

NEC Parallel Debugger User's Guide

(G2AT02E)

SX-Aurora TSUBASA

Proprietary Notice

The information disclosed in this document is the property of NEC Corporation (NEC) and/or its licensors. NEC and/or its licensors, as appropriate, reserve all patent, copyright and other proprietary rights to this document, including all design, manufacturing, reproduction, use and sales rights thereto, except to the extent said rights are expressly granted to others.

The information in this document is subject to change at any time, without notice.

Eclipse is a registered trademark of Eclipse Foundation.

All other product, brand, or trade names used in this publication are the trademarks or registered trademarks of their respective trademark owners.

Copyright 2018-2019 NEC Corporation.

Introduction

This document explains how to use NEC Parallel Debugger.

How to Read

This document is composed of the following chapters. The rightmost column is target readers of the corresponding chapter.

Chapter	Title	Description	Target Readers
1	Overview	The overview of NEC Parallel Debugger	Programmers
2	Creation of a Project	Creation of a project in Eclipse PTP	Programmers
3	Building of a Project	Building of a project in Eclipse PTP	Programmers
4	Creation of a Debug Configuration	Creation of a Debug Configuration in Eclipse PTP	Programmers
5	Operations for Debugging	Operations for debugging in Eclipse PTP	Programmers
6	Notices and Restrictions	Notices and restrictions for NEC Parallel Debugger	Programmers

Related Documents

- C/C++ Compiler User's Guide (G2AF01E)
- Fortran Compiler User's Guide (G2AF02E)
- NEC MPI User's Guide (G2AM01E)

Remarks

• This manual conforms to Release 1.0.0 and subsequent releases of the NEC Parallel Debugger.

Glossary

Term	Description	
Vector Engine (VE)	Vector Operation Engine implemented as a PCI Express Card attached to an x86 server. This is the core component of the SX-Aurora TSUBASA system.	
Vector Host (VH)	An x86 server equipped with VEs.	
Node	A VE, which has a shared memory.	
NQSV	The NEC Network Queuing System V, which is a batch processing system for high-performance cluster system.	
NEC MPIMPI (Message Passing Interface) implementatioMPI is A specification for a standard Icommunication. It can be used together with 0automatic parallelization.		
Eclipse Parallel Tools Platform (PTP)	ols An integrated software development environment for parallel applications, which is an open source software.	
View	A subwindow displayed in Eclipse window.	
Perspective	The name given to an initial collection and arrangement of views and an editor area.	
Target Process	A process targeted for debugging.	

1	Ove	erview	. 6
	1.1	NEC Parallel Debugger	. 6
	1.2	Operating Environment	. 6
	1.3	Steps for Debugging	. 7
2	Cre	eation of a Project	. 8
	2.1	Invocation of Eclipse	. 8
	2.2	Import of a Make Environment on a Remote Host	. 9
	2.3	Import of a Make Environment on the Local Host	11
3	Bui	lding of a Project	14
	3.1	Setting of the Build Configuration	14
	3.2	Execution of a Build Project	15
4	Cre	eation of a Debug Configuration	16
	4.1	Creation of a Debug Configuration	16
	4.2	Setting of Resources	17
	4.3	Setting of an Application	20
	4.4	Setting of the Debugger	21
	4.5	Setting of the Environment	21
	4.6	Other Settings	22
	4.7	Starting of a Debug Execution	22
5	Оре	erations for Debugging	24
	5.1	Available Views	24
	5.2	Debugging of Multiple Processes in a Collective Manner	24
	5.3	Debugging of One Process	25
	5.4	Display of the Stack Trace	26
	5.5	Reference of Variable Information	27
	5.6	Setting of Breakpoints	28
	5.7	Termination of a Debug Session	28
6	Not	tices and Restrictions	30
	6.1	Notices	30
	6.2	Miscellaneous	30
Aŗ	ppend	ix A: Change Log	31

1 Overview

1.1 NEC Parallel Debugger

NEC Parallel Debugger is an Eclipse PTP plugin for debugging of MPI applications. Eclipse PTP is an open source integrated software development environment for parallel programs. Eclipse PTP into which NEC Parallel Debugger is integrated enables effective debugging of distributed-memory parallel programs with MPI in addition to shared-memory parallel programs written in Fortran and C/C++ with OpenMP or automatic parallelization.

1.2 Operating Environment

Users invoke NEC Parallel Debugger on a front-end machine to which they have logged in via ssh etc, and can edit, compile, and debug software launched by ssh as shown in the first figure below. It is also possible to invoke NEC Parallel Debugger on the SX-Aurora TSUBASA to which they have logged in via ssh or qlogin which is used for interactive request supported by NQSV as shown in the second figure below. When a debug execution for a MPI application using NEC Parallel Debugger starts, Scalable Debug Manager (SDM) is invoked on the SX-Aurora TSUBASA system. Then SDM initiates gdb and an MPI application targeted for debugging.





1.3 Steps for Debugging

The following figure shows the debugging steps for the first time. The steps one through three are not needed for the second time and afterwards. The following chapters explain each step for debugging applications using NEC MPI launched on VE.



2 Creation of a Project

Development of applications with Eclipse requires creation of a project, which is an environment under which users develop and debug an application. This chapter explains the steps for creating a project and importing an existing make environment for the following two cases:

- Import of a make environment on a remote host
- Import of a make environment on the local host

2.1 Invocation of Eclipse

Execution of the following command invokes Eclipse, and a window opens.

```
% /<INST-PATH>/eclipse/eclipse
```

where *<INST* PATH> is the path under which Eclipse is installed.

By inputting the Workspace directory, which is the working directory for a project, on the window, the Welcome view is displayed as shown in the following figure.



Please close the Welcome view by clicking the X sign on the tab, and the C/C++ perspective, which is a set of views for C/C++, is displayed as shown in the following figure.



2.2 Import of a Make Environment on a Remote Host

It is recommended to create a synchronized project so that changes under the project such as modifications of source files are automatically reflected into the original environment on the remote host. The following is the steps for creating a synchronized project and import an existing make environment.

1. Right-click on empty space in the Project Explorer view and select "New > Synchronized C/C++ Project" or "Synchronized Fortran Project" to open the New Synchronized Project window.



2. Input a project name in the Project name field on the window, and then click on the New button in the Connection name line.

New Synchronized Project	<u></u>
Please select a project type to continue	
Project name: ve_mpi_bm Local directory Use default location Local directory: /usr/uhome/npdb/workspace/ve_mpi_bm Remote directory- Connection name: Please select a connection Remote directory:	Browse New Browse
Modify file filtering Project Type GAU Autotools Executable Executable (XL UPC) Shared Library Shared Library (XL UPC) Static Library (XL UPC) Static Library (XL UPC) Makefile project Show project types and toolchains only if they are supported on the platform	Remote Toolchain (select 1 or more)
0	<pre>< Back Next > Cancel Finish</pre>

3. Input in the entry fields such as Connection name, Host, User, and Passphrase, and click on the Finish button.

New Connection Specify properties of a new connection					
Connection na	me: remotehost				
Host informat	tion				
Host:	remotehost				
User:	npdb				
 Public ke 	Public key based authentication Keys are set at <u>Network Connections, SSH2</u>				
Passphrase:	•••••••				
Password	based authentication				
Password:					
 Advanced 					
?	Cancel Finish				

4. Input in the Remote directory field, select "Makefile project > Empty Project" in the Project Type field, and click on the Finish button.

New Synchronized Project Create synchronized project of the selected type	I.
Project name: ve_mpi_bm Local directory- Vuse default location Local directory: /usr/uhome/npdb/workspace/ve_mpi_bm Remote directory- Connection name: remotehost Remote directory: /usr/uhome/npdb/remotehost/ve_mpi_bm	Browse
Modify file filtering Project Type Conters Conters Conters Conters Conters Conters Conters Conters Conters Empty Project Empty Project - Fortran Demo - Hello World - Fortran Demo - Hello World - Fortran Demo - Hello World - Fortran using MPI Demo - Calculate Pi - Fortran using MPI Show project types and toolchains only if they are supported on the plane of the pl	Remote Toolchain (select 1 or more) Cygwin GCC GCC Fortran Local Toolchain (optional - select 0 or more) - Other Toolchain Cygwin GCC GCC Fortran tform
0	Sack Next > Cancel Finish

The creation of a synchronized project and import of a make environment are now complete.

Note that changes under the project are not reflected into the original environment at this point.

Please select "Synchronized Project" in the Project Explorer view and click on the button in the menu to activate the reflection.

<u>File Edit Source</u> F	Refac <u>t</u> or <u>N</u> avigate	Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
	• 🐔 🗶 🗟 🖌	i i → 12i → 12i → 13i → 14i → 10i → 9i → 12i →) 🕶 📃 🚺 🕶
∲ - ∦ - ∜- ¢	Synchronize the sel	cted project	
Project Explorer 🛿	- 6		- 8
🕨 😂 ve_mpi_bm			

2.3 Import of a Make Environment on the Local Host

The following is the steps for creating a project and import an existing make environment on the local host. The working directory for the project is the same as that of the original make environment.

1. Right-click on empty space in the Project Explorer view and select "Import" to open the Import window.

Project Explo	rer 🛛 🕹	
	E 🔄 🕯	~
	New	•
	<u>ک</u> Import	
	🛃 Exp <u>o</u> rt	
	🔊 Re <u>f</u> resh	F5
	Show Terminal	
	Show ferminar	

2. Select "C/C++ > Existing Code as Makefile Project" on the window, and click on the Next button to open another window.

Select Creates a new Makefile project in a directory containing existing code	Ľ
Select an import wizard:	
type filter text	•2
General General C/C++ C/C++ Executable C/C++ Project Settings Existing code as Autotools project General Gen	
 Comph Production Production Production 	
? < Back Next > Cancel	Finish

3. Input a project name in the Project Name field and the directory where the make environment exists in the Existing Code Location field, and select "none" in the Toolchain field on the window that opened at step 2.

Import Existing Code
Create a new Makefile project from existing code in that same directory
_Project Name
ve_mpi_bm
Existing Code Location
/usr/uhome/npdb/workspace/localhost/ve_mpi_bm Browse
Languages
✓ C ✓ C++
Toolchain for Indexer Settings
<none></none>
GCC Fortran
GNU Autotools Toolchain
IBM XL Fortran Tool Chain
Intel(R) Fortran Toolchain on Intel(R) 64
Linux Berkeley UPC
Linux GCC
XL C/C++ Tool Chain
XL UPC Tool Chain
Show only available toolchains that support this platform
(?) < Back Next > Cancel Finish

4. If your code is written in Fortran, select "Convert to Fortran Project" in the Project Explorer view to convert from a C/C++ project to a Fortran one.

Project Explorer 🕱)		
E 🕏	New	•	
🕨 😂 ve_mpi_bm	Go <u>I</u> nto		
	Open in <u>N</u> ew Window		
	Show in Local Terminal	►	
	📄 <u>С</u> ору	Ctrl+C	
	🖻 Paste	Ctrl+V	
	🗙 <u>D</u> elete		
	🚴 Remove from Context		
	Source	•	
	Move		
	Rena <u>m</u> e	F2	
	<u>≥ I</u> mport		
	🖾 Exp <u>o</u> rt		
	<u>B</u> uild Project		
	Clean Project		
	🔊 Re <u>f</u> resh	F5	
	Clo <u>s</u> e Project		
	Close Unrelated Projects		Proportion 1919 Coll (
	Build Targets	•	
	Index	•	Resour
	Build Configurations	•	
	Profiling Tools	•	
	Validate		
😂 ve_mpi_bm	👔 Convert to Fortran Project		
	Run As	•	a mpe ozrzoo ripi i o .

The creation of a project and import of a make environment are now complete.

3 Building of a Project

This chapter explains how to configure a project and generate an executable application.

3.1 Setting of the Build Configuration

The following explains the steps for setting the paths to compilers and header files required for generating an application.

1. Right-click on the project name in the Project Explorer view and select "Properties".

File Edit Source Refactor Navigate	Search Project	Run Window Help		
New	Shift+Alt+N 🕨	🖏 🔹 🖉 🕶 🖏 🖛 🚱 🔹 🎋 🔹 🔿 🤹	. 🚷	
Open File				Q
😂 Open Projects from File System				5
Close	Ctrl+W			
Close All	Shift+Ctrl+W			
📓 Save	Ctrl+S			ŕ
📓 Save As				
🐚 Save All	Shift+Ctrl+S			
Revert				
Move				
🖻 Rename	F2			
🐑 Refresh	F5			
Convert Line Delimiters To	+			
👜 Print	Ctrl+P			
≧ Import				
🖾 Export				
Properties	Alt+Enter			
<u>1</u> vempi.c [ve_mpi_bm]				
Switch Workspace	•			
Restart				24
Exit		asks 📮 Console 🔲 Properties 🕮 Call Graph		

2. Select "C/C++ Build > Environment > Add" and set the value of the environment variable PATH. Please select "C/C++ Build" for a Fortran project, too.

B	Environment			¢- •	• 🖒 • •
Resource Builders	Configuration:	Default [Active]	•	Manage Confi	gurations
Build Variables	Environment v	ariables to set			
Environment	Maniahla	Makes	Oninin		Add
Logging	CWD	value	DUILD SYSTEM		Select
Tool Chain Editor XL C/C++ Compiler	PWD	/usr/uhome/npdb/work	BUILD SYSTEM		Edit
 C/C++ General Fortran Build Linux Tools Path Paths and Symbols Project References Run/Debug Settings 					Undefine
Task Repository Task Tags	Append var Benlace pat	iables to native environment	000		
Validation WikiText		ave environment with specified	Restore	Defaults	Apply
?			Cancel	Apply	and Close

The value of the environment variable PATH needs to include the paths to all the commands required for generating the application.

Name:	PATH	
Value:	/opt/nec/ve/bin:/usr/bin:/bin	Variables
🗹 Add	to all configurations	
	Cancel	ОК

3. Select "C/C++ General > Paths and Symbols > Add" and set the directories under which include files referenced in the source files are placed. Please select "C/C++ General" for a Fortran project, too.

	Paths and Symbols	⟨⇒ ▼ ⇒ ▼
 Resource Builders C/C++ Build 	Configuration: Default [Active]	Configurations
 C/C++ General Code Analysis 	🖉 🕒 Includes 🗰 Symbols 🔿 Libraries 🕞 Libr	ary Paths 💙
Documentation File Types	Languages Include directories	Add
Formatter Indexer	Fortran	Edit
Language Mappin	GNU C // C	Delete
Preprocessor Inclu	UPC	Export
Profiling Categori XL C/C++ Langua		Move Up
Fortran Build Linux Tools Path	"Preprocessor Include Paths, Macros page may define additional entries Show built-in values	Move Down
Paths and Symbols Project References	🖓 Import Settings 🖗 Export Settings	
Run/Debug Settings	Restore Defaults	Apply
?	Cancel	Apply and Close

The following example is the window to input a directory.

Directory:			
/opt/nec/ve/ncc/1.0.3/include/	1		
Add to all configurations		Variables	
Add to all languages Add to all languages		Workspace	
		File system	
	ОК	Cancel	

3.2 Execution of a Build Project

Right-click on the project name in the Project Explorer view and select "Build Project" to generate the application.

4 Creation of a Debug Configuration

This chapter explains how to configure the settings for debug and start a debug execution.

4.1 Creation of a Debug Configuration

The following is the steps for creating a Debug Configuration.

1. Select "Window > Perspective > Open Perspective > Other" in the Project Explorer view.



2. Select the Parallel Debug perspective.

Open Perspective (on a1sb4_003)
C/C++ (default)
to Debug
Fortran
🔚 Git
🗱 Parallel Debug
() Planning
Resource
System Monitoring
SystemTap IDE
🖆 Team Synchronizing
Tracing
X XML
Use F2 to display the description for a selected perspective.
Cancel Open

3. Select "Debug (bug icon) > Debug Configurations" to open the Debug Configurations window

File Edit Navigate Search Project Run Window Help		
:	• 0 • 9 • i b s s i b s i c b s c s s c c s s c c s s c c s s s c c s s s c s s s s s s s s s s	ss 😰 🖬 🗱
Image: Second a job Image: Second a	(no launch history) Debug As Debug Configurations Organize Fayorites	

4. Select "Parallel Application > New" on the window

Create, manage, and run configurations Create a configuration to debug a parallel application		
Image: Constraint of the system Image: Constraint of the system <th> Configure launch settings from this dialog: Press the 'New' button to create a configuration of the selected type. Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. Press the 'Filter' button to configure filtering options. Edit or view an existing configuration by selecting it. Configure launch perspective settings from the <u>'Perspectives'</u> preference page. </th> <th></th>	 Configure launch settings from this dialog: Press the 'New' button to create a configuration of the selected type. Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. Press the 'Filter' button to configure filtering options. Edit or view an existing configuration by selecting it. Configure launch perspective settings from the <u>'Perspectives'</u> preference page. 	

5. Input a debug configuration name in the Name field.

Create, manage, and run configu No target system configuration has	rrations s been selected	Ś	
	Name: test_ve_mpi_bm		
type filter text	말 Resources 🖺 Application 🍽 Arguments 🕸 Debugger 🐷 Environment Synchronize 🕸 Source	"ı	
 C/C++ Application C/C++ Attach to Application 	Target System Configuration: Please select a target system configuration		
C/C++ Postmortem Debugger	O Local () Remote Please select a connection	New	
Group Launch Group Launch Group (Deprecated)			
▼ ➡ Parallel Application ➡ New_configuration			
Filter matched 8 of 8 items	Revert	Apply	
?	Close	Debug	

4.2 Setting of Resources

The following is the steps for configuring the settings required for execution of MPI applications such as runtime options.

1. Select "Generic NECMPI Interactive" in the Target System Configuration field on the Resources tab.

Create, manage, and run configura ல No target system configuration has b	tions been selected		Ť.
□ 🗎 🗶 🖻 🎲 🕶	Name: test_ve_mpi_bm		
type filter text	Resources 🖺 Application	n] 🚧= Arguments] 🕸 Debugger] 🌉 Environment] Synchronize] 🧤 Source] 🔲 <u>C</u> ommon]	
 C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application Launch Group (Deprecated) Launch Group (Deprecated) Parallel Application New_configuration 	Target System Configuration: Connection Type Local Remote Please	Please select a target system configuration Please select a target system configuration ALCF PBS BGP Batch ALCF PBS BGQ Batch BlueWaters PBS Batch Generic Grid Engine Batch Generic Grid Engine Batch Generic OpenMPI Interactive Generic PBS BGP Batch Generic PBS BGQ Batch Generic PBS BGQ Batch Generic PBS BICP Generic SLURM BGP Batch Generic Torque ALPS Batch Generic Torque Batch Generic Torque Batch	×
Filter matched 8 of 8 items		IBM LoadLeveler IBM LoadLeveler (Blue Gene) IBM Parallel Environment	
?		IBM Platform LSF IBM Platform LSF Interactive	

- 2. Select "Local" or "Remote" to specify where to launch an MPI application in the Connection Type field on the Resources tab. If user launches the MPI application on a remote VH, specify the VH information, too.
- 3. Specify the fields on the Basic Options tab to configure runtime options for the mpiexec command, which executes the MPI application.

Create, manage, and run configurations (Application): Application program not spece

Create, manage, and run configura (§ [Application]: Application program no	tions ot specifi	ed	Ť.
🗋 🗮 🗏 🖃 🛱 🖛	Name:	test_ve_mpi_bm	
type filter text 🛛 🕿	li⊒ Res	ources 🛽 🖺 Application 🖓 Arguments 🕸 Debugger 🔤 Environment Synchronize 🤤 Source	e 🔲 <u>C</u> ommon
C C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application Launch Group ► Launch Group (Deprecated) Parallel Application New_configuration	Target Conne Loc Loc Num VE N VE N Host Vie	System Configuration: Generic NECMPI Interactive Action Type al Please select a connection Advanced Options Advance	▼ New
Filter matched 8 of 8 items		Reve	rt Apply
?		Clo	se Debug

The following table shows the runtime options available on the Basic Options tab.

Option	Description
Number of	The total number of processes that execute an MPI application on VEs,
processes on VE	which is used in the –np option.
nodes	
VE node range	VE numbers on which an MPI application runs.
	Specify one VE number or a range of VE numbers, which is used in the
	-ve option. Without the specification, VE#0 is used.
	Examples of the specification:
	• One VE number : 1 (VE#1 is used)
	• Range of VE numbers: 0-7 (VE#0 through VE#7 are used)
Host file	File name that includes VHs from which an MPI application is
	launched. In the host file, VH names are descripted line by line. This is
	used in the -hostfile option. Without the specification of Host file and
	Host list, the local host is used.
Host list	Specify a comma-delimited list of VHs from which an MPI application
	is launched, which is used in the -hosts option. Without the
	specification of Host file and Host list, the local host is used.
	Example:
	host1,host2,host3

4. Click on "Advanced Options", select the checkbox "Extra Arguments", and input other options than those on the Basic Options tab.

Create, manage, and run configue	rations not specified
🗋 🗎 🗶 🛛 🔁 🌩 🗸	Name: test_ve_mpi_bm
type filter text	Resources Application ⋈= Arguments P Debugger Environment Synchronize Source "1
© C/C++ Application © C/C++ Attach to Application © C/C++ Postmortem Debugger © C/C++ Remote Application	Target System Configuration: Generic NECMPI Interactive Connection Type Local O Remote Please select a connection New
Group Launch Group Launch Group (Deprecated) Haunch Group (Deprecated)	Basic Options Advanced Options
i New_configuration	Arguments: -hosts host1,host2 -ve 0-3 -np 64 -ppn 2 Image: Comparison of the second
	Installation location Use default path Executables directory (bindir): Browse
	View Configuration Restore Defaults
Filter matched 8 of 8 items	Revert Apply
?	Close Debug

4.3 Setting of an Application

Specify a project name in the Project field and an executable file in the Application program field on the Application tab.

Create, manage, and run configurations

Create a configuration to debug a parallel application

5 5 1		- 2				
🗋 📔 🗶 🗏 🖻 🎲 🗶	Name: test_ve_mpi_bm					
type filter text	😫 Resources 🖹 Application 🛛 🕬= Arguments) 🕸 Debugger 🖾 Environment Synchronize 🦆 Source					
C/C++ Application		»1				
C/C++ Attach to Application	Project:					
C/C++ Postmortem Debugger	ve_mpi_bm					
Launch Group	Application program:					
Launch Group (Deprecated)	/usr/uhome/npdb/workspace/ve_mpi_bm/vempi.out	B <u>r</u> owse				
▼ Parallel Application	Copy executable from local filesystem					
;⇒test_ve_mpi_bm	Path to local executable:					
		Browse				
	Display output from all processes in a console view					
	Revert	Apply				
Filter matched 8 of 8 items						
		Dahua				
\bigcirc	Close	Debug				

The

4.4 Setting of the Debugger

- 1. Select "sxaurora-gdb-mi" in the Debugger backend field on the Debugger tab.
- 2. Uncheck the checkbox "Use built-in SDM if available for the target platform".

Create, manage, and run configura Create a configuration to debug a para	tions Ilel application	Ś
Image: Second Secon	Name: test_ve_mpi_bm Image: Application (M= Arguments (*) Debugger (mage Environment) Synchronize) (*) Source (mage Common Debugger: Debugger: SDM Image: Stop at main() on startup Debugger Options Debugger options Debugger backend sxaurora-gdb-mi Image: Jse built-in SDM if available for the target platform Path to SDM executable (if built-in is not used) /opt/nec/ve/npdb/bin/sdm Image: Advanced Options	Browse
Filter matched 8 of 8 items	Revert	Apply
?	Close	Debug

4.5 Setting of the Environment

If necessary, add Environment variables setting on Environment tab. These environment variables set on this tab are passed to the host where mpiexec command is launched (set in Connection Type field in Resources tab). If you want to pass the environment variables to all hosts where MPI application runs, please use NMPI_EXPORT environment variable supported by NEC MPI.

〕 🗎 🗶 │ 🗖 🆆 👻	Name: test_ve_mpi_bm		
type filter text 🛛 🗶	Resources Applicati	on 🕪 Arguments 🕸 Debugger 📧 Environment 🛛 Syr	chronize) 🦆 Source) 🔲 <u>C</u> ommo
C/C++ Application	Environment variables to s	et:	
C/C++ Attach to Application	Variable	Value	New
C/C++ Postmortem Debugger	ENV_VAR1	ABC	C-l-+t
C/C++ Remote Application	ENV_VAR2	XYZ	Select
🖶 Launch Group	NMPI_EXPORT	ENV_VAR1 ENV_VAR2	Edit
Launch Group (Deprecated)			
∃Parallel Application			Remove
New_configuration			
	• Append environment t	o native environment	
	Replace native enviror	nment with specified environment	
		A	
or metabod 9 of 9 itoms			Revert Apply

NEC Parallel Debugger supports the following environment variable.

Environment Variable	Contents				
NPDB_SDM_PORTRANGE	Range of the port number used for TCPIP connection among SDMs. Ports are separated by a colon. Default is 50000:50079.				

4.6 Other Settings

If necessary, add other settings on other tabs such as the Arguments tab.

4.7 Starting of a Debug Execution

Click on the Apply button to save the settings, and then the Debug button to start a debug execution. Please make sure that VHs from which an MPI application is launched have been configured to load the NEC MPI setup script by means like login shell so that the execution of MPI applications is enabled.

Create, manage, and run configurations

Create a configuration to debug a parallel application

□ 🗎 🗶 📄 🐡 🕶	Name: test_ve_mpi_bm	
type filter text C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application Launch Group Launch Group (Deprecated) Harallel Application New_configuration	Besources Application (# Arguments Debugger Environment Synchronize Source Common Debugger: SDM Stop at main() on startup Debugger Options	•
	Debugger backend sxaurora-gdb-mi Use built-in SDM if available for the target platform Path to SDM executable (if built-in is not used) /opt/nec/ve/npdb/bin/sdm Advanced Options	se
Filter matched 8 of 8 items	Revert	>
?	Close	g

T

5 Operations for Debugging

This chapter explains major operations for debugging with Eclipse PTP. For details, please refer to the "Parallel Development User Guide" published on the Eclipse official site.

5.1 Available Views

The following table shows the views available for the applications run on VE that are displayed by default in the Parallel Debug perspective. The name of view is displayed on the top of tab. In Editor, the name of the opened file is displayed on the top of tab.

View	Description
Parallel Debug	Perform a debug execution of multiple processes in a collective manner
Debug	Perform a debug execution of a target process and display the stack
	trace of the process
Editor	Display source code files
Breakpoints	Display the list of breakpoints
Variables	Display arguments and local variables in the stack selected on the
	target process
Arrays	Display information about arrays
Expressions	Test data by inputting conditional expressions
Signals	Display the list of signals the target process receives and the
	corresponding actions by the debugger.
Outline	Display definitions of variables and interfaces of functions in the file
	displayed in the Editor view
Console	Display the standard output and standard error from an application
Problems	Display error messages at the building of a project
Error Log	Display warning and error messages from plugins

5.2 Debugging of Multiple Processes in a Collective Manner

The Parallel Debug view enables debugging of multiple processes in a collective manner. Firstly, please select a set of processes targeted for debugging in the Parallel Debug view, and then a command to execute. If you do not select the set of processes, all the processes that are running (the root set) are selected by default.

뙗 Parallel Debug 없	1	90 F	3	R	<u>(%)</u> (%)	0	Q	0* -	() [×]	{ - }	{ ≅} ▼	
6964 - Root [4]												
🔬 6964	0 🔶 🔶 🔶	+										

The following table shows the available commands.

Icon	Command	Description
*	Resume Group	Resume execution of the target processes.
60	Suspend Group	Suspend execution of the target processes.
<i>₿</i> ₽	Step Into Group	Execute the current line of the source code on the target processes and suspend the execution at the beginning of the next line. If the current line includes reference of a procedure, suspend at the beginning of the first line of the procedure.
¢	Step Over Group	Execute the current line of the source code on the target processes and suspend the execution at the beginning of the next line. If the current line includes reference of a procedure, suspend at the beginning of the next line after the execution of the procedure.
<u>_@_</u>	Step Return Group	Resume execution of the procedure on the target processes from the line where the execution has been suspended, and suspend at the beginning of the next line of the line that invoked the procedure.
	Terminate Group	Terminate a debug session of the target processes.

5.3 Debugging of One Process

The Debug view enables debugging of a selected process. Firstly, please select a processes targeted for debugging in the Debug view, and then a command to execute.

The following is the detailed steps.

- 1. Double-click on a process in the Parallel Debug view to select the target of debugging. The stack trace of the selected process is added into the Debug view.
 - > A rhombus stands for a process. The rank of the process is displayed by placing the cursor over the rhombus.
 - > The rhombus corresponding to the process added into the Debug view is surrounded by a black border.

Parallel Debug 없	🀌 🗓 🖷 🦻 📀 🖄 🛞 O 🔍 🔍 Oʻ 🔻 Oʻ 🖂 🕲 🖛 🖃	
6964 - Root [4]		
👧 6964	8 🔁 🔁 🔶	
		- 11

2. Select a stack in the Debug view.

to Debug 23	×.		М	3	3	.e	 i⇒	ষ্ট	~	
The second secon										
🐨 🗗 Thread [1] (Suspended)										
<pre>1 main() bcast.c:28 4007d0</pre>										- 11
▼ ∂Process 2 (Suspended)										- 11
✓ IP Thread [1] (Suspended)										
≡ 1 main() bcast.c:28 4007d0										

3. Select a command to execute.

The following table shows the available commands.

Icon	Command	Description
	Resume	Resume execution of the target process.
00	Suspend	Suspend execution of the target process.
.₹.	Step Into	Execute the current line of the source code on the target process and suspend the execution at the beginning of the next line. If the current line includes reference of a procedure, suspend at the beginning of the first line of the procedure.
A	Step Over	Execute the current line of the source code on the target process and suspend the execution at the beginning of the next line. If the current line includes reference of a procedure, suspend at the beginning of the next line after the execution of the procedure.
_ 0 ² _	Step Return	Resume execution of the procedure on the target process from the line where the execution has been suspended, and suspend at the beginning of the next line of the line that invoked the procedure.
	Terminate	Terminate a debug session of the target process.
×	Remove All Terminated Launches	Delete the terminated debug session from the Debug view.

5.4 Display of the Stack Trace

The Debug view displays the stack trace of the target process.



The displayed information above is as follows:

```
Session Name
Rank of the MPI process
Thread [Thread number] (Execution status: The reason for suspension, if suspended)
Stack number Procedure name File name: Line number Address
Stack number Procedure name File name: Line number Address
:
```

5.5 Reference of Variable Information

Select a target process in the Parallel Debug view or Debug view, and then a stack of the process in the Debug view. The Variables view displays arguments and local variables in the selected stack.

®o Breakpoints ରିଙ୍କି Expressio	ons ^{(x)=} Variables ⊠	🗟 Signals	約 🛋 📄					
Name	Value	lue						
🗢 argc	0	0						
🕨 🔷 argv	9d36	9d3633ed						
🕨 🔷 buf	[0 -	[0 - 1024]						
🔷 myrank	1							
nprocs	4							

Information about a variable including external one is displayed on the source code in the Edit view by placing the cursor over the variable.

33	MPI_Init(0,0);	
34	<pre>MPI_Comm_rank(MPI_COMM_WORLD,</pre>	&myrank);
35	<pre>MPI_Comm_size(MPI_COMM_WORLD,</pre>	&nprocs);
36		$1 \cdot \text{pprocs} = 4$
37	calculate_data(buf, myrank);	1. 101003 - 4

5.6 Setting of Breakpoints

A breakpoint is enabled or disabled by double-clicking on the left of a source line number in the Edit view. The breakpoint is set for all processes by default.



The Breakpoints view displays the list of enabled breakpoints.



5.7 Termination of a Debug Session

To terminate a debug session, click on the button in the Debug view to terminate the target processes, and then click on the button to delete the debug session.

File Edit Navigate Search	Project Run Window Help
	💷 🗈 💷 🕺 R. 🤜 R. 🧮 🕱 🎋 🕶 🔿 🕶 🚱 🕶 😂 🖋 🖛
🗱 Parallel Debug 🛿	🐌 🗓 🖷 🧏 🥱 🖉 😸 O 🔍 🔍 O' ▾ O' (-) 😂 ▾ 🖓 🗖
26550 - Root [4]	
2 26550	0 4 4 4
🎋 Debug 🛿	🎇 🕨 💷 🛤 D. 🕫 🖈 😾 🔻 🗖 🗖
<pre>lileterminated>test_ve_mpi</pre>	bi Remove All Terminated Launches 0>Runtime process 20550

Next, click on the <u>button</u> button in the Parallel Debug view to terminate the processes not targeted for the debug, and then right-click on the debugging job to delete the debug session.

File Edit Navigate	Search Project	Run	Window	Hel	р								
📑 🗕 🔚 💼	● - 📮 🕨	00 0	N 3	3	r	5 F	*	- 0	•	8 -	0	1	•
Parallel Debug X	10		10	ê	9) {}	Q	0,	0* -	() [×]	{-}	(部) 🤜	, –	
26550 - Root [4]													
🧔 26550	0 -	₽ 4₽ 4	₽ ₽	_									
	🐘 🎇 Remove A	ll Term											
													- 1
													- 1
													- 1
L													
* Dobug S			52a 🛛	► m		• –		e)		ы –	v	,	-
No Debug 22			36 0		•••	V V		. U.S 9		· · ·	-		-

6 Notices and Restrictions

6.1 Notices

- Please use the Oxygen v3 version of Eclipse PTP as NEC Parallel Debugger supports only the version.
- The following message would be displayed in the Console view at the beginning of debugging. Please ignore it as there is no effect on debugging operations. (This message appears when gdb attaches to the mpiexec command at the beginning of debugging)

```
Missing separate debuginfos, use: debuginfo-install glibc-2.17-157.el7.x86_64
```

• The watch point feature and trace point feature are not supported.

6.2 Miscellaneous

- It is also possible to debug non-MPI applications for VE, using the CDT plugin or Fortran plugin for Eclipse. Please specify gdb for VE (/opt/nec/ve/bin/gdb) as a debugger on the Debugger tab at the creation of a Debug Configuration. For details, please refer to the following documents published on the Eclipse official site.
 - > C/C++ Development User Guide
 - > Fortran Development User Guide

Appendix A: Change Log

Edition	Issue	Category	Modified Item	Chapter/Section				
1	May 2018							
2	Dec 2018	DELETE	Restriction for referring variables.	6.2				
3	May 2019	UPDATE	Additional description	1.2				