VERTICA

11.0 新機能 紹介

マイクロフォーカスエンタープライズ株式会社
2021年8月16日

MICRO FOCUS

Eon Mode

- **Elastic K-safety**
  - Quorumの制御
    - Primary Nodeの半数以上（50% + 1台）が稼働している必要がある。
      これを満たさない場合はデータベースが停止。
    - 例: 6 Nodeの場合、4台以上が稼働必須。3台となった時点で停止。
    - Primary Nodeを削減する場合、必要台数は再計算される。
      ただし、最低構成の台数以上は削減できない。
  - Shard Coverageの制御
    - Shardは自身のデータをメンテナンスするためにTuple Moverを実行するPrimary Nodeに
      Subscribeしてもらう必要がある。（Shard Coverage）
    - K-safetyの値はShardをSubscribeしている冗長Nodeの数。
      - K-safety=1: Shardは2つのSubscriberを持つ。（Primary Subscriber, Secondary Subscriber）
    - Primary Subscriberが停止した場合、Secondary SubscriberがShardメンテナンスを受け持つ。
    - Secondary Subscriberも停止した場合、メンテナンスできないShardが発生するために
      データベースが停止。

Eon Mode

- **Elastic K-safety**（続き）
  - Elastic K-safety
    - Primary SubscriberまたはSecondary Subscriberのどちらかが停止した場合、他のNodeをShardのSubscriberとして追加。
    - ただし、Quorumの制御に必要な最低稼働台数に近づいた場合は追加しない。
    - 追加条件: N ÷ 2 + K + 1
      - N: Node数, K: K-safetyの値
    - 例: 10 Nodeの場合、7台（10 ÷ 2 + 1 + 1）以上の稼働であれば追加
    - データベース停止の発生を抑制。
    - 設定パラメータ: ElasticKSafety（デフォルト 1（有効））

Eon Mode

- Subcluster Interior Connection Load Balancing
  - NodeにNetwork Addressを作成するだけで有効となる新しいConnection Load Balancing。
  - 適用されるLoad Balancing PoliciesがなくClassic Load Balancingも有効でない場合、自動的に適用される。
  - 接続したSubclusterのNode間でLoad Balancingされる。
  - 設定パラメータ: EnableInteriorLoadBalancing （デフォルト 1（有効））
- Depotサイズの変更
  - データベースをReviveする際、以前の環境よりもディスク容量が大きかったり小さかったりする場合、Verticaは以前のDepotサイズの設定を再確認する。
    - Depotサイズがパーセントで指定されている場合、Reviveする環境のディスク容量に応じてDepotサイズを変更する。
    - Depotサイズが固定値で指定されている場合、ディスク容量の80%を超えてしまう場合に限り自動調整される。

Eon Mode

- Reviveの変更
  - 以前は停止しているEon ModeデータベースのすべてのsubclusterをReviveする必要があった。
  - 今後はPrimary Subclusterのみをreviveすることが選択できるようになった。

```
$ admintools -t revive_db --communal-storage-location=s3://verticadb -d verticadb
             -x auth_params.conf --hosts node01,node02,node03
```

- CLEAR_DATA_DEPOTのNodeレベルのサポート
  - CLEAR_DATA_DEPOTメタファンクションで指定できるレベルとしてデータベース, Subclusterに加えてNodeが追加された。

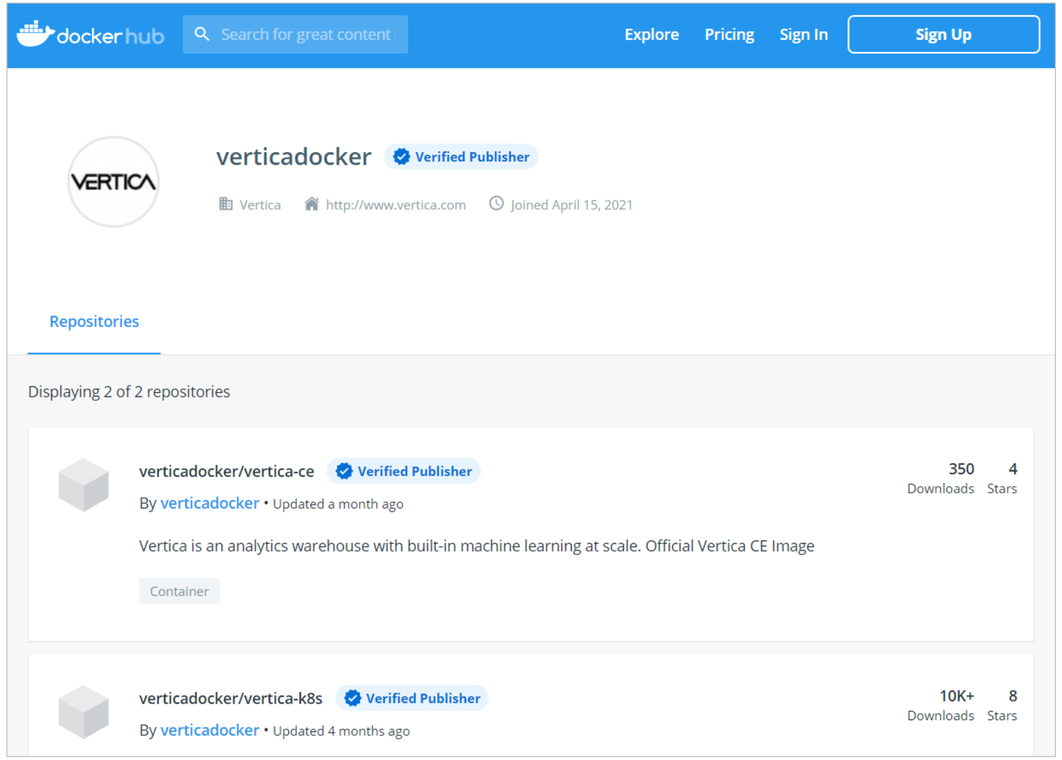
```
=> SELECT CLEAR_DATA_DEPOT('t1');
=> SELECT CLEAR_DATA_DEPOT('t1', 'subcluster_1');
=> SELECT CLEAR_DATA_DEPOT('t1', 'v_vmart_node0001');
```

Eon Mode

- Enterprise to Eon Mode Migration ToolによるAzureのサポート
  - Migration Toolを使ってAzureで稼働するEon Modeにマイグレーションできるようになった。
  - Communal StorageはAzure Blob storageを利用。

Container

- Docker ContainerとKubernetesのサポート
  - Vertica Docker Image: https://hub.docker.com/u/verticadocker
    - Vertica Kubernetes minimal image (without Tensorflow)
    - Vertica Kubernetes (with Tensorflow)
    - Vertica Community Edition
    - Vertica Kubernetes Operator
    - Webhook Admission Controller



Container

- Operator / Automation Tools for Kubernetes
  - Kubernetes上でVerticaのAutomate Lifecycle Taskを実行するHelm chartを提供。
    - Custom Resource Definition (CRD) file: Custom Resource (CR) Instanceの作成
    - Operator: CRの状態を管理するためにCRを監視
    - Admission Controller: CRの状態変更を確認
  - Operatorは次の管理者タスクを自動化
    - Verticaのインストール
    - Verticaデータベースの作成・Revive
    - Podの停止・再起動
    - Subclusterのスケーリング
    - Podの死活監視
    - 内部・外部トラフィックのロードバランシング

Voltage SecureData Integration

- Tweakパラメータ
  - VoltageSecureProtectにtweakの値を指定できるようになった。
  - ソルトに似たもので、より安全な暗号文を生成できる。
  - VoltageSecureAccessに同じtweakの値を指定しない限り復号化できなくなる。
- Maskingパラメータ
  - Voltage SecureDataでデータのマスキングが有効になっている場合、VoltageSecureAccessが返す値をマスキングすることができる。
- Format-Preserving Hash
  - VoltageSecureProtectはFormat-Preserving Encryption (FPE)に加えてFormat-Preserving Hash (FPH)を指定できるようになった。

Voltage SecureData Integration

- ファンクション

```
VoltageSecureProtect('plaintext' [, 'tweak'] USING PARAMETERS
                     format='format_name'
                     [, config_dfs_path='config_file']
                     [, identity=sd_identity]);

VoltageSecureAccess('ciphertext' [, 'tweak'] USING PARAMETERS
                     format='format_name'
                     [, mask=is_masked]
                     [, config_dfs_path='config_file']
                     [, identity=sd_identity]);
```

セキュリティ・認証

- TLS設定
  - TLS管理のためのTLS CONFIGURATIONオブジェクトが新たに追加され、ALTER TLS CONFIGURATIONで設定を行う。
  - デフォルトでserver, LDAPLink, LDAPAuthの3つのオブジェクトが作成され、削除することはできない。
  - 既存のTLS設定はアップグレード時にTLS CONFIGURATIONオブジェクトに引き継がれる。
  - admintoolsのset_ssl_paramsツールはTLS CONFIGURATIONオブジェクトを変更。
- Mutual TLS for LDAPLink, LDAPAuth
  - VerticaとLDAP serverとの接続にMutual TLSが使用できるようになった。
- 新しいLDAP Link Function: LDAP_LINK_SYNC_CANCEL
  - LDAP_LINK_SYNC_CANCELを用いて実行中のVertica・LDAP server同期処理をキャンセルさせることができる。

セキュリティ・認証

- 新しいSecurity Parameter: SystemCABundlePath
  - SystemCABundlePathを用いて、外部サービスとのTLS接続に使用するCA bundleを指定できる。
  - デフォルトのCA bundleは次の通り。
    - Red Hat系: /etc/pki/tls/certs/ca-bundle.crt
    - Debian系: /etc/ssl/certs/ca-certificates.crt
    - SUSE: /var/lib/ca-certificates/ca-bundle.pem
- 新しいSecurity Parameter: DHParams
  - DHParamsを用いて事前に作成した共通鍵交換で使用するDiffie-Hellman MODP groupを指定できる。
  - デフォルトはRFC 3526 2048-bit MODP Group 14を使用。

セキュリティ・認証

- 外部ファイルシステムのアクセスポリシー
  - デフォルトではVerticaはHDFSやクラウドのファイルシステムに接続するにはユーザが提供した認証情報を利用する。
  - 設定パラメータ: UseServerIdentityOverUserIdentity（デフォルト 0 （無効））
    ・有効にすることで、USER Storage Locationオブジェクトが必要となる。

End-to-End Machine Learning

- Time Series AnalyticsのModels
  - Time Seriesのデータセットに対する次の2つの作成とそれを使った予測。
  - 自己回帰モデル
    - AUTOREGRESSOR: 自己回帰モデルの作成
    - PREDICT_AUTOREGRESSOR: モデルをデータセットに適用し、予測作成
  - 移動平均モデル
    - MOVING_AVERAGE: 移動平均モデルの作成
    - PREDICT_MOVING_AVERAGE: モデルをデータセットに適用し、予測作成
- TensorFlow 2.xのサポート
  - TensorFlow 2.xのモデルのインポートと、それを使った予測。

End-to-End Machine Learning

- XGBoost: Column Subsampling
  - XGB_CLASSIFIERとXGB_REGRESSORに対しカラムの割合を次のパラメータに指定することで、より正確にColumn Subsamplingを行うことが出来る。
    - col_sample_by_tree: それぞれの木を構築する際にサンプリングするカラム割合を表す小数
    - col_sample_by_node: それぞれのノードを評価する際にサンプリングするカラム割合を表す小数

- PMMLに対するアップデート
  - サポートされるModelの拡張
    - RegressionModel
    - GeneralRegressionModel
  - サポートされるTag/Subtag
    - Output
    - OutputField
    - CategoricalPredictor

Stored Procedure

- PostgresのPL/pgSQLを採用
  - Phase 1としてVerticaの機能に適合する基本機能の実装が11.0で完了。
  - Phase 2として主にVerticaの機能に適合しない機能の実装を予定。
    - ・OUT, INOUTパラメータ
    - ・Cursor
    - ・トランザクション管理（Commit/Rollback。11.0ではAuto Commit。）
    - ・データ型（Complex Type, Numeric, Number, Decimal, Money, UUID）
    - ・INTO構文（SELECT INTO, EXECUTE INTO, FETCH INTO）
    - ・イベント駆動（Triggerなど）
    - ・その他（GET DIAGNOSTICSのROW_COUNT, CONTEXTオプションなど）
    - ・ドキュメント

Stored Procedure

- 構文

```
CREATE PROCEDURE [[database.]schema.]procedure([arg-list])
AS $$
  plpgsql-source
$$
LANGUAGE 'plpgsql'
[SECURITY { DEFINER | INVOKER }];
```

```
arg-list : [IN] [arg-name] arg-type
         | OUT arg-name arg-type
         | INOUT arg-name arg-type
```

```
plpgsql-source :
[ <<label>> ]
[ DECLARE
    declarations ]
BEGIN
    statements;
    ...
END [ label ];
```

Stored Procedure

- 構文（続き）

```
DROP PROCEDURE [IF EXISTS] [[database.]schema.]procedure([arg-list]);
```

```
SELECT [[database.]schema.]procedure([arg-list]);
CALL [[database.]schema.]procedure([arg-list]);
```

```
DO $$
  plpgsql-source
$$
```

Database Designer

- ZStandard Compression Encodingのサポート
  - Database Designerが次のZStandard Compression EncodingをProjectionのカラムに推奨するようになった。
    - ZSTD_COMP
    - ZSTD_FAST_COMP
    - ZSTD_HIGH_COMP

Data Export

- ORC・Delimited Formatファイルへのエクスポート
  - EXPORT TO PARQUETと同様にEXPORT TO ORCとEXPORT TO DELIMITEDによってORCファイルまたはDelimited Formatファイルへのエクスポートができるようになった。
  - エクスポート先はHDFS, S3, Google Cloud Storage (GCS), Azure Blob Storage, NFSマウントポイント。
    - ・HDFSまたはNFSマウントポイントにエクスポートする場合、Verticaは指定されたロケーションの一時ディレクトリにファイルを作成し、エクスポートが完了した時点でディレクトリ名を変更。
    - ・S3, GCS, Azure Blob Storageにエクスポートする場合、Verticaは指定されたロケーションに直接ファイルを作成。
  - S3EXPORTは非推奨機能に。

Data Export

- Linux File Systemへのエクスポート
  - VerticaはNFSでマウントされたLinux File Systemだけではなく、ローカルのLinux File Systemにエクスポートできるようになった。
  - ただし、それぞれのNodeが持つデータのみをローカルに出力する。
  - エクスポートしたファイルをロードする場合、すべてのNodeを指定する必要がある。

```
CREATE EXTERNAL TABLE sales (...)
AS COPY FROM '/data/sales/*.parquet' ON node01,
             '/data/sales/*.parquet' ON node02,
             '/data/sales/*.parquet' ON node03 PARQUET;
```

データ型

- ORC Complex Types
  - ParquetでサポートされているすべてのComplex Type（Array, Struct, ArrayとStructの混在, Map）がORCでもサポートされる。
  - ORCのStruct型をカラムに展開していた機能は削除。
- ORC・ParquetのMap型データ
  - External Tableの定義でArrayとStructの混在を用いることで、ORC・Parquet内のMap型のデータを読むことが可能となった。
  - INFER_EXTERNAL_TABLE_DDLファンクションの出力を参考にExternal Tableを作成。
- Null Struct
  - NullのStructがORC・Parquetにある場合、以前はNullのフィールドを持ったROWとして読み込んでいたが、今後はNullのROWとして扱うようになった。

データ型

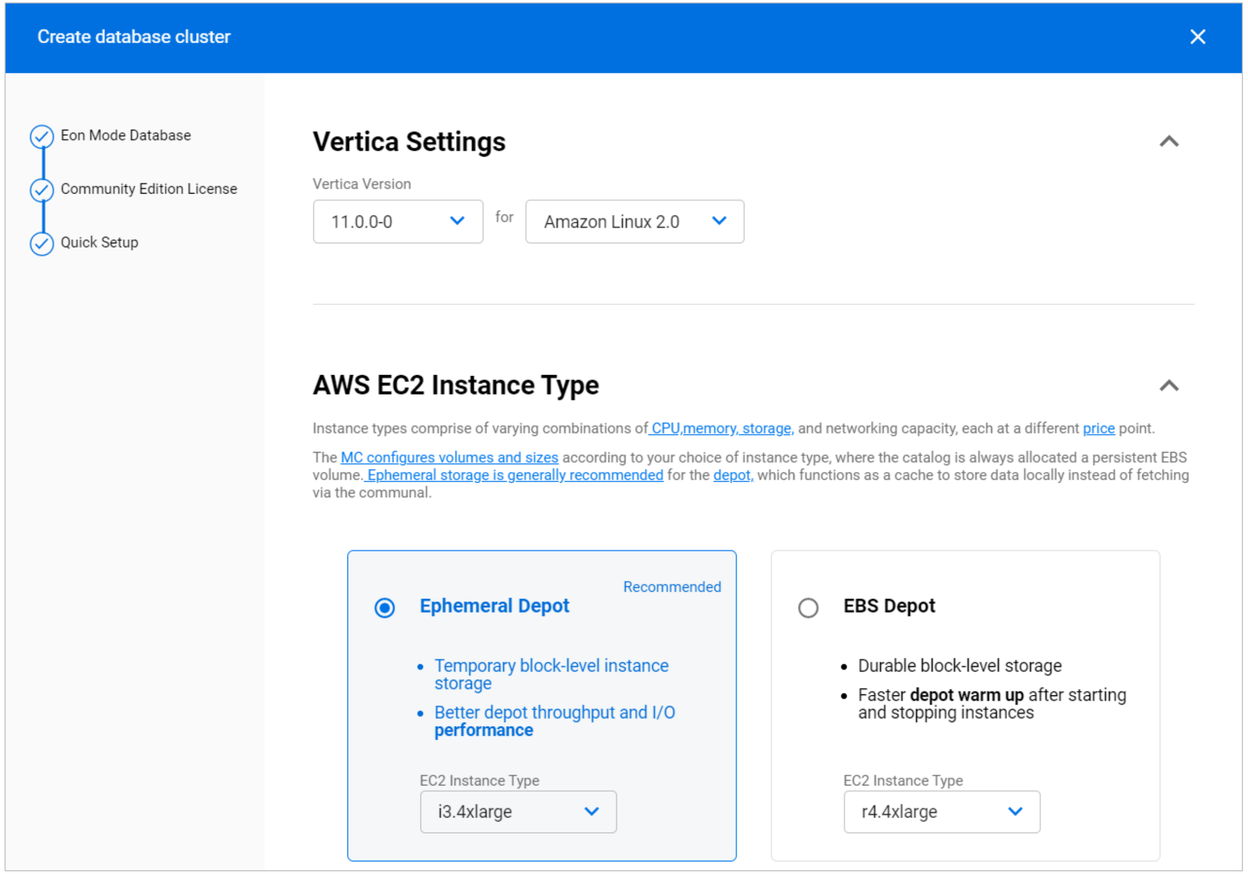
- 比較演算子
  - Array・Set型はいずれのデータ型・次元のデータも「=」「<>」「<=>」の演算子をサポート。

    1次元のデータは同じデータ型に限り「<」「<=」「>」「>=」の演算子をサポート。
  - Row型はいずれのデータ型のデータも「=」「<>」「<=>」の演算子をサポート。

    基本データ型のみを持つ場合、「<」「<=」「>」「>=」の演算子をサポート。

Management Console

- Eon Mode on Amazon Web Servicesの新しいセットアップパス
  - Eon Mode database on AWSを作成するための直感的なデザインの新しいQuick Setup wizardを提供。
  - 以前までのSetup wizardはAdvanced Setupとして利用可能。

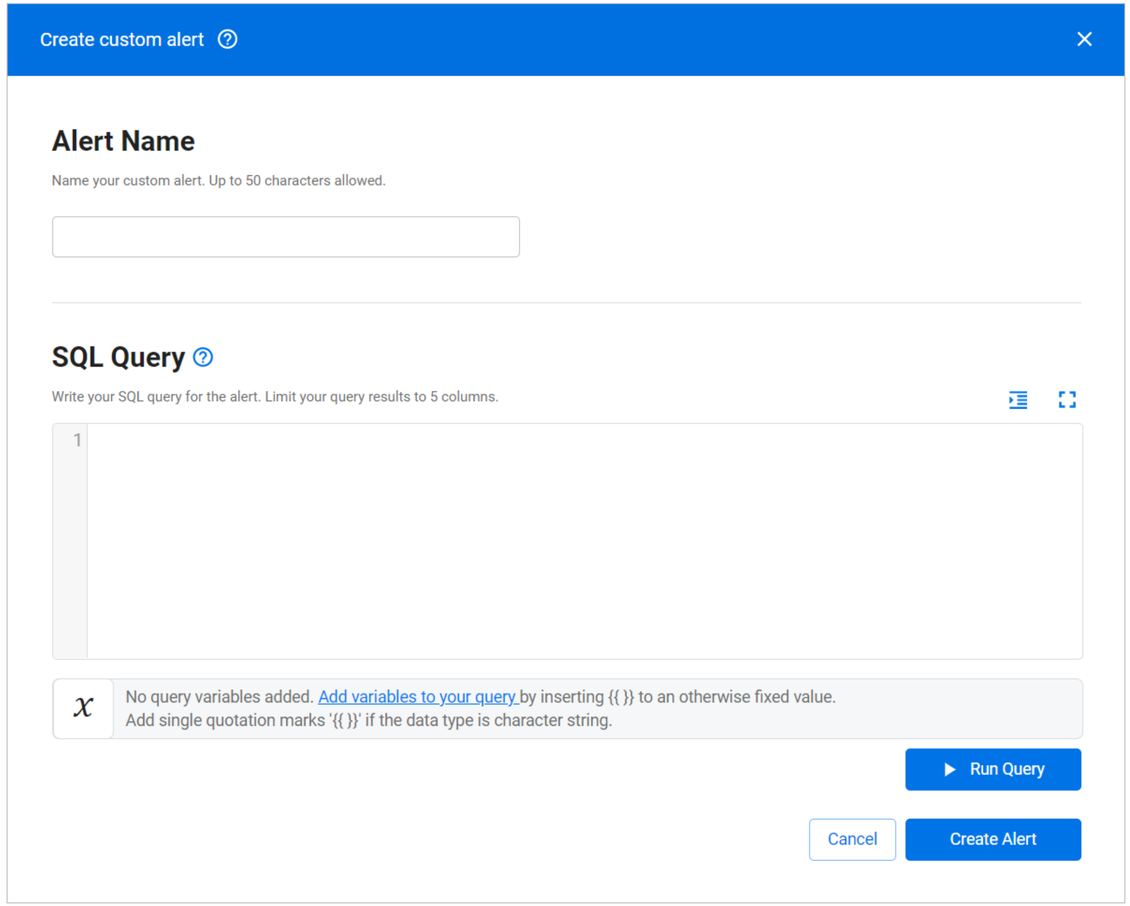


Management Console

- Microsoft Azureのサポート
  - Eon Mode database on Microsoft Azureをサポート。
  - 事前設定済みのMCを起動した後、Eon Mode databaseの作成が可能。
    データベースの起動・停止も可能。
  - MC on Azureの制約:
    - ログインユーザはSubscriptionの所有者かAzure Marketplaceからインスタンスを起動する管理者権限を持つ必要がある。
    - Subcluster Scalingはサポートされない。
    - 個々のNodeおよび仮想サーバの起動・停止はできない。
    - Subclusterの削除はできない。

Management Console

- データベース監視のカスタムアラート
  - カスタムアラートを作成し、事前定義済みのアラートで監視できないデータベースパフォーマンスの変化を監視できる。
  - カスタムアラートはユーザ定義のSQLクエリで、クエリの結果が定義された閾値を超えた際にメッセージ通知を行うことができる。



Projection

- Partition Range Projection

  Projectionの強みと欠点

  - Verticaの強みの一つにQuery Specific Projectionがある。
  - テーブルのすべてのカラムを持つSuper Projectionに加えて、ユーザは必要なカラムのみを持ち、Sort/Segmentationの定義が異なるProjectionを作成できる。
    これにより、SortやResegmentを避けてクエリの処理時間を短くすることができる。
  - 一方、実際に実行されているクエリの多くが直近に生成されたデータに対して実行されている。
  - 仮に99%以上のデータが参照されない状態でも、VerticaはQuery Specific Projectionにすべてのデータを格納する必要があり、これはストレージおよびCatalogの浪費につながり、Refreshなどのメンテナンスを難しくしている。

Projection

- Partition Range Projection
  - Partition Range ProjectionはProjectionにPartition Rangeをオプションとして指定できるようにし、Projectionに特定のPartitionのデータのみを保持させることを可能にする。

```
CREATE PROJECTION ... ORDER BY ... SEGMENTED BY ... / UNSEGMENTED ALL NODES
ON PARTITION RANGE BETWEEN min_val AND <max_val|NULL>;
```

  - Partition Range Projectionの制約:
    - テーブルはPartition Tableであること。
    - Partition Rangeに指定する式はテーブルのPartitionの指定と互換性があること。
    - min_valのPartition Keyはmax_valより小さい値または同値であること。
    - ProjectionがUnsegmentedの場合、少なくとも1つのSuper ProjectionがUnsegmentedであること。
    - Partition Rangeにはサブクエリを指定できない。
    - Live Aggregate Projection, Top-K ProjectionにはPartition Rangeは指定できない。

Tuple Mover

- Mergeoutの無効化
  - ALTER TABLE文を用いて特定のテーブルに対するMergeoutを無効化することができる。

```
ALTER TABLE public.store_orders_temp SET MERGEOUT 0;
```

  - ステージングテーブルなど一時的な目的で使用されるテーブルに対してMergeoutによるオーバーヘッドをなくすことが目的。

Query Optimization

- REFRESH_COLUMNSの最適化
  - Flattened TableのSET USINGまたはDEFAULT USINGのカラムに対して REFRESH_COLUMNSを実行することで、Target TableとSource TableをJoinする クエリが実行される。
  - デフォルトではSource Tableが常にInner TableとしてJoinが行われるが、 ほとんどの場合でSource TableのカーディナリティはTarget Tableより小さいので クエリは効率的にJoinできる。
  - ただし、Partition Tableに対してREFRESH_COLUMNSを実行する場合、Source Table のほうが大きくなることがあり、Join処理のパフォーマンスは理想的ではなくなる。
  - 設定パラメータ: RewriteQueryForLargeDim（デフォルト 0（無効））

    ・有効にすることでInner, Outerを逆にしてJoinを行うことができる。

SDK Updates

- Fenced ModeでのScalar FunctionのRange Optimizationのサポート
  - C++で開発されたUser-Defined Scalar Function (UDSF) は一連のデータを処理すべきかを事前に判断するためのgetOutputRange関数を実装できる。
  - これは依然まではUnfenced Modeだけで実装できていたが、今後はFenced Modeでも実装できる。

SQLファンクション・ステートメント

- Access Policyのエクスポート
  - EXPORT_TABLES, EXPORT_OBJECTS, EXPORT_CATALOGで出力されるSQLスクリプトにCREATE ACCESS POLICY文が追加される。
- ARRAY_CATによるComplex Typeのサポート
  - ARRAY_CATファンクションがComplex Typeの要素をサポート。
  - 両方の入力は同じデータ型の要素を持つ必要がある。
- INFER_EXTERNAL_TABLE_DDLによるORCファイルのサポート
  - INFER_EXTERNAL_TABLE_DDLメタファンクションがParquetとORCフォーマットの両方をサポート。formatパラメータに対象のフォーマットを指定。

```
INFER_EXTERNAL_TABLE_DDL('path' USING PARAMETERS
                              format=value,table_name=value[,vertica_type_for_complex_type=value])
```

SQLファンクション・ステートメント

- TO_JSONによるSET型のサポート
  - TO_JSONファンクションがSET型をサポート。
  - 以前までのようにSET型をARRAY型にキャストする必要なし。
- EXPLODEによるARRAY型・SET型のサポート
  - EXPLODEファンクションがComplex Typeを持つARRAY型, 複次元のARRAY型, SET型をサポート。
- RELOAD_ADMINTOOLS_CONFファンクション
  - RELOAD_ADMINTOOLS_CONFメタファンクションは現在のCatalog情報を基にすべての稼働Nodeのadmintools.confをアップデートする。
  - 例えば、再起動したNodeのadmintools.confを確実に他Nodeと同期させたい場合など。
  - アップデートされる内容は次の通り:
    - IPアドレスとCatalogのパス
    - データベース内のすべてのNode名

システムテーブル

- TLS_CONFIGURATIONS

  - 新しく導入されたTLS CONFIGURATIONオブジェクトの情報を格納するテーブル。

- USER_CONFIGURATION_PARAMETERS

  - ユーザレベルの設定パラメータを参照できるテーブル。

- 新しいCounterとEvent Type

  - Scalar FunctionのRange Optimizationに関する情報を収集するために、EXECUTION_ENGINE_PROFILESに次の3つの新しいCounterが追加。

    ・rows added by predicate analysis

    ・rows filtered by query predicate

    ・rows pruned by query predicates

  - QUERY_EVENTSシステムテーブルにはPREDICTS_DISCARDED_FROM_SCANのEventが追加。

システムテーブル

- PROJECTION_REFRESHESの新しいカラム
  - PROJECTION_REFRESHESにPERCENT_COMPLETEカラムが追加。

VERTICA

www.vertica.com

www.microfocus-enterprise.co.jp

MICRO FOCUS