



ioDrive2

PCIe SSD-365GB

PCIe SSD-785GB

PCIe SSD-1.2TB

ioDrive2 Duo

PCIe SSD-Duo 2.4TB

Linux 向け

ioMemory VSL ユーザー ガイド

2014 年 1 月

富士通株式会社

著作権および商標

Copyright © 2014 Fujitsu Technology Solutions GmbH.

All rights reserved.

お届けまでの日数は在庫状況によって異なります。技術的修正の権利を有します。

使用されているハードウェア名およびソフトウェア名は、各社の商標です。

- 本書の内容は、改善のため事前連絡なしに変更することがあります。
- 本書に記載されたデータの使用に起因する、第三者の特許権およびその他の権利の侵害については、当社はその責を負いません。
- 無断転載を禁じます。

Microsoft、Windows、Windows Server、および Hyper V は、米国およびその他の国における **Microsoft Corporation** の商標または登録商標です。

Intel、インテルおよび Xeon は、米国およびその他の国における **Intel Corporation** またはその子会社の商標または登録商標です。

Fusion-io、Fusion-io のロゴ、ioMemory、Virtual Storage Layer、VSL、Flashback、Flashback Protection、ioManager、および ioDrive は、Fusion-io, Inc. の米国およびその他の国における商標または登録商標です。

このドキュメントに記載されているその他の組織および製品の名称は、各所有者の商標またはサービス マーク（該当する場合）です。明記されていない限り、このドキュメントに記載されている他のいかなる組織や製品とも一切関係ありません。

本書をお読みになる前に

安全にお使いいただくために

本書には、本製品を安全に正しくお使いいただくための重要な情報が記載されています。

本製品をお使いになる前に、本書を熟読してください。特に、添付の『安全上のご注意』をよくお読みになり、理解されたうえで本製品をお使いください。また、『安全上のご注意』および当マニュアルは、本製品の使用中にいつでもご覧になれるよう大切に保管してください。

電波障害対策について

この装置は、クラスA 情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

VCCI-A

アルミ電解コンデンサについて

本製品のプリント板ユニットやマウス、キーボードに使用しているアルミ電解コンデンサは寿命部品であり、寿命が尽きた状態で使用し続けると、電解液の漏れや枯渇が生じ、異臭の発生や発煙の原因になる場合があります。

目安として、通常のオフィス環境（25℃）で使用された場合には、保守サポート期間内（5年）には寿命に至らないものと想定していますが、高温環境下での稼働等、お客様のご使用環境によっては、より短期間で寿命に至る場合があります。寿命を超えた部品について、交換が可能な場合は、有償にて対応させていただきます。なお、上記はあくまで目安であり、保守サポート期間内に故障しないことをお約束するものではありません。

ハイセイフティ用途での使用について

本製品は、一般事務用、パーソナル用、家庭用、通常の産業用等の一般的用途を想定して設計・製造されているものであり、原子力施設における核反応制御、航空機自動飛行

制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療器具、兵器システムにおけるミサイル発射制御など、極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、直接生命・身体に対する重大な危険性を伴う用途（以下「ハイセイフティ用途」という）に使用されるよう設計・製造されたものではございません。お客様は、当該ハイセイフティ用途に要する安全性を確保する措置を施すことなく、本製品を使用しないでください。ハイセイフティ用途に使用される場合は、弊社の担当営業までご相談ください。

瞬時電圧低下対策について

本製品は、落雷などによる電源の瞬時電圧低下に対し不都合が生じることがあります。電源の瞬時電圧低下対策としては、交流無停電電源装置などを使用されることをお勧めします。

（社団法人電子情報技術産業協会(JEITA)のパーソナルコンピュータの瞬時電圧低下対策ガイドラインに基づく表示)

外国為替及び外国貿易法に基づく特定技術について

当社のドキュメントには「外国為替及び外国貿易法」に基づく特定技術が含まれていることがあります。特定技術が含まれている場合は、当該ドキュメントを輸出または非居住者に提供するとき、同法に基づく許可が必要となります。

目次

はじめに	9
概要	9
ioMemory デバイスについて	9
ioMemory Virtual Storage Layer (VSL) について	9
Flashback Protection テクノロジーについて	10
ソフトウェアのインストール	11
インストール概要	11
ドライバのインストール	11
RPM パッケージのインストール	14
ioMemory VSL ドライバーの読み込み	16
ioMemory VSL の読み込みの制御	16
ioMemory デバイスとマルチパス ストレージ	19
ioMemory VSL の読み込み解除のハンドリング	19
ioMemory VSL オプションの設定	19
一度限りの構成	19
永続的な構成	20
ファームウェアのアップグレード	20
VMware のゲスト OS のアップグレード	21
PCIe 電力の有効化	21
デバイスのスワップとしての使用	21
論理ボリューム マネージャーの使用	22
RAID の構成	22
RAID 0	22
アレイの永続化 (再起動後に存在)	23
RAID 1	24
RAID 10	24
複数のデバイスにわたる RAID 10 の構築	24
Discard (TRIM) 機能のサポートについて	25
Linux における Discard (TRIM) 機能	25

メンテナンス	27
デバイスの LED インジケータ	27
GUI 管理	28
コマンドライン ユーティリティ.....	28
PCIe 電力優先 (PCIe Power Override) の有効化.....	29
上書きパラメータ (override parameter) の有効化	30
一般的なメンテナンス作業	31
ioMemory VSL ドライバーの読み込み解除	32
ioMemory VSL の RPM パッケージのアンインストール.....	32
ioMemory VSL ユーティリティとその他のサポート パッケージのアンインストール	32
2.x サポート パッケージのアンインストール	32
3.x サポート パッケージのアンインストール	32
自動接続 (auto-attach) の無効化	33
突然のシャットダウンの問題.....	33
ioMemory VSL の無効化	33
性能とチューニング	35
DVFS の無効化	35
ACPI C ステートの制限	35
ACPI C ステート オプションの設定	36
Linux における C ステート.....	36
NUMA アフィニティの設定	36
割り込みハンドラーのアフィニティの設定	36
付録 A – ユーティリティ リファレンス.....	38
fio-attach	38
fio-beacon.....	39
fio-bugreport.....	40
fio-detach	41
fio-format	42
fio-pci-check	43
fio-snmp-agentx.....	44
fio-status.....	44


fio-sure-erase	47
クリアのサポート.....	49
ページのサポート	49
fio-update-iodrive	50
付録 B – ioMemory デバイスの正常性の監視.....	52
NAND フラッシュとコンポーネントの障害	52
正常性ステータスの指標	52
正常性(ヘルスマニターリング)の監視手法.....	53
ソフトウェア RAID と正常性監視	53
付録 C – モジュール パラメータの使用	55
付録 D – SNMP の設定	57
SNMP マスター エージェント.....	57
SNMP マスター エージェントの起動	57
マスター エージェントの構成	57
Fusion-io の SNMP AgentX サブエージェント	58
Fusion SNMP サブエージェントのインストール	58
Fusion-io の SNMP サブエージェントの実行と構成	58
Fusion-io の SNMP サブエージェントの手動実行.....	59
サブエージェントのログ ファイル.....	59
SNMP サンプル構成ファイルの使用	60
SNMP テスト モードの有効化	60
SNMP MIB のサポート.....	63
付録 E – SMI-S インターフェイス.....	65
SMI-S インターフェイスの概要	65
Linux への SMI-S WMI プロバイダーのインストール	66
Linux のテスト.....	67
解説.....	68
実装.....	70
通知.....	78
付録 F – NUMA 構成	81
NUMA アーキテクチャについて	81

numa_node_override パラメータ	81
高度な構成の例.....	82
付録 G – VSL 2.x から 3.x へのデバイスのアップグレード	84
アップグレード手順	85
付録 H – 付属資料の使用許諾	89
AVR Bootloader	89
tree.h	89

はじめに

概要

富士通のソリッドステート ストレージ デバイスをご購入いただきありがとうございます。このガイドでは、ioMemory デバイス用ソフトウェアのインストール、トラブルシューティング、およびメンテナンスの方法について説明します。

 このマニュアルで **ioMemory デバイス** と記載されている箇所は、実際にお使いのデバイス (ioDrive2 デバイス、または ioDrive2 Duo デバイスを構成する 2 つの ioMemory デバイスのいずれかなど) に置き換えてお読みください。

複数のデバイスを備えた製品

ioDrive2 Duo デバイスは、実際には**複数の ioMemory デバイス** で構成されています。製品が複数の ioMemory デバイスで構成されている場合は、それぞれの ioMemory デバイスを独立したデバイスとして管理します。

たとえば、PCIe SSD-Duo 2.4TB を使用する場合は、2 つの ioMemory デバイスをそれぞれ独立して接続、切断、およびフォーマットできます。これらの 2 つのデバイスは、それぞれ独立したデバイスとしてシステムに認識されます。

ioMemory デバイスについて

ioMemory デバイスは、革新的なシリコンベースのストレージ アーキテクチャに基づいて設計された世界最先端の NAND フラッシュ ストレージ デバイスであり、DRAM に匹敵する性能と、今日のハード ディスクと同等のストレージ容量を備えているため、ストレージの性能が飛躍的に高まります。ioMemory デバイスにより、あらゆるコンピューターにおいてエンタープライズ SAN をしのぐ I/O 性能を発揮できます。

ioMemory デバイスは、I/O 負荷の高いアプリケーションにおける帯域幅性能の改善を目的として設計されたデータ アクセラレーターです。ioMemory デバイスは、今日のデータ センターが直面している難しい課題に対して理想的な解決策を提示する、他に類のないソリューションです。

ioMemory Virtual Storage Layer (VSL) について

ioMemory® Virtual Storage Layer™ (VSL: 仮想ストレージ レイヤー) は、単なるハードウェア ドライバーではなく、ioMemory デバイスが優れた性能を発揮するために必要不可欠な技術です。VSL™ は、RAM 仮想化サブシステムとディスク I/O サブシステムの両方の利点を取り入れた、ハイブリッド型のサブシステムです。一見すると、ブロックベースのアプリケーションやソフトウェアとのインターフェイスとして動

作するディスクのように見えますが、実際には、最大の性能を発揮するために RAM のような動作をしています。これにより、次のような大きなメリットがあります。

- 性能: VSL では、複数の CPU コアに直接かつ並列にアクセスできるため、ほぼニアな性能スケールが可能で、さまざまな読み書き負荷において安定した性能を発揮できます。また、割り込みやコンテキスト スイッチも最小限に抑えられるため、レイテンシーが短くなります。
- 拡張性: VSL を利用すると、フラッシュ最適化に対応したソフトウェア開発が可能になります。個々の ioMemory モジュールを、フラッシュ最適化されたデータ センター構築のための構成要素として柔軟に使用できます。

単なる追加ディスク ドライブ以上の役割を果たすフラッシュ ストレージ

他社提供の SSD では、PCIe カード型でも、ドライブ ベイ型でも、フラッシュを RAID コントローラー管理下の単なる追加のディスク ドライブとして扱っています。このアプローチには、次のような制約があります。

- 性能と信頼性が低下する
- レイテンシーが長くなり、複雑性が増す
- フラッシュ ストレージ メディアに関連するソフトウェア開発および最適化で発揮できる性能が制限される

ioMemory デバイスは、ioMemory VSL を利用することによって、このような制約のあるアプローチを回避し、性能向上と最適化の可能性を広げています。

Flashback Protection テクノロジーについて

NAND フラッシュは、他の多くのメモリー デバイスと同様、長年使用するうちにいつかは障害が発生します。発生する障害には、永続的なものと一時的なものがあります。Fusion Flashback™の冗長性機能は、永続的な障害が発生している ioMemory チップへの対策として設計されたものであり、ソフト エラーに対して ECC(誤り訂正符号)を超える保護機能を提供します。

Flashback テクノロジーは、RAID によく似たリアルタイムの冗長性をチップレベルで提供し、容量や性能を犠牲にすることなく耐障害性を実現します。物理的な RAID スキームを使用して冗長性や保護を実現しようとするソリューションでは、一般に、容量(RAID 1)または性能(RAID 5)を犠牲にせざるを得ません。

自己修復特性を備えた Fusion Flashback Protection™テクノロジーでは、他のどのフラッシュ ソリューションよりも高い性能、低い障害発生率、長い耐用性が保証されます。

ソフトウェアのインストール

このソフトウェアのインストールに進む前に、次の内容を確認してください。

1. ioMemory VSL をインストールする前に、ioMemory デバイスが適切に取り付けられていることを確認します。詳細およびハードウェア要件については、『ioMemory ハードウェア インストールガイド』を参照してください。

i すべてのコマンドには、管理者特権が必要です。sudo を使用するか、または「root」としてログインし、インストールを実行してください。

インストール概要

1. 以前のバージョンの ioDrive デバイスが取り付けられているシステムにこのバージョンの ioMemory VSL をインストールする場合は、「[付録 G - VSL 2.x から 3.x へのデバイスのアップグレード](#)」の手順に慎重に従う必要があります。

i 3.x.x 用のファームウェアにデバイスをアップグレードする必要はないが、システムに以前のバージョンの ioMemory VSL がインストールされている場合は、ioMemory VSL パッケージおよびユーティリティをアンインストールする必要があります。手順については、「[一般的なメンテナンス作業](#)」を参照してください。パッケージをアンインストールした後、このページに戻ります。

2. 最新バージョンの ioMemory VSL をインストールします。
3. ユーティリティおよび管理ソフトウェア(ドライバーのインストール手順に記載されています)をインストールします。
4. [ioMemory VSL を読み込み、オプションを設定](#)します。

ドライバのインストール

[PRIMERGY ダウンロードページ: http://jp.fujitsu.com/platform/server/primergy/downloads/](http://jp.fujitsu.com/platform/server/primergy/downloads/)より、ご使用中の OS 用デバイスバインドドライバをダウンロードし、インストールします。

1. ドライバパッケージは、Red Hat Driver Update Process 形式(RHDUP)となっています。含まれている ISO イメージ形式のパッケージより CD、または、USB メモリを作成してください。

i ダウンロードした ISO イメージ形式のパッケージをループマウントしてインストールすることも可能です。

```
$ mkdir -p /mnt/ioDrive
$ mount -o loop <FILENAME>.iso /mnt/ioDrive
```

この場合、CD-ROM マウンティングポイント: /media/OEMDRV を /mnt/ioDrive に置き換えて参照ください。

2. ドライバパッケージ(ISO イメージ内)には、YUM パッケージ用リポジトリが含まれています。“yum” (RHEL パッケージ管理ツール)を使用し、ドライバのインストールを行います。
3. 作成した CD または USB メモリのトップディレクトリにある YUM 用リポジトリファイル dud.repo を、ディレクトリ: /etc/yum.repos.d にコピーします。

```
$ cp /media/OEMDRV/dud.repo
/etc/yum.repos.d/iomemory_vsl_3.1.5.126_1.rhel5_u8.x86_64.repo
```

4. コピーしたリポジトリファイルをファイル内の記載に従い修正します。

以下は、リポジトリファイルの記載内容例です。

ネットワークに接続していない場合は、最終行の”#”を削除します。

```
# Sample yum repository file to use with this Fujitsu driver disk

# Copy this file to
/etc/yum.repos.d/iomemory_vsl_3.1.5.126_1_rhel5_u8.x86_64.repo
# and edit "baseurl" and "gpgkey" lines as indicated below.

[iomemory_vsl_3.1.5.126_1_rhel5_u8.x86_64]
name = iomemory-vsl 3.1.5.126-1 rhel5-u8.x86_64
enabled = yes

# Replace the path below (except the trailing "/rpms")
# by your mount point for the ISO image or CD-ROM
baseurl = file:///<INSERT_MOUNT_POINT_HERE>/rpms

# Check GPG signature on installed packages (recommended)
# Uncomment at EXACTLY ONE of the gpgkey= lines below!
gpgcheck = 1

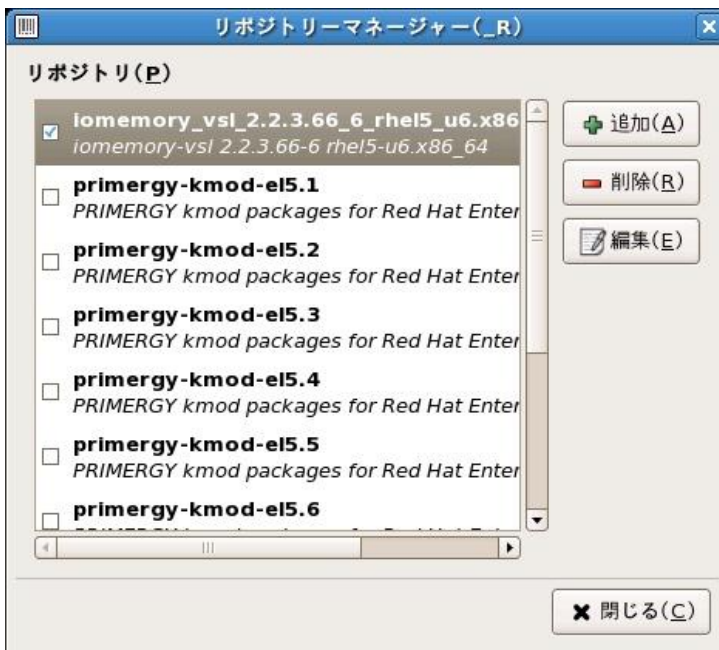
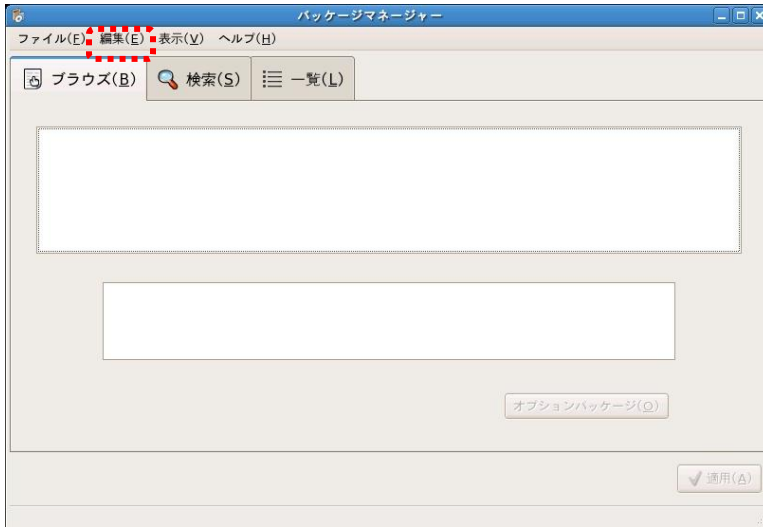
# If your system is connected to the internet, use this configuration:
# gpgkey=http://patches.ts.fujitsu.com/linux/pldp/gpg-pubkey-79444536-46837bd0.asc

# For systems without direct internet connection but with accessible
# mirror of the FTS package repository:
# gpgkey=http://<INSERT_YOUR_MIRROR_HERE>/linux/pldp/gpg-pubkey-79444536-46837bd0.asc

# For systems without direct internet connection and without mirror:
# The following configuration assumes that "primergy-dup" is installed
already.
# If it is not, install primergy-dup first with the command
```

```
# yum install --nogpgcheck primergy-dup
# gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fts-linux-driver-build
```

5. [アプリケーション]-[ソフトウェアの追加/削除]を選択し、「パッケージマネージャー」を起動します。
[編集(E)]のプルダウンメニューより[リポジトリ(R) Ctrl+R]を選択し、リポジトリが登録されていることを確認します。



6. 「パッケージマネージャー」に戻り、「一覧(L)」タブを選択、「有効なパッケージ(V)」を選択します。
表示されるデバイスドライバを選択しインストールを行ってください。



Red Hat Driver Update Process 形式(RHDUP)の詳細については、以下を参照ください。

http://www.ts.fujitsu.com/products/standard_servers/linux_readmes_popup.html
(英文です。)

RPM パッケージのインストール

PRIMERGY ダウンロードページ: <http://jp.fujitsu.com/platform/server/primergy/downloads/> から
ツールおよびユーティリティパッケージをダウンロードします。

このパッケージには、以下の RPM パッケージが含まれています。

パッケージ	インストール内容
fio-util-<VSL バージョン>.x86_64.rpm	ioMemory VSL ユーティリティ - 推奨
libvsl-<バージョン>.x86_64.rpm	管理ツールに必要なライブラリ - 推奨
fio-common-<VSL バージョン>.x86_64.rpm	init スクリプトに必要なファイル - 推奨
fio-sysvinit-<VSL バージョン>.x86_64.rpm	init スクリプト - 推奨
fio-smis-<バージョン>.x86_64.rpm	SMI-S CIM プロバイダー - オプション。「 付録 E - SMI-S インターフェイス 」を参照
fio-snmp-agentx-<バージョン>.x86_64.rpm	AgentX SNMP サブエージェント - オプション。 このパッケ


ン>.x86_64.rpm	ージのインストール前に net-snmp のインストールが必要。詳細については「 付録 D - SNMP の設定 」を参照
fio-snmp-mib-<バージョン>.x86_64.rpm	SNMP MIB - オプション。「 付録 D - SNMP の設定 」を参照

1. 次のコマンドを入力して、サポート ファイルをインストールします。

```
rpm -Uvh lib*.rpm
rpm -Uvh fio*.rpm
```

2. ioMemory VSL およびユーティリティは、次の場所にインストールされます。

パッケージ タイプ	インストール場所
ioMemory VSL	/lib/modules/<カーネル バージョン>/extra/fio/iomemory-vsl.ko
ユーティリティ	/usr/bin
ファームウェア	/usr/share/fio/firmware
SNMP MIB	/usr/share/fio/mib

 ioSphere アプリケーション(オプションの GUI 管理ソフトウェア)をインストールすることもできます。このアプリケーションは、インストール用ドキュメントやユーザー向けドキュメントと共に ioSphere ダウンロード フォルダに用意されています。


パッケージをインストールしたら、このセクションで後述する「[ioMemory VSL 機能\(ドライバー\)の読み込み](#)」に進みます。

ioMemory VSL ドライバーの読み込み

ioMemory VSL ドライバーを読み込むには:


1. 次のコマンドを実行します。

```
$ modprobe iomemory-vsl
```

 ioMemory VSL はシステム ブート時に自動的に読み込まれます。これにより、ioMemory デバイスは `/dev/fiox` として OS で使用可能な状態になります (x の部分は a、b、c などの文字で置き換えられます)。

- このコマンドを SLES システムで動作させる場合は、`/etc/init.d/iomemory-vsl.conf` ファイルの `init` 情報を編集して `udev` の部分を `boot.udev` に変更する必要があります。すると、このファイルは次のようになります。
 - `### BEGIN INIT INFO`
 - `# Provides:iomemory-vsl`
 - `# Required-Start:boot.udev`
 - SLES システムでは、このコマンドを動作させるために、サポートされていないモジュールを許可する必要もあります。
 - **SLES 11 アップデート 2 の場合:** `/etc/modprobe.d/iomemory-vsl.conf` ファイルを編集して、該当する行のコメントを解除します。
 - `# To allow the ioMemory VSL driver to load on SLES11, uncomment below`
 - `allow_unsupported_modules 1`
 - **SLES 10 SP4 の場合:** `/etc/sysconfig/hardware/config` ファイルを編集して、`LOAD_UNSUPPORTED_MODULES_AUTOMATICALLY` という変数を `yes` に設定します (以下に例を示します)。

```
LOAD_UNSUPPORTED_MODULES_AUTOMATICALLY=yes
```
2. ioMemory デバイスが接続されたことを確認するには、コマンドラインから `fio-status` ユーティリティを実行します。出力結果に、各ドライブとその状態 (接続されているかどうか) の一覧が表示されます。

 ioMemory デバイスが自動的に接続されない場合は、`/etc/modprobe.d` ファイルを参照して `auto_attach` オプションがオフになっていないか (0 に設定されていないか) 確認してください。

ioMemory VSL の読み込みの制御

ドライバーの読み込みは、`init` スクリプトまたは `udev` によって制御できます。

比較的新しい Linux ディストリビューションでは、ユーザーが udev デバイス マネージャーを利用して、取り付けられているハードウェア用のドライバーの探索と読み込みをブート時に自動的に行うことができます。なお、udev は無効にすることができ、init スクリプトはほとんどすべての場合に使用できます。こうした機能を備えていない古い Linux ディストリビューションでは、ユーザーがブート時の init スクリプトを利用して、必要なドライバーを読み込む必要があります。Fusion-io では、VSL ドライバーを以前の RHEL4 リリースや SLES10 ディストリビューションに読み込むための init スクリプトを `/etc/init.d/iomemory-vsl` に用意しています。

init スクリプトの使用

udev によるドライバーの読み込みが機能しない、または無効になっているシステムでは、init スクリプトを有効にしてブート時にドライバーを読み込むことができます。ディストリビューションによっては、init スクリプトがデフォルトで有効になっている場合があります。

i init スクリプトは `fio-sysvinit` パッケージに含まれています。このパッケージは、init スクリプトを有効にする前にインストールしておく必要があります。

ioMemory VSL の読み込みは、次のコマンドで無効にできます。

```
$ chkconfig --del iomemory-vsl
```

init スクリプトで ioMemory VSL の読み込みを再び有効にするには、次のコマンドを使用します。

```
$ chkconfig --add iomemory-vsl
```

ioMemory のインストール処理により、init スクリプトは `/etc/init.d/iomemory-vsl` に配置されます。次に、このスクリプトでは `/etc/sysconfig/iomemory-vsl` 内のオプション ファイルにある設定オプションが使用されます。init スクリプトを使用するには、このオプション ファイルで ENABLED を (ゼロ以外の値に) 設定する必要があります。

```
ENABLED=1
```

オプション ファイルには、さまざまな設定に関する説明が含まれています。そのうち、MOUNTS と KILL_PROCS_ON_UMOUNT の 2 つについては、このドキュメントで後述する「[ioMemory VSL の読み込み解除のハンドリング](#)」で詳しく説明します。

init スクリプト使用時のファイルシステムのマウント

ioMemory VSL は標準の手段 (`initrd` 内のもの、またはカーネルに組み込まれているもの) では読み込まれないので、ioMemory デバイス上でホストされるファイルシステムをマウントするための標準の方法を使用しても適切に動作しません。ioMemory デバイス上でホストされるファイルシステムの自動マウントを設定するには:

1. ファイルシステムをマウントするコマンドを/etc/fstab に通常どおりに追加します。
2. 次に示す 2 つのサンプル エントリのように、'noauto'オプションを etc/fstab に追加します。

```
/dev/fioa /mnt/fioa ext3 defaults,noauto 0 0  
/dev/fiob1 /mnt/ioDrive ext3 defaults,noauto 0 0
```

(fioa における a の部分は、システムに取り付けられている ioMemory デバイスの数に応じて、a、b、c、などで置き換えます。)

init スクリプトによってこれらのデバイスのマウントを ioMemory VSL の読み込みおよびアンマウント後、かつ ioMemory VSL の読み込み解除前に行うには、オプション ファイルに記されている手順に従ってマウント ポイントのリストをオプション ファイルに追加します。

先ほどの例に示したファイルシステム マウントの場合、オプション ファイル内の該当行は次のようになります。

```
MOUNTS="/mnt/fioa /mnt/iodrive"
```

udev の使用

udev を利用してドライバーを読み込むシステムで、udev によるブート時の ioMemory VSL の自動読み込みを禁止する場合は、ユーザーが ioMemory VSL のオプション ファイルを変更する必要があります。そのためには、既に次の行が存在する/etc/modprobe.d/iomemory-vsl.conf ファイルを探して編集します。

```
# blacklist iomemory-vsl
```

読み込みを無効にするには、行頭の“#”を削除してファイルを上書き保存します。

この blacklist コマンドの設定が済んだら、Linux を再起動します。udev による ioMemory VSL の読み込みが行われなくなります。

udev による ioMemory VSL の読み込み機能を復元するには、“#”を再び追加してこの行をコメントアウトします。

udev または init スクリプトによるシステム

ユーザーは、ブート時の ioMemory VSL の読み込みを無効にできます。そうすると、udev または init スクリプトによるシステムのどちらでも、診断またはトラブルシューティングの目的で自動接続の処理を止めることができます。自動接続機能の無効化または再有効化を行うには、「[自動接続の無効化](#)」の手順に従います。

あるいは、ブートローダーのカーネルコマンドラインで次のパラメータを追加しても、ioMemory VSL の読み込みを止めることができます。

```
iodrive=0
```

ただし、この方法は、ioMemory VSL がまったく機能しなくなり、実行できるトラブルシューティングの範囲が制限されるので、好ましくはありません。

ioMemory デバイスとマルチパスストレージ

ioMemory デバイスをマルチパスストレージと共に使用する場合は、ioMemory デバイスをブラックリストに登録して、device-mapper が ioMemory デバイスごとに dm-device を作成しようとするのを防ぐ必要があります。この作業は、dm-multipath のアクティブ化の前、またドライバーの読み込み前に行う必要があります。ioMemory デバイスがブラックリストに登録されていない場合は、ビジー状態にあるように見えるので、ioMemory デバイス上でファームウェアの接続、切断、更新を行うことができなくなります。

ioMemory デバイスをブラックリストに登録するには、`/etc/multipath.conf` ファイルを編集して以下の行を追加します。

```
blacklist {
  devnode      "^fio[a-z]"
}
```

ioMemory VSL の読み込み解除のハンドリング

ioMemory VSL の読み込み解除時には、特別な検討を行う必要があります。デフォルトでは、マウントされたファイルシステムを開いているすべてのプロセスが init スクリプトによって検出されて強制終了されます。その結果、ファイルシステムのアンマウントが可能になります。この動作は、オプションファイル内の `KILL_PROCS_ON_UMOUNT` というオプションによって制御されます。これらのプロセスが強制終了されない場合、ファイルシステムはアンマウントできません。こうした場合、ioMemory VSL の読み込み解除が正常に行われず、その後のブートで大幅な遅延が発生する可能性があります。

ioMemory VSL オプションの設定

このセクションでは、ioMemory VSL オプションの設定方法を説明します。設定に固有のオプションの詳細については、「[付録 C - モジュールパラメータの使用](#)」を参照してください。

一度限りの構成

ioMemory VSL オプションは、`insmod` または `modprobe` のコマンドラインで、インストール時に設定できます。たとえば、`auto_attach` という ioMemory VSL オプションを 0 に設定するには、次のコマンドを実行します。

```
$ modprobe iomemory-vsl auto_attach=0
```

このオプションは、この ioMemory VSL のその時点の読み込みでのみ有効です。その後の `modprobe` または `insmod` の呼び出しでは、このオプションは設定されません。

永続的な構成

オプションに対する永続的な設定を維持するには、必要なオプションを `/etc/modprobe.d/iomemory-vsl.conf` または類似のファイルに追加します。ioMemory デバイスの自動接続を止めるには、`iomemory-vsl.conf` ファイルに次の行を追加します。

```
options iomemory-vsl auto_attach=0
```

すると、この ioMemory VSL オプションが、その後のすべての ioMemory VSL の読み込みや、ブート時の ioMemory VSL の自動読み込みで有効になります。

ファームウェアのアップグレード


ioMemory VSL が読み込まれた場合は、ioMemory デバイスのファームウェアが最新のものであることを確認する必要があります。そのためには、[fio-status](#) コマンドライン ユーティリティを実行します。

デバイスが最小モード (minimal mode) で動作していることが出力結果に示された場合は、最新のファームウェアを [PRIMERGY ダウンロードページ](#): <http://jp.fujitsu.com/platform/server/primergy/downloads/> からダウンロードし、ioManager ソフトウェアまたは [fio-update-iodrive](#) ユーティリティを使用して、ファームウェアをアップグレードします。

アップグレード パス

ioMemory デバイスのアップグレード時には、所定のアップグレード パスがあります。ioMemory デバイスをアップグレードする前に、このリリースの ioMemory VSL のリリース ノートを参照してください。

ioMemory デバイスに最小ファームウェアのラベル (例: "MIN FW: XXXXXX") が付いていることがあります。このラベルは、デバイスと互換性がある最小のファームウェア バージョンを示しています。

 どの ioMemory デバイスでもファームウェアのダウングレードは行わないでください。ダウングレードを試みると保証が無効になる場合があります。

既存のデバイスが取り付けられている状態で新しい ioMemory デバイスを取り付ける場合は、新しいデバイスを取り付ける前に、現在取り付けられているすべてのデバイスを利用可能な最新のバージョンのファームウェアおよび ioMemory VSL にアップグレードする必要があります。

アップグレードに関する考慮事項については、このリリースの ioMemory VSL のリリース ノートを参照してください。

VMware のゲスト OS のアップグレード

(VM の DirectPath I/O を使用して) VMware のゲスト OS で ioMemory デバイスを使用している場合は、デバイスをアップグレードした後に、ホストの電源をいったんオフにして、再度オンにする必要があります。仮想マシンを再起動するだけでは、変更内容が適用されません。

PCIe 電力の有効化

ioDrive2 Duo デバイスなどのデュアル ioMemory デバイスを取り付けている場合、それらのデバイスが適切に機能するには、追加の電力 (PCIe Gen2 スロットによって供給される最低限の電力である 25 W を超える電力) を必要とする場合があります。

追加の電力は、電源ケーブル (『ioMemory ハードウェア インストール ガイド』を参照) または PCIe スロットから供給できます。PCIe スロットからデバイスに追加の電力を供給できるようにする方法については、「メンテナンス」の「[PCIe 電力優先 \(PCIe Power Override\) の有効化](#)」を参照してください。

デバイスのスワップとしての使用

ioMemory デバイスをスワップ領域として安全に使用するには、`preallocate_memory` というカーネル モジュール パラメータを渡す必要があります。このパラメータを指定する方法としては、`/etc/modprobe.d/iomemory-vsl.conf` ファイルに次の行を追加する方法が推奨されています。

```
options iomemory-vsl preallocate_memory=1072,4997,6710,10345
```

- ここで、1072、4997、6710、10345 は [fio-status](#) によって取得したシリアル番号です。

スワップでは、4K のセクター サイズのフォーマットが必要です。これにより、ioMemory VSL によるメモリの占有領域を適切なレベルに抑えることができます。ioMemory デバイスをセクター サイズ 4k でフォーマットするには、[fio-format](#) を使用します。

i アダプターではなく、ioMemory デバイスのシリアル番号を指定してください (該当する場合)。

i デバイスをスワップ領域として使用可能にするためには、`preallocate_memory` モジュール パラメータが必要です。このパラメータの設定の詳細については、「[付録 C - モジュール パラメータの使用](#)」を参照してください。

w ioMemory デバイスでスワップとして使用するための事前割り当てを有効にするには、十分な RAM が利用できる必要があります。十分な RAM がないまま事前割り当てを有効にして ioMemory デバイスを接続すると、ユーザー プロセスが強制終了したり、システムが不安定になったりする場合があります。

このバージョンの ioMemory VSL の RAM 要件については、リリース ノートを参照してください。

i `preallocate_memory` パラメータは ioMemory VSL によって読み込み時に認識されますが、要求されたメモリは、実際には指定されたデバイスが接続されるまでは割り当てられません。

論理ボリューム マネージャーの使用

次のようにして ioMemory デバイスをサポートされるタイプとして追加した場合は、ボリューム グループ管理アプリケーションである論理ボリューム マネージャー (LVM) によって、ioMemory デバイスのような大容量ストレージ デバイスが処理されます。

1. `/etc/lvm/lvm.conf` という構成ファイルを探して編集します。
2. このファイルに次のようなエントリを追加します。

```
types = [ "fio", 16 ]
```

“16”というパラメータは、デバイスでサポートされるパーティションの最大数を表しています。

i LVM または MD を使用する場合は、udev を使用して ioMemory VSL ドライバーを読みこまないでください。init スクリプトは、ioMemory デバイスの切断を試みる前に LVM ボリュームと MD デバイスが切断されていることを確認します。

RAID の構成

2 つ以上の ioMemory デバイスを構成して、ソフトウェアベースの RAID ソリューションを使用した RAID アレイにすることができます。

i RAID 1/ミラーを使用しており、1 つのデバイスで障害が発生した場合は、RAID を再構築する前に、(障害が発生していない既存のデバイスではなく) 交換したデバイスで `fio-format` を実行してください。以下に、mdadm ユーティリティを使用した一般的な RAID 構成の例をいくつか示します。

w Linux カーネルによる RAID 5 実装は、データ転送速度が速い場合に十分な性能を発揮できません。これは Linux カーネルの問題です。代替策としては、RAID10 の使用、場合によってはサードパーティによる RAID スタックの使用が挙げられます。

RAID 0

`fioa` と `fiob` という 2 つの ioMemory デバイスでストライピングする場合、ストライプ セットを作成するには次のコマンドを実行します。

```
$ mdadm --create /dev/md0 --chunk=256 --level=0 --raid-devices=2 /dev/fioa /dev/fiob
```


アレイの永続化(再起動後に存在)

Linux の一部のバージョンでは、構成ファイルが `/etc/mdadm.conf` ではなく、`/etc/mdadm/mdadm.conf` にあります。

`/etc/mdadm.conf` の内容を確認します。対象となるデバイスが複数の行で宣言されている場合は、それらの行の 1 つで "partitions" がオプションとして指定されていることを確認します。そうっていない場合は、次のように "partitions" を指定する新しい DEVICE 行をファイルに追加します。

```
DEVICE partitions
```

また、`fio` という ioMemory デバイスのデバイス指定子を次のように追加します。

```
DEVICE /dev/fio*
```

`/etc/mdadm.conf` の更新が必要かどうかを確認するには、次のコマンドを実行します。

```
$ mdadm --detail --scan
```

このコマンドの出力を `mdadm.conf` の現在の内容と比較し、必要なセクションがあれば `/etc/mdadm.conf` に追加します。

たとえば、アレイが 2 つのデバイスで構成されている場合、コマンドの出力には `mdadm.conf` ファイルに存在しない行が 3 つあります。1 行はアレイに関するもの、2 行はデバイスに関するもの(各デバイスに 1 行)です。これらの行を `mdadm.conf` に追加して、ファイルの内容がコマンドの出力に一致するようにしてください。

詳細については、お使いのディストリビューションの `mdadm` および `mdadm.conf` の man ページを参照してください。

これらの変更により、ほとんどのシステムでは RAID 0 アレイが再起動時に自動的に作成されます。ただし、再起動後に `/dev/md0` へのアクセスに問題がある場合は、次のコマンドを実行します。

```
$ mdadm --assemble --scan
```

また、必要に応じて、`udev` による ioMemory VSL ドライバーの読み込みを無効にし、ドライバーを読み込むための `init` スクリプトを使用することもできます。`init` スクリプトの使用の詳細については、このガイドの「[init スクリプトの使用](#)」を参照してください。

SLES 11 では、これらのサービスがブート時に実行されるように、次のコマンドの実行が必要になる場合があります。

```
chkconfig boot.md on
chkconfig mdadmd on
```

RAID 1

2 つの ioMemory デバイス `fioa` と `fiob` を使用してミラーセットを作成するには、次のコマンドを実行します。

```
$ mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/fioa /dev/fiob
```

RAID 10

4 つの ioMemory デバイス (`fioa`、`fiob`、`fioc`、`fiod`) を使用してストライプ化されたミラー アレイを作成するには、次のコマンドを実行します。

```
$ mdadm --create /dev/md0 -v --chunk=256 --level=raid10 --raid-devices=4 /dev/fioa /dev/fiob /dev/fioc /dev/fiod
```


複数のデバイスにわたる RAID 10 の構築


RAID 10 構成では、2 つのディスクによるセットをミラー化したうえで、それぞれのミラーをストライプ化しています。複数の ioMemory デバイスにわたる RAID 10 を設定するには、1 つの製品 (ioDrive Duo デバイスなど) を構成する 2 つの ioMemory デバイスだけでミラーを構成しないようにすることが適切です。

データを取得して適切にレイアウトするには、次の手順に従います。

- RAID 10 の作成時に `--layout=n2` オプションを使用します (ただし、このオプションはデフォルトになっているはずです)。
- 同じデバイスを構成する 2 つの ioMemory デバイスがリストで隣どうしにならないようにします。

以下のサンプル コードに、推奨構成を示します。

 以下のコマンドでは、すべての ioMemory デバイスが `fio-format` ユーティリティによって新たにフォーマット済みであることを前提としています。

 `fiox` デバイスの順序は非常に重要です。

```
# 2 Duos RAID10
$ mdadm --create --assume-clean --level=raid10 --layout=n2 -n 4 /dev/md0 \
/dev/fioa /dev/fioc \
/dev/fiob /dev/fiod
# Mirror groups are:fioa,fioc and fiob,fiod

# 3 Duos RAID10
$ mdadm --create --assume-clean --level=raid10 --layout=n2 -n 6 /dev/md0 \
/dev/fioa /dev/fiod \
/dev/fioc /dev/fiof \
/dev/fioe /dev/fiob

# 4 Duos RAID10
$ mdadm --create --assume-clean --level=raid10 --layout=n2 -n 8 /dev/md0 \
/dev/fioa /dev/fiod \
```



```
/dev/fioc /dev/fiof \  
/dev/fioe /dev/fioh \  
/dev/fiog /dev/fiob  
  
# 8 Duos RAID10  
$ mdadm --create --assume-clean --level=raid10 --layout=n2 -n 16 /dev/md0 \  
/dev/fioa /dev/fiod \  
/dev/fioc /dev/fiof \  
/dev/fioe /dev/fioh \  
/dev/fiog /dev/fioj \  
/dev/fioi /dev/fiol \  
/dev/fiok /dev/fion \  
/dev/fiom /dev/fiop \  
/dev/fioo /dev/fiob
```

Discard (TRIM) 機能のサポートについて

このバージョンの ioMemory VSL では、Discard (TRIM と呼ばれます) 機能がデフォルトで有効になっています。

Discard 機能は、ソリッドステート ストレージに特有の問題に対処するための機能です。ユーザーがファイルを削除した場合、デバイスはその領域が再利用可能であることを認識しません。デバイスは、データが引き続き有効であると見なします。


Discard は、比較的新しくリリースされたファイルシステムに備わっている機能です。Discard 機能によって、有効なユーザー データが含まれなくなった論理セクターがデバイスに通知されます。これにより、ウェアレベリング ソフトウェアは今後の書き込み操作を処理するためにその領域を再利用できます。

Linux における Discard (TRIM) 機能

このリリースでは、Discard 機能がデフォルトで有効になっています。ただし、Discard を実現するには、Linux ディストリビューションがこの機能をサポートしており、かつ Discard の機能がオンになっている必要があります。

つまり、お使いの Linux ディストリビューションが Discard をサポートしていて、システムでこの機能が有効になっていれば、Discard 機能が ioMemory デバイス上で実現されます。

Linux では、Discard リクエストはファイルシステムによって発行されるようには限定されておらず、カーネルの Discard ioctl を使用してユーザー空間のアプリケーションから直接 Discard リクエストを発行することもできます。

 Kernel.org 2.6.33 以前の ext4 には、Discard 機能が有効になっていると警告なしにデータを破壊する可能性があるという既知の問題が存在します。この問題は、ディストリビューション ベンダーによって提供される多くのカーネルで修正済みです。お使いのカーネルが Discard 機能を適切にサポートしていることをカーネル プロバイダーに確認してください。詳細については、このバージョンのソフトウ

エアのリリースノートにある正誤表を参照してください。

- ④ 現在 Linux では、MD および LVM が Discard (TRIM) リクエストを根底にあるデバイスに渡していません。そのため、MD または LVM アレイを構成する ioMemory デバイスがファイルシステムによって送信された Discard (TRIM) リクエストを受け取ることはありません。

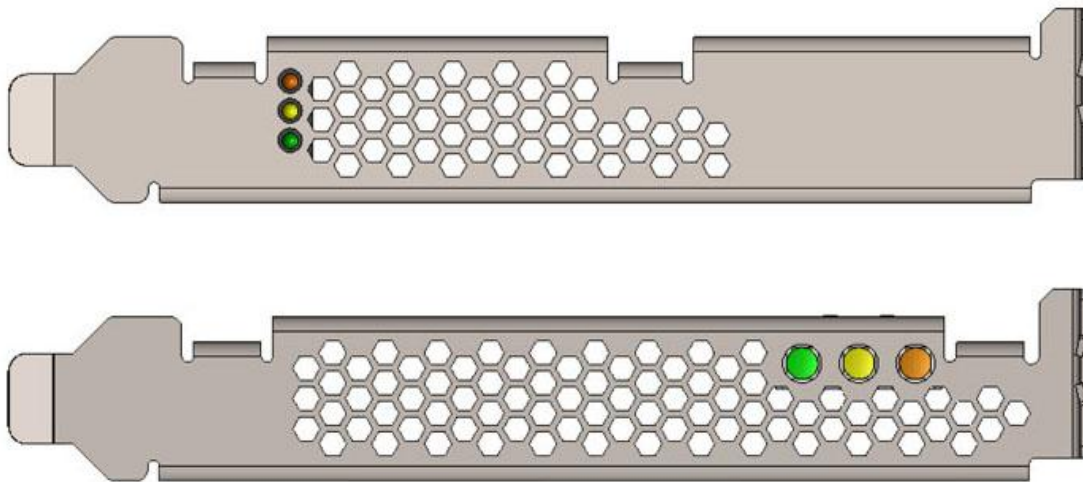
Red Hat 6.1 に含まれている LVM リリースは、(すべてではなく)いくつかのターゲットで Discard (TRIM) リクエストの引き渡しをサポートしています ([RHEL 6.1 のドキュメント](#))。詳細については、お使いのディストリビューションのドキュメントを参照してください。

メンテナンス

ioMemory VSL には、デバイスのメンテナンス用のソフトウェア ユーティリティが含まれています。監視用のオプションとして、SNMP をインストールすることもできます。


デバイスの LED インジケータ

ioMemory デバイスには、ドライブのアクティビティやエラー状態を示す 3 つの LED が用意されています。デバイスの LED は、次に示す図のいずれかに似た構成になっています。



これらの LED が示す情報について次の表で説明します。

Green	Yellow	Amber	Indicates	Notes
○	○	○	Power off	
○	○	●	Power on (Driver not loaded and device not attached)	Load Driver and attach device
●	○	○	Power on, Driver loaded (device may not be attached)	You may need to attach the device
●	(Flashing) ●	○	Writing (Rate indicates volume of writes)	Can appear in combination with the Read LED indication
(Flashing) ●	○	○	Reading (Rate indicates volume of reads)	Can appear in combination with the Write LED indication
●	●	●	Location Beacon	

 ioFX デバイスには、ioFX のロゴがイルミネーション表示される追加の LED があります。この LED には機能的な意味はなく、消灯することもできません。

GUI 管理

ioManager は、ioMemory デバイスを管理するための無料のブラウザベース ソリューションです。
[PRIMERGY ダウンロードページ: http://jp.fujitsu.com/platform/server/primergy/downloads/](http://jp.fujitsu.com/platform/server/primergy/downloads/) から入手することもできますが、ダウンロード場所はドキュメントが置いてあるサイト ページの別の場所になります。

ioManager は次のような多くの管理機能を実行できます。

- ファームウェアのアップグレード
- ローレベル フォーマット
- 接続および切断操作
- デバイスのステータス情報および性能情報の確認
- スワップおよびページングの構成
- バグ レポートの生成

コマンドライン ユーティリティ

ioMemory デバイスを管理するためのいくつかのコマンドライン ユーティリティがインストール パッケージに含まれています。

- fio-attach
- fio-beacon
- fio-bugreport
- fio-detach
- fio-format
- fio-pci-check
- fio-snmp-agentx
- fio-status
- fio-sure-erase
- fio-update-iodrive

コマンドライン ユーティリティの詳細については、「[付録 A - ユーティリティ リファレンス](#)」を参照してください。

PCIe 電力優先(PCIe Power Override)の有効化

ioDrive Duo デバイスなどの複数の ioMemory デバイスで構成された製品を取り付けている場合、それらのデバイスが適切に機能するには、追加の電力 (PCIe Gen2 スロットによって供給される最低限の電力である 25 W を超える電力) を必要とする場合があります。使用するデバイスで追加の電力が必要ではない場合でも、追加の電力を供給するとすべてのデュアル ioMemory デバイスで性能が向上する可能性があります。

ioDrive2 Duo デバイスが適切に機能するためには、追加の電力を供給する**必要があります**。追加の電力を必要とするデバイスの詳細については、『ioMemory ハードウェア インストール ガイド』の「複数のデバイスで構成された製品の電源ケーブル」を参照してください。

追加の電力は、2 つの方法で供給できます。

- **外部電源ケーブル:** このケーブルの取り付け方法の詳細については、『ioMemory ハードウェア インストール ガイド』を参照してください。

i 電源ケーブルを使用した場合は、すべての電力がケーブルから供給されます。PCIe スロットの電力は使用されません。

- **スロットからすべての電力を供給する:** 一部の PCIe スロットは、追加の電力を供給します (多くの場合、最大で 75 W)。スロットの定格電力が 55 W 以上の場合は、VSL モジュールのパラメータを設定することによって、デバイスのすべての電力を PCIe スロットから供給できます。この上書きパラメータ (override parameter) の有効化の詳細については、次のセクションの説明を参照してください。

i このパラメータによって、PCIe スロットからデバイスに 25 W を超える電力を供給するのを禁止する設定が上書きされます。このパラメータは、(デバイスのシリアル番号を使用して) デバイスごとに有効化されます。設定が上書きされると、各デバイスにおいて、最大の性能を発揮するのに必要な最大 55 W までの電力をスロットから供給できます。

警告
必要な電力をスロットから供給できない場合に PCIe スロットからすべての電力を供給する設定を有効化すると、デバイスが正しく機能しなくなったり、サーバのハードウェアが損傷したりする危険性があります。この上書きパラメータ (override parameter) の不適切な使用に起因する装置に対するすべての損傷については、お客様の責任となります。このような不適切な使用に起因するいかなる損傷についても富士通は一切の責任を負わないことをここに明記します。

この上書きパラメータ (override parameter) を有効化する前に、使用する各 PCIe スロットの定格電力が、すべてのスロット、デバイス、およびサーバ アクセサリに対して十分な電力を供給できるワット数で


あることを確認します。サーバのドキュメント、BIOS インターフェイス、セットアップ ユーティリティを参照するか、[fio-pci-check](#) を使用して、スロットの電力制限を確認してください。

サーバの製造業者への確認

各スロットおよびシステム全体の電力の制限と機能については、サーバの製造業者に問い合わせてください。

次に、重要な考慮事項を示します。

- 複数の ioDrive2 Duo デバイスを取り付けて、各デバイスに対して上書きパラメータ(override parameter)を有効にする場合は、マザーボードの定格電力が、使用する各スロットに対して 55 W の電力を供給できるワット数であることを確認します。

 たとえば、一部のマザーボードでは、任意の 1 つのスロットでは最大で 75 W まで安全に供給できますが、複数のスロットで 75 W を供給すると電力が制限されます。このような場合に複数のデバイスを取り付けたときも、サーバのハードウェアが損傷する危険性があります。PCIe スロット全体で使用可能な電力については、製造業者に問い合わせてください。

- 上書きパラメータ(override parameter)は、適切に有効化すると、システム内で永続的に保持されます。デバイスを取り外し、同じシステム内の別のスロットに取り付けた場合でも、パラメータが有効化されたデバイスではスロットからすべての電力が供給されます。ただし、定格電力が 55 W 未満のスロットにデバイスを取り付けると、サーバのハードウェアが損傷する危険性があります。
- この上書きパラメータ(override parameter)は、サーバごとの ioMemory VSL ソフトウェアの設定であり、デバイスには保存されません。デバイスを新しいサーバに移動した場合は、外部電源ケーブルを追加するか、または新しいサーバでデバイスに対してこの上書きパラメータ(override parameter)を有効にするまでの間、デバイスに対してデフォルトの 25 W の電力制限が適用されます。新しいサーバにおける PCIe スロット全体で使用可能な電力については、製造業者に問い合わせてください。

上書きパラメータ(override parameter)の有効化

シリアル番号の特定

このパラメータを有効化する前に、互換性のあるスロットに挿入する各デバイスのアダプターのシリアル番号を特定します。アダプターのシリアル番号を特定するには、`fio-status` コマンドライン ユーティリティを使用します。

シリアル番号のラベル

デバイスに貼付されているアダプターのシリアル番号のラベルを調べて、シリアル番号を特定することもできます。ただし、ベスト プラクティスとして、`fio-status` を実行し、それぞれのシリアル番号がア

アダプターのシリアル番号であることを確認することをお勧めします。アダプターのシリアル番号のラベルは、すべての ioDrive Duo デバイスおよび ioDrive2 Duo デバイスの背面に貼付されています。ioDrive Duo デバイスでは、PCIe コネクタに接続されたプリント基板に貼付されています。

fio-status の使用: `fio-status` コマンドライン ユーティリティを実行します。次に、サンプルの出力を示します。

```
fio-status
...
Adapter: Dual Controller Adapter
  Fusion-io ioDrive2 DUO 2.41TB, Product Number:F01-001-2T41-CS-0001, FIO SN:1149D0969
  External Power: NOT connected
  PCIe Power limit threshold: 24.75W
  Connected ioMemory modules:
    fct2:   SN:1149D0969-1121
    fct3:   SN:1149D0969-1111
```

この例では、1149D0969 がアダプターのシリアル番号です。


fio-beacon の使用: 複数のデバイスを取り付けている場合は、[fio-beacon](#) ユーティリティを使用して、各デバイスの物理的な場所を確認できます。詳細については、付録のユーティリティについてのドキュメントを参照してください。

パラメータの設定

`/etc/modprobe.d/iomemory-vsl.conf` ファイルを編集して `external_power_override` パラメータの値を変更することで、モジュール パラメータを設定します。例:


```
options iomemory-vsl external_power_override=<値>
```

このパラメータの<値>は、アダプターのシリアル番号をコンマで区切ったリストです。たとえば、1149D0969, 1159E0972, 24589 のように指定します。

 パラメータの変更を適用するには、ドライバーのリブート、または読み込み解除と再読み込みが必要です。

一般的なメンテナンス作業

以下では、コマンドライン ユーティリティを使用して ioMemory デバイスのメンテナンスを行うための非常に一般的な作業について説明します。

 すべてのコマンドには、管理者特権が必要です。“root”としてログインするか、`sudo` を使用して、コ

マンドを実行してください。

- ④ 「[ソフトウェアのインストール](#)」からこのセクションに移動してきた場合は、以前のバージョンのドライバーおよびユーティリティをアンインストールした後、元のセクションに戻ってください。

ioMemory VSL ドライバーの読み込み解除

ioMemory VSL の読み込みを解除するには、次のコマンドを実行します。

```
$ modprobe -r iomemory-vsl
```

ioMemory VSL の RPM パッケージのアンインストール

ioMemory VSL のバージョン 2.x 以降(3.x リリースを含む)では、アンインストールするパッケージのカーネル バージョンを指定する必要があります。インストールされているドライバー パッケージを調べるには、次のコマンドを実行します。

```
$ rpm -qa | grep -i iomemory
```

次に、サンプルの出力を示します。

```
iomemory-vsl-2.6.18-194.el5-2.2.2.82-1.0
```

ioMemory VSL をアンインストールする場合は、次の例に類似したコマンドを実行します(アンインストールするドライバーのカーネル バージョンを指定します)。

```
$ rpm -e iomemory-vsl-2.6.18-194.el5-2.2.0.82-1.0
```

ioMemory VSL ユーティリティとその他のサポート パッケージのアンインストール

2.x サポート パッケージのアンインストール

サポート RPM パッケージをアンインストールするには、次のコマンドを実行します(必要に応じてパッケージ名を追加または削除します)。

```
$ rpm -e fio-util fio-snmp-agentx fio-common fio-firmware iomanager-gui  
iomanager-jre libfio libfio-doc libfusionjni fio-sysvinit fio-smis fio-snmp-  
mib libfio-dev
```

3.x サポート パッケージのアンインストール

サポート RPM パッケージをアンインストールするには、次のコマンドを実行します(必要に応じてパッケージ名を追加または削除します)。


```
$ rpm -e fio-util fio-snmp-agentx fio-common fio-firmware libvsl libvsl-doc
fio-sysvinit fio-smis fio-snmp-mib libvsl-dev
```

自動接続(auto-attach)の無効化

ioMemory VSL をインストールすると、ioMemory VSL の読み込み時にすべてのデバイスを自動的に接続するように構成されます。場合によっては、自動接続機能の無効化が必要になります。その場合は次の手順に従います。

1. 次のファイルを編集します。

```
/etc/modprobe.d/iomemory-vsl.conf
```

2. このファイルに次の行を追加します。

```
options iomemory-vsl auto_attach=0
```

3. ファイルを保存します。自動接続を再び有効にするには、同じファイルを編集して、追加した行を削除するか次のように変更します。

```
options iomemory-vsl auto_attach=1
```

突然のシャットダウンの問題

停電その他の事情で突然のシャットダウンが発生した場合は、再起動時に ioMemory デバイスで強制的に整合性チェック(Consistency Check)が実行されることがあります。このチェックの完了には、数分間またはそれ以上の時間がかかることがあります。

突然のシャットダウンが発生した場合、ioMemory デバイスに書き込まれたデータが失われることはありませんが、重要なデータ構造が適切にデバイスにコミットされていない可能性があります。この整合性チェック(Consistency Check)では、このようなデータ構造が修復されます。

ioMemory VSL の無効化


ioMemory VSL の読み込みは、デフォルトではオペレーティング システムの起動時に自動的に行われます。診断またはトラブルシューティングのために、ioMemory VSL の自動読み込みを無効にできます。

ioMemory VSL の自動読み込みを無効にするには:

1. ブート ローダーのカーネル コマンド ラインに次のパラメータを追加します。

```
iodrive=0
```

ioMemory VSL が読み込まれなくなり、ユーザーはデバイスを操作できなくなります。

 ioMemory VSL をアンインストールして読み込まないようにしたり、ioMemory VSL を `/lib/modules/<カーネル バージョン>` ディレクトリの外に移動したりすることもできます。

2. トラブルシューティングを実行して問題を修正します。古いファームウェアに問題がある場合は、`iodrive=1` を使用して ioMemory VSL を最小モード(minimal mode)にします。その後、`fio-update-iodrive` または ioManager アプリケーションを使用してファームウェアを更新できます。
3. `fio-attach` ユーティリティまたは ioManager アプリケーションを使用して、ioMemory VSL をオペレーティング システムに接続します。

性能とチューニング

ioMemory デバイスは、広い帯域幅と高い IOPS (1 秒当たりの入出力操作数) を提供し、短いレイテンシーを実現するための特別な設計になっています。

ioMemory デバイスは IOPS とレイテンシーを向上させますが、デバイスの性能はオペレーティング システムの設定および BIOS の設定によって制限される可能性があります。ioMemory デバイスの性能を最大限に活用するためには、これらの設定のチューニングが必要になることがあります。

通常、Fusion-io デバイスはそのままで良好な性能を発揮しますが、このセクションでは、チューニングによって最適な性能を引き出せる可能性があるいくつかの領域について説明します。

DVFS の無効化

動的電圧・周波数制御 (DVFS) は、CPU の電圧または周波数を調整して CPU による消費電力を削減する電源管理手法です。これらの手法は、電力を節約し、CPU による熱の発生を抑えるのに役立ちますが、CPU が低電力状態と高性能状態との間で遷移することによって性能への悪影響が生じます。

これらの節電手法は、I/O レイテンシーおよび最大 IOPS に悪影響を与えることが知られています。性能の最大化を目的としてチューニングを行う場合、DVFS を制限するか完全に無効化することで効果を期待できますが、一方で消費電力は増加する可能性があります。

DVFS を使用できる場合は、オペレーティング システムの電源管理機能およびシステムの BIOS インターフェイスで設定を変更できます。DVFS の設定は、多くの場合、オペレーティング システムおよび BIOS 内の Advanced Configuration and Power Interface (ACPI) セクションにあります。詳細については、コンピューターのドキュメントを参照してください。

ACPI C ステートの制限

比較的新しいプロセッサは、使用率が低いときに低電力モードに切り替わる機能を備えています。このようなアイドル状態は ACPI C ステートと呼ばれます。C0 ステートは通常の最大電力での動作状態を意味し、それより上位の C ステート (C1、C2、C3 など) は低電力状態を意味します。

これらの ACPI C ステートは節電に役立ちますが、I/O レイテンシーおよび最大 IOPS に悪影響を与えることが知られています。通常は、C ステートのランクが上がるにつれて、節電のためにより多くのプロセッサ機能を制限するようになり、プロセッサを C0 ステートに復帰するまでの時間が長くなります。

これらの節電手法は、I/O レイテンシーおよび最大 IOPS に悪影響を与えることが知られています。性能の最大化を目的としてチューニングを行う場合、C ステートを制限するか完全に無効することで効果を期待できますが、一方で消費電力は増加する可能性があります。

ACPI C ステート オプションの設定

プロセッサで ACPI C ステートを利用できる場合は、通常、BIOS インターフェイス(セットアップ ユーティリティと呼ばれることもあります)でこれらを制限または無効化できます。ACPI C ステートの設定は、Advanced Configuration and Power Interface (ACPI) メニューなどに含まれています。詳細については、コンピューターのドキュメントを参照してください。

Linux における C ステート

比較的新しい Linux カーネルに含まれる一部のドライバーは、BIOS で ACPI C ステートが無効になっていても、ACPI C ステートを有効化しようとする場合があります。Linux では、カーネル起動オプションに次の行を追加することで、(BIOS 設定にかかわらず) C ステートを制限できます。

```
processor.max_cstate=1
```

この例では、最大で C1 までの C ステートが許可されます。C ステートを完全に無効にするには、値を 0 に設定します。

NUMA アフィニティの設定

NUMA (Non-Uniform Memory Access) アーキテクチャを備えたサーバでは、ioMemory デバイスの性能を最大限発揮するために特別なインストール手順が必要です。このようなサーバには、HP DL580、HP DL980、IBM 3850 サーバなどがあります。

NUMA アーキテクチャを備えたサーバでは、システムの起動時に BIOS によって NUMA ノード間で PCIe スロットが均等に配分されない場合があります。各 NUMA ノードには、複数の CPU が含まれています。スロットが均等に配分されないと、作業負荷が高まった場合に、半分以上の CPU がアイドル状態であるにもかかわらず、残りの CPU の使用率が 100% になる可能性があります。このような不均等を回避するために、利用可能な NUMA ノードに対して、ioMemory デバイスを手動で均等に割り当てる必要があります。

このアフィニティの設定の詳細については、「[付録 F - NUMA 構成](#)」を参照してください。

割り込みハンドラーのアフィニティの設定


NUMA システムに割り込みを割り当てることによって、デバイスのレイテンシーが影響を受ける場合があります。デバイスの割り込みは、アプリケーションが I/O を発行するのと同じ NUMA ソケットに割り当てる

ことをお勧めします。ユーザー アプリケーションのタスクによってこのソケットの CPU の使用率が非常に高くなる場合は、割り込みをリモート ソケットに移動してシステムの負荷を分散することによって性能を向上できる可能性があります。


多くのオペレーティング システムでは動的な割り込みの割り当てが試みられるため、通常は適切に負荷が分散されます。割り込みの割り当ての手動での調整は、特定のハードウェアにおけるアプリケーションのパフォーマンスについての分析を必要とする高度なオプションです。特定のデバイスの割り込みを特定の CPU に固定する方法の詳細については、オペレーティング システムのドキュメントを参照してください。

付録 A – ユーティリティ リファレンス

ioMemory VSL インストール パッケージには、さまざまなコマンドライン ユーティリティが含まれており、デフォルトでは各パッケージが `/usr/bin` にインストールされます。これらのユーティリティは、デバイスに対するアクセス、テスト、および操作に役立つ数々の方法を提供します。

 `/usr/bin` ディレクトリには、ここに示されていない追加のユーティリティもインストールされます。これらの追加のユーティリティは、他のユーティリティに依存するユーティリティ（メイン VSL ユーティリティによって使用されるユーティリティ）であり、カスタマー サポートによって指示された場合を除いて直接使用しないでください。

ユーティリティ	目的
<code>fio-attach</code>	ioMemory デバイスを OS で利用可能にします。
<code>fio-beacon</code>	ioMemory デバイスの外部 LED を点灯します。
<code>fio-bugreport</code>	問題のトラブルシューティングに使用するための詳細レポートを作成します。
<code>fio-detach</code>	一時的に ioMemory デバイスを OS のアクセスから切断します。
<code>fio-format</code>	ioMemory デバイスのローレベル フォーマットの実行に使用します。
<code>fio-pci-check</code>	PCI バス ツリーのエラー（特に ioMemory デバイスに関するもの）をチェックします。
<code>fio-snmp-agentx</code>	ioMemory VSL の SNMP FUSION-IODRV-MIB を実装する SNMP サブエージェントです。
<code>fio-status</code>	デバイスの情報を表示します。
<code>fio-sure-erase</code>	デバイスからデータをクリアまたはパーズします。
<code>fio-update-iodrive</code>	ioMemory デバイスのファームウェアを更新します。

 どのユーティリティでも、`-h` オプション（ヘルプ表示）と `-v` オプション（バージョン表示）を使用できます。`-h` および `-v` を使用すると、情報の表示後にユーティリティが終了します。

fio-attach

説明

ioMemory デバイスを接続して、オペレーティング システムで利用できるようにします。これにより、`fiox` という名前のブロック デバイスが `/dev` に作成されます（`x` の部分は `a`、`b`、`c` など置き換えられます）。接続後、ioMemory デバイスのパーティション分割やフォーマットを実行したり、ioMemory デバイスを

RAID アレイの一部としてセットアップしたりできます。このコマンドを実行すると、進捗状況を示すプログレス バーおよびパーセンテージが表示されます。

i ほとんどの場合、ioMemory VSL は、読み込み時に自動的にデバイスを接続して、スキャンを実行します。fio-attach は、fio-detach を実行した場合や、ioMemory VSL の auto_attach パラメータを 0 に設定した場合にのみ実行する必要があります。

構文

fio-attach <デバイス> [オプション]

<デバイス>には、デバイス ノード(/dev/fctx)の名前を指定します(x の部分には、0、1、2 などのカード番号を指定します)。たとえば、/dev/fct0 は、システムにインストールされている 1 つ目の ioMemory デバイスを示します。

複数の ioMemory デバイスを指定できます。たとえば、/dev/fct1 /dev/fct2 は、システムにインストールされている 2 つ目と 3 つ目の ioMemory デバイスを示しています。ワイルドカードを使用して、システム上のすべての ioMemory デバイスを指定することもできます(例: /dev/fct*)。

オプション	説明
-c	正常にシャットダウンされている場合にのみ接続します。
-q	表示抑制: プログレス バーおよびパーセンテージの表示を無効にします。

fio-beacon

説明

デバイスの位置を識別するために ioMemory デバイスの LED を点灯します。最初に ioMemory デバイスを切断してから、fio-beacon を実行する必要があります。

構文

fio-beacon <デバイス> [オプション]

<デバイス>には、デバイス ノード(/dev/fctx)の名前を指定します(x の部分には、0、1、2 などのカード番号を指定します)。たとえば、/dev/fct0 は、システムにインストールされている 1 つ目の ioMemory デバイスを示します。

オプション	説明
-0	オフ: (ゼロ)3 つの LED を消灯します。

-1	オン: 3 つの LED を点灯します。
-p	<デバイス>に指定したデバイスの PCI バス ID を標準出力に出力します。使用方法およびエラー情報は、標準エラー出力ではなく標準出力に出力される場合があります。

fio-bugreport

説明

問題のトラブルシューティングに使用するためのデバイスの詳細レポートを作成します。結果は/tmp ディレクトリ内のファイルに保存されます。保存先ファイル名には、このユーティリティの実行された日時が反映されます。

例:

```
/tmp/fio-bugreport-20100121.173256-sdv9ko.tar.bz2
```

構文

```
fio-bugreport
```

注記

このユーティリティは、デバイスの現在の状態を取り込みます。デバイスで性能または安定性に問題が発生している場合は、fio-bugreport ユーティリティを実行し、その出力を保存できます。

サンプル出力

```
-bash-3.2# fio-bugreport
Collecting fio-status -a
Collecting fio-status
Collecting fio-pci-check
Collecting fio-pci-check -v
Collecting fio-read-lebmap /dev/fct0
Collecting fio-read-lebmap -x /dev/stdout/dev/fct0
Collecting fio-read-lebmap -t /dev/fct0
Collecting fio-get-erase-count/dev/fct0
Collecting fio-get-erase-count -b /dev/fct0
Collecting lspci
Collecting lspci -vvvvv
Collecting lspci -tv
Collecting messages file(s)
Collecting procfusion file(s)
Collecting lsmod
Collecting uname -a
Collecting hostname
Collecting sar -r
Collecting sar
Collecting sar -A
```



```
Collecting syslog file(s)
Collecting proc file(s)
Collecting procirq file(s)
Collecting dmidecode
Collecting rpm -qa iodrive*
Collecting find /lib/modules
Please send the file /tmp/fio-bugreport-20090921.173256-sdv9ko.tar.bz2
along with your bug report to support@fusionio.com The file is in the /tmp
directory.
```


たとえば、`/tmp/fiobugreport-20090921.173256-sdvk0.tar.bz2` という名前のバグ レポート ファイルから次のことがわかります。

- 日付(2009 年 9 月 21 日)
- 時刻(17 時 32 分 56 秒)
- その他の情報(sdv9ko.tar.bz2)

fio-detach

説明

ioMemory デバイスを切断し、対応する ioMemory デバイスのブロック デバイス `fctx` を OS から削除します。`fio-detach` ユーティリティは、デバイスでのすべての読み込み/書き込み動作が完了するのを待ってから切断操作を実行します。このコマンドを実行すると、デフォルトで、切断の進行状況を示すプログレス バーおよびパーセンテージも表示されます。

 このユーティリティを使用する前に、切断するデバイスが現在マウントされていないこと、また使用されていないことを確認してください。

構文

```
fio-detach <デバイス> [オプション]
```

<デバイス>には、デバイス ノード(`/dev/fctx`)の名前を指定します(xの部分には、0、1、2などのカード番号を指定します)。たとえば、`/dev/fct0` は、システムにインストールされている 1 つ目の ioMemory デバイスを示します。

複数の ioMemory デバイスを指定できます。たとえば、`/dev/fct1 /dev/fct2` は、システムにインストールされている 2 つ目と 3 つ目の ioMemory デバイスを示しています。ワイルドカードを使用して、システム上のすべての ioMemory デバイスを指定することもできます(例: `/dev/fct*`)。

オプション	説明
<code>-i</code>	即時: 即時に強制的に切断を行います(メタデータは保存されません)。デバイスが OS によって使用中の場合、この操作は失敗します。

-q	表示抑制: プログレス バーおよびパーセンテージの表示を無効にします。
----	-------------------------------------

注記

このバージョンの ioMemory VSL では、デバイスが使用中であるというエラーが表示されて、ioMemory デバイスの切断に失敗する場合があります。通常、この状況は ioMemory デバイスがソフトウェア RAID (0、1、5) ボリュームの一部として使用されている場合、マウントされている場合、または何らかのプロセスで開かれている場合に発生します。

デバイスが何によって開かれているのかの判断には、`fuser`、`mount`、および `lsdf` の各ツールが役立ちます。

fiio-format

説明

i ioMemory デバイスは、あらかじめフォーマットされた状態で出荷されます。そのため、デバイスの論理サイズやブロック サイズを変更する場合や、デバイス上のユーザー データを消去する場合を除き、通常は `fiio-format` を実行する必要がありません。ユーザー データのより完全な消去を実施するには、[fiio-sure-erase](#) を使用します。

ioMemory デバイスのローレベル フォーマットを実行します。`fiio-format` を実行すると、デフォルトで、フォーマットの進行状況を示すプログレス バーおよびパーセンテージによるインジケーターが表示されません。

w このユーティリティを実行するとデバイス上のすべてのユーザー情報が削除されるので、慎重に使用してください。


i 4096 バイトなどの大きなブロック(セクター)サイズを使用すると、ioMemory VSL ホストのメモリー消費が最も多くなる場合のメモリー使用量を大幅に削減できます。ただし、一部のアプリケーションは 512 バイト以外のセクター サイズと互換性がありません。

i `-s` または `-o` オプションを指定しない場合、デバイスのサイズはデフォルトの公表されている容量に設定されます。`-s` および `-o` オプションを指定する場合は、サイズまたはパーセンテージも指定する必要があります。

構文

```
fiio-format [オプション] <デバイス>
```

<デバイス>には、デバイス ノード(`/dev/fctx`)の名前を指定します(x の部分には、0、1、2 などのデバイス番号を指定します)。たとえば、`/dev/fct0` は、システムにインストールされている 1 つ目の ioMemory デバイスを示します。

オプション	説明
-b <サイズ B K>	ブロック(セクター)サイズをバイトまたはキロバイト単位で指定します(キロバイトの計算は 2 を底とする指数を使用します)。デフォルトは 512 バイトです。たとえば、-b 512B または -b 4K と指定します(512B における B の指定は任意です)。
-f	通常のチェックおよび警告をバイパスして、フォーマット サイズを強制的に適用します。このオプションは、 <code>fio-format</code> が適切に進行しない例外的な状況で必要になる場合があります(-y オプションを使用しない場合は、“Are you sure?”(実行してもよろしいですか?)のプロンプトが表示されます)。
-q	表示抑制モード: プログレス バーおよびパーセンテージの表示を無効にします。
-s <サイズ M G T %>	<p>デバイスの容量を具体的なサイズ(TB、GB、MB 単位)または公表されている容量に対するパーセンテージで設定します。</p> <ul style="list-style-type: none"> • T: フォーマットするテラバイト(TB)数 • G: フォーマットするギガバイト(GB)数 • M: フォーマットするメガバイト(MB)数 • % パーセンテージ(70%など。パーセント記号の指定が必要)
-o <サイズ B K M G T %>	<p>デバイス サイズを公表されている容量よりも大きいサイズにオーバーフォーマットします。最大サイズは、最大物理容量です。パーセンテージを使用する場合は、デバイスの最大物理容量に対する割合を指定します(-o オプションにはサイズを指定する必要があります。サイズの単位の説明については、上記の-s オプションを参照してください)。</p> <p> このオプションを使用する前に、カスタマー サポートと具体的な使用例について相談してください。</p>
-R	正常にシャットダウンされていない場合の高速再スキャンを無効にします。この場合、容量の一部が使用できなくなる可能性があります。
-y	アプリケーションからのすべての問い合わせに対して自動的に「yes(はい)」と回答します(プロンプトをバイパスします)。

fio-pci-check

説明

PCI バス ツリーのエラー、特に ioMemory デバイスをチェックします。このユーティリティは、各 ioMemory デバイスの現在のステータスを表示します。PCI-Express の標準エラー情報の出力、および状態のリセットも行います。

❗ 最初に `fio-pci-check` を実行したときにいくつかのエラー（通常は最大 5 つ程度のエラー）が表示されますが、問題ありません。正常に動作している場合、2 回目以降は、数時間の動作中に 1 つまたは 2 つのエラーのみが発生します。

構文

`fio-pci-check` [オプション]

オプション	説明
<code>-d <値></code>	1 = リンクを無効にします。0 = リンクをアップ状態にします（推奨されません）。
<code>-e</code>	PCI-e エラー レポートを有効にします。
<code>-f</code>	システム内のすべてのデバイスをスキャンします。
<code>-i</code>	デバイスのシリアル番号を出力します。このオプションは、ioMemory VSL の読み込み時には無効です。
<code>-r</code>	リンクを維持されたままにします。
<code>-v</code>	詳細: ハードウェアについての追加データを出力します。

fio-snmp-agentx

説明

このユーティリティは、ioMemory VSL の SNMP FUSION-IODRV-MIB を実装する SNMP サブエージェントです。`fio-snmp-agentx` は、agentx プロトコルを介して SNMP マスター エージェントと通信します。

構文

`fio-snmp-agentx` [オプション]

オプション	説明
<code>-f</code>	サブエージェントをデーモンとしてではなく、強制的にフォアグラウンドで実行します。
<code>-l <ログ ファイル></code>	使用するログ ファイルです。
<code>-s</code>	Syslog の代わりに標準エラー出力にエラーを送信します。

fio-status

説明

取り付けられているデバイスに関する詳細情報を提供します。このユーティリティは、`fctx` または `fiox` デバイスに対して動作します。このユーティリティは root として実行する必要があります。また、このユー

ティリティを実行する場合は、ioMemory VSL が読み込まれている必要があります。ioMemory VSL が読み込まれていない場合は、返されるステータス情報が少なくなります。


`fio-status` では、最小モード(minimal-mode)、読み取り専用モード(read-only mode)、書き込み低減モード(write-reduced mode)などの特定のエラー モードに対して、警告およびその状態が発生した原因の説明が表示されます。

構文

`fio-status` [<デバイス>] [<オプション>]

<デバイス>には、デバイス ノード(/dev/fctx)の名前を指定します(x の部分には、0、1、2 などのカード番号を指定します)。たとえば、/dev/fct0 は、システムにインストールされている 1 つ目の ioMemory デバイスを示します。

<デバイス>が指定されない場合、`fio-status` はシステムのすべてのカードについて情報を表示します。ioMemory VSL が読み込まれていない場合、このパラメータは無視されます。

オプション	説明
-a	各デバイスで入手可能なすべての情報が報告されます。
-e	各デバイスのすべてのエラーおよび警告が表示されます。このオプションは問題の診断用であり、フォーマット サイズなどのその他の情報は表示されません。
-c	カウント: 取り付けられている ioMemory デバイスの数のみが報告されます。
-d	基本的な情報セット、および読み書きされたデータの合計量(累積データ量)が表示されます。-a オプションを指定した場合、このオプションは不要です。
-fj	JSON 形式: JSON 形式で出力を作成します。
-fx	XML 形式: XML 形式で出力を作成します。
-u	情報を入手できないフィールドを表示します。-fj または -fx と共に指定した場合にのみ有効です。
-U	情報を入手できないフィールド、およびその詳細な理由を表示します。-fj または -fx と共に指定した場合にのみ有効です。
-F<フィールド>	1 つのフィールドの値を出力します(フィールド名については次のオプションを参照)。デバイスを指定する必要があります。-F オプションは、複数指定できます。
-l	-F を使用して個別に指定できるフィールドを列挙します。
	出力の変更 バージョン 3.0.0 以降、 <code>fio-status</code> の出力の標準フォーマットが変更されました。この変更によって、このユーティリティの出力を使用していたすべてのカスタム管理ツールに影響があります。

基本情報: オプションを指定しない場合、`fio-status` では、次の基本情報が報告されます。

- システムに取り付けられているデバイスの数と種類
- ioMemory VSL のバージョン

アダプター情報:

- アダプター タイプ
- 製品番号
- 外部電源の状態
- PCIe 電力制限しきい値(利用可能な場合)
- 接続されている ioMemory デバイス

ブロック デバイス情報:

- 接続ステータス
- 製品名
- 製品番号
- シリアル番号
- PCIe アドレスおよびスロット
- ファームウェア バージョン
- デバイスのサイズ、および合計容量
- 平均温度(ioMemory VSL の読み込み以降の平均および最大、摂氏)
- ヘルス(正常性)ステータス: 正常(healthy)、消耗が限界に近付いている(nearing wearout)、書き込み低減(write-reduced)、読み取り専用(read-only)
- 予約容量(パーセンテージ)
- 警告容量しきい値(パーセンテージ)

データ ボリューム情報: -d オプションを指定した場合は、基本情報に加えて次のデータ ボリューム情報が報告されます。

- 書き込まれた物理バイト数
- 読み込まれた物理バイト数

すべての情報: -a オプションを指定した場合は、すべての情報(基本情報、データ ボリューム情報、および次に示す情報)が出力されます。

アダプター情報:

- 製造業者番号
- 部品番号
- 製造日付
- 停電時の保護ステータス
- PCIe バスの電圧(平均、最小、最大)

- PCIe バスの電流量(平均、最大)
- PCIe バス電力(平均、最大値)
- PCIe 電力制限のしきい値(ワット)
- PCIe スロットで利用可能な電力(ワット)
- PCIe でネゴシエートされたリンク情報(レーンおよびスループット)


ブロック デバイス情報:


- 製造業者のコード
- 製造日付
- ベンダーおよびサブベンダー情報
- フォーマット ステータスおよびセクター情報(デバイスが接続されている場合)
- FPGA ID およびローレベル フォーマット GUID
- PCIe スロットで利用可能な電力
- PCIe でネゴシエートされたリンク情報
- カードの温度(摂氏)
- 内部電圧(平均、最大値)
- 補助電圧(平均、最大値)
- 正常なブロック、データ、およびメタデータのパーセンテージ
- 累積データ ボリューム統計
- RAM の使用量


エラー モード情報: ioMemory VSL が最小モード(minimal mode)、読み取り専用モード(read-only mode)、または書き込み低減モード(write-reduced mode)の場合に `fio-status` を実行すると、これらのモードではない場合と比較して、出力に次の相違点があります。


- 接続状態が“Status unknown: Driver is in MINIMAL MODE:”(状態不明: ドライバーが最小モードになっています)になります。
- 最小モードになっている理由が表示されます(“Firmware is out of date. Update firmware.”(ファームウェアが期限切れです。ファームウェアを更新してください)など)。
- 「Geometry and capacity information not available.(形状および容量の情報は利用できません。)」と表示されます。
- メディアの正常性についての情報は表示されません。

fio-sure-erase

 ベスト プラクティスとして、クリアまたはパーズしたくない ioMemory デバイスがシステムに取り付けられている場合は、このユーティリティを使用しないことをお勧めします。最初に、誤って消去したくないすべてのデバイスを取り外してください。このユーティリティを使用してデータを削除すると、データは完全に削除されます。**復元することはできません。**

 このユーティリティを使用する前に、保護する必要があるすべてのデータをバックアップしてください。

 `fio-sure-erase` を使用した後、再度デバイスを使用する前に、[fio-format](#) を使用してデバイスをフォーマットします。

 デバイスが読み取り専用モード(read-only mode)の場合は、`fio-sure-erase` を実行する前に `fio-format` を使用してフォーマットを実行します。デバイスが最小モード(minimal mode)の場合は、`fio-sure-erase` を使用してデバイスを消去することはできません。ファームウェアを更新すると、デバイスの最小モード(minimal mode)が解除される可能性があります。

`fio-sure-erase` を実行するためには、ブロック デバイスを切断しておく必要があります。詳細については、「[fio-detach](#)」を参照してください。

説明

`fio-sure-erase` は、ioMemory デバイスから確実にデータを削除するコマンドライン ユーティリティです。このユーティリティは、次の標準規格における「Clear(クリア)」および「Purge(パージ)」レベルの破壊に準拠しています。


1. DOD 5220.22-M – フラッシュ EPROM に関する指示に準拠
2. NIST SP800-88 – フラッシュ EPROM に関する指示に準拠


クリアおよびパージのサポートの詳細については、後の説明を参照してください。

構文

`fio-sure-erase` [オプション] <デバイス>

<デバイス>には、デバイス ノード(/dev/fctx)の名前を指定します(xの部分には、0、1、2などのカード番号を指定します)。たとえば、/dev/fct0 は、システムにインストールされている1つ目のioMemory デバイスを示します。このデバイス ノードの情報を表示するには、[fio-status](#) を使用します。

 **複数のデバイスを備えた製品**
`fio-sure-erase` は、個別のioMemory デバイスに対して動作します。たとえば、ioDrive Duo デバイスのパージを計画している場合は、この操作をioDrive Duo デバイスの2つのioMemory デバイスそれぞれに対して実行する必要があります。

オプション	説明
-p	クリアではなくパージ: 書き込み、消去の順で実行します。パージの詳細については、後の説明を参照してください。  パージする必要があるデバイスのサイズによっては、デバイスのパージが完了するまでに数時間かかることがあります。
-y	確認なし: ユーティリティの実行時に「yes(はい)」または「no(いいえ)」を入力する必要がありません。
-q	表示抑制: ステータス バーを表示しません。

i オプションを指定しないで `fio-sure-erase` を実行すると、クリアが実行されます。詳細については、後の説明を参照してください。

メモリーの各ブロックは、1 または 0 のビットで均一に埋められます。

クリアのサポート

「クリア」は、`fio-sure-erase` を(オプションを指定しないで)実行するデフォルトの実行状態であり、書き込み回数超過消去ブロックを含む NAND メディア全体に対して完全なローレベル消去(すべてのセルの「1」への書き換え)を実行します。

動作に必要なメタデータ(メディア イベント ログ、消去回数、読み書きされた物理バイト数、温度履歴)は破棄されませんが、ユーザー固有のメタデータは破棄されます。

次に、クリア操作で実行される手順について説明します。

1. すべてのアドレス指定可能なブロックの統一マップが作成されます(これにより、以前にマッピングが解除された不良ブロックを含むすべてのブロックをアドレス指定できます)。
2. 各ブロックに対して、消去サイクルが実行されます(すべてのセルが「1」に書き換えられます)。
3. 不良ブロック マップが復元されます。
4. このユーティリティでは、クリア時にすべてのヘッダーが消去されるので、再度使用可能にするためにデバイスがフォーマットされます。

ページのサポート

「ページ」は、`fio-sure-erase` に `-p` オプションを指定することによって実行します。ページでは、最初に(書き込み回数超過消去ブロックを含む) NAND メディア全体が 1 つの文字で上書きされて(すべてのセルに論理表現の「0」が書き込まれます)、その後(書き込み回数超過消去ブロックを含む)メディア全体に対して全チップの消去(すべてのセルの「1」への書き換え)が実行されます。

動作に必要なメタデータ(メディア イベント ログ、消去回数、読み書きされた物理バイト数、温度履歴)は破棄されませんが、ユーザー固有のメタデータは破棄されます。

次に、ページ操作で実行される手順について説明します。

1. すべてのアドレス指定可能なブロックの統一マップが作成されます(これにより、以前にマッピングが解除された不良ブロックを含むすべてのブロックをアドレス指定できます)。
2. 各ブロックに対して、書き込みサイクルが実行されます(すべてのセルが「0」に書き換えられます)。
3. 各ブロックに対して、消去サイクルが実行されます(すべてのセルが「1」に書き換えられます)。
4. 不良ブロック マップが復元されます。

5. このユーティリティでは、クリア時にすべてのヘッダーが消去されるので、再度使用可能にするためにドライブがフォーマットされます。

fiio-update-iodrive

⚠ アップグレードを行う前に、念のため ioMemory デバイスのデータをバックアップする必要があります。

説明

ioMemory デバイスのファームウェアを更新します。このユーティリティでは、すべての ioMemory デバイスを検出するために PCIe バスがスキャンされて、検出されたデバイスが更新されます。各デバイスに対して更新の進行状況を示すプログレス バーおよびパーセンテージが表示されます。

- ⊖ デバイスで障害が発生する可能性があるため、ファームウェアのアップグレード中は、絶対に電源をオフにしないでください。UPS が設置されていない場合は、ファームウェアのアップグレードを行う前にシステムに UPS を追加することを検討してください。
- ⊖ ファームウェアのアップグレードを複数回連続して実行する場合は、ファームウェアを 1 回アップグレードすることにドライバーを読み込んでください。読み込まないと、ドライブ上のフォーマットが変更されず、データが失われます。
- ⊖ このユーティリティを使用して、ioMemory デバイスを以前のバージョンのファームウェアにダウングレードしないでください。ダウングレードを行うと、データが失われる危険性があり、また保証が無効となります。
- ⚠ デフォルトの動作 (-d オプションも -s オプションも使用しない場合)は、<iodrive_version.fff> ファイルに含まれているファームウェアを使用してすべての ioMemory デバイスをアップグレードすることです。更新の実行前に、すべてのデバイスでアップグレードが必要であることを確認してください。確信を持っていない場合は、-p (疑似実行) オプションを指定して、更新の推定結果を表示します。
- ⚠ ファームウェアを更新する前に、すべての ioMemory デバイスを切断する必要があります。
- ⚠ **アップグレードパス**
ioMemory デバイスのアップグレード時には、所定のアップグレードパスがあります。ioMemory デバイスをアップグレードする前に、このリリースの ioMemory VSL のリリース ノートを参照してください。
- ⓘ ファームウェアの更新時に MIDPROM 情報の更新を指示するエラー メッセージが表示された場合は、カスタマー サポートにお問い合わせください。

1 つ以上の特定のデバイスを更新するには:

- ioMemory VSL が読み込まれている場合は、-d オプションを使用してデバイス番号を指定します。

構文

fio-update-iodrive [オプション] <iodrive_version.fff>

<iodrive_version.fff>は、Fusion-io による指定のパスとファームウェア アーカイブ ファイルです。デフォルトのパスは /usr/share/fio/firmware です。このパラメータは必須です。

オプション	説明
-d	fctx(x は fio-status で表示されるデバイスの番号)で指定されたデバイスを更新します。このオプションを指定しない場合、すべてのデバイスが更新されます。 ⚠ 適切でない ioMemory デバイスを更新するとデバイスが破損する可能性があるため、-d または -s オプションは注意して使用してください。
-f	強制アップグレードを行います(主として以前のファームウェアバージョンへのダウングレードに使用します)。ioMemory VSL が読み込まれていない場合、このオプションは -s オプションを必要とします。 ⚠ -f オプションはカードの破損につながる可能性があるため、注意して使用してください。
-l	アーカイブ内の使用可能なファームウェアを一覧表示します。
-p	疑似実行:行われる更新の内容を表示します。ただし、実際のファームウェア変更は行われません。
-c	デバイスに設定されているロックをクリアします。
-q	プログレス バーやパーセンテージを表示せずに更新処理を実行します。
-y	すべての警告メッセージを承認します。
-s	'*'をデバイス用のワイルドカードとして使用して指定されたスロット内のデバイスを更新します。スロットは、(lspci で表示される)次の PCIe フォーマットで特定されます。 [[[<ドメイン>]:<バス>]:<スロット>]. [<機能>]]

アップグレード処理の実行中には、3 つの外部 LED インジケータがすべて点灯します。

付録 B – ioMemory デバイスの正常性の監視

このセクションでは、データの保護とデバイス耐用期間の延長のために ioMemory デバイスの正常性を計測して監視する方法を説明します。

NAND フラッシュとコンポーネントの障害

ioMemory デバイスは、コンポーネントの障害やソリッドステート ストレージの消耗特性に対する保護を多くのレベルで提供する、フォールトトレランス性の高いサブシステムです。然しながらどのようなストレージ サブシステムでも、コンポーネントに障害が発生する可能性があります。

デバイスの使用期間と正常性を予防的に監視することで、想定された製品寿命が尽きるまで信頼できる性能を確保できます。

正常性ステータスの指標

ioMemory VSL は、事前に定められた使用停止しきい値を使用して、ブロックの使用率を監視します。ioManager および `fio-status` ユーティリティでは、開始時の値が 100 で 0 までカウント ダウンされる正常性のインジケータが表示されます。一定のしきい値を下回ると、各種の措置が取られます。

10%の正常性しきい値では、警告が一度表示されます。このアラーム イベントを捕捉する方法については、以下の「[正常性の監視手法](#)」を参照してください。

0%になると、デバイスは正常でないと見なされます。その場合、デバイスは書き込み低限モード (write-reduced mode) に入ります。これは、データをデバイス外に安全に移行できるように、デバイスの寿命をある程度延ばすためのモードです。この状態では、書き込み性能が低下する点を除き、ioMemory デバイスは通常どおりに動作します。

0%のしきい値に達した後、デバイスはやがて読み取り専用モード (read-only mode) に入ります。このモードでは、ioMemory デバイスへの書き込みを行おうとしてもエラーになります。一部のファイルシステムでは、読み取り専用のブロック デバイスをマウントするために、そのマウントが読み取り専用であることを指定する以外に、特別なマウント オプションが必要になる場合があります。

たとえば、Linux の場合は、`ext3` で "`-o ro,noload`" を使用する必要があります。"`noload`" オプションは、ジャーナルの試行とリプレイを行わないようにファイルシステムに指示します。

読み取り専用モード (read-only mode) は、デバイスからデータを移行する最後の機会であると考えられています。デバイスに障害が発生する可能性は連続使用によって高まるからです。

ioMemory デバイスは、障害モードに入ることがあります。その場合、デバイスはオフライン状態になり、アクセス不能になります。この状態は、内部の致命的な障害、ファームウェアの不適切なアップグレード手順、またはデバイスの消耗によって引き起こされます。

- ① サービスまたは保証に関連するご質問については、デバイスの購入先にお問い合わせください。
- ② 複数の ioMemory デバイスを備えた製品の場合、これらのモードはデバイスごとに独立して管理されます。

正常性(ヘルスマonitoring)の監視手法

`fio-status:fio-status` ユーティリティの出力には、正常性のパーセンテージとデバイスの状態が表示されます。これらの項目は、以下のサンプル出力の"Media status"の部分に示されています。

```
Found 3 ioMemory devices in this system
Fusion-io driver version:3.0.6 build 364
```

```
Adapter: Single Adapter
```

```
Fusion-io ioDrive 1.30TB, Product Number:F00-001-1T30-CS-0001, SN:1133D0248,
FIO SN:1134D9565
```

```
...
```

```
Media status:Healthy; Reserves:100.00%, warn at 10.00%; Data: 99.12%
```

```
Lifetime data volumes:
```

```
Physical bytes written: 6,423,563,326,064
```

```
Physical bytes read : 5,509,006,756,312
```

ioManager の場合:[Device Report(デバイス レポート)] タブの右側の列にある[Reserve Space]のパーセンテージを参照します。このパーセンテージが高いほど、ドライブの正常性は高いと考えられます。

SNMP/SMI-S の場合:Windows または Linux での SNMP または SMI-S 正常性インジケータの構成方法の詳細については、対応する付録を参照してください。

`fio-status` ユーティリティでは、正常性の状態に関する以下のメッセージが生成されます。

- Healthy(正常)
- Read-only(読み取り専用)
- Reduced-write(書き込み低減)
- Unknown(不明)

ソフトウェア RAID と正常性監視


ソフトウェア RAID スタックは、通常、従来のストレージ メディアの障害モードを検出して、その影響を軽減できるように設計されています。ioMemory デバイスは、障害の影響を可能な限り軽減しようと試み

ますが、これらの新しい障害対応メカニズムは既存のソフトウェア RAID スタックと互換性があります。RAID グループ内の ioMemory デバイスでは、a) デバイスが書き込み低減 (write-reduced) 状態になり、b) 書き込み作業負荷が高い場合に、十分な速度でデータを書き込むことができなくなります。この場合、デバイスは RAID グループから切り離されます。読み取り専用モード (read-only mode) のデバイスは、デバイスから書き込み I/O に失敗したというエラーが返されたときに切り離されます。回復不可能な障害は、従来のストレージ デバイスでの障害と同様に検出および処理されます。


付録 C – モジュール パラメータの使用

次の表に、`/etc/modprobe.d/iomemory-vsl.conf` ファイルを編集して値を変更することで設定できるモジュール パラメータを示します。

この構成ファイル内の各モジュール パラメータの前には `options iomemory-vsl` が必要です。`/etc/modprobe.d/iomemory-vsl.conf` ファイルでは、いくつかのサンプル パラメータがコメントアウトされています。これらのサンプルはテンプレートとして使用したり、コメントを解除してそのまま使用したりできます。

 これらの変更を有効にするには、ioMemory VSL の読み込み前に変更を完了する必要があります。		
モジュール パラメータ	デフォルト (最小/最大)	説明
<code>auto_attach</code>	1	1=ドライバーの読み込み時に必ずデバイスに接続します。 0=ドライバーの読み込み時にデバイスに接続しません。
<code>external_power_override</code>	デバイス是非 選択	選択されたデバイスにおいて、PCIe スロットからすべての電力を供給することを許可します。このパラメータの<値>は、アダプター シリアル番号のコンマ区切りリストです。 ⚠ 慎重に指定してください。詳細については、「PCIe 電力優先 (PCIe Power Override) の有効化」を参照してください。
<code>fio_dev_wait_timeout_secs</code>	30	ドライバーの読み込み時に <code>/dev/fio*</code> ファイルを参照できるようにするまでに待つ秒数です。udev を使用していないシステムの場合は、0 に設定してタイムアウトを無効にし、ドライバー読み込み中の不要な一時停止を回避してください。
<code>force_minimal_mode</code>	0	1=デバイスを強制的に最小モード (minimal mode) にします。 0=デバイスを強制的には最小モード (minimal mode) にしません。
<code>numa_node_override</code>	選択対象なし	システム内のすべてのデバイスのアフィニティ設定を指定する<アフィニティの指定>カプレット (couplets) のリストです。カプレット内の各項目はコロンで、各カプレ

		<p>ットはコンマで区切ります。各<アフィニティの指定>カブレットの構文は次のとおりです。 <fct 番号>:<ノード番号>。</p> <p>このパラメータの使用法の詳細については、「付録 F - NUMA 構成」を参照してください。</p>
parallel_attach	1	<p>1=複数デバイスのパラレル接続を有効にします。 0=複数デバイスのパラレル接続を無効にします。</p>
preallocate_memory	デバイス是非選択	<p>選択されたデバイスに対し、ドライブをスワップ領域として使用できるようにするのに必要なすべてのメモリーを事前に割り当てます。このパラメータの<値>は、デバイス シリアル番号のコンマ区切りリストです。</p>
tintr_hw_wait	0(0、255)	<p>ハードウェア割り込みの待機間隔(マイクロ秒単位)です。割り込みコールスとも呼ばれています。0の場合はオフになります。</p>
use_workqueue	3(1 または 3)	<p>Linux 専用:3=OS の標準 I/O エレベーターを使用します。0=バイパスを行います。</p>

 preallocate_memory と external_power_override を除く各モジュール パラメータは、グローバルであり、コンピューター内のすべての Fusion デバイ스에適用されます。

付録 D – SNMP の設定

`fio-snmp-agentx` SNMP エージェントは、RFC 2741 準拠の AgentX サブエージェントです。このエージェントは、Net-SNMP など、任意の RFC 準拠 SNMP エージェントと共に動作します。マスター SNMP エージェントは、サポートされる MIB 上での `fio-snmp-agentx` クエリ参照に委ねられます。

SNMP マスター エージェント

`fio-util` パッケージに用意されている `fio-snmp-agentx` は、インストール済みの SNMP マスター エージェントを必要とします。SNMP マスター エージェントは、AgentX 接続をサポートし、また AgentX 接続向けに構成されている必要があります。`fio-snmp-agentx` は、Net-SNMP によってテストと検証が行われています。Net-SNMP は、ほとんどの Linux ディストリビューションに用意されている標準的な SNMP エージェントです。

この機能をサポートしている利用可能なエージェントはいくつかあります。Net-SNMP の使用を選択した場合は、以下のセクションの手順に従って Net-SNMP の構成と起動を行います。

SNMP マスター エージェントの起動

お使いの Linux のバージョン用のパッケージ マネージャーを使用して、Net-SNMP パッケージをインストールします。

Red Hat の場合

Net-SNMP を Red Hat にインストールするには、次のコマンドを使用します。

```
yum install net-snmp rsync
```

他の Linux バージョンの場合

Net-SNMP パッケージを Linux ディストリビューションにインストールするには、標準のシステム パッケージ マネージャーを使用します。`fio-snmp-mib` パッケージは、MIB ファイルを `/usr/share/fio/mib` に配置します。

マスター エージェントの構成

`snmpd.conf` テキスト ファイルを使用すると、ネットワーク通信パラメータ、セキュリティをはじめとするオプションを設定するように Net-SNMP マスター エージェント デモンを構成できます。このファイルの場所は、システムによって異なります。多くの場合は、`/etc/snmp` または `/usr/share/snmp` にあります。


簡単な `snmpd` 構成ファイルには、次のような記述が含まれます。

```
# set standard SNMP variables
syslocation "Data room, third rack"
syscontact itguy@example.com
# required to enable the AgentX protocol
master agentx
agentxsocket tcp:localhost:705
#set the port that the agent listens on (defaults to 161)
agentaddress 161
# simple access control (some form of access control is required)
rocommunity public
```

マスター エージェントの実行

マスター エージェントのインストールと構成が完了したら、新しいパラメータを有効にするために `snmpd` デーモンを起動または再起動する必要があります。`snmpd` は単純にそのインストール場所(多くの場合は `/usr/sbin`)から実行できます(オプションについては `snmpd` の man ページを参照してください)。通常、`snmpd` を適切に実行するには `root` 権限が必要です。また、`/etc/init.d` または `/etc/rc.d/init.d` にある `snmpd` 起動スクリプトを使用することもできます。セキュリティに関する懸念がある場合は、関連 man ページに記載の `rocommunity` および `rwcommunity` アクセス制御ディレクティブの代わりに高度な SNMPv3 アクセス制御を使用します。

Fusion-io の SNMP AgentX サブエージェント

 SNMP エージェントには、`libvsl` RPM パッケージが必要です。このパッケージは `ioMemory VSL` インストールの一部としてインストールされます。このパッケージは `PRIMERGY` ダウンロードページ <http://jp.fujitsu.com/platform/server/primergy/downloads/> からダウンロードできます。

Fusion SNMP サブエージェントのインストール

1. Fusion SNMP パッケージを [PRIMERGY ダウンロードページ](http://jp.fujitsu.com/platform/server/primergy/downloads/): <http://jp.fujitsu.com/platform/server/primergy/downloads/> からダウンロードします。
2. オペレーティング システムのパッケージ マネージャーを使用して、このパッケージをインストールします。たとえば、Red Hat では、次のコマンドを実行します。

```
rpm -Uvh fio-snmpp-*.rpm
```

`ioDrive` SNMP パッケージにより、その MIB ファイルが `/usr/share/fio/mib` に配置されます。

Fusion-io の SNMP サブエージェントの実行と構成

1. `fio-snmpp-agentx.conf` ファイルを作成して、サブエージェントを構成します。

2. この.conf ファイルを /opt/fio/etc/snmp ディレクトリに保存します。
3. 少なくとも、このファイル内のエージェント ネットワーク パラメータを次のように設定します。
4. # required to enable the AgentX protocol
5. agentxsocket tcp:localhost:705

この設定は、マスター エージェント用の snmpd.conf ファイルにある AgentX ネットワーク パラメータと一致している必要があります。

インストールと構成が完了すると、fio-snmp-agentx startup スクリプトがブート時に自動的に起動されます。

Fusion-io の SNMP サブエージェントの手動実行

Fusion-io の SNMP サブエージェントを手動で起動する必要がある場合は、以下の手順に従います。

1. SNMP マスター エージェントの起動後、次のコマンドを実行してサブエージェントを起動します。

```
/usr/bin/fio-snmp-agentx
```

このコマンドは、fio-snmp-agentx.conf という構成ファイルを使用して、Fusion-io サブエージェントを起動します。このファイルは、/opt/fio/etc/snmp ディレクトリの 1 つに存在する必要があります。

2. これで、SNMP MIB ブラウザー、または (/usr/share/fio/mib 内の) FIOioDrv.mib にアクセスするネットワーク管理システムを使用して、ioMemory デバイスの管理情報を表示できるようになります。

サブエージェントのログ ファイル

Fusion-io の SNMP サブエージェントは、自らのアクティビティに関するログ ファイルを管理できます。このファイルは、MIB から分離されており、エラーや断続的な問題など、サブエージェントとマスター エージェントとの通信に関するエントリを含んでいます。

サブエージェントでこのログ ファイルを管理するには、サブエージェント実行時にコマンドの一部として -l パラメータとログ ファイルへのパスを指定します。たとえば、次のコマンド

```
fio-snmp-agentx -l /usr/snmp/subagent.log
```

は、サブエージェントのログ ファイルを /usr/snmp ディレクトリ内の subagent.log として保持します。

これで、Fusion-io の SNMP サブエージェントはデバイスを監視できる状態になります。

SNMP サンプル構成ファイルの使用

SNMP をインストールすると、次のサンプル構成ファイルを使用できます。

- /usr/share/doc/fio-snmp-agentx/conf/snmpd.conf/(マスター エージェント用)
- /usr/share/doc/fio-snmp-agentx/conf/fio-snmp-agentx.conf/(サブエージェント用)

サンプル構成ファイルをカスタマイズして使用するには:

1. snmpd.conf ファイルと fio-snmp-agentx.conf ファイルの名前を(それぞれ snmpd-orig.conf、fio-snmp-agentx-orig.conf などに)変更します。通常、snmpd.conf ファイルは/etc/snmp または/usr/share/snmp にあります。fio-snmp-agentx.conf ファイルは/opt/fio/etc/snmp ディレクトリにあります。
2. /usr/share/doc/fio-snmp-agentx/conf/ディレクトリから、サンプルの snmpd.conf ファイルとサンプルの fio-snmp-agentx.conf ファイルをそれぞれ適切なディレクトリにコピーします。
3. コピーした各サンプル ファイルを編集して、変更結果をそれぞれ snmpd.conf、fio-snmp-agentx.conf として保存します。

SNMP テスト モードの有効化

SNMP AgentX は、実行時に fio-snmp-agentx config ファイルを読み取ります。

```
#####  
# Example config file for fio-snmp-agentx SNMP AgentX subagent.  
#  
# Fusion-io, Inc. #  
  
agentxsocket tcp:localhost:705  
  
# test_mode_enabled  
# set to 1, true or yes to enable 0, false or no to disable (default:false)  
test_mode_enabled true  
# traps_enabled  
traps_enabled true  
# testmode_file  
# name of test mode file (default:testmode.ini)  
testmode_file testmode.ini  
  
# update_delay  
# delay between agent polling requests in milliseconds (default: 250)  
update_delay 100  
  
# mib_select  
# set to fio for FUSIONIO-IODRV-MIB or cpq for CPQIODRV-MIB (default:fio)  
mib_select fio  
#####
```

以下にテスト モードの条件を示します。

1. 管理者が `test_mode_enabled` パラメータの設定を TRUE から FALSE に変更した場合、Fusion-io の SNMP はテスト モードの実行を試みません。代わりに、ioMemory VSL からのデータの処理を通常どおりに続行し、データを MIB に格納します。
2. CONF ファイルで `test_mode_enabled` が TRUE になっている場合、SNMP サブエージェントは `testmode.ini` を定期的に取り読んで変更の有無を確認します。サンプルの `testmode.ini` ファイルは、`/usr/share/doc/fio-snmp-agentx/conf` にインストールされています。
3. `testmode.ini` ファイルでテスト モードが ON に設定されている場合は、テスト モードになります。
4. テスト モードが ON の場合、SNMP Agentx は次の行 `TestModeIndex` を読み取って、テストする ioMemory モジュール (DIMM) を識別します。このパラメータ内の数値は、`fio-status` の使用時に表示される PCIe デバイス番号です (以下に例を示します)。

```
PCI:01:00.0
```

最初の 2 桁の数値は、PCIe バス番号を表しています (この場合は 01)。このバス番号は 16 進数で表示されるのに対し、`testmode.ini` ファイル内の `TestModeIndex` は 10 進数で指定する必要があります。`testmode.ini` には変換後の数値を入力してください。

`TestModeIndex` は、システムに取り付けられている ioMemory デバイスの有効なバス番号でなければなりません。

ここで、Fusion-io の SNMP サブエージェントは、(`TestModeIndex` で指定した ioMemory デバイスに対する) 既存のすべての ioMemory VSL データを、パラメータのリストにある設定済みのフィールドで置き換えます。フィールドが設定されていない場合、Agentx は既存のデータを保持し、そのデータを MIB にレポートします。フィールド内に値が存在する場合、Agentx はそのデータを置き換えて MIB にレポートします。

サブエージェントの動作は、`.INI` ファイル パラメータが OFF に設定されるまで、テスト モードで続行されます。以下に、`testmode.ini` ファイルに記述されているテスト モードの情報を示します。

```
# SNMP Test Mode sample file.  
# These values may be used to test the SNMP subsystem when it is in test mode.
```

```
[SNMP Agent Test Mode]  
TestMode           = off  
TestModeIndex      = 0
```

```
# InfoState:Note that the following states may change, but current  
definitions are:  
# 0 = unknown  
# 1 = detached  
# 2 = attached  
# 3 = minimal mode  
# 4 = error  
# 5 = detaching
```

```

# 6 = attaching
# 7 = scanning
# 8 = formatting
# 9 = updating firmware
# 10 = attach
# 11 = detach
# 12 = format
# 13 = update
InfoState = 2

InfoInternalTemp = 45
InfoAmbientTemp = 35
InfoWearoutIndicator = 2 ; 2=normal, 1=device is wearing out.
InfoWritableIndicator = 2 ; 2=normal, 1=non-writable, 0=write-
reduced, 3=unknown
InfoFlashbackIndicator = 2 ; 2=normal, 1=flashback protection
degraded.

ExtnTotalPhysCapacityU = 23
ExtnTotalPhysCapacityL = 215752192
ExtnUsablePhysCapacityU = 21
ExtnUsablePhysCapacityL = 7852192
ExtnUsedPhysCapacityU = 4
ExtnUsedPhysCapacityL = 782330816
ExtnTotalLogCapacityU = 18
ExtnTotalLogCapacityL = 2690588672
ExtnAvailLogCapacityU = 14
ExtnAvailLogCapacityL = 3870457856

ExtnBytesReadU = 18
ExtnBytesReadL = 3690588672
ExtnBytesWrittenU = 4
ExtnBytesWrittenL = 2578550816

InfoHealthPercentage = 95

InfoMinimalModeReason = 7 ; 0=unknown, 1=fw out of date, 2=low
power,
; 3=dual plane failure, 5=internal, 6=card limit,
; 7=not in minimal mode, 8=unsupported OS,
; 9=low memory

InfoReducedWriteReason = 0 ; 0=none, 1=user requested, 2=no md
blocks,
; 3=no memory, 4=failed die, 5=wearout,
; 6=adapter power, 7=internal, 8=power limit

InfoMilliVolts = 12000
InfoMilliVoltsPeak = 12100
InfoMilliVoltsMin = 11900
InfoMilliWatts = 6000
InfoMilliWattsPeak = 15000
InfoMilliAmps = 500
InfoMilliAmpsPeak = 1000

InfoAdapterExtPowerPresent = 1 ; 1=present, 2=absent

```

InfoPowerlossProtectDisabled = 2 ; 1=powerloss protection available but disabled
; 2=any other powerloss protection condition

SNMP MIB のサポート

Linux では、以下の SNMP MIB フィールドがサポートされています。

fusionIoDimmMibRevMajor	fusionIoDimmInfoAdapterType
fusionIoDimmMibRevMinor	fusionIoDimmInfoAdapterPort
fusionIoDimmMIBCondition	fusionIoDimmInfoAdapterSerialNumber
fusionIoDimmInfoIndex	fusionIoDimmInfoAdapterExtPowerPresent
fusionIoDimmInfoStatus	fusionIoDimmInfoPowerlossProtectDisabled
fusionIoDimmInfoName	fusionIoDimmInfoInternalTempHigh
fusionIoDimmInfoSerialNumber	fusionIoDimmInfoAmbientTemp
fusionIoDimmInfoPartNumber	fusionIoDimmInfoPCIBandwidthCompatibility
fusionIoDimmInfoSubVendorPartNumber	fusionIoDimmInfoPCIPowerCompatibility
fusionIoDimmInfoSparePartNumber	fusionIoDimmInfoActualGoverningLevel
fusionIoDimmInfoAssemblyNumber	fusionIoDimmInfoLifespanGoverningLevel
fusionIoDimmInfoFirmwareVersion	fusionIoDimmInfoPowerGoverningLevel
fusionIoDimmInfoDriverVersion	fusionIoDimmInfoThermalGoverningLevel
fusionIoDimmInfoUID	fusionIoDimmInfoLifespanGoverningEnabled
fusionIoDimmInfoState	fusionIoDimmInfoLifespanGoverningTgtDate
fusionIoDimmInfoClientDeviceName	fusionIoDimmExtnIndex
fusionIoDimmInfoBeacon	fusionIoDimmExtnTotalPhysCapacityU
fusionIoDimmInfoPCIAddress	fusionIoDimmExtnTotalPhysCapacityL
fusionIoDimmInfoPCIDeviceID	fusionIoDimmExtnTotalLogCapacityU
fusionIoDimmInfoPCISubdeviceID	fusionIoDimmExtnTotalLogCapacityL
fusionIoDimmInfoPCIVendorID	fusionIoDimmExtnBytesReadU
fusionIoDimmInfoPCISubvendorID	fusionIoDimmExtnBytesReadL
fusionIoDimmInfoPCISlot	fusionIoDimmExtnBytesWrittenU
fusionIoDimmInfoWearoutIndicator	fusionIoDimmExtnBytesWrittenL
fusionIoDimmInfoFlashbackIndicator	fusionIoDimmExtnFormattedBlockSize
fusionIoDimmInfoWritableIndicator	fusionIoDimmExtnCurrentRAMUsageU
fusionIoDimmInfoInternalTemp	fusionIoDimmExtnCurrentRAMUsageL
fusionIoDimmInfoHealthPercentage	fusionIoDimmExtnPeakRAMUsageU
fusionIoDimmInfoMinimalModeReason	fusionIoDimmExtnPeakRAMUsageL
fusionIoDimmInfoReducedWriteReason	fusionIoDimmWearoutTrap
fusionIoDimmInfoMilliVolts	fusionIoDimmNonWritableTrap
fusionIoDimmInfoMilliVoltsPeak	fusionIoDimmFlashbackTrap
fusionIoDimmInfoMilliVoltsMin	fusionIoDimmTempHighTrap
fusionIoDimmInfoMilliWatts	fusionIoDimmTempOkTrap

fusionIoDimmInfoMilliWattsPeak	fusionIoDimmErrorTrap
fusionIoDimmInfoMilliAmps	fusionIoDimmPowerlossProtectTrap
fusionIoDimmInfoMilliAmpsPeak	

付録 E – SMI-S インターフェイス

⚠ SMI-S プロバイダーには、`root/fio` という新しい CIM 名前空間があります。

SMI-S インターフェイスの概要

SMI-S インターフェイスは、既存の Distributed Management Task Force (DMTF) および Storage Networking Industry Association (SNIA) の Storage Management Initiative Specification (SMI-S) 標準に準拠しています。また、Web-Based Enterprise Management (WBEM) に基づいており、ioMemory デバイスおよび関連するソフトウェアを表す Common Information Model (CIM) モデルを提供します。このモデルでは、下位互換のある拡張が許可されており、Fusion-io によって開発される新規ハードウェアおよびソフトウェア機能に対応できます。

ここでは、WBEM、SMI-S、および DMTF 標準に精通しているユーザーを対象として説明します。このドキュメントおよび関連するモデルは、フィードバックを受けて任意の時点で変更される可能性があります。

リファレンス

CIM スキーマ v2.22

http://www.dmtf.org/standards/cim/cim_schema_v2220

DMTF DSP1011、物理アセット プロファイル

http://www.dmtf.org/standards/published_documents/DSP1011_1.0.2.pdf

DMTF DSP1023、ソフトウェア インベントリ プロファイル

http://www.dmtf.org/standards/published_documents/DSP1023_1.0.1.pdf

DMTF DSP1033、プロファイル登録プロファイル

http://www.dmtf.org/standards/published_documents/DSP1033_1.0.0.pdf

DMTF DSP1075 PCI デバイス プロファイル

http://www.dmtf.org/standards/published_documents/DSP1075_1.0.0.pdf

DMTF DSP1002、診断プロファイル

http://www.dmtf.org/standards/published_documents/DSP1002_2.0.0.pdf

SMI-S v1.4 アーキテクチャ

http://www.snia.org/sites/default/files/SMI-Sv1.4r6_Architecture.book_.pdf

SMI-S v1.4 共通プロファイル

http://www.snia.org/sites/default/files/SMI-Sv1.4r6_CommonProfiles.book_.pdf

SMI-S v1.4 ホスト プロファイル

http://www.snia.org/sites/default/files/SMI-Sv1.4r6_Host.book_.pdf

SMI-S v1.4 共通診断モデル

<http://www.dmtf.org/standards/mgmt/cdm/>

Linux への SMI-S WMI プロバイダーのインストール

Fusion-io の SMI-S プロバイダーは、ioMemory デバイスのリモート管理のために、DMTF および SNIA 規格に基づいた標準的な WBEM インターフェイスを実装しています。このプロバイダーは CMPI ベースのプロバイダーであり、SFCB、OpenPegasus、および OpenWBEM など、一般的な CIMOM と連携して動作する必要があります。当社では、Windows の WMI をサポートしているバージョンも用意しています。

このプロバイダーを使用するためには、CIMOM をインストールして構成する必要があります。その後、プロバイダーと関連 MOF ファイルをインストールし、CIMOM に登録する必要があります。MOF ファイルは、SMI-S プロバイダーから使用できる CIM オブジェクトを定義します。


元々、このプロバイダーは Linux の SFCB と Windows の WMI でテストされています。

1. SFCB CIM ブローカー (CIMOM) をインストールします。

- Debian 系:

```
$ sudo apt-get install sfcb sfcb-test wbemcli sblim-cmpi-base  
rsync
```

- その他:sblim-sfcc、sblim-sfcc-devel、cim-schema-2.21.0、sblim-sfcb、sblim-indication-helper、sblim-cmpi-base、および sblim-testsuite をインストールします。

 SLES、RHEL をはじめとするディストリビューション用の RPM は、OpenSUSE ビルドサービスで入手できます。

2. SFCB を構成します。そのためには、ファイル (`sfcb.cfg`) を `/etc/sfcb` にコピーします。
3. SMI-S プロバイダーをインストールします。そのためには、Fusion-io が配布する `fio-smis` パッケージをインストールし、`/usr/lib/fio/libfiosmis.so` を `/usr/lib/sfcb` (または `/usr/lib64/sfcb`) にコピーします。
4. 次のようにして SMI-S プロバイダーを SFCB に登録します。
5. `cd /usr/share/fio/cim`

6.

```
sh /usr/share/sblim-cmpi-base/provider-register.sh -r
fiosmis.registration cimv222-dmtf.mof
```

7. 次のようにして SFCB を再起動します。

```
$ /etc/init.d/sfcb restart
```

Linux のテスト

wbemcli ユーティリティを使用すると、SMI-S プロバイダーをテストできます。

1. ioMemory VSL のバージョンとシステム内の各 ioMemory デバイスのファームウェア バージョンをプロバイダーに問い合わせます。

```
$ wbemcli -nl ei http://localhost:5988/root/cimv2:FIO_SoftwareIdentity
```

出力は次のようになります(値は開発の進行に伴って変化する可能性があります)。

```
localhost:5988/root/cimv2:FIO_SoftwareIdentity.InstanceID="Fusion-io
drive driver"
-InstanceID="Fusion-io drive driver"
-TargetTypes=
-OtherExtendedResourceTypeDescription=
-MinExtendedResourceTypeRevisionNumber=
. . .
```

2. SMI-S プロバイダーに各 ioMemory デバイスの正常性を問い合わせます。

```
wbemcli -nl ei http://localhost:5988/root/cimv2:FIO_IoMemoryPort
```

出力は次のようになります(値は開発の進行に伴って変化する可能性があります)。

```
localhost:5988/root/cimv2:FIO_IoMemoryPort.DeviceID="fct1",CreationClas
sName="FIO_IoMemoryPort",SystemName="bigpu01",SystemCreationClassName="
FIO_ComputerSystem"
-DeviceID="fct1"
-CreationClassName="FIO_IoMemoryPort"
-SystemName="bigpu01"
. . .
```

3. 指定の ioMemory デバイス(以下の例では fct0)の容量と使用カウンターを問い合わせます。

```
$ wbemcli -nl gi
http://localhost:5988/root/cimv2:FIO_SSDStatistics.InstanceID="fct0"
```

出力は次のようになります(値は開発の進行に伴って変化する可能性があります)。

```
localhost:5988/root/cimv2:FIO_SSDStatistics.InstanceID="fct0"
```

```
-InstanceID="fct0"  
-WriteOperations=0  
-ReadOperations=6887  
-PhysicalMBytesWritten=1523769  
. . .
```

Linux の SMI-S プロバイダーは、上記の例の `localhost` をリモート ホストのホスト名または IP アドレスで置き換えることで、`wbemcli` ユーティリティを使用してリモートでテストできます。ただし、この方法は Windows の SMI-S プロバイダーをリモートでテストするためには使用できません。Windows は(当然ながら)新興の規格には追従していないからです。

SMI-S プロバイダーの通知も同様にテストできます。

解説

SMI-S は、もともとファイバー チャネル、iSCSI、SAS などの SCSI コマンド セットに基づくストレージエリア ネットワーク(SAN)システムに重点を置いた仕様のコレクションです。しかし、これらのストレージシステムのモデル化に使用される一般的なパターンは、Fusion-io が提供するシステムのような直接接続型のソリッドステート ストレージ システムにも適用できます。

ioMemory デバイスは、ストレージ HBA (Storage HBA)、直接接続ポート (Direct Attached (DA) Ports)、ホスト検出リソース (Host Discovered Resources) プロファイルで確立された SMI-S パターンを使用してモデル化されています。物理的資産プロファイル、ソフトウェア インベントリ プロファイル、PCI デバイス プロファイル、共同診断モデル プロファイルを含む公開済みの DMTF 仕様を使用して、ioMemory デバイスの物理的な部分、およびすべてのファームウェアと ioMemory VSL ソフトウェアがモデル化されています。

次の図に、ioMemory デバイスおよび関連するファームウェアとソフトウェアについての Fusion-io SMI-S CIM モデルを示します。見やすくするために、クラス名のプレフィクス「FIO_」は省略しています。

ProtocolEndpoint クラス(G)のインスタンスは、IOMemoryPort とソリッドステートストレージ間の論理データパスの両端を表します。モデルのこうした側面は、ポートがイニシエーターでもありターゲットでもある DA ポート プロファイルのパターンに由来しています。ProtocolEndpoint は、DeviceSAPImplementation(7)によって IOMemoryPort に、HostedAccessPoint(8)によって ComputerSystem に、それぞれ関連付けられています。

H:LogicalSSD クラス(ブロック デバイス)

アプリケーション(ファイル システム、データベース、論理ボリューム マネージャー)に公開されるブロック デバイスは、CIM_DiskDrive のサブクラスである LogicalSSD(H)のインスタンスを使用してモデル化されています。このブロック デバイスは、MediaPresent 関連(9)を使用して StorageExtent(J)と関連付けられていますが、StorageExtent は常に存在します。また、IOMemoryPort を表す ProtocolEndpoint(G)には SAPAvailableForElement(10)を使用して、スコープを設定する ComputerSystem には SystemDevice(3)を使用して、それぞれ関連付けられています。

ioMemory デバイスは、PCIe デバイスなので、PCIDevice クラス(B)のインスタンスによっても表されません。IOMemoryPort は、PCIDevice およびその関連制御デバイスの代替表現です。また、IOMemoryPort は ConcreteIdentity 関連によって PCIDevice に関連付けられています。

K:SoftwareIdentity

ioMemory VSL ソフトウェアは、SoftwareIdentity によっても表されています。SoftwareIdentity は、ElementSoftwareIdentity 関連(11)によって PCIDevice に関連付けられています。SoftwareIdentity(ファームウェア)は、スコープを設定する ComputerSystem に InstalledSoftwareIdentity 関連(12)によって関連付けられています。SoftwareInstallationService(L)のインスタンスは各 PCIDevice に関連付けられており、各 PCIDevice はデバイス ファームウェアの更新に使用できます。

M:物理的側面

ioMemory デバイスの物理的側面は、PhysicalPackage クラス(M)のインスタンスによって表され、このクラスは Realizes(13)によって PCIDevice に、スコープを設定する ComputerSystem には SystemPackaging(14)によって、それぞれ関連付けられています。ioMemory デバイス上の温度センサーは、TemperatureSensor(N)のインスタンスによって表され、AssociatedSensor によって PhysicalPackage に関連付けられています。

実装

このセクションでは、Fusion-io デバイスの CIM モデル用のインスタンスと関連の配置について説明します。ただし、すべてのクラス プロパティについて詳しく説明しているわけではありません。すべてのプロパティの詳細については、CIM スキーマを参照してください。

このモデルに基づく WBEM CIM プロバイダーは、将来開発される予定です。Fusion-io では、OpenPegasus、OpenWBEM、SFCB、Windows WMI など、一般的な CIMOM をサポートする予定です。

デバイスの正常性は、`HealthLevel` プロパティの値によって示されます。このプロパティは、正常 (Healthy)、警告 (Warning)、書き込み低減 (Reduced Write)、読み取り専用 (Read Only) の値を取ります。これらの値は、適宜 `standardHealthState` の値 - OK、低下/警告 (Degraded/Warning)、および致命的障害 (Critical Failure) - にマッピングされます。

デバイスのプロビジョニング用の外部メソッドとしては、接続 (attach)、切断 (detach)、フォーマット (format)、および更新 (update) があります。接続メソッドは、`ioMemory` デバイス用のブロック デバイスを作成します。切断は、このブロック デバイスを無効にします。フォーマット オプションは、ユーザーによるメガバイト単位またはパーセンテージのどちらかでのデバイス サイズの指定を可能にします。更新メソッドは、ユーザーによるデバイス上のファームウェアの更新を可能にします。

デバイスの耐用期間は、`HealthPercentage` プロパティの値によって示されます。

`FlashbackAvailability` は、`ioMemory` デバイスのこの機能がオンライン状態であるかどうかを示します。

各 `IOMemoryPort` は、`ControlledBy` 集約を介して `IOMemoryPortController` によって集約されます。`IOMemoryPort` のインスタンスは、`ConcreteIdentity` 関連によって対応する `PCIDevice` に関連付けられています。`IOMemoryPort` は、スコープを設定する `ComputerSystem` の論理デバイスであり、`SystemDevice` 集約によってそのように示されています。

`ioDrive Duo` デバイスなど、2 つ以上の `ioMemory` デバイスを備えた製品は、2 つの個別の `ioMemory` デバイスのように見えます。複数のデバイスを持つ製品の場合は、キャリア カードの種類、シリアル番号、製品全体の外部電源の接続に関する情報を含めるために `IOMemoryPort` クラスが拡張されます。

IOMemoryPort

`IOMemoryPort` のインスタンスは、`ComputerSystem` に取り付けられている `ioMemory` デバイスごとに 1 つ存在します。

`LocationIndicator` プロパティは、デバイス インジケータ ビーコン (例: ソリッド上のすべての LED) の状態を反映します。この値を読み取ることで、インジケータの現在の状態が与えられます。この値に "On" または "Off" を書き込むと、インジケータのオン/オフが切り替わります。この操作は、デバイスの物理的な場所を確認するために利用できます

SSDStatistics

SSDStatistics のインスタンスは、IOMemoryPort インスタンスごとに 1 つ存在します。このオブジェクトのプロパティは、性能と容量に関する情報を提供します。こうした情報の一部は、ドライブの接続時（関連する IOMemoryPort の状態が“Attached”（接続済み）になっているとき）に限り使用できます。

IOMemoryPortController

IOMemoryPortController のインスタンスは 1 つだけ存在し、IOMemoryPorts の制御に使用される ioMemory VSL ソフトウェアを表します。IOMemoryPortController は、CIM_PortController を特化したものです。

IOMemoryPortController は、スコープを設定する ComputerSystem に、SystemDevice 集約を使用して集約されます。IOMemoryPortController は、ioMemory VSL を表すインスタンスと、ElementSoftwareIdentity 関連を介して関連付けられています。

ProtocolEndpoint

ProtocolEndpoint のインスタンスは、IOMemoryPort のインスタンスごとに 1 つ存在します。このインスタンスは、DeviceSAPImplementation を使用して IOMemoryPort に、SAPAvailableForElement を使用して LogicalSSD に、それぞれ関連付けられています。IOMemoryPort はイニシエーター ポートとターゲット ポートの両方を表すので、IOMemoryPort と LogicalSSD 間の接続をモデル化するために必要な ProtocolEndpoint は、IOMemoryPort ごとに 1 つだけです。

LogicalSSD

CIM_DiskDrive のサブクラスである LogicalSSD のインスタンスは、ioMemory デバイスによって公開されるブロック デバイス(/dev/fioX)ごとに 1 つ存在します。オペレーティング システムのデバイス名に基づき、関連付け可能な ID が使用されます。これにより、クライアント アプリケーションは、このモデルによって検出されたブロック デバイスを、ホスト システム上に備わっている他の SMI-S モデルから検出されたリソースと関連付けることができます。

ComputerSystem は、SystemDevice を介して各 LogicalSSD を集約します。LogicalSSD インスタンスは、SAPAvailableForElement を介してそれぞれの ProtocolEndpoints に関連付けられています。エンドポイントに関連付けられている IOMemoryPort が接続されていない場合は、Availability プロパティが“Off Line”（オフライン）に設定され、DeviceID プロパティの値は“Unknown”（不明）になります。

StorageExtent

それぞれの LogicalSSD には StorageExtent のインスタンスが 1 つ関連付けられており、このインスタンスは関連するデバイスの論理ストレージを表します。

SoftwareIdentity

SoftwareIdentity のインスタンスは、ioMemory VSL ソフトウェアを表すために 1 つ存在します。ファームウェアもまた SoftwareIdentity を使用してモデル化されていますが、ファームウェアでは取り付けられている ioDrive ごとにインスタンスが必要です。IsEntity プロパティの値 True は、SoftwareIdentity インスタンスが ioMemory VSL ソフトウェアまたはファームウェアの分離されたコピーに対応することを示します。MajorVersion、MinorVersion、RevisionNumber、および BuildNumber の各プロパティは、ドライバーやファームウェアのバージョン情報を伝えます。Manufacturer プロパティは、Fusion-io の識別に使用できます。

ファームウェアに関する別のオプションは、ComputerSystem との InstalledSoftwareIdentity 関連を省略することです。というのも、ファームウェアは実際には ComputerSystem にインストールされていないからです。このオプションは、ユーザーがファームウェアをどのようにモデル化するかに依存しています。

SoftwareInstallationService

SoftwareInstallationService のインスタンスは PCIDevice ごとに存在し、関連付けられているデバイスのファームウェアを更新するために使用できます。

PCIDevice

PCIDevice のインスタンスは、コンピューター内の ioMemory デバイス (PCIe カード) ごとにインスタンス化されています。各プロパティは次のように設定されます。

- BusNumber - PCIe デバイスが存在するバスの番号です。DeviceNumber - このバスの PCI デバイスに割り当てられているデバイス番号です。
- FunctionNumber - PCI デバイスの関数番号に設定されます。
- SubsystemID、SubsystemVendorID、PCIDeviceID、VendorID、および RevisionID は省略できますが、PCI デバイスの構成レジスターから値を抽出できる場合には設定されることがあります。

PCIDevice は、ConcreteIdentity を使用して、その代替論理表現である IOMemoryPort と関連付けられています。また、PCIDevice は Realizes により、ioDrive の物理的側面を表す PhysicalPackage とも関連付けられています。

PhysicalPackage

PhysicalPackage のインスタンスは、コンピューター システム内に取り付けられている物理的な個々の ioDrive ごとに 1 つ存在します。Manufacturer、Model、SKU、SerialNumber、Version、および PartNumber の各プロパティは、物理カードの各側面を記述するために使用できます。

PhysicalPackage は、Realizes によって PCIDevice と、スコープを設定する ComputerSystem とは SystemPackaging によって、それぞれ関連付けられています。

TemperatureSensor

TemperatureSensor のインスタンスは、PhysicalPackage ごとに 1 つ存在します。ドライブの温度に関する情報は、このオブジェクトのプロパティに格納されます。

DiagnosticTest

DiagnosticTest のインスタンスは 1 つ存在することになります。RunDiagnostic() メソッドは、指定された ManagedElement (IoMemoryPort のインスタンスでなければなりません) のデバイス状態のスナップショットを取得するトリガーとなります。診断の実行は同期的であり、即座に行われます。結果として得られる ConcreteJob オブジェクトは、呼び出し側の DiagnosticTest インスタンスと、指定されたそれぞれの IoMemoryPort インスタンスに関連付けられます。この時点では、用意されているデフォルトの DiagnosticSettingData でのみ RunDiagnostic() を使用できます。

実行のたびに、DiagnosticSettingDataRecord の 1 つのエントリと、関連する

DiagnosticCompletionRecord が DiagnosticLog に追加されます。

DiagnosticCompletionRecord の RecordData プロパティは、実行時の深刻なデバイス状態を記録します。RecordData 文字列の書式は、RecordFormat プロパティ内に見つかります。

この書式は状態を表す文字列の系列であり、各文字列はアスタリスク(*) 文字で区切られた "Unknown" (不明)、"OK"、"Warning" (警告)、または "Error" (エラー) の各値の 1 つを保持できます。現時点では、WearoutStatus、WritabilityStatus、FlashbackStatus、TemperatureStatus、MinimalModeStatus、PciStatus、および InternalErrorStatus の 7 つの状態値が記録されています。通常の運用状況では、これらがすべて "OK" になるはずですが、

WearoutStatus は、デバイス上で予約されている残り領域が 10% 未満になると、"Warning" (警告) に設定されます。また、予約された領域がそれ以上なくなると、"Error" (エラー) に設定されます。

- WritabilityStatus は、デバイスがスロットリングによる書き込み調整中または読み取り専用モード (read-only mode) になると、"Error" (エラー) に設定されます。この状況は、デバイスの消耗や電力の不足など、さまざまな条件で発生する可能性があります。
- FlashbackStatus は、致命的なエラーによって Flashback 保護のレベルが低下すると、"Warning" (警告) になります。

- `TemperatureStatus` は、デバイスの温度が安全な最高温度付近まで上がると“Warning”（警告）になり、最大安全温度に到達またはその温度を超えると“Error”（エラー）になります。
- `MinimalModeStatus` は、デバイスが最小モード(minimal mode)に入ると、“Warning”（警告）または“Error”（エラー）のどちらかになります。
- `PciStatus` は、ホスト PCIe バスに互換性の問題が存在する場合に“Warning”（警告）または“Error”（エラー）になります。
- `InternalErrorStatus` は、ioMemory VSL に内部的な問題が存在する場合に“Error”（エラー）になります。

`CompletionState` プロパティは、これらの結果を要約し、不明(Unknown)、OK、警告(Warning)、または失敗(Failed)に設定されます。いずれかの状態がエラーになっている場合、このプロパティは失敗(Failed)になります。それ以外の場合は、何らかの警告状態になっていれば、警告(Warning)になります。`Message` プロパティは、何らかの警告またはエラーが存在する場合に、適切な措置を示すために設定されます。

DiagnosticSettingData

`DiagnosticTest` インスタンスに関連付けられた `DiagnosticSettingData` のインスタンスが存在することになります。これは、`RunDiagnostic` に対するそれぞれの呼び出しのデフォルト設定を記録します。

DiagnosticServiceCapabilities

`DiagnosticTest` サービスの機能を記録する `DiagnosticTest` インスタンスに関連付けられた `DiagnosticServiceCapabilities` のインスタンスが存在します。

DiagnosticLog

`DiagnosticLog` のインスタンスは、`DiagnosticTest` インスタンスと関連付けられており、それぞれの実行の結果を格納します。

DiagnosticSettingRecord

デフォルトの `DiagnosticSettingData` のコピーは、診断が実行されるたびに `DiagnosticSettingDataRecord` に格納され、`DiagnosticCompletionRecord` のインスタンスと関連付けられます。

DiagnosticCompletionRecord

`DiagnosticCompletionRecord` のインスタンスは、それぞれの `RunDiagnostic` 実行の結果を格納します。詳細については、`DiagnosticTest` の項を参照してください。

RegisteredDiskDriveLiteProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは `/root/interop` 名前空間にあり、ディスクドライブ ライト プロファイル(Disk Drive Lite Profile)の実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "SNIA:DiskDriveLiteProfile-1.4.0"に設定
- RegisteredOrganization - "11" (SNIA)に設定
- RegisteredName - "DirectAccess Ports Profile"に設定
- RegisteredVersion - "1.4.0"に設定

RegisteredDAPortsProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは `/root/interop` 名前空間にあり、DA ポート プロファイルの実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "SNIA:DAPortsProfile-1.4.0"に設定
- RegisteredOrganization - "11" (SNIA)に設定
- RegisteredName - "DirectAccess Ports Profile"に設定
- RegisteredVersion - "1.4.0"に設定

RegisteredStorageHBAProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは `/root/interop` 名前空間にあり、ストレージ HBA プロファイル(Storage HBA Profile)の実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "SNIA:StorageHBAProfile-1.4.0"に設定
- RegisteredOrganization - "11" (SNIA)に設定
- RegisteredName - "Storage HBA Profile"に設定
- RegisteredVersion - "1.4.0"に設定

RegisteredHostDiscoveredResourcesProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは `/root/interop` 名前空間にあり、ホスト検出リソース プロファイル(Host Discovered Resources Profile)の実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "SNIA:HostDiscoveredResourcesProfile-1.2.0"に設定
- RegisteredOrganization - "11" (SNIA)に設定
- RegisteredName - "Host Discovered Resources Profile"に設定
- RegisteredVersion - "1.2.0"に設定

RegisteredPCIDeviceProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは `/root/interop` 名前空間にあり、PCI デバイス プロファイル (PCI Device Profile) の実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "DMTF:DSP1075-PCIDevice-1.0.0a" に設定
- RegisteredOrganization - "2" (DMTF) に設定
- RegisteredName - "PCIDevice Profile" に設定
- RegisteredVersion - "1.0.0a" に設定

RegisteredSoftwareInventoryProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは `/root/interop` 名前空間にあり、ソフトウェア インベントリ プロファイル (Software Inventory Profile) の実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "DMTF:DSP1023-SoftwareInventory-1.0.1" に設定
- RegisteredOrganization - "2" (DMTF) に設定
- RegisteredName - "Software Inventory Profile" に設定
- RegisteredVersion - "1.0.1" に設定

RegisteredSoftwareUpdateProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは `/root/interop` 名前空間にあり、ソフトウェア更新プロファイル (Software Update Profile) の実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "DMTF:DSP1023-SoftwareUpdate-1.0.0" に設定
- RegisteredOrganization - "2" (DMTF) に設定
- RegisteredName - "Software Update Profile" に設定
- RegisteredVersion - "1.0.0" に設定

RegisteredPhysicalAssetProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは `/root/interop` 名前空間にあり、物理アセット プロファイルの実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "DMTF:PhysicalAssetProfile-1.0.2" に設定
- RegisteredOrganization - "2" (DMTF) に設定
- RegisteredName - "PhysicalAsset Profile" に設定
- RegisteredVersion - "1.0.2" に設定

RegisteredSensorsProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは /root/interop 名前空間にあり、センサープロファイル(Sensors Profile)の実装を示しています。以下に各プロパティの設定を示します。

- InstanceID - "SNIA:SensorsProfile-1.0.0"に設定
- RegisteredOrganization - "11" (SNIA)に設定
- RegisteredName - "Sensors Profile"に設定
- RegisteredVersion - "1.0.0"に設定

RegisteredCommonDiagnosticProfile

このクラスのインスタンスは 1 つだけ必要です。このクラスは /root/interop 名前空間にあり、共通診断モデル プロファイルの実装を示しています。InstanceID プロパティは、"DMTF:DiagnosicsProfile-2.0.0a"という値に設定されます。RegisteredOrganization プロパティは、値"2" (DMTF)に設定されます。RegisteredName プロパティは、"Diagnostics Profile"という値に設定されます。RegisteredVersion プロパティは、"2.0.0a"という値に設定されます。

通知

特定の ioMemory デバイスに重大な状態が発生している場合は、定期的に通知が生成されます。現在、WBEM プロバイダーでは、6 種類の通知がサポートされています。これらによって、SMI-S プロバイダーのユーザーに対して、間もなく発生する消耗、書き込み性能の低下、Flashback 機能の低下、温度の上昇、内部エラー状態などの状態について通知されます。

通知は、CIM_AlertIndication クラスを特化した FIO_AlertIndication クラスのインスタンスです。

FIO_AlertIndication のインスタンスのプロパティの値は現在策定中であり、テストの進行およびフィードバックの受領にともなって変更される可能性があります。

FIO_AlertIndication

プロパティ	値
IndicationIdentifier	それぞれのタイプについて後の説明を参照
IndicationTime	送信された時点のタイムスタンプ
AlertingManagedElement	IoMemoryPort.DeviceID=<デバイス ID>
AlertingElementFormat	CIMObjectPath(2)
AlertType	デバイス アラート(5)
PerceivedSeverity	それぞれのタイプについて後の説明を参照
ProbableCause	それぞれのタイプについて後の説明を参照

SystemCreationClassName	「FIO_AlertIndication」
SystemName	<ホスト名>
ProviderName	「fiosmis」
CorrelatedIndications	未使用
説明	クラスの説明
OtherAlertType	未使用
OtherSeverity	未使用
ProbableCauseDescription	未使用
EventID	IndicationIdentifierと同じ
OwningEntity	<ベンダー>
MessageID	未定
メッセージ	未定
MessageArguments	未定

書き込み性能低減通知

ioMemory VSL では、過度の消耗、温度の上昇、電力不足などのデバイスの状態を管理するために、動的に書き込みスループットが制限される場合があります。書き込み性能低減通知は、デバイスが書き込み低減モード(write-reduced mode)である間に生成されます。このモードの原因が過度の消耗である場合、IoMemoryPort の正常性パーセンテージでは正常性が 0%と報告されます。

プロパティ	値
IndicationIdentifier	<mfr>": "<ホスト名>":write"
PerceivedSeverity	低下/警告(3)
ProbableCause	しきい値超過(52) 温度の許容範囲超過(51) 電力の問題(36)

読み取り専用通知

ドライブが寿命に達すると、書き込むことができなくなり、読み取りのみを行うことができるようになります。このような場合に読み取り専用通知が送信されます。この場合、IoMemoryPort の正常性パーセンテージでは、引き続き正常性が 0%と報告されます。

プロパティ	値
IndicationIdentifier	<mfr>": "<ホスト名>":read_only"
PerceivedSeverity	低下/警告(3)

ProbableCause	しきい値超過(52)
---------------	------------

消耗通知

ドライブが消耗した場合、ドライブの正常性パーセンテージが 10%未満に低下した後、書き込みスループットが制限される前に、この通知が警告として生成されます。

プロパティ	値
IndicationIdentifier	<mfr>":"<ホスト名>":wearout"
PerceivedSeverity	低下/警告(3)
ProbableCause	しきい値超過(52)

Flashback 通知

部品の回復不可能な障害によって Flashback 機能の効率性が低下した場合は、この通知が送信されます。

プロパティ	値
IndicationIdentifier	<mfr>":"<ホスト名>":flashback"
PerceivedSeverity	低下/警告(3)
ProbableCause	冗長性の消失(88)

温度上昇通知

カードの温度が過度に上昇した場合にこの通知が送信されます。

プロパティ	値
IndicationIdentifier	<mfr>":"<ホスト名>":temperature"
PerceivedSeverity	重大(6)
ProbableCause	温度の許容範囲超過(51)

エラー通知

ioMemory VSL がエラー状態になった場合は、エラー通知が送信されます。


プロパティ	値
IndicationIdentifier	<mfr>":"<ホスト名>":error"
PerceivedSeverity	重大(6)
ProbableCause	その他(1)

付録 F – NUMA 構成

NUMA アーキテクチャについて

NUMA (Non-Uniform Memory Access) アーキテクチャを備えたサーバでは、ioMemory デバイスの性能を最大限発揮するために特別なインストール手順が必要です。このようなサーバには、HP DL580、HP DL980、IBM 3850 サーバなどがあります。

NUMA アーキテクチャを備えたサーバでは、システムの起動時に BIOS によって NUMA ノード間で PCIe スロットが均等に配分されない場合があります。各 NUMA ノードには、複数の CPU が含まれています。スロットが均等に配分されないと、作業負荷が高まった場合に、半分以上の CPU がアイドル状態であるにもかかわらず、残りの CPU の使用率が 100% になる可能性があります。このような不均等を回避するために、利用可能な NUMA ノードに対して、ioMemory デバイスを手動で均等に割り当てる必要があります。

 以降の例では、カスタム アフィニティ設定の最終的な実装を示しています。実装では、システムのアーキテクチャ、取り付けられている ioMemory デバイスの種類と数、使用されている特定の PCIe スロットなどの、特定のシステムについての分析が必要になります。特定の環境に対して、そのセットアップに応じたカスタムの分析が必要です。このような分析を行う場合は、特定のインストールに合わせてシステムの NMUA アーキテクチャを理解する必要があります。

サーバの構成によって、実際の設定は次の例と異なる場合があります。実際のシステムで正しい設定を行うために、`fio-status` を使用して、すべてのデバイス (fct 番号) をリスト表示します。次に、`fio-beacon` を使用して、各 PCIe スロットのそれぞれのデバイスを識別します。その後、`numa_node_override` パラメータを設定する以下の例をテンプレートとして使用し、このパラメータを実際のシステムに合わせて変更します。

NUMA アーキテクチャを採用したサーバで ioMemory デバイスを構成するには、`numa_node_override` パラメータを使用する必要があります。そのためには、`iomemory-vs1.conf` ファイルを変更します。

numa_node_override パラメータ

`numa_node_override` パラメータは、システム内のすべてのデバイスのアフィニティ設定を指定する <アフィニティの指定> カプレット (couplets) のリストです。カプレット内の各項目はコロンで、各カプレットはコンマで区切ります。

構文:

```
numa_node_override=<アフィニティの指定>[, <アフィニティの指定>...]
```

ここで、各<アフィニティの指定>の構文は次のとおりです。

<fct 番号>:<ノード番号>

簡単な例:

```
numa_node_override=fct4:1,fct5:0,fct7:2,fct9:3
```

この例では、次のアフィニティが作成されます。

デバイス	ノード/グループ	プロセッサのアフィニティ
fct4	ノード 1	ノード 1 内のすべてのプロセッサ
fct5	ノード 0	ノード 0 内のすべてのプロセッサ
fct7	ノード 2	ノード 2 内のすべてのプロセッサ
fct9	ノード 3	ノード 3 内のすべてのプロセッサ

高度な構成の例

このサンプルのサーバは、4 つの NUMA ノードで構成されており、ノードあたり 8 つのハーパースレッドコアがあります(ノードあたり 16 の論理プロセッサ、システム全体で 64 の論理プロセッサがあります)。また、このシステムでは拡張構成が使用されており、11 の PCIe 拡張スロットがあります。システムの起動時に、システムの BIOS によって PCIe スロット 1~6 が NUMA ノード 2 に、PCIe スロット 7~11 が NUMA ノード 0 に割り当てられます。NUMA ノード 1 および 3 には、PCIe スロットは割り当てられません。このように割り当てた場合、ioMemory デバイスのトラフィックが非常に多くなったときに、システムで負荷分散に関する問題が発生します。具体的には、デバイスの使用率が高くなると、システム内の CPU の半分がアイドル状態であるにもかかわらず、残りの半分の CPU の使用率が 100%になり、ioMemory デバイスのスループットが制限されます。

この問題を回避するためには、`numa_node_override` パラメータを使用して ioMemory デバイスのアフィニティを手動で構成し、すべての NUMA ノードに作業負荷を分散する必要があります。このパラメータを指定すると、ioMemory VSL ドライバーのデフォルトの動作が上書きされます。

`numa_node_override` パラメータの詳細については、上記の構文説明を参照してください。

次に、10 の ioDrive Duo デバイスを手動で構成する例を示します(各 ioDrive Duo デバイスは、2 つの ioMemory デバイスで構成されています)。スロット 1 は Gen1 PCIe スロットなので、ioDrive Duo デバイスとは互換性がありません。したがって、スロット 2~11 に ioDrive Duo デバイスを取り付けます。

① 各 ioDrive Duo デバイスは 2 つの ioMemory デバイスで構成されているため、それぞれの ioDrive Duo デバイスに対して 2 つのデバイス番号があります(各 ioMemory デバイスに対して 1 つの番号)。したがって、各スロットに 2 つのデバイス番号があります。

システム起動時の BIOS によるデフォルトの NUMA ノード割り当ては次のとおりです。


BIOS によって割り当てられる NUMA ノード	PCIe スロット	FCT デバイス番号	プロセッサの アフィニティ
0	7-11	8,9,13,14,18,19,23,24,28,29	ノード内のすべてのプロセッサ
1	なし	なし	なし
2	2-6	135,136,140,141,145,146,150,151,155,156	ノード内のすべてのプロセッサ
3	なし	なし	なし

BIOS によるデフォルトの割り当てでは、カードがシステム内の 2 つの NUMA ノードにしか割り当てられないため、負荷が均等に分散されません。作業負荷を均等に分散するために、次の手動設定を行います。

割り当てられる NUMA ノード	PCIe スロット	FCT デバイス番号	プロセッサのアフィニティ
0	7-9	8,9,13,14,18,19	ノード内のすべてのプロセッサ
1	10-11	23,24,28,29	ノード内のすべてのプロセッサ
2	2-3	135,136,140,141	ノード内のすべてのプロセッサ
3	4-6	145,146,150,151,155,156	ノード内のすべてのプロセッサ

ioMemory VSL ドライバーにこれらの上書き設定を構成するには、次の文字列を使用して `numa_node_override` パラメータを設定します。

```
numa_node_override=fct8:0,fct9:0,fct13:0,fct14:0,fct18:0,fct19:0,
fct23:1,fct24:1,fct28:1,fct29:1,fct135:2,fct136:2,fct140:2,fct141:2,
fct145:3,fct146:3,fct150:3,fct151:3,fct155:3,fct156:3
```

 上記の例では、見やすくするために改行が挿入されています。`numa_node_override` パラメータの実際の実装では、改行は挿入しません。

付録 G – VSL 2.x から 3.x へのデバイスのアップグレード

このバージョンの ioMemory VSL では、最新の ioMemory アーキテクチャや Flashback Protection 機能の向上など、新しい機能がサポートされています。これらの機能を使用するには、最新バージョンの Fusion-io ファームウェアが必要です。3.1.x 以降を実行するシステム内のすべての ioMemory デバイスは、最新バージョンのファームウェアにアップグレードする必要があります。

たとえば、ioMemory VSL 2.3.1 を実行しており、以前に ioDrive デバイスが取り付けられているシステムに新しい ioDrive2 デバイス(ioDrive2 デバイスには最新バージョンのファームウェアが必要)を取り付ける場合、既存のすべてのデバイスを最新のファームウェア バージョンにアップグレードする必要があります。

- ❌ デバイスをアップグレードした後は、デバイスのファームウェアを以前のバージョンに戻すことはできません(以前のバージョンに戻すと、保証が無効となります)。
- ❌ (VSL 2.x.x 用に構成されていた)デバイスをアップグレードして VSL 3.x.x に対応させるには、デバイスのローレベル メディア フォーマットが必要です。フォーマットすると、ユーザー データは消去されます。指示に従って、必ずすべてのデータをバックアップしてください。

⚠️ アップグレード パス

デバイスの現在のファームウェア バージョンによっては、内部構造を維持するために、デバイスのファームウェアのアップグレードが複数回必要になることがあります。次に、最低限実行する必要があるアップグレード パスを示します。システム上の ioMemory VSL ソフトウェアは、この順序でアップグレードします(ソフトウェアの各バージョンと互換性のあるバージョンに**ファームウェアもアップグレードします**)。


1.2.4→1.2.7→2.1.0→2.3.1→3.x

たとえば、デバイスで ioMemory VSL バージョン 2.2.0 用のファームウェアを使用している場合は、(ioMemory VSL および互換性のあるファームウェアの両方を)2.3.1 にアップグレードしてから、パスに従ってアップグレードします。必要なすべてのソフトウェア バージョンおよびファームウェア バージョンの一覧については、[PRIMERGY ダウンロードページ](http://jp.fujitsu.com/platform/server/primergy/downloads/)：


<http://jp.fujitsu.com/platform/server/primergy/downloads/>で参照してください。

アップグレード手順

必ず上記のアップグレードパスに従ってください。以前に取り付けられているすべての ioDrive デバイスが、2.3.1 に互換性のある適切なファームウェアに更新されていることを確認します。

 同じホストで ioDrive デバイスと ioDrive2 デバイスを使用する場合は、新しい ioDrive2 デバイスを取り付ける前に既存のすべての ioDrive デバイスでこのアップグレードを実行します。

1. 既存の各 ioDrive デバイスでアップグレードの準備を行います。
 1. 各デバイス上のユーザー データをバックアップします。

 アップグレードを行うには、デバイスのローレベル メディア フォーマットが必要です。フォーマットすると、ユーザー データは消去されます。すべてのデータをバックアップしてください。

2. 任意のバックアップ方法を使用します。過去に実績があるソフトウェアおよびバックアップデバイスを使用することをお勧めします。同じシステム上の別の ioMemory デバイスにはデータをバックアップしないでください。ローカル ディスク、または外部接続ボリュームにバックアップする必要があります。
3. [fio-bugreport](#) ユーティリティを実行して、出力を保存します。これにより、システム内の各デバイスのデバイス情報がキャプチャされます。このデバイス情報は、アップグレードの問題をトラブルシューティングする場合に役立ちます。コマンド例:

```
fio-bugreport
```

4. ioDrive デバイスを切断します。次に例を示します。

```
fio-detach /dev/fct*
```

詳細については、「[fio-detach](#)」を参照してください。

2. 現在の ioMemory VSL ドライバーをアンロードします。次に例を示します。

```
$ modprobe -r iomemory-vsl
```

3. ioMemory VSL 2.x ソフトウェアをアンインストールします。
 1. このソフトウェアをアンインストールするには、アンインストールするパッケージのカーネルバージョンを指定する必要があります。適切なコマンドを実行して、インストールされているパッケージを調べます。
 - RPM コマンド:

```
$ rpm -qa | grep -i iomemory
```

- DEB コマンド:

```
$ dpkg -l | grep -i iomemory
```

次に、サンプルの出力を示します。

```
iomemory-vsl-2.6.18-194.el5-2.2.2.82-1.0
```

2. 次の例に類似したコマンドを実行して、ioMemory VSL をアンインストールします（アンインストールするパッケージのカーネル バージョンを指定します）。

- サンプル RPM コマンド:

```
$ rpm -e iomemory-vsl-2.6.18-194.el5-2.2.0.82-1.0
```

- サンプル DEB コマンド:

```
$ dpkg -r iomemory-vsl-2.6.32-24-server
```

3. ユーティリティをアンインストールします。

- サンプル RPM コマンド:


```
$ rpm -e fio-util fio-snmp-agentx fio-common fio-firmware  
iomanager-gui iomanager-jre libfio libfio-doc libfusionjni  
fio-sysvinit fio-smis fio-snmp-mib libfio-deb
```

- サンプル DEB コマンド:

```
$ dpkg -r fio-util fio-snmp-agentx fio-common fio-firmware  
iomanager-gui iomanager-jre libfio libfio-doc libfusionjni  
fio-sysvinit fio-smis fio-snmp-mib libfio-deb
```

4. 新しい VSL と関連パッケージをインストールします。

1. お使いのカーネル向けの ioMemory VSL バイナリ パッケージとすべてのサポート パッケージを [PRIMERGY ダウンロードページ](http://jp.fujitsu.com/platform/server/primergy/downloads/): <http://jp.fujitsu.com/platform/server/primergy/downloads/> からダウンロードします。

 お使いのカーネル向けのバイナリが見当たらない場合は、「[ソースからの ioMemory VSL のビルド](#)」の手順に従います。現在のカーネル バージョンを確認するには、次のコマンドを実行します。

```
uname -r
```

2. 適切なコマンドを使用して、ioMemory VSL とユーティリティをインストールします。

- RPM コマンド:

- rpm -Uvh iomemory-vsl-<カーネル バージョン>-<VSL バージョン>.x86_64.rpm
- rpm -Uvh lib*.rpm

```
rpm -Uvh fio*.rpm
```

- DEB コマンド:
- `dpkg -i iomemory-vsl-<カーネル バージョン>_<VSL バージョン>_amd64.deb`
- `dpkg -i lib*.deb`
- `dpkg -i fio*.deb`

これらのパッケージの完全なインストール手順については、「[RPM パッケージのインストール](#)」または「[DEB パッケージのインストール](#)」を参照してください。

3. システムをリブートします。
5. `fio-update-iodrive` を使用して、各デバイスのファームウェアを最新バージョンに更新します。

⚠ 停電の回避
更新中に停電が発生しないように、UPS などの保護手段を用意してください。更新中に停電が起こると、デバイスで障害が発生する可能性があります。このユーティリティに関するすべての警告、アラート、およびオプションについては、付録の [fio-update-iodrive](#) ユーティリティ リファレンスを参照してください。

6. サンプルの構文:

```
fio-update-iodrive <iodrive_version.fff>
```

<iodrive_version.fff>は、ファームウェア アーカイブへのパスです。このコマンドによって、すべてのデバイスが選択されたファームウェアに更新されます。特定のデバイスを更新する場合のオプションについては、[ユーティリティ リファレンス](#)を参照してください。

7. システムを再起動します。
8. ioMemory VSL ドライバーを読み込みます。以下にコマンド例を示します。

```
$ modprobe iomemory-vsl
```

詳細については、「[ioMemory VSL ドライバーの読み込み](#)」を参照してください。


i `fio-status` を実行すると、アップグレードされたデバイスに `lebmap` がないという警告が表示されます。これは予期された動作です。次の手順で修正されます。

⚠ データの消去を伴う手順
次の手順で `fio-format` を実行すると、ユーザー データを含むデバイス全体が消去されます。このフォーマットを開始した後は、デバイスを 2.x のドライバーにダウングレードすることはできません。ダウングレードすると、保証が無効となります。

9. `fio-format` を使用して各デバイスをフォーマットします。次に例を示します。

```
fio-format <デバイス>
```

デバイス上のすべてのデータを消去するかどうかを確認するプロンプトが表示されます。

 デバイスの消耗度によっては、フォーマットの実行に長い時間がかかることがあります。

10. すべての ioDrive デバイスを接続します。次に例を示します。

```
fio-attach /dev/fct*
```

11. fio-status を使用してすべてのデバイスの状態を確認します。次に例を示します。

```
fio-status -a
```

これで、ioDrive デバイスがこのバージョンの ioMemory VSL 用に正常にアップグレードされました。ioDrive2 デバイスを取り付けることができます。

付録 H – 付属資料の使用許諾

ドライバーにバイナリ形式で付属する AVR bootloader および tree.h ファイルには、付属資料に関する次の著作権要件を有するコンテンツが含まれています。

AVR Bootloader

Copyright © 2002, 2003, 2004, 2005, 2006, 2007 Eric B. Weddington

All rights reserved.

改変の有無にかかわらず、次の条件をすべて満たす場合に限り、ソースおよびバイナリ形式での本ソフトウェアの再頒布及び使用を許可します。

- ソースコードの再頒布は、上記の著作権情報、本条件一覧、および次の免責条項を保持して行う必要があります。
- バイナリ形式で再頒布する場合は、頒布時に提供される文書またはその他の資料に上記の著作権情報、本条件一覧、および次の免責条項を再表示する必要があります。
- 書面による事前の許可を得た場合を除き、本ソフトウェアから派生する製品の推奨または販売促進のために著作権保持者または寄与者の名前を使用することはできません。

本ソフトウェアは著作権保持者および寄与者によって「現状のまま」提供され、商品性および特定の目的への適合性に関する暗黙の保証を含め、いかなる明示的保証または暗黙的保証も行いません。著作権保持者または寄与者は、事由のいかんを問わず、損害発生の原因いかんを問わず、かつ責任の根拠が契約であるか厳格責任であるか（過失その他の）不法行為であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、本ソフトウェアの使用によって発生した（代替品または代用サービスの調達、使用の喪失、データの喪失、利益の喪失、業務の中断も含め、またそれに限定されない）直接損害、間接損害、偶発的損害、特別損害、懲罰的損害、または必然的損害について、一切責任を負わないものとします。

tree.h

Copyright © 2002 Niels Provos <provos@citi.umich.edu>

All rights reserved.

改変の有無にかかわらず、次の条件をすべて満たす場合に限り、ソースおよびバイナリ形式での本ソフトウェアの再頒布及び使用を許可します。

1. ソースコードの再頒布は、上記の著作権情報、本条件一覧、および次の免責条項を保持して行う必要があります。
2. バイナリ形式で再頒布する場合は、頒布時に提供される文書またはその他の資料に上記の著作権情報、本条件一覧、および次の免責条項を再表示する必要があります。

本ソフトウェアは作成者によって「現状のまま」提供され、商品性および特定の目的への適合性に関する暗黙の保証を含め、いかなる明示的保証または暗黙的保証も行いません。作成者は、事由のいかんを問わず、損害発生の原因いかんを問わず、かつ責任の根拠が契約であるか厳格責任であるか(過失その他の)不法行為であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、本ソフトウェアの使用によって発生した(代替品または代用サービスの調達、使用の喪失、データの喪失、利益の喪失、業務の中断も含め、またそれに限定されない)直接損害、間接損害、偶発的損害、特別損害、懲罰的損害、または必然的損害について、一切責任を負わないものとします。