

# DS-Bench/Test-Env (D-Cloud) Specification

Version E1.01

2014/02/01

Edited by DEOS R&D Center



DEOS Project

JST-CREST

Research Area

"Dependable Operating Systems for Embedded Systems Aiming at Practical Applications"



Japan Science and Technology Agency

## Contents

---

1. Introduction .....	5
1.1. Overview of functions .....	5
1.1.1. DS-Bench/D-Cloud.....	5
1.1.2. Cooperation with the D-Case Editor / D-Case Weaver .....	5
1.2. System configuration .....	6
1.2.1. DS-Bench controller .....	6
1.2.2. DS-Bench target.....	6
1.2.3. D-Cloud controller.....	6
2. Implementation.....	7
2.1. System overview.....	7
2.1.1. DS-Bench.....	7
2.1.2. D-Cloud .....	8
2.1.3. D-Case Editor/D-Case Weaver cooperation .....	9
2.1.4. Scenario operation from the command line .....	12
2.2. Packaging.....	13
2.2.1. DS-Bench target and DS-Bench controller.....	13
2.2.2. D-Cloud controller.....	14
2.2.3. Sedna DB server .....	14
3. Memory Specification .....	15
4. File specification.....	16
4.1. The conf file of each benchmark .....	16
4.2. XML format to be stored in the Sedna XML DB .....	20
4.2.1. Benchmark scenario XML.....	20
4.2.2. Benchmark Result XML.....	23
4.2.3. Anomaly result XML.....	26
4.3. run_status_*.....	29
5. Communication specification .....	30
5.1. Communication message .....	33
5.1.1. Benchmark installation request.....	33
5.1.2. Benchmark installation response .....	33
5.1.3. Benchmark uninstallation request.....	33
5.1.4. Benchmark uninstallation response .....	34
5.1.5. Computational resource list request.....	34
5.1.6. Computational resource list response .....	35
5.1.7. Computational resource allocation request.....	35
5.1.8. Computational resource allocation being processed.....	36

5.1.9.	Computational resources preparation completion.....	36
5.1.10.	Computational resource preparation impossible.....	37
5.1.11.	Benchmark scenario transmission.....	37
5.1.12.	Benchmark scenario reception completion.....	38
5.1.13.	Benchmark scenario refusal response.....	38
5.1.14.	Benchmark start request.....	39
5.1.15.	Benchmark start response (ACK).....	39
5.1.16.	Benchmark start response (NAK).....	40
5.1.17.	Benchmark acquisition state request.....	40
5.1.18.	Benchmark acquisition state response.....	41
5.1.19.	Fault list request.....	41
5.1.20.	Daemon alive confirmation request.....	41
5.1.21.	Daemon alive confirmation response.....	42
5.1.22.	Fault list response (ACK).....	42
5.1.23.	Fault list response (NAK).....	43
5.1.24.	Benchmark start completion notice.....	43
5.1.25.	Benchmark execution completion notice.....	44
5.1.26.	Computational resource release request.....	44
5.1.27.	Computational resource release response (ACK).....	44
5.1.28.	Computational resource release response (NAK).....	45
5.2.	Request XML.....	46
5.2.1.	Computational resource list acquisition request XML.....	46
5.2.2.	Computational resource allocation request XML.....	46
5.2.3.	Benchmark scenario XML.....	54
5.2.4.	Benchmark start request XML.....	56
5.2.5.	Daemon alive confirmation request XML.....	56
5.2.6.	Calculating resource release request XML.....	56
5.3.	Response XML.....	57
5.3.1.	Computational resource list response XML.....	57
5.3.2.	Computing resources allocation response XML.....	58
5.3.3.	Benchmark scenario response XML.....	58
5.3.4.	Benchmark state XML.....	59
5.3.5.	Daemon existence confirmation response XML.....	61
5.3.6.	Fault list XML.....	62
5.4.	Communication with the D-Case Editor/D-Case Weaver.....	62
5.4.1.	Scenario execution start.....	62
5.4.2.	Scenario Result View.....	62
5.4.3.	Scenario stop.....	63

6.	Screen specification .....	64
6.1.	Common menu.....	64
6.2.	Main screen.....	64
6.3.	Benchmark screen.....	65
6.3.1.	Dialog to add targets .....	66
6.3.2.	Target Editing dialog .....	66
6.3.3.	Target list.....	67
6.3.4.	How to operate on the time scale.....	68
6.3.5.	Detailed benchmark settings dialog.....	69
6.3.6.	Use of Fault injection.....	71
6.3.7.	Add tags .....	72
6.3.8.	Execution of the scenario and its Result.....	72
6.3.9.	Cooperation with the D-Case.....	73
6.4.	Synopsis screen.....	76
6.5.	View Result .....	78
6.5.1.	View Result Details .....	79
6.6.	Configuration screen.....	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

## Revision History

---

Version	Change Description	Author	Date
1.01	* Corrected erroneous descriptions. * Added missing descriptions.	Dependable Embedded OS R&D Center	2014.Feb.1

# 1. Introduction

This document is a specification of the system using the Test-Env and DS-Bench, to perform a combination of benchmarks / anomaly load tests, and evaluate the test results.

DS-Bench is a measurement tool of dependability. The Test-Env (D-Cloud) is a tool to test the system quickly by managing the resources.

**※In subsequent sections "Test-Env (D-Cloud)" is simply referred to as "D-Cloud".**

**Each system name, product name, and service name in this document is the trade-mark or registered trade-mark of the company that created the respective system, product, or service.**

## 1.1. Overview of functions

### 1.1.1. DS-Bench/D-Cloud

The configuration of this system is as follows.

- One DS-Bench controller.
- More than one verification terminal DS-Bench target.
- One D-Cloud controller.
- Network-controllable power source taps .
- Network-controllable network switch.

Create a scenario that combines the benchmark / anomaly load. Run the benchmark / anomaly load according to your scenario on the target. Collect and calculate the results, and store these in the database. Through WebUI, you can easily create and confirm a scenario, check the progress while it is running, view the results, and determine the status of the scenario.

Also, if execution of a benchmark did not proceed as expected, an accurate error message is displayed to the user.

### 1.1.2. Cooperation with the D-Case Editor / D-Case Weaver

Get an existing scenario that was saved in the DS-Bench from D-Case Editor / D-Case Weaver. You set the expected values and the parameters for the benchmark scenario. You can carry out the scenario through D-Case Editor / D-Case Weaver. The results of executing the scenario are collected and calculated, and then passed to D-Case Editor / D-Case Weaver for evaluation.

The process of execution of a benchmark scenario can be observed through D-Case Editor/D-Case Weaver.

D-Case Editor / D-Case Weaver can stop a scenario that is running.

## 1.2. System configuration

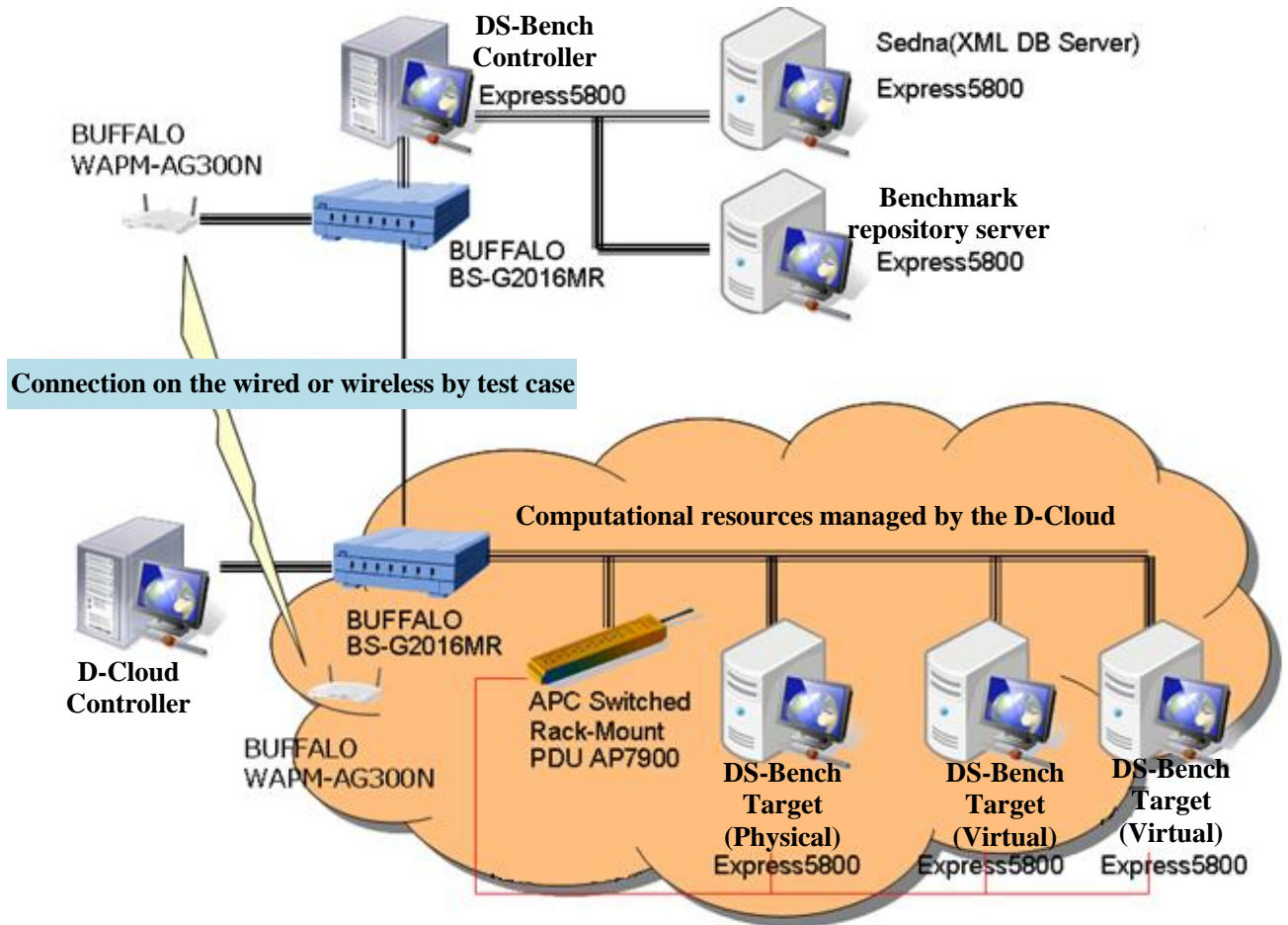


Fig. 1-1 System configuration diagram

### 1.2.1. DS-Bench controller

A benchmark tool and scenario data are stored. An order to execute a scenario is sent to a target, and the result can be acquired and saved. In addition, the controller has WebUI, whose functions can be intuitively grasped and which can be easily operated. Sedna is used as a database.

### 1.2.2. DS-Bench target

Receives a request from the DS-Bench controller, runs the test, generates a problem, and gets the results. The acquired results are calculated, organized, and returned. When using the DS-Bench target virtual machine, it is necessary to set the OpenStack in D-Cloud.

### 1.2.3. D-Cloud controller

In order to manage information in a unified manner, the computational resource's reservation, startup, information acquisition, etc. are performed via D-Cloud. Moreover, fault injection is managed and performed.

## 2. Implementation

### 2.1. System overview

#### 2.1.1. DS-Bench

There are CGI programs which run on Apache in a DS-Bench controller. A Web page is provided using Ajax technology. Most processing on the Web is done through the CGI, and is saved in the Sedna XML DB or files on the controller.

A daemon program called dsbd-ctrl is placed on the controller, where it confirms the list of computational resources for D-Cloud and secures computational resources. The command to install/uninstall a benchmark is sent to the secured target machine. Also, it receives a response from the target machine, and writes the results into Sedna XML DB or a file .

Daemon programs named "dsbd-target" are disposed in all targets. This program sends confirmation whether daemon exists in response to inquiries from the D-Cloud. A daemon installs, uninstalls, starts up, and ends a benchmark program on a TCP port based on the information received from dsbd-ctrl on the controller.

The results are returned to dsbd-ctrl controller in XML format after the benchmark program was executed.

##### 2.1.1.1. Benchmark scenario extension

This is a function giving names to the parameters needed to run the benchmark, to make them easy to recognize.

In case of collaboration with D-Case Editor/D-Case Weaver, a name can be given to a result passed to D-Case Editor /D-Case Weaver. That result can then be searched under that name specified by D-Case Editor/D-case Weaver.

The name of each parameter is set through the DS-Bench GUI.

##### 2.1.1.2. Reduction function

This is a function calculating or collecting the results from the same multiple benchmarks. The types of calculation operations are: MAX (maximum value), MIN (minimum value), SUM (total sum), PROD (total product), AVE (average), STDDEV (standard deviation).

##### 2.1.1.3. User Interface

The target machine or the network interface can be configured by the selection from the drop-down list.

The drop-down list is displayed by specifying attributes that are required to set up files for each benchmark. The drop-down list is created based on the list of computational resources supplied in response to a request.

### 2.1.2. D-Cloud

A daemon program called ds-ctrl is placed on the D-Cloud. "ds-ctrl" daemon sends requests for demon existence confirmation to all targets in response to a resource request from DS-Bench controller. The results are made into a target list and returned to a DS-Bench controller as a computational resource list response. The XML format of the response follows the XML format of the daemon existence confirmation request.

DS-Bench sends a computational resource release request to the D-Cloud after the completion of benchmark uninstallation. D-Cloud performs the computational resource release and transmits a computational resource release response (ACK/NAK) to DS-Bench.

#### 2.1.2.1. Target virtual machine

A virtual machine on DS-Bench is booted by OpenStack. The IP address of the virtual machine is unknown until OpenStack determines it upon the booting of the virtual machine.

Thus, the IP address is in the response XML, and is used when communicating with the target machine.

#### 2.1.2.2. Fault list function

DS-Bench sends a fault list request to D-Cloud. D-Cloud observes the target, gets the fault information, and sends a fault list response (ACK). If there is an abnormality, D-Cloud sends fault list response (NAK).

The contents of the response XML are in accordance with the fault list XML.

Fault injection is set on the timeline of each target machine on the GUI when a scenario is set.

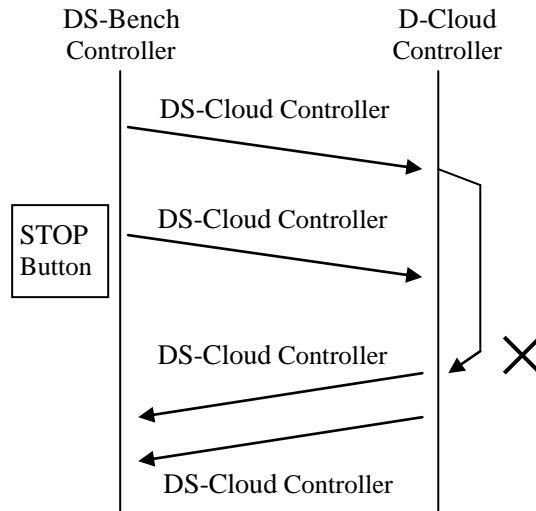
#### 2.1.2.3. Scenario stop function

In order to stop the scenario, it is necessary to stop allocation of computational resources. The computational resource release request is used as the sequence of communication with D-Cloud, and behavior of D-Cloud controller is modified. This process sets a flag signifying "stop allocation" and does not release the resources when a request is received for release of resources. When the flag is recognized by the allocation process, allocation is interrupted and computational resources are released. Interruption is completed as the failure to allocate the computational resources is recognized by the DS-Bench.

This method is used when the allocation stop request is sent by the D-Case Editor/D-Case Weaver.



Computational resource secured suspension process image



2.1.3. D-Case Editor/D-Case Weaver cooperation

2.1.3.1. Benchmark scenario list acquisition

D-Case Editor/D-Case Weaver runs the cgi to acquire the list of the benchmark scenarios. DS-Bench returns a list of the benchmark scenarios that have been designated based on certain parameters. Refer to the "Interface Specification between DS-Bench and D-Case Editor/D-Case Weaver for their Collaboration" for the format of the response XML and request parameters.

2.1.3.2. Benchmark scenario detail acquisition

D-Case Editor/D-Case Weaver runs the cgi to acquire the list of the benchmark scenarios. DS-Bench returns the details of the benchmark scenarios that have been designated based on certain parameters. Refer to the "Interface Specification between DS-Bench and D-Case Editor/D-Case Weaver for their Collaboration" for the format of the corresponding parameters, request parameters, and response parameters.

2.1.3.3. Benchmark scenario execution function

D-Case Editor/D-Case Weaver runs the cgi to request to start the benchmark. The requested processing is performed like the execution through a DS-Bench scenario editing display, securing computational resources for the specified scenario, installing the benchmark, running the benchmark, uninstalling the benchmark, and releasing computational resources.

The result of the processing that was received is returned to D-Case Editor/D-Case Weaver after the benchmark is completed.

Refer to the "Interface Specification between DS-Bench and D-Case Editor/D-Case Weaver for their Collaboration" for the format of the response XML and request parameters.

#### 2.1.3.4. Confirming benchmark execution status and results

D-Case Editor/D-Case Weaver runs the cgi to acquire the benchmark execution status and results. DS-Bench confirms the execution status of benchmark given scenario and returns the results.

Refer to the "Interface Specification between DS-Bench and D-Case Editor/D-Case Weaver for their Collaboration" for the format of the response XML and request parameters.

#### 2.1.3.5. Notification status

In order to inform the D-Case Editor/D-Case Weaver of the execution status of the benchmark scenario, codes for messages that indicate the nature of an error and status terms are provided. It is possible to report on the execution status of the scenario in more detail by providing a message code and a status term.

Error code / message table

code	message	Detail
0	success	Successful completion
1*		Error with regard to parser
11	D-Case Editor Error: request XML parser error	Failed to parse the XML request
12	DS-Bench Error: scenario XML parser error	Failed to parse the XML scenario
13	DS-Bench Error: benchmark XML parser error	Failed to parse the XML benchmark
14	DS-Bench Error: fault XML parser error	Failed to parse Fault XML
15	DS-Bench Error: result XML parser error	Failed to parse the XML result
2*		Error with regard to permission
21	DS-Bench Error: run_list file permission denied.	Error in opening run_list file
22	DS-Bench Error: run_status file permission denied.	Error in opening run_status file
23	DS-Bench Error: XML file permission denied.	Error in opening XML
3*		Error with regard to computational resources and exclusive control
31		Item with that number is missing
32	DS-Bench Error: the target is in use.	Target is in use

4*		Error regarding communication
41	DS-Bench Error: install failed.	Benchmark Installation Failed
42	DS-Bench Error: create invalid socket.	Socket creation failure
43	DS-Bench Error: list of calculation resources is unacquirable.	A failure to obtain computational resource list
44	DS-Bench Error: failure to reserve calculation resources.	Failure to obtain computational resources
45	DS-Bench Error: failure to release calculation resource.	Computational resource could not be released
5*		IO error
51	D-Case Editor Error: scenario do not exist.	The scenario that has been specified does not exist
52	D-Case Editor Error: parameter do not exist.	The specified parameter does not exist
53		Item with that number is missing
54	DS-Bench Error: scenario/benchmarks/ directory does not open.	Directory of scenarios / benchmarks does not exist
55	DS-Bench Error: packages.list do not exists.	List of packages does not exist
9*		Other
91	DS-Bench Error: Not be able to create JSON data.	JSON data creation failure
92	DS-Bench Error: failure to reduction.	Reduction failure
93	DS-Bench Error: failure to stop benchmark run.	Benchmark run stop failure
94	the scenario is not running.	The scenario is not running
95	D-Case Editor Error: Scenario has not executed from D-Case.	The specified scenario has not been run from D-Case

Scenario execution status table

status	Detail
error	Any error
ready	Normal acceptance of the scenario is complete

reserve	During securing of Calculations resources
install	During installation of Benchmark
run	Scenario running
reduction	Result aggregate / During store
finish	Scenario completion
stopping	Scenario aborted
stopped	Scenario complete stop
st_error	Error of scenario aborted

#### 2.1.3.6. Association with the DS-Bench ID Display

A Benchmark scenario which is run by the D-Case Editor/D-Case Weaver is assigned a unique ID. A result ID is added to a response to D-Case Editor at the time of the scenario execution request. The result ID is received, and a result ID which contains the response to the scenario execution request is used to access the result screen of the GUI. For "Communication specifications", see Sections 5.4.1 and 5.4.2. See Section 6.5.1 for the specification screen.

#### 2.1.3.7. Benchmark scenario abort function

A CGI is provided for stopping the benchmark scenario through D-Case Editor/D-Case Weaver. See "Communication specifications" for the "abort" scenario.

### 2.1.4. Scenario operation from the command line

#### 2.1.4.1. Run scenario from the command line

The value of a parameter may be rewritten for running a benchmark scenario from the command line.

The format of rewritable parameters is as follows

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

Substitute an explanation of the parameter for "description", the parameter name for "name", and the default value for "value". By using this format in the GUI of DS-Bench, you can change the parameters when running from the command line.

The command for execution is **dsbcui** and is executed as follows:

```
$ /var/dsbench/controller/bin/dsbcui {-r|--run} Δ scenario name Δ -name=value Δ -name=value Δ ...
```

**dsbcui** does not check mismatches among parameter names in the scenario and parameter names of command arguments. The execution module is responsible for checking the mismatch.

This command basically only accepts a request for starting a scenario. It shows an error message when there has been a failure to start the scenario up to the time of the request. Once the start request is accepted, it displays the result ID on the results page of the GUI that you can easily browse. The result is inspected

in the GUI or the DB. Refer to Table 2-4 of the command specification.

#### 2.1.4.2. Abort scenario from the command line

A scenario can be aborted from the command line just as a scenario can be started from the command line as explained in section 2.1.4.1.

A scenario can be aborted as follows:

```
$ /var/dsbench/bin/dsbrcui {-s|--stop} Δ scenario ID
```

This scenario ID is one obtained when the scenario was started.

This command basically only accepts a request to abort a scenario. The dsbrcui command shows an error message when there has been a failure. The command shows a message when the scenario was aborted successfully. Table 2-4 shows the error codes and their meanings.

dsbrcui return status table

Return value	Output	Conditions
0	Succeeded running scenario in scenario_id=***	Completion of scenario execution receptionist (run)
0	Succeeded stopping scenario with scenario_id=***	Completion of scenario stop receptionist (stop)
1	(usage: Outputs TBD)	No arguments
2	(usage: Outputs TBD)	Scenario name does not exist (run) Scenario ID does not exist (stop)
11~99	Message that depends on the execution module	Failed to accept scenario
0	(usage: Outputs TBD)	-h --help option

## 2.2. Packaging

The package is divided into D-Cloud controller / DS-Bench controller / DS-Bench target / Sedna DB server. Each is released as a Ubuntu deb package.

### 2.2.1. DS-Bench target and DS-Bench controller

Enter the following values during installation. The default values are shown in parentheses.

- Port of DS-Bench controller daemon ("9803")
- Port of DS-Bench target daemon ("9804")

For DS-Bench controller, enter the following additional values:

- The host name or IP address of the D-Cloud controller node ("127.0.0.1")
- Port of D-Cloud controller daemon ("9903")
- The host name or IP address of the Sedna DB server ("127.0.0.1")
- DB name that you created in Sedna ("dsbench")
- Name of collection for storing benchmarks that you created in Sedna ("result\_benchmark")
- Name of collection for storing scenarios that you created in Sedna ("result\_benchmark\_set")

In response to these input values, the configuration file is updated during installation.

### 2.2.2. D-Cloud controller

D-Cloud controller is packaged in two ways, depending upon whether or not Openstack is used. A package which does not use Openstack cannot handle virtual machines.

Enter the following values during installation. Default values are shown in parentheses.

- Port of DS-Bench target daemon ("9804")
- Port of D-Cloud controller daemon ("9903")
- Port of D-Cloud resource manager daemon ("9904")
- Port of map server daemon ("7000")

Enter the following additional value if you use Openstack.

- The path of the novarc ("/var/dcloud/openstack/creds/novarc")

Immediately after the daemons of a D-Cloud controller, D-Cloud resources manager and the map server are installed, they are ready to use. In addition, all these daemons are started automatically when the server is started.

### 2.2.3. Sedna DB server

An installer shell provided at "<http://www.sedna.org/>" wraps Sedna into a package so that it may be installed easily. Enter the following values after Sedna is installed. Default values are shown in parentheses.

- DB name to be used in the DS-Bench ("dsbench")
- Name of collection for storing benchmarks to be used in the DS-Bench ("result\_benchmark")
- Name of collection for storing scenarios to be used in the DS-Bench ("result\_benchmark\_set")

The DB used by DS-Bench and Sedna is started automatically when it is next started.

### 3. Memory Specification

The shared memory is not used as an inter-module interface.

## 4. File specification

### 4.1. The conf file of each benchmark

The "conf" file is a file prepared for each benchmark, and describes a setup for the benchmark.

The file is stored in the "conf/" folder, and can be edited from the "Benchmark Management" of the management screen of the DS-Bench.

Details of the configuration file are as follows:

```
<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>
```

The following describes information on each node type, including whether the node is mandatory or omissible, as well as whether the node may be repeated. The attribute value is filled in with necessary information, or left as blank as indicated.

**<dsbxml> </dsbxml>**



Root Node

Repeatable : No; Omittable: No

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

A benchmark is specified by this node.

- The "seq" attribute specifies the execution sequence seen from the DS-Bench controller.
- The "title" attribute may indicate a name if convenient, and may be left blank.
- The "path" attribute is the full path to the benchmark.
- The "type" attribute specifies the type of a benchmark, which is either a single benchmark [unit], or a C/S type benchmark [link].

Repeatable: Yes; Omittable: No

**<server\_input host="[IP]"></server>**

This node describes the host name or IP address of a server machine used as the "host" object, and is used for setting the server of the C / S type benchmark. It is significant only when "type" attribute of the parent "benchmark" node is "link".

Repeatable: No; Omittable: Yes (Mandatory in the case of type = link)

**<client\_input host="[IP]"></client>**

The node describes the host name or IP address of a machine used as the client of a single or a C/S type benchmark.

Repeatable: Yes; Omittable: No

**<stime interval="">Start time</stime>**

This node specifies the execution start time of a benchmark relative to the execution start time of the benchmark scenario, in seconds. The "interval" attribute specifies the interval before starting in milliseconds.

**<ftime interval="">End time</ftime>**

This node specifies the execution finish time of a benchmark relative to the execution start time of the benchmark scenario, in seconds. The "interval" parameter specifies the waiting interval before finishing in milliseconds.

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

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

This node specifies parameters which are passed to benchmarks as arguments etc.

Repeatable: Yes; Omittable: Yes

- "input"
  - text : the value may be entered in a text box.

- netif : a network interface may be selected from a drop down list.
- target : a target machine may be selected from a drop down list.

(※The drop down list is created from computational resources obtained from D-Cloud.)

- "label" describes the name of an option
- "arg" specifies an argument to which the hyphen is attached.
- "ui" specifies whether the configuration screen is shown or not when Web is executed.
- Enter "yes" when it is necessary to input communication interaction as the "interactive"command line is executed.

#### **<output/>**

Specify the output format of the result.

If this is omitted, results are not returned.

Repeatable: Yes; Omittable: No

#### **<caption/>**

Specify a regular character string or a new expression as the title of the result.

Repeatable: No; Omittable: Yes

#### **<table/>**

Settings for result output are made at this node.

Describe as many nodes as the number of tables that you want to retrieve.

Repeatable: Yes; Omittable: No

#### **<header>[str]</header>**

Specify the name of the part of the table whose value you want to extract.

Omittable: Yes

#### **<rheader>[str]</rheader>**

Separate out the name of the value you want to extract with commas (,).

Omittable: Yes

#### **<data>[str]...</data>**

Please enclose, in parentheses, the value retrieved from the results.

Note: Because strings between parentheses are extracted with regular expressions in the script, please do not use nested parentheses, (()).

Repeatable: No; Omittable: No

#### **<begin>[str]</begin>**

This node specifies the beginning portion of the table you want to extract.

The character string specified here is included in a search of results.

Omittable: Yes

**<end>[str]</end>**

This node specifies the end portion of the table you want to extract.

The character string specified here is included in a search of results.

Omittable: Yes

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

If the value zero (0) is specified, the name of the value is masked.

If entry of a value is omitted, the value one (1) is assigned automatically.

Omittable: Yes

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

When the value zero (0) is specified, the value is masked.

If the entry of a value is omitted, one (1) is assigned automatically.

Omittable: Yes

**<graph > Graph drawing method</graph>**

This node specifies parameters in the benchmark scenario such as extraction values and scales of the vertical axis and horizontal axis.

Based on the parameters, a program determines a label, a value, etc..

Parameters to be added are as follows.

Parameters	Content
title	Heading of the graph
xlabel	X-axis labels
ylabel	Y-axis label
image size	Image size
xrange	Scale of the graph
xdata	X-axis value
ydata	Y-axis value

## 4.2. XML format to be stored in the Sedna XML DB

The anomalies and benchmarks are stored in three XML formats, which are described in detail below.

### 4.2.1. Benchmark scenario XML

Whenever a benchmark scenario is performed, its results are stored in Sedna XML DB.

The primary information contained in an XML benchmark scenario is as follows.

- ID
- The date the benchmark was performed
- Target host name, hardware environment, software environment
- Comments
- ID for accessing benchmark result XML or anomaly result XML

- Specification of benchmark scenario XML

```

<dsbxml>
  <id>ID</id>
  <display-id> Result screen display number </display-id>
  <data> Benchmark start time </data>
  <display-time>Result screen display number </display-time>
  <controller>
    <host> Controller host name </host>
  </controller>
  <target>
    <host>Controller host name<env><cpu>x86_64</cpu><memory type='DDR3'><speed
type='mhz'>800</speed><size>2048</size>.....</env></host>
      .
      .
  </target>
  <benchmark-list>
    <model id=" ID of the benchmark result XML">Benchmark name </model>
      .
      .
  </benchmark-list>
  <anomaly-list>
    <model id=" ID of the Anomaly results XML">Anomaly program name </model>
      .
      .
  </anomaly-list>
  <result-viewer-list>
    <result-viewer>
      <result-name> Name that defines the benchmark results </result-name>
      <result-trgt> Benchmark results name </result-trgt>
      <result-desc> Comments on the benchmark results </result-desc>
    </result-viewer>
    <result-viewer>
      .
      .
    </result-viewer>
  </result-viewer-list>
  <reduction>
    <benchmark name="Benchmark name">
      <label name=" Key name of the results">
        <max target="Target name"> Maximum value </max>
        <min target="Target name"> Minimum value </min>
        <sum>Summation </sum>
        .
        .
      </label>
      .
      .
    </benchmark>
    .
    .
  </reduction>
  <description>Comments on the benchmark set</description>
</dsbxml>

```

The meanings of each tag are as follows:

**<id>ID</id>**

**<id>12456</id>** (The number of digits in a number is just an example, and changes with each actual number.)

result\_id indicates a unique number

**<display-id>Result screen display number</display-id>**

This node sets the ID number to be displayed in the leftmost column of the View Result screen. Please set the number zero for the first data registered, and increase by one as more data are entered in the Sedna XML DB. When you enter data, first obtain the highest number by searching the Sedna XML DB, and use that value +1 for the data.

**<data>Benchmark start time</data>**

The node specifies the date and time when you start the execution of the benchmark in the following format:

YYYY/MM/DD hh:mm:ss

**<host>Controller host name</host>**

This node sets the controller host name. Please enter empty string if the controller is not used.

**<host>Target host name<env>~</env></host>**

Set the target host name. If there is more than one target, enter that number of names continuously.

Describe the information on the environment of the target between <env> and </env>.

Store the environment information under the <target> tag of computational resource list acquisition response XML

**<model id="ID of the benchmark result XML">Benchmark name</model>**

This node sets the benchmark name and ID for accessing the benchmark results XML.

When there is more than one benchmark result (if you run more than one benchmark), add the same number of nodes in succession for all the results. See the following section for the form of ID. When benchmark results do not exist, only a benchmark-list tag is used, and a model tag is not used.

**<model id="ID of the Anomaly results XML">Anomaly program name</model>**

This node sets the anomaly name and ID for accessing the anomaly results XML. When there is more than one anomaly result (if you run more than one anomaly), add this node in succession for all the results.

When anomaly results do not exist, only a benchmark-list tag is used and a model tag is not used.

**<result-name>Name that defines the benchmark results</result-name>**

This node sets an identifier to access the benchmark results from D-Case Editor/D-Case Weaver. The creator of a scenario will set this.

**<result-trgt>Benchmark result name</result-trgt>**

The name of the benchmark results that are accessed from <result-name>. This name is that entered in benchmark configuration file and finally that in the benchmark. XML. If the name is invalid, that is, not set to any benchmark in the scenario, the corresponding <result-name> becomes invalid.

**<result-desc>Comments on the benchmark results</result-desc>**

An explanation of the corresponding <result-trgt>

**<reduction><benchmark name="Benchmark name"><label name="Key name of the result"><max target="Target name">Maximum value</max><min>...</benchmark></reduction>**

Saves the reduction results.

**<description>Comments on the benchmark scenario</description>**

Sets the comment that the user has written

#### 4.2.2. Benchmark Result XML

The benchmark result XML is stored in a DB so as to correspond to the benchmarks in the benchmark scenario that has been executed.

- ID
- Benchmark name
- Target name
- Start Time (Start condition)
- End Time
- Benchmark result

- Specification of benchmark results XML

```

<dsbxml>
  <id>ID</id>
  <benchmark> Benchmark name </benchmark>
  <target-list>
    <target> Target host name </target-list>
  <config>
    <stime interval=""> Start time </stime>
    <sstart> Start-pending benchmark ID </sstart>
    <sstop> Wait for end of benchmark ID </sstop>
    <etime interval="">End time</etime>
    <anomaly-load> </anomaly-load>
    <client_option>
      <option label="" arg="Label name" ui="UI flag" interactive="no">Option value
    </option>
      .
      .
      .
    </client_option>
    </server_option>
    <server_input host =" Server program execution host name">
      <stime>0 </stime>
      <ftime>Server program end time</ftime>
      <option label="Label name" arg="Argument flag" ui="UI flag"
interactive="on">Option value </option>
      .
      .
      .
    </server_input>
    </server_option>
  </config >
  <results>
    <data key=" Key of data" header=" Header of the data">Execution result value</data>
    .
    .
    .
  </results>
  <plain>
    Raw output of the benchmark
  </plain>
</dsbxml>

```

The meaning of each tag and how to set them are as follows.

**<id>ID</id>**

Set IDs in which sequential numbers are placed after the ID of the corresponding benchmark scenario XML. These sequential numbers are shared by the anomaly results. For example, if the ID of benchmark scenario XML is result\_id\_0000000001, and two benchmarks exist in the benchmark scenario, the values of the IDs of the XML benchmark results will be as follows.

result\_id\_0000000001\_1

result\_id\_0000000001\_2



**<benchmark>Benchmark name</benchmark>**

Specifies the benchmark name.

**<target>Target host name</target>**

Sets the host name of the target. If there is more than one target, that number of names are written successively.

**<stime interval="">Start time</stime>**

The relative execution start time of a benchmark where the execution start time of the benchmark scenario is 0, expressed in seconds.

The interval before starting is specified as an "interval" parameter, expressed in milliseconds.

**<sstart>Awaiting-start benchmark ID</sstart>**

Sets the ID of a benchmark to be started when the previous benchmark in the sequence starts. Disabled if the ID is "-1".

**<sstop>Awaiting-finish benchmark ID</sstop>**

Sets the ID of a benchmark to be started when the previous benchmark in the sequence finishes. Disabled if the ID is "-1".

**<etime interval="">End time</etime>**

The relative execution start time of a benchmark where the execution start time of the benchmark scenario is 0, expressed in seconds.

The interval before starting is specified as an "interval" parameter, expressed in milliseconds.

**<client\_option>**

A description is given of each option of the program for clients. The content depends on the type of benchmark. The server program is not described..

**<server\_option>**

A description is given of each option of the program for servers. The content depends on the type of benchmark. The client program is not described.

**<results>**

Describes the results of running the benchmark. The content depends on the type of benchmark. If you have more than one result, write a description .for each result.

**<plain>**

It describes the information which an benchmark outputs, without modification. May not exist depending on the type of benchmark.

#### 4.2.3. Anomaly result XML

Anomaly result XML is stored in SednaXML DB, for anomalies in a benchmark scenario that has been executed. The primary information contained in anomaly result XML is as follows.

- ID
- Anomaly program name
- Target host name
- Start time (Start condition)
- End time
- Classification of anomaly
- Anomaly result

- Specification of the anomaly result XML

The format of the anomaly result XML stored in Sedna XML DB is shown below. Save in this format the results of fault injection carried out by the D-Cloud.

```

<dsbxml>
  <id>ID</id>
  <benchmark> Anomaly program name </benchmark>
  <target-list>
    <target> Target host name </target>
  </target-list>
  <config>
    <stime interval=""> Start time </stime>
    <sstart> Start-pending benchmark ID </sstart>
    <sstop> Wait for end of benchmark ID </sstop>
    <etime interval=""> End time </etime>
    <anomaly-load>true</anomaly-load>
    <client_option>
      <option label="Label name" arg="Argument flag" ui="UI flag" interactive="no">Option
value </option>
      .
      .
      .
    </client_option>
    </server_option>
    </server_option>
  </config>
  <results>
    <data key=" Header of the data" header=" Header of the data">Execution result value </data>
    .
    .
    .
  </results>
  <plain>
    Raw output of the benchmark
  </plain>
</dsbxml>

```

The meaning and manner of setting of the tags are as follows.

**<id>ID</id>**

Sets IDs in which a sequential numbers are placed after the ID of the corresponding benchmark scenario XML.. The sequential number is shared by that benchmark result. For example, if the ID of benchmark scenario XML is result\_id\_0000000001, and two benchmarks exist in a benchmark scenario, the values of the IDs of the XML benchmark results will be as follows, respectively.

result\_id\_0000000001\_1

result\_id\_0000000001\_2

**<benchmark>Anomaly program name</benchmark>**

Specifies the benchmark name.

**<target>Target host name</target>**

Sets the host name of the target. Describe in succession all the targets if there is more than one target.

**<stime interval="">Start time</stime>**

The relative execution start time of a benchmark where the execution start time of the benchmark scenario is 0, expressed in seconds.

The interval before starting is specified as an "interval" parameter, expressed in milliseconds.

**<sstart>Awaiting-start benchmark ID</sstart>**

Sets the ID of a benchmark to be started when the previous benchmark in the sequence starts. Disabled if the ID is "-1".

**<sstop>Awaiting-finish benchmark ID</sstop>**

Sets the ID of a benchmark to be started when the previous benchmark in the sequence finishes. Disabled if the ID is "-1".

**<etime interval="">End time</etime>**

The relative execution start time of a benchmark where the execution start time of the benchmark scenario is 0, expressed in seconds.

The interval before starting is specified as an "interval" parameter, expressed in milliseconds.

**<anomaly-load>Classification of Anomaly program</anomaly-load>**

Set true if there is an anomaly, and set false if not.

**<client\_option>**

A description for each option of the program for clients. The content depends on the type of benchmark. The server program is not described.

**<server\_option>**

A description for each option of the program for clients. The content depends on the type of benchmark. The client program is not described.

**<results>**

Describe the results of running the anomaly. The content depends on the anomaly. Write a description for each result.

**<plain>**

It describes the information which an anomaly outputs, without modification. There may be no information with certain anomalies.

### 4.3. run\_status\_\*

A file giving a summary of the state of the scenario, under /var/dsbench/controller/running.

The formats of the file are as follows

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

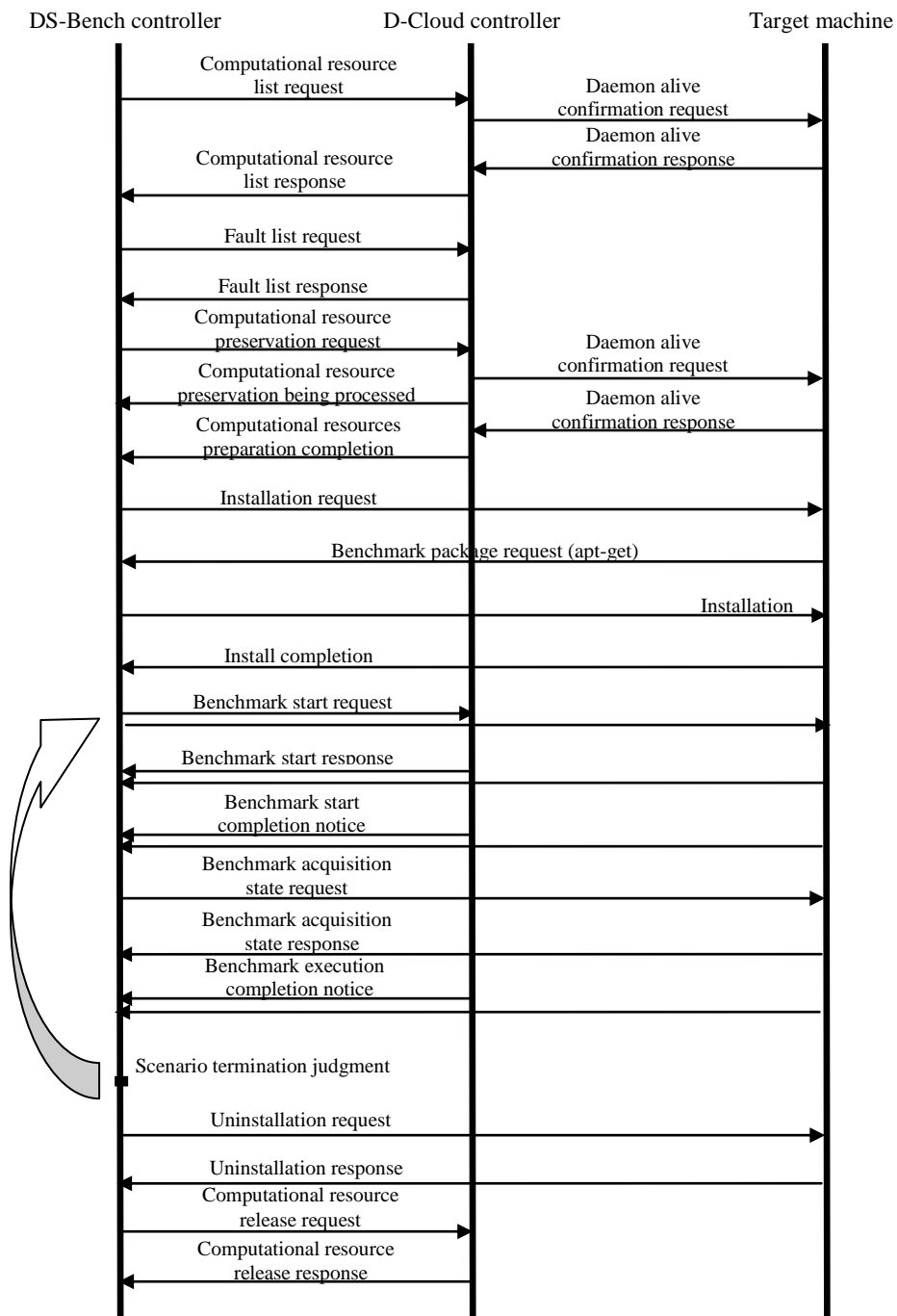
FROMWHERE gives the module from which scenarios have been executed. When the run from the GUI(run\_benchmark.cgi), this has the "gui" value. When run from the command line(run\_scenario.cgi) or D-Case Editor/D-Case Weaver, this has the "external" value.

## 5. Communication specification

This Chapter describes the communication messages between D-Cloud controller and DS-Bench.

The communication sequence involving the DS-Bench target and D-Cloud controller, and DS-Bench controller is as follows.

Communication sequence



See "Specification of Interface between DS-Bench and D-Case Editor for their Collaboration" for explanations of the interface and communication sequence of D-Case Editor/D-Case Weaver and DS-Bench.

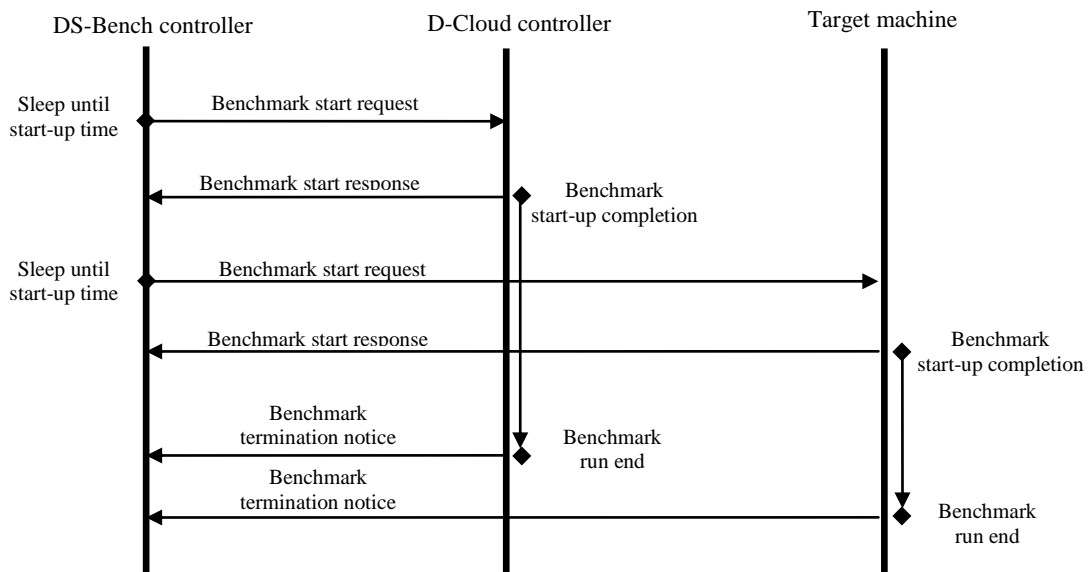
The following describes the function for execution synchronization of multiple target machines. There are three types of timings to execute a benchmark:

- Execution at a specified time
- Execution after confirming the start of a specified program
- Execution after confirming the end of a specified program

The following sequence diagram is an example where a benchmark is triggered at a specified time

After waiting until a specified time in the DS-Bench controller, the benchmark start-request is sent to each target machine.

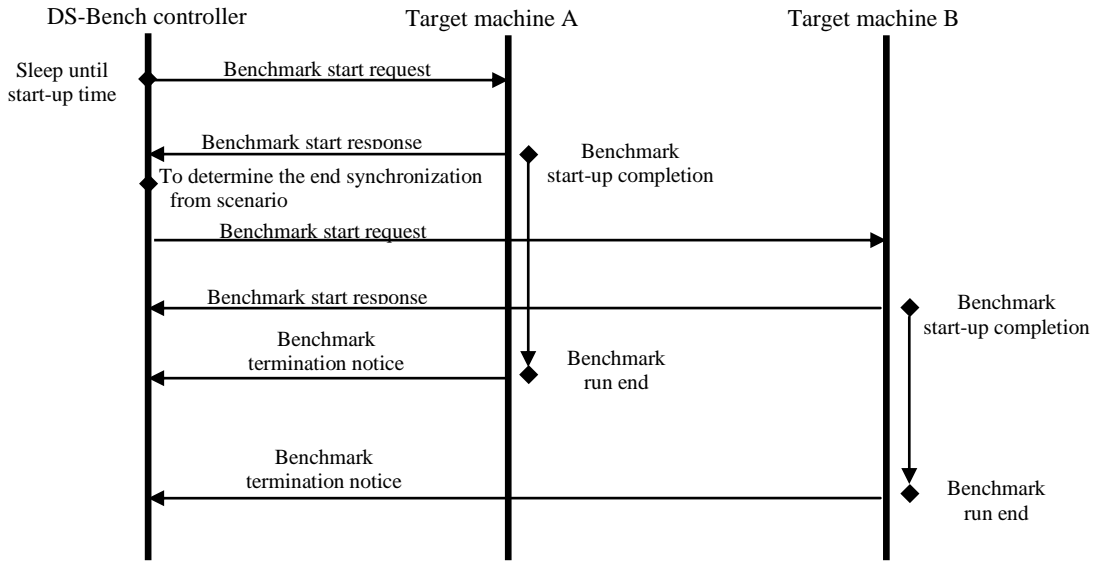
Communication sequence (The execution by the specified time)



The following sequence diagram is an example in which the benchmark is started after confirming the start of a program.

After receiving the benchmark start response that was sent from the target machine, DS-Bench controller determines whether there is a benchmark for performing start/end synchronization from the scenario and executes the benchmark against another target.

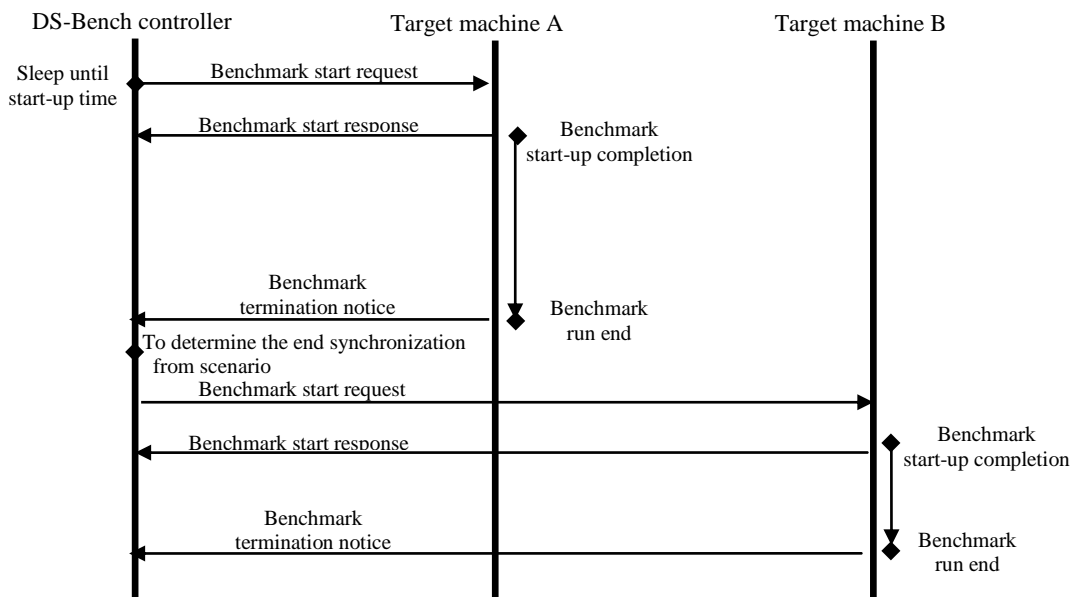
Communication sequence (Run after confirming the start of a program)



The following sequence diagram is an example in which the benchmark is started after confirming the end of a program.

After receiving the benchmark termination notifications sent from the target machine, DS-Bench determines whether there is a benchmark for performing startup/end synchronization from the scenario and requests another target to start benchmarking.

Communication sequence (Performed after confirming the end of the program)





## 5.1. Communication message

This section describes messages communicated among DS-Bench controller, DS-Bench target, and D-Cloud controller.

### 5.1.1. Benchmark installation request

Header	ID	ID	Fixed value
	CMD	'I'(0x49)(1byte)	Benchmark installation request
	SUB_CMD	'?'(1byte)	Server-side identifier
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Benchmark string (Separated by semicolons)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Response data

### 5.1.2. Benchmark installation response

Header	ID	ID	Fixed value
	CMD	'I'(0x49)(1byte)	Benchmark installation request
	SUB_CMD	'?' (1byte)	Server-side identifier
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Benchmark ID
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Response data

### 5.1.3. Benchmark uninstallation request

Header	ID	ID	Fixed value
	CMD	'U'(0x56)(1byte)	Benchmark uninstallation request
	SUB_CMD	'?' (1byte)	Server-side identifier
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of

			hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Benchmark string (Separated by semicolons)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Response data

#### 5.1.4. Benchmark uninstallation response

Header	ID	ID	Fixed value
	CMD	'U'(0x56)(1byte)	Benchmark uninstallation request
	SUB_CMD	'?' (1byte)	Server-side identifier
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Benchmark ID
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Response data

#### 5.1.5. Computational resource list request

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'L'(0x4C)(1byte)	Computational resource list request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	' '(0x20)(1byte)	Unused
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource allocation request XML
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)

	ANS	nothing	Response data
--	-----	---------	---------------

### 5.1.6. Computational resource list response

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'L'(0x4C)(1byte)	Computational resource list request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'A'(0x41) / 'N'( 0x4E)(1byte)	Response: NAK ('N') or ACK ('A')
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource allocation request XML (XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Response XML data

### 5.1.7. Computational resource allocation request

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'R'(0x52)(1byte)	Computational resource reservation request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	' '(0x20)(1byte)	Unused
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource allocation request XML
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)

	ANS	nothing	Response data
--	-----	---------	---------------

### 5.1.8. Computational resource allocation being processed

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'R'(0x52)(1byte)	Computational resource allocation request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'A '(0x41)(1byte)	ACK('A')
	UUID	(36bytes)	Identifier for identifying a sequence Set the UUID that was received in the computational resource allocation request
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource allocation request XML (XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Response data (XML reporting conditions as completely as possible)

### 5.1.9. Computational resources preparation completion

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'R'(0x52)(1byte)	Computational resource reservation request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'A '(0x41)(1byte)	ACK('A')
	UUID	(36bytes)	Identifier for identifying a sequence Set the UUID that was received in the computational resource reservation request
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource reservation request (XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)

	ANS	variable length	Response data (Normal response for XML)
--	-----	-----------------	---

## 5.1.10. Computational resource preparation impossible

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'R'(0x52)(1byte)	Computational resource reservation request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'N'(0x4E)(1byte)	NAK('N')
	UUID	(36bytes)	Identifier for identifying a sequence Set the UUID that was received in the computational resource reservation request
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource reservation request XML (XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Response data (XML reporting conditions as completely as possible)

## 5.1.11. Benchmark scenario transmission

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'S'(0x53)(1byte)	Benchmark scenario
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	' '(0x20)(1byte)	Unused
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Scenario XML
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)

	ANS	nothing	Response data
--	-----	---------	---------------

## 5.1.12. Benchmark scenario reception completion

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'S'(0x4C)(1byte)	Benchmark scenario
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'A '(0x41) / 'N '(0x4E)(1byte)	Unused
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Scenario XML(XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Response data (Normal response for XML)

## 5.1.13. Benchmark scenario refusal response

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'S'(0x45)(1byte)	Benchmark scenario
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	' '(0x20) (1byte)	Unused
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Scenario XML(XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)

	ANS	nothing	Response data
--	-----	---------	---------------

## 5.1.14. Benchmark start request

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'E'(0x45)(1byte)	Start benchmark
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	' '(0x20) (1byte)	Unused
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Scenario XML(XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Response data

## 5.1.15. Benchmark start response (ACK)

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'E'(0x4C)(1byte)	Start benchmark
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'A '(0x41) (1byte)	ACK('A ')
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Scenario XML(XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Response data (Normal response for XML)

5.1.16. Benchmark start response (NAK)

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'E'(0x4C)(1byte)	Start benchmark
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'N'(0x4E) (1byte)	NAK('N')
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
LEN_ALL	'???????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)	
Data	LEN_REQ	'???????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Scenario XML(XML at the time of the request)
	LEN_ANS	'00000000'	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Response data (As much as possible status report XML)

5.1.17. Benchmark acquisition state request

Header	ID	'U'(0x55)(1byte)	Fixed value
	CMD	'5'(0x34)(1byte)	Benchmark acquisition request
	SUB_CMD	'C'(0x43)(1byte)	Benchmark identifier
	RES_ID_LEN	'???????'(8bytes)	Number of bytes of scenario ID(8-digit hexadecimal character)
	RES_ID	variable length	Scenario ID
	LEN_ALL	'???????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'???????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Scenario XML (XML at the time of the request)
	LEN_ANS	'???????'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Response data (As much as possible status report XML)



## 5.1.18. Benchmark acquisition state response

Header	ID	'U'(0x55)(1byte)	Fixed value
	CMD	'6'(0x34)(1byte)	Benchmark acquisition request
	SUB_CMD	'C'(0x43)(1byte)	Benchmark identifier
	RES_ID_LEN	'????????'(8bytes)	Number of bytes of scenario ID(8-digit hexadecimal character)
	RES_ID	variable length	Scenario ID
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Scenario XML (XML at the time of the request)
	LEN_ANS	'????????'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Response data (As much as possible status report XML)

## 5.1.19. Fault list request

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'F'(0x46)(1byte)	Benchmark acquisition request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	''(0x20) (1byte)	Unused
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhh
	LEN_ALL	'00000000'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'00000000'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	nothing	Unused
	LEN_ANS	'00000000'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Unused

## 5.1.20. Daemon alive confirmation request

Header	ID	'T'(0x54)(1byte)	Fixed value
--------	----	------------------	-------------

	CMD	'L'(0x4C)(1byte)	Computational resource list request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource list request XML
	LEN_ANS	'????????'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Unused

### 5.1.21. Daemon alive confirmation response

Header	ID	'T'(0x54)(1byte)	Fixed value
	CMD	'L'(0x4C)(1byte)	Computational resource list request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Daemon alive confirmation request(XML at the time of the request)
	LEN_ANS	'????????'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Daemon alive confirmation response XML

### 5.1.22. Fault list response (ACK)

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'F'(0x46)(1byte)	Benchmark acquisition request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'A'(0x41) (1byte)	ACK('A')
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'00000000'(8bytes)	Number of bytes of REQ (8-digits of

			hexadecimal characters)
	REQ	nothing	Unused
	LEN_ANS	'????????'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Fault XML

## 5.1.23. Fault list response (NAK)

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'F'(0x46)(1byte)	Benchmark acquisition request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'N'(0x4E) (1byte)	ACK('N')
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'00000000'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	nothing	Unused
	LEN_ANS	'00000000'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Unused

## 5.1.24. Benchmark start completion notice

Header	ID	'U'(0x55)(1byte)	Fixed value
	CMD	'7'(0x33)(1byte)	Benchmark start completion notice
	SUB_CMD	'C'(0x43)(1byte)	Benchmark identifier
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Request XML data
	LEN_ANS	'????????'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Response XML data

## 5.1.25. Benchmark execution completion notice

Header	ID	'U'(0x55)(1byte)	Fixed value
	CMD	'8'(0x33)(1byte)	Benchmark execution completion notice
	SUB_CMD	'C'(0x43)(1byte)	Benchmark identifier
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Request XML data
	LEN_ANS	'????????'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	variable length	Response XML data

## 5.1.26. Computational resource release request

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'F'(0x46)(1byte)	Computational resource release request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	"(0x20) (1byte)	Unused
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'????????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'????????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource release XML
	LEN_ANS	'00000000'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Unused

## 5.1.27. Computational resource release response (ACK)

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'T'(0x54)(1byte)	Computational resource release request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'A'(0x41) (1byte)	ACK('A')
	UUID	(36bytes)	Identifier for identifying a sequence

			The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'???????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'???????'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource release request XML
	LEN_ANS	'???????'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Unused

### 5.1.28. Computational resource release response (NAK)

Header	ID	'D'(0x44)(1byte)	Fixed value
	CMD	'T'(0x46)(1byte)	Computational resource release request
	SUB_CMD	'C'(0x43)(1byte)	D-Cloud controller identifier
	STAT	'N'(0x4E) (1byte)	ACK('N')
	UUID	(36bytes)	Identifier for identifying a sequence The format is compliant to RFC4122 string format as follows hhhhh-hhhh-hhhh-hhhhhhhhhh
	LEN_ALL	'???????'(8bytes)	Number of bytes in the Data section (8-digits of hexadecimal characters)
Data	LEN_REQ	'00000000'(8bytes)	Number of bytes of REQ (8-digits of hexadecimal characters)
	REQ	variable length	Computational resource release request XML
	LEN_ANS	'00000000'(8bytes)	Number of bytes of ANS (8 hexadecimal digits)
	ANS	nothing	Unused

## 5.2. Request XML

### 5.2.1. Computational resource list acquisition request XML

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

The root node is **<dsb\_resource\_list>**

The following tags are placed under it.

- **<resource type='(virtual|physical|network\_switch)'\>**  
Specify the type of computational resource list to get.  
virtual : Get a computational resource list for Virtual PC  
physical : Get a computational resource list for Physical PC  
network\_switch : Get a list of network switches.  
power\_switch : Get a list of the power supplies.

### 5.2.2. Computational resource allocation request XML

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

The root node is **<dsb\_resource\_desc>**

The following three tags are placed under it.

**<target type=(virtual|physical)'>**

```

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

```

Specify the physical PC or Virtual PC.

Virtual PC or physical PC is specified in the parameter type.

When the parameter is omitted, "physical" is assumed.

The following tags are placed under **<target>** tag.

**<name>**

Specify the name of the PC

**<cpu>**

```

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

```

Information on the CPU is described.

The following tags are placed under <cpu> tag.

- **<arch>**

Describe the CPU architecture.

Description examples are as follows

- x86\_64
- x86\_32 or i386
- ia64
- s390
- alpha
- arm
- ppc32
- ppc64
- sparc
- pa\_risc

- **<vendor>**

Enter the CPU vendor name

Example) "intel" "amd"

- **<model>**

Enter the CPU model name

Refer to the models in "/proc/cpuinfo"

- **<mhz>**

Specify speed of CPU in MHz.

- **<cores>**

Specify the number of CPU cores. Number of physical CPUs × number of cores in 1 CPU.

The same as the cpu cores in "/proc/cpuinfo"

- **<threads>**

Enter the number of logical divisions, when the CPU has a function to divide one physical core into multiple logical cores, such as the Intel HTT (Hyper Threading Technology).

Specify 1 if there is no such function.

The same as the cpu cores in "/proc/cpuinfo"

- **<memory type='DDR|DDR2|DDR3'>**

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

Specify the size and speed of the memory in the PC, using the following tags.

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



Specify the speed of the memory. Unit of speed is MHz or Mibps(MiB/s).

- **<size>**

Specify the capacity of memory, in MiB.

- **<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>

```

Description of the devices connected to the PC.

The following tags are placed under **<devices>** tag.

- **<device type='Device type'>**

The device type can specify 'storage' or 'ethernet'

The following tags are placed under <device> tag.

- **<target dev='device name' ip='IP address of the device' priority='Priority' type='Detailed type of device' vendor='Vendor name' model='Model name' size='Size'>**

Specify device name, IP address, priority, type, vendor, model, etc. of the target device.

If the device has multiple IP addresses, priority ranking of the addresses is confirmed, and communication is performed using the primary IP address with the highest priority,.

- **<controller type=' device type' model=' model'>**

This is specified when there is a device controller.

When using HDD or the like, specify the device type to be the interface type (sata, etc.) and specify the model to be the IO controller hub (ICH9 etc), .

- **<os>**

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

Enter information on the OS.

The following tags are placed under **<os>** tag.

- **<family>**  
Enter the classification of OS  
The following can be specified: "Linux", "Unix", "Windows", "MacOS", "Other".
- **<vendor>**  
Specify the OS vendor name .
- **<name>**  
Specify the name of the OS
- **<version>**  
Specify the version of the OS
- **<arch>**  
Specify the architecture of the OS (refer to **<arch>** of **<cpu>**)

- **<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>

```

Enter information on software to be recorded in the benchmark results.

The following tags are placed under <softwares> tag.

- **<software>**

Use one tag for each software item to be recorded.

The following tags are placed under <software> tag

- **<version>**

Enter the version of the software

- **<name>**

Enter the name of the software

- **<family>**

Enter the classification of software

- **<lang>**

Specify the language. This is used mainly in the case of a compiler.

- **<status>**

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

- **<message>**

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

- **<time\_until\_opening>**

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

**<status>** tag shows the current usage status of each computational resource and **<message>** tag shows details.

Correspondence between "message" and "status" is as follows.

- 0 : success : Success of allocation for use
- 1 : target is offline : Waiting for allocation (Not started yet on a physical PC etc.)
- 2 : target is in use : Since it is being used by others, allocation is impossible
- -1 : target does not exist : The specified target does not exist

The **<time\_until\_opening>** tag specifies the time until the resource becomes available if it is being used presently (the status tag is 2). If "status" tag is not 2, the specified value is ignored.

The value of **<status>** tag of a target to be used in the scenario is set to 2 (target is in use).

The value of the tag **<time\_until\_opening>** is the estimated time to completion of the scenario.

### **<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>
```

Specify the network switch.

The following tags are placed under **<network\_switch>** tag.

#### **<name>**

Specify the name of the network switch.

#### **<control>**

Describe the control method for a network switch.

The following tags are placed under **<snmp>** tag.

- **<addr>**  
Specify the host name or IP address.
- **<version>**

Specify the version of SNMP.

➤ **<community>**

Specify the SNMP community name.

- **<port number='Port number'connected\_to='Connection device name'>**

Specify the number of the port holding the network switch and the name of the device connected to the port.

**<power\_switch>**

```

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

```

Specify the SNMP controllable power supply

The following tags are placed under <power\_switch> tag.

- **<name>**

Specify the name of the power supply.

- **<control>**

Describe the control method of the power supply.

The following tags are placed under <snmp> tag.

➤ **<addr>**

Specify the host name or IP address.

➤ **<version>**

Specify the version of SNMP.

➤ **<community>**

Specify the SNMP community name.

- **<port number=' Port number' connected\_to=' Connected device name'>**

Specify the number of the port holding the power supply and the name of the device connected to the port.

### 5.2.3. Benchmark scenario 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>

```

Benchmark scenario XML is transmitted as it is.

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

A benchmark is specified at this Node.

- Enter the execution sequence as seen from the DS-Bench controller side in "seq"
- Enter the ID for determining the benchmark at <sstop> and <sstart> in "ID"
- Enter a name to be used for convenience in "title" (can be left blank)
- Enter the full path to the benchmark in "path".

Enter a single benchmark [unit] or C / S type benchmark [link] in "type".

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

Enter the setting of the server of the C / S type benchmark.

This is valid only when the type of the benchmark node of a higher rank is "link".

Iteration : Impossible, Omission : Possible (If type=link, it can not be omitted),

Specify the host name or IP address of the Server machine of the "host".

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

Enter the setting for the client of the single or C / S type benchmark under this node.

Iteration : Impossible, Omission : Possible

Specify the host name or IP address of the Client machine of the "host".

**<stime interval="">Start time</stime>**

The execution start time of a benchmark relative to the start of the benchmark scenario is set in seconds.

The interval before starting is specified as an "interval" parameter, expressed in milliseconds.

**<ftime interval="">End time</ftime>**

The execution end time of a benchmark relative to the start of the benchmark scenario is set in seconds.

The waiting interval before ending is specified as an "interval" parameter, expressed in milliseconds.

**<output/>**

Specify the format of the result output.

If this is omitted, the result is not returned.

Iteration : Impossible, Omission : Possible

**<caption/>**

Set up the title of results.

Iteration : Impossible, Omission : Possible

**<table/>**

Make settings regarding the output of a result.

Iterate entries in this table tag for each table that you want to retrieve.

Iteration : possible, Omission : Impossible

**<data>[str]...</data>**

Values retrieved from the result are enclosed in parentheses.

\* When regular expressions are used in a script, values enclosed in parentheses are extracted . Therefore, please do not use nested parentheses like (()).

Iteration : Impossible, Omission : Impossible

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

Set "0" to hide the name of the value.

If omitted, "1" is assigned automatically.

Omission : Possible

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

Set "0" to hide the value.

If omitted, "1" is assigned automatically.

Omission : Possible

**<plain></plain>**

The intermediate step results are stored under this Node.

Iteration : Possible, Omission : Impossible

**<tags/>**

Tags that you entered are recorded in this tag, separated by commas.

#### 5.2.4. Benchmark start request XML

Benchmark start request XML is the same as the XML that was used to send scenario.

#### 5.2.5. Daemon alive confirmation request XML

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

The XML to be used for the daemon alive confirmation request is the XML of the target to be confirmed that is in the computational resource allotment request XML.

#### 5.2.6. Calculating resource release request XML

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



The XML to be used for the computational resource release request is equal to the computing resources allocation response XML.

## 5.3. Response XML

### 5.3.1. Computational resource list response 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>

```

One **<target>** tag or **<network\_switch>** tag, and one **<power\_switch>** tag are returned for each acquired computational resource.

For the contents of each tag, refer to section 5.2.2 Computational resource allocation request XML.

### 5.3.2. Computing resources allocation response XML

The same XML format is used as the XML in section 5.2.2 Computational resource allocation request XML.

### 5.3.3. Benchmark scenario response 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>
```

The XML in the almost same format as the benchmark scenario XML is used.

The difference is that each **<benchmark>** tag has a **<message>** tag and **<status>** tag.

Meanings of **<message>** tags and **<status>** tags and their correspondence are as follows

status	message	Meaning
0	success	Normal
1*		Error with regard to the parser
11	parser error	Failed to parse the XML request
12	fail in parse of schema	Failed to parse RelaxNG Schema
13	fail in validate	Failed to Validate
14	cannot resolve the host name	The failure to resolve the host name
2*		Error setting the benchmark hierarchy
21	seq is not numeric	non-numeric character mixed in seq
22	PATH is not a file	The PATH does not specify a file.

23	don't have permission in PATH	Do not have execution access privileges in the place the PATH specifies.
24	type is invalid	Designation of type is invalid
25	the combination is wrong	Combination of type = unit and subcmd = S
26	seq is overlapped	Seqs overlap
3*		Unused
4*		Setup error regarding C/S input
41	stime is not numeric	Unsupported non-numeric character is in the stime
42	ftime is not numeric	Unsupported non-numeric character is in the ftime
43	label is invalid character	Unsupported character is in the label
44	arg is invalid character	Unsupported character is in the arg
45	ui must be yes or no	Characters other than yes or no are contained in ui.
46	interactive must be yes or no	Characters other than yes or no are contained in interactive.
5*		Unused
6*		Error in setting output
61	parentheses is unbalanced	A missing parenthesis
62	parentheses is succession	There are nested parentheses
7*		Unused
-1	benchmark does not exist	The benchmark that is specified does not exist

### 5.3.4. Benchmark state XML

```

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

```

XML that is returned in the benchmark state acquisition response

Stored in "ANS" part

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

A benchmark is specified by this Node.

- The execution sequence seen from the DS-Bench controller side is written in "seq".
- The name used for convenience is entered in "title". May be left blank.
- The full path to the benchmark is entered in "path".
- Either a single benchmark "[unit]" or a C / S type benchmark "[link]" is entered in "type".

Iteration : possible, Omission : Impossible

**<caption>[str]</caption>**

Title of the results is specified in a regular expression or character string.

Iteration : Impossible, Omission : Possible

**<config></config>**

Unused, Not set

Iteration : Impossible, Omission : Possible

**<plain></plain>**

The intermediate step results are stored under this Node.

Iteration : Impossible, Omission : possible

## 5.3.5. Daemon existence confirmation response 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>

```

An XML in the same format as the computational resource allocation request XML is used

When the daemon exists, the latest software status at that time is stored in the **<software>** tag.

For the contents of each tag, refer to section 5.2.2 Computational resource allocation request XML.

### 5.3.6. Fault list 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>

```

The XML in almost the same format as a benchmark conf file is used.

The difference is that each <benchmark> tag has <constraints> attribute.

The possible values of the "constraints" attribute are as follows.

- vm : Fault injection dedicated to the virtual machines
- netlink : Fault injection that can disconnect network link using the switch control
- pdu : Fault injection that can control power with PDU
- ipmi : Fault injection that can control power with IPMI

There are multiple <benchmark> tags in the fault list XML. All the faults which can be used are described by this XML.

## 5.4. Communication with the D-Case Editor/D-Case Weaver

This section describes the XML to communicate with the D-Case Editor/D-Case Weaver.

### 5.4.1. Scenario execution start

In scenario execution start CGI, a response body is changed. The example of XML is shown.

```

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

```

Add a new view\_id attribute. This shows the ID of results displayed in the GUI of DS-Bench.

### 5.4.2. Scenario Result View

This new XML calls up the scenario result inspection CGI (view\_result\_detail.cgi). Here is an example of a request/response.

- Request method : GET
- Request header : view\_id=scenario1

- Response body : HTML that can be displayed in the browser (A screen example is shown in section 6.4 view\_result\_detail)

### 5.4.3. Scenario stop

A scenario stop CGI (stop\_scenario.cgi) is newly created. Here is an example of a request/response.

- Request method : POST
- Request body

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

- Response body

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

Below are specific messages and codes used in the response.

#### Scenario stop status

code	message	Detail
0	success	Successful completion
1*		Parser error
11	request XML parser error	Failed to parse the XML request
5*		IO error
52	parameter do not exist.	The specified parameters do not exist
9*		Other
94	the scenario is not running.	The scenario has stopped running
95	D-Case Editor Error: Scenario has not executed from D-Case.	The specified scenario is not executed from D-Case Editor

#### Scenario execution status

status	Detail
stopping	Scenario is stopping
stopped	Scenario completely stopped
st_error	Error while the scenario was stopping

## 6. Screen specification

### 6.1. Common menu

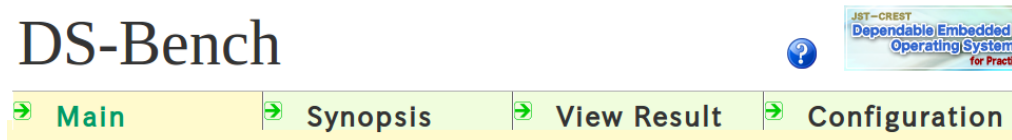


Fig. 6-1 Common header

Click the site title "DS-Bench" link, then move to the Main screen (main.cgi).

Click the "Main" link, then move to the Main screen (main.cgi).

Click the "Synopsis" link, then move to the Synopsis screen (synopsis.cgi).

Click the "View Result" link, then move to the Result screen (View Result.cgi).

Click the "Configuration" link, then move to the Configuration screen (configuration.cgi).

### 6.2. Main screen

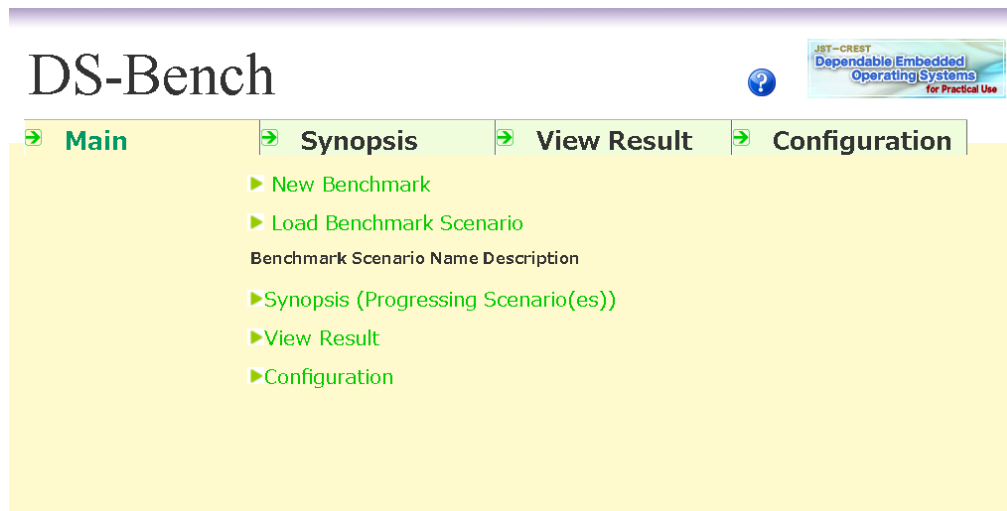


Fig. 6-2 Main screen

Click the "New Benchmark" link, then move to the Benchmark screen (benchmark\_main.cgi) in a new window → Benchmark screen.

Click the "Load Benchmark Scenario" link, then move to the Benchmark screen in a new window. Show the saved scenario list → benchmark scenario read screen

Click the "Synopsis (Progressing Scenario(es))" link, then move to the Synopsis screen (synopsis.cgi).

Click the "View Result" link, then move to the Result screen (View Result.cgi).

Click the "Configuration" link, then move to the Configuration screen (configuration.cgi).



## 6.3. Benchmark screen

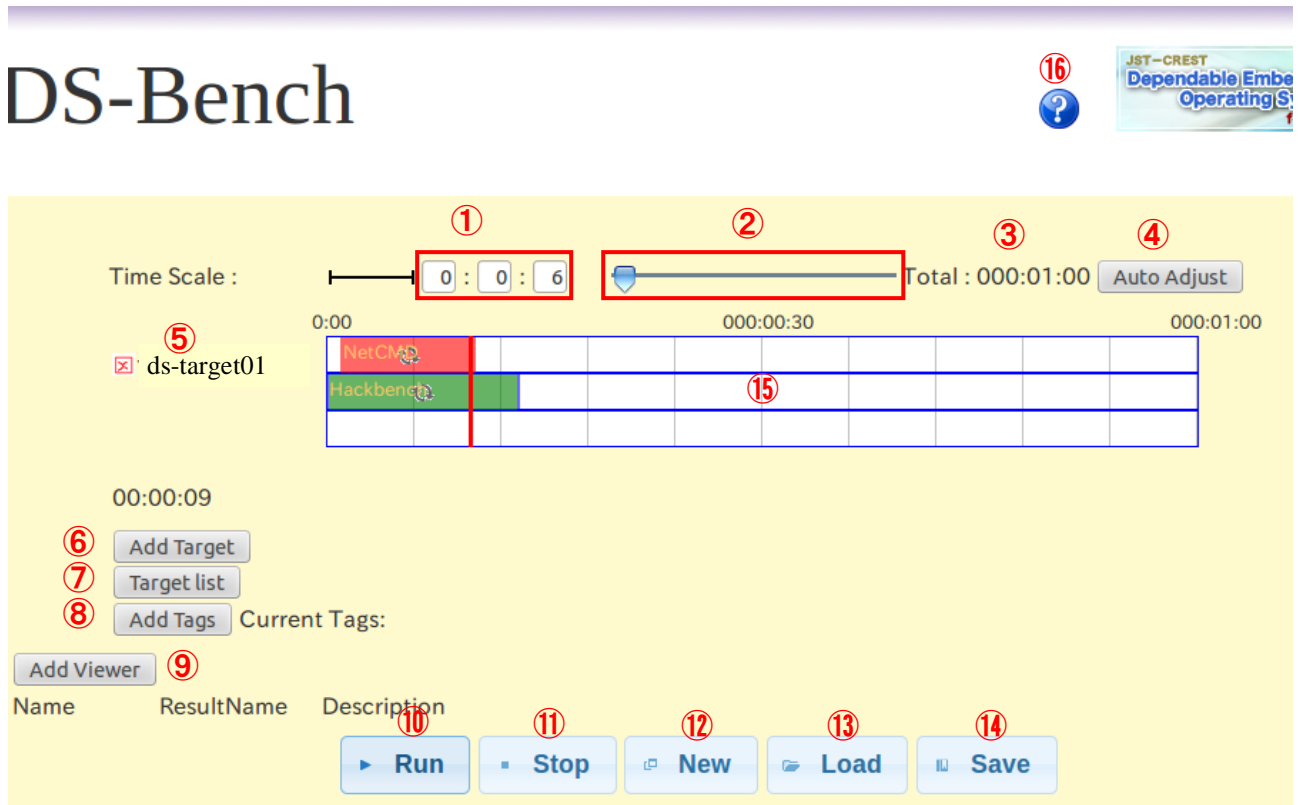


Fig. 6-3 Benchmark screen

In the following, the circled numbers in the Fig 6-3 Benchmark screen are explained simply.

① This is an input box where you can set the length of time of one unit length of the time scale. (It also changes the entire time of ③). From left to right, the small boxes set hours, minutes, and seconds.

In the case of 1:3:6, one unit length is 1 hour 3 minutes 6 seconds. Since there are ten unit lengths, the total time is 10 hours 31 minutes.

② This is a time scale bar. By dragging the indicator to the left or right, you change the time scale.

③ Displays the entire time of the time scale that is currently displayed.

④ "Auto Adjust" button ... Button that changes the time scale so that the longest benchmark time is 80% of the total time.

⑤ Displays the name of DS-Bench target, added using the "Add Target" button ⑥, when it was registered in D-Cloud.

Click the "delete" icon and the target is deleted.

Click the displayed name and the target setting screen appears.

⑥ "Add Target" button ... Displays the screen where a new target can be added.

⑦ "Target list" button ... Displays the target machine list.

⑧ "Add Tags" button ... Displays a screen where you can edit a tag in a benchmark scenario being edited.

⑨ "Add Viewer" button...Displays the name that corresponds to the name of the result to be passed to D-Case Editor / D-Case Weaver, the result of the benchmark, and the number of the clicks made for the explanatory note input box

⑩ "Run" button ...Start the Benchmark with the current benchmark configuration

⑪ "Stop" button ... End the benchmark.

⑫ "New" button ...Create a new benchmark configuration. At this time, the current benchmark configuration is discarded.

⑬ "Load" button ...Displays the screen for reading the benchmark configuration that has already been saved.

⑭ "Save" button ...Displays the screen for saving the current benchmark configuration.

⑮ Benchmark configuration on the timeline

⑯ Move to the Help screen

### 6.3.1. Dialog to add targets

Click "Add Target" and the Add Target dialog is displayed.

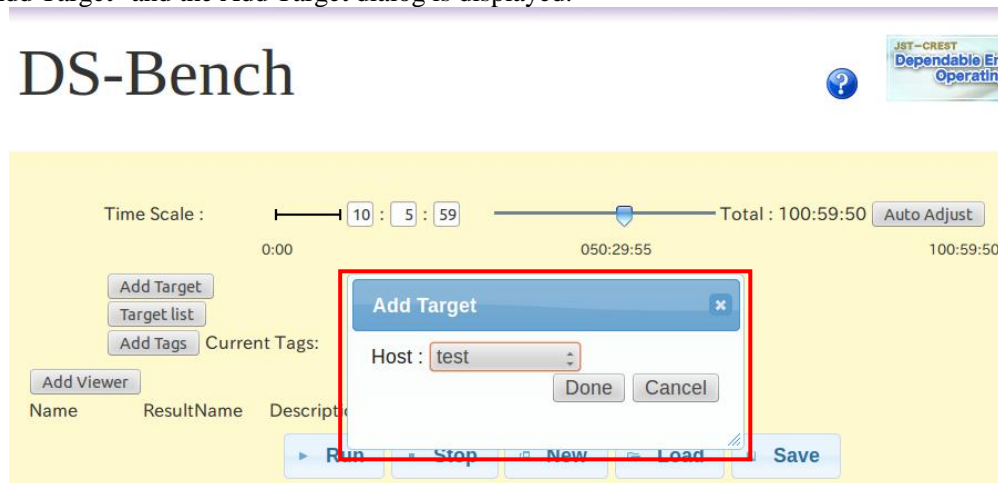


Fig. 6-4 Add Target dialog

Add the new target information.

Select the target on which you want to run the benchmark in the Host. After selecting the target, click the "Done" button, and then the dialog box is closed. To close the dialog box without doing anything, just click the "Cancel" button.

### 6.3.2. Target Editing dialog

Click the target name in ① in Fig 6-5.

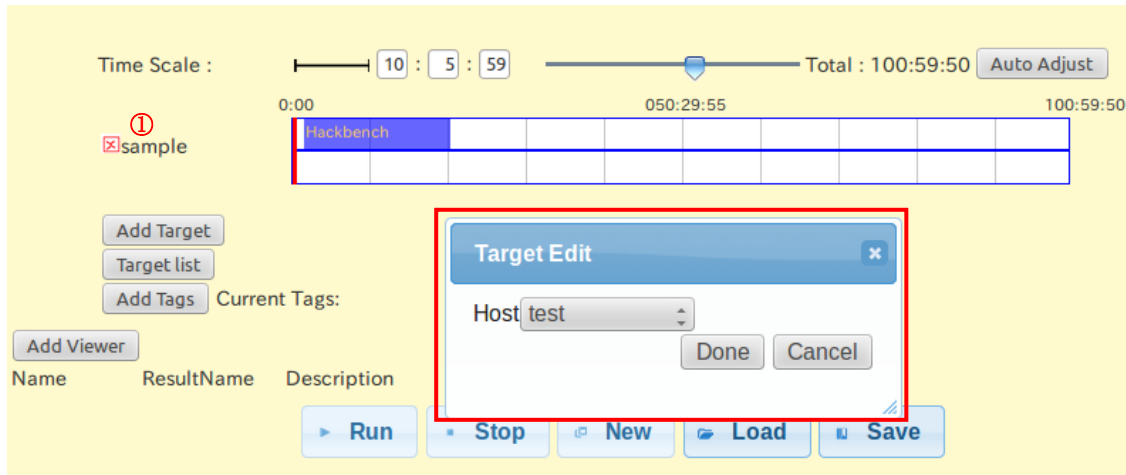


Fig. 6-5 Target Edit dialog

Edit the information of the target that was added using the Add Targets screen

Select the target on which you want to run the benchmark in the Host. After selecting the target, click the "Done" button and then the dialog box is closed. To close the dialog box without doing anything, just click the "Cancel" button.

### 6.3.3. Target list

Click the "Target list" button to display a list of target and their status below the "Run" button. Click the "Back" button below the list to return back.

This screen display is almost the same as the Configuration Target list.. The difference is that the "Back" button used in Configuration becomes a "Refresh" button.

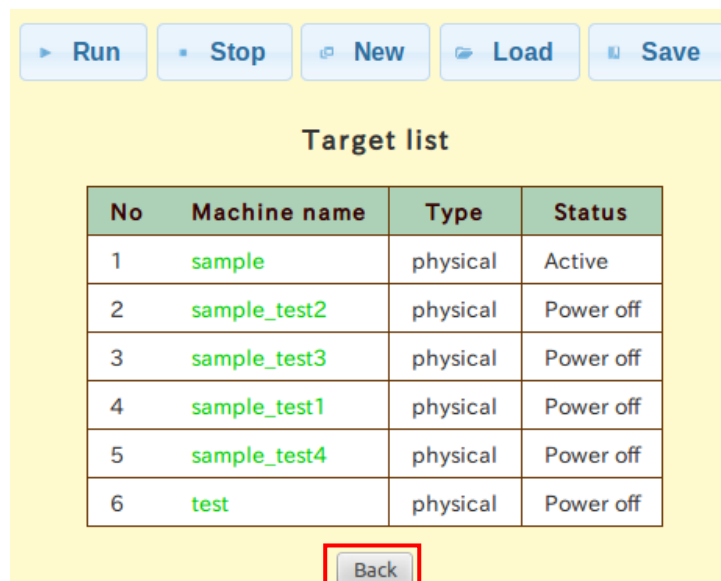


Fig. 6-6 Display of target list

For more information, see the " Configuration Target list".

### 6.3.4. How to operate on the time scale

#### 6.3.4.1. Relationship connections among benchmarks (ending times)

It is possible to register a Fault Injection or benchmark for a specified target by dragging on the time scale.

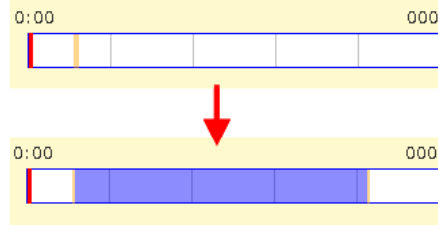


Fig. 6-7 Drag on the time line

This displays a blue area (in benchmark area) in the case of a benchmark, and a red area in the case of Fault Injection on the screen. The time setting is determined by the starting and ending positions of the dragging; the times of those positions are set. After dragging, the dialog for the detailed benchmark setting is displayed. Here, you can edit the types of benchmarks you want to run and their options. It is possible to edit the starting time and ending time of the benchmark on the time scale by using the mouse.

It is possible to change the starting time without changing the execution time by dragging the mouse cursor in the benchmark area (except for both edges) . If the cursor is dragged beyond the left end of the benchmark area, the starting time is set to 0:00 and the end time to is adjusted to match the execution time. If the cursor goes into another benchmark's execution time or goes beyond the right end of this benchmark's area, then the original position is reset.

Moreover, it is possible to change the execution time by dragging on the both ends of the benchmark area, changing the benchmark area size. If the cursor is dragged beyond the left end in the time scale, the starting time becomes 0:00. If the cursor is then dragged beyond the right end then the end time becomes the length of the time scale.

A number of Benchmarks / Anomaly loads can be registered in the same time period. The width of the gauge is one row in the initial stage. After using the one row, one more row is added. (The number of allocated rows is the number of rows used +1 row, with the maximum width being 8 rows). It is not possible to move up or down to a different line during the set time period.

#### 6.3.4.2. Relationship bonds among benchmarks

Relationship bonds are the application of multiple benchmarks/anomaly loads between targets of the same type or of different types. When two benchmarks / Anomaly loads are applied and one of the benchmarks/anomaly loads is dragged so as to overlap the other, a Confirmation dialog is displayed. Even in the case where the targets are different, we can make relationship bonds among them by overlapping

benchmarks that conduct the same action. However in this case, there is no vertical movement across the gauge.

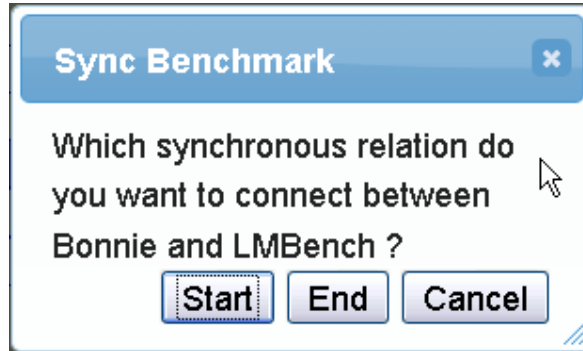


Fig. 6-8 Relationship bond dialog

A relationship can be bound between two Benchmarks / Anomaly loads by clicking the "start" or "end" button.

Click the "start" to make the following relationship bond between two benchmarks/Anomaly loads. After starting the first benchmark/Anomaly load, the 2nd benchmark / Anomaly Load is set to start after waiting a time that is set in the "Interval before starting (ms)". (We define the dropped benchmark/Anomaly load as the first and the dragged one as the second)

Click the "end" to make the following relationship bond between two benchmark/Anomaly loads. After finishing the first benchmark/Anomaly load, the 2nd benchmark / Anomaly Load starts after waiting a time that is set in the "Interval after termination (ms)". (We define the dropped benchmark/Anomaly load as the first and the dragged one as the second).

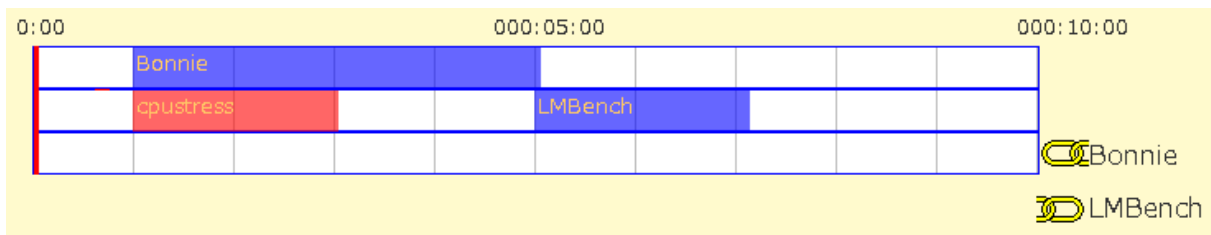


Fig. 6-9 Timeline of relationship bond runtime

### 6.3.5. Detailed benchmark settings dialog

The red frame in Figure 6-10 contains items common to all selected Programs. The blue frame contains items specific to one selected Program.

Details

Target : trg01

Program : NetCMD

Network load tool

Anomaly load

Begin time: 0 H 0 M 0 S

End time: 0 H 10 M 0 S

Interval before starting(ms): 0

Interval after termination(ms): 0

Mode: rate loss

order duplicate :

Value: bit % % % :

Interface : eth0

Done Delete Cancel

Fig. 6-10 Benchmark settings dialog

「Target : 」 ...Display the name of the target that is currently being edited

「Program : 」 ...Displays the Program drop-down list. Select the benchmark to run.

(There is a list for a physical machine and a list for a virtual machine.)

「Anomaly load check box」 ...Check this box if you want to run the benchmark as Anomaly load. (Check Yes = Anomaly load, Without Check = Benchmark)

「Begin Time : 」 ...Set the start time of the benchmark

「End Time : 」 ...Set the end time of the benchmark

「Interval before starting(ms) : 」 ...Set the estimated amount of time needed before starting the benchmark.

「Interval after termination(ms) : 」 .... Set the estimated amount of time needed before ending the benchmark.

Click the "Done" button to make the edited contents into the benchmark scenario.

Click the "Delete" button to delete the benchmark from the time line

Click the "Cancel" button to close the dialog box without taking any action

#### 6.3.5.1. Interval time

The estimated time before starting which is specified in the "Interval before starting (ms)" is used as a timer for the interval in which to determine that a certain Benchmark / Anomaly load has started, when the relationship-bond "Start when a certain benchmark/Anomaly load starts " is set.

The estimated time specified in the "Interval after termination (ms)" is used as a timer for the interval in

which to determine that a certain Benchmark / Anomaly load has ended, when the relationship bond "Start when a certain benchmark/Anomaly load finishes" is set.. If the benchmark program is not completed within this time then it is killed forcibly by the system.

### 6.3.5.2. Specific configuration items

Specific configuration items (parameters of the benchmark) are different for different types of benchmark. Some parameters can be selected in the drop-down list. The execution time can be passed on as a macro variable, or this value can be given to the D-Case.

Specifiable macro variables are as follows.

- \$(runsec) : converts the execution time to seconds.

When a text box where the macro variable in question can be specified is focused upon, macro variables that can be specified with a tool chip are displayed .

In the DS-Bench, when entries are made using the following format, the parameter names, descriptions, and values that are specified by the D-Case Editor / D-Case Weaver are set. However, this does not apply to all parameters. For instance, the parameter "Interface" in Figure 6-7 that is selected from drop down list is not applicable.

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

Below are the variable portions of the above format. Note: (\${}:) are half-width characters.

Parameter name	Meaning	Description
define-name	D-Case parameter	Name used to change the benchmark startup parameter value when the scenario is being run in the D-Case. Only half-width character strings with "-" and "_" can be input.
description	Description	Detailed description of the D-Case parameters. Can be omitted. "" ¥/: () {} <> \$ can be used.
value	Value	Benchmark startup parameter value

If it is not in the format \$(parame:...), we know that the benchmark start parameter value has been set. If a name in which "define-name" is entered is specified as the parameter by D-Case Editor/D-Case Weaver , then the benchmark startup parameter value is changed to that specified by D-Case Editor/D-Case Weaver.

When the parameters are to be displayed in a drop down list, the list must be prepared taking the resource list acquired from the D-Cloud into consideration. Parameters then can be selected from the list.

### 6.3.6. Use of Fault injection

It is possible to use the Fault injection provided by D-Cloud roughly in the same way as the benchmark.

Fault injection is displayed in a Program drop-down list, like the benchmarks. The candidate can be obtained from a fault list response. However, if a fault condition does not match the current target, that fault is not displayed. For example, if a fault is a program only for a virtual machine target, then it is not listed in the fault list when the current target machine is physical.

When a fault is selected in a Program, the check box of Anomaly load automatically becomes "selected / non-active". Other check boxes exhibit the same behavior when another benchmark is selected.

### 6.3.7. Add tags

Click the "Add Tag" button and a dialog is displayed for adding tags to the benchmark scenario.

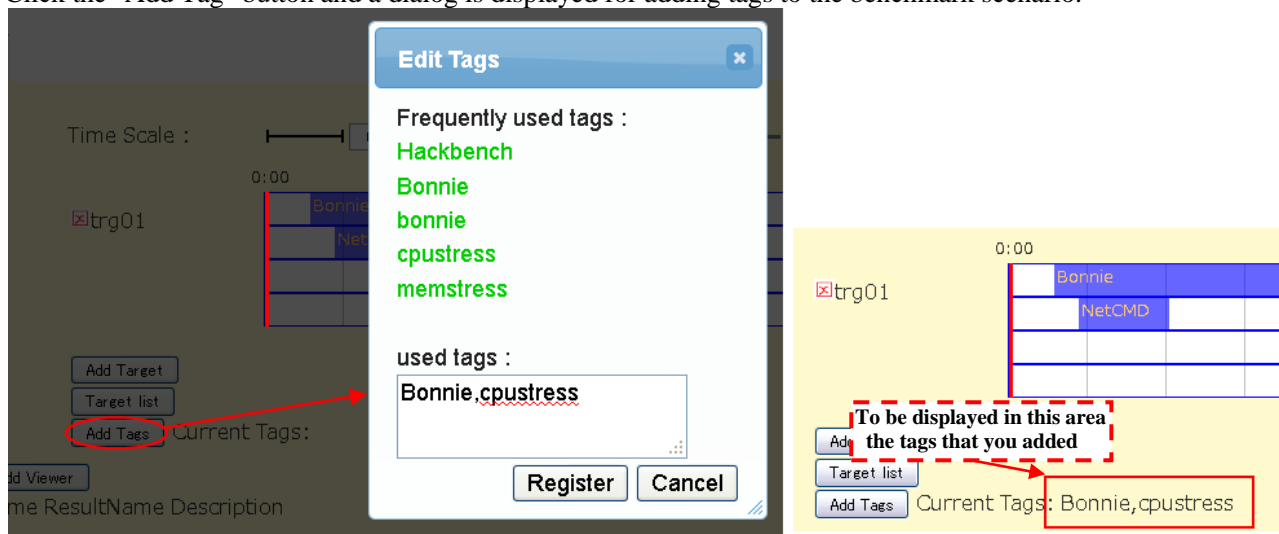


Fig. 6-11 Edit Tags dialog

"Frequently used tags:" can show at most five tags which are often used.

Click the anchor and the tag is added to "used tags :".

Input the tag name you want to register into the used tags text box. When you register more than one tag, you need to separate them by half-width commas.

Click the "Register" button to register the added tag into the scenario.

Click the "Cancel" button to close the Dialog box without any actions performed.

Registered tags are stored in <tags> in the benchmark scenario.

### 6.3.8. Execution of the scenario and its Result

Start the scenario by clicking the Run "button". While the benchmark is running, the current time is visually shown by the red line in the timeline. The progress log is displayed in the box below while the benchmark is running.



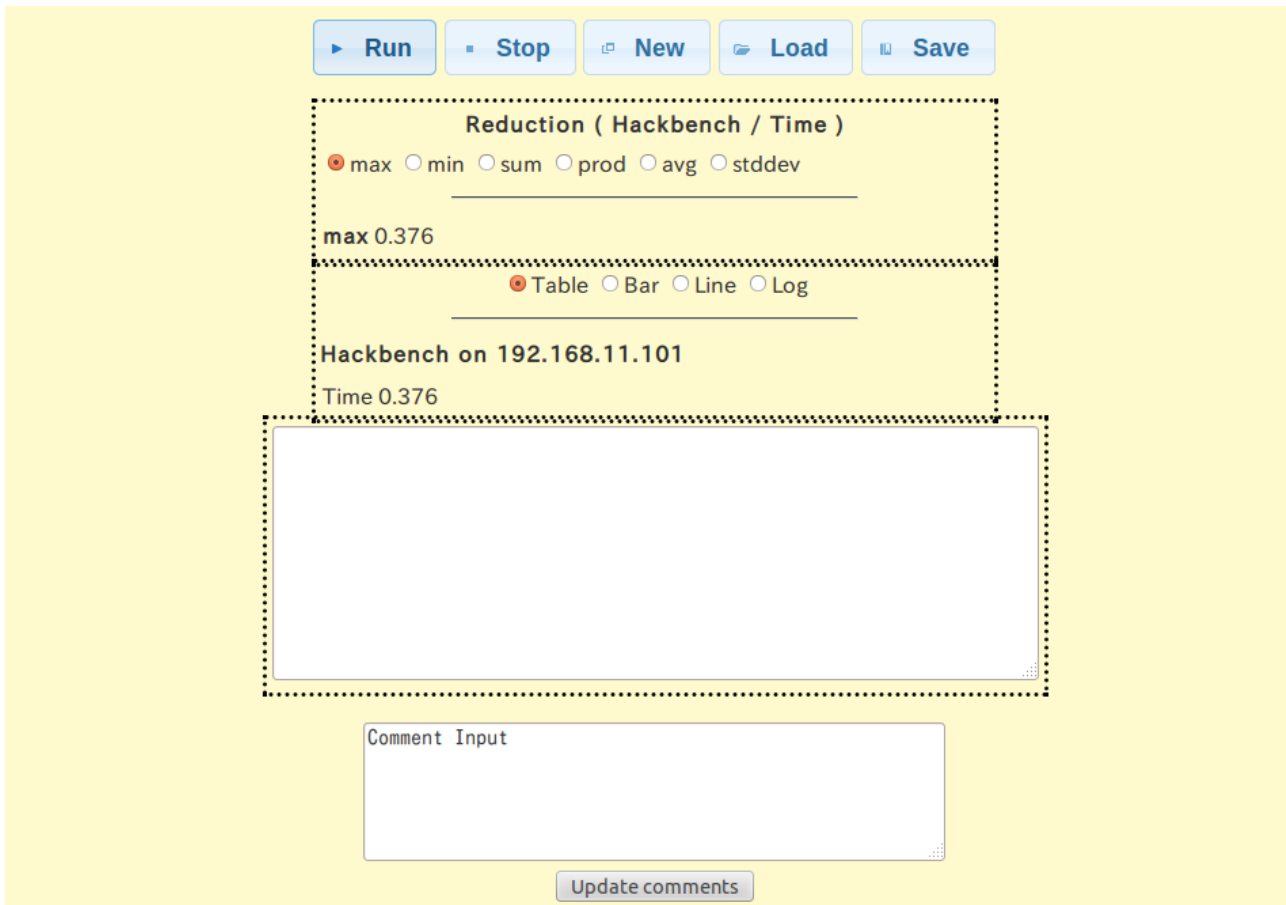


Fig. 6-12 View of Results

The result of the Reduction after a benchmark is terminated is displayed. The reduction function is a function that collects and calculates the results from multiple benchmarks of the same type. Types of calculations are MAX (maximum value), MIN (minimum value), SUM (total sum), PROD (total multiplication product), AVE (average), STDDEV (standard deviation). A result table like that produced by other benchmarks is displayed as the reduction, and there is a radio button to change the table displayed. A reduction table is also displayed for a scenario that does not need any calculation.

"Table", "Bar", "Line" and "Log" radio buttons are displayed if the all results are obtained in the DS-Bench controller.

It is possible to switch the display among "table", "bar", "line" and "progress log" by clicking.

Note: while the benchmark is running, it cannot display anything because the log is not recorded completely.

After the all benchmarks are finished, a finish message is displayed. A form for writing comments then appears. After writing comments, you can save the information by pressing the "Update comments" button.

### 6.3.9. Cooperation with the D-Case

Click the "Add Viewer" button to display the text box in which to enter the result information to be used in

the D-Case Editor/D-Case Weaver cooperation.

The text box consists of Name, ResultName and Description. These are respectively the name of the benchmark, benchmark result names, and explanatory descriptions (for display in D-Case Editor and D-Case Weaver). The ResultName (Benchmark results name) is specified by the order of benchmarks and their results.

There are two ways to specify the results you wish to access. The first way is to specify the number of the table in the displayed column, counting from the top.

In this case, the display command is as follows

`$(benchmark: Target name: number of the timeline: number of the benchmark: number of the displayed table)`

In Fig 6-13, the setting is the "trg3",target, the first timeline, the first benchmark in that timeline, and the 18<sup>th</sup> of the table listed in the results of running that benchmark. The setting for sending this to D-Case Editor/D-Case is as follows:

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

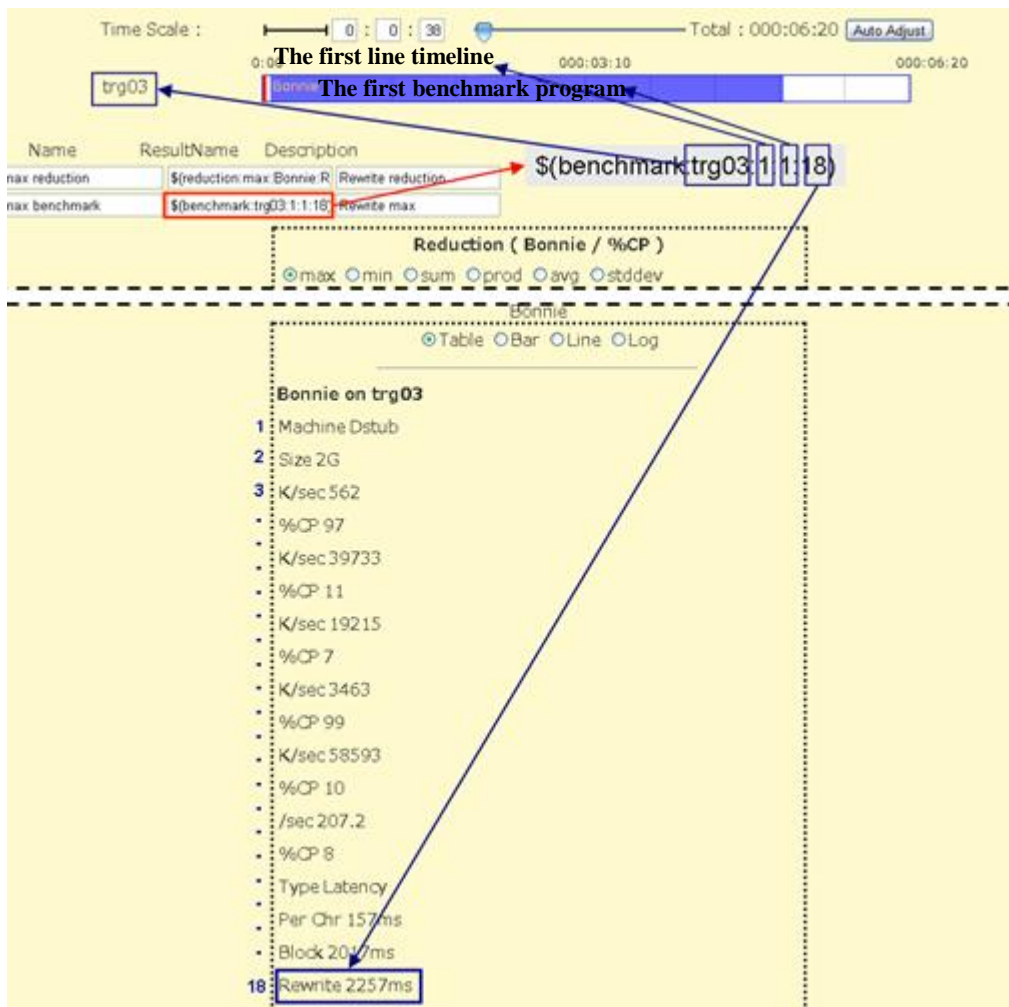


Fig. 6-13 View Results

The second way to specify results is to enter a name that represents a result of the reduction.

In this case the command is as follows

`$(reduction: Reduction selection: benchmark name: label name)`

Fig 6-14 shows the setting sending the maximum value of the Rewrite label from the reduction results of Bonnie to D-Case Editor/D-Case.

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

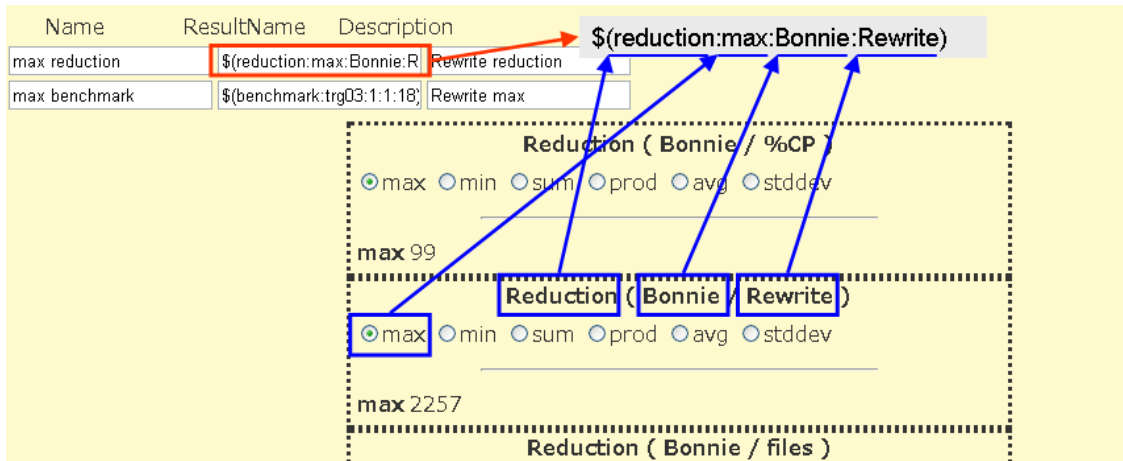


Fig. 6-14 View Results

## 6.4. Synopsis screen

ID	Date	Status Progress Remaining	Comment	Benchmark Program	Anomaly Load	Starter	View Scenario detail	Stop
193		reserve				gui	<a href="#">View</a>	<a href="#">Stop</a>
192	2013/05/31 15:18:16	run (56%) 0min	The CPU-load test by cpustress.	cpustress		gui	<a href="#">View</a>	<a href="#">Stop</a>
191	2013/05/31 15:16:11	finish	Benchmark by Httpperf. This benchmark accesses the Web server (ConnectionParSec) times a second until the number of accesses reaches (TotalConnections). It evaluates whether expected value((ResponseTime(ms))[ms]) is full filled or not.	Httpperf		external	<a href="#">View</a>	<a href="#">Stop</a>
190		error DS-Bench Error: failure to reserve calculation resources.				gui	<a href="#">View</a>	<a href="#">Stop</a>
189	2013/05/31 15:14:34	finish	The memory stress test by memstress.	Memstress		gui	<a href="#">View</a>	<a href="#">Stop</a>
			Benchmark by Httpperf This					

Fig. 6-15 Synopsis screen

This screen can be invoked from the main tab at the top of the DS-Bench screen.

The displayed items are as follows. These items are taken from the contents of a status file and a scenario XML file traced back from that status file.

- ID : Result ID
- Data : Scenario start date and time
- Status Progress Remaining : Progress (status), remaining time (min)
- Comment : Comments and Description of scenario
- Benchmark Program/Anomaly Load : The benchmark / Anomaly / Fault injection that was used
- Starter : The place from which the scenario was started (DS-Bench | External)

The result ID is a hyperlink. Click the result ID to move to scenario result screen.

In this screen, there are the "Reload" button, "Stop the scenario" button and "View the scenarios" button. The "Reload" button is acquired the new scenario status and updates part of the displayed information. Note that this screen is not updated automatically.

It is also possible to stop the scenario that is being executed in the D-Case Editor/D-Case Weaver through this screen. Click the "Stop" button and the benchmark scenario is stopped. This "Stop" button does not work on an ended scenario. Note that clicking the detailed scenario display button displays the corresponding detailed screen. With this action, a new window is opened in the screen, but a new window cannot be opened for the same scenario. Also note that this button does not work on an ended scenario.

To determine the scenario which is now running, use the status file (Run\_status\_ \*: progress rate and status are described for each scenario). This file is created using either D-Case Editor/D-Case Weaver or the DS-Bench. This file is used to make the list of existing scenarios.

It is inconvenient to display executed scenarios indefinitely. Therefore, a status file of a scenario is deleted at least 3days after it was started. If there are over 20 files, the older files are deleted until there are 20 (not including a currently running scenario). At most, 20 files whose scenarios were started in the past 3 days are kept. Note that this is not the case if over 20 scenarios are running.

If the scenario has been completed when a Benchmark screen is required from Synopsis screen, then redirect to the appropriate View Result Details screen. If the scenario is running, then display the detailed scenario, move the red execution bar to match the progress of the scenario and start. Note that this screen will open in a new window. For an ended or running scenario, open one window for one scenario; do not open a new window and access twice. There is no the limit set for scenario load and making new scenarios.

## 6.5. View Result

# DS-Bench



[Main](#)
[Synopsis](#)
[View Result](#)
[Configuration](#)

Search  in

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

<< 1 2 >>

Fig. 6-16 Synopsis screen

Display the history list of benchmark results that were executed in the past

- ID : Result ID (Appearing on the View Result screen)
- Data : Scenario start date and time
- Machines : Target machine name
- Comment : Comments and description of scenario
- Benchmark Program/Anomaly Load : benchmark/anomaly/fault injection that was used
- Tags : Configured tags are displayed. Add tags with the "add" button.

Searching can be done through the search-form at the top of the screen.

Click the red icon in any column and that displayed list is sorted.

Click the "add" button in the Tags and the Edit Tags dialog appears, allowing editing of Tags.

Put a check in the "Delete" column at the far right and click the "Delete" button to delete those benchmark results.

Click the "Check all" and the check marks appear in the "Delete" column boxes for all benchmark results. Also, click the "Clear all" button and the check marks disappear.

### 6.5.1. View Result Details

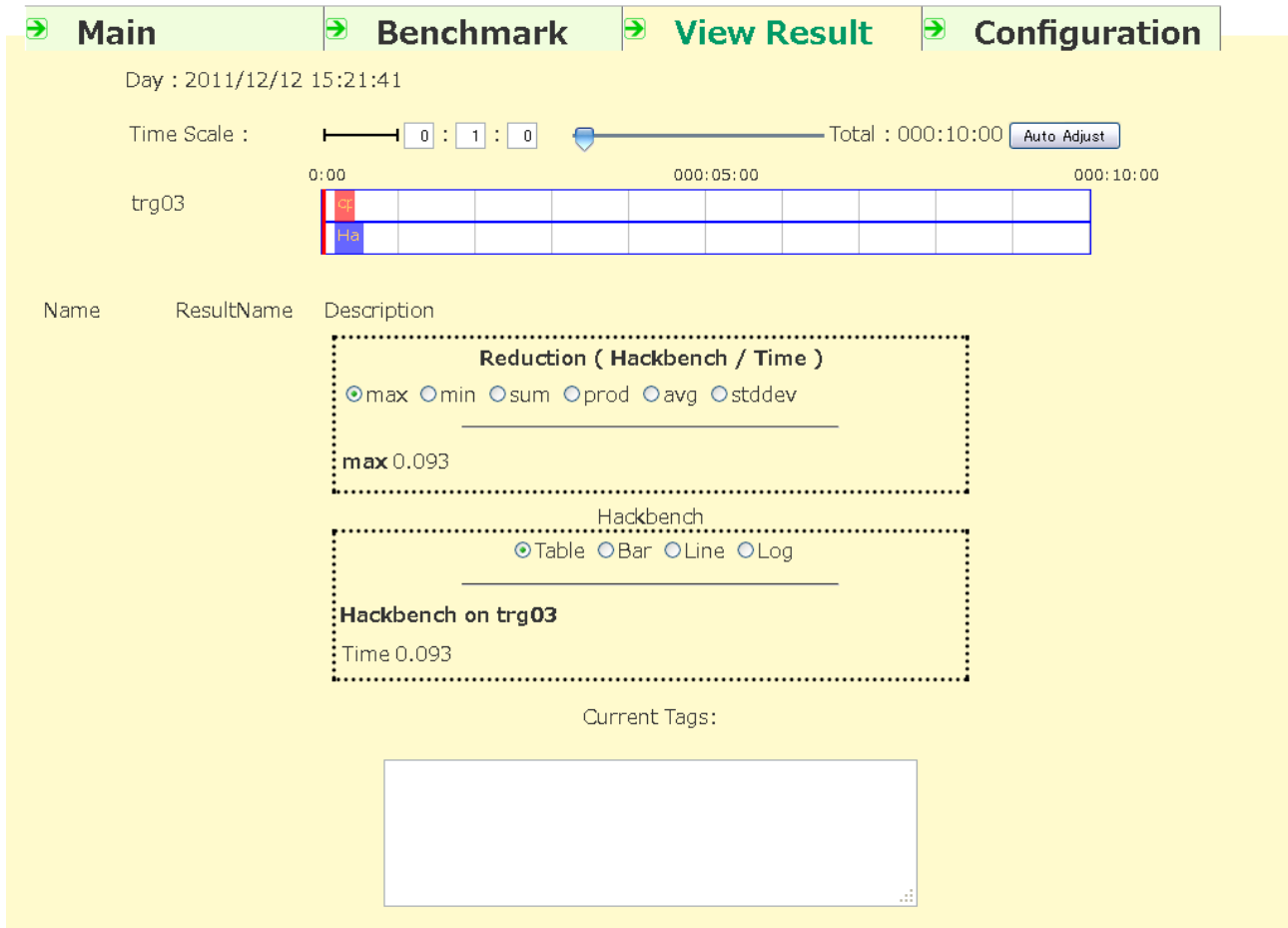


Fig. 6-17 View Result Details

Click the ID of the scenario from the View Result screen or choose the display (button/tag) in the D-Case Editor/D-Case Weaver then move to the benchmark execution result screen.

Display the benchmark results and the scenario details.

Click the "ReRun" button to move to the Benchmark screen, loading the same configuration that as the benchmark that was executed. Editing and execution can be carried out.

Click the "Delete" button and the displayed benchmark results will be deleted, after which the display moves to the View Result screen.

However, you cannot use the "ReRun" and "Delete" buttons if you display using D-Case Editor/D-Case Weaver. (Buttons not displayed)

## 6.6. Configuration screen

**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	
1	Bonnie	Bonnie++(IO benchmark program)	unit	<a href="#">Edit</a>
2	cpustress	CPU load tool	unit	<a href="#">Edit</a>
3	Hackbench	Hackbench(Load test)	unit	<a href="#">Edit</a>
4	httperf	httperf	unit	<a href="#">Edit</a>
5	IMB	Intel MPI Benchmarks	unit	<a href="#">Edit</a>
6	Iperf	Iperf(Network traffic benchmarks)	link	<a href="#">Edit</a>
7	LMBench	LMBench(Benchmarks suit)	unit	<a href="#">Edit</a>
8	Memstress	Memory load tool	unit	<a href="#">Edit</a>

Fig. 6-18 Configuration screen

Display the sub-menu on the left side of the Configuration screen.

"Benchmark": displays the Benchmark Management screen (display the initial Configuration screen)

"Setting": displays the time scale setting screen of DS-Bench

"Target list": displays Target list screen.

"Sedna": displays the Sedna setting screen

"D-Cloud": displays the D-Cloud setting screen

"Benchmark scenario": displays the benchmark scenario list



### 6.6.1. Benchmark

Displays the list of the benchmark programs comprising Benchmark Management.

Click the "Edit" anchor to display the Benchmark Edit dialog.

Benchmark Edit
✕

Benchmark Name :

Description :

Path :

Type :

Start time :  - Finish time :

Interval Before Starting :

Interval After Termination :

Client Input :

Label :  -Arg :  -Ui :  -Interactive :

-Value :

Label :  -Arg :  -Ui :  -Interactive :

-Value :

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+
\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>
                    
```

Fig. 6-19 Benchmark Edit dialog

Edit the information in the various benchmark programs. Contents edited here are reflected in the benchmark configuration file, conf-file.

Display the name of the benchmark to be edited in "Benchmark Name"

"Description" displays the description of the benchmark to be edited.

"Path" displays the path of the benchmark to be edited to where it is to be deployed.

「Client Input」 sets the default values and adds parameters to the benchmark.

To add items, use the 「add」 button. Set the display name in the "Label" box, set options in the "-Arg" box, selecting No|Yes in "-Ui" to display or hide , and input default values in "Value" boxes

Copy and paste an example of a log output from the benchmark in the "Source log".

Enter the rules for log cutting in "Output style" and then click the "Test" button to process the source log according to the Output style. The results are given in "Result".

Click the "Delete" button to remove the Benchmark whose Edit anchor has been clicked.

If the "Modify" button is clicked when there are no particular error elements, the display moves to Benchmark Management.

### 6.6.2. Setting



Fig. 6-20 Setting

Click the "Setting" to display the screen for setting the initial value of the benchmark time scale.

"Time Scale" is used as the initial value of the Time Scale when you display the initial screen of Benchmark. The boxes are hours, minutes, and seconds, from left to right.

Click the "Register" button to record the input value in the configuration file.

Click the "Reset" button to set the input value to that before it was changed.

Click the "Back" button to move to the previous screen and discard the input value.

### 6.6.3. Target list

**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

Fig. 6-21 Target list

Click the "Target list" to display the list of targets for D-Cloud controller to manage and their status .  
 When the status is "In Use", it displays the time that the machine user will release the machine.  
 Click the "Refresh" button to check and display the computational resources again.  
 Show the details of a target by clicking in the "Machine name" column.

#### 6.6.3.1. Target detail

**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		

Fig. 6-22 Configuration screen

Displays a table showing target information in the <dsb\_resource\_desc> tag.

Click the "Close" button then move to the Target list.

#### 6.6.4. Sedna

Sedna	
Sedna host	<input type="text" value="127.0.0.1"/>
Sedna DB name	<input type="text" value="dsbench"/>
Sedna DB user	<input type="text" value="SYSTEM"/>
Sedna DB password	<input type="password" value="....."/>
Result benchmark	<input type="text" value="result_benchmark"/>
Result benchmark scenario	<input type="text" value="result_benchmark_set"/>

Register Reset Back

Fig. 6-23 Sedna

Click the "Sedna" to display the environment setting input screen to connect to Sedna XML DB.  
 After "Sedna host", specify the host name or IP address of the Sedna XML Database.  
 After "Sedna DB name", specify the connection database name of Sedna XML Database.  
 After "Sedna DB user", specify the name of the connecting user of Sedna XML Database.  
 After "Sedna DB password", specify the password for the connecting user of Sedna XML Database.  
 After "Result benchmark", specify the storage destination of benchmark results.  
 After "Result benchmark scenario", set the storage location of the benchmark scenario.  
 Click the "Register" button to record the input values in the configuration file.  
 Click the "Reset" button then set input values to those before they were changed.  
 Click the "Back" button then move to the previous screen and discard the input values.

#### 6.6.5. D-Cloud

D-Cloud	
D-Cloud controller	<input type="text" value="192.168.11.133"/>
D-Cloud port	<input type="text" value="9903"/>

Register Reset Back

Fig. 6-24 D-Cloud

Click "D-Cloud" to display the environment setting input screen to connect to D-Cloud controller .  
 After "D-Cloud controller", specify the host name or IP address of the D-Cloud controller.  
 After "D-Cloud port", specify the connection port to the D-Cloud controller.  
 Click the "Register" button to record the input values in the configuration file.  
 Click the "Reset" button to set input values to those before they were changed.  
 Click the "Back" button to move to the previous screen and discard the input values.

6.6.6. Benchmark scenario

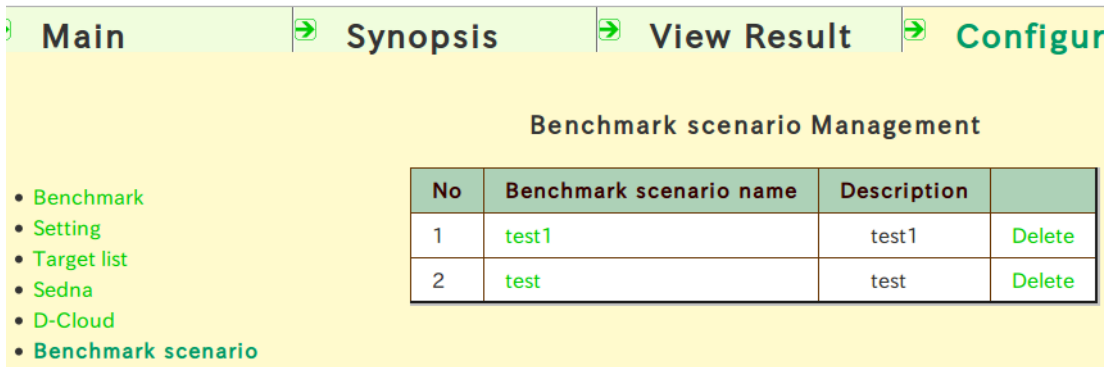


Fig. 6-25 Benchmark scenario Management

In this screen, you can edit, display in a list, and delete the benchmark scenarios that are registered by users.

When the "Delete" is clicked, a confirmation dialog box to confirm deletion of that benchmark scenario is displayed.

You can delete the benchmark scenario by clicking the "OK".

**List of tools in DS-Bench**

-----Supported-----

- NetCMD
- Memstress
- Httpperf
- cpustress
- bonnie
- terminator
- Hackbench
- Iperf
- NPB

-----Unsupported-----

- MPD
- LMBench
- IMB
- MemoryFlip
- NICFault
- MachineStop