

九州大学スーパーコンピュータと AWS クラウドサービスによる ハイブリッド計算環境の相互補完的利用方法に関する調査

南里 豪志¹⁾, 松山 和広²⁾, 田代 皓嗣³⁾, 原田 浩睦¹⁾

1) 九州大学, 2) 株式会社 Fusic, 3) アマゾンウェブサービスジャパン合同会社

nanri.takeshi.995@m.kyushu-u.ac.jp

Study on the complementary usage of hybrid computing environment with a supercomputer at Kyushu University and AWS cloud service

Takeshi Nanri¹⁾, Kazuhiro Matsuyama²⁾, Koji Tashiro³⁾, Hiroyoshi Harada¹⁾

1) Kyushu University, 2) Fusic Co., Ltd., 3) Amazon Web Services Japan

概要

本稿では、オンプレミス計算機である九州大学スーパーコンピュータとクラウド計算環境である AWS クラウドサービスを連携したハイブリッド計算環境において、それぞれの利点を活用する相互補完的な利用を実現する方法を調査し、提案した。オンプレミス計算機は、大規模かつ長時間の計算を行う際の費用対効果が高いという利点がある一方、保守等によるサービス停止期間や計算需要の急増に伴う資源割り当て待ち時間が問題となる。これに対してクラウド計算環境は、必要な時に必要な構成の計算資源を利用可能であるという可用性の高さや適用性の高さが利点である。そこで、これらの利点を活かした相互補完的な利用の基本形態として、ファイルの同期、オンプレミス計算機からのクラウド計算環境利用、およびクラウド計算環境からのオンプレミス計算機利用の3つを取り上げ、それぞれの実現方法を提案した。特にオンプレミス計算機は管理者権限が必要な操作が困難であることから、オンプレミス計算機側の操作は全て一般利用者権限のみで実行できるものを用いた。本稿では、実装方法の概要を説明するとともに、各操作の所要時間を評価した。

1 はじめに

九州大学情報基盤研究開発センター（以下、九州大学センター）は学際大規模情報基盤共同研究・共同利用拠点の一つとして、研究を目的とする計算のための計算機資源を提供している [1]。他のほとんどの拠点と同様に九州大学センターでは、運用する計算機資源を建物内に設置するオンプレミス型と呼ばれる方式を採用している。この方式は、機器の稼働率を高く維持できるだけの計算需要がある状況では高い費用対効果での運用が可能である。一方、この方法は機器の冗長性が低く保守時に一部または全部の計算サービスを停止する場合があるうえ、設置した計算資源を超える計算需要が発生すると資源割り当て待ちが発生する、といった問題がある。

一方、AWS (Amazon Web Services) [2] はクラウドコンピューティングサービスを提供している企業の一つである。このサービスは、個々の利用者がそれぞれ必要とする時に必要な量の計算資源を提供するもの

であり、特に近年は、演算アクセラレータや高性能インターコネクト等の高性能計算に必要な計算資源を選択可能となっていることから、可用性の高さや適用性の高い高性能計算環境として関心が高まっている。しかし、このような可用性の高い計算サービスの維持には、需要を大幅に上回る規模の計算資源を常に用意する必要があるため、特に大規模長時間の計算を実行する場合に、オンプレミス計算機よりも費用対効果が比較的低下する可能性が高い。

これに対して最近、オンプレミス計算機とクラウド計算環境を連携したハイブリッド計算環境が注目され始めている [3]。本稿では、このようなハイブリッド計算環境において、互いの特性を活かした相互補完的利用について検討する。このような相互補完的利用の基本形態として、ファイルの同期、オンプレミス計算機からのクラウド計算環境利用、クラウド計算環境からオンプレミス計算機の利用、の3つが考えられる。そこで本稿では、それぞれの利用形態について実現方法を提案する。通常、オンプレミス計算機では、管理

者権限での操作は許可されていない。そのため本提案では、オンプレミス計算機上での操作は全て一般利用者権限のみで実行できるものを用いる。また、利用者が自由にクラウド計算環境を選択できるように、オンプレミス計算機とクラウド計算環境はそれぞれ独立して契約することを前提とし、相互のアカウント連携などが無い状況で利用可能な実現方法を提案する。

2 オンプレミス計算機とクラウド計算環境の相互補完的利用

相互補完的利用は、オンプレミス計算機とクラウド計算環境のそれぞれの特性を活用した連携利用法である。このうち、オンプレミス計算機の特性としては、大規模長時間の計算における費用対効果の高さが挙げられる。オンプレミス計算機は、導入した計算資源を高い稼働率で運用することにより高い計算スループットを達成している。そのため、常に計算能力の限界に近い計算需要がある場合、計算機の導入コストに対して比較的高い計算能力を提供可能である。その反面、オンプレミス計算機の計算能力には余裕が無いため、保守作業等により一部または全部のサービスを利用不可となる時間帯が発生する。また、計算能力を超える計算需要がある場合、計算機の空きを待つ必要がある。その上、一度導入した計算機を数年間運用するため、通常、その間のハードウェア技術の進歩による恩恵を享受できない。

一方、クラウド計算環境の特性として、いつでも必要な計算能力を提供可能な可用性の高さと、最新ハードウェア技術を含む様々な装置から最適な構成を選択可能な適用性の高さが挙げられる。これらを実現するため、クラウド計算環境は基本的に予想需要を大幅に上回る計算能力を用意する必要があり、特に大規模長時間計算における費用対効果が低い場合が多い。

このようにオンプレミス計算機とクラウド計算環境は、片方の短所をもう片方の長所で補うことのできる相互補完的な関係にあると言える。そこで、それぞれの環境の特性を活かした相互補完的利用により、片方だけでは達成困難な高い利便性の実現が期待できる。

本稿では、オンプレミス計算機とクラウド計算環境の相互補完的利用における基本的な形態として以下の3点を挙げ、それぞれの実現方法を検討する。

ファイル同期

オンプレミス計算機のストレージ領域は、計算ノード群による高速なデータ処理に有効である。

一方クラウド計算環境上のストレージ領域は、運用停止期間がほとんど無い、永続的な利用が可能である。そこで、例えば不定期に発生するセンサーデータをクラウド計算環境のストレージ領域に収集し、オンプレミス計算機のストレージ領域と同期することで、大規模なセンサーデータの高速度処理を容易に実現可能となる。また、オンプレミス計算機における大量の計算結果をクラウド計算環境のストレージ領域に同期することで、常時アクセス可能な研究データとして公開することが可能となる。

オンプレミス計算機からのクラウド計算環境利用

オンプレミス計算機の計算能力を超える計算需要が発生した際、緊急性の高い計算を一時的にクラウド計算環境に移して実行することを可能とする。これにより、基本的な計算の費用対効果の高さと、計算機需要の急増への対応能力を実現可能となる。また、オンプレミス計算機の計算の一部のみをクラウド計算環境の最新ハードウェアで実行する、といった用途も考えられる。

クラウド計算環境からのオンプレミス計算機利用

クラウド計算環境上の装置やソフトウェアを組み合わせて実装されたクラウドネイティブと呼ばれるアプリケーションのうち、特に計算負荷が高い部分のみをオンプレミス計算機で実行させることにより、計算に要する費用の低減を図ることが出来る。

3 相互補完的利用の実現方法

3.1 ハイブリッド計算環境の準備

本節では、ハイブリッド計算環境における相互補完的利用の実現方法の概要を説明する。実際の操作に必要なコマンドやスクリプトなどの詳細な情報は、GitHubで公開している [6]。

本稿で想定するハイブリッド計算環境は、Linuxが動作するオンプレミス計算機とAWS上に構築したクラウド計算環境である。このうちオンプレミス計算機には、AWSに対する遠隔操作のためのツールであるAWSコマンドラインインタフェース(AWS CLI) [4]と、AWS上のクラスタを遠隔管理するツールParallelCluster [5]を予め導入する。これらのツールは一般利用者が自分のホームディレクトリにインストールすることが可能である。

一方、クラウド計算環境は、AWSの計算サービス

である EC2 とストレージサービスである S3 を組み合わせて利用する。オンプレミス計算機からクラウド計算環境の遠隔操作を行うために必要なアクセスキーとシークレットキーを予めコンソールで作成し、取得する。さらに、オンプレミス計算機上の AWS CLI でプロフィールを作成し、これらのキーを登録することで、利用可能となる。

なお、本提案における相互補完的利用の実現方法は、基本的にオンプレミス計算機からクラウド計算環境に対して処理を依頼することで各操作を実行する。これは、以下の二つの理由による。まず、オンプレミス計算機は保守等で利用できない場合があるのに対し、クラウド計算環境はほぼ常時利用可能である。そのため、操作の時点で利用可能である可能性の高いクラウド計算環境の方が遠隔操作対象として適している。また、遠隔からオンプレミス計算環境を利用するには、秘密鍵をクラウド計算環境にコピーするか、もしくは認証プロキシなどの代理認証方式を用意する必要がある。前者はセキュリティ上の問題からほとんどの計算機センターで推奨されていない。また後者は管理者権限による設定が必要であり、対象の計算機センターで予め設定されていない場合、利用できない。

3.2 オンプレミス計算機とクラウド計算環境のファイル同期

オンプレミス計算機とクラウド計算環境の間でファイルを同期するコマンドとして、AWS CLI には `aws s3` が用意されている。これは、同期元と同期先のストレージ領域を指定すると、その中の各ファイル群について、ファイルの大きさや更新日時に応じて転送の要否が判断され、更新が必要と判断されたファイルのみ、同期元から同期先に転送される。ディレクトリ内の階層的なファイル群を対象とする場合でも再帰的に処理される。なお、同期先から同期元への転送は行われなため、逆向きの同期が必要な場合、方向を入れ替えたコマンドを実行する。

3.3 オンプレミス計算機からクラウド計算環境へのジョブ投入

ここでは、オンプレミス計算機からクラウド計算環境上の計算クラスタへのジョブ投入を実現する方法を紹介する。今回想定しているジョブ投入の流れを図 1 に示す。

クラウド計算環境へのジョブ投入を希望する利用者は、まず投入する各ジョブを一意に識別するためのジョブ名を、例えば `job0001`、`job0002` のように、ジョブ毎に決定しておく。その後、オンプレミス計算

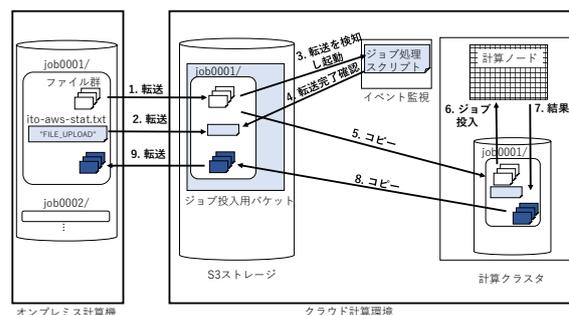


図 1 オンプレミス計算機からクラウド計算環境へのジョブ投入の流れ

機上でジョブ名をディレクトリ名とするディレクトリ内に、プログラムファイル、データファイルなど、必要なファイルを作成し、さらにジョブスクリプトを `ito-asw-run.sh` という名前で作成して、予めクラウド計算環境の S3 ストレージ領域に用意したジョブ投入用のバケットに、そのディレクトリ全体を転送する (図中 1.)。

次に、ジョブの状態を保存するプレインテキストファイルとして `ito-aws-stat.txt` という名前のファイルと同じディレクトリに作成し、中に `FILE_UPLOAD` という文字列を保存して、ジョブ投入用バケットに転送する (図中 2.)。

このバケットは AWS のイベント検知サービス EventBridge の機能を用いて監視されており、新しいファイルが追加されると、予め設定されていたジョブ処理スクリプトが起動する (図中 3.)。そのスクリプト中で、`ito-aws-stat.txt` の転送完了を確認し、このジョブのファイルを全て計算クラスタのストレージにコピーし (図中 5.)、計算ノードに対してジョブスクリプト `ito-aws-run.sh` を投入する (図中 6.)。

ジョブが終了すると (図中 7.)、Slurm のエピローグ機能により、計算クラスタのストレージとジョブ投入用バケットのファイルを同期し (図中 8.)、`ito-aws-stat.txt` の中身を `JOB_COMPLETE` に上書きする (図中 9.)。利用者はジョブ投入用バケットの `ito-aws-stat.txt` を参照することでジョブの完了を確認し、必要に応じて結果ファイルをオンプレミス計算機に転送する (図中 10.)。

このジョブ投入の流れを実現するために、利用者はあらかじめ、クラウド計算環境の計算クラスタを構築しておく必要がある。これは、鍵ペアの作成、計算ク

ラスタが使用する内部ネットワーク設定、およびインターネット接続設定を行ったのち、ParallelCluster の計算クラスタ作成コマンドで構築できる。さらに、S3 ストレージ領域上のジョブ投入用バケット作成、そのバケットを監視する EventBridge 設定、およびジョブ終了時のエビログ機能設定により、前述の流れによるジョブ投入が可能となる。

3.4 クラウド計算環境からオンプレミス計算機へのジョブ投入

クラウド計算環境からオンプレミス計算機へのジョブ投入は、前節で説明したものと同様の流れを、逆の方向に実行するものである。今回想定しているジョブの流れを図 2 に示す。

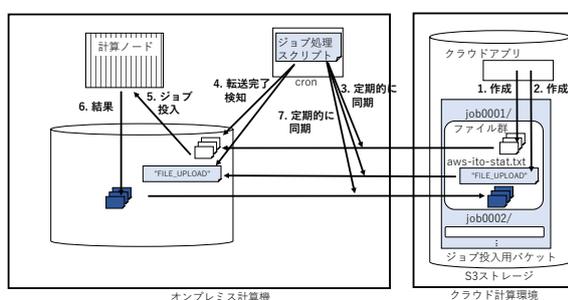


図 2 クラウド計算環境からオンプレミス計算機へのジョブ投入の流れ

前節のオンプレミス計算機からクラウド計算環境へのジョブ投入との違いは、オンプレミス計算機には EventBridge のようなイベント検知機能が無い点である。そこで本提案では、Linux の cron デーモンを利用して、定期的に S3 バケットの監視する方法を用いる。これは、crontab コマンドにより、一定の時間 (例えば 10 分) おきにジョブ監視スクリプトを実行するよう設定するものである。このスクリプトはクラウド環境上のジョブ投入バケットとオンプレミス計算機のジョブ処理ディレクトリの間で同期をとり (図中 3.)、新しいディレクトリが追加されると、状態を示すプレインテキストファイル aws-ito-stat.txt の到着を確認して (図中 4.)、aws-ito-run.sh をジョブスクリプトとして投入する (図中 5.)。さらにこのスクリプトは、ジョブが終了したことを確認すると、aws-ito-stat.txt の中身を JOB_COMPLETE に変更したうえで、ジョブのディレクトリをジョブ投入用ディレクトリと同期することによって、ジョブの結果

ファイルをクラウド環境に転送する (図中 7.)。

4 むすび

本稿では、オンプレミス計算機とクラウド計算環境を連携したハイブリッド計算環境において、互いの特性を活かした相互補完的利用の方法を調査し、基本的な利用形態として、ファイル同期、オンプレミス計算機からクラウド計算環境へのジョブ投入、およびクラウド計算環境からオンプレミス計算機へのジョブ投入のそれぞれについて、実現方法を提案した。ここで提案した方法は、オンプレミス計算機における操作として全て一般利用者権限で行えるもののみを用いている。そのため、オンプレミス計算機としては、今回試した九州大学のスーパーコンピュータ ITO だけでなく、ほとんどの Linux 計算機を利用可能である。一方、クラウド計算環境については、今回提案した方法では AWS のイベント検知機構を用いて遠隔から投入されたジョブの自動検知を実現している。しかし、この自動検知については、cron デーモンによる方法に置き換えることで、AWS 以外のクラウド環境も利用可能となると考えられる。

今後は、提案した方法の各操作に要する所要時間を評価し、実用性を検証するとともに、今回試した環境以外での実装を試みる。また、今回提案した基本的な利用形態をもとにした、具体的な相互補完的利用の事例を検討する。

謝辞

本調査にあたっては、アマゾンウェブサービスジャパン合同会社の宇津井峻氏、櫻田武嗣氏、佐々木啓氏、および株式会社 Fusic の新川拓也氏、早崎司氏、榎原竜之輔氏、室井慎太郎氏、藤村直美氏に助言と協力を頂きました。また、本調査は九州大学情報基盤研究開発センター研究用計算機システムの共同研究制度を利用しました。

参考文献

- [1] 上田将嗣, 小野真, 平島智将, 原田浩睦, 南里豪志, 新スーパーコンピュータ「ITO システム」の紹介, 大学 ICT 推進協議会 2017 年度年次大会
- [2] Amazon Web Service, <https://aws.amazon.com/jp/>
- [3] 滝澤真一郎, 清水正明, 中田秀基, 松葉浩也, 高野了成, セキュアな AI/HPC クラウドバースティング実現に向けた検討, 研究報告ハイパフォー

マンスコンピューティング (HPC) ,2022-HPC-185(17),1-7 (2022-07-20) , 2188-8841

- [4] AWS Command Line Interface、https://docs.aws.amazon.com/ja_jp/cli/
- [5] AWS ParallelCluster、https://docs.aws.amazon.com/ja_jp/parallelcluster/latest/ug/install.html
- [6] GitHub awsito <https://github.com/tnanri/awsito>