

記事

[Toshihiko Minamoto](#) · 2020年6月3日 6m read

InterSystems IRISによる水平スケーラビリティ

先週、私たちは[InterSystems IRIS Data Platform](#)

を発表しました。これは、トランザクション、分析、またはその両方に関係なく、あらゆるデータの取り組みに対応する新しい包括的なプラットフォームです。 CachéとEnsembleでお客様が慣れ親しんでいる多くの機能が取り込まれていますが、この記事では、プ

ラットフォームの新機能の1つである**SQLシャーディング**

についてもう少し詳しく説明します。これはスケーラビリティに関する強力な新機能です。

ちょうど4分41秒の時間がある方は、スケーラビリティに関する[こちらの詳しい動画](#)をご覧ください。ヘッドホンがない方や聞き心地の良いナレーションが同僚の方の迷惑になると思う方は、どうぞ読み進めてください！

スケールアップとスケールアウト

1日に何百万件という株取引を処理する場合でも、1日に数万人の患者を治療する場合でも、このような業務を支えているデータプラットフォームは、こういった大きなスケールに透過的に対処できなければなりません。

「**透過的に**

」というのは、プラットフォームがスケーリングの面を請け負い、開発者やビジネスユーザーは処理量を気にすることなく、それぞれが専門とする業務とアプリケーションに専念することができるという意味です。

長年にわたり、 Cachéは**垂直スケーラビリティ**

をサポートしてきました。ハードウェアの進歩がソフトウェアによって透過的に利用され、非常に高いコア数と大量のRAMを効率的に活用したスケーラビリティです。これは「スケールアップ」と呼ばれており、十分なサイジングの取り組みをあらかじめ行うことで、均衡のとれた完璧なシステムを得る一方で、単一のシステムでコスト効率よく達成できることには固有の制限があります。

そして、そこに**水平スケーラビリティ**

が登場します。ワークロードが単一のサーバーではなく、クラスターで稼働する複数の個別のサーバーに分散される方法です。 Cachéは、すでにしばらくの間、スケールアウトの手段としてECPアプリケーションサーバーをサポートしてきましたが、InterSystems IRISは、SQLシャーディングも追加しました。

何が新しいのか？

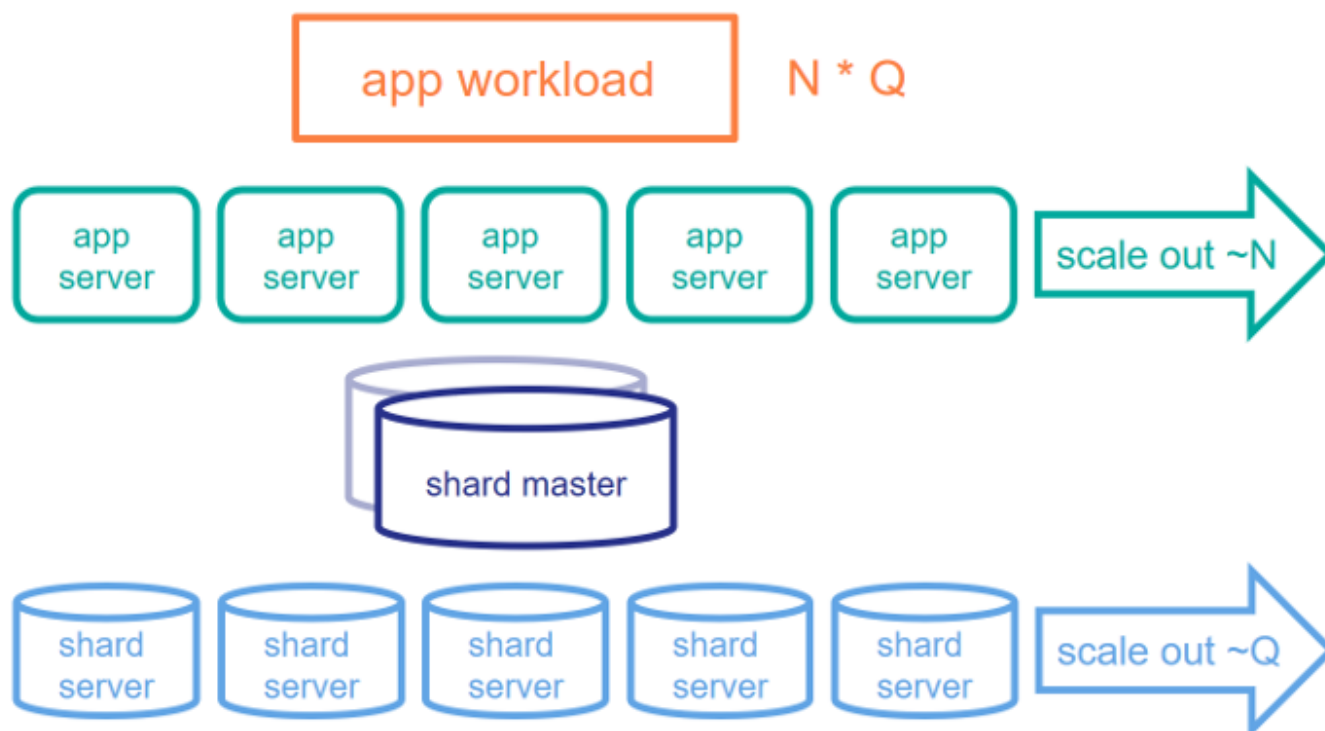
では、ECPアプリケーションサーバーと新しいシャーディング機能の違いは何でしょうか。

これらの違いを理解するために、ワークロードを詳しく見てみましょう。ワークロードは、数万個の小さなデバイスが小さなバッチのデータを連続的にデータベースに書き込む処理で構成されたり、少数のアナリストが一度に数ギガバイトのデータにまたがる分析クエリを発行する処理で構成されていたりします。

どちらの規模が大きいでしょうか。

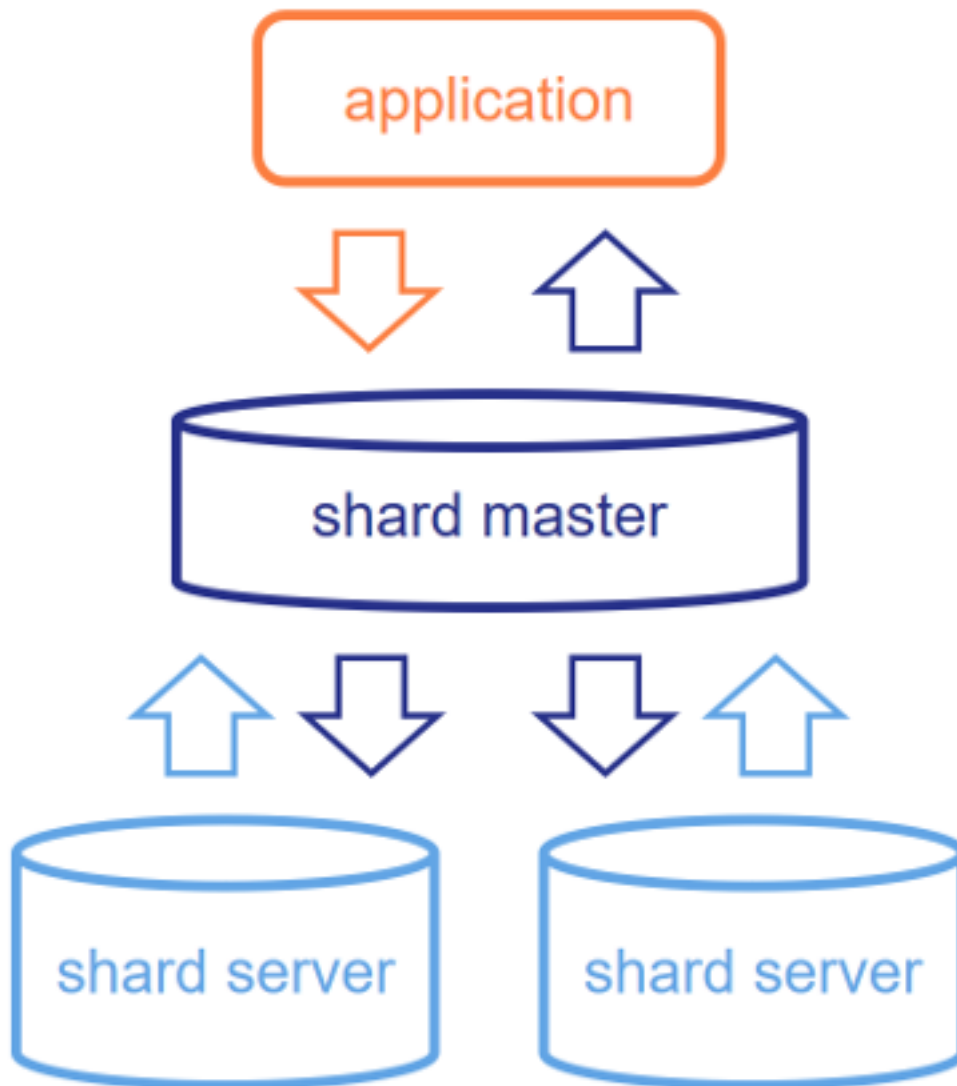
これは、釣り竿とビール樽のどちらが大きいかを答えるくらいわかりにくいものです。ワークロードには複数の次元があるため、これらをサポートするためのスケーリングには、もう少し微妙さが必要です。

大まかに単純化して、アプリケーションワークロード内の次のコンポーネントについて考えてみましょう。Nはユーザーワークロード、Qはクエリサイズを表します。前の例では、最初のワークロードのNが高くてQが低く、後者ではNが低くてQが高くなっています。ECPアプリケーションサーバーではアプリケーションユーザーを様々なサーバーに分割できるため、大きなNをサポートするのに非常に適しています。しかし、データセットが非常に大きくなり、ワーキングセットが単一マシンのメモリに収まらなくなると、それほど役に立つとは限りません。シャーディングは大きなQに対処し、サーバー間でデータセットを分割できるようにします。作業も可能な限りシャードサーバーに押し込まれます。



SQLシャーディング

では実際のところシャーディングは何を行うのでしょうか。これは、シャーディングされたテーブルのデータをシャードサーバーに格納されるばらばらのセットの行に分割するSQL機能です。シャードマスターに接続すると、このテーブルはすべてのデータを含む1つのテーブルのように見えますが、これに対するクエリはすべてのシャードサーバーに送信されるシャードローカルクエリに分割されます。そこで、シャードサーバーはローカルに保管されたデータに基づいて結果を計算し、その結果をシャードマスターに送り返します。シャードマスターはこれらの結果を集計して関連する組み合わせ論理を実行し、アプリケーションに結果を戻します。



このシステムは単純な「SELECT * FROM table」の場合は簡単ですが、並列処理を最大化するために（ほぼ）あらゆるSQLクエリを使用して最大量の作業がシャードにプッシュされるようにその中身には多くのスマートロジックが存在します。シャードキーにはどの行がどこに保存されるかが定義されており、典型的なクエリパターンを使用できます。最も重要なのは、JOINされることの多いテーブルが同じキーにシャーディングされる場合、JOINはシャードレベルで解決されるため、求めている高いパフォーマンスを得られるということです。

もちろん、これはシャーディングのほんの一部であり、ほかにもさまざまなことができますが、本質は上に示したとおりで、SQLシャーディングは、InterSystems IRISで作ることのできる非常にスケーラブルな料理がつまったレシピ本の新しい一品と言えます。ECPアプリケーションサーバーを補完するものであり、データセットのサイズに焦点を当てているため、多くの分析ユースケースに適しています。ECPアプリケーションサーバーと同様に、アプリケーションに対して完全に透過的であり、非常に具体的なシナリオに対応できるより創造的なアーキテクチャを備えています。

さらなる学習のためのリソース

以下のグローバルサミット2017でのセッションの録画を <http://learning.intersystems.com> でご覧いただけます。

- 「[What's Lurking in Your Data Lake](#)」 スケーラビリティと特にシャーディングに関する技術概要
- 「[We Want More! Solving Scalability](#)」 非常にスケーラブルなプラットフォームを必要とする関連ユースケースの概要

新しいプラットフォームのその他の機能については、learning.intersystems.com に掲載のInterSystems IRISの[リソースガイド](#)もご覧ください。

特定のユースケースでシャーディングを試してみるには

、<http://www.intersystems.com/iris>

をご覧ください。ページの下にあるフォームに必要事項をご入力の上アーリーアダプタープログラムにお申込みいただくか、今年後半に公開予定のフィールドテストバージョンをご期待ください。

[#Artificial Intelligence \(AI\)](#) [#ECP](#) [#SQL](#) [#シャーディング](#) [#分散データ管理](#) [#分析](#) [#Machine Learning \(ML\)](#)
[#InterSystems IRIS](#)

ソースURL:<https://jp.community.intersystems.com/post/intersystems-iris%E3%81%AB%E3%82%88%E3%82%8B%E6%B0%B4%E5%B9%B3%E3%82%B9%E3%82%B1%E3%83%BC%E3%83%A9%E3%83%93%E3%83%AA%E3%83%86%E3%82%A3>