NEC Network Queuing System V (NQSV) User's Guide

[Accounting & Budget Control]

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.

Preface

The NEC Network Queuing System V (NQSV) User's Guide [Accounting & Budget Control] describes how to use the accounting function and budget control function of NQSV.

Remarks

(1) This manual conforms to Release 1.00 and subsequent releases of the NQSV.

(2) All the functions described in this manual are program products. The typical functions of them conform to the following product names and product series numbers:

Product Name	product series numbers
NEC Network Queuing System V (NQSV) /ResourceManager	UWAF00 UWHAF00 (Support Pack)

(3) UNIX is a registered trademark of The Open Group.

(4) Intel is a trademark of Intel Corporation in the U.S. and/or other countries.

(5) OpenStack is a trademark of OpenStack Foundation in the U.S. and/or other countries.

(6) Red Hat OpenStack Platform is a trademark of Red Hat, Inc. in the U.S. and/or other countries.

(7) Linux is a trademark of Linus Torvalds in the U.S. and/or other countries.

(8) Docker is a trademark of Docker, Inc. in the U.S. and/or other countries.

(9) InfiniBand is a trademark or service mark of InfiniBand Trade Association.

(10) Zabbix is a trademark of Zabbix LLC that is based in Republic of Latvia.

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

About This Manual

This manual consists of the following chapters:

Chapter	Title	Contents
1	Overview of NQSV Accounting & Budget Control	Overview
2	Environment Creation and Operation	Installation of Accounting & Budget Control, Creation of Environment, and Starting- Stopping of Demon
3	Accounting	Referencing of Accounting Data, Daily Report, Monthly Report, Saving Accounting Data
4	Budget Management	Setting Accounting Rare, Referencing and Setting of Budget
5	Appendix	Appendix

Related manual at using this manual are as follows.

G2AD01E NQSV User's Guide [Introduction]

G2AD02E NQSV User's Guide [Management]

G2AD03E NQSV User's Guide [Operation]

G2AD04E NQSV User's Guide [Reference]

G2AD05E NQSV User's Guide [API]

G2AD07E NQSV User's Guide [JobManipulator]

How to Read This Manual

Symbols and Terms

The table below lists the text conventions used in this manual.

Elision marks	...	Indicates that you can repeat the preceding element as needed.
Vertical bar	|	Divides optional or required parameters or keywords.
Braces	{ }	Indicates that you must choose one of the parameters or keywords in braces.
Angular parenthesis	[ ]	Indicates that you can omit parameters or keywords in square brackets.

Glossary

Term	Definition
Vector Engine (VE)	The NEC original PCIe card for vector processing based on SX architecture. It is connected to x86-64 machine. VE consists of more than one core and shared memory.
Vector Host (VH)	The x86-64 architecture machine that VE connected.
Vector Island (VI)	The general component unit of a single VH and one or more VEs connected to the VH.
Batch Server (BSV)	Resident system process running on a Batch server host to manage entire NQSV.
Job Server (JSV)	Resident system process running on each execution host to manage the execution of jobs.
JobManipulator (JM)	JobManipulator is the scheduler function of NQSV. JM manages the computing resources and determines the execution time of jobs.
Accounting Server	Accounting server collects and manages account information and manages budgets.
Request	A unit of user jobs in the NQSV. It consists of one or more jobs. Requests are managed by the Batch Server.
Job	A job is an execution unit of user job. It is managed by Job Server.
Logical Host	A logical host is a set of logical (virtually) devided resources of an execution host.
Queue	It is a mechanism that pools and manages requests submitted to BSV.
BMC	Board Management Controller for short. It performs server management based on the Intelligent Platform Management Interface (IPMI).
HCA	Host Channel Adapter for short. The PCIe card installed in VH to connect to the IB network.
IB	InfiniBand for short.
MPI	Abbreviation for Message Passing Interface. MPI is a standard for parallel computing between nodes.
NIC	Network Interface Card for short. The hardware to communicate with other node.

CONTENTS

Chapter 1. Overview of NQSV Accounting & Budget Control

1.1. Introduction

NQSV Accounting & Budget Control function is the software that collects and manages accounting information using the accounting server. The accounting server also supports a budget management function that makes use of accounting data. The accounting server calculates charges based on the configured accounting rate and accumulated accounting data. And if budget has been exceeded, it is possible to refuse to submit NQSV requests and to refuse to make resource reservation section.

1.2. Components

NQSV Accounting & Budget Control consists of the following components.

- Accounting Server

  The Accounting server is the server to collect all accounting data. And the accounting data, budget setting and accounting rate can be referenced on the Accounting server host.

  Accounting server can be located with the Batch server or JobManipulator.

- Accounting Monitor

  The Accounting monitor is the component to send accounting data to the Accounting server. It sends the job accounting data and the request accounting data output by the Batch server, and also sends the reservation accounting data output by JobManipulator.

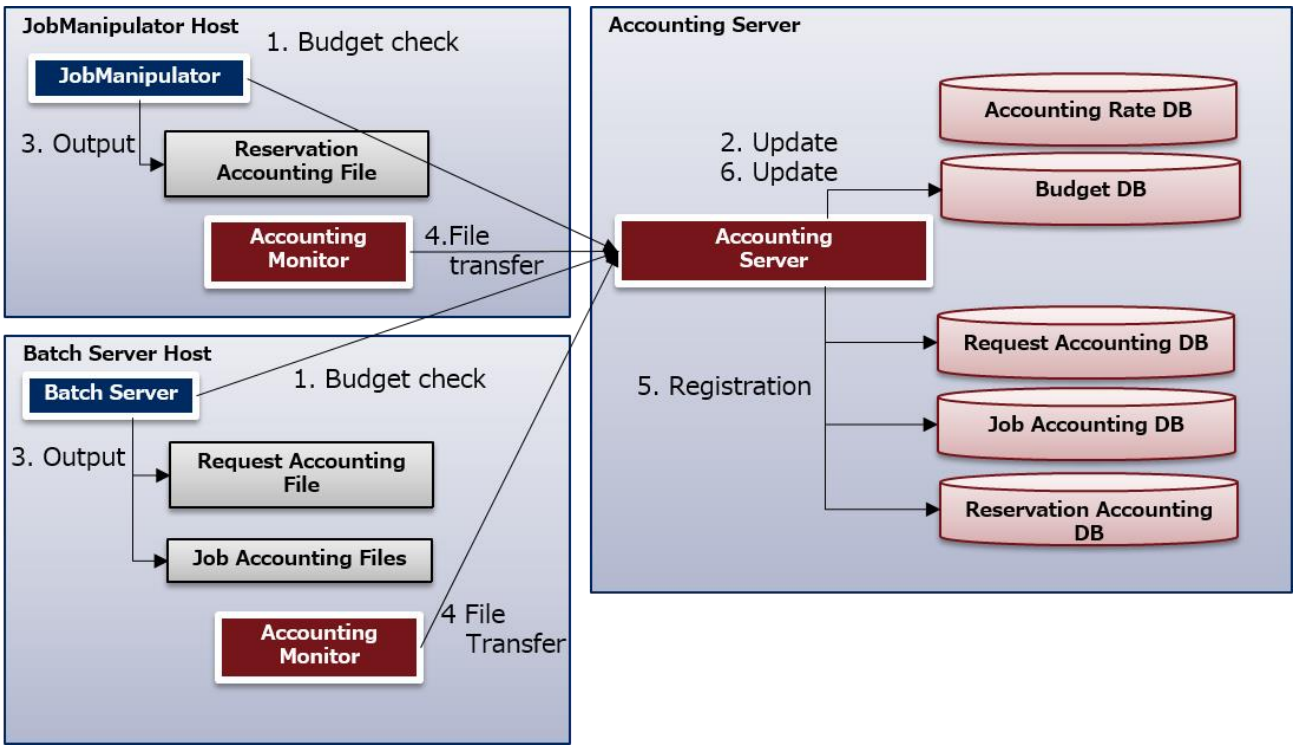
  The Accounting monitor must be located on each Batch server host and JobManipulator host.

1.3. Accounting Data Flow

NQSV can output the request accounting data and the job accounting data for each execution of NQSV requests. Accounting monitor collects these data and send them to the Accounting server. Also, the JobManipulator can output the reservation accounting data for each Resource Reservation Sections. And the Accounting monitor collects the reservation accounting data and send it to the Accounting server.

The Accounting server updates each accounting database and the charge data by the accounting data received from the Accounting monitor.

The following shows the accounting data flow.



1. Budget checking

   Accounting server checks the budget overruns at the request submission and at the creation of the Resource Reservation Sections.

2. Update the charge data

   Accounting server estimates the charge from the requested amount of resources of the request or the Resource Reservation Section, and check if it will cause the budget overruns. When it is within budget, the Accounting server updates the value of estimated charge in the budget database.

3. Output of accounting files

   Each accounting files are output by the Batch server of the JobManipulator at the time of termination of the request or the Resource Reservation Section.

4. Transferring accounting data to the Accounting server

   The Accounting monitor sends the accounting files created in Step 3 to the Accounting server

5. Registering on the Accounting server.

   The accounting data is extracted from the received data and registered to the Accounting DB on the Accounting server.

6. Update the charge data

   Accounting server calculates the actual charge from the received accounting data, and updates the charge data in the budget database.

Chapter 2. Environment Creation and Operation

2.1. Installation

To use NQSV Accounting & Budget Control, the Accounting server software should be installed on the server host and the Accounting monitor software should be installed on Batch server host and on each JobManipulator host. Please refer to NEC Network Queuing System V (NQSV) User's Guide [Introduction] about each package, and install the each package in accordance with the contents of the guide.

2.1.1. Installing Accounting server

Install NQSV/ResourceManager package on the Accounting server. The Accounting server daemon and the Accounting server commands are included in this package.

2.1.2. Installing Accounting Monitor

Install NQSV/ResourceManager package on each host that runs Batch Server and JobManipulator. The Accounting monitor daemon and the Accounting monitor commands are included in this package.

2.1.3. Installing AUI

The AUI (Accounting User Interface) are commands which access the Accounting server and display information. Install NQSV/Client package on the host that runs AUI. The AUI commands are included in this package.

2.2. Environment Creation

Before using the NQSV Accounting & Budget Control, perform the following environment settings. Please refer to the "Appendix A - Others" section for precautions regarding "Environment Creation".

2.2.1. Accounting Server Setting

- Editing the configuration file

  Set the following items in /etc/opt/nec/nqsv/asvd.conf file to activate the server daemon in the Accounting Server as needed.

Property	Property Name	Descriptions
RECV_PORT_FOR_ACCT	port number for receiving account	Set the port number that the Accounting server daemon uses for communication with the

	data	Accounting monitor. The default port number is 6542. It is unnecessary to change the default setting unless other applications use this port number.
ALLOW_CLIENTS	IP address of connectable hosts	Specify the IP address of the host which is allowed to connect with this Accounting server. A connection from all hosts is permitted by default. Multiple addresses can be specified by delimited with comma. Also, a range of addresses can be specified using a hyphen (-).
SBU_CHECK	Setting ON/OFF of Budget Management Function	Set "ON" to enable the budget management function; otherwise, set "OFF". "OFF" is set as default.
RECV_PORT_FOR_SBU	Port Number for receiving budget check request	Set the port number for budget management that the server daemon uses for communication with Batch server and JobManipulator. The default port number is 4595. It is unnecessary to change the default setting unless other applications use this port number.
LOG_FACILITY	Facility of syslog	Set facility of syslog. "LOG_LOCAL0" is set as default.
SBU_ORDER	order of priority to charge for resource usage	The Accounting server charges each resource usage to the target of user, group or account-code by the order specified in this item. The specification consists of the following characters and the left one is treated as a target first. A: Account-code U: User G: Group The target which is not included in this specification is not charged. The default setting is "GUA".
ACCT_DIR	Directory of accounting files	The Accounting server saves the received accounting files into this directory. The default directory is "/var/opt/nec/nqsv/asv/master".
LOCK_DIR	Directory of lock files	The Accounting server creates lock files in this directory. The default directory is "/var/opt/nec/nqsv/asv/master".
DBSIZE_LIMIT	Maximum size of account database file	Set the maximum size of each database file of the account in MB. The default is 1800MB. The setting range is 1 to 131072. If this size is exceeded, database registration of accounting data will be stopped. Increasing the maximum size of account database file may reduce the account aggregation performance and display performance, so 1800MB is recommended.
VEACCT_TIME_MARGIN	VE process accounting margin	The processes of VE job may end after the end time of the job. Set the adjustment margin of the time in seconds so that the process account output from VEOS is correctly aggregated to the

		job account. By this setting, the process account of the process that ended after the specified time has elapsed after the end time of the job can be aggregated to the job account. The default is 60 seconds.

- Setting for using budget management

  To use the NQSV budget management function, set prime time, non-prime time and holidays to /etc/opt/nec/nqsv/asv_holidays.

```
* Prime/Nonprime Table
*
* Curr  Prime   Non-Prime
* Year  Start   Start
*
  2016  0800    1730
*
* Day of    Calendar    Company
* Year      Date        Holiday
*
    1       Jan 1       New Year's Day
   47       Feb 16      Presidents' Day
  100       Apr 10      Spring Holiday
  145       May 25      Memorial Day
  184       Jul 3       Independence Day
  250       Sep 7       Labor Day
  330       Nov 26      Thanksgiving Day
  331       Nov 27      Day after Thanksgiving
  359       Dec 25      Christmas
```

  Please specify start time of prime time and non-prime time to the "Prime Start" and "Non-Prime Start" columns respectively in HH24MI format. Holidays can be specified after "Day Of Year". In the "Day Of Year" column, specify the number of days counted from New Year's Day. Please specify the comments for each "Day Of Year" in the "Calendar Date" and "Company Holiday" columns. Days set as holidays are treated as non-prime time throughout the day regardless of the prime time start time.

- User map file

  User can set or refer various data of the Accounting server under the NQSV user privilege. Therefore, when the Accounting server is placed on another host than the server that the BSV is installed, the Accounting server host is required to be set as follows.

  - User and group accounts are set as same as the BSV host.
  - /etc/opt/nec/nqsv/nqs_user.map file of the BSV host must be copied to the Accounting server host.

Please refer to NEC Network Queuing System V (NQSV) User's Guide [Management] for the format of the /nqs_user.map file.

- Account-code file

  When the budget management function is used with account-code, the account-code must be set in /etc/opt/nec/nqsv/acctcode file as follows.

```
acctcode1:user1,user2,user3
acctcode2:user1,user4
```

  Each line has a name of account-code and user names who can use the account-code in the format as

  name_of_account-code:user_name1,user_name2, ….

  Please note that no space character is permitted before and after colon and comma.

  The maximum length of name_of_account-code is 15 bytes.

- Setting automatic start the Accounting Server daemon

  To start the Accounting Server daemon automatically when the system starts, set as follows.

```
# systemctl enable nqs-asv.service
```

2.2.2. Accounting Monitor Setting

To start the Monitor, the following settings are required for Batch server host and JobManipulator host to collect accounting information.

- Editing the configuration file

  Set the following to the configuration file /etc/opt/nec/nqsv/acmd.conf to activate the Accounting monitor daemon.

Property	Property Name	Descriptions
LOG_FACILITY	Facility of syslog	Set facility of syslog. "LOG_LOCAL0" is set as default.
CONNECT_INTERVAL	Interval of sending account data	Set interval of sending account data from 1 second to 86400 seconds. The default interval is 30 seconds.
TURN_SIZE	The threshold file size for accounting file switching	The threshold file size can be set in MB unit. The value can be set from 0 to 2047, and the default value is 500MB.
TURN_SAVE	The number of generations of accounting files to be saved by switching.	The accounting files over the threshold size are saved. And the maximum number of the saved files can be set in this item. The value can be set from 0 to 99, and the default value is 10.
RACCT_FILE	Request Accounting File	Set the location of the request accounting file to be processed by the

		Accounting monitor daemon. The default location is /var/opt/nec/nqsv/bsv/account/reqacct.
JOBACCT_FILE	Job Accounting File	Set the location of the job accounting file to be processed by the Accounting monitor daemon. The default location is /var/opt/nec/nqsv/acm/jacct/jobacct.
RSVACCT_FILE	Reservation Accounting File	Set the location of the reservation accounting file to be processed by the Accounting monitor daemon. The default location is /var/opt/nec/nqsv/acm/rsvacct/rsvacct.
LOCK_DIR	Directory for lock files	The Accounting monitor creates lock files in this directory. The default directory is "/var/opt/nec/nqsv/acm".
SERVER_PORT_FOR_ACCT	Port number for sending account data	Set the port number for communication with the Accounting server daemon. The default port number is 6543. It is unnecessary to change the default setting unless the RECV_PORT_FOR_ACCT in the configuration file of the Accounting server is changed.
ACCTFILESIZE_LIMIT	Maximum size of accounting file	Set the maximum size of request accounting file, job accounting file, and reservation accounting file in MB. The default is 131072MB (128GB). If this size is exceeded, transfer of accounting data to Accounting server will be stopped.

- Setting automatic start of the Accounting monitor

  To start the Accounting monitor daemon automatically when the system starts, set as follows.

```
# systemctl enable nqs-acm.service
```

2.2.3. Batch server

To use NQSV Accounting & Budget Control, perform the following environment settings to the Batch server.

- Setting request accounting

  NQSV Accounting & budget control can collect NQSV request accounting information. To collect request accounting information and to use the NQSV budget control function, it is necessary to configure Batch server to output request accounting files. Please configure this

request accounting setting using the qmgr(1M) command on the Batch server host as follows.

```
# /opt/nec/nqsv/bin/qmgr -P m
Mgr: set batch_server req_account_file = /var/opt/nec/nqsv/bsv/account/reqacct
Set Req_Account_File Path
Mgr: set batch_server req_account ON
Set Req_Account ON.
Mgr: exit
```

* Do not set request accounting to OFF during NQSV Accounting & budget control operation.

- Setting job accounting

  NQSV Accounting & budget control can collect NQSV job accounting information from Batch server. By default, the Batch server outputs the job account file to the /var/opt/nec/nqsv/acm/jacct directory as a “jobacct” file. The output directory of job accounting files can be changed to other directory if necessary. To change the directory, after creating the output directory, use "set batch_server jacct_dir" sub-command of qmgr as follows.

```
# mkdir -p /usr/share/nqs/jacct
# /opt/nec/nqsv/bin/qmgr -P m
Mgr: set batch_server jacct_dir = /usr/share/nqs/jacct
Set Jacct_Directory Path.
Mgr: exit
```

  By setting the directory of job accounting files illustrated above, the directory is changed to /usr/share/nqs/jacct.

- Setting for Accounting server

  For NQSV Accounting & budget control, the information about the Accounting server must be set to the environment of Batch server so that the Batch server and JobManipulator can connect to the Accounting server.

  To set the Accounting server information, set the host name of the Accounting server first as follows.

```
# /opt/nec/nqsv/bin/qmgr -P m
Mgr: set batch_server acct_server host = acct-server
Set Acct Server Host.
Mgr: exit
```

  * *acct-server* is Accounting server's hostname.

And additionally set the port number if the port number which the Accounting server wait for the budget checking request from Batch servers or JobManipulators has been changed from the default value.

```
# /opt/nec/nqsv/bin/qmgr -P m
Mgr: set batch_server acct_server port = 1234
Set Acct Server Port Number.
Mgr: exit
```

The port number must be same as the number which is set as the port number for receiving budget check request at the Accounting server settings.

- Setting for resource reservation accounting

  When the accounting function or budget control function for resource reservation sections generated by JobManipulator are needed, please set the resource reservation accounting function to ON by qmgr command as follows.

```
# /opt/nec/nqsv/bin/qmgr -P m
Mgr: set batch_server reservation_acct = ON
Set Reservation Accounting
Mgr: exit
```

  By this setting, the budget checking for resource reservation sections created by JobManipulator, the collection of accounting data for the resource reservation sections and calculating the charge of resource reservation sections are enabled.

  However, the accounting functions for resource reservation sections can be enabled only for the reservation sections which are created by specifying queue name and number of hosts.

  When the resource reservation accounting is set to ON, resource usage is charged by resource reservation section, therefore each submitted NQSV requests using the resource reservation sections are not charged.

  And furthermore, by the resource reservation accounting is set to ON, JobManipulator outputs resource reservation accounting records to the reservation accounting file which is used for accounting functions of the Accounting server. The default path of the resource reservation accounting file is set to /var/opt/nec/nqsv/acm/rsvacct/rsvacct. If the resource accounting file path is changed to other than the default one, the file path must be set to the Batch server setting using qmgr command as follows.

```
# /opt/nec/nqsv/bin/qmgr -P m
Mgr: set batch_server rsvacct_file = /usr/share/nqs/rsvacct
Set Reservation Acct File.
Mgr: exit
```

  By setting the file path of reservation accounting file illustrated above, the file is changed to /usr/share/nqs/rsvacct. This setting must be same as the setting of RSVACCT_FILE in the Accounting monitor configuration file.

Note that the settings of resource reservation accounting function described above must be executed while the JobManipulator daemon program is down.

- Setting for Budget control

  NQSV Accounting & budget control function supports pay-as-you-go charge based on resource usage. By using the budget control function, budget overruns can be checked when submitting requests and when creating resource reservation sections. When using the budget control function, it is necessary the budget control function of the Accounting Server should be enabled. (Please refer to 2.2.1 for details.)

  To enable the budget control function, use the qmgr command's "set batch_server nqs_budget_chk" sub-command as follows.

```
# /opt/nec/nqsv/bin/qmgr -P m
Mgr: set batch_server nqs_budget_chk = ON
Set NQS Budgetchk
Mgr: exit
```

  As shown above, when the budget control function is set to "ON", budget overruns are checked both when submitting requests and when creating resource reservation sections.
  If you need the budget overrun check only for request submissions or creating resource reservation sections, please add the "target=*target*" option to the setting as follows.

```
# /opt/nec/nqsv/bin/qmgr -P m
Mgr: set batch_server nqs_budget_chk = ON target = ( request )
Set NQS Budgetchk
Mgr: exit
```

  This example shows the setting to perform the budget overrun check only for request submission. To enable the budget overrun check only for creating resource reservation sections, please specify "target = reservation".
  When the budget overrun check is performed for creating resource reservation sections, the requests that are submitted into resource reservation sections are not checked to be with in the budget.
  The default setting of the budget control function is OFF. If you don't need to use this NQSV budget control function (for example, when a flat rate billing is used), please use the default settings.

- Display of settings

  The Batch server's settings for the NQSV Accounting & budget control function can be displayed by qstat -Bf option.

  Example:

```
$ /opt/nec/nqsv/bin/qstat -Bf
Batch Server: batch_server
    NQSV Version = R1.00 (linux)
    Batch Server State = Active
    Batch Server Machine ID = 1
    Logfile Path       = /var/opt/nec/nqsv/batch_server_log
    Logfile Level      = 1
    Logfile Save Count = 3
    Logfile MAX Size   = UNLIMITED
    Accounting Server Host Name   = localhost
    Accounting Server Port Number = 4595
    Jacct directory = /var/opt/nec/nqsv/acm/jacct
    Budget function              = request,reservation
    Request Accounting           = ON
    Request Accounting File Path = /var/opt/nec/nqsv/bsv/account/reqacct
    Reservation Accounting       = ON
    Reservation Accounting File  = /var/opt/nec/nqsv/acm/rsvacct/rsvacct
    CPU Eco Mode = OFF
    Specify Group for Request = ON
    VM Control Timeout = 600
        :
        :
```

Location	Description
Accounting Server Host Name Accounting Server Port Number	The host name and the port number of the Accounting server are displayed.
Jacct directory	The directory path name in which the Job Accounting files are stored is displayed.
Budget function	The setting of Budget control function is displayed.
Request Accounting	The setting of the Request Accounting is displayed.
Request Accounting File Path	Path name of the Request Accounting file is displayed.
Reservation Accounting	The setting of the Reservation Accounting is displayed.
Reservation Accounting File	Path name of the Reservation Accounting file is displayed.

2.2.4. AUI Setting

To enable the AUI commands of NQSV Accounting & budget control function which can be used to refer accounting and budget information, the following settings should be needed on the hosts that NQSV/Client is installed.

- ssh setting

  Create "scacct" user. And to use the AUI commands, it is necessary to set the ssh connection so

that "scacct" user can use ssh from the host on which NQSV/Client is installed to the Accounting server host without password.

A) Create scacct user on client host.

```
# groupadd -rf scacct

# useradd -M -r -d /opt/nec/nqsv/scacct/.home -g scacct scacct

# mkdir -p /opt/nec/nqsv/scacct/.home

# chown scacct:scacct /opt/nec/nqsv/scacct/.home
```

B) Make a public key of the scacct user.

```
# su - scacct

% ssh-keygen
* Input Enter key 3 times so as to make NULL passphrase.
```

C) Confirm that /opt/nec/nqsv/scacct/.home/.ssh directory, id_rsa and id_rsa.pub file are made.

```
% ls -l /opt/nec/nqsv/scacct/.home/.ssh
-rw------- 1 scacct scacct 1675 Feb 19 14:27 2014 id_rsa
-rw-r--r-- 1 scacct scacct  419 Feb 19 14:27 2014 id_rsa.pub
```

D) Copy the public key of the scacct user to the Accounting server.

```
% su
Password: ********

# ssh acct-server 'mkdir -p /opt/nec/nqsv/scacct/.home/.ssh'

# scp -p /opt/nec/nqsv/scacct/.home/.ssh/id_rsa.pub \
acct-server:/opt/nec/nqsv/scacct/.home/.ssh/id_rsa.pub.send
```

* acct-server is Accounting server's hostname.

E) Log in to the Accounting server by a superuser and establish a public key file.

```
# ssh root@acct-server

# cat /opt/nec/nqsv/scacct/.home/.ssh/id_rsa.pub.send \
>> /opt/nec/nqsv/scacct/.home/.ssh/authorized_keys

# chown scacct:scacct /opt/nec/nqsv/scacct/.home/.ssh/authorized_keys

# chmod 600 /opt/nec/nqsv/scacct/.home/.ssh/authorized_keys
```

```
# chmod 700 /opt/nec/nqsv/scacct/.home/.ssh/

# chown scacct:scacct /opt/nec/nqsv/scacct/.home/.ssh/

# exit
```

F) Confirm that you can login to the Accounting server without password by the scacct user.

```
# su - scacct
# ssh acct-server
```

- sudo setting

  Add the following description to the setting of sudo using the visudo command as a super user on the client host.

```
## ACCT-CUI
Cmnd_Alias NQSACCTCUI = /opt/nec/nqsv/bin/*_cui
%group_name ALL=(scacct) NOPASSWD: NQSACCTCUI
```

  * group_name is group of the user who uses AUI commands on Accounting server.

2.3. Starting and Stopping

Please start daemon after the operating environment is configured. In this section, how to start and stop the server and monitor daemons is described.

2.3.1. Starting and Stopping Accounting server

To start and stop the Accounting server daemon, execute as follows.

- To start

```
# systemctl start nqs-asv.service
```

- To stop

```
# systemctl stop nqs-asv.service
```

2.3.2. Starting and Stopping Accounting monitor

To start and stop the Accounting monitor daemon, execute as follows.

- To start

```
# systemctl start nqs-acm.service
```

- To stop

```
# systemctl stop nqs-acm.service
```

Chapter 3. Accounting

The data collected with NQSV Accounting & budget control function can be referred to and edited using provided server commands. This chapter explains those commands.

The following commands can be used from a front-end machine by the setting described in 2.2.4. AUI. Please refer to racctreq(1)/racctjob(1)/racctrsv(1)/rbudgetcheck(1) online manuals for details.

- scacctreq
- scacctjob
- scacctrsv

3.1. Referencing Request Accounting Data

You can refer to the request accounting data by the scacctreq(1) command. Each item of the request accounting data is as the following table.

Name	Description	Option			Sort ID
		Display	Non Display	Search	
REQUEST ID	Request ID			-I	REQUEST_ID
REQUEST NAME	Request name	--long-request-name (*10)		-n	REQUEST_NAME
USER NAME	Submission User name			-u	USER_NAME
GROUP NAME	Group name	-Y		-g	GROUP_NAME
ACCOUNT CODE	Account code	-X		-x	ACCOUNT_CODE
QUEUE NAME	Submission Queue name			-Q	QUEUE_NAME
QUEUED TIME	Submit time		-2	-{s|e}	QUEUED_TIME
	Submit date and time	-2			
START TIME	Start time		-2	-{S|b}	START_TIME
	Start date and time	-2			
END TIME	End time		-2	-{A|E}	END_TIME(*9)
	End date and time	-2			
CPU (SECS)	CPU consumption time (system + user) (sec.)		-t		CPU

CPU (SECS) SYS	CPU consumption time (system) (sec.)	-t			CPU_SYS
CPU (SECS) USER	CPU consumption time (user) (sec.)	-t			CPU_USER
REAL (SECS)	Elapsed time (sec.) (*1)				REAL
REQUEST PRTY	Priority of the request	-p			REQUEST_PRTY
NICE	Nice value	-p			NICE
REQELAPS TIME(S)	Elapse Time Limit Value (sec.)	-T			REQELAPS_TIME
REQCPU TIME(S)	Requested CPU time per job (sec) In a hybrid request : Average requested CPU time for each job in the request (sec)	-q			REQCPU_TIME
REQCPU NUM	Number of requested CPU per job In a hybrid request : Average number of requested CPU for each job in the request	-q			REQCPU_NUM
REQMEM SIZE(K)	Requested memory size for each job (KB) In a hybrid request : Average requested memory size for each job in the request (KB)	-q			REQMEM_SIZE
REQGPU NUM	Number of requested GPU per job In a hybrid request : Average number of requested GPU for each job in the request	-q			REQGPU_NUM
STATUS	Exit status (simple format)				EXIT_STAT
EXIT STAT	Exit status (*3)	-h			
KCORE MIN	Total memory	-k			KCORE_MIN

	consumption (KB * MIN) (*8)				
MEAN SIZE(K)	Average memory consumption (KB)	-m			MEAN_SIZE
MAXMEM SIZE(K)	Max. memory consumption (KB)	-M			MAXMEM_SIZE
V KCORE MIN	Total virtual memory consumption (KB * MIN) (*8)	--vmem			V_KCORE_MIN
V MEAN SIZE(K)	Average virtual memory consumption (KB)	--vmem			V_MEAN_SIZE
V MAXMEM SIZE(K)	Max. virtual memory consumption (KB)	--vmem			V_MAXMEM_SIZE
QUE TYPE	Queue type	-K			QUE_TYPE
EHOST NUM	Number of execution hosts	-C			EHOST_NUM
JOBS	Number of jobs	-C			NUM_JOBS
SUBREQ	Number of sub requests (only parametric request)	-C			NUM_SUBREQ
RERUN COUNT	Rerun count	-r			RERUN_COUNT
PRERUN COUNT	Rerun count of the parent request	-r			PRERUN_COUNT
TEMPLATE NAME	Template name	-L		-l	TEMPLATE_NAME
crname (*2)	Requested amount of each custom resource. Requested amount for the request, when consumption unit of the custom resource is a request, Requested amount for each job, when consumption unit of the custom resource is a job.	-F		-c	CR_*crname*
	Requested amount of the specified custom resource	-G			
crname TOTAL	Total usage of each	-O			None

	custom resource				
	Total usage of the specified custom resource	-j			None
crname MEAN	Average usage of each custom resource	-O			None
	Average usage of the specified custom resource	-j			None
crname MAX	Maximum usage of each custom resource	-O			None
	Maximum usage of the specified custom resource	-j			None
UNIT(*4)	Unit of custom resource	-F -G -O -j			None
REQVE NUM(*5)	Number of requested VE nodes (*3)	-V			REQVE_NUM
RSVVE NUM	Number of reserved VE nodes (*3)	-V			RSVVE_NUM
VE CPU(SECS)	CPU consumption time on VE nodes [SEC] (*3)	-V –veps (*6)			When displayed with -V :VE_CPU when displayed with --veps: VE_CPU_PS
VE KCORE MIN	Total memory consumption on VE nodes [KB * MIN] (*3)	-V --veps (*6)			When displayed with -V : VE_KCORE_MIN when displayed with --veps: VE_KCORE_MIN_PS
VE MEAN SIZE(K)	Average memory consumption on VE nodes [KB] (*3)	-V --veps (*6)			When displayed with -V :VE_MEAN_SIZE when displayed with --veps: VE_MEAN_SIZE_PS
VE MAXMEM SIZE(K)	Max. memory consumption on VE nodes [KB] (*3)	-V			VE_MAXMEM_SIZE
REQCPUTIME SUM(S)	Total requested CPU time for each job in the	-z			REQCPUTIM_SUM

	request [SEC]				
REQCPUNUM SUM	Total number of requested CPUs for each job in the request	-z			REQCPUNUM_SUM
REQMEMSIZE SUM(M)	Total memory size required for each job in the request [MB]	-z			REQMEMSIZE_SUM
REQGPUNUM SUM	Total number of requested GPUs for each job in the request	-z			REQGPUNUM_SUM
HW FAILURE	Hardware trouble flag	--hw-failure			HW_FAILURE
VE TIME SLICE	Timeslice value on VE nodes[us] (*7)	--veps-perform			VE_TIME_SLICE
VE MFLOPS	MFLOPS value on VE nodes (*7)	--veps-perform			VE_MFLOPS
VE MFLOPS (REAL)	MFLOPS(REAL) value on VE nodes (*7)	--veps-perform			VE_MFLOPS_REAL
VE FPEC(K)	Floating point data element count on VE nodes (*7)	--veps-prog			VE_FPEC
VE SYSTEM CALL	Number of system calls on VE nodes (*7)	--veps-prog			VE_SYSTEM_CALL
VE-VH DATA TRANS	Data transfer amount between VE-VH (*7)	--veps-prog			VE_VH_DATA_TRANS
NUMBER OF VE PROCESS	Number of VE processes (*7)	--veps-prog			VE_PROC_NUM
INSTRCT(K)	Number of execution instructions on VE nodes (*7)	--veps-inst			VE_INSTRCT
VEC-INST(K)	Number of vector instructions on VE nodes (*7)	--veps-inst			VEC_INST
VEC-ELMT(K)	Vector elements count on VE nodes (*7)	--veps-inst			VEC_ELMT
VEC-EXE(SECS)	Vector execution time on VE nodes[s] (*7)	--veps-inst			VEC_EXE

VEC-INST LOAD(K)	Vector load element count on VE nodes (*7)	--veps-inst			VEC_INST_LOAD
VE L1 CACHE MISS TIME(SECS)	L1 cache miss time on VE nodes[s] (*7)	--veps-time			VE_L1_CACHE_MISS_ TIME
VE CPU PORT CONFLICT(SECS)	Port conflict time on VE nodes[s] (*7)	--veps-time			VE_CPU_PORT_ CONFLICT
VEC ARITH EXEC TIME(SECS)	Vector arithmetic execution time on VE nodes[s] (*7)	--veps-time			VE_ARITH_EXEC_ TIME
VEC LOAD EXEC TIME(SECS)	Vector load execution time on VE nodes[s] (*7)	--veps-time			VEC_LOAD_EXEC_ TIME
VE POWER THROTTLING (SECS)	Power throttling time on VE nodes[s] (*7)	--veps-time			VE_PORT_ THROTTLING
VE THERMAL THROTTLING (SECS)	Thermal throttling on VE nodes[s] (*7)	--veps-time			VE_THERMAL_ THROTTLING
VE MAX NTASK	Maximum number of threads on VE nodes (*7)	--veps-task			VE_MAX_NTASK
VEC-OPERAND RATIO	The ratio of vector operations to all operations on VE nodes (*7)	--veps-opratio			VEC_OP_RATIO

*1 "REAL" displayed in the request account is the time of the request in the RUNNING state.

*2 "crname" is specified custom resource name.

*3 These items are available only for the environment whose execution host is SX-Aurora TSUBASA.

*4 "UNIT" is a unit for requested amount or usage of custom resources. It is displayed at the same time as displaying the requested amount and usage amount of the custom resource.

*5 REQVE and RSVVE may have different values when requesting VE with --venode specified when submitting a job. For example, if you request VE with --venode = 11 and --venum-lhost = 4, REQVE will be 11 and RSVVE will be 12. When requesting VE using -b and --venum-lhost, or when the default number of VEs in the queue is 1 (default) and VE is requested by specifying only --venode, REQVE and RSVVE values are the same.

*6 The value displayed with -V is the value collected periodically with the ps(VE) command. VE MEAN SIZE(K) displayed by -V is the average amount of VE memory used by all job processes associated with the request. The value displayed by --veps is the value calculated based on the value

of the process account output from VEOS. VE MEAN SIZE(K) displayed by --veps is the average amount of VE memory used by one process of all job processes associated with the request. For details on how to use the process accounting, see "3.6 VE Process Accounting".

*7 These items are values calculated based on the process account values output from VEOS. For details on how to use the process accounting, see "3.6 VE Process Accounting".

*8 KCORE MIN and V KCORE MIN are the sum of KCORE MIN and V KCORE MIN for each job in the request. KCORE MIN for each job is the integrated value of "Average memory consumption * CPU consumption time (system + user)" of the execution host. V KCORE MIN for each job is the integrated value of "Average virtual memory consumption * CPU consumption time (system + user)".

*9 If the node health check at the end of the job takes a long time, the END TIME of request account will be the end time including the time.

*10 REQUEST NAME is displayed with a width of the request name column in default. When --long-request-name option is specified, REQUEST NAME shows full request name.

3.1.1. Example of scacctreq

The scacctreq(1) command searches and displays the request accounting data collected by the Accounting server.

The scacctreq(1) command displays the results as follows when it is executed without options.

```
=============================================================================================================
REQUEST                REQUEST  USER      QUEUE    QUEUED   START    END          CPU     REAL
ID                     NAME     NAME      NAME     TIME     TIME     TIME       (SECS)  (SECS) STATUS
=============================================================================================================
0.bsv01                mpi_al*  user01    bq1      11:34:49 11:34:56 11:40:07  3998.27     301 DONE
1.bsv01                mpi_al*  user01    bq1      11:35:01 11:35:06 11:40:16  3563.88     301 DONE
2.bsv01                b_eff_*  user2     bq1      11:53:11 11:53:21 11:56:56  1195.70     204 DELETED
14.bsv01               jm.req   user03    bq1      11:55:35 11:56:56 11:57:12    45.96      13 DELETED
35.bsv01               run128*  user0004  bq1      11:56:11 11:57:00 11:58:08  2038.12      60 DONE
36.bsv01               run128*  user0004  bq1      11:56:12 11:57:56 11:59:05  2269.37      60 RERUN
```

When this command is executed without options, the request information is displayed based on the user's privilege from the request database stored on the Accounting server. The privilege can be specified by -P option. The privileges and their reference ranges are as follows.

Privilege	Scope of reference
General User privilege	Requests which are submitted by the user.
Special User Privilege	
Group Manager Privilege	Requests whose group is the user's managing group.
Operator Privilege	All requests.
Manager Privilege	

If the amount of stored data is large, it takes a long time to display the results. In such case, please

designate some options and reduce the indicated amount of data.

When this command is executed with -u option as follows, the displayed data can be narrowed by the specified user.

```
$ scacctreq -u user01
=============================================================================================================
REQUEST                 REQUEST  USER      QUEUE    QUEUED   START    END          CPU     REAL
ID                      NAME     NAME      NAME     TIME     TIME     TIME       (SECS)  (SECS) STATUS
=============================================================================================================
0.bsv01                 mpi_al*  user01    bq1      11:34:49 11:34:56 11:40:07  3998.27     301 DONE
1.bsv01                 mpi_al*  user01    bq1      11:35:01 11:35:06 11:40:16  3563.88     301 DONE
3021.bsv01              mpi_al*  user01    bq1      13:32:00 13:45:16 13:50:26  1110.38     300 DELETED
3022.bsv01              mpi_al*  user01    bq1      13:32:02 13:50:15 13:55:35  1117.42     300 DELETED
3023.bsv01              mpi_al*  user01    bq1      13:32:03 14:17:25 14:22:28  1118.33     300 DELETED
3024.bsv01              mpi_al*  user01    bq1      13:57:03 14:22:25 14:27:29  1111.81     300 DELETED
```

It is also possible to specify the range such as date and time by using the -s or -e options and so on.

Accounting server stores various other items for request data than displayed in the execution example above. The following is an example of displaying group IDs (-Y option) and the queue type (-K option).

```
$ scacctreq -YK
=========================================================================================================================
REQUEST                 REQUEST  USER      GROUP    QUEUE    QUEUED   START    END          CPU     REAL QUE
ID                      NAME     NAME      NAME     NAME     TIME     TIME     TIME       (SECS)  (SECS) TYPE STATUS
=========================================================================================================================
0.bsv01                 mpi_al*  user01    grp1     bq1      11:34:49 11:34:56 11:40:07  3998.27     301 BAT  DONE
1.bsv01                 mpi_al*  user01    grp1     bq1      11:35:01 11:35:06 11:40:16  3563.88     301 BAT  DONE
2.bsv01                 b_eff_*  user2     gr2      iq1      11:53:11 11:53:21 11:56:56  1195.70     204 INT  DONE
14.bsv01                jm.req   user03    gr3      bq1      11:55:35 11:56:56 11:57:12    45.96      13 BAT  DELETED
35.bsv01                run128*  user0004  gr4      bq1      11:56:11 11:57:00 11:58:08  2038.12      60 BAT  DELETED
36.bsv01                run128*  user0004  gr4      bq1      11:56:12 11:57:56 11:59:05  2269.37      60 BAT  DELETED
```

Refer to the scacctreq(1) online manual for details.

And the request accounting data on the Accounting server can be displayed by racctreq(1) command from a host on which NQSV/Client is installed. Please refer to racctreq(1) manual page for details.

3.2. Referencing Job Accounting Data

You can refer to the job accounting data by the scacctjob(1) command. Each item of the job accounting data is as the following table.

Name	Description	Option			Sort ID
		Display	Non	Search	

			Display		
JOB ID	Job ID				JOB_ID
REQUEST ID	Request ID			-I	REQUEST_ID
REQUEST NAME	Request Name	--long-request-name (*11)		-n	NAME
USER NAME	Submit user name			-u	USER_NAME
GROUP NAME	Group name	-Y		-g	GROUP_NAME
ACCOUNT CODE	Account code	-X		-x	ACCOUNT_CODE
HOST NAME	Execution host name			-l	HOST_NAME
QUEUE NAME	Submit queue name			-Q	QUEUE_NAME
QUEUED TIME	Submit time		-2	-{s|e}	QUEUED_TIME
	Submit date and time	-2			
START TIME	Start time		-2	-{S|b}	START_TIME
	Start date and time	-2			
END TIME	End time		-2	-{E|A}	END_TIME(*10)
	End date and time	-2			
CPU (SECS)	CPU consumption time (system + user) (sec.)		-t		CPU
CPU (SECS) SYS	CPU consumption time (system) (sec.)	-t			CPU_SYS
CPU (SECS) USER	CPU consumption time (user) (sec.)	-t			CPU_USER
REAL (SECS)	Real time of job (sec.) (*1)				REAL
REQUEST PRTY	Priority of the request	-p			REQUEST_PRTY
NICE	Nice value	-p			NICE
REQELAPS TIME(S)	Elapse Time Limit Value (sec.)	-T			REQELAPS_TIME
REQCPU TIME(S)	CPU Time Limit Value (sec.)	-q			REQCPU_TIME
REQCPU NUM	Number of CPU Limit Value	-q			REQCPU_NUM
REQMEM SIZE(K)	Memory Size Limit Value (KB)	-q			REQMEM_SIZE
REQGPU_NUM	Number of requested GPU	-G			REQGPU_NUM
EXIT STAT	Exit status	-h			EXIT_STAT

KCORE MIN	Total memory consumption (KB * MIN) (*8)	-k			KCORE_MIN
MEAN SIZE(K)	Average memory consumption (KB)	-m			MEAN_SIZE
MAXMEM SIZE(K)	Max. memory consumption (KB) (*9)	-M			MAXMEM_SIZE
V KCORE MIN	Total virtual memory consumption (KB * MIN) (*8)	--vmem			V_KCORE_MIN
V MEAN SIZE(K)	Average virtual memory consumption (KB)	--vmem			V_MEAN_SIZE
V MAXMEM SIZE(K)	Max. virtual memory consumption (KB)	--vmem			V_MAXMEM_SIZE
QUE TYPE	Queue type	-K			QUE_TYPE
WAIT TIME(SEC)	Wait time (the time from scheduling start time until actual start time)	-w			WAIT_TIME
REQVE NUM	Number of requested VE nodes for the Job (*2)	-V			REQVE_NUM
RSVVE NUM	Number of reserved VE nodes for the Job (*2)	-V			RSVVE_NUM
VE CPU(S)	CPU consumption time on VE nodes [SEC] (*2)	-V --veps (*6)			When displayed with -V :VE_CPU when displayed with --veps : VE_CPU_PS
VE KCORE MIN	Total memory consumption on VE nodes [KB * MIN] (*2)	-V --veps (*6)			When displayed with -V :VE_KCORE_MIN when displayed with --veps : VE_KCORE_MIN_PS
VE MEAN SIZE(K)	Average memory consumption on VE nodes [KB] (*2)	-V --veps (*6)			When displayed with -V :VE_MEAN_SIZE when displayed with --veps : VE_MEAN_SIZE_PS
VE MAXMEM	Max. memory	-V			VE_MAXMEM_SIZE

SIZE(K)	consumption on VE nodes [KB] (*2)				
VE REQ NODELIST	List of assigned VE nodes for the Job (*2)	-Z			None
VE USE NODELIST	List of used VE nodes for the Job (*2)	-Z			None
VE RSV NODELIST	List of reserved VE nodes for the Job (*2)	-Z			None
RESERVATION ID	Resource Reservation Section ID	-r			RES_ID
crname (*3)	Requested amount of each custom resource	-F			None
	Requested amount of the specified custom resource	-B			None
crname TOTAL	Total usage of each custom resource	-O			None
	Total usage of the specified custom resource	-j			None
crname MEAN	Average usage of each custom resource	-O			None
	Average usage of the specified custom resource	-j			None
crname MAX	Maximum usage of each custom resource	-O			None
	Maximum usage of the specified custom resource	-j			None
UNIT(*4)	Unit of custom resource	-F -B -O -j			None
HW FAILIRE(*5)	Hardware trouble flag	--hw-failure			HW FAILURE
VE TIME SLICE	Timeslice value on VE nodes[us] (*7)	--veps-perform			VE_TIME_SLICE
VE MFLOPS	MFLOPS value on VE nodes (*7)	--veps-perform			VE_MFLOPS
VE MFLOPS	MFLOPS(REAL) value on	--veps-perfor			VE_MFLOPS_REAL

(REAL)	VE nodes (*7)	m			
VE FPEC(K)	Floating point data element count on VE nodes (*7)	--veps-prog			VE_FPEC
VE SYSTEM CALL	Number of system calls on VE nodes (*7)	--veps-prog			VE_SYSTEM_CALL
VE-VH DATA TRANS	Data transfer amount between VE-VH (*7)	--veps-prog			VE_VH_DATA_TRANS
NUMBER OF VE PROCESS	Number of VE processes (*7)	--veps-prog			VE_PROC_NUM
INSTRCT(K)	Number of execution instructions on VE nodes (*7)	--veps-inst			VE_INSTRCT
VEC-INST(K)	Number of vector instructions on VE nodes (*7)	--veps-inst			VEC_INST
VEC-ELMT(K)	Vector elements count on VE nodes (*7)	--veps-inst			VEC_ELMT
VEC-EXE(SECS)	Vector execution time on VE nodes[s] (*7)	--veps-inst			VEC_EXE
VEC-INST LOAD(K)	Vector load element count on VE nodes (*7)	--veps-inst			VEC_INST_LOAD
VE L1 CACHE MISS TIME(SECS)	L1 cache miss time on VE nodes[s] (*7)	--veps-time			VE_L1_CACHE_MISS_ TIME
VE CPU PORT CONFLICT(SECS)	Port conflict time on VE nodes[s] (*7)	--veps-time			VE_CPU_PORT_ CONFLICT
VEC ARITH EXEC TIME(SECS)	Vector arithmetic execution time on VE nodes[s] (*7)	--veps-time			VE_ARITH_EXEC_ TIME
VEC LOAD EXEC TIME(SECS)	Vector load execution time on VE nodes[s] (*7)	--veps-time			VEC_LOAD_EXEC_ TIME
VE POWER THROTTLING (SECS)	Power throttling time on VE nodes[s] (*7)	--veps-time			VE_PORT_ THROTTLING
VE THERMAL THROTTLING	Thermal throttling on VE nodes[s] (*7)	--veps-time			VE_THERMAL_ THROTTLING

(SECS)					
VE MAX NTASK	Maximum number of threads on VE nodes (*7)	--veps-task			VE_MAX_NTASK
VEC-OPERAND RATIO	The ratio of vector operations to all operations on VE nodes (*7)	--veps-oprati o			VEC_OP_RATIO
NUMA NODE	The number of NUMA node on VE nodes(*7)	--veps-numa node			None

*1 "REAL" displayed in the job account is the time from the start to the end of the job process on the execution host. This time may differ from the REAL displayed in scacctreq(1).

For multi-node MPI requests, the master job starts after all slave jobs have started when the request is in "PRE-RUNNIG" state. The request is in the "RUNNING" state, when the master job starts. The end of the “RUNNING” state is when the master job ends. Therefore, the section where the request is in the "RUNNING" state and the section where the job process exists may be different. (scacctreq(1) "REAL" and scacctjob(1) "REAL")

Also, the REAL of a job does not include the time taken for the node health check of the job. The node health check at the start of a multi-node job is executed in the order of a slave job -> master job, and the node health check at the end of a job is executed in the order of a master job -> slave job. Therefore, the health check time of a master job is included between the start time and the end time of a slave job. The health check time of the master job is included between the start time and the end time of the slave job. However, the REAL of the job does not include the health check time of the master job.

Translated with www.DeepL.com/Translator (free version)

*2 These items are available only for the environment whose execution host is SX-Aurora TSUBASA.

*3 Specified custom resource name

*4 "UNIT" is a unit for requested amount or usage of custom resources. It is displayed at the same time as displaying the requested amount and usage amount of the custom resource.

*5 JOB ID of the job whose Hardware trouble flag is 1 is displayed as "-0001", and END TIME is displayed as "????/??/?? ??:??:??".

*6 The value displayed with -V is the value collected periodically with the ps(VE) command. VE MEAN SIZE(K) displayed by -V is the average amount of VE memory used by all processes started by the job.The value displayed by --veps is the value calculated based on the value of the process account output from VEOS. VE MEAN SIZE(K) displayed by --veps

is the average amount of VE memory used by one process of the processes started by the job. For details on how to use the process accounting, see "3.6 VE Process Accounting".

*7 These items are values calculated based on the process account values output from VEOS. For details on how to use the process accounting, see "3.6 VE Process Accounting".

*8 KCORE MIN is the integrated value of "Average memory consumption * CPU consumption time (system + user)" of the execution host. V KCORE MIN is the integrated value of "Average virtual memory consumption * CPU consumption time (system + user)". All CPU consumption times are converted to minutes.

*9 If memory is shared between each process of the job, the maximum amount of memory used by those processes may exceed the amount of memory installed on the execution host. In that case, this value may also exceed the installed memory of the execution host.

*10 Even if the node health check at the end of the job takes a long time, the END TIME of the job account is the end time without including the time.

*11 REQUEST NAME is displayed with a width of the request name column in default. When --long-request-name option is specified, REQUEST NAME shows full request name.

3.2.1. Example of scacctjob

The scacctjob(1) command searches and displays the job accounting data collected by Accounting server. The job accounting database may have a large amount of job accounting information, so it is recommended to narrow the display of job accounting data by specifying search options.

The scacctjob(1) command displays the results as follows when it is executed without options.

```
$ scacctjob
==============================================================================================================================
JOB                                REQUEST  USER                          QUEUE    QUEUED   START    END          CPU     REAL
ID     REQUEST_ID                  NAME     NAME      HOST-NAME           NAME     TIME     TIME     TIME       (SECS)   (SECS)
==============================================================================================================================
03001  0.bsv01(0000)               mpi_al*  user01    node_0000a.domain1*  bq1      11:34:49 11:34:58 11:40:01  1104.36  300.80
03001  0.bsv01(0065)               mpi_al*  user01    node_0065a.domain1*  bq1      11:34:49 11:34:56 11:40:02  1187.32  304.15
03001  0.bsv01(0092)               mpi_al*  user01    node_0092a.domain1*  bq1      11:34:49 11:34:56 11:40:02  1187.08  304.06
03001  0.bsv01(0077)               mpi_al*  user01    node_0077a.domain1*  bq1      11:34:49 11:34:56 11:40:02  1187.13  304.20
03001  0.bsv01(0078)               mpi_al*  user01    node_0078a.domain1*  bq1      11:34:49 11:34:56 11:40:02  1187.50  304.20
03001  0.bsv01(0095)               mpi_al*  user01    node_0095a.domain1*  bq1      11:34:49 11:34:56 11:40:02  1187.12  304.10
```

When this command is executed without options, the job information is displayed based on the user's privilege from the job accounting database stored on the Accounting server. The privilege can be specified by -P option. The privileges and their reference ranges are as follows.

Privilege	Scope of reference
General User privilege	Jobs which are executed by the user.
Special User Privilege	
Group Manager Privilege	Jobs whose group is the user's managing group.
Operator Privilege	All jobs.

Manager Privilege	

If the amount of stored data is large, it may take a long time to display the results. In such case, please designate some options to reduce the number of displayed data.

When this command is executed with -u and -l options as follows, the data narrowed by users and execution host names is displayed.

```
$ scacctjob -u user01 -l node_0000a
==========================================================================================================================
JOB                                 REQUEST  USER                         QUEUE    QUEUED   START    END         CPU     REAL
ID    REQUEST-ID                    NAME     NAME     HOST-NAME           NAME     TIME     TIME     TIME      (SECS)   (SECS)
==========================================================================================================================
03001 0.bsv01(0000)                 mpi_al*  user01   node_0000a.domain1* bq1      11:34:49 11:34:58 11:40:01  1104.36  300.80
03227 6723.bsv01(0065)              mpi_al*  user01   node_0000a.domain1* bq1      08:26:13 09:28:26 09:33:31   922.54  302.54
03001 6753.bsv01(0000)              mpi_al*  user01   node_0000a.domain1* bq1      09:34:51 11:37:37 11:42:40  1195.17  300.44
03002 6763.bsv01(0000)              mpi_al*  user01   node_0000a.domain1* bq1      10:02:17 11:42:47 11:47:50  1189.49  300.83
03004 6764.bsv01(0108)              mpi_al*  user01   node_0000a.domain1* bq1      10:02:17 11:48:05 11:53:10  1186.45  303.26
03057 10675.bsv01(0000)             mpireq*  user01   node_0000a.domain1* bq1      11:19:31 11:19:52 11:21:54   478.30  119.88
```

It is also possible to specify the range of date and time by using such as the -s or -e options and so on.

NQSV Accounting & budget control function stores various items of the data other than the items displayed in the execution example above. The following is an example of displaying group IDs (-Y option) and the queue type (-K option).

```
$ scacctjob -YK
==================================================================================================================================
JOB                                REQUEST  USER     GROUP                        QUEUE  QUEUED   START    END          CPU    REAL QUE
ID    REQUEST-ID                   NAME     NAME     NAME     HOST-NAME           NAME   TIME     TIME     TIME       (SECS)  (SECS) TYPE
==================================================================================================================================
03001 0.bsv01(0000)                mpi_al*  user01   grp1     node_0000a.domain1* bq1    11:34:49 11:34:58 11:40:01  1104.36  300.80 BAT
03001 0.bsv01(0065)                mpi_al*  user01   grp1     node_0065a.domain1* bq1    11:34:49 11:34:56 11:40:02  1187.32  304.15 BAT
03001 0.bsv01(0092)                mpi_al*  user01   grp1     node_0092a.domain1* bq1    11:34:49 11:34:56 11:40:02  1187.08  304.06 BAT
03001 0.bsv01(0077)                mpi_al*  user01   grp1     node_0077a.domain1* bq1    11:34:49 11:34:56 11:40:02  1187.13  304.20 BAT
03001 0.bsv01(0078)                mpi_al*  user01   grp1     node_0078a.domain1* bq1    11:34:49 11:34:56 11:40:02  1187.50  304.20 BAT
03001 0.bsv01(0095)                mpi_al*  user01   grp1     node_0095a.domain1* bq1    11:34:49 11:34:56 11:40:02  1187.12  304.10 BAT
```

Refer to the scacctjob(1) online manual for details.

And the job accounting data on the Accounting server can be displayed by racctjob(1) command from a host on which NQSV/Client is installed. Please refer to racctjob(1) manual page for details.

3.3. Referencing Reservation Accounting Data

You can refer to the reservation accounting data by the scacctrsv(1) command. Each item of the reservation accounting data is as the following table.

Name	Description	Option			Sort ID
		Display	Non	Search	

			Display		
RESERVATION ID	Resource Reservation Section ID			-I	RES_ID
BATCH_SERVER HOST	Host name of the BSV	-B		-h	BSV
USER NAME	User name who creates the reservation	-U		-u	USER_NAME
GROUP NAME	Group name	-Y		-g	GROUP_NAME
ACCOUNT CODE	Account code	-X		-x	ACCOUNT_CODE
CREATE TIME	Creation date and time of the reservation	-C		-{s|e}	CREATE_TIME
RESERVATION TIME START	Scheduled Start date and time			-{S|b}	RES_START_TIME
RESERVATION TIME END	Scheduled End date and time			-{A|E}	RES_END_TIME
ACTUAL RESERVED TIME START	Actual Start date and time	-F		-f {S|b}	START_TIME
ACTUAL RESERVED TIME END	Actual End date and time	-F		-f {A|E}	END_TIME
EHOST NUM	Requested number of hosts				EHOST_NUM
CPU NUM	Requested number of CPUs per host				CPU_NUM
QUEUE	Queue name			-Q	QUEUE
STATUS	Exit status				STATUS
TEMPLATE NAME	Template name	-L		-l	TEMPLATE_NAME

3.3.1. Example of scacctrsv

The scacctrsv(1) command searches and displays the reservation accounting data collected by Accounting server.

The scacctrsv(1) command displays the results as follows when it is executed without options.

```
$ scacctrsv
================================================================================
   RES RESERVATION TIME                          NODE      CPU
    ID START               END                    NUM      NUM QUEUE    STATUS
================================================================================
  5237 2016/03/10 16:20:00 2016/03/10 16:21:40       2        1 bq1      DONE
  5238 2016/03/10 16:20:00 2016/03/10 16:21:40       2        1 bq1      DONE
  5236 2016/03/10 18:00:00 2016/03/10 19:00:00       1        1 bq2      DONE
  5242 2016/03/12 16:02:00 2016/03/12 16:02:01       2        1 bq1      DONE
```

```
  5244 2016/03/14 11:53:00 2016/03/14 11:54:40         1         0 bq2       DONE
  5245 2016/03/14 11:53:00 2016/03/14 11:54:40         1         1 bq2       DONE
```

When this command is executed without options, the reservation information is displayed based on the user's privilege from the reservation accounting database stored on the Accounting server. The privilege can be specified by -P option. The privileges and their reference ranges are as follows.

Privilege	Scope of reference
General User privilege	Reservations which are created by the user.
Special User Privilege	
Group Manager Privilege	Reservations whose group is the user's managing group.
Operator Privilege	All reservations.
Manager Privilege	

If the amount of stored data is large, it may take a long time to display the results. In such case, please designate some options to reduce the number of displayed data.

When this command is executed with -Q option as follows, the data narrowed by the queue name is displayed.

```
$ scacctrsv -Q bq1
================================================================================
   RES RESERVATION TIME                              NODE      CPU
    ID START               END                       NUM      NUM QUEUE    STATUS
================================================================================
  5237 2016/03/10 16:20:00 2016/03/10 16:21:40         2         1 bq1       DONE
  5238 2016/03/10 16:20:00 2016/03/10 16:21:40         2         1 bq1       DONE
  5240 2016/03/11 16:00:00 2016/03/11 17:00:00         2         1 bq1       DONE
  5241 2016/03/11 17:00:00 2016/03/11 18:00:00         2         1 bq1       DONE
  5239 2016/03/12 15:00:00 2016/03/12 16:00:00         2         1 bq1       DONE
  5242 2016/03/12 16:02:00 2016/03/12 16:02:01         2         1 bq1       DONE
```

It is also possible to specify the range of date and time by using such as the -s or -e options and so on.

Refer to the scacctrsv(1) online manual for details.

And the reservation accounting data on the Accounting server can be displayed by racctrsv(1) command from a host on which NQSV/Client is installed. Please refer to racctrsv(1) manual page in NEC Network Queuing System V (NQSV) User's Guide [Reference] for details.

3.4. Referencing Daily Report and Monthly Report

You can refer to the daily report and the monthly report by the scsumacct(1M) command. The

scsumacct(1M) command collects and displays the data ended at the specified date or the specified month from the request accounting data and the reservation accounting data collected by the Accounting server. Only super-user can execute scsumacct(1M).

Displayed items of the scsumacct(1M) are as follows.

Item	Description	Option		Display	
		Display	Non Display	Reservation	Request
USER	User Name			o	o
GROUP	Group Name			o	o
ACCTCODE	Account-code			o	o
REQYESTS	Total number of requests			-	o
JOBS	Total number of jobs			-	o
EHOST NUM	Total number of execution host			o	o
ELAPSE(H)	Total time of job's elapsed time[hours			-	o
CPU(SECS) SYS	System CPU Time [seconds]	-t		-	o
CPU(SECS) USER	User CPU Time [seconds]	-t		-	o
CPU(SECS)	CPU Time (System+User) [seconds]		-t	-	o
MEM	Total memory usage [KB x min.]			-	o
RESERVES	Total number of reservations			o	-
RSVTIME	Total time of reservation sections [hours]			o	-
USETIME	Total used time of reservation sections [hours]			o	-
ESTIMATE	Total estimated fee.	-b		o	o
FEE	Total real fee	-b		o	o

3.4.1. Daily Report

scsumacct(1M) command collects and displays aggregate data of the request accounting and reservation accounting for each user, group and account-code.

When executing scsumacct(1M) without any options, a today's daily report displayed as follows. Please refer to scsumacct(1M) for details of the displayed properties.

```
# scsumacct
DAILY REPORT [2017/12/04]
```

```
[REQUEST]
================================================================================
USER                      REQUESTS     JOBS   EHOST NUM ELAPSE[H]  CPU[SEC]       MEM
--------------------------------------------------------------------------------
user1                            1        2           2      0.00      0.00      0.00
user2                            4        7           7      0.05    128.73  14287.35
user3                           22       32          32      2.89      0.80     66.39
================================================================================
NUM OF USER    : 3              27       41          41      2.94    129.53  14353.74

================================================================================
GROUP                     REQUESTS     JOBS   EHOST NUM ELAPSE[H]  CPU[SEC]       MEM
--------------------------------------------------------------------------------
grp1                            23       34          34      2.89      0.80     66.39
grp2                             4        7           7      0.05    128.73  14287.35
================================================================================
NUM OF GROUP   : 2              27       41          41      2.94    129.53  14353.74

================================================================================
ACCTCODE                  REQUESTS     JOBS   EHOST NUM ELAPSE[H]  CPU[SEC]       MEM
--------------------------------------------------------------------------------
actc01                         152      640         640      0.68      0.92     84.20
================================================================================
NUM OF ACCTCODE: 2              27       41          41      2.94    129.53  14353.74

[RESERVE]
================================================================
USER                      RESERVES   EHOST NUM   RSVTIME   USETIME
----------------------------------------------------------------
                                 1           1      0.03      0.00
================================================================
NUM OF USER    : 1               1           1      0.03      0.00

================================================================
GROUP                     RESERVES   EHOST NUM   RSVTIME   USETIME
----------------------------------------------------------------
grp1                             1           1      0.03      0.00
================================================================
NUM OF GROUP   : 1               1           1      0.03      0.00

================================================================
ACCTCODE                  RESERVES   EHOST NUM   RSVTIME   USETIME
----------------------------------------------------------------
                                 1           1      0.03      0.00
================================================================
NUM OF ACCTCODE: 1               1           1      0.03      0.00
```

To display a daily report other than the current day, execute scsumacct(1M) with -s option as follows.

```
# scsumacct -s MM/DD[YY]
```

MM: month, *DD*: date, *YY*: year

Additionally, the -t option can be used to display user and system CPU time separately, and the -b option can be used to display both estimated fee and actual fee.

Refer to the scsumacct(1M) manual page for details.

3.4.2. Monthly Report

To display a monthly report, execute scsumacct(1M) with -S option. scsumacct(1M) command collects and displays aggregate data of the request accounting and reservation accounting for each user, group and account-code.

```
# scsumacct -S MM[YY]
```

MM: month, *DD*: date, *YY*: year

Refer to the scsumacct(1M) manual page for details.

3.5. Saving Accounting Data

The accounting data is continuously accumulated by NQSV Accounting & Budget Control as long as operation continues. If the amount of stored data becomes too large, it may take a long time to display the data. Therefore, it is necessary to delete (migrate) records except for the data required for operation.

3.5.1. Accounting server

The sceditacct(1M) command is used to deletes (migrates) accounting data collected by the Accounting server to optimize the current database. The command execution format is as follows.

```
# /opt/nec/nqsv/bin/sceditacct -j
      [-s MM/DD[/YY]] [-e MM/DD[/YY]] [-S MM/DD[/YY]] [-E MM/DD[/YY]]
      [-u user | -g group | -x acctcode] [-h host]
      [-d database,...] [-m database | -c database]

# /opt/nec/nqsv/bin/sceditacct -r
      [-s MM/DD[/YY]] [-e MM/DD[/YY]] [-S MM/DD[/YY]] [-E MM/DD[/YY]]
      [-q MM/DD[/YY]] [-Q MM/DD[/YY]]
      [-u user | -g group | -x acctcode]
      [-d database,...] [-m database | -c database]
      [-f]

# /opt/nec/nqsv/bin/sceditacct -y
      [-s MM/DD[/YY]] [-e MM/DD[/YY]] [-S MM/DD[/YY]] [-E MM/DD[/YY]]
      [-u user | -g group | -x acctcode]
      [-d database,...] [-m database | -c database]
```

```
# /opt/nec/nqsv/bin/sceditacct -N
```

MM: month, *DD*: date, *YY*: year

The -j option sets the job accounting data as the operation target. The -r option sets the request accounting data as the operation target. And the -y option sets the reservation accounting data as the operation target. Other options set the conditions to delete (migrate).

The following table lists the conditions to delete (migrate).

Option Name	Specification	Data to Be Deleted(Migrated)
u	User ID(or user name)	Data of the specified user ID (user name).
g	Group ID(or group name)	Data of the specified group ID (group name).
x	Account Code	Data of the specified account code.
s	[*MM*/*DD*[/*YY*]]	Data of which execution is terminated on the specified date or later.
e	[*MM*/*DD*[/*YY*]]	Data of which execution is started on the specified date or earlier.
S	[*MM*/*DD*[/*YY*]]	Data of which execution is started on the specified date or later.
E	[*MM*/*DD*[/*YY*]]	Data of which execution is terminated on the specified date or earlier.
h	Host Name(or IP Address)	Data of the specified host name (IP address). This option is effective only to job accounting database.

The start time and the end time of the request is registered as 0 in its database before execution is terminated. The sceditacct command with the above options does not operate the request accounting data that are not ended.

The sceditacct command with the above options and -f option operate the request accounting data including the request that is not ended. -f option is effective only to request accounting database.

The -f option is a mode to forcibly delete the data in the account database, and it may delete the data in the account database improperly depending on the usage. Do not use it normally.

The sceditacct with the following options can delete the request accounting data deleted by qdel before running without the request accounting data which has not ended.

Option Name	Specification	Data to Be Deleted(Migrated)
q	[*MM*/*DD*[/*YY*]]	Delete or save records of requests that queued at or after the specified date and deleted by qdel before running.
Q	[*MM*/*DD*[/*YY*]]	Delete or save records of requests that queued at or before the specified date and deleted by qdel before running.

These options can be specified with u, g, x, s, e, S and E option. When specifying q and Q option with s, e, S and E option, the data which met the condition of q and Q option or the condition of s, e, S and E option will be the operation target.

For example, if user1 is removed from management, the data related to user1 can be deleted by specifying as follows.

```
# /opt/nec/nqsv/bin/sceditacct -r -u user1 (for request accounting)
# /opt/nec/nqsv/bin/sceditacct -j -u user1 (for job accounting)
# /opt/nec/nqsv/bin/sceditacct -y -u user1 (for reservation accounting)
```

The -m option can be used to migrate the data to another file. For example, execute sceditacct(1M) command as follows in order to migrate the data of the previous month and leave the data of the current month in monthly processing. In the following example, the data for March 31 and earlier is migrated on April 1.

- For request/job accounting data, specify the destination directory to -m option.

```
# /opt/nec/nqsv/bin/sceditacct -r -E 3/31 -m /tmp/racct2019_03
# /opt/nec/nqsv/bin/sceditacct -j -E 3/31 -m /tmp/jacct2019_03
```

For request：racct, racct_tally, racct_cr, racct_cr_tally files are created in /tmp/racct2019_03. To back up the request accounting data deleted by qdel before running, add the -Q option.

For job：jacct, jacct_cr files are created in /tmp/jacct2019_03.

- For reservation accounting data, specify the destination database file to -m option.

```
# /opt/nec/nqsv/bin/sceditacct -y -E 3/31 -m /tmp/rsvacct2019_03
```

It is possible to automatically execute monthly processing by registering the sceditacct(1M) command execution to cron.
Execute scacctreq(1), scacctjob(1) or scacctrsv(1) to refer to these backup files.

- For request accounting data, specify the backup directory to -d option.

```
# /opt/nec/nqsv/bin/scacctreq -d /tmp/racct2014_03
```

- For job accounting data, specify the backup directory to -d option.

```
# /opt/nec/nqsv/bin/scacctjob -d /tmp/jacct2019_03
```

- For reservation accounting data, specify the backup database file to -d option.

```
# /opt/nec/nqsv/bin/scacctrsv -d /tmp/rsvacct2019_03
```

* Refer to the sceditacct(1M) manual page for details.

3.5.2. Accounting Monitor

The request accounting files, the job accounting files and the reservation accounting files on the Accounting monitor host are continuously added to even after transferred to the Accounting server. The file size will therefore continue to increase without limit. NQSV Accounting & Budget Control supports the scturnacct(1M) command to migrate these three kinds of accounting files and to manage generations. The accounting files are migrated only if their file sizes exceed the predefined size when the migration command, scturnacct(1M) is executed. The number of generations to be managed can also be specified on the Accounting monitor's configuration file (/etc/opt/nec/nqsv/acmd.conf).

The following table shows items to be set in the configuration file.

Item	Description	Default Value
TURN_SIZE	File size that scturnacct(1M) command starts to migrate accounting files. (unit:MB)	500(MB)
TURN_SAVE	Number of generations to migrate.	10

The following shows how to execute the scturnacct(1M) command.

```
# /opt/nec/nqsv/bin/scturnacct
```

The file size is compared with the TURN_SIZE value in the configuration file. If it is larger than TURN_SIZE, /var/opt/nec/nqsv/bsv/account/reqacct[.n] files are renamed as follows.

```
reqacct.n-1   ->      reqacct.n
              :
reqacct.1     ->      reqacct.2
reqacct       ->      reqacct.1
```

When the accounting file path is changed from the default value, the changed path is used for the migration by the scturnacct(1M) command.

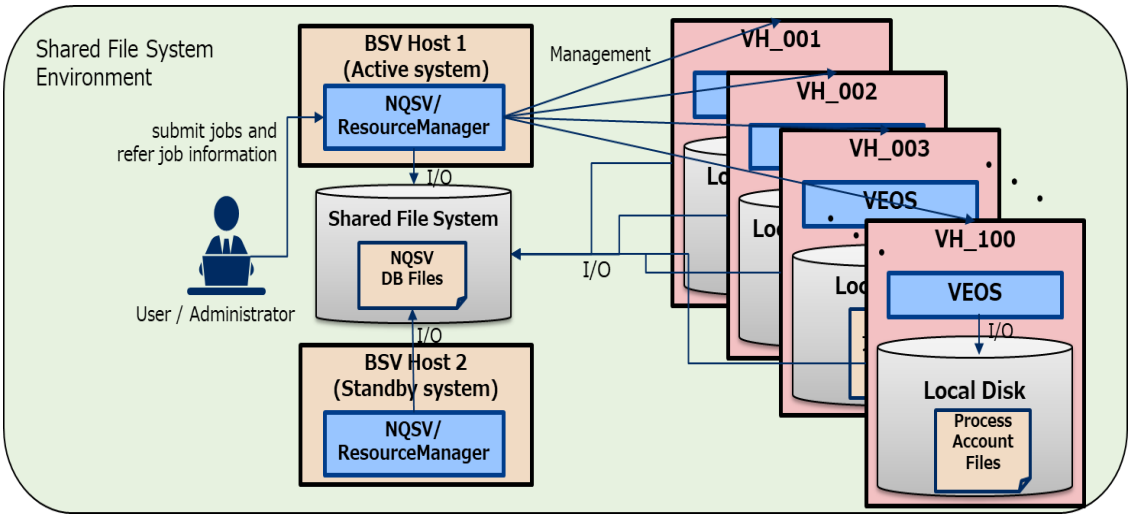
*Refer to the scturnacct(1M) manual page for details

3.6. VE Process Accounting

Using the VEOS feature, the account of the process executed on the VE can be referred. NQSV summarizes the process account output by VEOS on a per-job and per-request basis. The scacctjob(1)/scacctreq(1) command can view those values. For details on how to set the VEOS process accounting, see "SX-Aurora TSUBASA Installation Guide 4.13 Configuration for Process accounting".

3.6.1. Available Operational Configurations

This feature works in the environment of the shared file system as shown below.



Configuration Notes:

If you are building an environment that uses Docker to run VE jobs, be sure to specify "--pid=host" as an option for the docker-run command when starting the Docker container. The aggregation of VE process account can be used in the Docker environment started with this setting. Note that this feature cannot be used in a PID virtualized container image.

The flow of using this feature is as follows.

1. Set the VEOS process accounting and output the process account file on the VH's local disk.
2. Set a script that reads a process account file and aggregates account data by job (Process Account Aggregation Script) in the POST-RUNNING state of user EXIT. Aggregate results are output to a temporary file in CSV format (Process Account Aggregation Results CSV File). Set the output destination to a directory on a shared file system that can be accessed by the accounting server.
3. On the batch server host, set the scacctpsm command to cron and run it periodically. By this process, the contents of the CSV file aggregated in 2. are registered in the job /request account database.
4. The scacctjob(1)/scacctreq(1) command refers to aggregated information per job/request.

The following sections describe it in detail.

3.6.2. List of Aggregation Items

The following items of the VE process account are aggregated in the job account. The item name is a member name of the VE process account structure (VE10/VE20:struct ve_acct_v15, VE30: struct ve_acct_v16).

Process account item name	Description	Job account item name
ac_utime	Total VE CPU consumption time (users) [ticks]	VE CPU(SECS)
ac_total_mem	VE total memory usage [KB*ticks]	VE KCORE MIN Unit: KB*minute
ac_timeslice	VE time slice value [μs]	VE TIME SLICE
ac_fpec	Number of elements to execute floating point data in VE	VE FPEC(K)
ac_syscall	Number of VE system calls	VE SYSTEM CALL
ac_transdata	Data transfer rate between VE and VH [KB]	VE-VH DATA TRANS
ac_ex	Number of VE instructions executed	INSTRCT(K)
ac_vx	Number of vector instruction executions in VE	VEC-INST(K)
ac_ve2	Number of vector elements in VE	VEC-ELMT(K)
ac_vecc	Vector instruction execution time for VE [μs]	VEC-EXE(SECS)
ac_vlec	Number of vector load elements in VE	VEC-INST LOAD(K)
ac_l1mcc	L1 cash miss time for VE [μs]	VE L1 CACHE MISS TIME(SECS)
ac_pccc	VE CPU port contention time [μs]	VE CPU PORT CONFLICT(SECS)

ac_varec	Vector calculation execution time of VE [μs]	VEC ARITH EXEC TIME(SECS)
ac_vldec	Vector load execution time for VE [μs]	VEC LOAD EXEC TIME(SECS)
ac_ptcc	HW downtime due to VE energy factor [μs]	VE POWER THROTTLING(SECS)
ac_ttcc	HW downtime due to temperature factors in VE [μs]	VE THERMAL THROTTLING(SECS)
ac_max_nthread	Maximum number of tasks in VE	VE MAX NTASK
ac_numanode	NUMA node number of VE	NUMA NODE
ac_fmaec	Number of combined product sum elements * Totaled for the calculation of other items.	None
ac_etime	Elapse time [ticks] * Totaled for the calculation of other items.	None

The following items are calculated based on the process account items and output to the job account.

Job account item name	Description
VE MEAN SIZE(K)	Average memory usage for VE [KB]
NUMBER OF VE PROCESS	Number of processes in VE
VE MFLOPS	MFLOPS value of VE
VE MFLOPS(REAL)	MFLOPS(REAL) value of VE
VEC-OPERAND RATIO	Vectorization ratio of VE

3.6.3. Process Account Aggregation Results CSV File

Process Account Aggregation Script reads a VE process account file. It aggregates the data for each job and outputs it to a CSV-formatted file. The naming conventions for this file are as follows:

- When a Process Account Aggregation Script is executed in the user EXIT.
  <*IP_ADDRESS*>_jacct_ps_job.csv
  IP_ADDRESS is the IP address of VH.

The format of the CSV file is as follows:

column number	item name	Description

1	SID	Session ID
2	START_TIME	Earliest start time of all processes
3	END_TIME	Latest finish time of all processes
4	VE CPU	Total CPU consumption time (user) of VE [ticks] * 1tick = 10ms
5	VE KCORE MIN	Total memory used by VE [KB*ticks]
6	VE MEAN SIZE	Average memory usage for VE [KB]
7	VE TIME SLICE	Time Slice Value[μs]
8	VE MFLOPS	MFLOPS value
9	VE MFLOPS(REAL)	MFLOPS(REAL) value
10	VE FPEC	Number of floating point data elements
11	VE SYSTEM CALL	Number of system calls
12	VE-VH DATA TRNS	Data transfer between VE and VH [KB]
13	VE NUM PROCS	Number of processes executed
14	VE INSTRCT	Number of instructions
15	VECTOR INST	Number of vector instructions executed
16	VECTOR ELMT	Number of vector elements
17	VEC-EXE	Vector instruction execution time [μs]
18	VE L1CacheMiss	L1 cache miss time [μs]
19	VE CPUPortConf.	CPU port contention time [μs]
20	VE ArithExec.	Vector operation execution time [μs]
21	V.LoadExec.	Vector load execution time [μs]
22	V.LoadExecElem	Number of elements of vector loading
23	VE PowerThrottling	HW downtime due to power factor [μs]
24	VE ThermalThrottling	HW downtime due to temperature factor [μs]
25	VE MAX NTASK	Maximum number of tasks
26	V. Op. Ratio	Vectorization ratio
27	VE NUMA NODE NO	NUMA node number
28	FMAE	Number of combined sum elements
29	Elapse Time	Elapse time [ticks]

3.6.4. The Setting of Process Account Aggregation Script

1. Configure the aggregation script on all VH.

The following script for aggregation is included in the package of the job server installed on VH.
Set the VH IP address (IP address for communication between the NQSV batch server and the job server) and the output directory for the process account aggregation result CSV file in this script.

```
# vi /opt/nec/nqsv/sbin/tally_veacct.sh
 :
# === [NEED TO CHANGE VALUES] ===
#
# Please set execution host's IP address which management on NQSV
HOST_IP_ADDR="ip_addr"

# Please set NFS directory where CSV files output.
# This directory need to shared between BSV and JSV.
OUTPUT_DIR="/var/opt/nec/nqsv"
 :
```

- Set the HOST_IP_ADDR variable to the IP address of the VH.
- The OUTPUT_DIR variable should be set to the output directory of the process account aggregate result CSV file. This directory must be a shared file system directory that can be referenced from both VH and the accounting server.

2. Configure logrotate on all VH.

When using VE process accounting function, there may be a deficit in the aggregation of process account information when logrotate is executed on the process account file by Linux cron during job execution. Therefore, it is necessary to stop the logrotate of the process account file by cron and change the setting so that the aggregation script executes the logrotate only when the job is finished.

First, copy the configuration file for rotate the process account file installed on VH as follows.

```
# cp /etc/logrotate.d/psacct-ve /var/opt/nec/nqsv/psacct-ve0
```

This configuration file is set to rotate once a day by default, and is saved for 31 times (one month). If you want to change the setting, please edit the file.

Ex.) If you want to save more than 31 times (more than 1 month)
Ex.) If you want to increase the rotation frequency because the size of the process account file output per day is too huge.

```
# vi /var/opt/nec/nqsv/psacct-ve0
```

This configuration file is based on the Linux logrotate configuration file, so please refer to the logrotate page for detailed settings.

Next, copy the edited psacct-ve0 file as follows.

```
# cp /var/opt/nec/nqsv/psacct-ve0 /var/opt/nec/nqsv/psacct-ve1
# cp /var/opt/nec/nqsv/psacct-ve0 /var/opt/nec/nqsv/psacct-ve2
# cp /var/opt/nec/nqsv/psacct-ve0 /var/opt/nec/nqsv/psacct-ve3
# cp /var/opt/nec/nqsv/psacct-ve0 /var/opt/nec/nqsv/psacct-ve4
# cp /var/opt/nec/nqsv/psacct-ve0 /var/opt/nec/nqsv/psacct-ve5
# cp /var/opt/nec/nqsv/psacct-ve0 /var/opt/nec/nqsv/psacct-ve6
# cp /var/opt/nec/nqsv/psacct-ve0 /var/opt/nec/nqsv/psacct-ve7
```

After the copy is complete, edit the "?" at the beginning of the line to match the number at the end of the file name.

```
# vi /var/opt/nec/nqsv/psacct-ve0
Before) /var/opt/nec/ve/account/pacct_? {
After ) /var/opt/nec/ve/account/pacct_0 {

# vi /var/opt/nec/nqsv/psacct-ve1
Before) /var/opt/nec/ve/account/pacct_? {
After ) /var/opt/nec/ve/account/pacct_1 {

 - For psacct-ve2 to 7, please change them in the same way as above.
```

With the above settings, the process account file of the VE used by the job will be rotated according to the configuration file at the end of the job.

Finally, comment out all lines in the following file. This will prevent the process account file from being rotated by cron.

```
# vi /etc/logrotate.d/psacct-ve
```

3. If you have already applied the process account settings up to R1.08 **(version 1.08-270)** on all VH Please remove the following symbolic link.

```
# rm /etc/opt/nec/ve/veos/psacct-postrotate.d/logrotate_tally_veacct.sh
```

4. Set the user exit script on BSV.

In order to aggregate the process account at the time of the end of the request, it is necessary to set

the script for aggregation in user EXIT. Use the qmgr(1M) command to set up the queue to aggregate the process account. The command line image is as follows

```
$ qmgr  -Pm
Mgr: set execution_queue userexit location=post-running script=(uex_tally_veacct.sh) queue=<queue>
```

By this setting, the Process Account Aggregation Results CSV File (<IP_ADDRESS>_jacct_ps_job.csv) will be output to the path set in the OUTPUT_DIR for each VH used.

5. Generate the INDEX information on BSV.

In order to speed up the process of registering process account information, please execute the following script to create INDEX information in the DB of the job account. This operation must be performed while the operation is stopped.

```
# systemctl stop nqs-asv
# systemctl stop nqs-acm
# /opt/nec/nqsv/sbin/create_index_jacct.sh
```

- In an environment where NQSV is duplicated by EXPRESSCLUSTER etc., the following two conditions must be satisfied as the output area of the Process Account Aggregation CSV Results File.
  - This area can be read and written from both the VH and the accounting server (active system and stand-by system).
  - This area should also be duplicated.
    Since the process account information may be lost due to a disk failure, etc., duplicate the output area of the Process Account Aggregation CSV Results File.

3.6.5. Process Account Aggregation Settings in Account Database

The scacctpsm(1M) command registers a Process Account Aggregation Results CSV File in the account database. The execution format of the command is as follows:

scacctpsm -i <*input_directory*>

input_directory specifies the output directory of the CSV file created by the Process Account

Aggregation Script. To execute this command, you must have the root privileges. By executing this command, the process account data in the CSV file will be aggregated for each job and registered in the job account database. If the request has already been completed, the data per job is aggregated for each request and registered in the request account database. The Process Account Aggregation Results CSV File will be registered in account database and then deleted.
In addition, by registering scacctpsm(1M) command in cron, you can periodically register the Process Account Aggregation Results CSV File that is aggregated at the timing when the request ends in the account database.

You can use scacctjob(1)/scacctreq(1) to refer to the aggregate results. See "3.1 Referencing Request Accounting Data" and "3.2 Referencing Job Accounting Data" for details on the item names and how to refer to the aggregate results.

- Do not execute the saveing process by the sceditacct command for a job request that has not completed the aggregation of VE process account information by the scacctpsm command.
If it is executed, the scacctpsm command will not be able to register the information in the CSV file in which the relevant job information is recorded, and the CSV file will remain in the same state.

- In an environment where NQSV is duplicated by EXPRESSCLUSTER etc., avoid performing process account aggregation on the same timing from both the active system and the stand-by system when periodically aggregating process accounts with cron. After tuning the aggregation time, set the aggregation timing of the active system and the stand-by system so that there is an interval longer than that time.
If one of the active system and the stand-by system is operating and the area accessed by NQSV is mounted on the other, there is no problem even if the above aggregation is performed at the same timing. Aggregation error will occur on those who are not operating, but it will not affect those who are operating.

3.6.6. Using the VE Process Accounting in Docker Environment

The following four settings are required for the docker-run command options when starting a Docker container.

(1) Set the container's PID namespace to be the same as the host OS. (Use the *--pid=host* option)

(2) Mount the directory where the process account file is output on the host OS in the container with the same filepath as the host OS.
(Use the *–v /var/opt/nec/ve/account:/var/opt/nec/ve/account* option)

(3) Mount the NFS directory where the CSV file is output in the container with the same filepath as the host OS, as set in the Process Account Aggregation Script. This filepath must be the same as the directory specified in the 3.6.4. The Setting of Process Account Aggregation Script setting of OUTPUT_DIR.
(Use the *–v <The directory set in OUTPUT_DIR>:<The directory set in OUTPUT_DIR>* option)

(4) Mount the directory where the host OS Process Account Aggregation Script is installed to the container with a **different** filepath than the host OS.
(Use the *–v /opt/nec/nqsv/sbin/:/tmp/tally_veacct* option)

Then, edit the user exit script that runs POST-RUNNING and add the following bold lines. If you specified a directory other than "/tmp/tally_veacct" for the mount target in (4) above, please change the following script as well.

```
# vi /opt/nec/nqsv/sbin/uex_prog/uex_tally_veacct.sh
 :
case $UEX_LOCATION in
POSTRUNNING)
    #For Docker
    if [ -e /tmp/tally_veacct/tally_veacct.sh ]; then
        ¥cp -f /tmp/tally_veacct/tally_veacct.sh /opt/nec/nqsv/sbin/tally_veacct.sh
    fi
    /opt/nec/nqsv/sbin/tally_veacct.sh
    ;;
esac
exit 0
 :
```

By setting up the above, VE process accounts can be aggregated at the timing of request exit even in the Docker environment.

Chapter 4. Budget Management

When the budget management function of NQSV Accounting & Budget Control package is enabled, please set the accounting rate (sbuedit(1M)) and the budget (budgetedit(1M)) on the Accounting server. Please refer to each command reference in NEC Network Queuing System V (NQSV) User's Guide [Reference] for each operations.

4.1. Setting Accounting Rate

The table below lists the accounting items that NQSV Accounting & Budget Control can set.

Item Name	Description	Remarks
CPU	Accounting rate per second for CPU consumption time	Value for non-prime time can be set.
MEM	Accounting rate per unit memory usage (1KB * min.)	Value for non-prime time can be set.
JOB	Accounting rate per job	
GPUNUM	Accounting rate for elapsed time (per GPU * sec.)	
REQVE	Accounting rate for the elapsed time of a VE node per second. Requested VE nodes is used for calculation of the accounting. (*2)	Value >= 0
RSVVE	Accounting rate for the elapsed time of a VE node per second. Reserved VE nodes is used for calculation of the accounting. (*2)	Value >= 0
ELAPSE	Accounting rate for elapsed time (per job * sec.)	
RESERVE	Accounting rate for resource reservation section (per node * sec.)	
DEC	Weight for the declared amount of resources	
ACT	Weight for the amount of the actually used resources	
PRI_MAX	Weight for the maximum priority	Value >= 1.0
PRI_MIN	Weight for the minimum priority	0.0 <= Value <= 1.0
crname	Accounting rate per the custom resource consumption	Up to 20 resources.

*1 “crname” is specified custom resource name.

*2 These items are available only for the environment whose execution host is SX-Aurora TSUBASA.

The accounting rate can be set to each queue or each OpenStack/Docker template.
(OpenStack is NOT available for the environment whose execution host is SX-Aurora TSUBASA.)

When not using OpenStack/Docker template, set the accounting rate to only queue. The charge is calculated based on this rate for the user (or group, account code) that has been used the queue, and it is decreased from the budget. Requests submitted into a queue that the accounting rate is not set can be submitted without checking the budget even if the budget is not set. And also resource reservation sections created by specifying a queue that the accounting rate is not set can be created without checking the budget even if the budget is not set.

When using OpenStack/Docker template, set the accounting rate to queue and OpenStack/Docker template. The charge is calculated based on this rate for the user (or group, account code) that has been used the OpenStack/Docker template, and it is decreased from the budget. Requests submitted into a queue that the accounting rate is not set can be submitted without checking the budget even if the budget is not set. And also resource reservation sections created by specifying a queue that the accounting rate is not set can be created without checking the budget even if the budget is not set.

Item name REQVE and RSVVE is accounting rates for VE nodes. These are accounting rates for the elapsed time of a VE node per second. At calculation of the charges, it is possible to use both of "Number of requested VE nodes" and "Number of reserved VE nodes" or one of them. For example, if you want to charge to "Number of requested VE nodes" by the ratio of the 25% and "Number of reserved VE nodes" by the ratio of the 75%, set 25 to REQVE and 75 to RSVVE. If you set 0 to one of them, only another item is used.

4.1.1. sbuedit(1M)

sbuedit(1M) command sets the accounting rate to each queue or OpenStack/Docker template. The command execution format is as follows

```
# /opt/nec/nqsv/bin/sbuedit -N

# /opt/nec/nqsv/bin/sbuedit { add | upd }
        queue_name:acct_item=rate_in_prime[/rate_in_nonprime][,acct_item=...]

# /opt/nec/nqsv/bin/sbuedit del queue_name

# /opt/nec/nqsv/bin/sbuedit { add | upd } -t
          template_name:acct_item=rate_in_prime[/rate_in_nonprime][,acct_item=...]

# /opt/nec/nqsv/bin/sbuedit del -t template_name
```

The sbuedit(1M) command options must be set in order of operation, queue or OpenStack/Docker template, accounting item, and accounting rate. Only super-user can execute sbuedit (1M).
(template name of OpenStack is NOT available for the environment whose execution host is SX-Aurora TSUBASA.)

・ Operation

add	Add a new accounting rate settings of the queue.
upd	Update settings for a registered queue.
del	Delete a registered queue.

Execute sbuedit(1M) as follows to add a new accounting rate.

```
# /opt/nec/nqsv/bin/sbuedit add quename:CPU=1533.00/774.56,MEM=0.234,CR_Power=10.0
```

In this example, the accounting rates 1533.00 and 774.56 are added to CPU and 0.234 to MEM and 10.0 to Power(Custom resource), for the queue of quename. The 774.56 specified just after "/" of CPU is the accounting rate for non-prime time. If setting for non-prime time is omitted like MEM, the same accounting rate as prime time is used. For the definitions of prime time and non-prime time, refer to 2.2.Environment Creation. When setting the accounting rate of the custom resource, specify a "CR_" followed by a custom resource name as acct_item.

When this command is executed without any arguments, the accounting rate of each queue is displayed. The following is a display example for one queue.

```
--------------------------------------------------
QUEUE NAME [quename]
--------------------------------------------------
CPU             = 11.00/5.00
MEM             =  0.00/0.00
JOB             =  0.00
GPUNUM          =  0.00
REQVE           =  0.00
RSVVE           =  0.00
ELAPSE          =  0.00
RESERVE         =  0.00
DEC             =  0.00
ACT             =  1.00
PRI_MAX         =  1.00
PRI_MIN         =  1.00

Custom Resource Rate
Power           = 10.00
--------------------------------------------------
```

When a user uses CPU for 0.12 seconds in prime time in the queue of "quename", and the accounting rate is set as this example, the user's charge is calculated as follows.

0.12 * 11.00 = 1.32

*Refer to the sbuedit(1M) manual page in NEC Network Queuing System V (NQSV) User's Guide

[Reference] for details.

4.1.2. Handling of Accounting Rate When qmove

When you use the qmove(1) command to move a request between queues, it will be always charged at the accounting rate of the destination queue when the request ends. The queue name displayed in scacctreq(1) or scacctjob(1) is also the destination queue.

4.2. Referencing and Setting Budget

In NQSV Accounting & Budget Control function, the budget setting is performed by budgetedit(1M) command. The budgetedit(1M) edits the setting of the budget of each user, group or account-code. And budgetedit(1M) can also change the initial values of the budget, remaining values of the budget and the estimated fees.

When NQSV Budget Control function is used, it is necessary to set SBU_ORDER in the Accounting server's configuration. And the budget must be set for each users, groups or account codes according to the setting of SBU_ORDER. The Accounting server checks the budget overruns based on these settings.
For example, when the SBU_ORDER is set to "UG", a submitted request is checked for user's budget overrun first, and the request is charged to the user's budget if the amount of estimated resources is within the budget. If the user's budget is exceeded, the group's budget is checked and the request is charged to the group's budget when there are no exceeds. If the group's budget also exceeds, the request will be refused to submit.
When the budget is not set to the submitting user or group, the Accounting server treats the request as the user's or group's budget has exceeded.
And when the SBU_ORDER is set to use account code (A), a request submitted without specifying any account code is treated as account code's budget has exceeded by the Accounting server.
Also, when the SBU_ORDER is set to use group (G), a resource reservation section created without specifying any group is treated as group's budget has exceeded by the Accounting server.

4.2.1. budgetedit(1M)

The budgetedit(1M) command sets the budget for a user, group and account code. This command is used on the Accounting server. The command execution format is as follows.

```
# /opt/nec/nqsv/bin/budgetedit -N
# /opt/nec/nqsv/bin/budgetedit add|upd {-u|-g|-a} name:initial[/remain/estimate]
```

```
# /opt/nec/nqsv/bin/budgetedit del {-u|-g|-a} name
```

The budgetedit(1M) command options must be set in order of operation, target type, target name and the target's budget. Only super-user can execute budgetedit(1M).

- Operation

add	Add a new budget.
upd	Update the registered budget.
del	Delete the registered budget.

- Target type

-u	Set a budget to a user.
-g	Set a budget to a group.
-a	Set a budget to an account code. Specifies name of account code written in the account code file(/etc/opt/nec/nqsv/acctcode) as *name*. The maximum length of name of account code is 15 bytes.

Execute budgetedit(1M) as follows to add a new budget.

```
# /opt/nec/nqsv/bin/budgetedit add -u user1:125455.56,user2:5345566.44/2345566.44/0
```

In this example, the budgets 125455.56 and 5345566.44 are set to user1 and user2 respectively. The 2345566.44 and 0 delimited by "/" for user2 is the remaining budget and the estimated fee. If setting of the remaining budget and the estimated fee are omitted like user1, the initial value and remaining budget are set the same value and the estimated fee is set to 0.

When this command is executed without any arguments, the budget information is displayed in order of user, group, and account code as follows.

```
============================================================
USER                  REMAIN        ESTIMATE         INITIAL
============================================================
usr1                11223.41            0.00        12245.00
usr2              1395382.88            0.00      1399445.00
usr3               126555.98            0.00       126555.98

============================================================
GROUP                 REMAIN        ESTIMATE         INITIAL
============================================================
grp5                    0.00            0.00         1111.00
grp4                    0.00            0.00     19874344.00
```

```
=========================================================
ACCOUNT              REMAIN        ESTIMATE          INITIAL
=========================================================
acct1                 0.00            0.00          1111.00
acct2                 0.00            0.00      19874344.00
```

The displayed information is user name (group name, account-code), remaining budget, estimated fee and the initial budget. When the remaining budget is 0, it means the budget overrun occurs for the user (group, account-code).

To update the registered budget, do one of the following:
-When specifying an absolute value
Update the budget with the specified budget amount. You cannot specify the signed amount.

```
# /opt/nec/nqsv/bin/budgetedit upd -u user1:9999/8888/7777
```

-When specifying a relative value
Update the budget with the result of adding and subtracting the current budget amount with the specified relative value. A sign is required for the specified amount. You cannot specify a combination of minus and plus signs. Specify 0 as the value that is not changed.

*Refer to the budgetedit(1M) manual pages in NEC Network Queuing System V (NQSV) User's Guide [Reference] for details.

4.2.2. budgetcheck(1)

The budgetcheck(1) is a command to check the use of the budget for user, group and account code.
The budgetcheck(1) command searches the budget usage of user, group or account code according to the specified options (-u, -g or -a) and it displays remaining budget, estimated charge and initial budget for each user, group or account code to the standard output. As the result of execution, budgetcheck(1) returns a value that shows the budget exceeds or not. If the budget exceeds the limitation, exit status of the budgetcheck(1) will be equal to or greater than 1. This command can be used on the Accounting server.

budgetcheck(1) command execution format is as follows.

```
# /opt/nec/nqsv/bin/budgetcheck -N
# /opt/nec/nqsv/bin/budgetcheck [-u][-g][-a]
```

The exit status of budgetcheck(1) command is as follows. When more than 1 type of budget are exceeded, the exit status will be the logical additions of each exit status.

Exit status	Budget overruns
0	No budget overrun
1	User's budget is exceeded
2	Group's budget is exceeded
4	Account-code's budget is exceeded

budgetcheck(1) is executed as follows to check the user and group budgets.

```
$ /opt/nec/nqsv/bin/budgetcheck -u -g
usr1                    0.00            0.00        12245.00
grp4                    0.00            0.00     19874344.00
$ echo $?
3
```

The exit status will be 1 in case of exceeding the user budget, 2 in case of exceeding the group budget and 3 in case of exceeding both user and group budget.

The budgetcheck(1) command checks the budget usage of the following scope of users, groups and account codes according to the NQSV user privilege.

Privilege	Scope of reference
General User privilege	-u: Budget information of the executed user can be checked. -g: Budget information of the groups that the executed user belongs to can be checked. -a: Budget information of the account codes that the executed user belongs to can be checked.
Special User Privilege	
Group Manager Privilege	-u: Budget information of the executed user can be checked. -g: Budget information of the groups that the executed user manages can be checked. -a: Budget information of the account codes that the executed user belongs to can be checked.
Operator Privilege	Budget information of all users, groups and account codes can be checked.
Manager Privilege	

*Refer to the budgetcheck(1) manual page for details.

4.2.3. rbudgetcheck(1)

rbudgetcheck(1) command is used to check the budget on the host on which NQSV/Client is installed.

rbudgetcheck(1) command executes budgetcheck(1) on the Accounting server remotely to check the budget.

rbudgetcheck(1) command execution format is as follows. The first argument of the rbudgetcheck(1) command is the Accounting server's host name.

```
# /opt/nec/nqsv/bin/rbudgetcheck acct-server [-u][-g][-a]
```

How to specify the budget type (user, group or account code), how to specify privilege, output of the command and the exit code are the same as budgetcheck(1).

*Refer to rbudgetcheck(1) manual pages in NEC Network Queuing System V (NQSV) User's Guide [Reference] for details.

Appendix.A Others

A.1 Notes in the operations

(1) Optimization of accounting database

Though the accounting data sent from Batch server and JobManipulator is accumulated in the database on the Accounting server, the amount of the accounting data continuously grows with time and frequency of system use. As for the database used with NQSV Accounting & Budget Control, the overhead for searching data will increase as the amount of the registered data becomes larger.

The sceditacct(1M) command of the Accounting server can optimize the database. Execution time of the display command of the accounting data is shortened by optimizing the database.

The example of executing the sceditacct(1M) command is shown below.

```
/opt/nec/nqsv/bin/sceditacct -j -S 1/1 -m /tmp/jacct2019_03 (Job accounting)
/opt/nec/nqsv/bin/sceditacct -r -S 1/1 -m /tmp/racct2019_03 (Request accounting)
/opt/nec/nqsv/bin/sceditacct -y -S 1/1 -m /tmp/rsvacct2019_03 (Reservation accounting)
```

In this case, all job or request accounting data is saved to the file specified by -m option. As this database has been optimized, the display time of the scacctreq(1) command, the scacctjob(1) command and the scsumacct(1M) command is shortened.

The sample of executing the cron for saving the job accounting data monthly is shown below.

```
0 0 1 * * ( TODAY=`/bin/date +"%d\/%m\/%y"` ; export $TODAY ; FILEDATE=`/bin/date +"%d%m%y" ; export
$FILEDATE ; /opt/nec/nqsv/bin/sceditacct -j -E $TODAY -m /var/back/jacct.$FILEDATE )
```

Note that start time and end time of the request is registered as 0 in their database before execution is terminated. The sceditacct (1M) command excludes request accounting data execution has not ended from the operation target. When including such data in the operation target, use -f option without -s and -S option.
When including the request accounting data deleted before starting in the operation target and excluding the request accounting data execution has not ended from the operation target, specify the investment time by -q or -Q option. In this case, -f option need not be specified.

*Refer to the 3.5 Saving Accounting Data for details.

(2) The upper limit of the accounting database file size

When the size of each accounting database file is more than 1800MB, accounting data isn't registered any more. Delete (migrate) accounting data periodically via the sceditacct(1M) command. The upper limit can be changed by setting the value of DBSIZE_LIMIT in the accounting server configuration file.

(3) The upper limit of the account file size

When the size of each account file is more than 128GB, data will not be transferred and registration with the accounting database will stop. If file size becomes large, it may affect the aggregation performance of the account, so rotate the account file periodically via the scturnacct(1M) command. The upper limit can be changed by setting the value of ACCTFILESIZE_LIMIT in the accounting monitor configuration file.

(4) Each accounting file about the case where it save excluding scturnacct(1M) command

The Accounting monitor store the position of the last collected record in the request or reservation accounting file into a ".last" file. And the scturnacct(1M) command manages the generation of accounting file by storing the already collected records into other name of the accounting file.

If you save the accounting data without using scturnacct(1M), NQSV Accounting & Budget control function loses the control of the accounting data. Please make sure to use scturnacct(1M) command to save the accounting files.

If the accounting file is moved or deleted without using scturnacct(1M) command, deleted a ".last" file. In this case, the accounting data in the moved or deleted files cannot be sent to the Accounting server.

(5) List of used VE nodes for the Job

List of used VE nodes for the Job is made based on existence confirmation of processes which use VE per the "interval to get resource quantity of jobs". Therefore, when a process is generated and exited between the intervals, correct information may not collect.

(6) Regarding the script that needs to be regularly executed when the accounting function is enabled.

When the accounting function is enabled, the record information of a request has the request ID, BSV host's address, number of jobs and so on registered onto request account DB (racct) as the initial value when a request is submitted.

After that, the record information is updated by the actual achievement value and so on when a request end. When it isn't possible to update the actual achievement value, the record of the request account registered at request submitting keeps being left as the initial value. Even if the DB is saved by scacctedit -E and so on, this initial state record is not cleared.

If the initial state record remains and the request ID completes a cycle (default：0 to 999999) , and a request with the same basic information of the remaining record and request is submitted, a database duplicate error will occur.

At the time of request completion or deletion, the remaining initial state records are updated. However, certain columns such as planned accounting amount, number of jobs, user, and group continue to use the information from the old request, resulting in the incorrect reflection of accounting information for the correct user and group.

To prevent such issues, we recommend regularly executing a script (cleanup_racct.sh) to delete the remaining records that are still in their initial state. It is recommended to run this script before the request ID completes a cycle based on user operation status.

However, it is not possible to distinguish between the remaining records that are still in their initial state and the records in a state where information has not been updated yet, such as those in a pending or running state after the request has been submitted. Therefore, please execute this script only when all of the following conditions are met.

- There are no requests in a pending or running state after submitting a request using the qstat command.
- There are no files in the /var/opt/nec/nqsv/asv/master/<IP address>/ directory.
- Before stopping NQSV, the information of the last successfully executed request is displayed as scacctreq, and accounting amount aggregation and reflection are completed.
- The operation of NQSV has been stopped.

Here is the method for executing this script:

```
# /opt/nec/nqsv/sbin/cleanup_racct.sh
Deleted Request ID
  132321
  132322

Please confirm no requests are in progress before executing the deletion
```

When executing the script without any options, it will only perform a check to see if there are any

records to be deleted. If there are records to be deleted, the request IDs will be displayed on the standard output. If no request IDs are displayed, further processing is not necessary.

If request IDs are displayed, please execute the script with the -f option. This will delete the records to be deleted from the database.

```
# /opt/nec/nqsv/sbin/cleanup_racct.sh -f
Output /var/opt/nec/nqsv/asv/cleanup_backup/deleted_racct_records.bak_ YYYYMMDDhhmmss
Deleted records from racct
```

The records to be deleted will be stored in the /var/opt/nec/nqsv/asv/cleanup_backup directory, with the file name in the format "deleted_racct_records.bak_YYYYMMDDhhmmss".

If you accidentally delete the initial state records for queued/running/DB aggregation requests, they cannot be recovered. Therefore, please execute the script at the timing specified above to meet the aforementioned conditions.

A.2 Exit Status For Reservations

Exit status of reservation is confirmed by the STATUS of the scacctrsv(1) command. Following is the description of STATUS.

STATUS	DESCRIPTIONS
DONE	The reservation has ended.
DELETED	The reservation was deleted before starting.
QUIT	The reservation was deleted between its starting and ending.

A.3 Exit Status For Requests

Exit status of request is confirmed by STATUS or EXIT STAT of the scacctreq(1) command. When confirming the exit status of request simply, refer to STATUS. Following is the description of STATUS.

STATUS	DESCRIPTIONS
DONE	The request has ended.
DELETED	The request was deleted.
RERUN	The request was reruned.

When confirming the exit status of request in detail, refer to EXIT STAT that is displayed by the -h option of the scacctreq(1) command. Following is the format and the meaning of EXIT STAT. -R option displays details after the end of message code.

FORMAT	DESCRIPTIONS	
YYYYXXXX	**YYYY**	Operator
	XXXX	Termination code

* EXIT STAT is displayed by the hex digit. And the first 0 is not displayed.

Operator	DESCRIPTIONS
0x0001	owner
0x0002	manager, operator, system

* When the Termination code is 0x0002 or 0x0003, the operator is set.

Termination code	DESCRIPTIONS (batch, interactive, subrequest)	DESCRIPTIONS (parametric request)
0x0000	The master job has ended in exit(0).	All subrequests has ended in exit(0)
0x0001	The master job has ended besides exit(0).	More than 1 sub-request has ended besides exit(0).
0x0002	The request was deleted.	The parametric request was deleted.
0x0003	The request was reruned.	The parametric request was reruned.

A.4 Exit Status for Jobs

Exit status of job is confirmed by the EXIT STAT that is displayed by the -h option of the scacctjob(1) command. Following is the format and the meaning of EXIT STAT. -R option displays details after the end of message code.

FORMAT	DESCRIPTIONS	
ynnxxxx	**y**	Operation information.
or	**nn**	Termination code of a login shell.
yssxxxx	**ss**	Signal number.
	xxxx	Termination code of a job.

* EXIT STAT is displayed by the hex digit. And the first 0 is not displayed.
If termination code of a job is 0x402, and signal number is 15, and operation information is 1, EXIT STAT is **10f0402**. However, if operation information is 0, EXIT STAT is **f0402**.
If termination code of a job is 0x402, and there is no signal number, and operation information is 1, EXIT STAT is **1000402**. However, if operation information is 0, EXIT STAT is **402**.

EXIT STAT	DESCRIPTIONS
nn0000	Normal termination.
nn0001	Canceled by owner.
nn0003	System crash.
nn0004	Orphan Job.
ss0402	
nn0802	Deleted by owner.
ss2402	Rerun by owner.
nn4000	
nn4001	Migrated by owner
1nn0001	Canceled by operator.
1ss0402	
1nn0802	Deleted by operator.
1ss2402	Rerun by operator.
1nn4000	
1nn4001	Migrated by operator
2nn0202	Failed to execution.
2ss0402	Deleted by signal.
2nn0502	Deleted by Shutdown.
2nn1102	Failed to restart.

2nn1802	Failed to release.
2nn2502	Rerun by Shutdown.
3nn0001	Canceled by manager.
3ss0402	
3nn0802	Deleted by manager.
3ss2402	Rerun by manager.
3nn4000	
3nn4001	Migrated by manager.
4nn0001	Canceled by scheduler
4ss0402	
4nn0802	Deleted by scheduler.
4ss2402	Rerun by scheduler.
4nn4000	
4nn4001	Migrated by scheduler
5nn0001	Canceled by system
5ss0402	
5nn0802	Deleted by system.
5ss2402	Rerun by system.
5nn4000	Migrated by system.
5nn4001	

A.5 Charge Calculation

The charge is calculated by the following method. The charge is calculated based on the resource limits of the requests or resource reservation sections. Set the resource limits for the charge calculation.

<Calculation of the charge>

The charge = Total of the charge every resource

<Calculation of charge every resource>

The charge every resource = (DEC * the declaration value + ACT * the actual value) * the accounting rate for target resource * the priority rate

<Calculation of the declaration value and the actual value>

For resource reservation sections

resource	the declaration value	the actual value (*1)
RESERVE	number of hosts or number of machines ×The length of scheduled time	number of hosts or number of machines × The length of actual time

For requests

resource	the declaration value	the actual value (*1)
CPU	Total CPU time requested of all jobs	Total cpu time of all job
MEM	Total size of requested maximum memory of all jobs × Elapse time limit for request	Total memory usage of all job
JOB	Number of job	Number of job
GPUNUM	Total number of GPU Limit for each job × Elapse time limit for request	∑(Number of GPU Limit for each job × Elapse time of each job)
REQVE	Number of requested VE nodes ×Elapse time limit for request	∑(Number of requested VE nodes for the Job×Elapse time of each job)
RSVVE	Number of reserved VE nodes *Elapse time limit for request	∑(Number of reserved VE nodes for the Job×Elapse time of each job)
ELAPSE	Elapse time limit for request × Number of job	Total Elapse time of all job
cr_name (*2) (check_mode=off)	If consumption unit is request： Requested amount of Custom Resource If consumption unit is job：	

	Requested amount of Custom Resource × Number of job ("Requested amount of Custom Resource " is the value specified by "--custom" option at request submitting)	
cr_name (*2) (check_mode=integrate)	If consumption unit is request : Request amount of Custom Resource If consumption unit is job : Request amount of Custom Resource × Number of job	Total usage of Custom Resources of all jobs
cr_name (*2) (check_mode=moment)	if consumption unit is request : Custom Resource Amount to be used * Elapse time limit for request (min) if consumption unit is job : Custom Resource Amount to be used * Number of job * Elapse time limit for request (min)	∑(Total usage of Custom Resources of each job × Elapse time of each job)

*1 In the budget overage check, the estimated charge is calculated by regarding the actual value and the declared value as the same.

*2 Custom resource function can use check_mode option. The check_mode sets the resource monitoring of custom resources and whether to output them. No monitoring and output when check_mode is off. When check_mode is moment, custom resource usage is monitored and output as a momentary value. When check_mode is integrate, custom resource usage is monitored and output as an integrated value. See “NEC Network Queuing System V (NQSV) User’s Guide [Management]” for detail.

<Calculation of the priority rate >

Request Priority	Calculation of the priority rate
1 to 1023	1.0 + Request Priority * (PRI_MAX – 1.0)/1023
0	1.0
-1 to -1024	1.0 + Request Priority * (1.0 – PRI_MIN)/1024

Appendix.B List of Accounting files

In this section, each file and path are shown in tables.

- Accounting server

File	Description
/var/opt/nec/nqsv/asv/jacct	Database file for job accounting
/var/opt/nec/nqsv/asv/jacct_cr	Database file for job accounting (for custom resource)
/var/opt/nec/nqsv/asv/jacct_ps	Database file for job accounting (for VE process account)
/var/opt/nec/nqsv/asv/racct	Database file for request accounting
/var/opt/nec/nqsv/asv/racct_tally	Database file for request accounting(for aggregation data)
/var/opt/nec/nqsv/asv/racct_cr	Database file for request accounting(for custom resource data)
/var/opt/nec/nqsv/asv/racct_cr_tally	Database file for request accounting (for custom resource aggregation data)
/var/opt/nec/nqsv/asv/racct_ps_tally	Database file for request accounting (for VE process account aggregation data)
/var/opt/nec/nqsv/asv/rsvacct	Database file for reservation accounting
/var/opt/nec/nqsv/asv/sbu	Database file for accounting rate
/var/opt/nec/nqsv/asv/budget	Database file for budget
/var/opt/nec/nqsv/asv/master/XXX.XXX.XXX.XXX[1]/tjad_NNNNN[2]	Received job accounting intermediate file

/var/opt/nec/nqsv/asv/master/XXX.XXX.XXX.XXX[1]/racct_NNNNN[2]	Received request accounting intermediate file
/var/opt/nec/nqsv/asv/master/XXX.XXX.XXX.XXX[1]/rsvacct_NNNNN[2]	Received reservation accounting intermediate file

[1] XXX.XXX.XXX.XXX is IP address of the Accounting monitor.

[2] NNNNN is the sequential number which starts from 00000.

・ Accounting monitor

File	Description
/var/opt/nec/nqsv/bsv/account/reqacct[.N][1]	Request accounting data
/var/opt/nec/nqsv/bsv/account/reqacct_del[.N][1]	Request accounting data (for requests deleted before starting)
/var/opt/nec/nqsv/bsv/account/.reqacct.last	File for saving the position of transferred request accounting data
/var/opt/nec/nqsv/bsv/account/.reqacct.yet	Request accounting data which is not transferred yet
/var/opt/nec/nqsv/bsv/account/.reqacct.yet.last	File for saving the position of transferred request accounting data(for .reqacct.yet)
/var/opt/nec/nqsv/acm/jacct/jobacct	Job accounting data
/var/opt/nec/nqsv/acm/jacct/.jobacct.last	File for saving the position of transferred job accounting data
/var/opt/nec/nqsv/acm/jacct/.jobacct.yet	Job accounting data which is not transferred yet
/var/opt/nec/nqsv/acm/jacct/.jobacct.yet.last	File for saving the position of transferred job accounting data(for .jobacct.yet)
/var/opt/nec/nqsv/acm/rsvacct/rsvacct	Reservation accounting data
/var/opt/nec/nqsv/acm/rsvacct/.rsvacct.last	File for saving the position of transferred reservation accounting data
/var/opt/nec/nqsv/acm/rsvacct/.rsvacct.yet	Reservation accounting data which is not transferred yet
/var/opt/nec/nqsv/acm/rsvacct/.rsvacct.yet.last	File for saving the position of transferred reservation accounting data(for .rsvacct.yet)

[1] [.N] is a generation number. It is omitted when a generation number is 0.

Appendix.C Update history

C.1 List of update history

2018	February	1st edition
2021	December	11th edition
2023	January	12th edition
2023	March	13th edition
2023	June	14th edition
2024	March	15th edition

C.2 Details of additions and changes

・ 11th edition
  - 3.6.4 The Setting of Process Account Aggregation Script
    Corrected description.

・ 12th edition
  - 4.1 Setting Accounting Rate
    Corrected the description of applicable accounting rate at request qmove and move it to 4.1.2 Handling of Accounting Rate When qmove.

・ 13th edition
  - 3.1 Referencing Request Accounting Data, 3.2 Referencing Job Accounting Data
    Corrected the description of V KCORE MIN.
  - Added description of VE MEAN SIZE(K).3.6.2 List of Aggregation Items
    Updated structure name on VEOS side of VE process account.

・ 14th edition
  - 4.1.2. Handling of Accounting Rate When qmove
    Corrected the queue to which the accounting rate is applied due to the bugfix of the billing queue at qmove in NQSV R1.13.

・ 15th edition
  - 3.1 Referencing Request Accounting Data, 3.2 Referencing Job Accounting Data
    Corrected the description of REQUEST -NAME.
  - A.1 (6) Regarding the script that needs to be regularly executed when the accounting

function is enabled.

Added the description of the script that needs to be regularly executed.

NEC Network Queuing System V (NQSV) User's Guide
[Accounting & Budget Control]

March 2024 15th edition

NEC Corporation

Copyright: NEC Corporation 2018

No part of this guide shall be reproduced, modified or transmitted without a written permission from NEC Corporation.
The information contained in this guide may be changed in the future without prior notice.