

JST-CREST

研究領域

「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」

DEOS プロジェクト



DS-Bench/Test-Env(D-Cloud) 仕様書

Version 1.00

2013/05/01

DEOS 研究開発センター

目次

1.	はじめに	5
1.1.	機能概要	5
1.1.1.	DS-Bench/D-Cloud.....	5
1.1.2.	D-Case Editor/D-Case Weaver との連携.....	6
1.2.	システム構成	6
1.2.1.	DS-Bench コントローラ	6
1.2.2.	DS-Bench ターゲット	7
1.2.3.	D-Cloud コントローラ	7
2.	実現方式	8
2.1.	システム概要	8
2.1.1.	DS-Bench.....	8
2.1.2.	D-Cloud	9
2.1.3.	D-Case Editor 連携	10
2.1.4.	コマンドラインからのシナリオ実行.....	13
2.2.	パッケージ化	14
2.2.1.	DS-Bench コントローラおよび DS-Bench ターゲット	14
2.2.2.	D-Cloud コントローラ	15
2.2.3.	Sedna DB サーバ.....	15
3.	メモリ仕様	16
4.	ファイル仕様	17
4.1.	各ベンチマーク conf ファイル	17
4.2.	Sedna XML DB に格納する XML 形式	20
4.2.1.	ベンチマークシナリオ XML.....	20
4.2.2.	ベンチマーク結果 XML.....	23
4.2.3.	アノマリー結果 XML.....	26
4.3.	run_status_*	29
5.	通信仕様	30
5.1.	通信電文	33
5.1.1.	ベンチマークインストール要求	33
5.1.2.	ベンチマークインストール応答	33
5.1.3.	ベンチマークアンインストール要求.....	33
5.1.4.	ベンチマークアンインストール応答.....	34
5.1.5.	計算資源リスト要求	34
5.1.6.	計算資源リスト応答	35
5.1.7.	計算資源確保要求	35
5.1.8.	計算資源確保処理中	36

5.1.9.	計算資源準備完了	36
5.1.10.	計算資源準備不能	37
5.1.11.	ベンチマークシナリオ送信	37
5.1.12.	ベンチマークシナリオ受信完了	38
5.1.13.	ベンチマークシナリオ拒否応答	38
5.1.14.	ベンチマーク開始要求	39
5.1.15.	ベンチマーク開始応答(ACK)	39
5.1.16.	ベンチマーク開始応答(NAK)	40
5.1.17.	ベンチマーク取得状態要求	40
5.1.18.	ベンチマーク取得状態応答	41
5.1.19.	フォルトリスト要求	41
5.1.20.	デーモン生存確認要求	42
5.1.21.	デーモン生存確認応答	42
5.1.22.	フォルトリスト応答(ACK)	42
5.1.23.	フォルトリスト応答(NAK)	43
5.1.24.	ベンチマーク開始完了通知	43
5.1.25.	ベンチマーク実行完了通知	43
5.1.26.	計算資源解放要求	44
5.1.27.	計算資源解放応答(ACK)	44
5.1.28.	計算資源解放応答(NAK)	45
5.2.	要求 XML	46
5.2.1.	計算資源リスト取得要求 XML	46
5.2.2.	計算資源確保要求 XML	46
5.2.3.	ベンチマークシナリオ XML	53
5.2.4.	ベンチマーク開始要求 XML	55
5.2.5.	デーモン生存確認要求 XML	55
5.2.6.	計算資源解放要求 XML	56
5.3.	応答 XML	57
5.3.1.	計算資源リスト応答 XML	57
5.3.2.	計算資源確保応答 XML	58
5.3.3.	ベンチマークシナリオ応答 XML	58
5.3.4.	ベンチマーク状態 XML	59
5.3.5.	計算資源確保応答 XML	61
5.3.6.	計算資源確保応答 XML	62
5.4.	D-Case Editor との通信	62
5.4.1.	シナリオ実行開始	62
5.4.2.	シナリオ結果閲覧	62
5.4.3.	シナリオ中止	63

6.	画面仕様	64
6.1.	共通メニュー	64
6.2.	main 画面	64
6.3.	Benchmark 画面	65
6.3.1.	ターゲット追加ダイアログ	66
6.3.2.	ターゲット編集ダイアログ	66
6.3.3.	ターゲットリスト	67
6.3.4.	タイムスケール上での操作法について	68
6.3.5.	ベンチマーク詳細設定ダイアログ	69
6.3.6.	Fault injection の利用	71
6.3.7.	タグの追加	72
6.3.8.	シナリオ実行と結果	72
6.3.9.	D-Case との連携	73
6.4.	Synopsis 画面	76
6.5.	View Result	78
6.5.1.	View Result Detail	79
6.6.	Configuration 画面	80
6.6.1.	Benchmark	81
6.6.2.	Setting	82
6.6.3.	Target list	83
6.6.4.	Sedna	84
6.6.5.	D-Cloud	84
6.6.6.	Benchmark scenario	85

1. はじめに

本稿ではディペンダビリティの計測ツールである DS-Bench、および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 (D-Cloud) を用いベンチマークテストを組み合わせ実行し、テスト結果を評価するシステムの仕様書である。

※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。

1.1. 機能概要

1.1.1. DS-Bench/D-Cloud

本システムは DS-Bench コントローラ 1 台と、DS-Bench ターゲットと呼ばれる複数台の検証用端末と、これらの端末を管理する D-Cloud コントローラ 1 台、およびネットワーク制御可能なネットワークスイッチとネットワーク制御可能な電源タップにより構成され、DS-Bench ターゲットにテストを実行したい環境を構築し、WebUI からテストシナリオを作成する。シナリオではターゲットに対して何時、どのタイミングでどんなベンチマーク/Anomaly load を起動し、どのタイミングで終了させるかを設定し、シナリオ通りに DS-Bench ターゲット上で指定したベンチマーク/Anomaly load を実行する。使用できるターゲットは D-Cloud より確認メッセージを受け取り返却したターゲットを D-Cloud がリストにして DS-Bench コントローラに返す。そのリストにより DS-Bench ターゲットを選択する。1つのシナリオ中では、1つのターゲットに対してベンチマーク/Anomaly load は複数組み合わせ指定でき、またターゲットも複数指定できる。ベンチマークの起動条件は3通りあり、1つは指定時刻になった時、2つ目は指定プログラムが終了した時、3つ目は指定プログラムが起動したときである。他のシナリオで指定して実際に実行しているターゲットに対しては、シナリオでそのターゲットを選択することは出来るが終了するまで実行を行う事は出来ない。

実行したベンチマーク/Anomaly load の結果はターゲットから XML で DS-Bench コントローラに返す。返された結果に対して複数の同じベンチマークから結果を集約・演算し、その結果をデータベースに保存する。

実行中の進行状況、ステータス、実行結果は WebUI より確認することが出来る。

デーモンプロセス (dsb デーモン、Sedna、D-Cloud デーモン) に接続できなかった場合、設定ファイルフォーマットエラー、ベンチマークプログラムのインストール、起動の失敗、入力値チェックなどベンチマーク実行の処理が期待通りに動作しなかった際に、ユーザに対して正しいエラーメッセージを表示する。

1.1.2. D-Case Editor/D-Case Weaver との連携

D-Case Editor から DS-Bench にある既存のシナリオを取得、シナリオのベンチマークに対してパラメータ、期待値の設定、実行を D-Case Editor/D-Case Weaver から行う事が出来、実行した結果、または結果を集約・演算した結果を D-Case Editor/D-Case Weaver に渡し、期待値との比較により評価を行う。

D-Case Editor/D-Case Weaver からベンチマークシナリオの実行状況を閲覧出来る。

D-Case Editor/D-Case Weaver から実行中のシナリオに対して中止を行う事が出来る。

1.2. システム構成

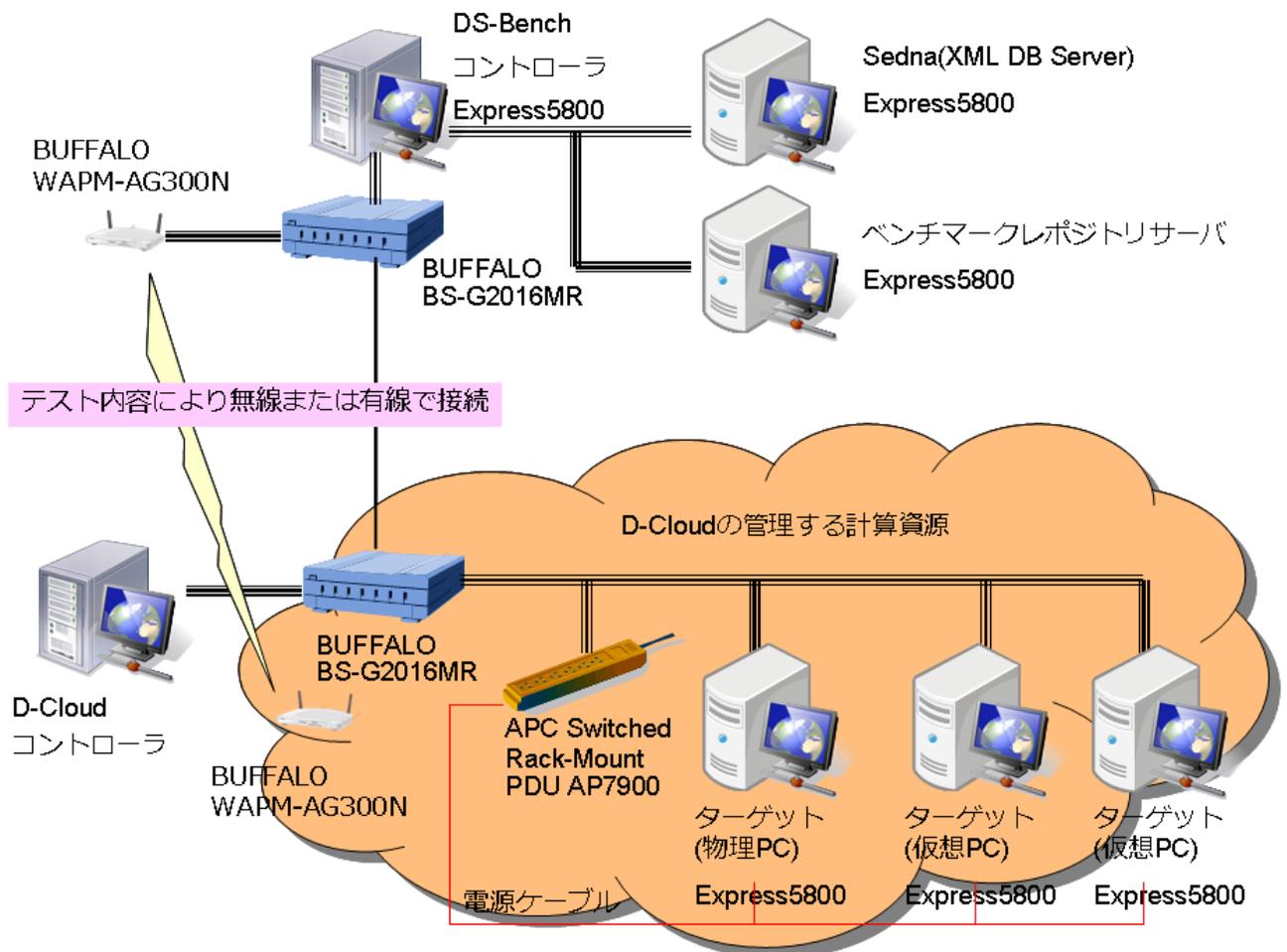


図 1-1 システム構成図

DS-Bench コントローラ 1 台と、ターゲットと呼ばれる複数台の検証用端末（物理 PC、仮想 PC）と、これらの端末を管理する D-Cloud コントローラ 1 台、およびネットワーク制御可能なネットワークスイッチとネットワーク制御可能な電源タップにより構成される。

1.2.1. DS-Bench コントローラ

ベンチマークツール、シナリオデータを持ち、ターゲットに対してシナリオ実行命令を送り、そ

の結果を取得、取得結果に対し演算を行い結果を生成、これらの結果を保存する機能を有す。また、WebUI を持ち、これらの機能を直観的、容易に操作できる。データベースとして Sedna を使用する。

1.2.2. DS-Bench ターゲット

DS-Bench コントローラからの要求を受け、テスト実行、異常生成を行い、その結果を成型して返却する。DS-Bench ターゲット仮想マシンを使用する場合、D-Cloud で OpenStack の設定を行う必要がある。

1.2.3. D-Cloud コントローラ

計算資源 (DS-Bench ターゲット) などの情報を一元管理するために、計算資源の情報取得、予約、起動等を D-Cloud 経由で行い、また Fault Injection の管理と実行を行う。

2. 実現方式

2.1. システム概要

2.1.1. DS-Bench

コントローラ上には Apache 上で動作する CGI プログラム類があり、Ajax 技術を使用した Web 画面を提供する。Web 上のほとんどの処理は CGI により処理され、コントローラ上のファイルまたは Sedna XML DB に保存される。

コントローラ上には dsbd-ctrl というデーモンプログラムが配置されており、D-Cloud に対して計算資源リストの確認を行い、計算資源を確保する。確保したターゲットマシンに対してベンチマークのインストール・アンインストール命令を送る。また、ターゲットマシンからの応答を受け取り、結果をファイルまたは Sedna XML DB に書き込む処理を行う。

全てのターゲット上には dsbd-target という同一のデーモンプログラムが配置されており、D-Cloud からのデーモン生存確認に対して確認応答を送信する。デーモンは TCP ポートでコントローラの dsbd-ctrl から受けた電文を基にしてベンチマークプログラムのインストール・アンインストール・起動・終了の操作を行う。

ベンチマークプログラム実行後はコントローラの dsbd-ctrl に結果を XML 形式で返却する。

2.1.1.1. ベンチマークシナリオ拡張

ベンチマークを実行する為のパラメータに、ユーザがわかりやすく名前を付与する為の機能である。

D-Case Editor と連携する場合に D-Case Editor に渡す結果についても名前を付与することができ、D-Case Editor より指定した名前で結果の問い合わせを行うことが出来る。

設定は DS-Bench の GUI より各パラメータの名前を設定する。

2.1.1.2. リダクション機能

複数の同じベンチマークから結果を集約・演算する機能を指し、演算の種類は MAX(最大値)、MIN(最小値)、SUM(総和)、PROD(総乗)、AVE(平均値)、STDDEV(標準偏差)の演算を行い結果を生成する。

2.1.1.3. ユーザインタフェース

ベンチマークの起動パラメータにターゲットマシン、ネットワークインターフェースを設定する場合、ドロップダウンリストより選択出来る。

各ベンチマークの設定ファイルに必要な属性を指定することで、GUI 表示でその情報を判断してドロップダウンリストの表示を行う。ドロップダウンリストは、計算資源リスト応答の結果より作成を行う。

2.1.2. D-Cloud

D-Cloud 上には ds-ctrl というデーモンプログラムが配置されており、DS-Bench コントローラからの計算資源要求に対して全てのターゲットにデーモン生存確認を送り、その結果を計算資源リスト応答として DS-Bench コントローラへ返す。応答 XML の形式についてはデーモン生存確認 XML に従う。

DS-Bench はベンチマークのアンインストール完了後に、D-Cloud に対して計算資源解放要求を送信する。D-Cloud は計算資源解放の処理を行い、DS-Bench に計算資源解放応答(ACK/NAK)を送信する。

2.1.2.1. ターゲット仮想マシン

仮想マシンの IP アドレスは、仮想マシン起動時に OpenStack が決めるため、事前には分からない。

そのため応答 XML には IP アドレスを持ち、ターゲットマシンとの通信時には IP アドレスを使用する。

2.1.2.2. フォルトリスト機能

DS-Bench はフォルトリスト要求を D-Cloud に対して送信し、D-Cloud はフォルト情報を取得して、フォルトリスト応答(ACK)を DS-Bench に送信する。異常時は、フォルトリスト応答(NAK)を送信する。

応答 XML の中身についてはフォルトリスト XML に従う。

Fault injection はシナリオを設定する際に GUI 上にある各ターゲットマシンのタイムライン上で設定を行う。

2.1.2.3. シナリオ中止機能

シナリオを中止するためには、処理中の計算資源確保を中止する必要がある。D-Cloud への通知シーケンスとしては計算資源解放要求を使い、D-Cloud コントローラ側の挙動を若干変更する。確保処理中の資源に対する解放要求を受信すると、このプロセスは資源を解放せず「確保中止」のフラグを立てる。確保処理の方でフラグを認知すると確保を中断し、必要に応じて資源解放を行う。DS-Bench 側では資源確保失敗とみなされ中断処理が完了する。

D-Case Editor からの中止においても、この方式は水平展開する。

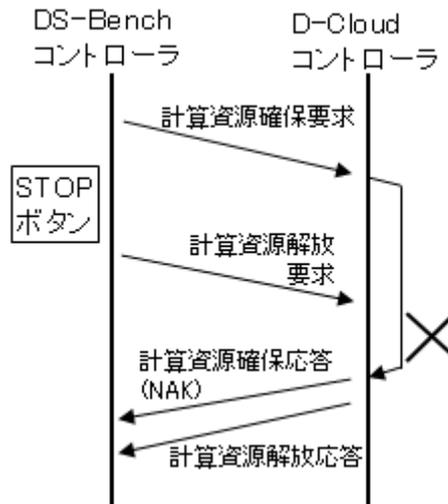


図 2-1 計算資源確保中断処理イメージ

2.1.3. D-Case Editor 連携

2.1.3.1. ベンチマークシナリオ一覧取得

D-Case Editor はベンチマークシナリオ一覧取得のための cgi を実行する。DS-Bench はパラメータより指定されたベンチマークシナリオの一覧を返却する。要求パラメータと応答 XML の形式については、「DS-Bench/D-Case Editor 連携 I/F 仕様書」に従う。

2.1.3.2. ベンチマークシナリオ詳細取得

D-Case Editor はベンチマークシナリオ一覧取得のための cgi を実行する。DS-Bench はパラメータより指定されたベンチマークシナリオの詳細を返却する。要求パラメータと応答パラメータと対応パラメータは「DS-Bench/D-Case Editor 連携 I/F 仕様書」に従う。

2.1.3.3. ベンチマークシナリオ実行機能

D-Case Editor はベンチマーク実行開始要求の為の cgi を実行する。DS-Bench シナリオ編集画面からの実行と同様に、指定されたシナリオの計算資源の確保、ベンチマークインストール、ベンチマーク実行、ベンチマークアンインストール、計算資源の解放までを行う。

ベンチマーク実行開始要求を受付けた後、D-Case Editor に処理を受付けた結果を返却する。

要求パラメータと応答 XML の形式については、「DS-Bench/D-Case Editor 連携 I/F 仕様書」に従う。

2.1.3.4. ベンチマーク実行状況・結果確認

D-Case Editor はベンチマーク実行状況・結果取得要求の為の cgi を実行する。DS-Bench は、指定されたベンチマークシナリオの実行状況を確認し、その結果を返却する。

要求パラメータと応答 XML の形式については、「DS-Bench/D-Case Editor 連携 I/F 仕様書」に従う。

2.1.3.5. 通知ステータス

ベンチマークシナリオの実行状況を D-Case Editor へ通知するためのステータス、またエラー内容を示すコード/メッセージが存在し、このステータスおよびコード/メッセージを用意することで、シナリオの状況をより詳細に伝達できるようにする。

code	message	詳細
0	success	正常終了
1*		Parser 周りのエラー
11	D-Case Editor Error: request XML parser error	リクエスト XML のパースに失敗
12	DS-Bench Error: scenario XML parser error	シナリオ XML のパースに失敗
13	DS-Bench Error: benchmark XML parser error	ベンチマーク XML のパースに失敗
14	DS-Bench Error: fault XML parser error	FaultXML のパースに失敗
15	DS-Bench Error: result XML parser error	結果 XML のパースに失敗
2*		Permission 周りのエラー
21	DS-Bench Error: run_list file permission denied.	run_list ファイルのオープンエラー
22	DS-Bench Error: run_status file permission denied.	run_status ファイルのオープンエラー
23	DS-Bench Error: XML file permission denied.	XML のオープンエラー
3*		排他制御及び計算資源周りのエラー
31		欠番
32	DS-Bench Error: the target is in use.	ターゲットが使用中
4*		通信周りのエラー
41	DS-Bench Error: install failed.	ベンチマークインストール失敗
42	DS-Bench Error: create invalid socket.	ソケット作成失敗
43	DS-Bench Error: list of calculation resources is unacquirable.	計算資源リストの取得失敗
44	DS-Bench Error: failure to reserve calculation resources.	計算資源確保失敗
45	DS-Bench Error: failure to release	計算資源解放失敗

	calculation resource.	
5*		IO エラー
51	D-Case Editor Error: scenario do not exist.	指定されたシナリオが存在しない
52	D-Case Editor Error: parameter do not exist.	指定されたパラメタが存在しない
53		欠番
54	DS-Bench Error: scenario/benchmarks/ directory does not open.	scenario/benchmarks ディレクトリが存在しない
55	DS-Bench Error: packages.list do not exists.	packages.list が存在しない
9*		その他
91	DS-Bench Error: Not be able to create JSON data.	JSON データ作成失敗
92	DS-Bench Error: failure to reduction.	リダクション失敗
93	DS-Bench Error: failure to stop benchmark run.	ベンチマーク実行停止失敗
94	the scenario is not running.	シナリオが実行停止している
95	D-Case Editor Error: Scenario has not executed from D-Case.	指定されたシナリオが D-Case Editor から実行されていない

表 2-2 エラーコード／メッセージ表

status	詳細
error	何らかのエラー
ready	シナリオの正常な受付が完了
reserve	計算資源資源確保中
install	ベンチマークインストール中
run	シナリオ実行中
reduction	結果集計／格納中
finish	シナリオ完了
stopping	シナリオ中止中
stopped	シナリオ中止完了
st_error	シナリオ中止中のエラー

表 2-3 シナリオ実行状況ステータス表

2.1.3.6. DS-Bench 表示 ID との関連付け

D-Case Editor から実行されたベンチマークシナリオは独自の ID を持っており、シナリオ実行要求の際に、D-Case Editor へのレスポンスに GUI 上での結果 ID を追加し、その ID を受け、GUI の結果画面へのアクセスにはシナリオ実行要求のレスポンスに含まれている結果 ID を使用する。通信仕様は [5.4.1 シナリオ実行開始](#) および [5.4.2 シナリオ結果閲覧](#) を参照。画面仕様については [6.5.1 View Scenario Result](#) を参照。

2.1.3.7. ベンチマークシナリオ中止機能

D-Case Editor からベンチマークシナリオを中止するための CGI を提供する。通信仕様はシナリオ中止を参照。

2.1.4. コマンドラインからのシナリオ実行

2.1.4.1. コマンドラインからのシナリオ実行

コマンドラインからベンチマークシナリオを実行する際に、指定されたあるパラメータの値を書き換えられるようする。

書き換え可能なパラメータの書式は以下のとおり

```
$(param:description{name}:value)
```

とする。ここに、description はパラメータ説明文、name はパラメータ名、value はデフォルト値を表す。DS-Bench の GUI でこの書式を記述することにより、コマンドラインからの実行でもパラメータを変更できるようになる。

実行のためのコマンドは dsbcui とし、以下のように実行する。

```
$ /var/dsbench/controller/bin/dsbcui {-r|--run} △シナリオ名 △-name=value △-name=value △...
```

シナリオ中のパラメータ名と、コマンド引数のパラメータ名の不整合については、dsbcui ではチェックを行わない。実行モジュールの仕様に任せる。

このコマンドは基本的にシナリオ実行の受付のみを行う。受付までに実行失敗と見なされたシナリオについてはエラーメッセージを出力する。受け付けられたら結果 ID (GUI で表示される) を表示し、GUI の結果ページを閲覧しやすいようにする。シナリオ結果の閲覧は GUI から、または DB から行う。コマンドの仕様は表 2-4 に定める。

2.1.4.2. コマンドラインからのシナリオ中止

2.1.4.1. コマンドラインからのシナリオ実行に対応して、シナリオの中止もできるようにする。以下のように実行する。

```
$ /var/dsbench/bin/dsbcui {-s|--stop} △シナリオ ID
```

ここにシナリオ ID は、実行時に取得した ID とする。

このコマンドは基本的にシナリオ中止の受付のみを行う。受付までに失敗と見なされたシナリオについてはエラーメッセージを出力する。正常に中止された旨を表示する。

コマンドの仕様は表 2-4 に定める。

返却値	出力	条件
0	Succeeded running scenario in scenario_id=***	シナリオ実行受付が完了(run)
0	Succeeded stopping scenario with scenario_id=***	シナリオ中止受付が完了(stop)
1	(usage:TBD を出力)	引数なし
2	(usage:TBD を出力)	シナリオ名が存在しない(run) シナリオ ID が存在しない(stop)
11~99	実行モジュールに依存したメッセージ	シナリオ受付に失敗
0	(usage:TBD を出力)	-h --help オプション

表 2-4 dsbcui 返却ステータス表

2.2. パッケージ化

パッケージは、D-Cloud コントローラ/DS-Bench コントローラ/DS-Bench ターゲット/Sedna DB サーバに分け、それぞれ Ubuntu 向け deb パッケージとしてリリースする。

2.2.1. DS-Bench コントローラおよび DS-Bench ターゲット

インストーラには以下の設定値の入力を促す処理を追加する。() 内はデフォルト値を表す。

- DS-Bench コントローラデーモンのポート ("9803")
- DS-Bench ターゲットデーモンのポート ("9804")

DS-Bench コントローラでは、加えて以下の入力も促す。

- D-Cloud コントローラノードの IP アドレスまたはホスト名 ("127.0.0.1")
- D-Cloud コントローラデーモンのポート ("9903")
- Sedna DB サーバの IP アドレスまたはホスト名 ("127.0.0.1")
- Sedna で作成した DB 名 ("dsbench")
- Sedna で作成した、ベンチマーク格納用 collection 名 ("result_benchmark")
- Sedna で作成した、シナリオ格納用 collection 名 ("result_benchmark_set")

これらの入力値を受けて、インストール時に設定ファイルを書き換える。

2.2.2. D-Cloud コントローラ

D-Cloud コントローラは、Openstack を使うタイプと使わないタイプに分けて 2 通りにパッケージ化する。Openstack を使わないタイプでインストールした場合、Openstack をインストールする必要はないが、仮想マシンを扱えなくなる。

衝突パッケージ (Conflicts) として、互いのパッケージを指定する。

依存パッケージ (Depends) として、特に DS-Bench ターゲットを追加する。

インストール中に入力を促す設定値は以下のとおり。() 内はデフォルト値を表す。

- DS-Bench ターゲットデーモンのポート ("9804")
- D-Cloud コントローラデーモンのポート ("9903")
- D-Cloud 資源マネージャデーモンのポート ("9904")
- マップサーバデーモンのポート ("7000")

Openstack を使うタイプでは、加えて以下の入力も促す。

- novarc のパス ("/var/dcloud/openstack/creds/novarc")

D-Cloud コントローラ、D-Cloud 資源マネージャ、マップサーバの 3 つのデーモンについては、インストール直後に起動し、すぐ使えるようにする。またサーバ起動時にも自動起動するようにする。

2.2.3. Sedna DB サーバ

Sedna については、<http://www.sedna.org/>よりインストーラシェルスクリプトが提供されているが、これをラップする形でパッケージ化するものとする。また、Sedna インストール後、以下の設定値を入力するよう促す。() 内はデフォルト値を表す。

- DS-Bench で使用する DB 名 ("dsbench")
- DS-Bench で使用する、ベンチマーク格納用 collection 名 ("result_benchmark")
- DS-Bench で使用する、シナリオ格納用 collection 名 ("result_benchmark_set")

これらの入力値を受けてインストーラ内で Sedna に DB および collection を作成する。パッケージインストール完了後、Sedna と上記作成 DB を起動してすぐ使えるようにする。

また次回起動時から Sedna および DS-Bench で使用する DB を自動起動する処理を行う。

3. メモリ仕様

モジュール間インタフェースで共有メモリ等は使用しない。

4. ファイル仕様

4.1. 各ベンチマーク conf ファイル

ベンチマーク毎に用意するベンチマーク固有の設定を記述するファイル。

conf/ フォルダ下に配置する。

管理画面の **Benchmark Management** から編集することが可能。

設定ファイルの詳細については以下の通り。

```
<dsbxml>
  <benchmark seq=" path=" title=" type=">
    <description/>
    <client_input host=">
      <stime interval="/>
      <ftime interval="/>
      <option input="netif" label=" arg=" ui=" interactive="/>
    </client_input>
    <server_input host=">
      <stime interval="/>
      <ftime interval="/>
      <option input="target" label=" arg=" ui=" interactive="/>
    </server_input>
    <output>
      <caption/>
      <table>
        <header/>
        <rheader/>
        <data/>
        <begin/>
        <end/>
        <key/>
        <value/>
      </table>
    </output>
    <graph>
      <title/>
      <ylabel/>
      <imagesize/>
      <xrange/>
    </graph>
  </benchmark>
</dsbxml>
```

ベンチマークの指定について、ノードは省略してもよいが **Tag** は必要事項を記入する。

空白 “ ” でもよいものはその旨を併記。

<dsbxml> </dsbxml>

ルートノード

繰り返し：不可、省略：不可

```
<benchmark seq="[num]" title="[str]" path="[path]" type="[unit|link]"></benchmark>
```

この Node でベンチマークの指定を行う。

- "seq"には、DS-Bench コントローラ側からみた実行順序が記入されている。
- "title"には、便宜上の呼称が記入されている。空白可
- "path"には、ベンチマークへのフルパスが記入されている。
- "type"には、単体ベンチマーク[unit]またはC/S型ベンチマーク[link]が記入されている。

繰り返し：可、省略：不可

```
<server_input host="[IP]"></server>
```

C/S 型ベンチマークのサーバ側についての設定を記述する。

上位の"benchmark"ノードが持つ"type"が"link"の場合にのみ意味を持つ。

省略：可 (type=link の場合、省略不可) 、繰り返し：不可

"host"対象となる Server マシンのホスト名または IP アドレスを指定する。

```
<client_input host="[IP]"></client>
```

このノード下で単体、C/S 型ベンチのクライアント側についての設定を記述する。

省略：可、繰り返し：不可

"host"対象となる Client マシンのホスト名又は IP アドレスを指定する。

```
<stime interval="">開始時刻</stime>
```

ベンチマークシナリオの実行開始を 0 とした、ベンチマークの相対的な実行開始時刻を、秒単位で設定する。

パラメータの interval に開始待ち遅延時間をミリ秒単位で指定する。

```
<ftime interval="">終了時刻</ftime>
```

ベンチマークシナリオの実行開始を 0 とした、ベンチマークの相対的な実行終了時刻を、秒単位で設定する。

パラメータの interval に終了待ち遅延時間をミリ秒単位で指定する。

```
<option input="[text|netif|target]" label="[str]" arg="[-c |--c|=|-c]" ui="[yes|no]"
```

```
interactive="[yes|no]">[str]</option>
```

ベンチマークに引数などがある場合指定する。

繰り返し：可、省略：可

- "input"
 - text：テキストボックスによる値の入力可能
 - netif：ドロップダウンリストにて、ネットワークインターフェースを選択可能
 - target：ドロップダウンリストにて、ターゲットマシン選択可能

(※ドロップダウンリストは D-Cloud から取得した計算資源より選択可能なリスト作成)

- “label”オプションの名称を記述。
- “arg” ハイフン付の引数を指定、ハイフンがつかないものはノードの値に設定する。
- “ui” Web 実行時にコンフィグ画面を表示するかを記述
- “interactive” コマンドライン実行時に対話的に入力する必要がある場合” yes”を記入。

<output/>

結果の出力形式について指定する。

省略した場合、結果を返却しない。

省略：可、繰り返し：不可

<caption/>

結果の表題、文字列か正規表現で指定する。

省略：可、繰り返し：不可

<table/>

結果出力についての指定をこのノード下で行う。

取り出したい表の個数だけこのノードを記述する。

省略：不可、繰り返し：可

<header>[str]</header>

抜き出す表内の値の名称部を指定する

省略：可

<rheader>[str]</rheader>

抜き出す値の名称をカンマ(,)区切りで設定する。

省略：可

<data>[str]…</data>

出力結果の取り出す値を()で括って表記する。

※スクリプト内で正規表現を用いて()で括った部分を抜き出しているため、(())等しないでください。

省略：不可、繰り返し：不可

<begin>[str]</begin>

抜き出す表の先頭部分を指定、ここで指定した文字列は結果検索に含まれる。

省略：可

<end>[str]</end>

抜き出す表の終了部分を指定、ここで指定した文字列は結果検索に含まれる。

省略：可

`<value>[0|1]*</value>`

値の名称をマスクする場合 0 を設定する。

省略した場合、自動的に 1 が入れられたものとする。

省略：可

`<key>[0|1]*</key>`

値をマスクする場合 0 を設定する。

省略した場合、自動的に 1 が入れられたものとする。

省略：可

`<graph>`グラフ描画方式`</graph>`

縦軸横軸の抽出値、軸のスケール等をベンチマークシナリオのパラメータとして保持し、プログラムではそのパラメータから使用するラベルや値等を決定する。

追加するパラメータは次の通り。

パラメータ	内容
title	グラフの見出し
xlabel	X 軸のラベル
ylabel	Y 軸のラベル
image size	画像サイズ
xrange	グラフのスケール
xdata	X 軸値
ydata	Y 軸値

4.2. Sedna XML DB に格納する XML 形式

ベンチマークおよびアノマリーに関連して、ベンチマークシナリオ XML、ベンチマーク結果 XML、およびアノマリー結果 XML の 3 種類の情報を XML 形式で保存している。以下はその詳細である。

4.2.1. ベンチマークシナリオ XML

ベンチマークシナリオ XML は、ベンチマークを実行する都度一つ Sedna XML DB に格納する。ベンチマークシナリオ XML に含まれている主な情報は以下になる。

- ID
- ベンチマークの行われた日付
- ターゲットホスト名、ハードウェア環境、ソフトウェア環境
- コメント
- ベンチマーク結果 XML またはアノマリー結果 XML にアクセスするための ID

● ベンチマークシナリオ XML の仕様

```

<dsbxml>
  <id>ID</id>
  <display-id>result 画面表示用番号</display-id>
  <data>ベンチマーク開始日時</data>
  <display-time>result 画面表示用日時</display-time>
  <controller>
    <host>コントローラホスト名</host>
  </controller>
  <target>
    <host>コントローラホスト名<env><cpu>x86_64</cpu><memory type='DDR3'><speed
type='mhz'>800</speed><size>2048</size>……</env></host>
    .
    .
  </target>
  <benchmark-list>
    <model id="ベンチマーク結果 XML の ID">ベンチマーク名</model>
    .
    .
  </benchmark-list>
  <anomaly-list>
    <model id="アノマリー結果 XML の ID">アノマリープログラム名</model>
    .
    .
  </anomaly-list>
  <result-viewer-list>
    <result-viewer>
      <result-name>ベンチマーク結果を定義する名前</result-name>
      <result-trgt>ベンチマーク結果名</result-trgt>
      <result-desc>ベンチマーク結果に対するコメント</result-desc>
    </result-viewer>
    <result-viewer>
      .
      .
    </result-viewer>
  </result-viewer-list>
  <reduction>
    <benchmark name="ベンチマーク名">
      <label name="結果のキー名">
        <max target="ターゲット名">最大値</max>
        <min target="ターゲット名">最小値</min>
        <sum>総和</sum>
        .
        .
      </label>
      .
      .
    </benchmark>
    .
    .
  </reduction>
  <description>ベンチマークセットに対するコメント</description>
</dsbxml>

```

各タグの意味と設定内容は以下の通り

`<id>ID</id>`

(数字の桁数は適当なので実際ものに合わせてください)

`result_id_一意となる番号`

`<display-id>result 画面表示番号</display-id>`

View Result 画面の一番左の欄に表示する ID 番号を設定する。Sedna XML DB に最初に登録したデータの番号を 0 番とし、以後、登録する際に 1 ずつ増加させた番号を設定する。

実際は、Sedna XML DB を検索（読み出し）して一番大きな番号を求め、その値に+1 した値を使用している。

`<data>ベンチマーク開始時間</data>`

ベンチマークの実行を開始した日時を以下の文字列で設定する。

YYYY/MM/DD hh:mm:ss

`<host>コントローラホスト名</host>`

コントローラのホスト名を設定する。コントローラを未使用の場合は空文字とする。

`<host>ターゲットホスト名<env>~</env></host>`

ターゲットのホスト名を設定する。ターゲットが複数ある場合、ターゲットの数だけ連続して記述する。

`<env>~</env>`の間にそのターゲットの環境情報を記述する。

環境情報は、計算資源リスト取得応答 XML の`<target>`タグ配下のものをそのまま格納する。

`<model id="ベンチマーク結果 XML の ID">ベンチマーク名</model>`

ベンチマーク結果 XML にアクセスするための ID と、そのベンチマーク名を設定する。

ベンチマーク結果が複数ある場合（複数のベンチマークを実行した場合）は、ベンチマークの数だけ連続して記述する。ID の形式は事項を参照。なお、ベンチマーク結果が存在しない場合は、`benchmark-list` タグだけ記述し、`model` タグは記述しない。

`<model id="アノマリー結果 XML の ID">アノマリープログラム名</model>`

アノマリー結果 XML にアクセスするための ID と、そのアノマニープログラム名を設定する。アノマニー結果が複数ある場合（複数のアノマニーを実行した場合）は、ベンチマークの数だけ連続して記述する。なお、アノマニー結果が存在しない場合は、`benchmark-list` タグだけ記述し、`model` タグは記述しない。

`<result-name>ベンチマーク結果を定義する名前</result-name>`

D-Case Editor からベンチマーク結果にアクセスするための識別子。シナリオ作成者が設定する。

<result-trgt>ベンチマーク結果名</result-trgt>

<result-name>からアクセスされるベンチマーク結果の名前。この名前はベンチマーク設定ファイルに記述してあるもので、最終的にベンチマーク XML に記載される。無効な名前、すなわちシナリオ中のどのベンチマークにも設定されていない結果名が指定されていた場合、対応する<result-name>は無効となる。

<result-desc>ベンチマーク結果に対するコメント</result-desc>

対応する<result-trgt>の説明文

<reduction><benchmark name="ベンチマーク名"><label name="結果のキー名"><max target="ターゲット名">最大値</max><min>・・・</benchmark></reduction>

リダクション結果を保存する。

<description>ベンチマークシナリオに対するコメント</description>

利用者が書き込んだコメントを設定する。

4.2.2. ベンチマーク結果 XML

ベンチマーク結果 XML は、実行したベンチマークシナリオに含まれるベンチマークに対応して Sedna XML DB に格納する。ベンチマーク結果 XML に含まれる主な情報は以下になる。

- ・ ID
- ・ ベンチマーク名
- ・ ターゲット名
- ・ 開始時間 (開始条件)
- ・ 終了時間
- ・ ベンチマーク結果

● ベンチマーク結果 XML の仕様

```

<dsbxml>
  <id>ID</id>
  <benchmark>ベンチマーク名</benchmark>
  <target-list>
    <target>ターゲットホスト名</target>
    .
    .
    .
  </target-list>
  <config>
    <stime interval="">開始時刻</stime>
    <sstart>開始待ちベンチマーク ID</sstart>
    <sstop>終了待ちベンチマーク ID</sstop>
    <etime interval="">終了時刻</etime>
    <anomaly-load> </anomaly-load>
    <client_option>
      <option label="ラベル名" arg="引数フラグ" ui="UI フラグ" interactive="no">オプション値</option>
      .
      .
      .
    </client_option>
    <server_option>
      <server_input host ="サーバプログラム実行ホスト名">
        <stime>0</stime>
        <ftime>サーバ用プログラム終了時刻</ftime>
        <option label="ラベル名" arg="引数フラグ" ui="UI フラグ" interactive="on">オプション値</option>
        .
        .
        .
      </server_input>
    </server_option>
  </config >
  <results>
    <data key="データのキー" header="データのヘッダ">実行結果値</data>
    .
    .
    .
  </results>
  <plain>
    ベンチマークの生出力
  </plain>
</dsbxml>

```

各タグの意味と設定方法は以下の通り。

<id>ID</id>

対応するベンチマークシナリオ XML の ID の後ろに数字の連番を付加したものを設定する。連番はアノマリー結果と共通である。例えば、ベンチマークシナリオ XML の ID が result_id_0000000001 で、ベンチマークシナリオ中のベンチマークが 2 つの場合、ベンチマーク結果 XML の ID の値はそ

れぞれ以下のようになる。

result_id_000000001_1

result_id_000000001_2

<benchmark>ベンチマーク名</benchmark>

ベンチマーク名を指定する。

<target>ターゲットホスト名</target>

ターゲットのホスト名を設定する。ターゲットが複数ある場合は、ターゲットの数だけ連続して記述する。

<stime interval="">開始時刻</stime>

ベンチマークシナリオの実行開始を 0 とした、ベンチマークの相対的な実行開始時刻を、秒単位で設定する。

パラメータの interval に開始待ち遅延時間をミリ秒単位で指定する。

<sstart>開始待ちベンチマーク ID</sstart>

開始待ちベンチマーク ID を ID にもつベンチマークが起動した際に本ベンチマークが起動する設定。ID が-1 の場合は無効。

<sstop>終了待ちベンチマーク ID</sstop>

終了待ちベンチマーク ID を ID にもつベンチマークが終了した際に本ベンチマークが起動する設定。ID が-1 の場合は無効。

<etime interval="">終了時刻</etime>

ベンチマークシナリオの実行開始を 0 とした、ベンチマークの相対的な実行終了時刻を、秒単位で設定する。

パラメータの interval に終了待ち遅延時間をミリ秒単位で指定する。

<client_option>

クライアント用プログラムのオプションを、オプションの数だけ記述する。記述内容はベンチマークの種類によって異なる。なお、サーバ用プログラムの場合は記述しない。

<server_option>

サーバ用プログラムのオプションを、オプションの数だけ記述する。記述内容はベンチマークの種類によって異なる。なお、クライアント用プログラムの場合は記述しない。

<results>

ベンチマークの実行結果を記述する。記述内容は、ベンチマークによって異なる。結果が複数ある場合は、結果の数だけ記述する。

<plain>

ベンチマークの出力する情報を無加工で記述する。ベンチマークの種類によっては何もない場合がある。

4.2.3. アノマリー結果 XML

アノマリー結果 XML は、実行したベンチマークシナリオに含まれるアノマリーに対応して Sedna XML DB に格納する。アノマリー結果 XML に含まれる主な情報は以下になる。

- ・ ID
- ・ アノマリープログラム名
- ・ ターゲットホスト名
- ・ 開始時間（開始条件）
- ・ 終了時間
- ・ アノマリーの識別
- ・ アノマリー結果

- アノマリー結果 XML の仕様

Sedna XML DB に格納するアノマリー結果 XML の形式を次図に示す。D-Cloud から提供される Fault injection の結果は、この形式で保存される。

```

<dsbxml>
  <id>ID</id>
  <benchmark>アノマニープログラム名</benchmark>
  <target-list>
    <target>ターゲットホスト名</target>
  </target-list>
  <config>
    <stime interval="">開始時刻</stime>
    <sstart>開始待ちベンチマーク ID</sstart>
    <sstop>終了待ちベンチマーク ID</sstop>
    <etime interval="">終了時刻</etime>
    <anomaly-load>true</anomaly-load>
    <client_option>
      <option label="ラベル名" arg="引数フラグ" ui="UI フラグ" interactive="no">オプション値</option>
      .
      .
      .
    </client_option>
    </server_option>
    </server_option>
  </config >
  <results>
    <data key="データのキー" header="データのヘッダ">実行結果値</data>
    .
    .
    .
  </results>
  <plain>
    ベンチマークの生出力
  </plain>
</dsbxml>

```

各タグの意味と設定内容は以下の通り。

<id>ID</id>

対応するベンチマークシナリオ XML の ID の後ろに数字の連番を付加したものを設定する。連番はベンチマーク結果と共通である。例えば、ベンチマークシナリオ XML の ID が result_id_0000000001 で、ベンチマークシナリオ中のベンチマークが 2 つの場合、ベンチマーク結果 XML の ID の値はそれぞれ以下のようになる。

result_id_0000000001_1

result_id_0000000001_2

<benchmark>アノマリープログラム名</benchmark>

ベンチマーク名を指定する。

<target>ターゲットホスト名</target>

ターゲットのホスト名を設定する。ターゲットが複数ある場合は、ターゲットの数だけ連続して

記述する。

`<stime interval="">開始時刻</stime>`

ベンチマークシナリオの実行開始を 0 とした、アノマリーの相対的な実行開始時刻を、秒単位で設定する。

パラメータの `interval` に開始待ち遅延時間をミリ秒単位で指定する。

`<sstart>開始待ちベンチマーク ID</sstart>`

開始待ちベンチマーク ID を ID にもつベンチマークが起動した際に本ベンチマークが起動する設定。ID が-1 の場合は無効。

`<sstop>終了待ちベンチマーク ID</sstop>`

終了待ちベンチマーク ID を ID にもつベンチマークが終了した際に本ベンチマークが起動する設定。ID が-1 の場合は無効。

`<etime interval="">終了時刻</etime>`

ベンチマークシナリオの実行開始を 0 とした、ベンチマークの相対的な実行終了時刻を、秒単位で設定する。

パラメータの `interval` に終了待ち遅延時間をミリ秒単位で指定する。

`<anomaly-load>アノマリープログラム判定</anomaly-load>`

アノマリーの場合は `true` を設定し、そうでない場合は `false` を設定する。

`<client_option>`

クライアント用プログラムのオプションを、オプションの数だけ記述する。記述内容はベンチマークの種類によって異なる。なお、サーバ用プログラムの場合は記述しない。

`</server_option>`

サーバ用プログラムのオプションを、オプションの数だけ記述する。記述内容はベンチマークの種類によって異なる。なお、クライアント用プログラムの場合は記述しない。

`</results>`

アノマリーの実行結果を記述する。記述内容は、アノマリーによって異なる。結果が複数ある場合は、結果の数だけ記述する。

`<plain>`

アノマリーの出力する情報を無加工で記述する。アノマリーの種類によっては何も無い場合がある。

4.3. run_status_*

/var/dsbench/controller/running 配下にあり、シナリオの状態のサマリを表すファイル。

ファイルの書式は以下の通り

- STATUS[LF]
- CODE[LF]
- MESSAGE[LF]
- FROMWHERE

FROMWHERE はどのモジュールからシナリオが実行されたかを表す。GUI (run_benchmark.cgi) から実行されたら”gui”、D-Case Editor またはコマンドライン (run_scenario.cgi) から実行されたら”external”の値を持つ。

5. 通信仕様

ここでは DS-Bench と D-Cloud コントローラ間の通信電文について記述する。

DS-Bench コントローラと D-Cloud コントローラ、及び DS-Bench ターゲットとの通信シーケンスは以下の通り。

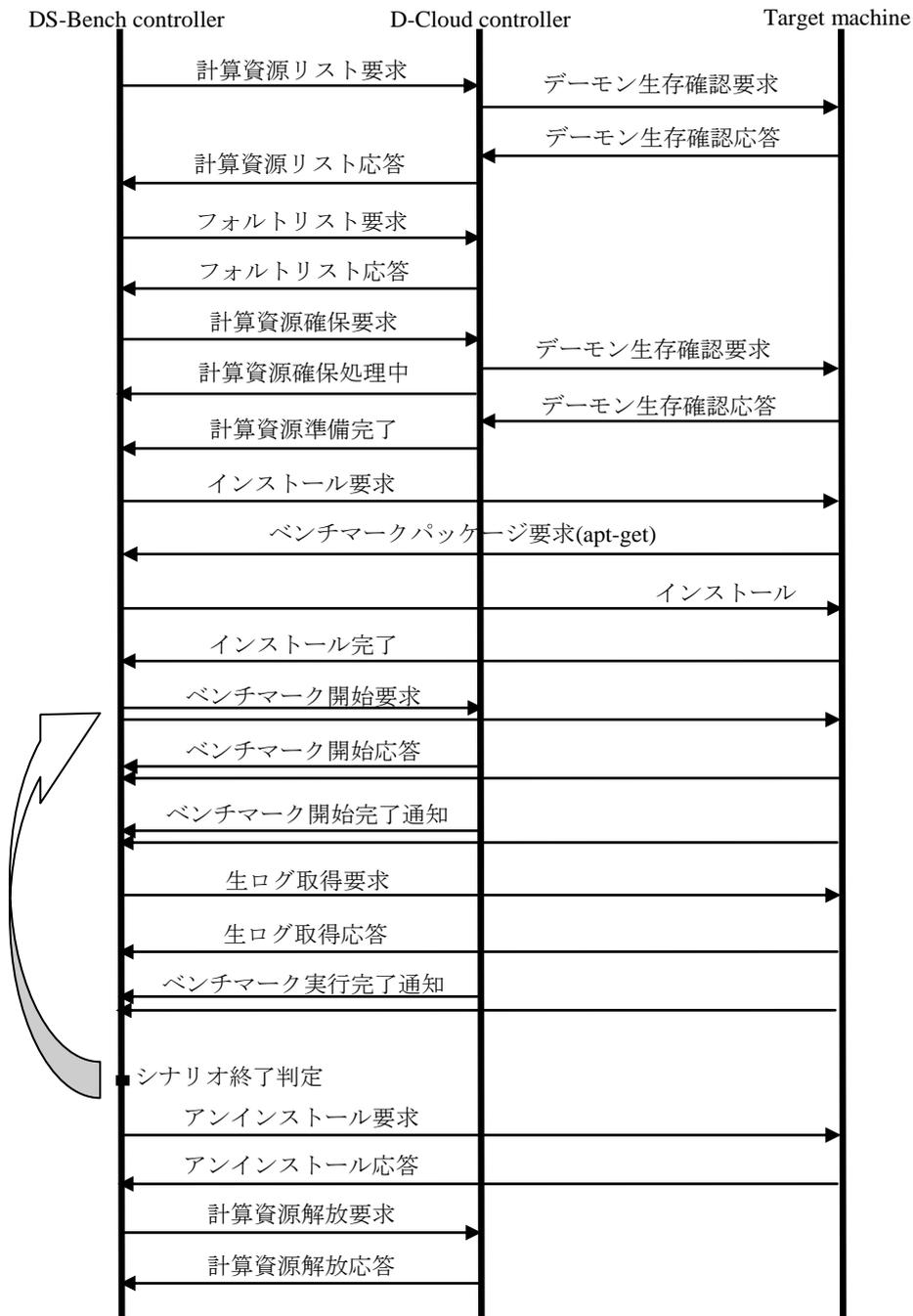


図 5-1 通信シーケンス

DS-Bench と D-Case Editor の通信シーケンス及びインタフェースについては DS-Bench/D-Case Editor 連携 I/F 仕様書を参照。

以下に、複数ターゲットマシンにまたがる実行同期機能について記述する。ベンチマークを実行する為のタイミングは以下の3通りである。

- ・ 指定された時間による実行
- ・ あるプログラムの実行開始を確認後の実行
- ・ あるプログラムの終了を確認後の実行

以下のシーケンス図は、指定された時間が来た場合にベンチマークが起動される例である。

DS-Bench コントローラにて指定時間まで待ち、各ターゲットマシンに対してベンチマーク開始要求を送信する。

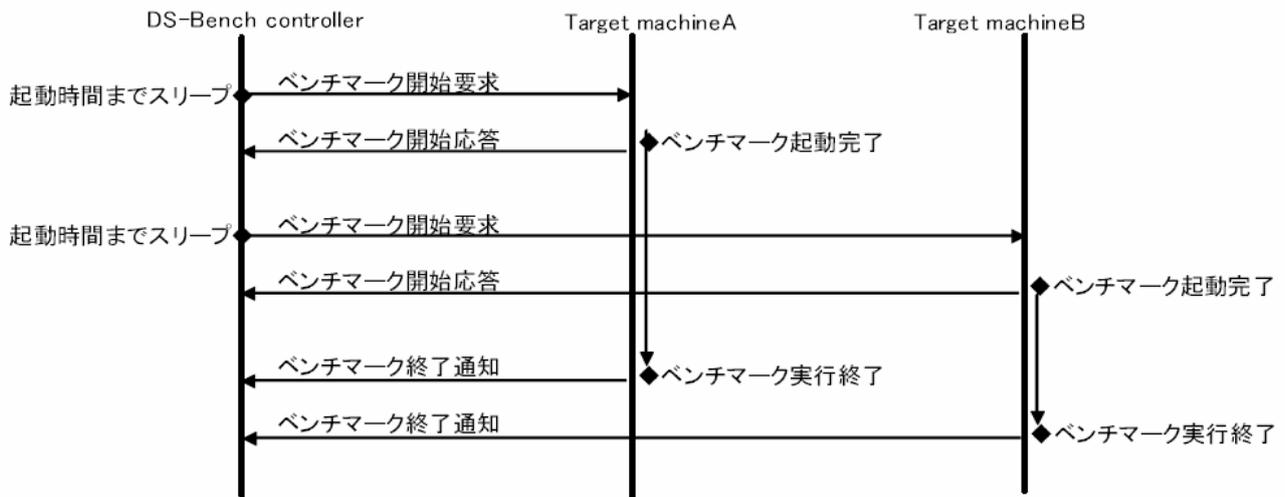


図 5-2 通信シーケンス(指定時間による実行)

以下のシーケンス図は、あるプログラムの実行開始を確認後にベンチマークが起動される例である。

DS-Bench コントローラにてターゲットマシンから送信されたベンチマーク開始応答を受信後に、シナリオから起動終了同期を行うベンチマークがあるかを判断し、別のターゲットに対してベンチマークを実行する。

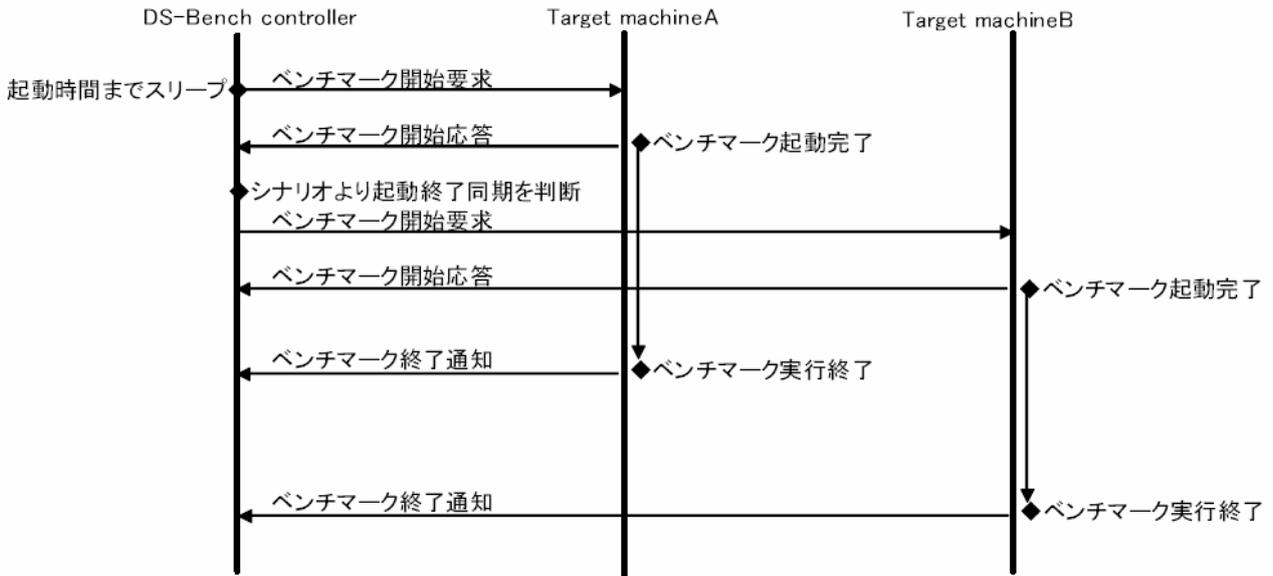


図 5-3 通信シーケンス(あるプログラムの実行開始を確認後の実行)

以下のシーケンス図は、あるプログラムの実行終了を確認後にベンチマークが起動される例である。
 DS-Bench コントローラにてターゲットマシンから送信されたベンチマーク終了通知を受信後に、シナリオから起動終了同期を行うベンチマークがあるかを判断し、別のターゲットに対してベンチマーク開始を要求する。

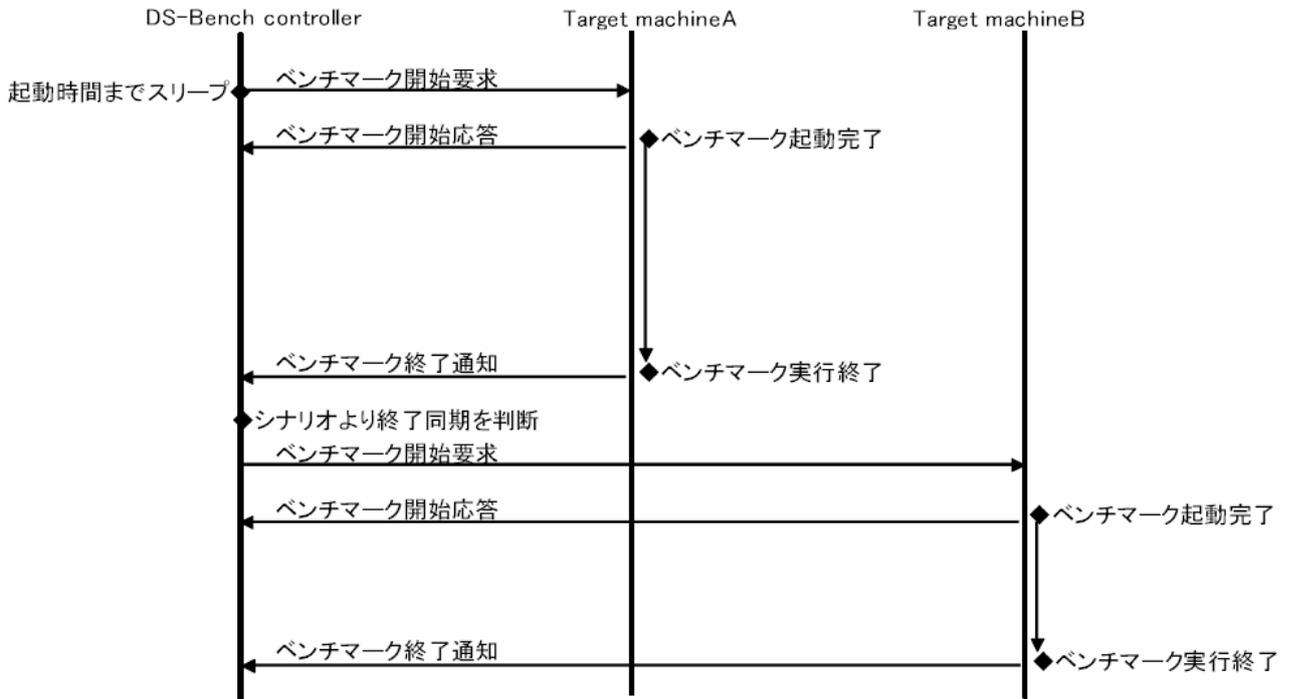


図 5-4 通信シーケンス(あるプログラムの終了を確認後の実行)

5.1. 通信電文

DS-Bench コントローラ、ターゲット、D-Cloud コントローラ間の通信電文について以下に記述する。

5.1.1. ベンチマークインストール要求

Header	ID	ID	固定値
	CMD	'I'(0x49)(1byte)	ベンチマークインストール要求
	SUB_CMD	'?'(1byte)	サーバ側識別子
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	ベンチマーク文字列 (セミコロン区切り)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ

5.1.2. ベンチマークインストール応答

Header	ID	ID	固定値
	CMD	'I'(0x49)(1byte)	ベンチマークインストール要求
	SUB_CMD	'?' (1byte)	サーバ側識別子
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	ベンチマーク ID
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ

5.1.3. ベンチマークアンインストール要求

Header	ID	ID	固定値
	CMD	'U'(0x56)(1byte)	ベンチマークアンインストール要求
	SUB_CMD	'?' (1byte)	サーバ側識別子
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 r)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	ベンチマーク文字列 (セミコロン区切り)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁 character)
	ANS	nothing	応答データ

5.1.4. ベンチマークアンインストール応答

Header	ID	ID	固定値
	CMD	'U'(0x56)(1byte)	ベンチマークアンインストール要求
	SUB_CMD	'?' (1byte)	サーバ側識別子
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 r)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	ベンチマーク ID
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ

5.1.5. 計算資源リスト要求

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'L'(0x4C)(1byte)	計算資源リスト要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	‘ ‘(0x20)(1byte)	未使用
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	計算資源リスト取得要求 XML
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ

5.1.6. 計算資源リスト応答

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'L'(0x4C)(1byte)	計算資源リスト要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'A'(0x41) / 'N'(0x4E)(1byte)	ACK('A')またはNAK('N')を応答
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	計算資源リスト取得要求 XML(要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答 XML データ

5.1.7. 計算資源確保要求

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'R'(0x52)(1byte)	計算資源確保要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	' '(0x20)(1byte)	未使用
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	計算資源リスト取得要求 XML
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ

5.1.8. 計算資源確保処理中

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'R'(0x52)(1byte)	計算資源確保要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'A '(0x41)(1byte)	ACK('A')
	UUID	(36bytes)	シーケンスを識別するための識別子 計算資源確保要求で受け取った UUID を設定
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	計算資源確保要求 XML(要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ(可能な限りの状況報告 XML)

5.1.9. 計算資源準備完了

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'R'(0x52)(1byte)	計算資源確保要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'A '(0x41)(1byte)	ACK('A')
	UUID	(36bytes)	シーケンスを識別するための識別子 計算資源確保要求で受け取った UUID を設定
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	計算資源確保要求 XML(要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	variable length	応答データ(正常応答用 XML)

5.1.10. 計算資源準備不能

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'R'(0x52)(1byte)	計算資源確保要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'N'(0x4E)(1byte)	NAK('N')
	UUID	(36bytes)	シーケンスを識別するための識別子 計算資源確保要求で受け取った UUID を設定
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	計算資源確保要求 XML(要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	variable length	応答データ(可能な限りの状況報告 XML)

5.1.11. ベンチマークシナリオ送信

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'S'(0x53)(1byte)	ベンチマークシナリオ
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	' '(0x20)(1byte)	未使用
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	シナリオ XML
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ

5.1.12. ベンチマークシナリオ受信完了

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'S'(0x4C)(1byte)	ベンチマークシナリオ
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'A '(0x41) / 'N '(0x4E)(1byte)	未使用
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	シナリオ XML(要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ(正常応答用 XML)

5.1.13. ベンチマークシナリオ拒否応答

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'S'(0x45)(1byte)	ベンチマークシナリオ
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	' '(0x20) (1byte)	未使用
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	シナリオ XML(要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ

5.1.14. ベンチマーク開始要求

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'E'(0x45)(1byte)	ベンチマーク開始
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	‘ ‘(0x20) (1byte)	未使用
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	シナリオ XML(要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	nothing	応答データ

5.1.15. ベンチマーク開始応答(ACK)

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'E'(0x4C)(1byte)	ベンチマーク開始
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	‘A ‘(0x41) (1byte)	ACK(‘A ‘)
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	シナリオ XML(要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	variable length	応答データ(正常応答用 XML)

5.1.16. ベンチマーク開始応答(NAK)

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'E'(0x4C)(1byte)	ベンチマーク開始
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'N '(0x4E) (1byte)	NAK('N')
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	シナリオ XML (要求時の XML)
	LEN_ANS	'00000000'	ANS のバイト数(16 進 8 桁)
	ANS	variable length	応答データ(可能な限りの状況報告 XML)

5.1.17. ベンチマーク取得状態要求

Header	ID	'U'(0x55)(1byte)	固定値
	CMD	'5'(0x34)(1byte)	ベンチマーク取得要求
	SUB_CMD	'C'(0x43)(1byte)	ベンチマーク識別子
	RES_ID_LEN	'????????'(8bytes)	シナリオ ID のバイト数(16 進 8 桁)
	RES_ID	variable length	シナリオ ID
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	シナリオ XML (要求時の XML)
	LEN_ANS	'????????'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	variable length	応答データ(可能な限りの状況報告 XML)

5.1.18. ベンチマーク取得状態応答

Header	ID	'U'(0x55)(1byte)	固定値
	CMD	'6'(0x34)(1byte)	ベンチマーク取得要求
	SUB_CMD	'C'(0x43)(1byte)	ベンチマーク識別子
	RES_ID_LEN	'????????'(8bytes)	シナリオ ID のバイト数(16 進 8 桁)
	RES_ID	variable length	シナリオ ID
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁 character)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁 character)
	REQ	variable length	シナリオ XML(要求時の XML)
	LEN_ANS	'????????'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	variable length	応答データ(可能な限りの状況報告 XML)

5.1.19. フォルトリスト要求

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'F'(0x46)(1byte)	ベンチマーク取得要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	' '(0x20) (1byte)	未使用
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhh の RFC4122 ストリング形式準拠
	LEN_ALL	'00000000'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'00000000'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	nothing	未使用
	LEN_ANS	'00000000'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	nothing	未使用

5.1.20. デーモン生存確認要求

Header	ID	T'(0x54)(1byte)	固定値
	CMD	L'(0x4C)(1byte)	計算資源リスト要求
	SUB_CMD	C'(0x43)(1byte)	D-Cloud コントローラ識別子
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	計算資源リスト要求 XML
	LEN_ANS	'????????'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	nothing	未使用

5.1.21. デーモン生存確認応答

Header	ID	T'(0x54)(1byte)	固定値
	CMD	L'(0x4C)(1byte)	計算資源リスト要求
	SUB_CMD	C'(0x43)(1byte)	D-Cloud コントローラ識別子
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	デーモン生存確認要求 XML (要求時の XML)
	LEN_ANS	'????????'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	variable length	デーモン生存確認応答 XML

5.1.22. フォルトリスト応答(ACK)

Header	ID	D'(0x44)(1byte)	固定値
	CMD	F'(0x46)(1byte)	ベンチマーク取得要求
	SUB_CMD	C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'A'(0x41) (1byte)	ACK('A')
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'00000000'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	nothing	未使用
	LEN_ANS	'????????'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	variable length	フォルトリスト XML

5.1.23. フォルトリスト応答(NAK)

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'F'(0x46)(1byte)	ベンチマーク取得要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'N'(0x4E) (1byte)	ACK('N')
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhh の RFC4122 スtring形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'00000000'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	nothing	未使用
	LEN_ANS	'00000000'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	nothing	未使用

5.1.24. ベンチマーク開始完了通知

Header	ID	'U'(0x55)(1byte)	固定値
	CMD	'7'(0x33)(1byte)	ベンチマーク開始完了通知
	SUB_CMD	'C'(0x43)(1byte)	ベンチマーク識別子
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	要求 XML データ
	LEN_ANS	'????????'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	variable length	応答 XML データ

5.1.25. ベンチマーク実行完了通知

Header	ID	'U'(0x55)(1byte)	固定値
	CMD	'8'(0x33)(1byte)	ベンチマーク実行完了通知
	SUB_CMD	'C'(0x43)(1byte)	ベンチマーク識別子
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	要求 XML データ
	LEN_ANS	'????????'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	variable length	応答 XML データ

5.1.26. 計算資源解放要求

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'F'(0x46)(1byte)	計算資源解放要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	‘^’(0x20) (1byte)	未使用
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhh の RFC4122 ストリング形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	計算資源解放 XML
	LEN_ANS	'00000000'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	nothing	未使用

5.1.27. 計算資源解放応答(ACK)

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'T'(0x54)(1byte)	計算資源解放要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	‘A’(0x41) (1byte)	ACK(‘A’)
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhh の RFC4122 ストリング形式準拠
	LEN_ALL	'????????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'????????'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	計算資源解放要求 XML
	LEN_ANS	'????????'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	nothing	未使用

5.1.28. 計算資源解放応答(NAK)

Header	ID	'D'(0x44)(1byte)	固定値
	CMD	'T'(0x46)(1byte)	計算資源解放要求
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud コントローラ識別子
	STAT	'N'(0x4E) (1byte)	ACK('N')
	UUID	(36bytes)	シーケンスを識別するための識別子 書式は hhhhhh-hhhh-hhhh-hhhhhhhhhh の RFC4122 ストリング形式準拠
	LEN_ALL	'???????'(8bytes)	Data 部のバイト数(16 進 8 桁)
Data	LEN_REQ	'00000000'(8bytes)	REQ のバイト数(16 進 8 桁)
	REQ	variable length	計算資源解放要求 XML
	LEN_ANS	'00000000'(8bytes)	ANS のバイト数(16 進 8 桁)
	ANS	nothing	未使用

5.2. 要求 XML

5.2.1. 計算資源リスト取得要求 XML

```
<dsb_resource_list>
  <resource type='physical'/?>
  <resource type='virtual'/?>
  <resource type='network_switch'/?>
  <resource type='power_swit'/?>
</dsb_resorce_list>
```

ルートノードは<dsb_resource_list>
その配下には配下のタグが配置される。

(1) <resource type='(virtual|physical|network_switch)'/>

取得する計算資源リストの種別を指定する。

virtual : 仮想 PC の計算資源リストを取得

physical : 物理 PC の計算資源リストを取得

network_switch : ネットワークスイッチのリストを取得

power_switch : 電源装置のリストを取得

5.2.2. 計算資源確保要求 XML

```
<dsb_resource_desc>
  <target type='physical'/?>
  ...
</target>
<network_switch>
  ...
</network_swich>
<power_switch>
  ...
</power_switch>
</dsb_resorce_desc>
```

ルートノードは<dsb_resource_desc>
その配下には以下の 3 タグが配置される。

(1) <target type=(virtual|physical)>

```

<target type='physical'>
  <!-- node A -->
  <name>nodeA</name>
  <cpu>
    ...
  </cpu>
  <memory type='DDR3'>
    ...
  </memory>
  <devices>
    ...
  </devices>
  <!-- ↑ libvirt から -->
  <os>
    ...
  </os>
  <softwares>
    ...
  </softwares>
</target>

```

仮想 PC あるいは物理 PC を指定する。

仮想 PC か物理 PC かの違いはパラメータ type で指定する。

パラメータ省略時は physical とする。

<target>タグの配下には以下のタグが配置される

- ① <name>
PC の名前を指定
- ② <cpu>

```

<cpu>
  <arch>x86_64</arch>
  <vendor>Intel</vendor>
  <model>core2quad</model>
  <mhz>2400</mhz>
  <cores>4</cores>
  <threads>2</threads>
</cpu>

```

CPU の情報を記述する。

<cpu>タグの配下には以下のタグが配置される

- <arch>
CPU アーキテクチャを記述する。
記述例は以下の通り
 - ・ x86_64
 - ・ x86_32 または i386
 - ・ ia64
 - ・ s390
 - ・ alpha
 - ・ arm
 - ・ ppc32
 - ・ ppc64
 - ・ sparc
 - ・ pa_risc
- <vendor>
CPU のベンダー名を記述
例) “intel” “amd”
- <model>
CPU のモデル名を記述
/proc/cpuinfo の model を参照
- <mhz>
CPU の速度を MH z で指定
- <cores>
CPU のコア数を指定、物理 CPU 数×1CPU 内のコア数
/proc/cpuinfo の cpu cores と同じ
- <threads>
Intel の HTT(Hyper Threading Technology)のような論理的に CPU コアを複数に見せる機能がある場合にその分割を記述
機能がない場合は 1 を指定
/proc/cpuinfo の cpu cores と同じ

③ <memory type='DDR|DDR2|DDR3'>

```
<memory type='DDR3'>
  <speed type='mhz'>800</speed><!-- ?? mhz or mibqs -->
  <size>2048</size>
</memory>
```

PC の持つメモリの速度およびサイズを指定する。指定方法は以下のタグを用いる。

- <speed type='(mhz|mibps)''>

メモリの速度を指定、速度の単位は MH z または Mibps(MiB/s)

- <size>

メモリの容量を MiB 単位で指定

④ <devices>

```
<devices>
  <device type='ethernet'>
    <target dev='eth0' type='lg' vendor='Intel' model='e1000' />
  </device>
  <device type='storage'>
    <controller type='sata' model='ich9'></controller>
    <target dev='/dev/hda' ip='' priority="" type='sata2' vendor='WesternDigital' model='Cavier
Green' size='1T' />
  </device>
</devices>
```

PC に接続されたデバイスについて記述

<devices>タグの配下には以下のタグが配置

- <device type='デバイスの種類'>

デバイス種別には 'ethernet' または 'storage' を指定可能

<device>タグの配下には以下のタグが配置される。

- <target dev='デバイス名' ip='デバイスの IP アドレス' priority='優先順位' type='デバイスの詳細タイプ' vendor='ベンダー名' model='モデル名' size='サイズ'>

対象となるデバイスのデバイス名、IP アドレス、優先順位、タイプ、ベンダー、モデル等を指定

デバイスの IP アドレスが複数存在している場合は優先順位を確認し、primary が設定されているものを使用して通信を行う。

- <controller type='デバイスタイプ' model='モデル'>

デバイスコントローラがある場合に指定

HDD 等の場合デバイスタイプにインタフェースタイプ(sata 等)を、モデルに IO コントローラ・ハブ(ICH9 等)を指定する。

⑤ <os>

```
<os>
  <family>Linux</family>
  <vendor>Fedora project</vendor>
  <name>Fedora</name>
  <version>11</version>
  <arch>x86_64</arch>
</os>
```

OS の情報を記述する。

<os>タグの配下には以下のタグが配置

- <family>
OS の分類を記述
記述可能な OS の分類は”Linux”, ”Unix”, ”Windows”, ”MacOS”, ”Other”
- <vendor>
OS のベンダー名を指定
- <name>
OS の名称を指定
- <version>
OS のバージョン指定
- <arch>
OS のアーキテクチャを指定 (<cpu>の<arch>参照)

⑥ <softwares>

```
<softwares>
  <software type='kernel'>
    <version>2.6.35</version>
  </software>
  <software type='library'>
    <name>libc</name>
    <version>glibc2.6.??</version>
  </software>
  <software type='package'>
    <family>RPM</family> <!--RPM or DEB -->
  </software>
  <software type='compiler'>
    <lang>C</lang>
    <family>GNU</family>
    <name>gcc</name>
    <version>4.4</version>
  </software>
</softwares>
```

ベンチマーク結果に記録すべきソフトウェアの情報を記述

<softwares>タグの配下に以下のタグを配置

- <software>
記録すべきソフトウェア 1 本につき本タグが 1 つ使われる。
<software>タグの配下には以下のタグを配置

- <version>
ソフトウェアのバージョンを記述
- <name>
ソフトウェアの名前を記述
- <family>
ソフトウェアの分類を記述
- <lang>
言語指定、主にコンパイラの場合に使用

⑦ <status>

```
<status>2</status>
```

⑧ <message>

```
<message>target is in usa</message>
```

⑨ <time_until_opening>

```
<time_until_opening>60</time_until_opening>
```

<status>タグで各計算資源の現在の利用状況を示し、<message>タグでその内容を記述する。
各 status と message の対応は以下の通り。

- 0 : success : 利用のため確保成功
- 1 : target is offline : 確保待ち(物理 PC 等で未起動)
- 2 : target is in use : 他で利用中のため確保不可
- -1 : target deos not exist : 指定されたターゲットが存在しない

また<time_until_opening>タグは、該当機器が使用中(status タグに 2 が設定)の場合に次に使用できるまでの時間を設定する。status タグが 2 以外の場合は、設定値を無視する。

<status>タグの値はシナリオで使うターゲットでは 2(target is in use)に設定する。

<time_until_opening>タグの値はシナリオ終了予想時間に設定する。

(2) <network_switch>

```
<network_switch>
  <name>dsbsw</name>
  <control>
    <snmp>
      <addr>dsbsw</addr>
      <version>1</version>
      <community>private</community>
    </snmp>
  </control>
  <port number="1" connected_to="nodeA/eth0" />
</network_switch>
```

ネットワークスイッチを指定する。

<network_switch>タグの配下には以下のタグが配置される。

<name>

ネットワークスイッチの名称を指定

<control>

ネットワークスイッチの制御方式を記述

<snmp>タグの配下には以下のタグが配置

- <addr>
IP アドレスまたはホスト名を指定
 - <version>
SNMP のバージョン指定
 - <community>
SNMP コミュニティ名を指定
- ③ <port number='ポート番号'connected_to='接続デバイス名'>
ネットワークスイッチの保有するポート番号とそこに繋がる機器のデバイス名を指定する

(3) <power_switch>

```

<power_switch>
  <name>dsbpdu</name>
  <control>
    <snmp>
      <addr>dsbpdu</addr>
      <version>1</version>
      <community>private</community>
    </snmp>
  </control>
  <port number="1" connected_to="nodeA" />
</power_switch>

```

SNMP 制御可能な電源装置を指定

<power_switch>タグの配下には以下のタグが配置される。

- ① <name>
電源装置の名称を指定
- ② <control>
電源装置の制御方式を記述
<snmp>タグの配下には以下のタグが配置

- (ア) <addr>
IP アドレスまたはホスト名を指定
- (イ) <version>
SNMP のバージョン指定
- (ウ) <community>
SNMP コミュニティ名を指定
- ③ <port number='ポート番号'connected_to='接続ノード名'>
ネットワークスイッチの保有するポート番号とそこに繋がる機器のデバイス名を指定する。

5.2.3. ベンチマークシナリオ XML

```
<dsbxml>
  <benchmark seq='1' id='1' path='/home/dsb/benchmarks/hackbench/hackbench' title='Hackbench'
type='unit'>
    <description>Hackbench(Load test)</description>
    <client_input host='dsbtarget2'>
      <stime>0</stime>
      <ftime interval='100'>36000</ftime>
      <option label='number of groups' arg=' ' ui='yes' interactive='no'>2</option>
    </client_input>
    <output>
      <caption>Time</caption>
      <table>
        <data>(Time): *(.*)</data>
        <key>10</key>
        <value>01</value>
      </table>
    </output>
    <plain>
    </plain>
    <tags>
      hackbench.core2duo
    </tags>
  </benchmark>
</dsbxml>
```

ベンチマークシナリオ XML をそのまま送信する。

```
<benchmark seq="[num]" title="[str]" path"[path]" type="[unit|link]"></benchmark>
```

この Node でベンチマークの指定を行う。

- ”seq”には、DS-Bench コントローラ側からみた実行順序が記入
- “ID”には、<ssstart><ssstop>でベンチマークを判断するための ID を記入

- "title"には、便宜上の呼称が記入（空白可）
- "path"には、ベンチマークへのフルパスが記入

"type"には、単体ベンチマーク [unit] または C/S 型ベンチマーク [link] が記入されている。

```
<server_input host="[IP]"></server>
```

C/S 型ベンチマークのサーバ側についての設定を記述する。

上位の"benchmark"ノードが持つ"type"が"link"の場合にのみ意味を持つ。

省略：可（type=link の場合、省略不可）、繰り返し：不可

"host"対象となる Server マシンのホスト名または IP アドレスを指定する。

```
<client_input host="[IP]"></client>
```

このノード下で単体、C/S 型ベンチのクライアント側についての設定を記述する。

省略：可、繰り返し：不可

"host"対象となる Client マシンのホスト名又は IP アドレスを指定する。

```
<stime interval="">開始時刻</stime>
```

ベンチマークシナリオの実行開始を 0 とした、ベンチマークの相対的な実行開始時刻を、秒単位で設定する。

パラメータの interval に開始待ち遅延時間をミリ秒単位で指定する。

```
<ftime interval="">終了時刻</ftime>
```

ベンチマークシナリオの実行開始を 0 とした、ベンチマークの相対的な実行終了時刻を、秒単位で設定する。

パラメータの interval に終了待ち遅延時間をミリ秒単位で指定する。

```
<output/>
```

結果の出力形式について指定する。

省略した場合、結果を返却しない。

省略：可、繰り返し：不可

```
<caption/>
```

結果の表題、文字列か正規表現で指定する。

省略：可、繰り返し：不可

```
<table/>
```

結果出力についての指定をこのノード下で行う。

取り出したい表の個数だけこのノードを記述する。

省略：不可、繰り返し：可

<data>[str]…</data>

出力結果の取り出す値を()で括って表記する。

※スクリプト内で正規表現を用いて()で括った部分を抜き出しているため、(())等しないください。

省略：不可、繰り返し：不可

<value>[0|1]*</value>

値の名称をマスクする場合 0 を設定する。

省略した場合、自動的に 1 が入れられたものとする。

省略：可

<key>[0|1]*</key>

値をマスクする場合 0 を設定する。

省略した場合、自動的に 1 が入れられたものとする。

省略：可

<plain></plain>

この Node 下に途中経過を格納する。

繰り返し：不可、省略：可

<tags/>

入力されたタグはカンマ区切りでこのタグ中に記録される。

5.2.4. ベンチマーク開始要求 XML

ベンチマーク開始要求に使用する XML はシナリオ送信に使用した XML をそのまま指定する。

5.2.5. デーモン生存確認要求 XML

```
<dsb_resource_desc>
  <target type='physical'>
    ...
  </target>
</dsb_resource_desc>
```

デーモン生存確認要求に使用する XML は計算資源確保要求 XML の中から生存確認を行うターゲットのみを抜き出したものである。

5.2.6. 計算資源解放要求 XML

```
<dsb_resource_desc>
  <target type='physical'>
    ...
  </target>
  <network_switch>
    ...
  </network_switch>
  <power_switch>
    ...
  </power_switch>
</dsb_resource_desc>
```

計算資源解放要求に使用する必要 XML は計算資源確保応答 XML に等しい。

5.3. 応答 XML

5.3.1. 計算資源リスト応答 XML

```
<dsb_resource_desc>
  <target type='physical'>
    <name>nodeA</name>
    <cpu>
      ...
    </cpu>
    <memory type='DDR3'>
      ...
    </memory>
    <devices>
      ...
    </devices>
    <os>
      ...
    </os>
    <softwares>
      ...
    </softwares>
    <status>2</status>
    <message>target is in use</message>
    <time_until_opening>60</time_until_opening>
  </target>
  <network_switch>
    ...
  </network_switch>
  <power_switch>
    ...
  </power_switch>
</dsb_resource_desc>
```

取得した計算資源 1 つに対して 1 つの<target>タグあるいは<network_switch>タグ、<power_switch>タグを返却する。

各タグの内容は [5.2.2 計算資源確保要求 XML](#) を参照。

5.3.2. 計算資源確保応答 XML

[5.2.2 計算資源確保要求 XML](#) と同じ形式の XML を返却する。

5.3.3. ベンチマークシナリオ応答 XML

```
<dsbxml>
  <benchmark seq='1' path='/home/dsb/benchmarks/hackbench/hackbench' title='Hackbench' type='unit'>
    <description>Hackbench(Load test)</description>
    <client_input host='dsbtarget2'>
      <stime>0</stime>
      <ftime>36000</ftime>
      <option label='number of groups' arg=' ' ui='yes' interactive='no'>2</option>
    </client_input>
    <output>
      <caption>Time</caption>
      <table>
        <data>(Time): *(.*)</data>
        <key>10</key>
        <value>01</value>
      </table>
      <plain>
      </plain>
    </output>
    <status>0</status>
    <message>success</message>
  </benchmark>
</dsbxml>
```

5.2.3 ベンチマークシナリオ XML とほぼ同じ形式の XML を返却する。

違いは、各<benchmark>タグの中に<status>タグと<message>タグが付いているところ。

<status>タグと<message>タグの関連とその意味は次の通り

status	message	意味
0	success	正常
1*		Parser 周りの Error
11	parser error	Parser の呼び出しと XML のパースに失敗
12	fail in parse of schema	RelaxNG Schema のパース失敗
13	fail in validator	Validate に失敗
14	cannot resolve the host name	ホスト名の解決に失敗
2*		benchmark 階層の設定 Error
21	seq is not numeric	seq に数字以外が混入
22	PATH is not a file	PATH 指定先がファイルではない
23	don't hava permission in PATH	PATH 指定先に実行権がない

24	type is invalid	type の指定が不正
25	the combination is wrong	subcmd=S と type=unit という組み合わせ
26	seq is overlapped	seq が重複
3*		未使用
4*		C/S input 周りの設定 Error
41	stime is not numeric	stime に数字以外が入っている
42	ftime is not numeric	ftime に数字以外が入っている
43	label is invalid character	label に未対応文字が入っている
44	arg is invalid character	arg に未対応文字が入っている
45	ui must be yes or no	ui に yes または no 以外の文字が入っている
46	interactive must be yes or no	interactive に yes または no 以外の文字が入っている
5*		未使用
6*		Output 周りの設定 Error
61	parentheses is unbalanced	括弧の閉じ忘れ
62	parentheses is succession	入れ子の括弧が存在
7*		未使用
-1	benchmark does not exist	指定されたベンチマークが存在しない

5.3.4. ベンチマーク状態 XML

```
<dsbxml>
  <benchmark seq="1" title="" path="" type="">
    <caption></caption>
    <config></config>
    <plain>
      </plain>
    </benchmark>
</dsbxml>
```

ベンチマーク状態取得応答で返却される XML

ANS 部分に格納されている。

```
<benchmark seq="[num]" title="[str]" path"[path]" type="[unit|link]"></benchmark>
```

この Node でベンチマークの指定を行う。

- ”seq”には、DS-Bench コントローラ側からみた実行順序が記入されている。
- ”title”には、便宜上の呼称が記入されている。空白可
- “path”には、ベンチマークへのフルパスが記入されている。
- “type”には、単体ベンチマーク[unit]またはC/S型ベンチマーク[link]が記入されている。

繰り返し：可、省略：不可

<caption>[str]</caption>

結果の表題、文字列か正規表現で指定。

省略：可、繰り返し不可

<config></config>

未使用、未設定

繰り返し：不可、省略：可

<plain></plain>

この Node 下に途中経過が格納される。

繰り返し：不可、省略：可

5.3.5. 計算資源確保応答 XML

```
<dsb_resource_desc>
  <target type='physical'>
    <name>nodeA</name>
    <cpu>
      ...
    </cpu>
    <memory type='*>
      ...
    </memory>
    <devices>
      ...
    </devices>
    <os>
      ...
    </os>
    <softwares>
      <software type='kernel'>
        <version>2.6.35</version>
      </software>
      <software type='library'>
        <name>libc</name>
        <version>glibc2.6.??</version>
      </software>
      <software type='package'>
        <family>RPM</family> <!-- RPM or DEB -->
      </software>
      <software type='compiler'>
        <lang>C</lang>
        <family>GNU</family>
        <name>gcc</name>
        <version>4.4</version>
      </software>
    </softwares>
    <status>2</status>
    <message>target is in use</message>
    <time_until_opening>60</time_unitl_opening>
  </target>
</dsb_resource_desc>
```

[5.2.2 計算資源確保要求 XML](#) と同じ形式の XML を返却する。

デーモンが存在していればその時点での最新のソフトウェア状況を<software>タグ内に格納して D-Cloud へ返却を行う。

各タグの内容については [5.2.2 計算資源確保要求 XML](#) を参照

5.3.6. 計算資源確保応答 XML

```
<dsbxml>
  <benchmark seq="1" path='${FAULTPATH}/fault' title="Fault" type="unit" constraints='netlink'>
    <description>xxxxxxxxxxxxxxxxxxxxxxxx</description>
    <client_input host='dsbtarget2'>
      <stime interval='15'>0</stime>
      <ftime interval='37'>36000</ftime>
      <option input='test' label='number of groups' arg=' ' ui='yes' interactive='no'>2
      </option>
    </client_input>
  </benchmark>

  <benchmark>...</benchmark>
  <benchmark>...</benchmark>
  <benchmark>...</benchmark>
  <benchmark>...</benchmark>
</dsbxml>
```

[4.1 各ベンチマーク conf ファイル](#) とほぼ同じ形式の XML を返却する。

違いは<benchmark>タグ内に<constraints>属性がつくことである。

constraints 属性の取りうる値は以下の通りです。

vm : 仮想マシン専用の fault injection

netlink : スイッチ制御によるネットワークリンク断が可能な fault injection

pdu : PDU による電源制御が可能な fault injection

ipmi : IPMI による電源制御が可能な fault injection

フォルトリスト XML には複数の<benchmark>タグがあり、利用できる全てのフォルトがこの XML に記述される。

5.4. D-Case Editor との通信

D-Case Editor との通信 XML について説明する。

5.4.1. シナリオ実行開始

シナリオ実行開始 CGI では、レスポンスボディが変更になる。XML の例を示す。

```
<?xml version="1.0" encoding="utf-8"?>
<Result scenario_name="" exec_id="" code="" message="" estimate_completion_time="" view_id="" />
```

view_id 属性を新たに追加し、これが DS-Bench の GUI で表示される結果 ID を表す。

5.4.2. シナリオ結果閲覧

シナリオ結果閲覧 CGI (view_result_detail.cgi) を新たに呼ぶようにする。リクエスト/レスポンスの例を示す。

- リクエストメソッド : GET
- リクエストヘッダ : view_id=scenario1

- レスポンスボディ：ブラウザで表示可能な HTML（画面例は 6.4.view_result_detail）

5.4.3. シナリオ中止

シナリオ中止 CGI (stop_scenario.cgi) を新たに作成する。リクエスト／レスポンスの例を示す。

- リクエストメソッド：POST
- リクエストボディ：

```
<?xml version="1.0" encoding="utf-8"?>
<Scenario name="" exec_id="" />
```

- レスポンスボディ：

```
<?xml version="1.0" encoding="utf-8"?>
<Result exec_id="" code="" message="" />
```

以下に、レスポンスで用いる code と message の詳細を示す

シナリオ中止状況ステータス

code	message	詳細
0	success	正常終了
1*		Parser 周りのエラー
11	request XML parser error	リクエスト XML のパースに失敗
5*		IO エラー
52	parameter do not exist.	指定されたパラメータが存在しない
9*		その他
94	the scenario is not running.	シナリオが実行停止している
95	D-Case Editor Error: Scenario has not executed from D-Case.	指定されたシナリオが D-Case Editor から実行されていない

シナリオ実行状況ステータス

status	詳細	前版から
stopping	シナリオ中止中	新規追加
stopped	シナリオ中止完了	新規追加
st_error	シナリオ中止中のエラー	新規追加

6. 画面仕様

6.1. 共通メニュー



図 6-1 共通ヘッダ

サイトタイトル「DS-Bench」リンクをクリック時、Main 画面 (main.cgi) に遷移

「Main」リンクをクリック時、Main 画面 (main.cgi) へ遷移

「Synopsis」リンクをクリック時、Synopsis 画面 (synopsis.cgi) へ遷移

「View Result」リンクをクリック時、Result 画面 (view_result.cgi) へ遷移

「Configuration」リンクをクリック時、Configuration 画面 (configuration.cgi) へ遷移

6.2. main 画面

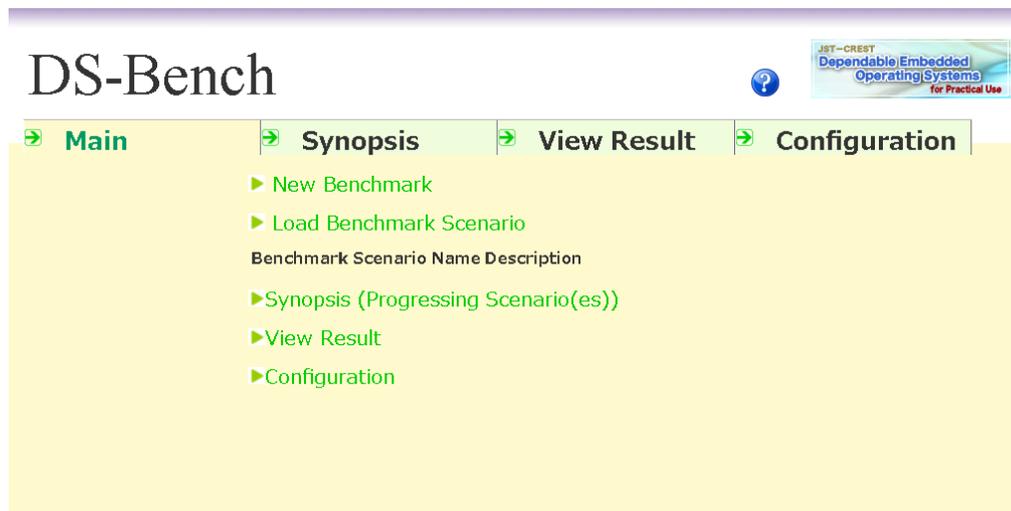


図 6-2 Main 画面

「New Benchmark」リンクをクリック時、新規ウィンドウで Benchmark 画面 (benchmark_main.cgi) へ遷移→Benchmark 画面

「Load Benchmark Scenario」リンクをクリック時、新規ウィンドウで Benchmark 画面へ遷移し、過去に保存したシナリオ一覧を表示→ベンチマークシナリオ読み込み画面

「Synopsis(Progressing Scenario(es))」リンクをクリック時、Synopsis 画面 (synopsis.cgi) へ遷移

「View Result」リンクをクリック時、Result 画面 (view_result.cgi) へ遷移

「Configuration」リンクをクリック時、Configuration 画面 (configuration.cgi) へ遷移

6.3. Benchmark 画面

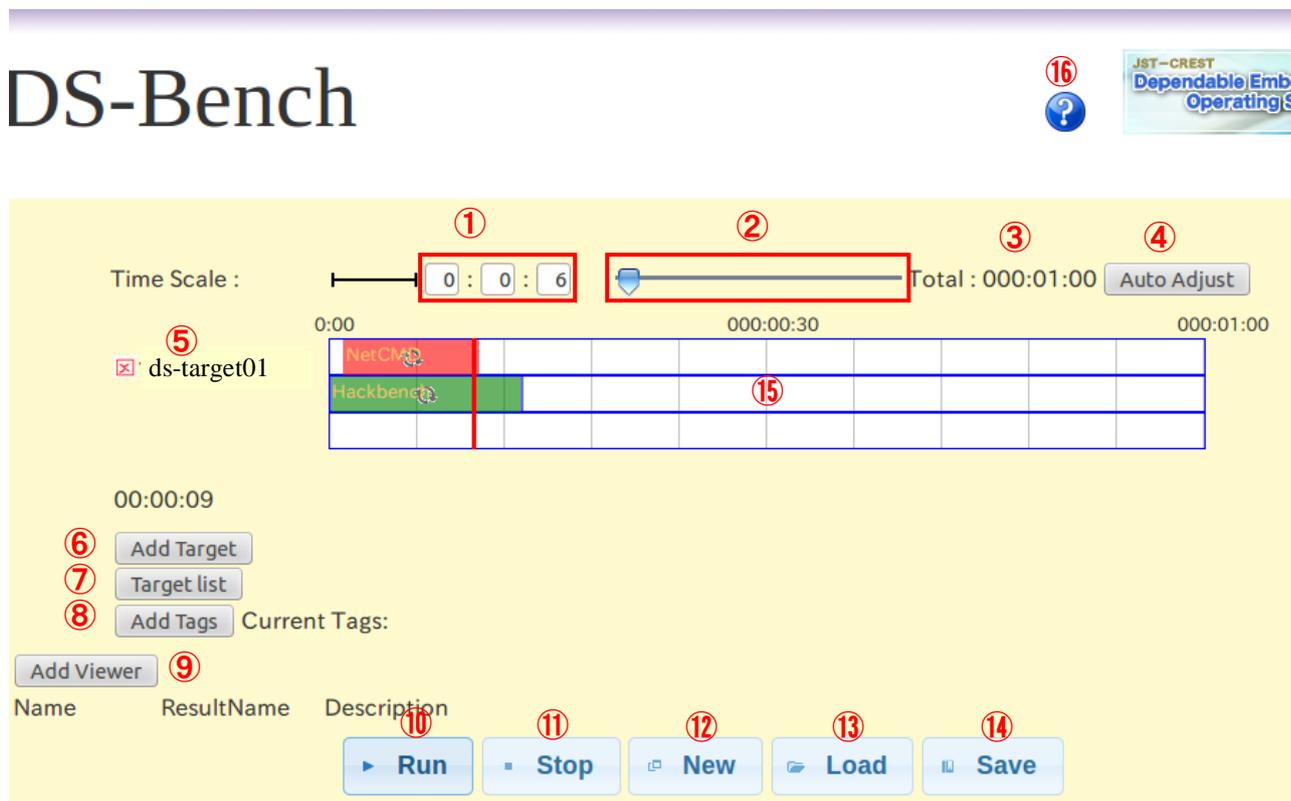


図 6-3 Benchmark 画面

以下、図 6-3 Benchmark 画面の丸数字について簡単説明します。

- ① タイムスケール⑮の 1 マス分の時間の長さを設定できるフォーム（③の全体時間も変化する）左側から、時、分、秒
例) 1 : 3 : 6 の場合、1 マス 1 時間 3 分 6 秒でトータル 10 時間 31 分となる
- ② 左右にドラッグすることにより、タイムスケールの時間を変更できるつまみ
- ③ 現在表示されているタイムスケール⑮の全体時間が表示される
- ④ 「Auto Adjust」ボタン・・・タイムスケール上に存在する最も長いベンチマーク終了時刻が全体の 80% の位置に来るように再配置するボタン
- ⑤ 「Add Target」ボタン⑥により追加された DS-Bench ターゲットの D-Cloud に登録した時の name が表示される
左端の削除アイコンをクリックするとターゲットが削除される
表示された name をクリックすると、ターゲットの設定を行える画面が表示される（ターゲット編集画面）
- ⑥ 「Add Target」ボタン・・・新しいターゲットの設定を追加する画面が表示される(6.3.1 ターゲット追加画面)
- ⑦ 「Target list」ボタン・・・ターゲットマシンの一覧表示

- ⑧ 「Add Tags」 ボタン・・・現在編集中のベンチマークシナリオにタグの編集を行える画面が表示される
- ⑨ 「Add Viewer」 ボタン・・・D-Case Editor/D-Case Weaver に渡す結果名に対応する名前、ベンチマーク結果、説明文の入力ボックスをクリック回数表示
- ⑩ 「Run」 ボタン・・・現在のベンチマーク構成でベンチマークを開始する
- ⑪ 「Stop」 ボタン・・・開始したベンチマークを終了する
- ⑫ 「New」 ボタン・・・現在のベンチマーク構成を破棄して新しいベンチマーク構成を作成する
- ⑬ 「Load」 ボタン・・・既に保存されているベンチマーク構成を読み込むための画面が表示される
- ⑭ 「Save」 ボタン・・・現在のベンチマーク構成を保存する為の画面が表示される
- ⑮ タイムライン上のベンチマーク構成
- ⑯ Help 画面へ遷移

6.3.1. ターゲット追加ダイアログ

「Add Target」 クリックで Add Target ダイアログ表示

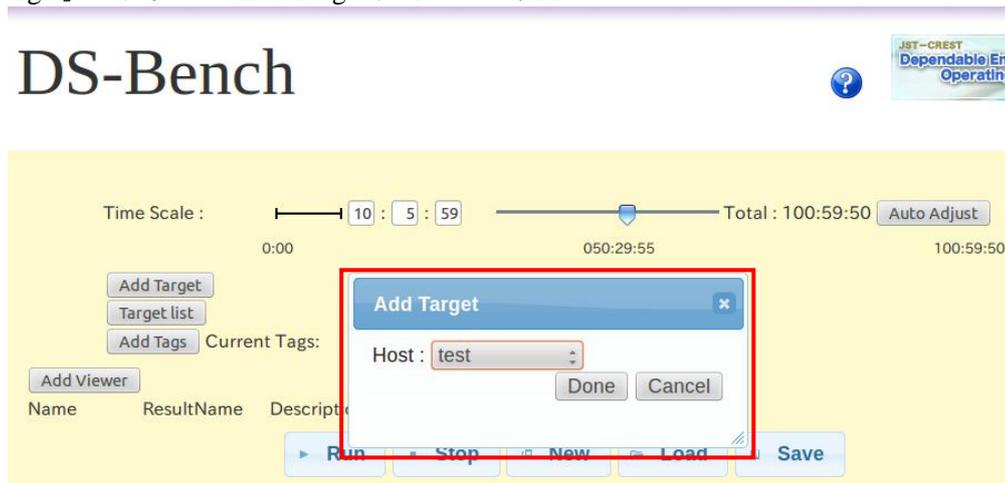


図 6-4 ターゲット追加ダイアログ

新しいターゲット情報を追加します。

Host にターゲットとして D-Cloud に登録した DS-Bench ターゲットが一覧で表示されるので使用するターゲットを選択でき、「Done」 ボタンをクリックで決定し、ダイアログボックスが閉じます。

「Cancel」 ボタンをクリックすると何もせずにダイアログボックスを閉じます。

6.3.2. ターゲット編集ダイアログ

表示されている①のターゲット名をクリック

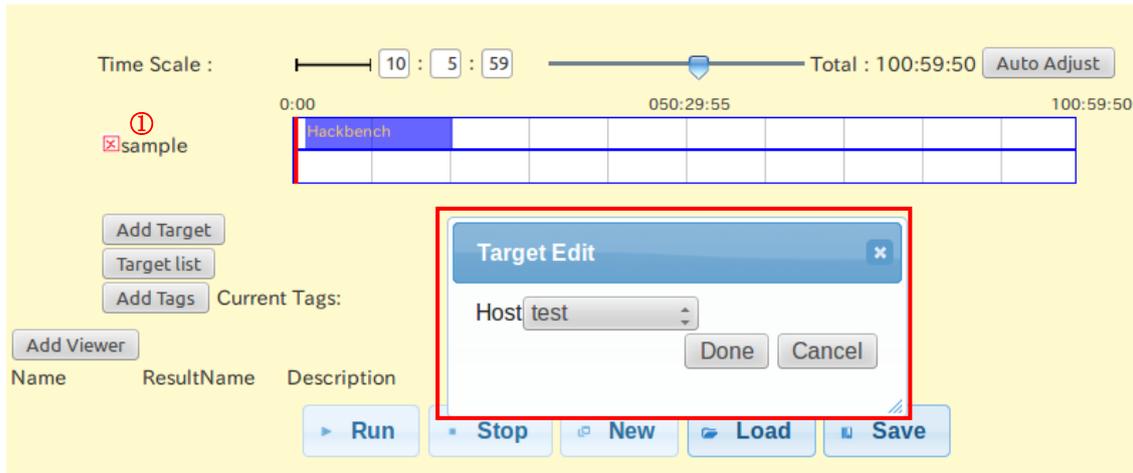


図 6-5 ターゲット編集ダイアログ

6.3.1 ターゲット追加画面によって追加されたターゲットの情報を編集します。

Host にベンチマークを実行したいターゲットを選択し、「Done」ボタンをクリックで決定し表示されていたターゲット名①が選択したターゲット名に変更しダイアログボックスを閉じます。

「Cancel」ボタンをクリックすると何もせずにダイアログボックスが閉じます。

6.3.3. ターゲットリスト

「Target list」ボタンクリックでターゲット一覧とそのステータスを「Run」ボタンの下に表示し確認を行える。また画面下の「Back」ボタンで元にもどる。

画面表示は Configuration の Target list と同じで Configuration との違いは、Configuration では「Back」ボタンが「Refresh」ボタンになる。

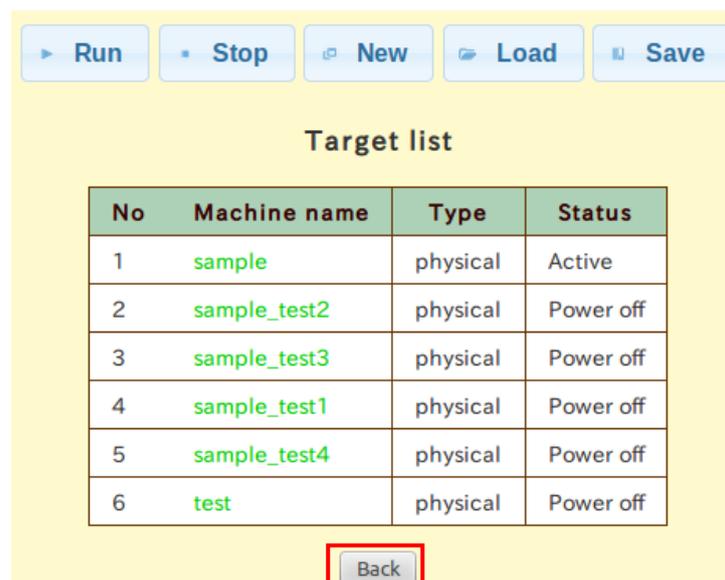


図 6-6 ターゲットリスト表示

詳しい説明は Configuration の Target list で

6.3.4. タイムスケール上での操作法について

6.3.4.1. ベンチマーク間の関係結合（終了同期）

タイムスケール上でドラッグ動作を行うことにより、指定したターゲットに対してベンチマークまたは Fault Injection を登録することができます。

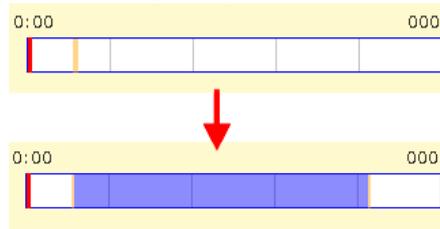


図 6-7 タイムスケール上でドラッグした際の表示

画面上では、ベンチマークの場合青の四角（ベンチマーク領域）で、Fault Injection の場合は赤の四角で表示されます。

このとき設定される時間はドラッグの開始位置と終了位置で決定しそれぞれの位置の時間がセットされます。

ドラッグ終了後、ベンチマーク詳細設定ダイアログが表示されるので、実行したいベンチマークの種類やオプションを編集できます。

作成したベンチマークはタイムスケール上でもマウス操作により開始時刻と終了時刻について編集可能です。

ベンチマーク領域の上（両端以外）にマウスカーソルを重ねドラッグ操作を行うと、そのベンチマークの実行時間を変えずに開始時刻を変更することが可能です。ドラッグ先がタイムスケール上において左端からはみ出していた場合、開始時刻を 0:00 とし、終了時刻が実行時間の長さに調整されます。他のベンチマークの実行時刻と重なってしまった場合や、右端からはみ出してしまった場合、元の位置に戻されます。

また、ベンチマーク領域の両端でドラッグ操作を行うと、ベンチマーク領域の大きさが変わり、ベンチマークの実行時間を変更できます。ドラッグ先がタイムスケール上において左端からはみ出していた場合、開始時刻が 0:00 となります。同様に右端からはみ出していた場合、終了時刻がタイムスケールの長さとなります。

同一時間帯に複数ベンチマーク/Anomaly load が登録でき、ゲージの幅は初期時 1 行分、1 行使用されるたびに 1 行分の幅が追加される。（使用行数+1 行確保され、最大幅は 8 行分）

設定した時間帯を上下の行に移動することはできない。

6.3.4.2. ベンチマーク間の関係結合

関係結合とは、同一あるいは異なるターゲット間において複数のベンチマーク/Anomaly load の関係付けを行うことを言い、2 つのベンチマーク/Anomaly load を配置し、どちらかのベンチマーク/Anomaly load をドラッグし、もう 1 つのベンチマーク/Anomaly load に重なるように移動すると確認

ダイアログ「Which synchronous relation do you want to connect between A to B?」が表示される。ターゲットが異なる場合も、ゲージの上下移動は行われませんが、動作は同じで片方のベンチマーク/Anomaly load をもう片方のベンチマーク/Anomaly load に重ねると、別々のターゲット同士で関係結合を行える。

(A、B にはそれぞれのベンチマーク/Anomaly load の名称が入る。)

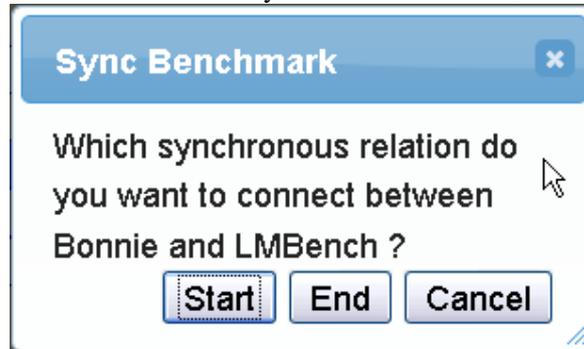


図 6-8 関係結合ダイアログ

「start」「end」ボタンをクリックすることで2つのベンチマーク/Anomaly load が関係付けられます。

「start」の場合、2つのベンチマーク/Anomaly load が関係付けられ、1つ目のベンチマーク/Anomaly load が起動後、各ベンチマーク/Anomaly load に設定された開始待ち時間「Interval before starting(ms)」で設定された時間が経過後に2つ目のベンチマーク/Anomaly load が起動するように設定される。(ドロップされたほうが1つ目、ドラッグしたほうが2つ目として解釈する。)

「end」の場合、2つのベンチマーク/Anomaly load が関係付けられ、1つ目のベンチマーク/Anomaly load が終了後、各ベンチマーク/Anomaly load に設定された開始待ち時間「Interval after termination(ms)」で設定した時間が経過後に2つ目のベンチマーク/Anomaly load が起動される。(ドロップされたほうが1つ目、ドラッグしたほうが2つ目として解釈する。)

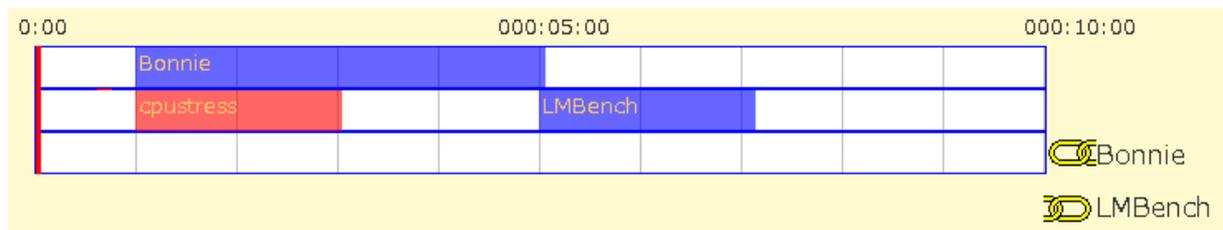


図 6-9 関係結合実行時のタイムライン

6.3.5. ベンチマーク詳細設定ダイアログ

図 6-7 の赤枠は全ての選択 Program において共通項目で、青枠はその選択 Program 固有の項目になります。

図 6-10 ベンチマーク設定ダイアログ

「Target : 」 現在編集を行っているターゲットの名称を表示

「Program : 」 Program ドロップダウンリストとして一覧表示、実行したいベンチマークを選択（ターゲットが物理マシンか仮想マシンかによってそれに対応したリストを表示する）

「Anomaly load チェックボックス」・・・実行するベンチマークを Anomaly load として起動するか判断する為のチェックボックス（チェック有=Anomaly load、チェックなし=ベンチマーク）

「Begin Time : 」 ベンチマークの開始時間を設定（初期状態は 6.3.3 でドラッグした時間が表示）

「End Time : 」 ベンチマークの終了時間を設定（初期状態は 6.3.3 でドラッグした時間が表示、指定時間が開始時刻前だと反せず初期値を表示）

「Interval before starting(ms) : 」 ベンチマークが起動するまでに要すると想定される時間を設定

「Interval after termination(ms) : 」 ベンチマークが終了するまでに要すると想定される時間を指定

「Done」・・・ボタンをクリックすると編集した内容でベンチマークシナリオを決定

「Delete」・・・ボタンをクリックするとタイムライン上からベンチマークを削除

「Cancel」・・・ボタンをクリックすると何もせずにダイアログボックスが閉じる

6.3.5.1. インターバルタイム

「Interval before starting(ms)」で指定した開始までの想定時間は、関係結合を行い「あるベンチマーク/Anomaly load が起動したら起動する」と設定された際に「あるベンチマーク/Anomaly load」が起動したと判断するためのインターバルタイマーとして利用される。

「Interval after termination(ms)」で指定した終了までの想定時間は、関係結合を行い「あるベンチ

マーク/Anomaly load が終了したら起動する」と設定された際に「あるベンチマーク/Anomaly load」が終了したと判断するためのインターバルタイマーとして利用される。また、この時間内にベンチマークプログラムが終了しなかった場合はシステムにより強制的に KILL される。

6.3.5.2. 固有の設定項目

固有の設定項目（ベンチマークのパラメータ）は、ベンチマークによって異なり、パラメータによってはドロップダウンリストで選択出来たり、実行時間をマクロ変数として受け渡すことが出来たり、D-Case に値を渡すことが出来る。

指定可能なマクロ変数は以下の通り。

- \$(runsec)実行時間を秒に変換して置き換える

本マクロ関数が指定可能なテキストボックスにフォーカスがあたり、ツールチップで指定可能なマクロ変数が表示される。

DS-Bench では次のような形式で記述することで D-Case Editor/D-Case Weaver から指定されるパラメータ名称とパラメータ説明、その設定される値を設定する。ただし、図 6-7 の Interface のようにドロップダウンリストより選択するパラメータについては対応しない。

`$(parame : description { define-name } : value)`

上記記述の可変部について以下に記述する。※記号({}):は、半角文字である。

パラメータ名	意味	説明
define-name	D-Case パラメータ	D-Case がシナリオ実行時にベンチマーク起動パラメータ値を変更するための名称。入力可能な文字列は半角-（半角ハイフン）_（半角アンダースコア）のみを使用可能とする。
description	説明	D-Case パラメータの詳細説明を記述。省略可能とする。\$<(){}:¥/”を使用可能とする。
value	値	ベンチマーク起動パラメータ値

\$(param:・・・)の形式でない場合、ベンチマーク起動パラメータ値が設定されていると判断する。D-Case Editor より define-name に記述された名称がパラメータで指定された場合に、ベンチマーク起動パラメータ値を D-Case Editor で指定された値に変更して実行する。

ドロップダウンリストで表示される場合は、D-Cloud から取得した資源リストから判断し、リストを作成し一覧で選択できる。

6.3.6. Fault injection の利用

ベンチマークと同様のイメージで D-Cloud から提供される Fault injection を利用することができる。Fault injection はベンチマーク同様に Program ドロップダウンリストに表示される。ここで表示される

候補はフォルトリスト応答で取得できる。ただし、フォルトの条件が現在のターゲットに合致していない場合、そのフォルトを表示しない。例えば、あるフォルトが仮想マシンのみを対象とするプログラムであった場合、現在のターゲット実マシンであればそのフォルトがリストに現れない。

Program にフォルトを選択すると、Anomaly load チェックボックスが自動的に選択/非活性状態になる。その他は他のベンチマークを選択した場合と同じ挙動を示す。

6.3.7. タグの追加

「Add Tag」ボタンは現在編集中のベンチマークシナリオにタグを追加するためのダイアログを表示。

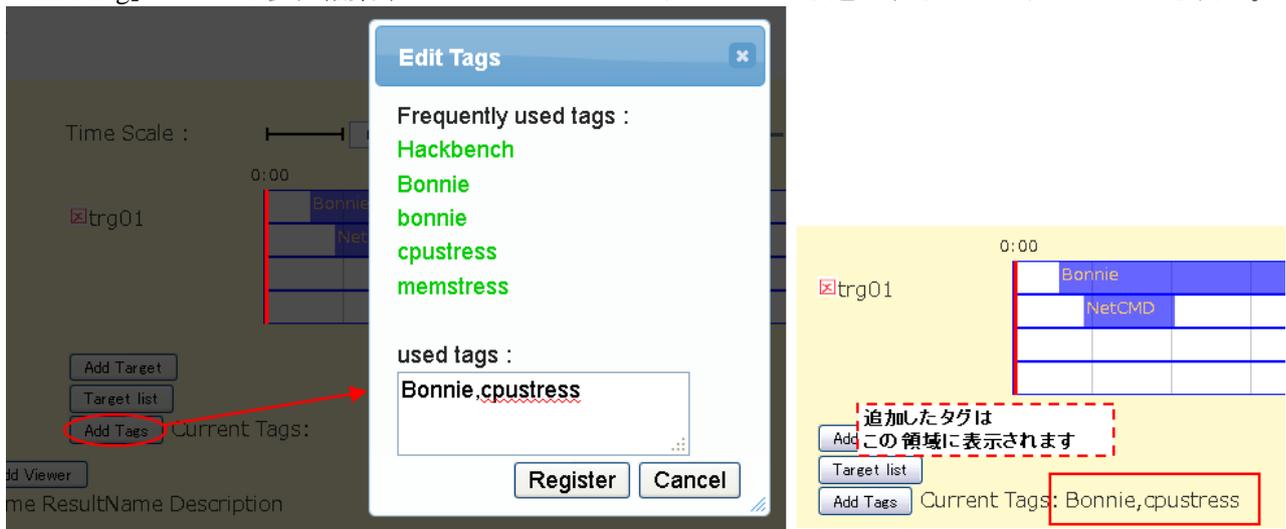


図 6-11 Edit Tags ダイアログ

「Frequently used tags :」によく付けられるタグを最大で上位 5 つまでアンカーとして表示する。アンカーをクリックすることでそのタグを used tags : に付与される。

「used tags :」では、テキストボックスに登録したいタグ名を入力する。タグを複数登録する場合はタグを、(半角カンマ) で区切る。

「Register」 ボタンをクリックすると追加したタグをシナリオに登録する。

「Cancel」 ボタンをクリックすると何もせずにダイアログボックスが閉じる。

登録したタグはベンチマークシナリオの<tags>内に格納される。

6.3.8. シナリオ実行と結果

「Run」 ボタンをクリックでシナリオを開始、ベンチマークを実行する。ベンチマーク実行中は、赤線で示した現在時刻がタイムライン上を移動し、視覚的にベンチマーク実行状態を確認できる。ベンチマーク実行中は生ログが下のボックスに表示される。

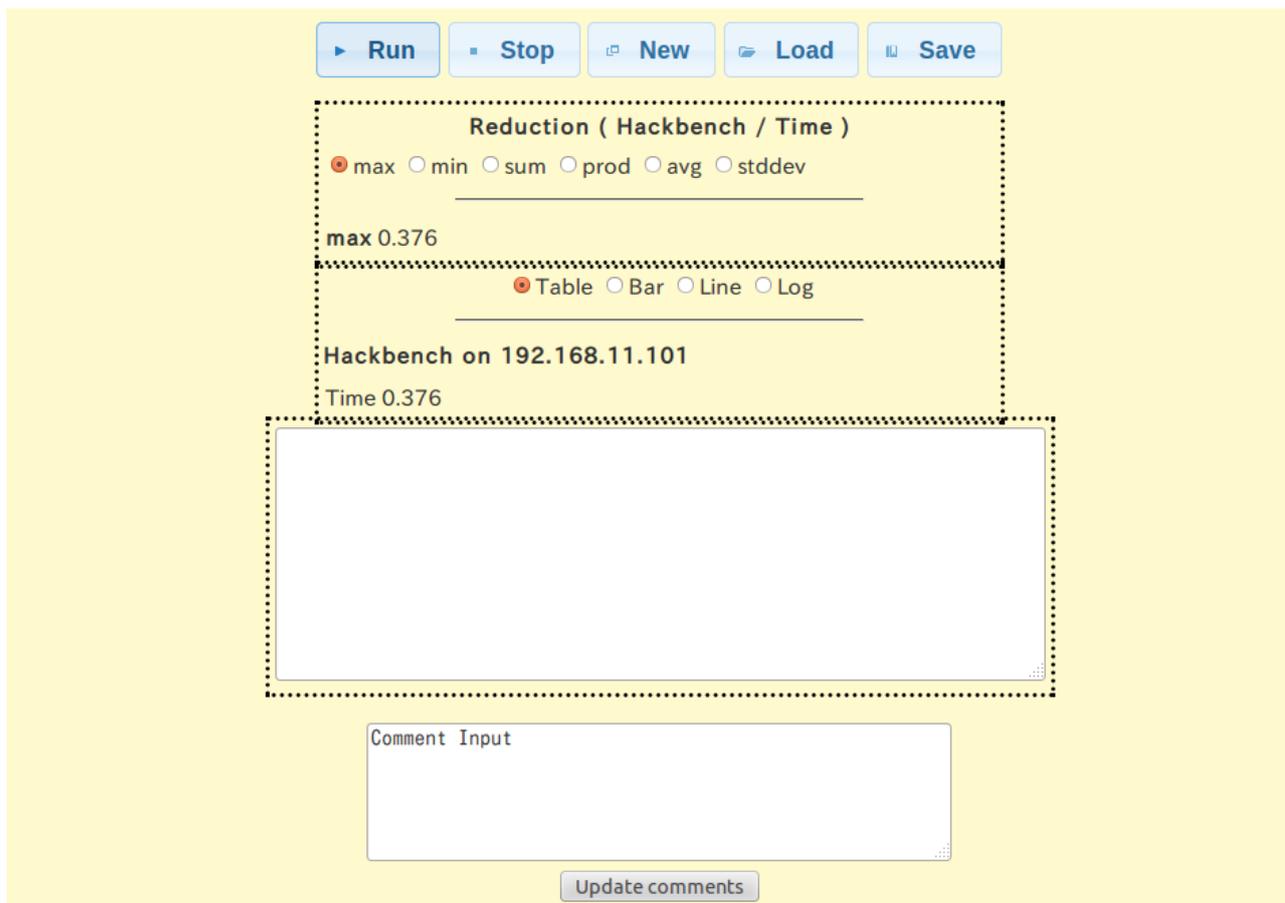


図 6-12 結果表示

ベンチマークが終了する際にリダクションの結果を表示する。リダクション機能とは、複数の同じベンチマークから結果を集約・演算する機能を指し、演算の種類は MAX (最大値)、MIN (最小値)、SUM (総和)、PROD (総乗)、AVE (平均値)、STDDEV (標準偏差) である。この結果は Reduction として他のベンチマークと同じように結果表を表示し、表示切り替え用のラジオボタンを配置する。Reduction テーブルは演算の必要のないようなシナリオであっても表示する。

結果が全て DS-Bench コントローラで取得済みの場合「Table」「Bar」「Line」「Log」のラジオボタンが表示される。

クリックで表示を切り替え、それぞれ「表形式」「棒グラフ」「折れ線グラフ」「生ログ」に切り替えることが出来る。※ベンチマーク実行中はログが中途半端なため切り出しルールが適用で出来ず表示できない。

全てのベンチマークが終了すると、その旨を伝えるメッセージが表示され、実行結果に対するコメントを書くフォームが表示され、コメント入力後「Update comments」ボタンで保存する。

6.3.9. D-Case との連携

「Add Viewer」ボタンをクリックすると、D-Case Editor 連携で使用する結果情報を記述するテキストボックスを表示します。

これは Name、ResultName、Description からなり、それぞれベンチマーク結果名に対応する名前、ベンチマーク結果名、説明文（D-Case Editor に表示させるためのもの）を表す。ResultName（ベンチマーク結果名）には、ベンチマークと結果の順番で指定する。

参照したい結果の指定方法には2通りあり、1つ目は Table 表示の上から何番目かを指定する方法。この場合の表記は以下の通り

`$(benchmark:ターゲット名: 何番目のタイムライン:何番目のベンチマーク: Table 表示の何番目)`

図の 6-13 ではターゲットは「trg3」、タイムライン 1 行目の 1 番目のベンチマークプログラムを実行した結果の Table の上から 18 番目の値を D-Case Editor に渡す場合の設定は以下の通りになる。

`$(benchmark:trg3:1:1:18)`

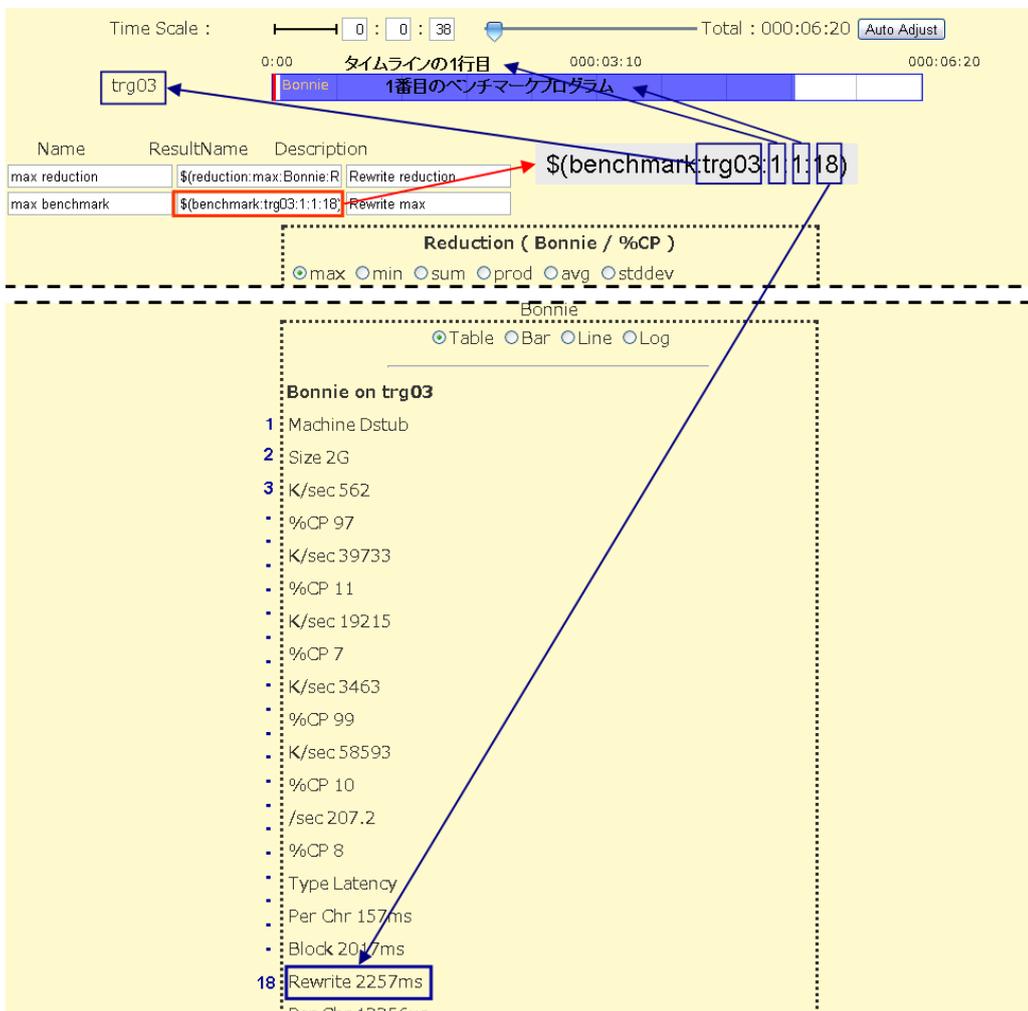


図 6-13 結果表示

もう一通りの方法はリダクションの結果を表わす名前を入れる。

この場合の表記は以下の通り

`$(reduction:リダクション選択:ベンチマーク名:ラベル名)`

図 6-14 では、Bonnie のリダクション結果から Rewrite ラベルの最大値を D-Case Editor に渡す場合の設定は以下になる。

`$(reduction:max:Bonnie:Rewrite)`

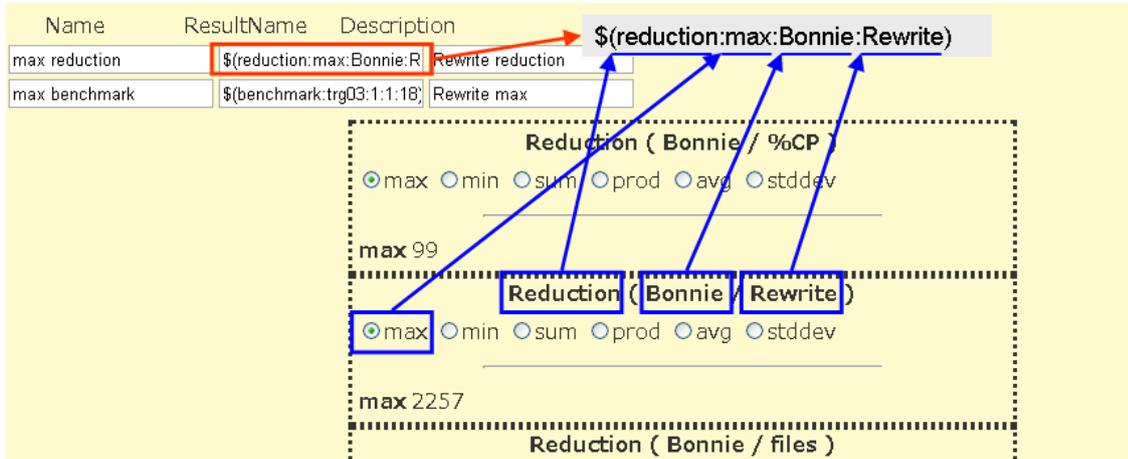


図 6-14 結果表示

6.4. Synopsis 画面

ID	Date	Status Progress Remaining	Comment	Benchmark Program	Anomaly Load	Starter	View Scenario detail	Stop
334	2012/03/14 18:16:04	reserve	D-Case Editor連携 試験用シナリオ 02(仮想マシン)	Iperf		gui	View	Stop
333	2012/03/14 18:05:52	run (35%) 2min	Bonnie単一シナリオ	Bonnie		external	View	Stop
331	2012/03/14 18:01:33	run (78%) 0min	D-Case Editor連携 試験用シナリオ 01(物理マシン)	Bonnie		gui	View	Stop
332		error DS-Bench Error: failure to reserve calculation resources.				gui	View	Stop
330	2012/03/14 09:09:27	finish	Httpperf and Hackbench,Bonnie	Httpperf Httpperf	Hackbench Hackbench Bonnie Bonnie	gui	View	Stop

図 6-15 Synopsis 画面

Synopsis 画面では、現在実行中および完了していても実行後 3 日間以内のベンチマークシナリオ(最大 20 件)が一覧として表示される。

この画面は DS-Bench 画面上部のメインタブから呼び出せる。

表示される項目は以下のとおり。これらはステータスファイルと、そこから逆引きしたシナリオ XML ファイルの内容から取得する。

- ID : 結果 ID (ViewResult 画面に現れる)
- Data : シナリオ開始日時
- Status Progress Remaining : 進捗状況 (ステータス) および消化割合、残り時間 (min)
- Comment : シナリオの説明・コメント
- Benchmark Program/Anomaly Load : 使用したベンチマーク/アノマリー/Fault injection
- Starter : どこから実行したシナリオか (DS-Bench | External)

このうち結果 ID はハイパーリンクになっており、クリックすることで対応するシナリオの結果画面に遷移する。

この画面では他に、更新 (**Reload**) ボタンと各シナリオに対する中止 (**Stop**) ボタン、シナリオ詳細表示 (**View**) ボタンを用意する。更新ボタンでは、新たにシナリオ状況を取得しなおし、一覧部の表示を更新する。なおこの画面は自動では更新されない。

中止ボタンではシナリオを中止できる。このボタンは完了済みのシナリオでは非活性にする。なおこの画面では **D-Case Editor** から実行されたシナリオも見、中止することができるが、中止のための確認ダイアログを二重にするものとする。

シナリオ詳細表示ボタンをクリックすると、対応するシナリオの詳細画面を表示する。このとき画面は新しいウィンドウで開くが、同じシナリオは重複して開かないような処理を加えるものとする。このボタンは完了済みのシナリオでは非活性にする。

現在どのシナリオが実行中なのか判断するために、**D-Case Editor** で使用しているステータスファイル (`run_status_*`: シナリオごとにステータスと消化割合が記述されている) を利用する。これは **D-Case Editor**、**DS-Bench** からの実行でも作成される。このファイルが存在するシナリオが一覧化の対象となる。

実行終了したシナリオをいつまでも表示しては不便なので、一覧画面がリクエストされた時点で、開始後 3 日以上経過しているシナリオのステータスファイルを削除し、さらに 20 件を上回るファイルが存在する場合、20 件になるまで古いものから削除する (進行中のシナリオのものは削除しない) 処理を追加する。これによって、この画面で表示するシナリオを開始後 3 日以内の最大 20 件に抑えることができる。20 件以上のシナリオが現在動いているような場合では、その限りではない。

Synopsis 画面から **Benchmark** 画面をリクエストした際、完了済みのシナリオだった場合は、該当する **ViewResultDetail** 画面へリダイレクトする。実行中のシナリオだった場合、シナリオの詳細を表示した後、赤の実行バーをシナリオ進行度に合わせて途中からスタートさせ、実行中であるように振舞う。なおこの画面は新規ウィンドウで開かれる。実行中または実行完了したシナリオに関しては、1 シナリオにつき 1 ウィンドウのみ開くものとする。連打などで 2 ウィンドウ目のアクセスがあってもそれを無視する。シナリオロードや新規シナリオ作成については、この制限は設けない。

6.5. View Result

DS-Bench



ID	Date	Machines	Comment	Benchmark Program	Anomaly Load	Tags	Delete
41	2013/04/23 15:23:14	sample_test2		Httpperf		httpperf add	<input type="checkbox"/>
40	2013/04/19 14:19:02	sample_test2		Httpperf		httpperf add	<input type="checkbox"/>
38	2013/04/19 10:50:34	sample_test2	Httpperf	Httpperf		add	<input type="checkbox"/>
12	2013/04/17 09:54:04	sample_test2		Hackbench Httpperf		add	<input type="checkbox"/>
11	2013/04/17 09:51:43	sample_test2		Hackbench		add	<input type="checkbox"/>
10	2013/04/17 09:50:52	sample_test2		Hackbench		add	<input type="checkbox"/>
9	2013/04/17 09:48:49	sample_test2		Hackbench		add	<input type="checkbox"/>

<< 1 2 >>

図 6-16 Synopsis 画面

過去に実施したベンチマーク結果の履歴一覧を表示する。

- ID : 結果 ID (View Result 画面に現れる)
- Data : シナリオ開始日時
- Machines : ターゲットマシン名
- Comment : シナリオの説明・コメント
- Benchmark Program/Anomaly Load : 使用したベンチマーク/アノマリー/Fault injection
- Tags : 設定したタグ表示、「add」ボタンでタグ追加

また、画面上部の検索フォームから指定したキーワードでの検索が可能。

各カラムの赤いアイコンをクリックすると表示されている一覧がソートされる。

Tags の「add」ボタンをクリックすると [6.3.7 タグの追加](#) Edit Tags ダイアログを表示しタグの編集を行える。

右端の「Delete」カラムにチェックを付け、「Delete」ボタンをクリックすると、そのベンチマーク結果が削除される。

「Check all」ボタンをクリックすると全てのベンチマーク結果の「Delete」カラムにチェックが入り、「Clear all」ボタンをクリックすると全てのベンチマーク結果の「Delete」カラムのチェックが外れる。

6.5.1. View Result Detail

Day : 2011/12/12 15:21:41

Time Scale : 0 : 1 : 0 Total : 000:10:00 Auto Adjust

trg03

Name	ResultName	Description
		<p>Reduction (Hackbench / Time)</p> <p><input checked="" type="radio"/> max <input type="radio"/> min <input type="radio"/> sum <input type="radio"/> prod <input type="radio"/> avg <input type="radio"/> stddev</p> <p>max 0.093</p>
		<p>Hackbench</p> <p><input checked="" type="radio"/> Table <input type="radio"/> Bar <input type="radio"/> Line <input type="radio"/> Log</p> <p>Hackbench on trg03</p> <p>Time 0.093</p>
		Current Tags:

図 6-17 View Result Detail

ベンチマーク実行結果を表示する画面で、View Result 画面から結果を表示させたいシナリオの ID をクリック、または D-Case Editor から表示で遷移する。表示方法は D-Case Editor から実行後、Evidence ノードを右クリック > Attachment > Open をクリックする。

シナリオの詳細とベンチマーク結果を表示する。

「ReRun」ボタンをクリックすると、Benchmark 画面に遷移し、実行したベンチマークと同じ構成がロードされ、編集・実行することができます。

「Delete」ボタンをクリックすると、表示されているベンチマーク結果を削除し、View Result 画面に遷移する。

ただし D-Case Editor から表示した場合は再実行する「ReRun」ボタンと削除を行う「Delete」ボタンは使えない（表示しない）。

6.6. Configuration 画面

DS-Bench

JST-CREST Dependable Embedded Operating Systems for Practical Use

➤ Main ➤ Synopsis ➤ View Result ➤ **Configuration**

Benchmark Management

- **Benchmark**
- Setting
- Target list
- Sedna
- D-Cloud
- Benchmark scenario

No	Benchmark Name	Description	Type	Edit
1	Bonnie	Bonnie++(IO benchmark program)	unit	Edit
2	cpustress	CPU load tool	unit	Edit
3	Hackbench	Hackbench(Load test)	unit	Edit
4	httperf	httperf	unit	Edit
5	IMB	Intel MPI Benchmarks	unit	Edit
6	lperf	lperf(Network traffic benchmarks)	link	Edit
7	LMBench	LMBench(Benchmarks suit)	unit	Edit
8	Memstress	Memory load tool	unit	Edit

図 6-18 Configuration 画面

Configuration 画面は左側にサブメニューを表示する。

「Benchmark」 Benchmark Management 画面を表示（Configuration 画面の初期表示）

「Setting」 DS-Bench のタイムスケールの設定画面を表示

「Target list」 Target list 画面を表示

「Sedna」 Sedna 設定画面を表示

「D-Cloud」 D-Cloud 設定画面を表示

「Benchmark scenario」 ベンチマークシナリオ一覧画面を表示

6.6.1. Benchmark

図 6-18 のように Benchmark Management としてベンチマークプログラムを一覧表示する。

「Edit」アンカーをクリックした場合そのベンチマークの Benchmark Edit ダイアログを表示する。

Benchmark Edit

Benchmark Name : Bonnie

Description : Bonnie++(IO benchmark program)

Path : /var/dsbench/target/benchmarks/bonnie++-1.96/sbin/bonnie++

Type : Unit test

Start time : 0 - Finish time : 20000

Interval Before Starting : 50

Interval After Termination : 1000

Client Input :

Label : number of group -Arg : -n -Ui : yes -Interactive : no -Value : 2

Label : user -Arg : -u -Ui : no -Interactive : no -Value : dsb

add

Source log :

Output style :

```
<table>
<rheader>Machine,Size,K/sec,%CP,K/sec,%CP,K/sec,%CP,K/sec,%CP,K/sec
</rheader>
<data>(.\b) +(d+.\d+)\d+\|++ +(d+.\d+)\d+\|++ +(d+.\d+)\d+\|++ +
\d+\|d+\|++ +(d+.\d+)\d+\|++ +(d+.\d+)\d+\|++ +(d+.\d+)\d+\|++ +(d+.\d+)\d+\|++
</data>
<begin>Version 1.96 -----Sequential Output----- --Sequential Input-
--Random-</begin>
<end>Version 1.96 -----Sequential Create----- -----Random
Create-----</end>
```

Result :

Graph style :

```
<title>Bonnie benchmark</title>
<ylabel>%CP</ylabel>
<imagesize>0.5,0.5</imagesize>
```

Modify Test Delete Cancel

図 6-19 Benchmark Edit ダイアログ

各種ベンチマークプログラムの情報を編集します。ここで行った編集内容はベンチマークの設定ファイル conf ファイルに反映される。

「Benchmark Name」に編集を行っているベンチマーク名を表示する。

「Description」編集中のベンチマークの説明を表示する。

「Path」編集のベンチマークの展開先 Path を表示する。

「Client Input」ベンチマークのパラメータの追加、デフォルト値の設定を行える。「add」ボタンで項目追加、「Label」に表示名、「-Arg」にオプション、「-Ui」表示、非表示を Yes|No で選択し、「Value」にデフォルト値を入力する。

「Source log」にベンチマークの出力するログの例をコピー&ペーストし、「Output style」にログ切り出しのルールを記載した状態で「Test」ボタンをクリックすると「Source log」を「Output style」で処理した結果を「Result」に出力する。

「Delete」ボタンをクリックすることで Edit アンカーをクリックした Benchmark を削除する。

「Modify」ボタンクリック時に特にエラーとなる要素がなければ Benchmark Management へ遷移

6.6.2. Setting



図 6-20 Setting

「Setting」をクリックすることでベンチマークの時間幅初期値を設定する画面を表示する。

「Time Scale」は Benchmark 画面を初期表示した際の Time Scale の初期値として利用される。

左から時:分:秒

「Register」ボタンをクリックすることで入力値を設定ファイルに記憶する。

「Reset」ボタンをクリックすることで入力値を変更前の状態に戻す。

「Back」ボタンをクリックすることで入力値を破棄して前の画面へ遷移する。

6.6.3. Target list

No	Machine name	Type	Status
1	dsb-target1	Virtual	Active
2	dsb-target2	Virtual	Power off
3	dsb-target3	Virtual	In use(60min)
4	dsb-physical1	Physical	Active

Refresh

図 6-21 Target list

「Target list」をクリックすることで D-Cloud コントローラが管理するターゲットの一覧と、それぞれの状態を一覧表示する。Status が In Use（使用中）の場合、マシン使用者が解放する時間を表示する。

「Refresh」ボタンを押すことで再度計算資源の取得を行い、再表示する。

「Machine name」カラムをクリックすることでターゲットの詳細（Target detail）を表示する。

6.6.3.1. Target detail

cpu			
arch	fuga		
memory()			
devices			
device(ethernet)			
eth0			
device(storage)			
/dev/hda			2T
os			
family	Linux		
name	Ubuntu		
version	11.10		
softwares			
software(kernel)			
version	3.0.0		
software(package)			
family	DPKG		

Close

図 6-22 Configuration 画面

<dsb_resource_desc>タグ内に記載したターゲット情報を表で表示する。

「Close」 ボタンで Target list に遷移する。

6.6.4. Sedna

Sedna	
Sedna host	127.0.0.1
Sedna DB name	dsbench
Sedna DB user	SYSTEM
Sedna DB password	*****
Result benchmark	result_benchmark
Result benchmark scenario	result_benchmark_set

Register Reset Back

図 6-23 Sedna

「Sedna」をクリックすることで Sedna XML DB へ接続するための環境設定用入力画面を表示する。

「Sedna host」は接続する Sedna XML Database の IP アドレスまたはホスト名を指定する。

「Sedna DB name」は接続する Sedna XML Database の接続データベース名を指定する。

「Sedna DB user」は接続する Sedna XML Database の接続ユーザ名を指定する。

「Sedna DB password」は接続する Sedna XML Database の接続ユーザのパスワードを指定する

「Result benchmark」はベンチマーク結果の格納先を指定する。

「Result benchmark scenario」はベンチマークシナリオの格納先を設定する。

「Register」 ボタンをクリックすることで入力値を設定ファイルに記憶する。

「Reset」 ボタンをクリックすることで入力値を変更前の状態に戻す。

「Back」 ボタンをクリックすることで入力値を破棄して前の画面へ遷移する。

6.6.5. D-Cloud

D-Cloud	
D-Cloud controller	192.168.11.133
D-Cloud port	9903

Register Reset Back

図 6-24 D-Cloud

「D-Cloud」をクリックすることで D-Cloud コントローラへ接続するための環境設定用入力画面を表示する。

「D-Cloud controller」は接続する D-Cloud コントローラの IP アドレスまたはホスト名を指定する。

「D-Cloud port」は接続する D-Cloud コントローラへの接続ポートを指定する。

「Register」 ボタンをクリックすることで入力値を設定ファイルに記憶する。

「Reset」 ボタンをクリックすることで入力値を変更前の状態に戻す。

「Back」 ボタンをクリックすることで入力値を破棄して前の画面へ遷移する。

6.6.6. Benchmark scenario

No	Benchmark scenario name	Description	
1	test1	test1	Delete
2	test	test	Delete

図 6-25 Benchmark scenario Management

本画面は利用者が作成、登録したベンチマークシナリオ編集、一覧表示及びその削除を行うことが出来る。

「Delete」アンカーをクリックで該当するベンチマークシナリオの削除をする確認ダイアログを表示する。

ダイアログの「OK」 ボタンクリックで該当のベンチマークシナリオを本システムから削除することが出来る。