

Tableau on SAP HANA:

パフォーマンスの追跡とワークロードの分析

目次

はじめに: 目的と前提条件	3
SAP HANA のランタイムトレースと統計の概要	4
エンドツーエンドの単一の実行トレース:	
Tableau と SAP HANA のパフォーマンス統計の組み合わせ	6
SAP HANA の SQL トレースのアクティブ化	6
Tableau のパフォーマンスレコーダーのアクティブ化	7
Tableau パフォーマンスワークブックおよび HANA SQL トレースの情報の分析	9
監視:	
長時間実行のメモリ負荷の高い SQL クエリの特定	12
Tableau の初期 SQL を通じた SAP HANA セッション情報の強化	12
Expensive Statements Trace:	
長時間実行のメモリ負荷の高いクエリの監視	15
Tableau での SAP HANA Expensive Statement Trace の結果の分析	18
SQL プランキャッシュの統計: 最適化の対象となり得る箇所の特定 (長時間および頻繁に実行)	18
Tableau での SAP HANA SQL プランキャッシュの分析	21
単一の SQL クエリの詳細分析:	
SAP HANA Explain Plan および Visualize Plan	22
SAP HANA Plan Explanation	22
SAP HANA Plan Visualizer	23
まとめ	25
Tableau について	25



はじめに: 目的と前提条件

SAP HANA ライブデータソースに関する Tableau ダッシュボードの実装を成功させるには、パフォーマンスと安定性がきわめて重要です。できる限り最高のパフォーマンスを達成するには、設計プロセスの間にダッシュボードのランタイムを分析し、その後、負荷の高いクエリについてシステムを継続的に監視することが重要です。このドキュメントは、SAP HANA と Tableau のパフォーマンストレースおよび監視機能の概要を示し、パフォーマンスのボトルネックを特定することを目的としています。ボトルネックを理解することは、適切なパフォーマンス最適化対策を特定するうえでの前提条件となります。効率的な Tableau ダッシュボードの設計方法に関する推奨事項については、このドキュメントでは扱いません。それらについては別のリソースをご参照ください。例: [効率的に作業できるワークブックの設計](#) (ホワイトペーパー)、[ダッシュボードのパフォーマンスに関するベストプラクティス](#) (プレゼンテーションの録画) など。

本書で取り上げる SAP HANA トレースをアクティブ化するためには、十分な権限または **SAP HANA 管理者** の支援が必要です。さらに、Tableau が測定したフロントエンドのランタイムデータを収集するために、**Tableau Creator** (アナリストまたはダッシュボード設計者など) が必要になります。Tableau と SAP HANA のトレースを組み合わせることで、完全なエンドツーエンドのランタイムディストリビューションを把握できます。これらのトレースは、パフォーマンスのボトルネックを特定するのに必要な情報を提供します。そのボトルネックは、パフォーマンスを最適化できる可能性の高い場所となります。

詳細なエンドツーエンドのトレースは、1 回の実行に対してのみ行うことをお勧めしますが (リソース消費のオーバーヘッドのため)、パフォーマンスとワークロードを継続的に監視して負荷の高いクエリを特定するために実行できる他のトレースがあります。高負荷の状況でボトルネックを回避するには、負荷が高く (CPU ランタイムおよびメモリ消費)、頻繁に実行される (実行数によって乗算される) クエリのソースを分析し、特定することが重要です。SAP HANA はこれらの統計を集計するように構成できます。そして、Tableau をそれらに接続し、分析することができます。

実行時間が長い、またはメモリを大量に消費する SQL ステートメントを特定した場合、SAP HANA Plan Explanation または SAP HANA Plan Visualizer などのツールを使用して、HANA 内でのその処理をさらに分析できます。これらのツールは、クエリ結果の計算方法と、どの処理ステップが負荷が高いのかについての、詳細なステップバイステップの情報を提供します。それを理解することで、フィルター の早期適用、異なるレベルでの集約など、最適化戦略を特定できるようになります。

SAP HANA のランタイムトレースと統計の概要

Expensive Statements Trace

Expensive Statements Trace は、設定された実行時間を超過した SQL ステートメントの情報をキャプチャします。そのしきい値はマイクロ秒 (μs 、100 万分の 1 秒) で入力します。

このトレースはさまざまな方法で使用できます。しきい値が 1 に設定されている場合は実質的にすべてのクエリを記録できますが、この設定は、専用パフォーマンス分析が必要な場合に限定的な時間でのみ使用する必要があります。一方、しきい値をたとえば 5 秒 (5,000,000 μs) などの高い値に設定した場合は、永続的にアクティブ化しておくことが可能です。SAP Note 2180165 ([FAQ: SAP HANA Expensive Statements Trace](#) (英語)) には、「(しきい値が妥当な場合は) 付加価値が大きく、オーバーヘッドが小さいため、そのトレースを永続ベースでアクティブ化しておくことが推奨される」と記載されています。

このトレースのその他の利点には、SAP HANA ビュー (スキーマ「SYS」の「M_EXPENSIVE_STATEMENTS」) から選択して結果を取得できるため、レコードを柔軟に分析できることがあります。つまり、Tableau の強力な分析機能を使用して、SAP HANA に記録された Expensive Statement を分析できるということです。

前述したような利点があるため (永続的にアクティブ化できる、SQL を使用してクエリできる)、Expensive Statements Trace については次章でさらに詳しく解説します。

SQL トレース

パフォーマンスの監視を目的として Expensive Statements Trace を永続的にアクティブ化しておけるとともに、SAP HANA は専用パフォーマンス分析を目的としたトレースも提供し、アクティブ化されている間すべての SQL クエリとそのランタイム統計をキャプチャできます。これは通常、専用ユーザーを対象として短時間のみ実行します。SQL トレースは詳細度が高く、より多くのリソース (ストレージと CPU) を消費するため、短時間の明示的なパフォーマンス分析用にものみアクティブ化することが推奨されます (長期間の監視には不適)。

SQL トレースは、収集した情報を、データベーステーブルではなく、.py テキストファイルに書き込みます。このファイルタイプには、トレースされたデータベース操作を再生できるという利点がありますが、トレースされたパフォーマンス情報を読み取って分析することは困難です。この問題に対処するために、SAP は Python ツールである SAP HANA SQL Trace Analyzer を提供しています。これはファイルからの情報を集約し、その分析を簡素化します。

トレースの結果例や SQL Trace Analyzer ツールの動作については、SAP HANA Academy による「[SAP HANA SQL Trace Analyzer の使用方法](#)」(英語) のデモビデオをご覧ください。

SQL プランキャッシュ

SQL プランキャッシュは、SAP HANA データベースの SQL プロセスの理解に役立つ重要なツールです。SQL プランキャッシュは、トレースではなく、デフォルトでアクティブ化されるキャッシュであり、貴重な統計を収集します。意図的にオンに切り替える必要はありません。

SQL プランキャッシュは、システム内で頻繁に実行されるステートメントの概要を提供し、ランタイム統計を追跡します。その統計は集計されるため、特定のダッシュボード実行のランタイム情報は明らかになりませんが、最適化の対象になり得る箇所を特定するために使用できます。たとえば、最も頻繁に実行されるクエリ、最も長く実行されているクエリなどです。SQL プランキャッシュは SAP HANA ビューとしてクエリできるため、Tableau を使用してそれを分析することができます。

SAP HANA メモリトラッキングの有効化

SAP HANA トレースの一部は、パフォーマンス KPI のほかにも、SQL ステートメントのメモリ消費をキャプチャできます。これは、メモリ不足の状況を分析する場合や、データベースのメモリを大量に消費しているユーザーを特定する場合に、重要な情報となることがあります。メモリ使用の追跡を行う場合は、global.ini ファイルの「resource_tracking」セクションにある次の 2 つのパラメーターをオンにする必要があります。

- enable_tracking
- memory_tracking

Overview	Landscape	Alerts	Performance	Volumes	Configuration	System Information	Diagnosis Files	Trace Configuration	
Filter: ✕									
Name ^						Default	System	Host - bw4hana	
▼ [1] resource_tracking									
enable_tracking						off	● on		
feature_usage_monitor_last_details						deprecated			
host_job_history_granularity						500			
load_monitor_granularity						10000			
load_monitor_max_samples						100000			
memory_tracking						off	● on		

エンドツーエンドの単一の実行トレース:

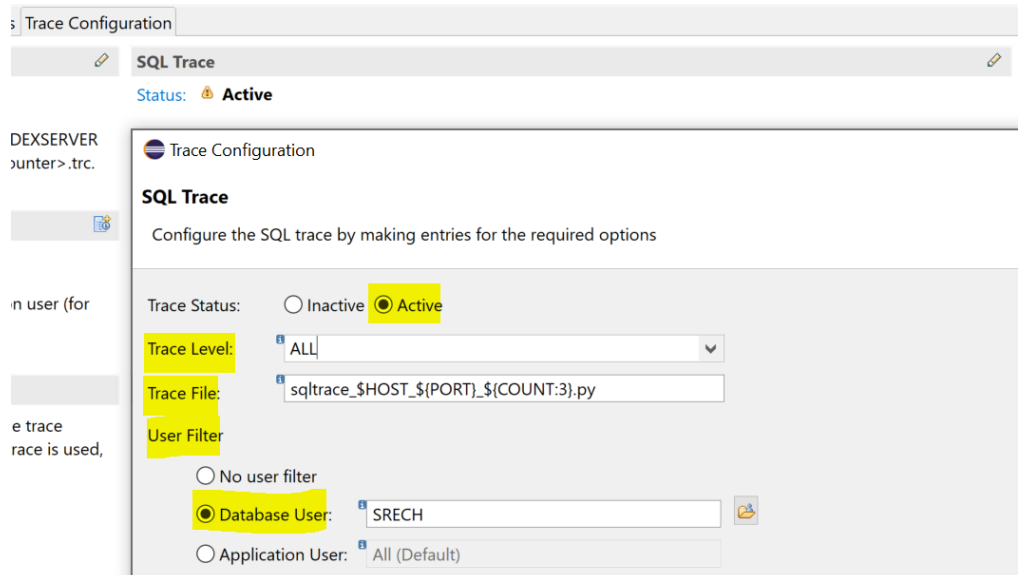
Tableau と SAP HANA のパフォーマンス統計の組み合わせ

Tableau ワークブックのランタイムディストリビューションを把握するためには、Tableau とデータベース側の両方でパフォーマンスをエンドツーエンドで追跡することが推奨されます。そうすることで、ランタイムの大部分が費やされているレイヤー (Tableau、ネットワーク、HANA) と、実行全体の速度を低下させているステップについて結論を導き出すことができます。これは、パフォーマンスを最適化できる可能性が最も高いステップを特定するのに役立ちます。

SAP HANA の SQL トレースのアクティブ化

最初のステップとして、SAP HANA の SQL トレースを、トレースを実行しているデータベースユーザーに対してアクティブ化する必要があります。これにより、ランタイム情報を含む、そのユーザーからのすべての SQL クエリをキャプチャできます。このトレースをアクティブ化する方法の 1 つは、SAP HANA Studio の [Administration] > [Trace Configuration] セクションでそれを設定することです。

トレースを有効にするには、トレースステータスを「Active」に設定し、トレースレベルを「ALL」または「ALL_WITH_RESULTS」にします。トレースファイルのフィールドには、生成されるトレースファイルの名前を指定できます。また、データベースユーザーは、Tableau ワークブックを実行しているユーザーにフィルターします。



エンドツーエンドの分析には、Tableau と SAP HANA の両方でトレースを一時的にアクティブ化することが必要になります。継続的な監視ではなく、単一の実行に対してのみトレースを行うことが推奨されます。

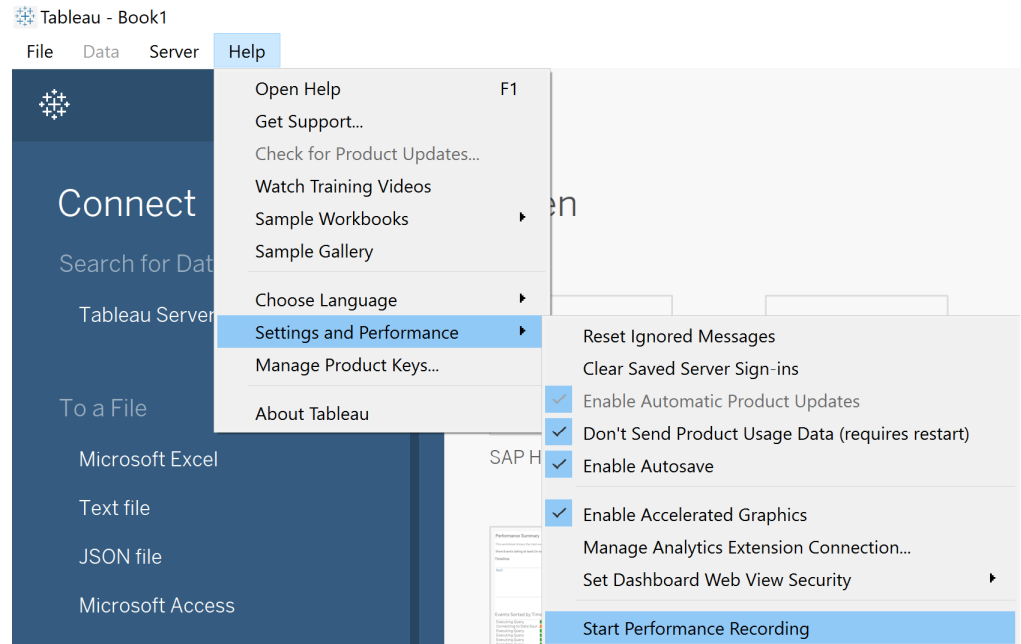
SQL トレースの構成に関する詳細説明については、SAP Note 2031647 (SAP HANA Studio で SQL トレースを有効化する方法 (英語)) を参照してください。

Tableau のパフォーマンスレコーダーのアクティブ化

Tableau 側でランタイムディストリビューションを記録するには、Tableau のパフォーマンスストレスを使用できます。

Tableau Desktop

Tableau Desktop を使用する場合は、次のようにクリックすることでパフォーマンスストレスが開始されます。[ヘルプ] > [設定とパフォーマンス] > [パフォーマンスの記録を開始]



これで、Tableau で実行されるすべてのステップが、パフォーマンスストレスの一環として記録されます。

記録を停止し、記録セッションの結果を含む一時的なワークブックを表示する場合は、同じボタンを再度クリックします ([パフォーマンスの記録の停止] となっています)。

[ヘルプ] > [設定とパフォーマンス] > [パフォーマンスの記録の停止]

その結果、トレースした情報を含む Tableau ワークブックが生成されます。[ファイル] > [名前を付けて保存] を使用することで、そのワークブックを保存し、後から参照できます。

Tableau Desktop でのトレースのアクティブ化については、Tableau ヘルプの「[ワークブックパフォーマンスの記録と分析](#)」(英語)をご覧ください。

Tableau Server

Tableau Server を使用する場合は、ビューの URL の最後に「:record_performance=yes&」を追加するとパフォーマンスをトレースできます。

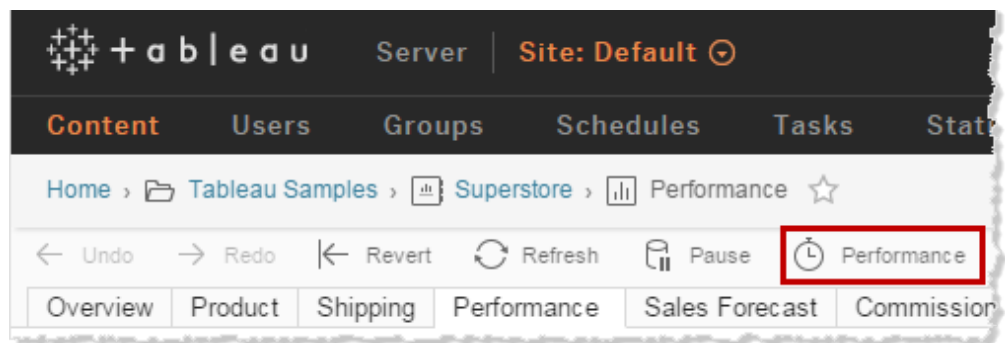
例: ビューの URL

```
http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:iid=1
```

例: ビューの URL + アクティブ化されたパフォーマンストレース:

```
http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:record_performance=yes&:iid=1
```

Tableau Server からパフォーマンス記録を見るには、[パフォーマンス] ボタンをクリックします。



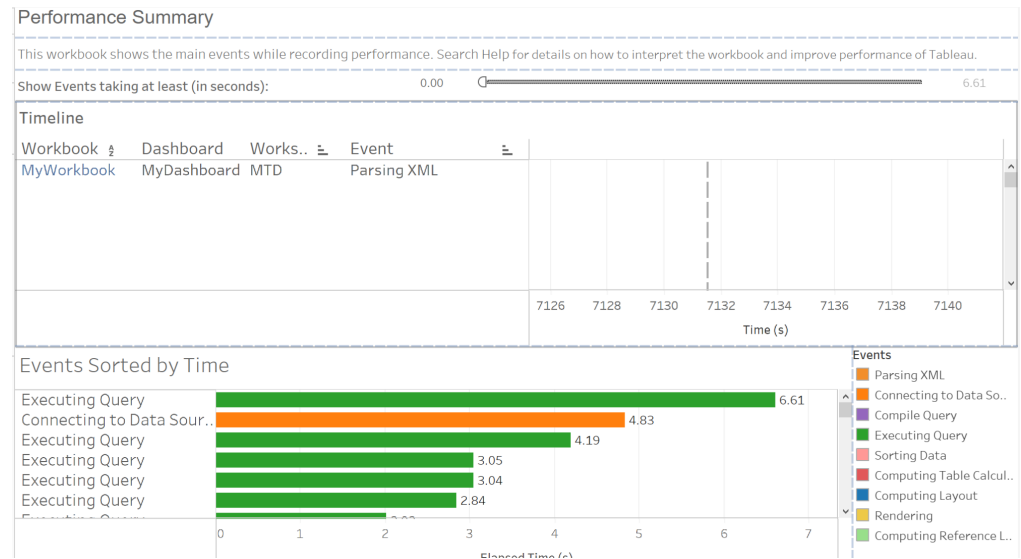
パフォーマンストレースを停止するには、別のページをクリックするか、URL から「:record_performance=yes」を削除します。

Tableau Server でのトレースのアクティブ化については、Tableau ヘルプの「[パフォーマンスの記録の作成](#)」をご覧ください。

Tableau パフォーマンスワークブックおよび HANA SQL トレースからの情報の分析

Tableau パフォーマンストレースの結果は、Tableau ワークブックの形式で提示されます。

[パフォーマンスサマリー] ダッシュボード に、実行されたステップのタイムラインの概要が表示され、下部のチャートではランタイム別にイベントがソートされます。

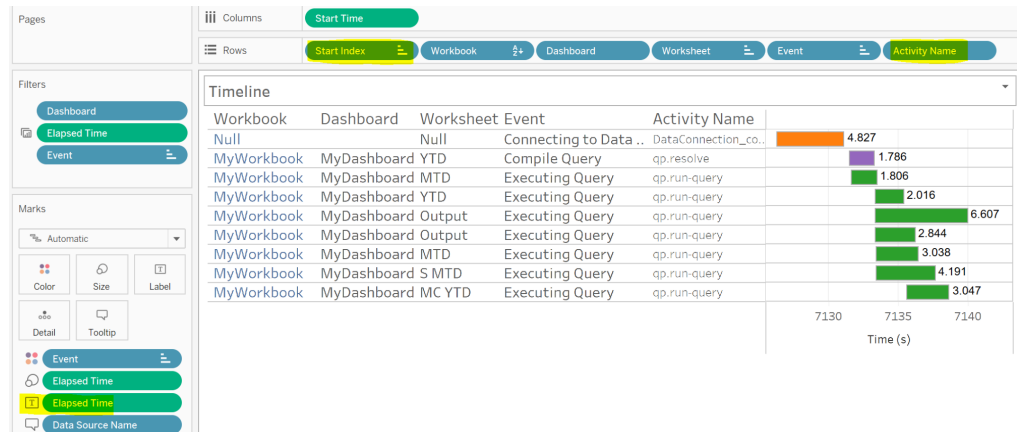


最も重要なイベントにフォーカスするために、0.5 秒以上のイベントをフィルタリングしてノイズを除去することをお勧めします。

Show Events taking at least (in seconds): 0.50

[タイムライン] シートを調節して、追加の情報を表示することができます。たとえば、「経過時間」をラベルとして追加したり、「開始インデックス」をドラッグして行の最初のフィールドにしたり (アクションが開始時間でソートされる)、行に「アクティビティ名」を追加したりできます。

その結果は以下のようになります。



このケースでは、ランタイムの大部分はクエリの実行に費やされています。「クエリの実行」は、Tableau がクエリの応答を SAP HANA から待っている時間です。これには、ネットワークを通じて応答が転送される時間が含まれています。

「クエリの実行」イベントに対応する完全な SQL ステートメントを特定するには、そのイベントを [パフォーマンスサマリー] で選択します。そうすることで、そのクエリのシートがフィルターされ、そのイベントからの SQL コマンドが表示されます。

Tableau ワークブックシートでは非常に小さいスペースにステートメントが表示されるため、それを選択してテキストエディターにコピーすることで全体を見ることができます。

```

SELECT ""BI.PLANT/PLANT_CV_PLNTMNG"".""BI_PLANT__T"" AS ""BI_PLANT__T""
FROM ""_SYS_BIC"".""BI.PLANT/PLANT_CV_PLNTMNG"" ""BI.PLANT/PLANT_CV_PLNTMNG""
GROUP BY ""BI.PLANT/PLANT_CV_PLNTMNG"".""BI_PLANT__T""
ORDER BY 1 ASC

```

ランタイムの大部分を消費するワークシートとクエリを特定した後、その情報を HANA SQL トレースと組み合わせることができます。

トレースファイルに対して SAP の SQL Trace Analyzer (SAP Note 2412519 [FAQ: SAP HANA SQL Trace Analyzer](#) (英語) を参照) を実行すると、その出力には HANA での各 SQL クエリのランタイム情報が表示されます。その情報には、実行、コンパイル、カーソル、フェッチのランタイムが含まれています。この情報はマイクロ秒単位で表示されるため、1,000,000 で除算すると秒に変換されます。

	A	B	C	D	H	L
1	STATEMENT_STRING	STATEMENT_HASH	SCHEMA	COUNT	TOTAL_EXECUTE_DURATION	TOTAL_COMPILE_DURATION
2	SELECT "t2"."X_measure_4" AS	270566e01b431ee3c8	SRECH	1	18900599	241
3	SELECT "t2"."X_measure_6" AS	2509cbf5a7db02d7a8	SRECH	1	18740081	432
4	SELECT "t2"."X_measure_4" AS	2925ec907324f130a9	SRECH	1	15357529	171

SAP HANA の SQL ステートメントランタイムと、これに対応する Tableau の「クエリの実行時間」ランタイムの比較によって、重要なインサイトを発見できます。大きな差がある場合は、HANA サーバーと Tableau フロントエンド間のネットワーク速度またはファイアウォールが関係している可能性があります。

HANA の SQL クエリのランタイムが大きい場合、それに対処するには複数の方法があります。たとえば次のような方法です。

- Tableau ワークシートを再設計することでクエリをシンプル化できないか確認する
- そのクエリ向けに HANA データモデルを最適化できないかチェックする (本書の「HANA Plan Explanation」および「HANA Plan Visualization」セクションを参照)。

SAP HANA の SQL 実行プランのコンパイル時間が長い場合や、クエリ実行時間よりも長い場合は、Tableau の [バインド変数機能](#) をアクティブ化することが推奨されます。この機能は SQL プラン キャッシュのヒット率を高めることができ、その結果、SQL 実行プランのコンパイルの必要性が低減されます。

「HAVING (COUNT(1) > 0)」で終わる、長時間実行されている SQL クエリがある場合は、TDC 設定 (「CAP_QUERY_HAVING_REQUIRES_GROUP_BY」) を有効にすることで、この条件を「GROUP BY」に置き換えることができます。この設定により、特定のユースケースでは SQL クエリのランタイムを格段に削減することができます。

これらは、Tableau トレースと SAP HANA トレースを分析することによって得られる調査結果の例となります。

監視:

長時間実行のメモリ負荷の高い SQL クエリの特定

SAP HANA は、パフォーマンス分析や監視に使用できる貴重な情報が含まれたさまざまなトレース、ログ、統計を提供します。たとえば、高負荷のリソースによってボトルネックが発生した場合に、それらを使用することで、パフォーマンスを最適化できる可能性のある箇所を特定することや、根本原因分析を行うことができます。

残念ながら、それらのログのどの情報が Tableau のクエリに関連しているか、さらにはどの特定のワークブックまたはシートに関連しているかを特定することは難しい場合があります。

Tableau クエリに関するそのような情報を追跡するために、Tableau の初期 SQL 機能が非常に役に立つ場合があります。この機能により、SAP HANA セッション変数を使用することで、SAP HANA での Tableau ワークロードおよびパフォーマンスを分析できる可能性が広がります。これは、Expensive Statements Trace や SQL プランキャッシュなど、SAP HANA ログからの情報とともに、SAP HANA の管理者および Tableau ワークブック設計者に新たな可能性をもたらします。

Tableau の初期 SQL を通じた SAP HANA セッション情報の強化

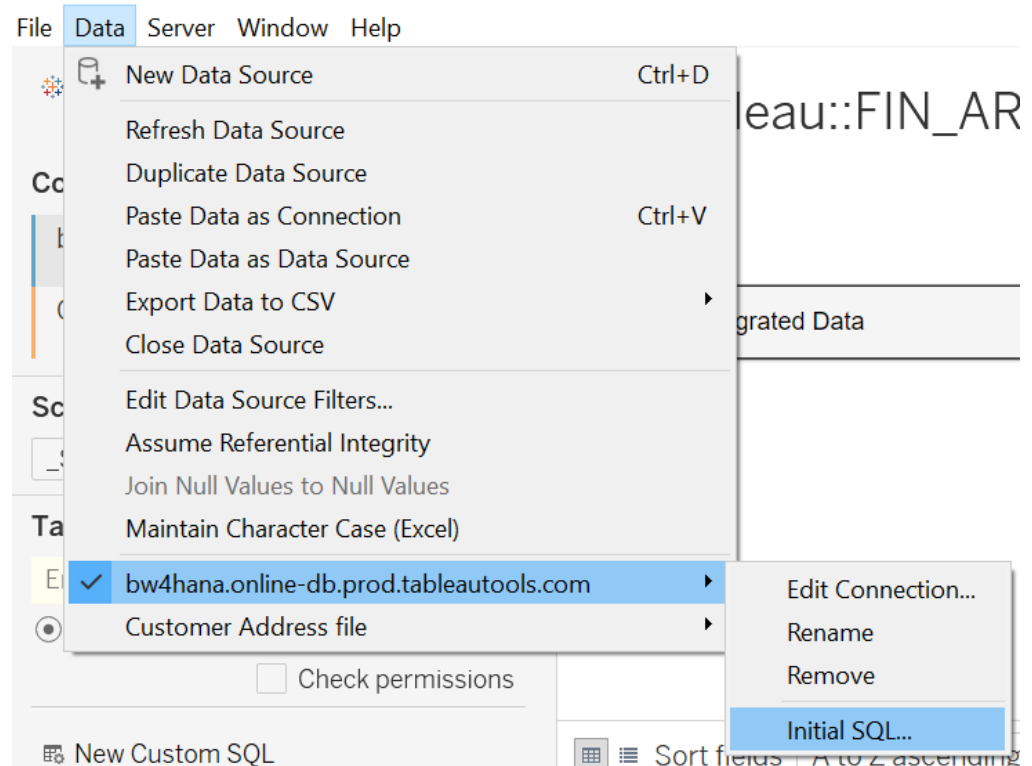
SAP HANA への新たな接続が作成されると、SAP HANA Session Management に新しいセッションが確立されます。そこでセッションのソースに関する情報と技術情報が保持され、それらはトレースおよび統計データのために SAP HANA が使用します。

事前定義されたセッション変数の例には、APPLICATION、APPLICATIONVERSION、APPLICATIONUSER、APPLICATIONSOURCE などがあります。事前定義されたセッション変数の全一覧とその使用については、「[SAP HANA SQL およびシステムビューのリファレンス](#)」(英語)をご覧ください。

Tableau が SAP HANA に接続すると、いくつかのセッション変数 (例: APPLICATION および APPLICATIONVERSION) が自動的に Tableau によって割り当てられます。この機能は、Tableau のクエリと、その他のアプリケーション/ツールからのワークロードを分離するのに役立ちます。Tableau の初期 SQL 機能を使用すると、Tableau ワークブック名や Tableau Server ユーザーなどの情報を格納するために追加のセッション変数を割り当てることができます。初期 SQL は、データソースレベルで定義された SQL コマンドのセットで、ワークブックが開かれた場合やデータが更新された場合など、データベース接続が行われたときに実行されます。

以下の例では、APPLICATIONSOURCE を使用して、接続を確立した Tableau ワークブックの名前を送信し、APPLICATIONUSER を使用して、クエリを実行している Tableau ユーザーを特定します。

初期 SQL を設定するには、[データソース] タブを開き、[データ] メニューで SAP HANA データソースを選択する必要があります。



セッション変数を設定または上書きする構文は次のとおりです。

```
SET [SESSION] <variable_string_literal> = <value_string_literal>
```

例として、初期 SQL で次のコマンドを使用して、SAP HANA セッション変数に Tableau アプリケーション、バージョン、ワークブック名を格納できます。

```
SET SESSION 'APPLICATIONSOURCE' = [WorkbookName];
```

```
SET SESSION 'APPLICATIONUSER' = [TableauServerUser];
```

想定している情報を初期 SQL が提供しているかどうかをチェックするために、M_SESSION_CONTEXT システムビューをクエリしてセッション変数を取得することができます (例: 'select * from M_SESSION_CONTEXT')。

```
select * from M_SESSION_CONTEXT
```

HOST	PORT	CONNECTION_ID	KEY	VALUE
bw4hana	30,203	326,188	APPLICATION	Tableau Desktop
bw4hana	30,203	326,188	XS_APPLICATIONUSER	SRECH
bw4hana	30,203	326,188	PROTOCOL_VERSION	4.1 (1, 1)
bw4hana	30,203	326,188	APPLICATIONVERSION	2020.2
bw4hana	30,203	326,188	APPLICATIONSOURCE	Initial SQL - Accounts Receivable - Live

結果として、Session Monitoring や Expensive Statements Trace などの SAP HANA 管理ツールにより、これまでは表示できなかったセッションのソースに関する情報を表示できます。

a) セッション監視:

Overview | Landscape | Alerts | Performance | Volumes | Configuration | System Information | Diagnosis Files | Trace Configuration

Threads | Sessions | Blocked Transactions | SQL Plan Cache | Expensive Statements Trace | Job Progress | Load

Summary

Enter your filter | Visible rows: 13/13

s	Application	Application Source	Application Version	Application User	Database User	Client Host	Client IP	Client Process ID
0	Tableau Desktop	Initial SQL - Accounts Receivable - Live	2020.2	?	SRECH	SRECH-LAP	192.195.4...	9,796
0	Tableau Desktop	Initial SQL - Accounts Receivable - Live	2020.2	?	SRECH	SRECH-LAP	192.195.4...	25,192
0	Tableau Desktop	Initial SQL - Accounts Receivable - Live	2020.2	?	SRECH	SRECH-LAP	192.195.4...	14,052

b) ログとトレース

SQL | Result

```
select * from M_SQL_PLAN_CACHE where USER_NAME = 'SRECH' ORDER BY LAST_EXECUTION_TIMESTAMP DESC
```

	HOST	STATEMENT_STRING	USER_NAME	APPLICATION_NAME	APPLICATION_SOURCE	ACCESSED_TABLE_NAMES
1	bw4hana	SELECT "tableau_FIN_AR_DFIAR30_CV"."DEBIT_AMOUNT_I...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...
2	bw4hana	SELECT "tableau_FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_I...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...
3	bw4hana	SELECT TO_DATE("tableau_FIN_AR_DFIAR30_CV"."CLEARIN...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...
4	bw4hana	SELECT ADD_DAYS(CAST(TO_DATE("tableau_FIN_AR_DFIAR...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...
5	bw4hana	SELECT "tableau_FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...
6	bw4hana	SELECT TOP 32 EXTRACT(MONTH FROM TO_DATE(ADD_DA...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...
7	bw4hana	SELECT ADD_DAYS(CAST(TO_DATE("tableau_FIN_AR_DFIAR...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...
8	bw4hana	SELECT "tableau_FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...
9	bw4hana	SELECT "tableau_FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB/BIO/TAC...

Expensive Statements Trace: 長時間実行のメモリ負荷の高いクエリの監視

SAP HANA は、実行時間が設定しきい値を超える SQL ステートメントをトレースできる機能を提供します。この広く利用されている機能は Expensive Statement Trace と呼ばれ、通常はパフォーマンス最適化が必要なクエリを特定するために使用されます。

SAP HANA セッション変数が Tableau の初期 SQL に割り当てられていない場合、長時間実行されている SQL ステートメントのどれが Tableau から来ているかを特定することは難しい場合があります (使用している Tableau バージョンによります)、さらには、そのクエリ元の Tableau ワークブックの特定もできません。

Tableau の初期 SQL にセッション変数を割り当てれば、非常に簡単に特定できるようになります。トレースを構成する際には、それをユーザーレベルのみではなく、アプリケーションレベルで制限することもできます。重要なのは、しきい値となる時間 (マイクロ秒: $1 \mu s = 100$ 万分の 1 秒) に注意を払うことです。長時間実行されているクエリのみをトレースする場合、この値を大きくする必要があります (既定は 1,000,000)。一方、すべてまたはほとんどの Tableau クエリをトレースする場合は、かなり小さい値にします (1,000 または 1 ms など)。

Overview	Landscape	Alerts	Performance	Volumes	Configuration	System Information	Diagnosis Files	Trace Configuration
If the database trace is configured, the traces for the trace components of the system (for example, INDEXSERVER and NAMESERVER) are written to files named <servicename>_<host>.<port_number>.<3_digit_file_counter>.trc. Some of these traces are always activated by default.				If the SQL trace is active, the database calls for the specified database, or application users are traced. The trace data is stored in files starting with sqltrace_<host>.<port_number>.<3_digit_file_counter>.py.				
User-Specific Trace Configuration: Not Specified				Performance Trace Status: Inactive				
If the user trace is configured, the traces for the trace components for a specific database or application user (for example, INDEXSERVER and NAMESERVER) are written to files named <servicename>_<host>.<port_number>.<context>.trc.				If the performance trace is running, the system performance is traced. The trace data is saved to the file specified.				
End-to-End Traces				Expensive Statements Trace Status: Inactive				
The predefined end-to-end traces are used by applications to record the steps through all the available trace components (INDEXSERVER and NAMESERVER for example) in a configuration. When an end-to-end trace is used, the traces for the trace components are written to files named <servicename>_<host>.<port_number>.<end-to-end-trace_name>.trc.				If the expensive statements trace is active, all statements that last longer than the specified threshold are traced. You can analyze the results on the Performance tab under Expensive Statements Trace.				

Expensive Statements Trace

Configure the expensive statements trace by specifying the necessary options.

Trace Status: Inactive Active

Threshold Duration (µs):

User Filter

No user filter

Database User:

Application User:

Table/View:

Application:

Passport Trace Level:

Trace parameter values

トレースの完了後、Expensive Statements ビューを選択し、APPLICATION_NAME 列をフィルターすることで、Tableau から送信されたステートメントが返されます。

```
select * from M_EXPENSIVE_STATEMENTS where APPLICATION_NAME like  
'%Tableau%'
```

結果セットには、各データベース操作の重要な情報が含まれています。その主な情報は、ユーザー名、ワークブック名、SQL ステートメント、開始時間、期間 (マイクロ秒)、影響を受けた DB テーブル、レコード数などです。「where」句で、このクエリをさらに特定のユーザーやワークブック、開始時間などに絞ることができます。

次の情報は特に役立つ場合があります。

- ・ ステートメント実行時の操作タイプ (OPERATION)
- ・ クエリの開始時間 (START_TIME)
- ・ クエリの継続時間 (DURATION_MICROSEC)
- ・ ステートメントの演算のための CPU 時間 (マイクロ秒) (CPU_TIME)
- ・ アクセスされたオブジェクト名 (OBJECT_NAME)
- ・ SQL ステートメント (STATEMENT_STRING)
- ・ ステートメント実行時のメモリのピーク使用量 (バイト) (MEMORY_SIZE)

記録された重要な操作タイプの一部:

操作	説明
AGGREGATED_EXECUTION	各データベースリクエストの総実行時間
CALL	プロシージャコールの実行時間
COMPILE	準備/解析時間
CURSOR_CLOSE	カーソルのクローズ時間
FETCH	フェッチ時間
SELECT, INSERT, UPDATE, DELETE	対応する操作の実行時間

Expensive Statements Trace の結果の例:

```

SQL Result
select * from M_EXPENSIVE_STATEMENTS where SESSION_VARIABLES like '%Tableau%'

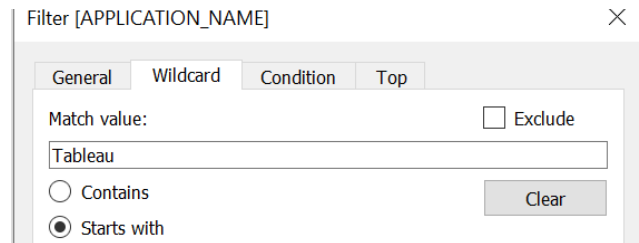
```

STATEMENT_STRING	CPU_TIME	STATEMENT_START_TIME	APPLICATION_SOURCE	APPLICATION_NAME	SESSION_VARIABLES
SELECT TO_DATE('tableau_FIN_AR_DFIAR30_CV','CLEARIN...	14.730	Jun 8, 2020 5:03:33.266443 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT TO_DATE('tableau_FIN_AR_DFIAR30_CV','CLEARIN...	14.653	Jun 8, 2020 5:03:33.266443 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT TO_DATE('tableau_FIN_AR_DFIAR30_CV','CLEARIN...	19.976	Jun 8, 2020 5:03:33.088574 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT 'tableau_FIN_AR_DFIAR30_CV'::ACCOUNTING_DO...	15.628	Jun 8, 2020 5:03:32.556884 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT 'tableau_FIN_AR_DFIAR30_CV'::ACCOUNTING_DO...	15.527	Jun 8, 2020 5:03:32.556884 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT 'tableau_FIN_AR_DFIAR30_CV'::ACCOUNTING_DO...	20.367	Jun 8, 2020 5:03:32.377876 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT ADD_DAYS(CAST(TO_DATE('tableau_FIN_AR_DFIAR...	22.388	Jun 8, 2020 5:03:32.098534 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT ADD_DAYS(CAST(TO_DATE('tableau_FIN_AR_DFIAR...	22.360	Jun 8, 2020 5:03:32.098534 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT 'tableau_FIN_AR_DFIAR30_CV'::CREDIT_AMOUNT_I...	14.600	Jun 8, 2020 5:03:32.01658 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT 'tableau_FIN_AR_DFIAR30_CV'::CREDIT_AMOUNT_I...	14.557	Jun 8, 2020 5:03:32.01658 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT ADD_DAYS(CAST(TO_DATE('tableau_FIN_AR_DFIAR...	21.459	Jun 8, 2020 5:03:31.908771 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR
SELECT 'tableau_FIN_AR_DFIAR30_CV'::CREDIT_AMOUNT_I...	18.963	Jun 8, 2020 5:03:31.838889 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	('APPLICATION': 'Tableau Desktop', 'APPLICATIONSOUR

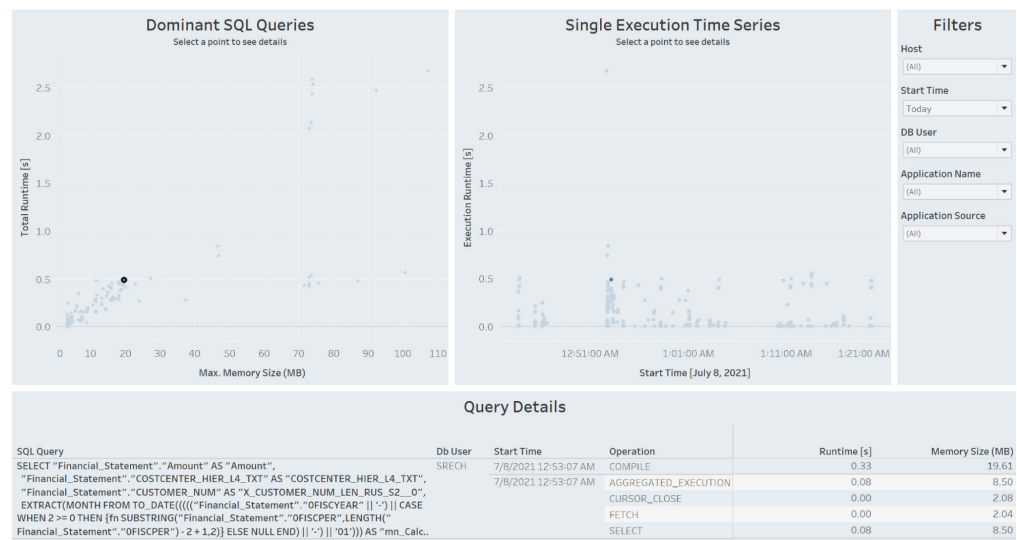
DURATION_MICROSEC	OBJECT_NAME	OPERATION	RECORDS
14,646	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	AGGREGATED_EXECUTION	35
14,604	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	SELECT	0
19,974	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	COMPILE	0
15,531	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	AGGREGATED_EXECUTION	70
15,480	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	SELECT	0
20,365	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	COMPILE	0
22,328	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	AGGREGATED_EXECUTION	1
22,317	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	SELECT	0
14,540	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	AGGREGATED_EXECUTION	31
14,514	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	SELECT	0
21,458	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	COMPILE	0
18,961	SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB...	COMPILE	0

Tableau での SAP HANA Expensive Statement Trace 結果の分析

Expensive Statements Trace の結果は、Tableau を使用して分析することができます。これを行うには、「SYS」スキーマの「M_EXPENSIVE_STATEMENTS」ビューへの接続を確立する必要があります。Tableau によってトリガーされたステートメントのみをフィルターするには、データソースフィルターを使用することができます (例: APPLICATION_NAME ディメンションに対する Tableau のワイルドカードフィルター)。



負荷の高いクエリを分析するためのダッシュボードの例:



SQL プランキャッシュの統計: 最適化の対象となり得る箇所の特定 (長時間および頻繁に実行)

SQL プランキャッシュは、SAP HANA データベースの SQL プロセスの理解に役立つ重要なツールです。システム内で実行されるステートメントの概要を提供し、実行ランタイムなどの統計を追跡します。これは、頻繁に実行されるクエリと遅いクエリに対するインサイトを提供するので、最適化が可能な箇所を特定するために使用できます。専用のトレースをアクティブ化する必要はありません。SQL プランキャッシュは SAP HANA ビューとしてクエリできるため、Tableau を使用してその情報を分析できます。

SAP HANA で SQL ステートメントを実行する前に、プランにコンパイルされます。プランがコンパイルされれば、次に同じステートメントが実行されるときに毎回新しいプランをコンパイルするよりも、そのプランを再利用するほうが無駄がありません。SAP HANA では、SQL プランキャッシュは以前の実行から生成されたプランを格納します。また、監視を目的として、各プランの統計を保持します。このため、実行数、最小/最大/合計/平均のランタイム、ロック/待機の統計などを分析できます。次の情報は特に役立つ場合があります。

- ・ 頻繁に実行されるステートメント (TOTAL_EXECUTION_TIME)
- ・ 長時間実行されているステートメント (AVG_EXECUTION_TIME)
- ・ 頻繁に実行されるプラン (EXECUTION_COUNT)
- ・ 返されたレコード数 (TOTAL_RESULT_RECORD_COUNT)

SAP Note 2000002: [FAQ: SAP HANA SQL 最適化 \(英語\)](#) には、SQL プランキャッシュでのランタイム統計の解釈方法について、詳細が記載されています。たとえば、操作は以下のアクションに分類されています。

操作	説明
CURSOR	SAP HANA サーバーの時間およびクライアントの時間を含む、すべてのカーソル時間が含まれます。データのフェッチ間にクライアントが他のタスクを実行した場合、カーソル時間は SAP HANA サーバーの時間よりも大幅に長くなります。
EXECUTION	SAP HANA サーバー側の実行時間 (オープン + フェッチ + ロック待機 + クローズ) が含まれます。テーブルの読み込みと準備時間は含まれません。
EXECUTION_OPEN	SAP HANA サーバー側でのオープン時間が含まれます。早期マテリアライゼーションによるリストアクセスの場合、データの実際の取得が含まれます。
EXECUTION_FETCH	SAP HANA サーバー側でのフェッチ時間が含まれます。行ストアアクセスまたは遅延マテリアライゼーションの場合、データの実際の取得時間が含まれます。
EXECUTION_CLOSE	SAP HANA サーバー側でのクローズ時間が含まれます。
TABLE_LOAD	準備時のテーブル読み込み時間が含まれており、準備時間の一部となります。
REPARATION	準備時間が含まれています。
LOCK_WAIT	トランザクションのロック待機時間が含まれます。内部ロックは含まれません。

SQL プランキャッシュの読み取り方のガイドについては、SAP のヘルプページにある [例: SQL プランキャッシュの読み取り \(英語\)](#) を参照してください。

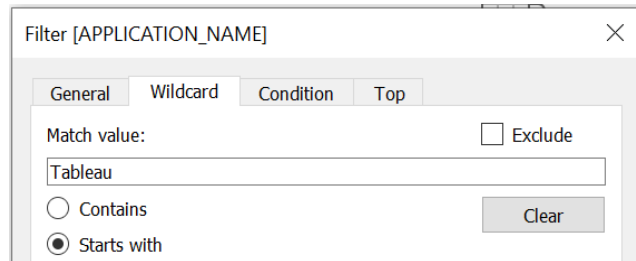
SQL プランキャッシュの分析に関する推奨事項については、SAP のヘルプページ、[SQL プランキャッシュ分析 \(英語\)](#) を参照してください。

Tableau での SAP HANA SQL プランキャッシュの分析

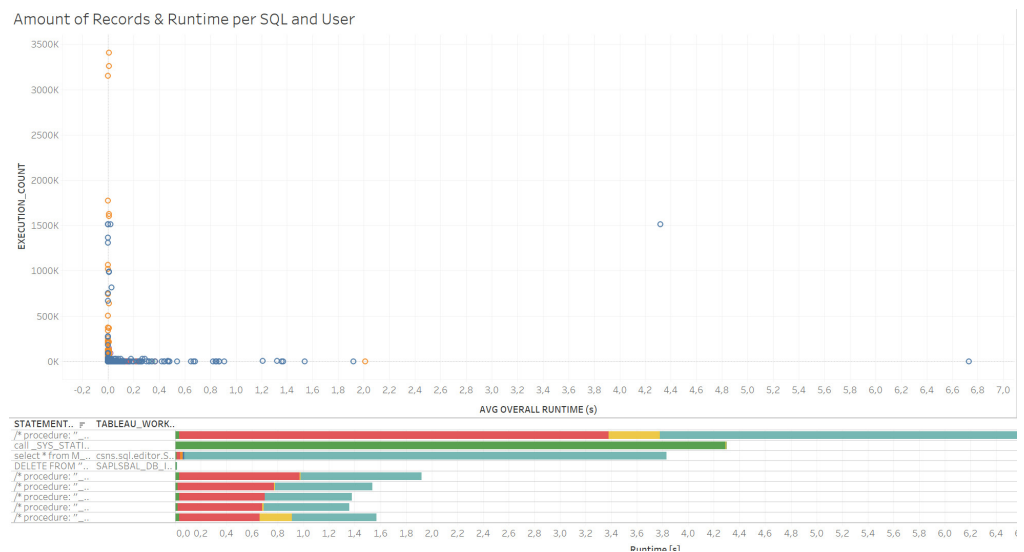
Tableau を使用して、SQL プランキャッシュ の結果を分析することができます。これを行うには、

「SYS」スキーマの「M_SQL_PLAN_CACHE」ビューへの接続を確立する必要があります。

Tableau によってトリガーされたステートメントのみをフィルターするには、データソースフィルターを使用することができます (例: APPLICATION_NAME ディメンションに対する Tableau のワイルドカードフィルター)。



ビジュアライゼーションの例には、実行回数と、ランタイム全体の平均およびランタイムの平均分布を含めることが可能です。



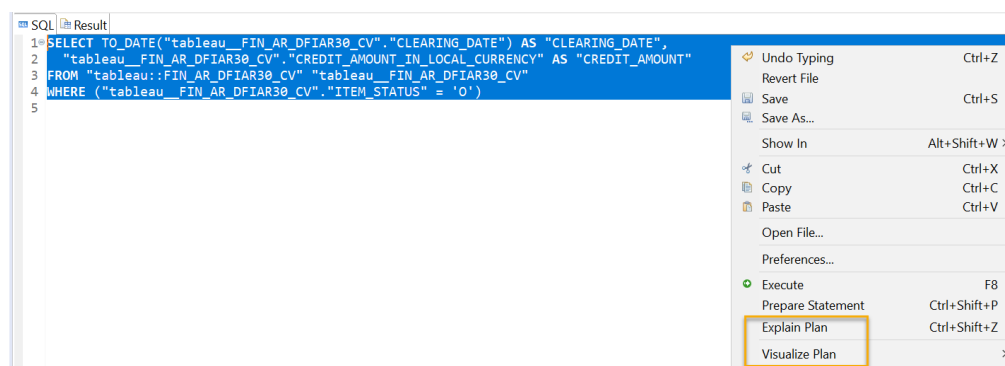
単一の SQL クエリの詳細分析:

SAP HANA Explain Plan および Visualize Plan

Expensive Statements Trace または SQL プランキャッシュ統計を使用して長時間実行のクエリを特定したら、次のステップはその実行に時間がかかっている理由を解明することです。

ランタイムの根本原因を特定するためには、SAP HANA が SQL ステートメントの実行をどう処理するかについての理解が役立ちます。これを調査する 2 つの方法が、SAP HANA の「Explain Plan」および「Visualize Plan」機能です。

これらのいずれかを実行する最も簡単な方法は、ステートメントを SQL コンソールにコピーし、コンテキストメニューで「Explain Plan」または「Visualize Plan」を選択することです。



それぞれの機能の詳細について、次の 2 つのセクションで説明します。

SAP HANA Plan Explanation

SQL ステートメントのプランの説明を生成すると、その結果には、クエリの実行と、処理中のデータベース操作に関する詳細情報が表示されます。

```
EXPLAIN PLAN FOR  
SELECT TO_DATE("tableau_FIN_AR_DFIAR30_CV"."CLEARING_DATE") AS "CLEARING_DATE",  
"tableau_FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_IN_LOCAL_CURRENCY" AS "CREDIT_AMOUNT"  
FROM "tableau::FIN_AR_DFIAR30_CV" "tableau_FIN_AR_DFIAR30_CV"  
WHERE ("tableau_FIN_AR_DFIAR30_CV"."ITEM_STATUS" = '0')
```

OPERATOR_NAME	OPERATOR_DETAILS	TABLE_NAME	OUTPUT_SIZE	SUBTREE_COST	OPERATOR_ID	LEVEL
1	COLUMN SEARCH TO_DATE(/B1H/AD_FIAR302.CLEARING_DATE), TO_DECIMALISUM(/B1H/AD_FIAR302.CRE...	?	348.718	0.0007147089139647216	1	1
2	AGGREGATION GROUPING: /B1H/AD_FIAR302.CLEAR_DATE, AGGREGATION: SUM(/B1H/AD_FIAR302.CRE...	?	348.718	0.00008454636905806701	2	2
3	COLUMN TABLE FILTER CONDITION: /B1H/AD_FIAR302.ITEM_STATUS = '0'	/B1H/AD_FIAR302	348.718	?	3	3

以下に、主な値の一部の簡単に説明し、それに続いて例を紹介します。詳細は、[SAP HANA SQL およびシステムビューのリファレンス \(英語\)](#) に記載されている EXPLAIN_PLAN_TABLE システムビューで確認してください。

地域	詳細
操作の詳細	OPERATOR_NAME 値は、結合、ユニオン、集約など、実行された操作タイプを示します。操作は、使用されたエンジン (基本的には行エンジンまたは列エンジン) に依存します。依存関係はインデントで示されます。以下の例をご覧ください。
エンジン	操作者が実行するエンジンのタイプは、EXECUTION_ENGINE 列に表示されます。ROW、COLUMN、OLAP、HEX、ESX
テーブルの詳細	テーブルの詳細には、アクセスされたテーブルの名前、タイプ、サイズ、またはオブジェクトの情報が含まれます。
推定コスト	コスト値には、推定出力行数 (OUTPUT_SIZE) および推定時間 (秒) (SUBTREE_COST) が含まれます。

Explain Plan の結果の解釈方法の例については、以下のページで説明しています。

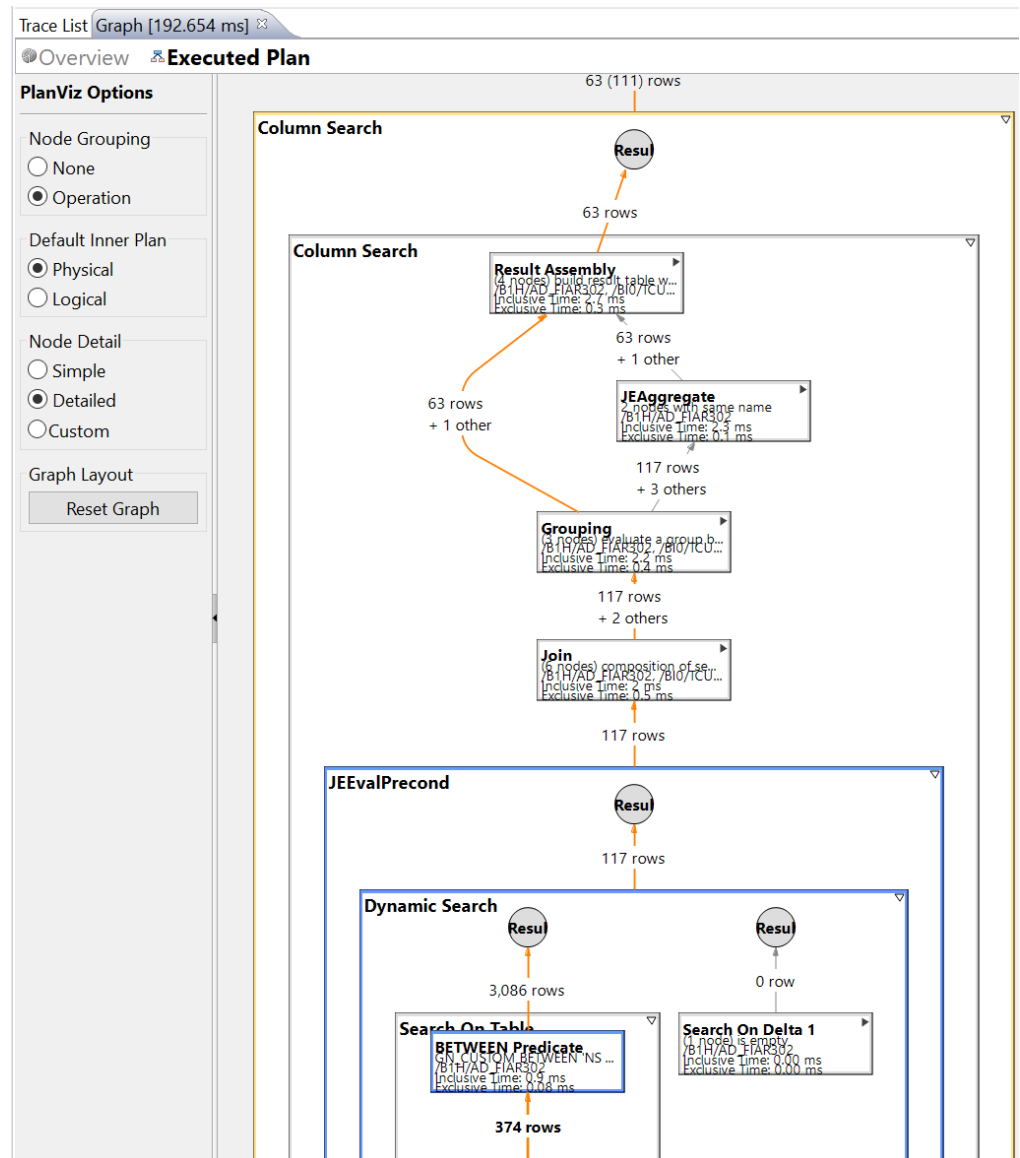
- ・ [Plan Explanation による SQL 実行分析 \(英語\)](#)
- ・ [SAP HANA SQL およびシステムビューのリファレンス: Explain Plan ステートメント \(英語\)](#)

SAP HANA Plan Visualizer

SAP HANA Plan Visualizer を使うと、SQL 実行プランをグラフで分析できます。これにより、処理に関連するステップと、レコードの量およびランタイムが時間の経過とともにどのようになるかが理解しやすくなります。

ランタイムは、「Exclusive」(ノードの実行時間) および「Inclusive」(子孫ノードを含む実行時間) の値として提示されます。

Time		Context	
Compilation	20.92 ms	SQL Query	SELECT TO_DATE("tableau_FIN_AR_DFIAR30_...
Execution	192.65 ms	System	bw4hana:30203
Dominant Operators		System Version	2.00.043.00.1569560581
Name	Execution Time	System Compile Type	rel
Basic Predicate	0.66 ms (0.34%)	Memory Allocated	11.7 MByte(s)
JEDistinctAttribute	0.25 ms (0.13%)	Data Flow	
Column Search	0.21 ms (0.11%)	Number of Tables Used [®]	2
Distribution		Result Record Count	63
Number of Nodes	1		
Number of Network Transfers	0		



Plan Visualizer は、分析のためのタイムラインビューやネットワークビューなどの追加のビューを提供します。それらの詳細については、SAP HANA トラブルシューティングおよびパフォーマンス分析ガイドの章「[Plan Visualizer による SQL 実行分析](#)」を参照してください。

また、SAP は PlanViz の使用方法に関するインストラクションを記載したブログをリリースしています。

- ・ [HANA PlanVisualizer \(PlanViz\) – 迅速で簡単 \(英語\)](#)
- ・ [Plan Visualizer \(PlanViz\) による SQL 実行分析 \(英語\)](#)

SAP HANA のクエリ処理のボトルネックを特定したら、それらに対処する必要があります。

SAP HANA 計算ビューのパフォーマンス最適化の可能性については、開発者向け SAP HANA パフォーマンスガイドの章「[計算ビューの最適化機能](#)」、および SAP のナレッジベース記事 2000002 ([FAQ: SAP HANA SQL 最適化 \(英語\)](#)) を参照してください。

Tableau が SAP HANA に送信する SQL クエリを変更するには、Tableau のダッシュボードの再モデリングを検討する必要があります。ダッシュボードのパフォーマンスに関するベストプラクティスについては、次のリソースを参照してください。[効率的に作業できるワークブックの設計](#)および[ダッシュボードパフォーマンスに関するベストプラクティス](#)。

まとめ

Tableau on SAP ダッシュボードのパフォーマンスを理解するには、Tableau 側と SAP HANA 側でいくつかのトレースを使用します。

単一のダッシュボードを最適化するには、Tableau のパフォーマンスレコーダーと SAP HANA の SQL トレースを組み合わせたエンドツーエンドの実行トレースをお勧めします。これにより、ランタイムが実行ステップと処理レイヤーにどのように分散されているかが明らかになり、ボトルネックを明らかにするのに役立ちます。

長期的なパフォーマンス監視には、SAP HANA Expensive Statements Trace が最適です。最小限のオーバーヘッドで永続的にアクティブ化しておくことができます。

長時間実行されている、またはメモリを大量に消費する SQL クエリを特定したら、SAP HANA の Plan Explanation または Plan Visualizer を使用してさらに分析し、DB の実行手順とランタイムの影響を理解することができます。

Tableau について

Tableau は、人々と組織がさらにデータドリブンになるのを支援する、エンタープライズ対応の完全な統合ビジュアル分析プラットフォームです。オンプレミスでもクラウドでも、また Windows でも Linux でも、Tableau はテクノロジーへの既存の投資を生かし、お客様のデータ環境の変化と成長に合わせた規模の拡張が可能です。お客様の最も貴重な資産であるデータと人材の力を解放します。