

SOAにおけるトランザクション

株式会社オージス総研
齋藤伸也

- 本発表について
 - 本発表で取り扱うのはシステムトランザクション。
 - トランザクションはデータの不整合が起きないようにする⇒品質に関わる。
 - ACIDトランザクションと新しいトランザクションをSOAのサービスの実装の観点で、考察する。

- 現在の一般的なシステム(RDBMS)が持つトランザクション機能
- X/Open の分散トランザクション処理参照モデル
 - SOAで実現する場合
 - 通信プロトコルはSOAP
 - WS-AT対応のAPサーバを利用しなければならない。
 - 実現するためのプラットフォームは限定される

ポイント

- データの不整合が起きないようにする仕組みである。

- クラウドプラットフォーム(key-value型データストア)におけるトランザクションの概念
- 基本的な原則：高い可用性、非同期連携
 - BASEトランザクション
 - 新しいACIDトランザクション
 - 不整合に対する対処：スターバックスは2相コミットをしない
 - ポイント
 - クラウドの制約の上で、データの不整合ができるだけ起きないようにする。
 - データの不整合に対処する。

- 新しいトランザクションは原子性、一貫性、分離性の保証が非常に限定的。

表：ACID特性の比較

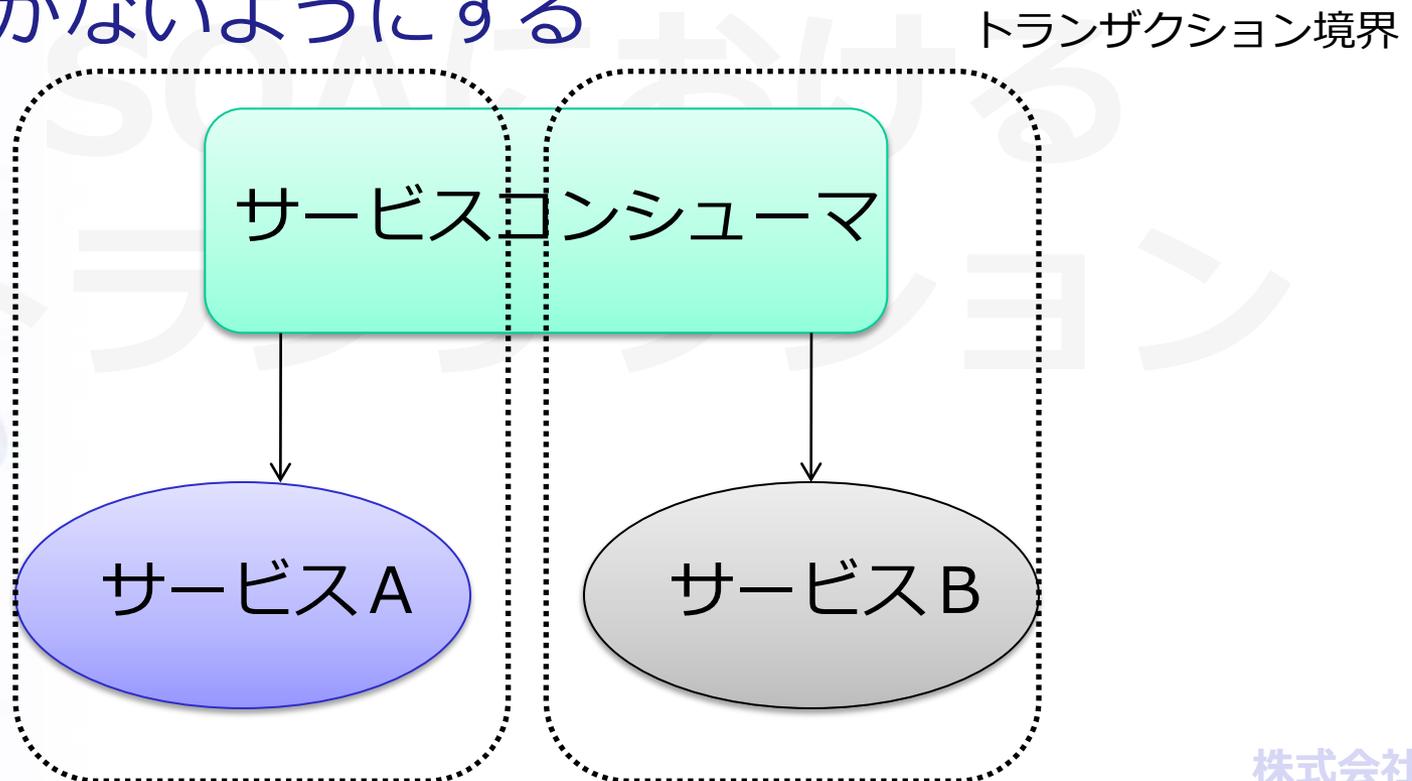
	ACIDトランザクション	新しいトランザクション
原子性	基本的に保証する 実現手段:2相コミット	限定的に保証する
一貫性	厳密に保証する 実現手段:2相コミット :データ制約	限定的に保証する 実現手段:結果整合性
分離性	厳密に保証する 実現手段:悲観的ロック	限定的に保証する 実現手段:楽観的ロック
永続性 (※)	基本的に保証する 実現手段:多重化、障害復旧 など	高いレベルで保証する 実現手段:多重化

※2相コミットプロトコル、BASE、新しいACIDで言及されている特性ではないが、従来とクラウドの基本的な特徴を記述。

新しいトランザクションの実現 1

SOAにおける
トランザクション

- データの不整合をできるだけ起こさないようにする
 - トランザクションはサービス内で完結させる
 - 複数のサービスにまたがるようなトランザクションがないようにする



- サービスコンシューマはサービスの処理の失敗を判断する
- サービスの処理に失敗した場合の処理を行う。

- サービスの処理の失敗の判断
 - サービスから処理の成否のメッセージを受け取る
 - サービスからメッセージが来なかった場合、ある一定時間を持って失敗とする。
- サービスの処理が失敗したとき
 - やりなおす。メッセージの再送信。
 - サービスにはメッセージのべき等性(Idempotent)を判断する処理が必要。
 - 補償する。補償処理を行うサービスを呼び出す。

- 全てのサービスに必要なわけではない
 - どのようなサービスに必要？
- 実現手段の洗い出し、パターン化
 - 今回、発表したものが全てではない。
- モジュール、フレームワークの作成