

AWSでのバッチ処理ノウハウ

～バッチパフォーマンスの出し方～

ICタイムリコーダー事業部

システム・アーキテクト 渡邊一夫

アジェンダ

- AWSでのバッチ処理実行環境
- バッチ処理の流れ
- パフォーマンス低下の原因調査
- パフォーマンス改善のまとめ

AWSでのバッチ処理実行環境



Amazon EC2

- Amazon EC2の上に実行環境を構築して実行
 - AWSが提供する仮想マシン環境
 - 様々なランタイム環境をセットアップ可能 → 自由度は高い
 - セットアップに手間がかかる
 - 定時実行はEC2内で実現

AWSでのバッチ処理実行環境



AWS Lambda

- AWS Lambdaで実行
 - AWSが提供するイベント駆動型の仮想マシン環境
 - セットアップ済みランタイムを使用
 - 実行時間に制限がある(最大 15分)
 - 定時実行はAWSサービス(Amazon CloudWatch Events)を利用

AWSでのバッチ処理実行環境



AWS Batch

- AWS Batchで実行
 - AWSが提供するバッチ実行用の仮想マシン環境
 - セットアップ済みランタイムを使用
 - 実行時間の制限はない
 - 定時実行はAWSサービス(Amazon CloudWatch Events)を利用

AWSでのバッチ処理実行環境

セットアップの自由度

セットアップの手間

実行時間の制限

定時実行の手間



↑ 高

↓ 多い

↑ ない

↓ 多い

Amazon EC2



↓ 低

↑ 少ない

↓ ある

↑ 少ない

AWS Lambda



↓ 低

↑ 少ない

↑ ない

↑ 少ない

AWS Batch

AWSでのバッチ処理実行環境

セットアップの自由度

セットアップの手間

実行時間の制限

定時実行の手間



↑ 高

↓ 多い

↑ ない

↓ 多い

Amazon EC2



↓ 低

↑ 少ない

↓ ある

↑ 少ない

AWS Lambda



↓ 低

↑ 少ない

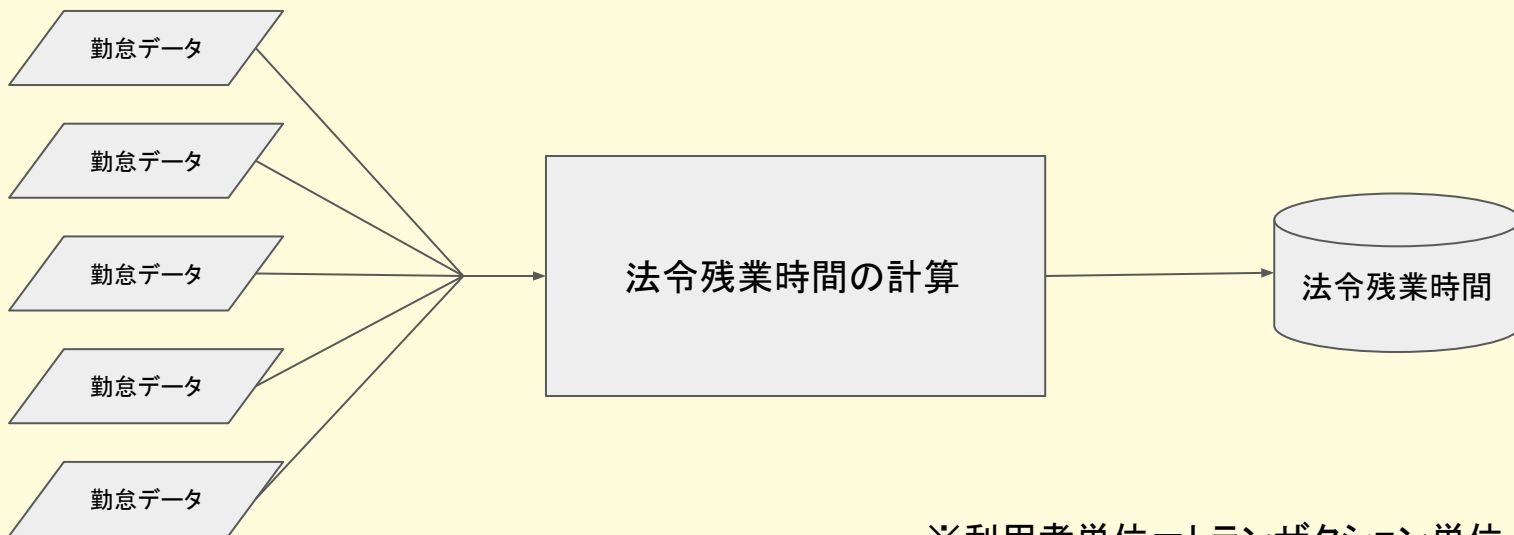
↑ ない

↑ 少ない

AWS Batch

バッチ処理の流れ

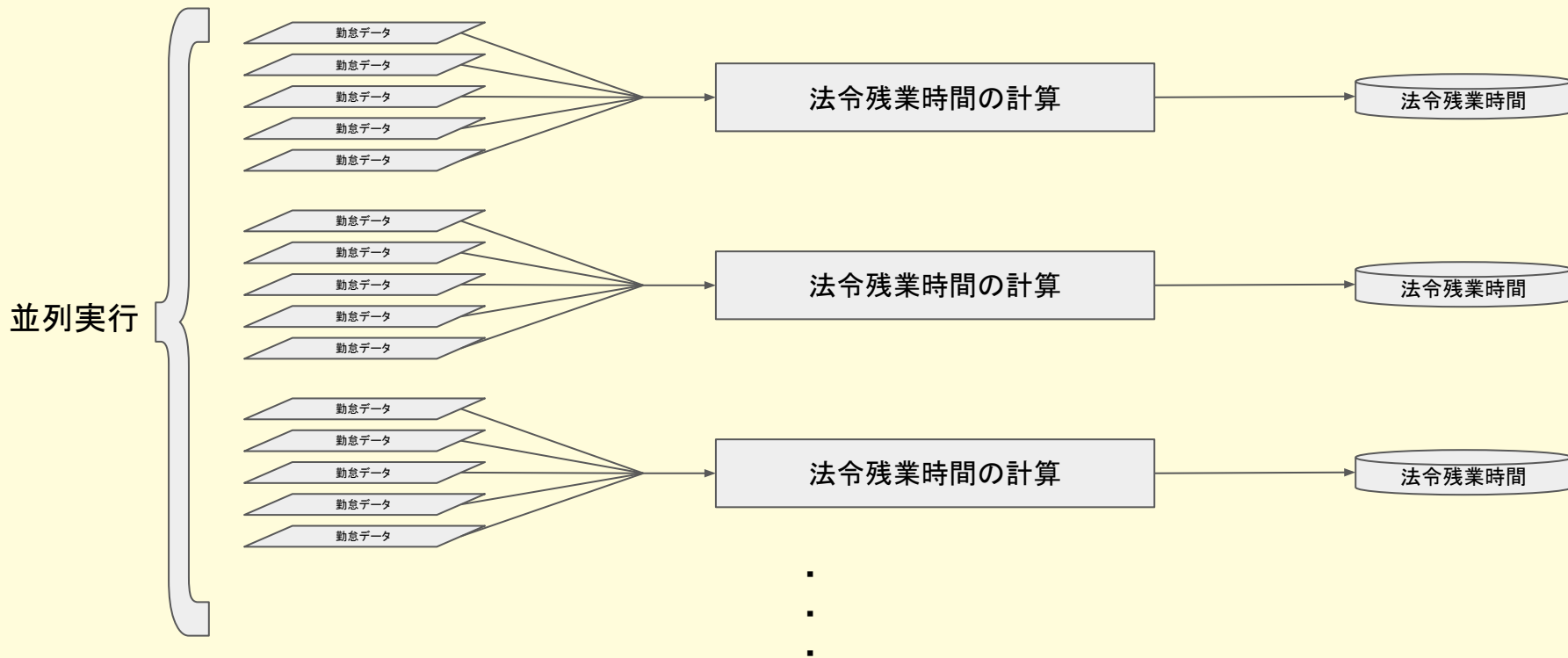
- 利用者単位に日々の勤怠データを読み込む
- 勤怠データを基に法令残業時間の算出をする
- 算出した法令残業時間をデータベース保存する



※利用者単位＝トランザクション単位

バッチ処理の流れ

- トランザクション単位に並列で処理する



バッチ処理の流れ

- 実行結果

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU	2 GB	t2.small
Amazon ECS	1 vCPU	2 GB	t2.small
Amazon RDS	2 vCPU	4 GB	db.t3.medium

実行時間 →

バッチ処理の流れ

- 実行結果

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU	2 GB	t2.small
Amazon ECS	1 vCPU	2 GB	t2.small
Amazon RDS	2 vCPU	4 GB	db.t3.medium

実行時間 → 20時間！

遅くても夜に実行して朝には終わる必要がある。

バッチ処理の流れ

- 実行結果

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU	2 GB	t2.small
Amazon ECS	1 vCPU	2 GB	t2.small
Amazon RDS	2 vCPU	4 GB	db.t3.medium

実行時間 → 20時間！

遅くても夜に実行して朝には終わる必要がある。



Point: 許容できる目標時間を設定する

パフォーマンス低下の原因調査

- CPUが足りない？
- メモリが足りない？
- ネットワーク帯域が足りない？

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU	2 GB	t2.small
Amazon ECS	1 vCPU	2 GB	t2.small
Amazon RDS	2 vCPU	4 GB	db.t3.medium

パフォーマンス低下の原因調査

- CPUが足りない？
- メモリが足りない？
- ネットワーク帯域が足りない？



Point: ボトルネックがどこなのか調査する

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU	2 GB	t2.small
Amazon ECS	1 vCPU	2 GB	t2.small
Amazon RDS	2 vCPU	4 GB	db.t3.medium

パフォーマンス低下の原因調査

- CPUが足りない？
- メモリが足りない？
- ネットワーク帯域が足りない？

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU	2 GB	t2.small
Amazon ECS	1 vCPU	2 GB	t2.small
Amazon RDS	2 vCPU	4 GB	db.t3.medium

パフォーマンス低下の原因調査

- CPUが足りない？
- メモリが足りない？
- ネットワーク帯域が足りない？

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU	2 GB	t2.small
Amazon ECS	1 vCPU	2 GB	t2.small
Amazon RDS	2 vCPU	4 GB	db.t3.medium

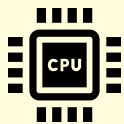


Point: クラウドのメリット(様々なインスタンスタイプ)を活用する

パフォーマンス低下の原因調査

- インスタンスタイプごとのCPU・メモリー


構成	インスタンスタイプ	CPU	メモリー
AWB Batch	t2.small	1 vCPU	2 GB
Amazon ECS	t2.small	1 vCPU	2 GB
Amazon RDS	db.t3.medium	2 vCPU	4 GB



パフォーマンス低下の原因調査

- インスタンスタイプごとのCPU・メモリー

構成	インスタンスタイプ	CPU	メモリー
AWB Batch	t2.small → m4.2xlarge	1 vCPU → 8 vCPU	2 GB → 32 GB
Amazon ECS	t2.small → t2.xlarge	1 vCPU → 4 vCPU	2 GB → 16 GB
Amazon RDS	db.t3.medium → db.r5.4xlarge	2 vCPU → 16 vCPU	4 GB → 128 GB




- データI/Oと検索を頻繁に行うAmazon RDSのスペックを上げると顕著
- バッチ処理が動作するAWS Batchのスペックを上げると顕著 → 並列化の効果が大きい
- ただし頭打ちがあるので計測することは大事

パフォーマンス低下の原因調査

- インスタンスタイプごとのCPU・メモリー

構成	インスタンスタイプ	CPU	メモリー
AWB Batch	t2.small → m4.2xlarge	1 vCPU → 8 vCPU	2 GB → 32 GB
Amazon ECS	t2.small → t2.xlarge	1 vCPU → 4 vCPU	2 GB → 16 GB
Amazon RDS	db.t3.medium → db.r5.4xlarge	2 vCPU → 16 vCPU	4 GB → 128 GB

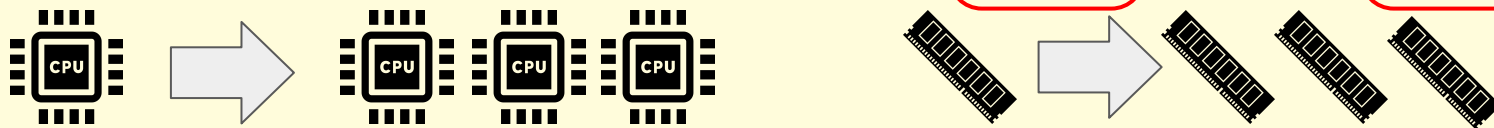


- データI/Oと検索を頻繁に行うAmazon RDSのスペックを上げると顕著
- バッチ処理が動作するAWS Batchのスペックを上げると顕著 → 並列化の効果が大きい
- ただし頭打ちがあるので計測することは大事

パフォーマンス低下の原因調査

- インスタンスタイプごとのCPU・メモリー

構成	インスタンスタイプ	CPU	メモリー
AWB Batch	t2.small → m4.2xlarge	1 vCPU → 8 vCPU	2 GB → 32 GB
Amazon ECS	t2.small → t2.xlarge	1 vCPU → 4 vCPU	2 GB → 16 GB
Amazon RDS	db.t3.medium → db.r5.4xlarge	2 vCPU → 16 vCPU	4 GB → 128 GB




- データI/Oと検索を頻繁に行うAmazon RDSのスペックを上げると顕著
- バッチ処理が動作するAWS Batchのスペックを上げると顕著 → 並列化の効果が大きい
- ただし頭打ちがあるので計測することは大事

Point: イマドキの処理はマルチコアを意識して並列実行前提の設計をする

パフォーマンス低下の原因調査

- インスタンスタイプごとのCPU・メモリー

構成	インスタンスタイプ	CPU	メモリー
AWB Batch	t2.small → m4.2xlarge	1 vCPU → 8 vCPU	2 GB → 32 GB
Amazon ECS	t2.small → t2.xlarge	1 vCPU → 4 vCPU	2 GB → 16 GB
Amazon RDS	db.t3.medium → db.r5.4xlarge	2 vCPU → 16 vCPU	4 GB → 128 GB



- データI/Oと検索を頻繁に行うAmazon RDSのスペックを上げると顕著
- バッチ処理が動作するAWS Batchのスペックを上げると顕著 → 並列化の効果が大きい
- ただし頭打ちがあるので計測することは大事

パフォーマンス低下の原因調査

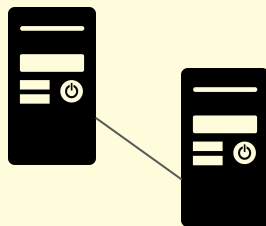
- CPUが足りない？
- メモリが足りない？
- ネットワーク帯域が足りない？

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU	2 GB	t2.small
Amazon ECS	1 vCPU	2 GB	t2.small
Amazon RDS	2 vCPU	4 GB	db.t3.medium

パフォーマンス低下の原因調査

- インスタンスタイプごとのネットワーク帯域

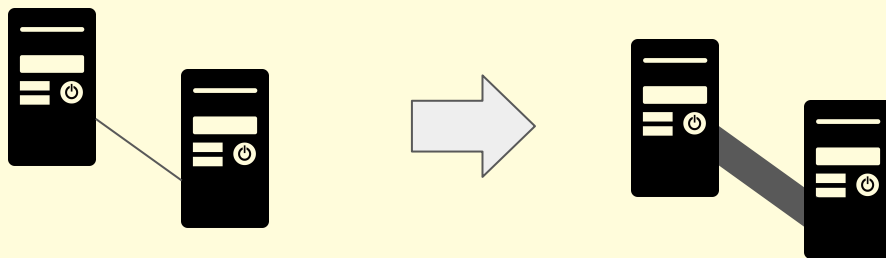
構成	インスタンスタイプ	ネットワーク帯域
AWB Batch	t2.small	低～中
Amazon ECS	t2.small	低～中
Amazon RDS	db.t3.medium	最大 5 Gbps



パフォーマンス低下の原因調査

- インスタンスタイプごとのネットワーク帯域

構成	インスタンスタイプ	ネットワーク帯域
AWB Batch	t2.small → m4.2xlarge	低～中 → 高
Amazon ECS	t2.small → t2.xlarge	低～中 → 高
Amazon RDS	db.t3.medium → db.r5.4xlarge	最大 5 Gbps → 最大 10 Gbps

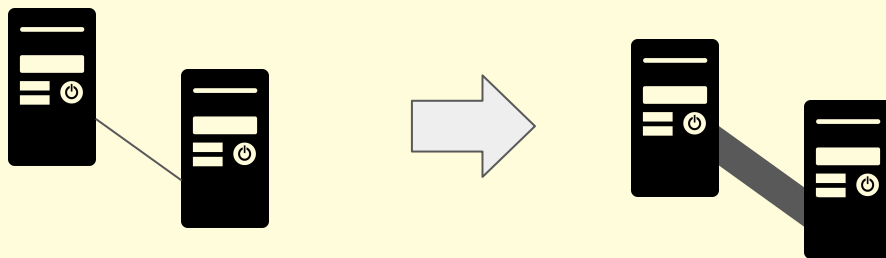


- ネットワークにうまくデータを流せる帯域のインスタンスタイプを選ぶことが重要

パフォーマンス低下の原因調査

- インスタンスタイプごとのネットワーク帯域

構成	インスタンスタイプ	ネットワーク帯域
AWB Batch	t2.small → m4.2xlarge	低～中 → 高
Amazon ECS	t2.small → t2.xlarge	低～中 → 高
Amazon RDS	db.t3.medium → db.r5.4xlarge	最大 5 Gbps → 最大 10 Gbps



- ネットワークにうまくデータを流せる帯域のインスタンスタイプを選ぶことが重要

パフォーマンス低下の原因調査

- 調査しチューニング実施

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU → 8 vCPU	2 GB → 32 GB	t2.small → m4.2xlarge
Amazon ECS	1 vCPU → 4 vCPU	2 GB → 16 GB	t2.small → t2.xlarge
Amazon RDS	2 vCPU → 16 vCPU	4 GB → 128 GB	db.t3.medium → db.r5.4xlarge

実行時間 →

パフォーマンス低下の原因調査

- 調査しチューニング実施

構成	vCPU	メモリー	インスタンスタイプ
AWB Batch	1 vCPU → 8 vCPU	2 GB → 32 GB	t2.small → m4.2xlarge
Amazon ECS	1 vCPU → 4 vCPU	2 GB → 16 GB	t2.small → t2.xlarge
Amazon RDS	2 vCPU → 16 vCPU	4 GB → 128 GB	db.t3.medium → db.r5.4xlarge

実行時間 → 8時間！

夜11時に実行して朝7時に終了・・・ギリギリ許容範囲。

パフォーマンス改善のまとめ

- 許容できる目標時間を決める
- ボトルネックがどこなのかを調査する
- クラウドのメリット(様々なインスタンスタイプ)を活用する
- イマドキの処理はマルチコアを意識して並列実行前提の設計をする

～ おまけ ～

- ログを出力しすぎてコストが上がったので、必要な情報に留めるのが重要

Thank you for watching.

質疑応答