沖縄科学技術大学院大学 HPC 環境への エンドユーザ向けグラフィカルインタフェース

川井 敦 [1)], タユフェール エディ [1)]

1) 沖縄科学技術大学院大学学園 研究支援ディビジョン
科学計算及びデータ解析セクション

atsushi.kawai@oist.jp

Graphical Enduser Interface to OIST HPC Environment

Atsushi Kawai[1)], Eddy Taillefer[1)]

1) Scientific Computing and Data Analysis section,
Research Support Division,
Okinawa Institute of Science and Technology Graduate University

概要

沖縄科学技術大学院大学 (OIST, 以降「本学」と記す) は理論性能約 2Pflops の演算クラスタを中核とする HPC システムを運用している。われわれはこのシステムのエンドユーザに対してグラフィカルなユーザインタフェースを提供する Web アプリケーション、HighSci (ハイサイ) の開発・運用を行っている。

本学の HPC システムは研究者だけでなく院生や研究室のサポートスタッフなどにも幅広く利用されている。すべてのユーザが必ずしも Linux やその上でのコマンドライン操作に慣れ親しんでいるとは限らず、直感的でグラフィカルなインタフェースによる操作と、資源の利用状況の可視化が従来より望まれていた。

HighSci の開発は 2017 年 3 月、運用は同年 11 月にはじまった。以来ユーザからのフィードバックをとりいれながら機能拡張を続け現在に至っている。2020 年 8 月現在、演算クラスタ上にジョブを投入しているユーザのうち約 20% が HighSci を常時利用している。

これまでに実装された HighSci の機能には、例えば以下のものがある: 全演算ノードの利用状況のヒートマップ表示機能; ユーザ自身および所属研究ユニットのストレージ占有量とその内訳、およびその月次変化の表示機能; ジョブキューの状況やユーザ自身の投入したジョブの状態の表示機能; 各研究室に割り当てられたストレージへのアクセス権限を、研究室の責任者 (教授等) が自身で管理するための機能;

1 背景

1.1 OIST の研究体制と HPC 資源の利用者

沖縄科学技術大学院大学 (以降 OIST と表記する) の研究主体は 74 個 [*1] の研究ユニットである (「研究ユニット」は一般的な大学の「研究室」に相当する) 。各研究ユニットの研究分野は物理学、化学、神経科学、海洋科学、環境・生態学、数学・計算科学、分子・細胞・発生生物学など多岐に渡る。研究ユニットは学部や学科などの階層構造には組み込まれておらず、互いにフラットな関係にある。

大学院生は各学年につき 40 人程度で、彼ら彼女らは第 2 年次までの基礎過程を経ていずれかの研究ユニットに配属される。また約 10 の研究支援セクションが研究ユニットを技術面で支援する (図 1)。

原則的には研究ユニットが独自に計算機資源を所有することはなく、すべての研究ユニットは全学に共通の HPC 環境を使用する。この HPC 環境は研究支援セクションのひとつである「科学計算及びデータ解析セクション」により提供される。著者らはこのセクションのメンバーである。

1.2 OIST HPC 環境とその特異性

HPC 環境の中核をなすのは 2020 年 7 月に導入された比較的新しい演算クラスタで、約 2 Pflops の理論性能を持つ。このクラスタの他に GPU クラスタ、容量 12 PByte の長期保存用ストレージなどが、InfiniBand ネットワークで相互接続されている (図 2)。

OIST における HPC 資源の利用形態には、典型的な研究機関のそれとは大きく異なる特徴がある。すな

[*1] 2020 年 8 月現在

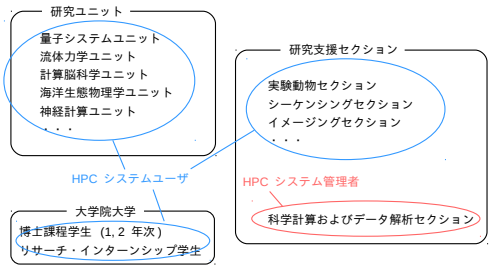

図 1　OIST 研究体制。研究ユニットが主体となって行う研究を、研究支援セクションがサポートする。研究支援セクションのひとつ「科学計算及びデータ解析セクション」が HPC 資源を提供する。すべての研究ユニットと研究支援セクションがこの資源を利用できる。

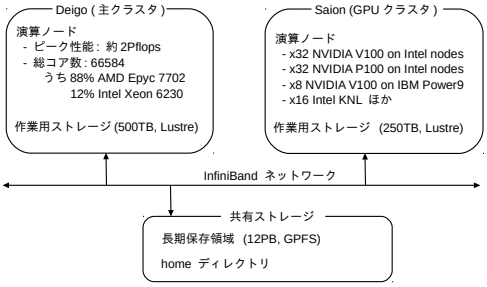

図 2　OIST の HPC 環境。主となる演算クラスタ Deigo (デイゴ) は AMD および Intel の CPU からなるクラスタ。2020 年 7 月に運用を開始した。

わち、学生や研究ユニットのメンバーであれば、原則的に誰でも利用できる。利用申請にあたって利用目的の妥当性などは審査されない。また CPU コア数やメモリサイズに課せられた上限値を超えない範囲内で、資源を無制限に利用できる。利用量に応じた課金はない。

ユーザは大きな自由を与えられている半面、つねに他のユーザに配慮して資源を利用することが求められる。他者への配慮を欠く不適切な利用は、たとえそれが意図せぬ誤操作によるものであっても可能な限り回避されねばならない。

1.3　HPC 向けグラフィカルインタフェース HighSci

上記の理由により、各ユーザが本学 HPC 資源全体の利用状況を正しく把握することが、OIST では他の研究機関にも増して重要である。しかしすべてのユーザが利用状況の把握を Linux のコマンドライン操作によって的確に行えると期待するのは現実的でない。研究ユニットの研究対象はさまざまで、研究者たちは必ずしも計算機や Linux に慣れ親しんでいるわけではない。HPC 環境へのアクセスや状態把握を誰もが容易に行えるような、何らかの直感的・視覚的なツールを、われわれ管理者サイドからユーザに提供することが望ましい。

HPC 環境のモニタリングツールやポータルは数多く知られている [1] [2] [3] [5] [9] [12] が、しかしこれらは管理者専用でエンドユーザの利用を想定していなかったり、あるいはジョブ管理など特定の機能のみに特化していてそれ以外の情報は得られないものであったりと、個々の機能は優れていても、本学のエンドユーザが必要とする機能を総括的には満たしてはいない。本学の多彩な要求に迅速かつ柔軟に対応するためには、われわれ自身でアプリケーションソフトウェアを開発するのが最適と考えられる。

以上の動機により、われわれはグラフィカルインタフェースをもつ HPC のエンドユーザ向け Web アプリケーションを開発し、これを HighSci (ハイサイ) を名づけた。以下ではこの Web アプリケーションについて詳細を説明する。

2　HighSci の全体像

2.1　システム構成

HighSci は Web アプリケーションであるので、クライアントプログラムは Web ブラウザ上で動作する。クライアントプログラムはユーザへグラフィカルなインタフェースを提供し (図:3)、またサーバ機上の Web サーバを介して各種のプログラムや外部サービスにアクセスする (図:4)。

2.2　ユーザに提供する機能

HighSci がユーザに提供する機能は多岐に渡るが、おおむね以下の 3 種に大別される

(a) 現在の状態を把握し反応・表示する機能
(b) 過去の状態の履歴を解析・表示する機能
(c) 現在の状態を変更する機能

(a) に分類される機能には例えば以下のものがある:

(a1) 演算クラスタの各ノードの現在の CPU コア利用率の表示 (図:5)
(a2) 自身の所属する研究ユニットが使用しているストレージサイズ、およびそのメンバーごとの内訳の表示 (図:3 A2)
(a3) 自身が投入したジョブの属性 (実行中/待機中、CPU コア数、メモリサイズ等) 表示 (図:3 A3)

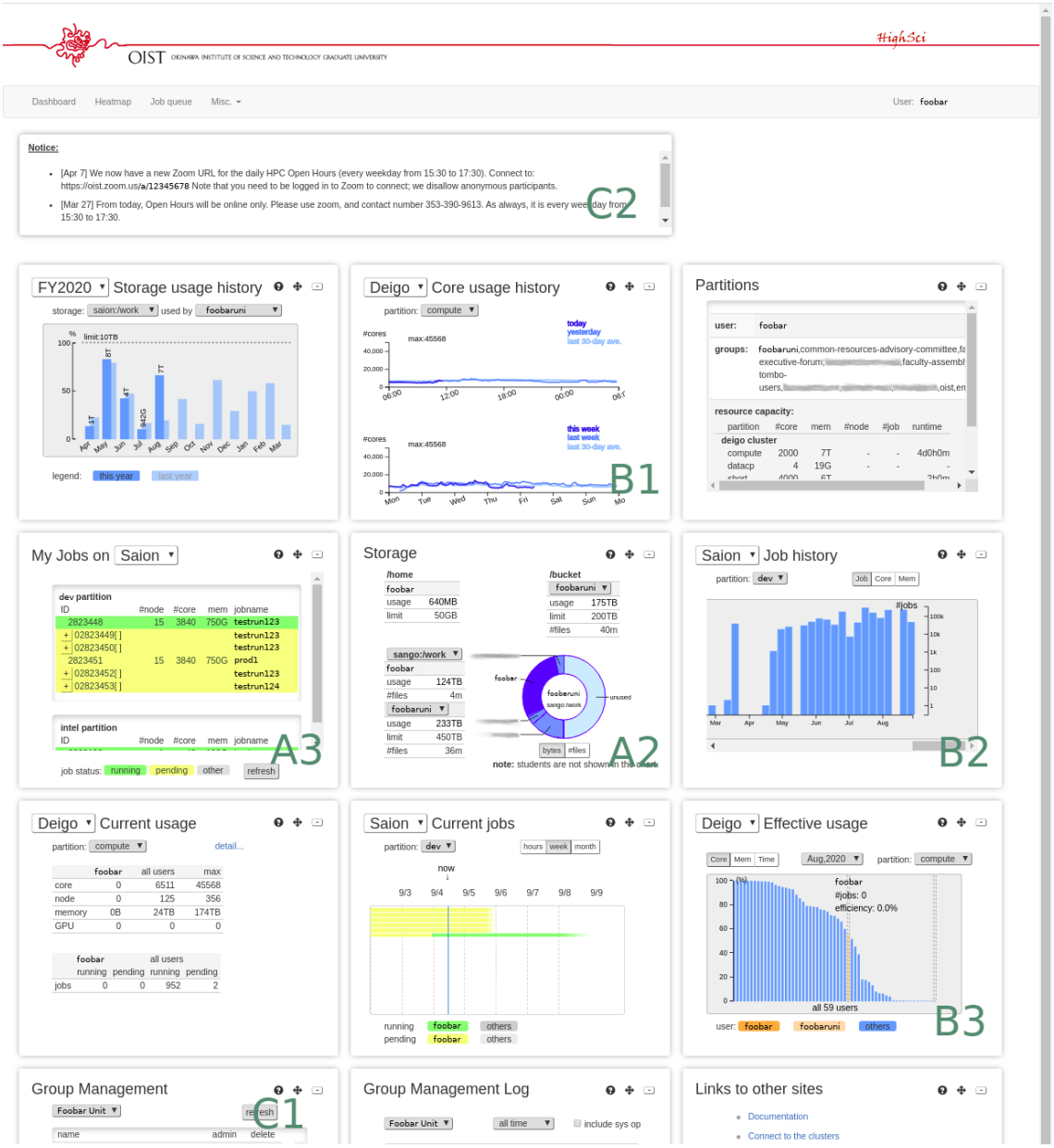

図 3　HighSci のダッシュボード (初期画面)。画面内の小さなウィンドウ (以降「パネル」と呼ぶ) それぞれが別個の機能を提供する。ユーザの好みに応じてパネルの順序を入れ替えたり、パネルをアイコン化できる。パネルでは表現の難しい複雑な機能はダッシュボードとは独立したページで扱う (例えば図 5 のヒートマップ)。

(b) に分類される機能には:

(b1) 演算クラスタの CPU コア利用率履歴の表示 (図:3 B1)

(b2) 自身がこれまでに投入したジョブの、週ごとの統計情報 (ジョブ数、CPU time、メモリサイズ) の表示 (図:3 B2)

(b3) 全ユーザが投入したジョブの、ひと月ごとの利用効率 (コア、メモリ、占有時間) 分布の表示 (図:3 B3)

等がある。(c) に分類される機能には:

(c1) 自身の研究ユニットのストレージへのアクセス権を、任意のユーザに与える/から剥奪する (図:3 C1、図:6)

(c2) 全ユーザ向けのアナウンス用メッセージを作成、公開する (図:3 C2)

がある。

ただし上記のすべての機能をすべてのユーザが利用できるわけではない。Web クライアント利用時には個人認証が行われ (ログイン操作)、利用できる機能が各人のもつ権限に応じて認可される。例えばストレージの使用量を確認できるのは、自分自身かあるいは自

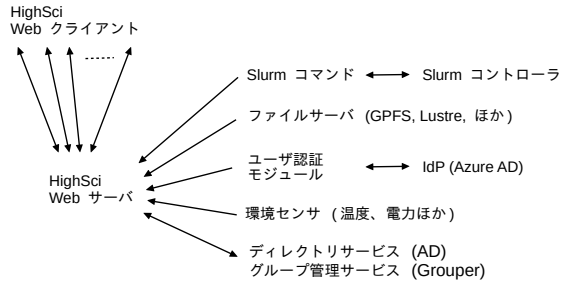

図 4　HighSci システム構成。ユーザは Web ブラウザを介してサーバ機上のバックエンドプログラムを利用する。バックエンドプログラムは Slurm コマンド、IdP、ファイルサーバ、各種環境センサ、アカウント管理関連サービスにアクセスし、得られた結果を Web サーバ経由でユーザに返す。

身の所属する研究ユニットの所有するフォルダのみに限られる。また現在投入されているジョブのうち、その所有者名を確認できるのは、自分自身かあるいは自身の所属する研究ユニットのメンバーのものに限られる。研究ユニットのストレージへの各ユーザのアクセス権を変更できるのは、そのユニットの責任者 (多くの場合は教授) のみに限られる。全ユーザ向けメッセージの発信を行えるのは、HighSci 管理者のみに限られる。このような粒度の細かい利用権限の制御は、セキュリティや利便性に配慮して HighSci 管理者が設定する。エンドユーザはこれを意識せずに HighSci を利用できる。

2.3　設計と内部の構成

HighSci の開発にあたっては HPC システムのモニタリングに特化したフレームワークやライブラリ等をアプリケーションの根幹部分に採用することは避け、プリミティブな要素の積み上げによってシステムを構築する方針をとっている。具体的にはフロントエンドは palin な JavaScript と HTML5、バックエンドは python3 で記述されている (現状では JavaScript コードの一部に jQuery に依存する部分があるが、plain な JavaScript への置き換えをすすめている)。既成のフレームワークは確かに開発コストを下げるかも知れないが、その代償として本学の環境や要求に合わせた柔軟な機能の実装や変更が難しくなる。

ただしユーザ認証やグラフィクス描画など、システム内において他の機能からの独立性の高い機能については、既存のライブラリや外部サービスを積極的に利用している。

- グラフィクス描画 – D3.js [4]
- Single Sign On の実装 – mod_auth_mellon [8]
- ActiveDirectory 連携 – Grouper [7]
- ソース・パッケージ管理 – webpack[13], git, npm

演算クラスタはジョブスケジューラとして Slurm [10] を採用しており、クラスタの状況把握は Slurm のコマンド群をバックエンドの python3 スクリプトが呼び出すことによって行っている。

ストレージの使用量は現在のところファイルサーバのもつメタ情報から抽出している。大規模ストレージ向けの管理ツール Starfish [11] が最近導入された。このツールとの連携によって使用量のより詳細な把握と管理を行えるよう、準備を進めている。

2.4　稼働状況

HighSci の開発は 2017 年 3 月に始まった。運用は同年 11 月にはじまった。以来ユーザからのフィードバックをとりいれながら機能拡張を続け現在に至っている。

2020 年 8 月現在、本学 HPC システムには 488 のユーザがアカウントを持っている。このうち約 350 のユーザが、過去 1 年間に少なくとも 1 回 HighSci を利用した。また演算ノードは典型的には常時 50∼100 ユーザ程度に利用されているが、このうち約 20% のユーザが HighSci を恒常的に利用している。

3　HighSci の機能例 — ストレージ管理

HighSci は多くの機能をもち、また現在もその機能は拡張され続けているが、ここでは一例として、主要な機能のひとつであるストレージ管理機能について紹介する。

ストレージ管理に関する機能には、使用量を把握するための機能と、アクセス権限を設定するための機能がある。前者の機能を実装するにあたっては、物理的実体の異なる複数のストレージ間の差異をどのように吸収するかが課題となる。後者については、個々のユーザに別個のアクセス権を付与し、それをユーザアカウントデータベースにどうやって反映させるかが課題となる。以下ではこれら二つの課題について、HighSci における解決方法を述べる。

3.1　ストレージ間の差異の隠蔽

本学 HPC システムのストレージは用途ごとに異なるハードウェアで実現されており、ファイルシステムもさまざま (GPFS, Lustre, Isilon OneFS) である。

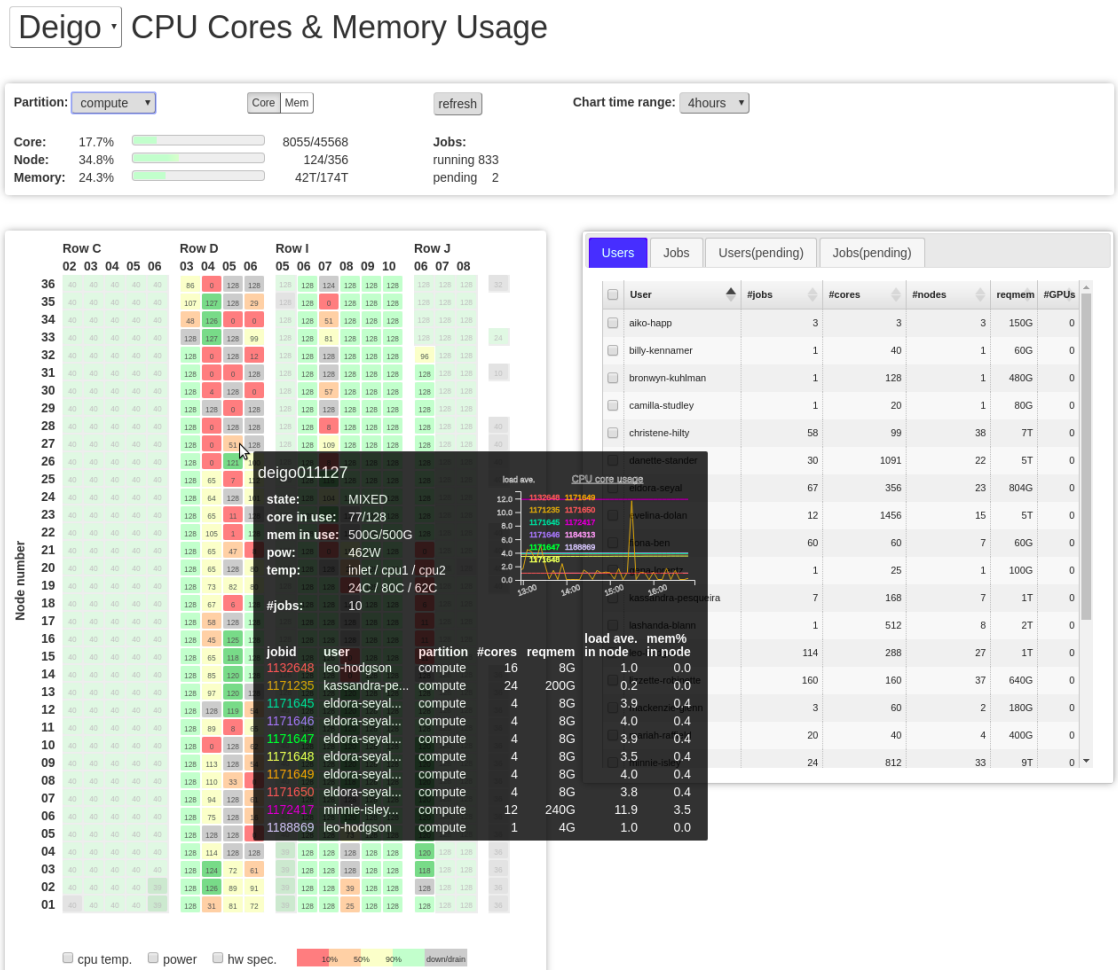

図 5　演算クラスタのヒートマップ。マス目ひとつが演算ノードひとつを表す。マス目の色がノード上の CPU コアおよびメモリの利用状況を表わす。マウスポインタをかざすとより詳細な情報がポップアップされる (図中の黒いウィンドウ)。

主要なストレージには以下の 3 種がある。

- データの長期保存用 (HDD, GPFS)
- ジョブ実行中の一時データ保存用 (SSD, Lustre)
- ホームディレクトリ用 (HDD, Isilon OneFS)

設定・取得できるメタ情報 (使用サイズ、ファイル数、クオータの設定等) やそのフォーマットはストレージごとに異なる。HighSci はそれらの差異を吸収し、いずれのストレージのメタ情報も同一の形式でユーザに提供する。ユーザはハードウェアの差異を意識することなくメタ情報を得られる。

3.2　アクセス管理

ストレージ内の各ディレクトリは、それぞれにアクセス権が設定されている。ホームディレクトリやシステムディレクトリへのアクセス権は典型的な Linux の慣習に従って設定されている。これらのディレクトリの他に、ストレージ内には各研究ユニットに専用に割り当てられたディレクトリが存在する。これらのディレクトリには本学独自の権限割り当てが行われている。

研究ユニットのディレクトリは、そのユニットに対応するグループ (unix group) が所有者として設定されており、このグループに所属するユーザだけがアクセス権を持つ。研究ユニットの正規メンバー (雇用契約に基づくメンバー) は、あらかじめこのグループに所属している。加えて、このグループには他のユニットに所属する任意のユーザをも追加することができる。この機能は複数の研究ユニットによる共同研究の際に、ユニット間でデータを共有することを念頭において実装されている。ただしグループへのユーザ追加を行えるのは、初期設定ではそのグループの管理者 (通常は教授) のみである。またたとえグループの管理者であっても正規メンバーからアクセス権を剥奪する権限は与えられていない。さらにまた、グループの管

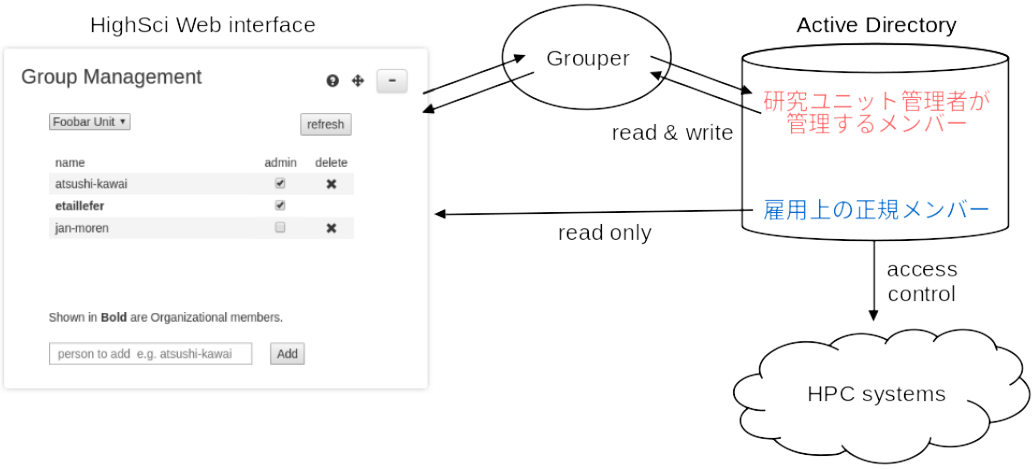

図 6　研究ユニットの管理者 (通常は教授) は自身の研究ユニット用フォルダへのアクセス権を、自身の裁量で任意のユーザに与えることができる (HPC システムの管理者はアクセス権の変更操作に関与しない)。さらに管理権限自体を他のユーザに与えることもできる。ただし雇用契約上の正規ユニットメンバーに対する操作は行えない。このような複雑な権限管理はミドルゥエア Grouper を利用して実装されている。

理者は自身の管理権限を他のメンバーに委譲することができる。

以上のような複雑で粒度の細かい権限設定を行うために、HighSci は Grouper [7] というミドルウェアを利用している。研究ユニットの正規メンバーの情報は Active Direcotry 上に記録されている。この記録は本学の雇用契約に基づくものであるので、ユーザによる書き換えは許可されていない。いっぽう Grouper は、Active Directory 上に、これとは異なる領域をもっており、この領域はユーザによる書き換えが許可されている。非正規メンバーの情報はこの領域に記録されている。とはいえユーザは記録を無制限に書き換えられるわけではなく、Grouper を介してのみ書き換えが許可されている。ユーザからの書き換え要求に不整合がある場合には Grouper がこれを拒否することで、データの整合性が保証されている。

4 まとめと今後

4.1 まとめ

われわれは OIST の HPC 利用者向けに、グラフィカルな Web インタフェース HighSci を開発・運用している。HighSci を用いれば、Linux のコマンドライン操作に不慣れなユーザであっても、演算資源やストレージ資源の利用量を直感的・視覚的に把握できる。さまざまな研究分野のユーザに利用されている OIST の HPC 環境においては、このような直感的インタフェースが特に有用である。

4.2 今後の課題

HighSci は一定の成功をおさめているが、ごく少数の学内スタッフ (実質 0.5 名未満) によって開発・運用されており、安定性や持続性の面で課題が残る。

HighSci の普及によって、本学 HPC 環境に必要とされるユーザ向け機能やその性能が具体的に洗い出されつつある。この意味で現在の HighSci を HPC 環境モニタリングツール の Proof of Concept とみなすことができる。将来的には既存の HighSci のもつ機能を要求仕様として定義し、開発や運用を外部に委託すれば、システムのより高い安定性や継続性を得られるだろう。

謝辞

HighSci の開発・運用にあたっては、本学研究支援ディビジョン 科学計算及びデータ解析セクションのメンバー Jan Moren、Pavel Puchenkov、田仲 康司から、設計方針の選定、動作環境の構築、不具合の報告、改善点の提案など、多方面に渡り多くの貢献があった。ここに感謝の意を表す。

参考文献

[1] Altair Access https://www.altairjp.co.jp/access/
[2] Atos Extreme Factory https://atos.net/en/solutions/high-performance-computing-hpc/bull-extreme-factory
[3] Bright Cluster Managr https://www.brightcomputing.com/brightclustermanager
[4] D3.js https://d3js.org/
[5] Fujitsu HPC CLuster Suite https://sp.ts.fujitsu.com/dmsp/Publications/public/ds-hpc-cluster-suite-en.pdf
[6] Grafana https://grafana.com/
[7] Grouper https://www.internet2.edu/products-services/trust-identity/grouper/
[8] mod_auth_mellon https://github.com/latchset/mod_auth_mellon/
[9] OPEN OnDemand https://openondemand.org/
[10] Slurm https://www.schedmd.com/index.php
[11] Starfish https://starfishstorage.com/
[12] Todd Evans et al.,"Comprehensive Resource Use Monitoring for HPC Systems with TACC Stats", proceedings of "2014 First International Workshop on HPC User Support Too", pp13-21, 2014.
[13] Webpack https://webpack.js.org/