



大規模AIクラウド計算システム「ABCI」 初めてのABCI

2020年11月11日 国立研究開発法人 産業技術総合研究所 情報・人間工学領域 ディジタルアーキテクチャ推進センター

クイックレファレンス https://abci.ai/ja/how_to_use/data/ABCI_referrence.html



目 次

<事前準備>

- 1. PC (Windows, Mac)へLinux, Python環境を構築 (Jupyter Notebook)
- 2. WindowsPCの場合 WSL (Windows Subsystem for Linux)によるLinux環境構築
- 3. 初回ログインとパスワード
- 4. 公開鍵の登録@利用ポータル
- 5. インタラクティブノードへのログイン (ABCI利用の入口)

<ハンズオン>

- 1. ABCI利用の概念的理解
- 2. 計算ノードにおける基本操作(Linux)
 - ジョブの投入 (qsub, qrsh)
 - クラウドストレージの利用
- 3. ABCI環境構築
 - ① Python仮想環境の構築
 - 2 モジュールコマンド:
 ABCIにインストール済のソフトウェアを利用
 例: python, cuda, cudnn, singularity
 - ③ インストールコマンド:
 利用したいソフトウェアをインストール
 例: TensorFlow, Jupyter Notebook
 - ④ コンテナイメージの利用:
 ソフトウェア環境をそのまま利用
 例: digits
- 4. ABCI利用ポータル
- 5. ABCIサービス諸元





PC (Windows, Mac)へLinux, Python環境を構築

Jupyter Notebookを利用する



Anaconda3をインストール



- Anacondaは、PythonやJupyter Notebook等のパッケージを提供するプラットフォーム
- 下記のリンクからAnaconda3のインストーラをダウンロード https://repo.continuum.io/archive/

*頻繁にアップグレードされているので、新しいバージョンを選択

<MacOS>

Anaconda3-2020.07-MacOSX-x86_64.pkg

<Windows>
Anaconda3-2020.07-Windows-x86.exe
*64bit版OSの場合は
Anaconda3-2020.07-Windows-x86_64.exe

• インストーラを実行

*「インストール先」は、「自分専用(Just Me)」とすること。 *インストール先のパスは「デフォルト」がお勧め(パスに日本語があるとうまく動作しない場合がある)。 *Windowsでは、「Register Anaconda as my default Python 3.8」にチェックを入れるのがお勧め。



Anaconda Navigatorを起動して Jupyter Notebookをインストール



Anaconda Navigatorを起動 ٠ <MacOS> <Windows> スタート -> Anaconda3 -> Anaconda Navigator Finder -> アプリケーション -> Anaconda-Navigator Anaconda Navigator **ANACONDA** NAVIGATOR Upgrade Now Sign in to Anaconda Cloud A Home Refresh Applications on python3 Channels Environments ۵ ۵ ¢ Projects (beta) lab Jupyter notebook glueviz jupyterlab Learning 0.12.4 0.31.12 5.4.0 Multidimensional data visualization across An extensible environment for interactive Web-based, interactive computing notebook files. Explore relationships within and among and reproducible computing, based on the environment. Edit and run human-readable Community related datasets. Jupyter Notebook and Architecture. docs while describing the data analysis. Jupyter Notebookの 「Install」をクリック Install Install ۵ ¢ ※「Launch」と表示されてい る場合は次ページ(P.9)へ進む Documentation rstudio Developer Blog orange3 atconsole 4.3.1 1.1.423 3.11.0 Feedback PyQt GUI that supports inline figures, proper A set of integrated tools designed to help Component based data mining framework. Data visualization and data analysis for multiline editing with syntax highlighting, you be more productive with R. Includes R novice and expert. Interactive workflows graphical calltips, and more. essentials and notebooks. with a large toolbox



Jupyter Notebookを起動









Jupyter Notebook起動時にこのような画面が出たら

🗂 Jupyter

Password or token: Log in

Invalid credentials

(2) 一度パスワードを登録しておけば、 登録したパスワードを入力 (毎回、トークンをコピペする必要なし)

(1) ここに、トークンとパスワードを入力

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you enable a password.

The command:

jupyter notebook list

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

Currently running servers: http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks

or you can paste just the token value into the password field on this page.

See <u>the documentation on how to enable a password</u> in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

Token

New Password

Log in and set new password





Jupyter Notebookのブラウザが自動で立ち上がらない場合

 Jupyter Notebookを起動してもブラウザが自動で立ち上がらない場合は、 ターミナルに出力されたURLをブラウザにコピペする。

To access the notebook, open this file in a browser: (Jupyter NotebookのURIが表示される) Or copy and paste one of these URLs: http://localhost:8888/?token=xxxxxx or http://127.0.0.1:8888/?token=xxxxxx

- インストールしたanaconda3のJupyter Notebookに再接続(「Logout」ボタンでログアウト 後に「login page」のリンクをクリックした場合等)した場合に、トークンを入力する画面 (P.10)が表示される場合がある。そのような使用を行う場合、事前に以下の手順でトークン を取得しておく。
 - 1. P.13の「Jupyter Notebookのターミナルを使う」の手順で、ターミナル画面を表示する。
 - 2. ターミナルで以下のコマンドを入力する。

jupyter notebook list

3. 以下のような結果が出力された場合、token=XX..XXの"XX..XX"の部分がトークンである。

Currently running servers: http://localhost:8888/?token=XX...XX ::







Jupyter ホーム画面

	C Home Page - Select or create a X +			
$\leftarrow \rightarrow$	C i localhost:8888/tree	🕶 🖈 🞯 🖬 🗯 🧓	Jupyter ターミナル画面	
アプリ	リ 🥝 JALカード - ログ 🍥 ログインページへ / 😒 【確定申告書等作 🧭 確定申告ソフト	、「 » 📄 その他のブックマーク	Home Page - Select or create a X >_ localhost:8888/terminals/1 X +	
	💭 Jupyter	Quit Logout	$\leftrightarrow \rightarrow \mathbb{C}$ () localhost:8888/terminals/1	🖈 🕝 🖬 🏚 :
			🏭 アプリ 🙆 JALカード‐ログ 🌔 ログインページへ/ 🚱 【確定申告書等作 🧬 確定申告ソフト「	» 🗎 その他のブックマーク
	Files Running Clusters		💭 Jupyter	Logout
	Select items to perform actions on them.	Upload New -	(hose) herichischerichischer Doch Durch herichisch	
	0 - Name -	Votebook:	(base) nagisnimakouteninomaeBook-Pro:~ nagisnimaş	
	ABCI.zip	Other		
	□ □ abci101	Text File		
	baboi_test	Folder		
		Terminal		
	Applications	2年前		
	🗆 🗅 build	2年前		
	Desktop	19時間前		
	New -> Terminal をクリック			





Jupyter NotebookのPythonを使う

Jupyter ホーム画面

	C Home Page - Select or create a X +		
$\leftrightarrow \rightarrow$	C (i) localhost:8888/tree	🕶 🖈 🚱 📼 🗯 🧓 🗄	Jupyter のPython画面
アプリ	② JALカード - ログ… ● ログインページへ /… ③ 【確定申告書等作… ◎ 確定申告ソフ ·	7ト「 » 🗎 その他のブックマーク	- 〇 〇 〇 M 受信ト x 🚋 Anacor x 👩 Pythor x 📕 Linux x 🔿 Anacor x 🤨 初心者 x 🔿 Home x 🧧 Untitle x 🕂
	C jupyter	Quit Logout	← → C ③ localhost:8889/notebooks/Untitled7.ipynb?kernel_name=python3 ☆ ③ □ ★ ⊕ :
			🛄 アプリ 🥝 JALカード - ログ 🌘 ログインページへ / 🤡 【確定申告書等作 🗭 確定申告ソフト「 » 🗎 その他のブックマーク
	Files Running Clusters		C Jupyter Untitled7 (unsaved changes)
	Select items to perform actions on them.	Upload New - 2	File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 O
	0 - Nam	Notebook:	
	ABCI.zip	Other	
	□ □ abci101	Text File	In []:
	abci_test	Folder	
		Terminal	
	C Applications	2年前	
		2年前	
	Desktop	19時間前	
	New -> Python3 をクリック		





Jupyter Notebookを終了させる

- Jupyter Notebookを起動した時に立ち上がったターミナルで、 「Control-C」を打つ。
- 以下のように聞かれるので、「y」を打つ。

The Jupyter Notebook is running at: http://localhost:8888/ Shutdown this notebook server (y/[n])?

(Windowsの場合)

Jupyter Notebookの画面右上の「Quit」をクリックし、以下のメッセージが表示されたらブラウザのタブを閉じる

Server stopped

×

You have shut down Jupyter. You can now close this tab. To use Jupyter again, you will need to relaunch it.





WSL (Windows Subsystem for Linux)によるLinux環境構築

- WSLを使えば、 WindowsPCでもターミナルでLinuxコマンドが使えるようになります。
- WSL (Windows Subsystem for Linux)によるLinux環境構築方法については、下記のサイト等をご参照ください。

https://qiita.com/tedkuma/items/01e5cfa09347176f4d1a





初回ログインURL、ABCIアカウント、仮パスワード をメール受信したら、、

ABCI利用ポータルへの 初回ログインとパスワード登録



1. 初回ログインURLへアクセス パスワードは、「仮パスワード」を入力
ABCIアカウント名 パスワード ログイン
2. 利用規定への同意(右) 「同意する」をクリック後、 「全てを同意して次へ進む」をクリック
利用規定への同意 下記を参照し、同意お願いします。同意いただけない場合ABCIをご利用できませんので、ご理解お願いいたします。 産総研外の利用者の方 産総研内の利用者の方 ○同意する ●同意しない
以下のセキュリティ上の遺守項目を読み、同意いただけたら「全てを同意して次へ進む」で次に進んでください。 利用責任者、利用管理者、利用者(以後、利用者等という)は、研究所から提供されるABCI利用に関するアカウント及びアカウントのパス ワードを研究所の承諾なく第三者に開示してはならず、かつ、第三者に推測されないように適切に設定し、管理しなければなりません。 利用者等は、利用者等のデータ等がいかなる法令にも違反していないことを表明及び保証し、利用者等のデータ等の開発、内容、運用、維持 及び利用につき、責任を負います。(利用者等のデータ等のセキュリティ及びパックアップ) 利用者等は、ABCIを適正に利用し、利用者等のデータ等について、セキュリティを確保し保護すること、及び定期的に保存することを含 め、適切なセキュリティ及び保護を行うことを誓約します。(安全保障輸出管理関係法令の遵守) 利用者等は、次の名号に該当する行為を行ってはなりません。 - 約款・規約及び回答書に記載されている事項に違反する行為 二 申請書に記載した利用目的以外にABCIを利用する行為 三 研究所若しくは第三者の著作権・商標権等の知的財産権を侵害する行為又はそのおそれがある行為 四 研究所者しくは第三者の常作権・商標権等の知的財産権を侵害する行為又はそのおそれがある行為 五 ABCIポイントを含めた研究所の電子情報を改ざん又は消去する行為 六 ウイルス等の有者なコンビュータブログラム等を開発する行為 七 研究所のネットワークやインターネット網、それらに接続されたサーバ設備等に不正にアクセスする行為 八 ABCI利用サービスの提供を妨害する行為又はが高するおそれのある行為

DAIST

3. パスワード変更

パスワード変更						
以下のフォームにパスワー	「をご入力の上、「変更」ボタンをクリックしてください。					
現在のパスワード						
新しいパスワード						
新しいパスワード (確認)						
パスワード規約						
 15以上の文字をランダ」 ません。文字をランダ」 ことを推奨します。 	ムに並べた文字列を指定してください。例えばLinux の辞書に登録されている単語は使用で ムに選ぶ方法として、パスワード作成用のソフトウェアを用いるなどして、自動的に生成す					
 現在のパスワードと異なる文字列を指定してください。 						
• 英小文字、英大文字、数字、特殊文字の4種類全てを含む文字列を指定してください。						
• 使用可能な特殊文字は以下の通りです。						
空白、!、"、#、\$、%、&、'、(、)、*、+、,、-、.、/、:、;、<、=、>、?、@、[、¥、]、^、、`、						
	+/					
● 王内乂子は使用できまし	1/V.					

+ 他の利用者や第三者に著しく迷惑をかけ、又は社会的に許されないような行為

→ AIST 初期アクセスURLがタイムアウトになってしまったら、 事前準備 、 、







ABCI利用開始前に公開鍵を登録する(必須)





公開鍵・秘密鍵ペアの生成

1)ターミナルを開き、公開鍵・秘密鍵のペアを生成する(ssh-rsaの例)。 <u>「xxx@yyy.co.jp</u>」は、自分のメルアドと置き換える!

yourpc\$ install -m 0700 -d ~/.ssh yourpc\$ cd ~/.ssh yourpc\$ ssh-keygen -t rsa -b 4096 -C "xxx@yyy.co.jp" -f ~/.ssh/id_rsa Generating public/private rsa key pair. Enter passphrase (empty for no passphrase): <-- パスフレーズを入力(任意) Enter same passphrase again: <-- もう一度、パスフレーズを入力(任意) Your identification has been saved in id_rsa. Your public key has been saved in id_rsa.pub. yourpc\$ ls ~/.ssh id_rsa id_rsa.pub

- * id_rsa (秘密鍵)・・・ クライアントPCに配置 * id_rsa.pub (公開鍵)・・・ ABCIに登録
- 2) ターミナルから公開鍵を参照する(ssh-rsaの例)。

* 公開鍵は架空のものです。

yourpc\$ cat ~/.ssh/id_rsa.pub

ssh-rsa

abcdefghijklmnopqrstuvwxz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOP QRSTUVWXabcdefghijklmnopqrstuvwxz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFG HIJKLMNOPQRSTUVWXabcdefghijklmnopqrstuvwxz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwx yzABCDEFGHIJKLMNOPQRSTUVWXgxVQ7WEhTBbCel xxx@dhcp012345.a01.yyy.co.jp yourpc\$







操作説明 画面 ①登録方式は[直接入力]を選択 😓 ユーザーサポート - ご利用方法 | A × │ р https://portal.abci.ai/user/login × 🕒 https://portal.abci.ai/user/usr/p × 🕂 ← → C https://portal.abci.ai/user/usr/public_key_regist.php Q \$ 功一 🏢 アプリ 🥑 Firefox を使いこな... 🗅 ほしい物リストに追... 🔹 アップル 🗅 Yahoo! Japan 🗅 Google マップ 🗅 YouTube 🗅 Wikipedia 🗎 ニュース 📄 お役立ち 🗅 Cisco.com 🗅 Apple » 🛅 その他のブックマーク ABCI利用ポータル **ABCI** ログインユーザ: 萩島 功一 ログアウト <u>ホーム</u> 公開鍵の登録 パスワード変更 公開鍵の登録 公開鍵の登録方式を選択しててください。 公開鍵の参照、削除 オブジェクトストレージ ◉ 直接入力 (2)公開鍵をコピペ 登録方式 ノード予約・キャンセル ファイルアップロード ABCIポイント使用履歴 ABCIポイント追加履歴 ②公開鍵をコピペ 利用グループ管理 ssh-rsa [登録]をクリックし公開鍵を登 abcdefghijklmnopgrstuvwxz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopgrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXabcdefghijklmnopgrstu vwxz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXabcdefghijklmnopqrstuvwxz1234567890ABCD 録 EFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXgxVQ7WEhTBbCel xxx@dhcp012345.a01.xxx.yy.jp **戻**(登録)(3) ③[登録]をクリック 公開鍵登録の際、以下の点にご注意ください。 ・改行文字が含まれていないこと。(特に末尾に改行が含まれていないことに注意してください) ・ヘッダ (ssh-rsa、ecdsa-sha2-nistp256、ecdsa-sha2-nistp384、ecdsa-sha2-nistp521、ssh-ed25519) を先頭に付与していること。 ・RSA公開鍵の場合、2048bit 以上で公開鍵を作成していること。 ※ 公開鍵は初回設定後も ・ECDSA公開鍵の場合、256bit、384bitもしくは521bitで公開鍵を作成していること。 利用ポータルで参照、削除、 ・Ed25519公開鍵の場合、256bit で公開鍵を作成していること。 ・全角文字などの不正文字が含まれないこと。 (追加)登録が可能です。



ABCI利用ポータル









ABCI利用ポータルヘアクセスし、 自分が所属する利用グループを確認する

	S WE	B CAS formulator	× 🛛 🚱 http	os://portal	l.abci.ai/us	er/login ×	٢	https://portal.abc	ai/user/usr/u	+							
$\leftarrow \ \rightarrow $	C A	portal.abci.ai/u	user/usr/user_info.php											☆	8 🗖	* 35-	
アプリ	🗎 Fire	efox ブックマーク	🎦 News 🗎 Popul	ar 🗎 ä	お役立ち	ג-ב= 🗎	0	【確定申告書等作	🗯 Apple	🙃 Coronavirus 2019) 闄 Firefox を使い	こな 🜖 Google Ma	ips	»	🗋 その他	のブック	マーク
X		BC Cloud Infrastructure							AB	CI利用ポ-	ータル		ログインコ	Lーザ:	萩島 功-	- <u>ログ</u>	<u>アウト</u>
• <u>ホー</u> / • <u>パスワ</u>	<u>ム</u> フード変	更	利用者情	靜報													
• <u>公開</u> 錄	鍵の登録	Ř	氏名	萩島	助一												
• <u>公開鍋</u>	<u>建の参照</u> ごろ約・1	<u>、削除</u> たわいわり	所属機関	産総	研												
• <u>ABCI</u>	ポイン	<u>ト使用履歴</u>	部署	ディ	ジタルア	ーキテクラ	チャ推	推進センター									
• ABCI	ポイン	ト追加履歴	メールアドレス	hag	ishima.k	o@aist.go	o.jp										
• <u>利用</u> ク	グループ	管理	所属ABCIグルー	プ								1	1				
• <u>クラウ</u>	<u> シドスト</u>	<u>・レージ管理</u>	ABCIグループ名		利用テ-	-7					利用責任者	利用管理者	利用期間				
			gaa50019										2020/04/0	2 - 20	21/03/	31	
			gxa50040										2020/04/0	2 - 20	21/03/	31	
			gaa50069										2020/06/1	5 - 20	21/03/	31	
			gaa50109										2020/04/0	2 - 20	21/03/	31	
			gaa50123										2020/04/0	2 - 20	21/03/	31	
			gca50152										2020/04/0	2 - 20	21/03/	31	
			gac50463		~								2020/06/1	8 - 20	21/03/	31	





インタラクティブノードへのログイン

Windowsユーザ(かつWSLを導入されていない)の皆様は、インタラクティブノードへのロ グインは、SSHクライアントソフトウェア (PuTTY, Teraterm等)を使用願います。 具体的な接続方法については、ABCI User Guideを参照願います。

Teratermの利用: https://docs.abci.ai/ja/tips/tera-term/ PuTTYの利用: https://docs.abci.ai/ja/tips/putty/

*所属する機関によっては下記のようなセキュリティポリシーを適用している場合があります。その際は、所属する機関のIT部門からのサポートを得てください。

- SSH接続をプロキシー経由としている
- 秘密鍵を(各自のPCではなく)サーバーに置くことになっている
- アクセス制御により、下記のようなポート番号を使用禁止にしている
 - 10022, 8888, 6006 ... (ABCIで利用するポート番号)



インタラクティブノードへのログイン(ssh)



「aaa12345xx」は、自分の「ABCI-ID」に置き換える!

yourpc\$ ssh -L 10022:*es*:22 -I *aaa12345xx as.abci.ai*

1) ターミナルからアクセスサーバ(as.abci.ai)にログインします。

Warning

* 秘密鍵ファイルへのパス指定サブコマンド: -i~/.ssh/秘密鍵ファイル名

上記状態で何らかのキーを入力するとSSH接続が切断されてしますので注意してください。

2) 別のターミナルで、インタラクティブノード(es)にポートフォーワーディングします。

yourpc\$ ssh -p 10022 -l *aaa12345xx* localhost

Warning

終了処理:終了する場合は、「exit」コマンドを投入。何も操作がなければ、1時間でセッションは自動的に切断されます。

ssh -L *10022:es:22* -l aaa12345xx as.abci.ai ssh -p *10022* -l aaa12345xx localhost localhostのポート番号:remote host:remote hostのポート番号 localhostのポート番号(ssh -Lで使った番号と一致させる)



*ハンズオンはここからスタートします。

これで、ABCIを利用する準備が整いました





計算ノードへのアクセス手順





ABCIでのオペレーション

- ABCIはLinux OSで構成されたシステム
 - コマンド実行による対話的な操作が主体



定型の操作を覚えれば利用は簡単



計算ノードにおける基本操作

- ジョブの投入 (qsub, qrsh)
- ファイルのアップロード、ダウンロード (scp)
- クラウドストレージの利用



バッチジョブによるABCI利用フロー





バッチジョブの準備 ジョブスクリプトの作成と実行権の付与



1) PCでジョブスクリプト(Sample.sh)を作成し、PCのホーム領域に保存。

		date	
		hostname	
		echo "Hello, world"	
2)	PC環境で		!

yourpc\$ scp −P 10022 *Sample.sh* aaa12345xx@localhost:Sample.sh Enter passphrase for key: +++++++ ← パスフレーズを入力 *Sample.sh* 100% file-size transfer-speed transfer-time

3-1) ジョブスクリプト(Sample.sh)の改行コードがCR+LFまたはCRになっている場合: 改行コードをLinux環境(LF)に変更し、sample.shで保存。

[username@es1 ~]\$ nkf -Lu Sample.sh > sample.sh

3-2) 文字コードがShift-JIS、改行コードがCR+LFになっている場合: 文字コードと改行コードをLinux環境(UTF-8, LF)に一括変換。

[username@es1 ~]\$ nkf -wd Sample.sh > sample.sh

3-3)文字コード、改行コードの確認。

[username@es1 ~]\$ nkf --guess sample.sh sample.sh: UTF-8 (LF)

4) インタラクティブノード(es)にログインして、ジョブスクリプトに実行権を付与。

[username@es1[~]]\$ chmod u+x sample.sh [username@es1[~]]\$ ls -I total 0 -rwxr--r-- 1 username usergroup 61 Jan 11 13:12 sample.sh

- * Linuxでは、文字コード(UTF-8)、改行コード(LF)が使用される。PCのOSによっては文字コード、改行コードの変換が必要。 • Windowsの場合: 文字コード(Shift-JIS)、改行コード(CR+LF)
- Mac OS X以前のMacの場合: 改行コード(CR)



バッチジョブ(qsub)



1) インタラクティブノード(es)にログインして、sample.shを実行。

「gaa12345xx」は、自分の「利用グループ名」に置き換える!

[aaa12345xx@es1 ~]\$ qsub -g *gaa12345 -*I *rt_C.small=1 sample.sh* # *gaa12345: グループ名, rt_C.small=1 :* 計算資源タイプ(CPU x 5コア), *sample.sh:* ジョブスクリプト Your job 151645 ("sample.sh") has been submitted

2) バッチジョブの状況を参照。

[aaa1234 job−ID	5xx@es1 ^ prior]\$ qstat name	user	state	submit/start at	queue	jclass	slots ja-task-ID
151646	0.25586	sample.sh	aaa12345xx	r	01/20/2019 15:16:53	gpu@g0002		10

3) バッチジョブの出力(ホームディレクトリ)。

[aaa12345xx@es1 ~]\$ ls -l	
-rw-rr 1 <i>aaa12345xx gaa12345</i>	172 1月 20 15:17 sample.sh.e151646 #エラー出力ファイル(数字はジョブ番号)
-rw-rr 1 <i>aaa12345xx gaa12345</i>	0 1月 20 13:51 sample.sh.o151235 # <mark>正常出力ファイル(数字はジョブ番号</mark>)

予約ノードの指定法: サブコマンド (-ar *ar_id*)

ar_id は「予約ID」(4桁の数字)

予約ID は、「利用ポータル」の「ノード予約・キャンセル」または、qrstatコマンドで参照できる。 優先度を上げるには: サブコマンド(-p-400)

#ポイントは通常の1.5倍

ジョブをプログラム(スクリプト)化 -> 多数のジョブを一度に投げられる



インタラクティブジョブ(qrsh)



1) インタラクティブノード(es)にログインして実行。

「gaa12345xx」は、自分の「利用グループ名」に置き換える!

[aaa12345xx@es1 ~]\$ qrsh -g *gaa12345* -l rt_F=1 -l h_rt=01:00:00 *# gaa12345: グループ名, rt_F=1 :* 計算資源タイプ(フルノードを1個), *h_rt=01:00:00*(最大1時間確保)

2) インタラクティブジョブの状況を参照。

[aaa1234 job−ID	5xx@es1 ົ prior]\$ qstat name	user	state	submit/start at	queue	jclass	slots ja-task-ID
151646	0.28027	QRLOGIN	aaa12345xx	r	01/21/2019 09:39:43	gpu@g0371		10

予約ノードの指定法: サブコマンド(-ar *ar_id*)

ar_id は「予約ID」(4桁の数字)

予約ID は、「利用ポータル」の「ノード予約・キャンセル」または、qrstatコマンドで参照できる。

ジョブ実行時に、インタラクティブに計算ノードへアクセス可能 -> デバッグ等に便利



ファイルのアップロード・ダウンロード(scp)



1) ファイルのアップロード: インタラクティブノード(es)にログインし、別のターミナルからscpを実行。

yourpc\$ scp -P 10022 *local-file* aaa12345xx@localhost:*remote-file # local-file* : PCからアップしたいファイル、*remote-file* : リモートファイル名 Enter passphrase for key: +++++++ ← パスフレーズを入力 *local-file* 100% file-size transfer-speed transfer-time

「aaa12345xx」は、自分の「ABCI-ID」に置き換える!

2) ファイルのダウンロード: インタラクティブノード(es)にログインし、別のターミナルからscpを実行。

yourpc\$ scp -P 10022 aaa12345xx@localhost:*sample* ./ *# sample*: PCへダウンロードしたいファイル Enter passphrase for key: +++++++ ← パスフレーズを入力 *sample* 100% file-size transfer-speed transfer-time



ABCI**クラウドストレージの特徴** Amazon S3 互換のオブジェクトストレージサービス

- ① AWS CLI (コマンドラインインタフェース) が使える
- ② ABCIにログインしなくても、httpsで外部から使える
- ③ サーバーサイド暗号化に対応している
- ④ 複数のクラウドストレージ・アカウントを持てる
- ⑤ バケット単位で、アクセス制御を設定できる (非公開、ABCI利用グループで共有、全世界への公開を選択できる)



ABCIクラウドストレージの利用例

■ <ユースケース1>データやプログラムの共有・公開の場として

データやプログラムを特定のグループ内(又は複数のグループ間)で共有。バケット単位で自由にアクセス制御ができるので、 柔軟に共有ができる。特に、クラウドストレージに保存されたデータを利用してABCIの計算ノードで処理する場合、データ転送を 低レイテンシーでゼロコストで可能となる。

■ <ユースケース2>IoTデバイスやエッジシステムとのデータ連携として

カメラやセンサーなどのIoTデバイスで取得したデータを直接、ABCIクラウドストレージに格納。取得したばかりのデータを 使って、ABCI計算ノードで処理させることができる。

■ <ユースケース3>ホーム領域の延長として

ホーム領域が不足してきた場合、データの保管場所として利用。低額の従量制課金なので、グループ領域(共有ディスク)をTB 単位、月単位で使用するよりも、低料金で柔軟に使うことができる。

■ <ユースケース4>複数のアカウントで使い分ける

研究者は複数のプロジェクトを推進している場合が殆ど。複数のアカウントを作れるので、プロジェクト毎にバケットの管理 (アクセス制御)を行うことができる。

■ <ユースケース5>暗号化してデータの情報漏洩を防止する

サーバーサイドの暗号化機能を使えば、万一、クラウドストレージから情報が漏洩してもデータを復元できない。



ABCIクラウドストレージの使い方



https://docs.abci.ai/ja/abci-cloudstorage/usage/ https://docs.abci.ai/portal/ja/03/#manage-cloudstorage

・ AWS-CLIモジュールのロード

[username@es1 ~]^{\$} module load aws-cli

• 認証情報などの設定

[username@es1 ~]\$ aws configure AWS Access Key ID [None]: ACCESS-KEY-ID AWS Secret Access Key [None]: SECRET-ACCESS-KEY Default region name [None]: us-east-1 Default output format [None]:(入力不要)

	•	各種	操	作	Default output format [None]:(人力不安)
AWS-CLI構文 aws [options]	<command/> <subcommand> [p</subcommand>	oarameters]	デー	タのコピ	ピー: ername@es1 ~l\$ awsendpoint-url https://s3 abci ai s3 cp. /images/0001 ing
例 awse	endpoint-url https://s3.abci.ai s3 l	S	s3:/	/datase	et-c0541/
s3: ⊐¬	マンド、ls: サブコマンド		<i>=</i> "_	uplo 夕の移動	oad: images/0001.jpg to s3://dataset-c0541/0001.jpg
バケットの作成:				[user	ername@es1 ~]\$ awsendpoint-url https://s3.abci.ai s3 mv annotations.zip
[username@es1 ~]\$	awsendpoint-url https://s3.ab	ci.ai s3 mb	s3:/	/datase	et-c0541/
s3://dataset-summer-2012	2012			mov	ve: ./annotations.zip to s3://dataset-c0541/annotations.zip
make_bucket: datas	set-summer-2012		オフ	ジェクト	トのリスト:
「username@es1 ~]\$	awsendpoint-url https://s3.ab	ci ai s3 ls		2010	19-07-05 17:33:05 4 test1 tyt
2019-06-15 10:47:	37 testbucket1			201	19-07-05 21:12:47 4 test2.txt
2019-06-15 18:10:	37 testbucket2		<u> </u>	カルディ	ィレクトリとクラウドストレージの同期:
オブジェクトのリスト:				[user	ername@es1 ~]\$ awsendpoint-url https://s3.abci.ai s3 sync ./sensor2
[username@es1 ~]\$	awsendpoint-url https://s3.ab	ci.ai s3 ls	s3:/	/mybuc	icket/
s3://testbucket1				uplo	oad: sensor2/0002.dat to s3://mybucket/0002.dat
2019-07-05 17:33:0	05 4 test1.txt		·	uplo	oad: sensor2/0004.dat to s3://mybucket/0004.dat
2019-07-05 21:12:4	47 4 test2.txt		オノ	シェクト	
			-2.1	Luser	ername@es1 ~j\$ awsendpoint-url https://s3.abci.ai s3 rm
			S3:/	/mybuc	
				aele	ete: s3://myducket/feadme.txt



さあ、ABCIを利用してみよう! ABCI環境構築

- モジュールコマンド: ABCIにインストール済のソフトウェアを利用 例: python, cuda, cudnn, singularity
- インストールコマンド:利用したいソフトウェアをインストール 例: TensorFlow, Jupyter Notebook
- コンテナイメージの利用:ソフトウェア利用環境をそのまま利用
 例: digits



アプリケーション環境の構築







Pythonモジュールのロード



• アプリ実行環境の適用

- ABCIで用意されたアプリ実行環境(環境変数)を設定可能

 I <mark>マンド</mark> \$ module [av	vail] [load MODULE] [unload MODULE] [list]
オプション	説明
avail	利用可能な environment module の一覧を表示。av に省略可。
load	moduleの読み込み
unload	module の解除
list	現在読み込んでいる module をリスト表示

実	行例
	[username@g0003 Sample]\$ which python3 Python3環境は未設定
	/usr/bin/which: no python3 in (/apps/) [username@g0003 Sample]\$ module load python/3.6/3.6.5
	[username@g0003 Sample]\$ module list
	Currently Loaded Modulefiles:
	1) python/3.6/3.6.5 ← Python3.6.5がロードされた



Python仮想環境の設定



- プロジェクトのアプリ実行環境をディレクトリ単位で管理可能 (他アプリ実行環境とのバージョン干渉を排除できる)
- python2 系は virtualenv, python3 系は venv など多数あり
- 仮想環境を使う度に activate で有効化、deactivate で終了

実行例:virtualenv (python2)
[username@g0003 ~]\$ virtualenv ~/Sample/v_tf_gpu ← 仮想環境を作成。ディレクトリapp1を作成 [username@g0003 ~]\$ source ~/Sample/v_tf_gpu/bin/activate ← 仮想環境を開始。(csh の場合 activate.csh) (v_tf_gpu)[username@g0003 ~]\$ pip install (v_tf_gpu)[username@g0003 ~]\$: ここここの間、pip などで取得するモジュールの ファイルは ~/Sample/v_tf_gpuディレクトリに配置される。 (v_tf_gpu)[username@g0003 ~]\$ deactivate ← 仮想環境を終了。
実行例:venv(python3)
[username@g0003 ~]\$ python3 -m venv ~/Sample/v_tf_gpu仮想環境を作成。ディレクトリapp2を作成[username@g0003 ~]\$ source ~/Sample/v_tf_gpu/bin/activate仮想環境を開始。(csh の場合 activate.csh)(v_tf_gpu)[[username@g0003 ~]\$ pip installこの間、pip などで取得するモジュールの(v_tf_gpu)[[username@g0003 ~]\$ deactivateファイルは ~/Sample/v_tf_gpuディレクトリに配置される。(v_tf_gpu)[[username@g0003 ~]\$ deactivate仮想環境を終了。



ABCIでJupyter Notebookを使う

https://docs.abci.ai/ja/tips/jupyter-notebook/



ABCIでJupyter Notebookを起動

(1) 計算ノードを一台占有し、Python仮想環境を作成し、

<u>pipでtensorflow-qpuとjupyterをインストール(一度、実施すればいい)。</u>

[username@es3 ~]\$ qrsh -g gaa12345 -l rt_F=1 [username@g0001 ~]\$ module load python/3.6/3.6.5 cuda/10.0/10.0.130.1 cudnn/7.4/7.4.2 [username@g0001 ~]\$ python3 -m venv ~/jupyter_env [username@g0001 ~]\$ source ~/jupyter_env/bin/activate (jupyter_env)[username@g0001 ~]\$ pip3 install tensorflow-gpu jupyter numpy==1.16.4

* 次回以降は、以下のようにモジュールの読み込みと~/jupyter_envのアクティベートだけでいい。

[username@es3 ~]\$ qrsh -g gaa12345 -l rt_F=1 [username@g0001 ~]\$ module load python/3.6/3.6.5 cuda/10.0/10.0.130.1 cudnn/7.4/7.4.2 [username@g0001 ~]\$ source ~/jupyter_env/bin/activate (jupyter_env)[username@g0001 ~]

(2) Jupyter Notebookを起動。 ローカルPCでJupyterを起動中でポート8888を使用している場合、「8889」を使用!

```
(jupyter_env)[username@g0001 ~]$ jupyter notebook --ip=`hostname` --port=88888 --no-browser
```

[I 20:41:12.082 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation). [C 20:41:12.090 NotebookApp]

To access the notebook, open this file in a browser: file:///home/username/.local/share/jupyter/runtime/nbserver-xxxxxx-open.html Or copy and paste one of these URLs: http://g0001.abci.local:8888/?token=token_string or http://127.0.0.1:8888/?token=token_string



Jupyter NotebookでABCIを利用

(3) 別ターミナルで。ローカルPCの8888番ポートを計算ノードの8888番ポートに 転送するSSHトンネルを作成。

yourpc\$ ssh -L 8888:g0001:8888 -l username -p 10022 localhost

(4) ブラウザで下記のURLを開く(トークンは、(2)をコピペ)。

http://127.0.0.1:8888/?token=token_string

- パスワードを登録しておけば、毎回トークンをコピペする必要はない。
 (詳細は、次ページを参照)
- * Jupyter Notebook利用上の注意点
- ## notebookコンフィグファイル
- Jupyter Notebookの利用には、notebookコンフィグファイルが必要です。

「jupyter_notebook_config.py」が下記のフォルダにない場合は、次のコマンドで notebookコンフィグファイルを作成してください。

- Windows: C:¥Users¥USERNAME¥.jupyter¥jupyter_notebook_config.py
- OS X: /Users/USERNAME/.jupyter/jupyter_notebook_config.py
- Linux: /home/USERNAME/.jupyter/jupyter_notebook_config.py

yourpc\$ jupyter notebook --generate-config



Jupyter Notebook:このような画面が出たら

🗂 Jupyter

Password or token: Log in Invalid credentials
Log in
Log in
(2)
の
(2

(2) 一度パスワードを登録しておけば、 登録したパスワードを入力 (毎回、トークンをコピペする必要なし)

(1) ここに、トークンとパスワードを入力

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you enable a password.

The command:

jupyter notebook list

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

Currently running servers: http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks

or you can paste just the token value into the password field on this page.

See <u>the documentation on how to enable a password</u> in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

Token

New Password



Jupyter NotebookでPythonプログラミング例

● ● ● 初心者向け Jupyte × 🗅 sample_mnist.py - × 既 jupyter notebook × 🔿 Home Page - Sele × 🧧 test_20191209 - J × 🕂
← → C ① localhost:8888/notebooks/test_20191209.ipynb ☆ Ø □ ★ ⊕ :
🏭 アプリ 🥝 JALカード - ログ 💿 ログインページへ / 📀 【確定申告書等作 🝼 確定申告ソフト「 » 🎦 その他のプックマーク
CJUPYTER test_20191209 Last Checkpoint: 2019/12/09 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 O
P + % 42 I + ↓ N Run ■ C → Code ✓ □
<pre>In [1]: import tensorflow as tf mnist = tf.keras.datasets.mnist (x_train, y_train),(x_test, y_test) = mnist.load_data() x_train, x_test = x_train / 255.0, x_test / 255.0 model = tf.keras.models.Sequential([tf.keras.layers.Flatten(input_shape=(28, 28)), tf.keras.layers.Dense(128, activation='relu'), tf.keras.layers.Dense(128, activation='relu'), tf.keras.layers.Dense(10, activation='softmax')]) model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy']) model.fit(x_train, y_train, epochs=5) model.evaluate(x_test, y_test) WARNING:tensorflow:From /home/aaa10005vk/iupyter_env/lib/python3.6/site-packages/tensorflow/python/op</pre>
wARNING:tensorriow:From / nome/aaa 1000/syk/jup/ter_env/iib/python3.6/site-packages/tensorriow/python/op s/init_ops.py:1251: calling VarianceScalinginit (from tensorflow.python.ops.init_ops) with dtype is deprecat ed and will be removed in a future version. Instructions for updating: Call initializer instance with the dtype argument instead of passing it to the constructor Epoch 1/5
Epoch 2/5 60000/60000 [=================================
60000/60000 [========] - 2s 37us/sample - loss: 0.1074 - acc: 0.9674 Epoch 4/5
Epoch 5/5 60000/60000 [=================================

Out[1]: [0.07159279510800261, 0.9777]



Jupyter Notebookの終了

以下の手順で終了。

- ローカルPCのJupyter Notebookホーム画面の「quit」ボタンで終了。
- ローカルPCの8888ポートを転送していたSSHトンネル接続を Control-C で終了。
- 計算ノードで起動中のJupyterプログラムが終了してない場合は、 Control-C で終了。



コンテナ(Singularity)の利用

https://docs.abci.ai/ja/09/



ABCIで利用可能なコンテナは、Singularity

- Singularity
 - ABCIではLinuxコンテナとしてSingularityが利用可能
 - 利用可能なバージョンは、moduleコマンドで確認可
 - NGC(※)コンテナレジストリなどから、Docker コンテナイメージ をSingularityイメージに変換し利用が可能
 <u>※ NGC (NVIDIA GPU Cloud) https://ngc.nvidia.com/</u> GPUに最適化されたディープラーニングフレームワークコンテナやHPCア プリケーションコンテナのDockerイメージと、それらを配布するための NGCコンテナレジストリを提供

実行例(module)	
\$ module avail	
/apps/modules/modulefiles/runtimes	
hadoop/2.9.1 singularity/2.6.1 spark/2.3.2	
hadoop/2.9.2 spark/2.3.1 spark/2.4.0	



Singularityイメージの作成

• NVIDIA NGCからDIGITSを取得





Singularityイメージの起動

• DIGITSの起動例

aaa12345xx@es3 ~]\$ qrsh -g gaa50123 -l rt_G.small=1 aaa12345xx@g0031 ~]\$ cd abci101/singularity_digits aaa12345xx@g0031 singularity_digits]\$ module load singularity/2.6.1 aaa12345xx@g0031 singularity_digits]\$ module list currently Loaded Modulefiles: 1) singularity/2.6.1 aaa12345xx@g0031 singularity_digits]\$ singularity runnv digits-20.06-tensorflow-py3.simg -d -p 8889
= DIGITS == ========
IVIDIA Release 20.06 (build 13420111) DIGITS Version 6.1.1
Container image Copyright (c) 2020, NVIDIA CORPORATION. All rights reserved. DIGITS Copyright (c) 2014-2019, NVIDIA CORPORATION. All rights reserved.
'arious files include modifications (c) NVIDIA CORPORATION. All rights reserved. IVIDIA modifications are covered by the license terms that apply to the underlying project or file.
IOTE: Legacy NVIDIA Driver detected. Compatibility mode ENABLED.
¥ / /) (¥ '_ _ / 6.1.1



ABCI利用ポータル

https://portal.abci.ai/user/

https://portal.a	abci.ai/user/ × +					
\leftarrow \rightarrow C $\$ portal.at	oci.ai/user/ 🔍 🖈 📿 🔹 🐲 :					
🏢 アプリ 🛛 🥑 Firefox を使いこ	な 🔇 ほしい物リストに追 🗯 アップル 🚱 Yahoo! Japan 🔇 Google マップ 🚱 YouTube 🚱 Wikipedia 📄 ニュース 📄 お役立ち 🚱 Cisco.com 🚱 Apple 🧼 📄 その他のブックマーク					
ABCI利用ポータル ABCI利用ポータル						
利用者ログ・	イン					
言語 / Language	● 日本語 / Japanese ● 英語 / English					
ABCIアカウント名						
画像認証 (数字6文字)						
	(数子6义子)					
	次へ					



ABCIアカウント名	
パスワード	
ログイン	



● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	5 Global Mobility 🗙	- チケット - aist. 🗙 📔 F	Fwd: ABCIログ ×	S https://portal. ×	🝐 マイドライブ	×	https://p	ortal. ×	+	
\leftarrow \rightarrow C $ ightarrow$ portal.abci.ai/	user/usr/user_info.	php						☆	功一	:
ABCI			ABCI	利用ポータ	フル □	グインニ	2ーザ:		ログア・	ウト
 <u>ホーム</u> <u>パスワード変更</u> 	利用者	青報								
 <u>公開鍵の登録</u> <u>公開鍵の参照、削除</u> ノード予約・キャンセル 	氏名 所属機関	產総研								
 ABCIポイント使用履歴 ABCIポイント追加履歴 利用グループ管理 	部署 メールアドレス 所属ABCIグル・	情報・人間工学領域 - ロープ	研究戦略部							
 クラウドストレージ管理 	ABCIグル ープ名	」用テーマ				利用 責任 者	利用 管理 者	利用期間	5	
							853353			



● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	bal x	M 受信ト x │ S h	https:// × 🚱 https:// ×	📥 あなた 🗙	● 2020_(× 🤆	Windo ×	a Amazo ×	🚯 在宅勤 🗙 🔇	https://× +
\leftrightarrow \rightarrow C $$ portal.abci.ai/u	iser/usr	/project_mng.php						☆ Q	м 🤫 ந— :
🏢 アプリ 🛛 🥑 Firefox を使いこな	🔂 🗔	しい物リストに 🗯 🕻	アップル YY Yahoo! Japan	S Google マップ	🕈 🕨 YouTube 🌘	S Wikipedia	E =1-7 E	□ お役立ち »	🛅 その他のブックマーク
ABCI A Bridging Cloud Infrastructure				ABC	利用ポ-	ータル		ログインユーザ:	萩島 功一 <u>ログアウト</u>
・ <u>ホーム</u> ・ <u>パスワード変更</u>	利	用グループ	プ管理						
 <u>公開鍵の登録</u> <u>公開鍵の参照、削除</u> 	選 択	ABCIグループ 名	利用テーマ			利用責任 者	ABCIポイン 量	ノト残 利用期間	9
 <u>ノード予約・キャンセル</u> <u>ABCIポイント使用履歴</u> <u>ABCIポイント追加履歴</u> 	0	gaa						0000 (0	4/00
・ <u>利用グループ管理</u> ・ <u>クラウドストレージ管理</u>	0	gxa							
	0	gaa							
	0	gaa							
		gca						2021/0	0/0 I
	戻る	ABCIポイント追加印	「グループディスク追加	加または削減申請	利用者追加申請	利用者編集	利用者削除	利用管理者権限設定	利用責任者変更申請



ABCIサービス諸元



1. 計算ノードの利用 ジョブ割当リソース

	資源タイプ	資源タイプ名	説明	割り当て物理 _{CPU} コア数	割り当て GPU数	メモリ(GiB)	ローカル ストレージ(GB)	資源タイプ 課金係数	割当可能な 個数
	Full	rt_F	ノード占有	40	4	360	1440	1.00	1-512 (1-32)*
	G.large	rt_G.large	ノード共有 GPU利用	20	4	240	720	0.90	1
	G.small	rt_G.small	ノード共有 GPU利用	5	1	60	180	0.30	1
	C.large	rt_C.large	ノード共有 CPUのみ利用	20	0	120	720	0.60	1
	C.small	rt_C.small	ノード共有 CPUのみ利用	5	0	30	180	0.20	1
	M.large	rt_M.large	ノード共有 CPUのみ利用	8	0	800	480	0.40	1
New!	M.small	rt_M.small	ノード共有 CPUのみ利用	4	0	400	240	0.20	1

*バッチジョブは最大512個、インタラクティブジョブは最大32個

資源タイプ名と数量で資源量を指定 例) 資源タイプ(Full)を1個指定: -l rt_F=1





hour:minute:second 時間:分:秒



57



3. 計算ノードの予約利用

項目	説明
最小予約日数	1日
最大予約日数	30日
システムあたりの最大同時予約可能ノード数	442ノード
1予約あたりの最大予約ノード数 *システムでの最大同時予約数の制限があるため、最大 予約ノード数未満又はノードを予約できない場合がある。	32ノード
1予約あたりの最大予約ノード時間積	12,288ノード時間積
予約受付開始時刻	30日前の午前10時
予約受付締切時刻	予約開始前日の午後9時
予約取消受付期間	予約開始前日の午後9時
予約開始時刻	予約開始日の午前10時
予約終了時刻	予約終了日の午前9時30分

予約ノードを利用する際は、qsubコマンド又はqrshコマンドにて、「-ar ar_id」オプションを使う。



4. ABCI料金表: 2019/2020年度

		インタラクティブ ジョブ	バッチジョブ	予約			
	rt_F	1.0 ポイン	36ポイント/日 (1.5ポイント/時間)				
	rt_G.large	0.9ポイン	ルト/時間				
己ケ次万	rt_C.large	0.6ポイン	・ト/時間				
訂昇頁源	rt_G.small	0.3ポイン	ΝΔ				
	rt_C.small	0.2ポイン	0.2ポイント/時間				
	rt_M.large	0.4ポイン					
	rt_M.small	0.2ポイン	▶/時間				
	グループ領域 (共有ディスク)	(ABCIアカウント毎1	のホーム領域は無償)				
New!	ABCIクラウド ストレージ		E				



5. ジョブ実行上の特長 大規模演算を必要とする場合、有効

- バッチジョブ(Spot)では、最大512ノードまでを利用可能
- インタラクティブジョブ(On-demand)でも、最大32ノードまで利用可能
- ジョブ投入数(ユーザあたり)
 - 1)アレイジョブあたりの最大投入可能タスク数: 75,000
 - 2)同時投入可能ジョブ数: 1,000
 - 3)同時実行ジョブ数: 200



ABCI参考サイト

より詳細な情報については、以下を参照下さい。

- ABCIユーザガイド https://docs.abci.ai/ja/
- ABCI利用ポータル
 <u>https://portal.abci.ai/user/</u>
- ABCI利用ポータルの利用ガイド <u>https://docs.abci.ai/portal/ja/</u>
- ABCIユーザサポート
 https://abci.ai/ja/how_to_use/user_support.html
- ABCI利用法に関するFAQ
 https://abci.ai/ja/how_to_use/yakkan.html#faq9



A Bridging Cloud Infrastructure

https://abci.ai/