

# DS-Bench Toolset: Tools for Dependability Benchmarking with Simulation and Assurance

Dependable Systems and Networks (DSN 2012), Boston, MA, USA

Hajime Fujita (U. of Tokyo, Japan)[\*1]

Yutaka Matsuno (U. of Tokyo, Japan) [\*2]

Toshihiro Hanawa (U. of Tsukuba, Japan)

Mitsuhisa Sato (U. of Tsukuba, Japan)

Shinpei Kato (UC Santa Cruz, USA) [\*2]

Yutaka Ishikawa (U. of Tokyo, Japan)

[\*1] Currently in the University of Chicago, USA

[\*2] Currently in Nagoya University, Japan

This work has been conducted as a part of the “Dependable Embedded Operating System for Practical Use (DEOS)” project, funded by JST (Japan Science and Technology Agency).

# Introduction (1)

---

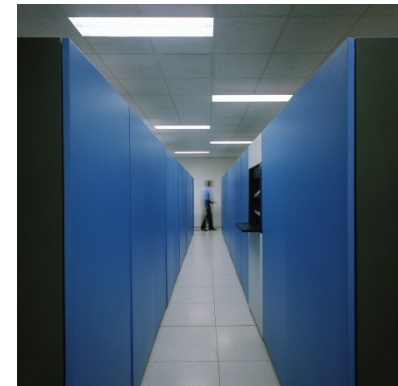
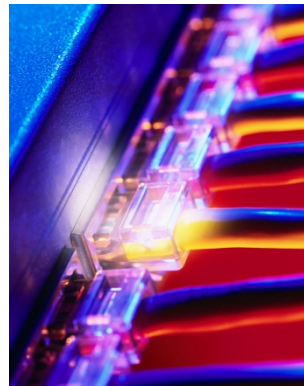
- ▶ Increasing demands for highly dependable system
  - ▶ Today's society totally depends on information systems, and suspension of services cost a lot
- ▶ A complex system involves a lot of stakeholders (e.g. developer, supplier, user, etc...)
- ▶ All of them must agree on what is the “dependability” for their system, e.g. minimum throughput or maximum latency
  - ▶ The dependability of the system should be expressed clearly and supported by clear evidences so that every stakeholder agrees that the system is in fact dependable



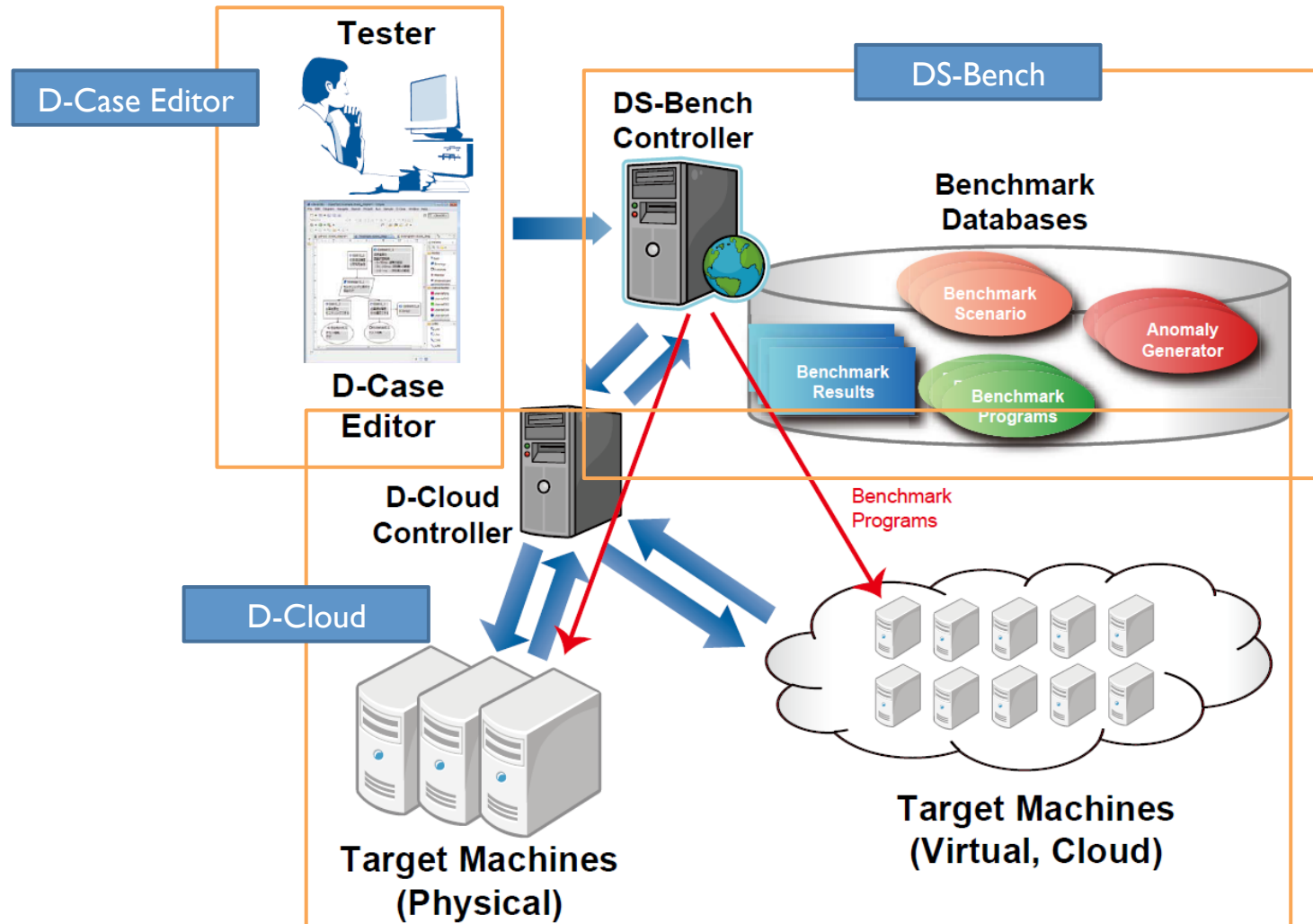
## Introduction (2)

---

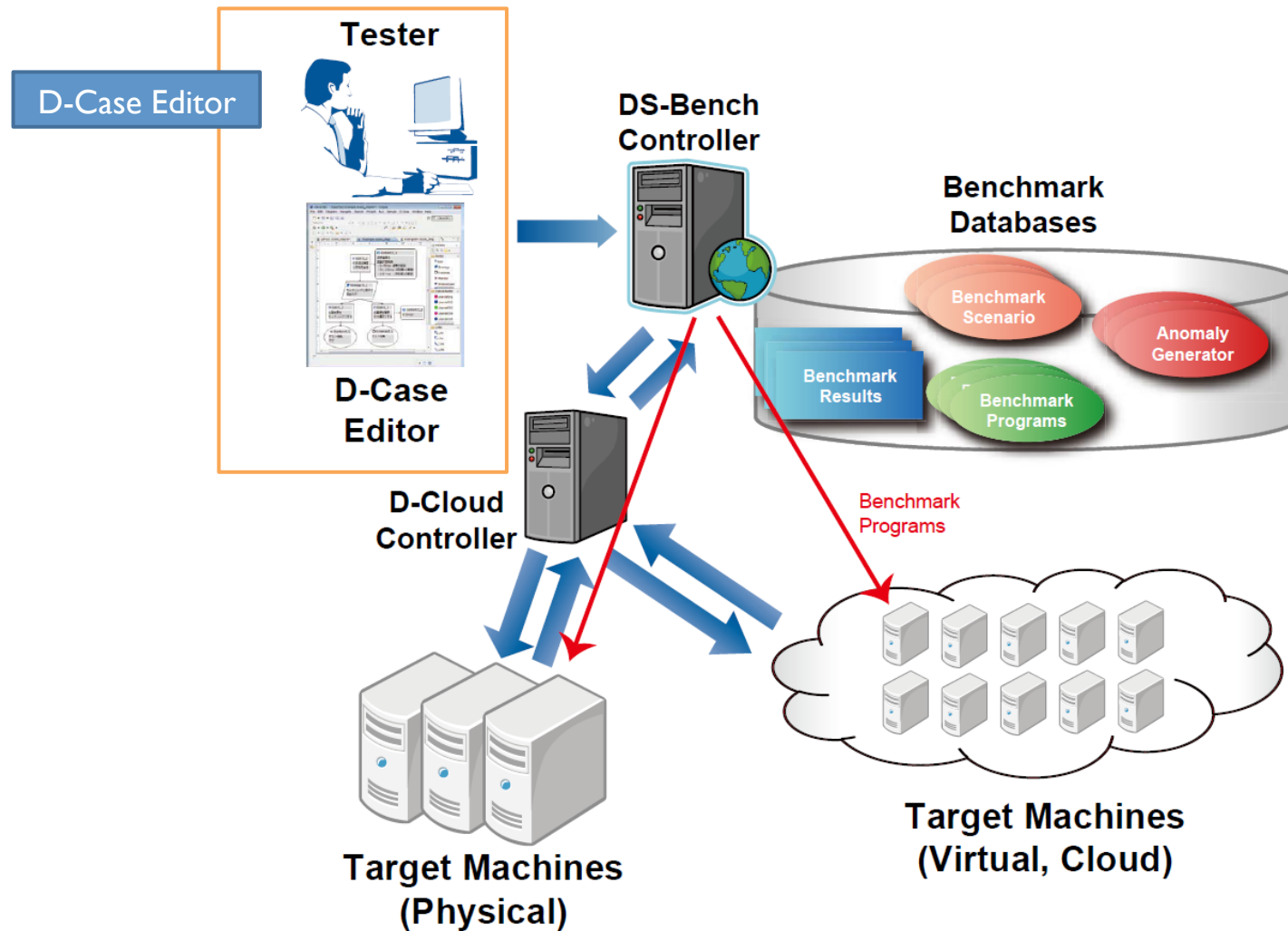
- ▶ Some argument on system's dependability may require quantitative evaluation of the system
- ▶ An automated benchmark testing tool is needed
  - ▶ Systems are getting more parallel and distributed
  - ▶ Testing takes much time and cost
- ▶ Results of the tests should be automatically collected as **evidences of dependability**



# DS-Bench Toolset: Overview



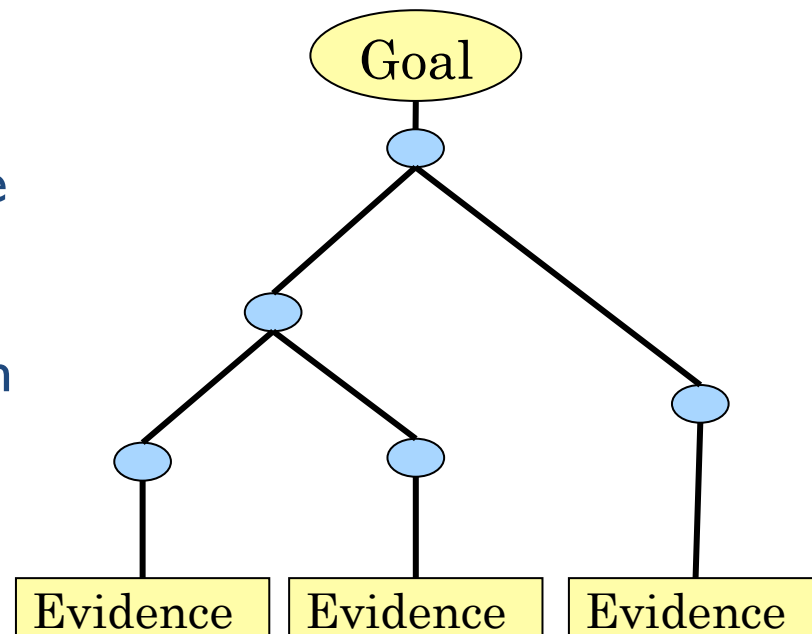
# D-Case Editor



## D-Case [Matsuno PRDC2010]

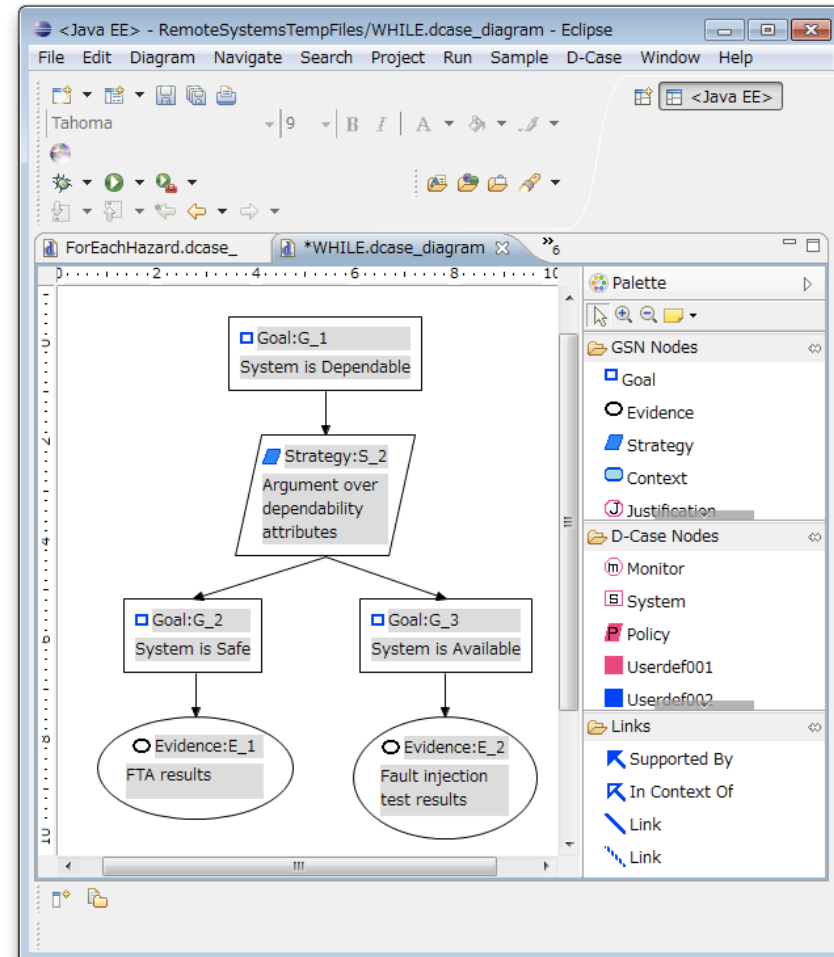
---

- ▶ A kind of Assurance Cases with development tools and runtime monitoring systems
- ▶ Assurance Cases
  - ▶ “A documented body of evidence that provides a convincing and valid argument that a system is adequately dependable for a given application in a given environment” [Adelard]
  - ▶ Becoming a standard for safety-critical systems
  - ▶ A Graphical notations GSN (Goal Structuring Notation)

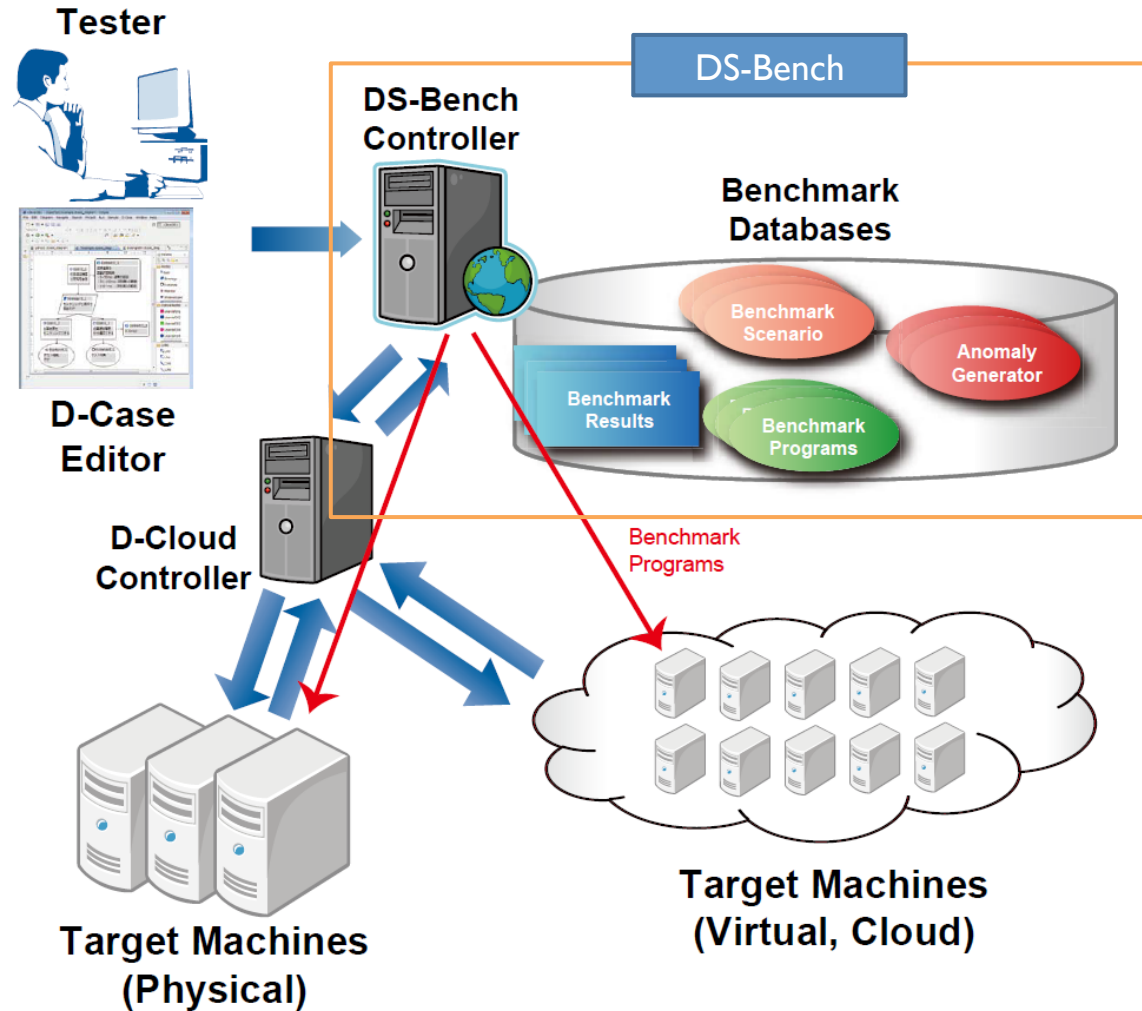


# D-Case Editor

- ▶ A free assurance case editor
  - ▶ An Eclipse plugin using Eclipse GMF
  - ▶ Supports GSN
- ▶ Key Features
  - ▶ Variable type checking and pattern library [Matsuno QSIC2011]
  - ▶ Conducting benchmark tests using DS-Bench



# DS-Bench

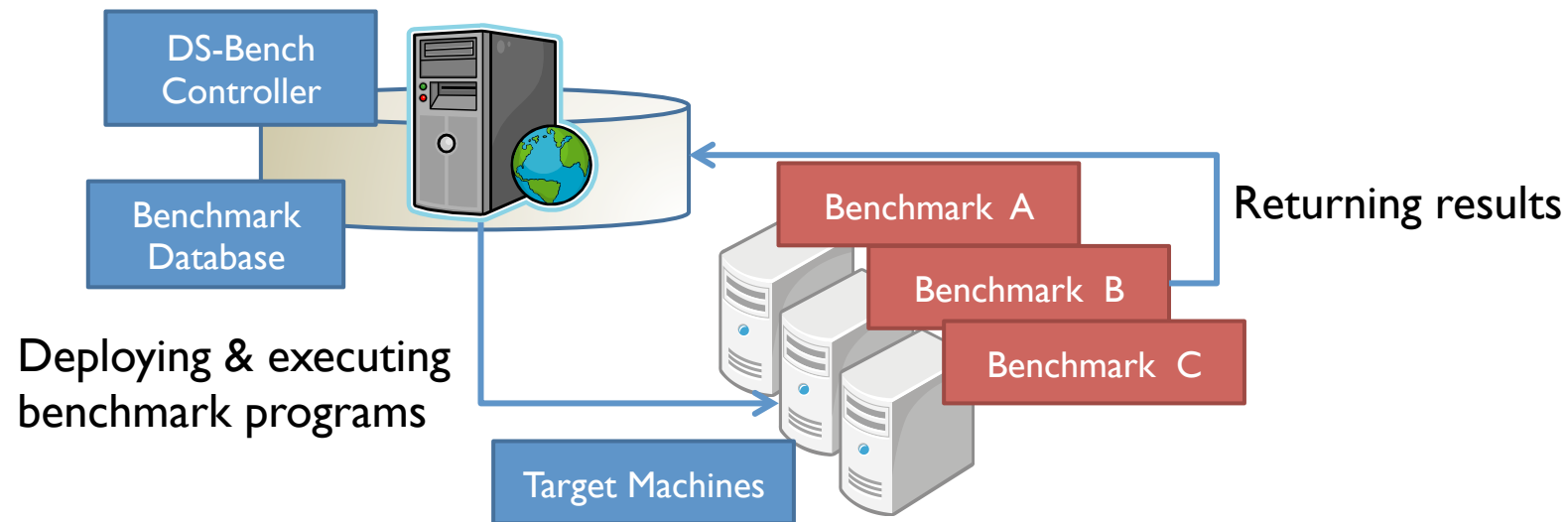




# DS-Bench: Overview

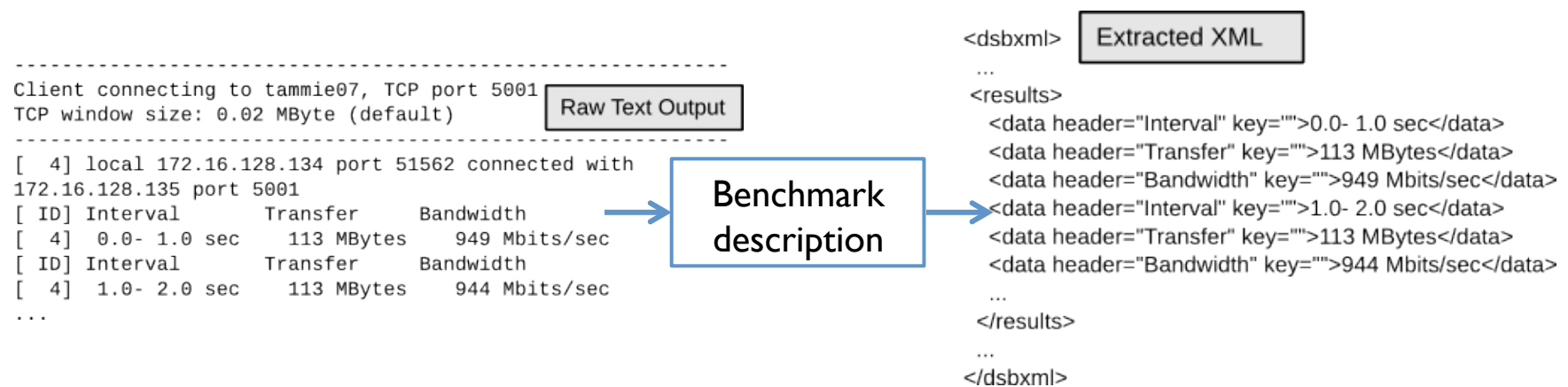
---

- ▶ A framework for automatic benchmark tests
  - ▶ Dependability metrics are defined and measured by each program. DS-Bench itself does not define them.
- ▶ Supports multiple benchmark programs
  - ▶ Existing benchmark programs, as well as user-developed ones, can be executed on DS-Bench



# DS-Bench: Benchmark Results

- ▶ Handling outputs from various benchmark programs
  - ▶ Output style of benchmark programs may vary
  - ▶ Usually the output is pre-formatted for human readability
- ▶ Benchmark description is prepared for each benchmark program
  - ▶ Describes cutting rules for interpreting the raw result so that a text table can be converted to a list of machine readable values
- ▶ Converted results are stored in an XML database



# DS-Bench: Anomaly Loads

---

- ▶ Anomaly loads simulate several irregular situations that may occur to the system
  - ▶ E.g. Whole machine failure, device failure, performance degradation, ...
- ▶ Essential for dependability benchmark testing
  - ▶ We want to know if the system is still dependable under such conditions

# DS-Bench: Pre-installed Programs

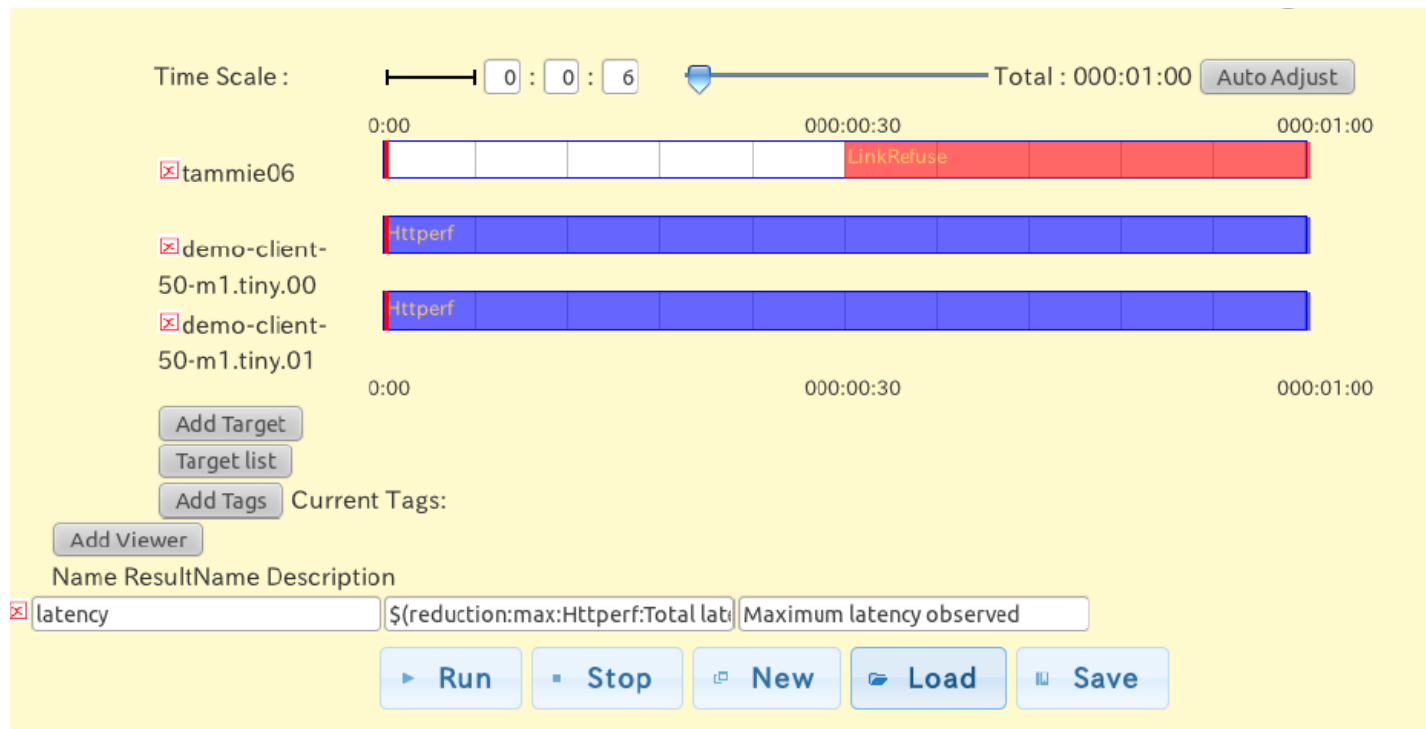
Name	Description
bonnie++	I/O benchmark
lmbench	Generic performance benchmark
hackbench	System benchmark; creates a lot of processes
httperf	Measures performance of HTTP servers
iperf	Measures network bandwidth
cpustress	Stresses CPU; just consumes CPU time
memstress	Stresses memory; just consumes memory
netcmd	Injecting network anomalies; delay, packet drop, reordering
terminator	Kills a process

Each program may be used as a benchmark program or anomaly generator. Some of them may be used as both.

e.g. bonnie++ can be used as a benchmark program to measure I/O bandwidth, as well as an anomaly generator that consumes I/O bandwidth.

# DS-Bench: Benchmark Scenario (1)

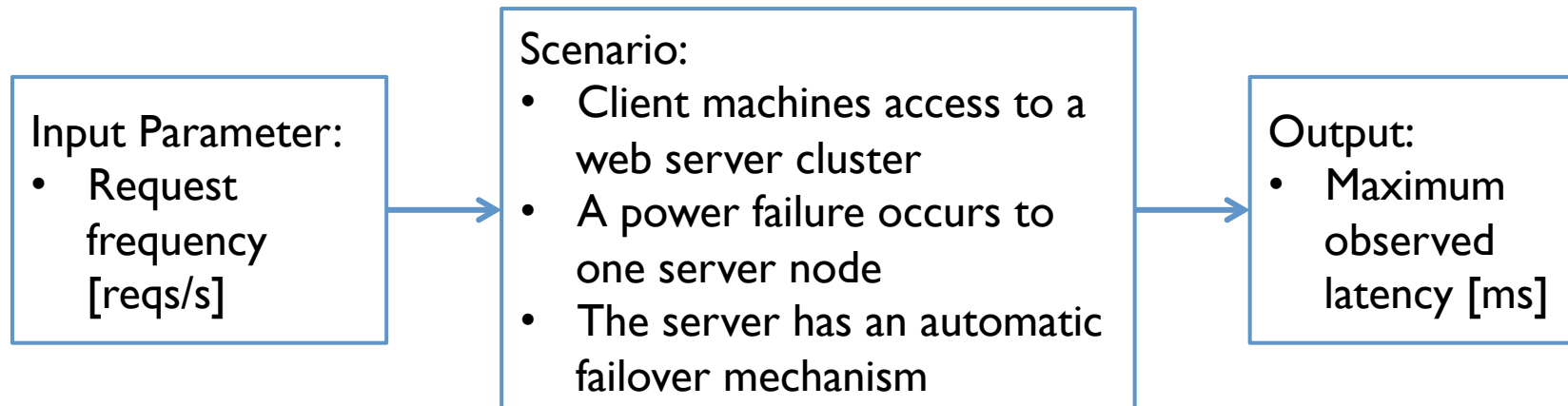
- ▶ Benchmark programs and anomaly loads are executed concurrently in a specific timing



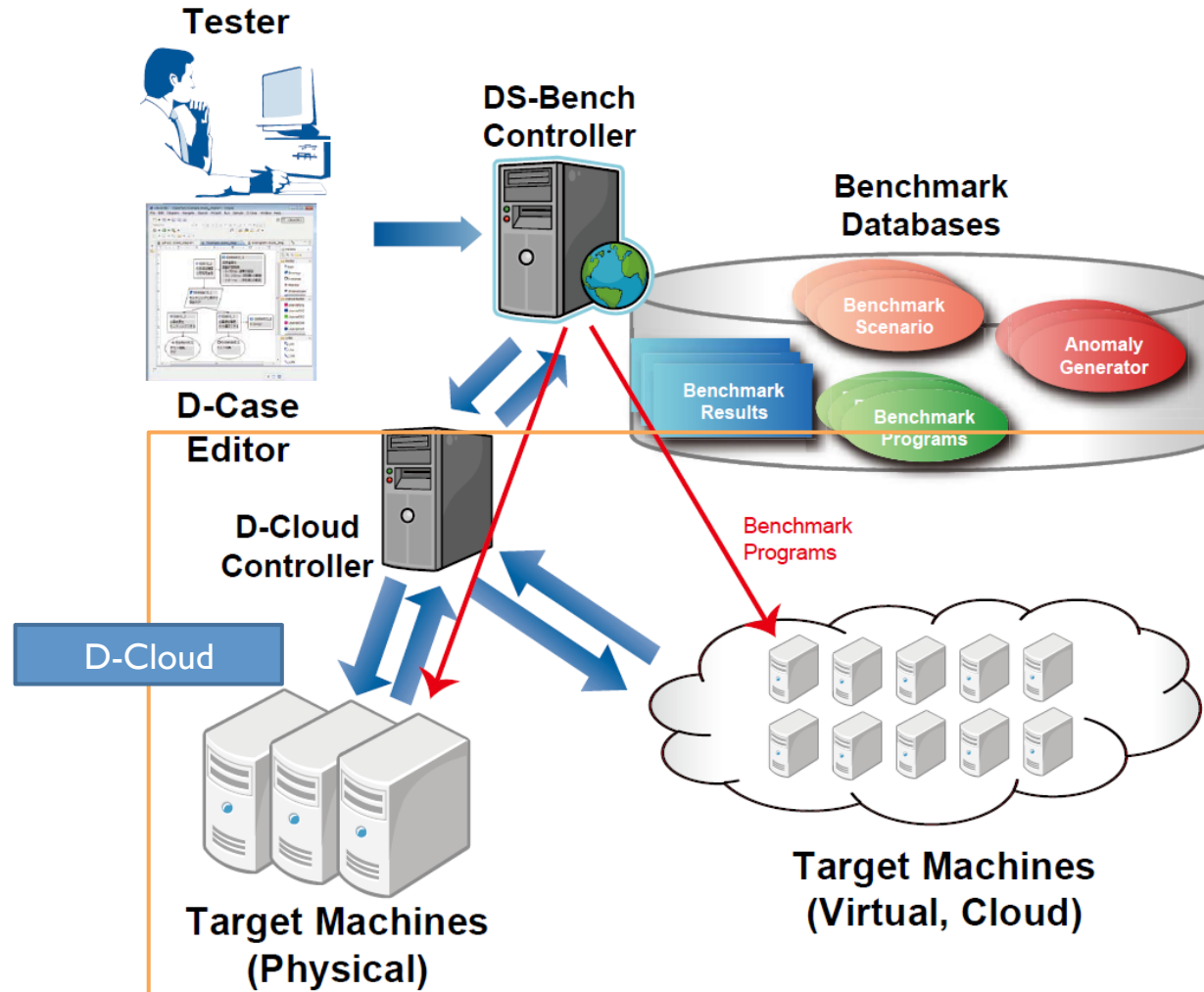
## DS-Bench: Benchmark Scenario (2)

---

- ▶ Each scenario may define input parameters and an output result
  - ▶ Input parameters are adjustable from D-Case Editor
  - ▶ An output result can be obtained from D-Case Editor
- ▶ Example:



# D-Cloud



# D-Cloud: Overview

---

- ▶ D-Cloud manages hardware resources needed for conducting benchmark tests
- ▶ Two types of computing resources are provided
  - ▶ **Physical machines**
    - ▶ For performance-sensitive tests
  - ▶ **Virtual machines**
    - ▶ Managed by OpenStack, a management software for private clouds
    - ▶ An arbitrary number of virtual machines can be created simultaneously
    - ▶ For functional, performance-insensitive tests



# D-Cloud: Fault Injection

---

- ▶ D-Cloud also performs fault injection
  - ▶ Adding anomalies from the outside of target machines
  - ▶ Mostly requires special equipment (e.g. intelligent PDUs, IPMI, SNMP-enabled switch), or special software (e.g.VMM)
- ▶ **FaultVM** [Banzai CLOUD2010][Hanawa PRDC2010]
  - ▶ A virtual machine monitor that comes with a hardware failure simulation feature
  - ▶ Based on QEMU, an open source full system emulator
- ▶ Fault injection functions are exported to DS-Bench and can be used as anomaly generators
  - ▶ i.e. These fault injectors can be put in a benchmark scenario

# D-Cloud: List of Fault Injection Methods

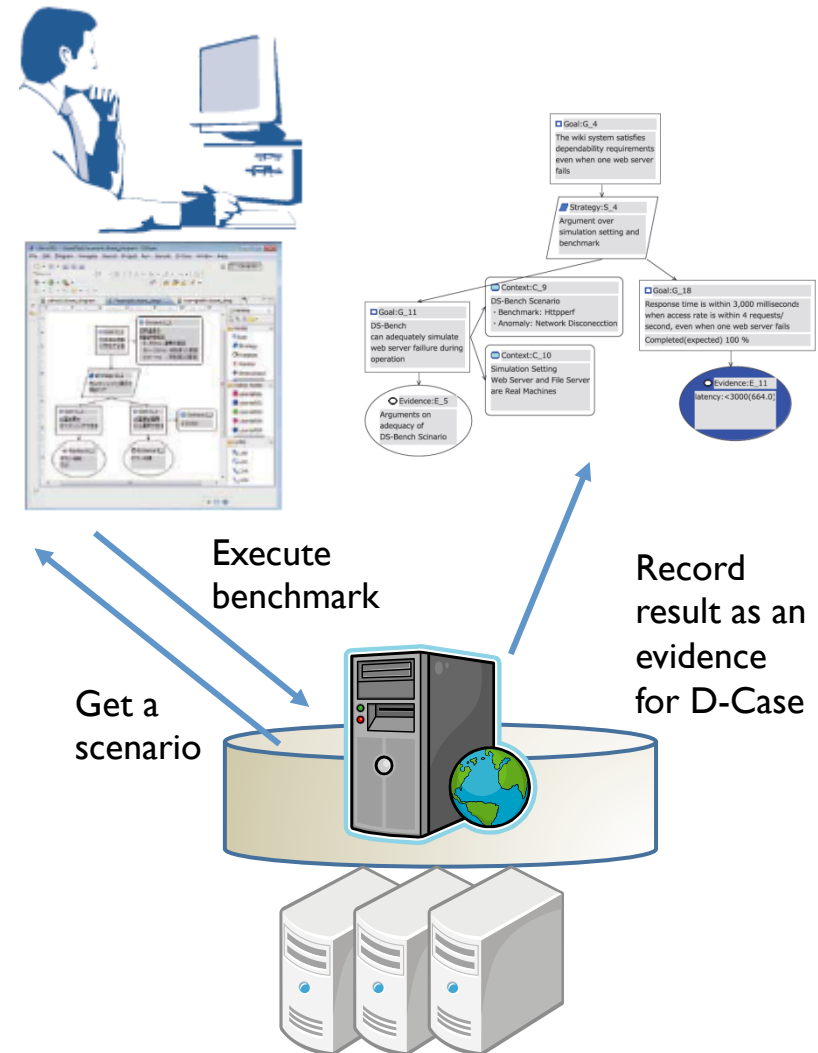
---

Name	Target	Description
LinkRefuse	Physical Machine	Shuts down a network switch port
SupplyRefuse	Physical Machine	Shuts down a power supply to a target
IPMICtrl	Physical Machine	Controls a power status of a target via IPMI
VMMemFlip	Virtual Machine	Injects memory flip into VM
VMStop	Virtual Machine	Pauses a VM
VMNicFault	Virtual Machine	Injects a various fault into VM's virtual NIC device

Fault injection methods for physical machines are implemented as a program, and can be added by users just as benchmark programs

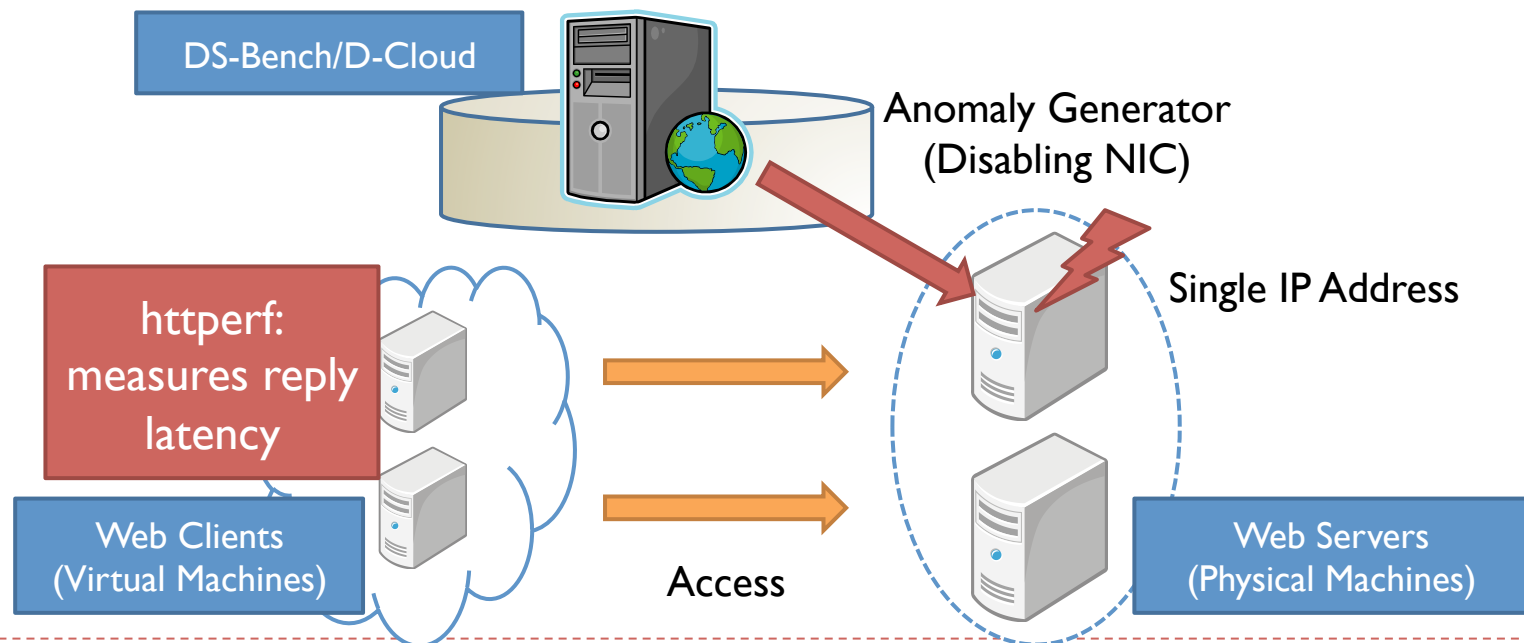
# Typical Workflow with the Toolset

1. Determine the required dependability metrics and describe it in D-Case
2. Find a suitable benchmark scenario in DS-Bench database
  - ▶ If not exist, create new one
3. Adjust parameters and run the benchmark test from D-Case Editor
4. Result is returned to D-Case Editor and recorded as an evidence for D-Case

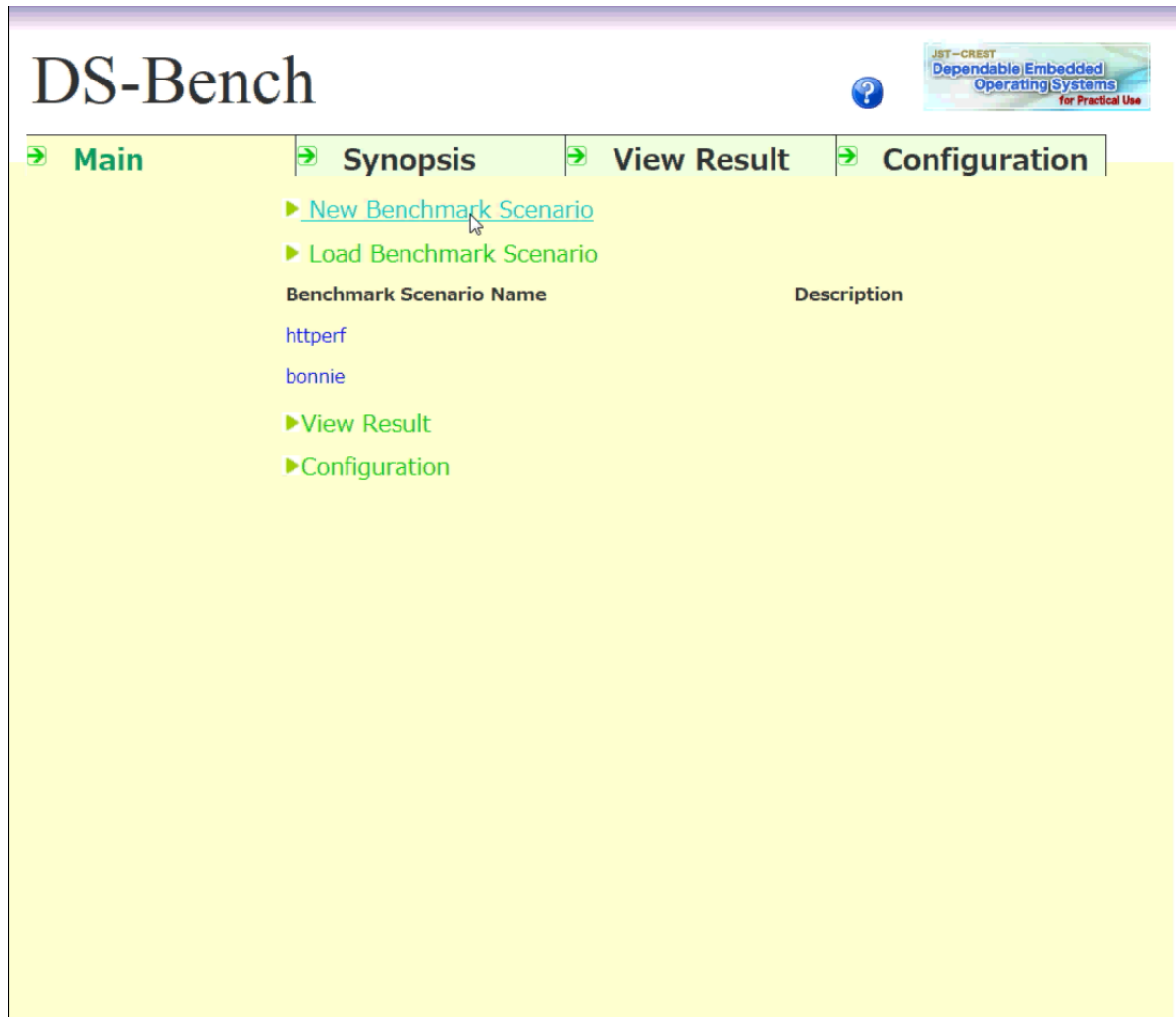


# Demonstration

- ▶ Evaluating a web server system
- ▶ Dependability requirement
  - ▶ Keep the access latency lower than 3s even one server node fails
- ▶ Environment
  - ▶ The server provides a single system view using SSPA, a load balancing/high availability mechanism [Fujita HASE2011]



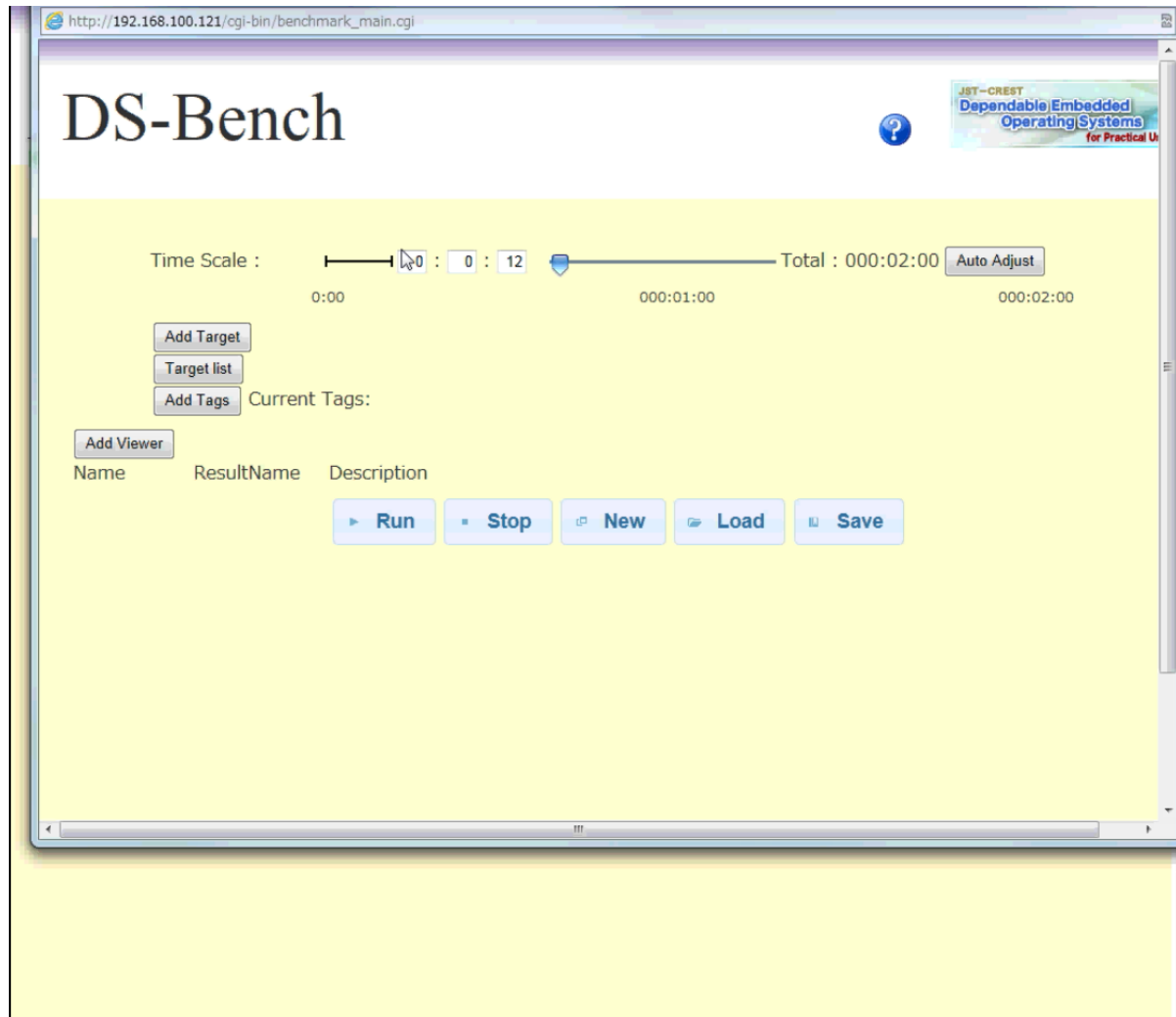
# Demo: Creating a New Scenario



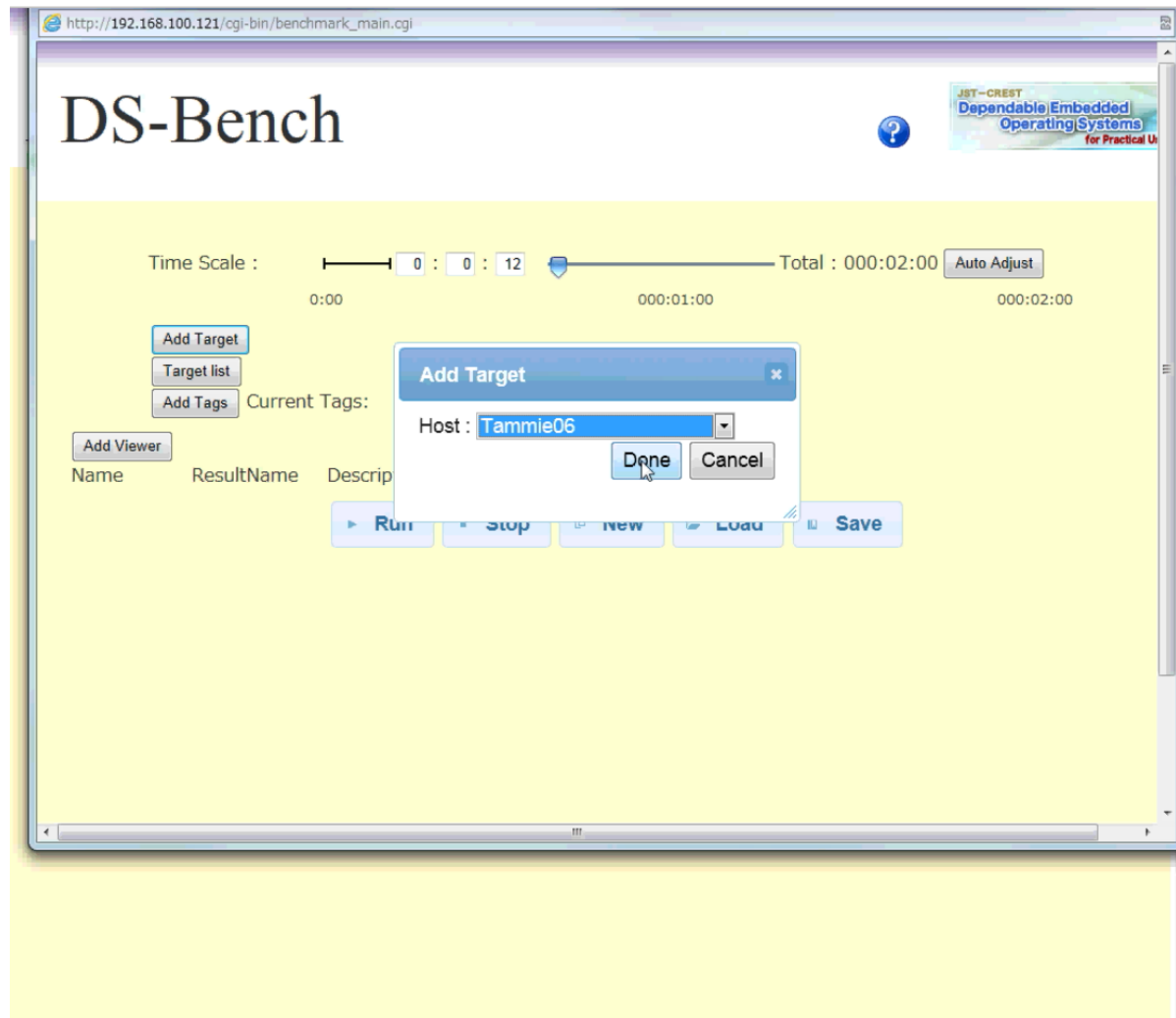
The screenshot shows the DS-Bench web application interface. At the top left, the title "DS-Bench" is displayed. To the right of the title is a help icon (a question mark in a blue circle) and a logo for "JST-CREST Dependable Embedded Operating Systems for Practical Use". Below the title is a navigation menu with four items: "Main", "Synopsis", "View Result", and "Configuration", each with a right-pointing arrow icon. The "Main" item is currently selected and highlighted in green. Below the navigation menu, there are two expandable menu items: "▶ New Benchmark Scenario" (with a blue underline) and "▶ Load Benchmark Scenario". Below these are two columns: "Benchmark Scenario Name" and "Description". Under "Benchmark Scenario Name", there are two entries: "httperf" and "bonnie". Below the table, there are two more expandable menu items: "▶ View Result" and "▶ Configuration".

Benchmark Scenario Name	Description
httperf	
bonnie	

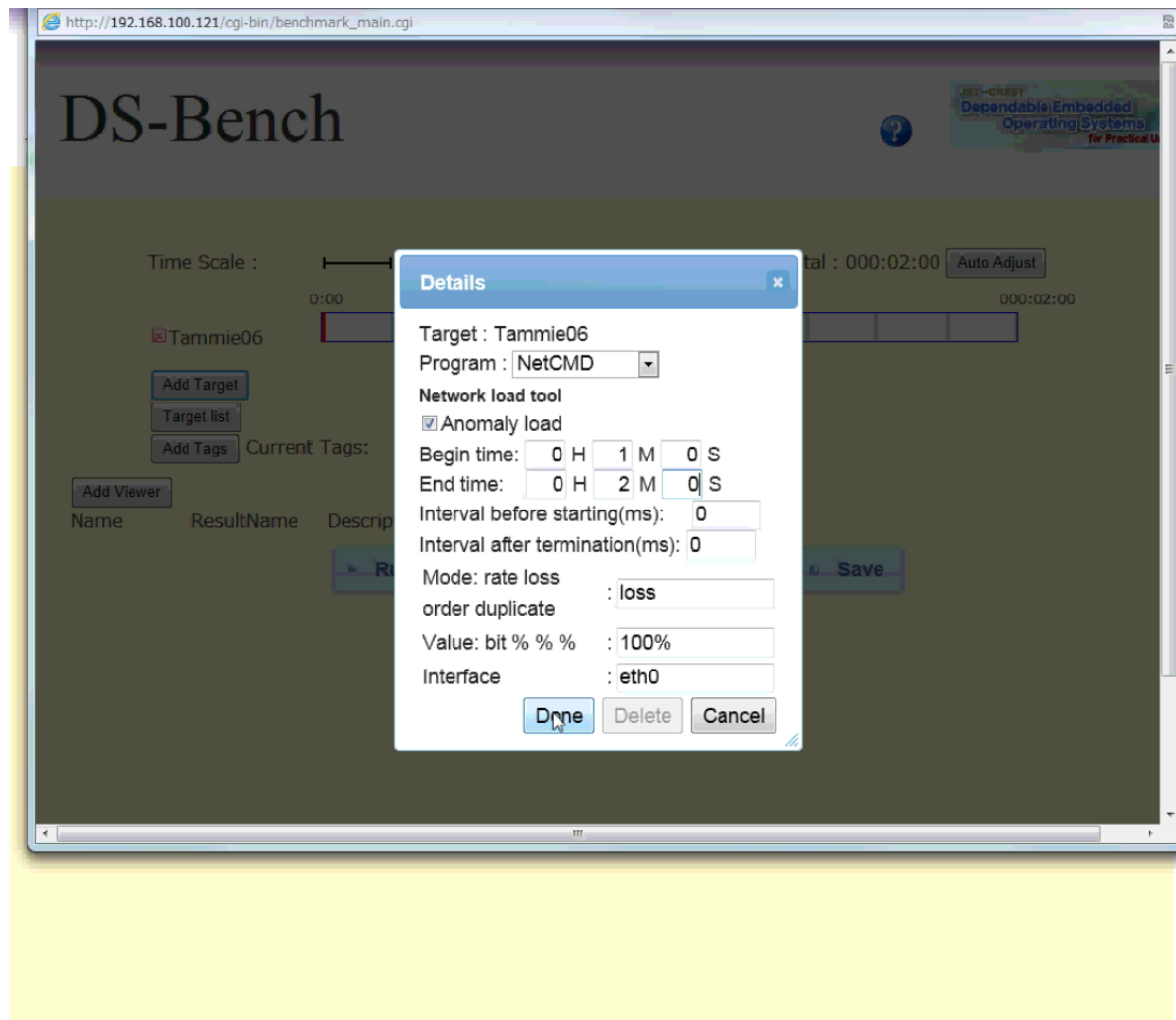
# Demo: Creating a New Scenario



# Demo: Creating a New Scenario



# Demo: Creating a New Scenario





# Demo: Creating a New Scenario

The screenshot shows the DS-Bench web interface. At the top, the URL is `http://192.168.100.121/cgi-bin/benchmark_main.cgi`. The main heading is "DS-Bench". In the top right corner, there is a logo for "JST-CREST Dependable Embedded Operating Systems for Practical U".

The interface features a "Time Scale" section with a slider set to 0:0:12 and a "Total" of 000:02:00. Below this is a timeline visualization for a scenario named "Tammie06". The timeline shows a red bar labeled "NetCMD" starting at 000:01:00 and ending at 000:02:00. A mouse cursor is hovering over the "Add Target" button.

Below the timeline, there are several buttons: "Add Target", "Target list", "Add Tags", "Add Viewer", "Run", "Stop", "New", "Load", and "Save". The "Current Tags:" label is positioned to the right of the "Add Tags" button.

Name	ResultName	Description
------	------------	-------------

# Demo: Creating a New Scenario

The screenshot shows the DS-Bench web interface. At the top, the URL is `http://192.168.100.121/cgi-bin/benchmark_main.cgi`. The page title is "DS-Bench". There is a logo for "JST-CREST Dependable Embedded Operating Systems for Practical U".

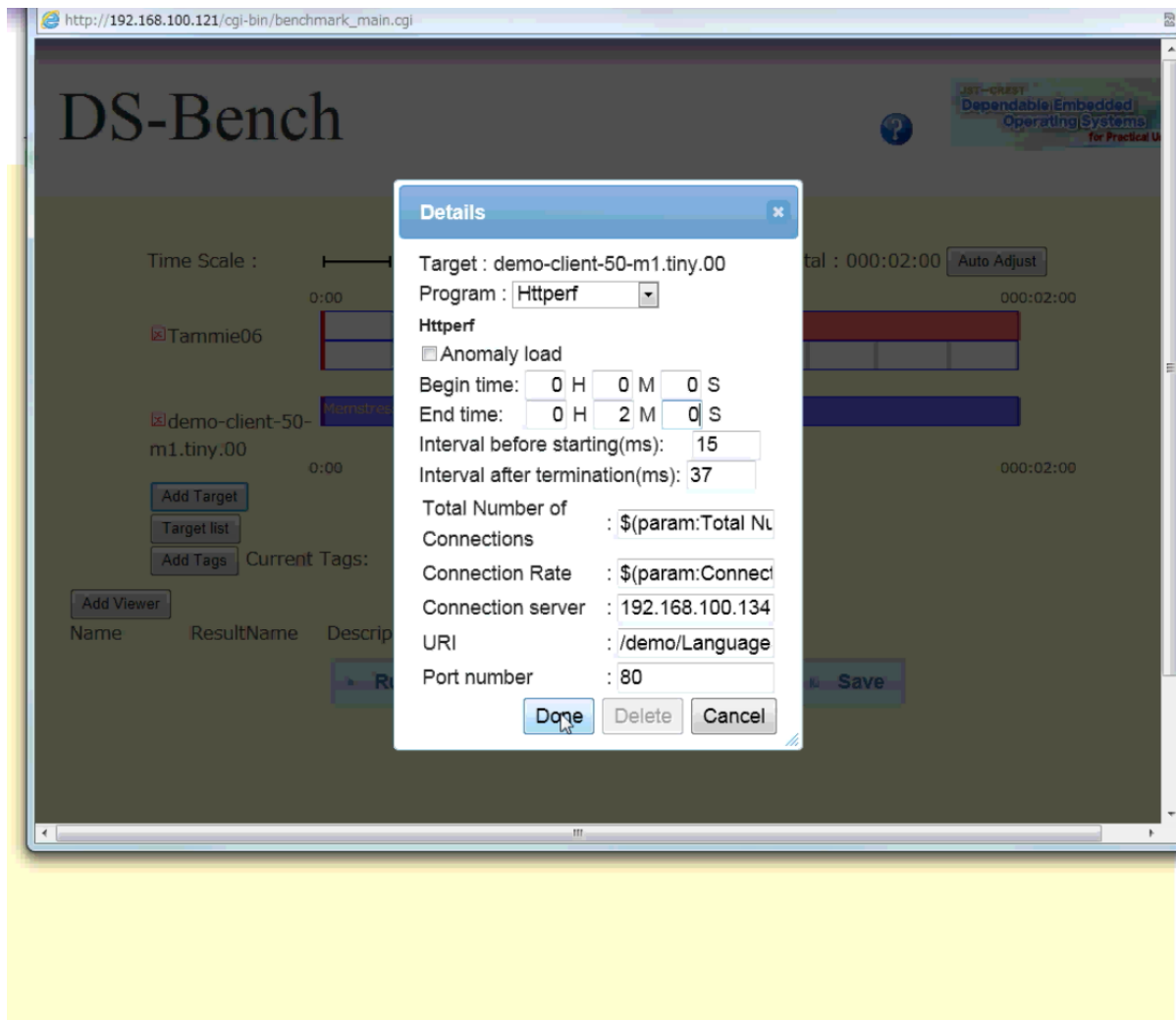
The main configuration area includes a "Time Scale" slider set to 0:0:12, with a "Total" of 000:02:00 and an "Auto Adjust" button. Below this is a timeline visualization for a scenario named "Tammie06", showing a red bar for "NetCMD" running from 0:00 to approximately 0:01:00.

An "Add Target" modal dialog is open, showing a dropdown menu for "Host" with the selected value "demo-client-50-m1.tiny.00". The dialog has "Done" and "Cancel" buttons.

On the left side, there are buttons for "Add Target", "Target list", "Add Tags", and "Add Viewer". Below these is a "Current Tags:" label.

At the bottom, there is a table with columns "Name", "ResultName", and "Description". Below the table are buttons for "Run", "Stop", "New", "Load", and "Save".

# Demo: Creating a New Scenario



# Demo: Creating a New Scenario

The screenshot shows the DS-Bench web interface. At the top, the URL is `http://192.168.100.121/cgi-bin/benchmark_main.cgi`. The main heading is "DS-Bench". A logo for "JST-CREST Dependable Embedded Operating Systems for Practical U" is visible in the top right.

The "Time Scale" section features a slider set to 0:0:12, with a "Total : 000:02:00" and an "Auto Adjust" button. Below this, two target configurations are shown:

- Tammie06: A red bar representing "NetCMD" running from 0:00 to 000:01:00.
- demo-client-50-m1.tiny.00: A blue bar representing "Httpperf" running from 0:00 to 000:02:00.

Control buttons include "Add Target", "Target list", "Add Tags", "Add Viewer", "Run", "Stop", "New", "Load", and "Save". The "New" button is highlighted with a mouse cursor.

Name	ResultName	Description
Tammie06	NetCMD	
demo-client-50-m1.tiny.00	Httpperf	

# Demo: Creating a New Scenario

The screenshot shows the DS-Bench web interface. At the top, the browser address bar displays `http://192.168.100.121/cgi-bin/benchmark_main.cgi`. The page title is "DS-Bench". In the top right corner, there is a logo for "JST-CREST Dependable Embedded Operating Systems for Practical U...".

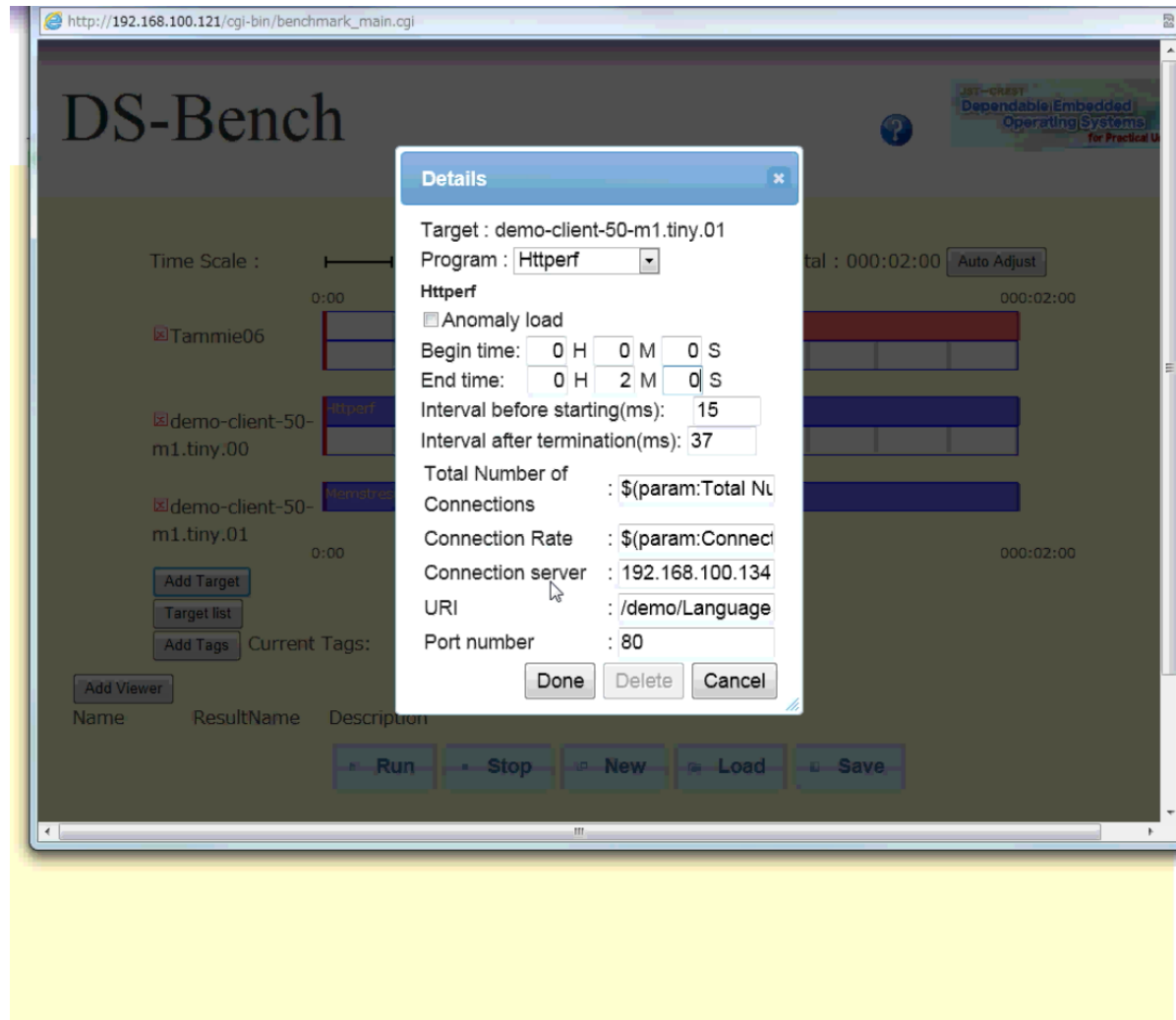
The main configuration area has a yellow background. At the top, there is a "Time Scale" section with a slider set to 0:0:12 and a "Total : 000:02:00" label, along with an "Auto Adjust" button. Below this is a timeline from 0:00 to 000:02:00. Two targets are listed:

- Tammie06: A red bar representing a "NetCMD" target.
- demo-client-50-m1.tiny.00: A blue bar representing a "Hitperf" target.

An "Add Target" dialog box is open in the center, showing "Host : demo-client-50-m1.tiny.01" and "Done" and "Cancel" buttons. Below the targets, there are buttons for "Add Target", "Target list", "Add Tags", and "Add Viewer". A "Current Tags:" label is also present.

At the bottom, there is a table with columns "Name", "ResultName", and "Description". Below the table are buttons for "Run", "Stop", "New", "Load", and "Save".

# Demo: Creating a New Scenario



# Demo: Creating a New Scenario

Time Scale : 0 : 0 : 12 Total : 000:02:00 Auto Adjust

0:00 000:01:00 000:02:00

Tammie06 NetCMD

demo-client-50-m1.tiny.00 Httpperf

demo-client-50-m1.tiny.01 Httpperf

0:00 000:01:00 000:02:00

Add Target Target list Add Tags Current Tags:

Add Viewer

Name	ResultName	Description
<input checked="" type="checkbox"/> latency	uction:max:Httpperf:Total	latency

Run Stop New Load Save

# Demo: Creating a New Scenario

The screenshot shows a web-based benchmarking tool interface. At the top, the URL is `http://192.168.100.121/cgi-bin/benchmark_main.cgi`. The main area features a timeline with a 'Time Scale' slider set to 12, a 'Total' of 000:02:00, and an 'Auto Adjust' button. Three scenarios are listed: 'Tammie06', 'demo-client-50-m1.tiny.00', and 'demo-client-50-m1.tiny.01'. A 'Save Benchmark Scenario' dialog box is open in the center, with 'Benchmark Scenario Name' set to 'demo' and 'Description' empty. Below the dialog are buttons for 'Add Target', 'Target list', 'Add Tags', 'Add Viewer', and 'Current Tags:'. At the bottom, there is a table with columns 'Name', 'ResultName', and 'Description', containing one row: 'latency', 'uction:max:Httpperf.Total', 'latency'. At the very bottom are buttons for 'Run', 'Stop', 'New', 'Load', and 'Save'.



# Demo: Creating a New Scenario

The screenshot shows a web-based interface for configuring a benchmarking scenario. At the top, there is a 'Time Scale' section with a slider set to 0:0:12 and a 'Total' time of 000:02:00. Below this, three targets are listed: 'Tammie06' (NetCMD), 'demo-client-50-m1.tiny.00' (Httpperf), and 'demo-client-50-m1.tiny.01' (Httpperf). A 'Dialog' box with the message 'Saved.' and an 'OK' button is overlaid on the interface. On the left side, there are buttons for 'Add Target', 'Target list', 'Add Tags', and 'Add Viewer'. Below these buttons is a table with columns 'Name', 'ResultName', and 'Description'. The table contains one entry: 'latency' with 'uction:max:Httpperf:Total' as the result name and 'latency' as the description. At the bottom, there are buttons for 'Run', 'Stop', 'New', 'Load', and 'Save'.

Time Scale : 0 : 0 : 12 Total : 000:02:00 Auto Adjust

0:00 000:01:00 000:02:00

Tammie06 NetCMD

demo-client-50-m1.tiny.00 Httpperf

demo-client-50-m1.tiny.01 Httpperf

0:00 000:01:00 000:02:00

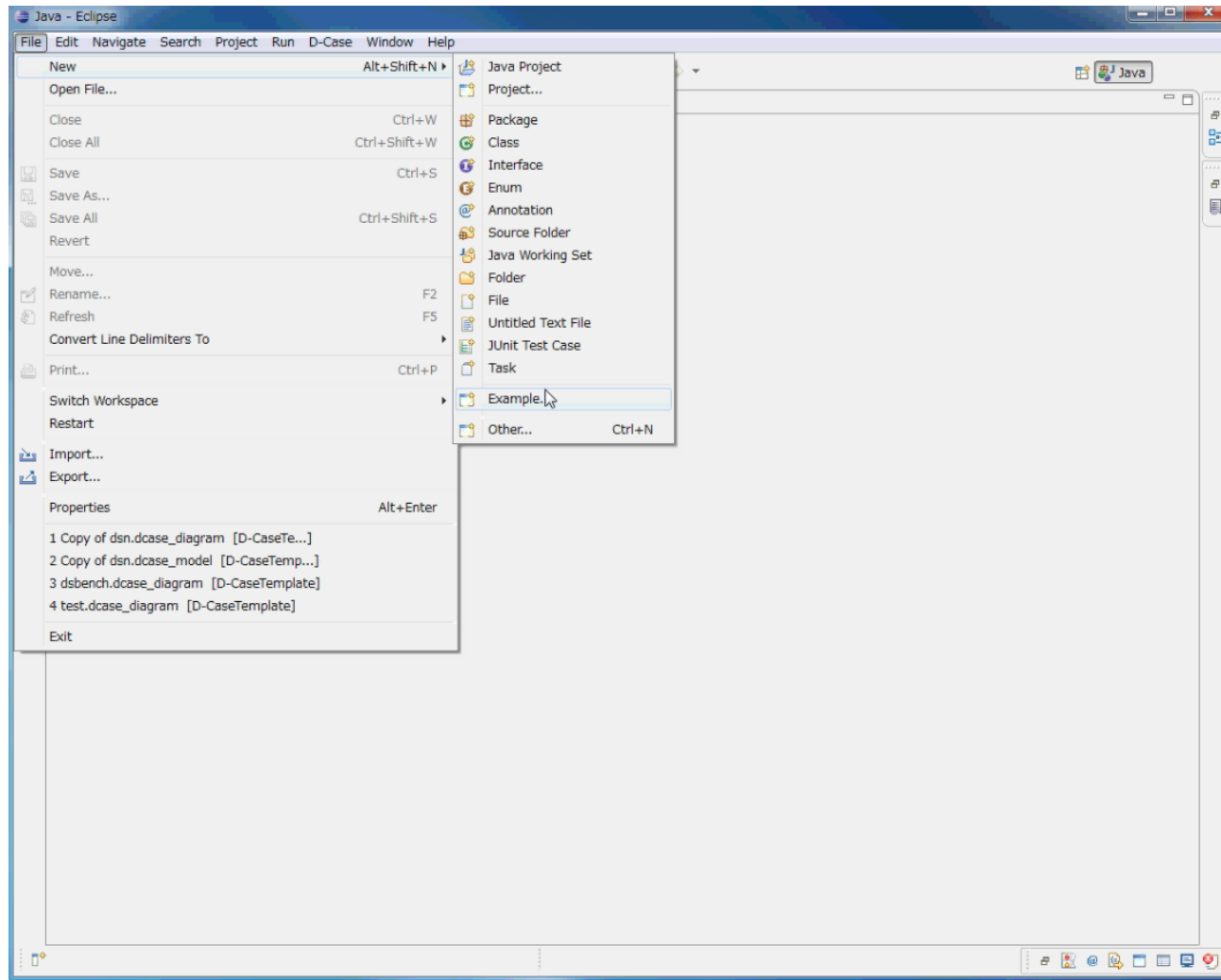
Add Target Target list Add Tags Current Tags:

Add Viewer

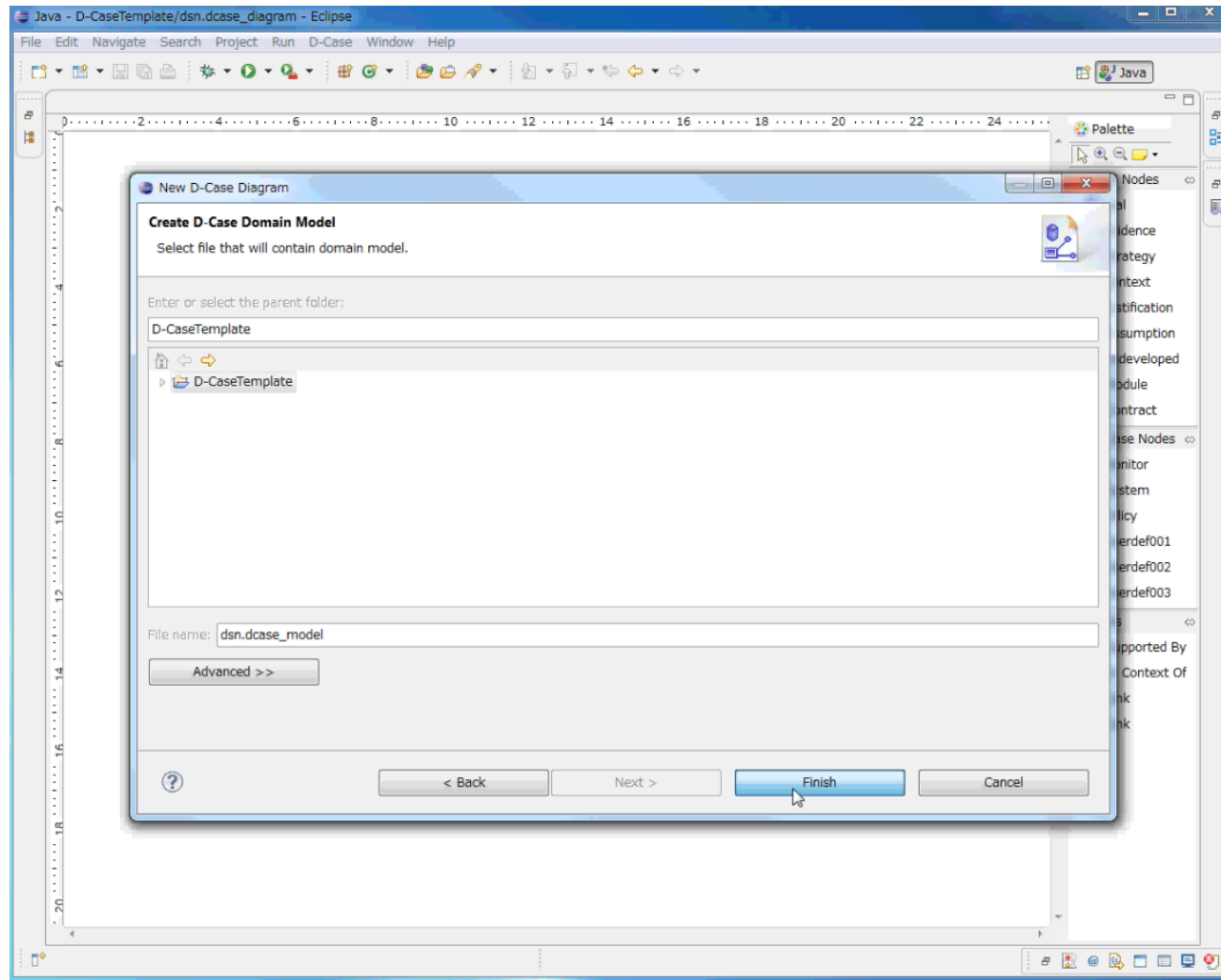
Name	ResultName	Description
<input checked="" type="checkbox"/> latency	uction:max:Httpperf:Total	latency

Run Stop New Load Save

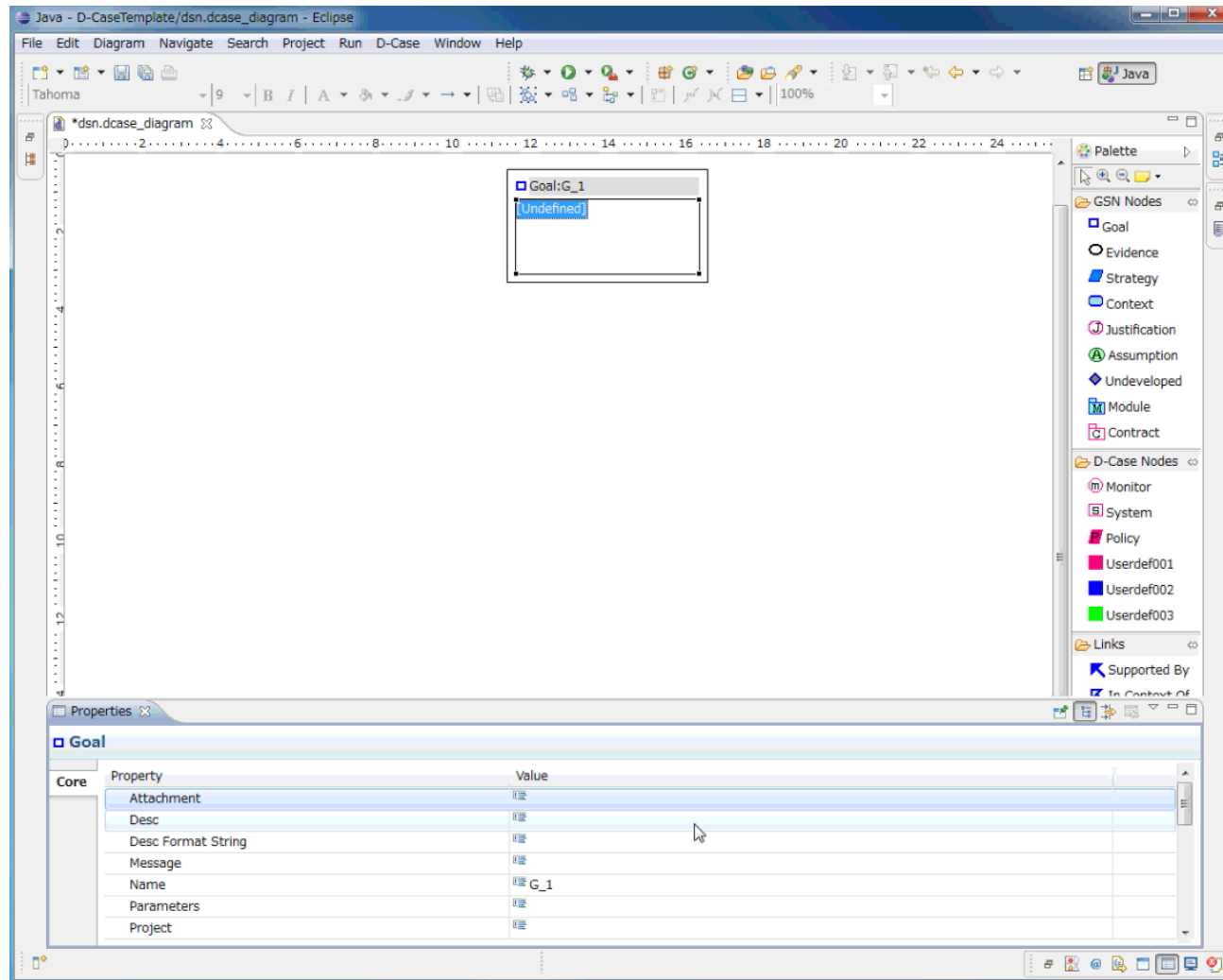
# Demo: Creating a D-Case Diagram



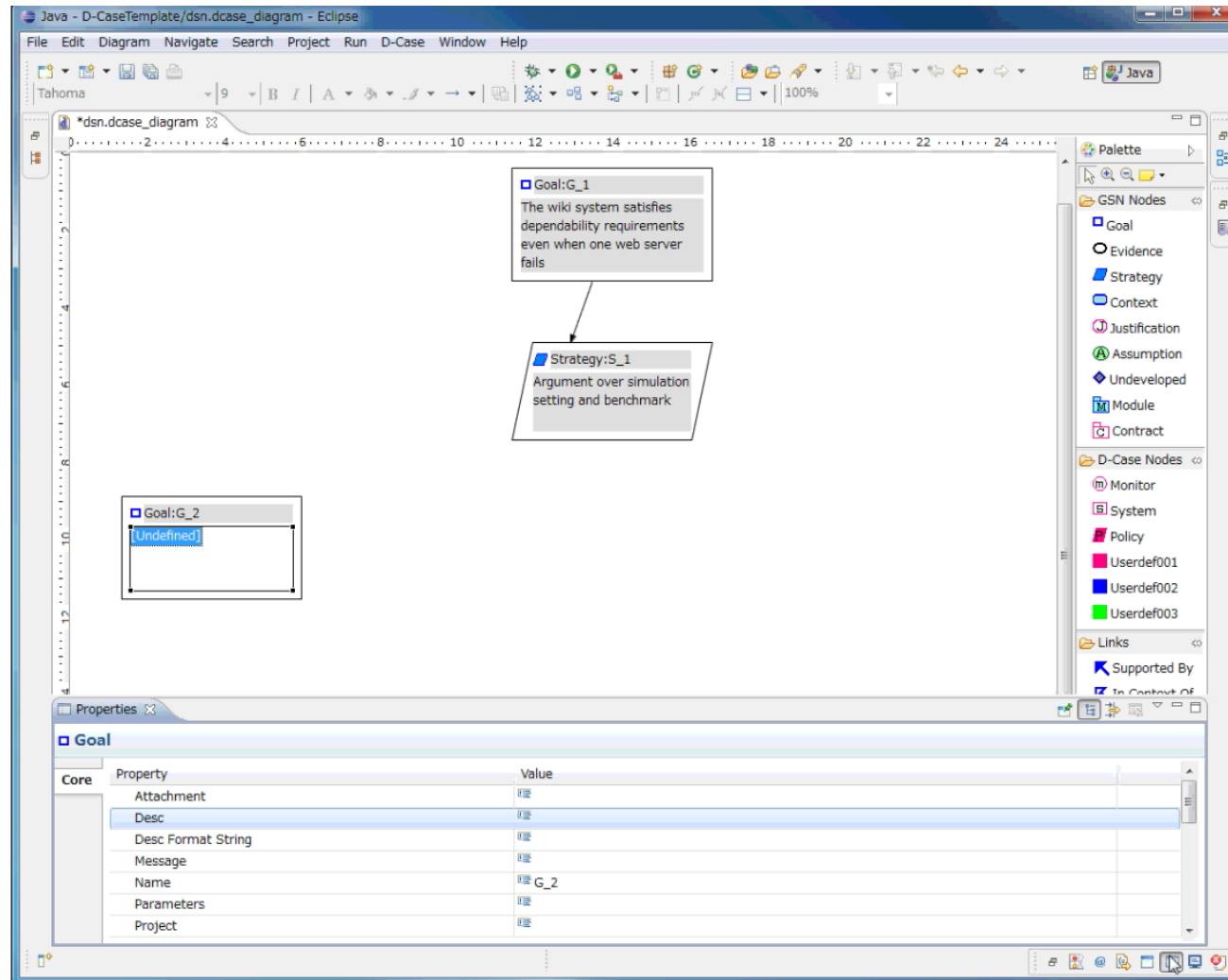
# Demo: Creating a D-Case Diagram



# Demo: Creating a D-Case Diagram



# Demo: Creating a D-Case Diagram



# Demo: Creating a D-Case Diagram

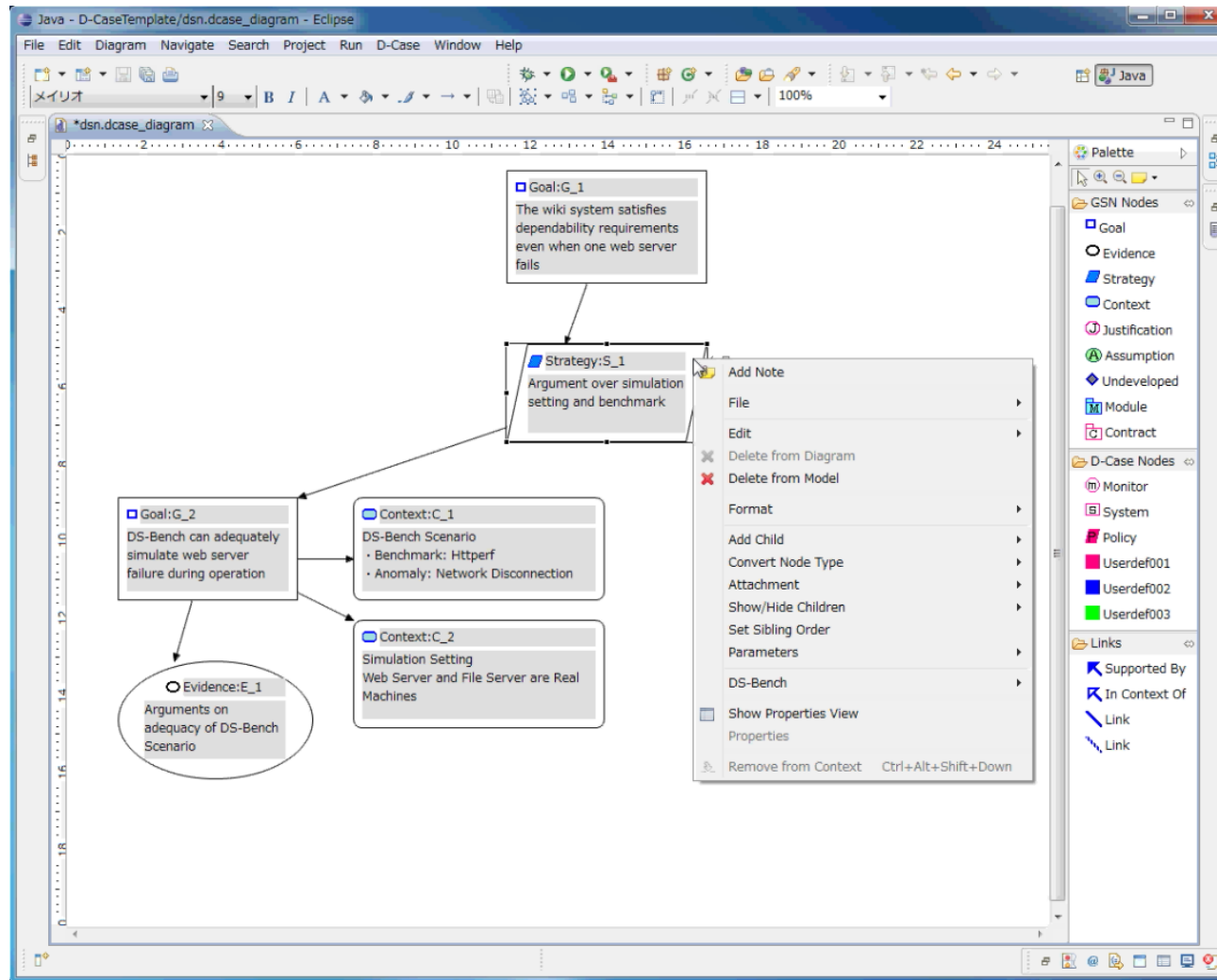
The screenshot shows the Eclipse IDE interface for editing a D-Case Diagram. The main editor displays a diagram with a goal node 'Goal:G\_1' and an evidence node 'Evidence:E\_1'. A dialog box titled 'Desc -- Context C\_1' is open, prompting the user to enter a value for the context. The dialog contains a text area with the following text:

Enter a value:  
DS-Bench Scenario  
• Benchmark: Httpperf  
• Anomaly: Network Disconnection

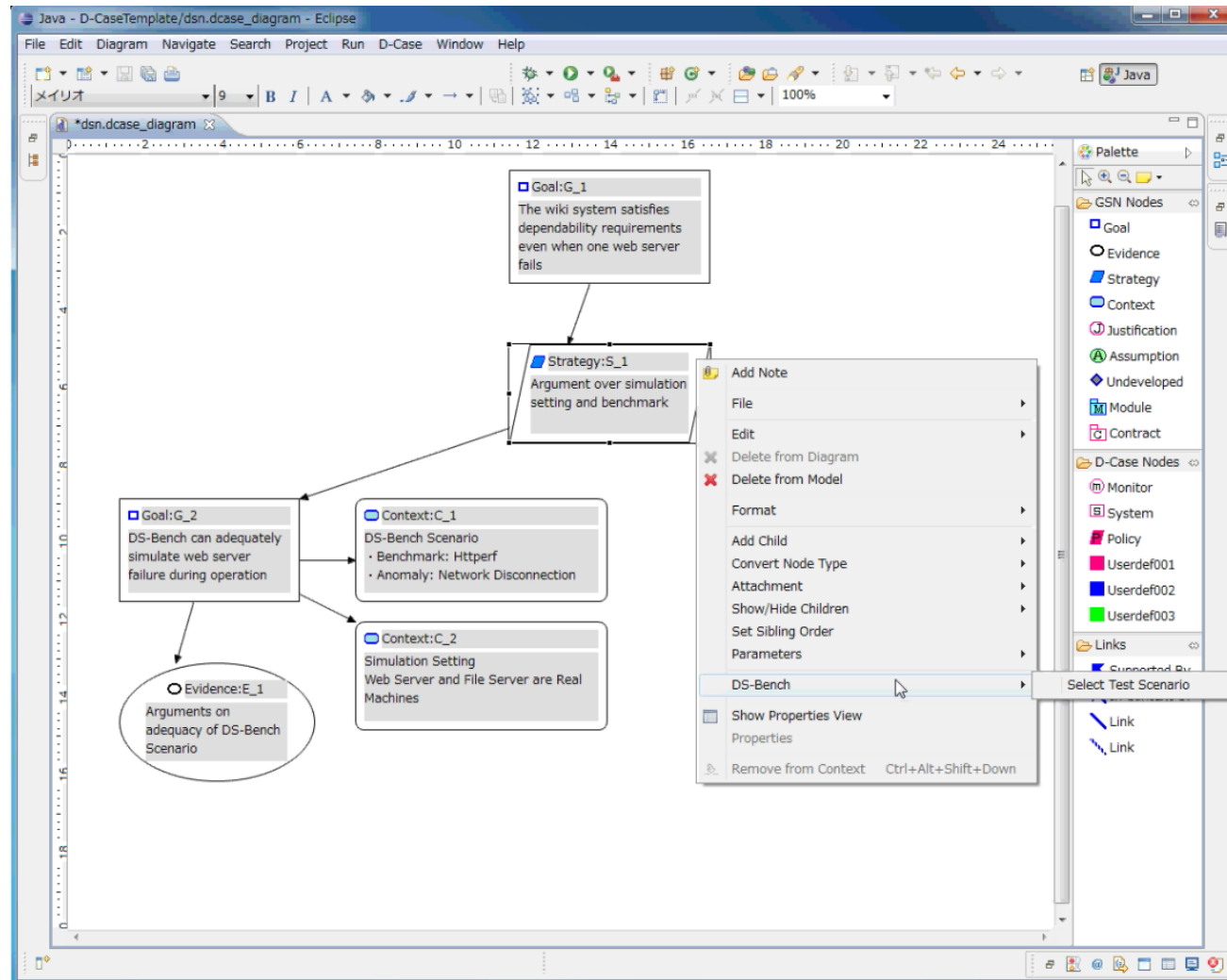
The 'Properties' window at the bottom shows the properties for the selected context:

Property	Value
Attachment	
Desc	
Desc Format String	
Message	
Name	C_1
Parameters	
Project	

# Demo: Selecting a Benchmark Scenario

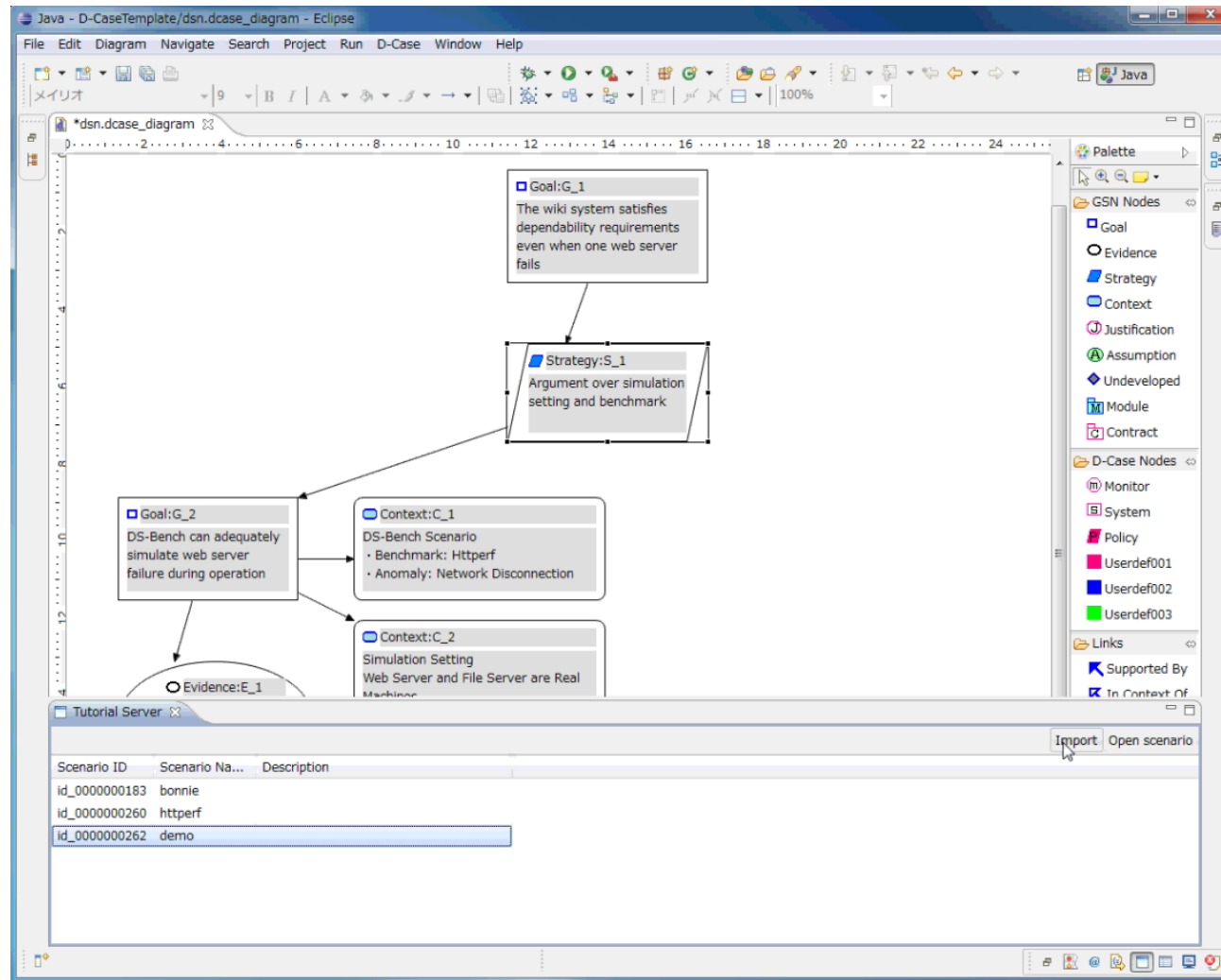


# Demo: Selecting a Benchmark Scenario

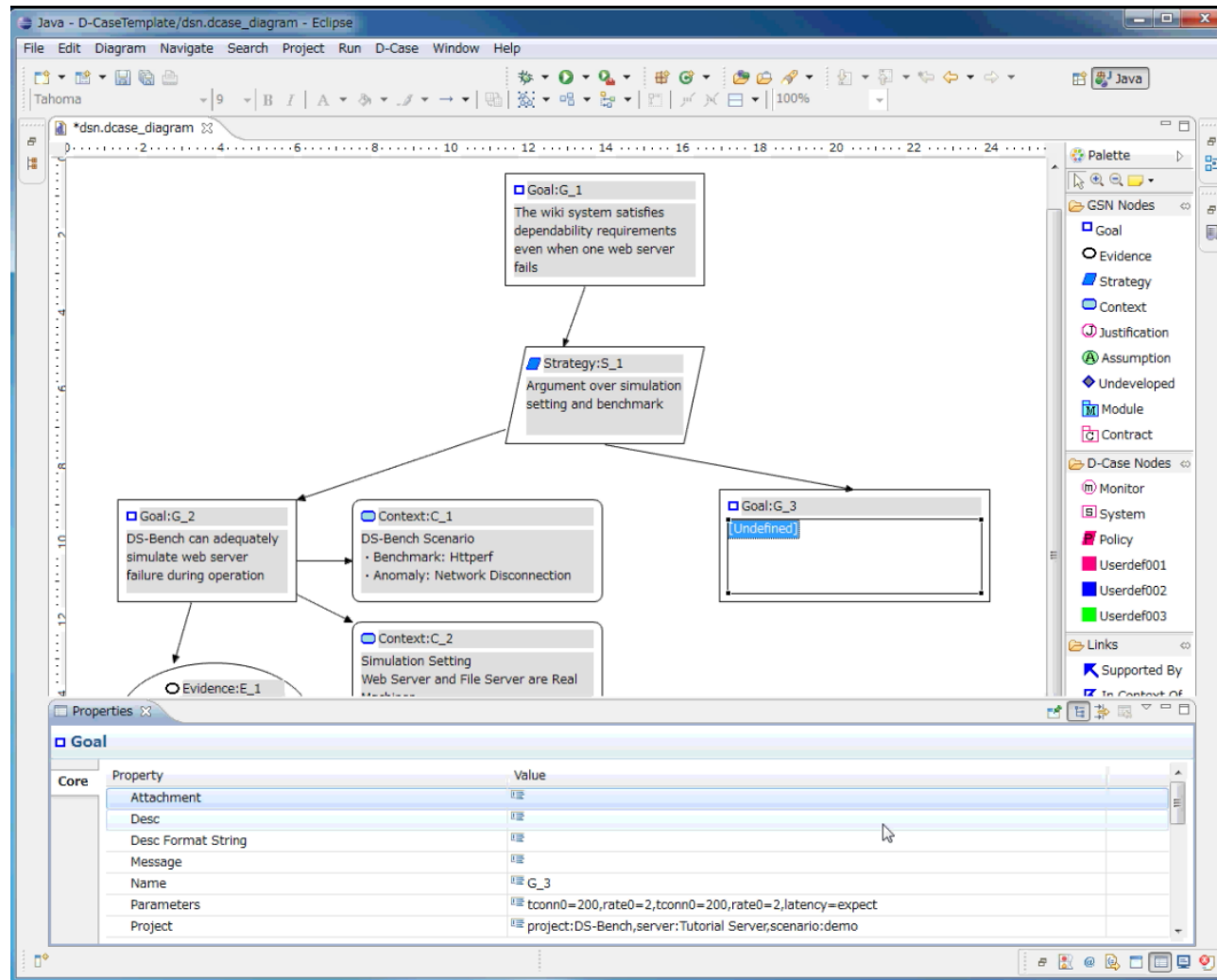




# Demo: Selecting a Benchmark Scenario



# Demo: Selecting a Benchmark Scenario

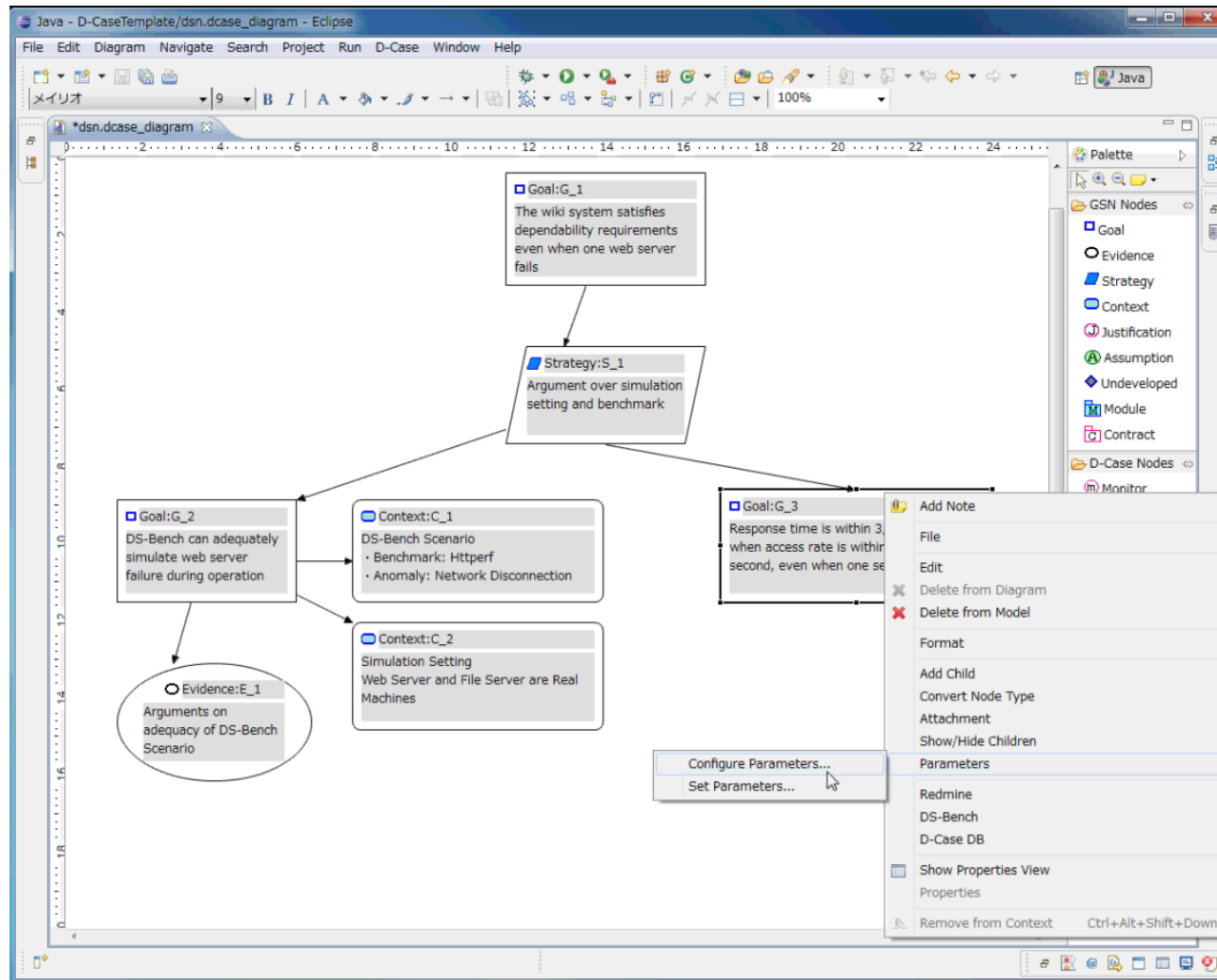


# Demo: Selecting a Benchmark Scenario

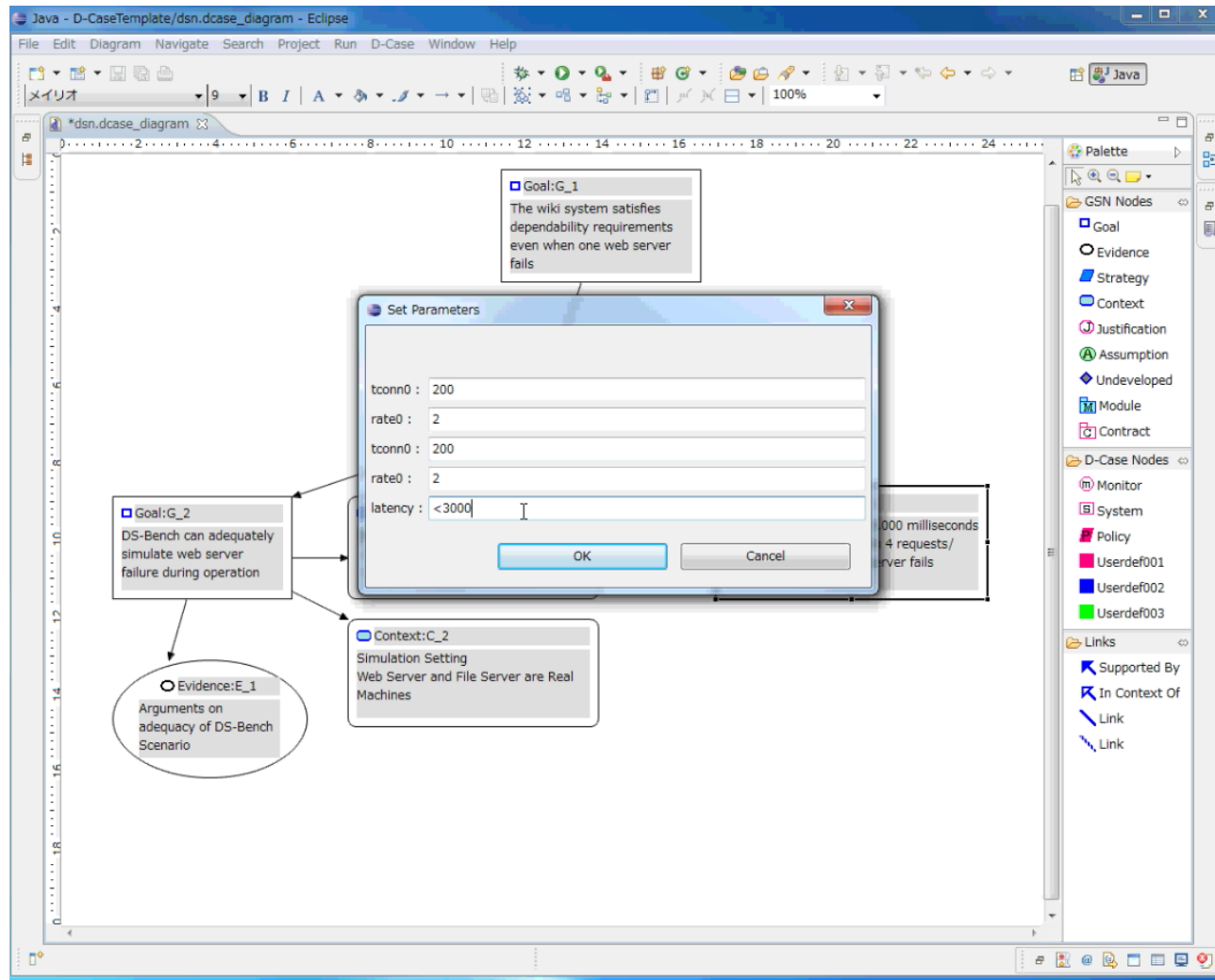
The screenshot shows the Eclipse IDE interface for a D-Case diagram. The main diagram area displays a goal node 'Goal:G\_1' with the description 'The wiki system satisfies dependability requirements even when one web server fails'. A dialog box titled 'Desc -- Goal G\_3' is open, prompting the user to enter a value. The text entered in the dialog is 'Response time is within 3,000 milliseconds when access rate is within 4 requests/second, even when one server fails'. The dialog has 'OK' and 'Cancel' buttons. Below the dialog, a 'Properties' window is visible, showing the details for the selected goal.

Property	Value
Attachment	
Desc	
Desc Format String	
Message	
Name	G_3
Parameters	tconn0=200,rate0=2,tconn0=200,rate0=2,latency=expect
Project	project:DS-Bench,server:Tutorial Server,scenario:demo

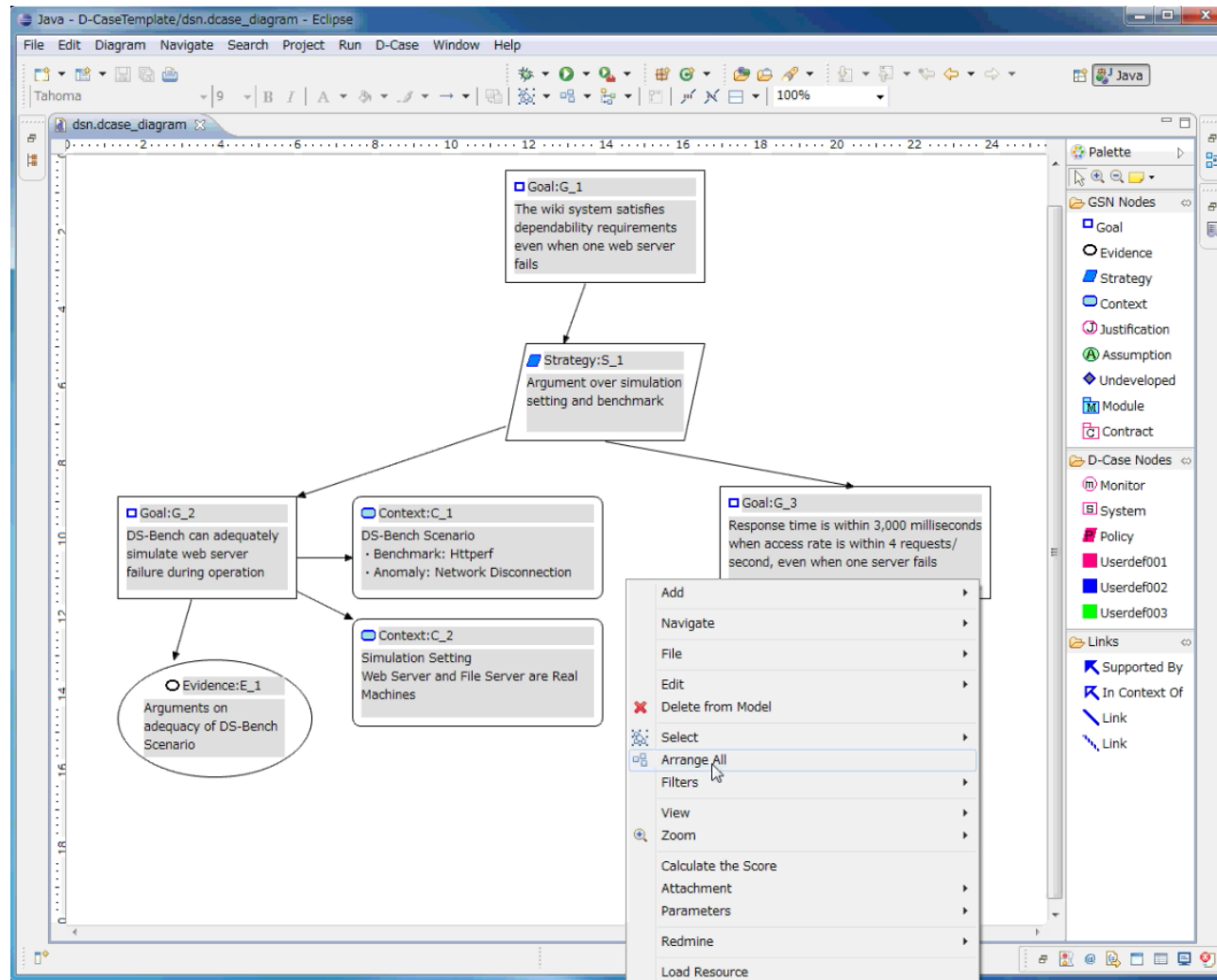
# Demo: Setting Benchmark Parameters



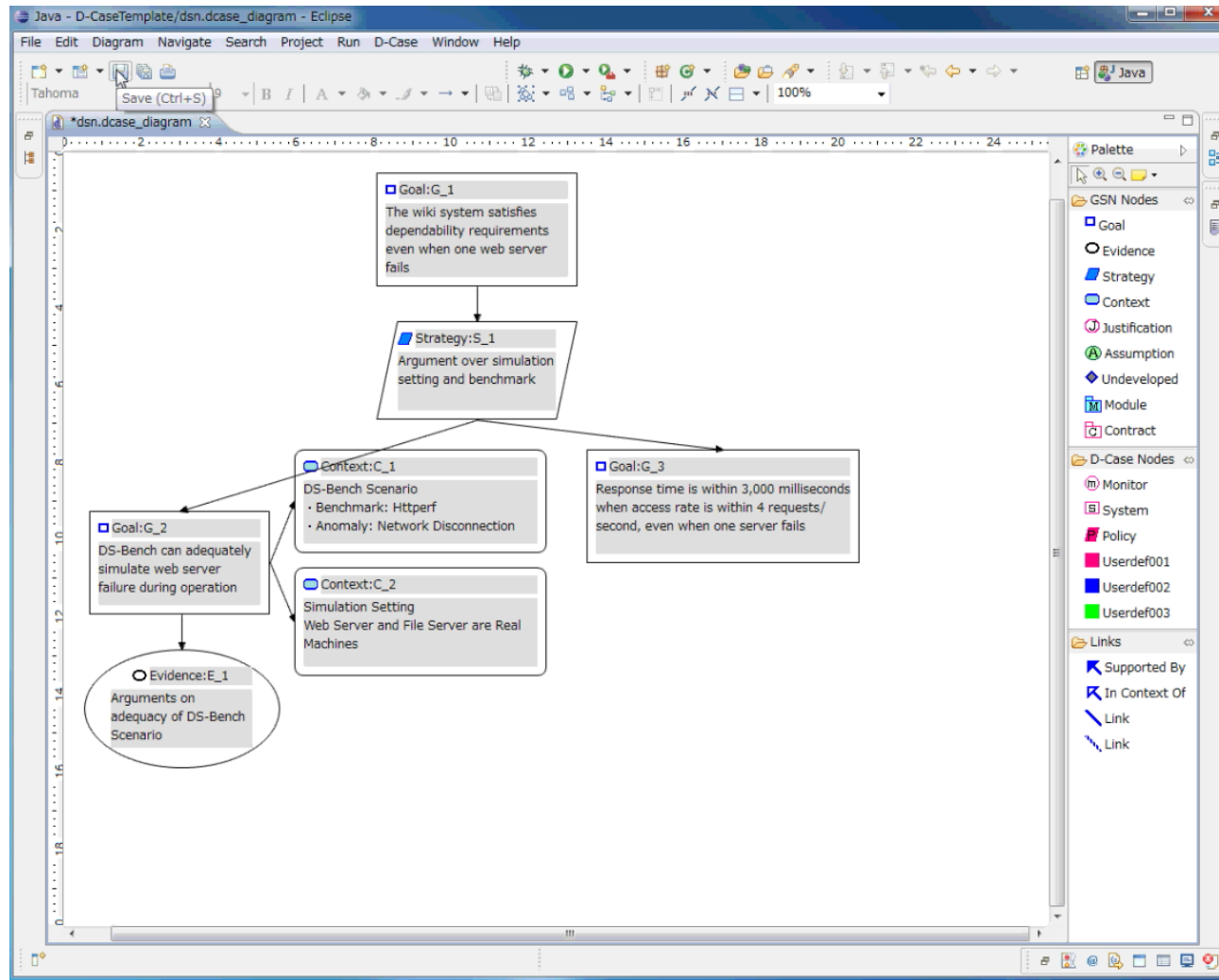
# Demo: Setting Benchmark Parameters



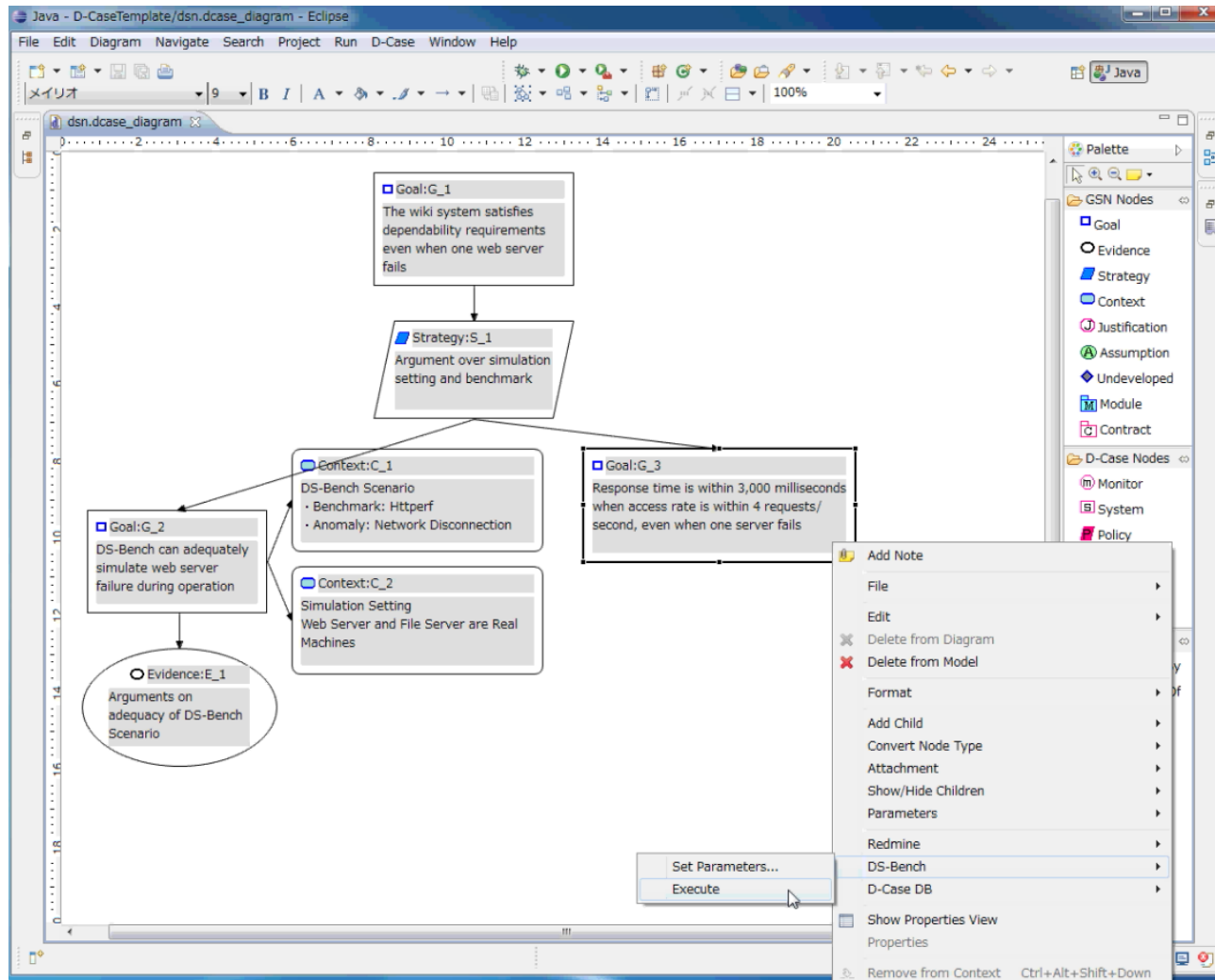
# Demo: Setting Benchmark Parameters



# Demo: Benchmark Execution

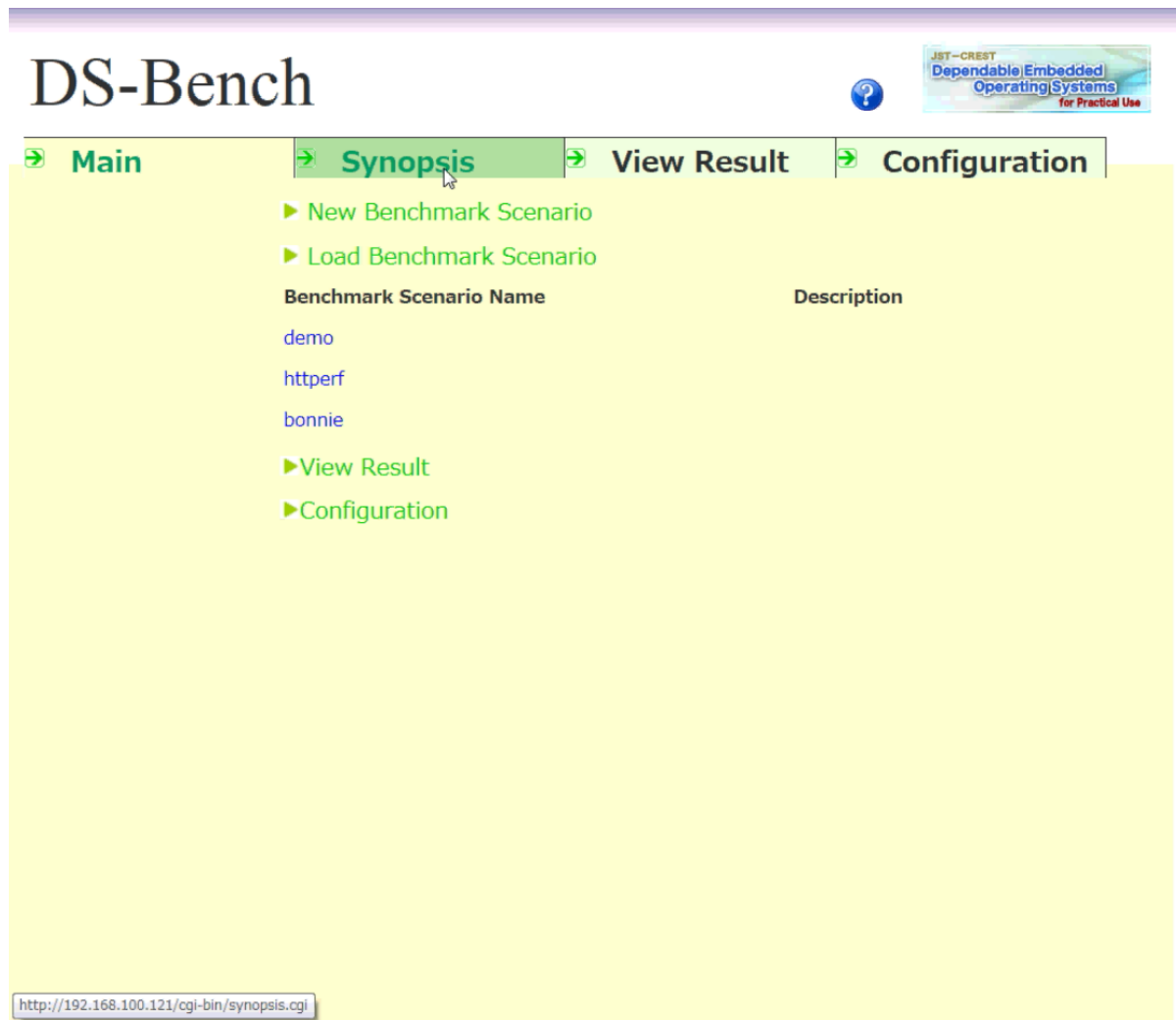


# Demo: Benchmark Execution





# Demo: Benchmark Execution



The screenshot displays the DS-Bench web application. At the top left, the title "DS-Bench" is visible. In the top right corner, there is a logo for "JST-CREST Dependable Embedded Operating Systems for Practical Use" and a help icon. Below the title, a navigation menu contains four items: "Main", "Synopsis", "View Result", and "Configuration". The "Synopsis" item is currently selected and highlighted in green. Below the navigation menu, there are two main sections. The first section contains two links: "▶ New Benchmark Scenario" and "▶ Load Benchmark Scenario". The second section is a table with two columns: "Benchmark Scenario Name" and "Description". The table lists three scenarios: "demo", "httperf", and "bonnie". Below the table, there are two more links: "▶ View Result" and "▶ Configuration". At the bottom left of the page, the URL "http://192.168.100.121/cgi-bin/synopsis.cgi" is displayed.

DS-Bench

JST-CREST  
Dependable Embedded  
Operating Systems  
for Practical Use

➔ Main   ➔ Synopsis   ➔ View Result   ➔ Configuration


▶ New Benchmark Scenario  
▶ Load Benchmark Scenario

Benchmark Scenario Name	Description
demo	
httperf	
bonnie	

▶ View Result  
▶ Configuration

http://192.168.100.121/cgi-bin/synopsis.cgi

# Demo: Benchmark Execution

DS-Bench 

➔ Main   ➔ Synopsis   ➔ View Result   ➔ Configuration

Reload

ID	Date	Status Progress Remaining	Comment	Benchmark Program	Anomaly Load	Starter	View Scenario detail	Stop
353	2012/06/21 18:39:36	run (10%) 1min		Httpperf Httpperf	NetCMD	external	<a href="#">View</a>	<a href="#">Stop</a>
321	2012/06/21 14:03:25	stopping		Httpperf Httpperf	NetCMD	gui	<a href="#">View</a>	<a href="#">Stop</a>
289	2012/06/21 11:15:43	install		Httpperf Httpperf	NetCMD	gui	<a href="#">View</a>	<a href="#">Stop</a>
156	2012/06/14 12:33:21	stopping		Httpperf Httpperf Hackbench		gui	<a href="#">View</a>	<a href="#">Stop</a>
108	2012/06/11 15:01:37	stopping	test	Memstress		gui	<a href="#">View</a>	<a href="#">Stop</a>
99		stopping				gui	<a href="#">View</a>	<a href="#">Stop</a>
96	2012/06/08 10:42:56	stopping	test	Memstress		gui	<a href="#">View</a>	<a href="#">Stop</a>
84		st_error DS-Bench Error: failure to release calculation resource.				gui	<a href="#">View</a>	<a href="#">Stop</a>
83		stopping				gui	<a href="#">View</a>	<a href="#">Stop</a>

# Demo: Benchmark Execution

The screenshot shows the DS-Bench web interface. At the top, the URL is `http://192.168.100.121/cgi-bin/benchmark_main.cgi?progress_id=result_id_000000356&display_id=356&progress=11`. The main heading is "DS-Bench".

The interface features a "Time Scale" slider set to 0:0:12, with a "Total" of 000:02:00 and an "Auto Adjust" button. Below this, three progress bars are shown:

- Tammie06**: A red bar labeled "NetCMD" indicating a failure or error.
- demo-client-50-m1.tiny.00**: A green bar labeled "Httpperf" indicating successful completion.
- demo-client-50-m1.tiny.01**: A green bar labeled "Httpperf" indicating successful completion.

Below the progress bars are buttons for "Add Target", "Target list", "Add Tags", "Add Viewer", and "Current Tags:". A table below these buttons shows a single entry:

Name	ResultName	Description
latency		\$(reduction.max:Httpperf.1 latency)

At the bottom of the interface are buttons for "Run", "Stop", "New", "Load", and "Save".

Below the main interface is a table with two rows of results:

84		DS-Bench Error: failure to release calculation resource.				gui	View	Stop
83		stopping				gui	View	Stop

# Demo: Benchmark Execution

The screenshot shows the DS-Bench web interface. At the top, the URL is `http://192.168.100.121/cgi-bin/benchmark_main.cgi?progress_id=result_id_0000000356&display_id=356&progress=11`. The main heading is "DS-Bench".

Below the heading, there is a "Time Scale" section with a slider set to 12 seconds, a "Total" time of 000:02:00, and an "Auto Adjust" button. A progress bar shows the current time at 00:01:00.

Three benchmarks are listed with their progress bars:

- Tammie06**: Progress bar is red, indicating a failure. The label "NetCMD" is visible on the bar.
- demo-client-50-m1.tiny.00**: Progress bar is green, indicating success. The label "Httpperf" is visible on the bar.
- demo-client-50-m1.tiny.01**: Progress bar is green, indicating success. The label "Httpperf" is visible on the bar.

Below the progress bars, there are buttons for "Add Target", "Target list", "Add Tags", "Add Viewer", "Run", "Stop", "New", "Load", and "Save".

At the bottom, there is a table with the following data:

Name	ResultName	Description	View	Stop
latency		\$(reduction.max.Httpperf.) latency		
84		DS-Bench Error: failure to release calculation resource.	gui	View Stop
83		stopping	gui	View Stop

# Demo: Benchmark Execution

DS-Bench

Time Scale : 0 : 0 : 12 Total : 000:02:00 Auto Adjust

Tammie06 NetCMD

demo-client-50-m1.tiny.00 Httpperf

demo-client-50-m1.tiny.01 Httpperf

00:02:00

Add Target  
Target list  
Add Tags Current Tags:

Add Viewer

Name	ResultName	Description
<input checked="" type="checkbox"/> latency		\$(reduction.max.Httpperf.1) latency

Run Stop New Load Save

Loading results

84		DS-Bench Error: failure to release calculation resource.				gui	View	Stop
83		stopping				gui	View	Stop

# Demo: Benchmark Execution

The screenshot shows the DS-Bench web interface. At the top, the browser address bar displays the URL: `http://192.168.100.121/cgi-bin/benchmark_main.cgi?progress_id=result_id_0000000356&display_id=356&progress=11`. The main heading is "DS-Bench".

Below the heading, there is a "Time Scale" section with a slider set to 12, showing a total time of 000:02:00. An "Auto Adjust" button is present. A progress bar below this shows the execution progress for three benchmarks: "Tammie06", "demo-client-50-m1.tiny.00", and "demo-client-50-m1.tiny.01". The "demo-client-50-m1.tiny.00" and "demo-client-50-m1.tiny.01" benchmarks are marked with "Httpperf" and show blue progress bars. The "Tammie06" benchmark shows a red progress bar, indicating it is the current or most recent benchmark.

An "Alert Dialog" is displayed in the center of the screen, with the message: "The benchmark has been completed." and an "OK" button.

Below the progress bars, there are several control buttons: "Add Target", "Target list", "Add Tags", "Current Tags:", "Add Viewer". Below these is a table with columns "Name", "ResultName", and "Description". The table contains one entry:

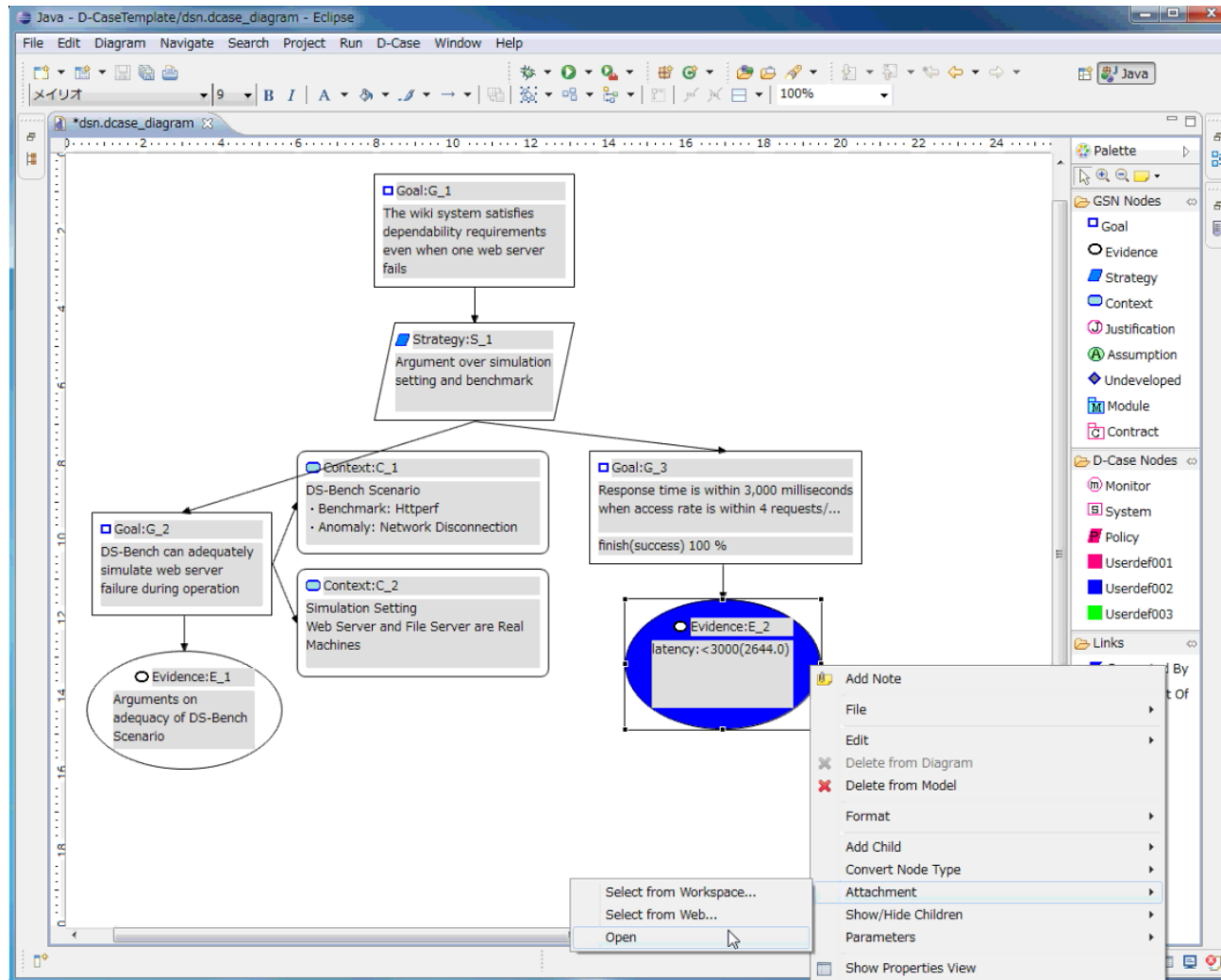
Name	ResultName	Description
latency		\$(reduction.max.Httpperf.1) latency

Below the table are buttons for "Run", "Stop", "New", "Load", and "Save".

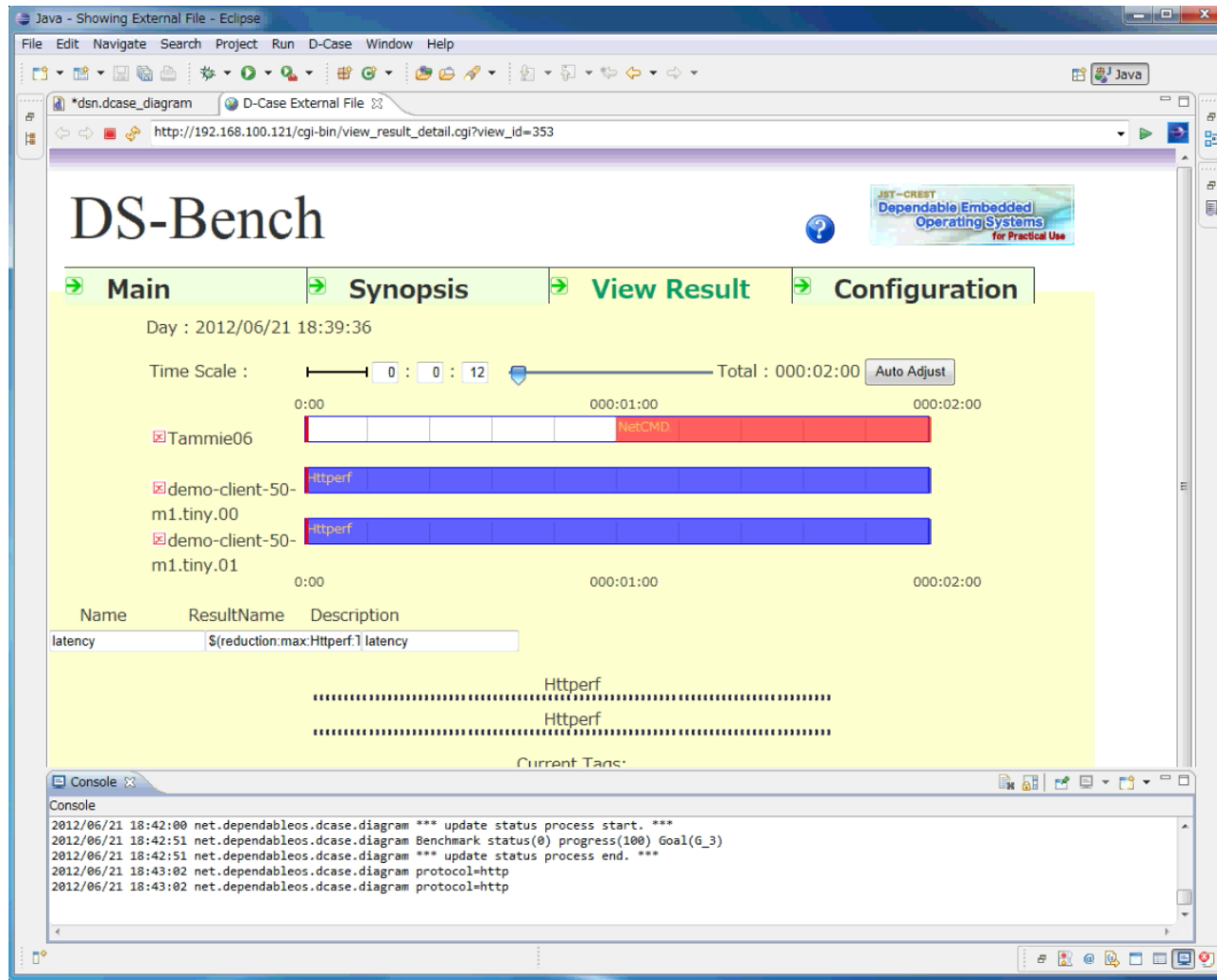
At the bottom of the interface, there is a table with the following data:

Reduction ( Httpperf / Total )							
84		DS-Bench Error: failure to release calculation resource.				gui	View Stop
83		stopping				gui	View Stop

# Demo: Benchmark Result

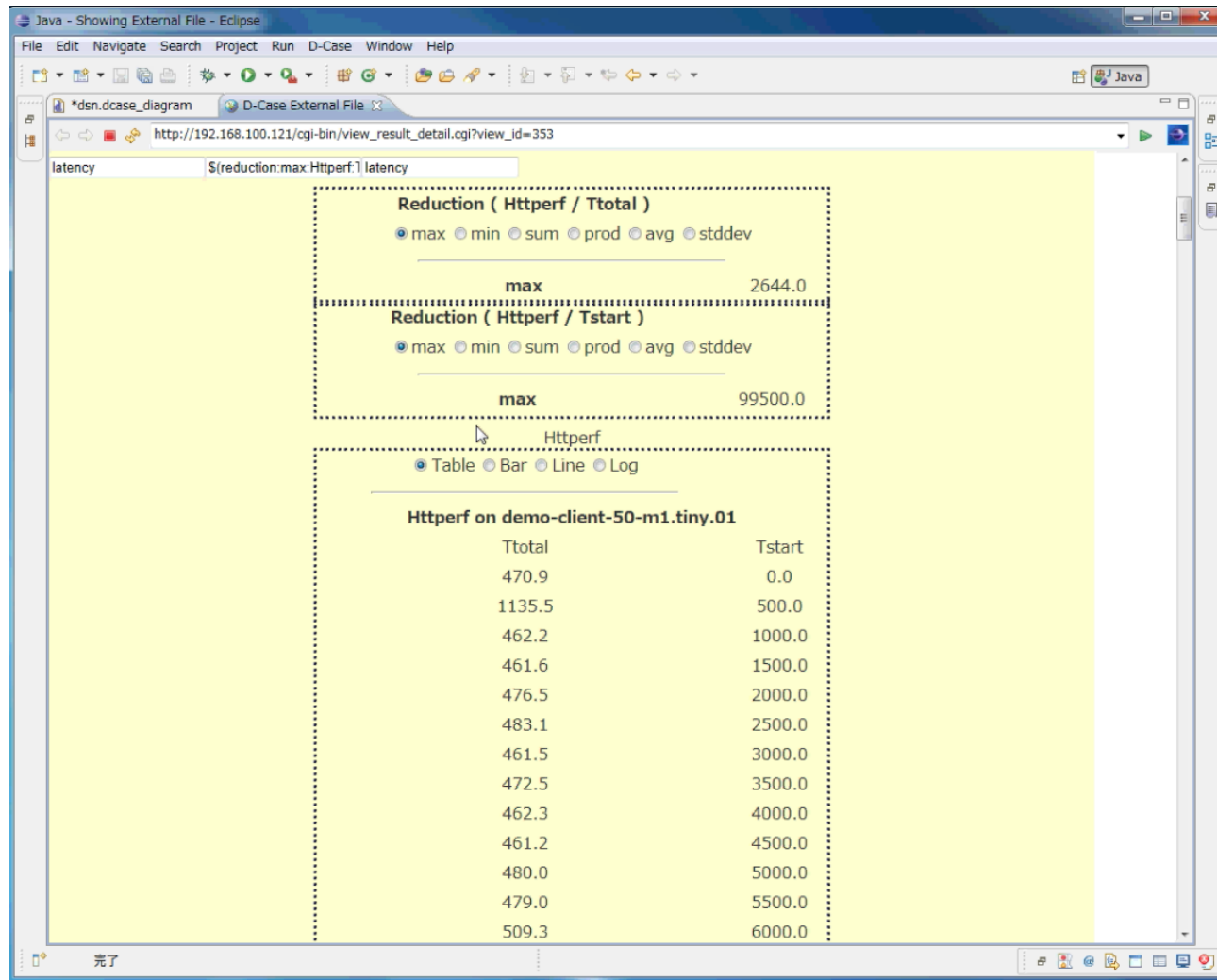


# Demo: Benchmark Result





# Demo: Benchmark Result



# Related Work

---

- ▶ **ASCE Tool [Adelard]**
  - ▶ An well-used assurance case editor. Mainly for generating certification documents
- ▶ **D-Bench [Kanoun 2004, Durães 2004]**
  - ▶ A project aimed at establishing dependability benchmarking methods for several domains
  - ▶ Proposed dependability metrics and measurement tools for several target domains, however did not provide a total software framework to integrate multiple benchmarks
- ▶ **Fault injection tools**
  - ▶ As far as we know, no effort have been made to combine assurance cases and benchmark results dynamically

# Summary

---

- ▶ An automated tool for dependability benchmarking, with dependability assurance, is needed
- ▶ DS-Bench Toolset
  - ▶ D-Case Editor, an assurance case editor
  - ▶ DS-Bench, a framework for benchmark test
  - ▶ D-Cloud, a system for managing hardware resources for benchmark test
- ▶ Tight collaboration between D-Case Editor and DS-Bench
  - ▶ Dependability requirements are described in D-Case
  - ▶ DS-Bench conducts benchmark tests to obtain a quantitative evidence for D-Case

# Questions?

---

- ▶ D-Case Editor is available from
  - ▶ <http://www.dependable-os.net/tech/D-CaseEditor/>
- ▶ DS-Bench/D-Cloud will be available from
  - ▶ <http://www.dependable-os.net/tech/DSBenchDCloud/>
- ▶ We thank Hajime Ueno for designing D-Case Editor with us. We also thank Shingo Takeda and Hideaki Koizumi for helping us with preparing the demonstration.