

担当学生 谷澤 悠太
指導教員 滝沢 寛之 教授

1. 背景

HPCは現在、機械学習や数値シミュレーション、統計解析など様々な科学分野で地盤されている。そのため、HPCを専門としない科学者がHPCを利用する事例が多く存在する。しかしながら、このような事例の場合、HPCを使いこなすための学習による時間や労力のコストが大きく、本来の研究に費やすコストが減少してしまうため問題となっている。

このような問題に対して、米国オハイオ・スーパーコンピュータセンターはWebポータル上からHPCシステムを簡易に利用可能とする『Open OnDemand』(OOD)というソフトウェアを開発した。開発当時のOODはSLURM, Torque, PBS Pro, LSFなどのスケジューラに対応していた。しかしながら、対応していないスケジューラも多く、使える環境が限定されていたといえる。

そこで先行事例として、OODをFujitsu.TCS(スーパーコンピュータ富岳で運用されているジョブスケジューラ)へ対応させたことによる『富岳』でのOOD利用という事例がある。[1]これにより、『富岳』の利用者はHPCシステムの視覚的理解が容易になったり、利用難易度が低下したりといったメリットを得ることができた。

2. 目的

本研究では東北大学で用いられているスーパーコンピュータ『AOBA』に着目して『AOBAへのOpen OnDemand導入によるスパコン利用者支援』を考える。前述の通り、OODには『AOBA』に用いられているジョブスケジューラNQSV(Network Queuing System V)に対応するためのアダプタが存在しない。そのため、AOBA用のアダプタを実装することでAOBA利用者の支援を目的とする。

3. 実装手順

具体的な実装について、大きく4つの手順に分けて紹介する。

まず、1つ目の手順としてSLURMスケジューラを試用しているsendaiサーバでOODの動作確認を行う。この手順でOODがどのようなものか理解したり、システムの設計について調査し、具体的な実装手順を見据える。

2つ目の手順として、SLURM用に書かれたアダプタファイルを自分で書き換えることでOODにその変更内容が反映されるかどうかを確認する。

3つ目の手順として、他のスケジューラのアダプタを参考にしてNQSVのアダプタを作成する。このとき、NQSVと親和性の高いPBS Proや動作確認で試用したSLURM、アダプタ作成例が紹介されているFujitsu.TCSのアダプタなどを参考にする。

最後の手順として、手順2や手順3の過程からアダプタの構造理解やその汎用性などの検討を行う。アダプタやスケジューラの理解を深めることで、様々なスケジューラに対応できる汎用性の高いアダプタについて考察したい。

4. 進捗

4.1. OOD 導入手順

現在の進捗として、手順1で示したSLURM環境でのOODの動作確認が終了している。以降に動作確認で用いた動作環境と導入手順を示す。

OODの導入は大きく4つのステップで手順が実行されていく。1ステップ目が環境構築、2ステップ目が必要なソフトウェアのインストール、3つ目がOODに必要な認証システムの構築である。

4.1.1. 環境構築

まず、ステップ1の動作環境についてである。SLURMスケジューラを用いたsendaiサーバで実行しているOSはUbuntu22.04.1 LTSを用いてsudo権のある一般ユーザーで実行している。

4.1.2. パッケージインストール

続いてOODのインストールに際して、必要なパッケージのインストールを行う。Apache HTTP Server 2.4とRuby 2.7, Node.js 14が用いられるためインストールを行う。その後、公式のドキュメントに従ってリポジトリを追加し、OODのインストールを行う。その後、SystemctlコマンドによりApacheを起動する。UbuntuではApacheによるデフォルトページが表示されてしまうため、/etc/ood/config/ood_portal.ymlのservernameをデフォルトの状態から設定したいドメインに変更する必要がある。これにより、設定ドメインにアクセスするとOODの認証システム構築待機画面が表示され、認証システムの構築を指示される。

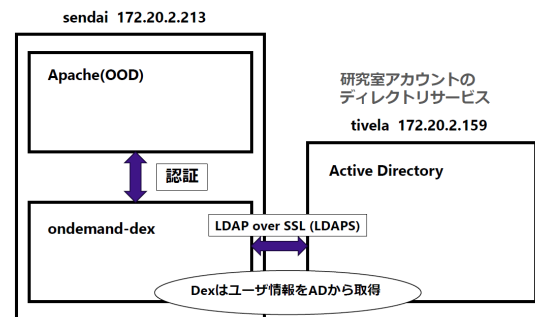


図1. 認証システムの設計

4.1.3. 認証システム

指示に従い、認証の設定を行う。今回はデフォルトで推奨されている「DexとのOpenIDコネクト」という方法を用いて認証を行う。これに付随してLDAPサーバが必要となるため、必要に応じてLDAPサーバを構築する必要がある。ここで、OpenIDコネクトとはユーザ認証用のプロトコルを指し、DexとはOpenIDコネクト認証のプロバイダー、LDAPサーバとは認証に必要なユーザ情報を管理するディレクトリサーバである。また、今回の動作環境では研究室内のアカウント情報を管理しているActiveDirectory(AD)を用いる。ADはWindows用のディレクトリサーバであり、中身はLDAPサーバと同じであるため、問題なく接続できる。

ここで、認証システムの設計を図1に示す。このように、今回はOODとDexサーバが同一のマシンで動作し、DexにtivelaサーバのADを接続して用いる。

認証システムの構築手順として、まずOnDemand-dexのパッケージのインストールを行い、続いてOnDemand-dexとADの接続、その後OnDemand-dexの起動を行う。OnDemand-dexとADの接続について、接続ファイルであるood_portal.ymlを書き換える必要があるため、以降に手順の詳細を示す。

ADとの接続設定は/etc/ood/config/ood_portal.ymlに記載されている。接続設定が書かれたファイル内のdexキー以下を図2に示す。hostキーはADのドメインとポート番号を示し、今回はSSLを用いたLDAPで接続するので636番を用いる。insecureSkipVerifyは証明書の検証の有無を示し、今回はスキップする。bindDNとbindPWはADの管理者アカウントの識別名とパスワードを示す。また、usersearch, groupsearchはそれぞれユーザ探索グループ探索に用いる情報を示していて、baseDNはそれぞれの探索の開始位置の識別名である。ADのディレクトリは木構造になっていて、すべてのユーザがDN(識別名)で認識される。参考として図3にツリー構成の例を示す。

また、Dexの背後にSQLが動いているため/etc/ood/dex/config.yamlに記載されているstorageキー配下を設定する必要がある。デフォルトで推奨されている

sqlite のインストールを行った後、図 4 に示すように storage の type の変更と config の file のパスを正しいものに修正する。

以上の作業により認証システムが機能し、AD に登録してある username と password を用いて OOD にログインすることが可能となった。

```
dex:
  connectors:
    - type: ldap
      id: ldap
      name: LDAP
      config:
        host: tivela.sc.cc.tohoku.ac.jp:636
        insecureSkipVerify: false
        bindDN: cn=admin,dc=users,dc=gp,dc=sc,dc=cc,dc=tohoku,dc=ac,dc=jp
        bindPW: ciero_cert29kbtcc
        userSearch:
          baseDN: cn=users,dc=gp,dc=sc,dc=cc,dc=tohoku,dc=ac,dc=jp
          filter: "(objectClass=person)"
          username: uid
          idAttr: uid
          emailAttr: mail
          nameAttr: gecos
          preferredUsernameAttr: uid
        groupSearch:
          baseDN: cn=users,dc=gp,dc=sc,dc=cc,dc=tohoku,dc=ac,dc=jp
          filter: "(objectClass=users)"
          userMatchers:
            - userAttr: DN
              groupAttr: member
              nameAttr: cn
```

図 2. ood.portal.yml

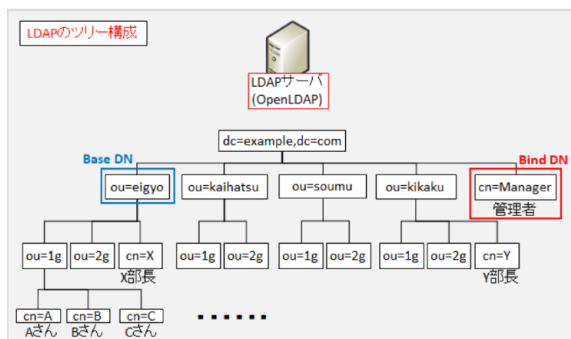


図 3. AD の構造

```
# The server name used for name-based Virtual Host
# Example:
#   servername: 'www.example.com'
# Default: null (don't use name-based Virtual Host)
#servername: null
servername: 'sendai.sc.cc.tohoku.ac.jp'
```

図 4. SQL の設定

4.2. クラスタ構築

最後のステップとしてクラスタ構成の設定を行う。クラスタの構成は/etc/ood/config/cluster.d/sendai.ymlを作成し、図 5 のように記述する。title はクラスタ名、host はログインするサーバのドメインを表し、adapter は試用するジョブスケジューラ、bin と conf は使用するスケジューラの bin ディレクトリと config ファイルのパスを表す。この設定により OOD が sendai クラスタを認識し、各アプリケーションが使用できるようになる。

```
v2:
  metadata:
    title: "Sendai"
  login:
    host: "sendai.sc.cc.tohoku.ac.jp"
  job:
    adapter: "slurm"
    bin: /usr/bin
    conf: /etc/slurm/slurm.conf
```

図 5. クラスタ構成

4.3. Job Composer

OOD の動作確認をするために設定したクラスタにジョブを投入する「Job Composer」というアプリケーションを使ってジョブを投入してみる。図 6 が「Job Composer」のジョブ選択画面である。New Job から From Specified Path を選択して、実行ファイルのパスと名前、実行クラスタを選択。その後、登録した実行ジョブを選択して Submit することでジョブの投入を行うことができる。実際に投入されたジョブの状態は「Active Job」から確認することが可能であり、実行されたジョブのステータスや実行時間などを確認することができた。

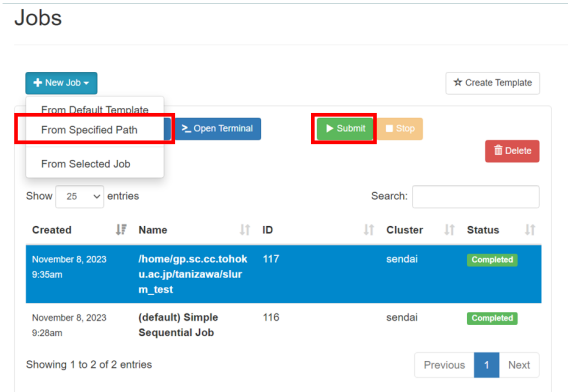


図 6. Job Composer

5. 今後の予定

短期目標として、

- ・各スケジューラのアダプタのコードの読み込みと理解
- ・SLURM アダプタの書き換えと書き換えたアダプタの動作確認

中長期目標として、

- ・NQS のアダプタを作成する際のテスト環境準備・NQS のアダプタの作成を考えている。

参考文献

- [1] 中尾 昌広, 三浦 信一, 山本 啓二. 『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』. 情報処理学会研究報告, No. 186, pp. 1-6, 9 2022.