

Alibaba Cloud Table Store

テスト環境

Document Version20190523

目次

1 テスト環境.....	1
2 テストツール.....	3
3 テストモデル.....	4
4 テスト指標.....	7
5 テスト結果 (高性能インスタンス).....	8
6 テスト結果 (容量インスタンス).....	11

1 テスト環境

Table Store インスタンス

このテストでは2つの Table Store インスタンスを使用します。

- ・ 高性能インスタンス: 中国 (杭州) リージョンにあります。
- ・ 容量インスタンス: 中国 (杭州) リージョンにあります。



注:

- ・ 高性能インスタンスと容量インスタンスは異なるディスクタイプを使用します。前者は SSD を使用し、後者は SSD と SATA ディスクの組み合わせを使用します。高性能インスタンスは、ディスクからデータを読み取る際により良いパフォーマンスを提供します。容量インスタンスは、データをディスクに書き込む際の高性能インスタンスに匹敵する性能で、単価が大幅に下がります。欠点は、ディスクデータの読み取りパフォーマンスが低いことです。2つのインスタンスは、キャッシュからデータを読み取る時に同様のパフォーマンスを提供します。
- ・ 従量課金の製品として、Table Store は1時間あたりに実際に使用するリソースに基づいて請求します。Table Storeは、非常に高い読み取り/書き込み同時実行性をサポートしています。大規模なパフォーマンステストは高額になる可能性があります。大規模なパフォーマンステストを実行する場合、妥当な費用で試験結果を得るためには [チケットを起票し](#)、[サポートセンターへお問い合わせください](#)。

ストレスランナー

すべてのテストは同じグループのランナーで行われます。

- ・ 数量: 5
- ・ タイプ: Alibaba Cloud ECS
- ・ リージョン: 中国(杭州) ゾーン F
- ・ モデル: 共有コンピューティング
- ・ 構成:
 - CPU: 8 コア
 - メモリ: 8 GB
 - インスタンスタイプ: 最適化された I/O
 - ネットワークタイプ: クラシックネットワーク
 - オペレーティングシステム: Ubuntu 16.04 64-bit



注：

パブリックネットワーク環境の品質は保証できず、Table Store の操作はネットワークのレイテンシに影響されるため、予測できないネットワーク要因による干渉を避けるために、Table Store インスタンスと同じリージョンの ECS インスタンスおよびプライベート Table Store アドレスを使用します。

2 テストツール

YCSB

Yahoo! Cloud Serving Benchmark (YCSB) は、NoSQL 製品の読み書きパフォーマンスをテストするためのオープンソースの分散パフォーマンステストツールです。

YCSB は、Cassandra、MongoDB、HBase、Redis、Infinispan などの一般的な NoSQL データベースおよびデータグリッド製品をサポートしています。インストールおよび使用方法は簡単です。テストデータタイプとサポートされているデータベースシステムを拡張できます。

インストール方法

『[YCSB ドキュメント](#)』をご参照ください。

YCSB では、まだ Table Store でのテストをサポートしていません。関連するテストツールコードはオープンソースで作成されています。

3 テストモデル

- ・ テーブル構造

プライマリーキー名	データ型	エンコード方法	長さ
userid	String	4-Byte-Hash + Long .toHexString	20

- ・ 属性列

属性列名	データ型	長さ
field0	String	100
field1	String	100
field2	String	100
field3	String	100
field4	String	100

- ・ パーティション数量

Table Store の自動ロードバランシング機能は、各パーティションのデータボリュームとアクセス要求に基づいてテーブルパーティションを動的に分割します。このプロセスは人間の介入を必要としません。このテストでは、通常 1、4、16 のパーティションを持つテーブルのパフォーマンスデータを選択します。

デフォルトでは、新しいデータテーブルには単一のデータパーティションがあります。新しいテーブルを手動で分割するには、[チケットを起票し、サポートセンターへお問い合わせください](#)。

- ・ テストケース

各ランナーで N スレッドを開始し、`com . alicloud . openservic es . tablestore . SyncClient` をスレッドごとに作成し、Table Store API を呼び出します。

- テストケースは次のとおりです。

- ランダム書き込み: テストは `SyncClient . putRow` を呼び出します。各要求は 1 行のデータを含み、1 時間持続します。
- バッチ書き込み: テストは `SyncClient . batchWrite Row` を呼び出します。各要求には 200 行のデータが含まれ、1 時間持続します。
- ランダム読み取り: テストは `BatchWrite Row` を呼び出します。各パーティションに 20 GB のデータを書き込み、次に `SyncClient . getRow` を呼び出します。各要求は 1 行のデータを受け取り、30 分間持続します。
- ランダムな取得範囲: テストは `BatchWrite Row` を呼び出します。各パーティションに 20 GB のデータを書き込み、その後 `SyncClient . getRange` を呼び出します。各要求は 100 行のデータを受け取り、30 分間持続します。

すべてのテストケースは、Table Store インスタンスのプライベートネットワークアドレスに直接要求を送信するので、ネットワーク環境によって引き起こされる影響は回避されます。

このパフォーマンステストは、サービスパフォーマンスの限界テストではありません。このテストは、Table Store サーバーの調整処理はトリガーされません。Table Store の自動ロードバランシング機能は、単一のテーブルで提供されるサービス機能の水平方向の拡大を保証します。大規模なパフォーマンステストはバックエンドの調整を引き起こし、高額になる可能性があります。大規模なパフォーマンステストを実行する場合は、妥当な費用で試験結果を得るために、[チケットを起票し、サポートセンターへお問い合わせください](#)。

Table Store の `BatchWriteRow` オペレーションは、パーティションによって同時に処理されます。各パーティションに書き込まれるデータは、単一のディスク書き込み操作です。各 `BatchWriteRow` の書き込みディスク操作を減らし、書き込みパフォーマンスを効果的に向上させるために、データパーティションキーによって `BatchWriteRow` 要求を集約することを推奨します。

ランダム読み取りおよびランダム取得範囲のテストケースでは、データは書き込まれません。テストが進むにつれて、キャッシュヒット率は増加します。負荷の低いシナリオで

は、キャッシュヒット率はゆっくりと増加するため、テストはディスク I/O 機能と密接に関係しています。負荷が高いシナリオでは、キャッシュヒット率が急激に増加するため、テストはディスク I/O 機能との関連性が低くなります。

BatchWriteRow および GetRange テストケースは、大量のネットワーク帯域幅を占有します。Table Store インスタンスの読み取りまたは書き込みのパフォーマンスが予想よりも低い場合は、ネットワーク帯域幅が完全に占有されているかどうかを確認してください。

Table Store の読み取りパフォーマンスは、データ量とキャッシュヒット率の影響を大きく受けます。シナリオによっては、GetRow および GetRange テストケースの制限を超える場合があります。これら 2 つのテストケースによって生成されたパフォーマンスデータを複製できます。つまり、このレポートのデータを同様のシナリオの参照として使用できます。実際の読み取りスループット、書き込みスループット、またはレイテンシがこのレポートのデータと大きく異なる場合は、原因を分析するためにチケットを起票し、サポートセンターへお問い合わせください。

4 テスト指標

すべてのストレステストは同期要求を使用するため、要求 QPS はランナー数と平均レイテンシと関連しています。この関係は次のとおりです。

合計 QPS = $1000 / \text{平均レイテンシ} * 5 \text{ (ランナー数)} * \text{ランナーあたりのスレッド数}$

- ・ Runner: ランナーの数
- ・ Thread: 各ランナーによって作成された YCSB スレッドの数
- ・ QPS (行): 1 秒間に読み書きされる行数
- ・ 平均レイテンシ (ミリ秒): ランナーによって計算された平均レイテンシ (ミリ秒)

たとえば、4つのパーティション、5つのランナー、ランナーあたり5つのスレッドおよび4ミリ秒のレイテンシを持つシナリオでは、単一のスレッドの理論上の QPS は $1000 / 4 = 250$ です。このシナリオにおける合計 QPS は、 $250 * 5 \text{ (ランナーの数)} * 5 \text{ (ランナーあたりのスレッド数)} = 6250$



注:

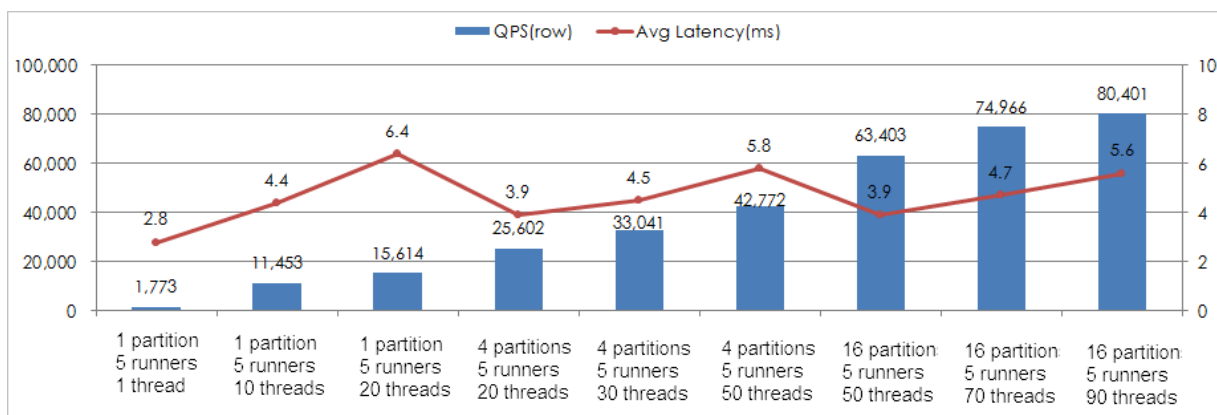
このパフォーマンステストは、サービスパフォーマンスの限界テストではありません。このテストは、Table Store サーバーの調整処理がトリガーされません。Table Store の自動負荷分散機能は、単一のテーブルで提供されるサービス機能の水平方向の拡大を保証します。

5 テスト結果 (高性能インスタンス)

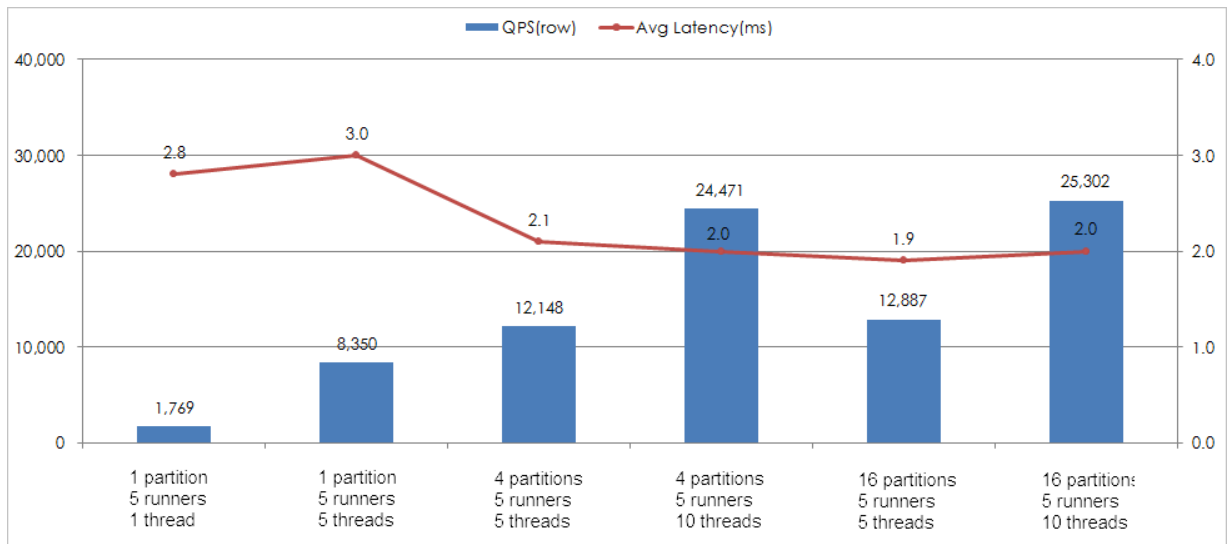
テスト分析

- ・ 同じシャードの場合、ストレスが増加してもランダム書き込み操作の遅延の変化がわずかであれば、このレイテンシのリソースは比較的適切です。ストレスが増加すると、QPS は大きくなります。
- ・ 同じシャードの場合、ストレスが増加してランダム書き込み操作の遅延がより長くなると、シャードがあるサーバーに要求のキューが表示されます。ストレスが増加すると、QPS は直線的には増加しません。
- ・ このパフォーマンステストは、サービスパフォーマンスの限界テストではありません。このテストは、Table Store サーバーの 調整処理がトリガーされません。Table Store の自動ロードバランシング機能は、単一のテーブルで提供されるサービス機能の水平方向の拡大を保証します。

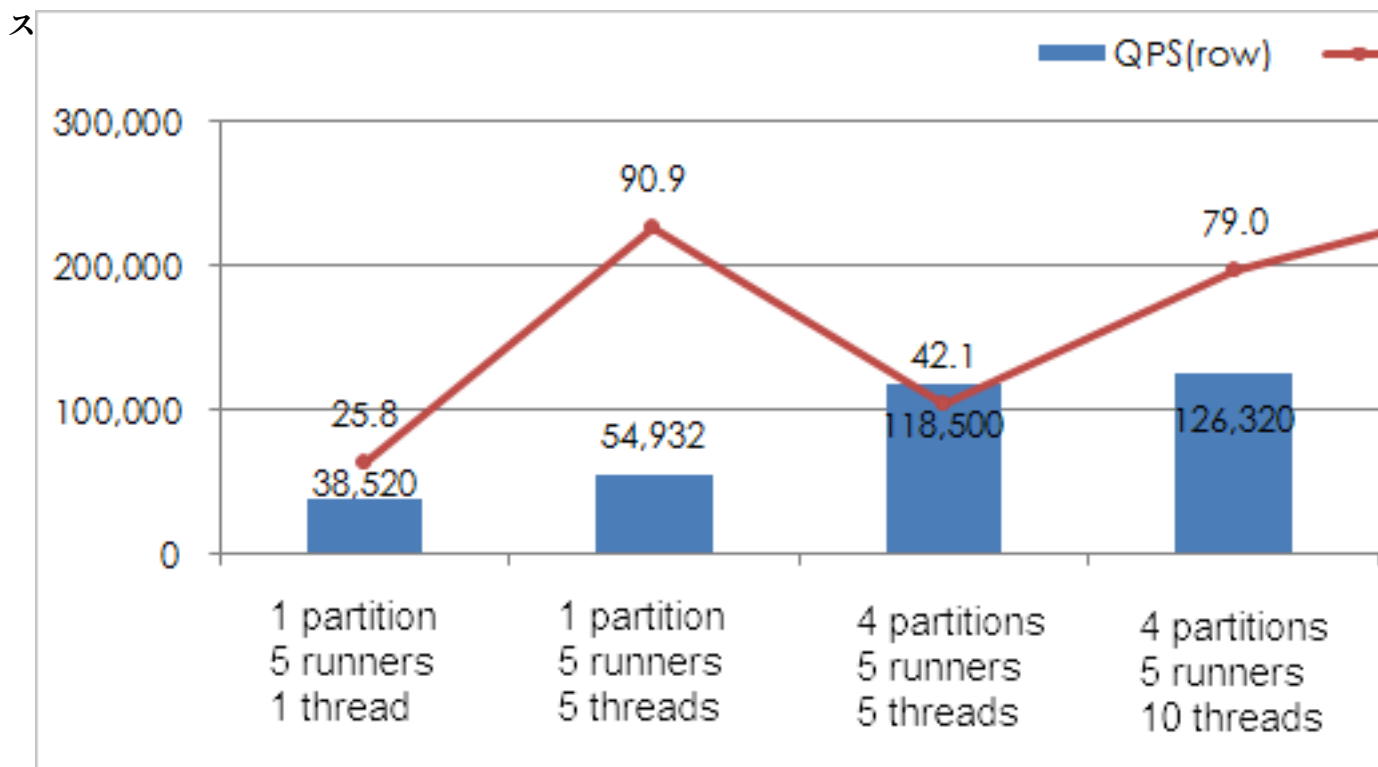
ランダム書き込みパフォーマンス



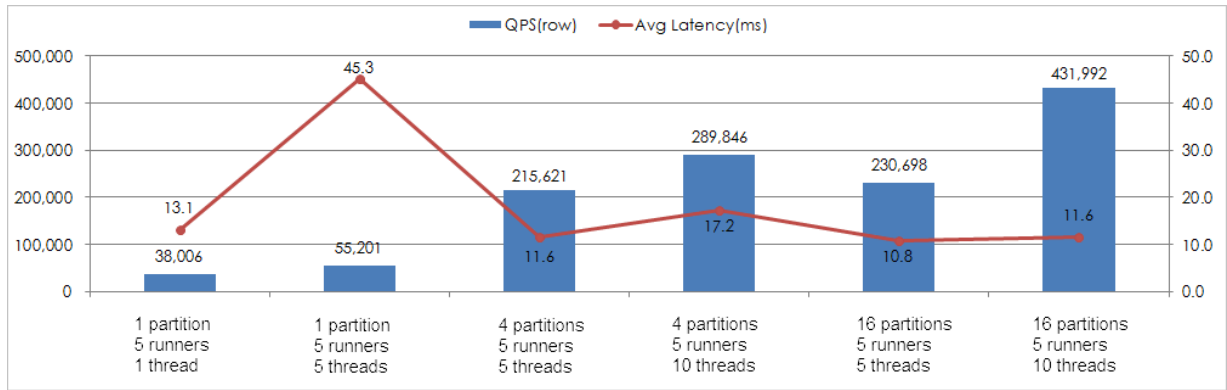
ランダム読み取りパフォーマンス



ランダムバッチ書き込みパフォーマンス



ランダム取得範囲パフォーマンス

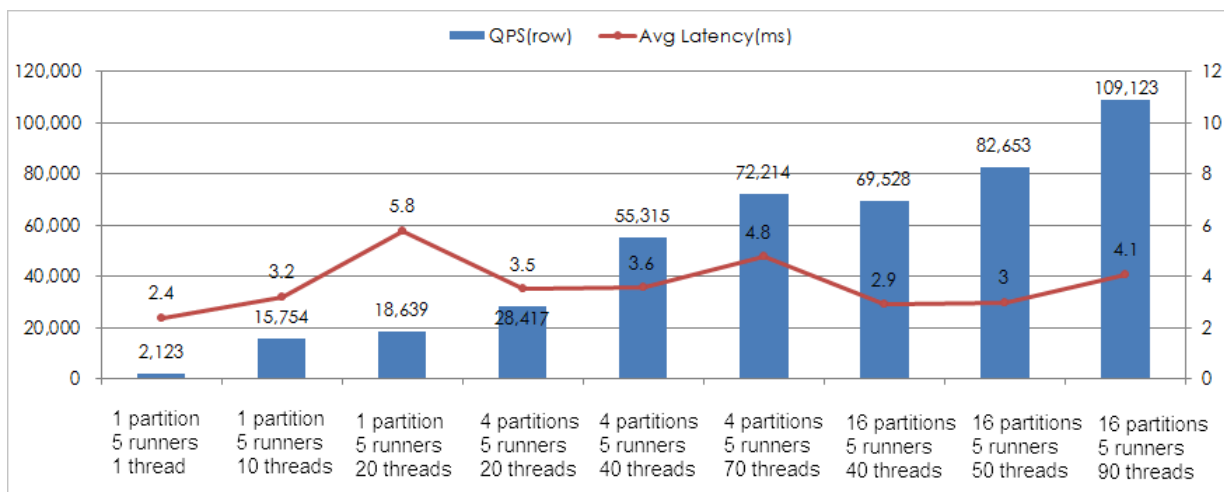


6 テスト結果 (容量インスタンス)

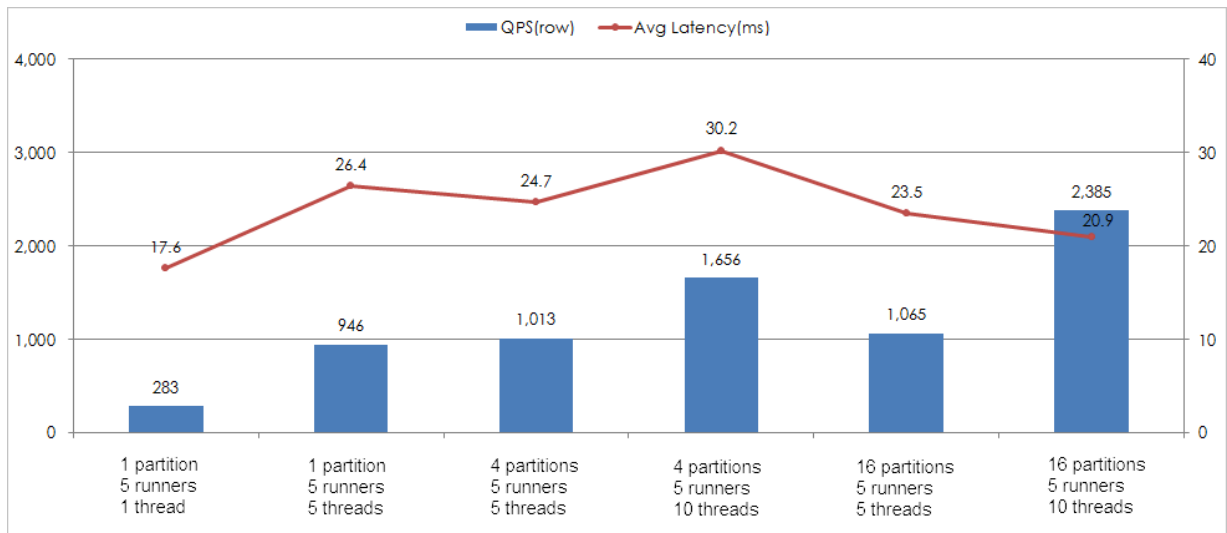
テスト分析

- ・ 同じシャードの場合、ストレスが増加してもランダム書き込み操作の遅延の変化がわずかであれば、このシャードのリソースは比較的適切です。ストレスが増加すると、QPS は大きくなります。
- ・ 同じシャードの場合、ストレスが増加してランダム書き込み操作の遅延が長くなると、シャードがあるサーバーに要求のキューが表示されます。ストレスが増加すると、QPS は直線的には増加しません。
- ・ ランダム読み取りおよびランダム取得範囲テストケースのパフォーマンスは、キャッシュヒット率の影響を大きく受けます。キャッシュヒット率がテスト結果に影響を与えないようにするために、ヒット率を極端に低いレベルに設定します。容量インスタンスの使用では、実際のパフォーマンスはこのテストの結果をかなり超える可能性があります。
- ・ このパフォーマンステストは、サービスパフォーマンスの限界テストではありません。このテストは、Table Store サーバーの調整処理がトリガーされません。Table Store の自動負荷分散機能は、単一のテーブルで提供されるサービス機能の水平方向の拡大を保証します。

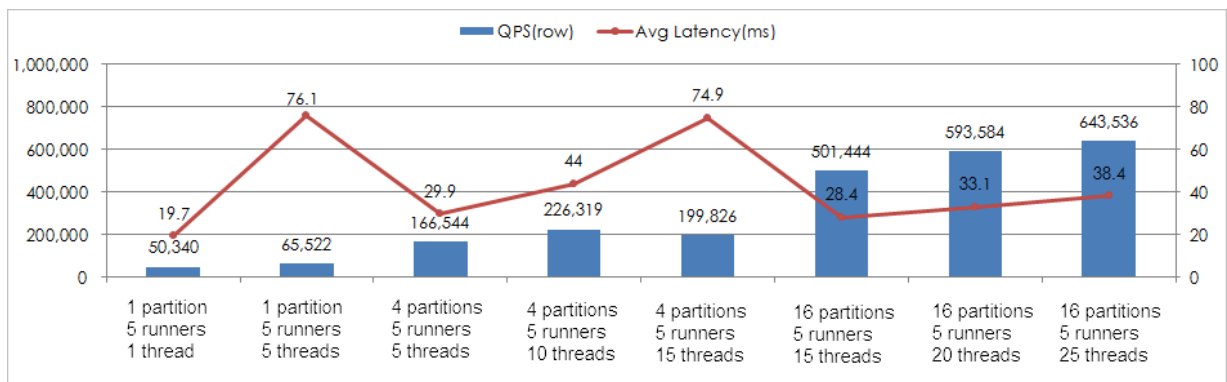
ランダム書き込みパフォーマンス



ランダム読み取りパフォーマンス



ランダムバッチ書き込みパフォーマンス



ランダム取得範囲パフォーマンス

