

高性能なMySQL Database Serviceを 採用したシステム統合計画

株式会社りらく

会社概要



株式会社りらく

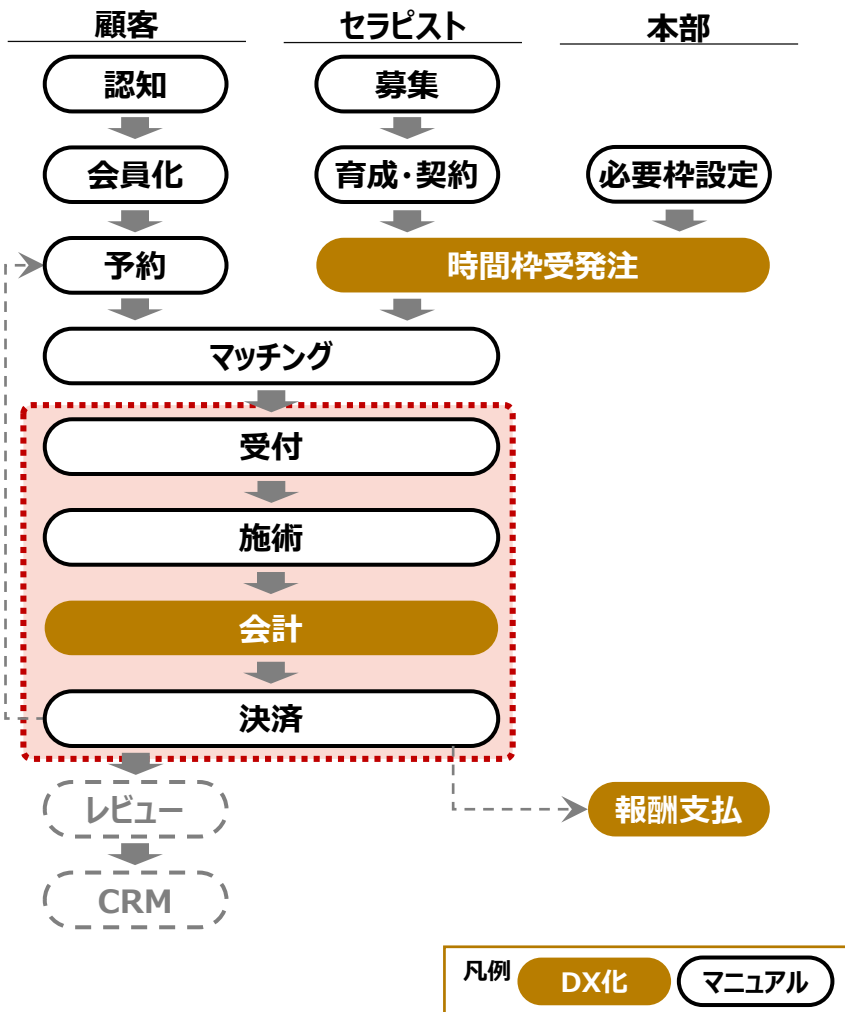


- 全国に「りらく」を615店舗*1を展開
- 年間総施術数500万施術超で業界1位、市場シェア20%
- 創業10年で600店舗にまで急拡大。

- 2009年に当時60分6,000円が主流であったリラクゼーション業界に60分2,980円(税抜)の低価格モデルで新規参入。現在は60分3,600円(税込、アプリ会員価格)で提供
- 2017年にスマホAPPを公開。現在のアプリ会員数は約190万人、直近アクティブユーザーは60万人。総施術に占めるアプリ会員比率は90%超*1。
- 2018年後半よりネット予約を順次導入、2020年6月に全国展開完了。現在のネット予約率は総施術の60%超*1。

「りらく」のDX化への取り組み

2018年1月時点では、ほぼ全てのプロセスがアナログ/マニュアル作業で運営されていた



認知



派手な看板+低価格による認知取得

→ 面前通行・交通量に依存
同価格帯の競合増加により看板効果は低減

予約



電話・ウォークインによる受付のみ

→ セラピストの空き時間を利用して対応
施術中顧客がいる場合への配慮不足
営業時間内でしか予約受付ができない

マッチング



ホワイトボードでアナログ予約管理

→ 本部での稼働状況のリアルタイム把握が困難
翌日以降の予約管理が困難

受付



対面での予約確認・受付

→ ホワイトボードの予約情報との突合
施術中顧客がいる場合への配慮不足

会計/決済

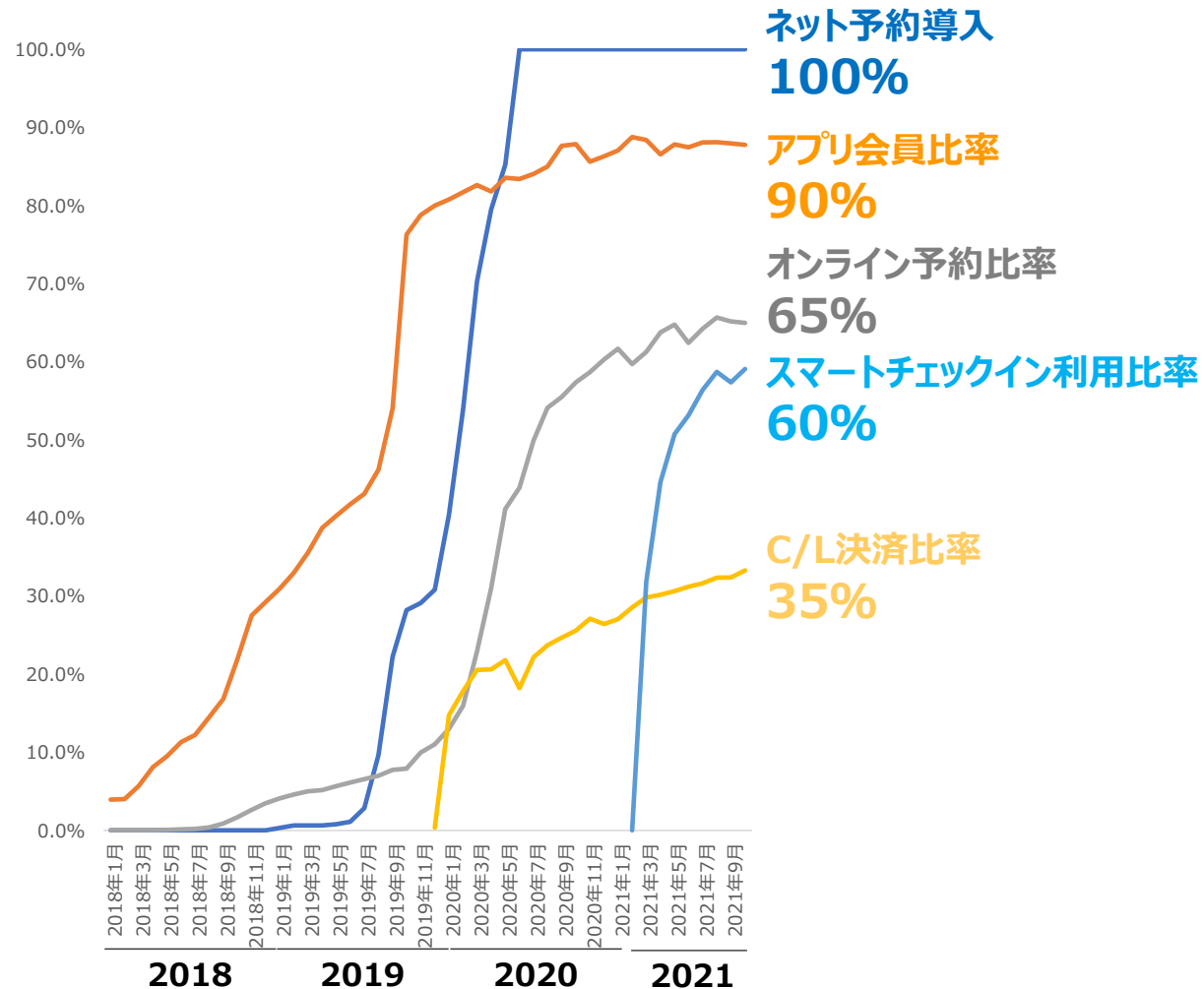
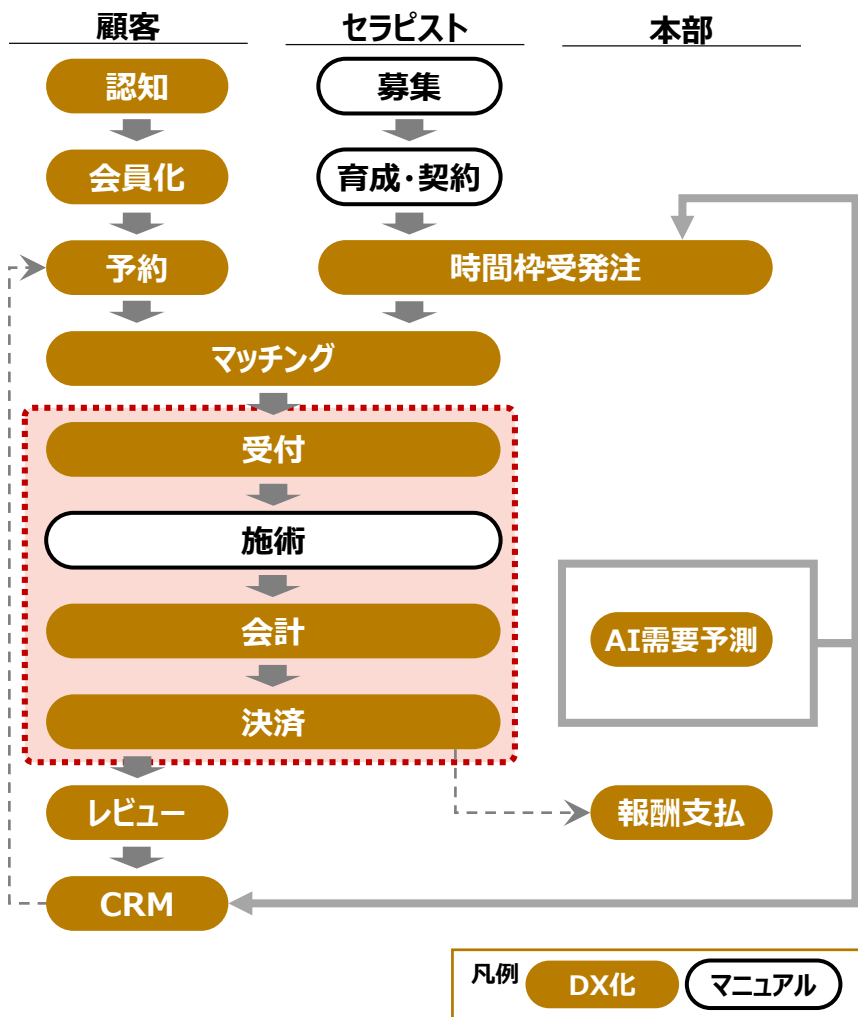


対面での会計・現金決済のみ

→ 店舗での現金・釣銭管理・売上送金業務

DX化3年間の現在地

2018年からの3年間で業務プロセスのほぼ全領域をシステム化。急激にDX化を進めてきた。



*1 2021年8月末現在

DX化を進めてきた中で発生している経営課題

1

システム障害の多発

- 矢継ぎ早にシステム導入を進めてきたため、全体構造が複雑化し、1カ所のシステム障害が他システムに波及し、**複数システムの同時停止が発生**。
- 復旧に向けた**原因分析に時間/費用が膨大にかかり**、システム復旧までに数日要することも多発。

2

システム開発期間の長期化/高コスト化

- システムが複雑に絡み合っているため、新たな機能を追加する場合に、影響範囲の調査に時間がかかり、**見積取得・開発開始までのリードタイムが長期化**
- 機能追加時の影響範囲が徐々に拡大し、単純な機能追加であっても周辺開発に費用/時間がかかり、**ビジネス上の機動性が失われていった**

3

経営分析に資するデータ抽出の難易度向上

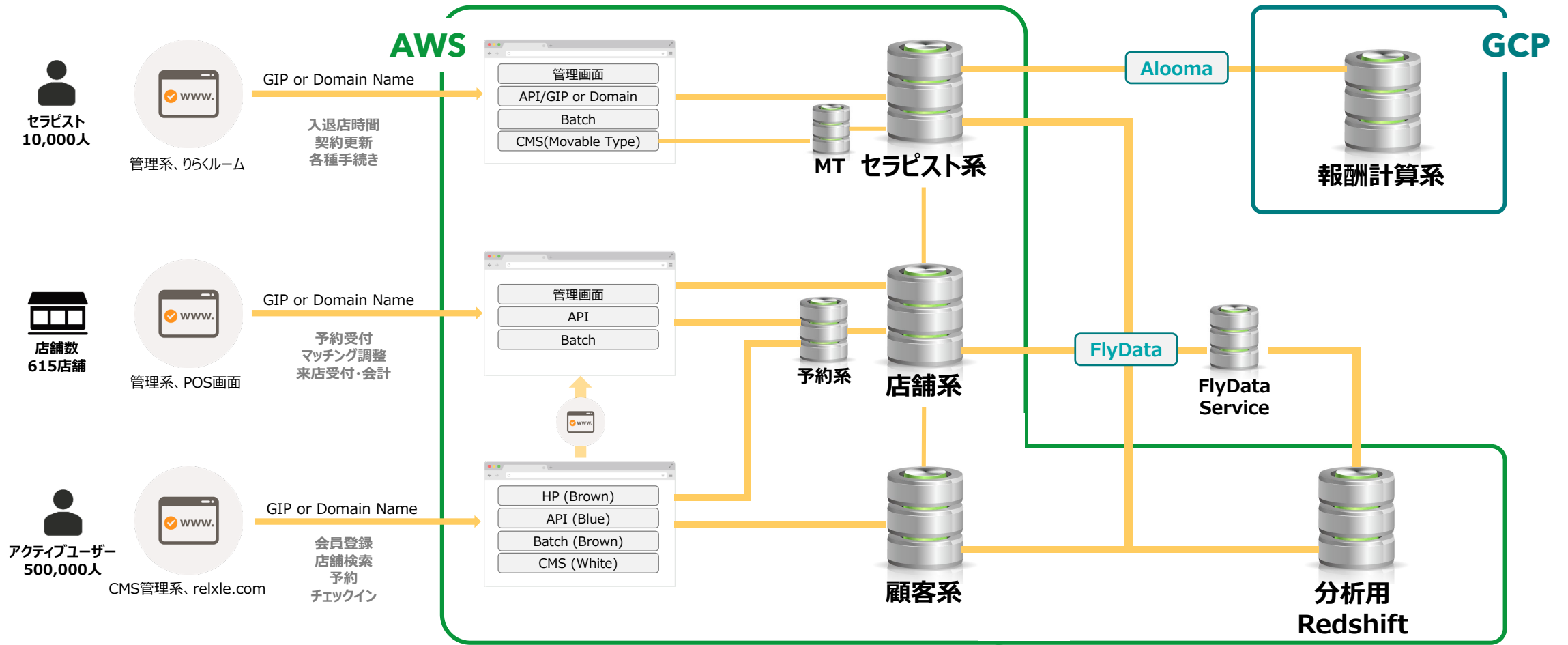
- 複数のシステムに類似のデータが蓄積され、抽出条件によっては複数システムを参照しながら追加/排除を繰り返す必要があり、抽出作業により結果が異なることが頻発。**何が正しい情報なのかわからない**ケースが発生。



本日のアジェンダ

1. 現状の課題
2. 課題を解決するための取り組み
3. Oracle Cloud Infrastructure(OCI)採用理由
4. OCI移行を担当したエンジニアの声
5. MDS移行時の考慮事項
6. OCIに今後期待すること

1. 現状の課題



1. 現状の課題

① 複数のサブシステムが乱立し、それぞれが独自に DB を保持

データが一元管理されておらず、複数のデータ連携バッチを使って、サブシステム間でデータを同期している

セラピスト
10,000人

管理系、予約システム

GIP or Domain Name

入退店時間
契約更新
各種手続き

管理画面

API/GIP or Domain

Batch

CMS(Movable Type)

MT

セラピスト系

② OLTP用DBとOLAP用DBが分離されておりリアルタイム性が低い

複数のサブシステムのDBからETLツール(FlyData)を使ってRedshiftにデータ連携している為、少し切り口を変えて分析をしたい場合にRedshift内に必要なデータが蓄積できていないケースが多い

店舗数
500,000人

管理系、POS画面

GIP or Domain Name

予約受付
マッチング調整
来店受付・会計

管理画面

Batch

予約系

店舗系

③ システムの拡張性が低い

データがサブシステム間で分断されているため、外部システムへのデータ提供や、外部システムで生成されたデータの格納が困難

500,000人

CMS管理系、relxle.com

予約
チェックイン

管理画面

API (Blue)

Batch (Brown)

CMS (White)

顧客系

Alooma

GCP

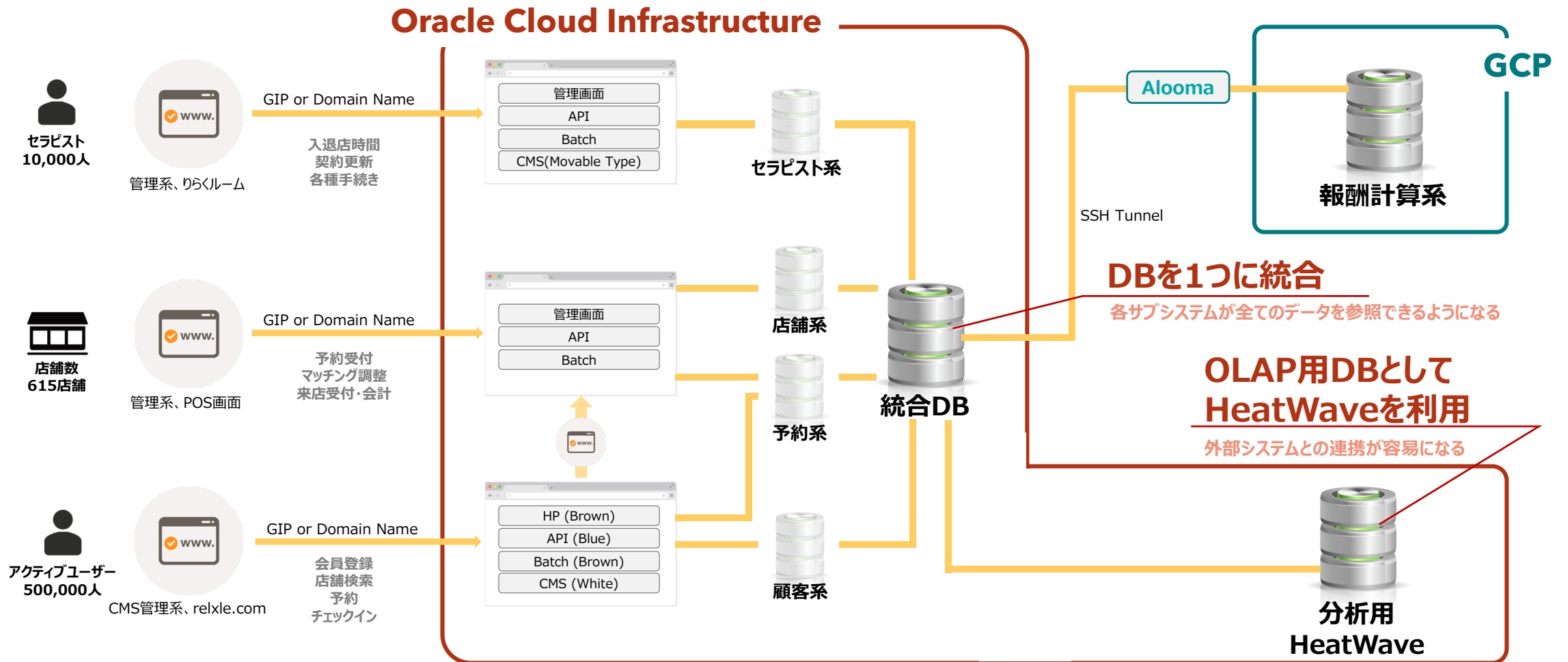
報酬計算系

FlyData

FlyData
Service

分析用
Redshift

2. 課題を解決するための取り組み



2. 課題を解決するための取り組み

① 複数のサブシステムが乱立し、それぞれが独自に DB を保持

データが一元管理されておらず、複数のデータ連携バッチを使って、サブシステム間でデータを同期している

② OLTP用DBとOLAP用DBが分離されておりリアルタイム性が低い

複数のサブシステムのDBからETLツール(FlyData)を使ってRedshiftにデータ連携している為、少し切り口を変えて分析をしたい場合にRedshift内に必要なデータが蓄積できていないケースが多い

③ システムの拡張性が低い

データがサブシステム間で分断されているため、外部システムへのデータ提供や、外部システムで生成されたデータの格納が困難

OCI上に統合DBを構築することでデータを一元管理

将来的にHeatWaveを活用することでETL不要で連携







DBを統合することで外部システムとのデータ連携が容易に

HeatWave

3. OCI採用の理由

圧倒的なコストパフォーマンス！

利用頻度の高いサービス3つ全てが低価格 VS MDS

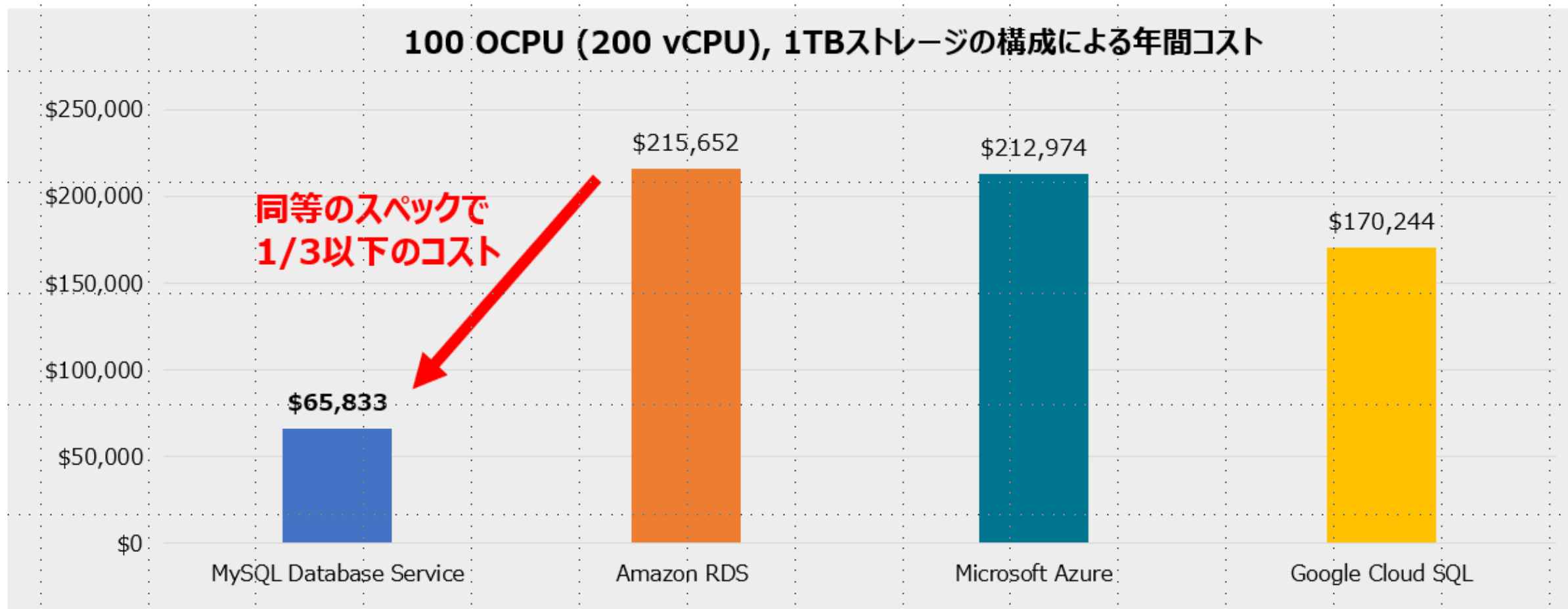
	Oracle の強み	Oracle	他社クラウド	
Compute 	<ul style="list-style-type: none">✓ 同一リソースを低価格で提供✓ より高性能なリソースも提供可能	¥61.25/時 Compute (VM.Standard2.8; 16vCPU, 120GB, Linux)	¥119.94/時 仮想マシン (16vCPU, 64GB, Linux)	 49% 低価格
Storage 	<ul style="list-style-type: none">✓ 他社標準ストレージ価格で、高性能ストレージを提供✓ IOPS設定 + SLA	¥5,100/月 Block Volume (1TB, 25K IOPS)	¥239,040/月 ブロック・ストレージ (1TB, 25K IOPS)	 97% 低価格
Network 	<ul style="list-style-type: none">✓ AD間無償✓ 10TB/月まで無償✓ 閉域網接続時はデータ転送無償	¥18,972/月 FastConnect (1Gbps, 100TB) *閉域網接続	¥517,445/月 接続サービス (1Gbps, 100TB) *閉域網接続	 96% 低価格

Compute: 同一リソースを低価格で提供(メモリは約2倍)

Storage/Network: トランザクションやデータ転送量による料金変動を低減する料金体系

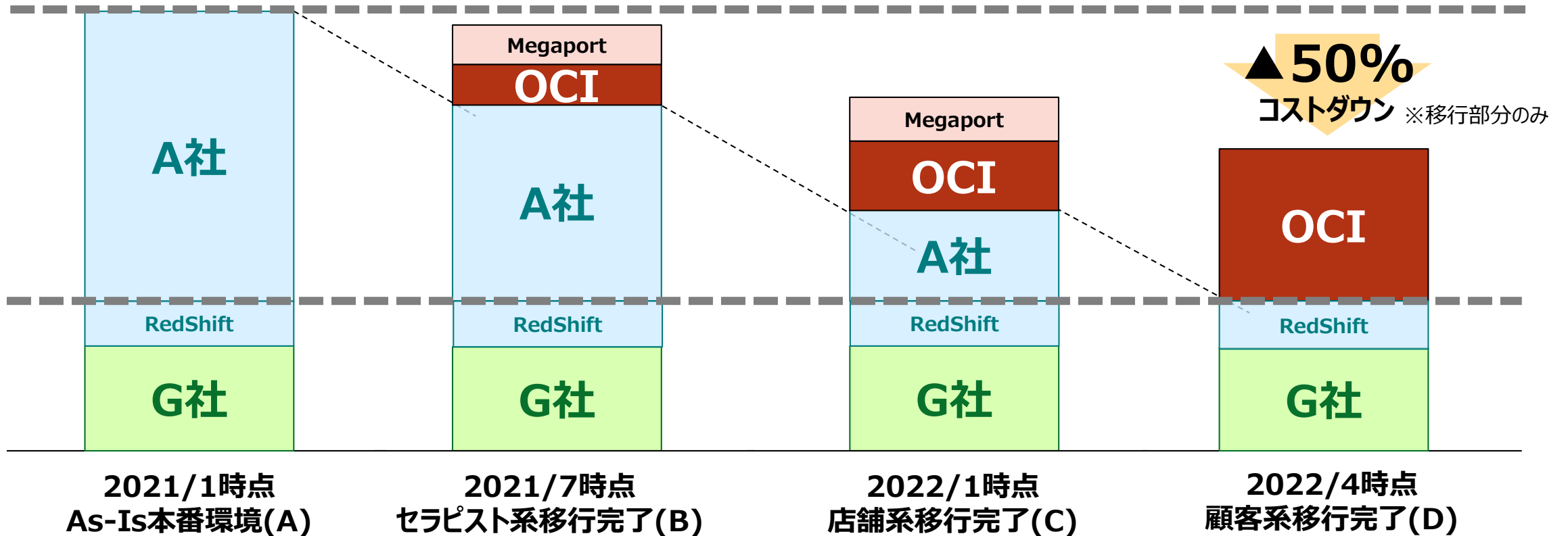
3. OCI採用の理由

MySQLのマネージドサービスも安い！



3. OCI採用の理由

- OCIへの移行後のコスト試算を実施した結果、50%のコストダウンが可能
- 現行のRedshiftをHeatWaveに置き換えることで、更なるコストダウンが可能



3. OCI採用の理由

MySQL Database Service(MDS) は高性能！
MySQL開発ベンダーであるオラクル社からのサポートも受けられる！



高性能

MDSでは高パフォーマンスなブロックストレージを標準採用
プロビジョンド IOPS 不要
(ブロックストレージのIOPS : 75 IOPS/GB)

MySQL開発 ベンダーが提供

オラクルのMySQLチームが100%開発、運用、サポート
MySQL部分についてもコアなサポートを受けられる
(チューニングに関する問合せもサポート対象範囲)

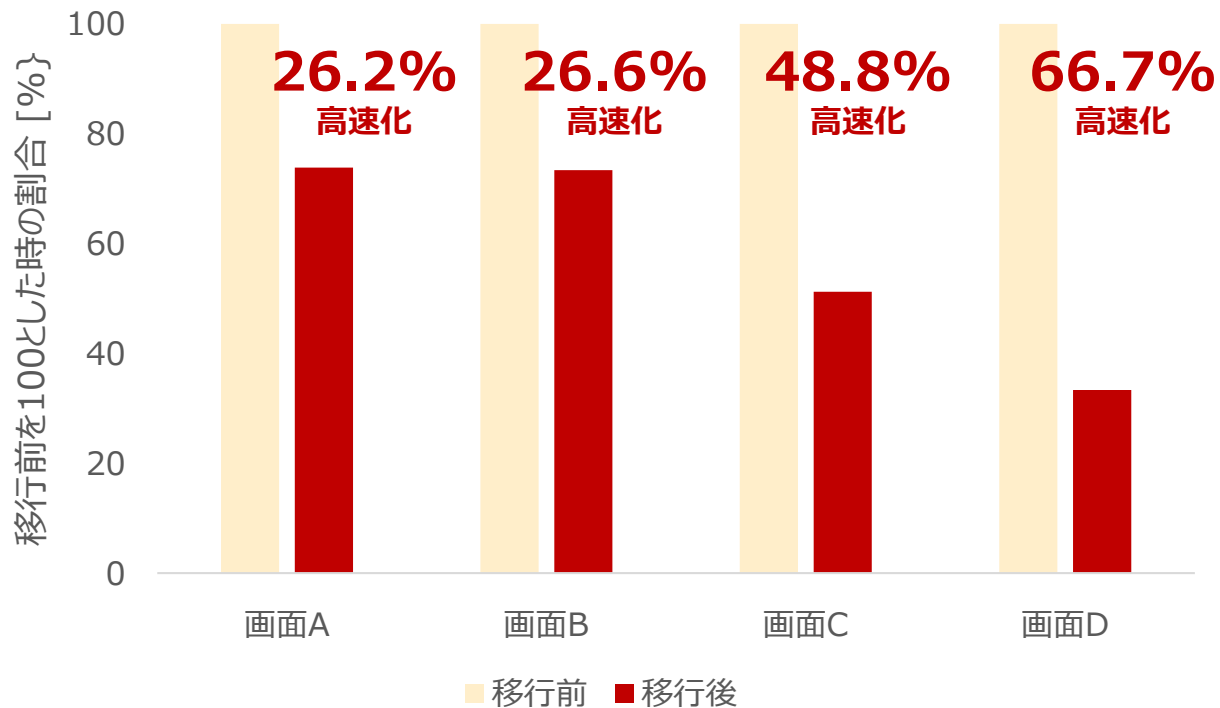
MySQL最新 バージョンを 使用可能

MDSではMySQL 8.0を利用可能
(MySQL 5.7は2023年10月でサポート終了
MySQL 8.0ベースのAmazon Auroraは未リリース)

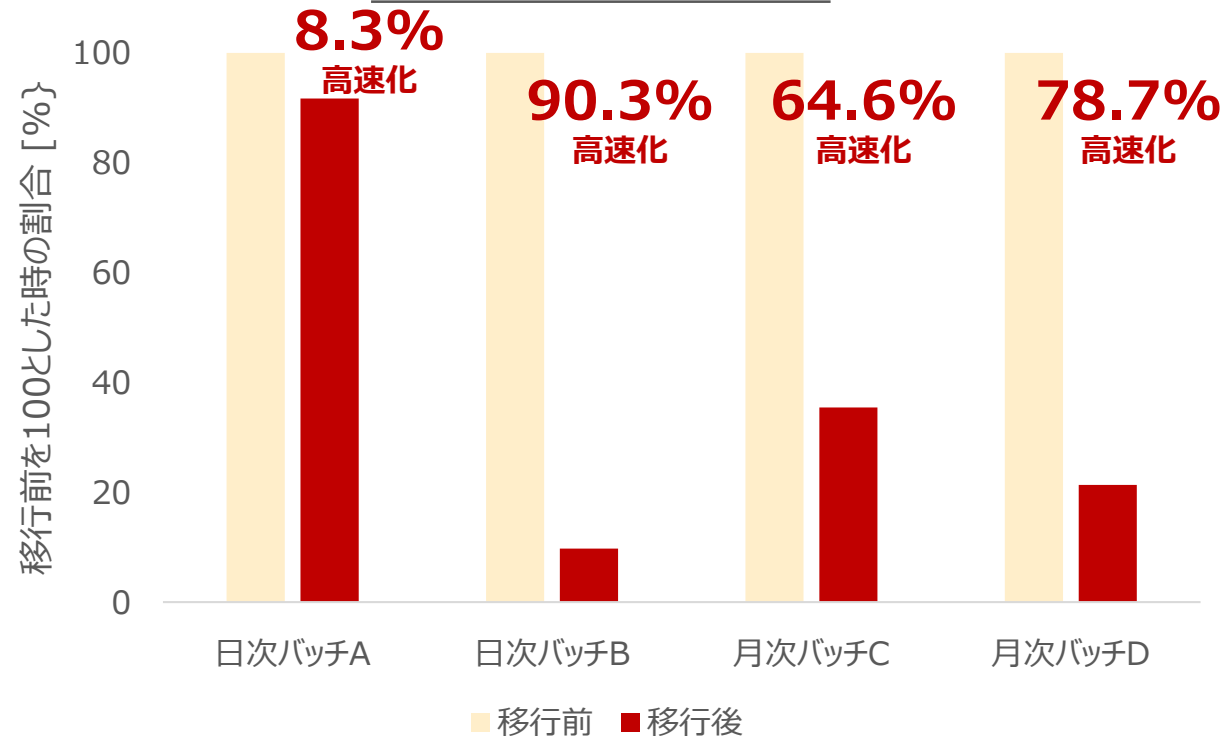
3. OCI採用の理由

OCI、MDSへの移行により、日中の処理もバッチ処理もパフォーマンス改善！！

TP系システムの画面表示時間



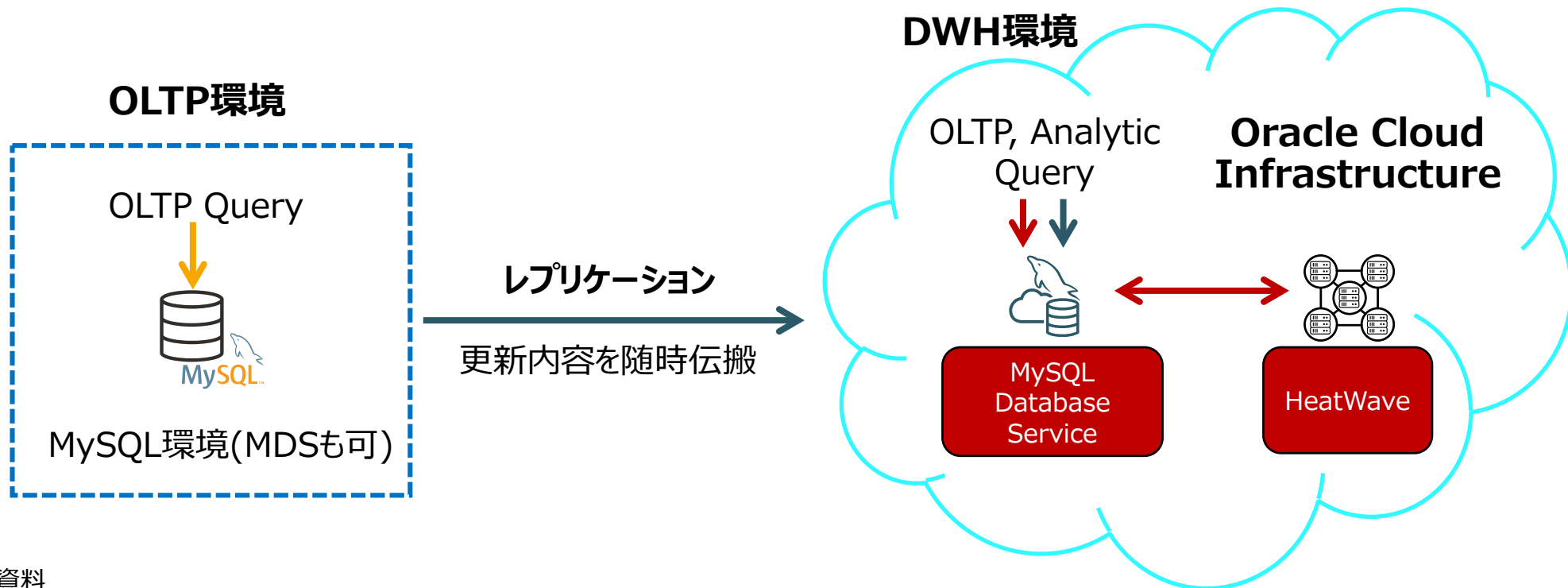
バッチ処理の処理時間



3. OCI採用の理由

- HeatWaveにより、MySQL上のデータをそのまま分析対象にできるため、FlyDataを使って複数のサブシステムからデータ連携する必要が無い
- ETLツール不要でシンプルなシステム構成のまま、最新データも分析が可能

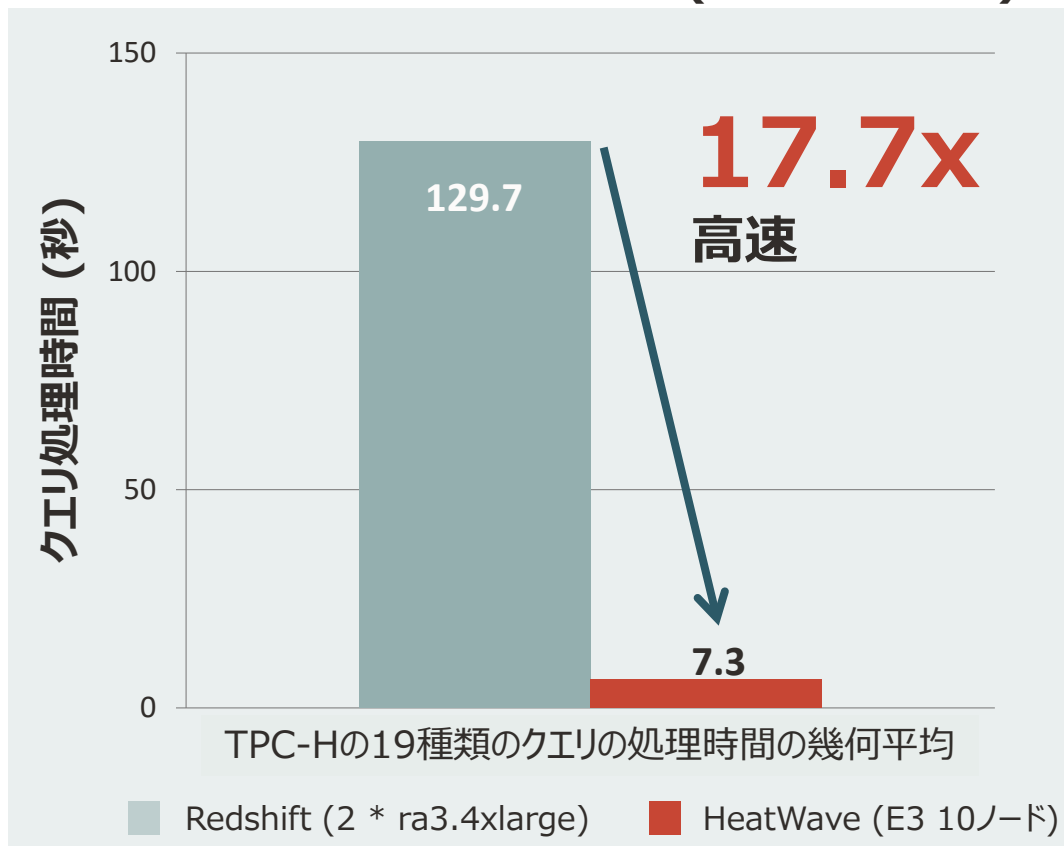
「Redshiftにデータはたくさん入っているが、分析に本当に必要なデータが入っていない」という問題を回避



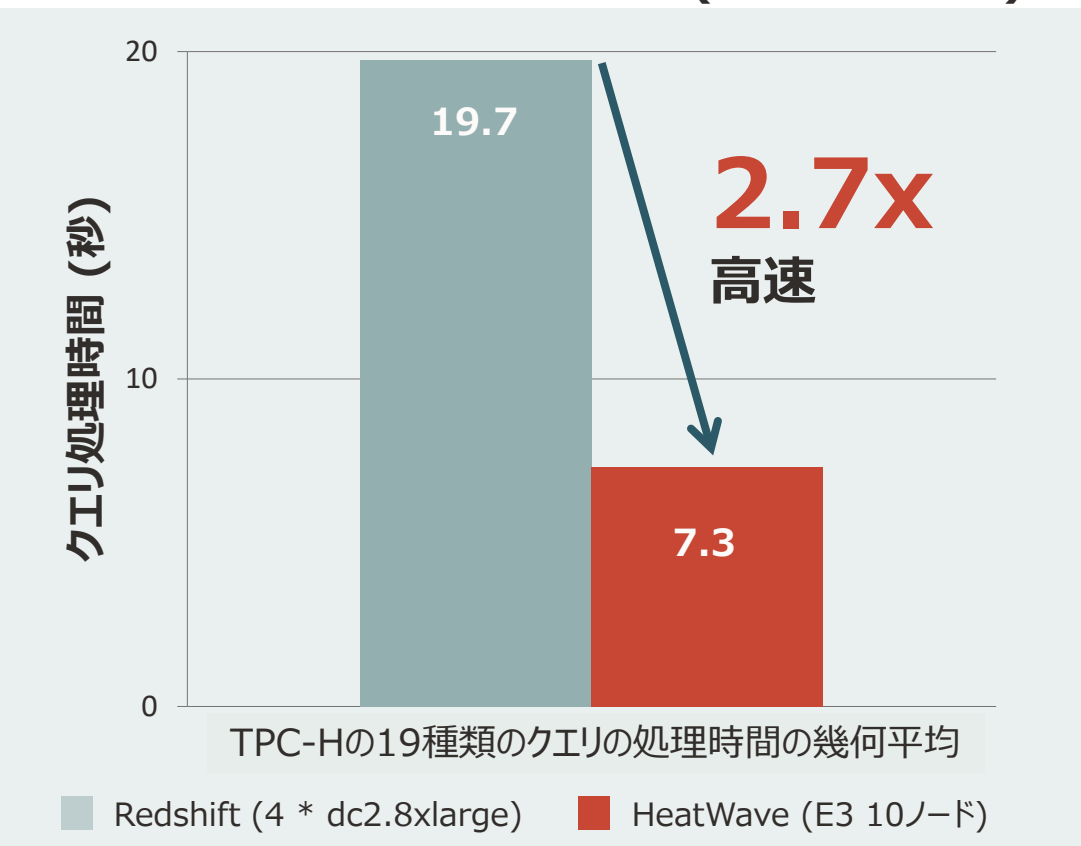
3. OCI採用の理由

HeatWaveの圧倒的な性能！

Redshiftの最安スペックとの比較(TPC-H、4TB)



Redshiftの最速スペックとの比較(TPC-H、4TB)



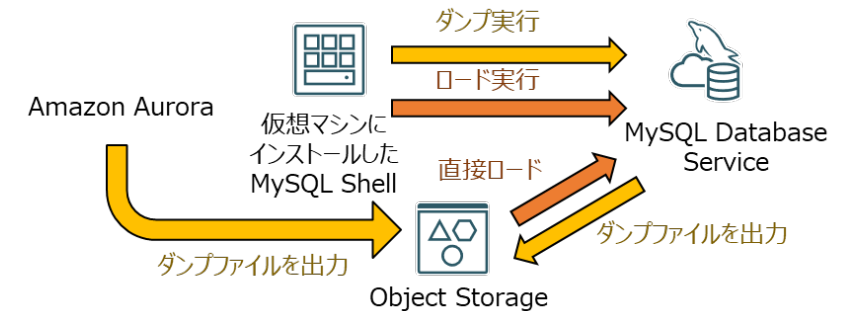
*Benchmark queries are derived from TPC-H benchmark, but results are not comparable to published TPC-H benchmark results since they do not comply with TPC-H specification

4. OCI移行を担当したエンジニアの声

1

MySQL ShellとOCI Object Storageの親和性が高く、作業がしやすい

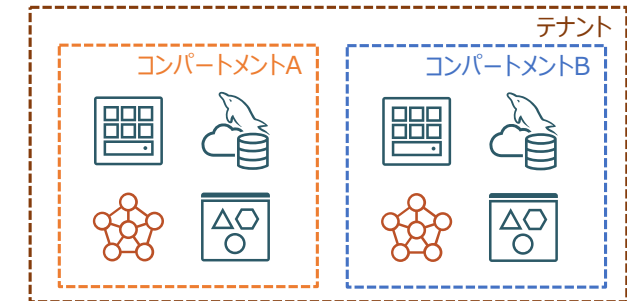
- MySQL DBのダンプをObject Storageに出力したり、Object Storage上のデータを直接ロード可能
- ローカルストレージの空き容量を気にせずダンプできるのは利便性が高い



2

コンパートメントを使うことで、シンプルな権限管理を実現でき、インフラ管理コストが下がった

- AWSには存在しない概念
- コンパートメントでは1テナント(契約)内に区画を区切ってリソース管理が可能
- 1アカウント内でコンパートメントを切り替えてアクセスするリソースを変更可能



3

強力な無償枠*₁があり、開発環境以外にもエンジニアが気軽に検証できる環境が確保できた

- 2021年5月にArmベースの仮想マシンも無償枠に追加され、4 OCPU (8vcpu)、メモリ24GBの仮想マシンも無償枠で利用できる

*1 Oracle Cloud Free Tier (<https://www.oracle.com/jp/cloud/free/>)

5. MDS移行時の考慮事項

1

MySQL 8.0へのバージョンアップが必要

- MDS固有の考慮点は特に無いが、事前にデータ移行やアプリケーションの確認等が必要であった
- バージョンアップの検証中に、MySQL部分の疑問点についてもサポートが受けられるのは良かった

2

MDSをHA構成で利用する時には、各テーブルに主キーが必須である

- 一部のフレームワークでは、フレームワークが内部的に主キー無しのテーブルを自動的に生成するので要注意
- MySQL 8.0.23から使用できるINVISIBLE COLUMNを使うことで大きな変更なく回避可能

3

リードレプリカ相当の機能がまだ無い

- MDSのバックアップをリストアして、MDS同士でレプリケーション構成を組むことで、参照用DBの追加は可能
- ただし、参照用のエンドポイントを制御する機能が無いため、現在はHAProxyを利用して制御している

4

ETLツールの制限事項

- いくつかのシステム変数を変更する必要があったが、MDSでは変更できないシステム変数が存在
- MySQL 8.0では既に非推奨になっているシステム変数も変更する必要があった

※MDSから仮想サーバー上のMySQLにレプリケーションを構成し、仮想サーバー上のMySQLに対してETLツールを接続することでこの問題を回避。将来的にHeatWaveを採用すれば、このあたりの運用管理コストも削減できる見込

6. OCIに今後期待すること : MDS、OCIの機能拡充

MDS

HA構成のMDSにおける
参照用エンドポイントの公開

AWS Auroraの読み込みエンドポイント、
カスタムエンドポイントのような機能の提供

シェイプ(スペック)の変更、ストレージ
サイズの拡張などをMDSの再起動だけで
適用できるようにしてほしい

- 単一のMDSだけで参照処理の負荷分散ができる方が利便性が高い
- 現時点では、次のPOSシステム統合時に参照用のMDSを追加予定
- 現状はHAProxyを使って参照用のエンドポイントを制御しているが、こういう機能もマネージドサービスとして提供して欲しい
- 現在の仕様では、MDSのバックアップを取得し、リストア時にこれらの点を変更してリストアしないとイケない

OCI 全般

メトリクスや監視周りの機能拡充

マネージドのキャッシュサービスの追加

- AWSのCloud Watchに比べると、OCIのメトリクスや監視周りがまだまだ弱いと感じている
- OCI Logging や Logging Analytics は興味深いけど、まだ使えていない
- Amazon ElastiCacheを使っているアプリケーションの移行の為に、Redisクラスタを独自にOKE(Oracle Container Engine for Kubernetes)上に立ち上げる必要があった

7. まとめ

1 OCIを採用することでA社よりもインフラコストが削減可能

2

MDSでは高スペックなシェイプも比較的安価に利用可能

- 追加料金不要でMySQL部分についてもコアなサポートが受けられる
- 最新バージョンのMySQL 8.0が使用可能
- AWS Auroraと比べて機能不足な点もあるが、活発に開発されており、どんどん機能強化されている

3

DB統合後は、HeatWave採用により
更なるパフォーマンス向上、インフラ管理コストの
削減も期待できる