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



# **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、	DEOS R&D Center	DEOS R&D Center	
		タイトル変更、3.4の内容変更			

# 目次

1.	はじ	こめに	.4
2.	動作	環境	5
/	2.1.	物理マシン単体を使用した構成	5
/	2.2.	仮想マシン単体を使用した構成	6
,	2.3.	物理マシン・仮想マシンを複数台組み合わせた構成	7
	2.4.	<b>D-Case Editor</b> を使用し、連携を行う場合の構成	8
3.	実行	〒手順の詳細	9
	3.1.	物理マシン単体でベンチマークを実行するケース	9
	3.1.	1. DS-Bench/D-Cloud のインストール	9
	3.1.2	2. 物理マシンを D-Cloud の管理下へ追加	9
	3.1.	3. ベンチマークシナリオの作成1	2
	3.1.4	4. ベンチマークシナリオの実行1	6
	3.1.	5. ベンチマークシナリオの結果確認1	7
	3.2.	仮想マシン単体でベンチマークを実行するケース1	9
	3.2.	1. 仮想 DS-Bench ターゲットの作成・登録1	9
	3.2.2	2. ベンチマークシナリオの作成2	0
	3.2.	3. ベンチマークシナリオの実行2	.3
	3.2.4	4. ベンチマークシナリオの結果確認2	.5
	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2	.7
	3.3.	1. 事前準備2	.8
	3.3.2	2. ベンチマークシナリオの作成	0
	3.3.	3. ベンチマークシナリオの実行3	5
	3.3.4	4. ベンチマークシナリオの結果確認3	6
•	3.4.	D-CaseEditor と連携を行うケース	8
	3.4.	1. シナリオの新規作成	9
	3.4.2	2. D-Case ダイアグラムの新規作成4	2
	3.4.	3. DS-Bench 連携のための設定4	4
	3.4.4	4. ベンチマークシナリオのインポート4	6
	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)4	.7
	3.4.	6. ベンチマークシナリオの実行4	.9
	3.4.	7. ベンチマークシナリオの結果確認、評価5	0

# 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  $\neg \gamma \land \neg \neg$ 

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  $\ddagger$  thi 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。

<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'</pre>
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					
		сри			
arch	x86_64				
vendor	Intel				
model	core2duo				
mhz	2400				
cores	2	]			
threads	2				
	memo	ory(DDR3)			
speed(mhz)	800				
	de	evices			
	device	(ethernet)			
eth0	1g	Intel	e1000		
	device	e(storage)			
/dev/hda	sata2	WesternDigital	Cavier Green	1T	
	•	os			
family	Linux				
vendor	ubuntu foundation	]			
name	Ubuntu				
version	10.04.3				
arch	x86_64				
	sof	twares			
	softwa	re(kernel)			
version	2.6.32-34-server				
	softwa	re(library)			
name	libc				
version	glibc2.6.??				
	softwar	e(package)			
family	RPM				
	softwar	e(compiler)			
lang	С				
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。

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	Туре	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 Scena	rio」	リンクをクリ	ックし	てください。	
	Synopsis	€	View Result	: €		
	▶ <u>New Benchmark Scena</u>	ario				
	► Load Benchmark Scen	ario	-			
	Benchmark Scenario Name			Descript		
	Concession in the second se		and the second second			
	and the second se					

新規ウィンドウでシナリオ作成画面が開きます。

Add Targ	et」ボタンを	ミクリックし、ベンラ	チマーク対象のマ	シンを追加	します。
	Time S	cale : 🛛 🖂 💿 :	2:0	Tota	
		0:00	000	0:10:00	
	Add Tar Target Add Tag	<del>وet</del> الغز ه Current Tags:			
	Add Viewer Name Re	sultName Description			
ダイアログ	が表示され	るので、「 <u>3.1.2 物理</u>	里マシンを D-Clou	ud の管理下~	<u>、追加</u> 」で追加したマ
シンを選択し	ます。ここ	では「 <mark>trg-tutorial</mark> 」を	選択します。		
trg-tutorial ?	を選択後、	「Done」ボタンをク	リックします。		
		Add Target	×		
		Host ·			

trg-tutorial Done Cancel

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。

Time Scale :		•	- Total : 000:20:00 Auto	Adjust
⊠trg-tutorial	0:00	000:10:00		000:20:00
Add Target Target list Add Tags Curren	t Tags:			
Add Viewer Name ResultName	Description			
	► Run - Stop	New Sea Load	Save	

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。

#### DEOS プロジェクト

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。

Details	×					
Target : trg-tutorial Program : Bonnie → Bonnie++(IO benchmark program) ■Anomaly load						
Begin time:         0         H         1         M         22         S           End time:         0         H         7         M         58         S						
Interval before starting(ms): 50 Interval after termination(ms): 1000 number of groups : 2						
user : Done Delete Cance						

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。
	⊠tra-tutorial Bonnie
	Add Target
	Add Tags Current Tags:
	Add Viewer
	Name ResultName Description

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。

Edit Tags
Frequently used tags :
virtual
iperf
Bonnie
single
used tags :
tutorial,physical
it.
Register Cancel

Current Tags に設定したタグが表示されます。

```
DEOS プロジェクト
```

Time Scale :		_
⊠trg-tutorial	0:00 000	):1
Add Target Target list Add Tags Current	t Tags: tutorial,physical	
Add Viewer Name ResultName	Description       Run     Stop	

「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「<u>3.4. D-Case Editor と連携を行うケース</u>」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。

Description				
► Run	• Stop	New	🖙 Load	Save

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。

Save Benchmark Scenario							
Benchmark	Scenario Name :						
Tutorial01							
Description							
physical tar	get						

※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。

#### Benchmark scenario Management

No	Benchmark scenario name	Description	
1	Tutorial01	physical target	Delete
2	100 C	and the second sec	Delete

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

「<u>3.1.3. ベンチマークシナリオの作成</u>」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。

Synopsis	2	View Result	-		C
New Benchmark Scenar	rio				
Load Benchmark Scena	<u>rio</u>				
Benchmark Scenario Name			Des	scripti	ion
Tutorial01	phy	sical target			

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。

「<u>3.1.3. ベンチマークシナリオの作成</u>」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。

Load Benchmark Scenario					
Benchmark	Scenario Name :				
Tutorial01	<b>•</b>				
Description	:				
physical ta	pet				
	Open Delete Cancel				

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。
	Target list Add Tags Current Tags: tutorial,physical
	Add Viewer           Name         ResultName         Description
	► Run = Stop 🖙 New 🖙 I
実行中の画面	

#### DS-Bench/Test-Env(D-Cloud)実行手順書

#### DEOS プロジェクト

© 2013 科学技術振興機構



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。

► Run	Stop	🛛 New	🖙 Load	Save
	Reduct	ion ( Bonnie	e / %CP )	
🖲 max 💿 min	🛇 sum 🛇	prod 💿 avg	© stddev	
<b>max</b> 97				
••••••	Reductio	n ( Bonnie	/ Rewrite )	
🖲 max 💿 min	💿 sum 💿	prod 💿 avg	© stddev	
: _				

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						
	max 97演算結果						

max	min	sum	prod	avg	stddev
最大値	最小值	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)やシナリオー覧

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。

	Main     Synopsis     View Result     Configuration       earch Include     in Comment     Search						n
ID	Date	Machines	Comment	Benchmark Program	Anormaly Load	Tags	Delete
198	2012/03/15 10:15:51	trg-tutorial	tutorial01 result	Bonnie		tutorial,physical add	
195	2012/03/15 09:58:29	trg-tutorial	physical target	Hackbench		tutorial01,physical add	
	2012/03/14	trav-shutdown01-9-	coustress				

※ 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
- シナリオー覧画面(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</u>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。

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 l	List
---------	-----------------	----	------------	------

		-			
No	Machine name	Туре	Status		
1	trg-tutorial	ph <b>y</b> sical	Active		
2	target-img-20-m1.tiny.00	virtual	Power off		
3	target-img-20-m1.tiny.01	virtual	Power off		
4	target-img-20-m1.tiny.02	virtual	Power off		
Refresh					

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。

Synopsis	View Result
▶ <u>New Benchmark Scenario</u>	
▶ Load Benchmark Scenario	-
Benchmark Scenario Name	De

新規ウィンドウでシナリオ作成画面が開きます。



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。

#### DS-Bench/Test-Env(D-Cloud)実行手順書

#### DEOS プロジェクト

Add Target 🛛 🗙
Host : target-img-20-m1.tiny.00 ▼ Done Cancel

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。

Time Scale :	► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ►	:
⊠target-img-20-	0:00 000:10:00 000	:20:00
m1.tinv.00 Add Target Target list Add Tags Current	t Tags:	
Add Viewer Name ResultName	Description       Run     Stop     New     Load     Save	

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。

#### DS-Bench/Test-Env(D-Cloud)実行手順書

© 2013 科学技術振興機構

DEOS プロジェクト

Details	×
Target : target-img-20-m1.tiny.00 Program : Memstress 🕥 Memory load tool	
Begin time: 0 H 1 M 6 S End time: 0 H 8 M 11 S	
Interval before starting(ms): 0 Interval after termination(ms): 0	
Memory size : 10240 Try count : 200	
Wait(us) : 1000000 Done Delete Can	cel

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。

Add	Target		
Tare	et list		
Add	Tags Current	Tags:	
Viewer			
ne	ResultName	Description	
		·	

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。

```
DEOS プロジェクト
```

		Edit 1	Fags		
		Frequ	ently used tags	S :	
		scena	IrioA		
		vinua			
		Bonni	•		
		single	e		
		Single			
		used	taas :		
		tutori	al.virtual		
			T		
			<b>_</b>		_
			Register	r Cancel	
о (т	レヨーション	*==*	トナナ		10
Current Tags	に設たしにダクか Starget-img	·衣小さ a-20-	れより。		
	m1.tiny.00	9 20			
	Add Target				
		urrent T	aas: tutorial.vir	tual	
	Name ResultN	Jame	Description		

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。 it Tags: tutorial, virtual



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「TutorialO2」、シナリオ説明を「virtual target」としています。

Save Benchmark Scenario 🛛 🗙
Benchmark Scenario Name :
Tutorial02
Description :
virtual target
Save Cancel

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

「<u>3.2.3. ベンチマークシナリオの作成</u>」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。

Synopsis	→ View Result  → Corrected  → Correc
New Benchmark Sce	nario
Load Benchmark Sce	nario
Benchmark Scenario Name	e Description
Tutorial02	virtual target
ALC: NO	Access and a contract

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。

「<u>3.2.3. ベンチマークシナリオの作成</u>」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。

Load Bench	ımark s	Scenar	io	×
Benchmark \$	Scenario	o Name	• :	
Tutorial02		-		
Description :				
virtual targe				
	Dpen	Delete	Canc	el

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 Tags Current Tags: tutorial, virtual
Add Viewer
Name ResultName Description
► Run - Stop @ New @ L
実行中は以下の画面が表示されます。

#### DS-Bench/Test-Env(D-Cloud)実行手順書

#### DEOS プロジェクト

© 2013 科学技術振興機構



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。

R	eduction (	Memstress	/ terminated	1)
🖲 max 💿 m	n 🔘 sum 🤇	∋prod ⊚avg	💿 stddev	
<b>max</b> 0				-
		⊚Table ⊙L	og	
Memstress	on <b>192.16</b>	8.255.4		-
terminated (	)			

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。

	Reduction	( Memst	ress / te	rminate	演算方法
🔊 max 🔘	min 💿 sum	©prod (	🛛 avg 🔘 s	tddev 🦳	
max 0 —	演算結	果			
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 ○ Log
	表示方法
Memstress on	表示結果
terminated 0 —	327148

- ◆ 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。

s	Dearch International International Provided	lain	Synopsis	S 🗩 🕽	/iew Resu	lt ⊇ C	onfiguration	
	ID	Date	Machines	Comment	Benchmark Program⊑	Anormaly Load	Tags∎	Delete
	266	2012/03/16 10:21:43	target-img-20- m1.tin <b>y.</b> 00	tutorial02 result	Memstress		tutorial,virtual	
	261	2012/03/16 10:03:40	target-img-20- m1.tiny.00	virtual target	cpustress		tutorial,virtual add	

- ※ 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
  - シナリオー覧画面 (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<u>3.1</u>.と<u>3.2</u>.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<u>2.3 物理マシン・仮想マシンを複数台組み合わせた構成</u>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.ml.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-clouding-amd64.img
-rw-rw-rw- 1 root root 4110432 YYYY-MM-DD hh:mi lucid-server-clouding-amd64-vmlinuz-virtual
-rw-r-r-- 1 root root 1476395008 YYYY-MM-DD hh:mi target-img
$
```

「<u>3.2.1. 仮想 DS-Bench ターゲットの作成・登録</u>」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。

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
\$				

回体化 「target-1mg-03」	を空球しより。以下は全球仮の唯語	ふ じ 9 。	
\$glance index ID Name	Disk Format	Container Format	Siz

同様に「target-img-03」を登録します。以下は登録後の確認です。

Π е 22 target-img03 1476395008 ami ami 21 target-img02 1476395008 ami ami 20 target-img ami ami 1476395008 org-kernel 4108960 1 aki aki \$

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
\$ 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

raiget list							
No	Machine name	Туре	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				

and dict

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。 物理マシンヘログインし、いくつかプロセスが起動していることを確認します。

DEOS プロジェクト

※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	apach	е			
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。

\$ Is -1 /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。

Ado	d Target			×
Hos tra-	st : tutorial		•	
)		Done	Ca	ncel

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。

Time Scale :		st
⊠trg-tutorial	0:00 000:10:00 00	0:20:00
Add Target Target list Add Tags Curren	nt Tags:	
Add Viewer Name ResultName	Description	
	► Run = Stop @ New @ Load   Save	

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。

E/R (X, Y) / A (F/K) 国 (A (X) + V) (A (C (X) / A))	追加後、	シナリ	オ作成画面は以下の内容となります	す。
--	------	-----	------------------	----

Time Scale :		•	— Total : 000:20:00 🛛	Auto Adjust
	0:00	000:10:00		000:20:00
⊠trg-tutorial				
Xtarget-				
img02-21-				
⊠tatiget00				
img03-22-	0:00	000:10:00		000:20:00
Add Target				
Target list	t Taga			
	u lays.			
Add Viewer DogultNamo	Description			
indille Resulundille	Description			
	► Run - Stop	🕫 New 🖙 Lo	ad 🛛 Save	

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。

⊠target- img02-21-	
	↓ I
⊠target- img02-21-	

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「<u>3.3.1 事前準備</u>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。

Details
Target : target-img02-21-m1.tiny.00 Program : <mark>Httperf →</mark> Httperf
Anomaly load
End time: 0 H 2 M 0 S
Interval before starting(ms): 15
Interval after termination(ms): 37
Total Number of Connections
Connection Rate : in Rate {rate0};2)
Connection :
URI : /index.html
Port number : 80
Done Delete Cancel

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<u>3.4.1</u>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。

```
DEOS プロジェクト
```

Details	×
Target : trg- <u>tutorial</u> Program : Hackbench → Hackbench(Load test) ☑Anomaly load	
Begin time: 0 H 0 M 20 S	
End time: 0 H 1 M 40 S	
Interval before starting(ms): 15	
Interval after termination(ms): 37	
number of groups : 480	
Done Delete Cancel	]

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。



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。

DEOS プロジェクト

Edit Tags
Frequently used tags : scenarioA virtual iperf Bonnie single
used tags : tutorial,physical,virtual, Register Cancel

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。

Save Benchmark Scenario
Benchmark Scenario Name :
Tutorial03
Description :
physical and virtual target
Save Cancel

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

「<u>3.3.2. ベンチマークシナリオの作成</u>」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。

Synopsis	View Result
New Benchmark Scenar	rio
Load Benchmark Scena	<u>rio</u>
Benchmark Scenario Name	De
Tutorial03	physical and virtual target

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。

Load Benchr	nark Scenario	×
Benchmark Se	cenario Name :	
Tutorial03	•	
Description :		
physical and	virtual target	
O	ben Delete	Cancel

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。

	Reduction ( Httperf / Ttotal )
⊚max ⊚min	⊙sum ⊙prod ⊙avg ⊙stddev
	Reduction ( Httperf / Tstart )
⊙max ⊙min	⊙sum ⊙prod ⊙avg ⊙stddev
	⊙Table ⊛Bar ⊙Line ⊙Log
– Httperf on	0.000
	Httperf benchmark
	350

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。

Reduction(Httperf / Ttotal) ● max ◎ min ◎ sum ◎ prod ◎ avg ◎ stddev						
max 599.9 演算結果						
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。

	∋ M	lain		Synopsis		View Result Configuration		
S	earch 🖪	arch Include			Search			
	ID	Date	Machines	Comment	Benchmark Program∎	Anormaly Load	Tags	Delete
	459	2012/03/23 10:59:44	trg-tutorial target- img02-21- m1.tiny.00 target- img03-22- m1.tiny.00	tutorial03 result	Httperf	Hackbench	tutorial,physical,virtual	
	453	5.02*		12112	-		add	

- ※ 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
  - シナリオー覧画面(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<u>http://www.dependable-os.net/osddeos/concept.html</u>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。

#### シナリオの新規作成 3.4.1.

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。

Synopsis	€	View Result	Co
▶ <u>New Benchmark Scer</u>	<u>nario</u>	]	
Load Benchmark Scent Load B	nario	-	
Benchmark Scenario Name		D	escription

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。
	Time Scale : 🛛 🗕 : 🔁 : 💿 🗧	
	0:00	
	Add Target Target list Add Tags Current Tags:	
	Add Viewer Name ResultName Description	

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。

Add Target	×
Host : target-img-20-m1.tiny.	00 -
Done	Cancel

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。

Time Scale :		0:2:0			— Total : 000	:20:00 🛛	Auto Adjust
	0:00		000	):10:00			000:20:00
⊠target-img-20-							
M1.tinv.00 Add Target							
Target list Add Tags Curren	t Tags:						
ver							

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。

Details
Target : target-img-20-m1.tiny.00 Program : Httperf 🗧 🗧
Httperf
Anomaly load
Begin time: 0 H 0 M 0 S
End time: 0 H 1 M 20 S
Interval before starting(ms): 15
Interval after termination(ms): 37
Total Number of Connections : nnections 1500)
Connection Rate : ectionParSec:25
Connection server : Webサイト
URI : /index.html
Port number : 80
Done Delete Cancel

※「\$(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]}:[值])

- [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。

Add Tags Currer	nt Tags:	
Add Viewer		
Name ResultName	Description	
Response Time(ms)	\$(reduction:max:Httperf:Total)	max of latency

- 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。

#### DS-Bench/Test-Env(D-Cloud)実行手順書

© 2013 科学技術振興機構

DEOS 7	プロジ	ェクト
--------	-----	-----

Save Benchmark Scenario	×
Benchmark Scenario Name :	
Tutorial04	
Description :	
Httperfによるベンチマーク。1	
Save Cance	

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。
```

<b>e</b> 1	Java - Eclipse											
File	Edit Source	Refactor	Navigate	Search	Project	Run	D-C	ase	Window	Help		_
	New				Alt-	+ Shift-	+N ►	鬯	Java Proj	ject		> ₹
	Open File								Project			_
	Close					Ctrl+	W	₿	Package			
	Close All				Ctrl+	-Shift+	W	C	Class			
	Save					Ctrl	+S	C	Interface	: 		
	Save As							<b>1</b>	Source F	older		
R	Save All				Ctrl	+ Shift	+S	G	Enum			
	Revert							ev Lå	Annotatio	on Juine Clet		
	Move							189 1100	Java wor	t Caro		
F4	Rename						F2		Junic Test Case			
8	Refresh						F5		Untitled	Text File		
	Convert Line De	elimiters To	D				►		Folder			
÷	Print					Ctrl	+P		File			
	Switch Workspa	ice					•	1	Example			
	Restart							C)	Other		Ctrl+N	
2	Import											
4	Export											

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



New	
Select a wizard Creates D-Case diagram.	
Wizards:	
type filter text	
<ul> <li>▷ ➢ General</li> <li>▷ ➢ CVS</li> <li>▷ ➢ D-Case Editor</li> <li>◎ Ď D-Case Diagram</li> <li>▷ ➢ Git</li> <li>▷ ➢ Java</li> <li>▷ ➢ Maven</li> <li>▷ ➢ Tasks</li> <li>▷ ➢ WindowBuilder</li> </ul>	E
Calibria     Calibria	ancel

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。

New D-Case Diagram	- • ×
Create D-Case Diagram Select file that will contain diagram model.	
Enter or select the parent folder:	
D-CaseTemplate	
CoseTemplate	
File name: Tutorial04.dcase_diagram	
Advanced >>	
	Cancel

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。

#### DEOS プロジェクト

D-Case ダイアグラムが新規作成されました。

🧶 Java - D-CaseTemplate/Tutorial04.dcase_diagram - Edipse	
File Edit Diagram Navigate Search Project Run D-Case Window Help	
<b>『 * 11 * 11 11 12 12 * 10 * 10 * 10 * 10 </b>	😭 🎒 Java
Tahoma ▼ 9 ▼ B I   A ▼ み ▼ J ▼ → ▼   階   凝 ▼ 昭 ▼ 録 ▼   凹   戸 区 日 ▼   100% ▼	
📔 Package E 🛞 🧮 🗖 🔃 Tutorial04.dcase_diagram 🗵	- 0
	· 24 🔅 Palette 👂
😢 D-CaseTemplate	^ <u></u> •
	(⇒ GSN Nodes ∞
	Goal
	OEvidence
	Strategy
	Context
	(ChrustiScation
	⊖ D-Case Nodes ⇔
 	Monitor
	System
	Policy
	Userderuul
	🗁 Links 🗠
	K Supported By
	K In Context Of
	Link
	, Link
🛞 Problems 🕼 Javadoc 🔞 Declaration 🥅 tammieŭ 🖨 Console 🗟 🔪 🗔 Properties 🌒 Error Log	
No consoles to display at this time.	
	🛯 🔍 A 般 🐸 🥔 🕐 🖏 🗧

### 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。
```

e_diagram - Edipse		
Project Run D-Case	Window Help	
<b>Q</b> ▼ <b># ©</b> ▼ <b>∅</b> <i>I</i>   A ▼ <b>∛</b> ▼ <i>.J</i> ▼	New Window New Editor	⇔ •   ≠
04.dcase_diagram 🛛	Open Perspective	•
2	Show View	• • 12
	Customize Perspective	
	Save Perspective As	
	Reset Perspective	
	Close Perspective	
	Close All Perspectives	
	Navigation	•
	Preferences	

「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。

#### © 2013 科学技術振興機構

Preferences		
type filter text	DS-Bench Servers	⇔ ▼ ⇔ ▼ ▼
General Ant D-Case Diagram Appearance Bookmarks Connections Converters DS-Bench Servers Parameters Printing	Server Name:	Tutorial Server
Rulers And Grid Help Install/Update Java Maven Model Validation Mylyn Run/Debug Team Usage Data Collector Validation WindowBuilder XML	All Scenario URL: Detail Scenario URL: Execute URL: Monitor and Search URL: Cancel URL: Scenario view URL: Result view URL: Add Delete Restore	/cgi-bin/get_scenario_list.cgi         /cgi-bin/get_scenario_detail.cgi         /cgi-bin/run_scenario.cgi         /cgi-bin/get_state_result.cgi         /cgi-bin/stop_scenario.cgi         /cgi-bin/view_scenario.cgi         /cgi-bin/view_result_detail.cgi         /cgi-bin/view_result_detail.cgi         Pefaults       Apply
?	ок	Cancel

#### 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。

default.dcase\_diagram 🛿



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 > [<u>3.4.2.</u>で設定した Server Name]」を選 択します。

#### DS-Bench/Test-Env(D-Cloud)実行手順書

DEOS フロシ	/エクト
----------	------

© 2013 科学技術振興機構

Strategy:5,4 ペンチマークに{	Add Note <u>File</u> <u>Edit</u> Delete from Diagram Delete from Model	>	G_68 数を監視できる	
-	Format Add Child Convert Node Type Attachment Show/Hide Children Set Sibling Order Parameters	> > > > >	onitor:M_11 arの入りロで測る	
	DS-Bench	>	Select Test Scenario	Tutorial Server
utorial Server 🔲 Pr	Show Properties View Properties Remove from Context	Shift+Ctrl+Alt+Down		>
Property	Input Methods	> Value	-	

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 <u>3.3.2.</u> で作成した「Tutorial04」を選択します。					
🔄 Console 🗖 Tutorial Server 🕱 🖾 Properties	Import Open scenario 🔻 🗖 🗖				
Scenario ID Scenario Nam Description					
id_000000004 Tutorial04 Httperfによるペンチマーク。1秒間に{ConnectionParSec}回Webサーバにアクセスし、計{TotalConnections}回に					

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:G\_103 Goal:G\_68 Httperfによるベンチマーク。1秒 アクセス数を監視できる 間に25回Webサーバにアクセ スし、計1 行します。 Add Note Time(m テストしま <u>F</u>ile > m Monitor:M\_11 Serverの入り口で測る <u>E</u>dit > Delete from Diagram Delete from Model For<u>m</u>at > Add Child > Convert Node Type > A<u>t</u>tachment > Show/Hide Children > rial Server Parameters > Redmine > D-Case DB perty > Set Parameters. Desc IW Show Properties View Desc Forma るベンチマーク。1秒間に{Conn Properties Message Shift+Ctrl+Alt+Down Remove from Context Name Input Methods >:ctions=1500,ConnectionPa Parameters

「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。

•		Set Parameters	×
Httperfによるハ	ペンチマーク。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({Response Time(ms)}[ms])を満たすかテ.	ストします
TotalConnections	5 : 1500		
ConnectionParSe	c: 25		
Response Time(r	ns) : <3000		
		Cancel	ж
	9		
	Httperfによるペンラ	-マーク。1秒間に{ConnectionPa	
	TotalConnections :	1500	
	ConnectionParSec :	25	
	Response Time(ms)	: <3000	

「Goal ノードを右クリック > Show Properties View」を選択します。 10 Goal:G\_103 Goal:G\_68 Httperfによるペンチマーク。1秒 間に25回Webサーバにアクセ スし、計1500回に達するまで実 行します。期待値({Response Time(ms)}[ms])を、 テストします Add I \_ アクセス数を監視 12 Add Note or:1 >(9 <u>F</u>ile <u>E</u>dit > 14 Delete from Diagram Delete from Model 19 For<u>m</u>at > Add Child > < Convert Node Type > 🖳 Console 📋 Tutorial Server 🔲 Prope A<u>t</u>tachment > Show/Hide Children > Goal Parameters > Property Core Redmine > D-Case DB > Desc -7 DS-Bench > Desc Format String Message Name Shift+Ctrl+Alt+Down 50 Parameters Remove from Context > ~ Project Input Methods Requirement

設定したパラメータを確認します。

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。

-		_			
Goal:G	103	t i	Goal:G_68		
Httperfに。 間に25回W	るベンチマーク。1秒 lebサーバにアクセ	1	アクセス数を監視できる		
スし、計1 行します。	Add Note				
Time(m テストしま	<u>F</u> ile	>	Monitor:M_11		
<u> </u>	<u>E</u> dit	>	Serverの入り口で測る		
	Delete from Diagram				
	Delete from Model				
	For <u>m</u> at	>			
	Add Child	>			
	Convert Node Type	>			
	Attachment	>			
	Show/Hide Children	>			
rial Server	<u>P</u> arameters	>			
	Red <u>m</u> ine	>			
perty	D-Case DB	>			
Attachment		>	Set Parameters		
Desc	Show Properties View		Execute W		
Desc Forma	Properties		るベンチマーク。1秒間に{Conn		
Message	Romovo from Contoxt Ch	ift i Ctrl i Alt i Down			
Name	Remove from Context Sh	III+CUI+AII+DOWN			
Parameters	Input Methods	>	ctions=1500,ConnectionPa		

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。

is default.dease_diagrafit ω		L						
· · · · · · · · · · · · · · · · · · · ·		2 · · · · · · 14 · · · · · · 16 · · · · · 18 ·						
16 •••••• 14 ••••••• 12 •••••• 8•••••	■ Goal:G_103 Httperficよるペンチマーク。1秒間に 25回Webサーバにアクセスし、計 1500回に達するまで実行します。期 待値{{Response Time{ms}} [ms])を満たすかテストします run 100 %	● Goal:G_68 アクセス数を整 ● Monitor Web Serverの入 ● Monitor ● Policy ● Dicy ● Context ● Dolicy ● Policy ● Dicy ● Dicy ● Userdef001 ● Links ■ Supported I ▼ In Context () ● Links						
		) Link						
Console      Tutorial Server      Properties     Properties     Console								
2013/04/10 13:56:19 net.dependableos.dcase.diagram Benchmark status(3) progress(100) Goal(G_103) 2013/04/10 13:56:19 net.dependableos.dcase.diagram *** update status process end. *** 2013/04/10 13:56:24 net.dependableos.dcase.diagram *** update status process start. *** 2013/04/10 13:56:24 net.dependableos.dcase.diagram Benchmark status(3) progress(100) Goal(G_103) 2013/04/10 13:56:24 net.dependableos.dcase.diagram *** update status process end ***								

# 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 <sup>■</sup> *Tutoriel04.dcese_diagram <sup>●</sup> ◆ ● ● http://	O -Case External File ☆     /cgi-bin/view_result_det	Open tail.cgi?view_id=461		
R.m DS-Ben	DS-Bench			-CREST pendable Embedded Operating Systems for Protoal Use
Time Scale : Time Scale : Target-img-i m1.tiny.00 Add Target Target list Add Target List Add Target C Target Viewer Name ResultNat Response Time(ms)	0:0:10 0:00 0:0- 0:0- 0:0- 0:0- 0:0- 0:0		tal : 000:01:40 Auto	o Adjust 000-01:40

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