

HP OpenView Performance Agent

Linux オペレーティング システム

ソフトウェア バージョン : C.04.60.000

インストール、設定ガイド

Manufacturing Part Number: B7491-99079

本書の発行日 : 2006 年 10 月

ソフトウェアのリリース日 : 2006 年 10 月



ご注意

保証書

HP 製品およびサービスに対する保証は、それらの製品およびサービスに付属している保証規定に明記された条項に限られます。本書に記載された内容は、追加の保証を規定するものではありません。HP は、本書の技術的および編集上の誤りや不備について、その責任を負わないものとします。

この情報は予告なしに変更されることがあります。

権利の制限

本書で取り扱っているコンピュータ ソフトウェアは秘密情報であり、その保有、使用、または複製には、HP から使用許諾を得る必要があります。FAR 12.211 および 12.212 に従って、商業用コンピュータ ソフトウェア、コンピュータ ソフトウェアドキュメンテーション、および商業用製品の技術データは、ベンダ標準の商業用ライセンスのもとで、米国政府にライセンスが付与されます。

著作権

© Copyright 1983-2006 Hewlett-Packard Development Company, L.P.

商標

UNIX® は、The Open Group の登録商標です。

Adobe®、Acrobat®、および PostScript® は Adobe Systems Incorporated の商標です。

Motif® は、米国およびその他の国々における Open Software Foundation の登録商標です。

その他の製品名は各社の商標またはサービス マークです。

サポートについて

次の HP OpenView のサポート専用 Web サイトを参照することができます。

<http://www.hp.com/managementsoftware/support>

HP OpenView オンライン サポートでは、対話型テクニカル サポート ツールをすぐにご利用いただけます。このサポート サイトでは、次の機能が用意されています。

- 技術情報の検索
- サポート ケースの登録とトラッキング、およびエンハンスメント要求の送信とトラッキング
- ソフトウェア パッチのダウンロード
- サポート 契約の管理
- HP サポート連絡先の検索
- 利用可能なサービスの参照
- ユーザー同士のディスカッション
- ソフトウェア トレーニングの検索と登録

ほとんどのサポート エリアでは、アクセスするために HP Passport ユーザーとして登録、サインインする必要があります。また、多くのサポート エリアでサポート契約が必要です。

アクセス レベルに関する詳細は以下を参照してください。

http://www.hp.com/managementsoftware/access_level

HP Passport ID の登録は以下で行うことができます。

<http://www.managementsoftware.hp.com/passport-registration.html>

目次

1	OpenView Performance Agent のインストールまたはアップグレード	7
	OpenView Performance Agent について	7
	インストール要件	9
	対応しているアーキテクチャおよびオペレーティング システムのバージョン	9
	ディスク スペース	9
	インストールまたはアップグレードの手順	11
	アップグレード	11
	初回インストールの場合	11
	アクティブなパフォーマンス ツール プロセスの終了	11
	OV Performance Agent のインストール	12
	Debian での OV Performance Agent のインストール	14
	OV Operations を使用している場合の OV Performance Agent のインストール	14
	仮想環境における OV Performance Agent	15
	サービス コンソール上の OV Performance Agent	15
	仮想マシン上の OV Performance Agent	15
	OV Performance Agent の削除	17
2	OV Performance Agent の実行方法	19
	OV Performance Agent の起動と終了	20
	自動的な起動と終了	21
	ステータスの確認方法	22
	ディレクトリのサンプル	22
	ファイアウォールを介した通信	23
	OVPA ポートの設定	24
	ファイアウォール環境での HTTP クライアントの設定	26
	ファイアウォール設定の確認	28

複数の IP アドレスを持つシステムの設定	28
安全な通信の設定	30
証明書の使用	30
クライアント認証の使用	30
authip ファイルによる認証の有効化	31
authip ファイルのフォーマット	32
データソースの設定	33
datasources 設定ファイルのフォーマット	33
データソースの削除	34
parm ファイル	35
アラームの定義	36
3 ドキュメント	37
OV Performance Agent ドキュメント一覧	38
Web 上でのドキュメントの参照	39
Adobe Acrobat ファイル	40
A coda の設定	41
はじめに	41
用語集	45
索引	53

1 OpenView Performance Agent のインストールまたはアップグレード

OpenView Performance Agent について

HP OpenView Performance Agent (OVPA) は、ご使用の Linux system から、パフォーマンス、リソース、およびトランザクションデータを取り込みます。また、最小限のシステムリソースを使用して、システム上の現在および過去のリソースデータに関するアラーム状態の連続的な収集、要約、タイムスタンプ作成、および検出を行います。

OV Performance Agent は、VMware ESX Server でサポートされるようになりました。詳細は、15 ページの「[仮想環境における OV Performance Agent](#)」を参照してください。

ローカル エクスポート機能、または OV Reporter や OV Performance Manager (OVPM) 4.0 以降などの相補的な OpenView 製品を使用して、データを分析することができます。さらに、OV Performance Agent は、OpenView Network Node Manager (NNM) および OpenView Operations (OVO) へのデータアクセスとアラーム通知機能を備えています。



本書における OV Performance Manager (OVPM) という語は、UNIX および Windows プラットフォーム対応のバージョン 4.0 およびそれ以降のみを指します。OVPM 3.x (PerfView) は、OV Performance Agent for Linux には接続しません。

OV Performance Agent は、データソース統合 (DSI: Data Source Integration) 技術を使用して、アプリケーション、データベース、ネットワーク、その他のオペレーティングシステムなどの外部データソースからデータを受信し、アラーム状態を検出し、ログを記録します。

システムから OVPA によって記録および保存された包括的なシステムパフォーマンスデータを使用して、次のようなタスクが可能となります。

- 環境の作業負荷の特徴づけ
- リソースの使用量と負荷バランスの分析

- 過去のデータの傾向分析とボトルネックの分離および特定
- エラー状態への対応
- トランザクション応答時間に基づいたサービスレベルの管理
- 容量計画の実行
- システム管理問題の事前の解決

OV Performance Agent に関する包括的な説明は、『HP OpenView Performance Agent for UNIX ユーザー マニュアル』を参照してください。

インストール要件

OV Performance Agent をインストールする前に、システムがこのセクションに記載されている要件を満たしていることを確認してください。

対応しているアーキテクチャおよびオペレーティング システムのバージョン

以下の表は、対応しているアーキテクチャおよびオペレーティング システムのバージョンと、各システムで使用可能なデータ通信プロトコルをリストしたものです。

アーキテクチャ	オペレーティング システム
32 ビット Intel Pentium (x86)	<ul style="list-style-type: none">RedHat Enterprise Linux (ES/AS/WS) 2.1、3 および 4SuSE Enterprise Server 8.x、9.x および 10.xDebian 3.0 r2/r3/r4、3.1TurboLinux Server 8 および 10
64 ビット システム (x86_64)	<ul style="list-style-type: none">RedHat Enterprise Linux (ES/AS/WS) 3 および 4SuSE Enterprise Server 9.x および 10.xAsianux 2.0

ディスク スペース

OV Performance Agent は /opt/perf/ および /opt/OV/ ディレクトリにインストールされます。ログ ファイルとステータス ファイルは /var/opt/perf/ および /var/opt/OV/ ディレクトリに作成されます。

- OV Performance Agent を初めてインストールする場合は、/opt/perf/ および /opt/OV/ ディレクトリに約 50 MB のディスク スペースが必要です。
- OVPA データベースおよびステータス ファイルには、/var/opt/perf/ および /var/opt/OV/ ディレクトリに 125 MB のディスク スペースが必要です。

OV Performance Agent の初回のインストールの際、デフォルトの parm ファイルは、ログ ファイルのデータを最大で約 60 MB まで収集するように設定されています。

parm ファイルを編集して scopeux を再開始することにより、ログ ファイルのサイズを変更できます。parm ファイルの編集、parm ファイルを使用してログ ファイルのデータ記憶領域を制限および構成する方法については、『HP OpenView Performance Agent for UNIX ユーザー マニュアル』の第 2 章を参照してください。

インストールまたはアップグレードの手順

OV Performance Agent は CD-ROM インストール メディアに収録されています。ドキュメントを含む製品のサイズはおよそ 50 MB です。

アップグレード

OVPA または GlancePlus をすでにシステムにインストールしている場合は、以下のタスクを実行します。

- 実行中と思われるパフォーマンス ツールまたはプロセスをすべて終了します。詳細は、「[アクティブなパフォーマンス ツール プロセスの終了](#)」を参照してください。
- OVPA をインストールします。OpenView Performance Agent のインストールに関する詳細は、「[OV Performance Agent のインストール](#)」を参照してください。

初回インストールの場合

OV Performance Agent を初めてインストールする場合は、「[OV Performance Agent のインストール](#)」を参照してください。

アクティブなパフォーマンス ツール プロセスの終了

- 1 **root** でログインします。
- 2 アクティブなパフォーマンス ツールを確認するには、次のように入力して `perfstat` を実行します。

```
/opt/perf/bin/perfstat
```

`perfstat` によって、アクティブなパフォーマンス ツール プロセスが報告された場合は、そのプロセスを終了します (プロセスを終了する前に、それらのツールを使用していないことを確認してください)。

- 3 以前にインストールしたバージョンの OV Performance Agent が動作中の場合は、次のように入力して終了します。

`/opt/perf/bin/ovpa stop`

▶ parm, alarmdef, ttd.conf, datasources などのカスタマイズした設定ファイルやその他のカスタマイズしたログ ファイルは、新しくインストールする際に上書きされることはありません。新しい設定ファイルは `/opt/perf/newconfig/` ディレクトリにインストールされます。

- 4 `midaemon` (測定インターフェイスデーモン) が終了していることを、次のように入力して確認します。

```
ps -ef | grep midaemon
```

`midaemon` がアクティブである場合は、次のように入力して終了します。

```
/opt/perf/bin/midaemon -T
```

- 5 `ttd` (トランザクションデーモン) が終了していることを、次のように入力して確認します。

```
ps -ef | grep ttd
```

`ttd` がアクティブである場合は、次のように入力して終了します。

```
/opt/perf/bin/ttd -k
```

▶ `ttd` を終了する場合は、ARM 装備のアプリケーションをすべて終了してから、`ttd` と `OV Performance Agent` のプロセスを再開始してください。

- 6 `perfstat` コマンドを再度実行して、アクティブなパフォーマンス ツールやプロセスがないことを確認します。すべてのツールまたはプロセスを終了し、カスタマイズした設定ファイルをバックアップしたら、インストールプロセスに進みます。

OV Performance Agent のインストール

▶ Debian にインストールする場合は、14 ページの「[Debian での OV Performance Agent のインストール](#)」を参照してください。

- 1 `root` でログインしていることを確認します。
- 2 インストール CD-ROM をドライブに挿入します。

通常、Linux システムは、自動的に CD-ROM をマウントするように設定されています。CD-ROM を自動マウントするように設定されていないシステムの場合は、ファイル マネージャを使用するか、または以下のようなコマンドを入力して、手動でマウントする必要があります。

```
mount -t <type> <device> <dir>
```

ここで、<type> はファイル システムのタイプ、<device> は CD-ROM デバイス、<dir> は CD-ROM ディレクトリがマウントされる場所です。次に例を示します。

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

- 3 ターミナル ウィンドウを開きます。コマンド ラインから `ovpa.install` スクリプトを実行します。
- 4 ターミナル ウィンドウで次のように入力し、現在の作業ディレクトリを変更します。

```
cd /<directory>
```

ここで、<directory> は、ご使用の CD-ROM ディレクトリです。

- 5 `ls` と入力して、正しいディレクトリに移動しているかどうかを確認します。README ファイル、`ovpa.install` および `glance.install` スクリプト、`paperdocs`、`packages`、および `ReleaseNotes` サブディレクトリが表示されます。
- 6 次のように入力し、インストール スクリプトを実行します。

```
./ovpa.install
```

`ovpa.install` スクリプトには、さらに高度なインストールに用いるコマンド ライン オプションが含まれています。コマンドの構文は次のとおりです。

```
ovpa.install [-n] [-f] [-v]
```

コマンド ライン オプションの意味は以下のとおりです。

- n インストールの最後に、`ovpa start` に関するメッセージを表示しません。
- f 強制インストール (`--force`、`--nodeps`)
- v 詳細インストール

OV Performance Agent のインストールが完了しました。

デフォルトでは、OV Performance Agent プロセスはシステムのリブート時に自動的に開始されるように設定されています。20 ページの「[OV Performance Agent の起動と終了](#)」を参照してください。

第 2 章「OV Performance Agent の実行方法」では、ご使用の環境に合わせて OV Performance Agent の設定を行う場合の作業を説明しています。



OV Performance Agent の新機能と変更点についての詳細は、`/opt/perf/ReleaseNotes/OVPA` にあるリリース ノートを参照してください。

Debian での OV Performance Agent のインストール

Debian システムでは、まず最初に `/usr/bin/rpm` (RedHat Package Manager) をインストールする必要があります (インストールされていない場合)。

OV Performance Agent のインストールには rpm が必要です。

これまでに rpm を使用していない場合は、次の手順を実行します。

- 1 `root` でログインしていることを確認します。
- 2 次のように入力して、rpm データベースを作成します。

```
# mkdir /var/lib/rpm
# rpm --initdb --dbpath /var/lib/rpm
```

- 3 12 ページの「[OV Performance Agent のインストール](#)」の手順 2 ~ 手順 6 を行います。

OV Operations を使用している場合の OV Performance Agent のインストール

OV Operations for UNIX 8.x またはそれ以降を使用している場合、管理サーバーから Linux の管理ノードに、OV Performance Agent をインストールすることができます。

OV Operations for UNIX バージョン 8.x でのインストールの方法については、『[HP OpenView Operations システム管理リファレンス ガイド](#)』の第 4 章「[HP OpenView Performance Agent](#)」を参照してください。

仮想環境における OV Performance Agent

このバージョンの OV Performance Agent は、VMware ESX Server でサポートされます。OV Performance Agent は VMware ESX Server のサービス コンソールにインストールする必要があります。詳細は、「[サービス コンソール上の OV Performance Agent](#)」を参照してください。

このバージョンの OV Performance Agent は、VMware ESX Server 2.5.2 以降にホストされた仮想マシン (VM) の監視に対応しています。詳細は、「[仮想マシン上の OV Performance Agent](#)」を参照してください。

サービス コンソール上の OV Performance Agent

OV Performance Agent は、VMware ESX Server のサービス コンソールにインストールする必要があります。インストールの手順については、「[インストールまたはアップグレードの手順](#)」を参照してください。OV Performance Agent は、以下の内容を取り込みます。

- システム全体のパフォーマンスおよび物理的なシステム リソース使用量に関するデータ
- 各ゲストのパフォーマンスおよび物理的なシステム リソース使用量に関するデータ

個々のゲスト オペレーティング システムのパフォーマンスとリソースに関するデータを記録するために、BYLS という新しいクラスのメトリックが導入されています。



すべてのメトリックとその説明が記載されたリストは、『[Dictionary of Performance Metrics](#)』を参照してください。

仮想マシン上の OV Performance Agent

このバージョンの OVPA は、VMware ESX Server 2.5.2 以降にホストされた仮想マシン (VM) の監視に対応しています。インストール手順については、[インストールまたはアップグレードの手順](#)を参照してください。

仮想マシンに **VMWare Tools** をインストールします。仮想マシンにインストールされた OSPA (VMWare ESX Server 2.5.2 以降にホスト) では、VMWare Tools を使用して、仮想マシンをゲスト オペレーティング システムとして認識させる必要があります。

VMWare ESX Server 3.0.1 にホストされた仮想マシンでは、仮想マシンの **VMware Guest API** を有効にします。仮想マシンによるシステムの物理リソースの使用量データを取り込むために、OSPVA では **VMWare Guest API** を使用します。**VMware Guest API** の詳細については、VMWare の **VMware Guest SDK** で利用できる最新の資料を参照してください。

 すべてのメトリックとその説明が記載されたリストは、リリース ノートおよび『Dictionary of Performance Metrics』を参照してください。

以下は、仮想環境におけるすべての機能のサポート状況の一覧です。

機能	Linux	VMware ESX Server
アラーム	利用可	利用可
データ ソース統合機能	利用可	利用可
アプリケーション メトリック クラス	利用可	利用不可
BYLS メトリック	利用不可	利用可
proc メトリック	利用可	利用不可
トランザクション メトリック クラス	利用可	利用不可
テーブル クラス	利用可	利用不可
ARM	利用可	利用不可

OV Performance Agent の削除

システムから OV Performance Agent を削除する場合は、`/opt/perf/bin/` ディレクトリの `ovpa.remove` スクリプトを実行します。OV Performance Agent を削除する前に、作成されたログ ファイルおよびデータ ファイルを削除するかどうかの確認が、このスクリプトによって行われます。

`ovpa.remove` スクリプトは、次のコマンド ライン オプションを使用することで、柔軟に製品を削除できるようになっています。コマンドのオプションは次のとおりです。

ovpa.remove [-f] [-h] [-r]

- f 強制削除します。製品を削除する際、確認画面を表示しません (非インタラクティブ モード)。
メモ : このオプションは、インストールされている GlancePlus も削除します。
- h ヘルプ : このコマンドの使用に関するメッセージを表示します。
- r OVPA の設定、データ、および製品ファイルを削除します。
メモ : `f` オプションと `r` オプションを共に使用すると、確認画面を表示せずに、インストールされている Glance を含め、すべての製品、設定、およびデータ ファイルを削除します。

他の OpenView 製品でも共有されていて、他のツールの使用に必要な一部の製品パッケージは、システムにインストールされたままになることがあります。ツールがすべて削除された時点で、これらのパッケージも削除されます。

2 OV Performance Agent の実行方法

この章では、OV Performance Agent をシステムにインストールした後のプログラムの起動と実行に関する作業について説明します。次のような内容を取り上げます。

- OV Performance Agent の起動と終了
- ファイアウォールを介した通信
- OVPA ポートの設定
- 安全な通信の設定
- データソースの設定
- アラームの定義



データソース統合 (DSI: Data Source Integration) 機能を使用して、他のソースのデータを記録する場合は、『HP OpenView Performance Agent for UNIX データソース統合ガイド』を参照してください。

OV Performance Agent の起動と終了

インストールが完了すると、OV Performance Agent を起動できます。OV Performance Agent の終了、起動、または再起動が必要な場合は、ovpa スクリプトを使用して、現在実行中のプロセスを終了または再起動してください。

OV Performance Agent とそのプロセスを起動する手順は次のとおりです。

- 1 **root** としてログインします。
- 2 次のように入力します。 **/opt/perf/bin/ovpa start**

ovpa start スクリプトは、OV Performance Agent とすべてのプロセスを起動します。起動されるプロセスには、scopeux (データコレクタ)、midaemon (測定インターフェイスデーモン)、ttd (トランザクショントラッキングデーモン)、coda、ovc、ovbbccb、およびアラームジェネレータが含まれます。スクリプトを実行すると、起動されたプロセスのステータスが画面に表示されます。

ovpa スクリプトと適切なオプションを使用することで、動作中の OV Performance Agent プロセスを終了し、再起動できます。

- ovpa stop により、ttd (トランザクショントラッキングデーモン)、ovc、ovbbccb を除く、OV Performance Agent のすべてのプロセスが終了します。これらのプロセスは常に実行されている必要があります。OpenView Operations エージェントがシステムで実行されている場合、ovpa stop は coda デーモンを終了しません。



ttd を終了する必要がある場合は、実行中の ARM 装備のアプリケーションを終了してから、ttd と OV Performance Agent のプロセスを再起動してください。

- ovpa restart server は、coda を終了させ、その後再起動します。その際、一時的にアラームが無効になり、OV Performance Manager などのクライアントにアクセスできなくなります。それから、datasources ファイルを再度読み込みます。また、このオプションは、perfalarm プロセスを終了させ、その後、再起動させて、alarmdef ファイルを再度読み込みます。
- ovpa restart は、scopeux およびサーバー プロセスを一時的に終了させ、その後、再起動させます。また、parm ファイルが読み込まれ、トランザクションデーモン ttd に設定ファイル ttd.conf を再度読み込ませます。

- ovpa restart alarmにより、アラームジェネレータプロセスは alarmdef ファイルを再度読み込みます。これにより、このファイルに変更を加えた場合に、すべての OVPA プロセスを再起動することなく、新しいアラーム定義を有効にすることができます。この動作によって他のプロセスが中断されることはありません。

自動的な起動と終了

システムの再起動時に自動で OV Performance Agent を起動するプロセス、およびシステムのシャットダウン時に OVPA を終了するプロセスは、OVPA 起動およびシャットダウン スクリプトと OVPA 起動設定ファイルにより制御されます。このスクリプトと設定ファイルは、ご使用の Linux ディストリビューションに固有の要件によって、以下のように異なります。

表 1 OVPA 起動スクリプトおよび設定ファイル

Linux ディストリビューション	OVPA 起動およびシャットダウン スクリプト	OVPA 起動設定ファイル
Red Hat	/etc/rc.d/init.d/ovpa	/etc/sysconfig/ovpa
SuSE	/etc/init.d/ovpa	/etc/ovpa.conf
TurboLinux	/etc/rc.d/init.d/ovpa	/etc/sysconfig/ovpa
Debian	/etc/init.d/ovpa	/etc/ovpa.conf
ESX Server	/etc/rc.d/init.d/ovpa	/etc/sysconfig/ovpa

OVPA 起動設定ファイルには、システム起動時の OV Performance Agent の動作を制御するさまざまな環境変数が含まれています。このファイルは、OVPA システム起動およびシャットダウン スクリプトのソースファイルです。

次の環境変数およびシェル変数を編集して、OV Performance Agent のデフォルトの動作を変更することができます。

- OVPA_START は、システムを再起動したときの OV Performance Agent の自動起動を制御します。この変数は、以下の値のうちのいずれかをとります。
 - 0 システム起動時に OV Performance Agent を起動しません。
 - 1 システム起動時に OV Performance Agent を起動します。

- `OVPA_START_COMMAND` は、システムの再起動時に、**OV Performance Agent** を起動するスクリプト オプションを指定する変数です。通常、この変数は次のように設定されます。

```
/opt/perf/bin/ovpa start
```



OVPA 起動設定ファイルは、**OV Performance Agent** のパッチの適用時または更新時に上書きされないため、加えた変更はすべて保持されます。

ステータスの確認方法

OV Performance Agent を起動すると、`/var/opt/perf/` および `/var/opt/OV/` ディレクトリに数個のステータス ファイルが作成されます。`perfstat` コマンドを使用して、**OV Performance Agent** の全プロセスまたは一部のプロセスのステータスを確認できます。

次のステータス ファイルには、**OV Performance Agent** のプロセスで生じる障害を解決するのに使用される診断情報が含まれます。

```
/var/opt/perf/status.scope  
/var/opt/perf/status.perfalarm  
/var/opt/perf/status.ttd  
/var/opt/perf/status.mi  
/var/opt/OV/log/coda.txt
```

ディレクトリのサンプル

`/opt/perf/examples/` ディレクトリには、設定ファイルや構文の例、および当社のパフォーマンス ツールをカスタマイズするために使用するプログラム ファイルのサンプルが含まれています。たとえば、`/opt/perf/examples/config/` サブディレクトリには、サンプルのアラーム定義と `parm` ファイルのアプリケーション指定パラメータの例が含まれています。詳細は、`/opt/perf/examples/README` ファイルを参照してください。

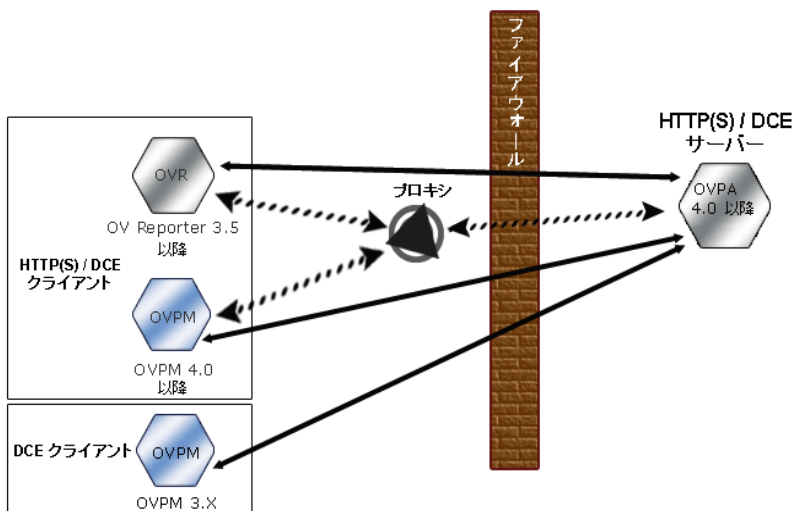
ファイアウォールを介した通信

ファイアウォールは、あるネットワークと他のネットワークの間のデータの流れをフィルタリングする方法として定義することができます。OV Performance Agent (OVPA) では、クライアントとサーバー アプリケーション間のデータ アクセスに関して、以前からパケットフィルタリング ネットワーク ファイアウォールを介した通信メカニズムに対応していましたが、今回さらに、HTTP ベースの通信インターフェイスに対応しました。OVPA は内部で Black Box Communication (BBC) コンポーネントを用いて、HTTP ベースのデータ通信を実現します。BBC コンポーネントは、HTTP または HTTPS (BBC5) をベースとした、ファイアウォール対応の単一ポートの通信インフラを提供します。HTTP ベースのインターフェイスはプロキシが使用できるため、柔軟で、ポート数が少なく済み、ファイアウォールに適しています。

2つのシステム間にネットワークファイアウォールがある場合は、OV Performance Manager および OV Reporter が OV Performance Agent システムからデータを取得します。

- ▶ 本書で使用されている OV Performance Manager (OVPM) という名前は、OVPM バージョン 4.0 以降を指します。OVPM 3.x は、OVPA for Linux システムとは接続できません。

図 1 ファイアウォール環境での OVPA との通信

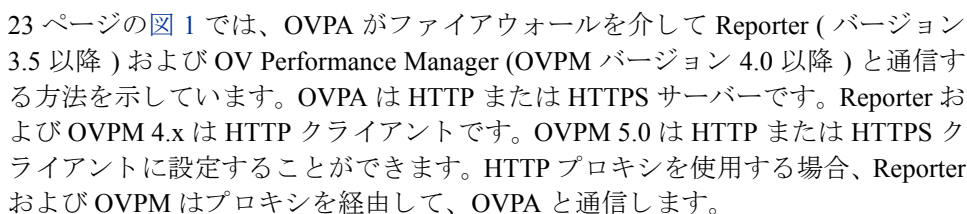


ファイアウォールを設定するためには、通信を開始するシステム (クライアント) および通信リクエストを受信するシステム (サーバー) を把握し、それに応じてファイアウォールのルールを設定することが重要です。典型的なリモート通信では、ソースポートを使用しているクライアントが、リモートシステムの宛先ポートをリッスンしているサーバーに接続します。クライアントおよびサーバーのデータフローを含む、ファイアウォール環境を把握します。以下のタスクを実行し、ファイアウォール環境での OVPA との通信を設定します。

1 OVPA ポートの設定

2 ファイアウォール環境での HTTP クライアントの設定

3 ファイアウォール設定の確認

23 ページの  図 1 では、OVPA がファイアウォールを介して Reporter (バージョン 3.5 以降) および OV Performance Manager (OVPM バージョン 4.0 以降) と通信する方法を示しています。OVPA は HTTP または HTTPS サーバーです。Reporter および OVPM 4.x は HTTP クライアントです。OVPM 5.0 は HTTP または HTTPS クライアントに設定することができます。HTTP プロキシを使用する場合、Reporter および OVPM はプロキシを経由して、OVPA と通信します。



Linux をご使用で、OVPM 3.x を OVPM 4.0 またはそれ以降にアップグレードできない場合は、extract プログラムの export 機能を使用して、ローカルで OVPA for Linux のデータにアクセスする必要があります。

OVPA ポートの設定

ファイアウォール環境の OVPA ポートは、次のいずれかの方法で設定し、確認することができます。

- 2 ポート通信の設定
- 単一ポート通信の設定
- ポート設定の確認

BBC5 を使用する OVPA システムでは、デフォルトで BBC 通信ブローカーがポート 383 を使用し、coda は動的に割り当てられたポートを使用します。

BBC 通信ブローカー用のポート設定

BBC 通信ブローカーが使用するデフォルト ポートを設定できます。ovconfchg コマンドを使用して、OV Performance Agent システムでのポート設定を変更します。以下のコマンドを入力します。

```
ovconfchg -ns bbc.cb.ports -set SERVER_PORT <port number>
ovpa restart server
```

2 ポート通信の設定

デフォルトでは、coda デーモンは、BBC 通信ブローカーが使用するポート 383 に加えて、動的に選択される 2 つ目のポートを使用します。ovconfchg コマンドを使用して、任意の well known ポートをリッスンするよう coda のポートを設定することができます。次のコマンドを入力します。

```
ovconfchg -ns coda.comm -set SERVER_PORT <portnumber>
ovpa restart server
```



ファイアウォールを介したりリモートからの OVPA への接続において、動的なポートの使用が難しくなることがあります。これは、ファイアウォールのポートが開かれるかわからない場合があるためです。

単一ポート通信の設定

OV Performance Agent システムでは、BBC 通信ブローカーがポート 383 を使用し、coda は動的に割り当てられたポートを使用します。ovconfchg コマンドを使用して、通信ブローカーが使用するポートを共有するよう coda のポートを設定することができます。次のコマンドを入力します。

```
ovconfchg -ns coda.comm -set SERVER_BIND_ADDR localhost
ovpa restart server
```

ポート設定の確認

ポート設定を確認するには、次のコマンドを入力します。

```
perfstat -d
```

次の情報が出力されます。

- coda が使用するポートのポート番号
- BBC 通信ブローカーが使用するポートのポート番号

- ポート設定
- 安全性の高い通信が有効かどうか
- coda メトリック収集が有効かどうか

次に例を示します。

```
Datacomm configuration :
```

```
-----
```

```
Coda Port                                49552 (Dynamic)
                                           Multi port Communication
BBC communication broker port            383
SSL security                              NONE
Coda Metric Collection (Prospector)      Disabled
```

ファイアウォール環境での HTTP クライアントの設定

ファイアウォール環境で HTTP クライアントを設定する方法は 2 つあります

- HTTP プロキシを用いた、HTTP クライアント (Reporter または OVPM) の設定
- HTTP プロキシを使用しない、HTTP クライアント (Reporter または OVPM) の設定

どちらの場合にも、OVPA ノードからデータにアクセスするには、HTTP サーバー (OVPA) 側のポートが 1 つだけ開いている必要があります。

HTTP プロキシを用いた、HTTP クライアント (Reporter または OVPM) の設定

ファイアウォールを介して通信する場合は、HTTP プロキシの使用をお勧めします。この方法では、環境内ですでに使用されているプロキシを用いることにより、設定が簡略化されます。プロキシが双方向で使用されている場合、ファイアウォールでは厳密に 1 つのポートを開く必要があります。OVPA により収集されたデータにアクセスするには、HTTP サーバー (OVPA) 用のポートと HTTP クライアント (Reporter および OVPM) 用のポートが開かれている必要があります。



デフォルトの 383 ポートは変更しないことをお勧めします。

HTTP プロキシを使用する場合は、Windows または UNIX 対応の Reporter および OVPM に対して、OVPA に接続するために使用するプロキシを指定する必要があります。

OVPM バージョン 5.0 以降および Reporter 3.7 を設定するには、次の手順を実行します。

次のコマンドを入力します。

```
ovconfchg -ns bbc.http -set PROXY proxy:port+(a)-(b)
```

変数 *a* および変数 *b* は、プロキシに適用するホスト名、ネットワーク、および IP アドレスをコンマで区切った形式のリストです。区切り記号「;」または「,」を使用して、1 つの PROXY キーに対して複数のプロキシを定義することができます。リストの前の「-」は、それらの要素では対象のプロキシが使用されないことを示し、リストの前の「+」は、それらの要素では対象のプロキシが使用されることを示します。最初に適合したプロキシが使用されます。

Reporter バージョン 3.6 以前および OVPM 4.x を設定するには、次の手順を実行します。

/var/opt/OV/conf/BBC/default.txt 設定ファイルを編集します。

default.txt ファイルの [DEFAULT] セクションで、PROXY に関連する行を特定し、次のように PROXY パラメータを設定します。

```
PROXY web-proxy.hp.com:8088-(localhost, *.hp.com) + (*)
```

この例では、ローカル マシン (localhost) への要求、および HP 内 (たとえば、**www.hp.com** のような、*.hp.com に合致するもの) への要求を除いて、web-proxy プロキシが、すべてのサーバー (*) に対してポート 8088 で使用されます。

HTTP プロキシを使用しない、HTTP クライアント (Reporter または OVPM) の設定

HTTP プロキシが利用できない場合は、HTTP クライアント (Reporter および OVPM システム) 上で、追加の設定が必要になります。

Reporter および OVPM for Windows が同一のシステムにインストールされていて、双方が同時に OVPA にアクセスする場合は、このセクションに記載されているポート範囲を指定します。それらが異なるシステムで実行されている場合は、それぞれに単一のポートを指定することができます。使用している OVPM および Reporter のバージョンに合わせて、次のオプションを選択します。

OVPM 5.0 以降および Reporter 3.7 を設定するには、次の手順を実行します。

次のコマンドを入力します。

```
ovconfchg -ns bbc.http -set CLIENT_PORT <port range>
```

<port range> は、使用するポート範囲です。

次に例を示します。

```
ovconfchg -ns bbc.http -set CLIENT_PORT 14000-14003
```

Reporter バージョン 3.6 以前および OVPM 4.x を設定するには、次の手順を実行します。

/var/opt/OV/conf/BBC/default.txt ファイルを、以下のように編集します。

- 1 CLIENT_PORT に適用される行を特定して、;CLIENT_PORT = の行をコメント解除します。
- 2 CLIENT_PORT パラメータに、ポート範囲を指定します。次に例を示します。
CLIENT_PORT = <port range>
<port range> は、使用するポート範囲です。次に例を示します。
CLIENT_PORT = 14000-14003

ファイアウォール設定の確認

設定を確認するには、次のコマンドを使用します。

```
ovcoda util -ping
```

このコマンドの出力には、通信設定の状態が表示されます。

複数の IP アドレスを持つシステムの設定

ご使用の環境に、複数のネットワーク インターフェイスおよび IP アドレスを持つシステムがあり、HTTP ベースの通信に専用のインターフェイスを使用する場合は、CLIENT_BIND_ADDR パラメータおよび SERVER_BIND_ADDR パラメータで、使用する IP アドレスを指定することができます。

- OVPA (サーバー) システム上に複数のネットワーク インターフェイスおよび IP アドレスがある場合は、SERVER_BIND_ADDR パラメータを次のように指定します。

```
ovconfchg -ns bbc.http -set SERVER_BIND_ADDR <IP Address>
```

- OVPA 5.0 (クライアント) システム上に複数のネットワーク インターフェイスおよび IP アドレスがある場合は、CLIENT_BIND_ADDR パラメータを次のように指定します。

```
ovconfchg -ns bbc.http -set CLIENT_BIND_ADDR <IP Address>
```

- Reporter/OVPM 4.x システム上に複数のネットワーク インターフェイスおよび IP アドレスがある場合は、CLIENT_BIND_ADDR パラメータを指定します。

/var/opt/OV/conf/BBC/default.txt ファイルを、以下のように編集します。

- a** CLIENT_BIND_ADDR に適用される行を特定して、以下の行をコメント解除します。

```
;CLIENT_BIND_ADDR =
```

- b** CLIENT_BIND_ADDR パラメータに、IP アドレスを指定します。

安全な通信の設定

OV Performance Agent は証明書ベースの安全性の高い通信、およびクライアント認証ベースの通信に対応しています。

証明書の使用

OV Performance Agent は、HP OpenView Operations 8.x 環境でのみ、証明書ベースの安全なデータ通信に対応しています。

OVO で安全性の高い通信を設定するには、『HP OpenView Operations for UNIX Firewall Concepts and Configuration Guide』を参照してください。OVO 8.x HTTPS エージェントについての詳細は、『HP OpenView Operations HTTPS Agent Concepts and Configuration Guide』を参照してください。

OVO 8.x 環境ですでに HTTPS 通信を設定している場合は、OV Performance Agent と OVPM 5.0 の間の安全な通信を設定するために、以下の変更を行います。



OV Reporter および OVPM 4.x は証明書ベースの安全性の高い通信には対応していません。

OV Performance Agent システムでは、coda に関して、SSL_SECURITY に REMOTE を設定します。次のコマンドを入力します。

```
ovconfchg -ns coda -set SSL_SECURITY REMOTE  
ovcodautl -config
```

クライアント認証の使用

オプションとして、OVPM または Reporter などの製品からのクライアント接続の認証を、OV Performance Agent で実行することができます。この認証機能を利用することで、特定の OV Performance Agent インスタンスについて、そのインスタンスへのクライアント接続を許可するホストを指定できます。

クライアント認証機能は、あらゆるバージョンの OVPM および Reporter クライアントからの接続を有効または無効にします。この機能を利用するにあたって、クライアント ソフトウェアを更新する必要はありません。

認証プロセスは、認証されるクライアントに透過的に行われるため、クライアント接続は表面上、OV Performance Agent の過去のバージョンを使用していたときと同じように処理されます。未認証のクライアントには、サービス拒否を示すメッセージが表示されます。

authip ファイルによる認証の有効化

authip というファイルが存在する場合、認証は有効になります。HTTP 通信が有効なシステムでは、authip ファイルは /var/opt/OV/conf/perf/ ディレクトリにあります。authip ファイルにはホストのリストが含まれ、そのホストからのクライアント接続が許可されます。

- authip ファイルが /var/opt/OV/conf/perf/ ディレクトリに存在する場合、クライアント接続が許可されるホストはこのファイルの内容によって決定されます。OV Performance Agent インスタンスとして同じホストで実行されているクライアントは自動的に認証されるため、エントリーを必要としません。長さがゼロの authip ファイルは、OV Performance Agent ホストで実行されているクライアントのみが接続できることを意味します。
- authip ファイルが /var/opt/OV/conf/perf/ ディレクトリに存在しない場合 (デフォルト) は、認証は行われず、OV Performance Agent の過去のバージョンと同様に、すべてのクライアントに接続が許可されます。

authip ファイルは、クライアントが OV Performance Agent によるサービスを要求して接続を試みるたびに照会されます。authip ファイルに変更が加えられた場合、変更内容を有効にするために OV Performance Agent を再起動する必要はありません。

OV Performance Agent クライアント認証機能では、authip ファイルによりネットワーク内でのクライアント エントリーが解決できるようになります。エントリーの性質によっては、DNS、NIS あるいは /etc/hosts/ ファイルによって提供されるようなネーム サービスを必要とする場合があります。

OV Performance Agent ホストから各 authip エントリーへ、「ping」が成功するかを確かめることで、これを検証することができます。クライアント認証は、authip ファイル中のクライアント エントリーが OV Performance Agent ホストからの ping に応答するという、同様の条件を備えたファイアウォールで機能します。

authip ファイルのフォーマット

authip ファイルは次のフォーマットに一致する必要があります。

- 1行あたり1つのクライアントホストをリストする
- クライアントエントリには、次のフォーマットのうち1つ(任意)を使用する
 - 完全修飾ドメイン名
 - エイリアス (alias)
 - IPアドレス(ドットで4つに区切ったIPv4アドレス)
- クライアントエントリにはスペースは使用できない
- 行の始めが「#」のラインは、コメントとして無視される
- ブランクあるいは長さがゼロの行は無視される
- IPアドレスは0から始めない。たとえば、23.10.10.10のIPアドレスを、023.10.10.10と表すことはできません。

/etc/hosts エントリは次のようになります。

```
123.456.789.1 testbox testbox.group1.thecompany.com
```

authip ファイル内で次のいずれかのエントリを使用することで、「testbox」ホストからのクライアント接続が可能となります。

```
#===== authip ファイルエントリの例 =====  
#  
# IP アドレスを使用  
123.456.789.1  
  
#  
# エイリアスを使用  
testbox  
  
#  
# 完全修飾ドメイン名を使用  
testbox.group1.thecompany.com  
  
#===== authip ファイルエントリの例の終わり =====
```


データ ソースの設定

OV Performance Agent は、coda デーモンを使用して、収集したデータをアラームジェネレータおよび分析製品に送信します。各データ ソースには、1 つのログファイル セットが含まれます。coda がアクセスするデータ ソースのリストは、`/var/opt/OV/conf/perf/` ディレクトリにある `datasources` 設定ファイルで管理されます。インストール後、初めて **OV Performance Agent** を実行する際には、デフォルトのデータ ソースである **SCOPE** が事前に設定されており、アクティブな `scopeux` ログ ファイル セットへのアクセスが提供されます。

他のデータ ソースを追加する場合は、`datasources` ファイルでそれらを設定することができます。その後、**OV Performance Agent** を再起動すると、coda デーモンにより `datasources` ファイルが読み込まれ、検出された各データ ソースの分析ツールへの通信リンクを経由して、データが利用可能になります。

datasources 設定ファイルのフォーマット

`datasources` 設定ファイルの各エントリは、1 つのログ ファイル セットで構成されるデータ ソースを示します。エントリは、データ ソースの名前とパスを示します。このフィールドでは、ログ ファイルのパス名以外は大文字小文字の区別はありません。構文は次のようになります。

```
datasource=datasource_name logfile=logfile_set
```

- **datasource** はキーワードです。**datasource_name** はデータ ソースを識別するための名称です。たとえば、データ ソース名はアラーム定義や分析ソフトウェアで使用されます。また、固有の名前である必要があります。これらの名前は大文字に変換されます。さらに、データ ソース名は最大 64 文字まで指定できます。
- **logfile** はキーワードです。また、**logfile_set** はログ ファイル (`dsilog` または `scope` プロセスにより作成され、`.log` で終わるファイル) を識別する完全修飾名で、大文字小文字の区別があります。

次に `datasources` ファイルのデータ ソース エントリを 2 例示します。

```
datasource=SCOPE logfile=/var/opt/perf/datafiles/logglob  
datasource=ASTEX logfile=/tmp/dsidemo/log/astex/ASTEX_SDL
```

datasources の更新後、ovpa restart server コマンドを使用して、coda デーモンを再開始します。まず、ovpa restart server により、すべての coda プロセスが終了されます。次に、coda デーモンが開始され、datasources ファイルを再読み込みして、このファイルに加えられた変更内容を取り込みます。

coda デーモンプロセスを終了すると、OV Performance Manager の現行の接続がすべて切断されることに注意してください。たとえば、データソースでグラフを描いていて、さらに別のグラフを描く場合には、coda デーモンが再起動されたときに、OV Performance Manager のデータソースを再選択して、接続を確立しなおす必要があります。

/var/opt/OV/log/coda.txt ファイルの内容を見て、coda デーモンがアクティブになったことを確認するか、またはエラーメッセージを参照します。

データソースの削除

OV Performance Manager からのログファイルデータを表示したり、またはログファイルのアラームを処理したりする必要がない場合は、OV Performance Agent 設定ディレクトリ /var/opt/OV/conf/perf/ の datasources ファイルを編集して、データソースやログファイルセットへのパスを削除することができます。その後、ovpa プログラムを実行して、前出の「[データソースの設定](#)」の手順で coda デーモンを再開始します。

データが必要ではなくなったときには、ログファイルセットを削除することもできます。datasources から、データソースは削除せずに、ログファイルセットを削除すると、coda により、そのデータソースは省略されます。

データのログファイルセットへの記録を停止する際に、coda デーモンを開いたままにしておくことが可能です。これにより、OV Performance Manager で履歴データを表示することができます。この場合は、datasources ファイルからデータソースを削除せずに、dsilog プロセスを停止します。

parm ファイル

parm ファイルはテキスト ファイルです。ログ ファイルの最大サイズや対象プロセスのしきい値の定義、アプリケーション定義といった scopeux データ コレクタの設定を指定します。ファイル中のコメントで、各種設定の概要を確認することができます。

OV Performance Agent では、parm ファイルは /opt/perf/newconfig/ ディレクトリに存在し、インストール時に /var/opt/perf/ ディレクトリにコピーされます。parm ファイルとそのパラメータに関する詳細は、『HP OpenView Performance Agent for UNIX ユーザー マニュアル』の第 2 章、「parm ファイル」セクションを参照してください。

アラームの定義

パフォーマンスをモニタするためにアラームを使用する場合、OV Performance Agent の alarmdef ファイルにあるアラーム定義セット内でアラーム発生条件を指定します。OV Performance Agent を初めてインストールした場合、alarmdef ファイルにはデフォルトのアラーム定義セットが含まれています。このデフォルトの定義を使用するか、または必要な場合はカスタマイズすることが可能です。

アラームを定義する手順については、『HP OpenView Performance Agent for UNIX ユーザー マニュアル』の「パフォーマンス アラーム」の章を参照してください。この章では、アラーム定義の構文、アラームの機能、パフォーマンスをモニタするためのアラームの使用方法についても説明しています。

3 ドキュメント

この章では、以下のトピックに関する OV Performance Agent のオンラインおよび印刷可能なドキュメントについて記載します。

- [OV Performance Agent ドキュメント一覧](#)
- [Web 上でのドキュメントの参照](#)
- [Adobe Acrobat ファイル](#)

OV Performance Agent ドキュメント一覧

OV Performance Agent ソフトウェアには、表示可能なファイルフォーマットと印刷可能なファイルフォーマットで設定される標準の OV Performance Agent ドキュメント一式が含まれています。Adobe Acrobat フォーマット (*.pdf) のドキュメントはオンラインで表示され、必要な場合は印刷できます。ASCII テキスト (*.txt) のドキュメントも印刷可能です。vi などの UNIX テキスト エディタを使用すると、画面上でテキストファイルを表示できます。次の表で、ドキュメント、使用されているファイル名、およびオンラインのパスを示します。

表 2 **OV Performance Agent ドキュメント一覧**

ドキュメント	ファイル名	パス
HP OpenView Performance Agent for Linux システム インストール、設定ガイド	ovpainst.pdf	/opt/perf/ paperdocs/ovpa/C
HP OpenView Performance Agent for UNIX ユーザー マニュアル	ovpausers.pdf	/opt/perf/ paperdocs/ovpa/C
HP OpenView Performance Agent for UNIX データ ソース統合ガイド	ovpadsi.pdf	/opt/perf/ paperdocs/ovpa/C
HP OpenView Performance Agent for UNIX および GlancePlus トランザクション追跡	tyt.pdf	/opt/perf/ paperdocs/arm/C
アプリケーション応答測定 API ガイド	armapi.pdf	/opt/perf/ paperdocs/arm/C
HP OpenView Performance Agent for Linux Metric Definitions	metlinux.txt metlinux.htm	/opt/perf/ paperdocs/ovpa/C
OVPA metrics list by Data Class for all operating systems	mettable.txt	/opt/perf/ paperdocs/ovpa/C

Web 上でのドキュメントの参照

ここにリストしている .pdf および .htm ドキュメントは、以下の HP OpenView マニュアルの Web サイトでも参照できます。

http://ovweb.external.hp.com/lpe/doc_serv

製品リスト ボックスから **Performance Agent** を選択し、リリース バージョン、OS、およびマニュアルのタイトルを選択します。**[Open]** をクリックして、オンラインでドキュメントを表示します。または **[Download]** をクリックして、ご使用のコンピュータにファイルを配置します。

Adobe Acrobat ファイル

Adobe Acrobat ファイルは、Acrobat 7.0 で作成されました。表示する場合は、Adobe Acrobat Reader バージョン 4.0 以降を使用してください。Web ブラウザに Acrobat Reader がインストールされていない場合は、Adobe の Web サイトからダウンロードできます。

<http://www.adobe.com>

Acrobat Reader でドキュメントを表示している間は、1 ページ、複数ページ、または全ページ分のドキュメントを印刷できます。

Linux で .PDF ファイルを表示するには、次のように入力します。

`acroread filename.pdf`

A coda の設定

はじめに

この付録には、coda および BBC 通信ブローカー ovbbccb を設定するためのオプションのリストが記載されています。OV Performance Agent 用の設定を行うには、ovconfchg ツールを使用します。

DISABLE_PROSPECTOR

このオプションでは、OVO エージェントおよび OVPA の両方がインストールされている場合の、coda を介したデータ収集について設定します。デフォルト値は false です。フォーマットは次のようになります。

ovconfchg -namespace coda -set DISABLE_PROSPECTOR < 値 >

- true: coda は、coda データ ソース用のデータを収集しません。
- false: coda は、coda データ ソース用のデータを収集します。

RESPONSE_SIZE_LIMIT

このオプションは、coda デーモンによってクエリ応答に割り当てられるメモリの最大容量を指定します。デフォルト値は 104857600 (100 メガバイト) です。フォーマットは次のようになります。

ovconfchg -namespace coda -set RESPONSE_SIZE_LIMIT < 値 >



指定した制限を超えた場合は、次のエラー メッセージが表示されます。

```
coda_out_of_resource
```

SSL_SECURITY

このオプションは、`coda` を介した安全性の高い通信を有効にします。デフォルト値は `NONE` です。フォーマットは次のようになります。

```
ovconfchg -namespace coda -set SSL_SECURITY <値>
```

- `NONE`: `coda` は、ローカルクライアントまたはリモートクライアントに対して `SSL` 接続を要求しません。
- `REMOTE`: `coda` は、すべてのリモート接続に対して `SSL` を要求します。
- `ALL`: `coda` は、すべての接続 (ローカルおよびリモート) に対して `SSL` を要求します。

SERVER_BIND_ADDR

このオプションでは、サーバーポートのバインドアドレスを指定します。デフォルト値は `localhost` です。値が `localhost` に設定されていると、すべてのクライアントは `ovbbccb` に接続し、`ovbbccb` によって要求が `coda` に転送されます。フォーマットは次のようになります。

```
ovconfchg -namespace coda.comm -set SERVER_BIND_ADDR  
localhost
```

LOG_SERVER_ACCESS

このオプションを使用すると、サーバーへのアクセスを有効 / 無効にできます。このオプションが `true` に設定されている場合、通信ブローカーの `BBC` はサーバーへのアクセスをすべて記録します。記録される情報には、送信側 `IP` アドレス、要求された `HTTP` アドレス、要求された `HTTP` メソッドおよび応答ステータスが含まれます。通常、この値は変更しません。

```
ovconfchg -namespace coda.comm -set LOG_SERVER_ACCESS false
```

PROXY

このオプションではプロキシの設定を行います。フォーマットは次のようになります。

```
ovconfchg -ns bbc.http -set PROXY proxy:port+(a)-(b)
```

ここで、変数 `a` および変数 `b` は、プロキシに適用するホスト名、ネットワーク、および `IP` アドレスをコンマで区切った形式のリストです。

区切り記号「;」または「,」を使用して、1つの PROXY キーに対して複数のプロキシを定義することができます。リストの前の「-」は、それらの要素では対象のプロキシが使用されないことを示し、リストの前の「+」は、それらの要素では対象のプロキシが使用されることを示します。最初に適合したプロキシが使用されます。

次に例を示します。

```
ovconfchg -ns bbc.http -set PROXY srv1.abc.com:8088+*
```

```
ovconfchg -namespace coda.comm -set SERVER_PORT 0
```

SERVER_PORT

BBC 通信ブローカーが使用するデフォルト ポートを設定できます。ovconfchg ツールを使用して、OV Performance Agent システムでのポート設定を変更します。以下のコマンドを入力します。

```
ovconfchg -ns bbc.cb.ports -set SERVER_PORT <port number>
```

```
ovconfchg -namespace coda.comm -set SERVER_PORT <port number>
```


用語集

alarmdef ファイル

アラームの状態を特定するアラーム定義を含むファイルです。

coda デーモン

収集したデータをアラーム ジェネレータや scopeux ログ ファイルまたは DSI ログ ファイルを含むデータ ソースの分析製品に提供するデーモンです。coda は detasources 設定ファイルにリストされているデータ ソースから、データを読み込みます。

datasources ファイル

/var/opt/OV/conf/perf/ ディレクトリにある設定ファイルです。ファイル内の各エントリは単一のログ ファイル セットから成る scopeux または DSI データ ソースを示しています。「perflbd.rc」、「coda」、および「データ ソース」も参照してください。

datasources 設定ファイル

/var/opt/OV/conf/perf/ ディレクトリにある設定ファイルです。ファイル内の各エントリは単一のログ ファイル セットから成る scopeux または DSI データ ソースを示しています。

default.txt

HP OpenView アプリケーションで通信パラメータをカスタマイズするために使用する通信設定ファイルです。

DSI

「データ ソース統合 (DIS)」を参照してください。

DSI ログ ファイル

OV Performance Agent の DSI プログラムによって作成される、自己記述型のデータを含むログ ファイルです。

extract

OV Performance Agent のプログラムの 1 つで、生のログ ファイルまたは事前に抽出したログ ファイルからデータを抽出（コピー）し、抽出ログ ファイルに書き込みます。また、分析プログラムで使用するために、データを書き出すことができます。

Glance

Glance (GlancePlus) は、ユーザー端末やワークステーションに現在のパフォーマンス データを直接表示するオンラインの診断ツールです。システムのパフォーマンス障害が発生した場合は、障害の識別と解決の支援ツールとして機能します。

logappl

ユーザーが定義した各アプリケーションのプロセスの測定値を含む生ログ ファイルです。

logdev

ディスクや netif などの個別デバイスのパフォーマンスの測定値を含む生ログ ファイルです。

logglob

システム全体の作業負荷、つまりグローバルな作業負荷の測定値を含む生ログ ファイルです。

logindx

他のログ ファイルに含まれるデータにアクセスする際に必要となる追加情報を含む生ログ ファイルです。

logproc

選択された「対象プロセス」の測定値を含む生ログ ファイルです。プロセスが初めて作成されたとき、プロセスが終了したとき、および CPU 使用量、ディスク使用量、応答時間などに対してユーザーが定義したしきい値を超えたときに、プロセスは「対象プロセス」となります。

logtran

トランザクション データの測定値を含む生ログ ファイルです。

midaemon

トレースデータを、メモリベースの MI パフォーマンス データベースを使用してカウンタを保持する Measurement Interface カウンタ データに変換する、OV Performance Agent プログラムです。このデータベースは、scopeux のようなコレクタ プログラムによりアクセスされます。

ovbbccb

ovc で制御される通信をベースとした HTTP(S) 用の OpenView Operations 通信ブローカーです。「**coda**」および「**ovc**」も参照してください。

ovc

OpenView Operations の制御プロセスおよび監視プロセスです。スタンドアロン OVPA のインストールでは、ovc によって coda および ovbbccb が監視および制御されます。OpenView Operations for UNIX 8.x エージェントがインストールされたシステムに OVPA をインストールする場合も、ovc によって OpenView Operations for UNIX 8.x プロセスが監視および制御されます。「**coda**」および「**ovbbccb**」も参照してください。

ovpa スクリプト

scopeux データ コレクタ、アラーム ジェネレータ、ttd、midaemon、および coda などの OV Performance Agent のプロセスを開始、終了、再開するオプションを含んだ OV Performance Agent スクリプトです。ovpa の man ページも参照してください。

OV Performance Manager (OVPM)

マルチベンダー分散型ネットワーク システムの統合型パフォーマンス管理を提供します。1 台のワークステーションを使用して、数十から数千のノード サイズに及ぶネットワーク上の環境パフォーマンスをモニタします。

parm ファイル

データ収集方法をカスタマイズするために、scopeux により使用されるパラメータを含む OV Performance Agent のファイルです。

perflbd.rc

/var/opt/perf/ ディレクトリにある設定ファイルです。このファイルは、datasources ファイルへのシンボリック リンクとして管理されます。「**datasources ファイル**」も参照してください。

perfstat

システムにインストールされたパフォーマンス プロセスの全ステータスを表示するプログラムです。

run ファイル

scopeux プロセスが動作中であることを示すために、scopeux コレクタによって作成されます。run ファイルを削除すると scopeux は終了します。

scopeux

パフォーマンス データを収集した後、分析または記録用として生ログ ファイルにデータを書き込む (記録する) OV Performance Agent データ収集プログラムです。「**生ログ ファイル**」も参照してください。

scopeux ログ ファイル

「**生ログ ファイル**」を参照してください。

status.scope ファイル

ステータス、データの不整合、およびエラーを記録するために、scopeux コレクタによって作成されます。

ttd.conf

トランザクション名、パフォーマンス分布範囲、サービス レベルの目標など、各トランザクションおよび追跡される情報を定義するトランザクション設定ファイルです。

utility

OV Performance Agent のプログラムの 1 つで、生ログ ファイルと抽出ログ ファイルのオープン、リサイズ (サイズ変更)、スキャン、およびレポート作成を可能にします。このプログラムを使用すると、parm ファイルと alarmdef ファイルの構文が確認でき、過去のログ ファイル データのアラーム情報が取得できます。

アプリケーション

関連のプロセスまたはプログラム ファイルのユーザー定義グループです。アプリケーションは、パフォーマンス ソフトウェアがパフォーマンス メトリックを収集し、プロセスおよびプログラムのアクティビティを一括して報告できるように定義されます。

アプリケーション ログ ファイル

「logappl」を参照してください。

アラーム

パフォーマンスがユーザー指定のアラームの基準に一致した時間、またはその基準を超えた時間を意味します。アラーム情報は、OVPM などの分析システムや OVO に送信されます。utility プログラムを使用すると、過去のデータ ログ ファイルでアラームを識別できます。

アラーム ジェネレータ

各アラーム情報間の通信を処理します。これは、perfalarm および agdb データベースで構成されます。agdb データベースは、アラームが送信される OV Performance Manager 分析ノード (存在する場合) のリスト、およびアラーム情報が送信される時間と場所を定義するために必要な各種の on または off フラグで構成されます。

グローバル

システム全体を意味する修飾子です。

グローバル ログ ファイル

「logglob」を参照してください。

システム識別番号

システムを識別する文字列です。デフォルトは `uname -n` で返されるホスト名です。

対象プロセス

プロセスが初めて作成されたとき、プロセスが終了したとき、および CPU 使用量、ディスク使用量、応答時間などに対してユーザーが定義したしきい値を超えたときに、プロセスは「対象プロセス」となります。

抽出ログ ファイル

extract プログラムによって作成されるログ ファイルです。ユーザーが選択したデータの範囲とデータ型を含みます。抽出ログ ファイルは、ワークステーションの分析ツール、OVPM により最適アクセスが可能になるようにフォーマットされています。このファイルのフォーマットは、extract プログラムと utility プログラムへの入力に最適であり、パフォーマンス データのアーカイブにも適しています。

データ ソース

単一の scopeux または DSI ログ ファイル セット内の 1 つ以上のデータ クラスで構成されます。たとえば、OV Performance Agent の SCOPE データ ソースはグローバルなデータから成る scopeux ログ ファイル セットです。「**datasources ファイル**」も参照してください。

データ ソース統合 (DSI)

OV Performance Agent がアプリケーション、データベース、ネットワーク、他のオペレーティング システムなどの外部ソースからデータを受信し、記録して、アラームを検出することを可能にする技術です。

デバイス

システムに接続されている入出力デバイスです。ディスクドライブ、テープドライブ、CD-ROM ドライブ、プリンタ、ユーザー端末などは共通デバイスです。

デバイス ログ ファイル

「**logdev**」を参照してください。

トランザクション追跡

情報技術 (IT) のリソース マネージャによるビジネス アプリケーション トランザクションのエンドツーエンドの応答時間の測定を可能にする、OV Performance Agent で使用されている技術です。

トランザクション ログ ファイル

「**logtran**」を参照してください。

生ログ ファイル

scopeux が収集するシステム データの要約された測定値を含むファイルです。「**logappl**」、「**logproc**」、「**logdev**」、「**logtran**」、および「**logindx**」を参照してください。

プロセス

プログラム ファイルを実行します。対話型のユーザー (normal、nice または realtime のいずれかの優先順位で実行中のプロセス) またはオペレーティング システムのプロセスを示します。

プロセス ログ ファイル

「logproc」を参照してください。

リアルタイム

イベントが発生する実際の時間です。

リサイズ

utilityプログラムのresizeコマンドを使用して、ログファイル全体のサイズを変更します。

ログ ファイル セット

単一のソースから収集されたデータからなるファイルの集まりです。

索引

A

authip ファイル, 31
 フォーマット, 32
 例, 32

C

coda.log ファイル, 34
coda デーモン, 33, 34

D

datasources 設定ファイル, 33
 フォーマット, 33
DSI データ ソース, 33

E

extract プログラム, 33

O

OV Operations
 OVPA のインストールに使用, 14

OVPA

 インストール, 12, 14
 起動, 20
 起動設定ファイル, 21
 起動と終了, 20
 削除, 17
 システム起動およびシャットダウン スクリプト, 21
 終了, 11
 スクリプト, 20
 ステータス ファイル, 22

ovpa.install スクリプト, 13

ovpa restart スクリプト, 20

OVPA_START_COMMAND 変数, 22

OVPA_START 変数, 21

ovpa stop スクリプト, 12

ovpa スクリプト
 ovpa start, 20

OVPA の起動と終了, 20

OVPA の削除, 17

OVPA の終了, 11

OV Performance Agent

 OVO を使用したインストール, 14

P

parm ファイル, 35

perfstat コマンド, 11

S

scopeux

起動, 20

データ ソース, 33

scopeux ログ ファイル セット, 33

SCOPE デフォルト データ ソース, 33

あ

アラーム ジェネレータ、開始, 20

い

印刷可能なドキュメント ファイル, 38

インストール

OVPA, 12, 14

手順, 11

要件, 9

インストールの手順

OVO を使用した OVPA のインストール,
14

インストール前にプロセスを終了, 11

インストール要件

ハードウェア, 9

か

環境変数, 21

き

起動

OVPA, 20

scopeux, 20

く

クライアント認証, 31

す

スクリプト

ovpa.install, 13

ovpa.remove, 17

ovpa restart alarm, 21

ovpa stop, 20

OVPA システム起動およびシャットダウン,
21

ステータス ファイル, 22

coda.log, 22

status.mi, 22

status.perfalarm, 22

status.scope, 22

status.ttd, 22

せ

設定, 33

データ ソース, 33

て

ディスク スペース、要件, 9

ディレクトリのサンプル

README, 22

データ ソース, 33

DSI, 33

SCOPE, 33

scopeux, 33

SCOPE デフォルト データ ソース, 33

削除, 34

データ ソースの削除, 34

と

ドキュメント, 37

Adobe Acrobat ファイル, 40

OVPA ドキュメント一覧, 38

Web で参照, 38

ふ

ファイアウォール

- OVPA 通信設定の概要 , 24
- ファイアウォールを介した通信 , 23
- 複数の IP アドレスを持つシステム , 28

ファイアウォール、通信 , 23

ファイル

- Adobe Acrobat, 40
- authip, 31
- coda.log, 34
- datasources, 33
- OVPA 起動設定 , 21
- parm, 35
- status.scope, 22

へ

変数 , 21

- OVPA_START, 21
- OVPA_START_COMMAND, 22
- 環境 , 21

