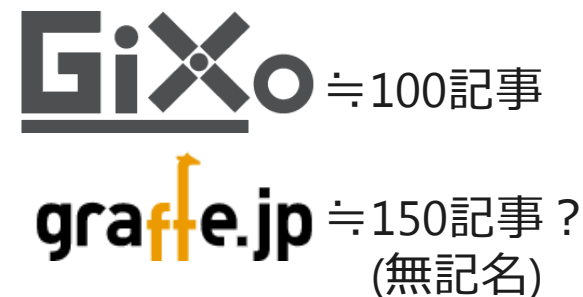


ベンチャー企業が取組んだ
ちょっと変わった
Redshift運用方法

2017.10.11
株式会社ギックス

名前	穴戸 栄一郎 (シシド エイチロウ)
所属	株式会社ギックス (2014年～) "Data Structuring"Sectionリーダー
担当業務	<ul style="list-style-type: none">• ETL処理などによる分析データ準備• 社内インフラ整備• 社内ツール開発• ブログ更新
連絡先	eiichiro.shishido@gixo.jp https://www.facebook.com/eiichiro.shishido
前職	中小SIerのプログラマー 主に保険システムのDB周りの設計・開発



社名	株式会社ギックス GiXo Ltd.
所在地	東京都 港区 三田 1-4-28 三田国際ビル 2F
設立	2012年12月12日
事業内容	<ul style="list-style-type: none">• データを活用した各種コンサルティング業務• データ分析に関わるツールの研究・開発• 上記ツールを用いた各種サービスの提供 (分析システム構築・レポート配信等)
WEB	https://www.gixo.jp/

$$P \{ D \mid C, A \} = ?$$

$$P^{\sim} D (C, A)^{\infty} = \times$$

- Redshift利用料金をケチる
- ゾンビクエリー監視
- なんちゃってAUTO VACUUM
- チームによるデータ分析体制

Redshift利用料金はオンプレDBに比べて安い、ベンチャー企業には大きな出費。

Dense Storage : ds2.xlarge × 2 node 運用を想定した場合

オンデマンド料金

$$\$1.190 \times 2 \text{ node} \times 24 \text{時間} \times 31 \text{日} = \$1,770.720 / \text{月}$$

リザーブドインスタンス料金

1年間：全額払い(\$6,125 × 2 node)

$$\$0.699 \times 2 \text{ node} \times 24 \text{時間} \times 31 \text{日} = \$1,040.112 / \text{月} (42\% \text{ off})$$

3年間：全額払い(\$7,585 × 2 node)

$$\$0.289 \times 2 \text{ node} \times 24 \text{時間} \times 31 \text{日} = \$430.032 / \text{月} (76\% \text{ off})$$

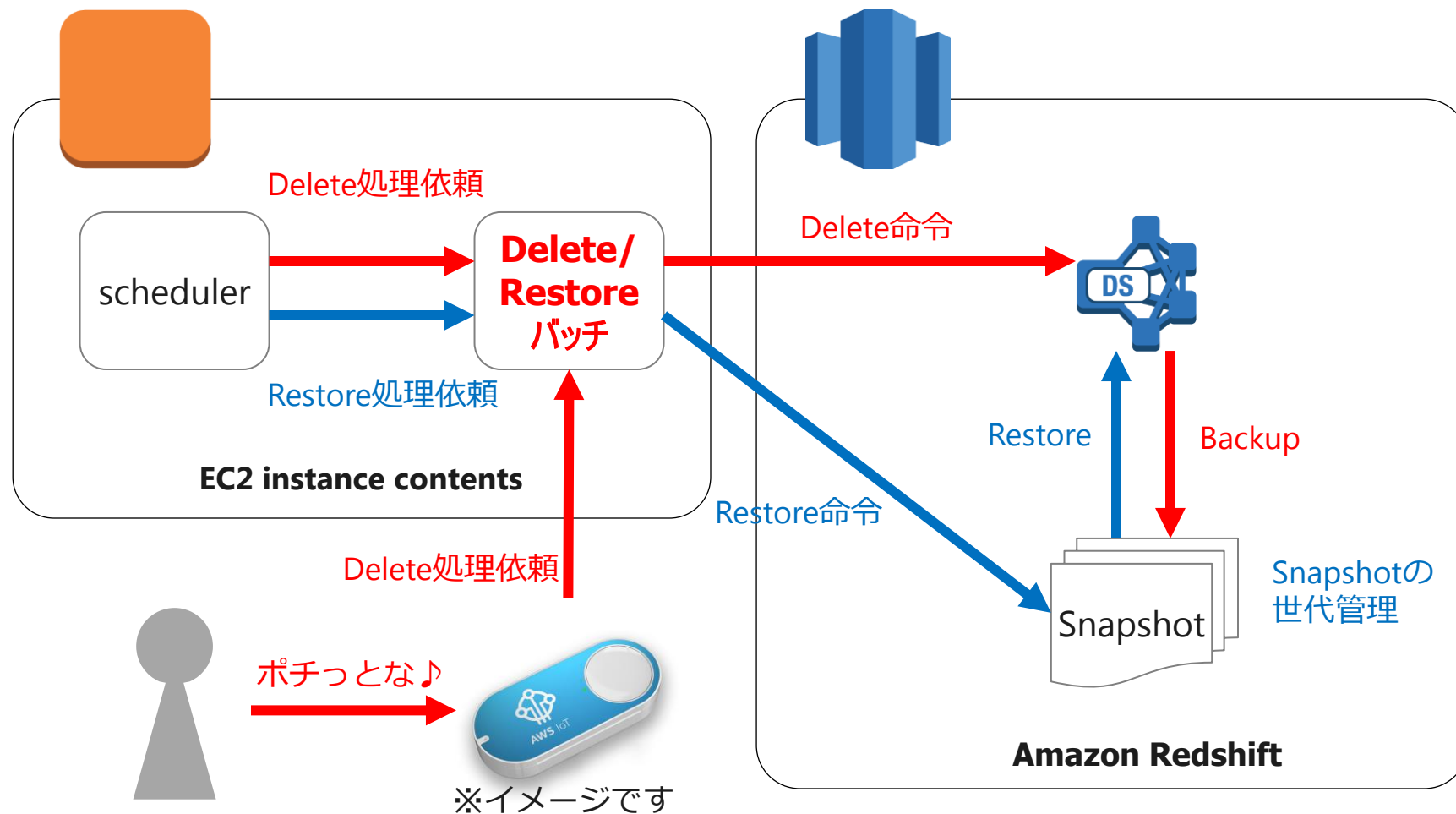
弊社運用中の料金

ちょっと変わった運用

 $\leq \$899.640 / \text{月} (50\% \text{ off})$

Redshift利用料金をケチる：Delete/Restoreバッチ

定時実行のDelete/Restoreバッチによって必要な時にだけRedshiftを稼働することにより、Redshiftの利用料金を最適化した



参考：Amazon Redshiftの利用料金を半額にしよう！ (<https://www.gixo.jp/blog/5556/>)

自動のDelete/Restoreバッチと手動のDelete実行でRedshiftコスト削減

Dense Storage : ds2.xlarge × 2 node 運用を想定した場合

オンデマンド料金

$$\$1.190 \times 2 \text{ node} \times 24 \text{ 時間} \times 31 \text{ 日} = \$1,770.720 / \text{月}$$

リザーブドインスタンス料金

1年間：全額払い(\$6,125 × 2 node)

$$\$0.699 \times 2 \text{ node} \times 24 \text{ 時間} \times 31 \text{ 日} = \$1,040.112 / \text{月 (42\% off)}$$

3年間：全額払い(\$7,585 × 2 node)

$$\$0.289 \times 2 \text{ node} \times 24 \text{ 時間} \times 31 \text{ 日} = \$430.032 / \text{月 (76\% off)}$$

弊社運用中の料金

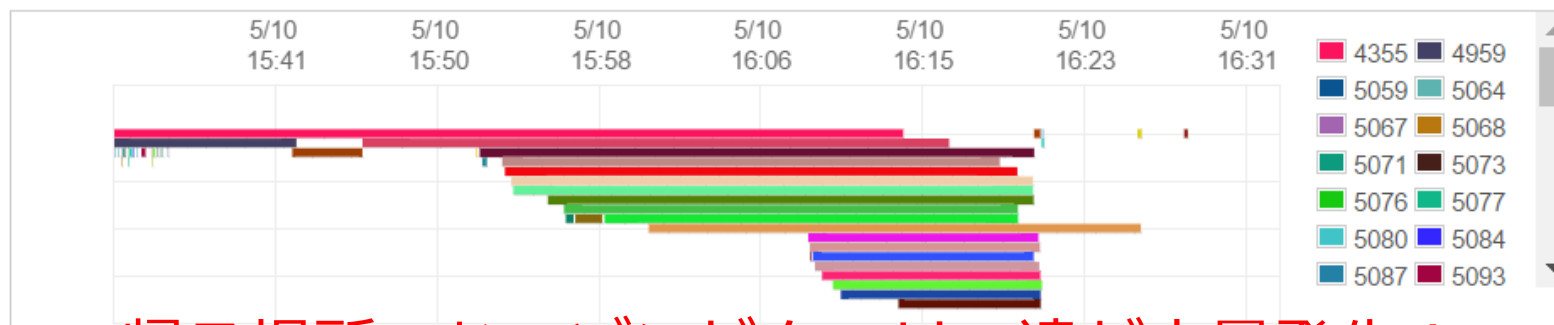
$$\$1.190 \times 2 \text{ node} \times 18 \text{ 時間} \times 21 \text{ 日} \leq \$899.640 / \text{月 (50\% off)}$$

Redshiftのリストアにはそれなりの時間が掛かるため、余裕をもってリストアを開始する必要がある。

1. リストア時間が日によってバラつきがある
2. リストア時間はノードの記憶容量に比例してそう
3. Dense Compute(SSD)のリストアは非常に速い

- Redshift利用料金をケチる
- ゾンビクエリー監視
- なんちゃってAUTO VACUUM
- チームによるデータ分析体制

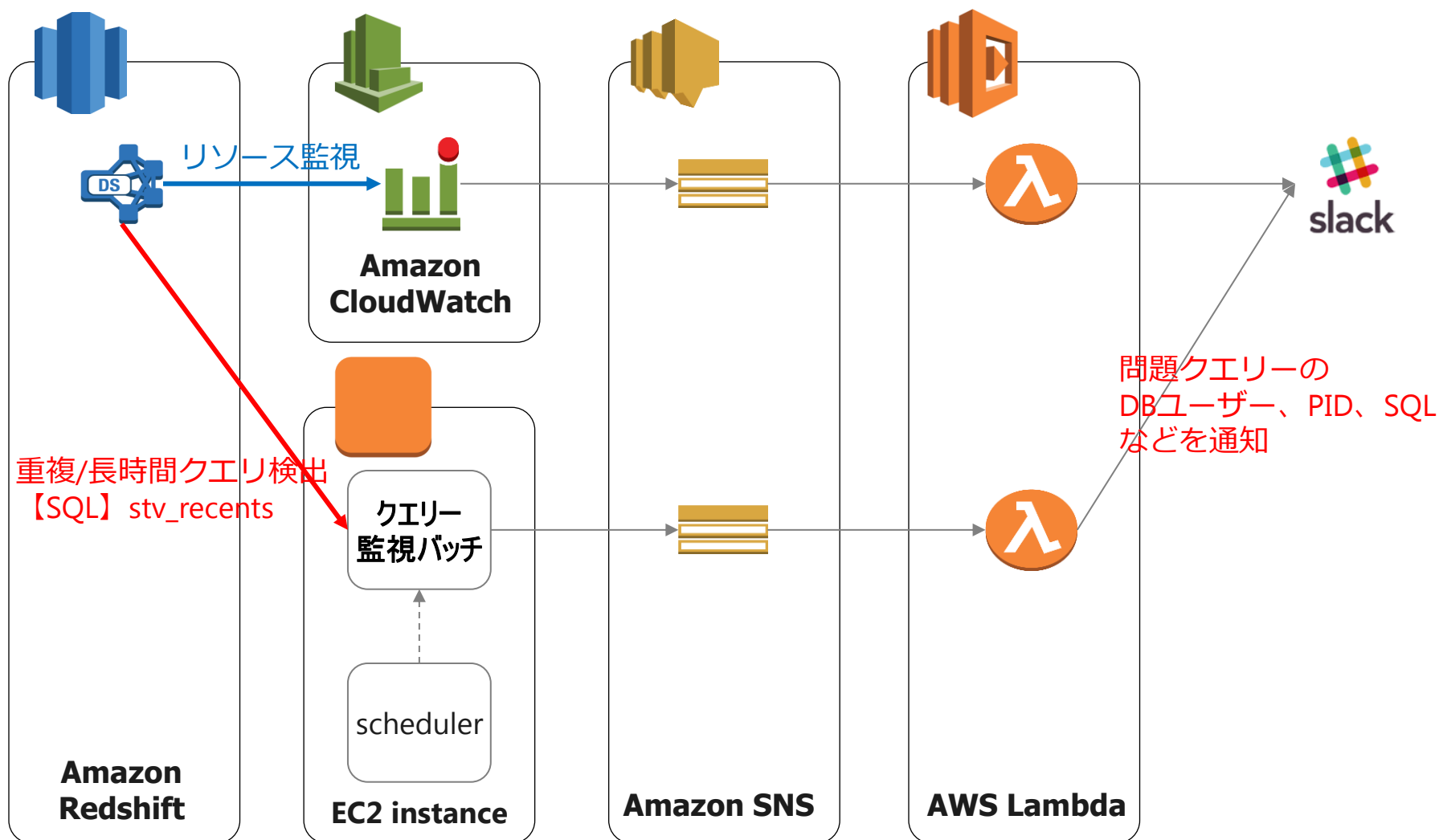
1. テーブルの結合条件が悪くクエリーが終わらない
2. クエリー実行中なのに何度も実行
3. 作業領域を使い過ぎてディスクがパンパン
4. そして、原因を作った本人は何が起きているか分からない



帰る場所のないゾンビクエリー達が発生！

ゾンビクエリー監視：処理フロー

Redshiftの様々な状況を収集してslackに通知することで「負荷の見える化」を実施
ゾンビクエリーが周知されることで早めに駆除できるようになった。



- Redshift利用料金をケチる
- ゾンビクエリー監視
- なんちゃってAUTO VACUUM
- チームによるデータ分析体制

定期的にAUTO VACUUMバッチを実行して、無駄領域がありそうなテーブルを特定して、VACUUM DELETE ONLYを実行。

1. スケジューラでAUTO VACUUMバッチを実行
2. 以下の2つのシステムテーブルからテーブル行数を取得
stv_tbl_perm テーブルに関する情報(未開放領域含む)
pg_class テーブルの統計情報
3. ある程度の行数差分があったテーブルで以下のクエリー実行
VACUUM DELETE ONLY
ANALYZE

db_name	space_name	table_name	table_all_count	table_count	kairi	now_no
			489895861	239849904	2.0425103	1
			472855024	230576704	2.0507493	2
			355865615	172832992	2.0590143	3
			194195108	183395872	1.0588849	4
			57370378	0.001	57370376	5
			57370378	0.001	57370376	6
			57370378	0.001	57370376	7
			15752964	14879292	1.0587173	8
			11411901	5611289.0	2.0337396	9
			5836032	5611289.0	1.0400519	10
			5615271	0.001	5615271	11

- Redshift利用料金をケチる
- ゾンビクエリー監視
- なんちゃってAUTO VACUUM
- チームによるデータ分析体制

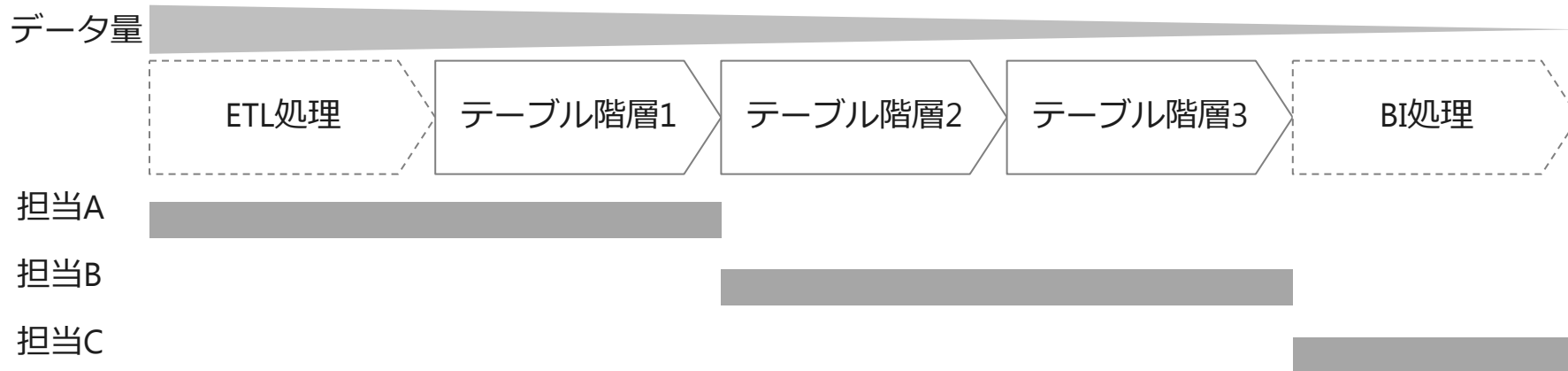
汎用化的な分析作業をテーブル階層ルールに落とし、作業者全員で作業を統一する

1. 分析作業を作業フロー化
2. 作業フローからテーブル階層化(特許取得済)
3. テーブル階層に沿った命名規則を作成
4. 作業フローと命名規則の絶対厳守

チームによるデータ分析体制：テーブル階層化のメリット

Redshiftのデータ容量の大きさを活かして多くの作業テーブルを作成。

- 作業範囲と成果物が明確化
- 作業の切り出しが容易になり下位分析者が参画しやすい
- 作業の後戻りが少なくなり分析サイクルを高速に回せる
- データ量が多い序盤にだけRedshiftパワーが必要
- BIツールのカスタムクエリの廃止



参考：データベースを“倉庫”ではなく“道具”として使う発想 (<https://www.gixo.jp/blog/9235/>)

