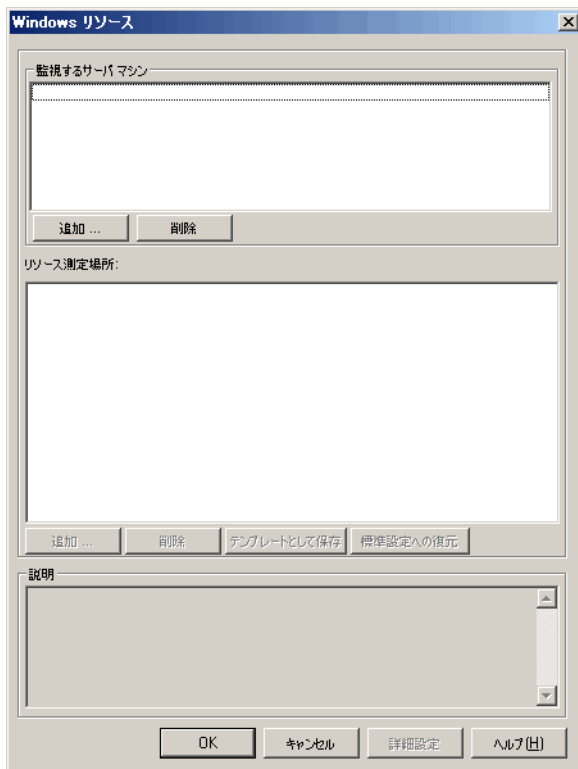
[Controller] ウィンドウで [実行] タブをクリックし、実行ビューを開きます。

[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。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。



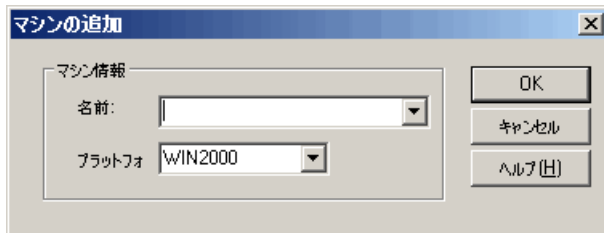
第7章・負荷テスト・シナリオの作成

[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、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。

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。

- ▶ 94 ページの「Controller の実行ビューの概要」
- ▶ 96 ページの「負荷テスト・シナリオの実行方法」
- ▶ 98 ページの「負荷下のアプリケーションの監視方法」
- ▶ 100 ページの「リアルタイムで実行しているユーザの監視方法」
- ▶ 101 ページの「ユーザ・アクションのサマリの表示場所」
- ▶ 102 ページの「テスト中に負荷を増やす方法」
- ▶ 103 ページの「負荷下のアプリケーションの動作方法」
- ▶ 104 ページの「アプリケーションでエラーが発生したかどうか」
- ▶ 106 ページの「テストの実行が完了したことを知る方法」
- ▶ 106 ページの「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] 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] 表示枠
- ▶ [シナリオ・ステータス] 表示枠
- ▶ [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 shows the HP LoadRunner Controller interface. On the left, there are labels pointing to different parts of the interface:

- [シナリオグループ] 表示枠**: Points to the Scenario Groups table.
- [シナリオステータス] 表示枠**: Points to the Scenario Status panel.
- 利用可能なグラフ**: Points to the Available Graphs tree.
- グラフ表示領域**: Points to the main graph display area.
- グラフの凡例**: Points to the legend table at the bottom.

The Scenario Groups table shows:

グループ名	保留中	初期化/準備	実行	リセット	成功	失敗	エラー	徐々に終了	終了中	中止
demo_scr	0	0	4	4	0	0	0	0	0	0

The Scenario Status panel shows:

- シナリオ ステータス: 実行中の Vuser
- 経過時間: 秒ごとのヒット数
- Vuser (U): 成功したトランザクション
- 実行/停止 (P): 失敗したトランザクション
- スケジューラ一時停止: エラー

The Available Graphs tree shows:

- Infrastructure Resource Graphs
 - Network Client
 - Runtime Graphs
 - Running Vusers
 - User Defined Data Points
 - Error Statistics
 - Vusers with Errors
 - Transaction Graphs
 - Trans Response Time
 - Trans/Sec (Passed)
 - Trans/Sec (Failed/Stopped)
 - Total Trans/Sec (Passed)
 - Web Resource Graphs
 - Hits per Second

The Graph Display Area shows four graphs:

- Running Vusers - シナリオ全体: Line graph showing the number of running users over time.
- Trans Response Time: Line graph showing response time in seconds over time.
- Hits per Second - シナリオ全体: Line graph showing hits per second over time.
- Windows Resource: Line graph showing Windows resource usage over time.

The Legend table at the bottom shows:

Color	Scale	Status	Max	Min	Avg	Std
Yellow	1	実行中	4.000	0.000	1.750	該当なし
Purple	1	準備完了	6.000	0.000	4.500	該当なし
Green	1	実行完了	0.000	0.000	0.000	該当なし
Red	1	エラー	0.000	0.000	0.000	該当なし

- ▶ **[シナリオグループ] 表示枠**: 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。
- ▶ **[シナリオステータス] 表示枠**: 右上の表示枠で、負荷テストのサマリーを表示できます。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、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] > [グラフを表示]）。
- ▶ **[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

このセクションでは、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] 表示枠の **[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	準備完了	実行	シデブ	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、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。

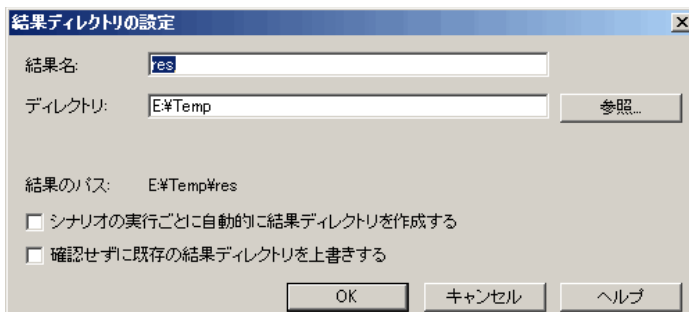
2 シナリオを開始します。



[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、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。

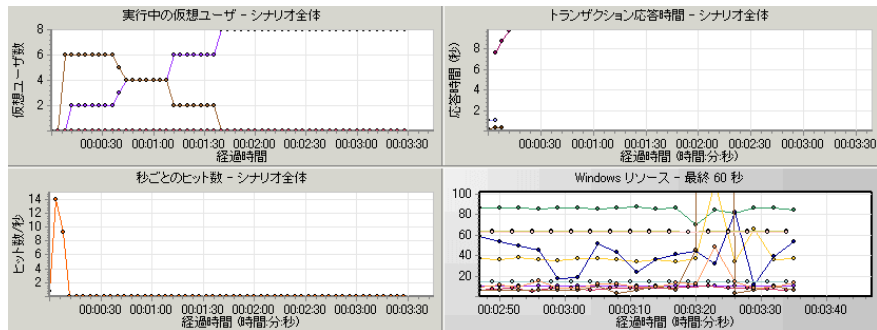
負荷下のアプリケーションの監視方法

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**：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。
- ▶ **[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。
- ▶ **[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。
- ▶ **[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。

[Windows リソース] グラフをダブルクリックし、拡大表示します。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。

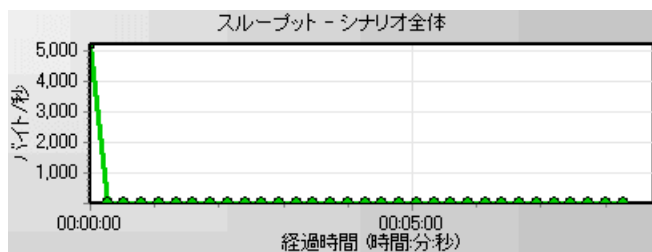
行を選択すると、対応するグラフの線が強調表示されます。また、その逆の場合も同様です。グラフを縮小するには、グラフをもう一度ダブルクリックします。

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。



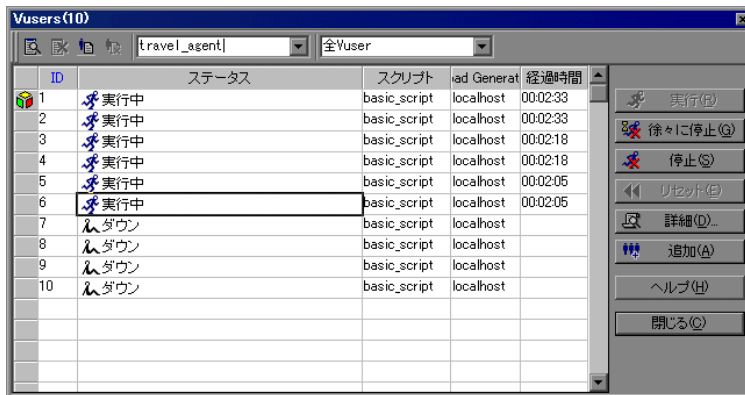
リアルタイムで実行しているユーザの監視方法

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、次の手順で行います。



- 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、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

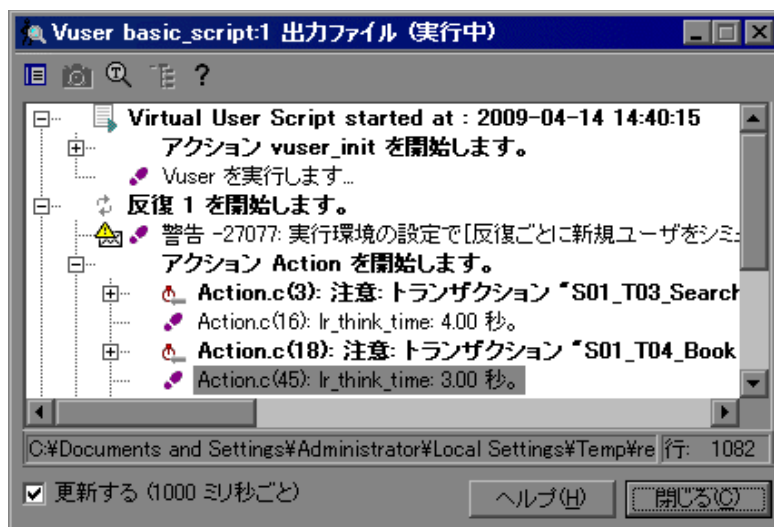
テスト実行中における個々の仮想ユーザの進行状況を検証するために、仮想ユーザのアクションのテキスト・サマリが格納されたログ・ファイルを表示できます。

イベントのテキスト・サマリを確認するには、次の手順で行います。



- 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、[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・ログ・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「Virtual User Script started」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。

- 2 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

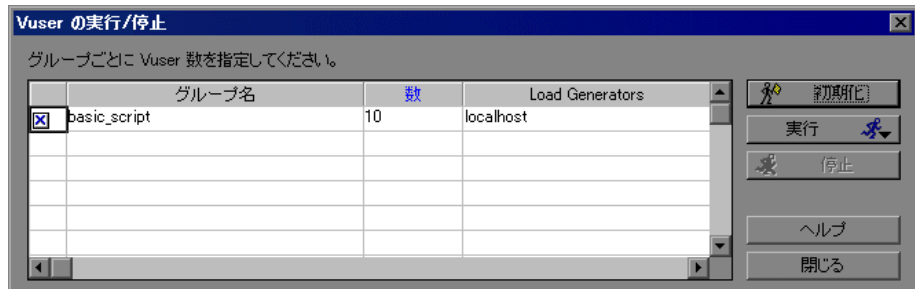
より多くのユーザを手動で追加することによって、負荷テストの実行中にアプリケーションにかかる負荷を増やすことができます。

負荷テスト中に負荷を増やすには、次の手順で行います。



- 1 実行ビューで [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] 表示枠ウィンドウには、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： 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・リソースが非常に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] 表示枠に実行中のシナリオのサマリが表示されているか確認し、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] 表示枠に、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	ダウン
実行中の Vuser	0
経過時間	00:02:20 (時間:分:秒)
秒ごとのヒット数	7.80 (最終 60 秒)
成功したトランザクション	251
失敗したトランザクション	0
エラー	10

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] 表示枠にある [成功したトランザクション] 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・ボックスが開きます。

名前	TPS	成功	失敗	中止
S01_T02_Login	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)

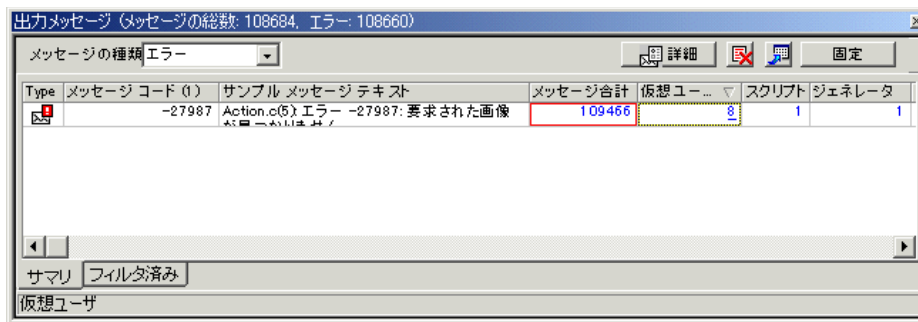
アプリケーションでエラーが発生したかどうか

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。

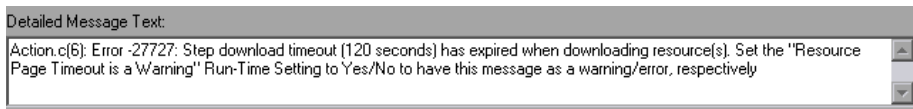
[表示] > [出力メッセージを表示] をクリックするか、[シナリオ ステータス] 表示枠にある [エラー] をクリックします。

[出力メッセージ] ダイアログ・ボックスが開き、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。



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ませんでした。



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。[出力カウインドウ] 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。



[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] 表示枠「**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] ダイアログ・ボックスで、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。

ID	ステータス	スクリプト	Load Generat	経過時間
1*	中止 反復試行回数: 1 成功数: 1	basic_script	localhost	00:00:18
2*	中止 反復試行回数: 1 成功数: 1	basic_script	localhost	00:00:18
3*	中止 反復試行回数: 1 成功数: 1	basic_script	localhost	00:00:17
4*	中止 反復試行回数: 1 成功数: 1	basic_script	localhost	00:00:17
5*	中止 反復試行回数: 1 成功数: 1	basic_script	localhost	00:00:01
6*	中止 反復試行回数: 2 成功数: 2	basic_script	localhost	00:00:32
7*	中止 反復試行回数: 2 成功数: 2	basic_script	localhost	00:00:33
8*	中止 反復試行回数: 2 成功数: 2	basic_script	localhost	00:00:32
9**	徐々に終了 実行中 状態から停止しました	basic_script	localhost	00:00:19

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、[上書き保存] 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。

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。この練習では、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。

- ▶ 108 ページの「使用するゴール・タイプ」
- ▶ 109 ページの「ゴール指向シナリオの作成方法」
- ▶ 110 ページの「Controller ウィンドウの概要（ゴール指向シナリオ）」
- ▶ 111 ページの「ゴールの定義方法」
- ▶ 113 ページの「負荷の動作の確認方法」
- ▶ 115 ページの「監視するオンライン・グラフ」
- ▶ 116 ページの「ゴール指向シナリオの実行方法」
- ▶ 119 ページの「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種類のゴールが用意されています。シナリオの目標とする、同時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、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**] 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**] の各ゴール・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**] 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

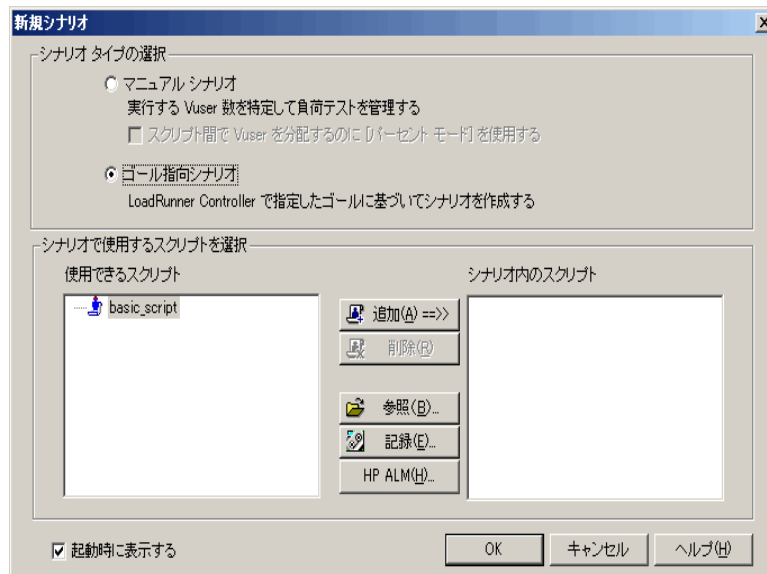
ユーザ・プロファイルの組み合わせを使用して現実のシステムをエミュレートするには、いくつかのスクリプトをシナリオに割り当てるとともに、スクリプトに負荷の割合を割り当てます。割合は期待負荷に従って設定する必要があります。

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・グループをエミュレートします。

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] 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。

- ▶ [シナリオのスクリプト] 表示枠
- ▶ [サービス・レベル・アグリーメント] 表示枠
- ▶ [シナリオのゴール] 表示枠

[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：この表示枠では、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 を確認します。シナリオの設定はここでを行います。

[サービス レベル アグリーメント] 表示枠：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**サービス・レベル・アグリーメント (SLA)** 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。

[シナリオのゴール] 表示枠：下部の表示枠には、テスト目標、ゴールを達成するために使用するユーザ数、シナリオの実行時間、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。

ゴールの定義方法

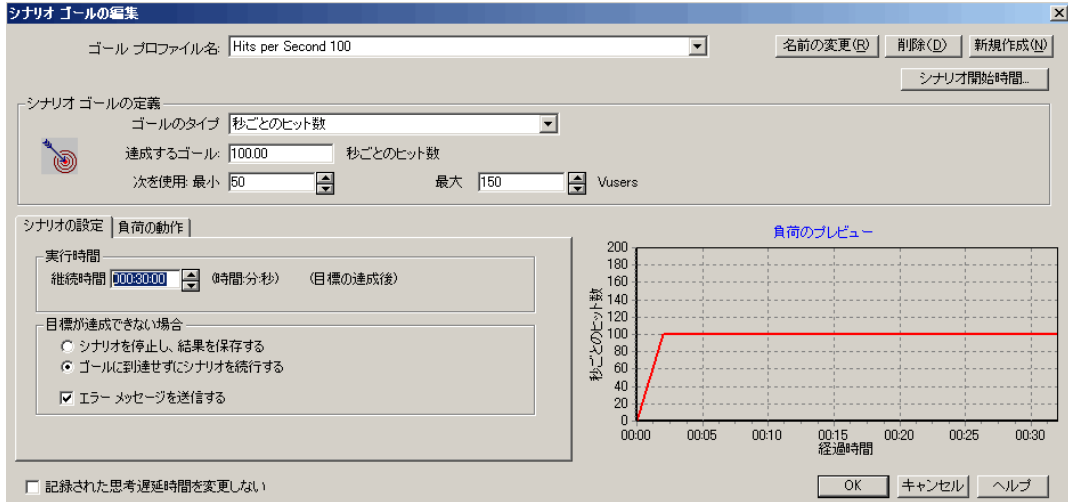
実行するスクリプトを選択したので、次は達成するゴールを定義する必要があります。

このセクションでは、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。



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」）を入力して、[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」と入力します。
- 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」と入力します。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

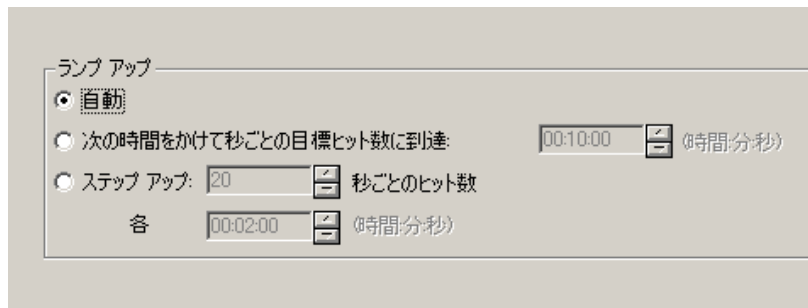
テストの目標を定義したので、次は Controller にいつどのように目標を達成させるかを指定する必要があります。

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、次の手順で行います。

1 仮想ユーザを同時に実行するようテストを設定します。

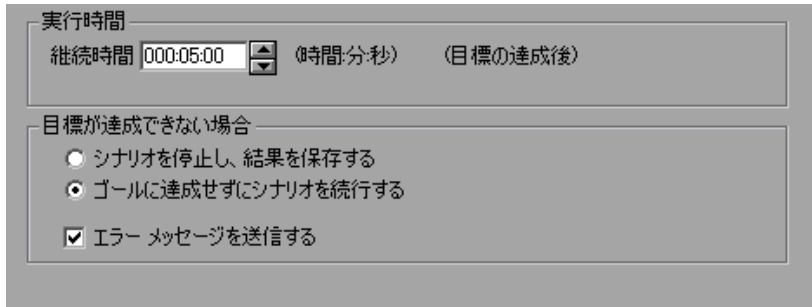
[シナリオ ゴールの編集] ダイアログ・ボックスで、[負荷の動作] タブを選択して、[自動] を選択します。



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

2 シナリオの設定を定義します。

[シナリオの設定] タブで、目標達成後にテストを継続する時間を 000:05:00 (5 分) と指定し、[**ゴールに到達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
- ゴールに到達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。

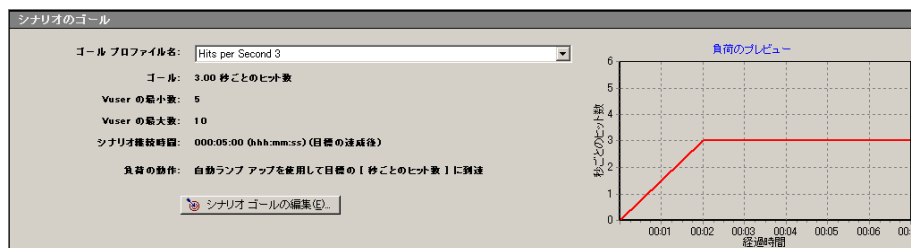
[シナリオ ゴールの編集] ダイアログ・ボックスの左下角で、[**記録された思考遅延時間を変更しない**] がクリアされていることを確認します。

このオプションを選択すると、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いけ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。



5 テスト用の Load Generator を定義します。

スクリプトを実行できるようにするには, 79 ページの「大きな負荷を生成する方法」の手順に従って, Load Generator をシナリオに追加します。

監視するオンライン・グラフ

テストのゴール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。このセクションでは、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 [Controller] 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。

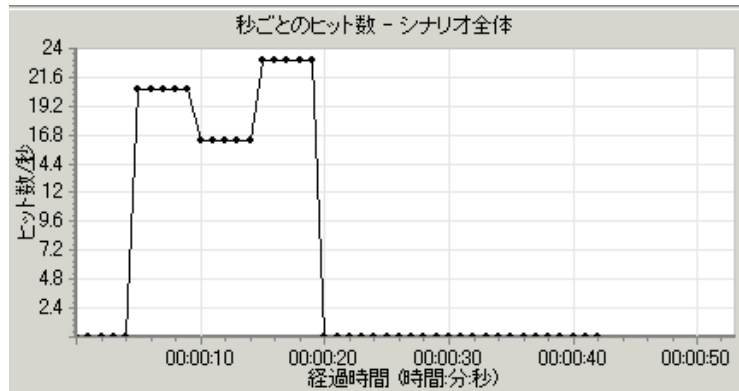
Controller がシナリオを開始します。

5 個の仮想ユーザが初期化されて実行が開始されます。LoadRunner は、必要なゴールである、秒ごとのヒット数 3 個を生成しようとします。

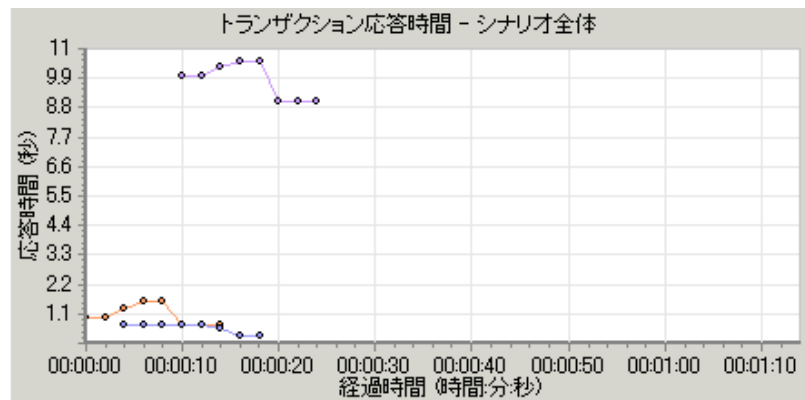
テスト中、Controller は仮想ユーザを自動的に開始、終了して、指定されたゴールを保持します。

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。

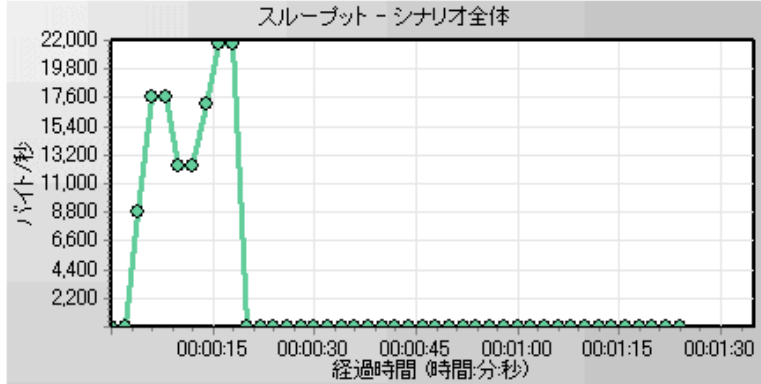


【トランザクション応答時間】 グラフには、各トランザクションが完了するのに要した時間が表示されます。トランザクション応答時間を監視して、サーバに負荷がかかっている状態で顧客への応答に要した時間を確認することはとても重要です。

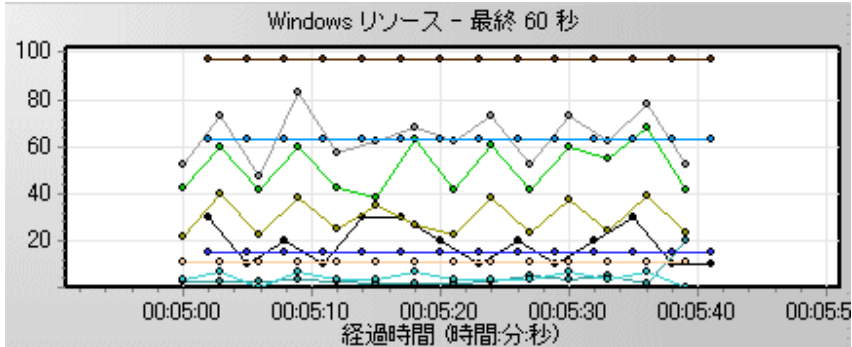


第9章・高度なゴール指向シナリオ

[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することで、[スループット] グラフを表示することもできます。[スループット] グラフは、秒ごとに仮想ユーザが Web サーバから受信する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目標の 3 個に到達しない場合、LoadRunner は、定義され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にシナリオの設定を保存します。シナリオを保存するには、[ファイル] > [保存] を選択するか [上書き保存] ボタンをクリックして、[シナリオを保存] ダイアログ・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。

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。

- ▶ 122 ページの「**Analysis** セッションの処理方法」
- ▶ 123 ページの「**Analysis** セッションの開始方法」
- ▶ 124 ページの「**Analysis** のウィンドウの概要」
- ▶ 125 ページの「ゴールの達成の確認 (サービス・レベル・アグリーメント)」
- ▶ 133 ページの「パフォーマンス・サマリの表示」
- ▶ 137 ページの「パフォーマンスのグラフによる表示方法」
- ▶ 140 ページの「サーバのパフォーマンスの確認」
- ▶ 144 ページの「問題の原因の特定方法」
- ▶ 148 ページの「シナリオ実行で収集可能なその他の情報」
- ▶ 150 ページの「結果の発行方法」
- ▶ 153 ページの「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応答時間はどうだったでしょうか。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応答時間はどうだったでしょうか。
- ▶ 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応答時間はどうだったでしょうか。
- ▶ 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。

次のセクションでは、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] > [すべてのプログラム] > [HP LoadRunner] > [LoadRunner] を選択します。[HP LoadRunner 11.00] ウィンドウが開きます。

2 LoadRunner Analysis を開きます。

[LoadRunner ランチャー] タブで、[テスト結果の分析] をクリックします。HP LoadRunner Analysis が起動します。

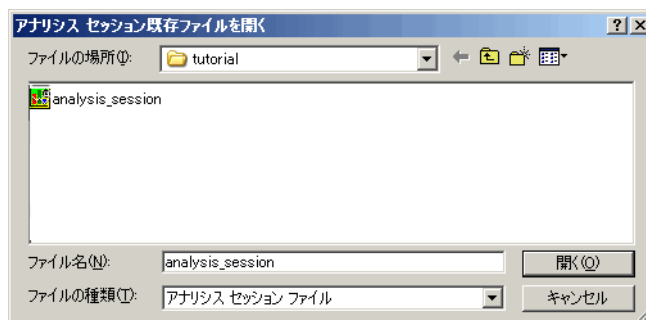
3 Analysis セッション・ファイルを開きます。

このセクションではより多様な結果を得るため、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。[Analysis セッション既存ファイルを開く] 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：セッションを古いバージョンの LoadRunner から新しいバージョンへ変換するよう求められた場合、[OK] をクリックします。

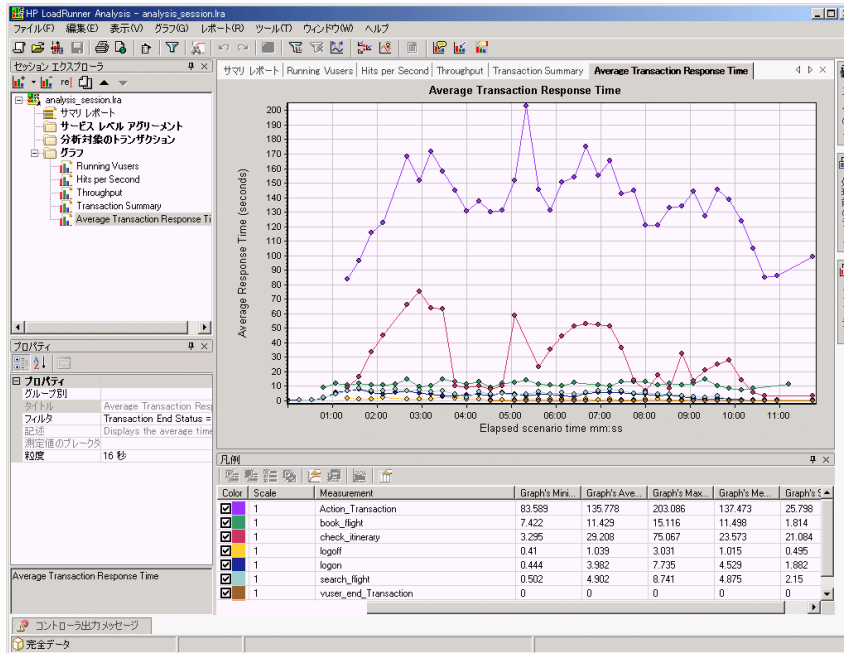


Analysis のウィンドウ内にセッション・ファイルが開きます。

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] 表示枠
- ▶ [プロパティ] ウィンドウ表示枠
- ▶ 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] 表示枠** : 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。
- ▶ **[プロパティ] ウィンドウ表示枠** : 左下の表示枠には、[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] ウィンドウに表示されます。黒で表示されたフィールドは編集可能です。

- ▶ **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。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。

ゴールの達成の確認（サービス・レベル・アグリーメント）

このセクションでは、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、スクリプト内の任意のトランザクションに対する、トランザクション応答時間測定値の平均に、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応答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応答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。

次のセクションでは、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応答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応答時間の最大値です。

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応答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応答時間は許容されます。

3 つの負荷シナリオを考慮するよう SLA を設定します：軽い負荷、平均負荷、重い負荷。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 を定義する方がより推奨されます。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。

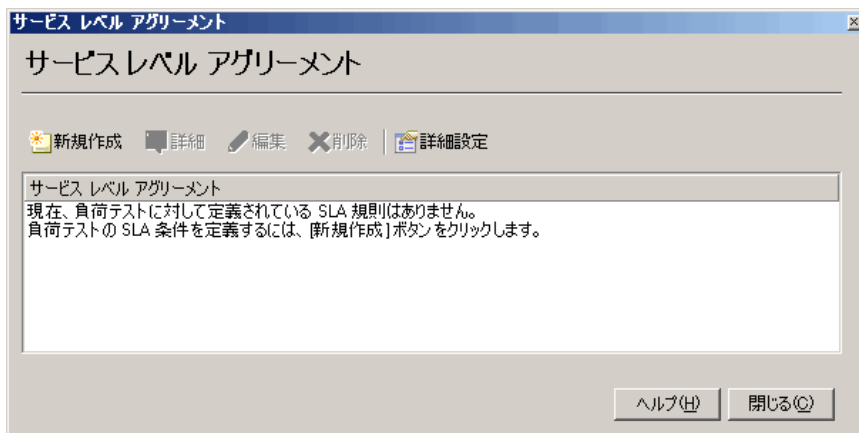
サンプルのセッション・ファイル内の **book_flight** および **search_flight** トランザクションの平均トランザクション応答時間を設定する SLA を定義します。

平均トランザクション応答時間は、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 設定ウィザードを開きます。

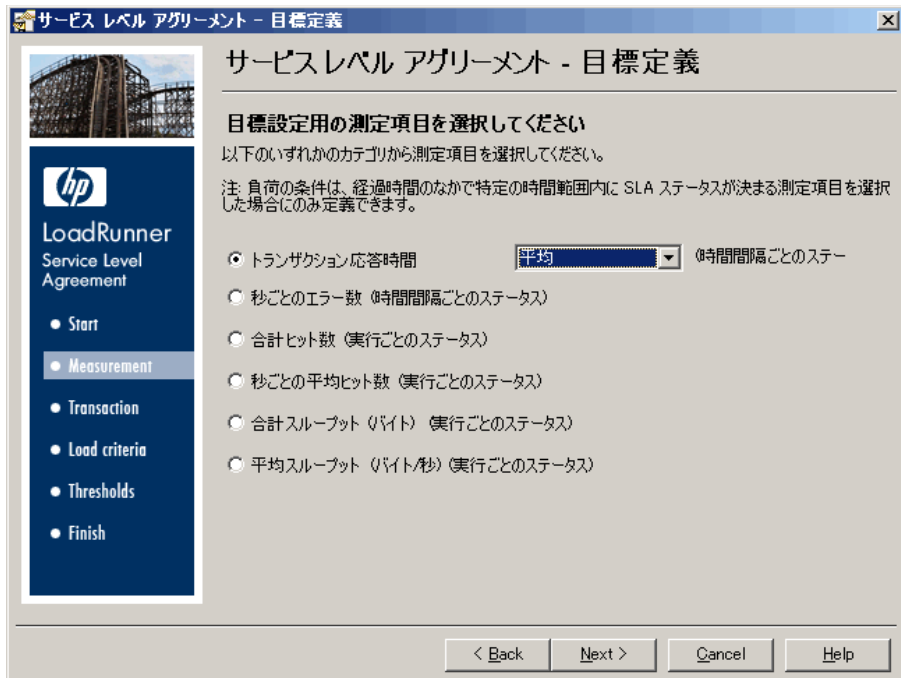
[ツール] > [SLA ルールの設定] を選択します。[サービス レベル アグリーメント] ダイアログ・ボックスが開きます。[新規作成] をクリックしてウィザードを開きます。



2 ゴールの測定方法を選択します。

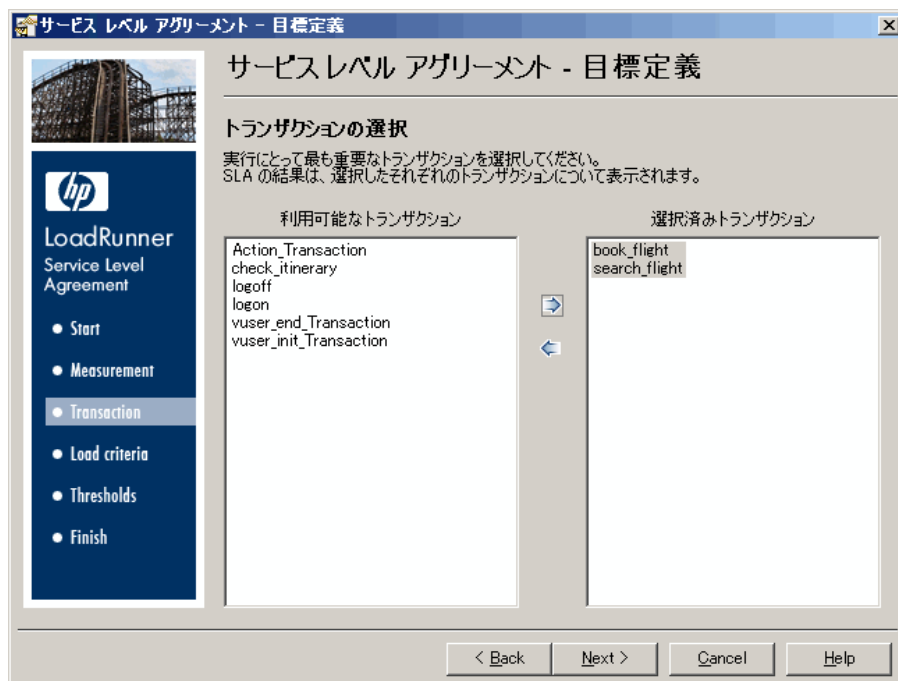
注： サービス・レベル・アグリーメント・ウィザードを初めて開く場合、ようこそページが表示されます。次にウィザードを実行したときにこのページが表示されないようにするには、[次回からこのページをスキップする] を選択します。

[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。[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。

[トランザクションの選択] ページで、スクリプト内のすべてのトランザクションのリストである [利用可能なトランザクション] リストから監視するトランザクションを選択します。book_flight および search_flight トランザクションをダブルクリックして選択します。[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。

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

サービス レベル アグリーメント - 目標定義

サービスレベル アグリーメント - 目標定義

負荷の条件の設定

選択済み測定項目:

テストしたい負荷の条件と負荷の値を設定してください。
目標のしきい値は、設定した値に照らして確認されます。

注: 負荷の値は、連続的で網羅的な範囲をカバーしなければなりません。

負荷の条件:

負荷の値:

<input checked="" type="checkbox"/>	未満	<input type="text" value="20"/>		
<input checked="" type="checkbox"/>	間	<input type="text" value="20"/>	-	<input type="text" value="50"/>
<input checked="" type="checkbox"/>	以上	<input type="text" value="50"/>		

< Back Next > Cancel Help

これで 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応答時間を SLA に設定しました。

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

5 しきい値を設定します。

[しきい値の設定] ページで、**book_flight** および **search_flight** 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応答時間を定義します。

次の例のようにしきい値を設定します。

サービスレベル アグリーメント - 目標定義

しきい値の設定
 選択済み測定項目:

各トランザクションの負荷の値ごとの上限しきい値を入力してください。このしきい値を上回る結果の場合、SLA ステータスは失敗となります。

ヒント: 1 セットのしきい値を以下の [全部に適用] テーブル内の全トランザクションに適用できます。

実行中の仮想ユーザ			
トランザクション名	<20	≥20 および <50	≥50
book_flight	5	10	15
search_flight	5	10	15

1 セットのしきい値を全トランザクションに適用するには、しきい値を入力して [全トランザクションに適用] をクリックします。

全部に適用:

<20	≥20 および <50	≥50
0	0	0

全トランザクションに適用

< Back Next > Cancel Help

これで選択したトランザクション両方に対して、次の平均トランザクション応答時間を許容すると設定しました。

- ▶ **軽い負荷** : 5 秒以下
- ▶ **平均の負荷** : 10 秒以下
- ▶ **重い負荷** : 15 秒以下

注： 選択したトランザクションに対する各しきい値は同じである必要はありません。各トランザクションに異なる値を割り当てられます。

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 設定を標準のサマリ・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。

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。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。

Analysis Summary		期間: 28/06/2004 09:59:49 - 28/06/2004 10:11:46
Scenario Name: C:\Tutorial_Scenario_gaby.lrs		
Results in Session: S:\run6\n6\1414\runner_download\prd\tutorial\tutorial_result\tutorial_result.lrs		
Duration: 11 分 および 57 秒.		
Statistics Summary		
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	View HTTP Responses Summary
Total Errors:	9,226	

最もパフォーマンスの悪かったトランザクション

[5 ワースト トランザクション] 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。

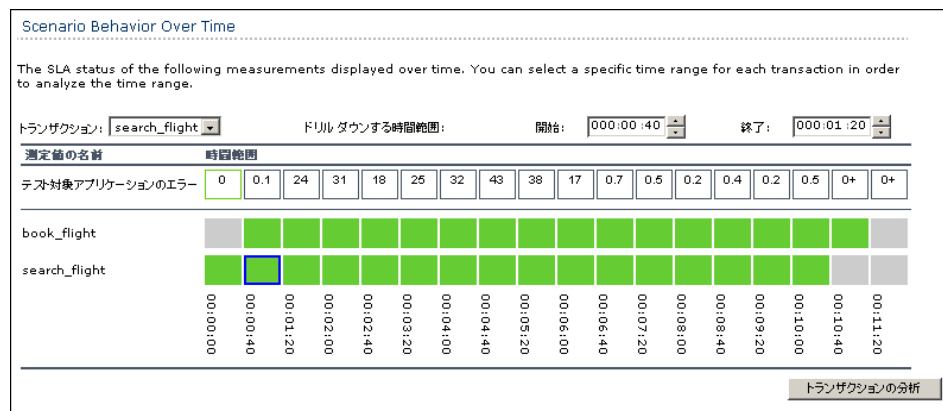
5 Worst Transactions		
トランザクション名	失敗の割合[%] (超過時間/トランザクション持続時間)	失敗値[%] (応答時間/SLA)
- book_flight	39.68	43.71
時間間隔	平均超過割合	最大超過割合
00:01:05-00:01:15	10.27	15.98
00:02:05-00:02:10	2.33	2.33
00:03:30-00:03:50	8.86	16.94
00:04:15-00:04:20	3.16	3.16
00:04:25-00:04:30	6.87	6.87
00:05:35-00:05:40	7.65	7.65
00:07:25-00:08:20	30.71	53.31
00:08:30-00:08:50	27.61	52.97
00:08:55-00:09:00	17.92	17.92
00:09:05-00:09:45	32.6	49.57
00:10:00-00:11:20	86.43	128.16

トランザクションの分析

book_flight トランザクションの実行で、処理時間のうちの 39.68% がしきい値を超えてい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 43.71% でした。

SLA しきい値を超えた時間間隔の確認

[経過時間ごとのシナリオ動作] セクションには、異なる時間間隔に各トランザクションがどのように実行し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間隔、赤の四角はトランザクションが失敗した時間間隔、そして灰色は関連する SLA が定義されていない時間間隔を表します。



SLA を定義した 2 つのトランザクションのうち、**search_flight** は測定した時間間隔のすべてにおいてしきい値内であり、一方の **book_flight** は複数の間隔でしきい値を超えたことがわかります。

全体的なトランザクションのパフォーマンスの確認

[トランザクション サマリ] 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 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

Service Level Agreement Legend: Pass Fail No Data

各トランザクションの応答時間を確認します。[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 65.754 秒であったことがわかります。これは、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。

[SLA ステータス] カラムにどのように関連する全体の SLA ステータスが表示されているかに注意してください。**book_flight** は [失敗]、**search flight** は [成功] となっています。

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] 表示枠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。

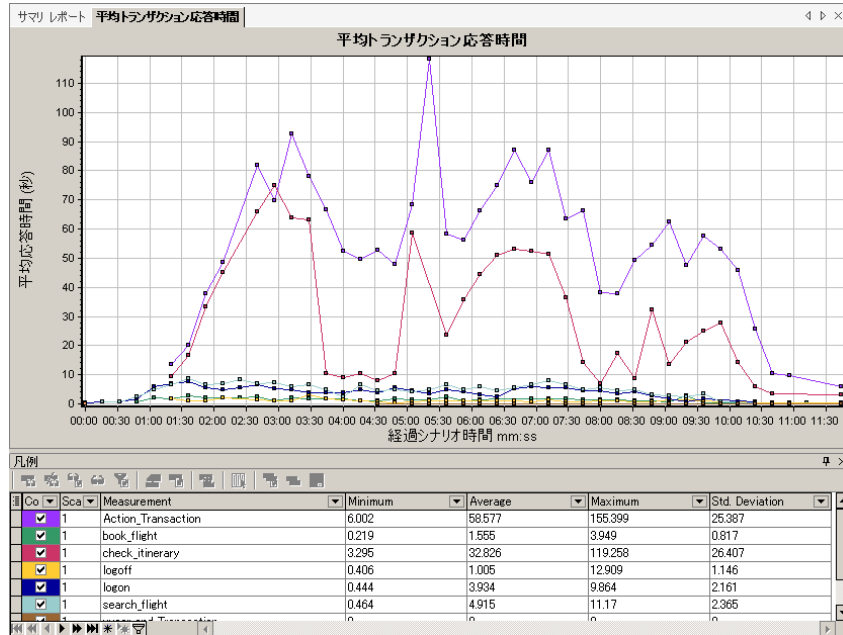
1 [平均トランザクション応答時間] グラフを開きます。

[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。

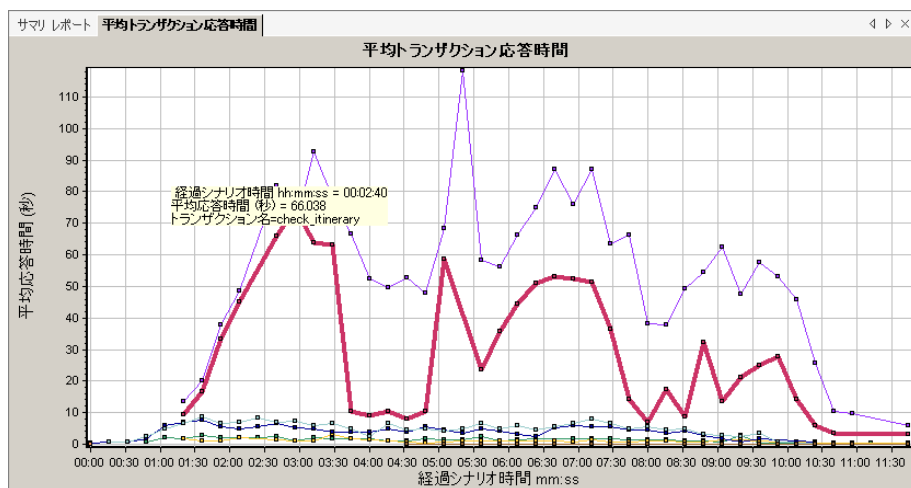
注: [セッション エクスプローラ] 表示枠にグラフが表示されない場合, [グラフ] ノードを右クリックし, [新規グラフを開く] ダイアログ・ボックスで [トランザクション: 平均トランザクション 応答時間] ノードを選択します。[グラフを開く] をクリックして [セッション エクスプローラ] 表示枠にグラフを追加します。

第 10 章 • シナリオの分析

[グラフの凡例] 表示枠で、**check_itinerary** トランザクションをクリックします。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。



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。

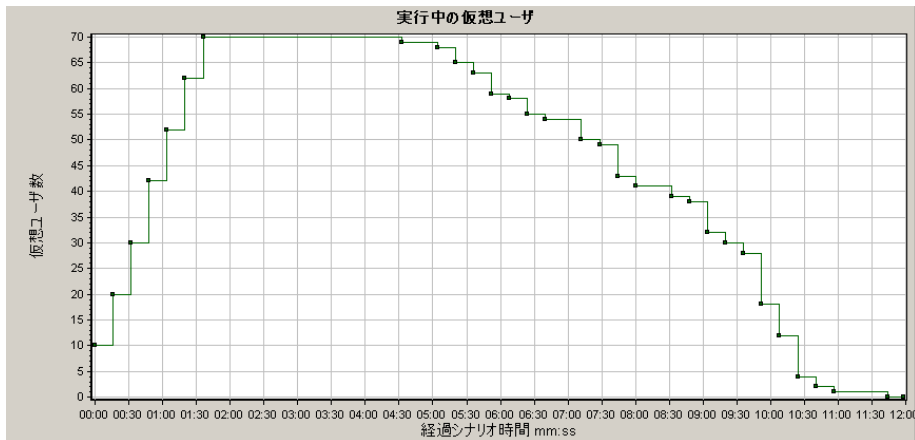
パフォーマンスの良いサーバでは、トランザクションの平均応答時間は比較的安定するでしょう。グラフの下部で、`logon` トランザクション、`logout` トランザクション、`book_flight` トランザクション、および `search_flight` トランザクションの平均応答時間がいくらか安定している点に注目してください。

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。

グラフ・ツリーで **[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。

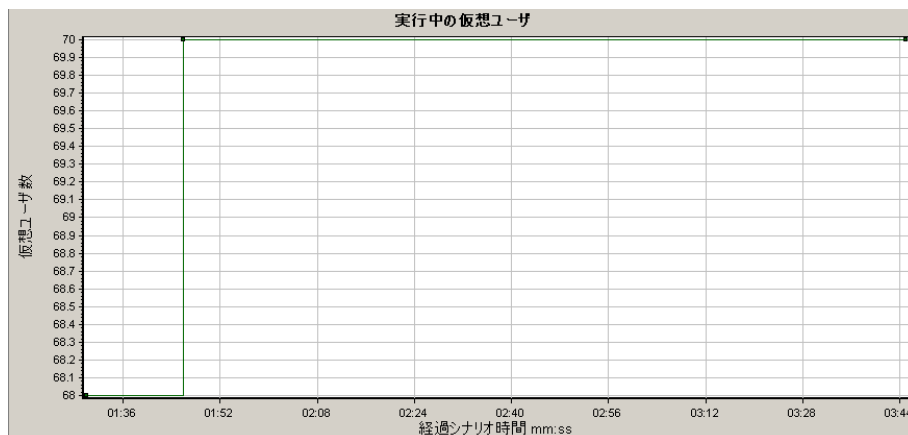


グラフを右クリックして、**[フィルタ / グループ化の設定]** を選択するか、またはツールバーの **[フィルタ / グループ化の設定 ...]** アイコンをクリックします。

[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 **000:01:30** から **000:03:45** までにします。[OK] をクリックします。

[グローバルフィルタ] 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。

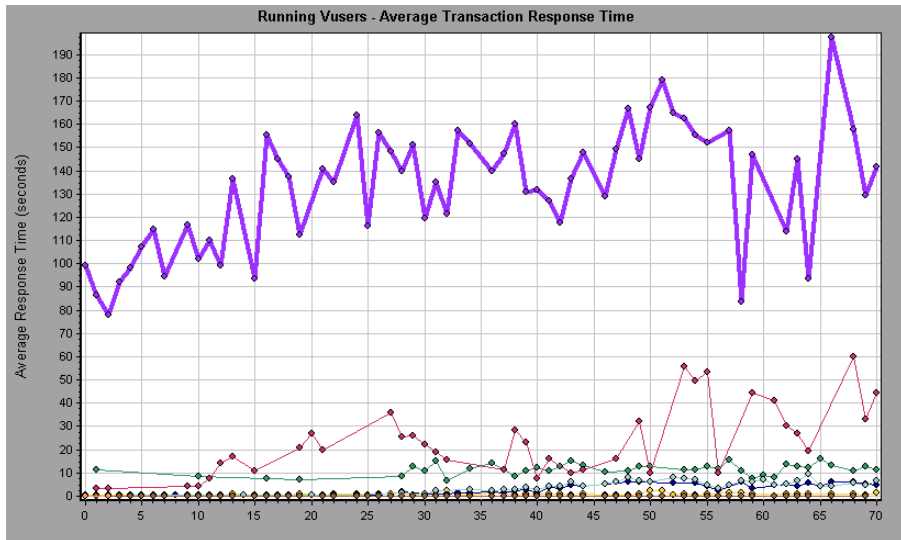
3 [実行中の仮想ユーザ] グラフと [平均トランザクション応答時間] グラフを相関させて, それらのデータを比較します。

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応答時間に与える影響を確認できます。

- a [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。
- b グラフを右クリックして、[グラフの結合] を選択します。
- c [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。
- d [結合タイプの選択] 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ー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。



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 つのグラフを関連させました。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、次の手順で行います。

- 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] 表示枠で、[新規作成] ボタンをクリックします。[新しいテンプレートを追加] 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、次の手順で行います。

- 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、[セッションに適用] をクリックします。

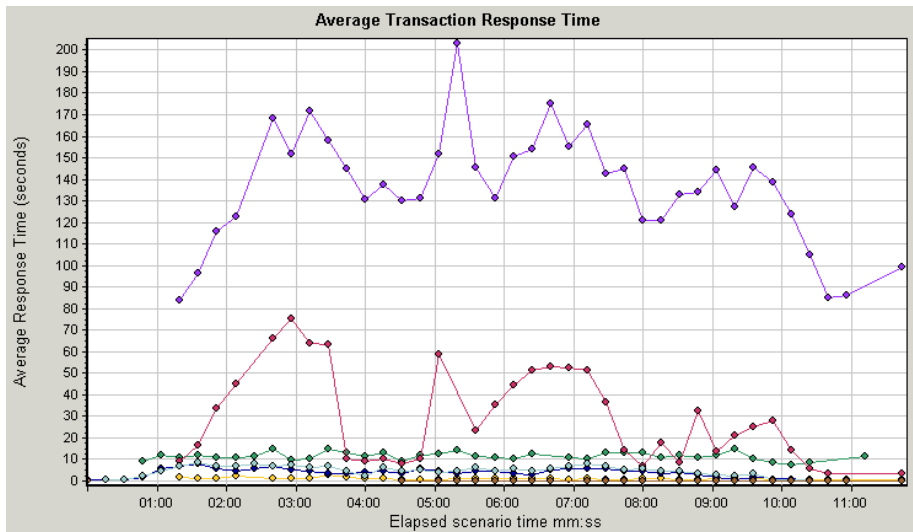
問題の原因の特定方法

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。

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。

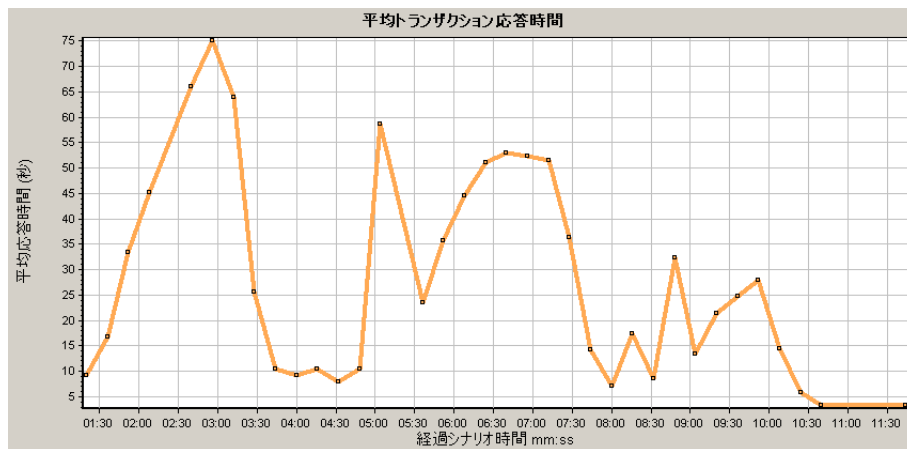


`check_itinerary` トランザクションの、特に経過時間が 1 分から 4 分の間を見ます。すぐに平均応答時間が増加し始め、ほぼ 3 分でピークに達するまで増加し続けました。

**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。**

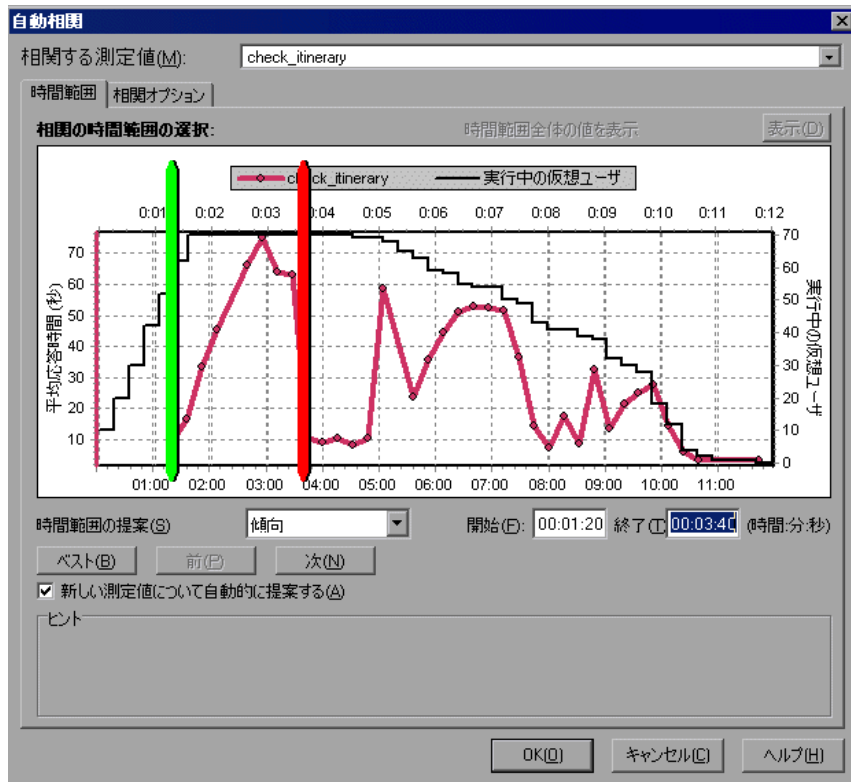
- a グラフを右クリックして、[フィルタ / グループ化の設定] を選択します。
- b [トランザクション名] の [値] カラムで **check_itinerary** を選択します。
- c [OK] をクリックします。

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。



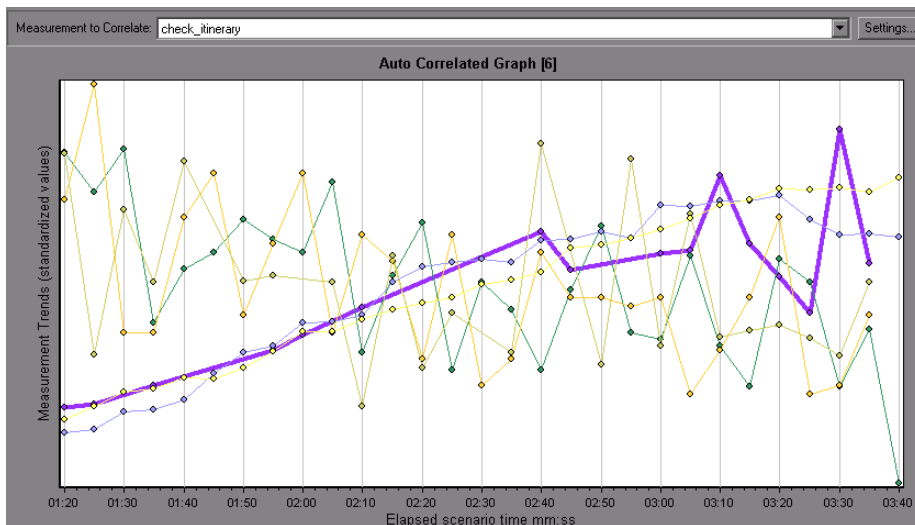
3 グラフを自動相関します。

- 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 [番号]**」が付けられます。

4 グラフの名前を変更します。

- a 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。
- 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。

グラフの下の凡例を見ます。

色	グラフ	倍率	測定値	相関一致	相関	マシン名	モニタの種類
✓	平均トランザク	標準化	S01_T06_CheckItinerary	100	直接関連している	なし	なし
■	秒ごとのトラン	標準化	Action_Transaction:Pass	24	直接関連している	なし	なし
■	秒ごとのトラン	標準化	S01_T03_SearchFlight:Pass	24	直接関連している	なし	なし
■	秒ごとのトラン	標準化	S01_T04_BookFlight:Pass	3	直接関連している	なし	なし
■	秒ごとのトラン	標準化	S01_T05_PaymentDetails:Pass	4	逆相関	なし	なし
✓	秒ごとのトラン	標準化	S01_T06_CheckItinerary:Pass	50	逆相関	なし	なし
■	秒ごとの合計	標準化	Pass	4	逆相関	なし	なし
✓	秒ごとのヒット	標準化	ヒット数	43	逆相関	なし	なし

[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 新しいグラフを表示します。

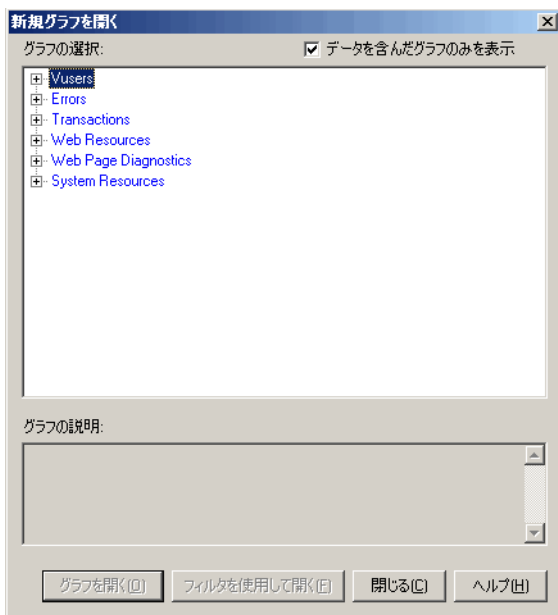


ツールバーで [新規グラフの追加] ボタンをクリックするか、[グラフ] > [新規グラフの追加] を選択します。

[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。

- ▶ **[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]** 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。



- [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、次の手順で行います。

- 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。

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、次の手順で行います。

1 [レポート] メニューから [新規レポート] を選択します。

[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 :

- ▶ [ベースのテンプレート] ドロップダウン・リストで、[詳細レポート (シングルラン)] を選択します。
- ▶ レポートのタイトルを入力します。
- ▶ 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 :

- ▶ 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。
- ▶ [追加] ボタンをクリックし、リストから [エグゼクティブ サマリ] を選択して [OK] をクリックします。[エグゼクティブ サマリ] 項目が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。
- ▶ 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。
- ▶ レポートの項目の表示順序を指定できます。
[ワークロードの特性] 項目で、[選択されたカラム] リストから [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。

