

salesforce

大量データボリューム 対策ポイントガイド

大量データ処理遅延の問題と対策

Salesforce

2022年10月31日



Forward Looking Statements



This presentation contains forward-looking statements about, among other things, trend analyses and future events, future financial performance, anticipated growth, industry prospects, environmental, social and governance goals, and the anticipated benefits of acquired companies. The achievement or success of the matters covered by such forward-looking statements involves risks, uncertainties and assumptions. If any such risks or uncertainties materialize or if any of the assumptions prove incorrect, Salesforce's results could differ materially from the results expressed or implied by these forward-looking statements. The risks and uncertainties referred to above include those factors discussed in Salesforce's reports filed from time to time with the Securities and Exchange Commission, including, but not limited to: the impact of, and actions we may take in response to, the COVID-19 pandemic, related public health measures and resulting economic downturn and market volatility; our ability to maintain security levels and service performance meeting the expectations of our customers, and the resources and costs required to avoid unanticipated downtime and prevent, detect and remediate performance degradation and security breaches; the expenses associated with our data centers and third-party infrastructure providers; our ability to secure additional data center capacity; our reliance on third-party hardware, software and platform providers; the effect of evolving domestic and foreign government regulations, including those related to the provision of services on the Internet, those related to accessing the Internet, and those addressing data privacy, cross-border data transfers and import and export controls; current and potential litigation involving us or our industry, including litigation involving acquired entities such as Tableau Software, Inc. and Slack Technologies, Inc., and the resolution or settlement thereof; regulatory developments and regulatory investigations involving us or affecting our industry; our ability to successfully introduce new services and product features, including any efforts to expand our services; the success of our strategy of acquiring or making investments in complementary businesses, joint ventures, services, technologies and intellectual property rights; our ability to complete, on a timely basis or at all, announced transactions; our ability to realize the benefits from acquisitions, strategic partnerships, joint ventures and investments, including our July 2021 acquisition of Slack Technologies, Inc., and successfully integrate acquired businesses and technologies; our ability to compete in the markets in which we participate; the success of our business strategy and our plan to build our business, including our strategy to be a leading provider of enterprise cloud computing applications and platforms; our ability to execute our business plans; our ability to continue to grow unearned revenue and remaining performance obligation; the pace of change and innovation in enterprise cloud computing services; the seasonal nature of our sales cycles; our ability to limit customer attrition and costs related to those efforts; the success of our international expansion strategy; the demands on our personnel and infrastructure resulting from significant growth in our customer base and operations, including as a result of acquisitions; our ability to preserve our workplace culture, including as a result of our decisions regarding our current and future office environments or work-from-home policies; our dependency on the development and maintenance of the infrastructure of the Internet; our real estate and office facilities strategy and related costs and uncertainties; fluctuations in, and our ability to predict, our operating results and cash flows; the variability in our results arising from the accounting for term license revenue products; the performance and fair value of our investments in complementary businesses through our strategic investment portfolio; the impact of future gains or losses from our strategic investment portfolio, including gains or losses from overall market conditions that may affect the publicly traded companies within our strategic investment portfolio; our ability to protect our intellectual property rights; our ability to develop our brands; the impact of foreign currency exchange rate and interest rate fluctuations on our results; the valuation of our deferred tax assets and the release of related valuation allowances; the potential availability of additional tax assets in the future; the impact of new accounting pronouncements and tax laws; uncertainties affecting our ability to estimate our tax rate; uncertainties regarding our tax obligations in connection with potential jurisdictional transfers of intellectual property, including the tax rate, the timing of the transfer and the value of such transferred intellectual property; uncertainties regarding the effect of general economic and market conditions; the impact of geopolitical events; uncertainties regarding the impact of expensing stock options and other equity awards; the sufficiency of our capital resources; our ability to comply with our debt covenants and lease obligations; and the impact of climate change, natural disasters and actual or threatened public health emergencies, including the ongoing COVID-19 pandemic.

Salesforce の定義する 大量データとは

Salesforce では次の定義をしています

“リリースに**数万人のユーザ**、**数千万件のレコード**、あるいは**合計で数百ギガバイト**ものレコードストレージが含まれる場合、そのリリースには「**大量のデータ**」があります”

[🔗 大量のデータを使用するリリースのベストプラクティス](#)

また、Salesforce において大量データを扱う場合には、マルチテナント・メタデータアーキテクチャの理解も不可欠です。マルチテナント・メタデータアーキテクチャの概要は、こちらの資料も参考になさってください。

[🔗 マルチテナンシーおよびメタデータの概要](#)



アジェンダ

I. Salesforce内でのバッチやレポートでの処理遅延対策

1. 検索性能の最適化
2. データ共有モデルの最適化
3. Apex バッチ処理の最適化
4. レポート実行処理の最適化

II. Salesforce 外部からの APIコールイン遅延処理対策

III. 大量データの事例

IV. まとめ

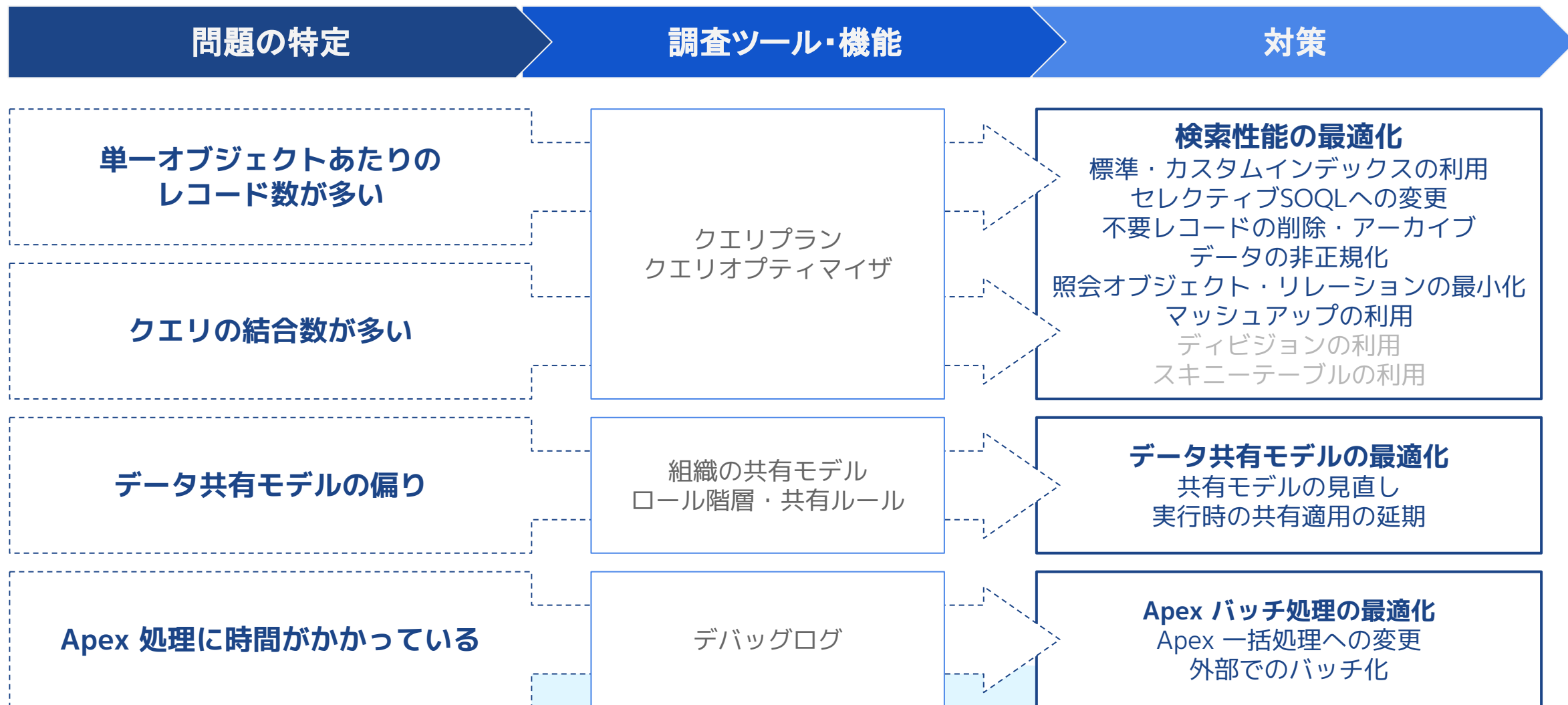


I. Salesforce内でのバッチや レポートでの処理遅延対策

Salesforce内でのバッチやレポートでの処理遅延対策



Salesforce 内だけで完結するApexバッチ、レポートなどで遅延の発生するケース



対策 #1 検索性能の最適化



1-1. 標準・カスタムインデックスの利用

クエリの処理速度を短縮化するためには、インデックスの利用が不可欠です

標準インデックス

次の項目は、自動的にインデックス作成

RecordTypeId
Division
CreatedDate
Systemmodstamp(LastModifiedDate)
Name
Email (取引先責任者とリード)
外部キーリレーション(参照と主従関係)
各オブジェクトの主キー (Salesforce ID / SFID)

※ 検索がインデックスにヒット (選択的) しないとフルスキャンとなり、パフォーマンスが劣化する原因となります

カスタムインデックス

弊社サポートにて一部付与可能

自動で付与される	外部ID (自動採番, メール, 番号, テキスト)
付与できない (非サポート)	<ul style="list-style-type: none"> ● 複数選択リスト ● ロングテキストエリア ● リッチテキストエリア ● 非決定性数式項目 ● 暗号化されたテキスト項目

カスタムインデックスを付与する場合は、サポートヘケースを上げて下さい



1-2. セレクティブ (選択的) SOQL への変更

SOQLで最高のパフォーマンスを得るためには、セレクティブであることが必須です

セレクティブ(選択的)にならない条件

1. クエリの対象が特定の閾値を超える*1-3
2. 検索条件の演算子が「非セレクティブ」
3. 検索条件に“CONTAINS”かつスキャン行数が 333,000 を超える場合
4. 空の値と比較する場合 (Name = “”)

非セレクティブな SOQL たち	
否定演算子	NOT EQUAL TO , != NOT CONTAINS NOT LIKE EXCLUDES
テキストとの比較演算子	text_field < text_field > text_field <= text_field >=
後方/中間一致 (頭に%)	LIKE '%string%'
非決定性数式項目	時間関数 (TODAY(), NOW())

1-3. セレクティブSOQLとなるインデックスの条件

オブジェクト内のレコード数に対して、インデックスが効果する上限があります

クエリオプティマイザ - 標準インデックスの場合

クエリ最適化の上限レコード数

$$\frac{1 \sim 100 \text{万件の} 30\%}{\uparrow 30 \text{万件が上限}} + \frac{100 \text{万件からの} 15\%}{\uparrow 70 \text{万件が上限}} = \text{最終的な上限} = 100 \text{万件}$$

レコード数	はじめの上限	以降の上限	最終的な上限
≤ 1,000,000	全体の30%	N/A	全体の30%
≤ 2,000,000	300,000	150,000	450,000
≤ 3,000,000	300,000	300,000	600,000
≤ 4,000,000	300,000	450,000	750,000
≤ 5,000,000	300,000	600,000	900,000
≤ 5,600,000	300,000	700,000	1,000,000

クエリオプティマイザ - カスタムインデックスの場合

クエリ最適化の上限レコード数

$$\frac{1 \sim 100 \text{万件の} 10\%}{\uparrow 10 \text{万件が上限}} + \frac{100 \text{万件からの} 5\%}{\uparrow 233,333 \text{件が上限}} = \text{最終的な上限} = 333,333 \text{件}$$

レコード数	はじめの上限	以降の上限	最終的な上限
≤ 1,000,000	全体の10%	N/A	全体の10%
≤ 2,000,000	100,000	50,000	150,000
≤ 3,000,000	100,000	100,000	200,000
≤ 4,000,000	100,000	150,000	250,000
≤ 5,000,000	100,000	200,000	300,000
≤ 5,600,000	100,000	333,333	333,333

※ 複数の項目の式を結合するようなクエリの場合、and/or で効果が異なります

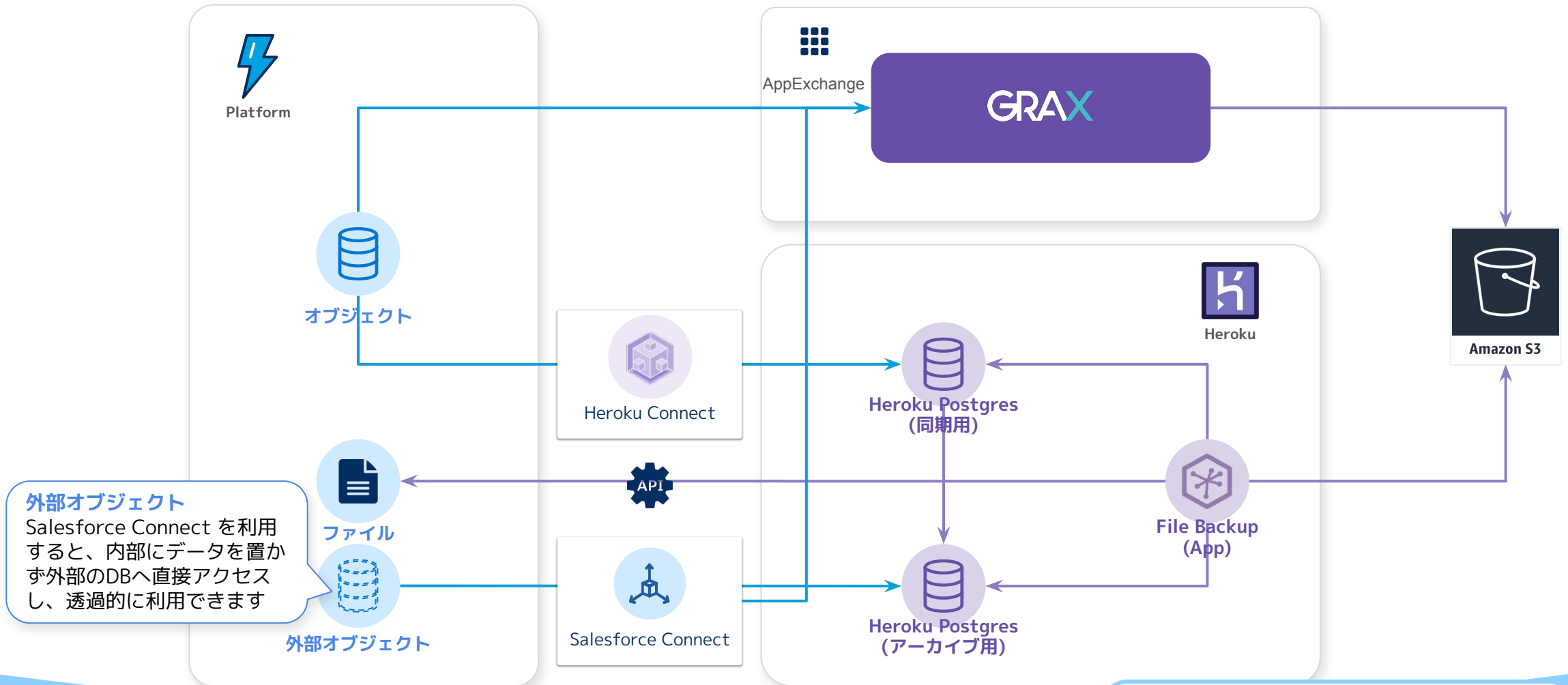
and : 数式のいずれかが、このセレクティブに当てはまれば効果がある

or : いずれの数式も、この上限に収まっていなければ効果がない(全てインデックスがあること)

※ LIMIT 句でこの数値未満になるよう絞り込みを行っても、インデックスは利用されません。検索条件がこの範囲に入ること。

1-4. 不要レコードの削除・アーカイブ

AppExchange のツールや、Heroku を活用し、不要なレコードをアーカイブ

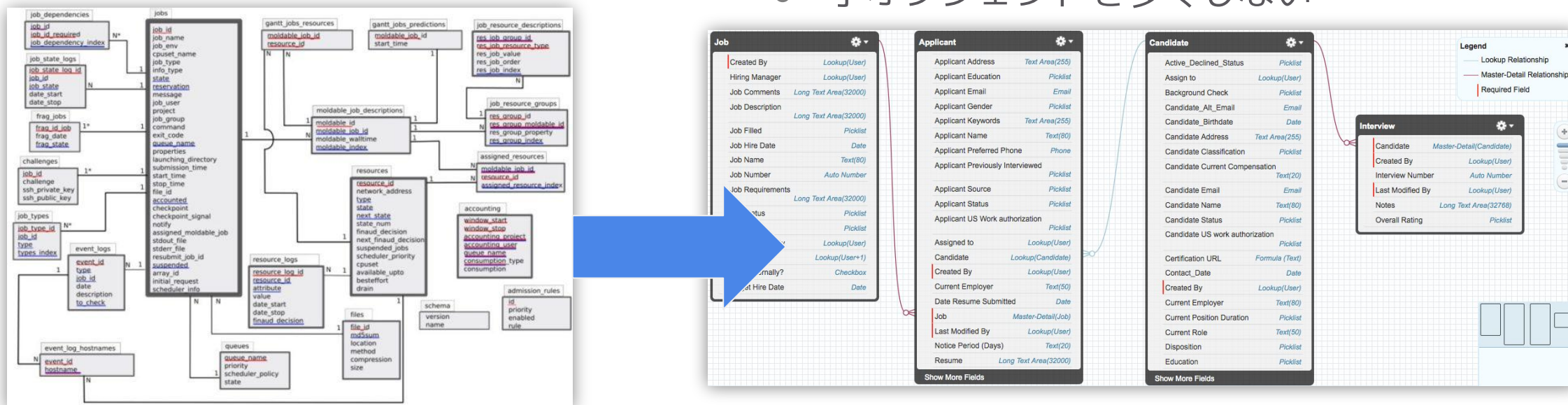


1-5. データの非正規化 照会オブジェクト/リレーションの最小化



通常のリレーショナルデータベースのように正規化しすぎないことが重要

- 一般的なリレーショナルデータベースでは、個々のデータの独立性を高め、不整合が生じないデータ構造へ正規化を行う
- Salesforce の場合、緻密な正規化がパフォーマンスに密に影響する
- 一般的なRDBMSレベルの正規化を行わず、業務の目的から考慮する
- システム移行時には、データモデルをそのまま移行しない
- レコードタイプの利用
- 子オブジェクトを多くしない



1-6. マッシュアップの利用

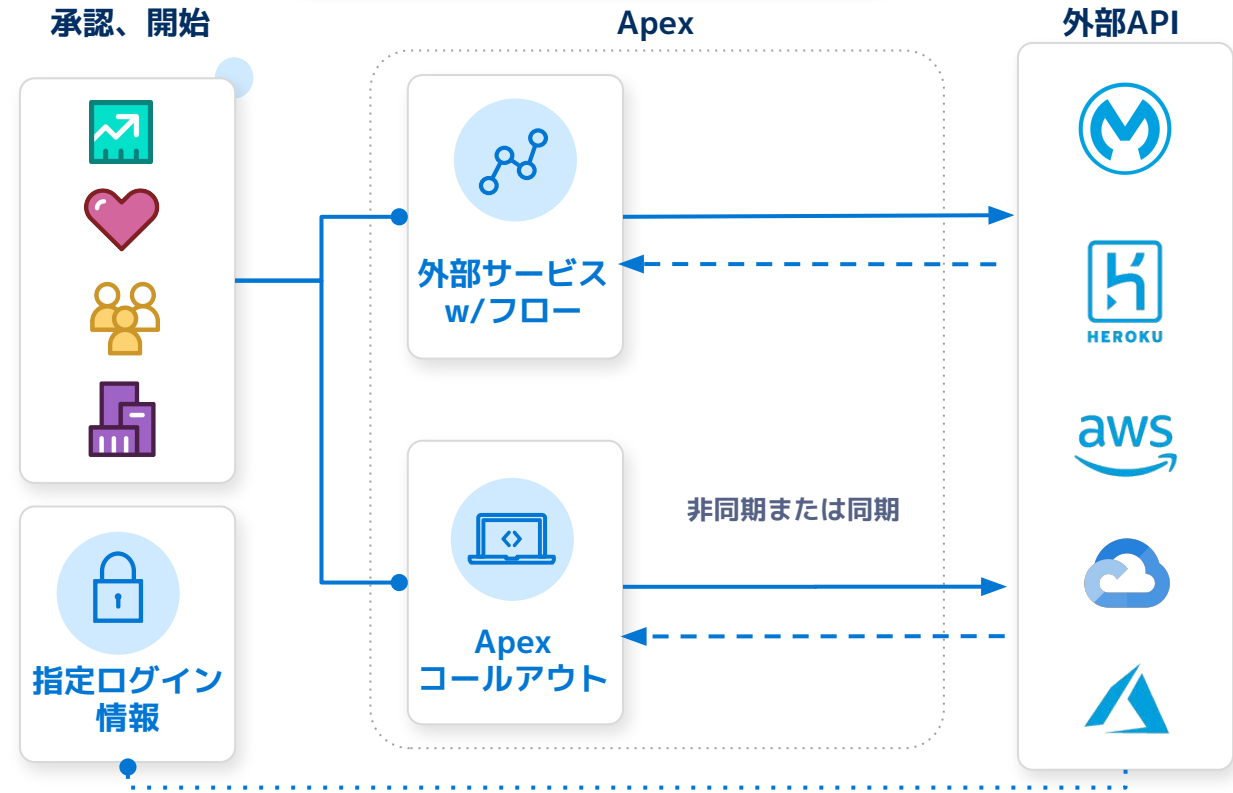
Salesforce 内ではなく、大量データを扱える外部のサービスを利用

Salesforce UI へ外部Webサイト表示



🔗 キャンパスの概要

外部エンドポイントへのコールアウト



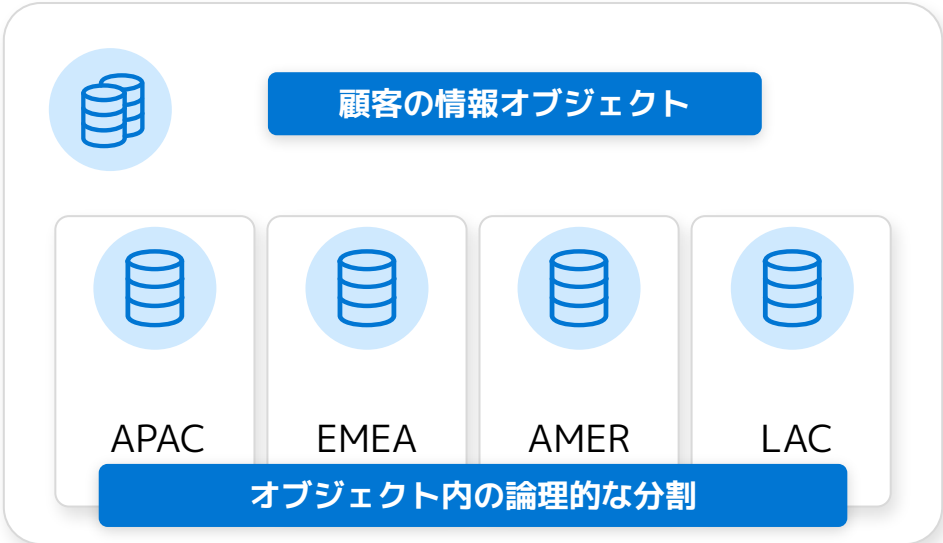
🔗 Apexインテグレーションサービス

🔗 非同期Apex

(参考) ディビジョン & スキニーテーブル

ディビジョン

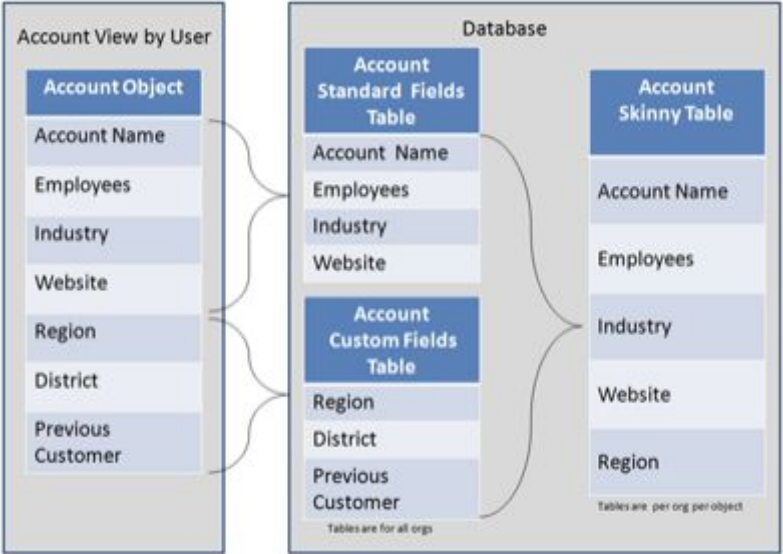
組織のデータを論理セクションに分類し、検索、レポート、リストビューなどのレコード数削減に寄与します。



[ディビジョンを使用してデータを整理する](#)

スキニーテーブル

頻繁に使用される項目を持つ特別なDBテーブルを作成します。



[スキニーテーブル](#)

(参考) 診断ツール



クエリ実行プランツール

パフォーマンスに問題があるクエリを評価し、原因を探ります

Id	CaseNumber
500100000meBuTAAU	00001584
500100000meBumAAE	00001604
500100000meBujAAE	00001605
500100000meBuAAE	00001592
500100000meBuWAAU	00001607
500100000meBuYAAU	
500100000meBuZAAU	
500100000meBudAAE	
500100000meBuXAAU	
500100000meBuAAE	
500100000meBuAAE	

Cardinality	Fields	Leading Operation Type	Cost	sObject Cardinality	sObject Type
83		TableScan	2.66481...	92	Case

Notes:
Not considering filter for optimization because unindexed. Table: Case Fields: ["IsDeleted"]
Not considering filter for optimization because the operator is not optimizable and the filter is not owner-optimizable. Table: Case Fields: ["Status"]

効率的なクエリの作成

Lightning ページ分析

Lightning アプリケーションビルダーから、該当ページのレスポンスタイム分析が可能

ページパフォーマンス (デスクトップ): 普通

予測ページ読み込み時間: 2.8 秒

このページの項目、コンポーネント、メタデータに加えて、ユーザーに固有の一部の総計値をチェックしました。Salesforce では体験ページ時間 (EPT) でパフォーマンスを測定しています。EPT を改善できる領域が見つかりました。Lightning ページのパフォーマンスを改善する方法については、[Salesforce ヘルプ](#)を参照してください。

Lightning ページのパフォーマンス分析

対策 #2 データ共有モデルの最適化



2-1. 共有モデルの見直し



公開グループが深いネスト構造になっていたり、
グループの合計数が多い場合に遅くなる

ネストは 5レベル以内
公開グループの合計数は 100,000 以内に抑える

プロファイル・権限セットや階層、共有ルールなど
ネイティブ共有コンポーネント利用時に遅くなる

Apex による共有管理 (プログラムによる共有)
方法を検討する
[🔗 Apex を使用したレコードの共有](#)

取引先が多い時(>200万)に遅くなる

チームやエンタープライズテリトリー管理が実装されて
いる場合には、モデルの見直しを行う

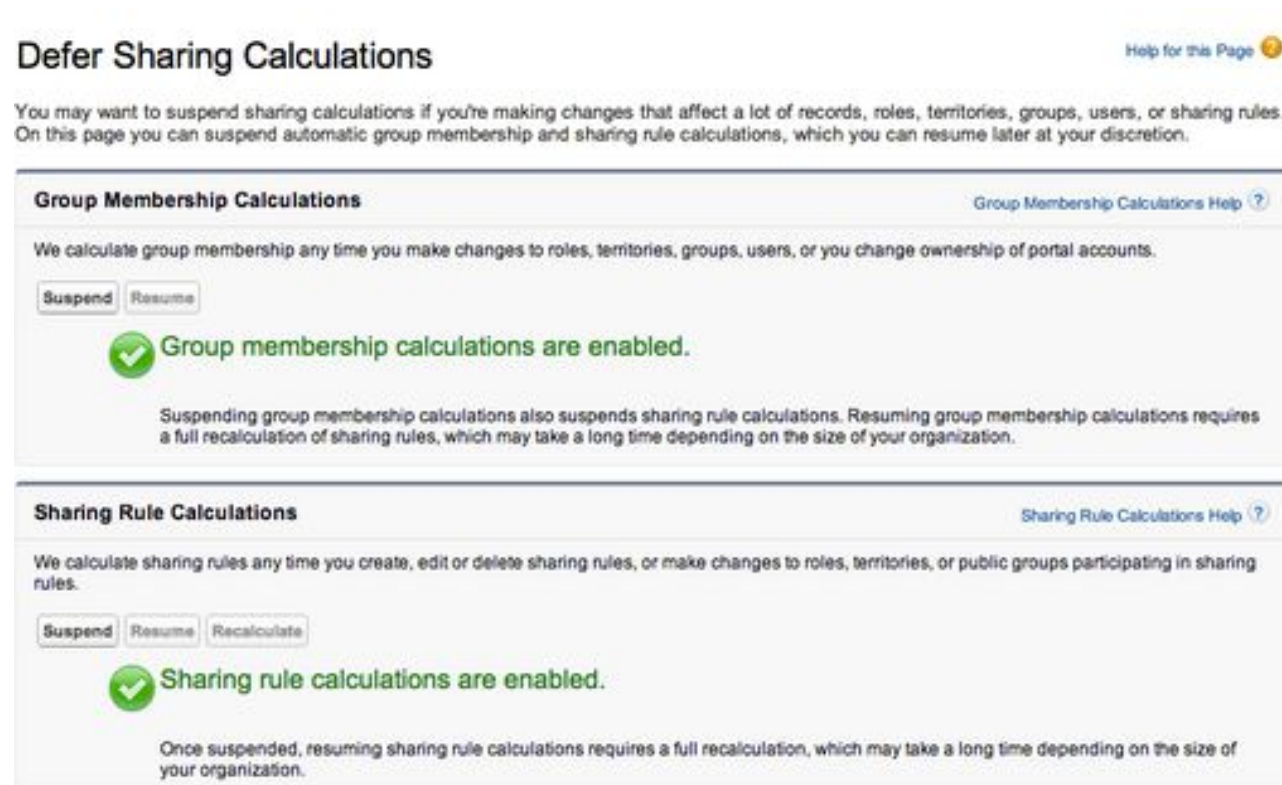
次のケースでデータスキューが発生し遅延する
親レコードに対し、子レコードが 10,000以上
単一の所有者がレコードを 10,000以上所有
データスキュー：1データに対し大量データが関連付けられている
[🔗 データモデルの設計](#)

子レコードを多く持たないオブジェクト構造にする
所有者ユーザレコードでロール階層内のロールを保持しない
or ロール階層の自分のブランチで最上位に配置する
[🔗 共有モデルの仕組み](#)

2-2. 実行時の共有適用の延期

複雑な共有ルールが適用されているときに大量のデータをロードするには

共有ルールの計算は時間がかかるため、初期や月次処理などで大量のデータをロードする際には「共有ルールの適用の延期」を行うことで、登録の時間を短縮化できます




Defer Sharing Calculations [Help for this Page](#)

You may want to suspend sharing calculations if you're making changes that affect a lot of records, roles, territories, groups, users, or sharing rules. On this page you can suspend automatic group membership and sharing rule calculations, which you can resume later at your discretion.

Group Membership Calculations [Group Membership Calculations Help](#)


We calculate group membership any time you make changes to roles, territories, groups, users, or you change ownership of portal accounts.

 **Group membership calculations are enabled.**

Suspending group membership calculations also suspends sharing rule calculations. Resuming group membership calculations requires a full recalculation of sharing rules, which may take a long time depending on the size of your organization.

Sharing Rule Calculations [Sharing Rule Calculations Help](#)

We calculate sharing rules any time you create, edit or delete sharing rules, or make changes to roles, territories, or public groups participating in sharing rules.

 **Sharing rule calculations are enabled.**

Once suspended, resuming sharing rule calculations requires a full recalculation, which may take a long time depending on the size of your organization.

対策 #3

Apex バッチ処理の最適化



3-1. Apex 一括処理への変更

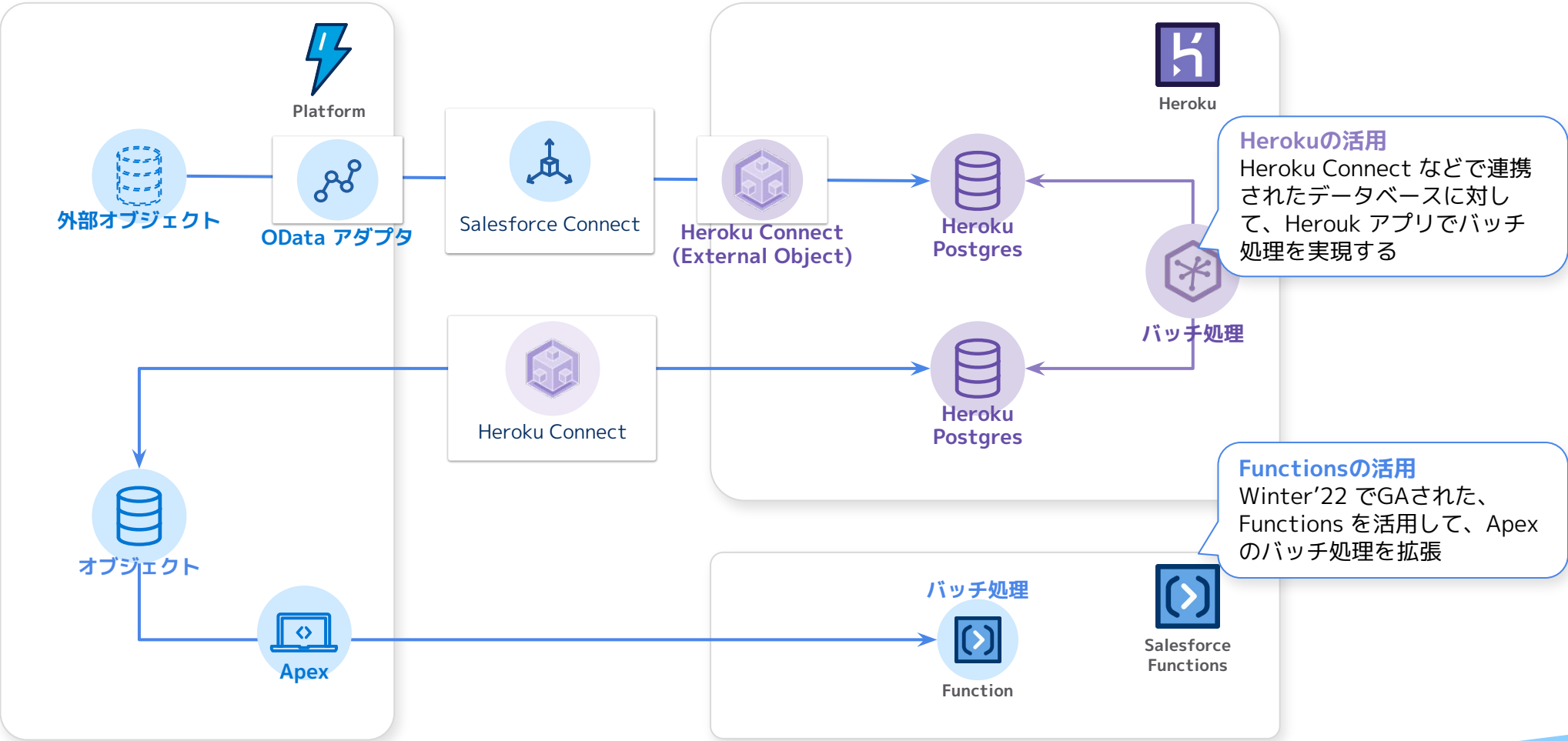
大量のデータを高速に処理するには、Apex一括処理を利用します



```
public class MyBatchClass implements Database.Batchable<sObject> {  
    public (Database.QueryLocator | Iterable<sObject>) start(Database.BatchableContext bc) {  
        // バッチ処理するためのオブジェクトまたはレコードを収集する  
    }  
    public void execute(Database.BatchableContext bc, List<P> records){  
        // 取得したクエリを処理単位毎に処理を行う (デフォルトのバッチサイズは 200レコード/順序は保証されない)  
    }  
    public void finish(Database.BatchableContext bc){  
        // 全てのバッチが処理された後に 1度だけ実行される  
    }  
}
```

3-2. 外部でのバッチ化

Salesforce 外の、よりリソースの多い、処理速度の速いアプリを活用する



Herokuの活用
Heroku Connect など連携されたデータベースに対して、Herouk アプリでバッチ処理を実現する

Functionsの活用
Winter'22 でGAされた、Functions を活用して、Apex のバッチ処理を拡張

対策 #4 レポート実行処理の最適化



レポート実行処理を最適化するための原因と対策

処理のパフォーマンスに影響する原因を把握し、対策を検討する

対策4-1

効率的な検索条件の使用

- 効率的な検索条件を使用し、返されるレコード数を制限する

対策4-2

不要な情報の削減

- レコード数削減と同様な観点で不要な情報を削減する

対策4-3

効率的な数式の記述

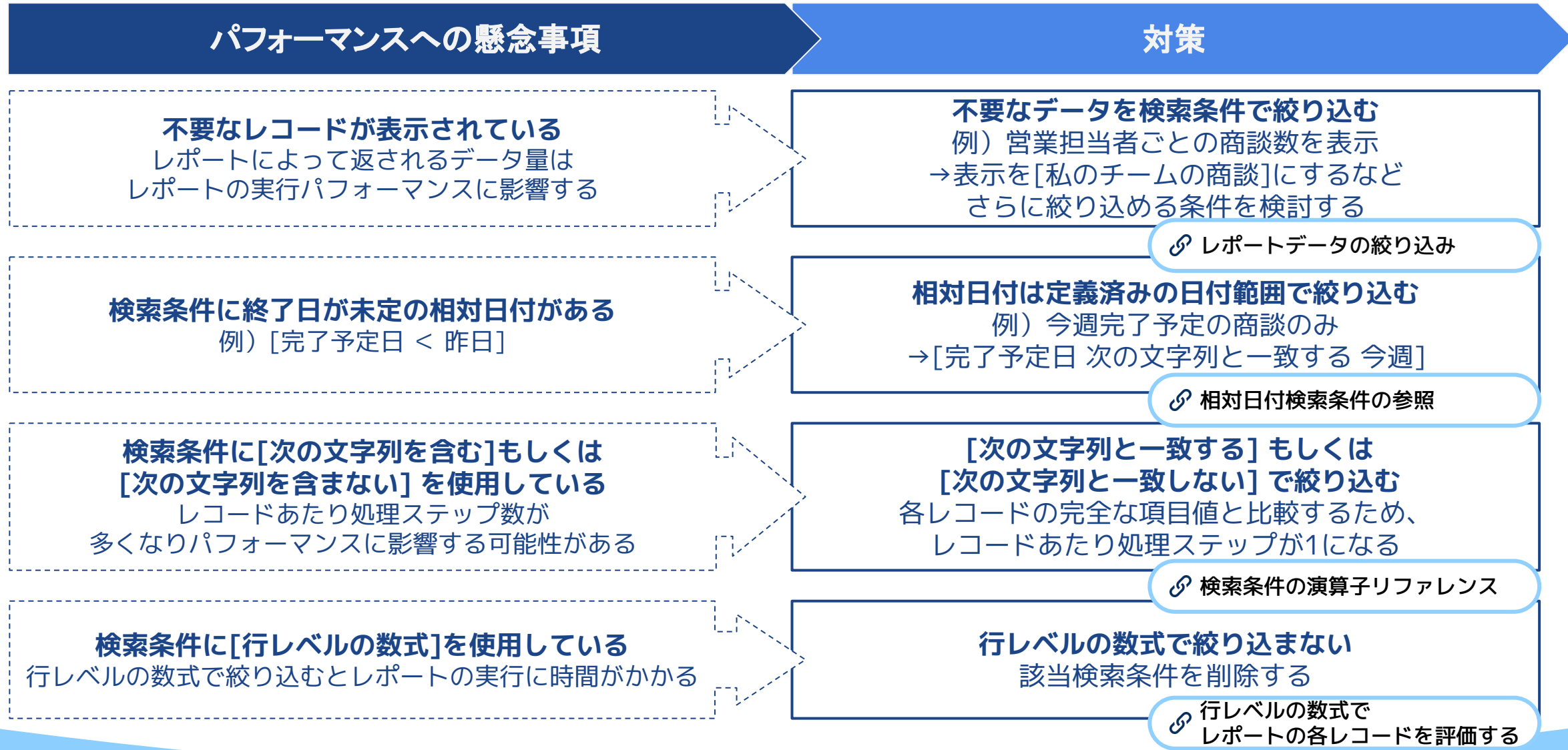
- 非効率的な記述を理解し、効率的な数式を記述する

対策4-4

その他対策

- その他パフォーマンス改善対策を実施する

4-1. 効率的な検索条件の使用



4-2. 不要な情報の削減

パフォーマンスへの懸念事項

対策

レポートに不要な列がある
多くのレコードが返されると時間がかかるのと同様に、
列数が多い場合も時間がかかる可能性がある

**よく使用されるレポートタイプから
デフォルトの列を削除する**

[カスタムレポートタイプから作成した
レポートの項目レイアウトの設計](#)

**グループ化されたデータを含むレポートに
表示不要な詳細行を表示している**

詳細行を非表示にする
個別のレコードはレポートに表示されないが、
グルーピング、合計、小計、レコード件数などの
主要な詳細は引き続き表示される

[レポートの詳細の表示と非表示](#)

4-3. 効率的な数式の記述



レポート数式を多用している
レポートの実行には
集計項目と行レベルの数式両方が評価される

**レポート数式の代わりに
レポートタイプに数式項目を追加する**
数式項目はワークフローで実行されるため、
レポートの実行時間に影響しない

[カスタムレポートタイプの設定](#)

不要な行レベルの数式を使用している
レポートの各レコードを評価するための
優れたツールではあるが、
レポートの実行に時間がかかる可能性がある

行レベルの数式を使用しないレポートに再設計する

- 合計値、最小値、最大値、平均値は集計機能を使用する
- 高度な数式はレポートタイプに数式項目を追加し使用する

行レベルの数式でデータをグループ化している

行レベルの数式でデータをグループ化しない

4-4. その他

パフォーマンスへの懸念事項

対策

多数のバケットを作成している

レポートで返される全てのレコードがバケットに属しているかを評価するため、時間がかかる可能性がある

慎重にバケットを追加する

多数のバケットがあり、レポートに時間がかかる場合は、より少ないバケットを使用するように再設計を検討する

複雑な共有ルールを採用している

レポートがオブジェクトからデータを取得するたびに各共有ルールを評価ため、時間がかかる可能性がある

共有ルールを簡略化する

可能な限り共有ルールは簡略化しておく

ごみ箱に大量データが存在する

論理削除（ごみ箱に存在する）の場合、データはまだ存在しているため時間がかかる可能性がある

レコードを「物理削除」する

ごみ箱を空にすることでレポートのパフォーマンスが向上する

[Lightning Experience でのごみ箱の管理](#)

上記の対策を実施してもレポート実行に時間がかかる

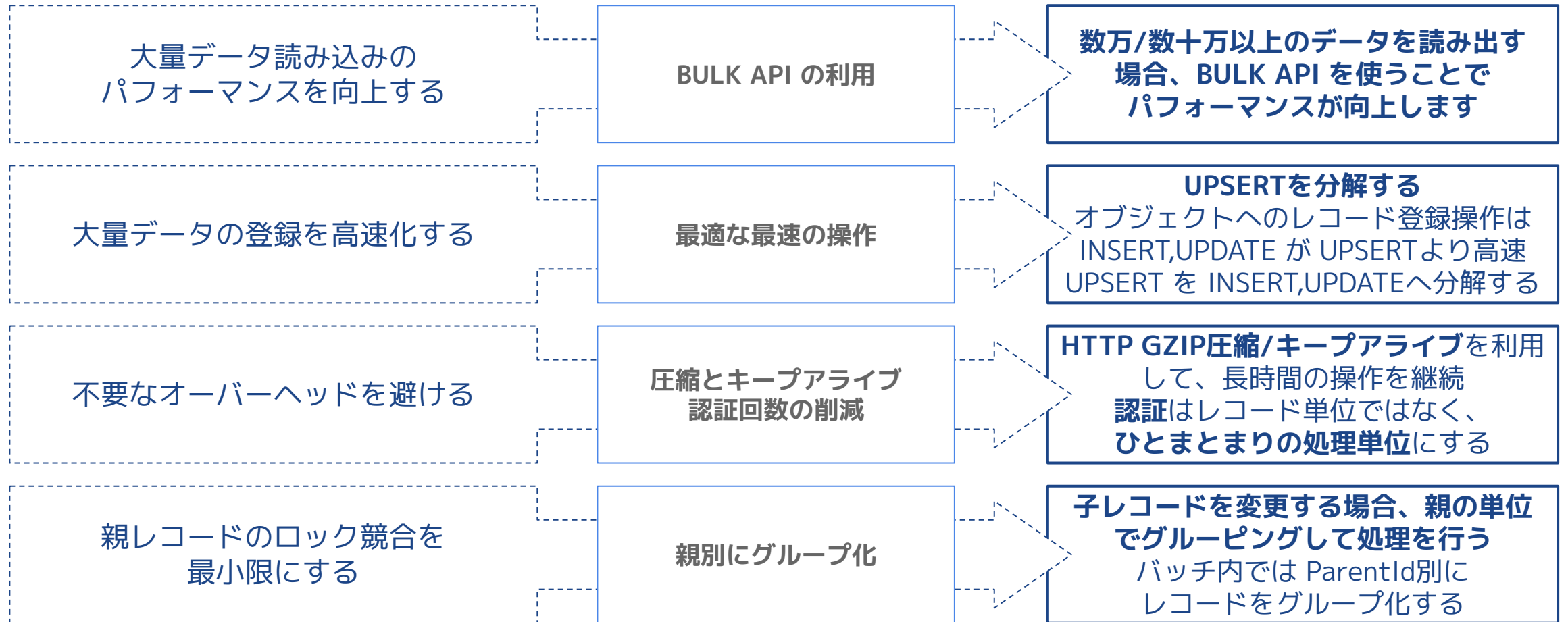
ピーク時間外にレポートを実行する

例) 毎日午前6時に実行するスケジュールを登録し、レポートの実行結果をメールで受け取る

[レポートのスケジュールと登録](#)

II. Salesforce 外部からの APIコールイン処理遅延対策

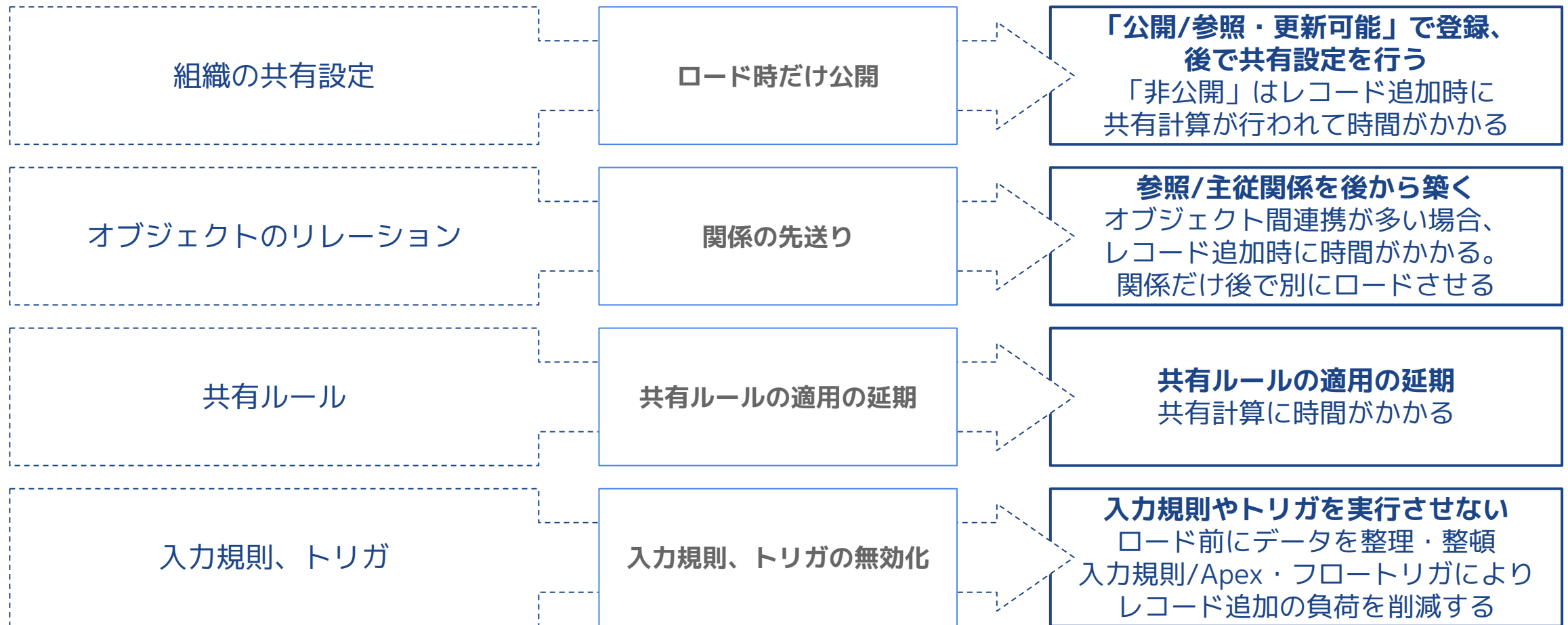
Salesforce 外部からのAPIコールイン処理遅延対策



Salesforce へ大量のデータをロードする時の考慮事項



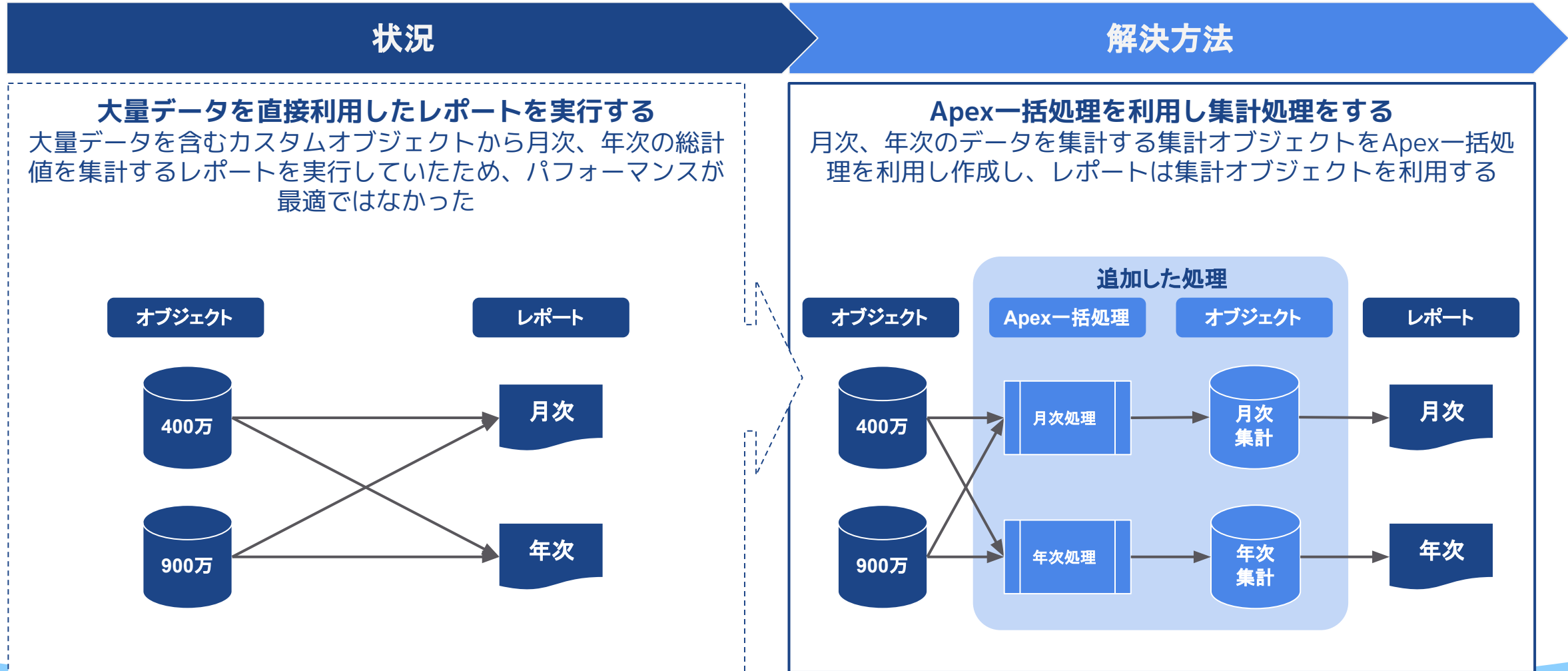
初期ロード時など、重要ではないプロセスを先送りにすることも検討する



III. 大量データの事例

1. データ集計

Apex一括処理を利用し、レポート作成を効率化する事例



2.null を使用したインデックス付け

項目の設計を変えずにクエリを早くする事例

状況

null値許可によりインデックスが使用できない

nullを許した項目をクエリで使用する必要があった。
しかし、選択リストおよび外部キー項目の単一列インデックスはnull行が除外されるため、以下のクエリにインデックスを使用できなかった。

インデックスが使用されない例

```
1 SELECT Name
2 FROM Object
3 WHERE Status__c = ''
```

解決方法

nullの代わりにN/Aやblankなど他の文字列を使用する

null値を許可しないことがベストプラクティスではあるが、難しい場合はnullの代わりにテキストを表示する数式項目を作成し、その項目にインデックスを付ける。

※参考：nullを含むカスタムインデックスをSalesforce カスタマーサポートに依頼することも可能

nullの場合、blankを表示する数式項目の作成

```
1 Status_Value__c
   = IF(ISBLANK(Status__c), "blank", Status__c)
```

インデックスが使用される例（上記で作成した数式項目の使用）

```
1 SELECT Name
2 FROM Object
3 WHERE Status_Value__c = 'blank'
```

3.大量のデータを含む関連リストの表示

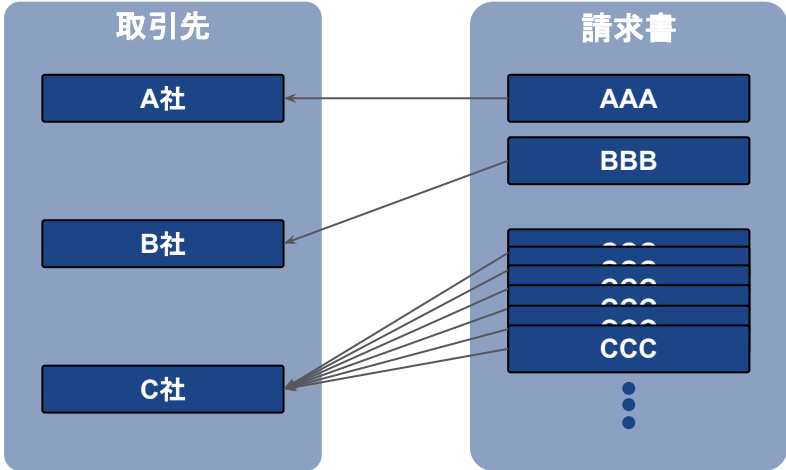
レコードのページレイアウト表示時間改善事例



状況

取引先レコード表示に時間がかかる

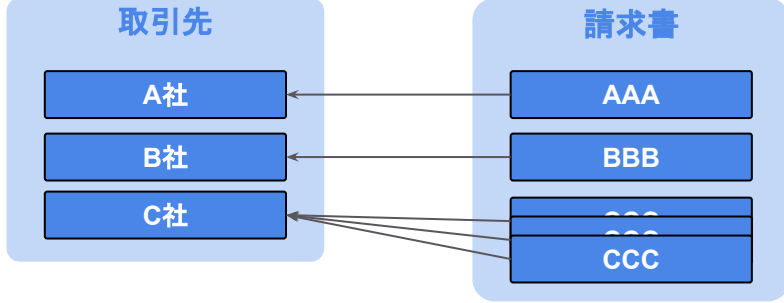
取引先（数十万）と請求書（1,500万）が主従関係にあり、請求書情報を関連リストとしてレイアウトに表示している。また、ほとんどの取引先に対する請求書は数件だったが、中には数千件の請求書を含むレコードも存在した。その結果、請求書関連リストの表示時間が長いいため、各取引先レコードが表示されるのに時間がかかっている。



解決方法

データスキューを解消する

子オブジェクトのデータスキューを最小限に抑える。



[関連リストの別途読み込み]を有効化する

※注意：Salesforce Classic にのみ有効
[設定]-[ユーザーインターフェース]-[ユーザーインターフェース]-[関連リストの別途読み込み]を有効化することで、関連リストのクエリが完了するのを待つ間に取引先詳細を先に表示できる。

- 関連リストの別途読み込みを有効化
- 外部オブジェクトの関連リストの別途読み込みを有効化

4.クエリの並び替えの最適化

インデックスのしきい値超過時のクエリ最適化事例

状況

しきい値超過によりインデックスが使用できない
クエリのデータ量が、合計レコード数の30%（最大で100万まで）という標準インデックスのしきい値を超えてしまったため、クエリのパフォーマンスが良くない。

過去3日間に作成された全てのレコードを検出するクエリ

```
1 SELECT Id,Product_Code__c
2 FROM Customer_Product__c
3 WHERE CreatedDate = Last_N_Days:3
```

解決方法

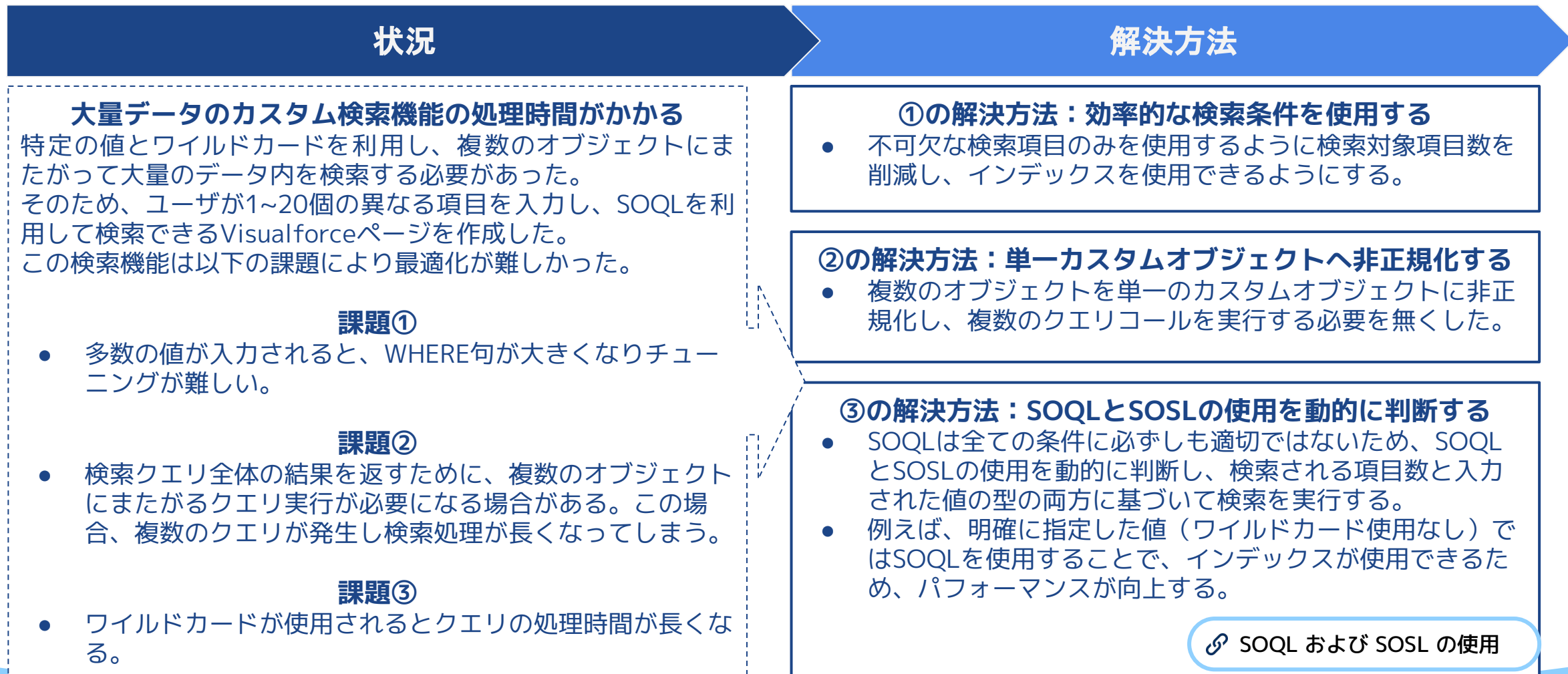
ORDER BY,LIMITによりクエリを最適化する
レコード数を100,000件未満に制限できる場合、インデックス付き項目に対しORDER BY,LIMIT句を指定すると、しきい値チェックを実行せず、インデックスを使用することができる。

ORDER BY,LIMIT指定によりインデックスが使用されるクエリ

```
1 SELECT Id,Product_Code__c
2 FROM Customer_Product__c
3 WHERE CreatedDate = Last_N_Days:3
4 ORDER BY CreatedDate LIMIT 99999
```

5. カスタム検索機能

SOQLの複雑なロジックによる最適化課題解決事例



6. 関連レポートのパフォーマンス

大量データの関連レポートの最適化事例

状況	解決方法
<p>複数オブジェクトによるレポート表示に時間がかかる 取引先 (31.4万)、販売注文 (76.9万)、販売詳細 (230万)、取引先会社形態 (120万) という4つの関連オブジェクトを使用してレポートを作成した。しかし、このレポートでは条件検索があまり行われず、最適化する必要があった</p> <div data-bbox="267 721 1065 1273" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">カスタムレポートタイプ</p> </div>	<div data-bbox="1286 444 2402 729" style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <p style="text-align: center;">検索条件の追加によるクエリの選択度を高める</p> <ul style="list-style-type: none"> 検索条件を追加してクエリの選択度を高め、できるだけ多くの検索条件にインデックス付けができるようにする。 </div> <div data-bbox="1286 729 2402 1015" style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <p style="text-align: center;">不要なデータを削減する</p> <ul style="list-style-type: none"> 不要な各オブジェクトのデータ量を削減した。 ごみ箱を空の状態に保つようにした。ごみ箱にデータが入っていると、クエリのパフォーマンスに影響及ぼす可能性がある。 </div> <div data-bbox="1286 1015 2402 1318" style="border: 1px solid black; padding: 10px;"> <p style="text-align: center;">複雑な共有ルールを削減する</p> <ul style="list-style-type: none"> 4つの関連オブジェクトに複雑な共有ルールが存在しないようにした。複雑な共有ルールは、パフォーマンスに顕著な影響を及ぼす可能性がある。 </div>

IV. まとめ

大量データ対策のまとめ



Salesforceでのデータ処理遅延対策

選択的SOQLなどを利用した検索性能の最適化

データ共有モデルを最適化する

Apex バッチ処理の最適化

レポート実行の最適化

Salesforce外部との大量データ対策

大量データの取得にはBULK APIを利用

UPSERT を INSERT/UPDATEへ分解する

親レコード毎のグルーピング

重要ではないプロセスの先送り

リソース



[大量のデータを使用するリリースのベストプラクティス](#)

[マルチテナンシーおよびメタデータの概要](#)

[レポートパフォーマンスの向上: ベストプラクティス](#)



[インデックス](#)

[非常に大きい SOQL クエリの処理](#)

[API からのデータ読み込み](#)



[共有モデルの仕組み](#)

[Apex を使用したレコードの共有](#)

[共有ルール適用の延期](#)



[Apex 一括処理の使用](#)

[Salesforce Functions](#)

[GRAX for Salesforce \(AppExchange\)](#)



[ディビジョンを使用してデータを整理する](#)

[スキニーテーブル](#)

[効率的なクエリの作成](#)

[Lightning ページのパフォーマンス分析](#)

[キャンバスの概要](#)

[Apex インテグレーションサービス](#)

[非同期 Apex](#)

オススの資格

LDV に関するポイントが試験範囲。Architect Journey Trailmix をご参考に

Salesforce 認定 Data アーキテクト

顧客のシナリオに沿って、大量データとソリューションのパフォーマンスを考慮して、拡張可能なデータモデルを設計する

顧客のシナリオに沿って、データストレージ管理の要件に最適なデータのアーカイブや削除の計画を推奨する



Architect Journey: Data Architecture

Salesforce 認定 Sharing and Visibility アーキテクト

特定の複雑な顧客組織の設定に従って、多数のユーザおよびレコードにおいてもメンテナンス可能なセキュリティモデルを設計する



Architect Journey: Sharing and Visibility



Thank You