HPC クラスタにおけるバッチジョブ投入のための Web アプリケーション Open Composer の開発

中尾 昌広^{1,a)} 山本 啓二¹

概要:HPC クラスタを利用するには,Linux コマンドやジョブスケジューラなどの前提知識が多いため, 初心者にとって学習コストが大きいという課題がある.そこで,我々は HPC クラスタの主な利用用途で あるバッチジョブの投入を簡易に行うための Web アプリケーション Open Composer の開発を行った. Open Composer は Web フォームを用いたジョブスクリプトの自動生成機能やジョブの投入などの機能を 持つ.本論文では,Open Composer の設計,開発,利用方法などについて述べる.

1. 背景

HPC クラスタは幅広い分野で利用されているが,その 利用には高い学習コストが伴うという課題がある.具体的 には,SSH クライアントのインストール,鍵ペアの生成, 公開鍵の登録といった準備に加え,CLI (Command Line Interface) による Linux コマンドやジョブスケジューラな どの知識が必要である.また,リモートデスクトップや JupyterLab などの GUI (Graphical User Interface) アプ リケーションを計算ノード上で対話的に利用したい場合, ポートフォワーディングなどの設定も必要である.これら の知識や設定は HPC クラスタごとに異なるため,新しい クラスタを利用するたびに再度学習が必要である.

この課題を解決するため、Ohio Supercomputer Center は HPC クラスタを操作するための Web ポータル Open OnDemand を開発している [1]. Open OnDemand を用い ると、ファイル管理、データ転送、ジョブの投入と管理、 ターミナル接続、GUI アプリケーションを対話的に利用す るための設定などを、ユーザが使い慣れた Web ブラウザか ら実行できる. Open OnDemand を用いることで、学習コ ストの削減と煩雑な準備や設定の省略を行うことができ、 さらに統一されたインタフェースで様々な HPC クラスタ を利用することができる. また、Open OnDemand はアプ リケーションプラットフォームとしての機能も持っており、 Open OnDemand には複数の便利な Web アプリケーショ ンがプリインストールされている. また、他の Web アプ リケーションの追加も行うことができる.

+ 7644 200 -		* Craste Temptate	
Ci Edit Files 🖉 Job Options 32, Open Terminel	• Game	E Corres	Job Datalia
Show 21 2 entries Overted IF Name	11 10	Search:	Job Name: (default) Simple SequentialJob Security:
November 11, 1914 - Moffault) Simple Sequential Jub Dillon Showing 1 to 1 of 1 antities		Pretox 1 Mod	R 6050 boud Acoust Mananet Sogh bollor: //Walkaret/sound.rs.manaput/Wene/Identifystery/register/001616/1/11 //Walkaret/sound.rs.manaput/Wene/Identifystery/register/001616/1/11 //Walkaret/sound.rs.manaput/Wene/Identifystery/register/001616/1/11
			nelis, (ds.)sh Yodar Garlantiz malin (Jobuh
			Salani Sojel mata Jobah Koje prostavi Pristavi se na se

☑ 1 Job Composer on Open OnDemand

HPC クラスタの主な利用方法は、実アプリケーション をバッチジョブとして実行することである. HPC クラス タにおけるバッチジョブとは、ジョブスケジューラによっ て非対話的に実行される計算タスクのことを指す.一般的 にバッチジョブはスクリプト形式で記述され、必要な計算 リソース(ノード数や実行時間など)をジョブスケジュー ラ用のディレクティブ(例:#SBATCH -N 4)を用いて指 定する. そして, バッチジョブがジョブスケジューラに投 入されると、ジョブスケジューラは HPC クラスタの空き 状況から判断した適切なノードを確保し、そのバッチジョ ブを実行する. Open OnDemand からバッチジョブをジョ ブスケジューラに投入したい場合, Open OnDemand に プリインストールされている Job Composer を利用できる (図 1). Job Composer は Web ブラウザからジョブの投入 と作成が可能であるが、ジョブスクリプトは手動で作成す る必要がある. つまり, ユーザはアプリケーションの実行 コマンドやジョブスケジューラのディレクティブを記述す る必要があるため、その学習コストは大きく、また書き間 違いなども発生すると考えられる.

理化学研究所 計算科学研究センター 兵庫県神戸市中央区港島南 町 7-1-26

^{a)} masahiro.nakao@riken.jp

IPSJ SIG Technical Report

_		$\overline{\mathbf{\xi}}$ 1 Real-world applications on Fugaku Open OnDemand
	Category	Application
	Climate	SCALE
	Computer Aided Engineering	FDS, FFVHC-ACE, FrontFlow (blue/X), FrontISTR, OpenFOAM (Foundation/OpenCFD)
	Condensed Matter Physics	ALAMODE, AkaiKKR, $\mathcal{H}\Phi$, mVMC, OpenMX, PHASE/0, Quantum Espresso, SALMON
	Experimental Data Processing	KIERTÄÄ
	Molecular Dynamics	GENESIS, GROMACS, LAMMPS, MODYLAS, PIMD
	Quantum Chemistry	ABINIT-MP, Gaussian, NTChem, SMASH
	Quantum Simulation	braket



2 How to start a GUI application

そこで本稿では、ジョブスクリプトを自動生成できる Web アプリケーション Open Composer の開発を行う. Open Composer では、実アプリケーション毎に用意され た Web フォームにパラメータを入力することでジョブス クリプトを生成するため、ジョブスケジューラに対する学 習コストを下げ、また書き間違いの頻度を低減できると考 えられる.また、Open Composer はジョブ投入や削除な どの管理を行う機能も提供している. Open Composer は Open OnDemand 上で動作するアプリケーションであり、 Open OnDemand のいくつかの機能と連携させている.

本稿の構成は次の通りである.2章ではジョブスクリプトの自動生成に関連する研究についてまとめる.3章と4章では,Open Composer の開発と利用方法について述べる.5章では本稿のまとめを行う.

2. 関連研究

2.1 Open OnDemand の Web フォームの利用

我々は「富岳」[2] 上で Open OnDemand の運用を行って いる [3], [4], [5]. 「富岳」で実アプリケーションのバッチジョ ブの投入を簡易に行うために, 1章で述べた Job Composer の代わりに Open OnDemand の GUI アプリケーションを 実行する Web フォームを利用している. 「富岳」の Open OnDemand で運用している実アプリケーションの一覧を **表 1** に示す.まず,ユーザが Open OnDemand 上で GUI アプリケーションを実行する手順を図 **2** を用いて説明す る.「富岳」の Open OnDemand のトップページ(図2左) から,実行したい GUI アプリケーションのアイコンをク リックすると,Webフォーム(図2中)が表示される.ジョ ブに対するパラメータを入力した後に「Launch」ボタンを クリックすると,Open OnDemand 内部で自動生成された ジョブスクリプトがジョブスケジューラに投入される.計 算ノード上で GUI アプリケーションが実行されると有効に なるボタンをクリックすると(図2右),Open OnDemand のリバースプロキシの機能により,計算ノード上で動作し ている GUI アプリケーションがユーザの Web ブラウザに 表示される.

我々が Open OnDemand を用いて運用しているバッチ ジョブの投入手順は、図 2 の (1) と (2) のみを利用してい る. バッチジョブの Web フォームの例を図 3 に示す. 図 3 で入力された情報と図 4 に示すジョブスクリプトのテンプ レートからジョブスクリプトを生成する. ジョブスクリプ トのテンプレートの形式は ERB (Embedded RuBy) が埋 め込まれた YAML (YAML Ain't Markup Language) であ り、 <%= %>の箇所は Web フォームで入力された値に置換 される. 1~6 行目はジョブスケジューラに渡すパラメータ を定義し、8~18 行目はジョブスクリプトを定義している.

このようなバッチジョブの投入手順は,アプリケーショ ンの実行コマンドやジョブスケジューラのディレクティブ を完全に隠蔽できるため,1章で述べた Job Composer の 課題を解決していると言える.しかし,そのトレードオフ

IPSJ SIG Technical Report

FDS	
Fire Dynamics Simulator (FDS) is a large-eddy simulation code.	
Queue	
fugaku-small	\$
Maximum run time (1 - 72 hours)	
1	٢
Number of nodes (1 - 384)	
1	٢
Total number of processes (1 - 18,432)	
1	٢
Total number of processes <= Number of nodes x 48.	
Number of threads (1 - 48)	
1	٢
Number of threads x Total number of processes <= Number of nodes x 48.	
Version	
6.7.9	*
Working directory	
/home/rccs-aot/a03011/	
Select Path	
Input file	
/home/rccs-aot/a03011/a.txt	
Select Path	
Email (You will receive an email when it starts and ends)	
Launch	
Launch	

3 Example of a web form for a real application

1	script:
2	queue_name: <%= queue %>
3	email: <%= email %>
4	native:
5	L elapse=<%= hours %>:00:00,node=<%= nodes %>
6	mpi proc=<%= procs %>
7	
8	<pre>batch_connect:</pre>
9	template: "basic"
10	<pre>script_wrapper:</pre>
11	#!/usr/bin/env bash
12	
13	. /vol0004/apps/oss/spack/share/spack/setup-env.sh
14	<pre>spack load fds@<%= version %></pre>
15	
16	export OMP_NUM_THREADS=<%= threads %>
17	cd <%= working_dir %>
18	mpiexec fds_mpi_fugaku <%= input_file %>

🛛 4 Template of a job script

として,生成されるジョブスクリプトの記述自由度の低さ が新たな課題となっている.事前に用意した Web フォー ムとテンプレートからジョブスクリプトが生成されるた め,例えばユーザはアプリケーションに対して任意のコマ ンドを追加することはできない.

Cluster	
rccs-cloud	\$
Edit	
Script	
hello_world.sh	\$
Edit	
Queue	
a100	\$
Edit	Remove

Add new option

☑ 5 Project Manager on Open OnDemand

HPC Port										
		Cr	eate a Ne	ew Job						
		t	est					~ or	create a new project	
		Job F	lesources: *							
		1	cpus, 4gb m	em, 30 min	s (Debug)				~
		Soft	vare: *							
		G	aussian 16							~
		Cust	om Configuratio	n:						
				_						v
		s	ubmit 🗲	—c	lick	(
				٦						
Job Number 💙	Software	Description	Resources	Status	Runtime	Submission time	Project	Directory	Repository	
00000004	Gaussian 16	an example Gaussian job	1 cpus, 4gb mem, 30 mins (Debug)	Completed	0.00:15	24/08/20218:57 p.m.	test	Open Download	10.5281/zenodo.5510302	Delete
00000003	Gaussian 16	-	1 cpus, 4gb mem 30 mins (Debug)	Completed	Unknown	24/08/20218:56 p.m.	-	Open Download	Publish	Delete
00000001	Gaussian 16	-	GPU, 12hrs	Completed	0.00:15	11/08/20212:14 p.m.	-	Open Download	Publish	Delete

☑ 6 CHAMP on Open OnDemand[6]

2.2 Open OnDemand 上で動作するアプリケーション

Open OnDemand は Project Manager という Web アプ リケーションがベータ版としてプリインストールされてい る^{*1}(図 5). Web フォームからジョブスケジューラの投 入コマンドにオプションを渡す機能(ディレクティブの記 述と同等の機能)を持つ.しかし,アプリケーションの実 行コマンドなどはユーザが手動で記入する必要がある.

CHAMP[6] は Open OnDemand 上で動作する Web アプ リケーションであり, FAIR (Findable, Accessible, Interoperable, and Reusabe) 原則に沿った研究データの生成と 公開を自動的に行う機能を持つ (図 6). CHAMP の特徴 として, CHAMP から実行されたジョブの状態を確認でき る Web ページを提供しており, その Web ページからジョ ブの結果をデータリポジトリに公開できる. CHAMP は 2.1 節で述べた方法と同様に Web フォームとテンプレート を用いたジョブの投入を行うため, ジョブスクリプトの記 述自由度が低いという課題がある.

^{*1} 原稿執筆時点(2024 年 11 月)で最新版の Open OnDemand 3.1.10 ではデフォルトで無効になっている.

IPSJ SIG Technical Report

Job Marso				
Job Name	test			
Location	Change	/home/users/masahiro.naki	ao/job_composer/test	
Environments	Fugaku			~ (+
Number of tasks 🕐	1			٢
Expected run time 🔿	0	1	۵ ا	٥
		Preview	_	
			Click	
ob Preview				
tem	plate.txt		driver.sh	
#!/usr/bin/env bash				
#!/usr/bin/env bash #PJMname "test" #PIM -l "node=1"				
#l/usr/bin/env bash #PJMname "test" #PJM -L "node=1" #PJM -L "elapse=1:0:	00"			
#l/usr/bin/env bash #PJMname "test" #PJM -L "node=1" #PJM -L "elapse=1:0: cd /home/users/masa)0" hiro.nakao/test			Α
#l/usr/bin/env bash #PJM -name "test" #PJM -L "node=1" #PJM -L "elapse=1:0: cd /home/users/masa	00" niro.nakao/test		Subr	nit Cancel

🗵 7 Drona Composer on Open OnDemand

	Parameters				
Limit this job to one node:					
Number of processor cores across all nodes:	1				
Number of GPUs:					
Memory per processor core:	1 GB 🕤				
Walltime:	1 hours 00 mins 00 secs				
#Submit this script with: sbatch the	filename				
<pre></pre>	filename				
#SBATCH	concor cores (i e tasks)				
#SBATCHnodes=1	per CPU core				

🛛 8 BYU Job Script Generator

Drona Composer[7] は Open OnDemand 上で動作する Web アプリケーションであり,アプリケーションの実行 コマンドとジョブスケジューラのディレクティブを含んだ ジョブスクリプトを生成できる(図 7).図7上でユーザ は Web フォームの項目を入力し,「Preview」ボタンをク リックすると,図7下が表示され,ジョブスクリプトの確 認および編集を行った後,「Submit」ボタンでジョブの投 入を行う.図7下のタブの「driver.sh」では,ジョブを投 入する際の前処理を設定できる.Drona Composer はジョ ブの状況確認や削除などを行う機能を持たないが,それら は既存の Open OnDemand の機能で代替可能である*².

2.3 Open OnDemand 以外の Web アプリケーション

BYU Job Script Generator[8] は, ジョブスケジューラで ある Slurm と PBS のディレクティブを生成する Web アプ リケーションである (図 8). BYU Job Script Generator の特徴として,Webフォームにパラメータを入力すると, リアルタイムでジョブスクリプトが更新される画面も表示 されるため,ジョブスクリプトの確認が行いやすい点が挙 げられる.他にも,Slurm に限定しているが同様のWeb アプリケーションが多数存在する [9], [10], [11], [12].しか し,これらのWebサービスはディレクティブ部分を生成 するのみであり,アプリケーションの実行コマンドは生成 できない.また,ジョブスケジューラと連携する機能も持 たないため,ジョブの投入などを行うことはできない.

3. Open Composer の開発

3.1 要件

Open Composer は、ジョブスクリプトの作成に伴う学 習コストを削減するとともに、バッチジョブを HPC クラ スタに容易に投入可能な環境を提供することにより、初心 者から熟練者までの幅広いユーザが HPC クラスタを効率 的に活用できることを目指している.そのための要件とし て、下記が挙げられる.

- ジョブスクリプトの自動生成とエラーチェック機能
- 様々な実アプリケーション用のジョブスクリプトを生成できる柔軟なカスタマイズ性
- ジョブ投入やジョブ管理などを行う機能
- HPC クラスタ間の差異を隠蔽する統一的なインタ フェース

3.2 設計

Open OnDemand は Web ブラウザを通じて HPC リソー スを利用できる Web アプリケーションのデファクトスタ ンダードであり,世界中の HPC クラスタで利用されてい る.そのため, Open Composer は Open OnDemand 上で 動作する Web アプリケーションとして作成する.この設 計により, Open Composer のデプロイを簡略化でき,さ らに Open OnDemand の認証機能やデータ転送機能など をそのまま利用できるため, Open Composer に対する開 発コストの低減を図ることができる.

Open Composer は、2.2 節で述べた Drona Composer と 同様の下記の機能を持つ.(1) アプリケーション毎に用意 された Web フォームからのジョブスクリプトの生成,(2) Web フォームで入力したパラメータに対するエラーチェッ ク、(3) ユーザがジョブスクリプトを直接編集できる機能, (4) ジョブを投入する際の前処理の定義.また、ジョブスク リプトの確認を行いやすくするため、2.3 節で述べた BYU Job Script Generator と同様に、リアルタイムでジョブス クリプトが更新される画面も表示する.さらに、ジョブの 投入と管理を行う機能も追加する.ここで、ユーザがジョ ブスクリプトを作成する手順としては、そのユーザが過去 に作成したジョブスクリプトの設定を少し変更して使用す

^{*2} Drona Composer の Youtube 動画 (https://www.youtube. com/watch?v=tgpP9LPG1YQ) では, Open OnDemand 上で動作 する他の Web アプリケーションを使ってジョブ管理を行ってい るが, このアプリケーションが公開されているかどうかは不明.

IPSJ SIG Technical Report

る場合が多いと考えられる.そのため,投入したジョブの パラメータなどの情報をジョブ管理機能に渡すことにより, ジョブスクリプトを簡易に再利用できる機能も開発する.

3.3 実装

Open OnDemand は Ruby on Rails で作成されているた め, Open OnDemand がインストールされた環境では必ず Ruby が利用できる. そのため, Open Composer のバッ クエンドには Ruby を用いたオープンソースの軽量フレー ムワーク Sinatra[13] を用いた. フロントエンドにはレス ポンシブなサイトを構築するためのツールキットである Bootstrap とジョブスクリプトをリアルタイムで更新する ために HTML5 と JavaScript を用いた.

Open Composer では、統一的な操作でジョブ管理を行う ため、ジョブスケジューラ間の差異を吸収するアダプタを 作成する必要がある。そこでアダプタのためのスーパーク ラスを定義し、ジョブスケジューラ毎にそのサブクラスを 作成する設計にした。スーパークラスで定義した関数は、 ジョブを投入するための submit()、ジョブを削除するため の cancel()、ジョブの状態を確認するための query() で ある。現在は、「富岳」で利用されているジョブスケジュー ラの Fujitsu_TCS と Slurm のアダプタを作成しており、そ れぞれのアダプタを開発するのに要した Ruby の行数は 79 と 76 である。

Open Composer の実装に要した各言語の行数(2024 年 11 月時点)は、Rubyは 1,269 行、JavaScriptは 737 行、ERB を含む HTML5 は 291 行である。Open Composer はオープンソースソフトウェアとして、https: //github.com/RIKEN-RCCS/OpenComposer で公開する予 定である。

4. Open Composer の利用方法

4.1 概要

Open Composer は「トップページ」・「アプリケーショ ンページ」・「履歴ページ」から構成される.本節ではそれ ぞれの概要について説明する.

トップページを図 9 に示す. トップページではカテゴリ 毎にアプリケーションのアイコンが表示される. そのアイ コンをクリックするとアプリケーションページに遷移する.

アプリケーションページを図 10 に示す. 図 10 左の Web フォームでパラメータの入力を行うと,図 10 右のテキス トエリアにジョブスクリプトがリアルタイムで生成され る.ジョブスクリプトの編集も可能である.図 10 右下の 「Submit」ボタンをクリックすると,そのジョブスクリプ トがジョブスケジューラに投入される.

履歴ページを図 11 に示す. 履歴ページでは, Open Composer を通じて投入された過去のジョブの履歴の閲覧と実行中のジョブの状態の確認などを行うことができる.また,



9 Top page on Open Composer



☑ 10 Application page on Open Composer

						Filter	
						All 🔵 Running 🔵 Queue	d 🔵 Comple
Job ID	Application	Partition	Script Location	Script Name	Job Name	Submission at	Status
29969	KIERTÄÄ	fx700	/Users/mnakao/	job.sh	job.sh	2024-11-08 13:35:20	Cemplete
29968	FDS	fx700	/Users/mnakao/ 🖿	job.sh	job.sh	2024-11-08 13:35:05	Cemplete
29861	Fujitsu TCS	fx700	/Users/mnakao/ 🖿	job.sh	job.sh	2024-11-07 15:10:41	Complete
29860	Fuiltsu TCS	fx700	/Users/mnakao/	job.sh	job.sh	2024-11-07 15:07:40	Cemplete
29856	Fujitsu TCS	fx700	/Users/mnakao/	job.sh	job.sh	2024-11-07 14:58:56	Cemplete
29855	Fujitsu TCS	fx700	/Users/mnakao/	job.sh	job.sh	2024-11-07 14:58:47	Cemplete
29854	Fujitsu TCS	fx700	/Users/mnakao/	job.sh	job.sh	2024-11-07 14:58:43	Cemplete
29853	Eujitsu TCS	fx700	/Users/mnakao/	job.sh	job.sh	2024-11-07 14:57:42	Complete
29852	Fulitsu TCS	fx700	/Users/mnakao/	job.sh	job.sh	2024-11-07 14:57:22	Complete
29851	KIERTÄÄ	fx700	/Users/mnakao/	job.sh	job.sh	2024-11-07 14:53:16	Complete

🗵 11 History page on Open Composer

過去のジョブスクリプトを簡易に再利用できる仕組みも持つ.「Script Name」列の文字列をクリックすると,図12

IPSJ SIG Technical Report



- 2 curegory: com
- 3 icon: icon.png
- 4 description: This application is test.

🗵 13 Sample of manifest.yml

のようなウィンドウが表示され,「Load Parameters」をク リックすると,そのジョブスクリプトがロードされた状態 のアプリケーションページに遷移する.

4.2 アプリケーションの設定例

Open Composer のインストール先(デフォルトでは /var/www/ood/apps/sys/OpenComposer)にアプリケー ション毎のディレクトリを作成し、その中に YAML 形 式の設定ファイル manifest.yml と form.yml を作成す る. それぞれに ERB を含めたい場合は、ファイル名を manifest.yml.erb と form.yml.erb にする.

manifest.yml にはアプリケーションの説明を記述する. manifest.yml の例を図 13 に示す. name はアプリケー ション名, category はアプリケーションのカテゴリ, icon はアイコン画像へのパスであり, それらはトップページの 表示に用いる. 4 行目の description はアプリケーション ページの左上でアプリケーションの説明を表示するために 用いる.

form.ymlは form, script, check というブロックで構 成されており, ジョブスクリプトの生成に用いる. form ブロックはジョブ情報を入力するためのウィジットの定 義, script ブロックはジョブスクリプトのテンプレート, check ブロックは生成されたジョブスクリプトのエラー チェックに用いる. Open Composer は下記のウィジット を用意している. これらのウィジットや 4.3 節で述べる Dynamic Form Widget は, Open OnDemand の仕様を参 考にしている*³.

- number:数値を入力する.
- text:テキストを入力する.
- email: email アドレスを入力する.
- select:セレクトボックスから項目を1つ選択する.
- multi_select:セレクトボックスから項目を複数選択 する.

```
1
   form:
 2
      partition:
 3
        widget: radio
        label: Partition
 4
 \mathbf{5}
        value: Small partition
 6
        direction: horizontal
 7
        options:
          - [Small partition, small]
 8
9
          - [Large partition, large]
10
      nodes:
11
        widget: number
12
        label: Number of nodes (1 - 128)
13
        value: 4
        min: 1
14
15
        max: 128
16
        step: 1
17
      time:
        widget: number
18
19
        size: 2
        label: [Max time (0-24 h), Max time (0-59 m)]
20
21
        value: [ 1, 0]
        min: [ 0, 0]
22
23
        max: [72, 59]
24
        step: [ 1, 1]
25
26
   script: |
27
      #!/usr/bin/env bash
28
      #SBATCH -p #{partition}
      #SBATCH -N #{nodes}
29
      #SBATCH -t #{time_1}:#{time_2}:00
30
31
       mpiexec ./a.out
32
33
   check: |
      if @time_1.to_i == 24 && @time_2.to_i > 0
34
35
       halt 500, "Exceeded Time"
36
      end
```



- radio: ラジオボタンから項目を1つ選択する.
- checkbox:チェックボックスから項目を複数選択する.
- path: Open Composer が動作しているサーバ上のファ イルやディレクトリのパスを選択する.

form.yml と生成されるアプリケーションのページの例 を図 14 と図 15 に示す.アプリケーションページで Web ブラウザの画面サイズが広い場合は図 10 のように 2 段組 みになるが,スマートフォンなどからの利用で画面サイ ズが狭い場合は図 15 のように 1 段組になる.図 15 上の 「Script Location」と「Script Name」は固定された path ウィジットと text ウィジットであり,それぞれジョブスク リプトの保存場所とジョブスクリプト名を入力する.path ウィジットで表示される「Select Path」ボタンをクリック すると、図 16 のような画面が表示され、ディレクトリや ファイルのパスをクリック操作で入力できる.図 14 の 2, 10,17 行目はウィジット名を示す.その次の行の widget と label と value は、それぞれウィジットの種類、Web フォームに表示されるラベル、初期値を示す.5 行目の

^{*3} https://osc.github.io/ood-documentation/ latest/how-tos/app-development/interactive/ dynamic-form-widgets.html

IPSJ SIG Technical Report

Top Application - History		Open OnDemand
Test This application is test.	Script Location /Users/mnakao/	Select Path
	Script Name	
	job.sh	
Partition O Small partition C Large partition		
Number of nodes (1 - 128)		
4		•
Max time (0-24 h)	Max time (0-59 m)	
1	0	٢
Script Contents		
#!/usr/bin/env bash #SBATCH -p small #SBATCH -N 4 #SBATCH -t 1:0:00 mpiexec ./a.out		
Sul	bmit	

15 Sample of application page

Type • Name • bin ruby-2.7.4 include ib64 ib65 ruby-3.3.0		Hide h	idden Filter
bin ruby-27.4 include ib64 ib6 ruby-3.0	Type 🔻	Name	•
ruby-2.7.4 include ib64 ib64 ruby-3.3.0		bin	
include ib64 ib ruby-3.3.0		ruby-2.7.4	
Ib64 Ib ruby-33.0		include	
iib ruby-3.3.0		lib64	
Display=1.10 ruby=3.3.0		lib	
		ruby-3.3.0	

☑ 16 Path widget on Open Composer

direction は、ラジオボタンの選択肢の並ぶ方向を示す. 7~9 行目はラジオボタンの選択肢を示す.1つ目の要素 (例:Small Partition) はラジオボタンに表記されるラベル であり、2つ目の要素(例:small)は script ブロック中で 用いられる値である. number ウィジットで用いられてい る min, max, step は, それぞれ最小値, 最大値, ステップ 幅を示す.19 行目の size は入力数を表し,そのウィジッ トの項目は多要素を持つ配列として記述する. 26~31 行 目の script ブロックでは、ジョブスクリプトを定義して いる. このブロック内で, #{WidgetName} で記述された箇 所は、そのウィジットの値に置換される. size に2以上 の値が用いられている場合, #{WidgetName_n} のようにア ンダースコアと数字をウィジット名の後に追加して指定 する. 33~36 行目の check ブロックでは,入力した値を チェックするための Ruby スクリプトを記述する. この例 では, time のウィジットで 24 時間を超過する値を入力し た場合、ジョブスクリプトの投入前にエラーが表示される.

4.3 前処理機能

表1に示した実アプリケーションの1つである KIERTAA を実行する手順は、まず図 17 のようなパラメータファ

L	user_id = "mnakao"
2	<pre>mode = "jasri_iap_s"</pre>
3	filter = ramp

3

zero_padding = 4 4

W	1 7	Development	£1.	- f	WIEDTÄ	Ä
凶	11	Parameter	ше	OI	KIEKI A.	A

1	script:
2	#!/usr/bin/env bash
3	cd <%= @SCRIPT_LOCATION %>
4	mv <%= @SCRIPT_NAME %> parameters.conf
5	genjs_ct parameters.conf > <%= @SCRIPT_NAME %>

🗵 18 submit.yml.erb

1	form:
2	partition:
3	widget: radio
4	label: Partition
5	value: Normal partition
6	options:
7	- [Normal partition, small, disable-cuda]
8	- [GPU partition, gpu]
9	cuda:
10	widget: number
11	label: CUDA version (12, 13, 14)
12	value: 12
13	min: 12
14	max: 14
15	step: 1

2 19 Sample of form.yml of Dynamic Form Widget



20 Sample of application page of Dynamic Form Widget

イルを記述し、そのパラメータファイルから KIERTÄÄ が提供するスクリプトを用いてジョブスクリプトを生成 する. このように Web フォームから直接ジョブスクリ プトを生成しないアプリケーションに対応するために、 ジョブを投入する前の処理を定義できる機能を開発し た. 図 18 のような submit.yml.erb をアプリケーショ ンのディレクトリに保存し、このファイルがある場合は その script ブロックがジョブの投入前に実行される仕 組みである. submit.yml.erb 中の@SCRIPT_LOCATION と **@SCRIPT_NAME**は、それぞれ図 15上の「Script Location」 と「Script Name」で入力した値に置換される.

4.2 節および 4.3 節の機能を用いることで、表1に示し たすべての実アプリケーションが Open Composer 上で動 作することを確かめた.

IPSJ SIG Technical Report

😣 👄 🥝 👘 🕘 Dashboard									
	0 B	_	_	_	_	_	_	û	
Fugaku Ondemand	Interactive App	- 0						0 -	4 0
	>_ Op	en in Termini	al b + New File b a Ne	w Directory	1. Upload	🕹 Downioa	d 🕒	Copy/Move	🖥 Delete
Home Directory	↑	/ home	/ 'ccs-aot / a03011 /	2 Change	directory			Ó	Copy path
			C Show Ow	ner/Mode	Show Dot!	iles Filte Showing '	er: 16 of 5	56 rows - 0 ro	ws select
		Type *,	Name		•.	Size		Modified at	
		•	bin2		1-	-		2022/7/18 18	8:26:04
		•	cuda-samples		1 -	-		2022/6/10 11	:33:21
		•	Desktop		1 -	-		2022/7/18 11	:02:18

21 Home directory on Open OnDemand

•••		a03011@on	demand-	test:~									
$\leftarrow \rightarrow$		<	2 81		_		_	_	_	_	습		
Host: one	demand	-teat.fugaku.r-	cca <i>r</i> iken	de l						Them	Default		v
op - 18:02:48 up 16 days, 23:24, 1 user, load average: 0.00, 0.00, 0.00 asks: 252 total, 1 running, 251 sleeping, 0 stopped, 0 zombie													
iB Mem		7674.4 t	otal,	6194.2	free,	584.3	usei,	89	5.9 buff/d	cache			
iB Swap		8084.0 t	otal,	8084.0	free,	0.0	usei.	6793	7.8 avail	Mem			
PID	USER	. PR	NI	VIRT	RES	SHR S	%CPJ	%MEM	TIME+	COMMAND			
1	reot	20	0	238100	10780	8144 S	0.0	0.1	0:31.19	systemd			
2	root	20	0	0	0	0 S	0.0	0.0	0:00.21	kthreadd			
3	reot	6	-20	0	0	0 I	0.)	0.0	0:00.00	rcu_gp			
4	reot	6	-20	0	0	0 I	0.0	0.0	0:00.00	rcu_par_gp			
6	root	6	-20	0	0	0 I	0.0	0.0	0:00.00	kworker/0:	0H-events_	highpr	
9	reot	6	-20	0	0	0 I	0.)	0.0	0:00.00	mm_percpu_	мd		
10	reot	20	0	0	0	0 S	0.0	0.0	0:00.00	rcu_tasks_	rude_		
11	root	20	0	0	0	0 S	0.0	0.0	0:00.00	rcu_tasks_	trace		
12	reot	20	0	0	0	0 S	0.)	0.0	0:00.29	ksoftirqd/	8		
10	reot	26	• •	Ð	ø	θI	0.9	0.0	0:39.07	rcu_sched			
14	root	rt	. 0	0	0	0 5	0.0	0.0	0:00.00	migration/	8		
15	root	rt	0	0	0	0 S	0.0	0.0	0:00.57	watchdog/0			
16	reot	20	0	0	0	0 S	0.0	0.0	0:00.00	cpunp/0			
17	root	20	0	0	0	0 S	0.0	0.0	0:00.00	cpuhp/1			

🗵 22 Shell on Open OnDemand

4.4 Dynamic Form Widget

Open Composer は動的なウィジットを作成するための 機能である Dynamic Form Widget を提供している. Dynamic Form Widget は選択肢を持つウィジットにおいて, ある選択肢を選択すると,他のウィジットの値の変更・無効 化・非表示化を行う機能である.本節では Dynamic Form Widget の無効化の機能について,図 19 と図 20 を用い て説明する.図 19 の 2~8 行目ではラジオボタンを定義 している.7行目の3つ目の要素の disable-cuda は cuda という名のウィジットを無効化することを意味している. 図 20 に示す通り,ラジオボタンで Normal partiton を選 択すると cuda ウィジットが無効化される.

4.5 Open OnDemand との連携

図 11 に示す履歴ページでは、ジョブの結果の確認な どを簡易に行うために、Open OnDemand にプリインス トールされているアプリケーションをワンクリックで起動 する機能を提供している.あるジョブ ID の行の「Script Location」の列の文字列をクリックすると、Web ブラウザ からデータの送受信やサーバ上のファイルの編集を行える Home directory アプリケーションが起動する(図 21). また、その文字列の横にあるターミナルのアイコンをク リックすると、ターミナル接続を行える(図 22). これら の機能により、ジョブの出力ファイルに素早くアクセスす ることができ、必要な場合は Linux コマンドを実行するこ とができる.

5. まとめ

本稿では、ジョブスクリプトの生成と投入を行うことが できる Web アプリケーション Open Composer の開発を 行った. Open Composer は、Web フォームと編集可能な テキストエリアの双方を活用することで、高い記述自由度 を維持しつつ、ジョブスケジューラの学習コストを低減す ることを目指している.また、ジョブの管理機能を備えて おり、ジョブ履歴ページから過去のジョブスクリプトを容 易に再利用できる仕組みを提供している.これらの機能に より、HPC クラスタ環境におけるジョブ投入作業の煩雑さ を解消し、ユーザの生産性向上に寄与すると考えている.

参考文献

- Dave Hudak et al. Open ondemand: A web-based client portal for hpc centers. *Journal of Open Source Software*, Vol. 3, No. 25, p. 622, 2018.
- [2] Mitsuhisa Sato et al. Co-Design for A64FX Manycore Processor and "Fugaku". In International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 651–665, Los Alamitos, CA, USA, 2020. IEEE Computer Society.
- [3] 中尾昌広他.スーパーコンピュータ「富岳」における hpc クラスタ用 web ポータル open ondemand の導入.研究報 告ハイパフォーマンスコンピューティング(HPC), No. 2022-HPC-186, 2022.
- [4] 中尾昌広他.スーパーコンピュータ「富岳」における hpc クラスタ用 web ポータル open ondemand の運用.研究報 告ハイパフォーマンスコンピューティング(HPC), No. 2022-HPC-191, 2023.
- [5] Masahiro et al Nakao. Introducing open ondemand to supercomputer fugaku. In Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis, SC-W '23, p. 720–727, New York, NY, USA, 2023. Association for Computing Machinery.
- [6] Christopher Cave-Ayland, Michael Bearpark, Charles Romain, and Henry S. Rzepa. Champ is a hpc access and metadata portal. *Journal of Open Source Software*, Vol. 7, No. 70, p. 3824, 2022.
- [7] Texas A&M High Performance Research Computing. Drona Composer. https://hprc.tamu.edu/kb/ User-Guides/Portal/Drona_composer/.
- Brigham Young University. BYU Job Script Generator. https://byuhpc.github.io/ BYUJobScriptGenerator/.
- [9] National Energy Research Scientific Computing Center. Jobscript Generator. https://my.nersc.gov/script_ generator.php.
- [10] University of Virginia. Slurm Script Generator. https://www.rc.virginia.edu/userinfo/hpc/ slurm-script-generator/.
- [11] University of Michigan. Job script generator for Slurm. https://websites.umich.edu/~greatlakes/ jobscript_generator/.
- [12] George Mason University. Slurm Script Generator. https://wiki.orc.gmu.edu/mkdocs/slurm_ generator/slurm_script_generator.html.
- [13] Sinatra. https://sinatrarb.com.