

# ハイブリッドクラウド環境における段階的クラウドバースティング方式の提案と評価

和田清美<sup>1</sup> 林真一<sup>1</sup>

**概要** : オンプレミス内の大量データを非定常分析する場合、クラウドバースティングにより低コストで高速化するための課題は、オンプレミスの計算リソース過負荷により目標クエリ処理時間を超過する前にクラウドバースティングすることと、帯域に上限があるオンプレミス・パブリッククラウド間で、大量データをパブリッククラウドへコピーすることである。本報告では、クラウドからオンプレミスへのデータアクセスによりコピー完了を待たずに分析を開始し、バックグラウンドのデータコピー完了後にパブリッククラウドのデータにアクセスする、段階的クラウドバースティング方式を提案する。実環境で本方式を適用した結果、クラウドバースティングで追加する DB ノード数の上限を設定することで、クラウドでのクエリ処理時間の長期化を防止できることを確認した。

**キーワード** : データ分析, クラウドバースティング, クエリ処理, バックグラウンドコピー, DB

## Proposal and Evaluation for Step-by-Step Cloud Bursting Method on Hybrid Cloud

KIYOMI WADA<sup>1</sup> SHINICHI HAYASHI<sup>1</sup>

**Abstract**: When performing nonroutine analysis of large amounts of data on-premises, there are challenges in cloud bursting for low-cost and fast processing. The first challenge is to perform cloud bursting before the target query processing time is exceeded due to overloading of computation resources on-premises. The second is to copy large amounts of data to the public cloud while there is a bandwidth limit between the on-premises and the public cloud. In this report, we propose a step-by-step cloud bursting method. The proposed method accesses on-premises data from the cloud to start analysis without waiting until the copying is completed, and accesses public cloud data after the background data copying is completed. We have applied the proposed method in an actual environment and confirmed that it can prevent prolonged query processing times in the cloud by setting an upper limit on the number of DB nodes to be added in cloud bursting.

**Keywords**: Data Analysis, Cloud Bursting, Query Processing, Background Copy, DB

### 1. はじめに

従来、Information Technology(IT)システムを活用する企業はオンプレミスデータセンタに IT システムを構築、運用してきたが、スケーラビリティやサービスの先進性で優れるパブリッククラウドの普及に伴い、オンプレミスデータセンタ

の所有者もパブリッククラウドを活用し始めている。一般的に構築、運用する IT システムが大規模な場合に、負荷が安定しており長時間稼働し続ける IT システムはパブリッククラウドよりオンプレミスデータセンタで稼働させる方が安価となり、一時的に稼働する IT システムはパブリッククラウドで動作させる方が安価となる。負荷変動が大きいデータ

<sup>1</sup> (株)日立製作所 研究開発グループ  
Hitachi Ltd. Research & Development Group, Kokubunji-shi, Tokyo  
185-8601, Japan  
i) Amazon Web Services は米国およびその他の国における Amazon

Technologies, Inc.の登録商標である。

分析処理において、定常的な負荷をオンプレミスデータセンタで処理し、一時的に大きくなる負荷をクラウドで処理するクラウドバースティングにより、コストを削減できる可能性がある。そのためには、オンプレミスのデータセンタ内の計算リソースが過負荷になる前にクラウドバースティングできなければならない。また帯域に上限があるオンプレミス・パブリッククラウド間で大量データを転送しなければならない。本報告では、大規模データのオンライン分析処理に対して、クラウドバースティングで高速処理を実現するための課題と解決方法を提案し、実現性を検証した結果を示す。

## 2. クラウドバースティングの既存技術と課題

### 2.1 本研究が対象とするデータ分析システムの要件

本研究が対象とするデータ分析システムは、各拠点のデータを集約して一元管理するデータベースと、本データベースにアクセスしてデータ分析するアプリケーションから成る。データ分析者は、レポート作成指示などの非定常業務で、サンプルデータを使って分析方法を試行錯誤し、最終的に対象となる全データで分析する。ここでは2つの課題がある。

1 つめは大量データ分析高速化・安定化(図 1)である。ネットワーク経由でアクセス可能なデータ分析システムでは、利用者数が増加するため、データ分析システムの過負荷が想定される。この場合、データ分析に長時間かかり、分析者の業務が遅延する。また、分析者の増減により分析時間が変動し、分析者が分析計画を立てにくくなり業務効率が低下する。2 つめはコスト削減(図 2)である。レポート作成指示により一時的に利用者数は増加するが、データ分析者は夜間に分析しない。このため、システムは夜間に使用されないが、ピーク負荷を処理可能なシステムリソースを用意しておく必要がある。

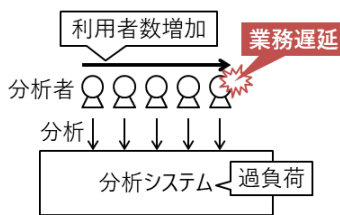


図 1 大量データ分析高速化・安定化

Figure 1 High-speed, stable mass data analysis

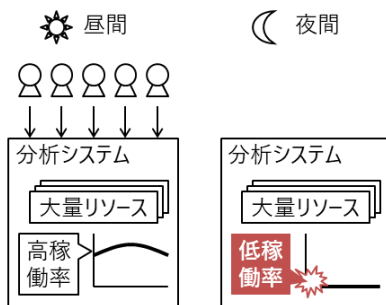


図 2 コスト削減

Figure 2 Reduce costs

### 2.2 従来技術と課題

2.1 節のデータ分析システム要件を解決するためにクラウドバースティング技術[1]が提案されている。

図 3 はクラウドバースティングを実現するためのオンプレミスとクラウドからなるシステム構成である。図 3 において、オンプレミスのデータ分析システムで利用者の増加により負荷が増加してリソースが枯渇し、アプリケーションがクラウドバースティングするとき、Database(DB)がクエリ処理のためにオンプレミスデータにアクセスするとクエリ処理時間が長くなるため、パブリッククラウドへデータをコピーし、DB をクラウドバースティングする。

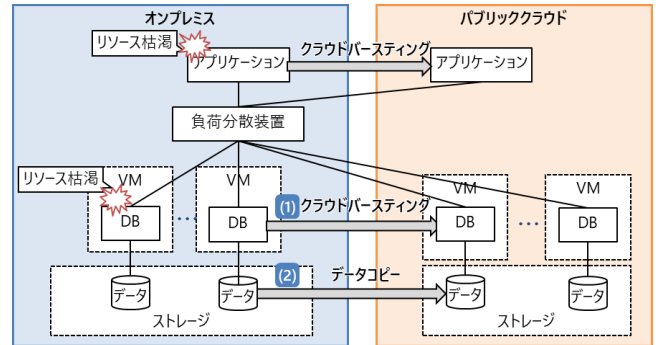


図 3 クラウドバースティングの概要

Figure 3 Overview of Cloud Bursting

DB のクラウドバースティングとデータコピーにおいて、下記(1)、(2)の課題がある。

#### (1) リクエスト増加からクラウドバースティングでの処理開始までの時間短縮

データ分析リクエストが増加し多数のクエリが負荷分散装置に投入されると、DB をクラウドバースティングし多数のクエリを DB に分配して並列実行する。ところが DB のクラウドバースティングが完了する前に多数リクエストが投入されると、各 DB のクエリ処理数が増加するため、全体処理時間が長くなる。

このため、リクエスト増加予測に基づいて事前に DB をクラウドバースティングする、あるいはリクエスト増加を検知後に迅速にクラウドバースティングする必要がある。

#### (2) 帯域に上限があるオンプレミス・パブリッククラウド間の大量データ転送待ち時間短縮

オンプレミスのデータをパブリッククラウド内のストレージに格納することで、パブリッククラウドの DB は高速にデータアクセスできる。しかし、オンプレミスとパブリッククラウド間帯域には上限があるため、大量データ転送に時間がかかる。そこで常時パブリッククラウド上のストレージにデータを格納すると、クラウドバースティングしないときもパブリッククラウドのリソース利用料金がかかるため、コストがかかる。

このため、クラウドバースティング時に、帯域に上限があるオンプレミス・パブリッククラウド間で大量データ転送待ち時間を短縮する必要がある。

### 3. 提案方式

2章の課題を解決するため、クラウドからオンプレミスへのデータアクセスによりコピー完了を待たずに分析を開始し、バックグラウンドのデータコピー完了後にパブリッククラウドのデータにアクセスする段階的クラウドバースティング方式を提案する。

図4は提案方式の段階的クラウドバースティングの概要とシステム構成である。Online Analytical Processing(OLAP)等の大規模データ処理向けに、オンプレミスのサーバと共有ストレージ、パブリッククラウドのコンピュータとストレージを活用したハイブリッド環境において、サービス利用者が目標クエリ処理時間を設定すると、クエリ処理時間とリソース使用状況などの性能情報を収集し、目標クエリ処理時間超過前にリソース不足を検知し、オンプレミス・パブリッククラウド跨り構成で迅速にクラウドバースティングする。段階的クラウドバースティングは図4のStep1~3を実行する。

Step1はオンプレミスのリソース不足時に、DBをクラウドバースティングし、パブリッククラウドのDBはオンプレミスのストレージにリモートアクセスする。またパブリッククラウドへオンプレミスデータのバックグラウンドコピーを開始する。Step1では、コピー完了を待たずに処理を開始することができる。

Step2はオンプレミスとパブリッククラウド間の通信帯域が逼迫したときに、読み出しデータの一部をキャッシュできるVM上のDBに切り替える。Step2では、キャッシュデータ参照によりクエリ処理時間が短くなり、オンプレミスとパブリッククラウド間通信量を削減できる。

Step3はバックグラウンドコピー完了後にパブリッククラウドのデータを読み出すDBに切り替える。Step3では、オンプレミス・パブリッククラウド間データ通信からパブリッククラウド内データ通信に変更するため、通信時間を含むクエリ処理時間が短くなり、DBのスケラビリティが上がる。

本提案方式により、安価なオンプレミスで定常負荷を処理し、不定期な高負荷を一時利用できるパブリッククラウドで処理することで高速化とコスト削減の両立が可能になる。

### 4. 提案方式に対する実現性検証

提案方式のStep1において、パブリッククラウドへのデータコピー完了を待たずにクラウドバースティングし、バックグラウンドコピーを開始するとき、帯域に上限があるオンプレミス・パブリッククラウド間で、クエリ処理とデータコピー処理の同時実行による性能影響を確認し実現を検証する。

#### 4.1 検証システム構成

本節では図4のStep1のDBスケールアウトとバックグラウンドコピーの実現性検証のためのシステム構成を示す。

図5は、オンプレミスのデータセンタの共有ストレージ

とパブリッククラウドAWSを専用線Direct Connect[2]で接続したハイブリッドクラウド環境で、段階的クラウドバースティング実現性検証のためのシステム構成である。パブリッククラウドのDBがオンプレミスの共有ストレージ内ボリュームからデータを読み出ししながら、オンプレミス共有ストレージ内ボリュームからパブリッククラウド内ストレージボリュームへデータをコピーする。

DBはPostgreSQLを使用し、DBをスケールアウトするためにPostgreSQLの負荷分散ミドルウェアPgpool-II[3]を使用する。Pgpool-IIは1つ以上のDBと接続し、リクエストのクエリを受け付けて、接続先DBへクエリを割り当てる。接続先のDBはそれぞれ全データを保持し、各DBは互いに独立してクエリ処理を実行できる。

バックグラウンドコピーでは、マスターデータを格納したボリュームのコピーボリュームを、パブリッククラウドのAmazon Elastic Block Store(EBS)ボリュームへコピーする。ここでは、オンプレミスのボリュームをマウントして、Linux cpコマンドを使って必要なデータファイルをコピーする。

表1は実行条件である。クエリが特定のDBノードに偏らず多くのクエリが実行可能とするため、vCPU数が多く、データがキャッシュされずストレージにアクセスするようメモリ容量が小さいインスタンスタイプ[4]を選択する。

表1 実行条件

Table 1 Execution Conditions.

項目	内容	詳細
インスタンス種別	c5.2xlarge	vCPU=8 メモリ=16GiB
EBS種別	gp3	3000IOPS 125MiB/s
オンプレミスとパブリッククラウド間通信	Direct Connect	10Gbps

#### 4.2 評価方法

評価プログラムはOLAP向けベンチマークプログラムのStar Schema Benchmark(SSB)[5]を使用する。SSBは、OLAPで使用される代表的な13種類のSQL文からなる。今回の評価では、同一クエリ(Query1-1)を同時に複数実行する。ファクトテーブルの総データサイズは100GBとする。

図5の検証システム構成で、帯域に上限があるオンプレミス・パブリッククラウド間で、クエリ処理とデータコピーを同時に実行するときの性能影響を確認する。

同時実行クエリ数は、DBノードが過負荷にならないようにvCPU数である8より少ない4クエリとする。また、目標クエリ処理時間は300秒とし、目標クエリ処理時間を超過する前にDBをスケールアウトする。

図6はオンプレミスとパブリッククラウド間の通信性能情報取得項目である。DBのクエリ処理時間とコピー時間、オンプレミスのストレージからパブリッククラウドのDBへの入力データ量を取得する。

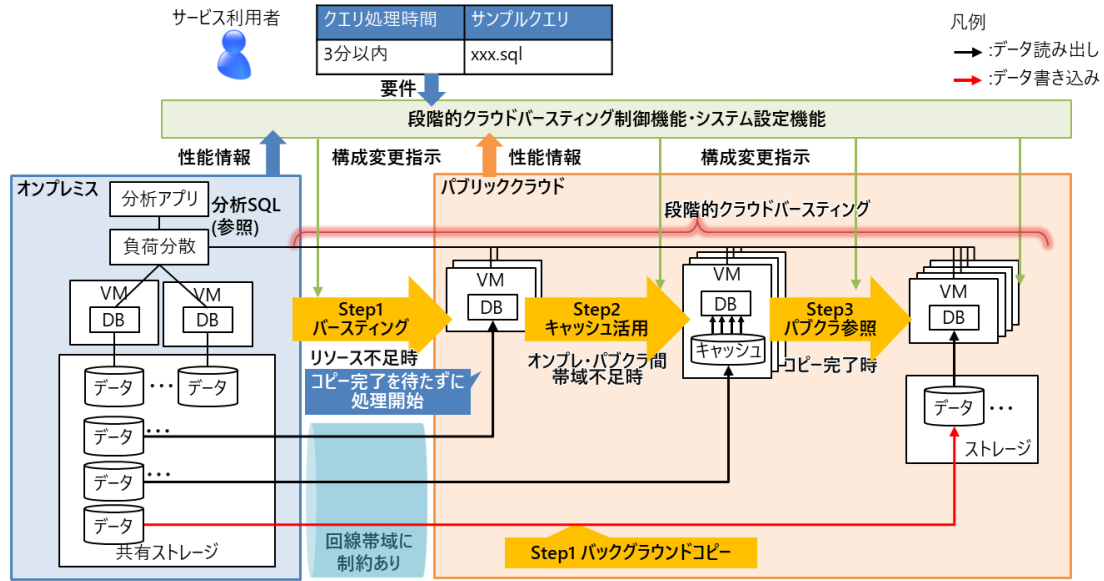


図 4 段階的クラウドバースティングの概要とシステム構成  
 Figure 4 Overview and System Configuration of Step-by-Step Cloud Bursting

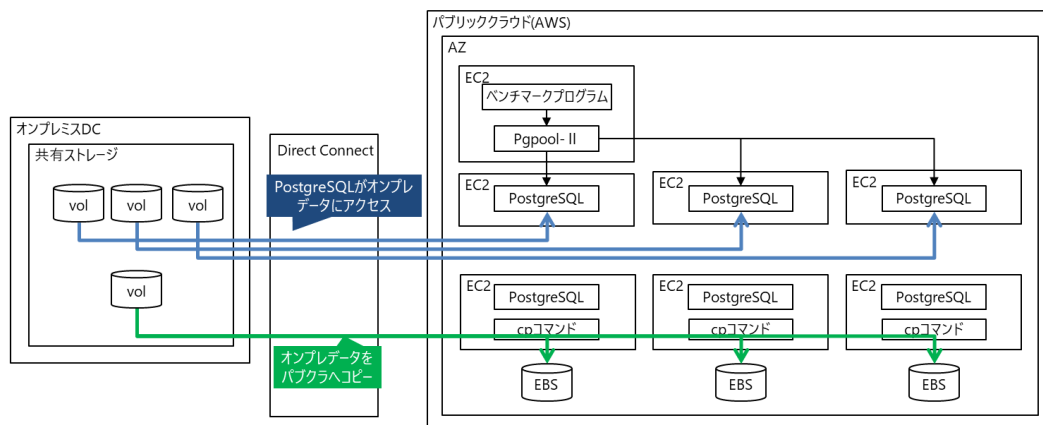


図 5 段階的クラウドバースティング実現性検証のためのシステム構成  
 Figure 5 System configuration for feasibility study of Step-by-Step Cloud Bursting

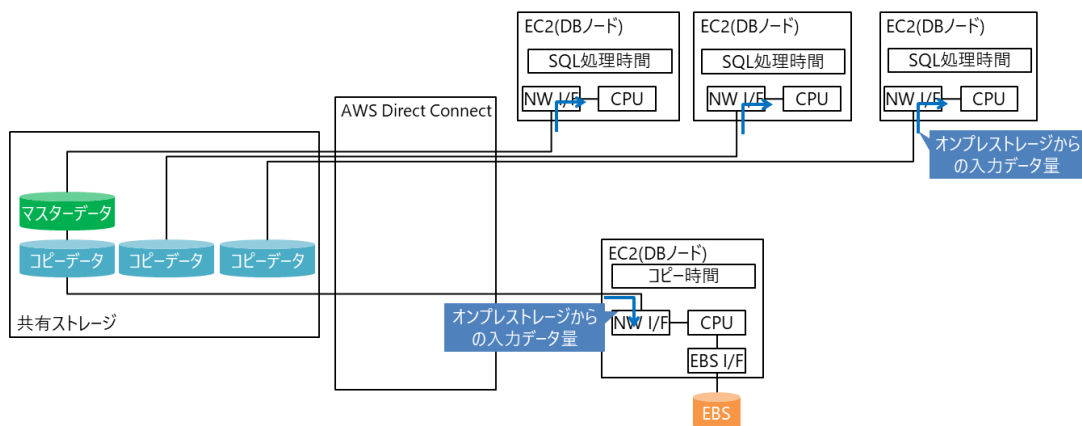


図 6 オンプレミスとパブリッククラウド間の通信性能情報取得項目  
 Figure 6 Performance data to be collected to evaluate on-premises and public cloud transfer performance

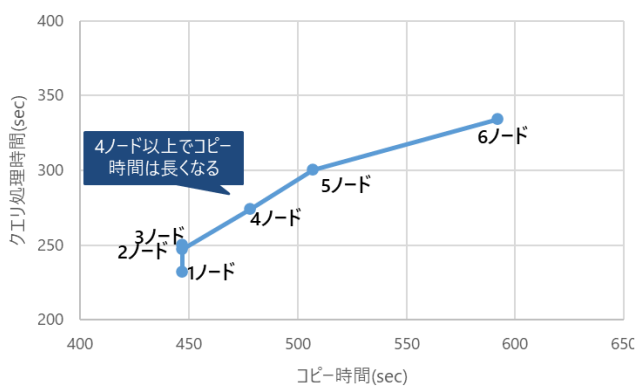


図 7-1 コピー時間とクエリ処理時間  
Figure 7-1 Copy time and query processing time

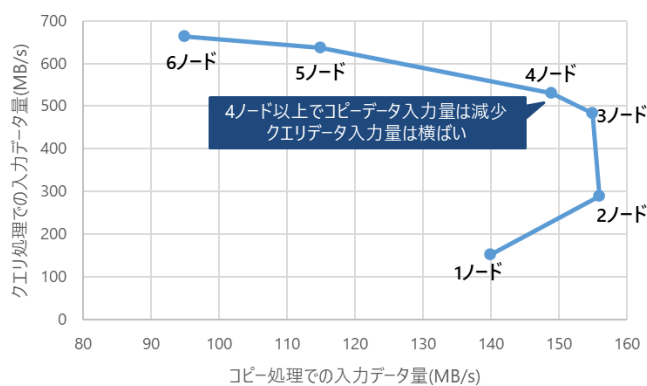


図 7-2 コピー処理とクエリ処理の入力データ量  
Figure 7-2 Input data for copy and query processing

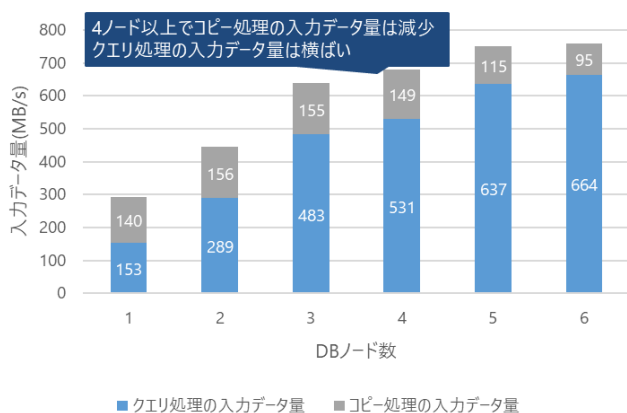


図 8-1 入力データ量の内訳  
Figure 8-1 Breakdown of input data volume

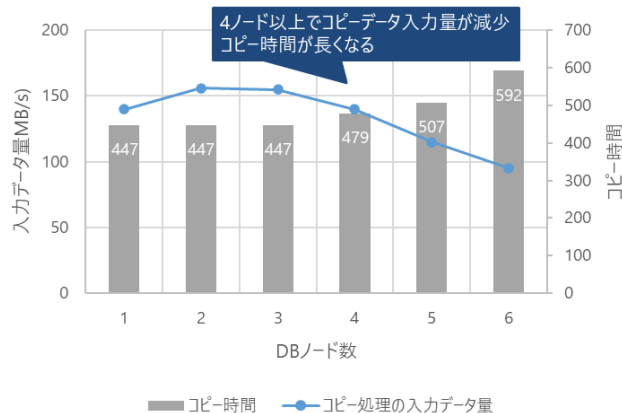


図 8-2 コピー時間とコピー処理の入力データ量  
Figure 8-2 Copy time and input data volume for copy

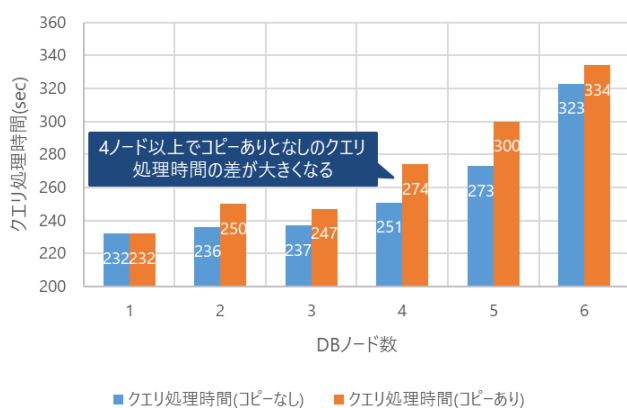


図 9-1 コピーによるクエリ処理時間への影響  
Figure 9-1 Impact of copying on query processing time

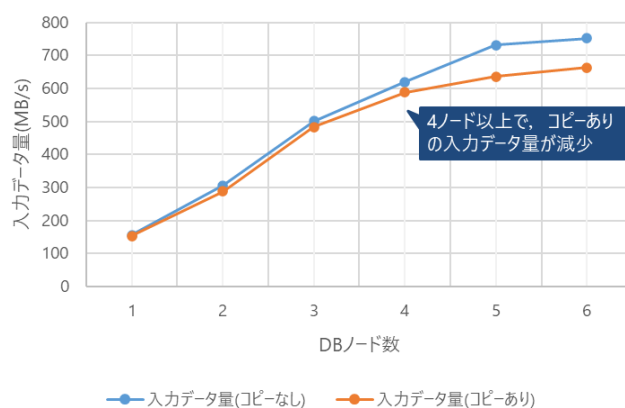


図 9-2 コピーによる入力データ量への影響  
Figure 9-2 Impact of copying on data entry volume

## 5. 評価結果

図 7-1 および図 7-2 は、コピーとクエリ処理の関係性を示している。図 7-1 は処理時間、図 7-2 は入力データ量に対するコピーとクエリ処理の関係である。図 7-1 のコピー時間とクエリ処理時間の関係より、DB が 4 ノード以上でコピー時

間とクエリ処理時間は長くなる。図 7-2 のコピー処理とクエリ処理の入力データ量の関係より、DB が 4 ノード以上でコピー処理の入力データ量は減少し、クエリ処理の入力データ量は横ばいになる。これより、DB が 4 ノード以上でコピー処理およびクエリ処理の性能が低下することを確認できた。

図 8-1 は、図 7-2 のオンプレミスからパブリッククラウド

への入力データ量の内訳である。図 8-1 より、クエリ処理とバックグラウンドコピーを同時に実行するとき、DB が 4 ノード以上で、コピー処理の入力データ量は減少し、クエリ処理の入力データ量は横ばいになる。図 8-2 は、コピー時間とコピー処理の入力データ量の関係である。図 8-2 より、DB が 4 ノード以上で、コピー処理の入力データ量は減少し、コピー時間が長くなる。これより、DB が 4 ノード以上でコピー時間が長くなると共に、クエリ処理の入力データ量が横ばいであるため、DB を追加してもクエリ処理性能が上がらないことを確認できた。

図 9 は、バックグラウンドコピーがクエリ処理性能に与える影響を示す、コピー処理ありとなしでの性能比較結果である。図 9-1 はクエリ処理時間を比較したもので、DB が 4 ノード以上でクエリ処理時間の差が大きくなる。図 9-2 はクエリ処理の入力データ量を比較したもので、DB が 4 ノード以上でコピーありの入力データ量が低くなる。これより、DB が 4 ノード以上で、コピー処理がクエリ処理に与える影響が大きくなることを確認できた。

以上より、帯域に上限があるオンプレミス・パブリッククラウド間で、クエリ処理のためのデータアクセスとバックグラウンドコピーを併用する場合、DB ノード数が増えるとクエリ処理およびバックグラウンドコピーの入力データ量が減少することから、入力データ量が低下しない DB ノード数の上限(本検証では 4)を設定することで、クラウドでのクエリ処理時間の長期化を防止できることを確認した。

バックグラウンドコピー完了前に DB アクセス増でクエリ処理時間が長くなる場合は、バックグラウンドコピー性能低下を防止するために、Step2 のキャッシュ活用を検討すべきである。

## 6. おわりに

本研究では、段階的クラウドバースティング方式を提案し、実現性を評価した。

OLAP のリクエストが増加してオンプレミスでのクエリ処理が目標時間を超過する前に、パブリッククラウドへのデータコピーの完了を待たずにクラウドバースティングし、バックグラウンドコピーを開始する。オンプレミスとパブリッククラウド間は帯域に上限があるので、クラウドバースティングで追加する DB ノード数は、クエリ処理およびバックグラウンドコピーの入力データ量が低下しないよう上限を設定することで、クラウドでのクエリ処理時間の長期化を防止できることを確認した。

## 参考文献

- [1] “Overcome these common cloud bursting challenges”  
<https://www.techtarget.com/searchcloudcomputing/tip/Overcome-these-common-cloud-bursting-challenges>
- [2] AWS Direct Connect  
[https://docs.aws.amazon.com/ja\\_jp/directconnect/latest/UserGuide/Welcome.html](https://docs.aws.amazon.com/ja_jp/directconnect/latest/UserGuide/Welcome.html)
- [3] Pgpool-II  
<https://www.pgpool.net/docs/latest/ja/html/intro-what-is.html>
- [4] “コンピュータ最適化インスタンス”  
[https://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/compute-optimized-instances.html](https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/compute-optimized-instances.html)
- [5] Jimi Sanchez, “A Review of Star Schema Benchmark”,  
arXiv:1606.00295v1[cs.DB] 1 Jun 2016,  
<https://arxiv.org/pdf/1606.00295.pdf>