

大島 聡史（九州大学 情報基盤研究開発センター※）
※調査研究開始時点では名古屋大学 情報基盤センター

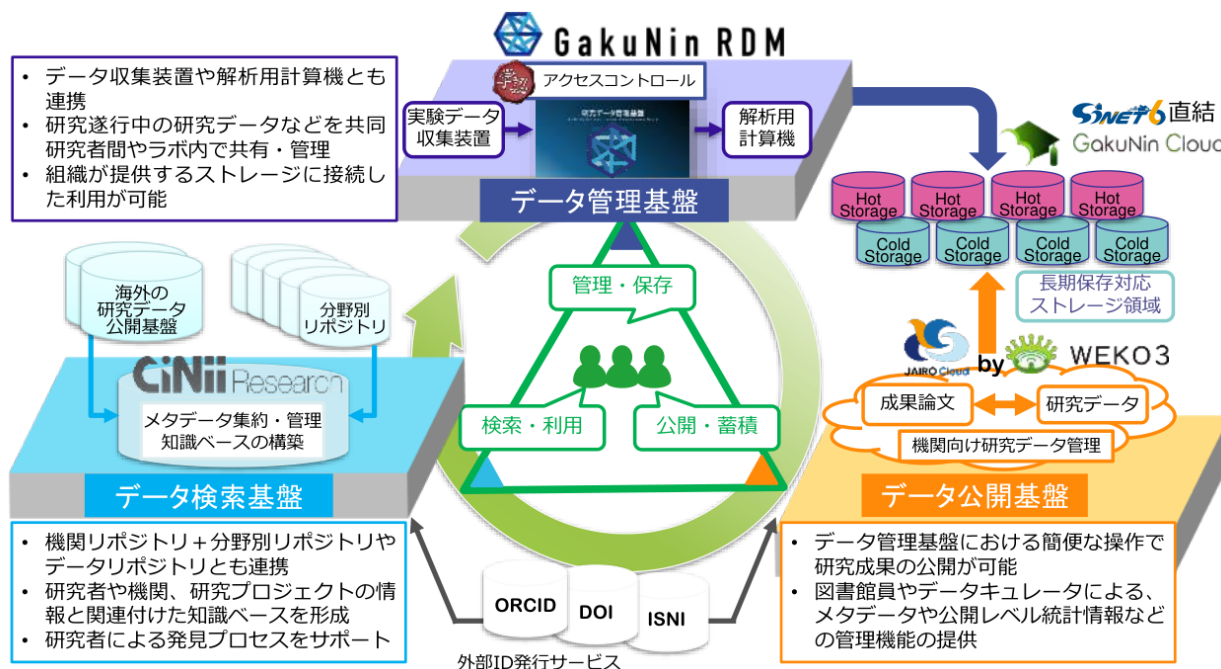
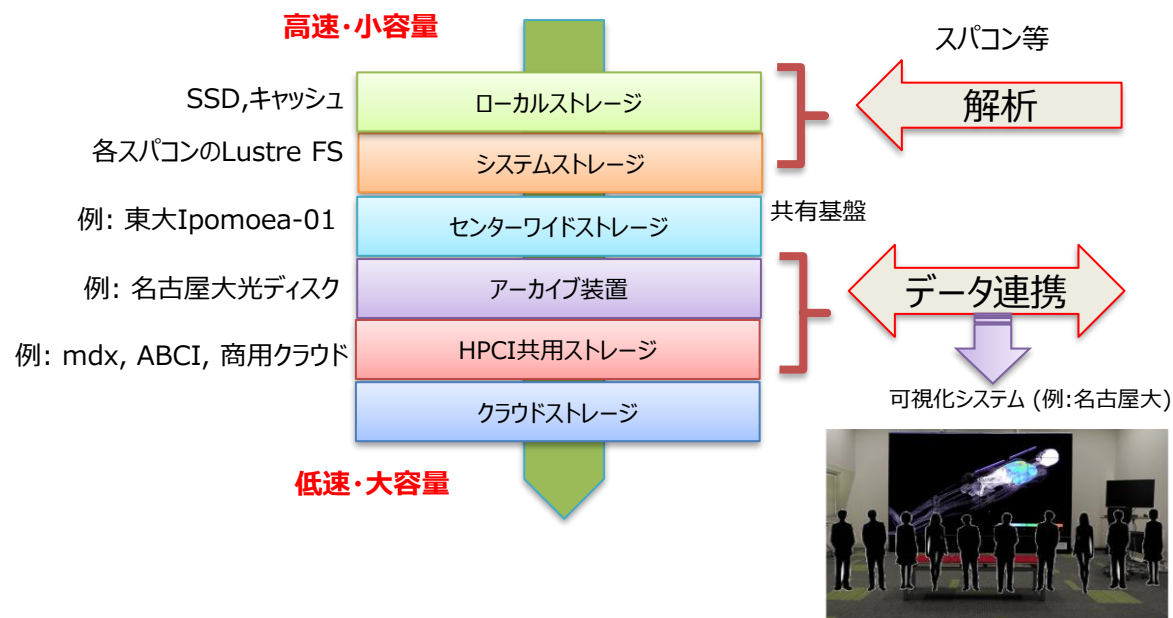
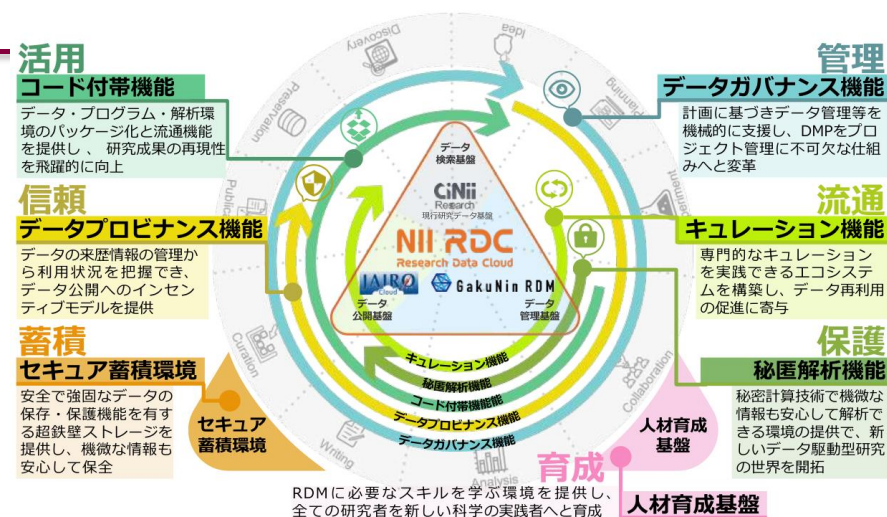
データ利活用とワークフロー

「データ利活用とワークフロー」グループとは？

- 計算資源やストレージの連携に関する調査や改善の検討を行うグループ
- 目的
 - 研究データ基盤をハブとして活用し、各計算資源およびストレージの密な連携によりデータ利活用の促進に繋げる。
- 検討項目
 - 研究データ基盤と各システム相互の連携
 - ストレージ階層の連携・管理・制御
 - 大規模データに関しては極力ストレージを移動することなく適切なセキュリティの維持、データ保全、連携
 - データ規模に応じた取り扱いの指針
- 具体的なアクション：研究データ基盤と各計算資源の連携に向けた調査検討
 - 各計算資源の性能調査：特にストレージと外部接続について
 - データ利活用に関するプログラム・アプリケーションの調査（ヒアリング）
 - データ利活用に資するツール・アプリケーションの調査と試行
- 資源管理グループと合同で定例ミーティングを実施（他グループ等からも多数参加🙏）

背景

- 計算資源やストレージの連携の重要性
 - システムにより異なるストレージ階層
 - システム間の連携や研究データ基盤との連携の重要性
- 状況の把握や性能・使い勝手の向上について調査・議論



右側図の引用元 https://www.nii.ac.jp/openforum/2024/day1_rdc-aieco.html

実施内容（本日の内容）

- 実施した調査についての報告
 - システム調査
 - アプリ開発者・アプリ利用者向けアンケート
 - アプリケーション調査
- 利便性向上に向けた活動の報告
 - Open OnDemand
 - Jupyter
 - その他のワークフローツール
 - Fuzzball PoC

※ 公開の場で詳細を話す前提で行ったものではないため概要のみ
(追跡調査は未実施)

システム調査

- スパコンの性能については既に様々な調査が行われている（性能指標がある）
 - CPU・GPU・メモリ・ネットワーク・ストレージ
- システム間や研究データ基盤との連携を考えると外部との接続に関する性能や機能が重要
 - ポスト「富岳」の性能が高くて必要なデータを必要なときに持ってこられない/提供できない
 - アプリケーション（サイエンス）ワークフロー全体としての性能こそが重要
 - 外部接続ネットワークやゲートウェイの性能
 - データの流れを考慮した際の本当のボトルネックはどこ？
 - ログインノードSSH+バッチジョブ 以外の利用方法
- 調査時期
 - 主に2022年度に実施
 - その後もシステムの更新にあわせて情報を更新

調査結果（概要）

- 調査時点（2022年度頃）のトレンド
 - 共有ストレージはHDDがメイン。徐々にSSDも導入。
 - 計算機システムにまたがるストレージやオブジェクトストレージなども徐々に
 - 東大 Ipomoea
 - 阪大 Onion-object
 - 名大 ODA（光ディスク）（→その後、開発・販売終了）
 - S3, NextCloud, Proselfなどの導入/連携も
 - Jupyter Labの利用（スパコンへのインストールと利用サポート）開始
 - 外部接続ネットワーク（スパコンより上流）は10Gbpsも残る：システム間連携のボトルネック
- 現在のトレンド
 - 共有SSDストレージの普及（HDDと併用が多い）
 - NextCloudやJupyter Labなどの導入例多数
 - 研究データ管理（長期保存）とも連携？
 - 外部接続ネットワークの100+Gbps化が進む
 - 回線速度よりもゲートウェイ・ファイアウォールの性能が課題？

アプリ開発者・アプリ利用者向けアンケート

- HPC関連アプリケーションが本当に必要としている入出力性能は？
- 2022年度にアプリFSとも連携してアンケートを実施
 - システムに対するデータの入出力の量や頻度などについて質問
 - 回答数 約80件
- 回答内容は回答者により大きく異なる。特にインパクトのあったものは……
 - 生命科学分野：毎週10TB生成して1年保存
 - 基礎科学分野：1PBクラスのデータを世界に公開（永久保存）
 - 気象系：リアルタイム実行のため1.5Gbps以上の帯域が必要
 - 気象系：1時間毎に100GB生成し数年間保存
 - その他：毎週10TB生成し1年保存

データ利活用アンケート

- データの共有や公開を行っている方を対象としたアンケート
- 2023年度に実施
 - 回答数 7名
- 利用状況に加えて将来に向けたコメントもいただいた
 - 厳しいコメントもいくつか
 - 現状が厳しい、将来の見通しが厳しい、という意味
 - HPCI資源提供機関のストレージが単一に見えると嬉しい
 - 大規模データのポスト処理・可視化・長期保存などもできてほしい
 - 既に取得データ量が保存データ量を上回っているため全ては保存できていない。科研費ベースのため予算が途切れると打撃。データの生産場所が都心部でないため回線が細い。
 - 合計数十PBのデータを無期限保存＋国際共有。セキュリティを保ちつつ高性能に通信したい。
 - PB級のデータ＋論文採択後10年管理義務が大変
 - オンプレミスのストレージとサーバ保守費が年間100万円以上。厳しい。

実施内容（本日の内容）

- 実施した調査についての報告
 - システム調査
 - アプリ開発者・アプリ利用者向けアンケート
 - アプリケーション調査
- 利便性向上に向けた活動の報告
 - Open OnDemand
 - Jupyter
 - その他のワークフローツール
 - Fuzzball PoC

※ 公開の場で詳細を話す前提で行ったものではないため概要のみ
(追跡調査は未実施)

利便性の向上：機能と使い勝手

- システム間連携を考えるうえではインターフェイス（UI、GUI）も重要な要素
 - 様々なシステムを同じように使えたら便利
 - スパコンごとにバッチジョブシステムが異なる
 - 解決に向けた取り組みの前例はある
 - 例：Xcrypt：様々なバッチスケジューラ向けの並列処理を統一的に記述できる
 - システムにまたがるジョブを実行できたら便利
 - スパコンをまたいだジョブ実行ができない
 - システムをまたいだアカウントの管理などの都合で難しい
 - 個別に起動したジョブ同士が連携してジョブを実行することは不可能ではない
 - バッチジョブシステムを使うことに抵抗がある・GUIで使いたい
 - 特にHPCでない分野の人がスパコンを使ううえでは障壁たりえる

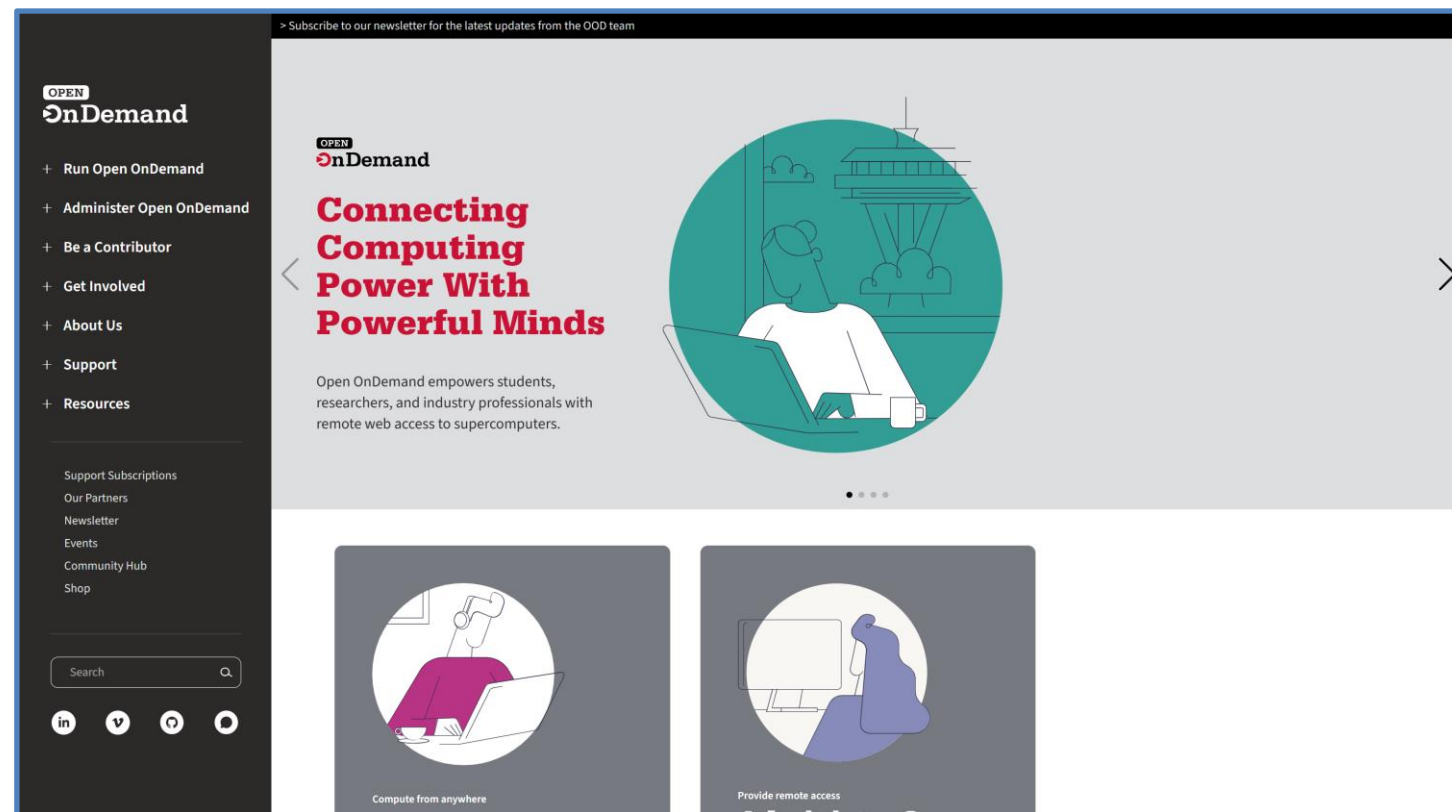
新しい（SSH+コマンドライン以外の）スパコンの使い方

- 前述の問題の全てを解決するのは難しいが、既に様々な取り組みがなされている
- 例えば
 - Open OnDemand
 - Jupyter (Jupyter Notebook, Jupyter Lab)：特にAI分野など
 - ワークフローツール：一部の計算科学分野など
 - miniwdl, Toil, Cromwell, Nextflow, Snakemake, WDL, etc.
- 本グループではこれらについて議論・情報共有・テストを実施

Open OnDemand (OOD)

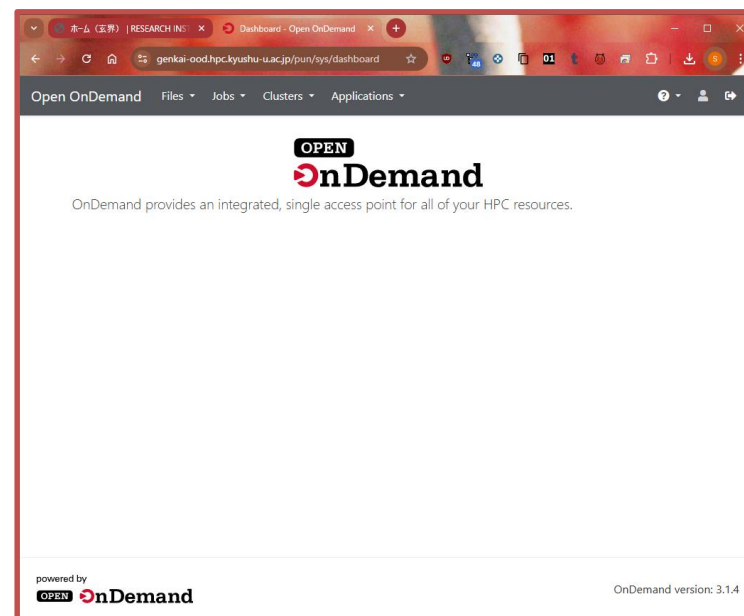
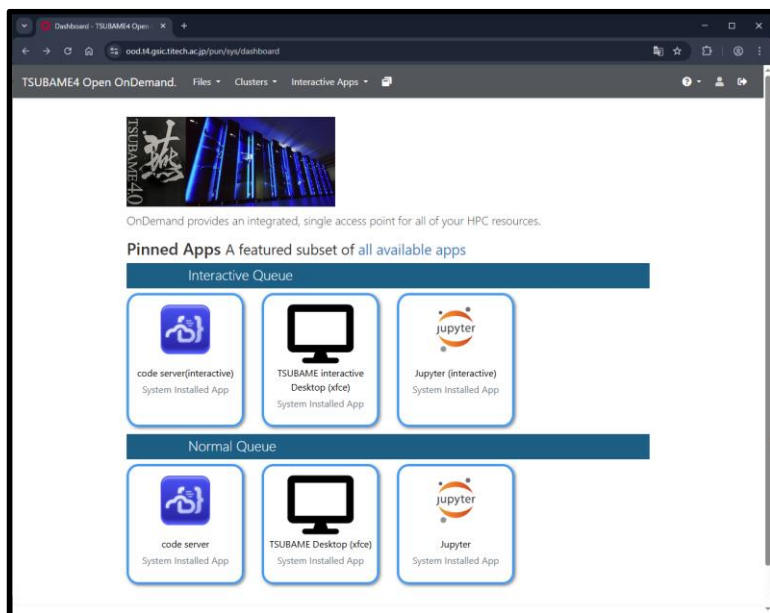
- HPCクラスタ用のWebポータル
- オハイオ州立大学が中心となって開発
 - オープンソースソフトウェア
- WebブラウザからHPCクラスタの操作が可能
 - ファイル操作やジョブ操作など
- 大手クラウドや研究所、大学など多くの機関で利用中
 - 100以上の国
 - 2,100以上の組織
 - (公式発表)

<https://openondemand.org/>



国内HPCIシステム等におけるOpen OnDemandの導入状況

- 少なくとも理研「富岳」、科学大 TSUBAME 4.0、産総研 ABCI 3.0、九大 玄界では一般ユーザ向けの通常サービスとして提供済み
- NIIのVirtual Cloud Provider(VCP)でも利用可能



「富岳」におけるOpen OnDemandの活用

- 2022年8月からテスト運用、2023年5月から本番運用
 - （おそらく国内スパコンでは最初）
 - ダッシュボードをカスタマイズ
 - アプリケーションを追加（約50個）
 - HPCI共用ストレージとGakuNin RDMとのデータ共有機能なども提供
- 主に中尾さん@理研らが尽力
 - OODに対してFujitsu TCS対応コードを作成し本家に提供
 - HPCIコンソーシアムの2025年度HPCIソフトウェア賞 普及部門賞 を受賞
 - <https://hpci-c.jp/hrdevelop/award.html>

The screenshot displays the Fugaku OnDemand web interface. At the top, there's a navigation bar with links like 'Batch Jobs', 'Interactive Apps', 'Passenger Apps', and 'My Sessions'. The main content area includes a 'Welcome to the supercomputer Fugaku' message, a 'Message of the day' section, and a 'Fugaku Schedule' calendar. Below these, there's a 'Pending jobs' section with a table of job details. The 'Accounting' section provides a detailed table of resource usage across different groups and volumes. At the bottom, there are sections for 'Recently Used Apps' and 'Passenger Apps', each with icons for various applications like Desktop, OpenFOAM, SCALE, Jupyter, Active Jobs, Home Directory, GakuNin RDM, HPCI Storage, Job Composer, and Fugaku Shell Access.

Accounting (Updated at 2023/07/25 02:32:10 (JST))

Group	Volume	Disk (GiB)				Disk (inode)				Resource (NH)			
		Limit	Usage	Avail.	Rate	Limit	Usage	Avail.	Rate	Limit	Usage	Avail.	Rate
rccs-aot	/vol0400	5,120	608	4,512	11%	1,500,000	265,259	1,234,741	17%	527,360	81,666	445,693	15%
f-op	/vol0403	614,400	160,260	454,140	26%	180,000,000	144,261,340	35,738,660	80%	-	-	-	-
ra030002	/vol0403	5,120	1	5,119	0%	1,500,000	2	1,499,998	0%	-	-	-	-
/home	/vol0400	20	5	15	25%	200,000	11,619	188,381	5%	-	-	-	-

Recently Used Apps

- Desktop
- OpenFOAM
- SCALE
- Jupyter

Passenger Apps

- Active Jobs
- Home Directory
- GakuNin RDM
- HPCI Storage
- Job Composer
- Fugaku Shell Access

参考：「富岳」のOpen OnDemandで提供されているアプリケーション

• GUI

ブラウザ操作で簡単にGUIの起動やジョブの実行が可能

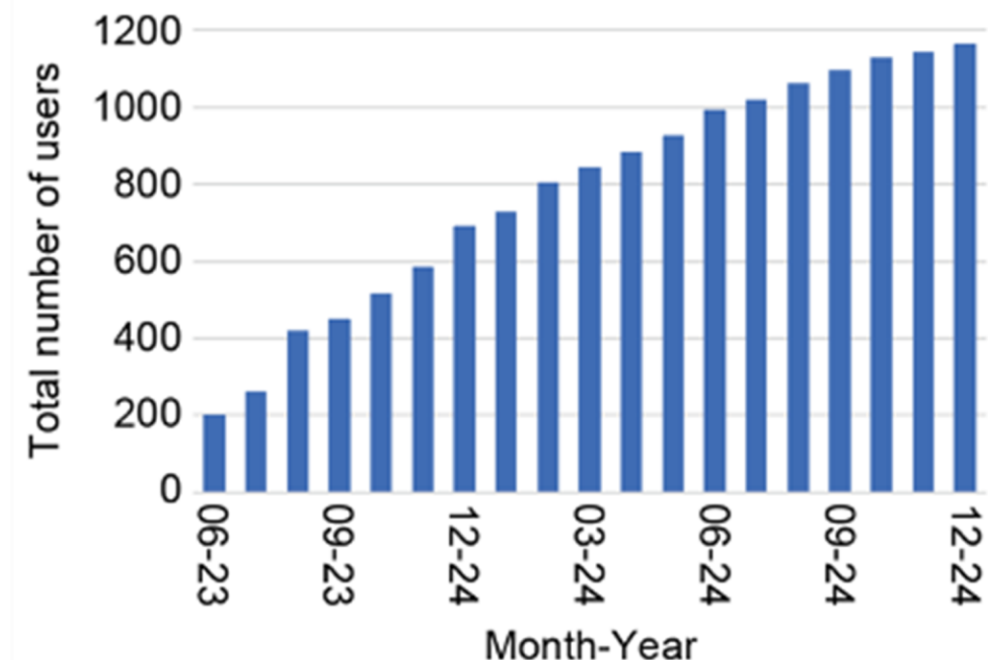
Category	Application
Development	Remote Desktop, JupyterLab, MATLAB*, VSCode, Terminal
Profiler	NVIDIA Visual Profiler*, NVIDIA Nsight Compute*, NVIDIA Nsight Systems*, Vampir*
Viewer	AVS/Express*, C-Tools, GaussView*, Gnuplot, GrADS, ImageJ, OVITO, Paraview, PyMOL, SALMON view, Smokeview, VESTA, VMD, VisIt, XCrySDen
Workflow	WHEEL

• バッチジョブ

赤字はSingularity管理アプリ、*は商用アプリ

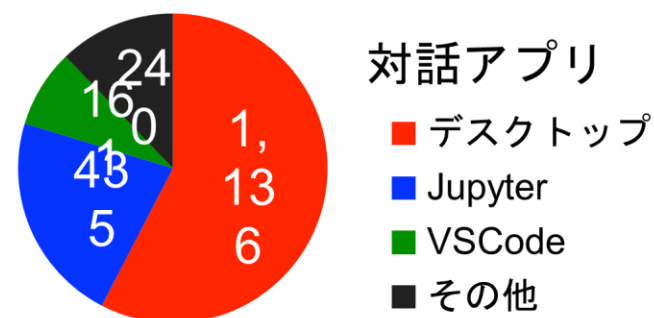
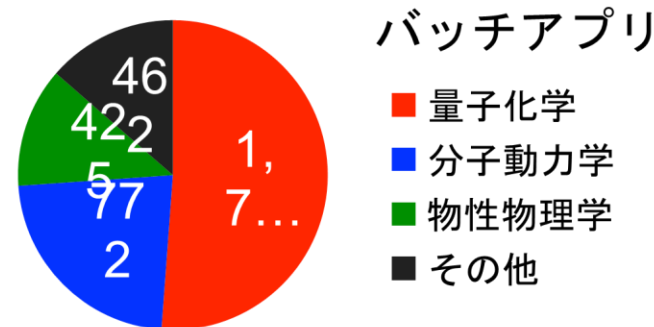
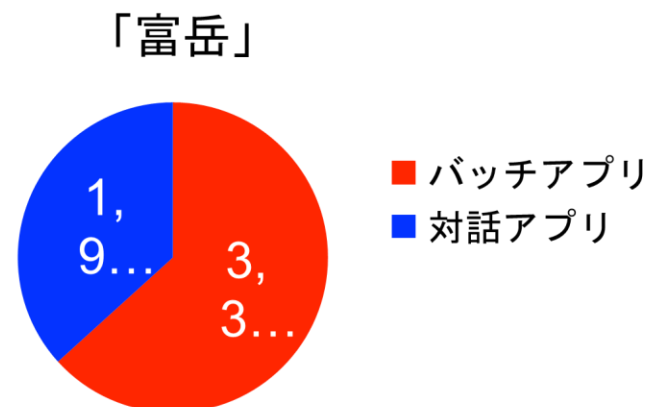
Category	Application
Climate	SCALE
Computer Aided Engineering	FDS, FFVHC-ACR, FrontFlow (blue/X), FrontISTR, OpenFOAM (Foundation/OpenCFD)
Condensed Matter Physics	ALAMODE, AkaiKKR, HΦ, mVMC, OpenMX, PHASE/0, Quantum Espresso, SALMON
Molecular Dynamics	GENESIS, GROMACS, LAMMPS, MODYLAS, PIMD
Quantum Chemistry	ABINIT-MP, Gaussian*, NTChem, SMASH
Quantum Simulation	braket
Experimental Data Processing	KIERTÄÄ

「富岳」におけるOpen OnDemandの利用状況

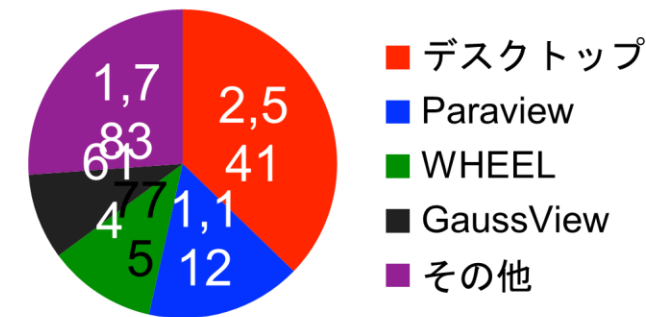


「富岳」における利用者数

現在の富岳のアカウント数は4900くらい。
上のグラフは、累積数なので、削除済のユーザも含まれている



プリポスト環境（対話のみ）



2023年6月～2024年12月の
アプリケーション起動数

OODの普及と情報交換のためのイベントを開催

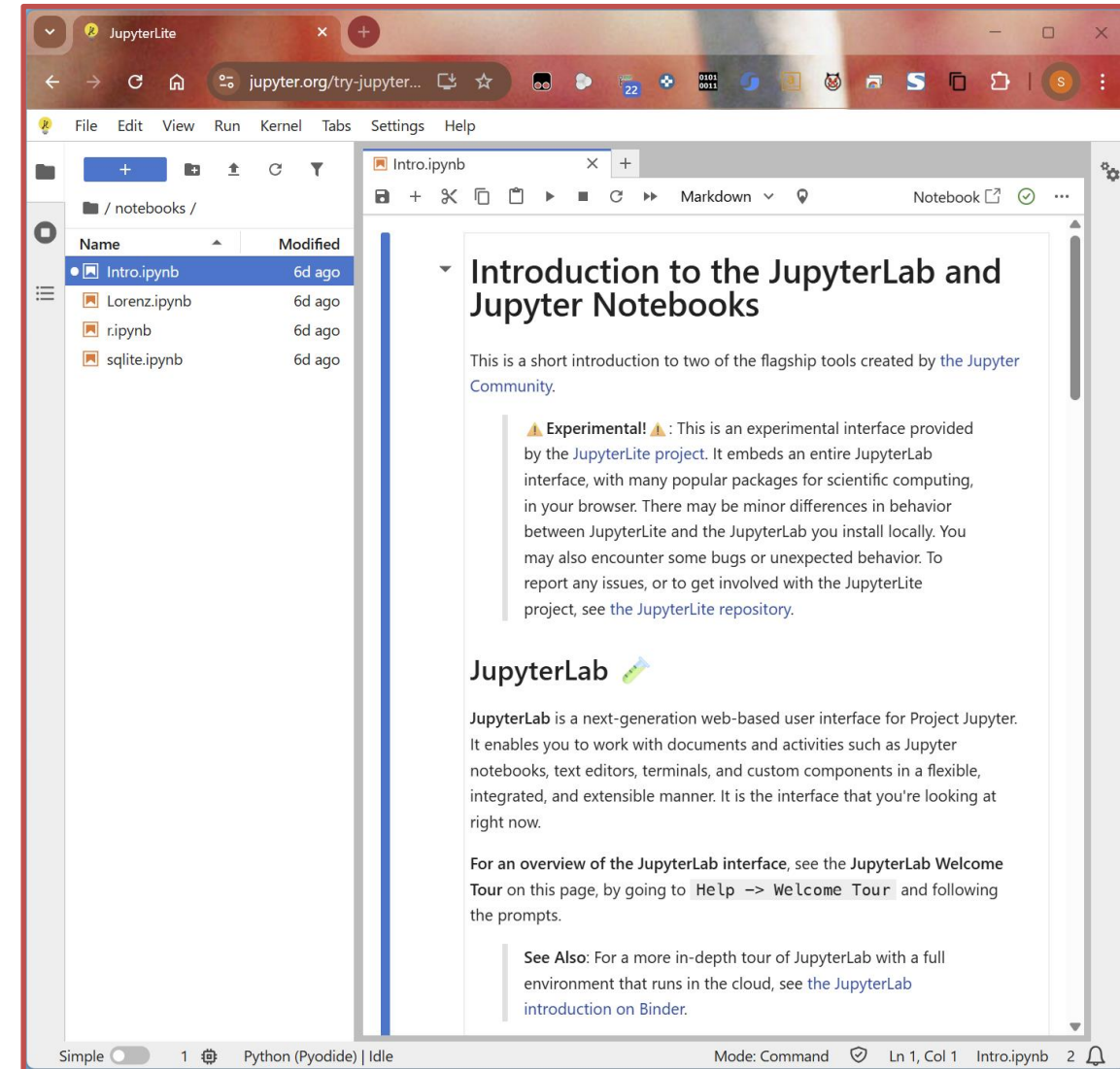
- PCクラスタコンソーシアムの部会イベントとして実施
 - PCCC Open OnDemandワークショップ、運営者や講演者として本FSのメンバーが多数参加
 - 第1回：2023年10月、理研R-CCS
 - 第2回：2025年1月、PCCC HPC OSSワークショップ in 九州大学、Kokkosと2日開催
 - 情報共有のための国内コミュニティを構築 <https://openondemandjp.github.io/>

13:00-13:10	オープニング	1：参加者135名
13:10-13:30	「Open OnDemand の概要」 中尾 昌広 (理化学研究所R-CCS)	
13:30-14:00	「Open OnDemandハンズオン」 中尾 昌広 (理化学研究所R-CCS)	
14:00-14:30	事例紹介1：「『富岳』におけるOpen OnDemandの導入と運用」 中尾 昌広 (理化学研究所R-CCS)	
14:45-15:00	事例紹介2：「三菱ケミカルにおけるOpen OnDemandへの取り組み」 杉山 肇 (三菱ケミカル株式会社)	
15:00-15:15	「クラウド環境構築支援システムVCPによるOpen OnDemand環境の簡易構築」 大江 和一 (国立情報学研究所)	
15:15-15:30	「JAXAにおけるOpen OnDemand導入の狙いと期待」 宇野 俊司 (宇宙航空研究開発機構)	
15:30-16:00	質疑応答	
16:00-	富岳見学会 (現地参加者のみ)	

8:50-9:00	オープニング	2：参加者69名
9:00-9:30	開発ロードマップ：「Open OnDemand: Connecting Computing Power with Powerful Minds」 Julie Ma (MGHPCC)	
9:30-10:00	活用事例 (1)：「『富岳』におけるOpen OnDemandの導入と運用」 中尾 昌広 (理化学研究所R-CCS)	
10:15-11:45	ハンズオン：中尾 昌広 (理化学研究所R-CCS)	
11:45-13:00	昼食	
13:00-13:20	活用事例 (2)：「玄界におけるOpen OnDemandの利用状況」 大島 聡史 (九州大学 情報基盤研究開発センター)	
13:20-13:40	活用事例 (3)：「北大クラウドウィーク内で実施したOCS+Open OnDemandハンズオンセミナー実施報告」 大江 和一 (国立情報学研究所)	
13:40-14:00	活用事例 (4)：「TSUBAMEシリーズにおけるブラウザベース利用機能の導入」 野村 哲弘 (東京科学大学)	
14:00-14:20	活用事例 (5)：「ABCIでのOpen OnDemand活用事例と今後の展望」 滝澤 真一朗 (産業技術総合研究所)	
14:50-16:00	ディスカッション	
16:00-16:30	TrinityX の紹介「Introducing TrinityX Cluster Management Software with Seamless Open OnDemand Integration」 Antoine Schonewille (ClusterVision)	
16:30-16:40	クロージング	
16:40-17:20	スーパーコンピュータ玄界 見学会	

Jupyter (JupyterLab, Jupyter Notebook)

- Webブラウザ上で使える対話型実行環境
 - Pythonの対話型利用に便利
 - (設定により) Linuxコマンドなども実行可能
 - 機械学習関係のフロントエンドとしても普及
 - おそらくHPC以外の分野の方が普及している
- HPCI計算環境の多くにもインストール済み
- ジョブスクリプトの差異を気にせず
スパコンの計算ノードを利用可能
 - (実行できるアプリ・ワークロードは限られるが)
- Jupyterの起動自体も隠蔽すれば
スパコンの利用ハードルが何もなくなる
 - 「富岳」やTSUBAME 4.0では
Open OnDemandから起動可能



研究データ基盤を活用したワークフロー

- 「研究データ基盤を活用したワークフロー」？

- 具体例

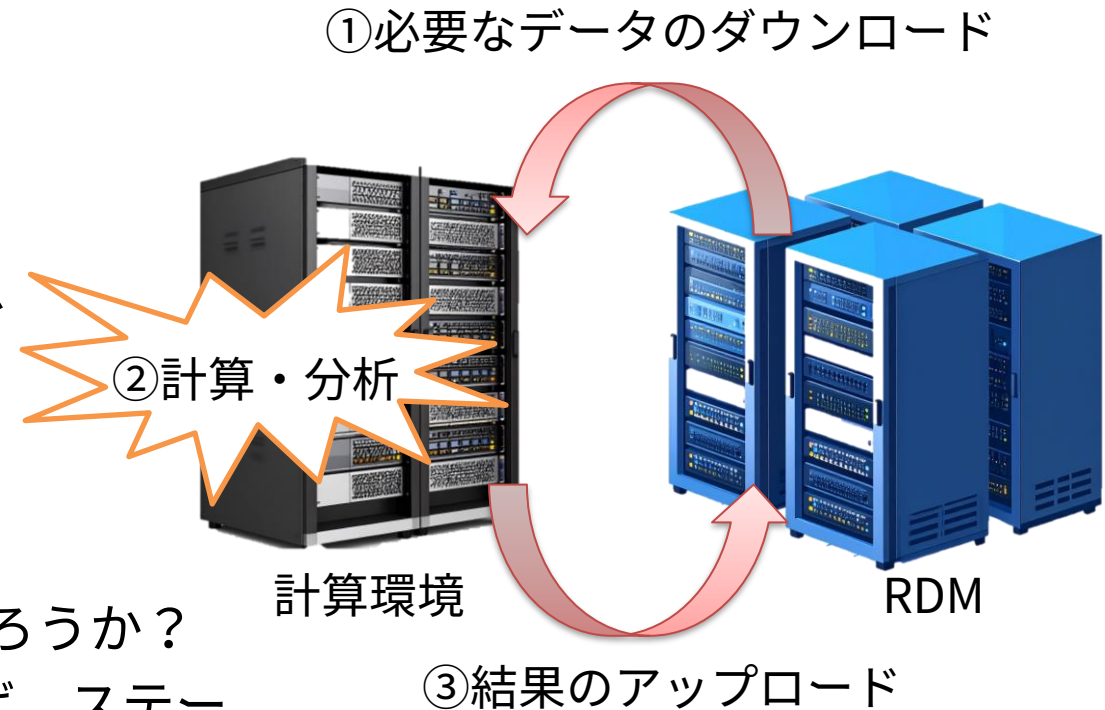
1. GakuNin RDM（以下GRDM）にデータを配置
2. 必要なデータをGRDMからスパコンにダウンロード
3. 計算ノードで計算（バッチジョブ実行）
4. 計算結果をGRDMへアップロード

ー 一連の作業をスムーズに行うにはどうすれば良いだろうか？

- ・ バッチジョブシステムにステージング機能があれば、ステージング時にデータをダウンロード・アップロードが妥当か？

↓

- ・ ステージング機能自体がない（有効にしていない）
- ・ ステージング処理はシステム内でのデータコピーのみ対応



GakuNin RDMと連携するデータ解析機能

- NIIが提供する解析環境を用いて実現する方法は確立している

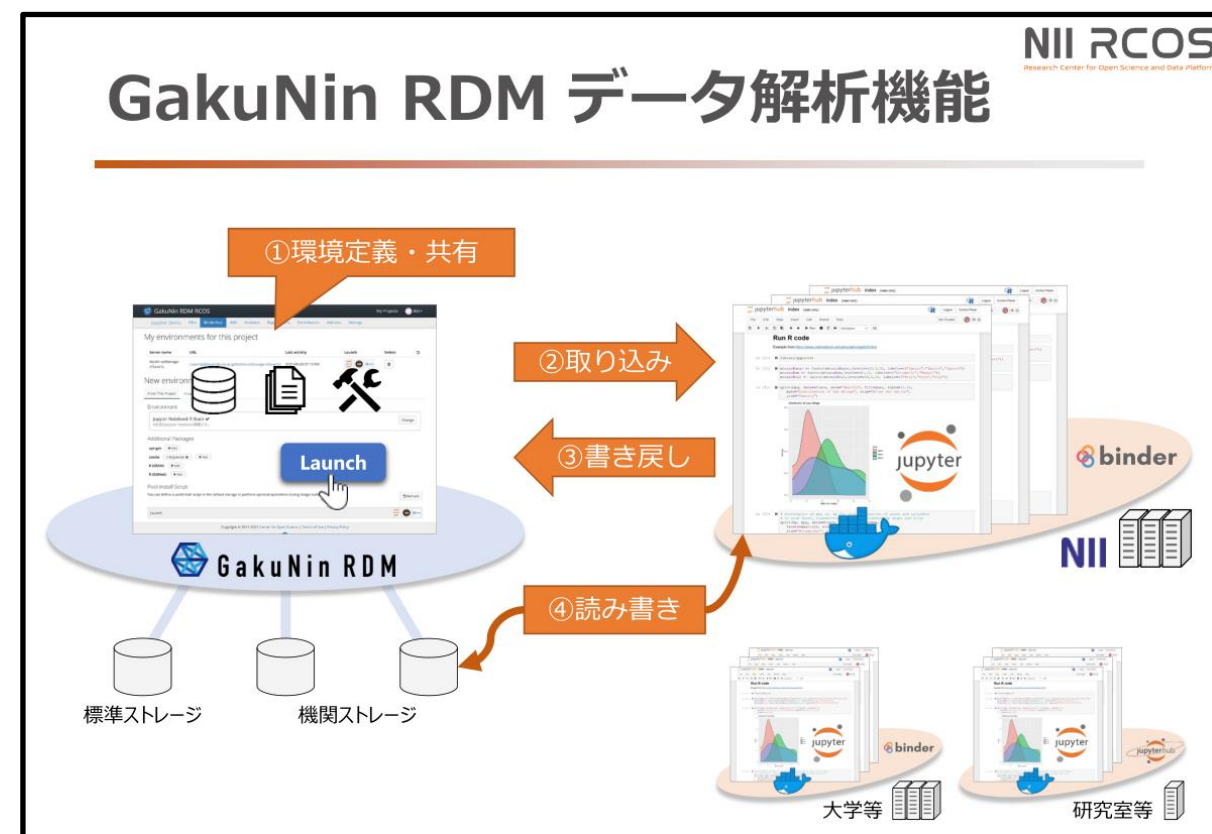
- NII自身が情報を公開

データ解析機能の概要 | GakuNin RDM サポートポータル
<https://support.rdm.nii.ac.jp/usermanual/DataAnalysis-01/>

- 仮想マシン上のJupyterLabから操作可能

- HPCI計算環境などで実行するには？

- Dockerや特定のワークフローエンジンが必要
 - Dockerを使えるようにするのは非常に難しい
 - ワークフローエンジンさえ対応していればGRDMと連携したジョブの実行は可能
(ただしFujitsu TCSは非対応)



「富岳」+Open OnDemandとGakuNin RDMの連携

- 2023年から公式にサポート済みの機能
 - <https://www.r-ccs.riken.jp/outreach/topics/20230726-1/>
 - https://riken-rccs.github.io/ondemand_fugaku/index_ja.html
- 「富岳」上のOpen OnDemandからGRDMの操作が可能
 - Web GUIで簡単にファイル転送
 - OODのオーバーヘッドはわずか
 - 計算ノード利用も含めたワークフローがサポートされているわけではない

GakuNin RDM

You can perform file operations on [GakuNin RDM](#), a research data management platform provided by the National Institute of Informatics.

- Mount Path : Mount path on Fugaku (e.g. /home/rccs-aot/a00000/abc)
- RDM Node ID : A part of URL of your project in GakuNin RDM (xxx in <https://rdm.nii.ac.jp/xxx/>)
- ROM Token : Personal access token in GakuNin RDM

GakuNin RDM

Available Action	Mount Path	RDM Node ID	RDM Token
mount			

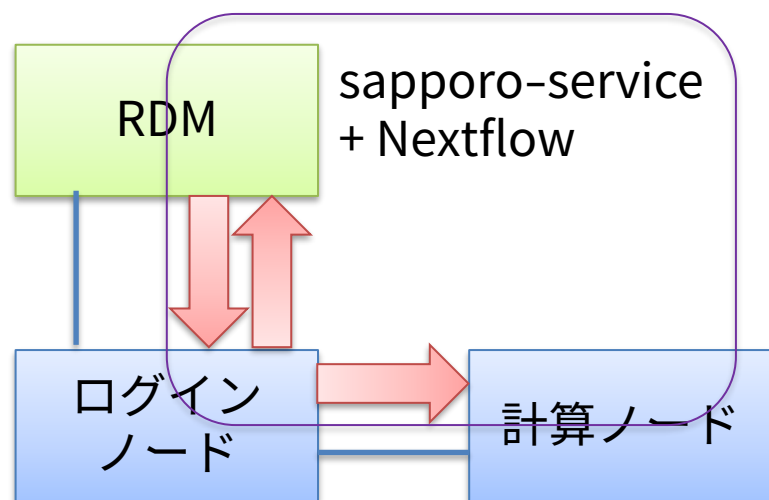
When clicking mount button, a project directory in GakuNin RDM is mounted to Fugaku.

GakuNin RDM + Fujitsu TCS連携（「富岳」/センタースパコン）

- 前述の連携機能はNIIで開発しているsapporo-serviceにより実現
 - <https://github.com/RCOSDP/CS-sapporo-service>
 - Global Alliance for Genomics & Health(GA4GH)が開発したもののfork
- sapporo-serviceは外部のワークフローエンジンを使ってジョブを実行可能
 - 複数のワークフローエンジン（タスク実行システム）に対応
 - Nextflow, Streamflow, Snakemake……
- Slurmなどを使っているスパコンではそのまま使えると思われるが、「富岳」などで用いられているFujitsu TCSには対応していない
 - 偶然にも私が所属する名大・九大もTCS
- NextflowにTCS対応機能を追加してGakuNin RDMと「富岳」等の連携を可能にした
 - Nextflow <https://www.nextflow.io/>
 - sapporo-serviceから改造版Nextflowを経由してRDM+TCSの連携ジョブを実行
 - Nextflowに対する変更は本家リポジトリへ取り込んでもらうべく作業中

実行例（新たにできるようになったこと）

- sapporo-serviceを通じてGakuNin RDMとのデータのやりとりと計算ノードでのジョブ実行からなるワークフローの実行が可能に
 - 理研「富岳」、名大「不老」、九大 玄界 で動作を確認
 - 計算ノードで実行するジョブのための設定は個別に行う必要がある（利用するジョブキューなどを設定ファイルに書く）が、Nextflowによる記述は同じものを利用可能
 - スパコン+GRDM連携を汎用的な（共通の）記述で利用可能



ユーザがログインノード上で行う操作

1. サービスプログラムの起動（httpポート待ち受け）
2. ワークフローの投入（待ち受けポートへ投入）

sapporo-service + Nextflowによる行われる処理

1. RDMからデータをダウンロード
2. TCSを操作して計算ノードへジョブ投入・終了待ち
3. RDMに結果データをアップロード

実装概要

- Nextflowの実装はGroovy（Javaベース）
- バックエンドのバッチジョブシステムごとに制御用のクラスを備える
- Slurm向けのクラスなどを元にTCS向けのクラスを追加
 - ジョブ処理コマンドの違いにあわせて変更

```

56     protected List<String> getDirectives(TaskRun task, List<String> result) {
57
58         if( task instanceof TaskArrayRun ) {
59             final arraySize = task.getArraySize()
60             result << '--array' << "0-${arraySize - 1}".toString()
61         }
62
63         result << '-J' << getJobNameFor(task)
64
65         // -o OUTFILE and no -e option => stdout and stderr merged to stdout/OUTFILE
66         result << '-o' << (task.isArray() ? '/dev/null' : quote(task.workDir.resolve(Ta
67
68         result << '--no-requeue' << '' // note: directive need to be returned as pairs
69
70         if( !hasSignalOpt(task.config) ) {
71             // see https://github.com/nextflow-io/nextflow/issues/2163
72             // and https://slurm.schedmd.com/sbatch.html#OPT_signal
73             result << '--signal' << 'B:USR2@30'
74         }
75
76         if( task.config.getCpus() > 1 ) {
77             result << '-c' << task.config.getCpus().toString()
78         }
79
80         if( task.config.getTime() ) {

```

Slurm

```

44     protected List<String> getDirectives( TaskRun task, List<String> result ) {
45         assert result !=null
46
47         if( task instanceof TaskArrayRun ) {
48             final arraySize = task.getArraySize()
49             result << '--bulk --sparam' << "0-${arraySize - 1}".toString()
50         }
51
52         result << '-N' << modName(getJobNameFor(task))
53
54         result << '-j' << ''
55         result << '-S' << ''
56         if( !task.workDir ) {
57             result << '-o' << (task.isArray() ? '/dev/null' : quote(task.workDir.resolve
58         }
59         //result << '-o' << (task.isArray() ? '/dev/null' : quote(task.workDir.resolve(
60
61         // max task duration
62         if( task.config.getTime() ) {
63             final duration = task.config.getTime()
64             result << "-L" << "elapsed=${duration.format('HH:mm:ss')}".toString()
65         }

```

TCS

例：sbatchの引数処理を参考にpjsubの引数処理を記述

and script wrapped file name
config, result)

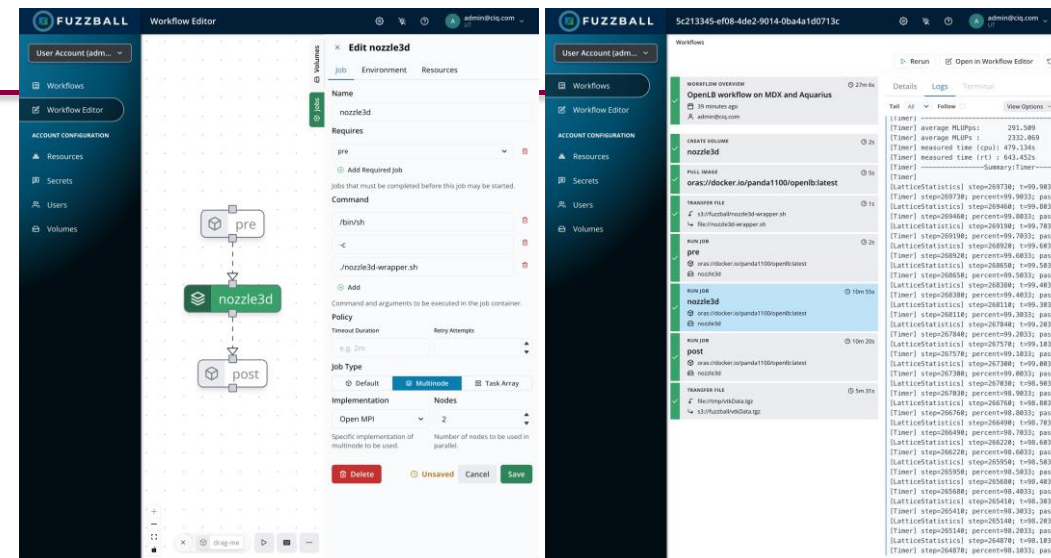
Fuzzball PoC

- 計算機クラスタのシステム管理やジョブ/ワークフロー管理に資するソフトウェアの開発もいくつか行われている
- 今回は米国CIQ社が開発中のFuzzballを対象として複数拠点HPCシステム間の統一的管理及びワークフローの概念実証（PoC）を実施
- Fuzzball
 - コンテナベースのHPCプラットフォーム
 - 直感的なWeb UIを提供
 - MPI・PGAS・GPUなどを用いた並列計算をサポート
- PoC 実施内容
 - Fuzzballを用いたmdxとAquariusの統一的な管理・利用
 - Fuzzballと既存のワークフローツールなどとの機能の比較

統一的な利用の実証

実施内容

- mdxのゲストOSにインストールされたFuzzball (Fuzzball Orchestrate)から、mdx上のVMとAquarius上の計算ノード2台を用いたワークフローを実行。



Fuzzballを用いたワークフローの作成と実行の様子

具体的な手順

1. mdx上のs3ストレージサーバからデータを取得
2. コンテナレジストリからORASプロトコルを用いてOpenLB 1.7コンテナ(SIFイメージ)を取得
3. mdx側の計算ノード1台を用いてプリプロセッシングを実行
4. Aquarius側の計算ノード2台（GPU16枚）を用いて格子ボルツマン法による流体計算フレームワークOpenLBを用いたnozzle3dシミュレーションを実行
5. mdx側の計算ノード1台を用いてポストプロセッシングを実行
6. mdx上のs3ストレージサーバに処理後のデータを保存

機能の比較

- 既存のジョブ管理システムやワークフローエンジン等より高機能
- コンテナを中心に設計されたHPCワークフローのオーケストレータの有用性を確認

	Fuzzball	Slurm (ジョブ管理システム)	Nextflow (ワークフローエンジン)	Open OnDemand
Web UI	☑	☑		☑
CLI	☑	☑	☑	☑
ワークフロー	☑		☑	
ジョブ管理	☑	☑		
対話型シェルアクセス	☑	☑		☑
コラボレーションツール	☑			
カスタムホスト名	☑			
ボリューム管理	☑			
シークレット管理	☑		☑	
ジョブアレイ	☑	☑	☑	
MPI、PGAS	☑	☑	☑	
GPU	☑	☑	☑	
高速ネットワーク	☑	☑		
エンティティ管理	☑			
クライアントSDK	☑	☑		
サンプルワークフロー	☑	☑	☑	☑
デプロイと構成	☑	☑	☑	☑
ストレージ管理	☑			

まとめ

- 当グループでは計算資源やストレージの連携に関する調査や改善の検討を実施
 - 扱う話題の都合上、定例ミーティングは資源管理グループと合同
- システムやアプリケーションの調査
 - HPC向けの調査では注目度が低めのストレージや外部接続ネットワークまで含めた性能調査を実施
 - システム間連携処理を行ううえでのボトルネックを確認し情報共有
 - 利用者が直面している課題を共有
 - （調査しきれていないものも残る：最新のAIワークロード事情など）
- 利用環境の改善・高度化に向けた調査やテスト
 - Open OnDemand・Jupyter・ワークフローエンジン等の情報を共有
 - Open OnDemandの国内普及（運用ノウハウの共有）やコミュニティ構築に貢献
 - 最新の統合ツールに対するPoCを実施
 - スパコンを使ってこなかった層へのアプローチ・新たなユーザの取り込みにつながることに期待