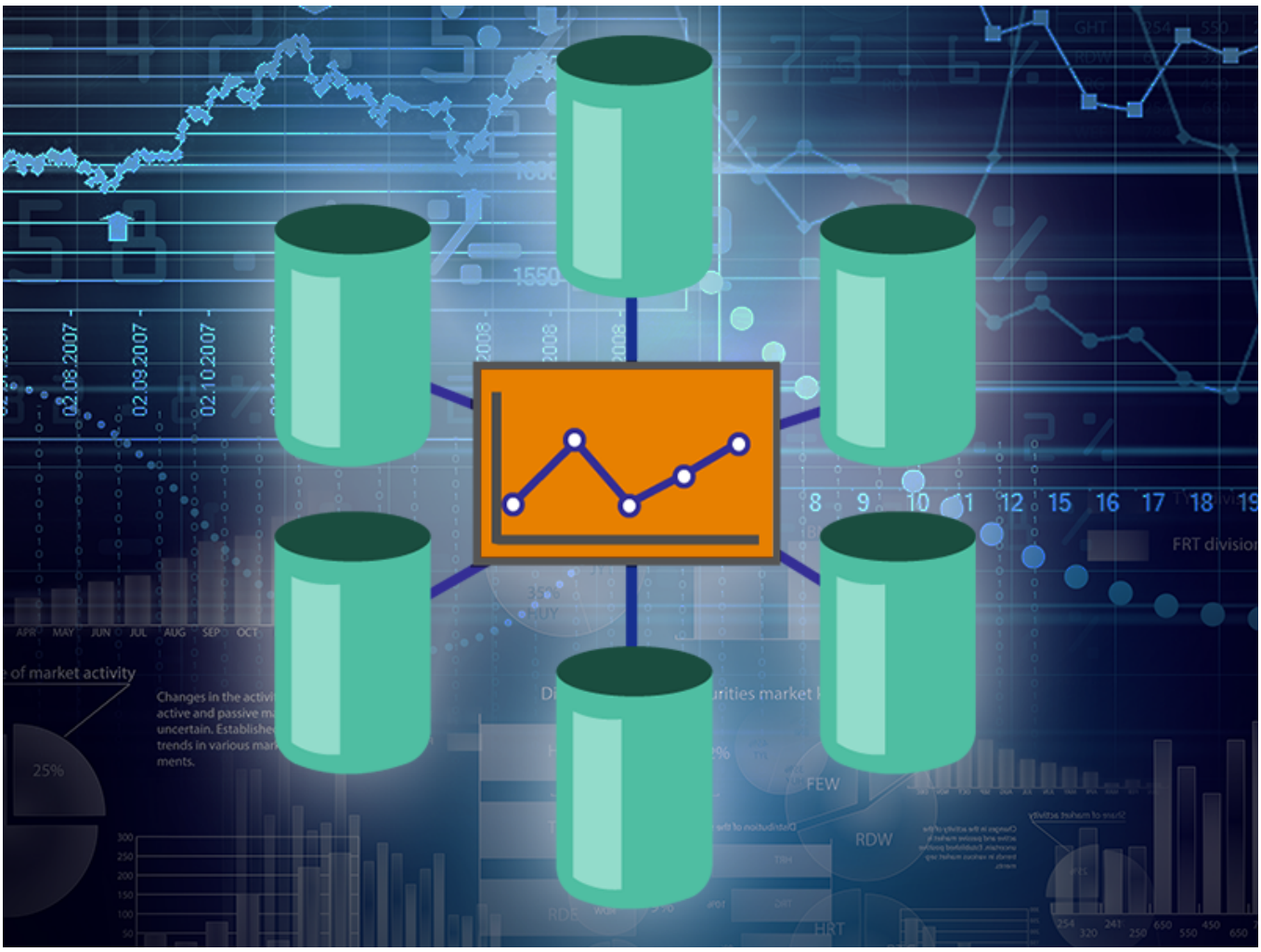
記事
Toshihiko Minamoto · 2021年5月24日 3m read

DeepSee: データベース、ネームスペース、マッピング（2/5）

以下の記事は、DeepSee の基本的なアーキテクチャを実装するためのガイドです。 この実装には、DeepSee キャッシュ用のデータベースと DeepSee 実装と設定用のデータベースが含まれています。



例 1: 基本アーキテクチャ

データベース

アナリティクスサーバー用のこの構成には、APP-CACHE と APP-DEEPSEE データベースが含まれています。
DeepSee が円滑に実行するためには、**DeepSee キャッシュを決してジャーナリングしない**ことが重要な設定となります。
ジャーナリングしてしまうと、ハイパージャーナリングやディスク容量の問題が発生するだけでなく、DeepSee エンジンのパフォーマンスが低下してしまいます。 このため、DeepSee キャッシュは、ジャーナリングが無効になっている別の DeepSee Cache データベース「APP-CACHE」に格納されます。

APP-DEEPSEE は、^DeepSee.* グローバルが含まれる、DeepSee の実装と設定用のデータベースです。
これらのグローバルは、定義と設定、Cube Manager、ユーザー設定など、ほとんどの DeepSee
実装を定義しています。 次に示すスクリーンショットに見られるように、すべてのデータベースは読み取り/書き込みが可能であり、APP-DEEPSEE
でのみジャーナリングが有効となるように決定されていることに注意してください。 このデータベースにはすべての定義、設定、およびユーザーデータが含まれているため、これをジャーナリングすることをお勧めします。

グローバルマッピング

次のスクリーンショットは、APP
ネームスペース上のこの基本アーキテクチャ実装のマッピングを示しています。 ^DeepSee.Cache.* と
^DeepSee.JoinIndex は DeepSee キャッシュを APP-CACHE データベースにマッピングしています。
^DeepSee.* グローバルはとりわけ、DeepSee の実装と設定を APP-DEEPSEE
データベースにマッピングしています。

コメント

基本アーキテクチャのこの例では、DeepSee キャッシュは専用のデータベースに保存されています。
このため、^DeepSee.Cache* と ^DeepSee.JoinIndex
グローバルのジャーナリングを無効にすることができます。

中断が発生した場合に DeepSee 実装（キューブ、サブジェクトエリア、DeepSee
アイテム、ユーザー設定など）の復元を実現できるのが、APP-DEEPSEE データベースのジャーナリングです。

この例に説明されている構成には、いくつかの欠点があります。
まず、同期をサポートするグローバルが処理されていない点です。 2 つ目は、APP-DEEPSEE
データベースには、ファクトテーブル、インデックス、およびその他の DeepSee
グローバルも含まれている点です。 そのため、APP-DEEPSEE
のサイズが肥大し、ジャーナリングと復元が実用的でなくなる可能性があります。
この構成は、たとえばキューブに大量のデータが含まれていない場合などに適用できます。

この連載の次の例
では、キューブ同期グローバル、ファクトテーブル、およびインデックスを個別のデータベースにマッピングする方法を説明します。

#マッピング #チュートリアル #デプロイ #データベース #分析 #初心者 #InterSystems IRIS BI (DeepSee)

ソースURL:
https://jp.community.intersystems.com/post/deepsee-%E3%83%87%E3%83%BC%E3%82%BF%E3%83%99%E3%83%BC%E3%82%B9%E3%80%81%E3%83%8D%E3%83%BC%E3%83%A0%E3%82%B9%E3%83%9A%E3%83%BC%E3%82%B9%E3%80%81%E3%83%9E%E3%83%83%E3%83%94%E3%83%B3%E3%82%B0%EF%BC%8825%EF%BC%89