DS-Bench/Test-Env(D-Cloud)
実行手順書
1.01 版

2013 年　05 月 01 日

DEOS 研究開発センター

 DEOS プロジェクト

Revision History

Ver.	年月日	内容	改訂者	査閲	備考
1.00	2012/03/30	新規作成	中田	黒崎	
1.01	2013/05/01	全体の説明内容追加、修正、タイトル変更、3.4 の内容変更	DEOS R&D Center	DEOS R&D Center	

目次

1. はじめに

DS-Bench はベンチマークテストを組み合わせて実行し、テスト結果を評価するシステムです。この DS-Bench/Test-Env(D-Cloud)実行手順書（以下、本書）では、物理マシン単体でベンチマークを実行するケース、仮想マシン単体でベンチマークを実行するケース、物理マシン・仮想マシンを複数台組み合わせたケース、D-Case Editor と連携を行うケースの 4 ケースを基に、それぞれのシステム構成、ベンチマーク構成を挙げて DS-Bench/Test-Env(D-Cloud)の操作方法を説明します。

4 ケース目の D-Case Editor と連携するケースについて、同じ連携を D-Case Editor のかわりに D-Case Weaver で行うこともできます。本書では D-Case Editor との連携方法のみ説明します。D-Case Weaver との連携については別途、D-Case Weaver 仕様書(DEOS-FY2013-CW-01J)を参照してください。

本書は DS-Bench/ Test-Env(D-Cloud)の環境が構築されていることを前提としています。

環境構築については別途、DS-Bench/Test-Env 環境構築手順書(DEOS-FY2013-BI-01J)を参照して行ってください。

※資源を管理してシステムテストを迅速に行うツールである Test-Env について、本書では D-Cloud と表記します。

また、本書に記載されているシステム名、製品名、サービス名などは一般に各社の商標または登録商標です。

2. 動作環境

本書では以下の 4 ケースを 4 つの動作環境で実施しています。各環境、パッケージのインストールについては、以下の構成を基に DS-Bench/Test-Env(D-Cloud)環境構築手順書を参照してください。

また、D-Case Editor と連携するケースでは、D-Case Editor のインストール及び設定が必要になります。

2.1. 物理マシン単体を使用した構成

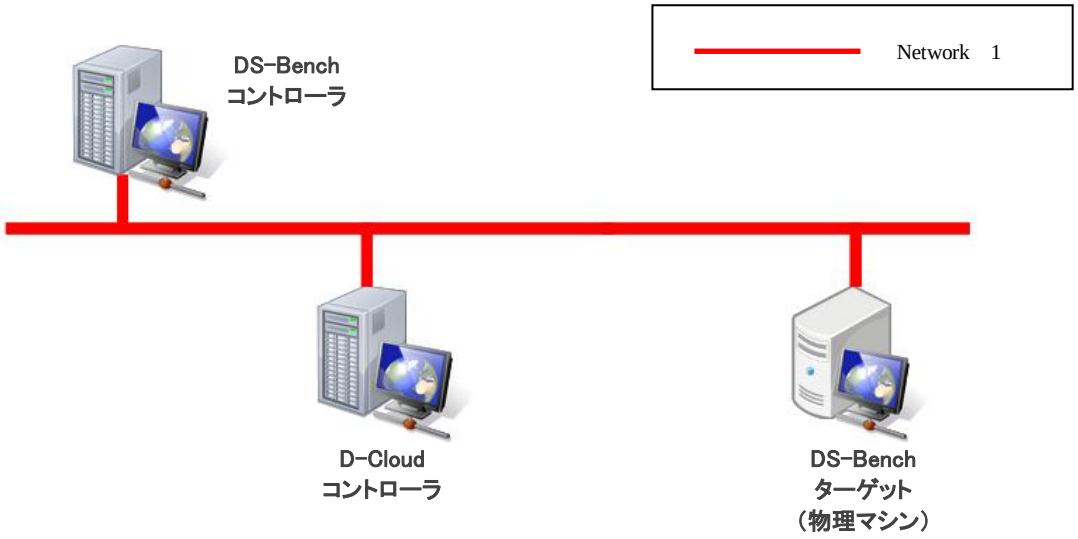
- DS-Bench コントローラ

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  Sedna XML DB
- DS-Bench ターゲット（物理マシン）

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1
- D-Cloud コントローラ

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1



2.2. 仮想マシン単体を使用した構成

- DS-Bench コントローラ

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  Sedna XML DB
- DS-Bench ターゲット（仮想マシン）

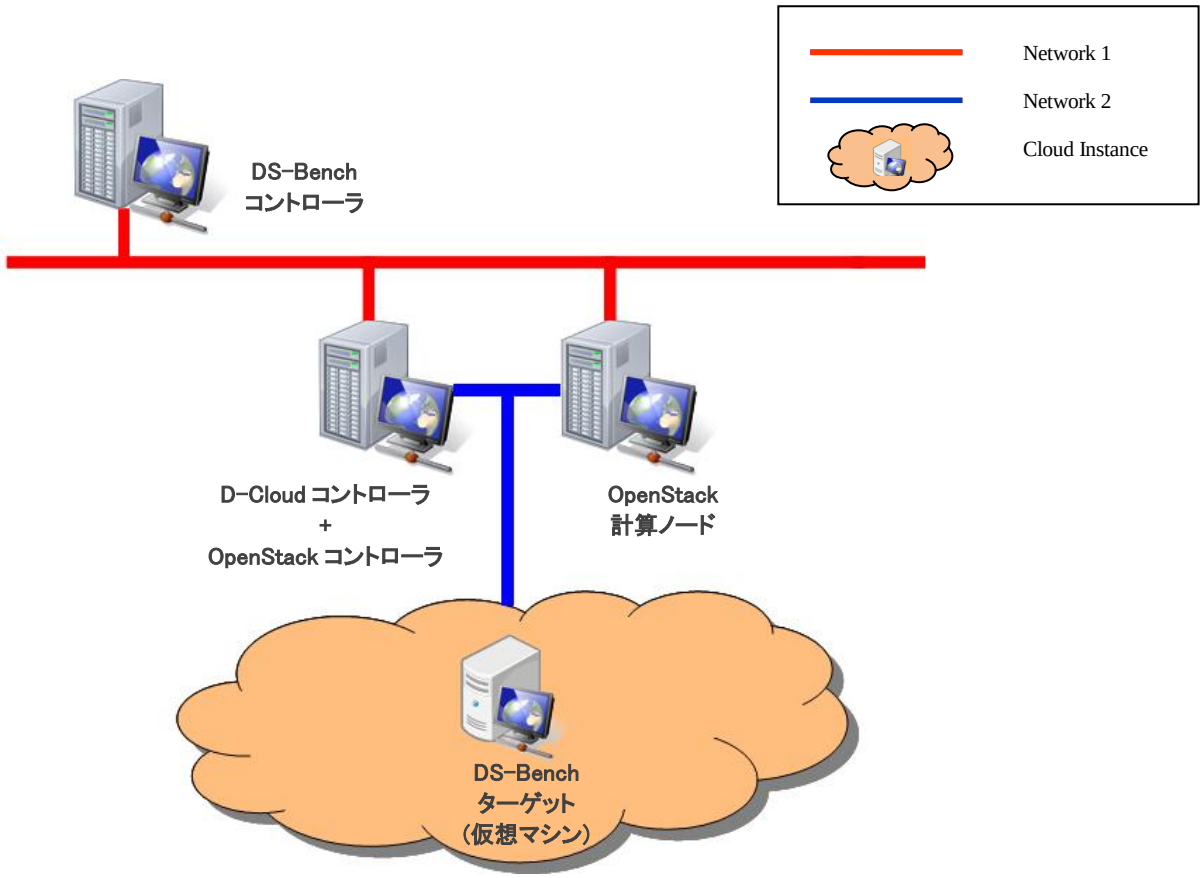
  ubuntu-10.04-server-cloudimg-amd64 または ubuntu-12.04-server-cloudimg-amd64
- D-Cloud コントローラ + OpenStack コントローラ

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  OpenStack 2011.3（diablo）または OpenStack 2012.1.3（Essex）
- 計算ノード

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  OpenStack 2011.3（diablo）または OpenStack 2012.1.3（Essex）



2.3. 物理マシン・仮想マシンを複数台組み合わせた構成

- DS-Bench コントローラ

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  Sedna XML DB
- DS-Bench ターゲット（物理マシン）

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  Apache HTTP Server
- DS-Bench ターゲット（仮想マシン）

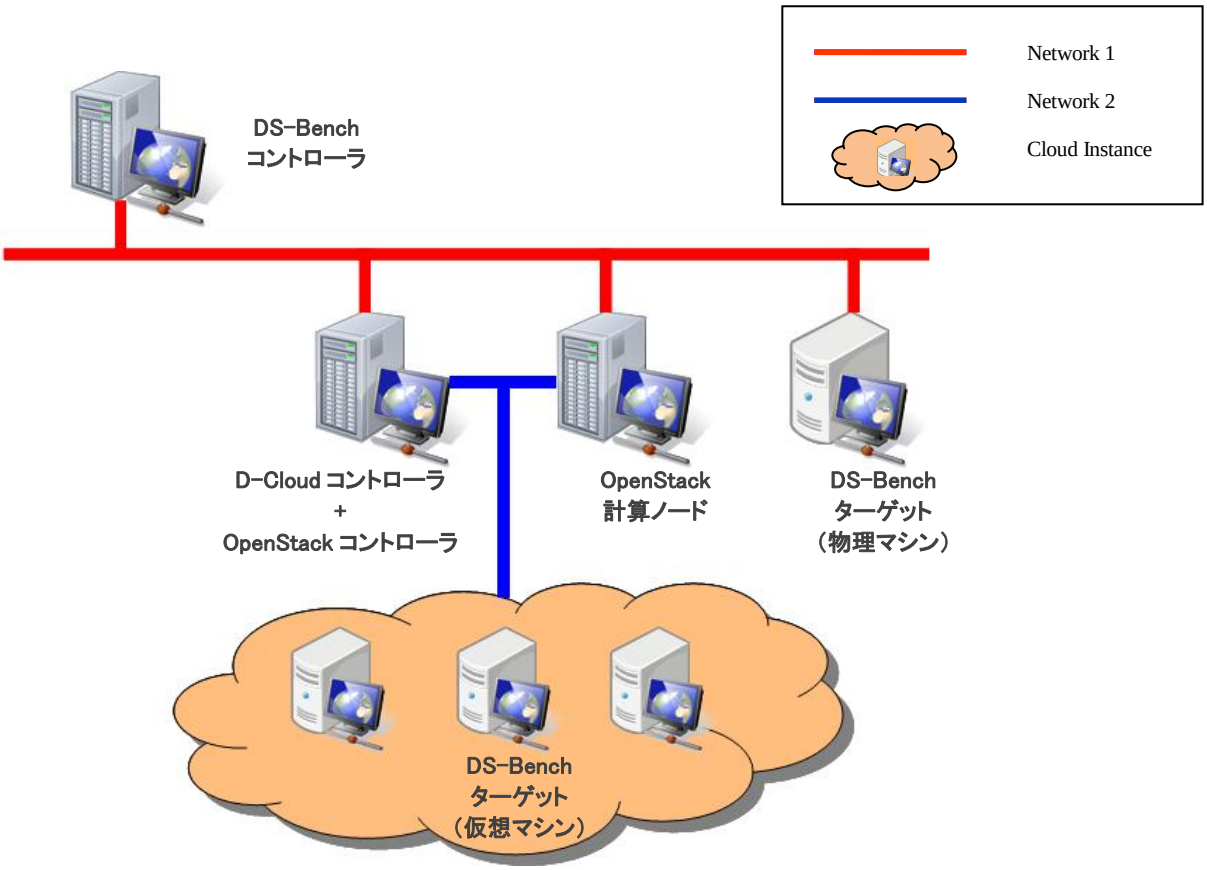
  ubuntu-10.04-server-cloudimg-amd64 または ubuntu-12.04-server-cloudimg-amd64
- D-Cloud コントローラ ＋OpenStack コントローラ

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  OpenStack 2011.3（diablo）または OpenStack 2012.1.3（Essex）
- OpenStack 計算ノード

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  OpenStack 2011.3（diablo）または OpenStack 2012.1.3（Essex）



2.4. D-Case Editor を使用し、連携を行う場合の構成

- DS-Bench コントローラ

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  Sedna XML DB
- DS-Bench ターゲット（仮想マシン）

  ubuntu-10.04-server-cloudimg-amd64 または ubuntu-12.04-server-cloudimg-amd64
- D-Cloud コントローラ + OpenStack コントローラ

  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  OpenStack 2011.3（diablo）または OpenStack 2012.1.3（Essex）
- OpenStack 計算ノード

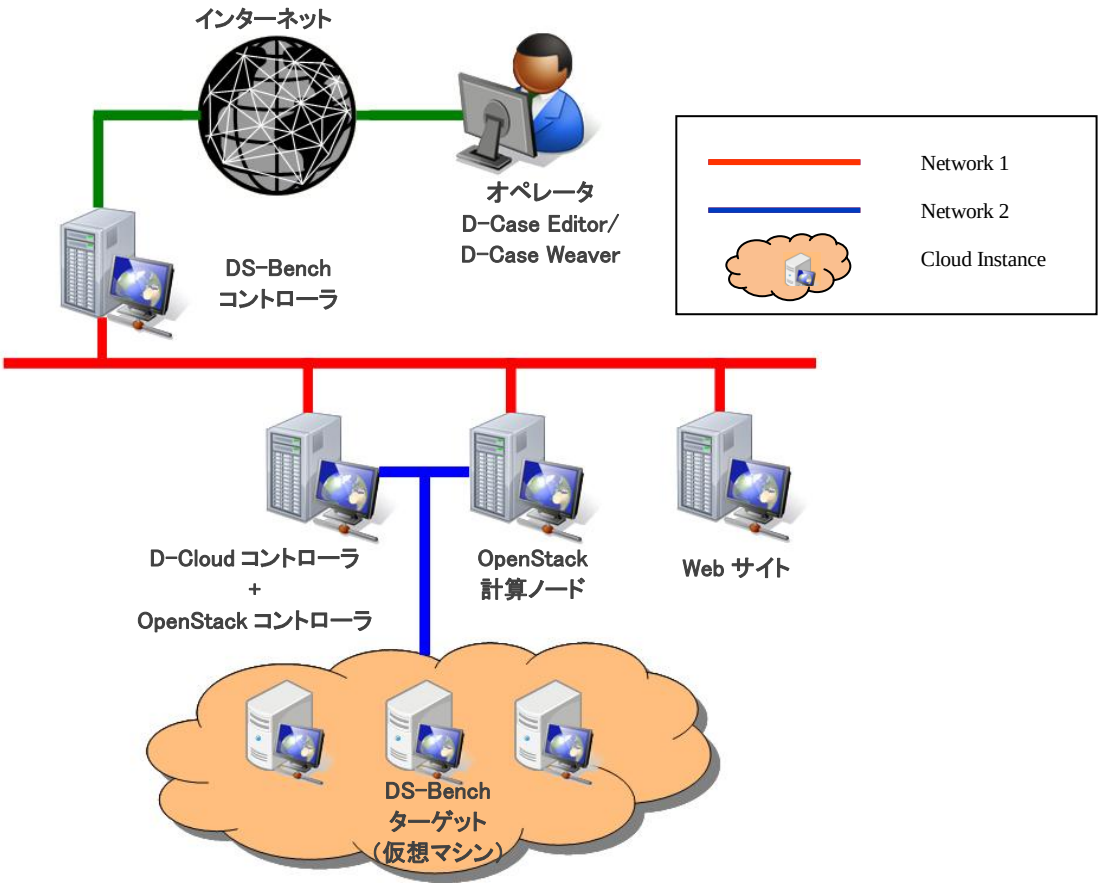
  Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1

  OpenStack 2011.3（diablo）または OpenStack 2012.1.3（Essex）
- オペレータ

  D-Case Editor

  D-Case Weaver
- Web サイト

  Apache HTTP Server



3. 実行手順の詳細

3.1.物理マシン単体でベンチマークを実行するケース

2.1 物理マシン単体を使用した構成通りに DS-Bench コントローラ、D-Cloud コントローラ、DS-Bench ターゲット（物理マシン）をインストールした環境を用意し、DS-Bench ターゲットを D-Cloud 管理下に置きます。この登録した DS-Bench ターゲット単体を用いたシナリオを作成／実行し、結果を確認するケースです。

このケースでは、使用するベンチマークに「Bonnie」を選択し、サーバのシーケンシャルアクセス(read/write)、ランダムアクセス等の性能測定を一度に行い、その結果を保存し、結果を確認する方法について、その一連の手順を説明します。

3.1.1. DS-Bench/D-Cloud のインストール

DS-Bench コントローラ、D-Cloud コントローラ、DS-Bench 物理ターゲットをインストールします。手順については「DS-Bench/Test-Env(D-Cloud)環境構築手順書」を参照してください。

3.1.2. 物理マシンを D-Cloud の管理下へ追加

D-Cloud コントローラサーバへログインします。

管理ファイルの所在を確認します。「DCLOUD_DESC_PATH」に設定されている値が管理ファイルのパスです。ここでは「/var/dcloud/config/resource_dsb.xml」とします。

```
$ cat /etc/dcloud.conf
.
.
export DCLOUD_DESC_PATH=/var/dcloud/config/resource_dsb.xml
.
.
$
```

管理ファイルへ対象の物理マシンの情報を追記します。

```
$ vi /var/dcloud/config/resource_dsb.xml
<dsb_resource_desc>
    <target type='physical'>
.
.
    </target>
    <target type='physical'>
.
.
</dsb_resource_desc>
```

<target>・・・</target>までにマシン情報が記述されています。

追加するマシン情報を管理ファイルへ追記します。以下はサンプルになります。

```
<target type='physical'>
    <name>trg-tutorial</name>
    <cpu>
        <arch>x86_64</arch>
        <vendor>Intel</vendor>
        <model>core2duo</model>
        <mhz>2400</mhz>
        <cores>2</cores>
        <threads>2</threads>
    </cpu>
    <memory type='DDR3'>
        <speed type='mhz'>800</speed>
        <size>2048</size>
    </memory>
    <devices>
        <device type='ethernet'>
            <target dev='eth0' type='1g' vendor='Intel' model='e1000' ip='*.*.*.*' priority='primary'/>
        </device>
        <device type='storage'>
            <controller type='sata' model='ich9'></controller>
            <target dev='/dev/hda' type='sata2' vendor='WesternDigital' model='Cavier Green' size='1T'
ip='' priority=''/>
        </device>
    </devices>
    <os>
        <family>Linux</family>
        <vendor>ubuntu foundation</vendor>
        <name>Ubuntu</name>
        <version>10.04</version>
        <arch>x86_64</arch>
    </os>
    <softwares>
        <software type='kernel'>
            <version>2.6.32-21-server</version>
        </software>
        <software type='library'>
            <name>libc</name>
            <version>glibc2.6.??</version>
        </software>
        <software type='package'>
            <family>RPM</family>
        </software>
        <software type='compiler'>
            <lang>C</lang>
            <family>GNU</family>
            <name>gcc</name>
            <version>4.4</version>
        </software>
    </softwares>
</target>
```

赤字で記述したものは必須項目です。**この項目で D-Cloud の管理下へ追加します**。

「ip」は DS-Bench コントローラ、D-Cloud コントローラと通信が出来る IP アドレスを指定してください。

また、「priority」が「primary」のインタフェースを通信で使用します。

青字で記述したものが DS-Bench の「Target detail」画面上に表示されます。任意で指定してください。

※任意で項目を指定するので、指定された項目間での整合性のチェックは行っていません。（ubuntu ではパッケージ管理は「APT」を使用していますが、サンプルでは「RPM」となっている。等）

※上記サンプルの場合は以下のように表示されます。

Target detail

cpu				
arch	x86_64			
vendor	Intel			
model	core2duo			
mhz	2400			
cores	2			
threads	2			
memory(DDR3)				
speed(mhz)	800			
devices				
device(ethernet)				
eth0	1g	Intel	e1000	
device(storage)				
/dev/hda	sata2	WesternDigital	Cavier Green	1T
os				
family	Linux			
vendor	ubuntu foundation			
name	Ubuntu			
version	10.04.3			
arch	x86_64			
softwares				
software(kernel)				
version	2.6.32-34-server			
software(library)				
name	libc			
version	glibc2.6.??			
software(package)				
family	RPM			
software(compiler)				
lang	C			
family	GNU			
name	gcc			
version	4.4			

Close

※最低限の項目を記述したものは以下になります。

```
<target type='physical'>
    <name>trg-tutorial</name>
    <devices>
        <device type='ethernet'>
            <target ip='*.*.*.*' priority='primary'/>
        </device>
    </devices>
</target>
```

管理ファイルへ追記完了後、D-Cloud のデーモンを再起動します。

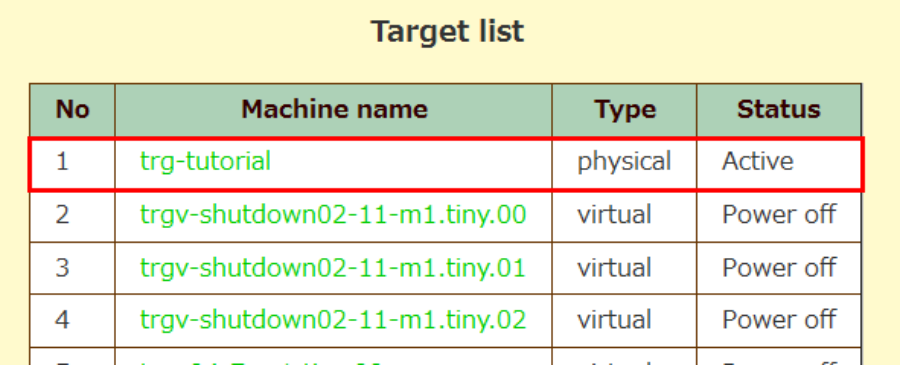
再起動するデーモンは「dc-resource-dsb」です。

```
$ sudo /etc/init.d/dc-resource-dsb restart
Stopping  ... success
Starting  ... success
$
```

DS-Bench 画面からターゲットリストへ追加されていることを確認します。

サンプルでは「trg-tutorial」が表示されます。

メイン画面 > Configuration 画面 > Target List



Target list

No	Machine name	Type	Status
1	trg-tutorial	physical	Active
2	trgv-shutdown02-11-m1.tiny.00	virtual	Power off
3	trgv-shutdown02-11-m1.tiny.01	virtual	Power off
4	trgv-shutdown02-11-m1.tiny.02	virtual	Power off

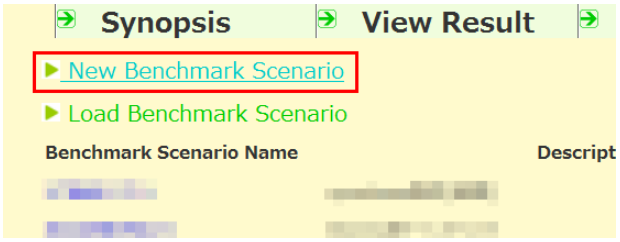
3.1.3. ベンチマークシナリオの作成

ブラウザを立ち上げ、DS-Bench メイン画面を開きます。

http://[DS-Bench コントローラの IP アドレス or ホスト名]/cgi-bin/main.cgi

（DS-Bench コントローラのインストール時に指定した DocumentRoot と Apache の DocumentRoot が同じ場合）

メインメニューの「New Benchmark Scenario」リンクをクリックしてください。

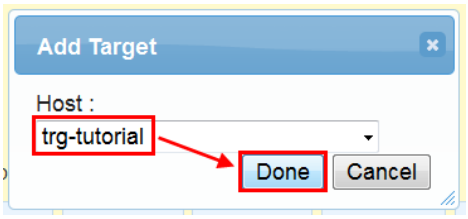


新規ウィンドウでシナリオ作成画面が開きます。

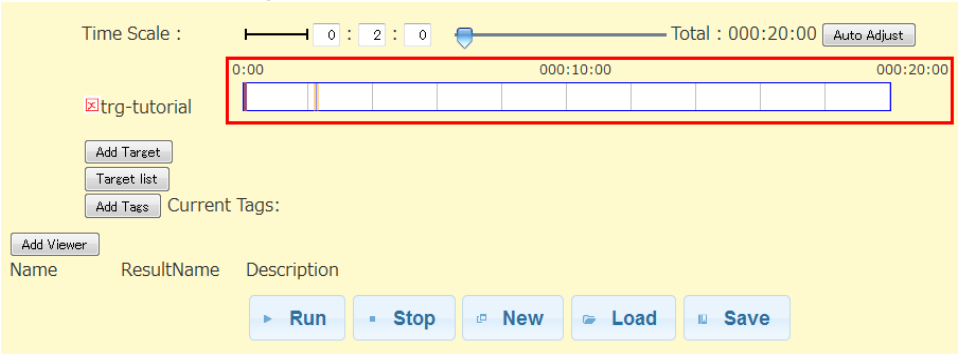
「Add Target」ボタンをクリックし、ベンチマーク対象のマシンを追加します。

ダイアログが表示されるので、「3.1.2 物理マシンを D-Cloud の管理下へ追加」で追加したマシンを選択します。ここでは「trg-tutorial」を選択します。

trg-tutorial を選択後、「Done」ボタンをクリックします。

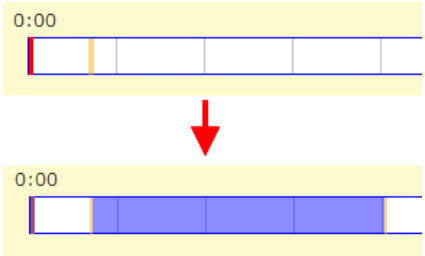


シナリオ作成画面に「trg-tutorial」のタイムラインが表示されます。



タイムライン上にベンチマークシナリオを追加します。

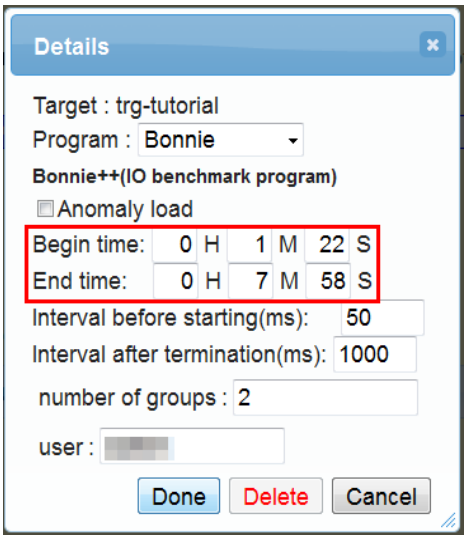
タイムライン上でクリックすると薄橙色のバーが表示されるので、そのまま右方向にドラッグしてください。ドラッグした分だけ青色のバーが伸びます。そして終了したいところでドロップしてください。この青色のバーの始めと終わりがベンチマーク設定ダイアログの開始、終了時間に入ります。



ドロップしたらベンチマーク設定のダイアログが表示されます。

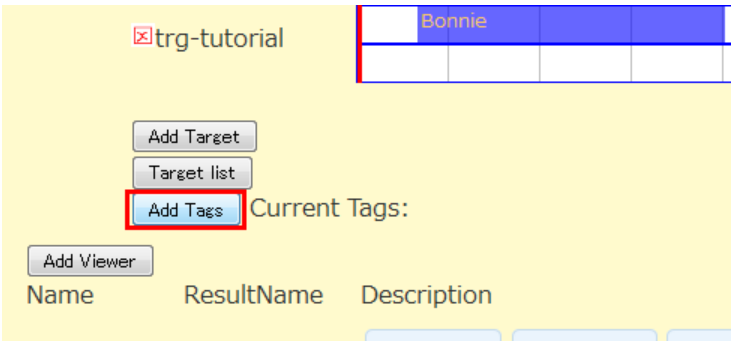
今回のケースでは Program に「Bonnie」を選択、user には存在するユーザ名を指定し、「Done」ボタンをクリックしてください。ベンチマークプログラムのパラメータはお使いの環境にあわせて調整してください。

実行からどれだけ時間がたったら開始、終了するか詳細に指定する場合は「Begin time」、「End time」で指定してください。



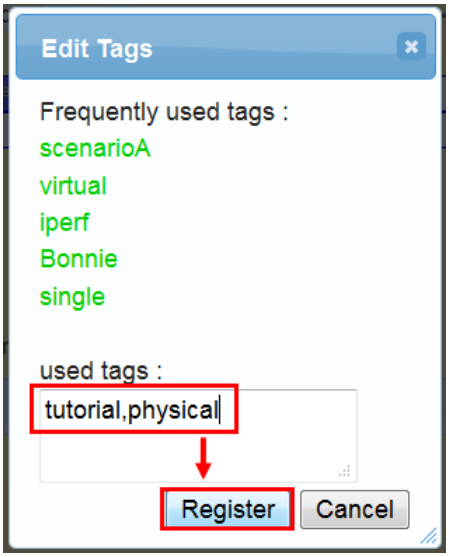
タグを設定します。設定は任意ですが、シナリオ結果を簡単に検索することができます。

「Add Tags」ボタンをクリックしてください。

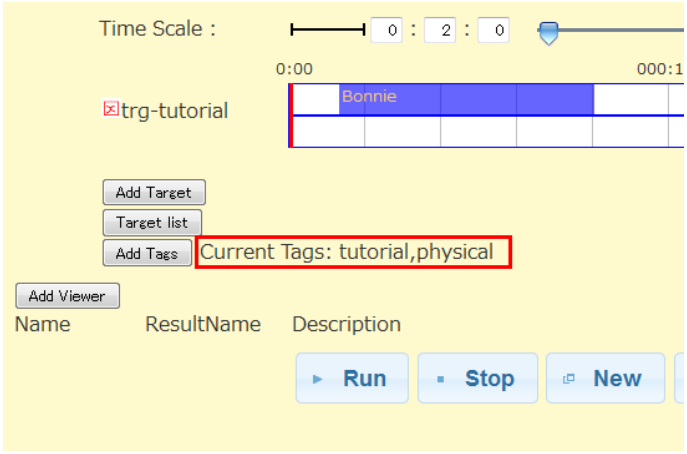


ダイアログが表示されるので、テキストエリアにタグを記入後、「Register」ボタンをクリックしてください。タグはカンマ区切りで複数指定できます。

ここでは「tutorial」と「physical」を登録します。

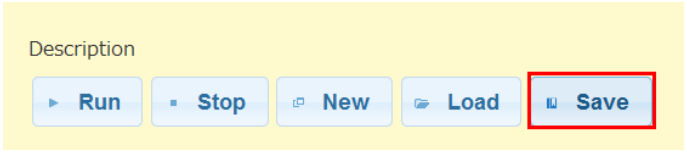


Current Tags に設定したタグが表示されます。



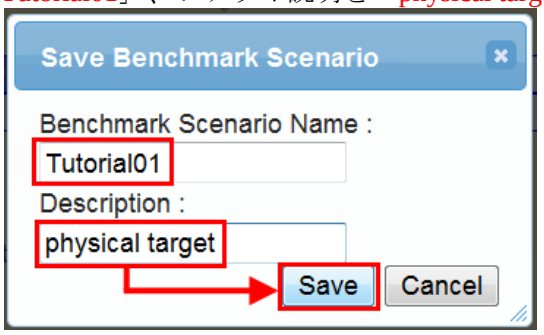
「Add Viewer」ボタンは D-Case Editor から実行した際に、D-Case Editor に渡す結果値のパラメータを設定するためのボタンです。詳しくは「3.4. D-Case Editor と連携を行うケース」で説明します。

作成したシナリオを保存します。画面下部の「Save」ボタンをクリックしてください。

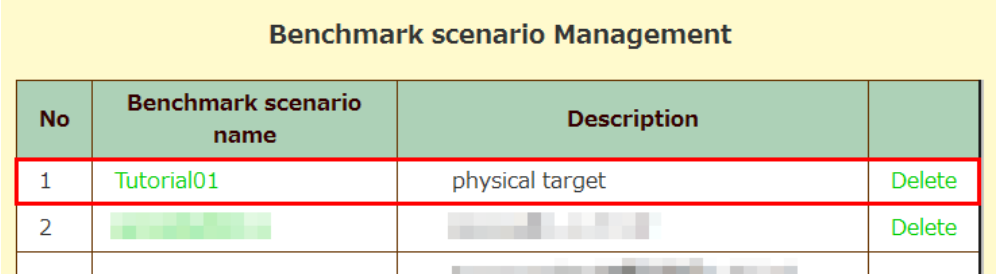


ダイアログが表示されます。シナリオ名、シナリオ説明を入力後、「保存」ボタンをクリックしてください。

ここではシナリオ名を「Tutorial01」、シナリオ説明を「physical target」としています。



※保存したシナリオは「メイン画面 > Configuration タブ > Benchmark Scenario」リンクから確認、削除ができます。



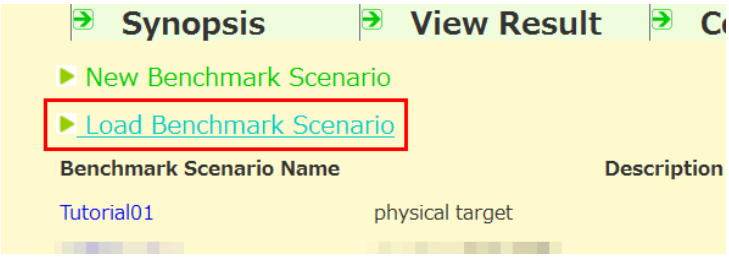
シナリオ作成は以上です。

試しにサンプルをいくつか作成し、保存してみてください。

3.1.4. ベンチマークシナリオの実行

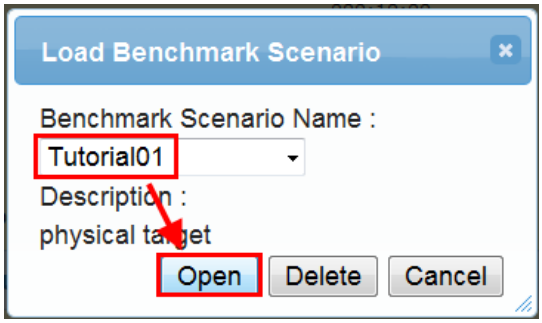
「3.1.3. ベンチマークシナリオの作成」で作成したシナリオを読み込みます。

DS-Bench メイン画面の「Load Benchmark Scenario」リンクをクリックしてください。



新規ウィンドウが立ち上がり、シナリオ選択ダイアログが表示されます。

「3.1.3. ベンチマークシナリオの作成」で作成したシナリオを選択し、「Open」ボタンをクリックしてください。ここでは「Tutorial01」を選択します。

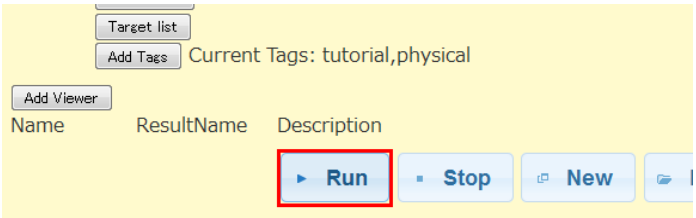


作成したベンチマークシナリオを読み込み、表示されます。

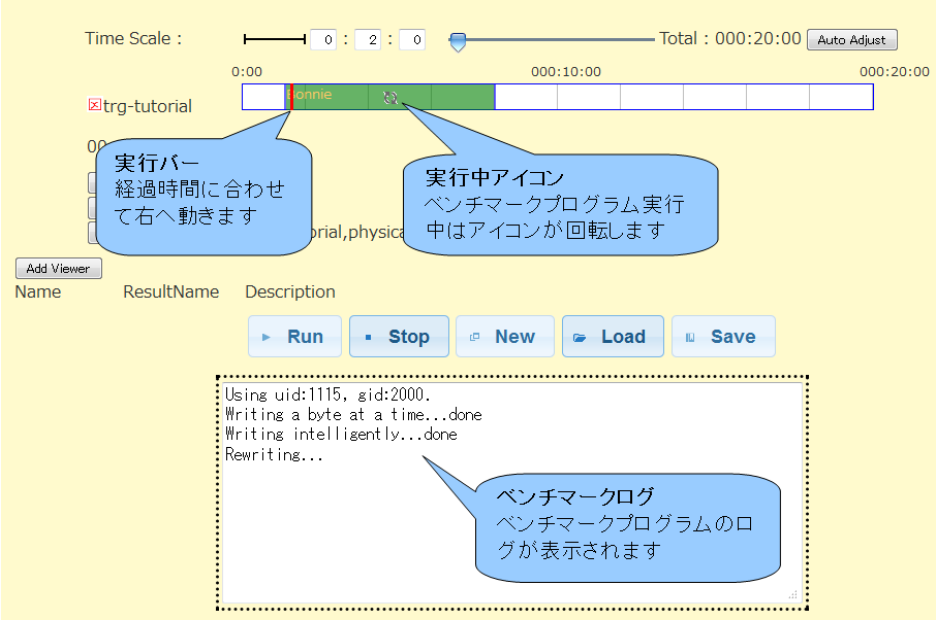
※シナリオの読み込みは以下の方法でも可能です。

- メイン画面の Load Benchmark Scenario リンク下部のシナリオ名リンク
  メイン画面 ＞ シナリオ名リンク
- ベンチマーク作成画面の「Load」ボタン
  メイン画面 ＞New Benchmark Scenario リンク ＞Load
- ベンチマークシナリオ設定画面
  メイン画面 ＞Configuration タブ ＞Benchmark Scenario＞ シナリオ名リンク

ベンチマークシナリオを実行します。画面下部の「Run」ボタンをクリックします。



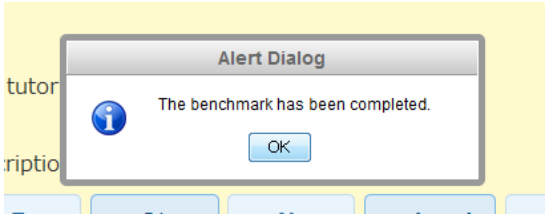
実行中の画面



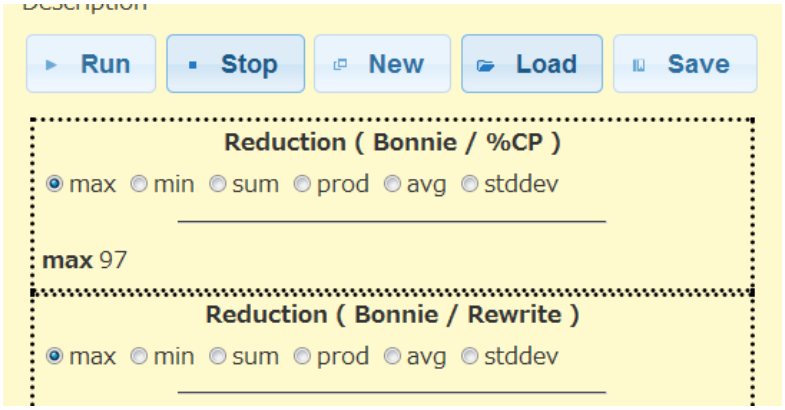
実行バーがベンチマークバーの終端に到達すると、シナリオの実行完了となります。

3.1.5. ベンチマークシナリオの結果確認

シナリオの実行が完了すると、ダイアログが表示されます。



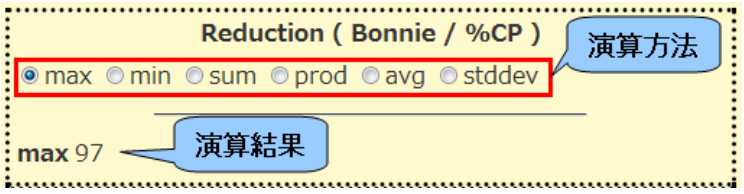
ベンチマークシナリオの結果は下部に表示されます。



- リダクション結果

ベンチマークプログラムの実行結果を集計し、演算をします。

演算方法はラジオボタンで変更することができます。

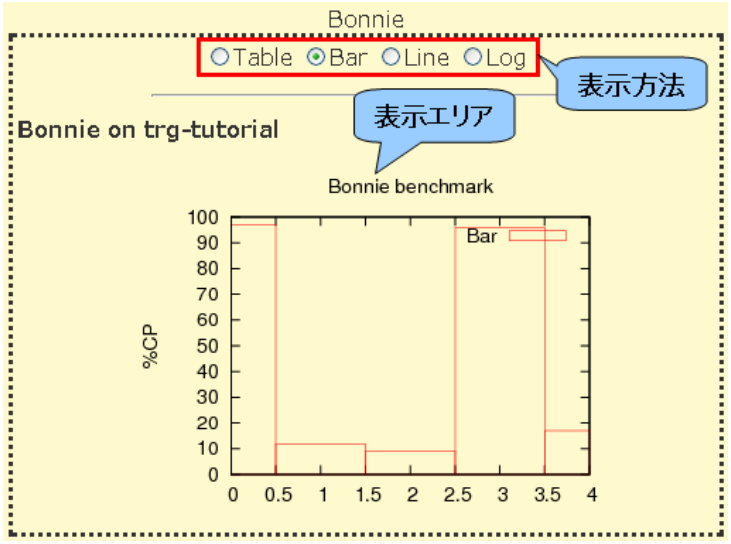


max	min	sum	prod	avg	stddev
最大値	最小値	合計	総乗	平均	標準偏差

- ベンチマーク結果

ベンチマークプログラムの実行結果が表示されます。

ラジオボタンでベンチマーク結果の表示方法を変更することができます。

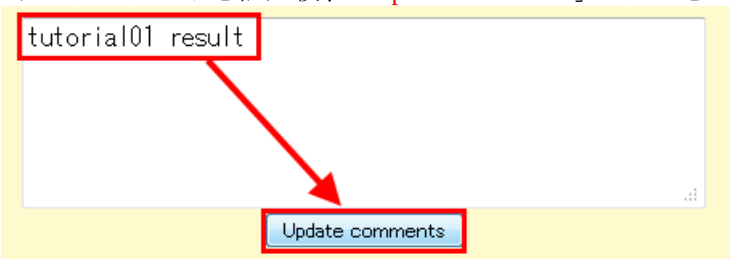


- ◆ Table　結果がテーブル形式で表示されます。
- ◆ Bar　結果が棒グラフで表示されます。
- ◆ Line　結果が折れ線グラフで表示されます。
- ◆ Log　ベンチマーク実行ログが表示されます

- コメントの登録

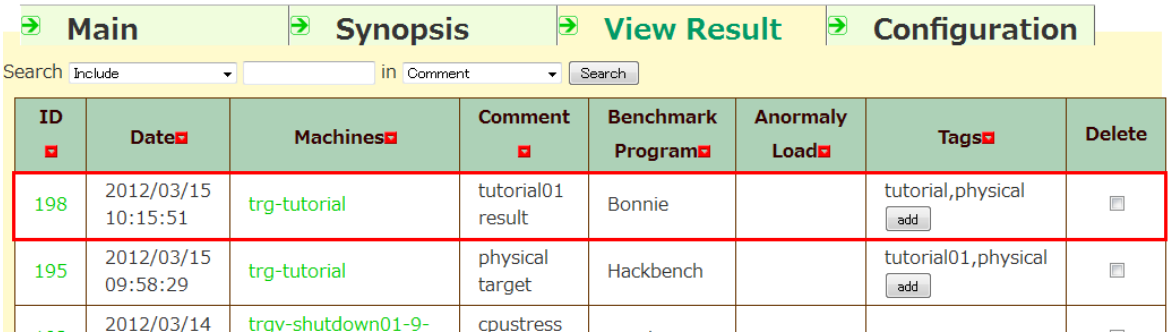
シナリオ実行結果に対して、コメントを登録することができます。

テキストエリアにコメントを記入後、「Update comments」ボタンをクリックしてください。



コメントを登録した場合、シナリオ実行結果一覧画面（View Result 画面）やシナリオ一覧

画面（Synopsis 画面）の「Comment」列に表示され、どのようなシナリオの実行結果か一覧から確認できるようになります。



※ シナリオ実行結果の確認方法

- シナリオ実行結果一覧画面（View Result 画面）

  メイン画面 > View Result タブ > ID リンク

- シナリオ一覧画面（Synopsis 画面）

  メイン画面 > Synopsis タブ > ID リンク

3.2. 仮想マシン単体でベンチマークを実行するケース

仮想マシン（イメージ）を DS-Bench ターゲットとして使えるように設定を行います。2.2. 仮想マシン単体を使用した構成を参考に環境構築を行い、作成した仮想マシンだけを用いたシナリオを作成／実行し、結果を確認するケースです。使用するベンチマークは「Memstress」を選択し、対象の仮想マシンに割り振られたメモリ領域に負荷をあたえます。

3.2.1. 仮想 DS-Bench ターゲットの作成・登録

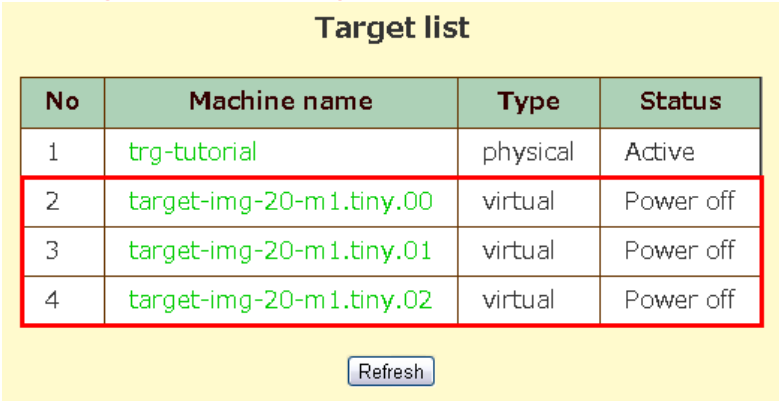
仮想 DS-Bench ターゲットを作成し、D-Cloud へ仮想マシンを登録します。手順については「DS-Bench/Test-Env(D-Cloud)環境構築手順書」の 4. OpenStack の項目を参照してください。

※登録完了後は必ず D-Cloud のデーモンを再起動してください。

```
$ sudo /etc/init.d/dc-mapserver restart
Stopping  ... success
Starting  ... success
$
$ sudo /etc/init.d/dc-resource-dsb restart
Stopping  ... success
Starting  ... success
$
$ sudo /etc/init.d/dcld restart
Stopping  ... success
Starting  ... success
$
```

DS-Bench 画面からターゲットリストへ追加されていることを確認します。

Main 画面 > Configuration 画面 > Target List



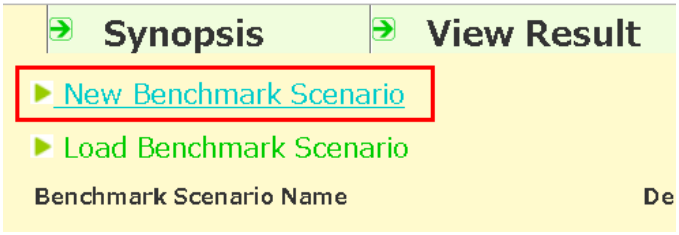
3.2.2. ベンチマークシナリオの作成

ブラウザを立ち上げ、DS-Bench メイン画面を開きます。

http://[DS-Bench コントローラの IP アドレス or ホスト名]/cgi-bin/main.cgi

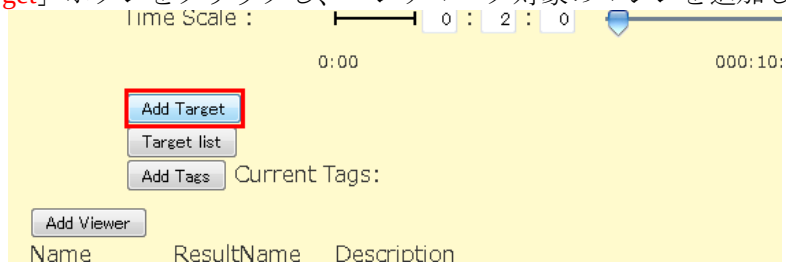
（DS-Bench コントローラのインストール時に指定した DocumentRoot と Apache の DocumentRoot が同じ場合）

メインメニューの「New Benchmark Scenario」リンクをクリックしてください。



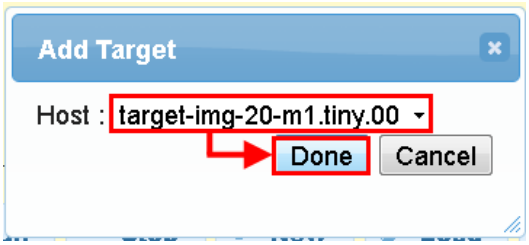
新規ウィンドウでシナリオ作成画面が開きます。

「Add Target」ボタンをクリックし、ベンチマーク対象のマシンを追加します。

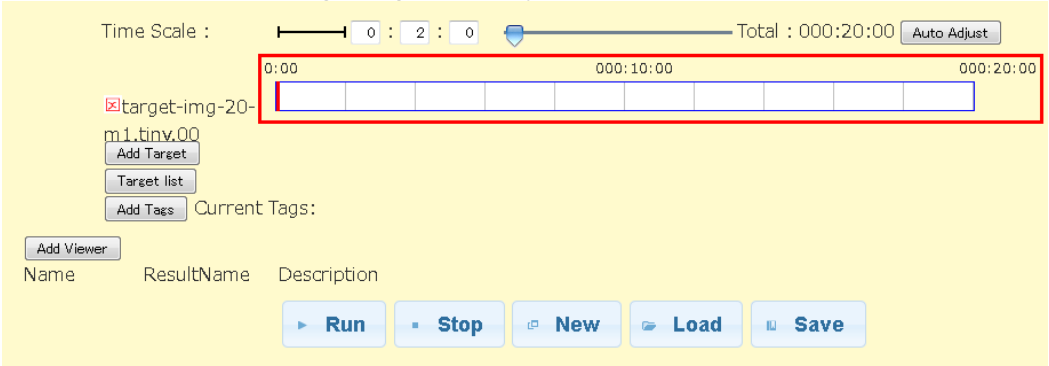


ダイアログが表示されるので、D-Cloud に追加したマシンを選択します。ここでは「target-img-20-m1.tiny.00」を選択します。

trg-tutorial を選択後、「Done」ボタンをクリックします。

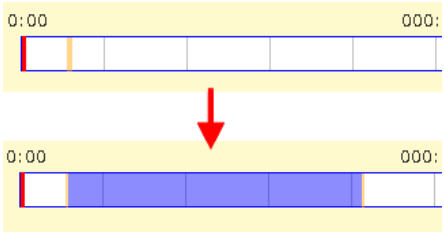


シナリオ作成画面に「target-img-20-m1.tiny.00」のタイムラインが表示されます。



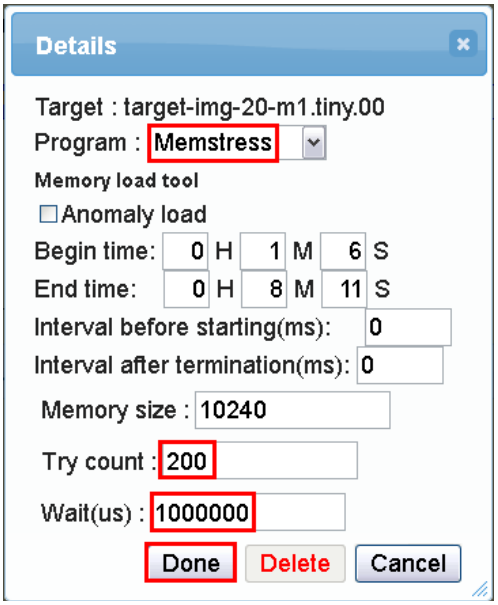
タイムライン上にベンチマークシナリオを追加します。

タイムライン上でクリックを行うと薄橙色のバーが表示されます。そのバーを右方向にドラッグし終了したいところでドロップしてください。

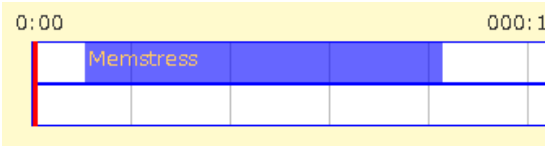


ドロップを行ったらベンチマーク設定のダイアログが表示されます。

Program の「Memstress」を選択し、パラメータを調整します。ここでは 1 秒おき（Wait=1000000 マイクロ秒）に指定したメモリサイズを確保し、それを 200 回（Try count=200）繰り返すことでメモリに負荷をかけます。ベンチマークプログラムのパラメータはお使いの環境にあわせて調整してください。調整完了後、「Done」ボタンをクリックしてください。

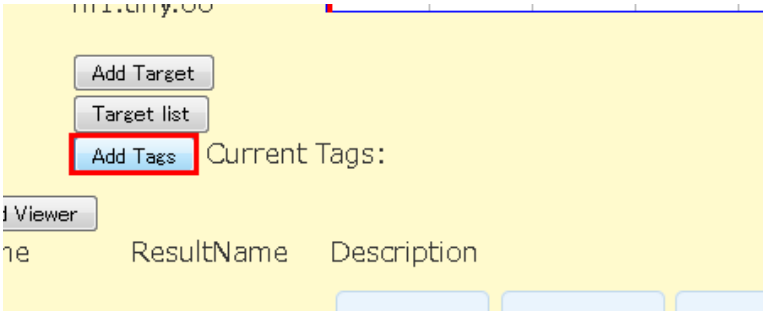


タイムライン上にベンチマークバーが表示されます。

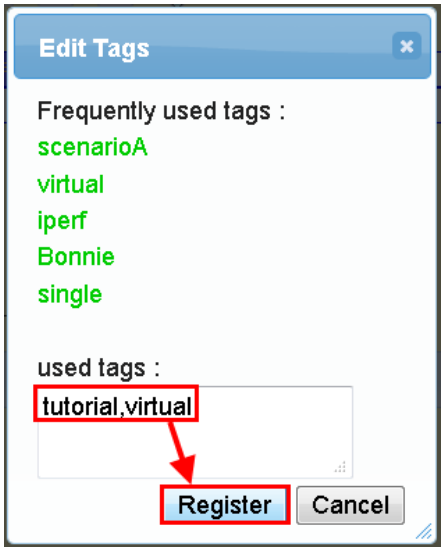


タグを設定します。設定は任意ですが、設定することでシナリオ結果を簡単に検索することが出来るようになります。

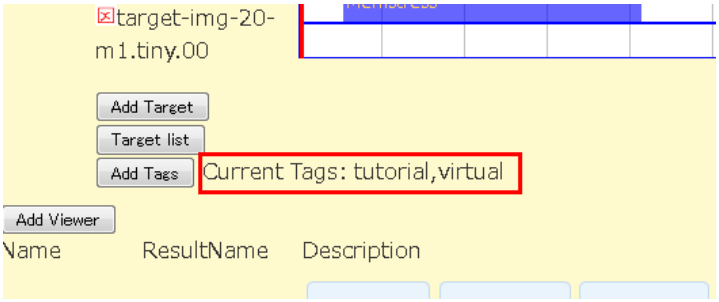
「Add Tags」ボタンをクリックしてください。



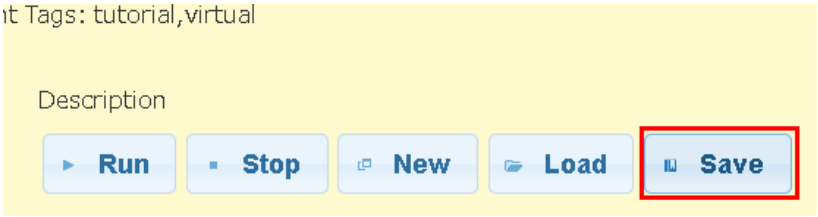
ダイアログが表示されるので、テキストエリアにタグを記入後、「Register」ボタンをクリックしてください。タグはカンマ区切りで複数指定できます。ここでは「tutorial」と「virtual」を登録します。



Current Tags に設定したタグが表示されます。

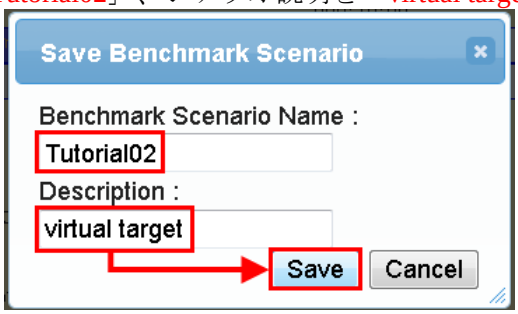


作成したシナリオを保存します。画面下部の「Save」ボタンをクリックしてください。



ダイアログが表示されます。シナリオ名、シナリオ説明を入力後、「Save」ボタンをクリックしてください。

ここではシナリオ名を「Tutorial02」、シナリオ説明を「virtual target」としています。

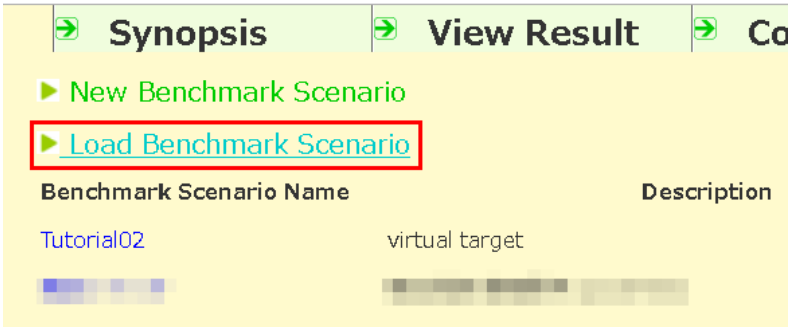


シナリオ作成は以上です。

3.2.3. ベンチマークシナリオの実行

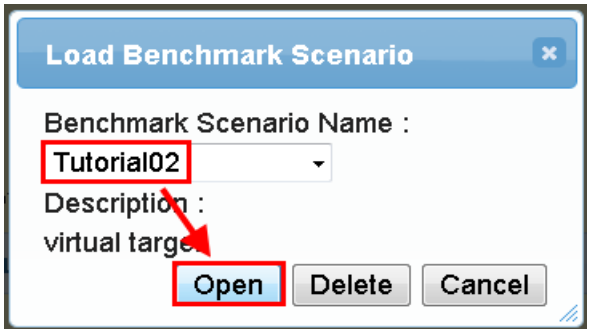
「3.2.3. ベンチマークシナリオの作成」で作成したシナリオを読み込みます。

DS-Bench メイン画面の「Load Benchmark Scenario」リンクをクリックしてください。



新規ウィンドウが立ち上がり、シナリオ選択ダイアログが表示されます。

「3.2.3. ベンチマークシナリオの作成」で作成したシナリオを選択し、「Open」ボタンをクリックしてください。ここでは「Tutorial02」を選択します。

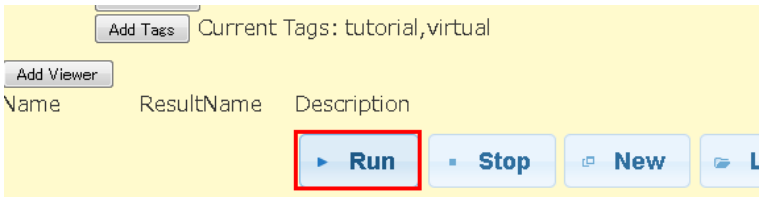


作成したベンチマークシナリオを読み込み、表示されます。

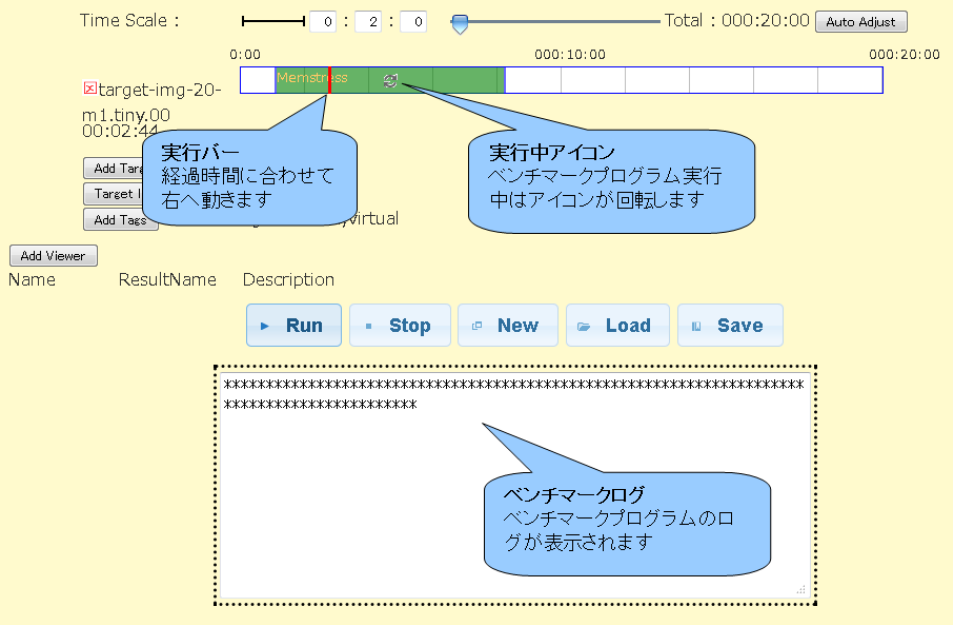
※シナリオの読み込みは以下の方法でも可能です。

- メイン画面の Load Benchmark Scenario リンク下部のシナリオ名リンク
  メイン画面 ＞ シナリオ名リンク
- ベンチマーク作成画面の「Load」ボタン
  メイン画面 ＞New Benchmark Scenario リンク ＞Load
- ベンチマークシナリオ設定画面
  メイン画面 ＞Configuration タブ ＞Benchmark Scenario＞ シナリオ名リンク

ベンチマークシナリオを実行します。画面下部の「Run」ボタンをクリックします。



実行中は以下の画面が表示されます。



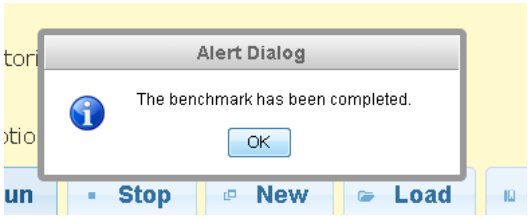
Memstress では実行中、ログ画面に領域を確保した回数「*」を表示します。

確保した分メモリ使用量が増えていくので仮想マシンのメモリをモニターすると状況がよくわかります。

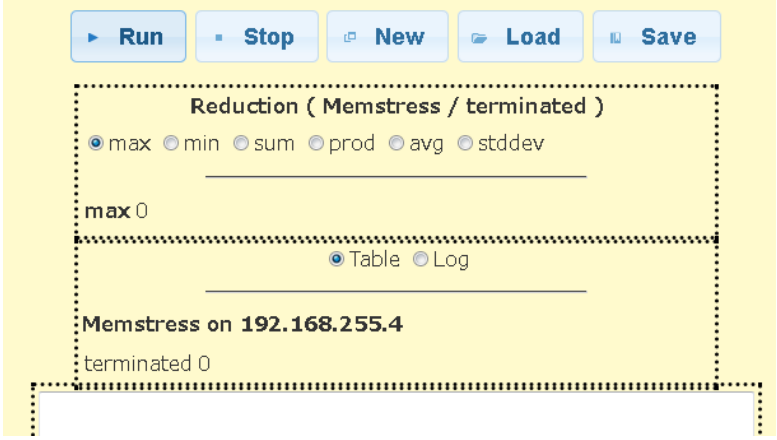
実行バーがベンチマークバーの終端に到達すると、シナリオの実行完了となります。

3.2.4. ベンチマークシナリオの結果確認

シナリオの実行が完了すると、ダイアログが表示されます。



ベンチマークシナリオの結果は下部に表示されます。



- リダクション結果

ベンチマークプログラムごとの実行結果を集計し、演算をします。

演算方法はラジオボタンで変更することができます。

max	min	sum	prod	avg	stddev
最大値	最小値	合計	総乗	平均	標準偏差

- ベンチマーク結果

ベンチマークプログラムの実行結果が表示されます。

ラジオボタンでベンチマーク結果の表示方法を変更することができます。

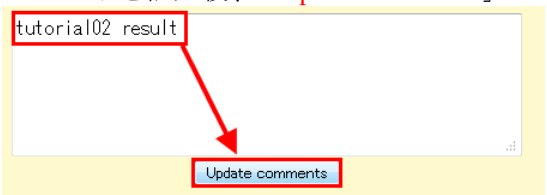
※ベンチマークツールによってはグラフ設定がされていないものがあります。この場合「Bar」、「Line」は表示されません。

- ◆ Table　結果がテーブル形式で表示されます。
- ◆ Bar　結果が棒グラフで表示されます。
- ◆ Line　結果が折れ線グラフで表示されます。
- ◆ Log　ベンチマーク実行ログが表示されます。

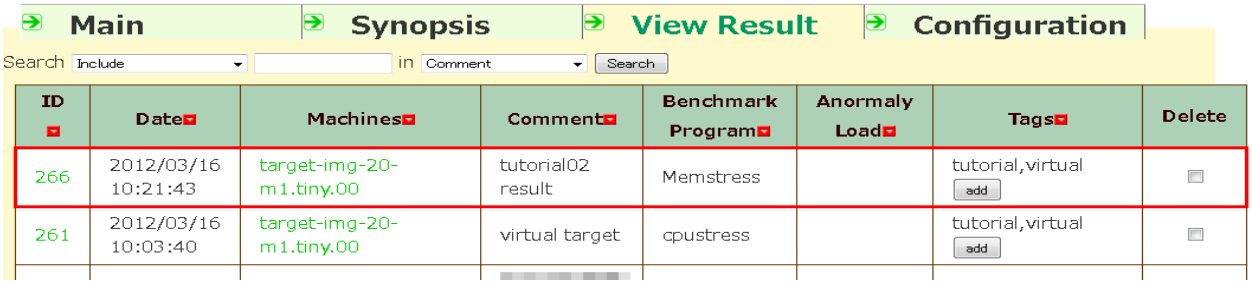
- コメントの登録

シナリオ実行結果に対して、コメントを登録することができます。

テキストエリアにコメントを記入後、「Update comments」ボタンをクリックしてください。



コメントを登録した場合、シナリオ実行結果一覧画面（View Result 画面）やシナリオ一覧画面（Synopsis 画面）の「Comment」列に表示され、どのようなシナリオの実行結果か一覧から確認できるようになります。



※ シナリオ実行結果の確認方法

- シナリオ実行結果一覧画面（View Result 画面）メイン画面 > View Result タブ > ID リンク
- シナリオ一覧画面（Synopsis 画面）メイン画面 > Synopsis タブ > ID リンク

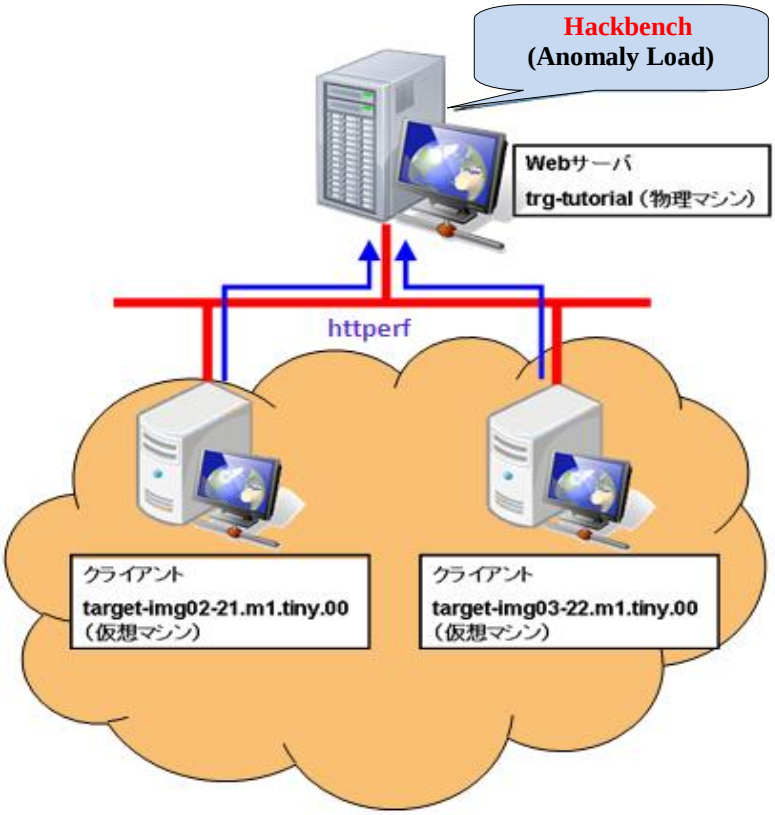
3.3.物理マシン・仮想マシンを複数台組み合わせたケース

マシンは3.1.と3.2.で作成した DS-Bench ターゲットを用います。ここでは仮想マシン 2 台をクライアント、物理マシンをサーバとした httperf を使用します。また、サーバ側では Anomaly Load として Hackbench を使用します。

Anomaly Load を設定した Web サーバに対して、それぞれの仮想クライアントから一秒間に 2 コネクション、総コネクション数が 200 回に到達するまで繰り返し行い、その時のレスポンスタイムを取得し結果を保存します。

環境構成は2.3 物理マシン・仮想マシンを複数台組み合わせた構成を、実行イメージは以下の図のように行います。

サーバ	trg-tutorial（物理マシン）
クライアント 1	target-img02-21.m1.tiny.00（仮想マシン）
クライアント 2	target-img03-22.m1.tiny.00（仮想マシン）



3.3.1. 事前準備

クライアントの仮想マシンを増設します。

OpenStack をインストールしたサーバへログインします。

```
$ cd /var/nova/images/test/
$ pwd
/var/nova/images/test/
$
$ ls –l
total 1991356
-rw-r--r-- 1 root root 1476395008 YYYY-MM-DD hh:mi lucid-server-cloudimg-amd64.img
-rw-rw-rw- 1 root root    4110432 YYYY-MM-DD hh:mi lucid-server-cloudimg-amd64-vmlinuz-virtual
-rw-r--r-- 1 root root 1476395008 YYYY-MM-DD hh:mi target-img
$
```

「3.2.1. 仮想 DS-Bench ターゲットの作成・登録」で仮想ターゲットイメージを保存したディレクトリへ移動します。ここでは、「/var/nova/images/test」になります。

「target-img」をコピーします。

```
$ sudo cp -p target-img target-img-02
$
$ ls –l
total 2986876
-rw-r--r-- 1 root root 1476395008 YYYY-MM-DD hh:mi lucid-server-cloudimg-amd64.img
-rw-rw-rw- 1 root root    4110432 YYYY-MM-DD hh:mi lucid-server-cloudimg-amd64-vmlinuz-virtual
-rw-r--r-- 1 root root 1476395008 YYYY-MM-DD hh:mi target-img
-rw-r--r-- 1 root root 1476395008 YYYY-MM-DD hh:mi target-img-02
$
```

コピーしたイメージを OpenStack へ登録します。同様にカーネルは「org-kernel」を使用します。

```
$ sudo nova-manage image image_register target-img-02 admin --name=target-img02 --kernel=00000001
--cont_format=ami --disk_format=ami
Image registered to 21 (00000015).
$
```

登録確認を行います。

```
$ glance index
ID          Name                        Disk Format       Container Format  Size
----------- --------------------------- ----------------- ----------------- -----------
21          target-img02                ami               ami                 1476395008
20          target-img                  ami               ami                 1476395008
1           org-kernel                  aki               aki                    4108960
$
```

同様に「target-img-03」を登録します。以下は登録後の確認です。

```
$ glance index
ID           Name                         Disk Format       Container Format  Size
------------ ---------------------------- ----------------- ----------------- -----------
22           target-img03                 ami               ami                 1476395008
21           target-img02                 ami               ami                 1476395008
20           target-img                   ami               ami                 1476395008
1            org-kernel                   aki               aki                    4108960
$
```

登録完了後、D-Cloud のデーモンを再起動します。

```
$ sudo /etc/init.d/dc-mapserver restart
Stopping  ... success
Starting  ... success
$
$ sudo /etc/init.d/dc-resource-dsb restart
Stopping  ... success
Starting  ... success
$
$ sudo /etc/init.d/dcld restart
Stopping  ... success
Starting  ... success
$
```

DS-Bench 画面にて、仮想ターゲットマシンが登録されていることを確認します。

main 画面 > Comfiguration 画面 > Target list

Target list

No	Machine name	Type	Status
1	trg-tutorial	physical	Active
2	target-img03-22-m1.tiny.00	virtual	Power off
3	target-img03-22-m1.tiny.01	virtual	Power off
4	target-img03-22-m1.tiny.02	virtual	Power off
5	target-img02-21-m1.tiny.00	virtual	Power off
6	target-img02-21-m1.tiny.01	virtual	Power off
7	target-img02-21-m1.tiny.02	virtual	Power off
8	target-img-20-m1.tiny.00	virtual	Power off
9	target-img-20-m1.tiny.01	virtual	Power off
10	target-img-20-m1.tiny.02	virtual	Power off

Refresh

次にサーバとなる物理マシンで Apache が起動しているか確認します。

物理マシンへログインし、いくつかプロセスが起動していることを確認します。

※DS-Bench ターゲット（物理マシン）に Apache がインストールされていなければ別途インストールを行ってください。

```
$ ps -ef | grep apache
root      1905     1  0 MMMDD ?        00:00:06 /usr/sbin/apache2 -k start
www-data  1907  1905  0 MMMDD ?        00:00:04 /usr/sbin/apache2 -k start
www-data  1909  1905  0 MMMDD ?        00:00:04 /usr/sbin/apache2 -k start
www-data  1911  1905  0 MMMDD ?        00:00:04 /usr/sbin/apache2 -k start
*******   8985  7025  0 hh:mi pts/0    00:00:00 grep apache
$
```

httperf にて使用する「index.html」が DocumentRoot に存在するかを確認します。

```
$ ls -l /var/www/index.html
-rw-r--r-- 1 root root 177 YYYY-MM-DD hh:mi /var/www/index.html
$
```

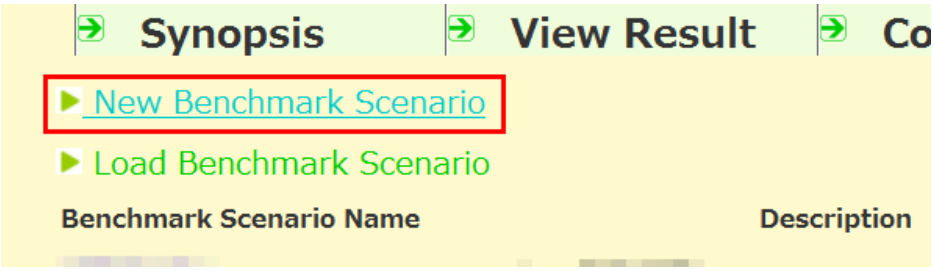
3.3.2. ベンチマークシナリオの作成

ブラウザを立ち上げ、DS-Bench メイン画面を開きます。

http://[DS-Bench コントローラの IP アドレス or ホスト名]/cgi-bin/main.cgi

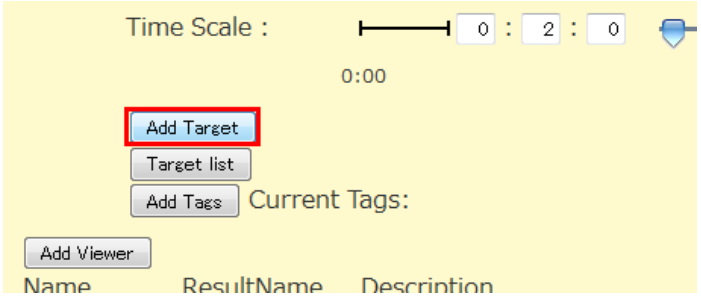
（DS-Bench コントローラのインストール時に指定した DocumentRoot と Apache の DocumentRoot が同じ場合）

メインメニューの「New Benchmark Scenario」リンクをクリックしてください。



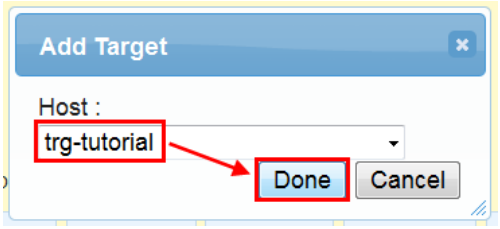
新規ウィンドウでシナリオ作成画面が開きます。

「Add Target」ボタンをクリックし、ベンチマーク対象のマシンを追加します。

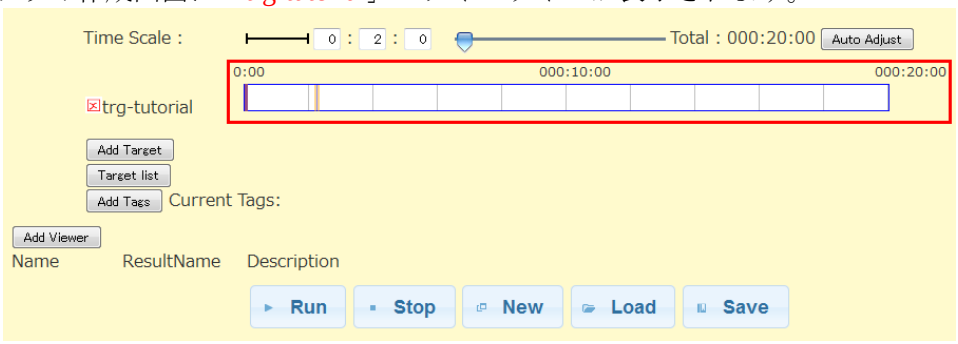


ダイアログが表示されるので、サーバとなる物理マシンを選択します。ここでは「trg-tutorial」を選択します。

trg-tutorial を選択後、「Done」ボタンをクリックします。

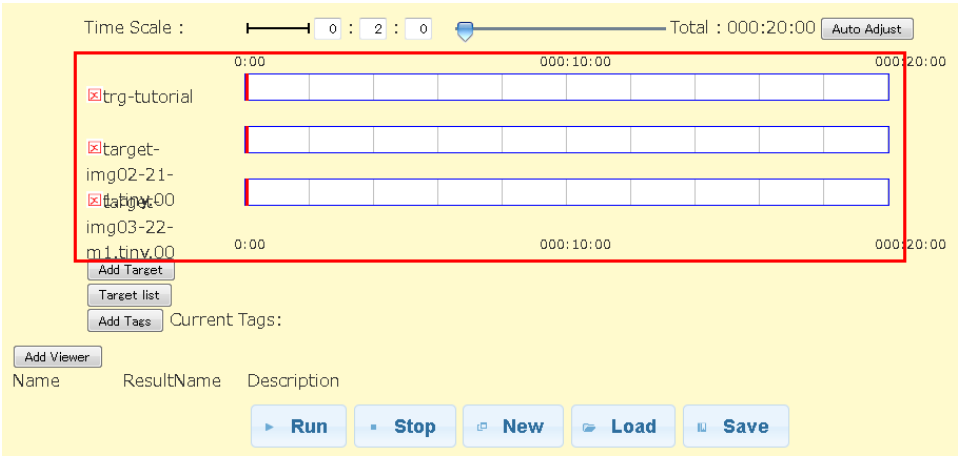


シナリオ作成画面に「trg-tutorial」のタイムラインが表示されます。



同様にクライアントとなる仮想マシン 2 台を追加します。

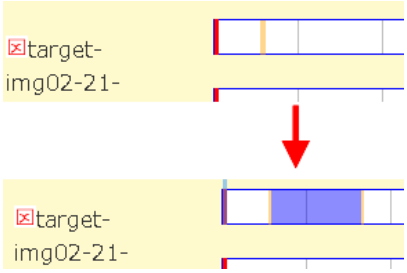
追加後、シナリオ作成画面は以下の内容となります。



ベンチマークプログラムを追加します。

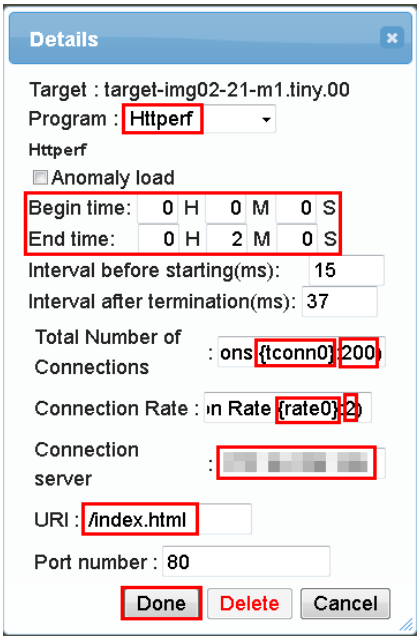
まず仮想マシンへ「httperf」を追加します。

いずれかの仮想マシンのタイムライン上でドラッグ＆ドロップします。



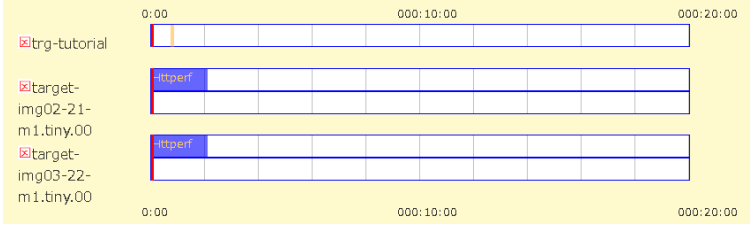
ベンチマーク設定のダイアログが表示されます。

Program の「httperf」を選択、開始時間「0H 0M 0S」、終了時間「0H 2M 0S」、コネクション数の合計（Total Number of Connections）「$(param:Total Number of Connections {tconn0}:200)」、秒間アクセス数（Connection Rate）「$(param:Connection Rate {rate0}:2)」、接続先サーバ（Connection server）「接続先 IP アドレス」、参照 URL「3.3.1 事前準備で確認した index.html の URL」を入力し、接続先を Web サーバ、1 秒間に 2 回のコネクションを 200 回に達するまで行うシナリオを作成し、「Done」ボタンをクリックしてください。



もう一方の仮想マシンにも同様に「httperf」を追加します。

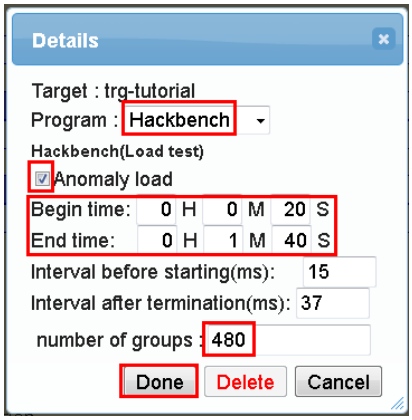
※「$(param～)」で始まるパラメータは D-Case ditor から実行する際に、値を変更するためのものなので、ここではデフォルトのままお使いください。説明は3.4.1を参照してください。



次にサーバ側で負荷を発生させるため、物理マシン「trg-tutorial」へ「Hackbench」を追加します。物理マシンのタイムライン上で同様にドラッグ＆ドロップしてください。

ベンチマーク設定のダイアログが表示されるので、以下の項目を選択します。

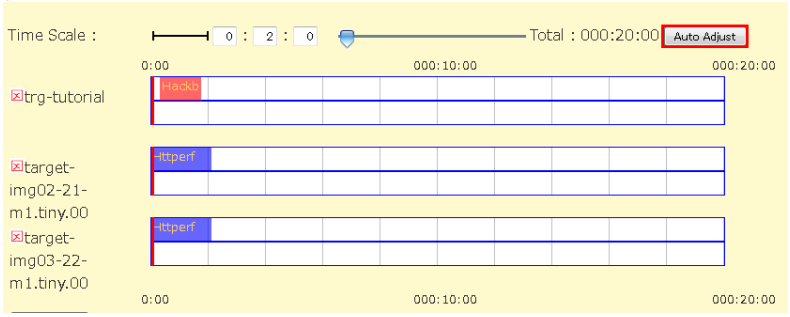
Program の「Hackbench」を選択、Anomaly load にチェック、開始時間「0H 0M 20S」、終了時間「0H 1M 40S」、負荷の大きさ（number of groups）「480」を入力し、「Done」ボタンをクリックしてください。



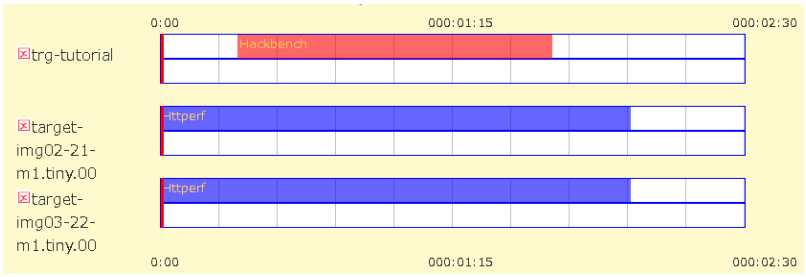
追加後のシナリオ作成画面は以下になります。

タイムラインのスケールが合っていないので、自動調整を行います。

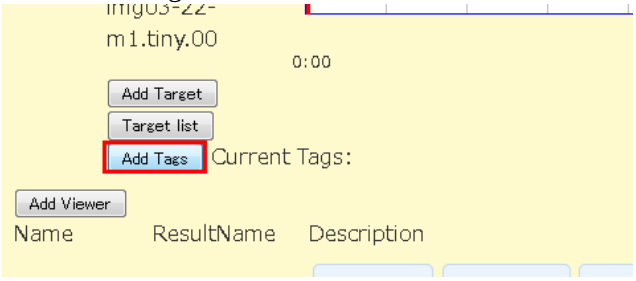
「Auto Adjust」ボタンをクリックしてください。



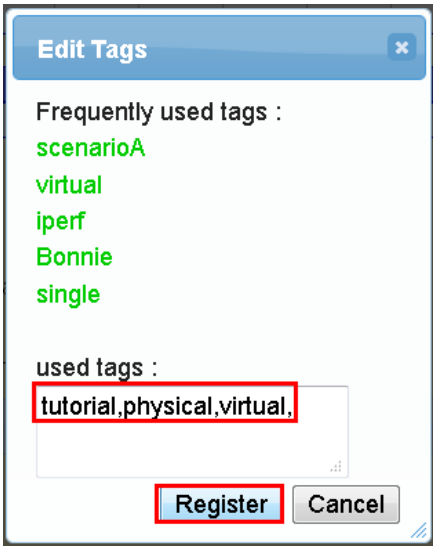
スケールが自動的に調整され、見やすくなりました。



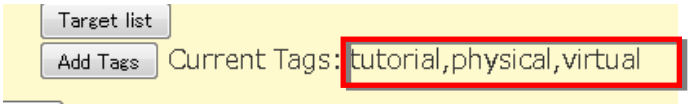
タグを追加します。「Add Tags」ボタンをクリックしてください。



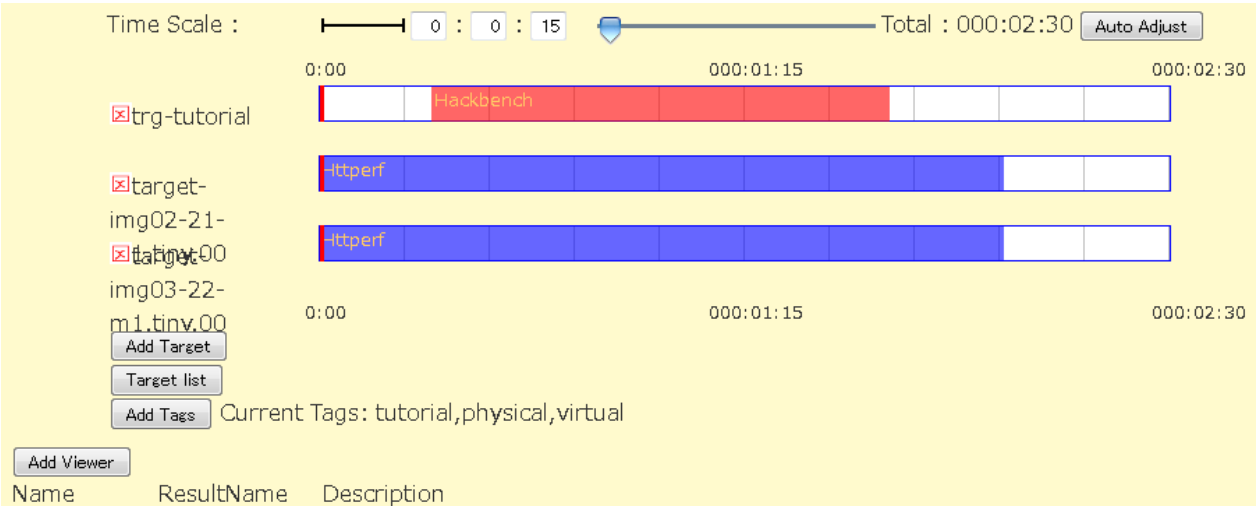
ダイアログが表示されるので、ここでは「tutorial」「physical」「virtual」を登録します。入力後、「Register」ボタンをクリックしてください。



シナリオ作成画面にタグが追加されます。

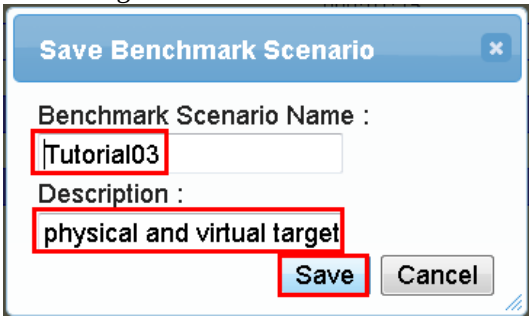


入力完了後のシナリオ作成画面は以下になります。



作成したシナリオを保存します。

「Save」ボタンをクリックします。ダイアログが表示されるので、シナリオ名へ「Tutorial03」、説明文へ「physical and virtual target」と入力し、「Save」ボタンをクリックします。

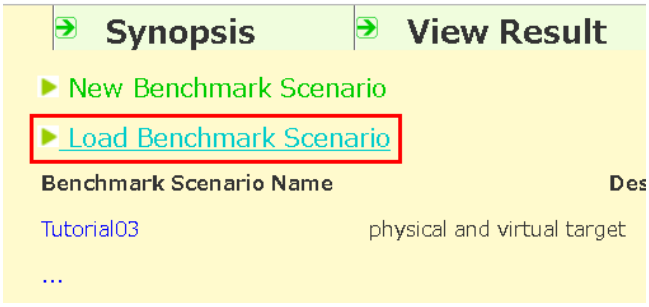


シナリオ作成は以上です。

3.3.3. ベンチマークシナリオの実行

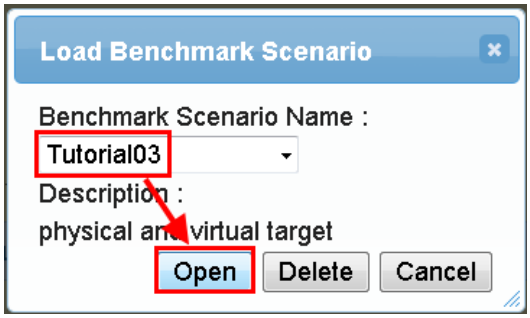
「3.3.2. ベンチマークシナリオの作成」で作成したシナリオを読み込みます。

DS-Bench メイン画面の「Load Benchmark Scenario」リンクをクリックしてください。



新規ウィンドウが立ち上がり、シナリオ選択ダイアログが表示されます。

ここでは先ほど作成した「Tutorial03」シナリオを選択し、「Open」ボタンをクリックしてください。



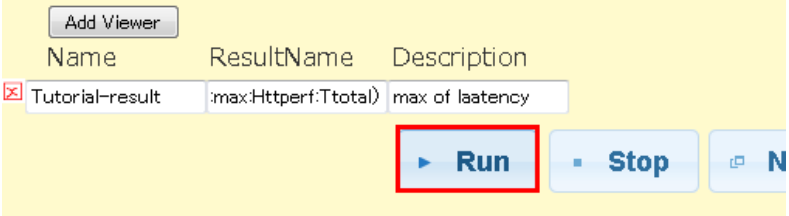
作成したベンチマークシナリオを読み込み、表示されます。

※シナリオの読み込みは以下の方法でも可能です。

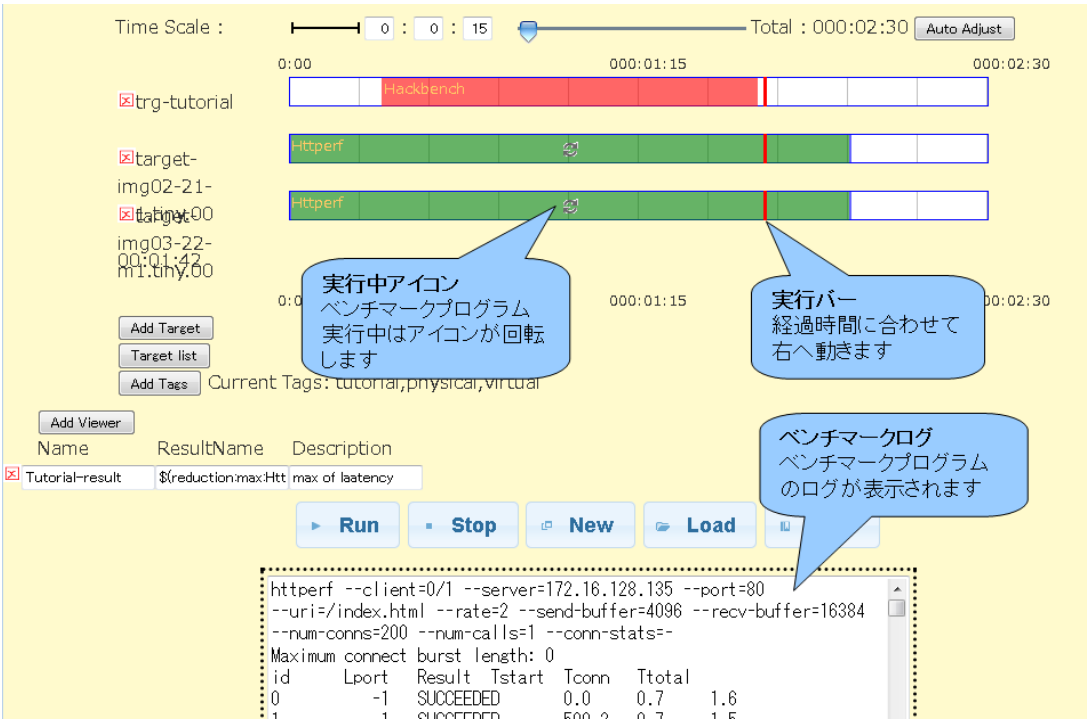
- メイン画面の Load Benchmark Scenario リンク下部のシナリオ名リンク
  メイン画面 > シナリオ名リンク
- ベンチマーク作成画面の「Load」ボタン
  メイン画面 > New Benchmark Scenario リンク > Load
- ベンチマークシナリオ設定画面
  メイン画面 > Configuration タブ > Benchmark Scenario > シナリオ名リンク

ベンチマークシナリオを実行します。画面下部の「Run」ボタンをクリックします。

※Add Viewer については次の D-Case Editor との連携で説明します。



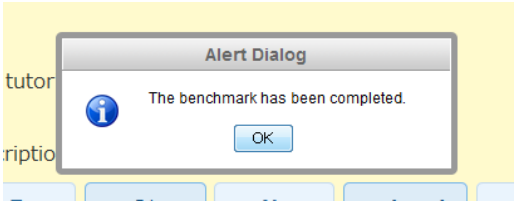
実行中は以下の画面が表示されます。



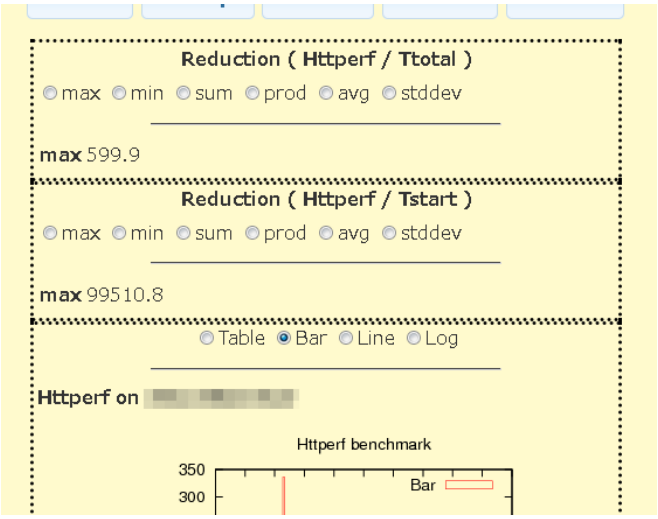
実行バーがベンチマークバーの終端に到達すると、シナリオの実行完了となります。

3.3.4. ベンチマークシナリオの結果確認

シナリオの実行が完了すると、ダイアログが表示されます。



ベンチマークシナリオの結果は下部に表示されます。



- リダクション結果

ベンチマークプログラムの実行結果を集計し、演算をします。

演算方法はラジオボタンで変更することができます。

以下の場合、「httperf」の最大応答時間が「599.9 ミリ秒」を表しています。

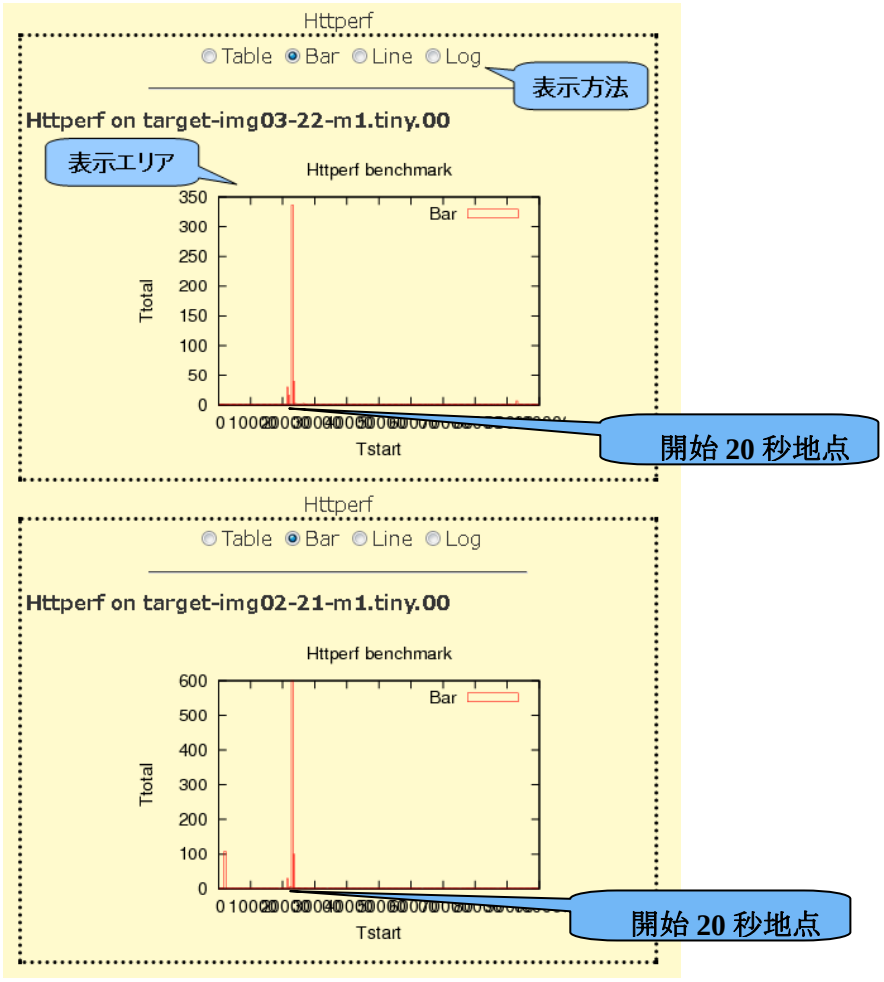
max	min	sum	prod	avg	stddev
最大値	最小値	合計	総乗	平均	標準偏差

- ベンチマーク結果

ベンチマークプログラムの実行結果が表示されます。

ラジオボタンでベンチマーク結果の表示方法を変更することができます。

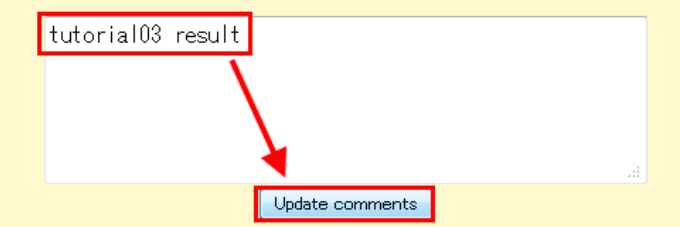
以下の図よりサーバ側で開始から 20 秒後に負荷（Hackbench）が発生し、その時に応答時間が遅くなっていることが確認できます。



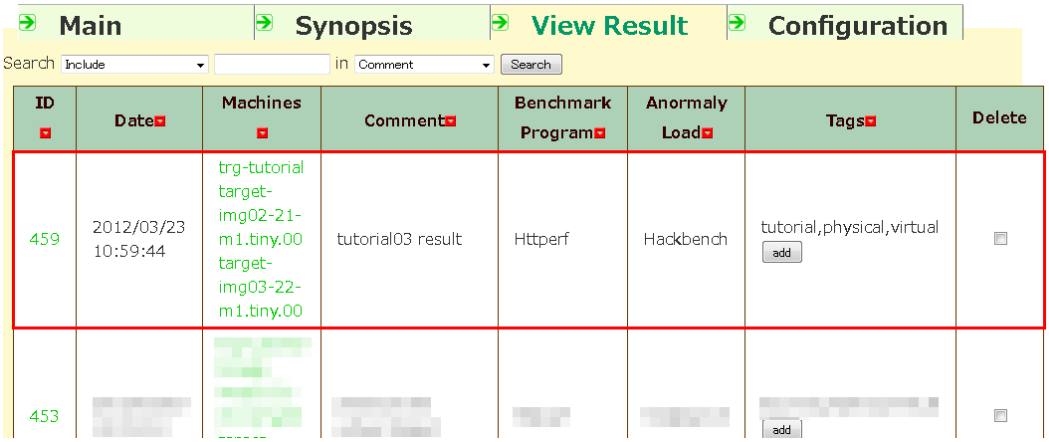
- コメントの登録

  シナリオ実行結果に対して、コメントを登録することができます。

  テキストエリアにコメントを記入後、「Update comments」ボタンをクリックしてください。



コメントを登録した場合、シナリオ実行結果一覧画面（View Result 画面）やシナリオ一覧画面（Synopsis 画面）の「Comment」列に表示され、どのようなシナリオの実行結果か一覧から確認できるようになります。



※ シナリオ実行結果の確認方法

- シナリオ実行結果一覧画面（View Result 画面）

  メイン画面 > View Result タブ > ID リンク

- シナリオ一覧画面（Synopsis 画面）

  メイン画面 > Synopsis タブ > ID リンク

3.4. D-CaseEditor と連携を行うケース

はじめに、D-Case については http://www.dependable-os.net/osddeos/concept.html の 3 D-Case を参照してください。D-Case Editor は http://www.dependable-os.net/tech/D-CaseEditor/D-Case_Editor_J.html を、D-Case Weaver については http://www.dependable-os.net/tech/DCaseWeaver/index_J.html を参照してください。

D-Case の作成時において、ベンチマークの測定結果を Evidence にする必要が生じた場合に、DS-Bench を D-Case Editor/D-Case Weaver と連携させ使用します。

ここでは、Web サーバにアクセス数が毎分 1500 回以下であれば、レスポンスタイムは 3 秒以内であることを検証します。

マシンは DS-Bench コントローラ、D-Cloud コントローラ+OpenStack コントローラ、Web サーバ、クライアントとして3.2で作成した DS-Bench ターゲット（仮想マシン）を、2.4. 構成で使用します。

以上の検証を行うためのシナリオを DS-Bench で作成します。

ここでは仮想マシンをクライアント、Web サイトをサーバとした httperf を使用します。

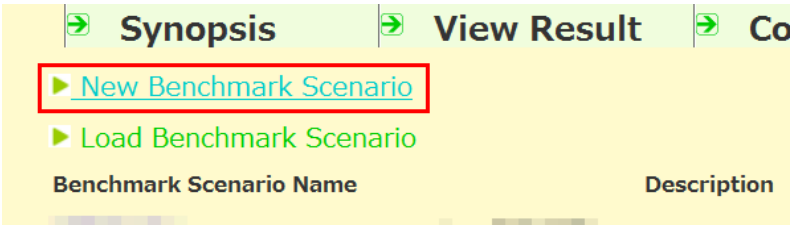
D-Case Editor にて作成したシナリオを読み込み、実行しゴールとして「最大応答時間が 3 秒未満」を設定し検証します。

※対象の Web サーバは、Apache をインストールし、/var/www/に index.html ファイルがあることを確認し、この index.html を対象に httperf を実行します。

3.4.1. シナリオの新規作成

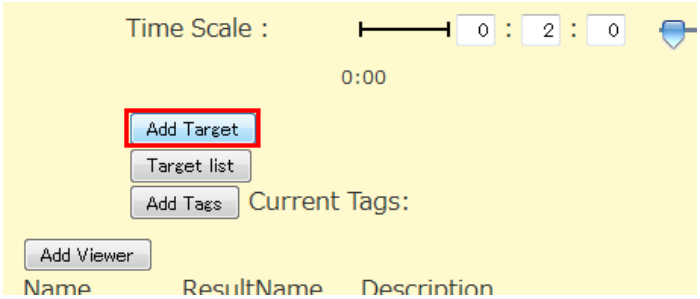
検証を行うため、1 分間に Web サーバに対して 1500 回アクセスを行い、そのレスポンスタイムを取得するシナリオを作成します。

シナリオの作成はブラウザの DS-Bench メイン画面より「**New Benchmark Scenario**」をクリックしてください。



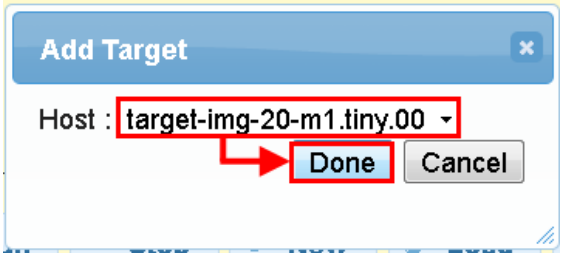
新規ウィンドウでシナリオ作成画面が開きます。

「Add Target」ボタンをクリックし、ベンチマーク対象のマシンを追加します。

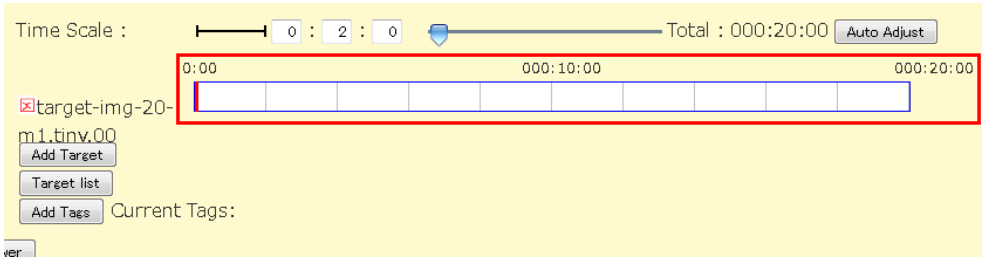


ダイアログが表示されるので、クライアントになる仮想マシンを選択します。ここでは「target-img-20-m1.00」を選択します。

Target-img-20-m1.00 を選択後、「Done」ボタンをクリックします。



シナリオ作成画面に「target-img-20-m1.00」のタイムラインが表示さるのでタイムライン上でドラッグ＆ドロップを行います。

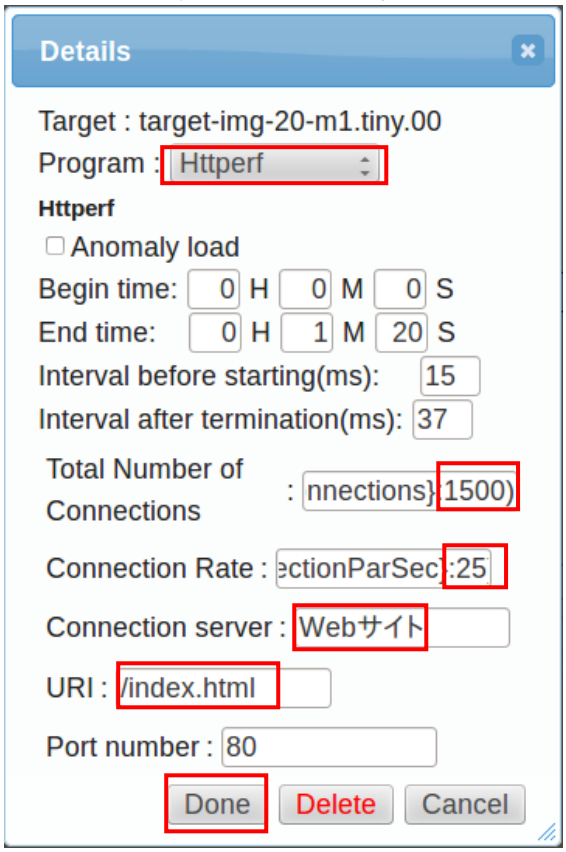


ベンチマーク設定のダイアログが表示されます。

Program に使用するベンチマーク「Httperf」を選択し、開始、終了時間を設定します。

1 分間に Web サーバに対して 1500 回アクセスを行うため、総コネクション数を 1500、毎秒のコネクション数を 25 と設定するため、コネクション数の合計（Total Number of Connections）「$(param:Total Number of Connections {TotalConnections}:1500)」、秒間アクセス数（Connection Rate）「$(param:Connection Rate {ConnectionParSec}:25)」、接続先サーバ（Connection server）「接続先 IP アドレス」、参照 URL「index.html の URL」を入力し、「Done」ボタンをクリックしてください。

※接続先 IP アドレスは Web サイトと表示されていますが Web サイトの IP で指定ください。



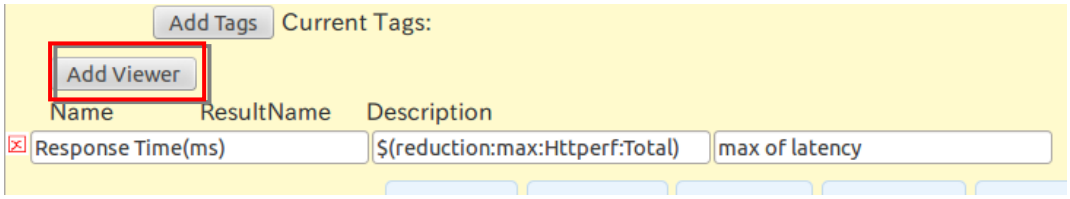
※「$(param～)」で始まるパラメータは D-Case Editor から実行する際に、値を変更するためのものです。なお入力規則は以下の通りです。

$(param:[説明]{[変数名]}:[値])

- [説明] 変数名の説明文です。任意の値を指定してください。
- [変数名] D-Case Editor で表示される変数名です。シナリオ内で一意となる値を指定してください。
- [値] 変数へ代入する値です。お使いの環境に合わせて指定してください。

D-Case Editor 側でレスポンスが 3 秒以内か判定を行うため、実行結果変数を設定します。

「Add Viewer」ボタンをクリックしてください。シナリオ作成画面に 3 つのテキストボックスが表示されるので、それぞれに値を入力します。



- Name 変数名を入力します。ここでは「Response Time(ms)」とします。
- Result Name 変数の値を設定します。入力規則は以下の通りです。

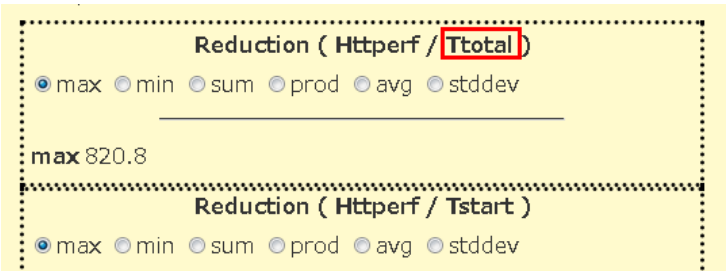
  $(reduction:[演算方法]:[ベンチマークプログラム名]:[ラベル名])

  今回は Httperf のリダクション結果の最大値を取得したので、入力内容は以下になります。

  $(reduction:max:Httperf:Ttotal)

  ※ラベル名「Ttotal」は応答時間のリダクション結果値の変数です。

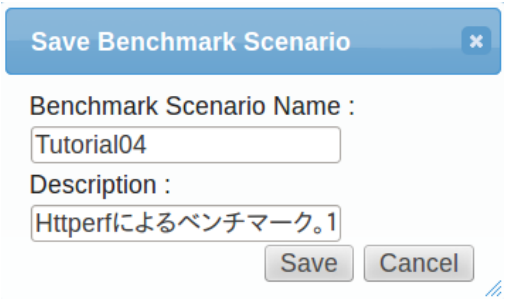
  （リダクション結果の以下の値がラベル名です）



- Description 変数の説明文を入力します。ここでは「max of latency」と入力します。

作成したシナリオを保存します。

「Save」ボタンをクリックします。ダイアログが表示されるので、シナリオ名(Benchemark Scenario Name)へ「Tutorial04」、説明文へ「Httperf によるベンチマーク。1 秒間に **{ConnectionParSec}**回 Web サーバにアクセスし、計**{TotalConnections}**回に達するまで実行します。期待値は**({Response Time(ms)}[ms])**を満たすかテストします」と入力し、「Save」ボタンをクリックします。



ここで指定した説明文(Description)が、D-Case 側でシナリオを呼び出したときの Goal ノードの説明として表示されます。

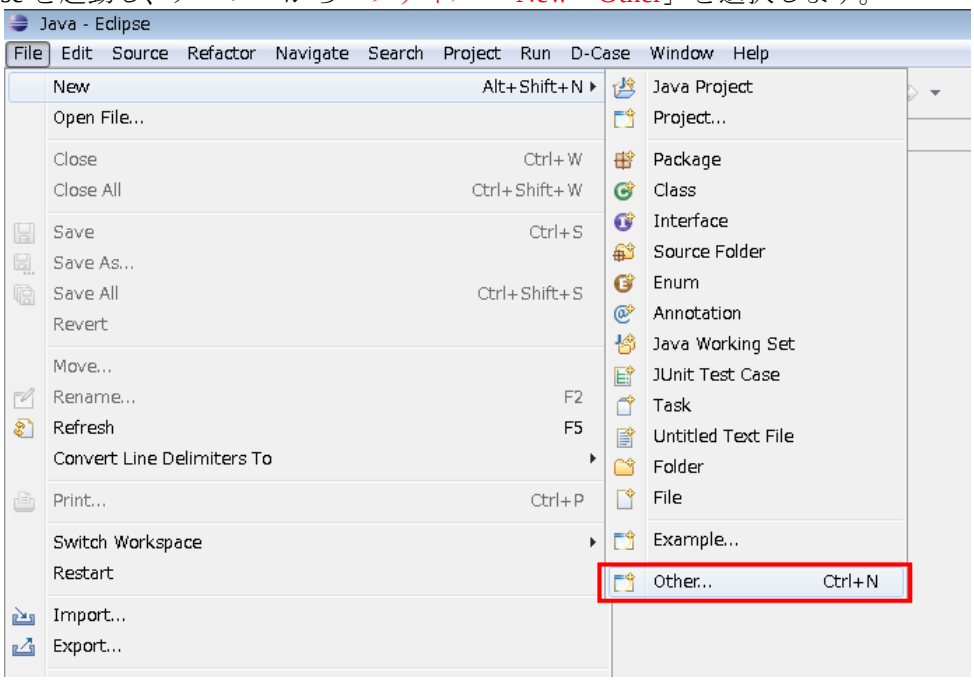
シナリオ作成時、(param:[説明] {[変数名]}:[値])で設定した変数名を{}で説明文に記述すると、呼び出された状態ではその変数名の値が表示されます。

シナリオ作成は以上です。

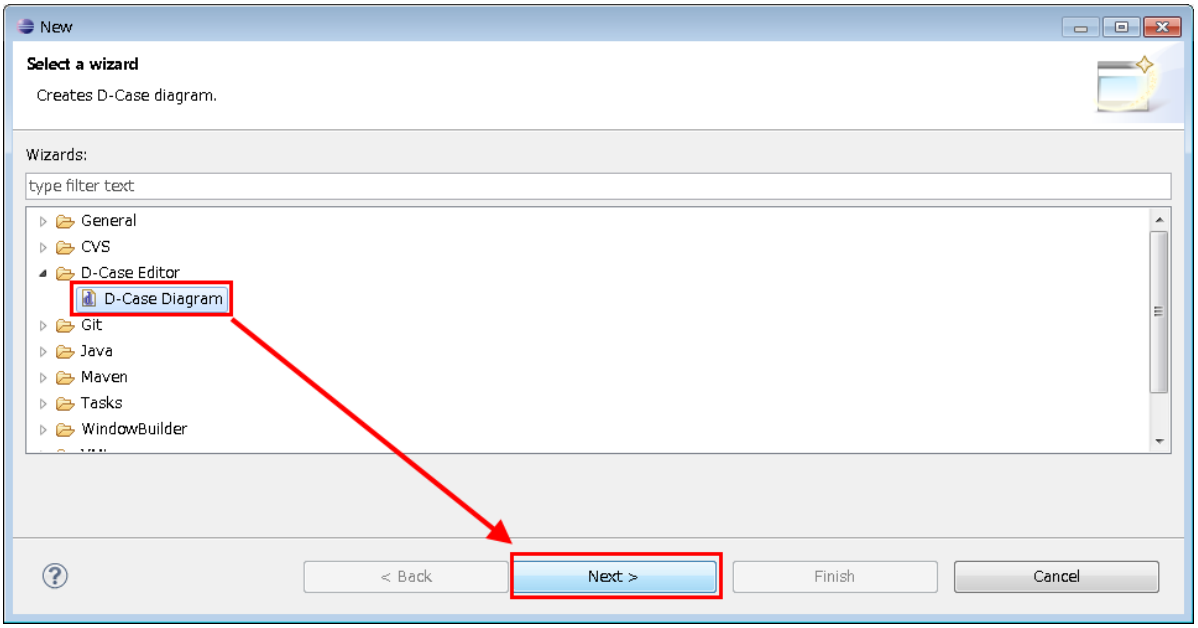
3.4.2. D-Case ダイアグラムの新規作成

D-Case Editor からベンチマークシナリオを実行するための事前準備を行います。

Eclipse を起動し、メニューから「ファイル ＞New＞Other」を選択します。

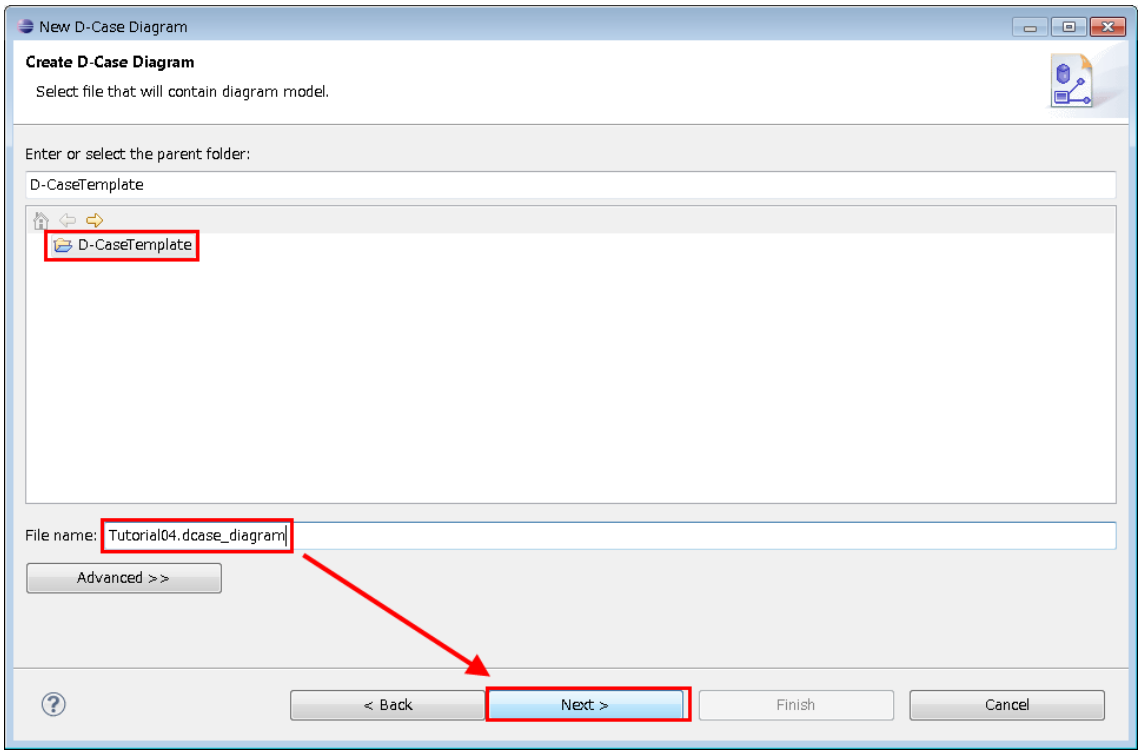


「D-Case Editor＞D-Case Diagram」を選択し、「next」ボタンをクリック



ここでは「D-CaseTemplate」フォルダの下へ新規作成します。

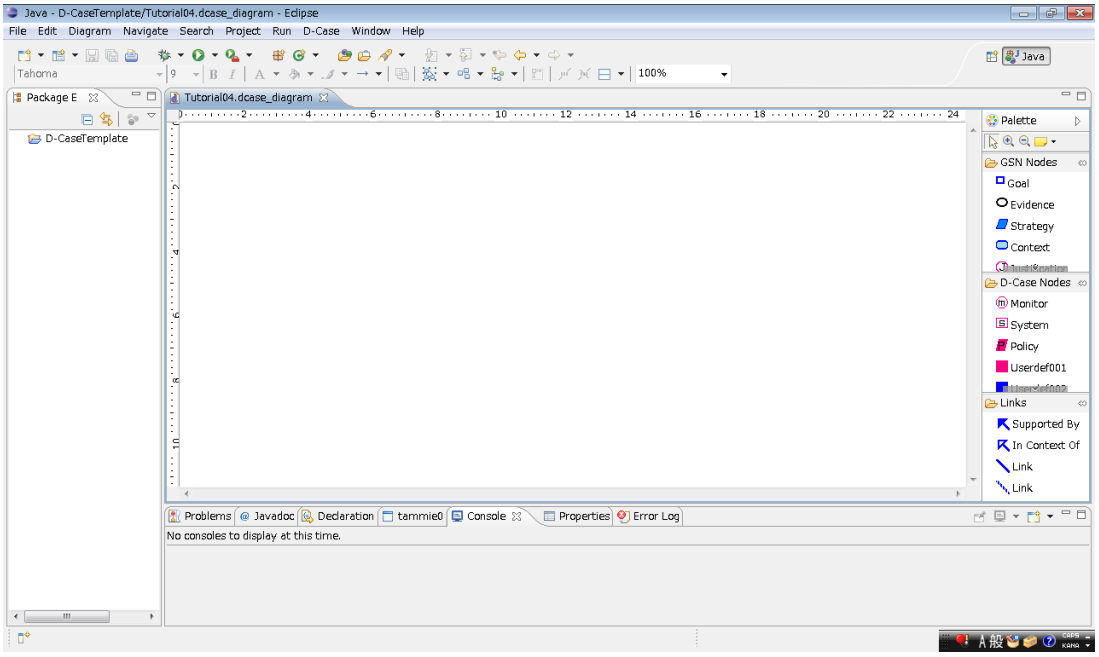
Diagram 名を指定し、拡張子は「.dcase_diagram」です。ここでは「Tutorial04.dcase_diagram」とし、ファイル名を指定後、「Next」ボタンをクリックします。



model 名を指定します。

自動で model 名が入っていますので、そのまま Finish ボタンをクリックします。

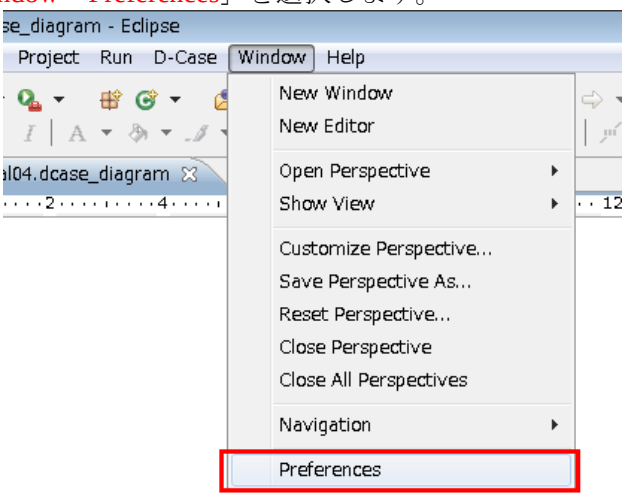
D-Case ダイアグラムが新規作成されました。



3.4.3. DS-Bench 連携のための設定

DS-Bench と連携を行うための設定をします。

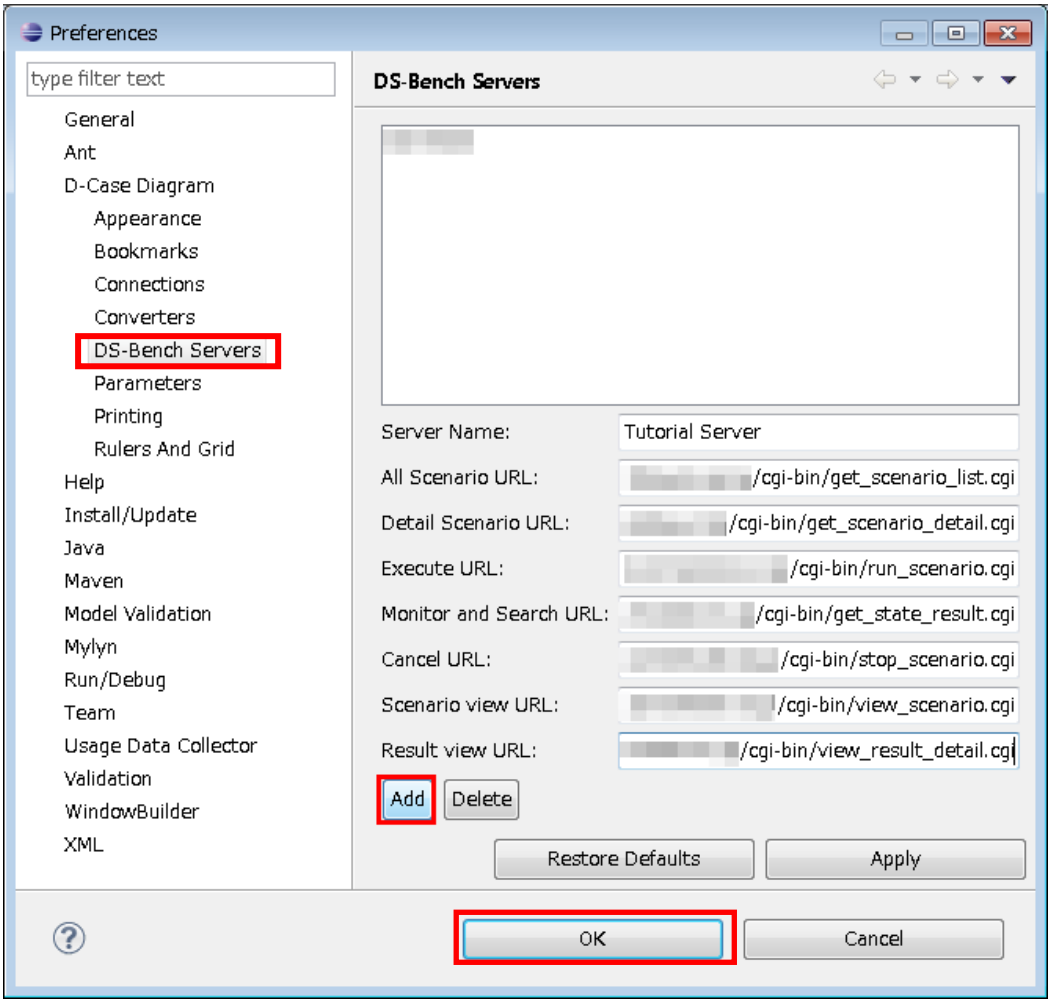
メニューから「Window > Preferences」を選択します。



「D-Case Diagram > DS-Bench Servers」を選択します。

DS-Bench の設定をします。

設定完了後、「Add」ボタンをクリックしてください。



- **Server Name**

  任意で指定してください。D-Case Editor 内での DS-Bench サーバ名です。

- **All Scenario URL**

  DS-Bench からシナリオリストを取得するための URL です。

  http://[DS-Bench コントローラの IP アドレス or ホスト名]/cgi-bin/get_scenario_list.cgi

- **Detail Scenario URL**

  DS-Bench からシナリオ詳細情報を取得するための URL です。

  http://[DS-Bench コントローラの IP アドレス or ホスト名]/cgi-bin/get_scenario_detail.cgi

- **Execute URL**

  D-Case Editor からシナリオを実行するための URL です。

  http://[DS-Bench コントローラの IP アドレス or ホスト名] /cgi-bin /run_scenario.cgi

- **Monitor and Search URL**

  DS-Bench からシナリオ実行状況を取得するための URL です。

  http://[DS-Bench コントローラの IP アドレス or ホスト名] /cgi-bin /get_state_result.cgi

- **Cancel URL**

  D-Case Editor からシナリオを中止するための URL です。

http://[DS-Bench コントローラの IP アドレス or ホスト名] /cgi-bin /stop_scenario.cgi

- **Scenario View URL**

  DS-Bench からシナリオ実行結果を取得するための URL です。

  http://[DS-Bench コントローラの IP アドレス or ホスト名] /cgi-bin /view_scenario.cgi

- **Result View URL**

  DS-Bench のシナリオ実行結果画面を表示するための URL です。

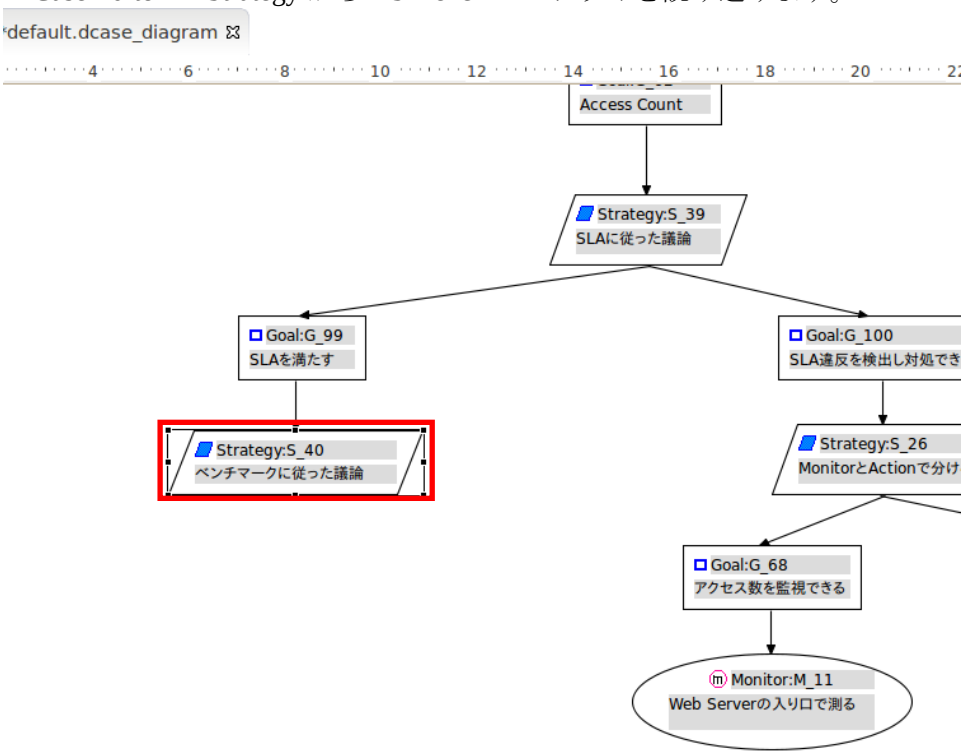
  http://[DS-Bench コントローラの IP アドレス or ホスト名] /cgi-bin /view_result_detail.cgi

設定した DS-Bench サーバ名が上部エリアへ表示されます。

表示確認後、OK ボタンをクリックしてください。

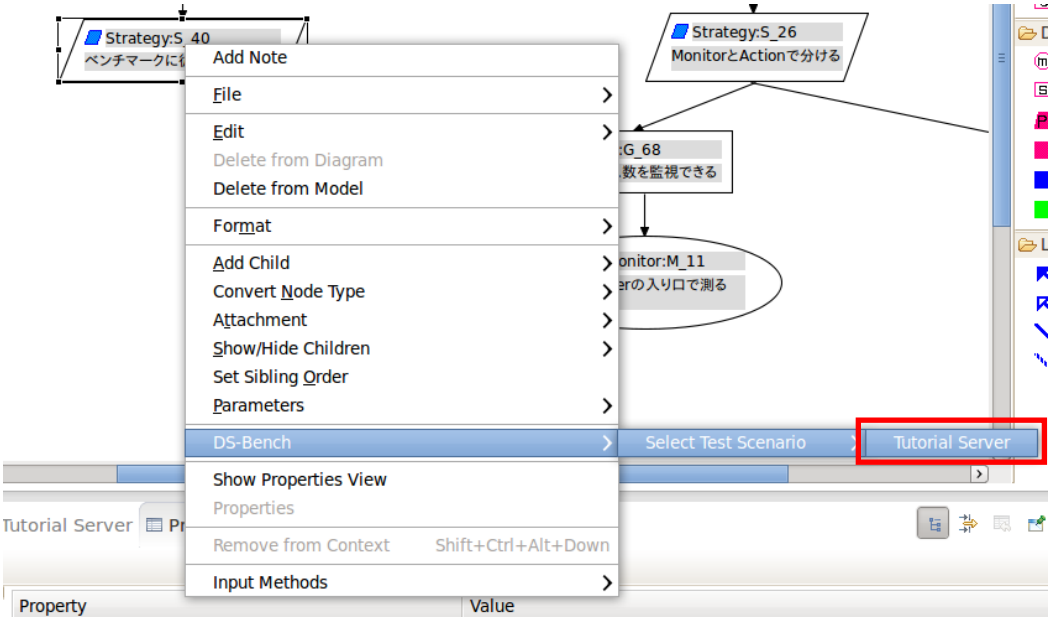
3.4.4. ベンチマークシナリオのインポート

D-Case Editor の Strategy から DS-Bench のシナリオを読み込みます。



ベンチマークシナリオのリストを取得します。

「Strategy を右クリック > DS-Bench > Select Test Scenario > [3.4.2.で設定した Server Name]」を選択します。

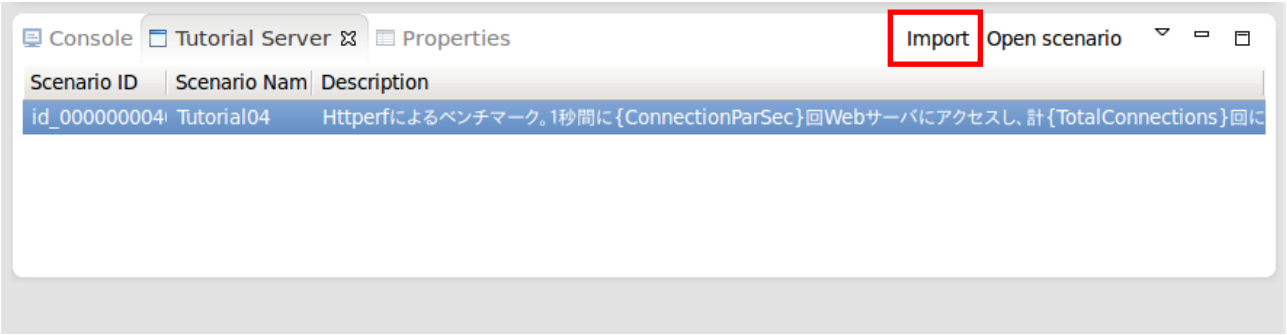


下部にシナリオリストが表示されます。

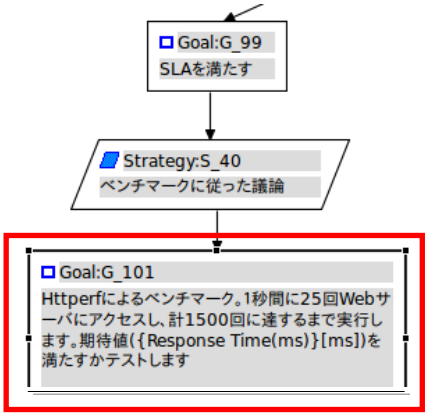
3.4.5. Goal ノードの追加（ベンチマークシナリオのインポート）

DS-Bench で作成したシナリオを D-Case Editor と関連付けます。

シナリオリストから実行するシナリオを選択し、「Import」ボタンをクリックします。
ここでは3.3.2.で作成した「Tutorial04」を選択します。

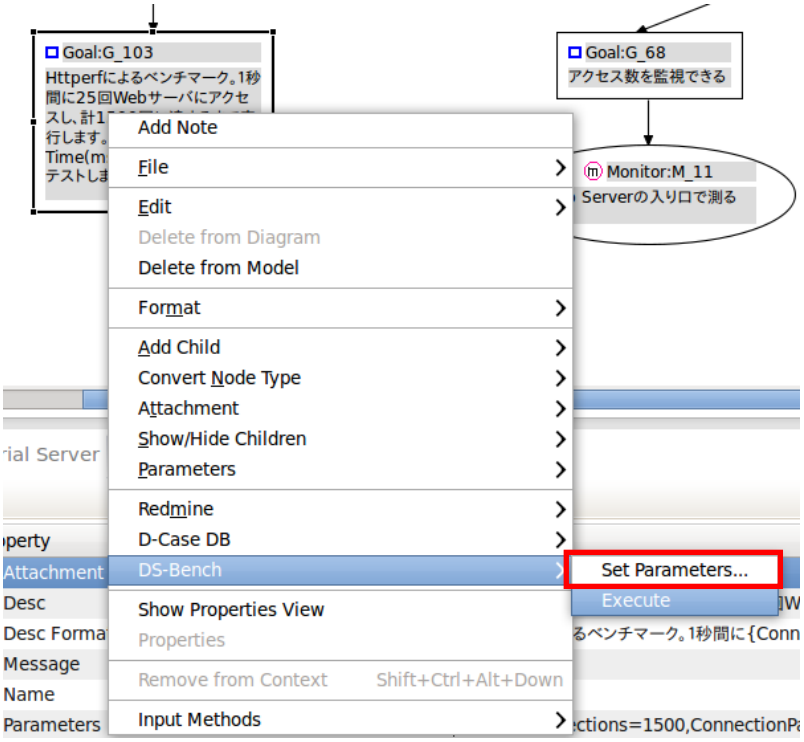


キャンバス上に、Goal ノードが追加されます。



DS-Bench で設定したシナリオのパラメータを指定します。

「Goal ノードを右クリック ＞DS-Bench＞Set Parameters…」を選択

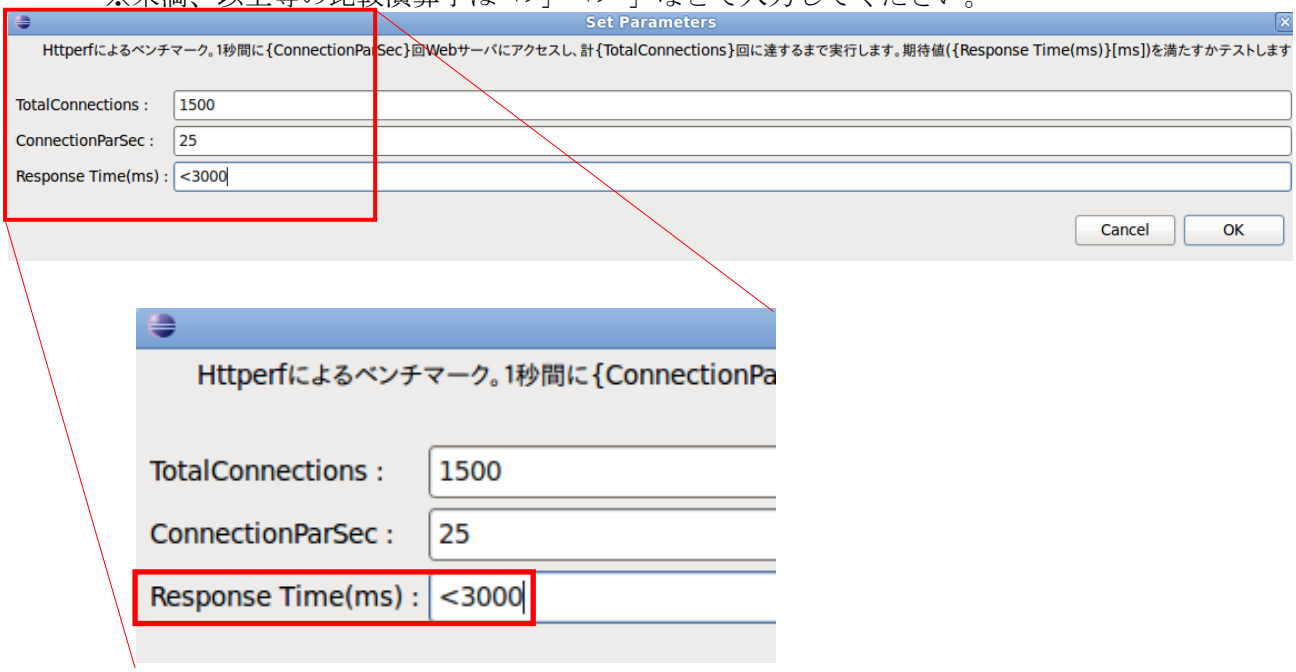


パラメータ設定のダイアログが表示されるので、各パラメータを指定します。

パラメータ指定後、OK ボタンをクリックしてください。

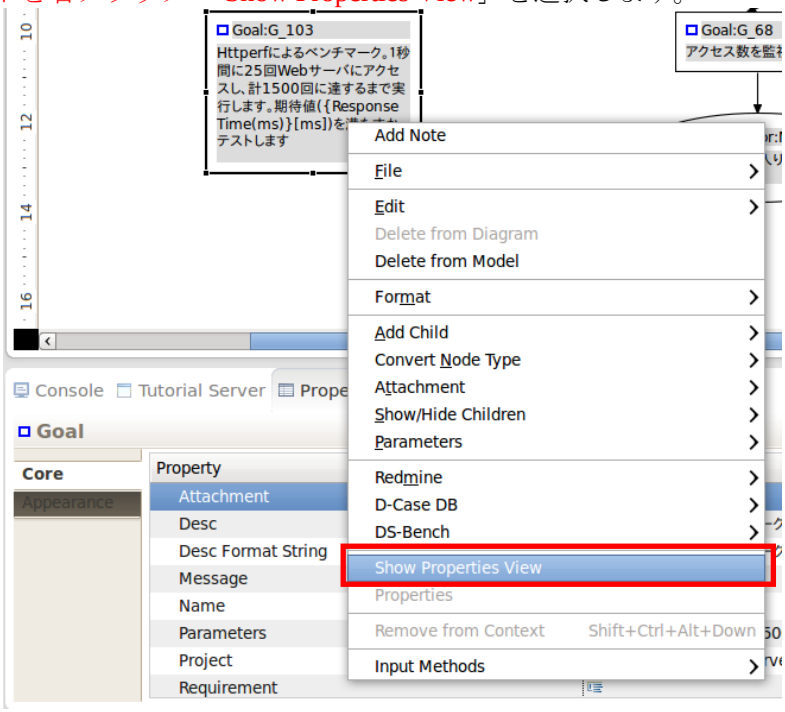
作成した「Tutorial04」シナリオに対して、httperf の最大応答時間（変数名：Respons Time(ms)）が 3 秒未満（<3000 ミリ秒）となるゴールを設定します。

※未満、以上等の比較演算子は「>」「>=」などで入力してください。



設定したパラメータを確認します。

「Goal ノードを右クリック > Show Properties View」を選択します。



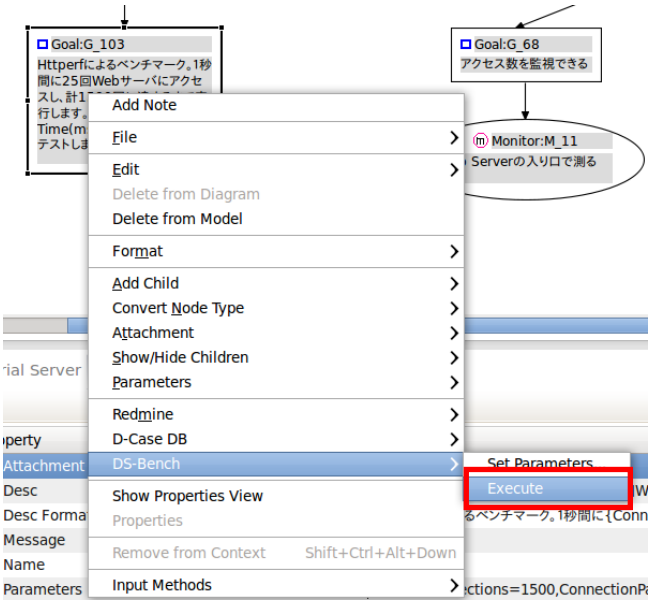
下部にプロパティが表示されます。

Parameters に設定したパラメータが表示されていることを確認します。

3.4.6. ベンチマークシナリオの実行

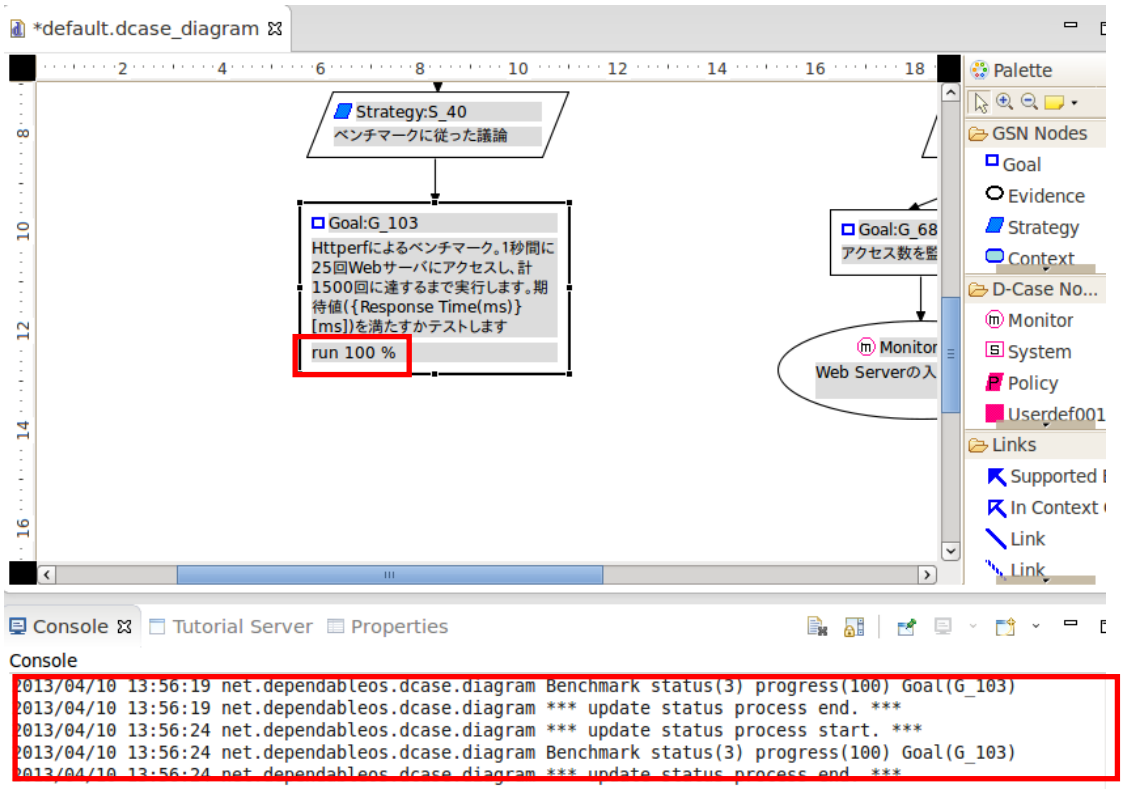
D-Case Editor からベンチマークシナリオを実行します。

「Goal ノードを右クリック > DS-Bench > Execute」を選択します。



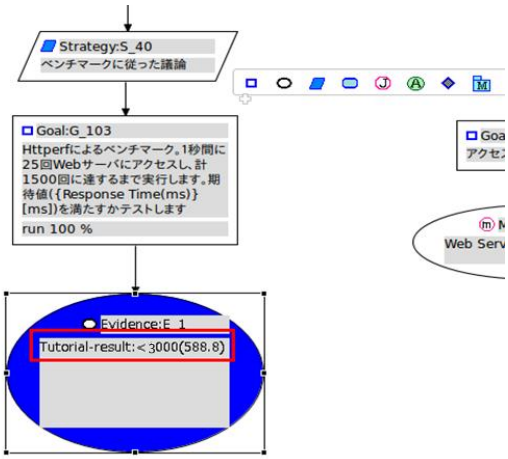
モニタリングを行うか、行わないかを確認するダイアログが表示されます。
ここではモニタリングを実施するため、「OK」ボタンをクリックします。

実行中はログが下部に表示されます。また進捗率が Goal ノードに表示されます。



3.4.7. ベンチマークシナリオの結果確認、評価

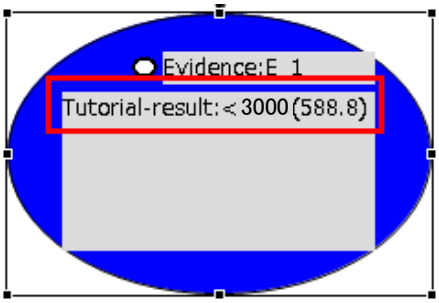
ベンチマークシナリオを実行後、自動的に結果が表示されます。
ベンチマークシナリオが終了すると、自動的に Evidence ノードが作成されます。



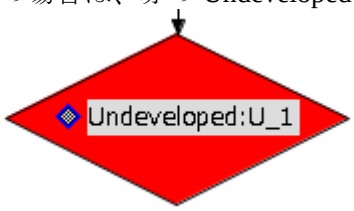
Evidence ノード内に設定した比較パラメータの結果が表示されます。

括弧内の値が実行結果の値です。

ゴールとして設定した「httperf の最大応答時間が 3 秒（3000 ミリ秒）未満」に対し、実行結果が「588.8 ミリ秒」であるため、条件を満たす結果となりました。

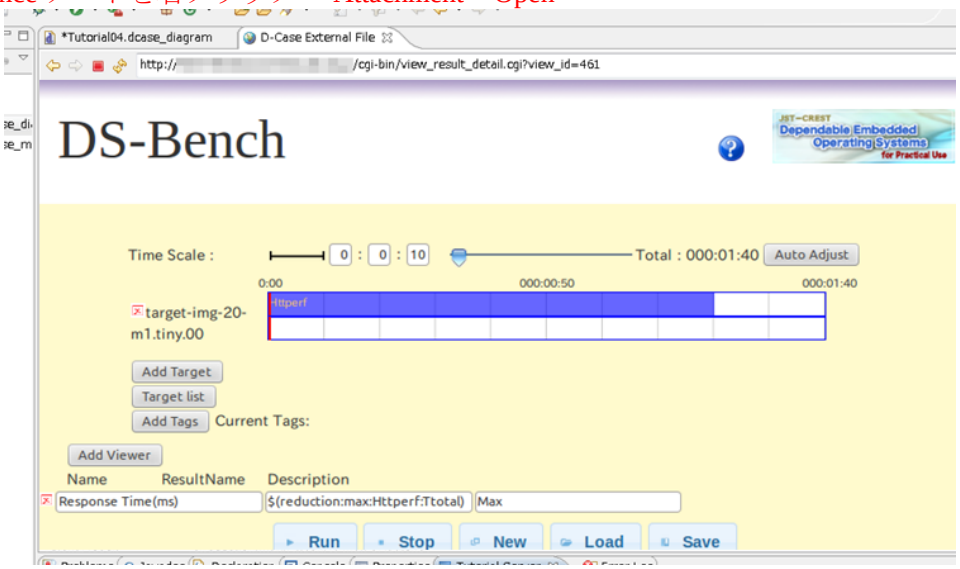


※評価が期待値に対し未達成の場合は、赤の Undeveloped ノードが追加されます。



※D-Case Editor から DS-Bench のシナリオ実行結果詳細画面を表示することができます。

Evidence ノードを右クリック ＞ Attachment ＞ Open



D-Case Editor と連携を行い、D-Case 作成時にベンチマークの測定結果を Evidence にする必要が生じた場合、DS-Bench を用いてその結果を D-Case Editor へ渡し検証を行い、検証結果が期待値を満たした場合はそのシナリオ内容（今回の場合は Web サーバにアクセス数が毎分 1500 回以下であれば、レスポンスタイムは 3 秒以内であること）を保証する方法は以上です。

DEOS プロジェクト