
NEC Network Queuing System V (NQSV) 利用の手引

[アカウントティング・予算管理編]

輸出する際の注意事項

本製品（ソフトウェアを含む）は、外国為替および外国貿易法で規定される規制貨物（または役務）に該当することがあります。

その場合、日本国外へ輸出する場合には日本国政府の輸出許可が必要です。

なお、輸出許可申請手続きにあたり資料等が必要な場合には、お買い上げの販売店またはお近くの当社営業拠点にご相談ください。

は し が き

本書は、NQSⅤにおけるアカウンティング・予算管理機能の利用方法について説明したものです。

備考

- (1) 本書はNEC Network Queuing System V (NQSV) R1.00以降に対応しています。
- (2) 本書に説明しているすべての機能はプログラムプロダクトであり、以下のプロダクト名およびプロダクト番号に対応しています。

プロダクト名	型番
NEC Network Queuing System V (NQSV) /ResourceManager	UWAF00 UWHAF00 (サポートパック)

- (3) UNIX は The Open Group の登録商標です。
- (4) Intelは、アメリカ合衆国およびその他の国における Intel Corporation の商標です。
- (5) OpenStack は、アメリカ合衆国およびその他の国における OpenStack Foundation の商標です。
- (6) Red Hat OpenStack Platformは、アメリカ合衆国およびその他の国における Red Hat, Inc. の商標です。
- (7) Linux は Linus Torvalds氏の米国およびその他の国における登録商標あるいは商標です。
- (8) Dockerはアメリカ合衆国およびその他の国におけるDocker, Inc.の商標です。
- (9) InfiniBand は、InfiniBand Trade Associationの商標またはサービスマークです。
- (10) Zabbixはラトビア共和国にあるZabbix LLCの商標です。
- (11) その他、記載されている会社名、製品名は、各社の登録商標または商標です。

本書の対象読者

本書は、システム管理者と一般利用者を対象読者として書かれたものです。

とくにシステム管理者には、本書は有効な説明書となります。

本書は、対象読者が Linux の一般操作および、NQSV に関する一般的な事項について知っていることを前提として記述しています。

本書の読み進め方

本書は、次の構成となっています。章ごとに対象読者の範囲は異なっており、表の一番右の列にその範囲を示しています。記載された対象読者の後に(*)がついている章については、該当する読者は必ずお読みください。

章	タイトル	内容	対象読者
1	NQSVのアカウントティング・予算管理機能の概要	NQSVのアカウントティング・予算管理機能の概要	システム管理者(*) 一般利用者
2	環境構築・起動	アカウントティング・予算管理機能のインストール、環境構築およびデーモンの起動・停止	システム管理者(*)
3	アカウントティング	アカウントデータの参照、日報・月報データの参照、アカウントデータの退避	システム管理者(*) 一般利用者
4	予算管理	課金レートの設定、予算の設定や参照	システム管理者(*)
5	付録	各種終了ステータス、課金額の計算方法、コマンドリファレンスおよびアカウントファイル一覧	システム管理者(*) 一般利用者

関連説明書

NEC Network Queuing System V (NQSV)のマニュアルは以下で構成されています。

マニュアル名称	内容
NEC Network Queuing System V (NQSV) 利用の手引 [導入編]	システムの全体像および基本的なシステムの構築方法について説明しています。
NEC Network Queuing System V (NQSV) 利用の手引 [管理編]	NQSV の管理者が実施する各種設定について説明しています。
NEC Network Queuing System V (NQSV) 利用の手引 [操作編]	NQSV の一般利用者が使用する各種機能について説明しています。
NEC Network Queuing System V (NQSV) 利用の手引 [リファレンス編]	コマンドリファレンスに関して説明しています。
NEC Network Queuing System V (NQSV) 利用の手引 [API 編]	NQSV を操作する C 言語のプログラミングインタフェース (API) について説明しています。
NEC Network Queuing System V (NQSV) 利用の手引 [JobManipulator 編]	NQSV のスケジューラコンポーネント JobManipulator に関して説明しています。
NEC Network Queuing System V (NQSV) 利用の手引 [アカウントティング・予算管理編]	NQSV のアカウントティング機能に関して説明しています。

表記上の約束

本書では次の表記規則を使用しています。

省略記号 ... 前述の項目を繰り返すことができることを表しています。ユーザは同様の項目を任意の数だけ入力することができます。

縦棒 | オプションまたは必須の選択項目を分割します。

中かっこ {} 1つを選択しなければならない一連パラメータまたはキーワードを表しています。

角かっこ [] 省略可能な一連パラメータまたはキーワードを表しています。

用語定義・略語

用語・略語	説 明
ベクトルエンジン (VE、Vector Engine)	SX-Aurora TSUBASA の中核であり、ベクトル演算を行う部分です。複数のコアと、 共用メモリが実装されています。PCI Express カードであり、x86 サーバに搭載して使用します。
ベクトルホスト (VH、Vector Host)	ベクトルエンジンを保持するサーバ、つまり、ホストコンピュータを指します。
ベクトルアイランド (VI)	VH1台にVEを1枚ないし複数枚組み込んだ単位を指します。
バッチサーバ (BSV)	バッチサーバホスト上に常駐し、NQSV全体を管理します。
ジョブサーバ (JSV)	実行ホスト上に常駐し、ジョブの実行を管理します。
JobManipulator (JM)	NQSVの提供するスケジューラ機能です。計算リソースの空き容量を管理し、ジョブの開始時刻を割り当てます。
アカウントティングサーバ	アカウント情報の収集・管理、予算管理を行います。
リクエスト	NQSVにおけるユーザジョブの単位であり、1つまたは複数のジョブから構成されたものです。バッチサーバによって管理されます。
ジョブ	ユーザジョブの実行単位であり、ジョブサーバによって管理されます。
論理ホスト	実行ホストの資源を論理的（仮想的）に分割し、ジョブに割り当てたものです
キュー	受け取ったリクエストをプールし、管理する機構です。
BMC	Board Management controllerの略語です。IPMI(Intelligent Platform Management Interface)と呼ばれる業界標準のサーバーマネジメントインタフェースに準拠したサーバ管理を行います。
HCA	Host Channel Adapterの略語です。IBネットワークに接続するためにサーバ側に取り付けるPCIeカードです。
IB	InfiniBandの略語です。
MPI	Message Passing Interfaceの略語です。主にノード間で並列コンピューティングを行うための標準規格です。
NIC	Network Interface Cardの略語です。他ノードと通信するためのハードウェアです。

目 次

は し が き	i
用語定義・略語	vi
目 次	vii
図目次	ix
第 1 章 NQSV のアカウンティング・予算管理機能の概要	1
1.1 はじめに	1
1.2 コンポーネント	1
1.3 各アカウントデータの流れ	2
第 2 章 環境構築・起動	4
2.1 インストール	4
2.1.1 アカウンティングサーバ	4
2.1.2 アカウンティングモニタ	4
2.1.3 AUI	4
2.2 環境の構築	5
2.2.1 アカウンティングサーバ	5
2.2.2 アカウンティングモニタ	9
2.2.3 バッチサーバ	10
2.2.4 AUI	15
2.3 起動・終了	17
2.3.1 アカウンティングサーバ	17
2.3.2 アカウンティングモニタ	18
第 3 章 アカウンティング	19
3.1 リクエストアカウントデータの参照	19
3.1.1 scacctreq の実行例	24
3.2 ジョブアカウントデータの参照	26
3.2.1 scacctjob の実行例	30
3.3 予約アカウントデータの参照	32
3.3.1 scacctrsv の実行例	33
3.4 日報・月報データの参照	35
3.4.1 日報	36
3.4.2 月報	37
3.5 アカウントデータの退避	38

3.5.1	アカウンティングサーバ.....	38
3.5.2	アカウンティングモニタ.....	41
3.6	VE プロセスアカウント.....	43
3.6.1	利用可能な運用構成.....	43
3.6.2	集計項目一覧.....	43
3.6.3	プロセスアカウント集計結果 CSV ファイル.....	45
3.6.4	プロセスアカウント集計スクリプトの設定.....	46
3.6.5	アカウント DB のプロセスアカウント集計設定.....	49
3.6.6	Docker 環境でプロセスアカウント機能を利用する場合.....	50
第 4 章	予算管理.....	52
4.1	課金レートの設定.....	52
4.1.1	sbuedit(1M).....	53
4.1.2	qmove 時の課金レートの扱い.....	55
4.2	予算の参照、設定.....	56
4.2.1	budgetedit(1M).....	56
4.2.2	budgetcheck(1).....	58
4.2.3	rbudgetcheck(1).....	59
付録 A	注意点および補足説明.....	61
A.1	運用上の注意点.....	61
A.2	予約の終了ステータスについて.....	64
A.3	リクエストの終了ステータスについて.....	65
A.4	ジョブの終了ステータスについて.....	66
A.5	課金額の計算について.....	68
付録 B	アカウントファイル一覧.....	70
付録 C	発行履歴.....	72
C.1	発行履歴一覧表.....	72
C.2	追加・変更点詳細.....	72
索 引		74

図目次

図 1 アカウントデータの流れ.....	2
----------------------	---

第1章 NQSV のアカウントティング・予算管理機能の概要

1.1 はじめに

NQSV では、アカウントティングサーバを利用してアカウントデータの管理を行うことができます。

また、アカウントティングサーバは予算管理機能をサポートしており、アカウントデータからシステムの利用料金を算出して予算情報を更新したり、予算を超過している利用者のリクエスト投入や予約区間の作成を制限したりすることができます。

1.2 コンポーネント

アカウントデータ・予算管理は以下のコンポーネントにより行われます。

- アカウントティングサーバ

アカウントティングサーバはアカウント情報の収集を行うサーバです。このサーバの存在するマシン上でアカウントデータの参照、予算の設定、課金レートの設定を行います。

アカウントティングサーバは、バッチサーバ、JobManipulatorと同じホスト上に配置することが可能です。

- アカウントティングモニタ

アカウントティングモニタはアカウントティング情報をアカウントティングサーバへ送信します。バッチサーバからはジョブアカウントデータおよびリクエストアカウントデータを採取し、JobManipulatorからは予約アカウントデータを採取してアカウントティングサーバに送信します。

アカウントティングモニタは、バッチサーバホスト、および、JobManipulatorと同じホスト上に配置する必要があります。

1.3 各アカウントデータの流れ

NQSV では、リクエストが実行されると、バッチサーバがジョブアカウント情報とリクエストアカウント情報を出力します。アカウントティングモニタはこれらのアカウント情報をアカウントティングサーバへ送信します。また、リソース予約区間が生成されると、JobManipulator が予約アカウント情報を出力します。アカウントティングモニタはこのアカウント情報をアカウントティングサーバへ送信します。アカウントティングサーバでは、アカウントティングモニタから受信したアカウントデータをもとに、各アカウントのデータベースの更新および予算実績の更新を行います。

下図は各アカウントデータの流れを示したものです。

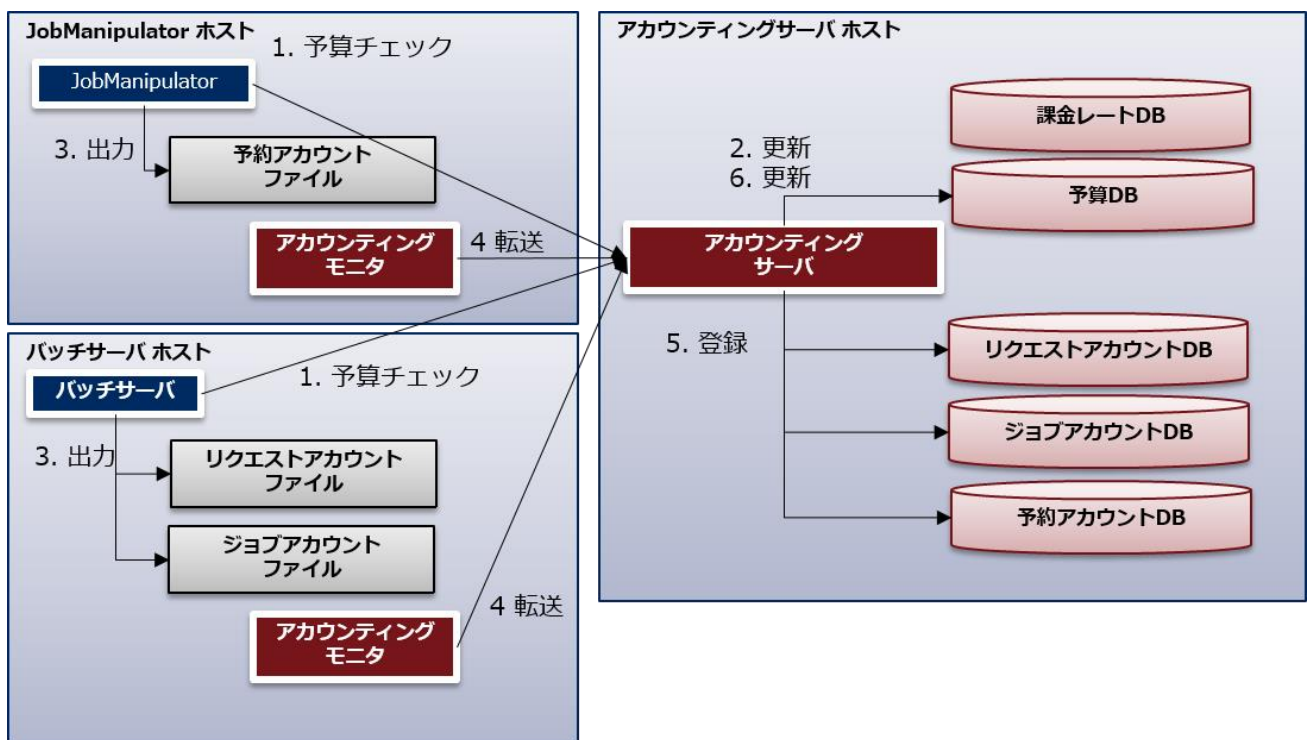


図 1 アカウントデータの流れ

(1) 予算チェック

リクエスト投入もしくはリソース予約区間の生成を行うと、アカウントティングサーバへ予算チェック要求が行われます。

(2) 予算データの更新

アカウントティングサーバは予算チェック要求の情報をもとに予定課金額を見積り、予算を超過しないか確認します。予算が足りている場合は予算データの予定課金額を更新します。

(3) アカウントファイルの出力

予約区間もしくはリクエストが終了すると、アカウントファイルが出力されます。

(4) アカウントファイルの転送

アカウントティングモニタは3で出力されたアカウントファイルをアカウントティングサーバへ転送します。

(5) アカウントデータの登録

アカウントティングサーバは4で受信したアカウントデータを各アカウントDBへ登録します。

(6) 予算データの更新

アカウントティングサーバは4で受信したアカウントデータおよび課金レートをもとに実課金額を計算し、予算データを更新します。

第2章 環境構築・起動

2.1 インストール

NQSV のアカウントिंग・予算管理機能を利用するためには、アカウントिंगサーバの構築が必要です。また、バッチサーバおよび JobManipulator を配置する各ホスト上にアカウントिंगモニタをインストールする必要があります。

各パッケージについては、NEC Network Queuing System V (NQSV)利用の手引 [導入編]を参照してください。各パッケージのインストールについても、NEC Network Queuing System V (NQSV)利用の手引 [導入編]の内容に従って実施してください。

2.1.1 アカウントिंगサーバ

アカウントングサーバ上に、NQSV/ResourceManager パッケージをインストールしてください。このパッケージ中に、アカウントングサーバデーモンとアカウントングサーバ付属のコマンドが含まれています。

2.1.2 アカウントングモニタ

バッチサーバおよび JobManipulator を配置する各ホスト上に、NQSV/ResourceManager パッケージをインストールしてください。このパッケージ中に、アカウントングモニタデーモンとアカウントングモニタ付属のコマンドが含まれています。

2.1.3 AUI

AUI(Accounting User Interface)はアカウントングサーバにアクセスして情報を表示するコマンドです。クライアントホスト上に、NQSV/ResourceManager の中の NQSV-Client-X.XX-X.x86_64.rpm パッケージをインストールしてください。このパッケージ中に、AUI コマンドが含まれています。

2.2 環境の構築

NQSV のアカウントティング・予算管理機能の利用に際して、以下の環境設定を行ってください。

環境構築・運用における注意事項については「付録 A 注意点および補足説明」節を参照してください。

2.2.1 アカウントティングサーバ

- 設定ファイルの編集

アカウントティングサーバにおいて、サーバーデーモン起動のために、設定ファイル (/etc/opt/nec/nqsv/asvd.conf) に次の設定をしてください。

項目	意味	内容
RECV_PORT_FOR_ACCT	アカウントデータ受信用ポート番号	アカウントティングサーバデーモンがアカウントティングモニタとの通信に使用するポート番号を設定します。デフォルトでは 6542 が設定されています。他に重複するアプリケーションがなければ変更する必要はありません。
ALLOW_CLIENTS	接続を許可するホストのIPアドレス	接続を許可するバッチサーバ、および、JobManipulatorのIPアドレスを指定します。デフォルトではすべてのホストからの接続を許可します。IPアドレスを複数指定する場合はカンマ(,)で区切ってください。またハイフン(-)でIPアドレスの範囲を指定することができます。
SBU_CHECK	予算管理機能のON/OFF	NQSVの予算管理機能を有効にする場合 ON、無効にする場合は OFF を設定します。デフォルトでは OFF が設定されています。
RECV_PORT_FOR_SBU	予算要求受付用ポート番号	アカウントティングサーバが、バッチサーバ、および、JobManipulatorからの予算要求を受け付けるための通信に使用するポート番号を設定します。デフォルトでは 4595 が設定されています。他に重複するアプリケーションがなければ変更する必要はありません。
LOG_FACILITY	syslogのファシリティ	syslogのファシリティを指定します。デフォルトはLOG_LOCAL0です。
SBU_ORDER	課金対象とする予算種別の優先順位	課金対象とする予算種別の優先順位を以下の文字で指定します。ここで指定された予算種別に対して左から順番に予算のチェックを行います。文字の意味は以下の通りです。 A : アカウントコード U : ユーザ G : グループ 指定に含まれない予算種別に対しては予算のチェックを行いません。デフォルトでは、GUAが設定

		されています。
ACCT_DIR	アカウントファイルの格納ディレクトリ	受信したアカウントファイルの格納先ディレクトリを指定します。 デフォルトのディレクトリは、 /var/opt/nec/nqsv/asv/masterです。
LOCK_DIR	ロックファイルの格納ディレクトリ	ロックファイルの格納先ディレクトリを指定します。デフォルトのディレクトリは、 /var/opt/nec/nqsv/asv/masterです。
DBSIZE_LIMIT	アカウントDBファイルの上限サイズ	アカウントの各DBファイルの上限サイズをMB単位で設定します。デフォルトは1800MBです。設定可能な範囲は、1～131072です。このサイズを超えると、アカウントデータのDB登録が停止します。アカウントDBファイルの上限サイズを大きくするとアカウントの集計性能や表示性能が劣化する可能性がありますので、1800MBを推奨します。
VEACCT_TIME_MARGIN	VEプロセスアカウント集計の時刻マージン	VEジョブのプロセスは、ジョブの終了時刻よりも後に終わる場合があります。VEOSから出力されるプロセスアカウントを正しくジョブアカウントに集計するために、その時刻の調整マージンを秒単位で設定します。この設定により、ジョブの終了時刻よりも遅れて終了したプロセスのプロセスアカウントが設定値の時間範囲内であれば、ジョブアカウントへ集計されます。既定値は60秒です。

- 予算管理を実施する際の設定

NQSV のアカウントティング・予算管理機能において予算管理を実施する場合、プライムタイム、ノンプライムタイム、および休日の設定が必要です。設定は /etc/opt/nec/nqsv/asv_holidays を編集します。

* 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

Prime Start のカラムにはプライムタイムの開始時刻を、Non-Prime Start のカラムにはノンプライムタイムの開始時刻を HH24MI 形式で指定します。また、Day Of Year 以降に休日を登録することが可能です。Day Of Year のカラムには年の開始からの日数を指定します。Calendar Date、Company Holiday は Day Of Year に対するコメントです。なお、休日を設定した場合、プライムタイムの開始時刻にかかわらず、ノンプライムタイムの扱いとなります。

• ユーザマップファイル

アカウンティングサーバでは、NQSV のユーザ権限に応じて各種データの設定、参照ができます。このため、アカウンティングサーバをバッチサーバと異なるホスト上に配置する場合は、アカウンティングサーバホストにおいて、バッチサーバと同じアカウントが利用できるように設定の上、バッチサーバ上のユーザマップファイルを/etc/opt/nec/nqsv/nqs_user.map にコピーしてください。

ユーザマップファイルの形式に関しては、NEC Network Queuing System V (NQSV)利用の手引[管理編]を参照してください。

• アカウントコードファイル

NQSV のアカウンティング・予算管理機能において、アカウントコードに対する予算管理を行う場合は、各ユーザが利用可能なアカウントコードを/etc/opt/nec/nqsv/acctcode ファイルに設定してください。

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

各行のフォーマットは、

アカウントコード名:ユーザ名,ユーザ名,・・・

のように、アカウントコード名と、そのアカウントコードを利用可能なユーザのユーザ名を設定します。コロン、カンマの前後に空白文字を挿入しないでください。アカウントコード名の最大長は15バイトまでです。

- 自動起動設定

システムの起動時にアカウントिंगサーバデーモンを自動的に起動するように設定する場合は、以下のように設定してください。

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

2.2.2 アカウンティングモニタ

- 設定ファイルの編集

バッチサーバホスト、および JobManipulator ホスト上で、アカウンティングモニタデーモン起動のために、設定ファイル(/etc/opt/nec/nqsv/acmd.conf) に次の設定をしてください。

項目	意味	内容
LOG_FACILITY	syslogのファシリティ	syslogのファシリティを指定します。デフォルトはLOG_LOCAL0です。
CONNECT_INTERVAL	アカウントデータ送信間隔 [秒]	アカウントデータの送信間隔を秒単位で指定します。1から86400(1日)まで指定できます。デフォルトは30秒です。
TURN_SIZE	アカウントデータ退避の閾値 [MB]	アカウントデータを退避するサイズの閾値をMB単位で指定します。0から2047まで指定可能で、デフォルトは、500MBです。
TURN_SAVE	アカウントデータを退避する際の保存世代数	アカウントデータを退避する際に保存するアカウントファイルの最大世代数を指定します。0から99まで指定可能で、デフォルト値は10です。
RACCT_FILE	リクエストアカウントファイル	リクエストアカウントファイルの位置を設定します。デフォルトでは /var/opt/nec/nqsv/bsv/account/reqacct が設定されています。
JOBACCT_FILE	ジョブアカウントファイル	ジョブアカウントファイルの位置を設定します。デフォルトでは /var/opt/nec/nqsv/acm/jacct/jobacct が設定されています。変更可能なのはディレクトリ名部分です。ファイル名(jobacct)は変更しないでください。変更すると誤動作します。
RSVACCT_FILE	予約アカウントファイル	予約アカウントファイルの位置を設定します。デフォルトでは /var/opt/nec/nqsv/acm/rsvacct/rsvacct が設定されています。
LOCK_DIR	ロックファイル用ディレクトリ	ロックファイルを格納するディレクトリを指定します。デフォルトは、/var/opt/nec/nqsv/acm/です。
SERVER_PORT_FOR_ACCT	アカウントデータ送信用ポート番号	アカウンティングサーバデーモンとの通信に使用するポート番号を設定します。デフォルトでは6542 が設定されています。アカウンティングサーバの設定ファイルでRECV_PORT_FOR_ACCTを変更していなければ設定は不要です。
ACCTFILESIZE_LIMIT	アカウントファイルの上限サイズ	リクエストアカウントファイル、ジョブアカウントファイル、予約アカウントファイルの上限サイズをMB単位で設定します。デフォルトでは131072MB(128GB)が設定されています。このサイズを超えるとアカウントファイルの内容はアカウンティングサーバに転送されません。

- 自動起動設定

システムの起動時にアカウントिंगモニタデーモンを自動的に起動するように設定する場合は、以下のように設定してください。

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

2.2.3 バッチサーバ

NQSV のアカウントिंग・予算管理機能を利用するには、バッチサーバに対して以下の環境設定が必要です。

- リクエストアカウントの設定

NQSV のアカウントिंग・予算管理機能では、NQSV のリクエストアカウント情報の集計が可能です。リクエストアカウントの集計、予算管理を行うためには、NQSV のバッチサーバがリクエストアカウントファイルの出力を行うよう設定する必要があります。

リクエストアカウントファイルの出力設定は、qmgr(1M)によって次のように設定してください。

```
# /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
```

※アカウントिंग・予算管理機能運用中にリクエストアカウントの出力をオフにしないでください。

- ジョブアカウントの設定

NQSV のアカウントिंग・予算管理機能では、バッチサーバが出力するジョブアカウントファイルからアカウントिंग情報の収集を行います。

バッチサーバによるジョブアカウントファイルは、既定値では、/var/opt/nec/nqsv/acm/jacct ディレクトリの下に jobacct というファイルとして出力されます。

この出力先ディレクトリを変更する必要がある場合は、変更後の出力先ディレクトリを作成の上、以下のように qmgr コマンドにより設定を行ってください。

```
# 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
```

上記のように設定することで、ジョブアカウントファイルの出力先ディレクトリを /usr/share/nqs/jacct に変更することができます。

- アカウンティングサーバの設定

バッチサーバ、および、JobManipulator が、アカウンティングサーバと連携して予算超過の判断を行えるようにするため、バッチサーバにアカウンティングサーバの情報を設定する必要があります。

アカウンティングサーバ情報の設定は、まず、アカウンティングサーバのホスト名を qmgr コマンドにより以下のように設定します。

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

※*acct-server*にはアカウンティングサーバのホスト名を指定してください。

また、アカウンティングサーバの設定の際に、予算要求受付用ポート番号を既定値から変更した場合は、さらに、以下のようにアカウンティングサーバの予算要求受付用ポート番号を設定してください。

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

設定するポート番号は、アカウンティングサーバの予算要求受付用ポート番号と同じ番号を設定してください。

- JobManipulatorのリソース予約区間に対する課金の設定

NQSV のアカウンティング・予算管理機能において、JobManipulator により生成されるリソース予約区間に対するアカウンティング情報の集計、予算管理を行う場合は、バッチサーバで、qmgr コマンドにより以下のように、リソース予約区間のアカウンティング機能を ON にしてください。

```
# /opt/nec/nqsv/bin/qmgr -P m
```

```
Mgr: set batch_server reservation_acct = ON
Set Reservation Accounting
Mgr: exit
```

本設定により、リソース予約区間のアカウンティング機能を ON にすることによって、JobManipulator でリソース予約区間を生成する際の予算のチェック、生成されたリソース予約区間に対する予約アカウントファイルの集計、および、リソース予約区間に対する課金の計算が行えるようになります。

ただし、リソース予約区間に対する課金が有効になるのは、キューおよびホスト数を指定して生成するリソース予約区間に対してのみで、ホスト指定でのリソース予約区間に対する課金はいりません。

また、リソース予約区間に対する課金機能を ON に設定した場合、JobManipulator により、予約アカウントファイルが出力され、アカウンティングサーバにより集計を行うことができます。

この予約アカウント情報は、既定値では JobManipulator の実行ホスト上で、`/var/opt/nec/nqsv/acm/rsvacct/rsvacct` ファイルに出力されます。

アカウンティングモニタの設定で、予約アカウントファイルの設定を変更した場合は、バッチサーバにより、以下のように予約アカウントファイルの出力ファイルの設定を合わせて変更してください。

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

上記のように設定することで、予約アカウントファイルを `/usr/share/nqs/rsvacct` に変更することができます。この設定はアカウンティングモニタの設定ファイル内の `RSVACCT_FILE` に合わせて設定する必要があります。

上記の、JobManipulator のリソース予約区間に対する課金の設定は、JobManipulator を起動していない状態で実行してください。

- 予算管理機能

NQSV のアカウンティング・予算管理機能では、アカウンティングサーバの予算管理機能と連携して、リクエストの投入時、および、リソース予約区間の生成時に、予算超過をチェックすることができます。

NQSV による予算超過チェックの機能を有効にする場合は、アカウンティングサーバの予算管理機能の設定も ON に設定して運用する必要があります(アカウンティングサーバの設定については 2.2.1 を参照してください)。

NQSV による予算超過チェックの機能を有効にするには、以下のように `qmgr` コマンドで、`"set batch_server nqs_budget_chk"`サブコマンドにより設定を行う必要があります。

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

上記のように、`"set batch_server nqs_budget_chk = ON"`で予算超過チェックの機能を有効にした場合、リクエストの投入、および、リソース予約区間の生成に対して予算超過チェックを行います。

予算超過チェックの機能を有効にする際に、リクエストに対してのみ、または、リソース予約区間に対してのみ予算超過チェックを行う場合には、下記のように`"target = target"`の指定を追加してください。

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

例のように、`target`に対して`"request"`を指定した場合は、リクエストに対してのみ予算超過チェックを行う場合の指定になります。リソース予約区間に対してのみ予算超過チェックを行う場合は、`target`に対して`"reservation"`を指定してください。

リソース予約区間に対して予算超過チェックを行うように設定した場合、リソース予約区間の生成時に予算超過チェックを行い、リソース予約区間に対して課金の計算が行われます。

このとき、そのリソース予約区間を指定して投入したリクエストに対しては、予算超過チェックと課金の計算は実行されません。

ただし、予算超過チェックの設定を変更した場合、リソース予約区間を指定して投入したリクエストに対する予算超過チェックと課金の計算は、リソース予約区間の生成時の設定とは関係なく、投入

時点の設定にしたがって処理されます。

NQSV による予算超過チェック機能の既定値は、OFF です。課金の運用において、アカウントイングサーバがサポートしている従量課金の機能を使用しない場合(定額課金の運用を行う場合など)は、本設定を OFF のまま運用してください。

- 設定内容の表示

NQSVのアカウントイング・予算管理機能に関するバッチサーバの設定内容は、qstatコマンドの -Bf オプションにより、確認することができます。

表示例

```
$ /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
  :
  :
```

表示箇所	表示内容
Accounting Server Host Name Accounting Server Port Number	アカウンティングサーバのホスト名とポート番号が表示されます。
Jacct directory	ジョブアカウントファイルの出力ディレクトリが表示されます。
Budget function	予算管理機能のON/OFFの設定が表示されます。
Request Accounting	リクエストアカウントファイルの出力のON/OFFの設定が表示されます。
Request Accounting File Path	リクエストアカウントファイルのパス名が表示されます。
Reservation Accounting	予約アカウントのON/OFFの設定が出力されます。
Reservation Accounting File	予約アカウントファイルのパス名が表示されます。

2.2.4 AUI

クライアントホスト上で、アカウンティングサーバからのアカウンティング情報、予算情報の参照を行うためのコマンドを有効にするためには、次の設定をしてください。

- scacctユーザのssh接続

scacct ユーザを作成し、この scacct ユーザに対して、クライアントホストからアカウンティングサーバホストに対して、パスワードなしで ssh の接続が行えるよう設定しておく必要があります。

- (1) クライアントホストでscacctユーザを作成します。

```
# 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
```

- (2) scacctユーザの公開鍵を作成します。

```
# su - scacct

% ssh-keygen

※3回Enterのみを入力し、空のパスフレーズを設定します。
```

- (3) /opt/nec/nqsv/scacct/.home/.sshディレクトリとid_rsa、id_rsa.pubファイルが作成されていることを確認します。

```
$ ls -l /opt/nec/nqsv/scacct/.home/.ssh
-rw----- 1 scacct scacct 1675 2月 19 14:27 2014 id_rsa
-rw-r--r-- 1 scacct scacct 419 2月 19 14:27 2014 id_rsa.pub
```

- (4) scacctユーザの公開鍵をアカウントिंगサーバに配置します。

```
$ 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 にはアカウントिंगサーバのホスト名を指定してください。

- (5) スーパーユーザでアカウントिंगサーバにログインし、公開鍵ファイルを設定します。

```
# 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
```

- (6) scacctユーザでアカウントティングサーバにパスワードなしでログインできることを確認します。

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

- sudoの設定

クライアントホストにてスーパーユーザで visudo コマンドを起動し、sudo の設定に以下の記述を追加してください。

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

※group_name にはコマンドをアカウントティングサーバ上で利用するユーザのグループを指定します。

2.3 起動・終了

環境の構築が終わったら、各デーモンを起動します。各デーモンの起動、終了方法は下記の通りです。

2.3.1 アカウントティングサーバ

アカウントティングサーバの起動、終了方法は下記の通りです。

- 起動

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

- 終了

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

2.3.2 アカウンティングモニタ

アカウンティングモニタの起動、終了方法は下記の通りです。

- 起動

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

- 終了

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

第3章 アカウンティング

NQSV アカウンティング・予算管理システムで収集されたアカウントデータは、NQSV アカウンティング・予算管理サーバ付属のコマンド群で参照することが可能です。本章ではそれらのコマンドの説明をします。

また、一部のコマンドは「[2.2.4 AUI](#)」の設定を行うことで、アカウンティングサーバマシン以外のフロントエンドマシンなどから実行、参照することが可能です。詳細は `racctreq(1)/racctjob(1)/racctrsv(1)/rbudgetcheck(1)` のオンラインマニュアルを参照してください。

3.1 リクエストアカウントデータの参照

リクエストアカウントデータは `scacctreq(1)` コマンドで参照できます。リクエストアカウントの各項目は、下表のとおりです。

項目名	内容	オプション			ソートオプション(-o)で指定する文字列
		表示	非表示	検索	
REQUEST ID	リクエストID			-I	REQUEST_ID
REQUEST NAME	リクエスト名	--long-request -name (*10)		-n	REQUEST_NAME
USER NAME	投入ユーザ名			-u	USER_NAME
GROUP NAME	グループ名	-Y		-g	GROUP_NAME
ACCOUNT CODE	アカウントコード	-X		-x	ACCOUNT_CODE
QUEUE NAME	投入キュー名			-Q	QUEUE_NAME
QUEUED TIME	投入時刻		-2	-{s e}	QUEUED_TIME
	投入日付/時刻	-2			
START TIME	実行開始時刻		-2	-	START_TIME
	実行開始日付/時刻	-2		{S b}	
END TIME	実行終了時刻		-2	-	END_TIME (*9)
	実行終了日付/時刻	-2		{A E}	
CPU (SECS)	CPU消費時間（システム+ユーザ） [秒]		-t		CPU
CPU (SECS)	CPU消費時間（システ	-t			CPU_SYS

SYS	△) [秒]				
CPU (SECS) USER	CPU消費時間 (ユーザ) [秒]	-t			CPU_USER
REAL (SECS)	経過時間 [秒] (*1)				REAL
REQUEST PRTY	リクエスト優先度	-p			REQUEST_PRTY
NICE	ナイス値	-p			NICE
REQELAPS TIME(S)	経過時間制限値 [秒]	-T			REQELAPS_TIME
REQCPU TIME(S)	ジョブ毎の要求CPU時間[秒]。ハイブリッドリクエストの場合、リクエストの各ジョブの要求 CPU 時間の平均値 [秒]	-q			REQCPU_TIME
REQCPU NUM	ジョブ毎の要求 CPU 数。ハイブリッドリクエストの場合、リクエストの各ジョブの要求 CPU 数の平均値	-q			REQCPU_NUM
REQMEM SIZE(K)	ジョブ毎の要求メモリ量[KB]。ハイブリッドリクエストの場合、リクエストの各ジョブの要求メモリ量の平均値 [KB]	-q			REQMEM_SIZE
REQGPU NUM	ジョブ毎の要求GPU数。ハイブリッドリクエストの場合、リクエストの各ジョブの要求 GPU数の平均値	-q			REQGPU_NUM
STATUS	終了ステータス (簡易表示)				EXIT_STAT
EXIT STAT	終了ステータス	-h			
KCORE MIN	延べメモリ使用量[KB*分] (*8)	-k			KCORE_MIN
MEAN SIZE(K)	平均メモリ使用量 [KB]	-m			MEAN_SIZE
MAXMEM SIZE(K)	最大メモリ使用量 [KB]	-M			MAXMEM_SIZE
V KCORE MIN	延べ仮想メモリ使用量 [KB*分] (*8)	--vmem			V_KCORE_MIN
V MEAN SIZE(K)	平均仮想メモリ使用量 [KB]	--vmem			V_MEAN_SIZE
V MAXMEM SIZE(K)	最大仮想メモリ使用量 [KB]	--vmem			V_MAXMEM_SIZE

QUE TYPE	キューのタイプ	-K			QUE_TYPE
EHOST NUM	実行ホスト数	-C			EHOST_NUM
JOBS	ジョブ数	-C			NUM_JOBS
SUBREQ	サブリクエスト数(パラ メトリックリクエスト のみ)	-C			NUM_SUBREQ
RERUN COUNT	RERUN回数	-r			RERUN_COUNT
PRERUN COUNT	パラメトリックリクエ ストのRERUN回数	-r			PRERUN_COUNT
TEMPLATE NAME	テンプレート名	-L		-l	TEMPLATE_NAME
<i>crname</i> (*2)	カスタムリソースの宣 言量。カスタムリソー スの消費単位がrequest の場合は、リクエスト 全体の宣言量、消費単 位がjobの場合は、ジョ ブ毎の宣言量	-F		-c	CR_ <i>crname</i>
	指定したカスタムリソ ースの宣言量	-G			
<i>crname</i> TOTAL	カスタムリソースのの べ使用量	-O			(なし)
	指定したカスタムリソ ースののべ使用量	-j			(なし)
<i>crname</i> MEAN	カスタムリソースの平 均使用量	-O			(なし)
	指定したカスタムリソ ースの平均使用量	-j			(なし)
<i>crname</i> MAX	カスタムリソースの最 大使用量	-O			(なし)
	指定したカスタムリソ ースの最大使用量	-j			(なし)
UNIT(*4)	カスタムリソースの単 位	-F -G -O -j			(なし)
REQVE NUM(*5)	要求VEノード数 (*3)	-V			REQVE_NUM
RSVVE NUM	確保VEノード数 (*3)	-V			RSVVE_NUM
VE CPU(SECS)	VEのCPU消費時間 (ユ ーザ) の合計[秒] (*3)	-V --veps (*6)			-Vで表示した場合はVE_CPU --veps で 表 示 し た 場 合 は VE_CPU_PS
VE KCORE MIN	VEの延べ使用メモリ [KB・分] (*3)	-V --veps (*6)			-V で 表 示 し た 場 合 は VE_KCORE_MIN --veps で 表 示 し た 場 合 は

					VE_KCORE_MIN_PS
VE MEAN SIZE(K)	VEの平均メモリ使用量 [KB] (*3)	-V --veps (*6)			-V で 表 示 し た 場 合 は VE_MEAN_SIZE --veps で 表 示 し た 場 合 は VE_MEAN_SIZE_PS
VE MAXMEM SIZE(K)	VEの最大メモリ使用量 [KB] (*3)	-V			VE_MAXMEM_SIZE
REQCPUTIM E SUM(S)	リクエストの各ジョブ の要求CPU時間の合計 [秒]	-z			REQCPUTIM_SUM
REQCPUNUM SUM	リクエストの各ジョブ の要求CPU数の合計	-z			REQCPUNUM_SUM
REQMEMSIZ E SUM(M)	リクエストの各ジョブ の要求メモリ量の合計 [MB]	-z			REQMEMSIZE_SUM
REQGPUNU M SUM	リクエストの各ジョブ の要求GPU数の合計	-z			REQGPUNUM_SUM
HW FAILURE	HW障害フラグ	--hw- failure			HW_FAILURE
VE TIME SLICE	VEのタイムスライス値 (*7)	--veps- perform			VE_TIME_SLICE
VE MFLOPS	VEのMFLOPS値(*7)	--veps- perform			VE_MFLOPS
VE MFLOPS (REAL)	VEのMFLOPS(REAL) 値(*7)	--veps- perform			VE_MFLOPS_REAL
VE FPEC(K)	VEの浮動小数点データ 実行要素数[K] (*7)	--veps- prog			VE_FPEC
VE SYSTEM CALL	VEのシステムコール回 数(*7)	--veps- prog			VE_SYSTEM_CALL
VE-VH DATA TRANS	VE-VH間データ転送量 [KB] (*7)	--veps- prog			VE_VH_DATA_TRANS
NUMBER OF VE PROCESS	VEの実行プロセス数 (*7)	--veps- prog			VE_PROC_NUM
INSTRCT(K)	VEの命令実行数(K) (*7)	--veps- inst			VE_INSTRCT
VEC-INST(K)	VEのベクトル命令実行 数(K) (*7)	--veps- inst			VEC_INST
VEC-ELMT(K)	VEのベクトル要素数(K) (*7)	--veps- inst			VEC_ELMT
VEC-EXE (SECS)	VEのベクトル命令実行 時間[秒] (*7)	--veps- inst			VEC_EXE
VEC-INST LOAD(K)	VEのベクトルロード実 行要素数(K) (*7)	--veps- inst			VEC_INST_LOAD
VE L1 CACHE	VEのL1キャッシュミス	--veps-			VE_L1_CACHE_MISS_TIME

MISS TIME (SECS)	時間[秒] (*7)	time			
VE CPU PORT CONFLICT (SECS)	VEのCPUポート競合時間[秒]	--veps- time			VE_CPU_PORT_CONFLICT
VEC ARITH EXEC TIME (SECS)	VEのベクトル演算実行時間[秒] (*7)	--veps- time			VE_ARITH_EXEC_TIME
VEC LOAD EXEC TIME (SECS)	VEのベクトルロード実行時間[秒] (*7)	--veps- time			VEC_LOAD_EXEC_TIME
VE POWER THROTTLING (SECS)	VEの電量要因によるHW停止時間[秒] (*7)	--veps- time			VE_PORT_ THROTTLING
VE THERMAL THROTTLING (SECS)	VEの温度要因によるHW停止時間[秒] (*7)	--veps- time			VE_THERMAL_THROTTLIN G
VE MAX NTASK	VEの最大タスク数(*7)	--veps- task			VE_MAX_NTASK
VEC- OPERAND RATIO	VEのベクトル化率(*7)	--veps- opratio			VEC_OP_RATIO

*1 リクエストアカウントで表示するREALはリクエストのRUNNING状態の時間です。

*2 *crname*はカスタムリソース名です。

*3 これらの項目は実行ホストがSX-Aurora TSUBASAの場合のみ有効です。

*4 UNITはカスタムリソースの宣言量または使用量の単位です。カスタムリソースの宣言量と使用量を表示するときに同時に表示されます。

*5 REQVEとRSVVEは、ジョブ投入時に--venodeを指定してVEを要求した場合に値が異なる可能性があります。例えば、--venode=11、--venum-lhost=4としてVEを要求した場合、REQVEは11、RSVVEは12となります。-bと--venum-lhostを使用してVEを要求した場合、または、キューの既定VE搭載数が1（既定値）でかつ--venodeのみを指定してVEを要求した場合は、REQVEとRSVVEの値は同じです。

*6 -Vで表示する値は、ps(VE)コマンドで定期的に採取した値です。-Vで表示するVE MEAN SIZE(K)は、リクエストに紐づく全ジョブのプロセスが使用したVEのメモリ量の平均値です。--vepsで表示する値は、VEOSから出力されるプロセスアカウントの値を基に計算した値です。--vepsで表示するVE MEAN SIZE(K)は、リクエストに紐づく全ジョブのプロセスの、1プロセスが使用したVEのメモリ量の平均値です。プロセスアカウントの利用方法については「[3.6 VEプロセスアカウント](#)」節を参照してください。

*7 これらの項目はVEOSから出力されるプロセスアカウントの値を基に計算した値です。プロセスアカウントの利用方法については「[3.6 VEプロセスアカウント](#)」節を参照してください。

*8 KCORE MINとV KCORE MINは、リクエストの各ジョブのKCORE、V KCORE MINの合計です。ジョブ単位のKCORE MINは、ジョブの実行ホストの平均メモリ使用量×CPU消費時間(システム+ユーザ)の積算値になります。ジョブ単位のV KCORE MINは、ジョブの平均仮想メモリ使用量×CPU消費時間(システム+ユーザ)の積算値になります。

*9 ジョブ終了時のノードヘルスチェックに時間を要した場合、リクエストアカウントのEND TIMEは当該時間を含めた終了時刻となります。

*10 REQUEST NAMEは標準でREQUEST NAMEの表示カラムの幅で表示されます。--long-request-nameオプションを指定することで最大63文字のリクエスト名が表示できます。

3.1.1 scacctreq の実行例

scacctreq(1)コマンドは NQSV アカウンティング・予算管理で収集されたリクエストアカウントデータを検索、表示するコマンドです。

オプションを指定せずに実行した場合は、下記のような結果が表示されます。

```
$ scacctreq
```

REQUEST ID	REQUEST NAME	USER NAME	QUEUE NAME	QUEUED TIME	START TIME	END TIME	CPU (SECS)	REAL (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

scacctreq(1)コマンドは、実行時の NQSV の権限に応じて、アカウンティングサーバ上のリクエスト DB 内のデータのうち、以下に示す範囲の情報が表示されます。実行時の権限は、-P オプションにより指定します。既定値は、一般ユーザ権限です。

権限	参照可能な情報
一般ユーザ権限	自分が投入したリクエスト情報
特別ユーザ権限	
グループ管理者	自分が管理しているグループのリクエスト情報
操作員権限	全てのユーザのリクエスト情報
管理者権限	

蓄積しているデータが多い場合は、表示に時間がかかる場合があります。そのような場合は、さまざまな抽出オプションを指定して表示するデータ量を削減してください。

例えば、下記のようなオプションを指定して実行すると、ユーザ名で絞り込んだデータが表示されます。

```
$ scacctreq -u user01
```

REQUEST ID	REQUEST NAME	USER NAME	QUEUE NAME	QUEUED TIME	START TIME	END TIME	CPU (SECS)	REAL (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

また、-s, -e オプションなどを併用することで、日時による範囲指定が可能です。

NQSV アカウンティング・予算管理システムでは、上記の例で表示している項目の他にも、様々なデータを保持しています。下記にグループ名(-Y)、キュータイプ(-K)を表示する例を示します。

```
$ scacctreq -YK
```

REQUEST ID	REQUEST NAME	USER NAME	GROUP NAME	QUEUE NAME	QUEUED TIME	START TIME	END TIME	CPU (SECS)	REAL (SECS)	QUE 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	j.m. 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

※詳細は、scacctreq(1) のオンラインマニュアルを参照してください。

さらに、クライアントホストから、racctreq(1)コマンドによりアカウンティングサーバ上のリクエストアカウントデータを参照することもできます。詳細は、racctreq(1)のマニュアルページを参照してください。

3.2 ジョブアカウントデータの参照

ジョブアカウントデータは `scacctjob(1)` コマンドで参照できます。ジョブアカウントの各項目は、下表のとおりです。

項目名	内容	オプション			ソートオプションで 指定する文字列
		表示	非表示	検索	
JOB ID	ジョブID				JOB_ID
REQUEST ID	リクエストID			-I	REQUEST_ID
REQUEST NAME	リクエスト名	--long-request-name (*11)		-n	NAME
USER NAME	投入ユーザ名			-u	USER_NAME
GROUP NAME	グループ名	-Y		-g	GROUP_NAME
ACCOUNT CODE	アカウントコード	-X		-x	ACCOUNT_CODE
HOST NAME	実行ホスト名			-l	HOST_NAME
QUEUE NAME	投入キュー名			-Q	QUEUE_NAME
QUEUED TIME	投入時刻		-2	-{s e}	QUEUED_TIME
	投入日付/時刻	-2			
START TIME	実行開始時刻		-2	-{S b}	START_TIME
	実行開始日付/時刻	-2			
END TIME	実行終了時刻		-2	-{E A}	END_TIME(*10)
	実行終了日付/時刻	-2			
CPU (SECS)	CPU消費時間 (システム+ユーザ) [秒]		-t		CPU
CPU (SECS) SYS	CPU消費時間 (システム) [秒]	-t			CPU_SYS
CPU (SECS) USER	CPU消費時間 (ユーザ) [秒]	-t			CPU_USER
REAL (SECS)	経過時間 [秒] (*1)				REAL
REQUEST PRY	リクエスト優先度	-p			REQUEST_PRTY
NICE	ナイス値	-p			NICE
REQELAPS TIME(S)	経過時間制限値 [秒]	-T			REQELAPS_TIME
REQCPU TIME(S)	要求CPU時間 [秒]	-q			REQCPU_TIME
REQCPU NUM	要求CPU数	-q			REQCPU_NUM
REQMEM SIZE(K)	要求メモリ量 [KB]	-q			REQMEM_SIZE
REQGPU NUM	要求GPU数	-G			REQGPU_NUM
EXIT STAT	終了ステータス	-h			EXIT_STAT
KCORE MIN	延べメモリ使用量 [KB*分]	-k			KCORE_MIN

	(*8)				
MEAN SIZE(K)	平均メモリ使用量 [KB]	-m			MEAN_SIZE
MAXMEM SIZE(K)	最大メモリ使用量 [KB] (*9)	-M			MAXMEM_SIZE
V KCORE MIN	延べ仮想メモリ使用量 [KB*分] (*8)	--vmem			V_KCORE_MIN
V MEAN SIZE(K)	平均仮想メモリ使用量 [KB]	--vmem			V_MEAN_SIZE
V MAXMEM SIZE(K)	最大仮想メモリ使用量 [KB]	--vmem			V_MAXMEM_SIZE
QUE TYPE	キューのタイプ	-K			QUE_TYPE
WAIT TIME(SEC)	待ち時間（実行開始予定時刻 と実行開始時刻の差）	-w			WAIT_TIME
REQVE NUM	ジョブの割当VEノード数 (*2)	-V			REQVE_NUM
RSVVE NUM	ジョブの確保VEノード数 (*2)	-V			RSVVE_NUM
VE CPU(SECS)	VEのCPU消費時間（ユーザ） の合計[秒] (*2)	-V --veps (*6)			-Vで表示した場合は VE_CPU --vepsで表示した場合は VE_CPU_PS
VE KCORE MIN	VEの延べ使用メモリ [KB・ 分] (*2)	-V --veps (*6)			-Vで表示した場合は VE_KCORE_MIN --vepsで表示した場合は VE_KCORE_MIN_PS
VE MEAN SIZE(K)	VEの平均メモリ使用量 [KB] (*2)	-V --veps (*6)			-Vで表示した場合は VE_MEAN_SIZE --vepsで表示した場合は VE_MEAN_SIZE_PS
VE MAXMEM SIZE(K)	VEの最大メモリ使用量 [KB] (*2)	-V			VE_MAXMEM_SIZE
VE REQ NODELIST	ジョブの割当VEノード番号 のリスト (*2)	-Z			(なし)
VE USE NODELIST	ジョブの使用VEノード番号 のリスト (*2)	-Z			(なし)
VE RSV NODELIST	ジョブの確保VEノード番号 のリスト (*2)	-Z			(なし)
RESERVATION ID	リソース予約区間ID	-r			RES_ID

<i>crname</i> (*3)	カスタムリソースの宣言量	-F			(なし)
	指定したカスタムリソースの宣言量	-B			(なし)
<i>crname</i> TOTAL	カスタムリソースののべ使用量	-O			(なし)
	指定したカスタムリソースののべ使用量	-j			(なし)
<i>crname</i> MEAN	カスタムリソースの平均使用量	-O			(なし)
	指定したカスタムリソースの平均使用量	-j			(なし)
<i>crname</i> MAX	カスタムリソースの最大使用量	-O			(なし)
	指定したカスタムリソースの最大使用量	-j			(なし)
UNIT(*4)	カスタムリソースの単位	-F -B -O -j			(なし)
HW_FAILURE (*5)	HW障害フラグ	--hw-failure			HW_FAILURE
VE TIME SLICE	VEのタイムスライス値(*7)	--veps-perform			VE_TIME_SLICE
VE MFLOPS	VEのMFLOPS値(*7)	--veps-perform			VE_MFLOPS
VE MFLOPS (REAL)	VEのMFLOPS(REAL)値(*7)	--veps-perform			VE_MFLOPS_REAL
VE FPEC(K)	VEの浮動小数点データ実行要素数[K] (*7)	--veps-prog			VE_FPEC
VE SYSTEM CALL	VEのシステムコール回数(*7)	--veps-prog			VE_SYSTEM_CALL
VE-VH DATA TRANS	VE-VH間データ転送量[KB] (*7)	--veps-prog			VE_VH_DATA_TRANS
NUMBER OF VE PROCESS	VEの実行プロセス数(*7)	--veps-prog			VE_PROC_NUM
INSTRCT(K)	VEの命令実行数(K) (*7)	--veps-inst			VE_INSTRCT
VEC-INST(K)	VEのベクトル命令実行数(K) (*7)	--veps-inst			VEC_INST
VEC-ELMT(K)	VEのベクトル要素数(K) (*7)	--veps-inst			VEC_ELMT
VEC-EXE (SECS)	VEのベクトル命令実行時間[秒] (*7)	--veps-inst			VEC_EXE
VEC-INST LOAD(K)	VEのベクトルロード実行要素数(K) (*7)	--veps-inst			VEC_INST_LOAD
VE L1 CACHE	VEのL1キャッシュミス時間	--veps-			VE_L1_CACHE_MI

MISS TIME (SECS)	[秒] (*7)	time			SS_TIME
VE CPU PORT CONFLICT (SECS)	VEのCPUポート競合時間 [秒] (*7)	--veps- time			VE_CPU_PORT_CO NFLICT
VEC ARITH EXEC TIME (SECS)	VEのベクトル演算実行時間 [秒] (*7)	--veps- time			VE_ARITH_EXEC_ TIME
VEC LOAD EXEC TIME(SECS)	VEのベクトルロード実行時 間[秒] (*7)	--veps- time			VEC_LOAD_EXEC_ TIME
VE POWER THROTTLING (SECS)	VEの電量要因によるHW停 止時間[秒] (*7)	--veps- time			VE_PORT_THROT TLING
VE THERMAL THROTTLING (SECS)	VEの温度要因によるHW停 止時間[秒] (*7)	--veps- time			VE_THERMAL_TH ROTTLING
VE MAX NTASK	VEの最大タスク数(*7)	--veps- task			VE_MAX_NTASK
VEC-OPERAND RATIO	VEのベクトル化率(*7)	--veps- opratio			VEC_OP_RATIO
NUMA NODE	VEのNUMAノード番号(*7)	--veps- numanod e			なし

*1 ジョブアカウントで表示するREALは実行ホストにおいてジョブのプロセスが起動してから終了するまでの時間です。この時間は、scacctreq(1)で表示するREALは下記の理由で誤差が生じる場合があります。

マルチノードMPIリクエストの場合、リクエストがPRE-RUNNING状態においてすべてのスレーブジョブが起動してからマスタージョブが起動します。マスタージョブの起動をもってリクエストはRUNNING状態となります。マスタージョブが終了した時点のリクエストのRUNNING状態の終了とします。これにより、リクエストがRUNNING状態として扱われる区間（scacctreq(1)のREAL）とジョブのプロセスが存在する区間（scacctjob(1)のREAL）が異なる可能性が生じます。

また、ジョブのREALにはジョブのノードヘルスチェックにかかった時間を含めません。マルチノードジョブ開始時のノードヘルスチェックは、スレーブジョブ→マスタージョブ、ジョブ終了時のノードヘルスチェックはマスタージョブ→スレーブジョブの順で実行されるため、スレーブジョブの開始時刻から終了時刻までの間にマスタージョブのヘルスチェック時間が含まれます。しかし、ジョブのREALにはマスタージョブのヘルスチェック時間は含まれません。

*2 これらの項目は実行ホストがSX-Aurora TSUBASAの場合のみ有効です。

*3 *crname*はカスタムリソース名になります。

*4 カスタムリソースの宣言量または使用量の単位です。カスタムリソースの宣言量と使用量を表示するときに同時に表示されます。

*5 HW障害フラグが1のジョブは、JOB IDが「-0001」、「END TIME」の日付と時刻は「????/??/?? ??:??:??」と表示されます。

*6 -Vで表示する値は、ps(VE)コマンドで定期的に採取した値です。-Vで表示するVE MEAN SIZE(K)は、ジョブとして起動した全てのプロセスが使用したVEのメモリ量の平均値です。--vepsで表示する値は、VEOSから出力されるプロセスアカウントの値を基に計算した値です。--vepsで表示するVE MEAN SIZE(K)は、ジョブとして起動したプロセスの、1プロセスが使用したVEのメモリ量の平均値です。プロ

セスアカウントの利用方法については「3.6 VEプロセスアカウント」節を参照してください。

*7 これらの項目はVEOSから出力されるプロセスアカウントの値を基に計算した値です。プロセスアカウントの利用方法については「3.6 VEプロセスアカウント」節を参照してください。

*8 KCORE MINは、実行ホストの平均メモリ使用量×CPU消費時間(システム+ユーザ)の積算値です。V KCORE MINは、平均仮想メモリ使用量×CPU消費時間(システム+ユーザ)の積算値です。CPU消費時間はいずれも分単位に換算しています。

*9 ジョブの各プロセス間においてメモリを共有している場合は、それらのプロセスが使用するメモリ量の最大値が実行ホストの実装メモリ量を超えることがあります。その場合は、本値も実行ホストの実装メモリを超えることがあります。

*10 ジョブ終了時のノードヘルスチェックに時間を要した場合でも、ジョブアカウントのEND TIMEは当該時間を含めない終了時刻となります。

*11 REQUEST NAMEは標準でREQUEST NAMEの表示コラムの幅で表示されます。--long-request-nameオプションを指定することで最大63文字のリクエスト名が表示できます。

3.2.1 scacctjobの実行例

scacctjob(1)コマンドは NQSV アカウンティング・予算管理で収集されたジョブアカウントデータを検索、表示するコマンドです。ジョブアカウント情報は非常に数が多いため、検索系のオプションを使用して、データを絞り込んで表示することをおすすめします。

オプションを指定せずに実行した場合には、下記のような結果が表示されます。

=====										
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

scacctjob(1)コマンドは、実行時の NQSV の権限に応じて、アカウンティングサーバ上のジョブ DB 内のデータのうち、以下に示す範囲の情報が表示されます。実行時の権限は、-P オプションにより指定します。既定値は、一般ユーザ権限です。

権限	参照可能な情報
一般ユーザ権限	自分が投入したリクエストのジョブ情報
特別ユーザ権限	
グループ管理者	自分が管理しているグループのリクエストのジョブ情報

操作員権限	全てのユーザのジョブ情報
管理者権限	

蓄積しているデータが多い場合は、表示に時間がかかる場合があります。そのような場合は、さまざまな抽出オプションを指定して表示するデータ量を削減してください。例えば、下記のようなオプションを指定して実行すると、ユーザ、および実行ホスト名で絞り込んだデータが表示されます。

\$ 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

また、-s, -e オプションなどを併用することで、日時による範囲指定が可能となります。

NQSV アカウンティング・予算管理システムでは、上記の例で表示している項目の他にも、様々なデータを保持しています。下記にグループ名(-Y)、キュータイプ(-K)を表示する例を示します。

\$ 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

※詳細は、scacctjob(1) のオンラインマニュアルを参照してください。

さらに、クライアントホストから、`racctjob(1)`コマンドによりアカウンティングサーバ上のジョブアカウントデータを参照することもできます。詳細は、`racctjob(1)`のマニュアルページを参照してください。

3.3 予約アカウントデータの参照

予約アカウントデータは `scacctrsv(1)` コマンドで参照できます。予約アカウントの各項目は、下表のとおりです。

項目名	内容	オプション			ソートオプションで指定する文字列
		表示	非表示	検索	
RESERVATION ID	リソース予約区間ID			-I	RES_ID
BATCH_SERVER HOST	バッチサーバのホスト名	-B		-h	BSV
USER NAME	予約作成ユーザ名	-U		-u	USER_NAME (現在、リソース予約においてアカウントコードはサポートしていません。)
GROUP NAME	グループ名	-Y		-g	GROUP_NAME
ACCOUNT CODE	アカウントコード	-X		-x	ACCOUNT_CODE (現在、リソース予約においてアカウントコードはサポートしていません。)
CREATE TIME	予約生成時刻(日付つき)	-C		-{s e}	CREATE_TIME
RESERVATION TIME START	予定開始時刻(日付つき)			- {S b}	RES_START_TIME
RESERVATION TIME END	予定終了時刻(日付つき)			- {A E}	RES_END_TIME
ACTUAL RESERVED TIME START	実開始時刻(日付つき)	-F		-f {S b}	START_TIME
ACTUAL RESERVED TIME END	実終了時刻(日付つき)	-F		-f {A E}	END_TIME
EHOST NUM	要求ホスト数				EHOST_NUM
CPU NUM	要求CPU数 (1ホスト当たり)				CPU_NUM
QUEUE	投入キュー名			-Q	QUEUE
STATUS	終了ステータス				STATUS
TEMPLATE	テンプレート名	-L		-l	TEMPLATE_NAME

NAME					
------	--	--	--	--	--

3.3.1 scacctrsv の実行例

scacctrsv(1)コマンドは NQSV アカウンティング・予算管理で収集された予約アカウントデータを検索、表示するコマンドです。オプションを指定せずに実行した場合は、下記のような結果が表示されます。

```
$ 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

scacctrsv(1)コマンドは、実行時の NQSV の権限に応じて、アカウンティングサーバ上の予約 DB 内のデータのうち、以下に示す範囲の情報が表示されます。実行時の権限は、-P オプションにより指定します。既定値は、一般ユーザ権限です。

権限	参照可能な情報
一般ユーザ権限	自分が利用可能な予約情報
特別ユーザ権限	
グループ管理者	自分が管理しているグループの予約情報
操作員権限	全ての予約情報
管理者権限	

蓄積しているデータが多い場合は、表示に時間がかかる場合があります。そのような場合は、さまざまな抽出オプションを指定して表示するデータ量を削減してください。

例えば、下記のようなオプションを指定して実行すると、キュー名で絞り込んだデータが表示されます。

```
$ scacctrsv -Q bq1
```

RES RESERVATION TIME				NODE	CPU		STATUS
ID	START	END		NUM	NUM	QUEUE	
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

また、-s, -e オプションなどを併用することで、日時による範囲指定が可能となります。

※詳細は、scacctrsv(1) のオンラインマニュアルを参照してください。

さらに、クライアントホストから、racctrsv(1)コマンドによりアカウンティングサーバ上の予約アカウントデータを参照することもできます。詳細は、NEC Network Queuing System V (NQS) V) 利用の手引 [リファレンス編] の racctrsv(1)を参照してください。

3.4 日報・月報データの参照

日報・月報データの参照は、scsumacct(1M)コマンドを使用します。scsumacct(1M)コマンドは、アカウンティングサーバに収集されたリクエストアカウントデータ、予約アカウントデータから、指定した日、または、指定した月に実行が終了したデータを集計して表示します。scsumacct(1M)コマンドの実行ができるのは、スーパーユーザのみです。

集計データとして表示される項目は以下の通りです。

項目名	内容	オプション		表示対象	
		表示	非表示	予約	リクエスト
USER	ユーザ名			○	○
GROUP	グループ名			○	○
ACCTCODE	アカウントコード			○	○
REQUESTS	リクエストの総数			—	○
JOBS	ジョブの総数			—	○
EHOST NUM	実行ホストの総数			○	○
ELAPSE(H)	ジョブのELAPSE時間の総和 [時間]			—	○
CPU(SECS) SYS	CPU消費時間 (システム) [秒]	-t		—	○
CPU(SECS) USER	CPU消費時間 (ユーザ) [秒]	-t		—	○
CPU(SECS)	CPU消費時間 (システム+ユーザ) [秒]		-t	—	○
MEM	のべメモリ使用量[KB×min]			—	○
RESERVES	予約の総数			○	—
RSVTIME	リソース予約区間の総時間 [時間]			○	—
USETIME	リソース予約区間の総使用時間 [時間]			○	—
ESTIMATE	予定課金額の総和	-b		○	○
FEE	実課金額の総和	-b		○	○

3.4.1 日報

scsumacct(1M)は、ユーザ、グループ、アカウントコード毎にリクエストアカウントデータ、予約アカウントデータを集計し、表示します。

オプションを指定せずに実行すると当日の日報が下記のように表示されます。表示される項目の内容については、scsumacct(1M)を参照してください。

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

USER	REQUESTS	JOB	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	JOB	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	JOB	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

当日以外の日報を表示したい場合は、-s を指定し、下記のように実行します。

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

MMは月、DDは日、YYは西暦を指定します。

そのほかにも、-t オプション(ユーザ/システム別 CPU 消費時間表示)、-b オプション(予定課金額と実課金額表示)が使用可能です。

※詳細は、sctsumacct(1M)のオンラインマニュアルを参照してください。

3.4.2 月報

月ごとの集計データを表示する場合には-S オプションを指定してください。sctsumacct(1M)は、日報出力時と同様に、ユーザ、グループ、アカウントコード毎にリクエストアカウントデータ、予約アカウントデータを集計し、表示します。

```
# sctsumacct -S MM[/YY]
```

MMは月、YYは西暦を指定します。

※詳細は、sctsumacct(1M)のオンラインマニュアルを参照してください。

3.5 アカウントデータの退避

NQSV アカウンティング・予算管理で収集する、アカウントデータは運用を続ける限り蓄積され続けます。データ量が増えすぎるとデータの表示などに時間がかかる場合もあるため、運用に必要なデータを残して、削除(退避)する必要があります。

3.5.1 アカウンティングサーバ

アカウンティングサーバでは `sceditacct(1M)` コマンドを使って各アカウントデータを削除(退避)し、カレントのデータベースのサイズを小さくします。実行形式は下記の通りです。

```
# /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*は月、*DD*は日、*YY*は西暦を指定します。

-j オプションはジョブアカウントデータを、-r オプションはリクエストアカウントデータを、-y オプションは予約アカウントデータを操作の対象にします。その他のオプションは削除(退避)するデータの条件を指定します。下記に退避する条件の一覧を示します。

オプション名	指定方法	削除(退避)されるデータ
u	ユーザID (ユーザ名)	指定したユーザID (ユーザ名) のデータ
g	グループID (グループ名)	指定したグループID (グループ名) のデータ
x	アカウントコード	指定したアカウントコードのデータ
s	[MM/DD[YY]]	指定した月日あるいはそれ以降に実行が終了したデータ
e	[MM/DD[YY]]	指定した月日あるいはそれ以前に実行が開始されたデータ
S	[MM/DD[YY]]	指定した月日あるいはそれ以降に実行が開始されたデータ
E	[MM/DD[YY]]	指定した月日あるいはそれ以前に実行が終了したデータ
h	ホスト名(IPアドレス)	指定したホスト名(IPアドレス)のデータを削除(退避)します。このオプションは ジョブアカウントのデータベースに対してのみ有効です。

リクエストアカウントデータは実行が終了するまでは開始時刻および終了時刻が 0 で DB に登録されています。上記の各オプション指定時は、そのような実行が終了していないデータは操作の対象外となります。

上記各オプションとともに-f オプションを指定すると、実行が終了していないデータも含めて、削除(退避)するデータの条件を判断します。-f オプションが指定されない場合は、実行が終了していないデータは操作の対象としません。なお、-f オプションは リクエストアカウントのデータベースに対してのみ有効です。

-f オプションはアカウント DB のデータを強制的に削除するモードです。使い方によってはアカウント DB のデータを不適切に削除してしまう恐れがあります。したがって、通常運用でのご利用は控えください。

実行が終了していないデータのうち、実行が開始前に削除されたリクエストアカウントデータを退避し、実行中のリクエストアカウントデータを残す場合は、次のオプションで条件を指定することで可能です。

オプション名	指定方法	削除(退避)されるデータ
q	[MM/DD[YY]]	指定した月日あるいはそれ以降に投入され、かつ実行が開始前に削除されたデータ
Q	[MM/DD[YY]]	指定した月日あるいはそれ以前に投入され、かつ実行が開始前に削除されたデータ

これらオプションは、u、g、x、s、e、S および E の各オプションと同時に指定することができます。s、e、S および E の各オプションと併用した場合は、q および Q オプションで指定した条件、または s、e、S、E オプションの条件のいずれかの条件を満たしたものが操作の対象となります。

たとえば、運用中にユーザ user1 が管理の対象からはずれた場合、下記のように対象となるデータを削除することが可能です。

```
# /opt/nec/nqsv/bin/sceditacct -r -u user1 (リクエストアカウントの場合)
# /opt/nec/nqsv/bin/sceditacct -j -u user1 (ジョブアカウントの場合)
# /opt/nec/nqsv/bin/sceditacct -y -u user1 (予約アカウントの場合)
```

また、-m オプションを指定することで、削除するデータを別ファイルに退避することが可能です。たとえば、月次処理などで前月のデータを退避し、当月のデータのみ有効とする場合には、下記のように実行します。ここでは、4/1 に 3/31 日以前のデータを退避する場合を想定しています。

- リクエストアカウント、ジョブアカウントの場合は -m オプションで退避先のディレクトリ名を指定します。

```
# /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
```

リクエストアカウントの場合は、/tmp/racct2019_03 配下に racct、racct_tally、racct_cr、racct_cr_tally ファイルが作成されます。実行が開始前に削除されたリクエストアカウントのデータも含めて退避する場合は、-Q オプションをつけてください。

ジョブアカウントの場合は、/tmp/jacct2019_03 配下に jacct、jacct_cr ファイルが作成されます。

- 予約アカウントの場合は -m オプションで退避先のファイル名を指定します。

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

cron に登録することで、自動的に月次処理を実行することが可能です。また、上記のバックアップファイルを参照するには、scacctreq(1)、scacctjob(1)、scacctrsv(1) コマンドで、下記のように実行します。

- リクエストアカウントの場合は-dオプションで退避先のディレクトリ名を指定します。

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

- ジョブアカウントの場合は--db-dirオプションで退避先のディレクトリ名を指定します。

```
# /opt/nec/nqsv/bin/scacctjob --db-dir /tmp/jacct2019_03
```

- 予約アカウントの場合は-dオプションで退避先のファイル名を指定します。

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

※詳細は、sceditacct(1M) のオンラインマニュアルを参照してください。

3.5.2 アカウンティングモニタ

アカウンティングモニタ上で処理されるリクエストアカウントファイル、ジョブアカウントファイルおよび予約アカウントファイルはアカウンティングサーバに送信した後も追記され、そのサイズは無制限に大きくなります。

モニタでは scturnacct(1M) コマンドを使ってリクエストアカウントファイル、ジョブアカウントファイル、予約アカウントファイルを退避し、世代管理を行います。

リクエストアカウントファイル、ジョブアカウントファイル、予約アカウントファイルは scturnacct(1M) コマンド実行時に、設定されたファイルサイズを超えていた場合にのみ退避されます。

また、世代管理する世代数に関しても、モニタの設定ファイル(/etc/opt/nec/nqsv/acmd.conf)にて設定可能です。設定する内容は下表の通りです。

設定項目	内容	デフォルト値
TURN_SIZE	コマンド実行時に退避を開始するファイルサイズ(MB単位)	500(MB)
TURN_SAVE	退避する世代数	10

scturnacct(1M) の実行方法は下記の通りです。

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

例えば、/var/opt/nec/nqsv/bsv/account/reqacct ファイルのサイズが設定ファイルの TURN_SIZE

以上の場合、以下のようにファイルをリネームします。(n は設定ファイルの TURN_SAVE)

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

各アカウントファイルのパスを設定ファイルによって変更していた場合、世代管理されるファイルのファイル名は、設定ファイルに設定されたファイル名.1 ～設定ファイルに設定されたファイル名.n となります。

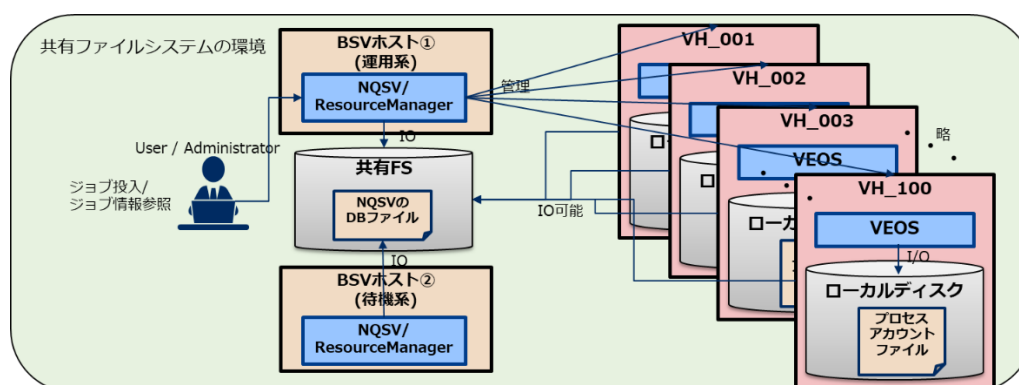
※詳細は、scturnacct(1M) のオンラインマニュアルを参照してください。

3.6 VEプロセスアカウント

VEOS の機能により VE 上で実行されたプロセスのアカウントを参照することができます。NQSV では、VEOS が出力したプロセスアカウントをジョブ単位およびリクエスト単位に集計します。scacctjob(1)/scacctreq(1)コマンドでそれらの値を参照することができます。VEOS のプロセスアカウントの設定方法については、「SX-Aurora TSUBASA インストレーションガイド」の「4.13 Process accounting の設定」節を参照してください。

3.6.1 利用可能な運用構成

本機能は、下図のような共有ファイルシステムを利用する環境で動作します。



本機能の利用の流れは以下の通りです。

1. VEOS のプロセスアカウントを設定し、VH のローカルディスク上にプロセスアカウントファイルを出力させます。
2. プロセスアカウントファイルを読み込んでジョブ単位でアカウント情報を集計するスクリプト（以降、プロセスアカウント集計スクリプト）を、ユーザ EXIT の POST-RUNNING 状態の実行スクリプトとして設定します。集計結果は CSV 形式の中間ファイル（以降、プロセスアカウント集計結果 CSV ファイル）に出力されます。ファイルの出力先は、アカウンティングサーバが参照可能な共有ファイルシステム上のディレクトリとしてください。
3. バッチサーバホストで、scacctpsm コマンドを cron に登録し、定期的に行わせてください。この処理により、2 で集計された CSV ファイルの内容をジョブアカウントおよびリクエストアカウント DB に登録します。
4. scacctjob(1)/scacctreq(1)コマンドでジョブ単位/リクエスト単位で集計した情報を参照します。

以降の節で詳細を説明します。

3.6.2 集計項目一覧

VE プロセスアカウントの下記の項目をジョブアカウントに集計します。項目名は VE プロセスア

カウントの構造体(VE10/VE20 : struct ve_acct_v15、VE30 : struct ve_acct_v16)のメンバー名です。

プロセスアカウントの 項目名	説明	ジョブアカウントの項目名
ac_utime	VE の CPU 消費時間 (ユーザ) の 合計[ticks]	VE CPU(SECS)
ac_total_mem	VE の延べ使用メモリ [KB・ticks]	VE KCORE MIN ※単位 : KB・分
ac_timeslice	VE のタイムスライス値[μs]	VE TIME SLICE
ac_fpec	VE の浮動小数点データ実行要素数	VE FPEC(K)
ac_syscall	VE のシステムコール回数	VE SYSTEM CALL
ac_transdata	VE-VH 間データ転送量[KB]	VE-VH DATA TRANS
ac_ex	VE の命令実行数	INSTRCT(K)
ac_vx	VE のベクトル命令実行数	VEC-INST(K)
ac_ve2	VE のベクトル要素数	VEC-ELMT(K)
ac_vecc	VE のベクトル命令実行時間[μs]	VEC-EXE(SECS)
ac_vlec	VE のベクトルロード実行要素数	VEC-INST LOAD(K)
ac_l1mcc	VE の L1 キャッシュミス時間[μs]	VE L1 CACHE MISS TIME(SECS)
ac_ptcc	VE の CPU ポート競合時間[μs]	VE CPU PORT CONFLICT(SECS)
ac_varec	VE のベクトル演算実行時間[μs]	VEC ARITH EXEC TIME(SECS)
ac_vldec	VE のベクトルロード実行時間[μs]	VEC LOAD EXEC TIME(SECS)
ac_ptcc	VEの電力量要因によるHW停止時間[μs]	VE POWER THROTTLING(SECS)
ac_ttcc	VEの温度要因によるHW停止時間[μs]	VE THERMAL THROTTLING(SECS)
ac_max_nthread	VEの最大タスク数	VE MAX NTASK
ac_numanode	VEのNUMAノード番号	NUMA NODE
ac_fmaec	結合積和要素数 ※他項目計算のために集計。	なし
ac_etime	Elapse時間[ticks]	なし

プロセスアカウントの項目名	説明	ジョブアカウントの項目名
	※他項目計算のために集計。	

プロセスアカウントの項目を基に値を計算して、ジョブアカウントに出力する項目は以下の通りです。

ジョブアカウント項目名	説明
VE MEAN SIZE(K)	VEの平均メモリ使用量 [KB]
NUMBER OF VE PROCESS	VEの実行プロセス数
VE MFLOPS	VEのMFLOPS値
VE MFLOPS(REAL)	VEのMFLOPS(REAL)値
VEC-OPERAND RATIO	VEのベクトル化率

3.6.3 プロセスアカウント集計結果 CSV ファイル

プロセスアカウント集計スクリプトは、VE プロセスアカウントファイルを読み込んでジョブ単位にデータを集計し、CSV 形式のファイルに出力します。このファイルの命名規則は下記の通りです。

- ・ユーザ EXIT でプロセスアカウント集計スクリプトを実行された場合

<IP_ADDRESS>_jacct_ps_job.csv

IP_ADDRESSは VH の IP アドレスです。

CSV ファイルの形式は下記の通りです。

列番号	項目名	説明
1	SID	セッション ID
2	START_TIME	全プロセスの最早開始時刻 (※1)
3	END_TIME	全プロセスの最遅終了時間 (※2)
4	VE CPU	VE の CPU 消費時間 (ユーザ)の合計[ticks] ※1tick は 10ms
5	VE KCORE MIN	VE の延べ使用メモリ[KB・ticks]
6	VE MEAN SIZE	VE の平均メモリ使用量[KB]
7	VE TIME SLICE	タイムスライス値[μs]
8	VE MFLOPS	MFLOPS 値

列番号	項目名	説明
9	VE MFLOPS(REAL)	MFLOPS(REAL)値
10	VE FPEC	浮動小数点データ実行要素数
11	VE SYSTEM CALL	システムコール回数
12	VE-VH DATA TRNS	VE-VH 間データ転送量[KB]
13	VE NUM PROCS	実行プロセス数
14	VE INSTRCT	命令実行数
15	VECTOR INST	ベクトル命令実行数
16	VECTOR ELMT	ベクトル要素数
17	VEC-EXE	ベクトル命令実行時間[μs]
18	VE L1CacheMiss	L1 キャッシュミス時間[μs]
19	VE CPUPortConf.	CPU ポート競合時間[μs]
20	VE ArithExec.	ベクトル演算実行時間[μs]
21	V.LoadExec.	ベクトルロード実行時間[μs]
22	V.LoadExecElem	ベクトルロード実行要素数
23	VE PowerThrottling	電力要因による HW 停止時間[μs]
24	VE ThermalThrottling	温度要因による HW 停止時間[μs]
25	VE MAX NTASK	最大タスク数
26	V. Op. Ratio	ベクトル化率
27	VE NUMA NODE NO	NUMA ノード番号
28	FMAE	結合積和要素数
29	Elapse Time	Elapse 時間[ticks]

3.6.4 プロセスアカウント集計スクリプトの設定

① 全 VH 上で集計スクリプトの設定をする

VH にインストールするジョブサーバのパッケージ内に、以下の集計用スクリプトが含まれています。
本スクリプト内に VH の IP アドレス（NQSV のバッチサーバとジョブサーバの通信用 IP アドレス）

と、プロセスアカウント集計結果 CSV ファイルの出力ディレクトリを設定してください。

```
# vi /opt/nec/nqsv/sbin/tally_veacct.sh
: (略)
# === [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"
: (略)
```

- ・ HOST_IP_ADDR 変数には VH の IP アドレスを設定してください。
- ・ OUTPUT_DIR 変数にはプロセスアカウント集計結果 CSV ファイルの出力ディレクトリを設定してください。このディレクトリは VH 上及びアカウンティングサーバ上の両方から参照可能（共有ファイルシステムなど）である必要があります。

② 全 VH 上で logrotate の設定を行う

本機能を使用する場合、ジョブの実行中に Linux の cron によってプロセスアカウントファイルに logrotate が実行された時、プロセスアカウント情報の集計に欠損が出てしまう場合があります。このため、cron によるプロセスアカウントファイルの logrotate を停止し、ジョブが終了したタイミングでのみ、集計スクリプトによって logrotate が実行されるように設定を変更する必要があります。

まず、VH 上にインストールされているプロセスアカウントファイルのローテート用コンフィグファイルを、以下の様にコピーしてください。

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

このコンフィグファイルは、既定値で 1 日に 1 回ローテートを実行し、31 回分（1 か月）保存される設定となっております。設定を変更する場合は当該ファイルを編集してください。

例 1) 保存回数を 31 回分以上（1 か月以上）保存したい場合

例 2) 1 日に出力されるプロセスアカウントファイルのサイズが巨大なため、ローテート頻度を上げたい場合

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

なお、本コンフィグファイルは Linux の logrotate のコンフィグファイルに準拠しておりますので、詳細な設定方法については logrotate に関するページをご確認ください。

次に、編集が完了した psacct-ve0 ファイルを、以下の様にコピーしてください。

```
# 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
```

コピーが完了したら、以下の様にファイルを編集し、行頭の「?」をファイル名末尾の番号と一致させてください。イメージは以下の通りです。

```
# vi /var/opt/nec/nqsv/psacct-ve0
変更前) /var/opt/nec/ve/account/pacct_? {
変更後) /var/opt/nec/ve/account/pacct_0 {

# vi /var/opt/nec/nqsv/psacct-ve1
変更前) /var/opt/nec/ve/account/pacct_? {
変更後) /var/opt/nec/ve/account/pacct_1 {

※psacct-ve2~7についても、上記と同様に変更してください。
```

上記の設定により、ジョブの終了時にコンフィグファイルに従って、そのジョブが使用した VE のプロセスアカウントファイルがローテートされます。

最後に、以下のファイルの全行をコメントアウトしてください。これにより、cron によるプロセスアカウントファイルのローテートが実行されなくなります。

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

③ 全 VH 上で R1.08 (**バージョン 1.08-270**) までのプロセスアカウントの設定を適用済みの場合以下のシンボリックリンクを削除してください

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

④ BSV 上でユーザ EXIT スクリプトを設定する

リクエスト終了のタイミングでプロセスアカウントを集計するために、ユーザ EXIT に集計用スクリプトを設定する必要があります。プロセスアカウントを集計する対象のキューに対して、qmgr(1M) コマンドで設定して下さい。コマンドラインイメージは以下の通りです。

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

この設定により、使用した VH 単位のプロセスアカウント集計結果 CSV ファイル (<IP_ADDRESS>_jacct_ps_job.csv) が、リクエスト終了時に OUTPUT_DIR にこのパスに出力されます。

⑤ BSV 上で DB の INDEX 情報を生成する

プロセスアカウント情報の登録処理を高速化するために、以下のスクリプトを実行してジョブアカウントの DB に INDEX 情報を作成してください。この操作は必ず運用停止中に実施してください。

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

- CLUSTERPRO 等により NQSV を二重化して利用する環境では、プロセスアカウント集計結果 CSV ファイルの出力領域としては以下の 2 つの条件を満たす必要があります。
 - VH 側とアカウンティングサーバ側(運用系と待機系)の両方から当該領域にアクセス（読み取りと書き込みの両方）できること。
 - 当該領域も合わせて二重化すること。ディスク障害等によりプロセスアカウント情報の欠損が発生する可能性がありますので、プロセスアカウント集計結果 CSV ファイルの出力領域の二重化も行ってください。

3.6.5 アカウント DB のプロセスアカウント集計設定

scacctpsm(1M)コマンドは、プロセスアカウント集計結果 CSV ファイルをアカウント DB に登録します。コマンドの実行形式は下記の通りです。

```
scacctpsm -i <input_directory>
```

*input_directory*には、プロセスアカウント集計スクリプトで作成した CSV ファイルの出力ディレクトリを指定します。このコマンドを実行するには root 権限が必要です。このコマンドを実行すると、CSV ファイルのプロセスアカウント情報がジョブ単位で集計され、アカウント DB に登録されます。当該リクエストが既に終了している場合は、ジョブ単位の情報がさらにリクエスト単位で集計されてアカウント DB に登録されます。アカウント DB に登録できた後は、プロセスアカウント集計結果 CSV ファイルを削除します。

また、scacctpsm(1M)コマンドを cron に登録することでリクエストが終了したタイミングで集計されたプロセスアカウント集計結果 CSV ファイルを定期的にアカウント DB に登録することができます。

集計結果は `scacctjob(1)/scacctreq(1)` で参照できます。集計後の項目名および参照方法については「[3.1 リクエストアカウントデータの参照](#)」および「[3.2 ジョブアカウントの参照](#)」節を参照してください。

・ `scacctpsm` コマンドで VE プロセスアカウント情報の集計が完了していないジョブ・リクエストに対しては、`sceditacct` コマンドによる退避処理を実行しないでください。実行した場合、`scacctpsm` コマンドが当該ジョブ情報の記録されている CSV ファイルの情報を登録できず、CSV ファイルが残る状態になります。

・ CLUSTERPRO 等により NQSV を二重化して利用する環境では、`cron` によりプロセスアカウントの集計を定期的実施する際に、運用系と待機系の両方から同じタイミングで集計処理を行うことは避けてください。集計にかかる時間を考慮し、その時間以上の間隔が空くように運用系と待機系の集計タイミングを設定してください。

なお、運用系または待機系の片方が稼働している場合に、もう一方で NQSV がアクセスする領域がマウントされていれば、前述の集計は同じタイミングで行っても問題はありません。稼働していないノードでは集計エラーが発生しますが、稼働しているノードには影響を与えません。

3.6.6 Docker 環境でプロセスアカウント機能を利用する場合

Docker コンテナ起動時の `docker-run` コマンドのオプションに、以下の4つの設定が必要です。

- ① コンテナの PID 名前空間をホスト OS と同じにする。(`--pid=host` オプション)
- ② ホスト OS でプロセスアカウントファイルが出力されるディレクトリを、コンテナ内にホスト OS と同じパスでマウントする。(例: `-v /var/opt/nec/ve/account:/var/opt/nec/ve/account`)
- ③ 集計スクリプト内で設定した、CSV ファイルが出力される共有ファイルシステムのディレクトリを、コンテナ内にホスト OS と同じパスでマウントする。なお、[3.6.4.プロセスアカウント集計スクリプトの設定](#)にて集計スクリプトの `OUTPUT_DIR` に設定したディレクトリと一致させる必要があります。
(例: `-v <OUTPUT_DIR に設定したディレクトリ>:<OUTPUT_DIR に設定したディレクトリ>`)
- ④ ホスト OS の集計スクリプトがインストールされているディレクトリを、コンテナ内にホスト OS と異なるパスでマウントする。
(例: `-v /opt/nec/nqsv/sbin:/tmp/tally_veacct`)

次に、POST-RUNNING 時に実行されるユーザ Exit スクリプトを編集し、以下の太字行を追加します。なお、上記の④でマウント先のディレクトリを「`/tmp/tally_veacct`」以外のパスに指定した場合

は、以下のスクリプトの該当箇所も併せて変更してください。

```
# 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
:
```

以上の設定により、Docker 環境でもリクエストが終了したタイミングで VE のプロセスアカウント情報を集計できるようになります。

第4章 予算管理

予算管理機能を利用する場合、アカウントティングサーバ上で、課金レートの設定(sbudget(1M))、および予算の設定(budgetedit(1M))を行ってください。設定方法は、NEC Network Queuing System V (NQSV)利用の手引 [リファレンス編]のそれぞれのコマンドリファレンスを参照してください。

4.1 課金レートの設定

NQSV の予算管理機能で設定可能な課金項目を下表に示します。

項目名	内容	備考
CPU	CPU消費時間一秒あたりの金額です。	ノンプライムタイムの設定可
MEM	単位メモリ使用量(1KB × 分)あたりの金額です。	ノンプライムタイムの設定可
JOB	1ジョブあたりの金額です。	
GPUNUM	経過時間1GPU 1秒あたりの金額です。	
REQVE	経過時間1VE 1秒あたりの金額です。課金額の計算において要求VEノード数を使用します。(*2)	0以上の値が指定可
RSVVE	経過時間1VE 1秒あたりの金額です。課金額の計算において確保VEノード数を使用します。(*2)	0以上の値が指定可
ELAPSE	経過時間1ジョブ 1秒あたりの金額です。	
RESERVE	リソース予約区間 1 ノード 1 秒あたりの金額です。	
DEC	宣言量に対する重みづけを指定します。	
ACT	実績量に対する重みづけを指定します。	
PRI_MAX	優先度が最大の場合の重みづけを指定します。	1.0以上の値が指定可
PRI_MIN	優先度が最小の場合の重みづけを指定します。	0.0から1.0の間の値が指定可
cname (*1)	カスタムリソースの消費量1あたりの金額です。	20種類まで登録可

*1 cnameはカスタムリソース名になります。

*2 これらの項目は実行ホストがSX-Aurora TSUBASAの場合のみ有効です。

課金レートはキュー単位および OpenStack や Docker のテンプレート単位で設定することが可能です。(OpenStack は実行ホストが SX-Aurora TSUBASA の場合は利用できません。)

OpenStack や Docker と連携したプロビジョニング環境を使用しない場合は、キュー単位の課金レートのみを設定してください。課金レートが設定されているキューへのリクエストの投入、および、リソース予約区間の生成に対しては、キューごとに設定された課金レートから、当該キューを使用したユーザ(グループ、アカウントコード)の課金額を計算し、予算から減算されます。

課金レートが設定されていないキューへのリクエストの投入、および、リソース予約区間の生成に対しては、予算の設定の有無にかかわらず、常に投入、生成が可能です。また、このとき、リクエスト、リソース予約区間の実行に対しては、予算に対する課金の計算は行いません。

OpenStack や Docker と連携したプロビジョニング環境を使用する場合は、キュー単位の課金レ

トとテンプレート単位の課金レートを設定してください。

課金レートが設定されているキューへのリクエストの投入、および、リソース予約区間の生成に対しては、テンプレートごとに設定された課金レートから課金額を計算し、予算から減算されます。

課金レートが設定されていないキューへのリクエストの投入、および、リソース予約区間の生成に対しては、予算の設定の有無にかかわらず、常に投入、生成が可能です。また、このとき、リクエスト、リソース予約区間の実行に対しては、予算に対する課金の計算は行いません。

項目名 REQVE と RSVVE は VE ノードに対する課金レートです。これらは 1VE につき経過時間 1 秒当たりの課金項目の設定です。課金額の計算において要求 VE ノード数と確保 VE ノード数の両方もしくはいずれかで課金するかを設定可能です。例えば 1VE1 秒当たり 100 の金額に対し、要求 VE ノード数で 25%、確保 VE ノード数で 75%の比率で課金する場合は、REQVE に 25、RSVVE に 75 を設定します。一方の値を 0 にすれば他方の課金項目でのみ課金されます。

4.1.1 sbuedit(1M)

sbuedit(1M) コマンドはキューおよびテンプレートに対して課金レートを設定するコマンドです。実行形式は下記の通りです。

```
# /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
```


コマンドの使用方法は オペレーション→キュー名 or テンプレート名 → 課金項目 → 課金レートの順に設定します（OpenStack のテンプレート名は、実行ホストが SX-Aurora TSUBASA の場合は利用できません）。sbuedit (1M)コマンドの実行ができるのは、スーパーユーザのみです。

- オペレーション

add	新たにキューに対する課金レートを追加します。
upd	既に登録済みのキューの設定を更新します。
del	登録されているキューの課金レートの設定を削除します。

新たに課金レートを追加する場合、下記のように実行します。

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

上記は、キュー名が quename のキューに対する課金レートとして、CPU に 1533.00 と 774.56、MEM に 0.234、カスタムリソース Power に 10.0 の設定を追加する例です。ここで、CPU の"/" で区切られた2つの値は、"/"の前の 1533.00 が、プライムタイムの課金レートで、"/"の後の 774.56 が、ノンプライムタイムの課金レートです。MEM のように省略した場合にはプライムタイムとノンプライムタイムに対して同じ課金レートが設定されます。

プライムタイム、ノンプライムタイムの定義に関しては、2.2 環境の構築を参照してください。

カスタムリソースの課金レートはカスタムリソースの名前の前に”CR_”を付けて指定します。

sbuedit(1M) コマンドを、引数を指定せずに実行した場合、キュー毎の課金レートが、順に表示されます。以下にキュー 1 つ分の表示例を示します。

```
-----
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

上記の設定の場合、あるユーザが `quename` のキューに投入したリクエストにより、プライムタイムΔ時間内に、CPU を 0.12 秒使用した場合、

$0.12 * 11.00 = 1.32$
が課金されます。

※詳細は、NEC Network Queuing System V (NQSV)利用の手引 [リファレンス編]の `sbuedit(1M)` を参照してください。

4.1.2 qmove 時の課金レート of 扱い

`qmove(1)` コマンドを使用して、キュー間でリクエストを移動した場合は、リクエスト終了時に常に移動先のキューの課金レートで課金されます。`scacctreq(1)` や `scacctjob(1)` で表示されるキュー名も移動先のキューになります。

4.2 予算の参照、設定

NQSV の予算管理機能において予算の設定は、`budgetedit(1M)` コマンドで行います。`budgetedit(1M)` コマンドにより、ユーザ、グループ、アカウントコード別に、予算の初期値を設定してください。また、`budgetedit(1M)` コマンドで、予算の初期値に加えて、予算の残高、予定課金額の変更を行うこともできます。

NQSV の予算管理機能で予算管理を行う場合、アカウントティングサーバの設定で `SBU_ORDER` に指定した課金種別（ユーザ、グループ、アカウントコード）に対して予算の設定を行う必要があります。たとえば、`SBU_ORDER=UG` と指定されている場合、リクエストの投入時には、まず投入リクエストのユーザの予算の超過をチェックし、超過していなければ、そのユーザに対して課金が行われます。

投入ユーザの予算が超過している場合は、次にそのリクエストのグループの予算をチェックして、超過していなければ、そのグループに対して課金が行われます。投入グループの予算も超過している場合は、予算超過により投入がエラーとなります。このとき、投入ユーザ、または、投入グループに対して、予算が設定されていない場合は、そのユーザ、グループに対しては常に予算超過しているものとして処理されます。

また、`SBU_ORDER` で、アカウントコードに対して課金を行うように設定されている場合、アカウントコードを指定しないで投入されたリクエストに対しては、アカウントコードに対する予算が常に超過しているものとして処理されます。

同様に、グループに対して課金を行うように設定されている場合、グループを指定しないリソース予約区間の生成に対しては、グループに対する予算が常に超過しているものとして処理されます。

4.2.1 budgetedit(1M)

`budgetedit(1M)` コマンドはユーザ、グループ、またはアカウントコードに対して、予算を設定するコマンドです。このコマンドはアカウントティングサーバ上で使用します。実行形式は下記の通りです。

```
# /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
```

コマンドの使用方法は オペレーション → 適応範囲 → 名前 → 予算の順に設定します。`budgetedit(1M)` コマンドの実行ができるのは、スーパーユーザのみです。

- オペレーション

add	新たに予算を追加します。
-----	--------------

upd	既に登録済みの予算を更新します。
del	登録されている予算を削除します。

- 適用範囲

-u	ユーザの予算を設定します。
-g	グループの予算を設定します。
-a	アカウントコードの予算を設定します。 <i>name</i> にアカウントコードファイル (/etc/opt/nec/nqsv/acctcode) に記述されているアカウントコード名を指定します。アカウントコード名の最大長は15バイトまでです。

新たに予算を追加する場合、下記のように実行します。

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

上記のように設定した場合、user1 には 125455.56 の予算が、user2 には 2345566.44 が設定されます。ここで、"/" で区切って初期値/残高/予定課金額を指定することができます。"/" で区切らず初期値のみを指定した場合は残高には初期値を同じ値が設定され、予定課金額は 0 に設定されます。

budgetedit(1M) コマンドを、引数を指定せずに実行した場合、予算が設定されているユーザ、グループ、アカウントコードについて順に表示されます。下記に表示例を示します。

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
-------	------	------	-------------

表示はユーザ名（グループ名、アカウントコード）残高、予定課金額、初期値の値が表示され、残高が 0 のユーザ（グループ、アカウントコード）は予算超過となります。

登録済みの予算を更新する場合、下記のいずれかの方法で実行します。

- ・絶対値指定の場合

指定した予算額で予算データを更新します。符号の付いた金額を指定することはできません。

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

- ・相対値指定の場合

現在の予算額を、指定した相対値で加減算した結果で予算データを更新します。指定する金額には符号が必要です。マイナスとプラスの符号を混ぜて指定できません。変更しない値は 0 で指定してください。

```
# /opt/nec/nqsv/bin/budgetedit upd -u user1:+1000/+1000/0
```

※詳細は、NEC Network Queuing System V (NQSV)利用の手引 [リファレンス編]の budgetedit(1M)を参照してください。

4.2.2 budgetcheck(1)

budgetcheck(1)コマンドは、アカウントティングサーバ上でユーザ、グループ、または、アカウントコードの予算実績を確認するためのコマンドです。-u オプション、-g オプション、-a オプションの指定により、ユーザ、グループ、または、アカウントコードの予算実績を検索し、標準出力に、各ユーザ、グループ、アカウントコードごとに、ユーザ名（グループ名、アカウントコード名）と予算残高、予定課金額、予算初期値を表示し、予算超過しているかどうかを返却します。オプションが指定されていない場合はユーザの予算実績をチェックします。予算が超過している場合には終了ステータスが 1 以上となります。このコマンドはアカウントティングサーバ上で使用します。

budgetcheck(1)の実行形式は下記の通りです。

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

budgetcheck(1)の終了ステータスは、以下の通りです。複数の予算が超過した場合は、その論理和

が終了ステータスとなります。

終了ステータス	意味
0	予算超過なし
1	ユーザの予算が超過
2	グループの予算が超過
4	アカウントコードの予算が超過

ユーザおよびグループの両方で予算超過を確認する場合は下記のように実行します。

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

ユーザの予算が超過している場合は 1 が、グループの予算が超過している場合は 2 が、ユーザ、グループ両方が超過している場合は 3 が budgetcheck(1) の終了ステータスとなります。

budgetcheck(1) コマンドは、実行時の NQSV の権限に応じて、以下に示す範囲の予算の状況をチェックします。実行時の権限は、-P オプションにより指定します。既定値は、一般ユーザ権限です。

権限	参照可能な情報
一般ユーザ権限	-u の場合、自分の予算実績をチェック
特別ユーザ権限	-g の場合、自分が属するグループの予算実績をチェック -a の場合、自分が使用可能なアカウントコードの予算実績をチェック
グループ管理者	-u の場合、自分の予算実績をチェック -g の場合、自分が管理しているグループの予約実績をチェック -a の場合、自分が使用可能なアカウントコードの予算実績をチェック
操作員権限	全ユーザ、全グループ、または全アカウントコードの予算実績をチェック
管理者権限	

※詳細は、NEC Network Queuing System V (NQSV) 利用の手引 [リファレンス編] の budgetcheck(1) を参照してください。

4.2.3 rbudgetcheck(1)

rbudgetcheck(1) コマンドは、クライアントホスト上で予算の超過を確認するためのコマンドで、アカウントティングサーバの budgetcheck(1) コマンドをリモート実行することにより、予算超過を確認することができます。

rbudgetcheck(1) の実行形式は下記の通り、第一引数にアカウントティングサーバのホスト名を指定します。

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

rbudgetcheck(1) のユーザ、グループ、アカウントコードの指定、実行時の権限によるチェック範囲、表示内容、および、終了ステータスは、budgetcheck(1)コマンドと同じです。

※詳細は、NEC Network Queuing System V (NQSV)利用の手引 [リファレンス編]の rbudgetcheck(1)を参照してください。

付録 A 注意点および補足説明

A.1 運用上の注意点

(1) アカウントデータベースの最適化

アカウントティングサーバ上のデータベースにはバッチサーバおよび JobManipulator から送信されるアカウントデータが蓄積されますが、時間の経過、ユーザの使用頻度によってそのアカウントデータの量は大きくなります。

アカウントティングサーバで使用するデータベースは登録されるデータの量が多くなるにつれデータ検索のためのオーバーヘッドが大きくなります。

アカウントティングでサポートされる `sceditacct(1M)` コマンドを使用してデータの退避を行うことで、アカウントデータの表示コマンドの実行時間が短縮されます。

下記に、`sceditacct(1M)` コマンドの実行例を示します。

```
/opt/nec/nqsv/bin/sceditacct -j -S 1/1 -m /tmp/jacct2019_03 (ジョブアカウント)
/opt/nec/nqsv/bin/sceditacct -r -S 1/1 -m /tmp/racct2019_03 (リクエストアカウント)
/opt/nec/nqsv/bin/sceditacct -y -S 1/1 -m /tmp/rsvacct2019_03 (予約アカウント)
```

この場合、全てのリクエスト(ジョブ)アカウントデータは `-m` オプションで指定された場所に退避されますが、このデータは最適化されているため、`scacctreq(1)` コマンド、`scacctjob(1)` コマンドでの表示、`scsumacct(1M)` コマンドの集計表示などにかかる時間が短縮されます。

ジョブアカウントについて月次処理でデータベースを退避する場合のサンプルを以下に示します。

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

なお、リクエストアカウントは実行が終了するまでは開始時刻および終了時刻が 0 で DB に登録されています。`sceditacct(1M)` コマンドは、実行が終了していないデータを除外して退避します。このようなデータを削除するときは、`-f` オプションを指定し、`-s` および `-S` オプションは指定しないで `sceditacct(1M)` コマンドを実行してください。

`-s` および `-S` オプションは範囲指定条件の開始時刻条件の指定なので、開始時刻が 0 のデータを削除する指定である `-f` の効果が打ち消されるためです。

また、実行が開始前に削除されたリクエストは、開始時刻および終了時刻が 0 で DB に登録されています。実行が開始前に削除されたリクエストアカウントデータを退避し、実行中のリクエストアカウントデータを残す場合は、`-q` または `-Q` オプションで投入時刻を指定してください。この場合、`-f` の指定は不要です。

※詳細は、[3.5 アカウントデータの退避](#) を参照してください。

(2) データベースファイルの上限について

各アカウント用のデータベースファイルはサイズが 1800MB を超えるとデータが登録されなくなります。定期的に `sceditacct` でデータを退避してください。なお、アカウントティングサーバのコンフィグファイルに `DBSIZE_LIMIT` を設定することで、データベースファイルの上限を変更することが可能です。

(3) アカウントファイルの上限について

各アカウントファイルはサイズが 128GB を超えるとデータが転送されなくなり、アカウントデータベースへの登録が止まります。ファイルサイズが大きくなると、アカウントの集計性能に影響を与える可能性があるため、定期的に `scturnacct` でデータを退避してください。なお、アカウントティングモニタのコンフィグファイルに `ACCTFILESIZE_LIMIT` を設定することで、アカウントファイルの上限を変更することが可能です。

(4) 各アカウントファイルを `scturnacct(1M)` 以外で退避した場合について

アカウントティングモニタはアカウントファイルの既に収集済みの最終レコードの位置を、`".last"` ファイルに格納します。`scturnacct(1M)` コマンドは、既に収集済みのレコードを別名のアカウントファイルに格納することで、アカウントファイルの世代管理を行いません。アカウントデータに対して、`scturnacct(1M)` 以外のコマンドで退避した場合は、アカウントデータが欠落します。必ず、`scturnacct(1M)` コマンドを利用した退避を行ってください。

`scturnacct(1M)` 以外のコマンドでアカウントファイルを移動または、削除してしまった場合には、`.last` ファイルを削除して下さい。この場合、移動や削除を行ったファイルのアカウントデータはサーバに送信されません。

(5) ジョブの使用 VE ノード番号のリストについて

ジョブの使用 VE ノード番号のリストは、バッチサーバの設定のジョブの資源使用量採取間隔の設定値毎に VE を使用するプロセスの存在確認を行い作成します。従って当該間隔の間にプロセスの生成消滅が発生したなどの場合、正確な情報が採取されない可能性があります。

(6) 課金機能有効時に定期的に行う必要があるスクリプトについて

課金機能が有効な場合、リクエスト投入時にリクエスト ID、BSV ホストのアドレス、ジョブ数等を初期状態として、リクエストアカウント DB(racct)に登録します。その後リクエスト終了後に当該レコードに実績値等を更新しますが、実績データが正常にアカウントサーバに送信できなかった等の要因で実績値を更新できない場合、リクエスト投入時に登録されたリクエストアカウントのレコードは初期状態のまま残り続けます。この初期状態レコードは DB 退避(sceditacct -E 等による退避)を行っても退避されません。

初期状態レコードが残ったまま、リクエスト ID が一周(デフォルトは 0~999999)し、残ったままのレコードとリクエストの基本情報が同じリクエストが投入された場合、DB 重複エラーが発生します。この時、リクエスト終了・削除時には残ったままの初期状態レコードに対して更新されますが、予定課金額・ジョブ数・user・グループ等一部のカラムは古いリクエストの情報をそのまま使用するため、正しいユーザ・グループへの課金情報の反映が行われません。

このような問題を防ぐため、初期状態のまま残存したレコードを削除するスクリプト (cleanup_racct.sh) を定期的に行ってください。ユーザの運用状況からリクエスト ID が一周する前に実行することを推奨します。

ただし、初期状態のまま残存したレコードとリクエスト投入後実行待ちや実行中等でまだ情報が更新されていない状態のレコードとの判別ができません。そのため、本スクリプトは以下の条件を全て満たすタイミングで実行してください。

- ・ qstat でリクエスト投入後実行待ちや実行中のリクエストが存在しない
- ・ /var/opt/nec/nqsv/asv/master/<IP アドレス>/ ディレクトリ以下 にファイルがない
- ・ NQSV 停止前、最後に実行完了したリクエストの情報が scacctreq で表示され、課金額の集計、反映が完了されている
- ・ NQSV の運用が停止されている

本スクリプトの実行方法は以下の通りです。

```
# /opt/nec/nqsv/sbin/cleanup_racct.sh
```

```
Deleted Request ID
```

```
132321
```

```
132322
```

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

本スクリプトをオプション無しで実行すると、削除対象のレコードがあるかどうかのチェックのみ行います。削除対象のレコードがある場合、リクエスト ID が標準出力に表示されます。リクエスト

ID が表示されなかった場合、この後の処理は不要です。

リクエスト ID が表示された場合、スクリプトを `-f` オプションをつけて実行してください。DB から削除対象のレコードが削除されます。

```
# /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
```

削除対象レコードは `/var/opt/nec/nqsv/asv/` 以下に `cleanup_backup` ディレクトリを作成し、ファイル名 `deleted_racct_records.bak_YYYYMMDDhhmmss` の形式で保存します。

誤って実行待ち/実行中/DB 集計待ちリクエストの初期状態レコードを削除してしまった場合、復旧することはできません。そのため、前述の条件にあったタイミングでスクリプトを実行してください。

A.2 予約の終了ステータスについて

予約の終了ステータスは、`scacctrsv(1)` コマンドの `STATUS` で確認できます。`STATUS` の意味を下表に示します。

STATUS	意味
DONE	予約が終了したことを示します。
DELETED	予約が開始する前に削除されたことを示します。
QUIT	予約が開始され終了予定時刻よりも前に削除されたことを示します。

A.3 リクエストの終了ステータスについて

リクエストの終了ステータスは、scacctreq(1)コマンドの STATUS もしくは EXIT STAT で確認できます。リクエストの終了ステータスを簡易的に確認する場合は STATUS を参照してください。STATUS の意味を下表に示します。

STATUS	意味
DONE	リクエストが終了したことを示します。
DELETED	リクエストがqdelされたことを示します。
RERUN	リクエストがrerunされたことを示します。

リクエストの終了ステータスを詳細に確認する場合は scacctreq -h で出力される EXIT STAT を参照してください。EXIT STAT のフォーマットと意味を下表に示します。-R オプションを併用すると終了コードに対応したメッセージが出力されます。

フォーマット	意味	
YYYYXXXX	YYYY	オペレータ
	XXXX	終了コード

※EXIT STAT は 16 進数で表示します。また、先頭の 0 は表示しません。

オペレータ	意味
0x0001	リクエストのオーナー
0x0002	管理者、操作員、またはシステム

※オペレータは、終了ステータスが 0x0002 または 0x0003 の場合に設定されます。

終了コード	意味 (バッチ、会話、サブリクエスト)	意味 (パラメトリックリクエスト)
0x0000	マスタージョブがexit(0)で終了したことを示します。	全サブリクエストがexit(0)で終了したを示します。
0x0001	マスタージョブがexit(0)以外で終了したことを示します(シグナル終了を含む)。	1つ以上のサブリクエストがexit(0)以外で終了したことを示します。
0x0002	リクエストがqdelされたことを示します。	パラメトリックリクエストがqdelされたことを示します。
0x0003	リクエストがrerunされたことを示します。	パラメトリックリクエストがrerunされたことを示します。

A.4 ジョブの終了ステータスについて

ジョブの終了ステータスは `scacctjob -h` で出力される EXIT STAT で確認できます。EXIT STAT のフォーマットと意味を下表に示します。`-R` オプションを併用すると終了コードに対応したメッセージが出力されます。

フォーマット	意味	
y nn or y ss or xxxx	y	オペレータ情報
	nn	ログインシェルの終了コード
	ss	シグナル番号
	xxxx	終了コード

※ EXIT STATUSは16進数で表示します。また、先頭の0は表示しません。終了コードが0x402でシグナル番号が15、オペレータ情報が1なら10f0402、オペレータ情報が0ならf0402と表示し、終了コードが0x402でシグナル番号が無く、オペレータ情報が1なら1000402、オペレータ情報が0なら402と表示します。

終了コード	意味
nn0000	ジョブが正常に終了したことを示します。
nn0001	一般ユーザ権限でキャンセルされたことを示します。
nn0003	システムがクラッシュしたことを示します。
nn0004	リクエストのマスタージョブが存在しないことを示します。
ss0402	一般ユーザ権限で qdel(1) されたことを示します。
nn0802	
ss2402	一般ユーザ権限でリランされたことを示します。
nn4000	一般ユーザ権限でジョブマイグレーションされたことを示します。
nn4001	
1nn0001	操作員権限でキャンセルされたことを示します。
1ss0402	操作員権限で qdel(1) されたことを示します。
1nn0802	
1ss2402	操作員権限でリランされたことを示します。
1nn4000	操作員権限でジョブマイグレーションされたことを示します。
1nn4001	
2nn0202	ジョブの実行に失敗したことを示します。
2ss0402	ジョブにシグナルが送信されたことを示します。
2nn0502	シャットダウンによって qdel(1) されたことを示します。
2nn1102	ジョブのリスタートが失敗したことを示します。
2nn1802	ジョブのリリースが失敗したことを示します。
2nn2502	シャットダウンによってリランされたことを示します。
3nn0001	管理者権限でキャンセルされたことを示します。
3ss0402	管理者権限で qdel(1) されたことを示します。
3nn0802	
3ss2402	管理者権限でリランされたことを示します。
3nn4000	管理者権限でジョブマイグレーションされたことを示します。
3nn4001	

4nn0001	スケジューラによってキャンセルされたことを示します。
4ss0402	スケジューラによって qdel(1) されたことを示します。
4nn0802	
4ss2402	スケジューラによってリランされたことを示します。
4nn4000	スケジューラによってジョブマイグレーションされたことを示します。
4nn4001	
5nn0001	システムによってキャンセルされたことを示します。
5ss0402	システムによって qdel(1) されたことを示します。
5nn0802	
5ss2402	システムによってリランされたことを示します。
5nn4000	システムによってジョブマイグレーションされたことを示します。
5nn4001	

A.5 課金額の計算について

課金額は以下の方法で算出されます。課金額はリソース区間作成時およびリクエスト投入時に指定された資源制限値に基づいて算出しますので、課金するリソースに応じた資源制限値が適切に設定されるように運用する必要があります。

【課金額の算出方法】

課金額 = リソース毎の課金額の合計

【リソース毎の課金額の算出方法】

リソース毎の課金額 = (DEC × 宣言値 + ACT × 実績値) × リソースの課金レート × 優先度係数

【宣言値と実績値の算出方法】

リソース予約区間の場合		
リソース	宣言値	実績値(*1)
RESERVE	ホスト数またはマシン数 ×リソース予約区間の予定期間	ホスト数またはマシン数 ×リソース予約区間の実期間
リクエストの場合		
リソース	宣言値	実績値(※)
CPU	各ジョブの要求CPU時間の合計	各ジョブのCPU時間の合計
MEM	各ジョブの要求最大メモリ量の合計 ×要求Elapse時間	各ジョブのメモリ使用量の合計
JOB	ジョブ数	ジョブ数
GPUNUM	各ジョブの同時実行GPU台数制限値の合計×要求Elapse時間	Σ(各ジョブの同時実行GPU台数制限値 ×各ジョブのElapse時間)
REQVE	要求VEノード数 ×要求Elapse時間	Σ(各ジョブの割当VEノード数 ×各ジョブのElapse時間)
RSVVE	確保VEノード数 ×要求Elapse時間	Σ(ジョブ毎の確保VEノード数 ×各ジョブのElapse時間)
ELAPSE	要求Elapse時間×ジョブ数	各ジョブのElapse時間の合計
cr_name*2 (check_mode=off)	消費単位がリクエストの場合：カスタムリソースの宣言量 消費単位がジョブの場合：カスタムリソースの宣言量×ジョブ数	
cr_name*2 (check_mode=integrate)	消費単位がリクエストの場合： カスタムリソースの宣言量 消費単位がジョブの場合： カスタムリソースの宣言量× ジョブ数	各ジョブのカスタムリソースの使用量の合計
cr_name*2 (check_mode=moment)	消費単位がリクエストの場合： カスタムリソースの宣言量×	Σ(各ジョブのカスタムリソースの使用量×各ジョブのElapse時間)

	要求Elapse時間(分) 消費単位がジョブの場合： カスタムリソースの宣言量× ジョブ数×要求Elapse時間(分)	
--	--	--

*1 予算超過チェックでは実績値を宣言値と同値とみなして予定課金額が算出されます。

*2 カスタムリソース作成時は、check_modeのオプションが指定可能です。check_modeとは、カスタムリソースの資源を監視および出力するか否かのモードのことです。offの場合は、カスタムリソースの使用量は監視および出力されません。momentの場合はカスタムリソースの使用量を瞬間値として監視および出力されます。integrateの場合は、カスタムリソースの使用量を積算値として監視および出力されます。詳細について、「NEC Network Queuing System V (NQS) 利用の手引 [管理編]」を参照してください。

【優先度係数の算出方法】

リクエストの優先度	優先度係数の算出方法
1 ~ 1023	$1.0 + \text{リクエストの優先度} \times (\text{PRI_MAX} - 1.0) \div 1023$
0	1.0
-1 ~ -1024	$1.0 + \text{リクエストの優先度} \times (1.0 - \text{PRI_MIN}) \div 1024$

付録 B アカウントファイル一覧

• アカウンティングサーバ

ファイル	説明
/var/opt/nec/nqsv/asv/jacct	ジョブアカウント用DBファイル
/var/opt/nec/nqsv/asv/jacct_cr	ジョブアカウント用DBファイル（カスタムリソース用）
/var/opt/nec/nqsv/asv/jacct_ps	ジョブアカウント用DBファイル（VEプロセスアカウント用）
/var/opt/nec/nqsv/asv/racct	リクエストアカウント用DBファイル
/var/opt/nec/nqsv/asv/racct_tally	リクエストアカウント用DBファイル（集計データ用）
/var/opt/nec/nqsv/asv/racct_cr	リクエストアカウント用DBファイル（カスタムリソース用）
/var/opt/nec/nqsv/asv/racct_cr_tally	リクエストアカウント用DBファイル（カスタムリソースの集計データ用）
/var/opt/nec/nqsv/asv/racct_ps_tally	リクエストアカウント用DBファイル（VEプロセスアカウントの集計データ用）
/var/opt/nec/nqsv/asv/rsvacct	予約アカウント用DBファイル
/var/opt/nec/nqsv/asv/sbu	課金レートDBファイル
/var/opt/nec/nqsv/asv/budget	予算DBファイル
/var/opt/nec/nqsv/asv/master/XXX.XXX.XXX.XXX ¹ /tjad_NNNNN ²	受信したジョブアカウントファイル
/var/opt/nec/nqsv/asv/master/XXX.XXX.XXX.XXX ¹ /racct_NNNNN ²	受信したリクエストアカウントファイル
/var/opt/nec/nqsv/asv/master/XXX.XXX.XXX.XXX ¹ /rsvacct_NNN ²	受信した予約アカウントファイル

¹ XXX.XXX.XXX.XXXはアカウンティングモニタのIPアドレスです。

² NNNNNは00000から始まる通番です。

• アカウンティングモニタ

ファイル	説明
/var/opt/nec/nqsv/bsv/account/reqacct[.M] ¹	リクエストアカウントデータ
/var/opt/nec/nqsv/bsv/account/reqacct_del[.M]	リクエストアカウントデータ（開始前削除）
/var/opt/nec/nqsv/bsv/account/.reqacct.last	送信済みリクエストアカウントデータ位置保存ファイル
/var/opt/nec/nqsv/bsv/account/.reqacct.yet	未送信のリクエストアカウントデータ
/var/opt/nec/nqsv/bsv/account/.reqacct.yet.last	送信済みリクエストアカウントデータ位

	置保存ファイル (.reqacct.yet用)
/var/opt/nec/nqsv/acm/jacct/jobacct	ジョブアカウントデータ
/var/opt/nec/nqsv/acm/jacct/.jobacct.last	送信済みジョブアカウントデータ位置保存ファイル
/var/opt/nec/nqsv/acm/jacct/.jobacct.yet	未送信のジョブアカウントデータ
/var/opt/nec/nqsv/acm/jacct/.jobacct.yet.last	送信済みジョブアカウントデータ位置保存ファイル (.jobacct.yet用)
/var/opt/nec/nqsv/acm/rsvacct/rsvacct	予約アカウントデータ
/var/opt/nec/nqsv/acm/rsvacct/.rsvacct.last	送信済み予約アカウントデータ位置保存ファイル
/var/opt/nec/nqsv/acm/rsvacct/.rsvacct.yet	未送信の予約アカウントデータ
/var/opt/nec/nqsv/acm/rsvacct/.rsvacct.yet.last	送信済み予約アカウントデータ位置保存ファイル (.rsvacct.yet用)

¹ [.N]は世代数です。世代数が 0 の場合は世代数を付加しません。

付録 C 発行履歴

C.1 発行履歴一覧表

2018 年	2 月	初版
2021 年	12 月	第 12 版
2023 年	1 月	第 13 版
2023 年	3 月	第 14 版
2023 年	6 月	第 15 版
2024 年	3 月	第 16 版

C.2 追加・変更点詳細

- ・ 第12版
 - 「3.6.4 プロセスアカウント集計スクリプトの設定」の⑤を修正
- ・ 第13版
 - 「4.1 課金レートの設定」
リクエストqmove時の適用課金レートの記述を修正して当該内容を4.1.2 qmove時の課金レートの扱いに移動
- ・ 第14版
 - 「3.1 リクエストアカウントデータの参照」、「3.2 ジョブアカウントデータの参照」
V KCORE MINの説明を修正
VE MEAN SIZE(K)の説明を追記
 - 「3.6.2 集計項目一覧」
VEプロセスアカウントのVEOS側における構造体名を更新
- ・ 第15版
 - 「4.1.2 qmove時の課金レートの扱い」
NQSV R1.13におけるqmove時の課金キューの修正に伴い、課金レートが適用されるキ

ユーを修正

- ・ 第16版
 - 「3.1 リクエストアカウントデータの参照」「3.2 ジョブアカウントデータの参照」
REQUEST_NAMEの説明を修正
 - 「付録A.1 (6) 課金機能有効時に定期的に行う必要があるスクリプトについて」
課金機能有効時に実行するスクリプトの説明を修正

索引

B	V
BMC vi	VE vi
	VI クラスター vi
H	
HCA vi	か
	課金レート 52
I	環境の構築 5
IB vi	
M	へ
MPI vi	ベクトルエンジン vi
	ベクトルホスト vi
N	
NIC vi	

**NEC Network Queuing System V (NQSⅤ)
利用の手引〔アカウントィング・予算管理編〕**

2024年3月 第16版

日本電気株式会社

東京都港区芝五丁目7番1号

TEL(03)3454-1111 (大代表)

© NEC Corporation 2018

日本電気株式会社の許可なく複製・改変などを行うことはできません。

本書の内容に関しては将来予告なしに変更することがあります。