

---

# 富岳 高速転送ガイド

リリース 1.3

**RIKEN R-CCS**

2023 年 08 月 02 日

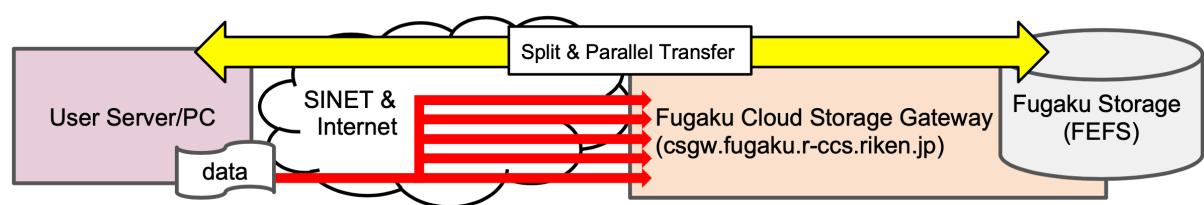
# 目次:

<b>第 1 章</b>	<b>はじめに</b>	1
1.1	クラウドストレージゲートウェイについて . . . . .	1
1.2	表記について . . . . .	2
1.3	更新履歴 . . . . .	2
<b>第 2 章</b>	<b>GridFTP について</b>	3
2.1	GridFTP サーバの設定について . . . . .	3
<b>第 3 章</b>	<b>GridFTP クライアントソフトを用いたデータ転送方法</b>	4
3.1	globus-url-copy コマンド . . . . .	4
3.2	uberftp コマンド . . . . .	13
<b>第 4 章</b>	<b>その他の並列転送方法</b>	18
4.1	mscp コマンド . . . . .	18
4.2	BCCP コマンド . . . . .	18

# 第1章

## はじめに

本ドキュメントは「富岳」環境へのデータ転送を高速に行う方法について説明します。GridFTP を用いて複数並列で、ご利用者様のサーバや端末と富岳間でデータを転送する方法を説明します。長時間・多並列でのデータ転送については、クラウドストレージゲートウェイ ([csgw.fugaku.r-ccs.riken.jp](http://csgw.fugaku.r-ccs.riken.jp)) のご利用をお願いいたします。



### 1.1 クラウドストレージゲートウェイについて

クラウドストレージゲートウェイは、富岳と外部(クラウドや共用ストレージ、ご利用者様のサーバや端末)間のデータ転送などを利用するためのサーバです。クラウドストレージゲートウェイからは富岳のログインノードと同様に、富岳グローバルストレージへのアクセスが可能です。クラウドストレージゲートウェイへのログイン方法は以下です。SSH 公開鍵は富岳ログインノードと同様に設定されます。

#### ■ ssh コマンドでのログイン

```
$ ssh csgw.fugaku.r-ccs.riken.jp
```

#### ■ gsissh コマンドでのログイン

```
$ gsissh -p 2222 csgw.fugaku.r-ccs.riken.jp
```

上記の通りログインすると、以下の 2 サーバに分散されます。

- [csgw1.fugaku.r-ccs.riken.jp](http://csgw1.fugaku.r-ccs.riken.jp)
- [csgw2.fugaku.r-ccs.riken.jp](http://csgw2.fugaku.r-ccs.riken.jp)

## 1.2 表記について

ご利用者様の端末 (User) とクラウドストレージゲートウェイ (csgw.fugaku.r-ccs.riken.jp) のどこでコマンド実行が行われるのかを明確にするため、本書では以下のプロンプト表記を使用します。

プロンプト	利用端末
[User]\$	ご利用者様の端末やサーバでのコマンド実行
[CSGW]\$	クラウドストレージゲートウェイでのコマンド実行

- 利用者 PC で複数の CLI コンソールを利用する場合は「User\_1」、「User\_2」などと番号で識別します。

## 1.3 更新履歴

### 第 4 版 (2023 年 8 月 2 日)

globus-url-copy の実行に関するタイプの修正  
mscp の説明を追加

### 第 3 版 (2023 年 6 月 16 日)

globus-url-copy および uberftp のインストール方法を修正  
誤記を修正

### 第 2 版 (2021 年 7 月 5 日)

パフォーマンス情報を追加  
NAT 配下での利用方法を追加  
globus-url-copy コマンドの並列転送方法やオプションについて追加

### 初版 (2021 年 5 月 24 日)

---

Copyritht (c) 2021 理化学研究所計算科学研究センター  
本マニュアルに記載されている内容の無断転載・複製を禁じます。

## 第2章

# GridFTPについて

TCP通信による並列転送可能なデータ転送ソフトウェア(プロトコル)です。大きなファイルも分割して並列転送しますので、scp等より高速に転送できることが期待できます。クラウドストレージゲートウェイ(csgw.fugaku.r-ccs.riken.jp)では、GridFTPサーバが動作しています。GridFTPクライアントソフトを用いて、ご利用者様のサーバや端末等と富岳のストレージ間で並列でデータ転送できます。

### 2.1 GridFTPサーバの設定について

クラウドストレージゲートウェイ(csgw.fugaku.r-ccs.riken.jp)ではGridFTPサーバの利用向けに以下のポートを開放しています。GridFTPで利用されるデフォルトのポートが開放されています。GridFTPサーバではSSH公開鍵認証とGSI認証を許可しています。ご利用しやすい認証を方式を選択してください。ご利用されるクライアント環境でOUTBOUNDのFirewallなどが設定されている場合、以下のポートをご利用いただけるようポート開放をお願いいたします。

Port番号	プロトコル	用途
22	tcp	アクセスポート/GridFTPサーバとの認証でSSH公開鍵を用いるのに利用
2811	tcp	アクセスポート/GridFTPサーバとの認証でGSI認証を用いるのに利用
50000 - 51000	tcp	データ転送用ポート

## 第3章

# GridFTP クライアントソフトを用いたデータ転送方法

### 3.1 globus-url-copy コマンド

本項では Globus が提供している GridFTP 転送プロトコルを用いたデータ転送ツール"globus-url-copy"コマンドを利用してデータを転送する方法を説明します。globus-url-copy コマンドでは、インタラクティブでの操作は提供していませんが、UberFTP Clinet に比べ、ファイル単位の並列転送やバッファサイズの指定等のオプションが提供されています。globus-url-copy コマンドの詳細については、以下 URL をご参照ください。

- <https://gridcf.org/gct-docs/6.2/gridftp/user/index.html#gridftp-user-command>

また NAT 環境では一部のオプションを指定すると読み出しの際に、データ転送ができない問題があります (version 10.8 で確認)。本問題の回避策を「*NAT 環境下や Firewall 配下でのデータ転送時の注意点*」に記載しています。NAT 環境でご利用される場合はご注意ください。

#### 3.1.1 globus-url-copy コマンドのインストール方法

globus-url-copy コマンドは以下の方法でインストールできます。また Windows や Mac をご利用の場合は、HPCI から Docker コンテナが提供されていますので、こちらをご利用いただくことで簡単に環境を構築できます。

##### RHEL/CentOS

epel リポジトリを利用してインストールができます。(以下の例では GSI 代理証明書をダウンロードするための myproxy パッケージも合わせてインストールしています)

##### RHEL/CetOS 8 系の例

```
(User) $ sudo dnf localinstall https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
(User) $ sudo dnf install --enablerepo=epel globus-gass-copy-progs myproxy
```

##### RHEL/CetOS 7 系の例

```
(User)$ sudo yum localinstall https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm -y
(User)$ sudo yum install --enablerepo=epel globus-gass-copy-progs myproxy
```

## Ubuntu

```
(User)$ sudo apt-get install globus-gass-copy-progs myproxy
```

## Mac/Windows

Mac および Widnows から利用する方法としては、WSL 等の Linux 仮想環境を利用する方法と、Docker コンテナを利用する方法で簡単に環境をセットアップすることができます。

### (1) WSL/Lima 等の Linux 仮想環境を利用する

Windows では WSL を用いて Linux の仮想環境を利用することができます。powershell で以下を実行することで、Ubuntu の仮想環境が用意されます。

```
(User@Windows)$ wsl --install
```

WSL の詳細な利用方法は以下等をご参照ください。

- <https://learn.microsoft.com/ja-jp/windows/wsl/install>
- <https://www.hpci.nii.ac.jp/gt6/docker/HPCI-Login-noDesktop-win10.html.ja>

Mac では Lima などの環境を利用して Linux の仮想環境を利用することができます。以下は Lima を利用する手順です。

```
(User)$ brew install lima
(User)$ limactl start default
(User)$ limactl list
NAME      STATUS      SSH          VMTYPE      ARCH      CPUS      MEMORY
DISK      DIR
default   Running    127.0.0.1:60022  qemu       aarch64   4         4GiB
100GiB   ~/.lima/default
(User)$ limactl shell default
(User@lima-default):<path>$ sudo apt update
(User@lima-default):<path>$ sudo apt install myproxy globus-gass-copy-progs
```

homebrew や Lima については以下を参照ください。

- brew: [https://brew.sh/index\\_ja](https://brew.sh/index_ja)
- Lima: <https://github.com/lima-vm/lima>

### (2) HPCI Globus 環境向け Docker コンテナを利用する

以下の通り HPCI より Globus 環境が整備された Docker コンテナが提供されています。こちらを用いることで簡単に globus-url-copy コマンドの利用環境を構築できます。

- Mac 版: <https://www.hpci.nii.ac.jp/gt6/docker/HPCI-Login-Docker-macOS.html>
- Windows 版: <https://www.hpci.nii.ac.jp/gt6/docker/HPCI-Login-Docker-win10.html>

上記 Docker コンテナを起動後、yum コマンドで globus-gass-copy-progs パッケージをインストールしてください。(以下は gsi.openssh-20201215 版の例です。docker run のオプションなどはご利用の環境に合わせて設定してください)

```
(User)$ docker load -i gsi.openssh-20201215.tar.bz2
(User)$ docker run -d --rm --name gsi.openssh hpci/gsi.openssh:20201215
(User)$ docker exec -i -t gsi.openssh /bin/bash
[hpciuuser@docker_container]$ sudo yum install globus-gass-copy-progs
```

### 3.1.2 globus-url-copy コマンドを用いた転送方法

globus-url-copy コマンドを利用して、SSH 公開鍵認証、GSI 認証を利用して転送する方法を説明します。globus-url-copy コマンドは次のようなフォーマットで転送元のデータと転送先のパス (URL) を指定します。

```
(User)$ globus-url-copy <OPTION> <転送元 URL> <転送先 URL>
```

転送先・転送元 URL は「<プロトコル>://<user 名>@<ホスト名 (IP アドレス)>:<ポート番号>/<ファイルパス>」の様な形で記載します。プロトコルは、クラウドストレージゲートウェイでは、sshftp(SSH 認証で転送) と gsiftp(GSI 認証で転送) がサポートされています。例えば SSH 認証でクラウドストレージゲートウェイへ ~/file.txt を転送する例は以下です。

- ローカルのファイルやディレクトリを指定する場合はプロトコルに file:// と指定することもできます。以下の例では file:// は省略しています。
- sshftp の場合は認証にポート番号 22, gsiftp の場合はポート番号は 2811 がデフォルトのポート番号として利用されます。これらも省略が可能です。
- ディレクトリを再帰的に転送する場合は -r オプションを指定する必要があります。その他のオプションは後述の転送オプションの項を参照ください。

```
(User)$ globus-url-copy ~/file.txt sshftp://<user>@csgw.fugaku.r-ccs.riken.jp/~/  
→file.txt
```

#### globus-url-copy コマンドの転送オプション

globus-url-copy では並列数や転送バッファサイズを指定できます。転送距離や転送帯域などによって、変更することでより高速に転送が可能になります。また、チェックサム比較などもオプションも用意されています。必要に応じてオプションの指定、指定値の調整をご検討ください。以下は転送によく利用するオプションです。詳細は globus-url-copy のマニュアルをご参照ください。

オプション	内容
-r	ディレクトリを転送する際に指定します。ディレクトリに保存されたファイル・ディレクトリを再起的に転送します
-cd / -create-directory	転送先に指定したディレクトリがない場合に自動でディレクトリを作成します。
-v / -verbose	Verbose モードです。転送時に詳細な情報を表示します
-p / -parallel	1つのファイルを分割して転送する際の並列数を指定できます、このオプションは NAT 環境でファイルの読み出しを行う際にデータ転送に失敗する可能性があります。もし失敗するようでしたら、NAT 環境下での回避策または本オプションではなく -cc オプションでのファイルごとの並列転送をご利用ください。
-cc / -concurrency	複数のファイルを並列して転送する際に指定します。このオプションは -p と異なり NAT 環境でのファイルの読み出しにも影響がございません。
-tcp-bs	ftp 転送チャンネルが利用する転送時のバッファサイズを指定できます
-bs	転送方法別に利用される転送バッファサイズを指定できます
-verify-checksum	転送元・転送先でチェックサムを取得して比較します(異なる場合は転送失敗)

## SSH 公開鍵認証を用いた転送方法

SSH 公開鍵認証を用いるには、クラウドストレージゲートウェイ (csgw.fugaku.r-ccs.riken.jp) に SSH 公開鍵を設定してください。SSH 公開鍵の設定については、富岳のスタートアップガイドをご参照ください。

- 富岳マニュアルページ: <https://www.fugaku.r-ccs.riken.jp/docs/manuals>

以下の例は、SSH 公開鍵認証で test.d ディレクトリに保存されたファイルを富岳ホーム領域(グローバル・ファイルシステム)に保存する例です。

- <FUGAKU\_USERNAME>は富岳のアカウントを指定してください。
- <User>はご利用者様の端末のユーザ名です。
- 転送先にディレクトリがないと転送に失敗します。転送前に予め転送用のディレクトリを作成してください。以下の例では富岳ホーム領域に./test.d を作成しています。

### ■ 富岳グローバルファイルシステムへの書き込み

```
(User)$ ls ~/test.d
test.01  test.02  test.03  test.04  test.05  test.06  test.07  test.08  test.09 ↘
→test.10
(User)$ globus-url-copy -cd -r -p 8 -v ~/test.d/ sshftp://<FUGAKU_USERNAME>@csgw.
→fugaku.r-ccs.riken.jp:22/~/test.d/
Source: file:///home/<User>/test.d/
Dest:   sshftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp:22/~/test.d/
       test.01
Source: file:///home/<User>/test.d/
Dest:   sshftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp:22/~/test.d/
       test.02
(snip)
Dest:   sshftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp:22/~/test.d/
       test.10
```

(次のページに続く)

(前のページからの続き)

**■ 確認**

```
(User) $ ssh <FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp "ls test.d/"
test.01
test.02
test.03
test.04
test.05
test.06
test.07
test.08
test.09
test.10
(User) $
```

以下は、SSH 公開鍵認証で富岳のホーム領域(グローバル・ファイルシステム)のデータを手元のサーバに保存する例です。転送ポートを指定するため GLOBUS\_TCP\_PORT\_RANGE を指定してください。本例では 50000~51000 を利用して転送されます。本例では-v オプション(Verbose)と-r オプション(Recurive)、-p オプション(Parallel)を指定しています。

- <FUGAKU\_USERNAME>は富岳のアカウントを指定してください。
- <User>はご利用者様の端末のユーザ名です。

**■ 環境変数設定(転送ポート設定)**

```
(User) $ export GLOBUS_TCP_PORT_RANGE=50000,51000
■ 富岳グローバルファイルシステムからの読み出し
(User) $ globus-url-copy -cd -r -p 8 -v sshftp://<FUGAKU_USERNAME>@csgw.fugaku.r-
→ccs.riken.jp:22/~/test.d/ ~/test.d/
Source: sshftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp:22/~/test.d/
Dest:   file:///home/<User>/test.d/
        test.01
Source: sshftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp:22/~/test.d/
Dest:   file:///home/<User>/test.d/
        test.02
(snip)
Source: sshftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp:22/~/test.d/
Dest:   file:///home/<User>/test.d/
        test.10
■ 確認
(User) $ ls ~/test.d
test.01  test.02  test.03  test.04  test.05  test.06  test.07  test.08  test.09 ↵
→test.10
(User) $
```

## GSI 認証を用いた転送方法

HPCI が提供している証明書を用いて、GSI 認証での並列転送を行うこともできます。GSI 認証の詳細については、HPCI クイックスタートガイドをご参照ください。

- HPCI マニュアルページ: [https://www.hpci-office.jp/pages/hpci\\_info\\_manuals](https://www.hpci-office.jp/pages/hpci_info_manuals)

以下の例は、GSI 認証で test.d ディレクトリに保存されたファイルを富岳ホーム領域（グローバル・ファイルシステム）に保存する例です。

- HPCI 代理証明書を取得しておいてください。
- <HPCI-ID>はご利用者の HPCI-ID を指定してください。
- <FUGAKU\_USERNAME>は富岳のアカウントを指定してください。
- <User>はご利用者様の端末のユーザ名です。
- 転送先にディレクトリがないと転送に失敗します。転送前に予め転送用のディレクトリを作成してください。以下の例では富岳ホーム領域に./test.d を作成しています。

### ■ 代理証明書のダウンロード

```
(User) $ myproxy-logon -s portal.hpci.nii.ac.jp -l <HPCI-ID> -t 168
Enter MyProxy pass phrase:
(User) $ grid-proxy-info | grep timelife
timelife : 13:50:08
■ 富岳グローバルファイルシステムへの書き込み
(User) $ ls ~/test.d
test.01  test.02  test.03  test.04  test.05  test.06  test.07  test.08  test.09  ↵
  ↵test.10
(User) $ globus-url-copy -cd -r -p 8 -v ~/test.d/ gsiftp://<FUGAKU_USERNAME>@csgw.
  ↵fugaku.r-ccs.riken.jp/~/test.d/
Source: file:///home/<User>/test.d/
Dest:   gsiftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp/~/test.d/
       test.01
Source: file:///home/<User>/test.d/
Dest:   gsiftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp/~/test.d/
       test.02
(snip)
Dest:   gsiftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp/~/test.d/
       test.10
■ 確認
```

```
(User) $ gsiss -p2222 <FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp "ls test.d/"
test.01
test.02
test.03
test.04
test.05
test.06
test.07
test.08
test.09
test.10
(User) $
```

以下は、GSI 認証で富岳のホーム領域（グローバル・ファイルシステム）のデータを手元のサーバに保存する例です。転送ポートを指定するため GLOBUS\_TCP\_PORT\_RANGE を指定してください。本例では 50000~51000 を利用して転送されます。本例では -v オプション (Verbose) と -r オプション (Recursive)、-p オプション (Parallel) を指定しています。

- <FUGAKU\_USERNAME>は富岳のアカウントを指定してください。
- <User>はご利用者様の端末のユーザ名です。

#### ■ 代理証明書のダウンロード

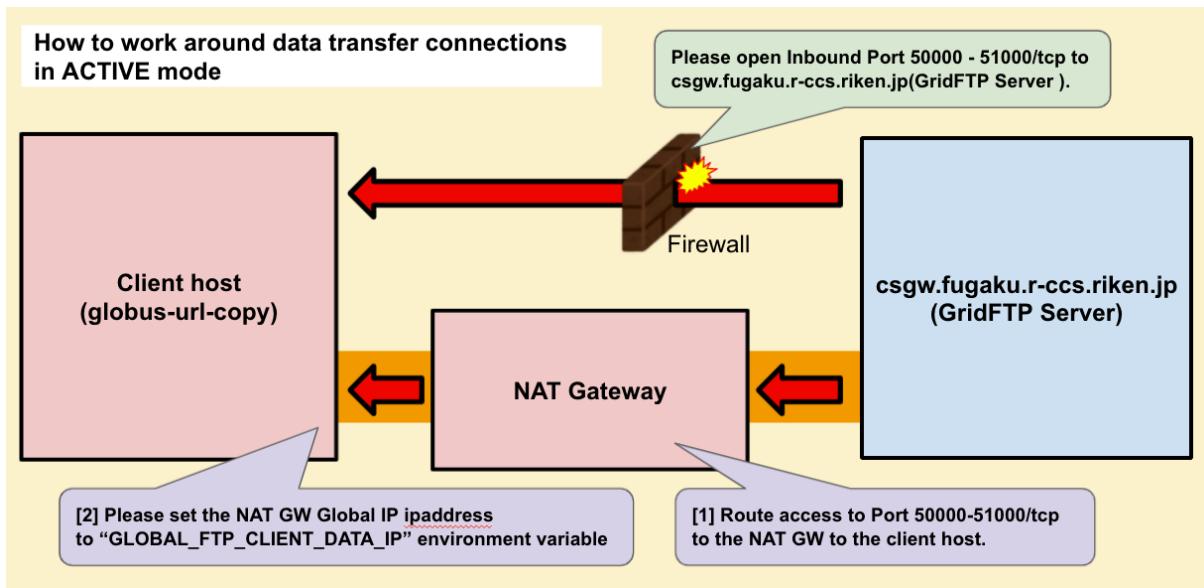
```
(User) $ myproxy-logon -s portal.hpci.nii.ac.jp -l <HPCI-ID> -t 168
Enter MyProxy pass phrase:
(User) $ grid-proxy-info | grep timelife
timelife : 13:50:08
■ 環境変数設定（転送ポート設定）
(User) $ export GLOBUS_TCP_PORT_RANGE=50000,51000
■ 富岳グローバルファイルシステムからの読み出し
(User) $ globus-url-copy -cd -r -p 8 -v gsiftp://<FUGAKU_USERNAME>@csgw.fugaku.r-
→ccs.riken.jp/~/test.d/ ~/test.d/
Source: gsiftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp/~/test.d/
Dest:   file:///home/<User>/test.d/
       test.01
Source: gsiftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp/~/test.d/
Dest:   file:///home/<User>/test.d/
       test.02
(snip)
Source: gsiftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp/~/test.d/
Dest:   file:///home/<User>/test.d/
       test.10
■ 確認
(User) $ ls ~/test.d
test.01  test.02  test.03  test.04  test.05  test.06  test.07  test.08  test.09 ↵
→test.10
(User) $
```

### 3.1.3 NAT 環境下や Firewall 配下でのデータ転送時の注意点

globus-url-copy では、通常 PASSIVE モード（クライアントから GridFTP サーバへデータ転送用のコネクション張る）でデータを転送します。しかしながら、"-p" オプションなどの一部のオプションを指定した際に、ACTIVE モード（GridFTP サーバ側からクライアント環境へデータ転送コネクションを張る）での転送方法が利用されてしまい、NAT 環境下や Firewall 設定環境下では、GridFTP サーバからクライアント（globus-url-copy 実行環境）へのデータ転送コネクションが張れず、データの読み出しを行う際に転送に失敗します。

回避策として、Firewall が原因の場合は Inbound ポートの開放、NAT 環境下ではルーティングや IP マスカレードなどでデータを転送するよう NAT サーバやルータ設定をしていただく必要があります。

ご利用の環境によってはこのような回避策が設定いただけない可能性があります。設定いただけない場合につきましては、"-p" オプションを利用した同一ファイルの分割転送オプションの指定はせず、-cc オプションの複数ファイルを並列転送するオプションを用い、転送速度を上げていただくようお願いいたします。



### Firewall 環境でのファイル読み出し方法

図の通り、csgw.fugaku.r-ccs.riken.jp からクライアントホストへの Inbound Port 50000 - 51000/tcp を開放いただくようお願いいたします。

### NAT 環境でのファイル読み出し方法

NAT 配下への転送を行う場合、NAT サーバやルータからクライアント環境へのルーティング設定と、globus-url-copy コマンド実行前の環境変数の設定が必要です。

1. NAT サーバやルータで、csgw.fugaku.r-ccs.riken.jp からの Port 50000 - 51000/tcp のアクセスをクライアントホストに転送するよう、ルーティングや IP マスカレード設定を実施（ご利用の NAT サーバやルータに合わせて設定してください）
2. globus-url-copy コマンド実行ホストに GLOBUS\_FTP\_CLIENT\_DATA\_IP 環境変数を設定し、globus-url-copy コマンドを実行

```
(User) $ export GLOBUS_FTP_CLIENT_DATA_IP=<NAT_GW_IPADDRESS>
(User) $ globus-url-copy -r -p 8 -v ~/test.d/ sshftp://<FUGAKU_USERNAME>@csgw.fugaku.r-ccs.riken.jp:22/~/test.d/
```

### 3.1.4 パフォーマンス

HPCI 共用ストレージの R-CCS ログインノード das04.r-ccs.riken.jp から csgw1.fugaku.r-ccs.riken.jp にデータ転送した際のパフォーマンスです。

- 計測は 1GB x 128File の合計 128GB で行なっています。
- das04(client) から csgw への転送は 100Gbps で接続されています。

- das04(client) の書き込み元データ及び読み出し先のデータの保存先は/dev/shm(on memory) を指定しています。
- 後述の通り転送先/元の FEFS 領域は/data 以下を利用してお、Strip の設定値はデフォルト (Strip Count:1 / BlockSize 1MB) のままでです。

ユーザの利用状況によって転送速度は変化しますが、-p(ファイルを分割して転送) と -cc(ファイルを並列して転送) オプションの双方を指定することで効率よく転送が可能です。-p オプションが利用できない NAT 環境等の場合も、-cc オプションを増やすことで転送帯域が向上いたします。本結果を参考にし並列数やバッファサイズ、ブロックサイズなどをご調整ください。

### 書き込み計測結果

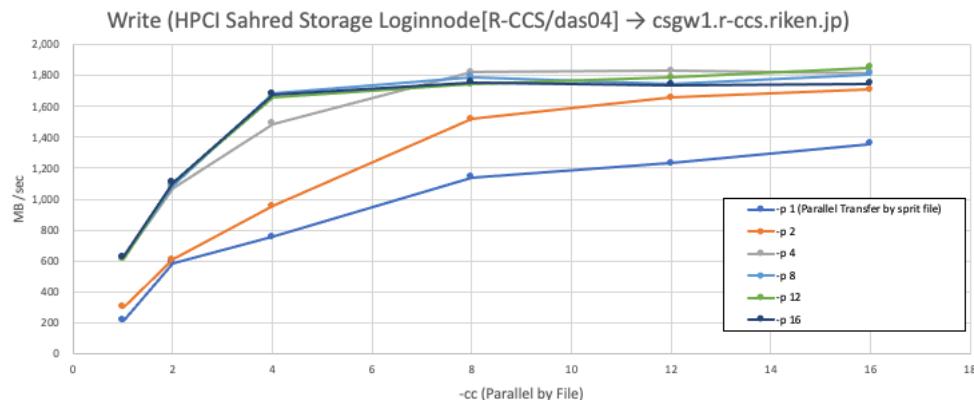
#### 書き込み計測コマンド

```
(das04)$ time -p globus-url-copy -bs 1048576 -rst -cd -v \
    -p <分割転送数> -cc <並列転送数> -r \
    /dev/shm/test.d/
    sshftp://${USER}@csgw1.r-ccs.riken.jp/data/.../test.d/
```

### 書き込み計測結果

- p オプション: 1 ファイルを分割して並列転送と、その分割並列数
- cc オプション: 複数のファイルを並列で転送と、その並列数

MB/sec	cc=1	cc=2	cc=4	cc=8	cc=12	cc=16
p=1	212.635	584.555	756.054	1141.804	1231.796	1356.747
p=2	300.066	609.572	957.033	1516.767	1657.745	1709.173
p=4	616.461	1072.805	1486.126	1822.583	1828.920	1816.118
p=8	619.445	1087.819	1681.040	1791.713	1743.078	1808.335
p=12	611.991	1106.979	1656.386	1743.632	1785.133	1853.819
p=16	626.898	1105.322	1679.496	1756.231	1738.264	1749.032



## 読み出し計測結果

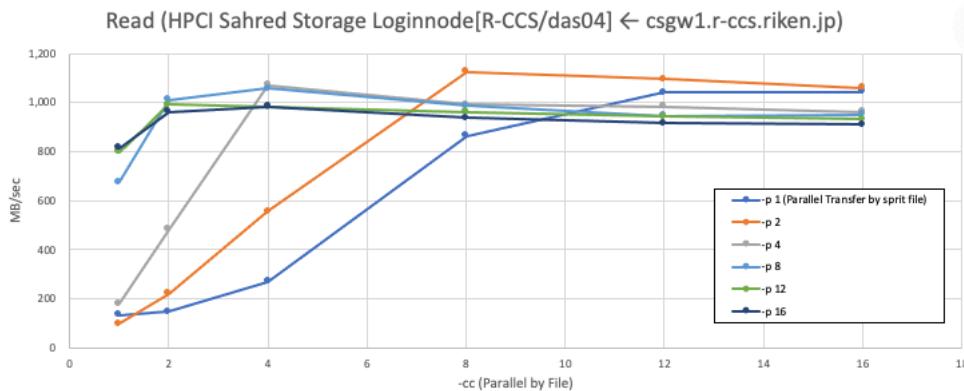
### 読み出し計測コマンド

```
(das04)$ time -p globus-url-copy -bs 1048576 -rst -cd -v \
-p <分割転送数> -cc <並列転送数> -r \
sshftp://${USER}@csgw1.fugaku.r-ccs.riken.jp/data/.../test.d/
/dev/shm/test.d/
```

## 読み出し計測結果

- p オプション: 1 ファイルを分割して並列転送と、その分割並列数
- cc オプション: 複数のファイルを並列で転送と、その並列数

MB/sec	cc=1	cc=2	cc=4	cc=8	cc=12	cc=16
p=1	132.999	146.944	269.853	865.235	1041.158	1043.748
p=2	97.617	220.283	555.353	1124.899	1095.365	1060.744
p=4	179.808	483.512	1072.161	992.748	985.108	962.816
p=8	676.819	1012.378	1058.026	986.360	945.766	952.168
p=12	797.677	991.971	981.670	962.927	943.619	933.135
p=16	813.773	961.779	984.691	939.013	916.993	911.356



## 3.2 uberftp コマンド

本項では、GSI 認証を用いてインタラクティブに利用できる GridFTP 転送プロトコルを用いたデータ転送ツール"uberftp"コマンドでデータを転送する方法を説明します。uberftp は 2023 年時点ですでに開発が中止されていますので、可能な限り globus-url-copy をご利用ください。

uberftp は globus-url-copy コマンドと異なり、対話型でも利用可能です。本説明では主に対話型での利用方法について説明します。

uberftp コマンドでは、SSH 認証には対応していません。GSI 認証をご利用ください。またファイルの分割転送は可能ですが、ファイルの並列転送ができません (version 2.8)。globus-url-copy と同様に、NAT 環境でファイルの読み出しなどを行う際に、分割転送に失敗する可能性がございますのでご注意ください。もし、

NAT 環境でファイルの読み出しを並列で行いたい場合は、globus-url-copy 同様の回避策をご利用いただくか、globus-url-copy コマンドのファイル並列オプション (cc オプション) をご利用いただくようお願いいたします。

### 3.2.1 uberftp コマンドのインストール方法

uberftp コマンドは以下の方法でインストールできます。また Windows や Mac をご利用の場合は、HPCI から Docker コンテナが提供されています。こちらにインストールすることで GSI 環境が簡単にご利用できます。

#### ■ RHEL/CentOS

epel リポジトリを利用してインストールができます。(以下の例では GSI 代理証明書をダウンロードするための myproxy パッケージも合わせてインストールしています)

#### RHEL/CetOS 8 系の例

```
(User)$ sudo dnf localinstall https://dl.fedoraproject.org/pub/epel/epel-release-
↪latest-8.noarch.rpm
(User)$ sudo dnf install --enablerepo=epel uberftp myproxy
```

#### RHEL/CetOS 7 系の例

```
(User)$ sudo yum localinstall https://dl.fedoraproject.org/pub/epel/epel-release-
↪latest-7.noarch.rpm -y
(User)$ sudo yum install --enablerepo=epel uberftp myproxy
```

#### ■ Ubuntu

※ Ubuntu 20.23 にはパッケージインストールする方法がございません。

#### ■ Mac/Windows

##### (1) WSL/LIMA を利用する

Widows では WSL、Mac では LIMA 等を利用することで、Linux 環境を構築することができます。利用方法については、globus-url-copy のインストール方法をご参照ください。

UberFTP は Ubuntu 版のパッケージがないため、必要に応じて RHEL 系 (CentOS や Almalinux など) をご利用ください。

起動後は以下の通り、RHEL/CentOS のインストールと同じように dnf または yum を利用して epel リポジトリから uberftp をインストールしてください。

```
(User@WSL or LIMA)$ sudo dnf localinstall https://dl.fedoraproject.org/pub/epel/
↪epel-release-latest-8.noarch.rpm
(User@WSL or LIMA)$ sudo dnf install --enablerepo=epel uberftp myproxy
```

##### (2) HPCI 提供の Docker コンテナを利用する

以下の通り HPCI より Globus 環境が整備された Docker コンテナが提供されています。こちらを用いることで簡単に globus-url-copy コマンドの利用環境を構築できます。

- Mac 版: <https://www.hpci.nii.ac.jp/gt6/docker/HPCI-Login-Docker-macOS.html>
  - Windows 版: <https://www.hpci.nii.ac.jp/gt6/docker/HPCI-Login-Docker-win10.html>

上記 Docker コンテナを起動後、yum コマンドで `uberftp` パッケージをインストールしてください。(以下は `gsi-openssh-20201215` 版の例です。`docker run` のオプションなどはご利用の環境に合わせて設定してください)

```
(User)$ docker load -i gsi-openssh-20201215.tar.bz2
(User)$ docker run -d --rm --name gsi-openssh hpc1/gsi-openssh:20201215
(User)$ docker exec -i -t gsi-openssh /bin/bash
[hpc1user@docker_container]$ sudo yum install uberftp
```

### 3.2.2 `uberftp` コマンドを用いた転送方法

`uberftp` コマンドを利用してインタラクティブに利用して転送する方法を説明します。認証は GSI 認証になります。本例では、ローカルから富岳のグローバル・ファイルシステムにデータを書き込み、その後、富岳のグローバル・ファイルシステムからローカルにデータを読み出す例です。

- ・<FUGAKU\_USERNAME>は富岳のアカウントを指定してください。
  - ・他のFTPクライアントと同様に先頭に"I"をつけるとローカルを対象にしたコマンドになります。

#### ■ 代理証明書のダウンロード

```
(User)$ myproxy-logon -s portal.hpci.nii.ac.jp -l <HPCI-ID> -t 168  
Enter MyProxy pass phrase:  
(User)$ grid-proxy-info | grep timyleft  
timyleft : 13:50:08
```

■ `uberftp` コマンドでクラウドストレージゲートウェイにアクセス

```
(User)$ uberftp -u <FUGAKU_USERNAME> csgw.fugaku.r-ccs.riken.jp  
220 sm02sv11 GridFTP Server 13.21 (gcc64, 1607703417-0) [Grid Community Toolkit 6.  
→2] ready.  
230 User <FUGAKU_USERNAME> logged in.
```

ÜberFTP (2.8) >

```
■ ローカルの転送元ディレクトリを確認

UberFTP (2.8)> lcd ./test.d
UberFTP (2.8)> lls
drwxrwxr-x  2 <user> <group>          200 May 23 19:15 .
drwx----- 72 <user> <group>        16384 May 23 19:15 ..
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.01
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.02
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.03
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.04
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.05
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.06
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.07
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.08
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.09
-rw-rw-r--  1 <user> <group> 104857600 May 23 19:43 test.10

UberFTP (2.8)> lcd ..
```

CET①80-212(緒)

(前のページからの続き)

```
/home/<user>/test.d
■ クラウドストレージゲートウェイの src.dir ディレクトリに転送
UberFTP (2.8)> mkdir src.dir
UberFTP (2.8)> cd src.dir
UberFTP (2.8)> pwd
/home/<FUGAKU_GROUP>/<FUGAKU_USER>/src.dir
UberFTP (2.8)> mput * ./
test.01: 104857600 bytes in 0.078164 Seconds (1.249 GB/s)
test.02: 104857600 bytes in 0.062107 Seconds (1.572 GB/s)
test.03: 104857600 bytes in 0.057231 Seconds (1.706 GB/s)
test.04: 104857600 bytes in 0.057512 Seconds (1.698 GB/s)
test.05: 104857600 bytes in 0.059141 Seconds (1.651 GB/s)
test.06: 104857600 bytes in 0.064720 Seconds (1.509 GB/s)
test.07: 104857600 bytes in 0.066412 Seconds (1.470 GB/s)
test.08: 104857600 bytes in 0.069008 Seconds (1.415 GB/s)
test.09: 104857600 bytes in 0.069565 Seconds (1.404 GB/s)
test.10: 104857600 bytes in 0.068166 Seconds (1.433 GB/s)
UberFTP (2.8)> ls
drwxr-xr-x 12 <FUGAKU_USER> <FUGAKU_GROUP> 4096 May 23 20:42 .
drwx----- 14 <FUGAKU_USER> <FUGAKU_GROUP> 4096 May 23 20:42 ..
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.01
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.02
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.03
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.04
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.05
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.06
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.07
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.08
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.09
-rw-r--r-- 1 <FUGAKU_USER> <FUGAKU_GROUP> 104857600 May 23 19:43 test.10
■ クラウドストレージゲートウェイの src.dir のデータを、ローカルの dest ディレクトリに転送
UberFTP (2.8)> lmkdir ~/dest.dir
UberFTP (2.8)> lcd ~/dest.dir
UberFTP (2.8)> lpwd
/home/<user>/dest.dir
UberFTP (2.8)> pwd
/home/<FUGAKU_GROUP>/<FUGAKU_USER>/src.dir
UberFTP (2.8)> mget * ./
test.01: 104857600 bytes in 0.300731 Seconds (332.523 MB/s)
test.02: 104857600 bytes in 0.395436 Seconds (252.885 MB/s)
test.03: 104857600 bytes in 0.292810 Seconds (341.518 MB/s)
test.04: 104857600 bytes in 0.295335 Seconds (338.599 MB/s)
test.05: 104857600 bytes in 0.296257 Seconds (337.545 MB/s)
test.06: 104857600 bytes in 0.299359 Seconds (334.047 MB/s)
test.07: 104857600 bytes in 0.331494 Seconds (301.665 MB/s)
test.08: 104857600 bytes in 0.300032 Seconds (333.298 MB/s)
test.09: 104857600 bytes in 0.294818 Seconds (339.192 MB/s)
test.10: 104857600 bytes in 0.314017 Seconds (318.454 MB/s)
UberFTP (2.8)> ll
total 1024020
drwx----- 2 <user> <group> 200 May 23 20:50 .
```

(次のページに続く)

(前のページからの続き)

```
drwx----- 73 <user> <group> 16384 May 23 20:45 ..
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.01
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.02
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.03
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.04
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.05
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.06
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.07
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.08
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.09
-rw----- 1 <user> <group> 104857600 May 23 19:43 test.10
UberFTP (2.8)>
```

## 第4章

# その他の並列転送方法

### 4.1 mscp コマンド

mscp コマンドは、指定したファイル・ディレクトリを複数の ssh(SFTP) 接続でコピーするコマンドです。

mscp: <https://github.com/upa/mscp>

mscp コマンドは、標準の openssh-server などが稼働しているホストに対して、通常の scp と同じように実行することで並列・分割転送が利用できます。

#### 4.1.1 mscp コマンドのインストール方法

Mac および Linux へのインストール手順は [github](#) をご参照ください。Windows では WSL を使って構築した Linux 環境などの上で mscp をインストールしてご利用ください。mscp: <https://github.com/upa/mscp>

##### ■ 富岳ログインノードへのインストール

Linux バイナリ版を利用できます。

```
(csgw1)$ wget https://github.com/upa/mscp/releases/latest/download/mscp.linux.x86.  
→static -O ./mscp  
(csgw1)$ chmod 755 ./mscp  
(csgw1)$ ./mscp ./directory <hostname>:~/
```

### 4.2 BBCP コマンド

BBCP コマンドは高速化・安全にデータを転送するコマンドです。本項では BBCP コマンドのインストール方法と、富岳ログインノードへのデータ転送の方法を説明します。

- BBCP を実行するためには転送元と転送先で BBCP が利用可能 (PATH が通っている状態) である必要があります。
- 通信には SSH が利用されるため、転送前に転送元・先間で SSH アクセスができるか確認してください。

BBCPについては、現在 snapd によるインストールまたは spack によるインストール方法を記載しております。ARM 環境(例えば Mac)などでは Source build が必要になるため、こちらについては本マニュアルではサポートしません。

BBCP: <https://www.slac.stanford.edu/~abh/bbcp/>

### 4.2.1 BBCP コマンドのインストール方法

BBCP コマンドは以下の方法でインストールできます。富岳ログインノードで利用する場合は、以下の通り Spack を利用してインストールを実施してください。

#### ■ 富岳ログインノードへのインストール

spack のバージョン v0.17 でのインストール例です。新しいバージョンが出ている場合は、新しいバージョンを利用してインストールしてください。

##### (1) spack リポジトリをチェックアウト

```
csgw1$ cd ~
csgw1$ git clone https://github.com/spack/spack.git
Cloning into 'spack'...
remote: Enumerating objects: 389791, done.
remote: Counting objects: 100% (86/86), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 389791 (delta 26), reused 65 (delta 14), pack-reused 389705
Receiving objects: 100% (389791/389791), 178.18 MiB | 12.03 MiB/s, done.
Resolving deltas: 100% (167901/167901), done.
Updating files: 100% (9513/9513), done.
csgw1$ cd spack
csgw1$ git checkout releases/v0.17
Updating files: 100% (8028/8028), done.
Branch 'releases/v0.17' set up to track remote branch 'releases/v0.17' from 'origin
→'.
Switched to a new branch 'releases/v0.17'
csgw1$
```

##### (2) コンパイラを確認

gcc@8.4.1 が存在することを確認

```
csgw1$ . ~/spack/share/spack/setup-env.sh
csgw1$ spack compiler find
==> Found no new compilers
==> Compilers are defined in the following files:
/home/<group>/<username>/.spack/linux/compilers.yaml
csgw1$ spack compilers
==> Available compilers
-- gcc_rhel8-x86_64 -----
gcc@11.2.0^^c2^^a0 gcc@8.4.1
csgw1$
```

## (3) gettext コマンドをインストール

ログインノードとストレージゲートウェイには、bbcp のビルドに必要になる libntl ライブラリが存在していません。このため、Spack で gettext をビルドして libntl ライブラリを利用できるようにします。

- bbcp コマンドのビルドには libntl ライブラリが必要です
- 通常 libntl ライブラリは gettext パッケージに含まれていますが、RHEL の gettext パッケージには libntl ライブラリが含まれていません。(Spack では gettext パッケージがインストールされていると libntl ライブラリが存在していると判定しているのですが、RHEL の gettext パッケージには libntl が含まれていないため、Spack で libntl ライブラリが依存関係が含まれるようなソフトウェアのビルドの際にはエラーとなります)
- このため、Spack で gettext をビルドします(これにより libntl が含まれた gettext がインストールされます)

```
csgw1$ spack install gettext %gcc@8.4.1 target=x86_64
(snip)
[+] /vol0004/(Group)/(User)/spack/opt/spack/linux-rhel8-x86_64/gcc-8.4.1/gettext-0.
→21-5dwjdvq....
```

## (4) BBCP コマンドをインストール

ビルドした gettext をロードし BBCP コマンドをインストールします。

```
csgw1$ spack load gettext | grep x86_64
==> Error: gettext matches multiple packages.
Matching packages:
seu57rr gettext@0.19.8.1%fj@4.7.0 arch=linux-rhel8-a64fx
hi2wxq5 gettext@0.19.8.1%fj@4.7.0 arch=linux-rhel8-a64fx
cbhtryb gettext@0.19.8.1%fj@4.8.0 arch=linux-rhel8-a64fx
ykn373u gettext@0.19.8.1%gcc@11.2.0 arch=linux-rhel8-cascadelake
5dwjdvq gettext@0.21%gcc@8.4.1 arch=linux-rhel8-x86_64
Use a more specific spec.
csgw1$ spack load /5dwjdvq
csgw1$ spack load --list
==> 7 loaded packages
-- linux-rhel8-skylake_avx512 / gcc@8.4.1 -----
bzip2@1.0.6^^c2^^a0 libxml2@2.9.7^^c2^^a0 ncurses@6.1^^c2^^a0 xz@5.2.4

-- linux-rhel8-x86_64 / gcc@8.4.1 -----
gettext@0.21^^c2^^a0 libiconv@1.16^^c2^^a0 tar@1.34
csgw1$
```

BBCP コマンドのリポジトリをチェックアウトしてビルドします。

```
csgw1$ git clone https://www.slac.stanford.edu/~abh/bbcp/bbcp.git bbcp
Cloning into 'bbcp'...
Fetching objects: 705, done.
csgw1$
```

(次のページに続く)

(前のページからの続き)

```
csgw1$ spack dev-build bbcp@master ^gettext@0.21 %gcc@8.4.1 arch=linux-rhel8-x86_64
(snip)
[+] /vol0004/...../spack/opt/spack/linux-rhel8-skylake_avx512/gcc-8.4.1/bbcp-
→master-ckx5ixewcv5goj6jyaiuxni2eu6357fi
```

### (5) BBCP コマンドをロード

csgw で bbcp を利用できるようにするには、以下のように bash で spack のパスを通す必要があります。

```
csgw1$ spack load bbcp
csgw1$ bbcp --help
Usage: ^c2^a0 bbcp [Options] [Inspec] Outspec

Options: [-a [dir]] [-A] [-b [+bf] [-B bsz] [-c [lvl]] [-C cfn] [-D] [-d path]
[-e] [-E csa] [-f] [-F] [-g] [-h] [-i idfn] [-I slfn] [-k] [-K]
[-L opts[@logurl]] [-l logf] [-m mode] [-n] [-N nio] [-o] [-O] [-p]
[-P sec] [-r] [-R [args]] [-q qos] [-s snum] [-S srcxeq] [-T trgxeq]
[-t sec] [-v] [-V] [-u loc] [-U wsz] [-w [=]wsz] [-x rate] [-y] [-z]
[-Z pnf[:pn1]] [-4 [loc]] [-~] [-@ {copy|follow|ignore}] [-$] [-#] [--]

I/Ospec: [user@] [host:] file

Function: Secure and fast copy utility.
(snip)
csgw1$ vim ~/.bashrc
. ~/spack/share/spack/setup-env.sh
spack load bbcp
csgw1$ bash
```

## ■ Linux

基本的にソースコードの build が必要になります。または snapd を利用することでインストールすることができます。※: snapd の bbcp Arm 版は存在していないようです

```
(User)$ sudo dnf localinstall https://dl.fedoraproject.org/pub/epel/epel-release-
→latest-8.noarch.rpm
(User)$ sudo dnf install --enablerepo=epel globus-gass-copy-progs myproxy
```

### (B) snapd を利用する場合

snapd はクロスプラットフォームで利用可能なパッケージ管理ツールです。Ubuntu 等ではデフォルトでインストールされています。CentOS 等は snapd のインストールが必要です。snapd のインストール方法などは以下の参照してください。snapd: <https://snapcraft.io/docs/installing-snapd>

```
(User)$ sudo snap install bbcp
```

## ■ Mac/Windows

Mac および Widnows から利用する場合は、WSL 等の Linux 仮想環境を用いて環境を構築してください。Arm 版 Mac については snapd 対応の Arm 版 BBCP が 2023 年 06 月時点では存在していないため、ソースインス

トールが必要です。WSL および Lima 等の Linux 仮想環境については「globus-url-copy のインストール」のページを参考にしてください。

## BBCP の利用方法

詳細な利用方法については以下をご参照ください。BBCP: <https://www.slac.stanford.edu/~abh/bbcn/>

通信には SSH が利用されるため、転送前に転送元・先間で SSH アクセスができるることを確認してください。富岳 (csgw.fugaku.r-ccs.riken.jp) へ転送する場合、bbcpc を事前に spack で構築しておく必要があります。環境の構築方法は前述のインストール方法をご参照ください。

BBCP コマンドの基本的な利用方法は以下です。

```
(User)$ bbcpc local-file <User>@csgw.fugaku.r-ccs.riken.jp:/home/<group>/<user>/<保存先ディレクトリ>
```

多数のファイルを転送する場合は、再帰的な転送をする方法とファイルリストを作成して転送する方法があります。

以下は再帰的に転送する例です。-A はディレクトリを作成するオプション、-r は再帰的に転送するためのオプション、-P <N> は N 秒ごとに実行状況を表示するオプション -s <N> は N 並列で転送するオプションです。.. code-block:: shell-session

```
(User)$ bbcpa -A -r -P 1 -s 4 local-file <User>@csgw.fugaku.r-ccs.riken.jp:/home/<group>/<user>/<保存先ディレクトリ> bbcp: Indexing files to be copied... bbcp: Copying 24 files and 0 links in 1 directory plus 1 empty directory. bbcp: Creating /home/<group>/<user>/<保存先ディレクトリ> bbcp: 230619 17:27:03 95% done; 105.7 MB/s bbcp: 230619 17:27:03 95% done; 105.6 MB/s (snip)
```

-I オプションを指定することでファイルリストに指定されたファイルを並列転送することができます。以下はファイルリストを作成して転送する例です。

```
(User)$ cat filelist.txt
/home/.../testfile00
/home/.../testfile01
/home/.../testfile02
(snip)
/home/.../testfile99
(User)$ bbcpa -A -r -I filelist.txt -P 1 -s 4 <User>@csgw.fugaku.r-ccs.riken.jp:/
→home/<group>/<user>/<保存先ディレクトリ>
```