FUJITSU

Fujitsu Software
Technical Computing Suite V4.0L20

LLIO ユーザーズガイド

J2UL-2555-01Z0(08)
2023年3月

まえがき

本書の目的

本書では、Technical Computing Suite V4.0L20 に含まれる「LLIO(Lightweight Layered IO-Accelerator)」の導入、運用管理、LLIOを利用したジョブの操作方法について説明します。

FEFSについては、「FEFS ユーザーズガイド」を参照してください。

ジョブ運用ソフトウェアについては、以下のマニュアルを参照してください。

- 「ジョブ運用ソフトウェア 概説書」
- 「ジョブ運用ソフトウェア 管理者向けガイド ジョブ管理編」
- 「ジョブ運用ソフトウェア エンドユーザ向けガイド」

本書の読者

本書は、以下の読者が対象です。

- LLIOの導入および運用管理を行うシステム管理者
- LLIOを使用したジョブを操作するエンドユーザ

本書を読むためには、以下の知識が必要です。

- Linux に関する基本的な知識
- ストレージ一般に関する知識
- 「ジョブ運用ソフトウェア 概説書」による、ジョブ運用ソフトウェアの概要についての知識

本書の構成

本書の構成は、以下のとおりです。システム管理者は全章が対象です。エンドユーザは、1章、2章が対象です。

第1章 LLIOの概要

LLIOの概要および構成について説明します。

第2章 LLIOの機能

LLIOの機能について説明します。

第3章 システム運用

LLIOの運用について説明します。

付録A リファレンス

LLIOのシステムコール、コマンド、ライブラリのリファレンスマニュアルです。

付録B メッセージ

LLIOが出力するメッセージについて説明します。

付録C 統計情報の出力項目

統計情報の出力項目について説明します。

用語集

LLIOにおける主な用語を説明します。

本書の表記について

単位の表現

本書では、単位を表現する際の接頭語は以下のとおりです。基本的にディスクサイズは10のべき乗、メモリサイズは2のべき乗で表現します。コマンドの表示や入力時の指定において注意してください。

接頭語	値	接頭語	値
K (kilo)	10^3	Ki (kibi)	2^{10}
M (mega)	10^6	Mi (mebi)	2^{20}
G (giga)	10^9	Gi (gibi)	2^{30}
T (tera)	10^{12}	Ti (tebi)	2^{40}
P (peta)	10^{15}	Pi (pebi)	2^{50}

機種名の表現

本書では富士通製CPU A64FXを搭載した計算機を「FX サーバ」、FUJITSU server PRIMERGYを「PRIMERGY サーバ」(または単に「PRIMERGY」)と呼びます。

コマンド入力例におけるプロンプトについて

コマンド操作を行うために必要な管理者権限によって、プロンプトを区別しています。

- # は管理者権限 (スーパーユーザ) で実行することを意味します。
- $ は管理者権限以外で実行することを意味します。

マニュアル内のアイコンについて

本書では、以下のアイコンを使用しています。

注意

特に注意が必要な事項を説明しています。必ずお読みください。

参照

詳細な情報が書かれている参照先を示しています。

参考

LLIOに関連した参考記事を説明しています。

輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

商標

- Lustreは米国 Seagate Technology LLC の登録商標です。
- Linux® は米国およびそのほかの国における Linus Torvalds の登録商標です。
- Red Hat、Red Hat Enterprise Linux は米国およびそのほかの国において登録されたRed Hat, Inc.の商標です。
- Intel は、アメリカ合衆国およびそのほかの国における Intel Corporation またはその子会社の商標です。
- そのほか、本マニュアルに記載されている会社名および製品名は、それぞれ各社の商標または登録商標です。

出版年月および版数

版数	マニュアルコード
2023年3月 第1.8版	J2UL-2555-01Z0(08)
2022年9月 第1.7版	J2UL-2555-01Z0(07)

版数	マニュアルコード
2022年3月 第1.6版	J2UL-2555-01Z0(06)
2021年11月 第1.5版	J2UL-2555-01Z0(05)
2021年8月 第1.4版	J2UL-2555-01Z0(04)
2020年12月 第1.3版	J2UL-2555-01Z0(03)
2020年9月 第1.2版	J2UL-2555-01Z0(02)
2020年6月 第1.1版	J2UL-2555-01Z0(01)
2020年2月 初版	J2UL-2555-01Z0(00)

変更履歴

変更内容	変更箇所	版数
llio_transfer コマンドの --sync オプションを追加しました。	A.2.3	第1.8版
第2階層ストレージのキャッシュ領域に関する注意事項を追加しました。	2.1.1	第1.7版
llio_transfer コマンドの復帰値に関する記述を変更しました。	A.2.3	
第2階層ストレージのキャッシュ領域のキャッシュが削除される契機の説明に、具体例を追加しました。	2.1.1	第1.6版
そのほか、誤記を修正しました。	-	
LLIO 領域の選択に関する説明を追加しました。	2.2.1	第1.5版
LLIO 性能情報の採取時の注意事項を追加しました。	2.7.2	
第1階層ストレージの3つの領域に関して、ジョブあたりにオープン可能または書き込み可能なファイル数に関する注意事項を追加しました。	2.1	第1.4版
ノード内テンポラリ領域の説明を修正しました。	2.1.3	
第2階層ストレージに残存した共有テンポラリ領域と第2階層ストレージのキャッシュ領域のファイルやディレクトリについて、説明を修正しました。	3.3.4 3.3.4.1 3.3.4.2	第1.3版
LLIO性能情報の出力項目"Uncompleted-file"の説明を訂正し、注意事項を追加しました。	C.1	
共有テンポラリ領域の説明を修正しました。	2.1	第1.2版
第1階層ストレージに対するDirect I/Oでのreadおよびwriteサイズの単位を64KBに訂正しました。	2.1	
非同期クローズ機能が有効であるときのファイルデータの書出し失敗について、注意事項を追加しました。	2.4.4	
クライアントノード間でのメタデータの一貫性保証に関する仕様の対処について注意事項を追加しました。	3.4	
llio_transfer コマンドの注意事項を修正しました。	A.2.3	
クライアントノード間での一貫性保証についての説明を改善しました。	2.1.1 2.1.2 2.1.3	第1.1版
共有テンポラリ領域とノード内テンポラリ領域において、ファイルの管理のために消費されるサイズについて記載しました。	2.1.2 2.1.3	
trash配下のディレクトリはジョブ運用中でも削除できることを追加しました。	3.3.4.2	
OSの更新パッケージ適用時の注意事項は不要になったため削除しました。	3.4	
LLIO性能情報の説明を改善しました。	C.1	

変更内容	変更箇所	版数
llio_transferコマンドのメッセージ6458を追加しました。	B.2.3	

目　次

第1章 LLIOの概要

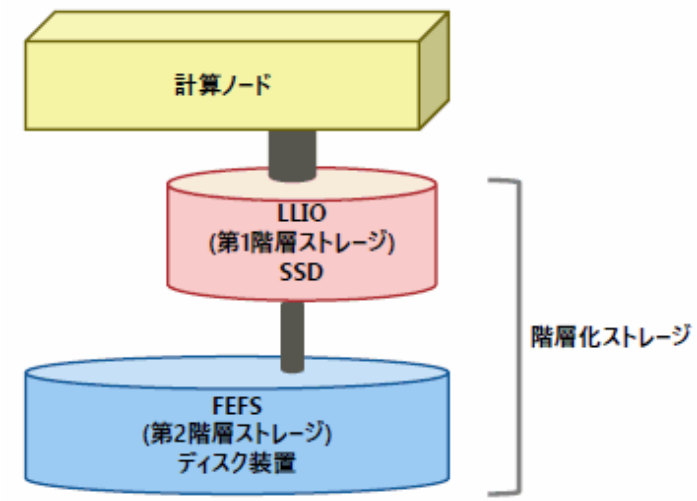
ここでは、LLIO(Lightweight Layered IO-Accelerator)の概要について説明します。システム管理者とエンドユーザが対象です。

1.1 特長

LLIOは、並列分散ファイルシステムであるFEFSと計算ノードの間に位置する、SSD を使用した高性能なファイルシステム、またはそれを実現する技術です。ジョブ用の一時ファイルをFEFSに書き出さない、またはLLIOからFEFSへの書出しを計算処理中に非同期に行うことで高速化を実現します。

FEFSとLLIOを組み合わせることで、それぞれの特徴を活かした高速かつ大容量な階層化ストレージを実現します。計算ノードから見たビューで上位層になるLLIOを第1階層ストレージ、下位層になるFEFSを第2階層ストレージと呼び、この2つを合わせて階層化ストレージと呼びます。

図1.1 階層化ストレージ



参考

LLIOは、Linux標準のPOSIXインターフェースに対応しているため、アプリケーションの修正は不要です。

LLIOは、以下の機能を使用して高性能なファイルシステムを実現します。これらの機能はジョブ投入時に指定できます。

ジョブに最適な第1階層ストレージ領域

第1階層ストレージには異なる特性を持つ3種類の領域があります。エンドユーザはジョブ投入時に、それぞれの領域に対して、最適な大きさを指定できます。

参照

第1階層ストレージ領域の詳細は、”2.1 第1階層ストレージの3つの領域”を参照してください。

ストライプ機能

1つのファイルを複数のストレージI/Oノードに分割して同時に転送するストライプ機能を提供します。

参照

ストライプ機能の詳細は、”2.4.2 第1階層ストレージへのストライプ機能”、”2.4.3 第2階層ストレージへのストライプ機能”を参照してください。

共通ファイル配布機能

ジョブ実行時、負荷集中が想定される第2階層ストレージ上のファイル(a.out や入力ファイルなど)の複製を第1階層ストレージに作成し負荷分散を実現する、共通ファイル配布機能を提供します。
負荷集中が想定される第2階層ストレージ上のファイルを第1階層ストレージ上に作成することで、計算ノードからストレージI/Oノードへのアクセスを分散させます。この処理はアプリケーションを起動する前から開始させるために、ジョブスクリプトの先頭に専用のコマンドを記述することで実現できます。

参照

共通ファイル配布機能の詳細は、"2.3 共通ファイル配布機能"を参照してください。

計算ノード内キャッシュ機能

アプリケーションが使用していない計算ノードの空きメモリをページキャッシュとして使用する計算ノード内キャッシュ機能を提供します。計算ノードのユーザ用メモリを多く使用しないジョブは、空きメモリを持ったままジョブを実行することになります。小さなI/Oサイズで書き出す場合や同じファイルを何回も読み出す場合、空きメモリをページキャッシュとしてできるだけ大きく割り当てることで、性能を引き出すことができます。空きメモリをページキャッシュとして使用することで、ジョブからのI/Oを計算ノード内で折り返しI/Oの高速化を実現します。計算ノード内キャッシュは、ユーザメモリ領域の中から割り当てられるので、アプリケーションが使用する領域を意識して指定する必要があります。

参照

計算ノード内キャッシュ機能の詳細は、"2.5 計算ノード内キャッシュ機能"を参照してください。

またLLIOの利用をサポートするための機能として、以下を実現します。これらの機能はジョブ投入時に指定できます。

統計情報

LLIOは、多くの統計情報を採取しジョブ実行者であるエンドユーザおよびシステム管理者に提供します。システム管理者やエンドユーザはこの情報を参照し、I/Oチューニングやシステムトラブルの調査に役立てることができます。

参照

- エンドユーザ向けの統計情報の詳細は、"2.7 統計情報"を参照してください。
- システム管理者向けの統計情報の詳細は、"2.8.2 システム統計情報"を参照してください。

ノード異常時のジョブの取り扱い

ジョブ実行中に発生した計算ノードおよびストレージI/Oノードの異常はジョブ運用ソフトウェアによって検出され、当該ノードはジョブ運用から切り離されます。この時当該ノードで実行されていたジョブは異常終了し、第2階層ストレージに書き出されなかったファイルは、エンドユーザに通知されます。

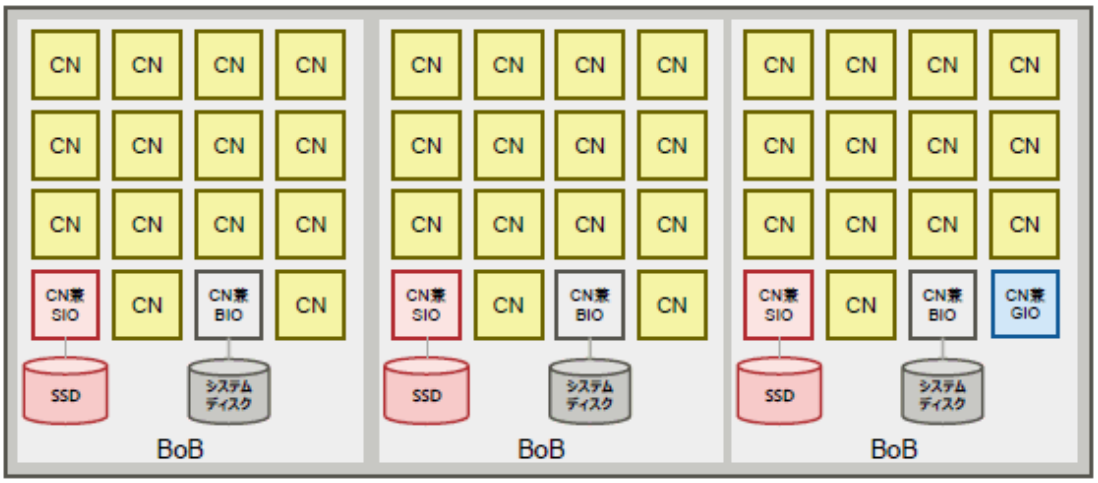
参照

第2階層ストレージに書き出されなかったファイルをエンドユーザに通知する機能を未書出しファイル一覧取得機能といいます。未書出しファイル一覧取得機能の詳細は、"2.6 未書出しファイル一覧取得機能"を参照してください。

1.2 システム構成

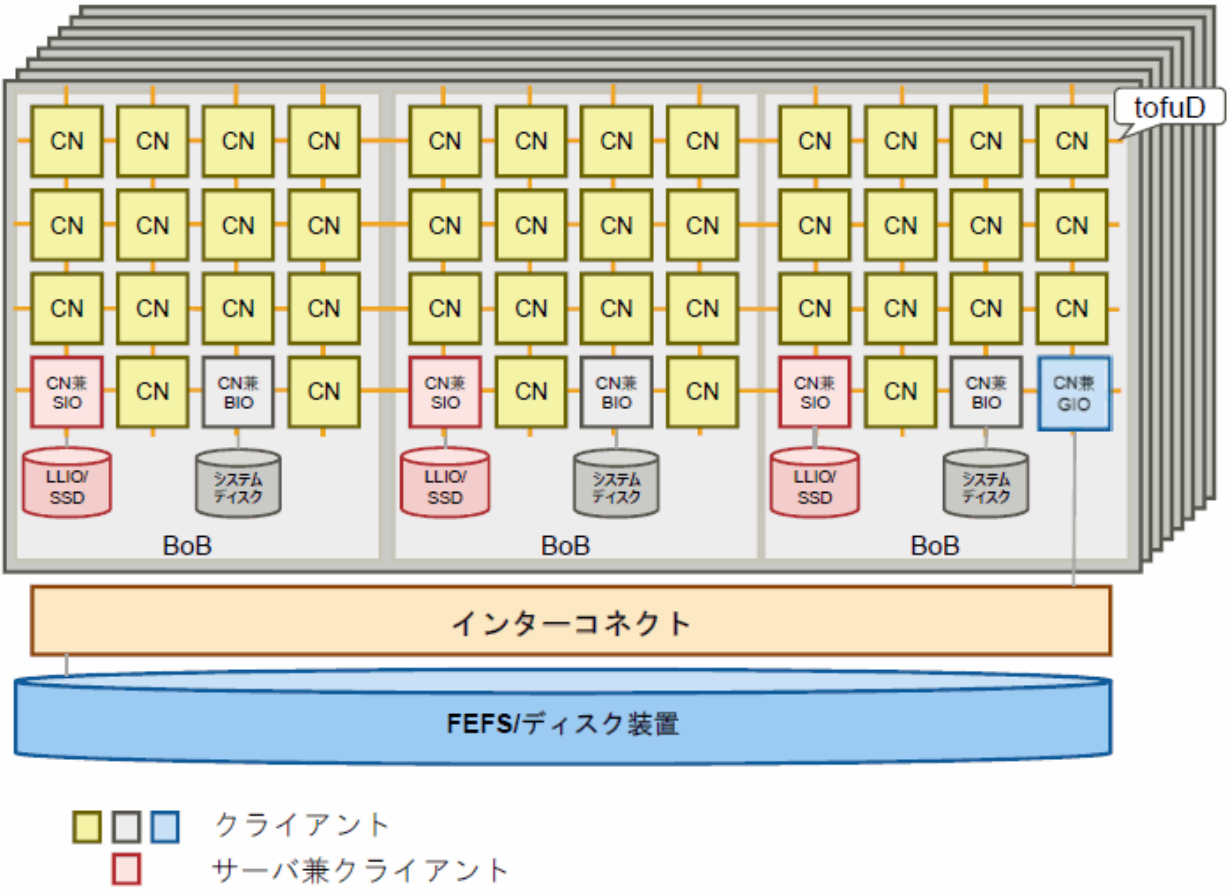
LLIOはサーバ、クライアント構成を持つファイルシステムです。LLIOのシステム構成について計算クラスタの基本構成単位であるBoB(16ノードの計算ノードを制御するブート単位)をもとに説明します。LLIOの物理構成を以下に示します。

図1.2 LLIOの物理構成



LLIOのシステム環境は、計算ノード、ストレージI/Oノード、グローバルI/Oノード、ブートI/Oノードから構成されます。

図1.3 LLIOのシステム構成



参照

LLIOで使用するノード種別を以下に示します。各ノードの詳細は、マニュアル「ジョブ運用ソフトウェア 概説書」を参照してください。

表1.1 ノード種別名と意味

ノード名	略称	説明
ストレージI/Oノード	SIO	第1階層ストレージに対する入出力を担うI/Oノードです。 ストレージI/Oノードには、第1階層ストレージを構成するSSDが接続されています。 ストレージI/Oノードは、計算ノードも兼ねます。
グローバルI/Oノード	GIO	第2階層ストレージ(グローバルファイルシステム)に対する入出力を中継するノードです。 グローバルI/Oノードは、計算ノードも兼ねます。
ブートI/Oノード	BIO	ノードのブートサーバになるI/Oノードです。 ブートI/Oノードは計算ノードも兼ねます。
計算ノード	CN	ジョブが動作するノードです。

1.2.1 サーバ構成

LLIOサーバとして利用できるノード種別を以下に示します。

- ストレージI/Oノード

参考

ストレージI/Oノードは第2階層ストレージのクライアントを兼ねています。

1.2.2 クライアント構成

LLIOクライアントとして利用できるノード種別を以下に示します。

- 計算ノード
- ストレージI/Oノード
- グローバルI/Oノード
- ブートI/Oノード

1.2.3 ネットワーク構成

以下のネットワークをサポートします。

- TofuインターコネクトD
  - 計算ノード、ストレージI/Oノード、グローバルI/Oノード、ブートI/Oノードの間

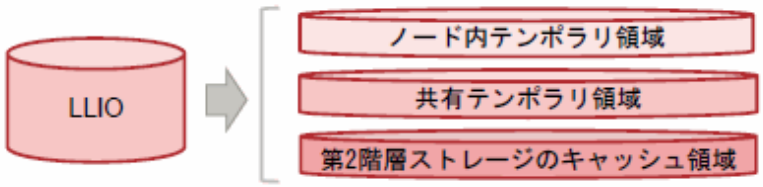
第2章 LLIOの機能

ここではLLIOの機能について説明します。システム管理者とエンドユーザが対象です。

2.1 第1階層ストレージの3つの領域

第1階層ストレージの3つの領域である、第2階層ストレージのキャッシュ領域、共有テンポラリ領域、ノード内テンポラリ領域の特性について以下に示します。

図2.1 第1階層ストレージの利用形態



第2階層ストレージのキャッシュ領域

ジョブからFEFSへのI/Oをキャッシュし、アクセスを高速化する領域です。

共有テンポラリ領域

ジョブに割り当てられた計算ノード間で共有できるジョブの一時領域です。
LLIOは、共有テンポラリ領域の通常ファイルのデータを第1階層ストレージ上に格納し、各種ファイルのメタデータを第2階層ストレージ上に格納します。statfs(2)では、ブロック情報は第1階層ストレージ上の値、inode情報は第2階層ストレージ上の値を報告します。共有テンポラリ領域に作成できるデータあり通常ファイルの最大数は、「領域サイズ*25/100/300」です。ただし、第1階層ストレージへのストライプカウントが1より大きければ、最大数はストライプカウントで割った値になります。

ノード内テンポラリ領域

ジョブに割り当てられたそれぞれの計算ノードで利用できるジョブの一時領域です。
ノード内テンポラリ領域の最大ファイル数は、「領域サイズ*25/100/300」です。

表2.1 3つの領域の特性

領域名	参照範囲	第2階層ストレージへの書出し	生存期間
第2階層ストレージのキャッシュ領域	すべての計算ノード	する	ジョブ終了時に削除されない
共有テンポラリ領域	同一ジョブ内の計算ノード	しない	ジョブ終了時に削除される
ノード内テンポラリ領域	1つの計算ノード	しない	ジョブ終了時に削除される

参考

LLIO ファイルシステムとしての単一ディレクトリ内のサブディレクトリ/ファイルの最大数や、ACLの最大エントリはFEFS の仕様に準じます。

注意

- 第2階層ストレージのキャッシュ領域、共有テンポラリ領域、ノード内テンポラリ領域に対しDirect I/Oで read、writeを実行する場合には、64KB単位で実行してください。
- ジョブがオープンしているファイルの数が多くなると、ストレージI/Oノードのメモリが不足しファイルアクセスに失敗する場合があります。ジョブあたりにオープン可能なファイル数の目安は以下のとおりです。

  1,024 × ジョブに割り当てられた計算ノード数

※ 同一のファイルを複数の計算ノードからオープンする場合は、計算ノード数によらず1ファイルとして計算してください。

また、非同期クローズが有効なとき、多数のファイルに書き込むとストレージI/Oノードのメモリが不足しファイルアクセスに失敗する場合があります。
ジョブあたりに書き込み可能なファイル数の目安は以下のとおりです。

1,024 × ジョブに割り当てられた計算ノード数

※ 同一のファイルに複数の計算ノードから書き込む場合は、計算ノード数によらず1ファイルとして計算してください。

上記のファイル数を見積る際の注意事項は以下のとおりです。

a. ファイル数は第2階層ストレージのキャッシュ領域、共有テンポラリ領域、ノード内テンポラリ領域のファイル数の合計で計算してください。

b. LLIO のストライプカウントが2以上の場合、ストライプ毎に1ファイルとして計算してください。
例えば、ファイルA をオープンしている場合、ストライプを考慮した「オープンしているファイル数」は以下の式で求められます。

ファイルAについての「オープンしているファイル数」 =
(ファイルサイズ / ストライプサイズ) (*1) と (ストライプカウント) の最小値

(*1) 小数点以下は切り上げてください。

c. llio_transfer コマンドで転送した共通ファイルは、以下をファイル数として計算してください。

共通ファイル数 × ジョブに割り当てられた計算ノード数

転送した共通ファイルに対しては、b. に記載のストライプの考慮は不要です。

以降では第1階層ストレージの3つの領域の詳細を説明します。

2.1.1 第2階層ストレージのキャッシュ領域

LLIOは、第2階層ストレージへのより高速なアクセスを実現するため、第1階層ストレージ上に第2階層ストレージがキャッシュされた領域を用意します。これを第2階層ストレージのキャッシュ領域といいます。
エンドユーザは、この領域を第2階層ストレージと同じように扱うことができます。第2階層ストレージのキャッシュは、ジョブの計算とは非同期にジョブの出力結果などを第2階層ストレージに書き出します。

第2階層ストレージのキャッシュ領域の仕様を説明します。

表2.2 第2階層ストレージのキャッシュ領域の仕様

項目	特性
参照範囲	すべての計算ノードから参照できます。
第2階層ストレージへの書出し	計算ノードからの書出し要求とは非同期に書き出します。
生存期間	第2階層ストレージのキャッシュ領域は、ジョブ起動前に初期化され、ジョブ終了時に削除されます。 第2階層ストレージへの書出し途中でジョブ実行可能時間制限を超えるなどしてジョブが中断された場合は、第2階層ストレージのキャッシュ領域は書出しが完了していない場合でも削除されます。 書き出されなかったファイルは、未書出しファイルの一覧としてファイルに出力します。 また、上記以外で第2階層ストレージのキャッシュ領域のキャッシュが削除されるタイミングは以下のとおりです。 ・ ファイルを削除した場合 ・ 第2階層ストレージのキャッシュ領域が一杯になった場合 (この場合は LRU 方式で古いものから削除されます) ・ Direct I/O (write または read) が実行された場合 (この場合は Direct I/O した箇所のキャッシュが削除されます)
クライアントノード間での一貫性保証	NFS(version3)相当です。 このため、NFS と同様に以下のような事象が発生することがあります。

項目	特性
	• 同一ファイルに対するノード間での書き込み順序は保証されないため、同一ファイルの重複するオフセットにデータをWRITEした場合、結果のファイルデータはクライアント上でWRITEが行われた順序とは異なっていることがあります。 • あるノードで行われたデータの更新をほかのノードで即時検出できないことがありますが、一定時間後には検出できるようになります。 • あるノードで実行されたファイル操作(ファイル/ディレクトリの作成や削除、属性情報の変更など)をほかのノードで即時検出できないことがありますが、一定時間後には検出できるようになります。

エンドユーザはジョブ投入時に、共有テンポラリ領域の大きさとノード内テンポラリ領域の大きさを指定することで第2階層ストレージのキャッシュ領域の大きさを間接的に指定できます。

参照

使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

注意

LLIO の第2階層ストレージのキャッシュ領域上で保持・管理されているファイルに第2階層ストレージから直接 write した場合、LLIO ではそれを認識することができず、第2階層ストレージのキャッシュ領域から当該ファイルにアクセスしても write されたデータが読み込めないことがあります。

例えば、会話型ジョブを実行し、そのジョブで使用している第2階層ストレージのキャッシュ領域のファイルをログインノード上で第2階層ストレージへ直接アクセスして更新し、そのファイルを再度ジョブで使用するようなケースが該当します。

このような問題を避けるため、同一ファイルに対して第2階層ストレージのキャッシュ領域と第2階層ストレージから並行しての read/write は行わないようにしてください。

2.1.2 共有テンポラリ領域

LLIOは、同一ジョブ内の複数ジョブプロセス間で使用するファイルの一時領域として共有テンポラリ領域を提供します。この領域は同一ジョブ内であれば複数の計算ノード間で共有できます。共有テンポラリ領域は、ジョブ開始時に使用可能になり、ジョブ終了時に使用できなくなります。そのためエンドユーザはジョブ実行結果ファイルなど退避しておきたいファイルを、第2階層ストレージに配置する必要があります。

共有テンポラリ領域の仕様を説明します。

表2.3 共有テンポラリ領域の仕様

項目	特性
参照範囲	ジョブに割り当てられた計算ノード間で参照できます。同じジョブの中であれば複数の計算ノードから参照できます。
第2階層ストレージへの書出し	ファイルは第1階層ストレージ上だけに存在し、第2階層ストレージへは書き出されません。
生存期間	ジョブ起動前に初期化され、ジョブ終了時に削除されます。
クライアントノード間での一貫性保証	NFS(version3)相当です。

共有テンポラリ領域には、第2階層ストレージへのメタアクセスから分離するために、専用の MDT を指定することができます。

また、エンドユーザはジョブ投入時に、共有テンポラリ領域の大きさを指定できます。
共有テンポラリ領域は、ファイルのデータ以外にファイルの管理データによっても消費されます。したがって、ファイルのデータの合計が、領域のサイズ未満でもENOSPCになることがあります。ファイルの管理データによって消費される容量の目安は以下のとおりです。

300byte×ファイル数

参照

使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

2.1.3 ノード内テンポラリ領域

LLIOは、ジョブに割り当てられた個々の計算ノードの中にあるプロセス間でローカルに使用するファイルの一時領域として、ノード内テンポラリ領域を提供します。ノード内テンポラリ領域は、ジョブ開始時に使用可能になり、ジョブ終了時に使用できなくなります。そのためエンドユーザは、ジョブ実行結果ファイルなど退避しておきたいファイルを、第2階層ストレージに配置する必要があります。

ノード内テンポラリ領域の仕様を説明します。

表2.4 ノード内テンポラリ領域の仕様

項目	特性
参照範囲	ジョブに割り当てられた個々の計算ノードの中で動作するプロセス間でのみ参照できます。
第2階層ストレージへの書出し	ファイルは第1階層ストレージ上だけに存在し、第2階層ストレージへは書き出されません。
生存期間	ジョブ起動前に初期化され、ジョブ終了時に削除されます。

エンドユーザはジョブ投入時に、ノード内テンポラリ領域の大きさを指定できます。
ノード内テンポラリ領域は、ファイルのデータ以外にファイルの管理データによっても消費されます。したがって、ファイルのデータの合計が、領域のサイズ未満でもENOSPCになることがあります。ファイルの管理データによって消費される容量の目安は以下のとおりです。

```
300byte×ファイル数
```

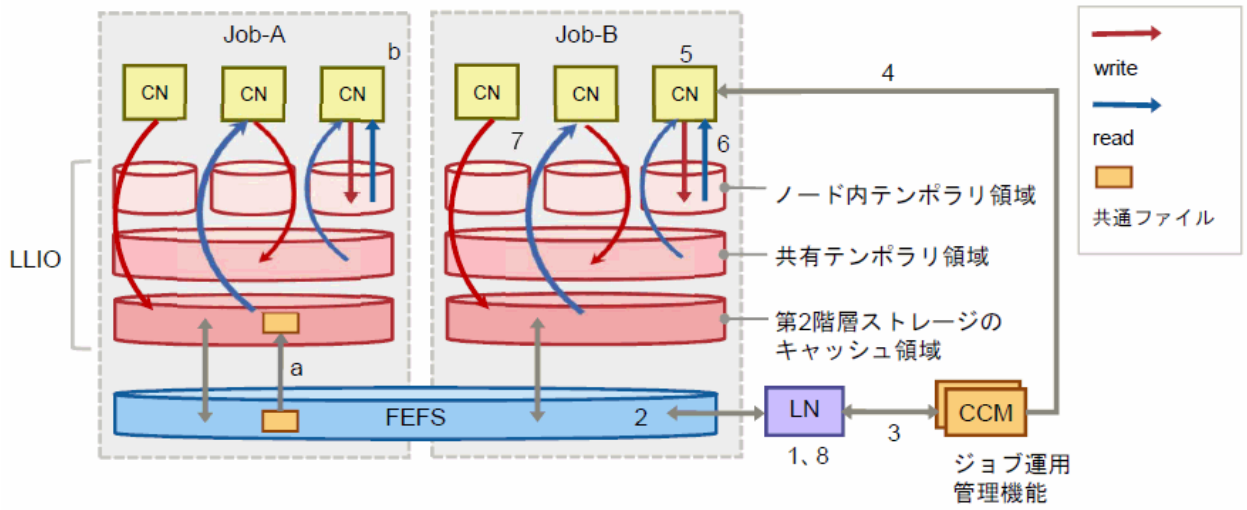
参照

使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

2.2 第1階層ストレージとジョブ

第1階層ストレージとジョブの関係について、ジョブの利用イメージをもとに説明します。

図2.2 第1階層ストレージとジョブ



エンドユーザは、プログラムをジョブ運用ソフトウェアが提供するジョブ運用環境で実行し結果を得ます。

以下に流れを示します。

1. エンドユーザは、システム利用のための入り口であるログインノード(LN)にログインします。

2. エンドユーザは、第2階層ストレージであるFEFS上にジョブスクリプトを配置します。ジョブ運用管理機能のコマンドで投入されたジョブの情報は、ジョブ運用管理機能へ送られ、バッチ処理されます。

   a. エンドユーザは、第2階層ストレージ上にある実行ファイルや設定ファイルなど、計算ノードからアクセスが集中すると予想されるファイルへの負荷集中を避けるため、事前に第1階層ストレージ上に配置したいファイルをジョブスクリプトに指定します。

   参照

   詳細は、”2.3 共通ファイル配布機能”を参照してください。

   b. エンドユーザは、計算ノード統計情報を取得するライブラリ関数を使ったプログラムをジョブスクリプト内に指定できます。

   参照

   詳細は、”2.7.3 計算ノード統計情報”を参照してください。

3. エンドユーザは、ジョブの実行を計算クラスタ管理ノード(CCM)上のジョブ運用ソフトウェアのジョブ運用管理機能に依頼します。これをジョブの投入と呼びます。
   エンドユーザはジョブを投入する際、“表2.5 ジョブ投入時に指定できるLLIOのパラメーター”を指定することで、ジョブに最適なLLIO環境を構築できます。
4. ジョブ運用管理機能はジョブを計算ノードに割り当てます。
5. ジョブが計算ノードで実行されます。
6. 計算ノード上で実行しているジョブは、ノード内テンポラリ領域、共有テンポラリ領域、第2階層ストレージのキャッシュ領域上のジョブの実行に必要なファイルにアクセスします。
7. ジョブは、第1階層ストレージ上の”第2階層ストレージのキャッシュ領域”を通じて、第2階層ストレージに結果を出力します。
8. エンドユーザは、ログインノードで第2階層ストレージ上のジョブ出力結果を参照します。

表2.5 ジョブ投入時に指定できるLLIOのパラメーター

分類	用途	機能説明
第1階層ストレージのサイズの指定	第2階層ストレージのキャッシュの領域の指定	2.1.1 第2階層ストレージのキャッシュ領域
	共有テンポラリ領域の指定	2.1.2 共有テンポラリ領域
	ノード内テンポラリ領域の指定	2.1.3 ノード内テンポラリ領域
第1階層ストレージの高速化	第2階層ストレージから計算ノードへ読み込んだファイルを第1階層ストレージへキャッシュする指定	2.4.1 read時における第1階層ストレージへのキャッシュ機能
	第1階層ストレージにファイルを分散配置する場合の、ストライプサイズとストライプカウントの指定	2.4.2 第1階層ストレージへのストライプ機能
	第2階層ストレージにファイルを分散配置する場合の、ストライプサイズとストライプカウントの指定	2.4.3 第2階層ストレージへのストライプ機能
	階層化ストレージ上のファイルのクローズを非同期クローズにする指定	2.4.4 非同期クローズ機能
計算ノード内キャッシュの利用	ジョブが利用できるメモリ資源量の中から、計算ノード内キャッシュに割り当てるメモリサイズの指定	2.5.1 計算ノード内キャッシュのメモリサイズ
	階層化ストレージからファイルを読むときに、計算ノード内キャッシュにキャッシュする指定	2.5.2 read時における計算ノード内キャッシュへのキャッシュ機能
	階層化ストレージ内の連続した領域を計算ノード内キャッシュに読み込む場合、自動的に先読みする指定	2.5.3 計算ノード内キャッシュへの自動先読み機能
	計算ノード内キャッシュの使用有無を切り替えるwrite サイズの閾値の指定	2.5.4 計算ノード内キャッシュの使用有無を切り替えるwriteサイズの閾値

分類	用途	機能説明
ジョブ異常時の扱い	ジョブ終了時、第1階層ストレージ上に、第2階層ストレージに対して未書出しのファイルが残っていた場合、ファイル名の一覧を出力する指定	2.6 未書出しファイル一覧取得機能
統計情報	ジョブ統計情報の取得	2.7.1 ジョブ統計情報
	LLIO性能情報の取得	2.7.2 LLIO性能情報
	計算ノード統計情報の取得	2.7.3 計算ノード統計情報

2.2.1 LLIO 領域の選択

エンドユーザはジョブ投入時に"表2.6ジョブ投入時に指定できるLLIOの環境変数"にある環境変数を指定することでアクセス可能にしたいLLIO の領域を選択することができます。

表2.6 ジョブ投入時に指定できる LLIO の環境変数

環境変数名	詳細	注意事項
PJM_LLIO_GFSCACHE	使用する第 2 階層ストレージのキャッシュ領域を指定します。 "pjsub -x PJM_LLIO_GFSCACHE=*パス名*" で指定することができます。 複数指定する場合は、":" 区切りで指定します。	本環境変数を指定しない場合は、エンドユーザのホームディレクトリ上にある第 2 階層ストレージのキャッシュ領域が使用されます。 本環境変数が設定されている場合、エンドユーザのホームディレクトリがある第 2 階層ストレージのキャッシュ領域は、本環境変数に指定されていなくても使用することができます。
PJM_LLIO_SHAREDTMP	共有テンポラリ領域の MDT として使用する第 2 階層ストレージを指定します。 "pjsub -x PJM_LLIO_SHAREDTMP=*パス名*" で指定することができます。	本環境変数を指定しない場合は、エンドユーザのホームディレクトリ上にある第2階層ストレージが使用されます。

以下に環境変数の指定例を示します。

[pjsub コマンド実行例]

```
pjsub -x PJM_LLIO_GFSCACHE=/fefs1:/fefs2 -x PJM_LLIO_SHAREDTMP=/fefs3 ...
```

[ジョブスクリプト例]

```
:
#PJM -x PJM_LLIO_GFSCACHE=/fefs1:/fefs2
#PJM -x PJM_LLIO_SHAREDTMP=/fefs3
:
```

注意

ファイルシステムの故障により、ジョブ運用を継続するためにファイルシステムが切り離されている場合があります。環境変数に切り離されているファイルシステムを指定すると、ジョブはエラーになります。ファイルシステムの切離しについては、マニュアル「FEFS ユーザーズガイド」を参照してください。

2.3 共通ファイル配布機能

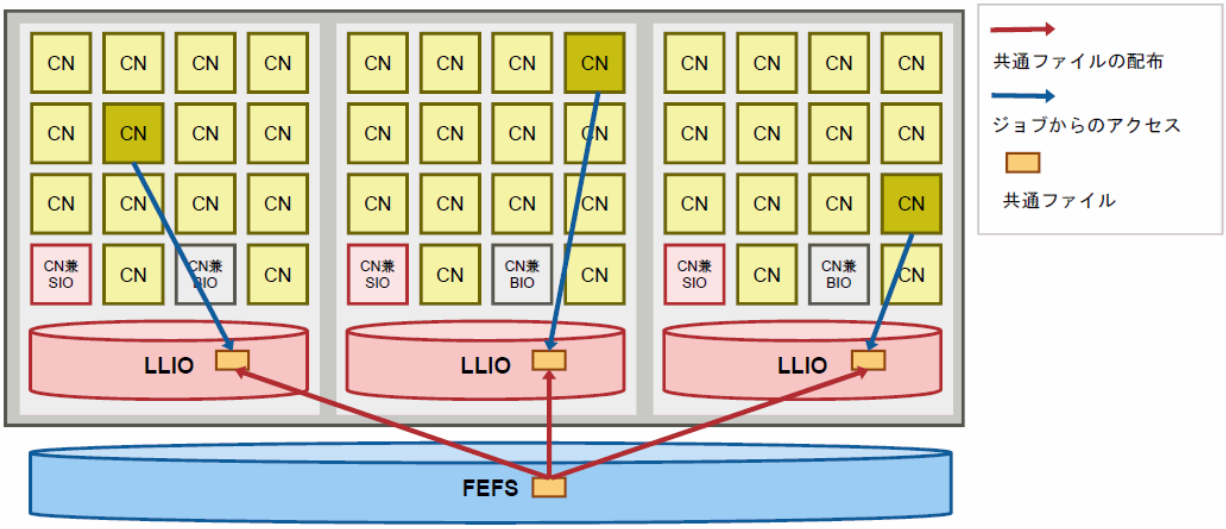
ジョブ開始時、第2階層ストレージ上にある実行ファイルや設定ファイルなど、すべての計算ノードから読み込まれるファイル(共通ファイル)には、アクセスが集中します。その結果、特定のストレージI/Oノードにアクセスが集中し、性能劣化を引き起こす可能性があります。LLIOは、そのような状態を避けるため、第1階層ストレージ上の第2階層ストレージのキャッシュ領域に共通ファイルを配布することで、アクセスを分散させるための機能を提供します。これを共通ファイル配布機能といいます。エンドユーザはジョブスクリプト内に共通ファイルの配布を指定できます。

参照

使用方法の詳細は、”A.2.3 llio_transferコマンド”、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

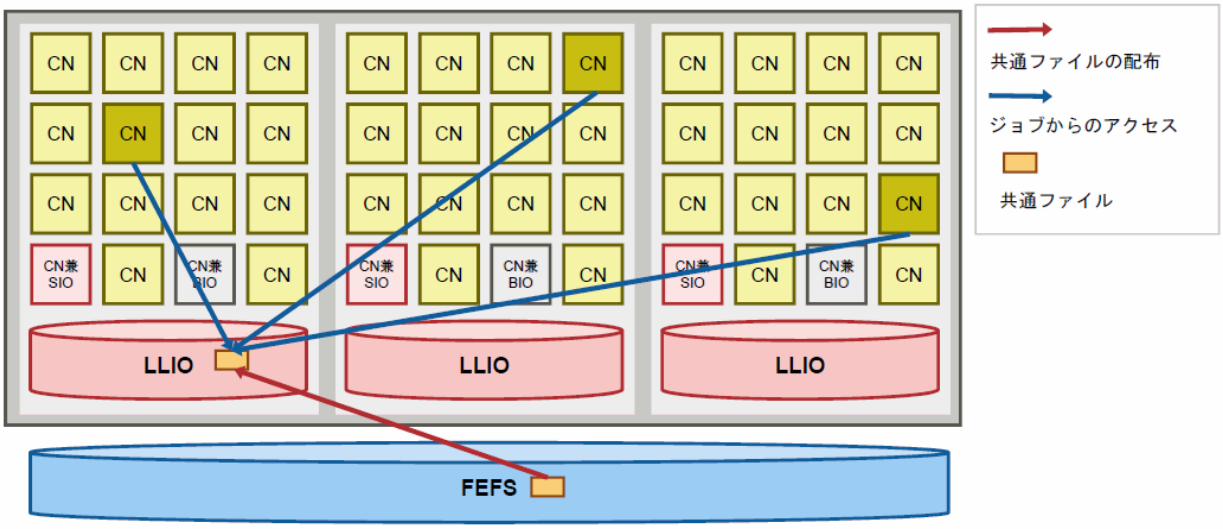
以下に実行イメージを示します。

図2.3 共通ファイル配布機能を使用する場合



共通ファイル配布機能は、ジョブが割り当てられたすべてのストレージI/Oノードの第1階層ストレージの第2階層ストレージのキャッシュ領域にファイルを配布します。計算ノード上のジョブは、物理的に一番近いストレージI/Oノードに対しアクセスすることでストレージI/Oノードのアクセス負荷を分散します。

図2.4 共通ファイル配布機能を使用しない場合



共通ファイル配布機能を使用しない場合、計算ノード上のジョブは、1つのストレージI/Oノードにアクセスが集中し、性能劣化を引き起こす可能性があります。

2.4 第1階層ストレージの高速化

2.4.1 read時における第1階層ストレージへのキャッシュ機能

LLIOは、第2階層ストレージから計算ノード内キャッシュに読み込んだファイルを第1階層ストレージにキャッシュする機能を提供します。この機能は、第1階層ストレージにキャッシュすることでジョブ内が同じファイルを複数回読み込む場合に性能向上を実現します。

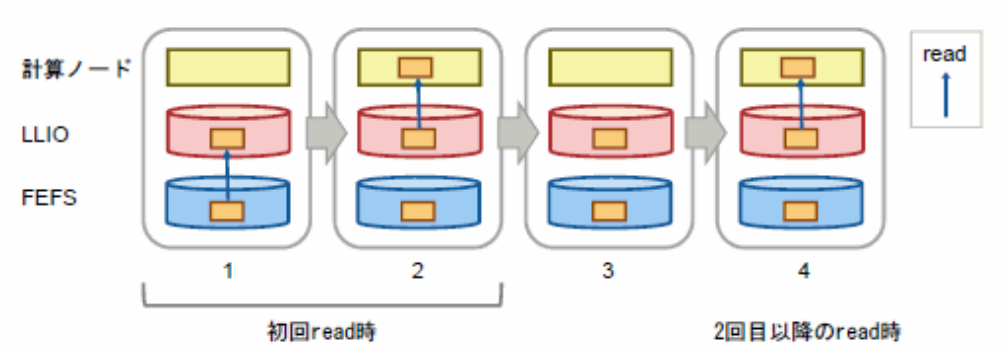
エンドユーザはジョブ投入時に、read時における第1階層ストレージへのキャッシュを指定できます。

参照

- 計算ノード内キャッシュの詳細は、"2.5 計算ノード内キャッシュ機能"を参照してください。
- 使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

以下に第1階層ストレージへのキャッシュ機能の動作を示します。

図2.5 第1階層ストレージへのキャッシュ機能の動作



1. 初回read時、第2階層ストレージのファイルを第1階層ストレージに読み込みます。
2. 第1階層ストレージのファイルを計算ノード内キャッシュに読み込みます。
3. ファイルは第1階層ストレージにキャッシュされています。
4. 2回目以降のread時、計算ノードは第1階層ストレージのファイルを読み込みます。
   第2階層ストレージのファイルを読み込む必要はありません。

2.4.2 第1階層ストレージへのストライプ機能

LLIOは、1つのファイルを複数のストレージI/Oノードにストライピングする第1階層ストレージへのストライプ機能を提供します。第1階層ストレージへのストライプ機能には、以下の効果があります。

- 1つのSSDの物理的な容量を超えるサイズのファイルを作成できます。
- 1つのファイルを複数のストレージI/Oノードの第1階層ストレージに分散して格納することで、ファイルアクセスの帯域幅が向上します。

エンドユーザはジョブ投入時に、アプリケーションのI/O特性に合わせてストライプサイズやストライプカウントを指定します。これにより、第1階層ストレージへの効率の良い転送が可能になります。

参照

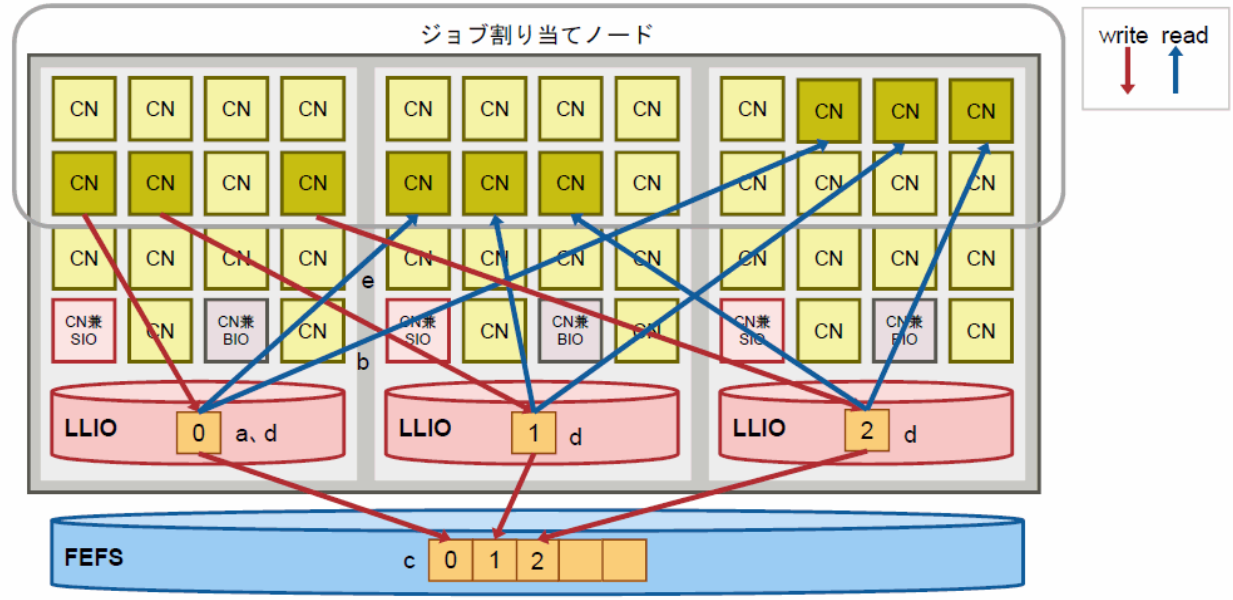
使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

参考

ストライプ機能が使用できるのは、共有テンポラリ領域と第2階層ストレージのキャッシュ領域です。ノード内テンポラリ領域には使用できません。

第1階層ストレージへのストライプ機能を使用して、第2階層ストレージのキャッシュ領域に対しストライプサイズに1MiB、ストライプカウントに3を指定した場合の例を以下に示します。

図2.6 第1階層ストレージへのストライプ機能の動作



a. LLIOは、先頭のストレージI/Oノードを決定します。

b. LLIOは、3つのストレージI/Oノードに対しストライプサイズごとにラウンドロビンでファイルを書き出します。

c. 3つのストレージI/Oノードの第1階層ストレージに格納したファイルは、必要に応じてバックグラウンドで第2階層ストレージへ書き出します。

d. エンドユーザは第2階層ストレージへ書き出したあとも、第1階層ストレージにファイルを残しておく指定もできます。

e. 計算ノードが、第1階層ストレージに書き出されたファイルを読み込みます。

2.4.3 第2階層ストレージへのストライプ機能

LLIOは、計算ノードから第2階層ストレージへのストライプ設定、参照機能を提供します。第2階層ストレージに対するストライプサイズやストライプカウントを指定することで、第2階層ストレージのキャッシュから第2階層ストレージへの効率の良いデータ転送が可能になります。

エンドユーザはジョブスクリプト内に第2階層ストレージへのストライプ設定や参照を指定できます。

参照

使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

参考

第2階層ストレージのクライアントから第2階層ストレージへのストライプ設定、参照には、lfsコマンドのsetstripeサブコマンド、getstripeサブコマンドが提供されています。詳細は、マニュアル「FEFS ユーザーズガイド」を参照してください。

2.4.4 非同期クローズ機能

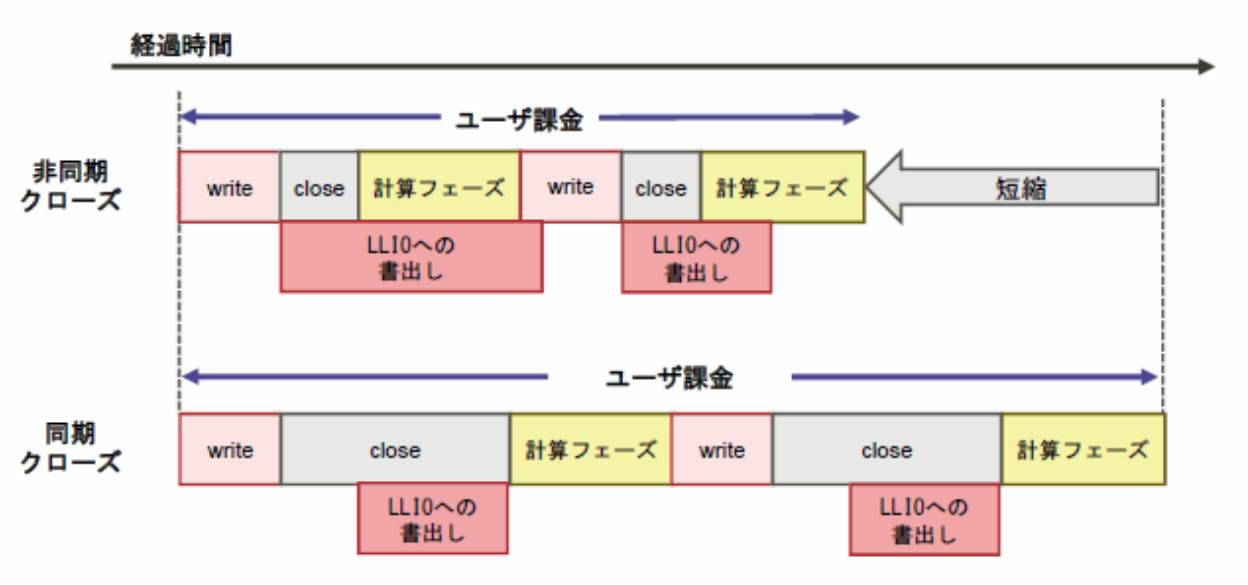
LLIOは、計算ノード内キャッシュから第1階層ストレージ、および第2階層ストレージのキャッシュから第2階層ストレージへの書出しにおいて、ファイルクローズ時の書出しを非同期に行う非同期クローズ機能を提供します。この機能を使用することで書出し完了を待たされることがなくなりジョブの終了が早くなる効果があります。この機能が有効でも、ジョブ終了までのファイルの書出しは保証されます。

エンドユーザはジョブ投入時に、非同期クローズ機能を指定できます。

参照

- 計算ノード内キャッシュの詳細は、"2.5 計算ノード内キャッシュ機能"を参照してください。
- 使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

図2.7 非同期クローズ機能の動作



非同期クローズは、クローズと同時に計算ノード内キャッシュから、第1階層ストレージ、および第2階層ストレージへの書出しが始まります。クローズ終了の時点で、第1階層ストレージおよび第2階層ストレージへの書出しを保証しません。書出しは、次の計算フェーズ内で終了する場合と終了しない場合があります。ただし、非同期クローズ機能が有効でもジョブ終了時のファイルの書出しは保証します。

同期クローズは、クローズ終了の時点で、第1階層および第2階層ストレージへの書出し終了を保証します。

注意

- 非同期クローズが有効なとき、ファイルをO_WRONLYまたはO_RDWRでオープンし、当該ファイルをクローズした直後に同ファイルに対してexecve(2)を実行した場合、execve(2)がETXTBSYで失敗することがあります。このようなファイルアクセスを行う場合は、非同期クローズを無効にしてジョブを実行するようにしてください。
- 非同期クローズが有効なとき、計算ノード内キャッシュからのファイルデータの書出し失敗はclose(2)で通知されません。第2階層ストレージのキャッシュ領域については、書出し失敗を未書出しファイル一覧取得機能で確認できます("2.6 未書出しファイル一覧取得機能"参照)。共有テンポラリ領域およびノード内テンポラリ領域については、書出し失敗を未書出しファイル一覧取得機能で確認できません。
  共有テンポラリ領域またはノード内テンポラリ領域で、領域のサイズを超える書出しをした場合、ENOSPCエラーでの書出し失敗を確認できないことがあります。このため、領域のサイズは、ジョブが格納するファイルデータのサイズを見積もり、十分な値を指定してください。

2.5 計算ノード内キャッシュ機能

2.5.1 計算ノード内キャッシュのメモリサイズ

LLIOは、ジョブが利用できるメモリ資源量の中から計算ノード内キャッシュのメモリサイズを指定する機能を提供します。ジョブの実行に多くのメモリを使用しないアプリケーションなどは、アプリケーションで使用していないメモリ資源量を計算ノード内キャッシュとして使用することで高速なI/Oを実現します。

エンドユーザはジョブ投入時に、計算ノード内キャッシュに割り当てるメモリサイズを指定できます。

参照

使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

2.5.2 read時における計算ノード内キャッシュへのキャッシュ機能

LLIOは、階層化ストレージから計算ノードに読み込んだファイルを計算ノード内キャッシュにキャッシュする機能を提供します。この機能は、階層化ストレージから読み込んだファイルを計算ノード内キャッシュにキャッシュすることで、アプリケーションが同じファイルを複数回読み込む場合の性能向上を実現します。

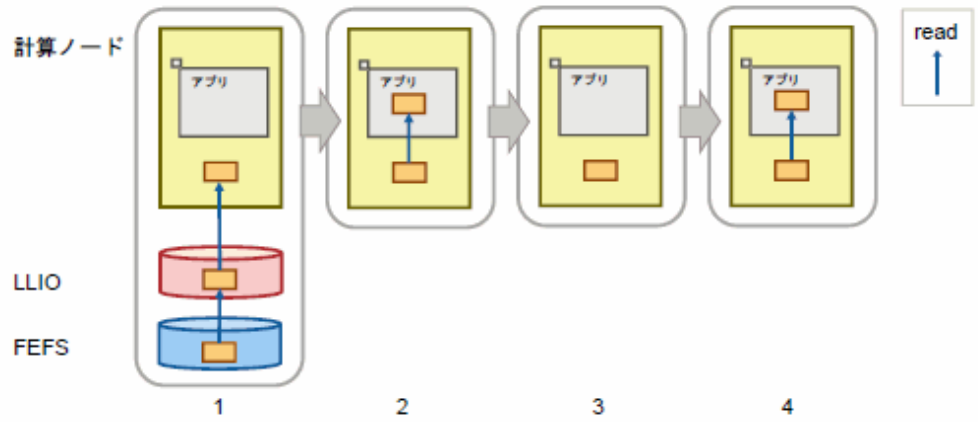
エンドユーザはジョブ投入時に、read時における計算ノード内キャッシュへのキャッシュ機能を指定できます。

参照

使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

以下に、計算ノード内キャッシュにキャッシュする機能の動作を示します。

図2.8 計算ノード内キャッシュにキャッシュする機能の動作



1. 第2階層ストレージから第1階層ストレージに読み込まれたファイルは、計算ノード内キャッシュにキャッシュされます。
2. 計算ノード内キャッシュに読み込まれたファイルは、アプリケーションバッファに読み込まれます。
3. 第1階層ストレージから計算ノードに読み込まれたファイルは、計算ノード内キャッシュにキャッシュされています。
4. アプリケーションが再び同じファイルを読み込む場合、計算ノード内キャッシュのファイルを読み込みます。

2.5.3 計算ノード内キャッシュへの自動先読み機能

LLIOは、ジョブが階層化ストレージ内の連続した領域を読み込もうとした場合、自動的に計算ノード内キャッシュに先読みするかを指定する自動先読み機能を提供します。この機能はファイルを計算ノード内キャッシュに先読みし、読込み処理の高速化を実現する機能です。エンドユーザはジョブ投入時に、計算ノード内キャッシュへの自動先読み機能を指定できます。

参照

使用方法の詳細は、「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

2.5.4 計算ノード内キャッシュの使用有無を切り替えるwriteサイズの閾値

LLIOは、計算ノード内キャッシュの使用有無を切り替えるwriteサイズの閾値を指定する機能を提供します。この機能は、アプリケーションから第1階層ストレージへの書込みファイルを一時的に計算ノード内キャッシュにキャッシュし、計算ノードとストレージI/Oノード間の最大転送サイズの単位でまとめて転送することで、計算ノードから第1階層ストレージへの高速転送を実現する機能です。

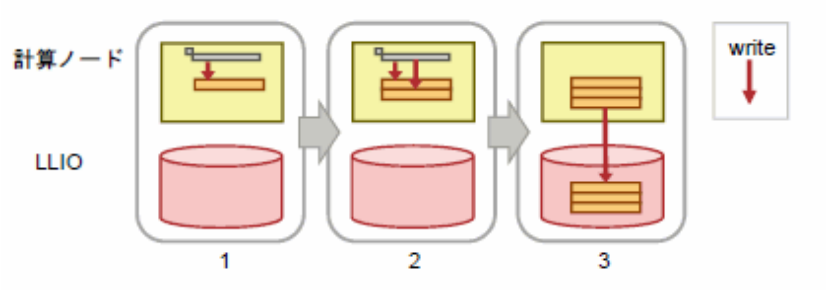
エンドユーザはジョブ投入時に、計算ノード内キャッシュの使用有無を切り替えるwriteサイズの閾値を指定できます。

参照

使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

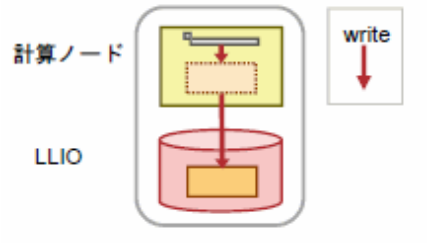
以下に、計算ノード内キャッシュの使用有無を切り替えるwriteサイズの閾値を指定する機能の動作を示します。

図2.9 writeサイズが指定値以下の場合の動作



1. アプリケーションから第1階層ストレージへ書き出すデータサイズが指定値以下の場合、データは計算ノード内キャッシュにキャッシュされます。
2. アプリケーションから第1階層ストレージへ書き出すデータサイズの合計が指定値以下の場合、引き続きデータは計算ノード内キャッシュにキャッシュされます。
3. キャッシュデータはfsyncシステムコールの呼び出しやOSの非同期書出し処理に従って第一階層ストレージへ書き出されます。また書き出すキャッシュデータの合計値が計算ノード内キャッシュのサイズより大きくなる場合は同期的な書出しが行われます。

図2.10 writeサイズが指定値を超える場合の動作



アプリケーションから第1階層ストレージへ書き出すデータサイズが指定値を超える場合、データは計算ノード内キャッシュに書き出されず、アプリケーションと同期して第1階層ストレージに書き出されます。計算ノード内キャッシュに第1階層ストレージへの書出しが必要なファイルがある場合は、計算ノード内キャッシュ内のデータは先に第1階層ストレージに書き出されます。

2.6 未書出しファイル一覧取得機能

ジョブが実行中の計算ノードに異常が発生した場合、または、書出し処理中にジョブがジョブの経過時間制限に達した場合なども、計算ノード内キャッシュから第1階層ストレージ、および第1階層ストレージから第2階層ストレージへの書出しが完了できないファイルが残ることがあります。

LLIOは、書出しが完了しなかったファイルの一覧である未書出しファイル一覧を出力する機能を提供します。エンドユーザは、この一覧を分析することで、出力が完了しなかったファイルを知るだけではなく、次に実行するジョブの実行可能時間を調整するための情報を得ることができます。

エンドユーザはジョブ投入時に、未書出しファイル一覧取得機能を指定できます。

参照

使用方法の詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

2.7 統計情報

LLIOは、エンドユーザがアプリケーションの第1階層ストレージへのファイルアクセスを把握するための手段として、3種類の統計情報を提供します。

ジョブ統計情報

ジョブの投入条件や、第1階層ストレージのアクセス状況を確認できます。

LLIO性能情報

第1階層ストレージへのメタアクセス情報や入出力情報を、ジョブ統計情報よりも詳細に確認できます。

計算ノード統計情報

ジョブ内でライブラリを使用することで、計算ノードごとの第1階層ストレージへのメタアクセス情報や入出力情報を確認できます。

2.7.1 ジョブ統計情報

LLIOは、ジョブ運用ソフトウェアと連携しジョブ統計情報の一部にLLIOの統計情報を出力します。エンドユーザは、ジョブ投入コマンドでジョブの出力結果として、ジョブ統計情報を得ることができ、それを参照することでジョブの実行の様子を後から分析できます。

参照

- ジョブの出力結果としてジョブ統計情報を得る方法については、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」の"ジョブの操作方法"にある"ジョブ統計情報出力の指定"を参照してください。
- 出力されるジョブ統計情報の詳細は、ジョブ運用ソフトウェアのmanマニュアルpjstatsinfo(7)を参照してください。

2.7.2 LLIO性能情報

LLIOは、ジョブ運用ソフトウェアと連携しLLIO性能情報を提供します。この機能は、計算クラスタ管理ノードに集約したストレージI/Oノードの統計情報を、ジョブ運用ソフトウェアがストレージI/Oノード単位に集計しジョブ終了時にLLIO性能情報としてファイルに出力する機能です。

エンドユーザはジョブ投入時に指定することで、LLIO性能情報をジョブの出力結果として得ることができます。それを参照することで、自分のジョブに関する第1階層ストレージへのアクセス状況を後から分析できます。また、管理者は、ジョブ運用ソフトウェアのログファイルllioinfoから、ジョブごとのLLIO性能情報を参照できます。

参照

- ジョブの出力結果としてLLIO性能情報を得る方法については、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」の"ジョブの操作方法"にある"LLIO性能情報の採取"を参照してください。

・ 出力されるLLIO性能情報の詳細は、"C.1 LLIO性能情報の出力項目"を参照してください。

注意

LLIO 性能情報の採取はジョブが RUNOUT 状態の時に実施しています。

LLIO 性能情報は、ジョブを実行したノード数が多くなるほどその採取に時間がかかり、これにともなってジョブがRUNOUT 状態となっている時間も増えていきます。

1 万ノード程度であれば数分内で完了しますが、2 万ノードで 30 分、5 万ノードで 2 時間程度かかる場合があります。

2.7.3 計算ノード統計情報

LLIOは、計算ノードから第1階層ストレージへのI/O情報を取得するライブラリ関数を提供します。エンドユーザは、計算ノード統計情報が提供するライブラリ関数を使用することで、その計算ノードに関するI/Oの状況を確認できます。

参照

計算ノード統計情報の採取方法と、採取項目の詳細は、”C.2 計算ノード統計情報の採取方法と出力項目”を参照してください。

参考

取得できる統計情報は、計算ノード統計情報取得を実行したノードの情報だけです。

2.8 システム管理者向けの機能

2.8.1 LLIO状態確認

LLIOは、ジョブ運用ソフトウェアと連携しLLIOが動作する計算ノード、ストレージI/Oノード、グローバルI/Oノードの状態監視をする機能を提供します。システム管理者は、状態監視機能コマンドであるpashowclstコマンドを使用して各ノードの状態確認ができます。

参照

LLIOに関するサービスの状態監視方法の詳細は、”3.2.1 LLIOの状態監視” を参照してください。

2.8.2 システム統計情報

LLIOは、システム管理者がファイルへのI/O状況を把握し、運用状況の監視やトラブル調査、チューニングをするための手段として、システム統計情報を提供します。システム統計情報は、オープンソースである collectlのサービスを自動起動するように設定することで、ストレージI/Oノード、グローバルI/Oノードの統計情報を得ることができます。

参照

・ システム統計情報の採取方法の詳細は、”3.2.3 システム統計情報の採取”を参照してください。

・ システム統計情報の採取項目の詳細は、”C.3 システム統計情報の出力項目”を参照してください。

第3章 システム運用

ここではLLIOのシステム運用について説明します。システム管理者が対象です。

3.1 導入

LLIOの導入は、FEFSの導入と同時に行います。

注意

FEFSの導入の詳細は、マニュアル「FEFSユーザーズガイド」を参照してください。

LLIOの導入は、以下の順に行います。

1. LLIO構成の設計
2. LLIOパッケージとcollectlパッケージの適用
3. FEFSデザインシートの作成
4. LLIOの構築
5. LLIOの状態確認
6. ジョブACL機能の設定
7. ジョブ運用ソフトウェアとの連携のための設定

注意

- collectlパッケージは別途入手して適用します。システム統計情報の採取をしない場合は、インストールする必要はありません。
- collectlはバージョン4.3.0のみサポートしています。

3.1.1 LLIO構成の設計

LLIOの構成について、以下の観点で具体的に設計します。

- SSDの決定
  - SSDの容量
  - SSDデバイスのパス名
- 構成の決定
  - ストレージI/Oノードの台数
  - 計算ノードの台数
  - ストレージI/Oノードと計算ノードの構成

3.1.2 LLIOパッケージの適用

LLIOのパッケージを以下に示します。

LLIO本体パッケージ

1. FJSVllio-*.aarch64.rpm
2. FJSVllio-modules-*.aarch64.rpm

3. FJSVllio-stats-*.aarch64.rpm

※*には版数とリリース名が入ります。

パッケージと適用ノードの関係を以下に示します。

表3.1 LLIO本体パッケージと適用ノード

パッケージ名	ノード種別			
	SIO	GIO	BIO	CN
FJSVllio	○	○	○	○
FJSVllio-modules	○	○	○	○
FJSVllio-stats	○	○	○	○

LLIO周辺パッケージ

1. FJSVllio-*.x86_64.rpm
2. FJSVllio-stats-*.x86_64.rpm

※*には版数とリリース名が入ります。

パッケージと適用ノードの関係を以下に示します。

表3.2 LLIO周辺パッケージと適用ノード

パッケージ名	ノード種別		
	CCM	LN	多目的※1
FJSVllio	○	○	○
FJSVllio-stats		○	

※1:多目的ノードを統計情報収集ノードとして使用する場合のみ適用してください。多目的ノード以外を統計情報収集ノードとして使用する場合は、そのノードに適用してください。

参考

統計情報収集ノードに関する詳細は、"複数ストレージI/Oノードのシステム統計情報の出力方法"を参照してください。

注意

collectlパッケージは別途入手し、以下のノードに適用します。

表3.3 collectlパッケージと適用ノード

パッケージ名	ノード種別		
	SIO	GIO	多目的※1
collectl	○	○	○

※1:多目的ノードを統計情報収集ノードとして使用する場合のみ適用してください。多目的ノード以外を統計情報収集ノードとして使用する場合は、そのノードに適用してください。

3.1.3 FEFSデザインシートの作成

LLIOの構成は、FEFSのデザインシートの「LLIOシート」に記入します。FEFSデザインシートはWindows端末で作成します。FEFSデザインシートのひな形は、FEFS製品に同梱されています。FEFSデザインシートのファイル名は、"FEFSDesignSheet.xlsm"です。FEFSデザインシート作成作業を始めるときは、最初にExcelのマクロを有効にしてください。なお、セルの色が赤の入力項目は設定が必須の項目です。必ず値を入力してください。

注意

サポートするWindowsとExcelのバージョンは、Windows 8、Windows 10 上のExcel 2010、および2013です。これ以外の環境については、担当保守員(SE)または当社Support Deskに相談してください。

LLIOシートには、LLIOのマウント情報およびストレージI/Oノード、第1階層ストレージデバイスの設定をします。

1. LLIO SETTINGセクション

図3.1 LLIO SETTINGセクションの記入例

■ LLIO SETTING	
FUNCTIONS	**USE / UNUSE**
Shared temporary	USE
Local temporary	USE
Global	USE

LLIOを利用する場合はすべての項目にUSEを指定してください。

LLIOを利用しない場合は初期値のままUNUSEとし、以降のFEFSデザインシートでのLLIOの設定は不要です。

2. LLIOセクション
以下の"LLIOセクションの記入例"に沿って説明します。

図3.2 LLIOセクションの記入例

■ LLIO			
FUNCTIONS	**MOUNT POINT**	**MOUNT OPTION**	
Shared temporary	/share		Configured
Local temporary	/local		Configured
Global	* Same as "MOUNT POINT [FEFS]" in GFS sheet *		Configured

   a. MOUNT POINT [Shared temporary]
   ジョブ内共有テンポラリ用のマウントポイントを指定します。

   b. MOUNT OPTION [Shared temporary]
   ジョブ内共有テンポラリ用のマウントオプションを指定します。通常は変更する必要はありません。

   c. MOUNT POINT[Local temporary]
   ノード内テンポラリ用のマウントポイントを指定します。

   d. MOUNT OPTION[Local temporary]
   ノード内テンポラリ用のマウントオプションを指定します。通常は変更する必要はありません。

   e. MOUNT OPTION [Global]
   ジョブ内第2階層ストレージのキャッシュのマウントオプションを指定します。通常は変更する必要はありません。

マウントポイントなしにマウントオプションを定義することはできません。定義が適切に行われている場合は、入力欄の右に「Configured」が入力されます。

3. SIOセクション
以下の"SIOセクションの記入例"に沿って説明します。

図3.3 SIOセクションの記入例

■ SIO				
SIO HOSTNAME	**SSD VOLUME**	**SSD SIZE(Mib)**	**MKFS OPTION**	**MOUNT OPTION**
sio1	/dev/nvme0n1	819200	-f	pquota
sio2	/dev/nvme0n1			
sio3	/dev/nvme0n1			

a. SIO HOSTNAME
ストレージI/Oノードのホスト名を記述します。

b. SSD VOLUME
当該ストレージI/Oノードに接続された第1階層ストレージデバイスのパスを定義します。

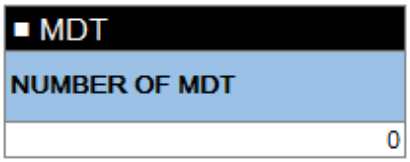
c. SSD SIZE(Mib)
第1階層ストレージが利用するデバイスサイズをMib単位で指定します。

d. MKFS OPTION
第1階層ストレージのmkfsオプションを指定します。通常は変更する必要はありません。

e. MOUNT OPTION
第1階層ストレージデバイスのマウントオプションを指定します。通常は変更する必要はありません。

4. MDT セクション
以下の "MDT セクションの記入例" に沿って説明します。

図3.4 MDTセクションの記入例



a. NUMBER OF MDT
共有テンポラリ領域向けに専用の MDT を用意している場合、専用の MDT 数を指定してください。
共有テンポラリ領域向けに専用の MDT を用意していない場合は、0 を指定してください。(デフォルト0)

参考

本項は共有テンポラリ領域が使用するFEFS MDTの数を指定します。指定する場合、共有テンポラリ領域が使用するFEFS MDTにはインデックス番号の大きなものから割り当てられます。システム管理者がFEFSのリモートディレクトリおよびストライプディレクトリの機能を利用して、ホームディレクトリなどで使用するFEFS MDTを共有テンポラリ領域が使用するFEFS MDT以外に設定することにより、共有テンポラリ領域へのファイルアクセスと、第2階層ストレージのキャッシュ領域およびFEFSへのファイルアクセスで使用するFEFS MDTを分離できます。

3.1.4 LLIOの構築

FEFSデザインシートの作成以降の以下の作業はFEFSの構築と同時に行います。

- FEFSセットアップツール用構成定義ファイルの作成
- FEFSセットアップツール用構成定義ファイルの配置
- FEFSの構築

参照

FEFSの構築の詳細は、マニュアル「FEFSユーザーズガイド」を参照してください。

3.1.5 LLIOの状態確認

"3.1.4 LLIOの構築"のFEFSの構築で、FEFSサービスを起動するとLLIOサービスは自動的に起動されます。LLIOのサービスがストレージI/Oノードと計算ノードで正常に起動されたことを、運用系システム管理ノードでpashowclstコマンドを実行して確認します。

```
[システム管理ノード]
# pashowclst -v --nodetype CN
```

LLIOの状態がFEFSSR(o)およびFEFS(o)に遷移していれば、LLIOのサービスは正常に起動されています。

サービスの状態の詳細は、"3.2.1 LLIOの状態監視"を参照してください。

注意

第2階層ストレージのキャッシュ領域、共有テンポラリ領域、ノード内テンポラリ領域のマウントはジョブ投入時に行われます。サービス開始時には行われません。

3.1.6 ジョブACL機能の設定

ジョブACLの設定をpmjacladmコマンドで行います。ジョブACL機能はジョブ運用ソフトウェアの機能で、ジョブに対して指定できる資源量の上限やジョブ運用に関するコマンドの利用権限などを制御します。

参照

ジョブACL機能の概念や機能全般に関する詳細は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド ジョブ管理編」を参照してください。

3.1.7 ジョブ運用ソフトウェアとの連携のための設定

ジョブ運用ソフトウェアと連携するために「フック」の設定をします。フックとは、ジョブ運用ソフトウェアの機能で、ジョブ実行処理の中で管理者が任意の処理を組み込んで実行させることができます。LLIOはフックの「ジョブマネージャー出口機能」を使ってジョブがLLIOを使用するための準備や後処理を行います。

参照

フックの詳細は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド ジョブ運用管理機能フック編」を参照してください。

以下にジョブマネージャー出口機能の設定方法について説明します。

ジョブマネージャー出口機能の設定

システム管理ノードのリソースユニットごとの /etc/opt/FJSVtcs/Rscunit.d/*リソースユニット名*/pmpjm.confファイルのExitFuncサブセクションに、LLIOの出口関数ライブラリlibllliohook.soを設定します。

```
ResourceUnit {
    ResourceUnitName = リソースユニット名
        ...
        ExitFunc {                          ← リソースユニットに対する設定
            ExitFuncLib = libllliohook.so   ← 出口関数ライブラリの指定
            ExitFuncPri = 200               ← 出口関数の実行優先度
            ExitFuncType = pjm              ← 出口関数の種類
        }
}
```

項目のExitFuncLibとExitFuncTypeは、上記のとおりに記述してください。ExitFuncPriは、特に条件はありません。ほかの出口関数ライブラリの実行優先度と重複した場合には、記載した順に実行されます。

LLIOの出口関数ライブラリをジョブ運用ソフトウェアに組み込むためにpmpjmadmコマンドを実行します。

```
[システム管理ノード]
# pmpjmadm -c クラスタ名 --set --rscunit リソースユニット名
```

3.2 運用

3.2.1 LLIOの状態監視

システム管理者は、pashowclstコマンドでLLIOの状態監視ができます。状態監視には、FEFSSR監視サービスとFEFS監視サービスの2種類があります。状態監視対象ノードと状態監視項目を以下に示します。

表3.4 状態監視対象ノードと状態監視項目

監視サービス名	状態監視項目	状態監視対象ノード
FEFSSR	LLIOのサーバ機能	ストレージI/Oノード
	グローバルI/Oノードの中継機能	グローバルI/Oノード
FEFS	FEFSのクライアント機能	ストレージI/Oノード、グローバルI/Oノード、計算ノード

例) pashowclst コマンドの -n オプションで対象ノードのサービス状態を確認します。

```
[システム管理ノード]
# pashowclst -c clstname -n nodeid
[ CLST: clstname ]
[ NODE: nodeid ]
NODE        NODETYPE  STATUS    REASON    PWR_STATUS     ARCH_STATUS    SRV_STATUS
nodeid      SIO,CN    Running   -         os-running     ICC_Running    PLE(o),NRD(o),FEFSSR(o),FEFS(o),PWRD(o)
```

状態監視対象ノードで出力されるサービスの状態の意味を以下に示します。

表3.5 ストレージI/OノードにおけるFEFSSRの状態と意味

サービス状態	意味
o	監視項目がすべて正常
!	ネットワークで縮退または異常が発生
x	監視項目のいずれかで異常
s	起動後に設定。各状態が正常になるまで状態を遷移しない
b	FEFSおよびLLIOが未設定状態

表3.6 グローバルI/OノードにおけるFEFSSRの状態と意味

サービス状態	意味
o	監視項目がすべて正常
x	FEFSサービスの停止、または異常
!	ネットワークで縮退または異常が発生
s	起動後に設定。各状態が正常になるまで状態を遷移しない
b	FEFSおよびLLIOが未設定状態

表3.7 ストレージI/Oノード、グローバルI/Oノードおよび計算ノードにおけるFEFSの状態と意味

サービス状態	意味
o	監視項目がすべて正常
x	監視項目のいずれかで異常
!	ネットワークで縮退または異常が発生
s	起動後に設定。各状態が正常になるまで状態を遷移しない
a	ストレージI/Oノード、またはグローバルI/OノードのFEFSで異常があり使用できない
b	FEFSおよびLLIOが未設定状態

参照

pashowclstコマンドの表示については、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド システム管理編」を参照してください。

3.2.2 ジョブACL機能の設定の変更

運用中、ジョブACL機能の設定を変更する場合はpmjacladmコマンドを使用します。ジョブACL機能はジョブ運用ソフトウェアの機能で、ジョブに対して指定できる資源量の上限やジョブ運用に関するコマンドの利用権限などを制御します。

参照

ジョブACL機能の概念や機能全般に関する詳細は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド ジョブ管理編」を参照してください。

3.2.3 システム統計情報の採取

3.2.3.1 ストレージI/Oノードのシステム統計情報の採取

ストレージI/Oノードのシステム統計情報は、collectl用のプラグインであるlliosv.phで採取します。collectlのサービスを自動起動するように設定することで自動で定期的に採取できます。

ストレージI/Oノードのシステム統計情報の採取方法

ストレージI/Oノードのシステム統計情報を採取するためには、システム管理者は以下の手順でcollectl設定ファイル(/etc/collectl.conf)を設定します。

1. collectl.confファイルを作成し、collectlデーモン起動オプション設定行DaemonCommandsに以下を記述します。

```
[システム管理ノード]
# vi collectl.conf
DaemonCommands = -i <systemstat_interval> -f <systemstat_dir> -r<time>,<systemstat_rolllogs> -m -F<flush_time> (※)
-s C --import /opt/FJSVllio/bin/lliosv.ph
```

備考) 紙面の都合で、上記表示例は(※)の箇所で改行しています。実際には、1行として表示されます。

collectl設定ファイルのcollectl デーモン起動オプション設定行DaemonCommandsの設定項目を以下に示します。

表3.8 DaemonCommandsの設定項目

設定項目	説明
-i <*systemstat_interval*>	<*systemstat_interval*>に統計情報の採取間隔を秒単位で指定します。 推奨値は600です。 指定しなかった場合のデフォルト値はcollectlに従います。
-f <*systemstat_dir*>	<*systemstat_dir*>に統計情報を出力するディレクトリを指定します。 推奨ディレクトリは/var/opt/FJSVllio/lliostat/です。 指定しなかった場合のデフォルト値はcollectlに従います。
-r<*time*>,<*systemstat_rolllogs*>	<*time*>にログローテートする時刻を、<*systemstat_rolllogs*>に蓄積する日数を指定します。 時刻は*hh:mm* 形式、日数は数値で指定します。 ログファイルは1日ごとに切り替わり、指定日数を超えた場合は、古いファイルから順に削除します。 <*systemstat_rolllogs*>の推奨値は10 です。 指定しなかった場合のデフォルト値はcollectl に従います。
-m	-f オプションで指定したディレクトリにcollectlの動作ログを作成します。
-F<*flush_time*>	<*flush_time*>に統計情報のデータをメモリ上からディスクに書出す間隔を秒単位で指定します。

設定項目	説明
	0を指定するとデータを採取するたびにディスクに書き出します。 指定しなかった場合のデフォルト値はcollectlに従います。
-s C	各CPUコアの負荷を採取します。

2. 編集したcollectl.confファイルをすべてのストレージI/Oノードに配布します。

```
[システム管理ノード]
# pmscatter -c clstname --nodetype SIO ./collectl.conf /etc/collectl.conf
```

3. collectlサービスを自動起動させます。

```
[システム管理ノード]
# pmexe -c clstname --nodetype SIO "/usr/bin/systemctl enable collectl"
# pmexe -c clstname --nodetype SIO "/usr/bin/systemctl start collectl"
```

4. 運用中に設定を変更する場合は、以下のようにcollectlサービスを再起動します。

```
[システム管理ノード]
# pmexe -c clstname --nodetype SIO "/usr/bin/systemctl restart collectl"
```

ストレージI/Oノードのシステム統計情報の出力方法

ストレージI/Oノードのシステム統計情報を出力するためには、collectl で採取したログファイルを使用します。

以下にストレージI/Oノードのシステム統計情報を出力する方法を示します。

```
[ストレージI/Oノード]
# collectl -p <data file> -oD -s-C --plot [<collectl option>] --import /opt/FJSVllio/bin/lliosv.ph [,from=<value>] (※)
[,stat=<value>] [,jobid=<ジョブID>] [,kind=<value>] [,type=<value>] [,section=<value>]
```

備考) 紙面の都合で、上記表示例は(※)の箇所で改行しています。実際には、1行として表示されます。
<*data file*>には、ストレージI/Oノードで採取したcollectlのログファイルを指定します。

collectlコマンドのオプションには、出力項目のセパレータや出力する時間範囲などが指定できます。

ストレージI/Oノードのシステム統計情報では、以下のcollectl拡張プラグインのオプションが指定できます。

参照

ジョブIDの詳細はマニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」を参照してください。

表3.9 ストレージI/Oノードのcollectl拡張プラグイン

オプション	説明
from=<*value*>	出力する統計情報を日時で絞り込みます。 <*value*>には以下を指定できます。 yyyymmdd:hh:mm:ss-yyyymmdd:hh:mm:ss
stat=<*value*>	出力する統計情報を統計情報種別で絞り込みます。 <*value*>には以下を指定できます。 j:ジョブごとのI/O情報 c:通信層のコネクション数に関する情報 r:リクエスト処理に関する情報
jobid=<*ジョブID*>	出力する統計情報をジョブIDで絞り込みます。
kind=<*value*>[/<*value*>]	ジョブごとのI/O 情報の種類による出力情報の絞り込みを行います。 本オプションで指定した統計情報のみを出力します。"/"で複数のI/Oの情報の種類が指定できます。本オプションを指定しない場合はすべての情報を出力します。 <*value*>には以下を指定できます。

オプション	説明
	io：I/O 情報 meta：メタアクセス情報 rsc：リソース情報 async：非同期転送情報
type=<*value*>[/<*value*>]	利用形態の種類によるジョブごとのI/O情報の絞り込みを行います。 本オプションで指定した統計情報のみを出力します。"/"で複数の利用形態が指定できます。本オプションを指定しない場合はすべての情報を出力します。 <*value*>には以下を指定できます。 local：ノード内テンポラリ領域の情報 share：共有テンポラリ領域の情報 cache：第2 階層ストレージのキャッシュの情報
section=<*value*>[/<*value*>]	I/O箇所の種類によるジョブごとのI/O情報の絞り込みを行います。 本オプションで指定した統計情報のみを出力します。"/"区切りで複数のI/O箇所が指定できます。本オプションを指定しない場合はすべての情報を出力します。 <*value*>には以下を指定できます。 Total：全体のI/O情報 CN-CBCN：計算ノードと通信用バッファ間のI/O情報 CBCN-Dev：通信用バッファと第1階層ストレージデバイス間の転送のI/O情報(計算ノードとの通信) CBGFS-Dev：通信用バッファと第1階層ストレージデバイス間の転送のI/O情報(第2階層ストレージとの通信) CBCN-GFS：通信用バッファと第2階層ストレージ間の転送のI/O情報(Direct I/O) CBGFS-GFS：通信用バッファと第2階層ストレージ間の転送のI/O情報(Cached I/O)

以下に出力例を示します。

```
#Date Time Hostname j JOBID Cache-Total-WR(4K-)-Count Cache-Total-WR(1M-)-Count Cache-Total-WR(4M-)-Count …<略>
#Date Time Hostname c Conn InProConn InProDisconn MaxConn AccumConn AccumDisconn
#Date Time Hostname r JOBID IO-Wait-Num IO-Wait-Min IO-Wait-Max IO-Wait-Sum IO-Qdepth-Num IO-Qdepth-Min …<略>
20181113 11:40:00 sio-node1 j 744 0 0 0 0 0 0 0 0 0 0 0 …<略>
20181113 11:40:00 sio-node1 c 1 0 0 0 1 0
20181113 11:40:00 sio-node1 r 744 4 3025 7549 17256 4 0 0 0 4 1 1 4 47 2779 5587 152640 47 0 0 0 47 1 1 47
20181113 11:50:00 sio-node1 j 744 0 20 0 0 0 1310720 0 0 0 1310934 0 0 …<略>
20181113 11:50:00 sio-node1 r 744 39 -422 0 137259 39 0 0 0 39 0 0 39 2 0 0 7070 2 0 0 0 2 0 0 2
(以下略)
```

1行目から3行目にヘッダー情報(行頭の "#" の後に項目名)を出力し、4行目以降に情報を採取するタイミングごとの統計情報を出力します。

参照

各出力項目の詳細は、"C.3.1 ストレージI/Oノード向けシステム統計情報の出力項目"を参照してください。

複数ストレージI/Oノードのシステム統計情報の出力方法

収集された複数のストレージI/Oノードのシステム統計情報のログファイルから、特定のジョブのシステム統計情報を抽出するためにログファイルを1つずつcollectlコマンドで参照すると時間がかかります。showsiostatsコマンドは、1つのノードに複数のストレージI/Oノードのシステム統計情報を収集することで、システム統計情報の中の特定のジョブのシステム統計情報をまとめて出力できます。以降では、システム統計情報を収集するノードを"統計情報収集ノード"と表記します。

参照

showsiostatsコマンドの詳細は、"A.2.4 showsiostatsコマンド"を参照してください。

複数のストレージI/Oノードのシステム統計情報の収集方法として、ログファイルを直接収集する方法と、ノードを中継して収集する方法の2種類があります。ノードを中継して収集する方法は、大規模構成のシステムに対し負荷分散が見込めます。収集方法や統計情報収集ノード、

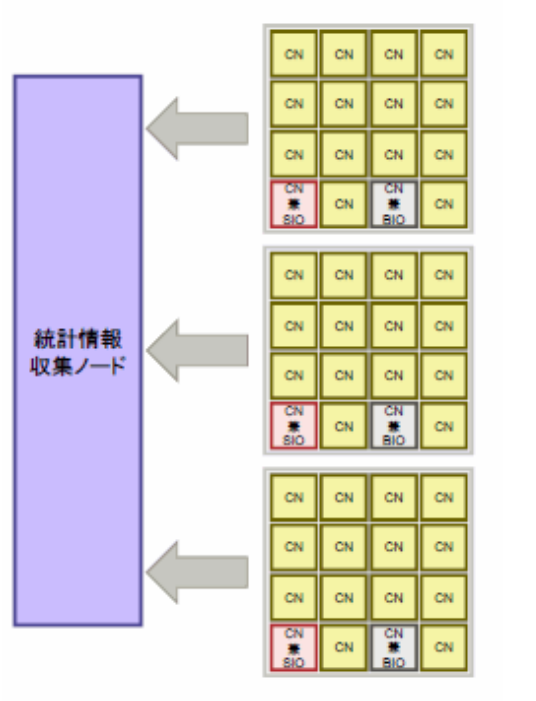
中継ノードの選択については、システム管理者がシステム環境に合わせて検討します。システム環境内にログ収集ノードがある場合は、ログ収集ノードと統計情報収集ノードを同じにすると、効率よくログもしくは統計情報が確認できます。

注意

以下に示す方法は、cronコマンドやrsyncコマンドを利用した方法です。転送ノード間のrsyncコマンドをパスフレーズなしで実行できることを前提としています。

1. ログファイルを直接収集する方法

図3.5 ログファイルを直接収集する



1. 統計情報収集ノードの任意の場所に、収集対象となるストレージI/Oノードの数だけcollectlで採取したログファイルの収集用ディレクトリを作成します。

```
[統計情報収集ノード]
# mkdir -p /var/opt/FJSVllio/rsync/<ストレージI/Oノード名>
```

2. 収集対象となるストレージI/Oノードの数だけrsyncコマンドのコマンドラインを作成します。

```
rsync -auv --delete --log-file=<任意ログファイル名> (※)
<BIO IP>:/export/nfsroot/cnode<xxx>/var/opt/FJSVllio/lliostat /var/opt/FJSVllio/rsync/<ストレージI/Oノード名>
```

備考)紙面の都合で、上記表示例は(※)の箇所で改行しています。実際には、1行として表示されます。

<*BIO IP*>には、pashowclstコマンドで表示されるMNG_NETのIPアドレスを指定します。

```
[システム管理ノード]
# pashowclst -v -l --nodetype BIO | grep "¥." |  awk '{print $3}'
```

<*xxx*>には、pashowclstコマンドで表示されるCTRL_NETの()の番号を指定します。

```
[システム管理ノード]
# pashowclst -v -l --nodetype SIO | grep "¥." | awk -F'(' '{print $2}' | awk -F')' '{print sprintf("cnode%03d",
$1);}'
```

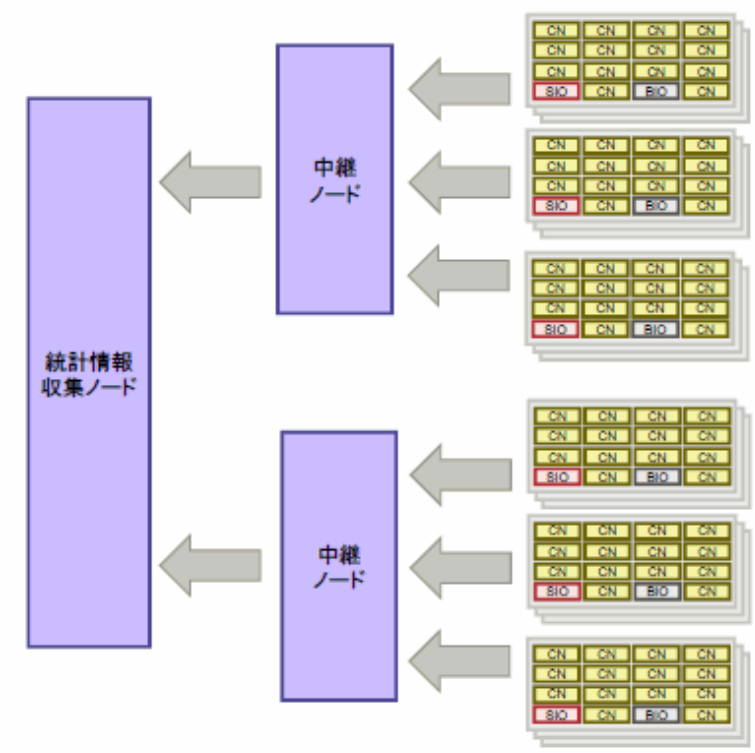
3. 統計情報収集ノードのcronを設定します。

```
[統計情報収集ノード]
# crontab -e
```

- 並列実行する場合
  crontabコマンドを使用して、手順2で作成したコマンドラインを設定します。
- 逐次実行する場合
  手順2で作成したコマンドラインでスクリプトを作成し、crontabコマンドにスクリプトを設定します。

2. ログファイルをノードを中継して収集する方法

図3.6 ログファイルをノードを中継して収集する



1. 統計情報収集ノードの任意の場所に、collectlで採取したログファイルの収集用ディレクトリを作成します。

```
[統計情報収集ノード]
# mkdir -p /var/opt/FJSVllio/rsync
```

2. 中継ノードの任意の場所に、収集対象となるストレージI/Oノードの数だけcollectlで採取したログファイルの収集用ディレクトリを作成します。

```
[中継ノード]
# mkdir -p /var/opt/FJSVllio/rsync/<ストレージI/Oノード名>
```

3. 中継ノードと収集対象となるストレージI/Oノード間のrsyncコマンドのコマンドラインを作成します。

```
rsync -auv --delete --log-file=<任意ログファイル名> (※)
<BIO IP>:/export/nfsroot/cnode<xxx>/var/opt/FJSVllio/lliostat /var/opt/FJSVllio/rsync/<ストレージI/Oノード名>
```

備考) 紙面の都合で、上記表示例は(※)の箇所で改行しています。実際には、1行として表示されます。

<*BIO IP*>には、pashowclstコマンドで表示されるMNG_NETのIPアドレスを指定します。

```
[システム管理ノード]
# pashowclst -v -l --nodetype BIO | grep "¥." |  awk '{print $3}'
```

<*xxx*>には、pashowclstコマンドで表示されるCTRL_NETの()の番号を指定します。

```
[システム管理ノード]
# pashowclst -v -l --nodetype SIO | grep "¥." | awk -F'(' '{print $2}'| awk -F')' '{print sprintf("cnode%03d", $1);}'
```

4. 統計情報収集ノードと中継ノード間のrsyncコマンドのコマンドラインを作成します。

```
rsync -auv --delete --log-file=<任意ログファイル名> /var/opt/FJSVllio/rsync/<ストレージI/Oノード名> <統計情報収集ノードIP>:/var/opt/FJSVllio/rsync
```

5. 手順3、4の順番にコマンドラインを記述し、スクリプトを作成します。

```
rsync -auv --delete --log-file=<任意ログファイル名> (※)
<BIO IP>:/export/nfsroot/<cnode>xxx/var/opt/FJSVllio/lliostat /var/opt/FJSVllio/rsync/<ストレージI/Oノード名>
rsync -auv --delete --log-file=<任意ログファイル名> /var/opt/FJSVllio/rsync/<ストレージ I/O ノード名> (※)
<統計情報収集ノード IP>:/var/opt/FJSVllio/rsync
```

備考) 紙面の都合で、上記表示例は(※)の箇所で改行しています。実際には、1行として表示されます。

6. 中継ノードのcronに手順5のスクリプトを設定します。

```
[中継ノード]
# crontab -e
```

3.2.3.2 グローバルI/Oノードのシステム統計情報の採取

グローバルI/Oノードのシステム統計情報は、collectl用のプラグインであるlnetsv.phで採取します。collectlのサービスを自動起動するように設定することで自動で定期的に採取できます。

グローバルI/Oノードのシステム統計情報の採取方法

グローバルI/Oノードのシステム統計情報を採取するためには、システム管理者は以下の手順でcollectl設定ファイル(/etc/collectl.conf)を設定します。

1. collectl.confファイルを作成し、collectlデーモン起動オプション設定行DaemonCommandsに以下を記述します。

```
[システム管理ノード]
# vi collectl.conf
DaemonCommands = -i <systemstat_interval> -f <systemstat_dir> -r<time>,<systemstat_rolllogs> -m -F<flush_time> (※)
-s C --import /opt/FJSVllio/bin/lnetsv.ph
```

備考) 紙面の都合で、上記表示例は(※)の箇所で改行しています。実際には、1行として表示されます。

"表3.8 DaemonCommandsの設定項目"を参考に、collectl設定ファイルのcollectlデーモン起動オプション設定行DaemonCommandsを設定します。

2. 編集したcollectl.confファイルをすべてのグローバルI/Oノードに配布します。

```
[システム管理ノード]
# pmscatter -c clstname --nodetype GIO ./collectl.conf /etc/collectl.conf
```

3. collectlサービスを自動起動させます。

```
[システム管理ノード]
# pmexe -c clstname --nodetype GIO "/usr/bin/systemctl enable collectl"
# pmexe -c clstname --nodetype GIO "/usr/bin/systemctl start collectl"
```

4. 運用中に設定を変更する場合は、以下のようにcollectlサービスを再起動します。

```
[システム管理ノード]
# pmexe -c clstname --nodetype GIO "/usr/bin/systemctl restart collectl"
```

グローバルI/Oノードのシステム統計情報の出力方法

グローバルI/Oノードのシステム統計情報を出力するためには、collectlで採取したログファイルを使用します。

以下にグローバルI/Oノードのシステム統計情報を出力する方法を示します。

```
[グローバル I/O ノード]
# collectl -p <data file> -oD -s-C --plot [<collectl option>] --import /opt/FJSVllio/bin/lnetsv.ph[,stat=<value>]
```

<*data file*>には、グローバルI/Oノードで採取したcollectlのログファイルを指定します。

collectlコマンドのオプションには、出力項目のセパレータや出力する時間範囲などが指定できます。

グローバルI/Oノードのシステム統計情報では、以下のcollectl拡張プラグインのオプションが指定できます。

表3.10 グローバルI/Oノードのcollectl拡張プラグイン

オプション	説明
stat=<*value*>	出力する統計情報をデータの種類で絞り込みます。 <*value*>には以下を指定できます。 s:データ転送回数およびデータ量の情報 b:転送バッファの情報

以下に出力例を示します。

```
#Date Time TransferCount TransferAmount MinFreeBufZero MinBufSend MinFreeBufRDMA
20181113 23:05:00 216 108288 2047 1531 96
20181113 23:10:00 284 59072 2048 1531 96
20181113 23:15:00 268 55744 2048 1531 96
(以下略)
```

各出力項目の詳細は、"C.3.2 グローバルI/Oノード向けシステム統計情報の出力項目"を参照してください。

複数グローバルI/Oノードのシステム統計情報の出力方法

1つのノードに複数のグローバルI/Oノードのシステム統計情報を収集する場合は、複数ストレージI/Oノードのシステム統計情報の出力方法を参照の上、<ストレージI/Oノード名>を、<グローバルI/Oノード名>に置き換えて収集してください。

3.3 保守

3.3.1 LLIO構成の変更

SIO構成の変更など、ハードウェアの保守を伴うLLIO構成の変更を実施する場合は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド 保守編」を参照の上、以下の手順に従い作業します。

1. FEFSデザインシートの作成デザインシートの作成
   "3.1.3 FEFSデザインシートの作成"に従い、FEFSデザインシートを作成します。

2. LLIOサービスの停止
   ストレージI/Oノード、計算ノードのLLIOサービスを停止します。これらは運用系システム管理ノードで実施します。

   ```
   [システム管理ノード]
   # fefs_sync --stop --compute=<cluster>[,…] --llio
   ```

   LLIOを構成するすべてのクラスタを指定してください。

3. LLIOの再構築
   "3.1.4 LLIOの構築"に従い、以下の作業を実施します。

   - FEFSセットアップツール用構成定義ファイルの作成
   - FEFSセットアップツール用構成定義ファイルの配置

4. LLIOサービスの起動
   ストレージI/Oノード、計算ノードのLLIOサービスを起動します。これらは運用系システム管理ノードで実施します。

```
[システム管理ノード]
# fefs_sync --start --compute=<cluster>[,…] --llio
```

   LLIOを構築するすべてのクラスタを指定してください。

3.3.2 ローリングアップデート

システム全体を停止せずにLLIOのパッケージを更新できます。

注意

パッケージには、ローリングアップデートの可否情報が記載されています。適用済のパッケージに対しては、rpm -qi コマンドで、適用予定のパッケージに対しては、rpm -qpi コマンドで事前に作業の可否・条件などを確認しておいてください。詳細は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド 保守編」の「ソフトウェア保守」を参照してください。

操作はSIOグループ単位で行います。

1. 事前準備
   「運用からの切り離し」と「ソフトウェアメンテナンスモードへの移行」を実施します。

   ### 参考

   詳細は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド 保守編」の「ソフトウェア保守の事前準備」を参照してください。

2. 対象範囲における LLIOサービスの停止
   運用系システム管理ノードで以下を実行してください。

```
[システム管理ノード]
# fefs_sync --stop --compute=<cluster> --llio --nodeid=<nodeid> --siogrp
```

   --compute :計算クラスタ名を指定してください。

   --nodeid :<*nodeid*> を含むSIOグループに対してコマンドが実行されます。

   ### 注意

   指定した範囲のノードの中で、LLIOパッケージが適用されていないノードに対して操作は行われません。(例:範囲指定オプションに --nodegrp (ノードグループ範囲)を指定した場合の計算クラスタサブ管理ノードに対するLLIOサービスの操作など)

3. 保守適用作業
   1. で停止した範囲においてパッケージ適用を行ってください。

4. 対象範囲における LLIOサービスの起動
   運用系システム管理ノードで以下を実行してください。

```
[システム管理ノード]
# fefs_sync --start --compute=<cluster> --llio --nodeid=<nodeid> --siogrp
```

   --compute : 計算クラスタ名を指定してください。

   --nodeid : <*nodeid*> を含む SIO グループに対してコマンドが実行されます。

5. 運用組み込み
   事前準備で運用から切り離していた保守対象を運用に組み込みます。

参照

詳細は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド 保守編」の「ソフトウェア保守後の運用組み込み」を参照してください。

参照

LLIOパッケージとFEFSクライアントパッケージを併せて適用する場合については、マニュアル「FEFSユーザーズガイド」を参照してください。

3.3.3 トラブル発生時の対処

運用中にトラブルが発生したときは、以下の資料を採取してください。

表3.11 トラブル発生時に必要な資料

採取資料の種類	対象ノード	採取ファイル/採取コマンド
システムログ	全ノード	/var/log/messages*
PANIC DUMP	DUMP が採取されたノード	padumpmgrコマンドで採取された、ダンプファイル
システムの資料	全ノード	pasnapコマンドで採取された、OSの調査資料
FEFSの資料	全ノード	以下のコマンドを実行し、作成された <outputdir>/fefssnap_<タイムスタンプ>.tgz # /usr/sbin/fefssnap -d <outputdir> ※<タイムスタンプ>はコマンドの実行時間(yyyymmddHHMMSS)です。 pasnap コマンドで採取された、FEFSの調査資料
LLIOの資料	全ノード	以下のコマンドを実行し、作成された <outputdir>/lliosnap_<タイムスタンプ>.tgz # /usr/sbin/lliosnap -d <outputdir> ※<タイムスタンプ>はコマンドの実行時間(yyyymmddHHMMSS)です。 pasnap コマンド(--directオプション指定)で採取された、LLIOの調査資料

参照

- fefssnapコマンドを実行することで、lliosnapコマンドが実行されます。lliosnapコマンドの詳細は、"A.2.2 lliosnapコマンド" を参照してください。
- fefssnapコマンドの詳細は、マニュアル「FEFS ユーザーズガイド」を参照してください。
- pasnapコマンドの資料採取方法およびpadumpmgrコマンドによるダンプファイルの採取方法の詳細は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド 保守編」を参照してください。

参考

- FEFSのサービスに問題がない場合、FEFSの資料は採取する必要はありません。
- トラブルの調査のためにFEFSの内部ログである fefs.logが必要となることがあります。fefs.logは、状況によっては存在しない場合があります。fefs.logに、何も出力するものがないのか、ログ出力が停止しているのかを調査するには、該当ノード上でpsコマンドを実行し

ます。
以下のコマンドが動作していればログ採取ができている状態です。

```
[当該ノード]
# ps -ef | grep fefslog
lctl fefslog start /var/opt/FJSVfefs/fefs.log
```

3.3.4 第2階層ストレージに残存した共有テンポラリ領域と第2階層ストレージのキャッシュ領域のファイルやディレクトリについて

本項では、ノードダウンなどにより第2階層ストレージに残存した共有テンポラリ領域と第2階層ストレージのキャッシュ領域のファイルやディレクトリの確認および削除方法について説明します。共有テンポラリ領域に作成したファイルやディレクトリは、第1階層ストレージに加えて第2階層ストレージ上にも作成されます。第2階層ストレージのキャッシュ領域では、削除ファイルを第2階層ストレージ上に一時的に残す場合があります。これらは、通常、ジョブ終了時に削除されますが、ノードダウンなどによって第2階層ストレージ上に残る場合があるため、削除する必要があります。

3.3.4.1 残存したファイルやディレクトリの確認方法

残存したファイルやディレクトリは、以下に示す4か所のディレクトリ<*ジョブID*>配下に存在している場合があります。

<*第2階層ストレージのマウントパス*(※1)>/.llio_share/<*job000-1ff*(※2)>/<*ジョブID*>
<*第2階層ストレージのマウントパス*(※1)>/.llio_sillyrename/<*job000-1ff*(※2)>/<*ジョブID*>
<*第2階層ストレージのマウントパス*(※1)>/.llio_share/trash/<*ストレージI/0ノードのホスト名*>/<*ジョブID*>
<*第2階層ストレージのマウントパス*(※1)>/.llio_sillyrename/trash/<*ストレージI/0ノードのホスト名*>/<*ジョブID*>

※1 第2階層ストレージを複数マウントしている場合は、それぞれ確認してください。
※2 ジョブごとのディレクトリを、job000～job1ffの512個のディレクトリに分散して配置しています。

3.3.4.2 残存したファイルやディレクトリの削除方法

残存したファイルやディレクトリはログインノードでルートユーザが削除してください。
ただし、以下のディレクトリはジョブ運用中に削除しないでください。これらは保守中に削除できます。

<*第2階層ストレージのマウントパス*>/.llio_share/<*job000-1ff*>
<*第2階層ストレージのマウントパス*>/.llio_share/<*job000-1ff*>/<*ジョブID*>
<*第2階層ストレージのマウントパス*>/.llio_sillyrename/<*job000-1ff*>
<*第2階層ストレージのマウントパス*>/.llio_sillyrename/<*job000-1ff*>/<*ジョブID*>
<*第2階層ストレージのマウントパス*>/.llio_share/trash/<*ストレージI/0ノードのホスト名*>
<*第2階層ストレージのマウントパス*>/.llio_sillyrename/trash/<*ストレージI/0ノードのホスト名*>

以下のディレクトリはジョブ運用中でも削除できます。

<*第2階層ストレージのマウントパス*>/.llio_share/trash/<*ストレージI/0ノードのホスト名*>/<*ジョブID*>
<*第2階層ストレージのマウントパス*>/.llio_sillyrename/trash/<*ストレージI/0ノードのホスト名*>/<*ジョブID*>

3.4 注意事項

ファイルロックについて

ファイルロックの有効範囲

LLIO のファイルロック機能を利用した場合のロックの有効範囲は、以下のとおりです。

- ノード内テンポラリ領域:
  名前空間の範囲 (1CN内) と同じ
- 共有テンポラリ領域:
  名前空間の範囲 (ジョブ内) と同じ

- 第 2階層ストレージのキャッシュ領域:
  名前空間の範囲と異なり、ロックの有効範囲はジョブ内に閉じる。つまり、同一ファイルに対して衝突するロックを複数のジョブから獲得しようとした場合は、すべてのジョブで排他の獲得が可能。

強制ロックのサポート

LLIO では NFS と同様、強制ロックはサポートしません。

fcntlとflockの相互作用

LLIO のファイルロック機能を利用した場合のfcntlとflockの相互作用は、ファイルシステムごとに以下の仕様となります。

- ノード内テンポラリ領域:
  Linuxの仕様に則りfcntlとflockの相互作用は存在しない。つまり、同一ファイルに対してfcntlで排他ロックをとるプロセスとflockで排他ロックをとるプロセスが同時に存在しても、いずれのプロセスのロック獲得も成功する。
- 共有テンポラリ領域:
  NFSの仕様に則り、fcntlとflockの相互作用は存在する。
- 第2階層ストレージのキャッシュ領域:
  NFSの仕様に則り、fcntlとflockの相互作用は存在する。

updatedbコマンドによるアクセスの抑止について

mlocateを導入している場合、定期的にupdatedbコマンドが実行されLLIOファイルシステムへ意図しないアクセスが行われることがあります。

LLIO ファイルシステムへのupdatedbコマンドによるアクセスを行わないようにするには、以下の設定を行い検索対象から除外してください。

【設定方法】

updatedbコマンドの検索対象からLLIO ファイルシステムを除外するには、LLIOクライアントにおいて"/etc/updatedb.conf"ファイルの"PRUNEFS"に "lliofs" を追加します。

例:/etc/updatedb.conf の設定例

```
PRUNEFS = "auto afs gfs gfs2 iso9660 sfs udf lliofs"
                                            ~~~~~~
```

クライアントノード間でのメタデータの一貫性保証に関する仕様の対処について

第2階層ストレージのキャッシュ領域または共有テンポラリ領域において、計算ノードAでファイルAを削除または別名に変更した後に再作成した場合、計算ノードBでは、ファイルAのopen(2)が失敗することや、計算ノードAで削除または別名に変更したファイルがopen(2)されることがあります。

この仕様は、以下で対処できます。

ー 計算ノードBで、ファイルAをopen(2)する前に、ファイルAの親ディレクトリに対してlsコマンドを実行する。

ー 計算ノードBで、ファイルAをopen(2)する前に、計算ノードAのファイルAの再作成から60秒間待つ。

付録A　リファレンス

A.1　システムコール

LLIOが対応しているシステムコールの種類を以下に示します。

LLIOが対応しているシステムコール

システムコール	対応状況
_llseek	○
access	○
bdflush	—
chdir	○
chmod	○
chown	○
chown32	○
chroot	○
close	○
creat	○
dup	—
dup2	—
execve	○
fchdir	○
fchmod	○
fchown	○
fchown32	○
fcntl	△ 勧告ロックのみ対応(強制ロックは非対応)
fcntl64	△ 勧告ロックのみ対応(強制ロックは非対応)
fdatasync	○
fgetxattr	○
flistxattr	○
flock	○
fremovexattr	○
fsetxattr	○
fstat	○
fstat64	○
fstatfs	○
fstatfs64	○
fsync	○
ftruncate	○
ftruncate64	○

システムコール	対応状況
getdents	○
getdents64	○
getxattr	○
ioctl	△ ストライプ機能のみ対応
lchown	○
lchown32	○
lgetxattr	○
link	○
listxattr	○
llistxattr	○
lremovexattr	○
lseek	○
lsetxattr	○
lstat	○
lstat64	○
mkdir	○
mknod	○
mmap	○
mount	○
munmap	—
open	○
pipe	—
pivot_root	×
pread64	○
pwrite64	○
read	○
readdir	○
readlink	○
readv	○
removexattr	○
rename	○
rmdir	○
setrlimit	—
setxattr	○
stat	○
stat64	○
statfs	○
statfs64	○
swapon	×

システムコール	対応状況
swapoff	×
symlink	○
sync	—
sysfs	×
truncate	○
truncate64	○
umount	○
umount2	○
unlink	○
utime	○
utimes	○
write	○
writev	○

○：対応　△：一部機能のみ対応　－:VFSレベルで対応　×:非対応　□:動作保証なし

A.2 コマンド

A.2.1 lfsコマンド

lfs getstripe

【名前】

lfs getstripe - 計算ノードから第2階層ストレージのストライプパターンを表示します。

【書式】

/usr/local/bin/lfs getstripe <*dirname* | *filename*>

【説明】

指定したファイルやディレクトリのストライプパターンの情報を表示します。<*dirname*>を指定することでディレクトリのストライプパターンを表示します。<*filename*>を指定することでファイルのストライプパターンを表示します。

lfs setstripe

【名前】

lfs setstripe - 計算ノードから第2階層ストレージのストライプパターンを設定します。

【書式】

/usr/local/bin/lfs setstripe [option] <*dirname* | *filename*>

【説明】

ストライプパターンを持った新規ファイルを作成、または既存のディレクトリのストライプパターンを設定します。

<*dirname*>を指定することでディレクトリのストライプパターンを指定します。<*filename*>を指定することでファイルのストライプパターンを指定します。

指定できるストライプサイズとストライプカウントの上限値、下限値は、第2階層ストレージの仕様を確認してください。

【オプション】

[--stripe-size | -S stripe_size]

ストライプサイズを設定します。

-S #k、-S #m、-S #g とすることで、サイズをKiB、MiB、GiB 単位で設定できます。

[--stripe-count | -c stripe_count]

ストライプカウントを設定します。

-1 と設定した場合、すべてのOST に書込みが行われます。

A.2.2 lliosnapコマンド

【名前】

lliosnap - 第1階層ストレージの調査に必要な資料の採取を行います。

【書式】

```
/usr/sbin/lliosnap -d <outputdir>
/usr/sbin/lliosnap --help
```

【説明】

lliosnap はトラブル調査に必要な資料の採取を行います。

採取されたデータは tar + gzip の形式で圧縮され、以下の名前で指定したディレクトリに格納されます。

lliosnap_<*タイムスタンプ*>.tgz

※*タイムスタンプ*: yyyymmddHHMMSS

本コマンドは管理者権限を持つユーザーのみが利用できます。

【オプション】

-d <*outputdir*>

採取した資料を格納するディレクトリを指定します。

資料採取時、本オプションは必須オプションです。

--help

usageを表示して終了します。

【終了ステータス】

以下の終了ステータスが返されます。

0: 正常終了

1: 異常終了

A.2.3 llio_transferコマンド

【名前】

llio_transfer - 共通ファイルを配布、または削除します。

【書式】

配布操作（非同期モード）：

```
/usr/bin/llio_transfer <path> [<path> ...]
```

配布操作（同期モード）：

```
/usr/bin/llio_transfer --sync <path> [<path> ...]
```

削除操作：

```
/usr/bin/llio_transfer --purge <path> [<path> ...]
```

使用方法表示：
```
/usr/bin/llio_transfer --help
```

【説明】

llio_transferコマンドはジョブスクリプト内で用いられるコマンドで、第2階層ストレージ上の共通ファイルを、第1階層ストレージ上の第2階層ストレージのキャッシュ領域へ配布します。
また、オプションを使用して配布した共通ファイルを第1階層ストレージ上の第2階層ストレージのキャッシュ領域から削除します。
<*path*> には、第2階層ストレージ上の共通ファイルの絶対パスまたは相対パスを指定します。
共通ファイルは、ストレージI/Oノードにまたがってストライプされません。
シンボリックリンクファイルを指定した場合、リンク先のファイルを第1階層ストレージ上の第2階層ストレージのキャッシュ領域へ配布します。
複数のファイルを指定したときは、先頭のファイルから順に処理し、途中のファイルで配布または削除に失敗しても、継続不可能なエラー以外は、最後のファイルまで処理を継続します。
「配布操作 (非同期モード) 」の場合は、<*path*> に指定したファイルの配布開始の指示までを行い、llio_transferコマンドは復帰します。
「配布操作 (同期モード) 」の場合は、<*path*> に指定したファイルの配布完了後に、llio_transferコマンドは復帰します。

【オプション】

--sync

共通ファイルを同期モードで配布します。

--purge

配布した共通ファイルを、第1階層ストレージ上の第2階層ストレージのキャッシュ領域から削除します。

--help

標準出力に使用方法のメッセージを出力して正常終了します。

注意

llio_transfer コマンドで第2階層ストレージのキャッシュへコピーしたファイルを共通ファイルと呼びます。共通ファイルに関して、以下に注意してください。

- 共通ファイルに対する第2階層ストレージのキャッシュ領域経由の削除、ファイルデータ更新やファイル属性更新の操作はエラーで失敗します。
- 共通ファイルに対するファイルロックは未サポートです。
- ジョブ実行中は第2階層ストレージにある共通ファイルのコピー元を変更したり削除したりしないでください。これを変更した場合は、第2階層ストレージのキャッシュにコピーした共通ファイルの内容が不定になる可能性があります。また、これを削除した場合は、共通ファイルをllio_transferコマンドの--purgeオプションで削除できなくなり、ジョブが終了するまで第2階層ストレージのキャッシュ領域が共通ファイルに占有されたままになります。
- ジョブスクリプト内で共通ファイルを削除する場合は、llio_transferコマンドに--purgeオプションを指定して削除してください。rmコマンドはエラーで失敗し、削除できません。
- llio_transfer --purgeで第2階層ストレージのキャッシュから共通ファイルを削除した直後は、ジョブからの共通ファイルのコピー元ファイルの削除、ファイルデータ更新やファイル属性更新の操作がエラーで失敗する場合があります。その場合は、60秒待ってから、操作を再実行してください。
- 共通ファイルのコピー先である、第1階層ストレージの "第2階層ストレージのキャッシュ領域" には共通ファイル全体を格納できるだけの容量が必要です。第2階層ストレージのキャッシュ領域にコピーされた共通ファイルは、この領域が一杯になっても削除されることはありません。第2階層ストレージのキャッシュ領域の空き容量が必要な場合には、ジョブの途中で、llio_transferコマンドに--purgeオプションを指定してキャッシュ領域から共通ファイルを削除してください。なお、第2階層ストレージのキャッシュ領域の共通ファイルは、ジョブ終了後には自動的に削除されます。
- コマンドが共通ファイルの領域を確保する前に、ジョブ内で転送元ファイルのキャッシュデータが第1階層ストレージに存在する場合、llio_transferコマンドはエラーになります。ファイルオープンをした場合にキャッシュデータが作成される場合があるため、llio_transferコマンド実行前にジョブ内でファイルオープンをしないでください。以下のコマンドも、ファイルオープンする場合があるため、使用しないでください。
  - lfs setstripe

- lfs getstripe

なお、llio_transfer コマンドがエラーになった場合にジョブを続行すると、共通ファイルが配布されていない状態でジョブは実行されます。

ー llio_transferコマンドに指定できるファイルは以下の条件を満たす必要があります。

- 本コマンド実行ユーザーがファイルにアクセス可能、かつ、
- 第2階層ストレージのファイル、かつ、
- ファイルの種類が通常ファイル、かつ、
- ファイルサイズが1Byte以上のファイル

【終了ステータス】

以下の終了ステータスが返されます。

0: 正常終了

1: 異常終了

複数のファイルを指定したときは、すべてのファイルの配布または削除処理が正常終了した場合に0を返します。

A.2.4 showsiostatsコマンド

【名前】

showsiostats - システム統計情報のログファイルから、ジョブに関するシステム統計情報を出力します。

【書式】

```
/opt/FJSVllio/bin/showsiostats -d <dirname> -j <jobid> [--from=<value>] [--stat=<value>] [--kind=<value>]
                                  [--type=<value>] [--section=<value>]
                               --help
```

【説明】

showsiostatsコマンドは、<*dirname*>に指定したディレクトリ配下のストレージI/Oノード向けシステム統計情報のログファイルから、<*jobid*>で指定されたジョブに関するシステム統計情報を出力するコマンドです。

本コマンドは、システム統計情報を収集したノードでシステム管理者の権限を持つユーザーが利用できます。

【オプション】

-d <*dirname*>

抽出元のログファイルが検索するディレクトリを指定します。なお、指定したディレクトリのサブディレクトリも検索対象です。

-j <*jobid*>

抽出対象のジョブIDを指定します。

--from=<*value*>

統計情報の日時による出力情報の絞り込みを行います。<value>には以下を指定できます。

yyyymmdd:hh:mm:ss-yyyymmdd:hh:mm:ss : 指定時刻範囲

--stat=<*value*>

統計情報の種別による出力情報の絞り込みを行います。<value>には以下を指定できます。

j: ジョブごとのI/O情報
c: 通信層のコネクション数に関する情報
r: リクエスト処理に関する情報

--kind=<*value*>

ジョブごとのI/O 情報の種類による出力情報の絞り込みを行います。本オプションで指定した統計情報のみを出力します。

"/"区切りで複数指定ができます。本オプションを指定しない場合はすべての情報を出力します。

<value>には以下を指定できます。

io：I/O 情報
meta：メタアクセス情報
rsc：リソース情報
async：非同期転送情報

--type=<*value*>

利用形態の種類によるジョブごとのI/O情報の絞り込みを行います。本オプションで指定した統計情報のみを出力します。"/"区切りで複数指定ができます。本オプションを指定しない場合は、すべての情報を出力します。

<*value*>には以下を指定できます。

cache：第2 階層ストレージのキャッシュの情報
share：共有テンポラリ領域の情報
local：ノード内テンポラリ領域の情報

--section=<*value*>

I/O箇所の種類によるジョブごとのI/O 情報の絞り込みを行います。本オプションで指定した統計情報のみを出力します。"/"区切りで複数指定ができます。本オプションを指定しない場合はすべての情報を出力します。

<*value*>には以下を指定できます。

Total：全体のI/O情報
CN-CBCN：計算ノードと通信用バッファ間のI/O情報
CBCN-Dev：通信用バッファと第1階層ストレージ間の転送のI/O情報(計算ノードとの通信)
CBGFS-Dev：通信用バッファと第1階層ストレージ間の転送のI/O情報(第2階層ストレージとの通信)
CBCN-GFS：通信用バッファと第2階層ストレージ間の転送のI/O情報(Direct I/O)
CBGFS-GFS：通信用バッファと第2階層ストレージ間の転送のI/O情報(Cached I/O)

--help

本コマンドの使用方法を表示します。
このオプションを指定した場合、引数およびそのほかのオプションはすべて無効になります。

【終了ステータス】

以下の終了ステータスが返されます。

0: 正常終了

1: 異常終了

A.3 ライブラリ

A.3.1 getlliostat

【書式】

```
#include <llio_stat.h>

getlliostat (lliostat_t * lliostat);
```

【説明】

計算ノード上で第1階層ストレージが管理している統計情報を引数 lliostat に設定して返します。

本ライブラリを呼び出すごとにジョブ開始時点から関数実行時までの累積値を返す項目と、現在値を返す項目があります。

例えば、使用量を返す項目 (キャッシュ未使用領域量、キャッシュ済み領域量、未書出しキャッシュ領域量)についてはライブラリ実行時点での値を取得します。

また時間や回数を返す項目(Writeの総回数、Writeの総転送量、Writeの総処理時間)などについてはジョブ開始時点からの累積値を出力します。

なお、統計情報の各項目はuint64_t 型で管理され、オーバーフローした場合は0からカウントしなおします。

【引数】

lliostat_t *lliostat

統計情報を格納する構造体を指定します。構造体の定義は“C.2 計算ノード統計情報の採取方法と出力項目”を参照してください。

【終了ステータス】

以下の終了ステータスが返されます。

0: 正常終了

1: 異常終了

【ライブラリパス】

/opt/FJSVllio/lib/liblliostat.so

ジョブ実行にはLD_LIBRARY_PATHに/opt/FJSVllio/libを指定し実行してください。

【インクルードヘッダファイルパス】

/opt/FJSVllio/include/llio_stat.h

付録B　メッセージ

B.1　システムログに出力されるメッセージ

以下は、LLIOがシステムログに出力するメッセージの形式です。

```
[ERR.] [LLIO] 0001 jobid lliofs Job is already running.
(1)    (2)   (3)  (4)   (5)    (6)
```

1. メッセージ種別

   メッセージの出力レベルを表します。3. のメッセージID と連動しており、内容は以下のとおりです。

   - [ERR.]: ERRORメッセージ(0001から5999)
   - [WARN]: WARNINGメッセージ(6000から6999)
   - [NOTE]: NOTICE メッセージ(7000から7999)
   - [INFO]: INFO メッセージ(8000から8999)

2. LLIO プレフィックス

   メッセージが 第1階層ストレージ 関連の出力であることを示す識別子です。

   内容は以下のとおりです。

   - [LLIO]: 第1階層ストレージ本体が出力するメッセージです。

3. メッセージID

   メッセージの識別ID です。1. のメッセージ種別と連動しています。

4. ジョブID

   ジョブIDです。該当するジョブIDがない場合は、”-“ が出力されます。

5. サブコンポ名

   LLIOのサブコンポ名です。

6. メッセージ内容

   メッセージの内容です。

[ERROR メッセージ] (0001から5999)

[ERR.] [LLIO] 0001 *jobid* lliofs Job is already running.

意味

第1階層ストレージを指定したジョブが実行中です。

対処

第1階層ストレージを使用する複数のジョブが同じジョブIDでジョブ実行を開始しようとしている可能性があります。担当保守員(SE)、または当社SupportDeskに連絡してください。

[ERR.] [LLIO] 0002 *jobid* lliofs Unable to unmount(*mountpoint*).

意味

第1階層ストレージのアンマウント処理が失敗しました。

mountpoint: マウントポイント

対処

担当保守員(SE)、または当社SupportDeskに連絡してください。

[ERR.] [LLIO] 0005 *jobid* lliofs Cannot allocate memory.
[ERR.] [LLIO] 1000 *jobid* lliosv Cannot allocate memory.
[ERR.] [LLIO] 2000 *jobid* lliost Cannot allocate memory.
[ERR.] [LLIO] 2300 *jobid* lliolog Cannot allocate memory.
[ERR.] [LLIO] 2600 *jobid* lliosv_qos Cannot allocate memory.
[ERR.] [LLIO] 2800 *jobid* lliorpc Cannot allocate memory.
[ERR.] [LLIO] 2900 *jobid* lliolib Cannot allocate memory.

意味

第1階層ストレージモジュールの処理に必要なメモリの獲得に失敗しました。

対処

メッセージが出力されたノードのメモリの使用状況を確認してください。原因が不明の場合は、担当保守員(SE)、または当社SupportDeskに連絡してください。

[ERR.] [LLIO] 0007 jobid lliofs Mount(*path*) failed(*reason*).

意味

path へのマウントが *reason* により失敗しました。

対処

reason が以下の場合、指定したマウントオプションを確認してください。

"unknown option"

未サポートのオプションが指定されています。

"incompatible filesystem type for lazystatfs option"

lazystatfs オプションが共有テンポラリ以外に指定されています。

"invalid min/max relationship"

最大値、最小値を設定するオプションに対し、大小関係が不当となる値が指定されています。

"too long JOBID"

指定された JOBID の文字列が長すぎます。

"bad option value for attr_dir_min"

"bad option value for attr_dir_max"

"bad option value for attr_reg_min"

"bad option value for attr_reg_max"

オプション引数に指定している値が不当です。

"lack of mandatory option"

JOBID オプションが指定されていません。

"unknown filesystem type"

マウントするファイルシステムとして、local/share/global 以外が指定されています。

"invalid specified format for global filesystem"

global のマウント対象の指定方法が不当です。

"JOB not found"

指定されたJOBが開始されていません。

reason が以下の場合、担当保守員(SE)、または当社SupportDeskに連絡してください。

"insufficient memory"

メモリ不足が発生しました。

"system error"

内部エラーが発生しました。

[ERR.] [LLIO] 1001 *jobid* lliosv System error(*detail*).
[ERR.] [LLIO] 2001 *jobid* lliost System error(*detail*).
[ERR.] [LLIO] 2301 *jobid* lliolog System error(*detail*).
[ERR.] [LLIO] 2400 *jobid* llio_transfer System error(*detail*).
[ERR.] [LLIO] 2901 *jobid* lliolib System error(*detail*).

意味

第1階層ストレージモジュールの処理でシステムエラーが発生しました。

detail: 保守用の詳細情報

対処

担当保守員(SE)、または当社SupportDeskに連絡してください。

[ERR.] [LLIO] 1100 *jobid* lliosv Module load error. Specified(*filesystem*) mount point(*mountpoint*) is not an actual mount point.

意味

第1階層ストレージモジュールのロードに失敗しています。デザインシートで指定されたパスはマウントポイントではありません。

filesystem:ファイルシステム名です。以下のどちらかが出力されます。

2nd_layer storage:第2階層ストレージのマウントポイント

shared temporary namespace file system:共有テンポラリ領域の名前空間管理用ファイルシステムのマウントポイント

mountpoint: 指定されたマウントポイント

対処

FEFSデザインシートで指定したマウントポイントに誤りがないか確認してください。

[ERR.] [LLIO] 1101 *jobid* lliosv Module load error. Specified(*filesystem*) mount point(*mountpoint*) does not exist.

意味

第1階層ストレージモジュールのロードに失敗しています。デザインシートで指定されたパスがファイルシステム内に存在しないディレクトリです。

filesystem:ファイルシステム名です。以下のどちらかが出力されます。

2nd_layer storage:第2階層ストレージのマウントポイント

shared temporary namespace file system:共有テンポラリ領域の名前空間管理用ファイルシステムのマウントポイント

mountpoint: 指定されたマウントポイント

対処

FEFSデザインシートで指定したマウントポイントに誤りがないか確認してください。

[ERR.] [LLIO] 2002 *jobid* lliost 1st_layer storage device mkfs error(dev=*devname*).

意味

第1階層ストレージデバイスのmkfsに失敗しました。

devname: デバイス名

対処

デバイスが使用可能な状態になっていることを確認してください。原因が不明の場合は、担当保守員(SE)、または当社SupportDeskに連絡してください。

[ERR.] [LLIO] 2200 *jobid* lliost The state of the 1st_layer storage device(*SSD_path*) is abnormal(*critical_warning*).

意味

第1階層ストレージデバイスの状態が正常ではありません。

SSD_path: 第1階層ストレージデバイスのパス名

critical_warning: *critical warning*の値。以下のいずれか、もしくは OR を取った値が表示されます。

値	説明
0x02	温度異常
0x04	信頼性低下
0x08	読込み専用モードに移行
0x10	バックアップデバイス異常

対処

第1階層ストレージデバイスの状態を確認し、必要に応じて保守を行ってください。

[WARNING メッセージ] (6000から6999)

[WARN] [LLIO] 6300 *jobid* lliost The available spare space of the 1st_layer storage device(*SSD_path*) is below the hardware threshold(current=*avail_spare*%, threshold=*spare_thresh*%).

意味

第1階層ストレージデバイスの予備領域の残量がハードウェアの閾値未満となりました。

SSD_path: 第1階層ストレージデバイスのパス名

avail_spare: メッセージ出力時の *avail_spare* の値

spare_thresh: *avail_space* に対するハード規定の閾値の値

対処

予備領域が枯渇しないように、必要に応じて保守を行ってください。

[WARN] [LLIO] 6400 *jobid* llio_transfer Failed to transfer common file(*path*).

意味

llio_transferコマンドの引数に指定されたファイルの転送に失敗しました。

共通ファイルの転送中に、以下の操作が行われた可能性があります。

- 第2階層ストレージ上のファイルを削除
- 第2階層ストレージ上のファイルを更新
- 同一ジョブスクリプト内で llio_transferコマンドを--purgeオプション指定で実行

path: 転送に失敗したファイルのパス

対処

第2階層ストレージ上のファイルが存在することを確認してください。
共通ファイルの転送中に、第2階層ストレージ上のファイルを更新しないでください。また、共通ファイルの転送中に、同一ジョブスクリプト内で llio_transferコマンドを--purgeオプション指定で実行しないでください。

[NOTICE メッセージ] (7000から7999)

[NOTE] [LLIO] 7350 *jobid* lliolog Starting lliolog daemon.

意味

第1階層ストレージのログデーモンを起動しています。

対処

対処不要です。

[NOTE] [LLIO] 7351 *jobid* lliolog Shutting down lliolog daemon.

意味

第1階層ストレージのログデーモンを終了します。

対処

対処不要です。

[INFO メッセージ] (8000から8999)

[INFO] [LLIO] 8300 *jobid* lliost The available spare space of the 1st_layer storage device(*SSD_path*) is below the software threshold(current=*avail_spare*%, threshold=*spare_thresh*%).

意味

第1階層ストレージデバイスの予備領域の残量を示します。

SSD_path: 第1階層ストレージデバイスのパス名

avail_spare: メッセージ出力時の *avail_spare* の値

spare_thresh: *avail_space* に対するソフト規定の閾値の値

対処

対処不要です。

B.2 コマンドの出力するメッセージ

コマンド実行時に異常が発生した場合、以下のメッセージを標準エラー出力に出力します。以下は、コマンドが出力するメッセージの形式です。

```
[ERR.] LLIO 2750 lliosnap -d: Outputdir: No such directory.
(1)    (2)  (3)  (4)      (5)
```

1. メッセージ種別

   メッセージの出力レベルを表します。3. のメッセージID と連動しており、内容は以下のとおりです。

   - [ERR.]: ERRORメッセージ(0001から5999)
   - [WARN]: WARNINGメッセージ(6000から6999)
   - [NOTE]: NOTICE メッセージ(7000から7999)
   - [INFO]: INFO メッセージ(8000から8999)

2. LLIO プレフィックス

   メッセージが 第1階層ストレージ関連の出力であることを示す識別子です。

   内容は以下のとおりです。

   - LLIO: 第1階層ストレージ本体が出力するメッセージです。

3. メッセージID

   メッセージの識別ID です。1. のメッセージ種別と連動しています。

4. コマンド名

   コマンド名です。

5. メッセージ内容

   メッセージの内容です。

B.2.1 lfsコマンド

共通

[ERR.] LLIO 3150 lfs : <*command*> is not supported.

意味

指定したサブコマンドはサポートされていません。

パラメーターの説明

command : 指定されたサブコマンド

対処

指定できるサブコマンドは setstripe, getstripeのみです。
指定するサブコマンドを見直してください。

lfs setstripe

[ERR.] LLIO 3151 lfs setstripe: Missing filename|dirname.

意味

ファイル名またはディレクトリ名の指定がありません。

対処

ファイル名またはディレクトリ名を指定してください。

[ERR.] LLIO 3152 lfs setstripe: Bad stripe size(*size*).

意味

オプション-S で指定されたストライプサイズが適切ではありません。

size : 指定したストライプサイズ値

対処

指定したストライプサイズの値を見直してください。

[ERR.] LLIO 3153 lfs setstripe: <*path*> is not on a global file system cache.

意味

指定されたパスは第 2階層ストレージキャッシュ上ではありません。

path : 指定したパス名

対処

指定されたパス名を見直してください。

[ERR.] LLIO 3154 lfs setstripe: Bad stripe size(*size*), must be multiple of 65536 bytes Invalid argument(22).

意味

ストライプサイズの指定は65536バイトの倍数である必要があります。

size : 指定したストライプサイズ値

対処

ストライプサイズは65536バイトの倍数を指定してください。

[ERR.] LLIO 3155 lfs setstripe: Stripe size 4G or larger is not currently supported and would wrap Invalid argument (22).

意味

ストライプサイズの指定は 4194240KiB (4GiB-64KiB) 以下である必要があります。

対処

ストライプサイズは 4194240KiB(4GiB-64KiB) 以下を指定してください。

[ERR.] LLIO 3156 lfs setstripe: bad stripe count(*count*).

意味

オプション-cで指定されたストライプカウントが適切ではありません。

count : 指定したストライプカウント値

対処

指定したストライプカウントの値を見直してください。

[ERR.] LLIO 3157 lfs setstripe: Unable to open(*path*) *errmsg*(*err*).

意味

path に指定されたファイルまたはディレクトリがオープンできませんでした。

path: 指定したパス名
errmsg: エラーメッセージ
err: エラーコード

対処

指定したパスを見直してください。

[ERR.] LLIO 3158 lfs setstripe: ioctl() failed for(*path*) *errmsg*.

意味

ストライプ情報を設定するioctl()が失敗しました。

path: 指定したパス名
errmsg: エラーメッセージ

対処

エラーメッセージが”stripe already set”の場合は、指定したパスにすでにファイルが存在しています。パス名を見直して再実行してください。

上記以外の場合は担当保守員(SE) 、または当社SupportDeskに連絡してください。

[ERR.] LLIO 3159 lfs setstripe: setstripe failed <*errmsg*>.

意味

ストライプ情報の設定が失敗しました。

errmsg: エラーメッセージ

対処

エラーメッセージに出力されているメッセージを参考に対処してください。

lfs getstripe

[ERR.] LLIO 3160 lfs getstripe: Failed for(*<path>*).

意味

ディレクトリ名またはファイル名の指定が適切ではありません。

path: 指定したパス名

対処

指定したディレクトリ名またはファイル名の指定を見直してください。

[ERR.] LLIO 3161 lfs getstripe: ioctl() failed for(*<path>*) *<errmsg>*.

意味

ストライプ情報を取得するioctl()が失敗しました。

path: 指定したパス名
errmsg: エラーメッセージ

対処

エラーメッセージが "*<path>* is not on a global file system cache."の場合は、指定したパス名は第2階層ストレージキャッシュ上のファイルではありません。パス名を見直して再実行してください。

エラーメッセージが "Stripe information is not set." の場合は、ストライプ情報が設定されていません。ストライプ情報を設定してください。

上記以外の場合は担当保守員(SE) 、または当社SupportDeskに連絡してください。

[ERR.] LLIO 3162 lfs getstripe: getstripe failed *<errmsg>*.

意味

ストライプ情報の取得に失敗しました。

errmsg: エラーメッセージ

対処

エラーメッセージに出力されているメッセージを参考に対処してください。

B.2.2 lliosnapコマンド

[ERR.] LLIO 2750 lliosnap -d: (*Outputdir*) No such directory.

意味

指定されたディレクトリが見つかりません。

Outputdir : -d オプションで指定したディレクトリパス

対処

-d オプションの指定を見直してください。

[ERR.] LLIO 2752 lliosnap Exist temporary directory(*Tmpdir*).

意味

作業領域がすでに存在します。

Tmpdir: 作業領域のディレクトリパス

対処

*Tmpdir*に書かれているディレクトリパスは本コマンドで作業領域として使用するディレクトリパスです。*Tmpdir*に書かれているディレクトリパスを別のパス名に変更してください。

[ERR.] LLIO 2753 lliosnap Cannot create temporary directory(*Tmpdir*).

意味

作業領域が作成できませんでした。

Tmpdir: 作業領域のディレクトリパス

対処

作業領域が作成できる状態になっているか確認してください。

[ERR.] LLIO 2754 lliosnap Cannot create output file.

意味

出力ファイルが作成できませんでした。

対処

出力先ディレクトリの状態を確認してください。

B.2.3 llio_transferコマンド

[ERR.] LLIO 2450 llio_transfer Not enough disk space.

意味

ジョブで使用可能な第1階層ストレージの空き容量が不足しています。

対処

ジョブで使用可能な第1階層ストレージの容量を増やしてください。

[ERR.] LLIO 2451 llio_transfer Command is already running.

意味

llio_transferコマンドは現在実行中です。

対処

llio_transferコマンドが実行中でないことを確認してください。

[ERR.] LLIO 2452 llio_transfer System error(*detail*).

意味

llio_transferコマンドでの処理でシステムエラーが発生しました。

detail: 保守用の詳細情報

対処

担当保守員(SE)、または当社SupportDeskに連絡してください。

[ERR.] LLIO 2453 llio_transfer Unrecognized option(*input*).

意味

不明なオプションが指定されました。

input: 不明なオプション

対処

指定したオプションを確認してください。

[ERR.] LLIO 2454 llio_transfer Missing path operand.

意味

ファイルのパスが指定されていません。

対処

ファイルのパスを指定してください。

[ERR.] LLIO 2455 llio_transfer Missing option format.

意味

オプションの指定方法が正しくありません。

対処

指定したオプションを確認してください。

[ERR.] LLIO 2456 llio_transfer System error(*detail*) detail code(*code*) ¥n.

意味

llio_transferコマンドでの処理でシステムエラーが発生しました。

detail: 保守用の詳細情報
code: 保守用の詳細エラーコード

対処

担当保守員(SE)、または当社SupportDeskに連絡してください。

[WARN] LLIO 6450 llio_transfer Permission denied(*path*).

意味

llio_transferコマンドの引数に指定されたファイルの権限が不正です。

path:アクセス権のないファイルのパス

対処

llio_transferコマンドの引数に指定したファイルのアクセス権限を確認してください。

[WARN] LLIO 6451 llio_transfer File does not exist(*path*).

意味

llio_transferコマンドの引数に指定されたファイルが存在しません。

path:存在しないファイルのパス

対処

llio_transferコマンドの引数に指定したファイルが存在するか確認してください。

[WARN] LLIO 6452 llio_transfer File was already cached(*path*).

意味

llio_transferコマンドの引数に指定したファイルは、アプリケーションによってジョブ内で過去にキャッシュされていました。

path:アプリケーションが過去にキャッシュしたファイルのパス

対処

ファイルオープンをした場合にキャッシュデータが作成される場合があるため、llio_transferコマンド実行前にジョブ内でファイルオープンをしないでください。以下のコマンドも、ファイルオープンする場合があるため、使用しないでください。

- lfs setstripe
- lfs getstripe

[WARN] LLIO 6453 llio_transfer File is not supported(*path*) detail (*reason*).

意味

llio_transferコマンドの引数に指定したファイルは、共通ファイル配布の対象外です。

path:コマンドがサポートしていないファイルのパス
reason: サポート外と判定された理由

"file is not a global file": 第 2階層ストレージのファイルではありません。
"not regular file": ファイルの種類が通常ファイルではありません。
"file size is zero": ファイルのサイズが 1byte 以上ではありません。

対処

reasonの内容を確認し、適切なファイルを指定してください。

[WARN] LLIO 6454 llio_transfer File has already been distributed(*path*).

意味

llio_transferコマンドの引数に指定したファイルは、共通ファイルを配布済みです。

path: 共通ファイルを配布済みのファイルのパス

対処

対処不要です。

[WARN] LLIO 6455 llio_transfer There is no file to be deleted(*path*).

意味

llio_transferコマンドの引数に指定したファイルは、削除対象の共通ファイルがありません。

path: 削除対象の共通ファイルがないファイルのパス

対処

対処不要です。

[WARN] LLIO 6456 llio_transfer Pathname is not suitable(*path*).

意味

llio_transferコマンドの引数に指定したファイルのパス名が不適切です。

path: 不適切なパス名

対処

引数のパス名の中で示すディレクトリが、正しいディレクトリを指しているかを確認してください。

あるいは、パス名の中のシンボリックリンクの利用回数を減らしてください。

[WARN] LLIO 6457 llio_transfer Filename is too long(*path*).

意味

llio_transferコマンドの引数に指定したパス名が長すぎます。

path: 長すぎるパス名

対処

引数に指定するパス名を短くしてください。

[WARN] LLIO 6458 llio_transfer Old file is being deleted(*path*).

意味

llio_transferコマンドの引数に指定したファイルは、過去に共通ファイルとして配布されており、現在削除中です。

path: 削除中の共通ファイルのパス

対処

ファイルがクローズ済か確認してください。クローズ済でない場合はクローズしてからコマンドを再実行してください。
非同期クローズ機能が有効でファイルがクローズ済でも本メッセージが出る場合、数秒待ってからコマンドを再実行してください。

B.2.4 showsiostatsコマンド

[ERR.] LLIO 2510 showsiostats Invalid option specified(option=*Opt*).

意味

不明なオプションが指定されました。

対処

指定したオプションを確認してください。

[ERR.] LLIO 2511 showsiostats Required option not specified(option=*Opt*).

意味

必要なオプションが指定されていません。

対処

指定したオプションを確認してください。

[ERR.] LLIO 2512 showsiostats Directory not exist(path=*path*).

意味

-dオプションで指定したディレクトリパスが存在しません。

対処

指定したディレクトリパスを確認してください。

[ERR.] LLIO 2513 showsiostats --from option format error(from=*from*).

意味

--fromオプションで指定した書式が正しくありません。

対処

--fromオプションで指定した書式を確認してください。

付録C　統計情報の出力項目

C.1　LLIO性能情報の出力項目

LLIO性能情報は、第1階層ストレージ(LLIO)に対するジョブの入出力の統計情報です。LLIO性能情報の出力先には、以下に示す2種類のファイルがあります。

- ジョブの実行結果として出力されるファイル(ジョブ投入時にpjsubコマンドの--llio perfオプション指定時)

  LLIO性能情報は以下のように出力されます。

```
I/O     2ndLayerCache   NodeTotal                       Count           Amount(Byte)        Time(us)
                Write(Cached I/O)       Sum             〈値〉           〈値〉              〈値〉
                                        [1,4Ki)         〈値〉           〈値〉              〈値〉
                                        [4Ki,1Mi)       〈値〉           〈値〉              〈値〉
                                        [1Mi,4Mi)       〈値〉           〈値〉              〈値〉
                                        [4Mi+           〈値〉           〈値〉              〈値〉
                Read(Cached I/O)        Sum             〈値〉           〈値〉              〈値〉
                                        [1,4Ki)         〈値〉           〈値〉              〈値〉
                                        ...
I/O     2ndLayerCache   ComputeNode<->CommBuf          Count           Amount(Byte)        Time(us)
                Write(Cached I/O)       Sum             -               -                   〈値〉
...
                        ※ここまでは、ジョブのサマリ情報
SIO Information         ※ここからは、ストレージI/Oノードごとの情報
Node ID : 〈ノードID〉
I/O     2ndLayerCache   NodeTotal                       Count           Amount(Byte)        Time(us)
                Write(Cached I/O)       Sum             〈値〉           〈値〉              〈値〉
                                        [1,4Ki)         〈値〉           〈値〉              〈値〉
                                        [4Ki,1Mi)       〈値〉           〈値〉              〈値〉
                                        [1Mi,4Mi)       〈値〉           〈値〉              〈値〉
                                        [4Mi+           〈値〉           〈値〉              〈値〉
                Read(Cached I/O)        Sum             〈値〉           〈値〉              〈値〉
                                        [1,4Ki)         〈値〉           〈値〉              〈値〉
                                        ...
I/O     2ndLayerCache   ComputeNode<->CommBuf          Count           Amount(Byte)        Time(us)
                Write(Cached I/O)       Sum             -               -                   〈値〉
...

Node ID : 〈ノードID〉
I/O     2ndLayerCache   NodeTotal           Count   Amount(Byte)    Time(us)
                Write(Cached I/O)       Sum             〈値〉           〈値〉              〈値〉
                ...
```

  各行の意味については"表C.1 LLIO性能情報の出力項目"を参照してください。

  項目の出力値<*値*>には以下があります。

  ― Count：総回数

  ― Amount(Byte)：総データ量(バイト)

  ― Time(us)：総処理時間(マイクロ秒)

- ジョブ運用ソフトウェアのログファイル(計算クラスタ管理ノードの/var/opt/FJSVtcs/shared_disk/pjm/jsti/llioinfo)

  LLIO性能情報は以下のようにCSV形式で出力されます。

```
JLLIO,〈ジョブID〉,〈ステップ番号〉,〈バルク番号〉,〈#1〉,〈#2〉,〈#3〉,〈#4〉,...
SLLIO,〈ノードID〉,〈#1〉,〈#2〉,〈#3〉,〈#4〉,...
SLLIO,〈ノードID〉,〈#1〉,〈#2〉,〈#3〉,〈#4〉,...
...
```

"JLLIO"で始まる行は、当該ジョブのLLIO性能情報です。"SLLIO"で始まる行は、そのジョブのLLIO性能情報を、ジョブの入出力を処理したストレージI/Oノードごとに出力したものです。"JLLIO"行と"SLLIO"行は1つのジョブについて連続で出力されます。値*<#n>*は、"表C.1 LLIO性能情報の出力項目"に示される内容に対応します。

表C.1 LLIO性能情報の出力項目

I/O 2ndLayerCache NodeTotal 第2階層ストレージのキャッシュ領域へのI/O情報(サマリ)		llioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	*<#1>*	*<#2>*	*<#3>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	*<#4>*	*<#5>*	*<#6>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	*<#7>*	*<#8>*	*<#9>*
	[4Mi+：4Miバイト以上	*<#10>*	*<#11>*	*<#12>*
Read(Cached I/O) O_DIRECT未指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	*<#13>*	*<#14>*	*<#15>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	*<#16>*	*<#17>*	*<#18>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	*<#19>*	*<#20>*	*<#21>*
	[4Mi+：4Miバイト以上	*<#22>*	*<#23>*	*<#24>*
Write(Direct I/O) O_DIRECT指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	*<#25>*	*<#26>*	*<#27>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	*<#28>*	*<#29>*	*<#30>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	*<#31>*	*<#32>*	*<#33>*
	[4Mi+：4Miバイト以上	*<#34>*	*<#35>*	*<#36>*
Read(Direct I/O) O_DIRECT指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	*<#37>*	*<#38>*	*<#39>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	*<#40>*	*<#41>*	*<#42>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	*<#43>*	*<#44>*	*<#45>*
	[4Mi+：4Miバイト以上	*<#46>*	*<#47>*	*<#48>*

I/O 2ndLayerCache ComputeNode<->CommBuf 第2階層ストレージのキャッシュ領域へのI/O情報 (計算ノードと通信用バッファ間のI/O情報)		llioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#49>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#50>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#51>*
	[4Mi+：4Miバイト以上	-	-	*<#52>*
Read(Cached I/O) O_DIRECT未指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#53>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#54>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#55>*
	[4Mi+：4Miバイト以上	-	-	*<#56>*

I/O 2ndLayerCache ComputeNode<->CommBuf 第2階層ストレージのキャッシュ領域へのI/O情報 （計算ノードと通信用バッファ間のI/O情報）		IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Direct I/O) O_DIRECT指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	<#57>
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	<#58>
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	<#59>
	[4Mi+：4Miバイト以上	-	-	<#60>
Read(Direct I/O) O_DIRECT指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	<#61>
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	<#62>
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	<#63>
	[4Mi+：4Miバイト以上	-	-	<#64>

I/O 2ndLayerCache CommBuf<->1st&2ndStorageDev 第2階層ストレージのキャッシュ領域へのI/O情報 （通信用バッファと階層ストレージ間の転送のI/O情報）		IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	<#65>
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	<#66>
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	<#67>
	[4Mi+：4Miバイト以上	-	-	<#68>
Read(Cached I/O) O_DIRECT未指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	<#69>
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	<#70>
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	<#71>
	[4Mi+：4Miバイト以上	-	-	<#72>
Write(Direct I/O) O_DIRECT指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	<#73>
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	<#74>
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	<#75>
	[4Mi+：4Miバイト以上	-	-	<#76>
Read(Direct I/O) O_DIRECT指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	<#77>
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	<#78>
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	<#79>
	[4Mi+：4Miバイト以上	-	-	<#80>

I/O 2ndLayerCache 1stStorageDev->CommBuf(Async) 第2階層ストレージのキャッシュ領域へのI/O情報 (第1階層ストレージから通信用バッファへの非同期転送のI/O情報)	IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
	Count	Amount	Time
Sum : 合計	-	-	-
[1,TS) : 1バイト～(TS-1)バイト (TS=最大転送長)	*<#81>*	*<#82>*	*<#83>*
[TS] : 最大転送長	*<#84>*	*<#85>*	*<#86>*

I/O 2ndLayerCache CommBuf->2ndStorageDev(Async) 第2階層ストレージのキャッシュ領域へのI/O情報 (通信用バッファから第2階層ストレージへの非同期転送のI/O情報)	IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
	Count	Amount	Time
Sum : 合計	-	-	-
[1,TS) : 1バイト～(TS-1)バイト (TS=最大転送長)	*<#87>*	*<#88>*	*<#89>*
[TS] : 最大転送長	*<#90>*	*<#91>*	*<#92>*

I/O SharedTmp NodeTotal 共有テンポラリ領域のI/O情報(サマリ)		IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum : 合計	-	-	-
	[1,4Ki) : 1バイト～(4Ki-1)バイト	*<#93>*	*<#94>*	*<#95>*
	[4Ki,1Mi) : 4Kiバイト～(1Mi-1)バイト	*<#96>*	*<#97>*	*<#98>*
	[1Mi,4Mi) : 1Miバイト～(4Mi-1)バイト	*<#99>*	*<#100>*	*<#101>*
	[4Mi+ : 4Miバイト以上	*<#102>*	*<#103>*	*<#104>*
Read(Cached I/O) O_DIRECT未指定のRead	Sum : 合計	-	-	-
	[1,4Ki) : 1バイト～(4Ki-1)バイト	*<#105>*	*<#106>*	*<#107>*
	[4Ki,1Mi) : 4Kiバイト～(1Mi-1)バイト	*<#108>*	*<#109>*	*<#110>*
	[1Mi,4Mi) : 1Miバイト～(4Mi-1)バイト	*<#111>*	*<#112>*	*<#113>*
	[4Mi+ : 4Miバイト以上	*<#114>*	*<#115>*	*<#116>*
Write(Direct I/O) O_DIRECT指定のWrite	Sum : 合計	-	-	-
	[1,4Ki) : 1バイト～(4Ki-1)バイト	*<#117>*	*<#118>*	*<#119>*
	[4Ki,1Mi) : 4Kiバイト～(1Mi-1)バイト	*<#120>*	*<#121>*	*<#122>*
	[1Mi,4Mi) : 1Miバイト～(4Mi-1)バイト	*<#123>*	*<#124>*	*<#125>*
	[4Mi+ : 4Miバイト以上	*<#126>*	*<#127>*	*<#128>*
Read(Direct I/O) O_DIRECT指定のRead	Sum : 合計	-	-	-
	[1,4Ki) : 1バイト～(4Ki-1)バイト	*<#129>*	*<#130>*	*<#131>*
	[4Ki,1Mi) : 4Kiバイト～(1Mi-1)バイト	*<#132>*	*<#133>*	*<#134>*
	[1Mi,4Mi) : 1Miバイト～(4Mi-1)バイト	*<#135>*	*<#136>*	*<#137>*
	[4Mi+ : 4Miバイト以上	*<#138>*	*<#139>*	*<#140>*

I/O SharedTmp ComputeNode<->CommBuf 共有テンポラリ領域のI/O情報(計算ノードと通信用バッファ間のI/O情報)		IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#141>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#142>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#143>*
	[4Mi+：4Miバイト以上	-	-	*<#144>*
Read(Cached I/O) O_DIRECT未指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#145>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#146>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#147>*
	[4Mi+：4Miバイト以上	-	-	*<#148>*
Write(Direct I/O) O_DIRECT指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#149>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#150>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#151>*
	[4Mi+：4Miバイト以上	-	-	*<#152>*
Read(Direct I/O) O_DIRECT指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#153>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#154>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#155>*
	[4Mi+：4Miバイト以上	-	-	*<#156>*

I/O SharedTmp CommBuf<->1stStorageDev 共有テンポラリ領域のI/O情報(通信用バッファと第1階層ストレージ間の転送のI/O情報)		IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#157>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#158>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#159>*
	[4Mi+：4Miバイト以上	-	-	*<#160>*
Read(Cached I/O) O_DIRECT未指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#161>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#162>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#163>*
	[4Mi+：4Miバイト以上	-	-	*<#164>*
Write(Direct I/O) O_DIRECT指定のWrite	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#165>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#166>*

I/O SharedTmp CommBuf<->1stStorageDev 共有テンポラリ領域のI/O情報(通信用バッファと第1階層ストレージ間の転送のI/O情報)		Iliioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#167>*
	[4Mi+:4Miバイト以上	-	-	*<#168>*
Read(Direct I/O) O_DIRECT指定のRead	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	-	-	*<#169>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	-	-	*<#170>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#171>*
	[4Mi+:4Miバイト以上	-	-	*<#172>*

I/O LocalTmp NodeTotal ノード内テンポラリ領域のI/O情報(サマリ)		Iliioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	*<#173>*	*<#174>*	*<#175>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	*<#176>*	*<#177>*	*<#178>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	*<#179>*	*<#180>*	*<#181>*
	[4Mi+:4Miバイト以上	*<#182>*	*<#183>*	*<#184>*
Read(Cached I/O) O_DIRECT未指定のRead	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	*<#185>*	*<#186>*	*<#187>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	*<#188>*	*<#189>*	*<#190>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	*<#191>*	*<#192>*	*<#193>*
	[4Mi+:4Miバイト以上	*<#194>*	*<#195>*	*<#196>*
Write(Direct I/O) O_DIRECT指定のWrite	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	*<#197>*	*<#198>*	*<#199>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	*<#200>*	*<#201>*	*<#202>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	*<#203>*	*<#204>*	*<#205>*
	[4Mi+:4Miバイト以上	*<#206>*	*<#207>*	*<#208>*
Read(Direct I/O) O_DIRECT指定のRead	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	*<#209>*	*<#210>*	*<#211>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	*<#212>*	*<#213>*	*<#214>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	*<#215>*	*<#216>*	*<#217>*
	[4Mi+:4Miバイト以上	*<#218>*	*<#219>*	*<#220>*

I/O LocalTmp ComputeNode<->CommBuf ノード内テンポラリ領域のI/O情報(計算ノードと通信用バッファ間のI/O情報)		Iliioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	-	-	*<#221>*

I/O LocalTmp ComputeNode<->CommBuf ノード内テンポラリ領域のI/O情報(計算ノードと通信用バッファ間のI/O情報)		Iliioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	-	-	*<#222>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#223>*
	[4Mi+:4Miバイト以上	-	-	*<#224>*
Read(Cached I/O) O_DIRECT未指定のRead	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	-	-	*<#225>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	-	-	*<#226>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#227>*
	[4Mi+:4Miバイト以上	-	-	*<#228>*
Write(Direct I/O) O_DIRECT指定のWrite	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	-	-	*<#229>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	-	-	*<#230>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#231>*
	[4Mi+:4Miバイト以上	-	-	*<#232>*
Read(Direct I/O) O_DIRECT指定のRead	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	-	-	*<#233>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	-	-	*<#234>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#235>*
	[4Mi+:4Miバイト以上	-	-	*<#236>*

I/O LocalTmp CommBuf<->1stStorageDev ノード内テンポラリ領域のI/O情報 (通信用バッファと第1階層ストレージ間の転送のI/O情報)		Iliioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Write(Cached I/O) O_DIRECT未指定のWrite	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	-	-	*<#237>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	-	-	*<#238>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#239>*
	[4Mi+:4Miバイト以上	-	-	*<#240>*
Read(Cached I/O) O_DIRECT未指定のRead	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	-	-	*<#241>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	-	-	*<#242>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#243>*
	[4Mi+:4Miバイト以上	-	-	*<#244>*
Write(Direct I/O) O_DIRECT指定のWrite	Sum:合計	-	-	-
	[1,4Ki):1バイト～(4Ki-1)バイト	-	-	*<#245>*
	[4Ki,1Mi):4Kiバイト～(1Mi-1)バイト	-	-	*<#246>*
	[1Mi,4Mi):1Miバイト～(4Mi-1)バイト	-	-	*<#247>*
	[4Mi+:4Miバイト以上	-	-	*<#248>*

I/O LocalTmp CommBuf<->1stStorageDev ノード内テンポラリ領域のI/O情報 (通信用バッファと第1階層ストレージ間の転送のI/O情報)		IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
		Count	Amount	Time
Read(Direct I/O) O_DIRECT指定のRead	Sum：合計	-	-	-
	[1,4Ki)：1バイト～(4Ki-1)バイト	-	-	*<#249>*
	[4Ki,1Mi)：4Kiバイト～(1Mi-1)バイト	-	-	*<#250>*
	[1Mi,4Mi)：1Miバイト～(4Mi-1)バイト	-	-	*<#251>*
	[4Mi+：4Miバイト以上	-	-	*<#252>*

Meta 2ndLayerCache NodeTotal 第2階層ストレージのキャッシュ領域のメタアクセス情報	IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)	
	Count	Time
open	*<#253>*	*<#254>*
close	*<#255>*	*<#256>*
lookup	*<#257>*	*<#258>*
mknod	*<#259>*	*<#260>*
link	*<#261>*	*<#262>*
unlink	*<#263>*	*<#264>*
mkdir	*<#265>*	*<#266>*
rmdir	*<#267>*	*<#268>*
readdir	*<#269>*	*<#270>*
rename	*<#271>*	*<#272>*
getattr	*<#273>*	*<#274>*
setattr	*<#275>*	*<#276>*
getxattr	*<#277>*	*<#278>*
setxattr	*<#279>*	*<#280>*
listxattr	*<#281>*	*<#282>*
removexattr	*<#283>*	*<#284>*
statfs	*<#285>*	*<#286>*
sync	*<#287>*	*<#288>*
lock	*<#289>*	*<#290>*
getstripe	*<#291>*	*<#292>*
setstripe	*<#293>*	*<#294>*
transfer	*<#295>*	*<#296>*

Meta SharedTmp NodeTotal 共有テンポラリ領域のメタアクセス情報	IIioinfoファイル内の出力値<#n> ("-"の項目は出力されません)	
	Count	Time
open	*<#297>*	*<#298>*
close	*<#299>*	*<#300>*

Meta SharedTmp NodeTotal 共有テンポラリ領域のメタアクセス情報	lliioinfoファイル内の出力値<#n> ("-"の項目は出力されません)	
	Count	Time
lookup	<#301>	<#302>
mknod	<#303>	<#304>
link	<#305>	<#306>
unlink	<#307>	<#308>
mkdir	<#309>	<#310>
rmdir	<#311>	<#312>
readdir	<#313>	<#314>
rename	<#315>	<#316>
getattr	<#317>	<#318>
setattr	<#319>	<#320>
getxattr	<#321>	<#322>
setxattr	<#323>	<#324>
listxattr	<#325>	<#326>
removexattr	<#327>	<#328>
statfs	<#329>	<#330>
sync	<#331>	<#332>
lock	<#333>	<#334>

Meta LocalTmp NodeTotal ノード内テンポラリ領域のメタアクセス情報	lliioinfoファイル内の出力値<#n> ("-"の項目は出力されません)	
	Count	Time
open	<#335>	<#336>
close	<#337>	<#338>
lookup	<#339>	<#340>
mknod	<#341>	<#342>
link	<#343>	<#344>
unlink	<#345>	<#346>
mkdir	<#347>	<#348>
rmdir	<#349>	<#350>
readdir	<#351>	<#352>
rename	<#353>	<#354>
getattr	<#355>	<#356>
setattr	<#357>	<#358>
getxattr	<#359>	<#360>
setxattr	<#361>	<#362>
listxattr	<#363>	<#364>
removexattr	<#365>	<#366>

Meta LocalTmp NodeTotal ノード内テンポラリ領域のメタアクセス情報	llioinfoファイル内の出力値<#n> ("-"の項目は出力されません)	
	Count	Time
statfs	*<#367>*	*<#368>*
sync	*<#369>*	*<#370>*
lock	*<#371>*	*<#372>*

Resource 2ndLayerCache CacheOperation 第2階層ストレージのキャッシュ領域のリソース情報	llioinfoファイル内の出力値<#n> ("-"の項目は出力されません)		
	Count	Amount	Time
Hit：キャッシュヒット	*<#373>*	-	-
Miss：キャッシュミス	*<#374>*	*<#375>*	*<#376>*
Shortage_wait：キャッシュ書出し待ち(キャッシュ領域不足時のキャッシュI/O時)	*<#377>*	*<#378>*	*<#379>*
Nonssd_io_wait：キャッシュ無効化(sio-read-cache=offのread時またはDirect I/O時)	*<#380>*	*<#381>*	*<#382>*
Ssd_io_wait：キャッシュ書出し待ち(sio-read-cache=onのread時またはキャッシュI/Oのwrite時)	*<#383>*	*<#384>*	*<#385>*
Invalidate：キャッシュ無効化(キャッシュ領域不足時のキャッシュI/O時)	*<#386>*	*<#387>*	*<#388>*

Resource 2ndLayerCache CacheUsage 第2階層ストレージのキャッシュ領域のUsage	llioinfoファイル内の出力値<#n> ("-"の項目は出力されません)
	Amount
MaximalUsedSpace：最大使用容量 ※llioinfoファイルでは最小空き容量が出力されます。	*<#389>*
Uncompleted-file：未書出しデータ量	*<#390>*

[注意]
LLIO性能情報に出力される未書出しデータ量は、第1階層ストレージから第2階層ストレージに書出しが完了しなかったファイルの情報です。
一方、"2.6 未書出しファイル一覧取得機能"で説明した未書出しファイル一覧には、以下の情報が出力されます。

- 計算ノード内キャッシュから第1階層ストレージに書出しが完了しなかったファイル
- 第1階層ストレージから第2階層ストレージに書出しが完了しなかったファイル

そのため、未書出しファイルの合計サイズに関しては、以下のとおりとなります。

未書出しファイル一覧での値 ≧ LLIO性能情報に出力される未書出しデータ量

Resource SharedTmp CacheUsage 共有テンポラリ領域のUsage	llioinfoファイル内の出力値<#n> ("-"の項目は出力されません)
	Amount
MaximalUsedSpace：最大使用容量	*<#391>*

Resource LocalTmp CacheUsage ノード内テンポラリ領域のUsage	llioinfoファイル内の出力値<#n> ("-"の項目は出力されません)
	Amount
MaximalUsedSpace：最大使用容量	*<#392>*

C.2 計算ノード統計情報の採取方法と出力項目

C.2.1 計算ノード統計情報の採取方法

ジョブの中で実行しているアプリケーションから計算ノードの統計情報を採取するために、getlliostatライブラリ関数を使用します。

参照

getlliostatライブラリ関数の詳細については、"A.3.1 getlliostat" を参照してください。

getlliostatライブラリ関数の書式を以下に示します。

```
#include <llio_stat.h>
int getlliostat(lliostat_t * lliostat);
```

以下にプログラム例(sample.c)を示します。

```
#include <llio_stat.h>                              ← インクルードファイルをインクルードする処理を記述する

int main(int argc, char *argv[]) {
    lliostat_t stat;
    <略>
    memset(&stat, 0, sizeof(lliostat_t));
    int stat_status = getlliostat(&stat);
    if (stat_status == 0) {
        // 性能情報出力処理
        // lliostat_t構造体のメンバを参照して出力を作成する
    } else {
        // エラー処理
    }
    <略>
}
```

以下にsample.c をコンパイルする例を示します。

```
# fcc -I/opt/FJSVllio/include -L/opt/FJSVllio/lib  -llliostat -o sample -D_GNU_SOURCE sample.c
      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  ← fcc のオプションに追加する
```

以下にジョブスクリプト例を示します。

```
#!/bin/sh

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/FJSVllio/lib
/gfs/sample -f 4m -b 4m -w /gfs/aaa
```

実行後、ジョブの標準出力ファイルに結果が出力されます。

出力内容については、プログラム例の性能情報出力処理の部分で記載した内容が出力されます。

C.2.2 計算ノード統計情報の出力項目

getlliostatライブラリが使用する構造体と列挙型を説明します。これらはllio_stat.hに定義されています。

```
enum LLIO_IOSIZERANGE {                                 // 計算ノードで採取するI/Oサイズレンジの列挙型
   LLIO_IOSIZERANGE_GE_1B,                              // 1B から 4B-1
   LLIO_IOSIZERANGE_GE_4B,                              // 4B から 16B-1
   LLIO_IOSIZERANGE_GE_16B,                             // 16B から 64B-1
   LLIO_IOSIZERANGE_GE_64B,                             // 64B から 256B-1
   LLIO_IOSIZERANGE_GE_256B,                            // 256B から 1KiB-1
   LLIO_IOSIZERANGE_GE_1KIB,                            // 1KiB から 4KiB-1
   LLIO_IOSIZERANGE_GE_4KIB,                            // 4KiB から 16KiB-1
   LLIO_IOSIZERANGE_GE_16KIB,                           // 16KiB から 64KiB-1
   LLIO_IOSIZERANGE_GE_64KIB,                           // 64KiB から 256KiB-1
   LLIO_IOSIZERANGE_GE_256KIB,                          // 256KiB から 1MiB-1
   LLIO_IOSIZERANGE_GE_1MIB,                            // 1MiB から 4MiB-1
   LLIO_IOSIZERANGE_GE_4MIB,                            // 4MiB から 16MiB-1
   LLIO_IOSIZERANGE_GE_16MIB,                           // 16MiB から 64MiB-1
   LLIO_IOSIZERANGE_GE_64MIB,                           // 64MiB から 256MiB-1
   LLIO_IOSIZERANGE_GE_256MIB,                          // 256MiB以上
   LLIO_NUM_IOSIZERANGES                                // 計算ノードで採取したI/Oサイズの種別数
};
```

```
enum LLIO_METAOPS{                                      // 第1階層ストレージへのメタアクセスの種類の列挙型
   LLIO_METAOPS_OPEN,                                   // open
   LLIO_METAOPS_CLOSE,                                  // close
   LLIO_METAOPS_LOOKUP,                                 // lookup
   LLIO_METAOPS_MKNOD,                                  // mknod
   LLIO_METAOPS_LINK,                                   // link
   LLIO_METAOPS_UNLINK,                                 // unlink
   LLIO_METAOPS_MKDIR,                                  // mkdir
   LLIO_METAOPS_RMDIR,                                  // rmdir
   LLIO_METAOPS_READDIR,                                // readdir
   LLIO_METAOPS_RENAME,                                 // rename
   LLIO_METAOPS_GETATTR,                                // getattr
   LLIO_METAOPS_SETATTR,                                // setattr
   LLIO_METAOPS_GETXATTR,                               // getxattr
   LLIO_METAOPS_SETXATTR,                               // setxattr
   LLIO_METAOPS_LISTXATTR,                              // listxattr
   LLIO_METAOPS_REMOVEXATTR,                            // removexattr
   LLIO_METAOPS_STATFS,                                 // statfs
   LLIO_METAOPS_SYNC,                                   // sync
   LLIO_METAOPS_LOCK,                                   // lock
   LLIO_METAOPS_GETSTRIPE,                              // getstripe
   LLIO_METAOPS_SETSTRIPE,                              // setstripe
   LLIO_METAOPS_TRANSFER,                               // transfer
   LLIO_NUM_METAOPS                                     // 第1階層ストレージへのメタアクセスの種別数
};
```

```
struct llio_cnstat_io {                                 // 第1階層ストレージへのI/O総量構造体
   uint64_t write_count[LLIO_NUM_IOSIZERANGES];         // Write の総回数
   uint64_t write_amount[LLIO_NUM_IOSIZERANGES];        // Write の総転送量（バイト単位）
   uint64_t write_time[LLIO_NUM_IOSIZERANGES];          // Write の総処理時間（マイクロ秒単位）
   uint64_t read_count[LLIO_NUM_IOSIZERANGES];          // Read の総回数
   uint64_t read_amount[LLIO_NUM_IOSIZERANGES];         // Read の総転送量（バイト単位）
   uint64_t read_time[LLIO_NUM_IOSIZERANGES];           // Read の総処理時間（マイクロ秒単位）
};
```

```
enum LLIO_IO_TYPE {                                     // 第1階層ストレージへのI/Oタイプ列挙型
   LLIO_IO_CACHED,                                      // キャッシュI/O
   LLIO_IO_DIRECT,                                      // DirectI/O
   LLIO_NUM_IO_TYPES                                    // I/Oタイプの種別数
};
```

```
struct llio_cnstat_meta {                               // 第1階層ストレージのメタ情報構造体
   uint64_t meta_count[LLIO_NUM_METAOPS];               // メタアクセスの総回数
```

```
   uint64_t meta_time[LLIO_NUM_METAOPS];                    // 総処理時間（マイクロ秒単位）
};
```

```
struct llio_cnstat_cache {                                  // 第1階層ストレージのキャッシュ情報構造体
   uint64_t cache_hit_count;                                // キャッシュヒット回数（累計）※1
   uint64_t cache_miss_count;                               // キャッシュミス回数（累計）※2
   uint64_t cache_wait_count;                               // キャッシュ空き待ち回数（累計）
   uint64_t cache_wait_amount;              // キャッシュの空き容量不足のために追い出したデータ量(累計，バイト単位)
   uint64_t cache_wait_time;                                // キャッシュ空き待ち時間の累計（マイクロ秒単位）
   uint64_t cache_free;                                     // 現時点での、キャッシュの未割当て領域量(バイト単位)
   uint64_t cache_used;                                     // 現時点での、キャッシュの使用領域量(バイト単位)
   uint64_t cache_dirty;                                    // 現時点での、キャッシュの未書出し領域量(バイト単位)
};
```

制限事項:上記 ※1、※2 は、mmap(2)でメモリマッピングしたファイルについては採取されません。

```
struct llio_cnstat_fs {                                     // 第1階層ストレージへの全I/O情報構造体
   struct llio_cnstat_io io[LLIO_NUM_IO_TYPES];             // 第1階層ストレージへのI/O総量構造体
   struct llio_cnstat_meta meta;                            // 第1階層ストレージのメタ情報構造体
   struct llio_cnstat_cache cache;                          // 第1階層ストレージのキャッシュ情報構造体
};
```

```
enum LLIO_FS_TYPE {                                         // 第1階層ストレージの利用形態の列挙型
   LLIO_GFS_CACHE,                                          // 第2階層ストレージのキャッシュ
   LLIO_SHARED_TEMP,                                        // 共有テンポラリ領域
   LLIO_LOCAL_TEMP,                                         // ノード内テンポラリ領域
   LLIO_NUM_FS_TYPES                                        // 第1階層ストレージの利用形態の種別数
};
```

```
struct lliostat {                                           // lliostat 構造体
   struct llio_cnstat_fs fs[LLIO_NUM_FS_TYPES];             // 第1階層ストレージ全I/O情報構造体
};
#define lliostat_t struct lliostat
extern int getlliostat(lliostat_t *);
```

C.3 システム統計情報の出力項目

C.3.1 ストレージI/Oノード向けシステム統計情報の出力項目

ストレージI/Oノード向けシステム統計情報の出力項目を以下に示します。

表C.2 システム統計情報の種別

種別	説明
j	ジョブごとのI/O 情報
c	通信層のコネクション数に関する情報
r	ジョブごとのリクエスト情報

表C.3 I/O情報の項目

項目名と例	説明
<利用形態>-<I/O 箇所>-[Direct]<I/O 種別>(<サイズ>)-<I/O 項目> 【例】Cache-CN-SIO-RD(4K)-Amount	<利用形態>用途の<I/O 箇所>におけるDirect/CachedI/O の<I/O 種別>の<サイズ>ごとの<I/O 項目> ※Direct I/O の場合は<I/O箇所>の後にDirect が付加される。

表C.4 メタアクセス情報の項目

項目名と例	説明
<利用形態>-<メタアクセス種別>-<メタアクセス項目> 【例】Share-close-Count	<利用形態>用途の<メタアクセス種別>ごとの<メタアクセス項目>

表C.5 リソース情報の項目

項目名と例	説明
<利用形態>-<リソース情報項目> 【例】Local-MinFreeSpace	<利用形態>用途の<リソース情報項目>

表C.6 利用形態

略称	説明
Cache	第2 階層ストレージのキャッシュ
Share	共有テンポラリ
Local	ノード内テンポラリ

表C.7 I/O箇所の略称と意味

略称	意味
Total	全体のI/O
CN-CBCN	計算ノードと通信用バッファ間のI/O 情報
CBCN-Dev	通信用バッファと第1 階層ストレージデバイス間の転送のI/O 情報(計算ノードとの通信)
CBGFS-Dev	通信用バッファと第1 階層ストレージデバイス間の転送のI/O 情報(第2 階層ストレージとの通信)
CBCN-GFS	通信用バッファと第2 階層ストレージ間の転送のI/O 情報(DirectI/O)
CBGFS-GFS	通信用バッファと第2 階層ストレージ間の転送のI/O 情報(CachedI/O)

表C.8 I/O略称

略称	説明
RD	Read
WR	Write

表C.9 I/Oサイズの略称

略称	説明
4K-	1B ～ 4KiB-1B
1M-	4KiB ～ 1MiB-1B
4M-	1MiB ～ 4MiB-1B
4M+	4MiB ～
TrSz-	1B ～ <転送サイズ>-1B
TrSz+	<転送サイズ>B

表C.10 I/O項目の略称

略称	説明
Count	総回数
Amount	総データ量
Time	総処理時間

表C.11 メタアクセス種別

略称	メタアクセス
open	open
close	close
lookup	lookup
mknod	mknod
link	link
unlink	unlink
mkdir	mkdir
rmdir	rmdir
readdir	readdir
rename	rename
getattr	getattr
setattr	setattr
getxattr	getxattr
setxattr	setxattr
listxattr	listxattr
removexattr	removexattr
statfs	statfs
sync	sync
lock	lock
getstripe	getstripe
setstripe	setstripe
transfer	transfer
allocate	allocate
purge	purge

表C.12 メタアクセス項目

略称	メタアクセス
Count	総回数
Time	総処理時間

表C.13 リソース情報項目

略称	説明
CacheHitCount	キャッシュヒット回数
CacheMissCount	キャッシュミス回数
CacheMissAmount	キャッシュミスデータ量
CacheMissTime	キャッシュミス時間
CacheShortageWaitCount	キャッシュの空き容量不足が原因で、キャッシュ追出しが発生した回数の合計
CacheShortageWaitAmount	キャッシュの空き容量不足が原因で、キャッシュ追い出ししたデータ量の合計
CacheShortageWaitTime	キャッシュの空き容量不足が原因で、キャッシュ追出しに要した時間の合計

略称	説明
CacheNonSSDIOWaitCount	古いキャッシュデータが存在するブロックに対してSSDを利用しないI/Oを行ったことが原因で、キャッシュ追出しが発生した回数の合計
CacheNonSSDIOWaitAmount	古いキャッシュデータが存在するブロックに対してSSDを利用しないI/Oを行ったことが原因で、キャッシュ追い出ししたデータ量の合計
CacheNonSSDIOWaitTime	古いキャッシュデータが存在するブロックに対してSSDを利用しないI/Oを行ったことが原因で、キャッシュ追出しに要した時間の合計
CacheSSDIOWaitCount	古いキャッシュデータが存在するブロックに対してSSDを利用するI/Oを行ったことが原因で、キャッシュ追出しに要した回数の合計
CacheSSDIOWaitAmount	古いキャッシュデータが存在するブロックに対してSSDを利用するI/Oを行ったことが原因で、キャッシュ追出しに要したデータ量の合計
CacheSSDIOWaitTime	古いキャッシュデータが存在するブロックに対してSSDを利用するI/Oを行ったことが原因で、キャッシュ追出しに要した時間の合計
CacheInvalCount	キャッシュ無効化回数
CacheInvalAmount	キャッシュ無効化データ量
CacheInvalTime	キャッシュ無効化時間
MaximalUsage	最大使用容量
CacheUnflushedAmount	未書出しデータ量

システム統計情報種別が"c"の場合は、通信層のコネクション数に関する情報であることを示します。出力項目は"表C.14 コネクション数項目の略称"で示す当該ノードのコネクションの接続数と切断数です。

表C.14 コネクション数項目の略称

略称	説明
Conn	現在接続済のコネクション数
InProConn	現在接続中のコネクション数
InProDisconn	現在切断中のコネクション数
MaxConn	接続済、接続中、切断中のコネクション数の合計の過去最大値
AccumConn	過去の新規接続数の累計
AccumDisconn	過去の切断数の累計

表C.15 リクエスト項目の略称

略称	説明
<リクエスト種別>-Wait-Num	待ち時間の記録回数
<リクエスト種別>-Wait-Min	待ち時間の最小値
<リクエスト種別>-Wait-Max	待ち時間の最大値
<リクエスト種別>-Wait-Sum	待ち時間の合計値
<リクエスト種別>-Qdepth-Num	キューの長さの記録回数
<リクエスト種別>-Qdepth-Min	キューの長さの最小値
<リクエスト種別>-Qdepth-Max	キューの長さの最大値
<リクエスト種別>-Qdepth-Sum	キューの長さの合計値
<リクエスト種別>-Active-Num	同時リクエスト処理数の記録回数
<リクエスト種別>-Active-Min	同時リクエスト処理数の最小値
<リクエスト種別>-Active-Max	同時リクエスト処理数の最大値

略称	説明
<リクエスト種別>-Active-Sum	同時リクエスト処理数の合計値

<リクエスト種別>にはIO、または Meta が入ります。

C.3.2 グローバルI/Oノード向けシステム統計情報の出力項目

グローバルI/Oノード向けシステム統計情報の出力項目を以下に示します。

表C.16 データ転送回数とデータ量項目の略称

略称	意味
TransferCount	LNET 層転送回数
TransferAmount	LNET 層転送量(Byte)

表C.17 転送バッファ項目の略称

略称	意味
MinFreeBufZero	ゼロバイト転送用バッファの最小空き数
MinBufSend	SEND転送用バッファの最小空き数
MinFreeBufRDMA	RDMA転送用バッファの最小空き数

用語集

以下の用語に加えて、マニュアル「ジョブ運用ソフトウェア 用語集」および「FEFS ユーザーズガイド」の用語集を参照してください。

インターコネクト

ノード間でプログラムの計算データ、入出力データを転送するための高速なネットワークです。TofuインターコネクトDや InfiniBand があります。

階層化ストレージ

第1階層ストレージと第2階層ストレージの2階層の構成のことです。

共通ファイル

第2階層ストレージ上にある実行ファイルや設定ファイルなど、ジョブ開始時にすべての計算ノードから読み込まれるファイルのことです。アクセス集中による性能劣化を避けるため、第2階層ストレージのキャッシュ領域に共通ファイルをコピーしておくことができます (共通ファイル配布機能)。

第1階層ストレージ

LLIOが管理するジョブ専用の高速一時領域のことです。

第2階層ストレージ

FEFSが管理するファイルシステムのことです。

ローリングアップデート

パッケージ適用で、システムまたはクラスタ全体を停止せず、一部の計算ノードでクラスタ内のジョブ運用を継続しながら部分的な保守をすることです。

BoB (Bunch of Blades)

FXサーバの制御単位。16ノードで構成されます。

FEFS (Fujitsu Exabyte File System)

富士通が開発した並列分散ファイルシステムです。

LNET

Ethernet、InfiniBandなど複数の異なるネットワークを共通に利用してファイルシステムにアクセスするための機能です。

LLIO (Lightweight Layered IO-Accelerator)

FEFSと計算ノードの間に SSD を使用したストレージ階層を設け、FEFSのキャッシュ領域やジョブの一時領域として使用することで高性能を実現する技術、またこれによって実現されるファイルシステムを指すこともあります。

SIOグループ

計算クラスタの中で、同じストレージI/Oノードを利用する計算ノード群です。

TofuインターコネクトD

FXサーバにおけるノード同士を接続する高速ネットワーク。計算ノード同士をつなぐネットワークをジョブ運用ソフトウェアでは、計算用ネットワークと呼びます。