

記事

[Toshihiko Minamoto](#) · 2021年7月8日 6m read

[Open Exchange](#)

ランダム読み出しIOストレージパフォーマンスツール

目的

このツールは、データベース内からランダム読み出し入力/出力 (IO) を生成するために使用されます。このツールの目的は、目標IOPSを達成し、許容範囲内のディスク応答時間の維持を確保するために、可能な限り多くのジョブを駆動することです。IOテストから収集された結果は、IOサブシステムに基づいて構成ごとに異なります。これらのテストを実行する前に、対応するオペレーティングシステムとストレージレベルの監視が、今後の分析のためにIOパフォーマンスの測定データを保存するように設定されていることを確認してください。

方法論

少数のプロセスでプロセスごとに10,000回の反復から始めます。オールフラッシュストレージレイでは、プロセスごとに100,000回の反復を使用します。次に、10ジョブから開始して、10、20、40など追加し、プロセスの数を増やしていきます。応答時間が一貫して10ミリ秒を超えるか、計算されたIOPSが直線的に増加しなくなるまで、個々のテストを実行し続けます。

目安として、8KBおよび64KBのデータベースランダム読み出し (キャッシュされていない) の次の応答時間は、通常、オールフラッシュレイで許容されます。

- 平均2ミリ秒以下
- 5ミリ秒を下回らない

このツールでは、事前に拡張された空のIRIS.DATデータベースが、サーバーのメモリサイズの少なくとも2倍であり、ストレージコントローラーのキャッシュサイズの少なくとも4倍である必要があります。読み出しがファイルシステムキャッシュにキャッシュされてしまわないように、データベースをメモリ容量よりも大きくする必要があります。

このツールは、データベースブロックをメモリに読み込ませるObjectScript VIEWコマンドを使用するため、期待した結果が得られない場合は、恐らくすべてのデータベースブロックがすでにメモリ内にあるということになります。

仕様と目標

次の表に、環境の仕様と目標を記入してください。

仕様	例
ストレージ	ストレージレイの仕様
物理サーバー	CPU、メモリ仕様
仮想マシン	Red Hat Enterprise Linux 7 24 vCPU, 40GB vRAM
データベースサイズ	200GB
共有メモリ	Huge Pagesを使用して割り当てられた26956MBの共有メモリ；24000MBのグローバルバッファ、1000MBのルーチンバッファ
目標IOPS	2000
目標応答時間	5ミリ秒以下

インストール

[こちら](#)からGitHubでPerfTools.RanRead.xmlツールをダウンロードします。

テストが速すぎる場合や結果が期待どおりでない場合は、反復回数を10000に増やします。

```
USER> do ##class(PerfTools.RanRead).Run("/usr/iris/db/zranread",20,10000)
```

```
InterSystems Random Read IO Performance Tool
```

```
-----  
Starting 20 jobs in the background.  
To terminate jobs run: do ##class(PerfTools.RanRead).Stop()  
Waiting for jobs to finish.....  
Random read background jobs finished.  
20 processes (1000 iterations) average response time = 7.18ms  
Calculated IOPS = 2787
```

結果

各実行の結果は、USERのPerfTools.RanReadというSQLテーブルに保存されます。次のSQLクエリを実行すると、結果の概要を確認できます。

```
SELECT RunDate,RunTime,Database,Iterations,Processes,  
       {fn ROUND(AVG(ResponseTime),2)} As ResponseTime,  
       {fn ROUND(AVG(IOPS),0)} As IOPS  
FROM PerfTools.RanRead  
GROUP BY Batch
```

結果セットをタブ区切りのテキストファイルにエクスポートするには、次の手順を実行します。

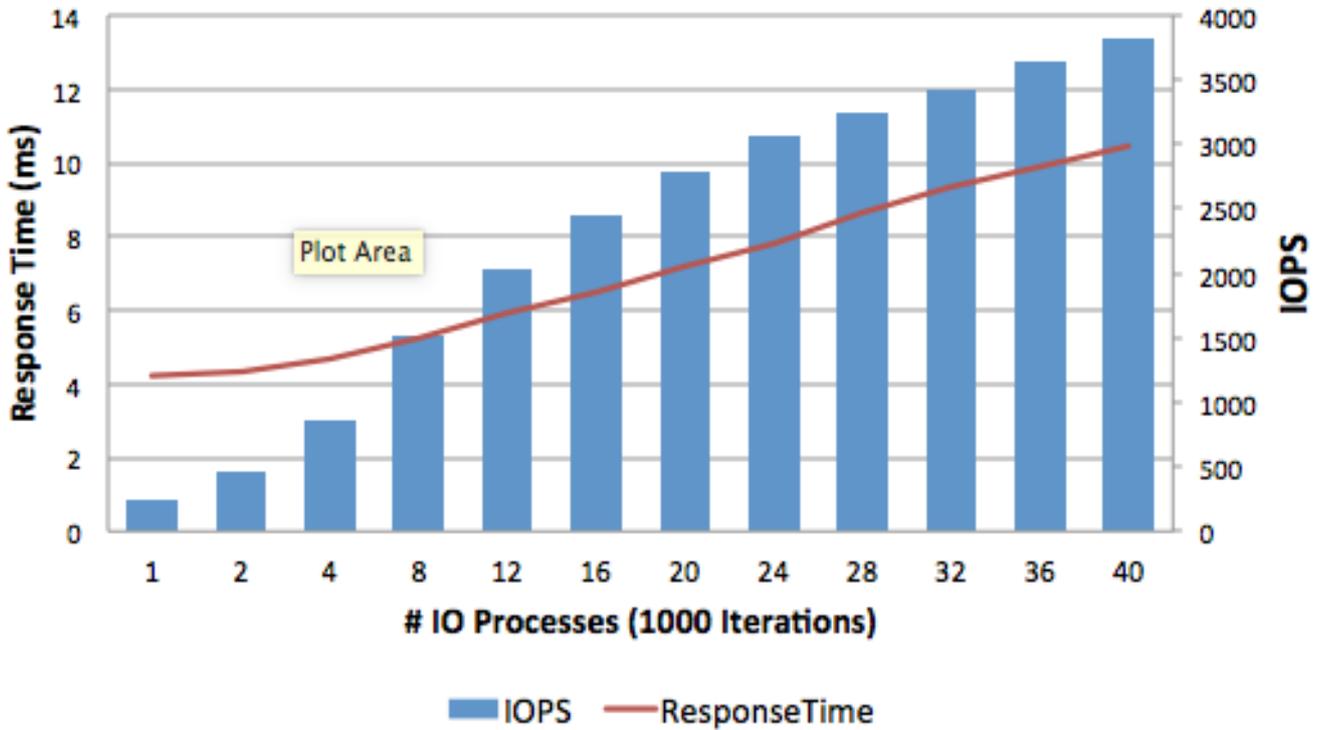
```
USER> do ##class(PerfTools.RanRead).Export("/usr/iris/db/zranread/")
```

```
Exporting summary of all random read statistics to /usr/iris/db/zranread/PerfToolsRan  
Read_20160523-1408.txt  
Done.
```

分析

エクスポートしたテキストファイルをExcelで開いてコピーし、PerfToolsRandomRead /Analysis / Template.xlsxスプレッドシートに貼り付け、グラフ化します。

Random Read IO Performance Results



サンプルスプレッドシートは[こちら](#)からGitHubでダウンロードすることができます。

クリーンアップ

テストの実行が終了したら、次のコマンドを実行して履歴を削除します。

```
%SYS> do ##class(PerfTools.RanRead).Reset()
```

[#データベース #パフォーマンス #InterSystems IRIS](#)
[InterSystems Open Exchange](#)で関連アプリケーションを確認してください

ソースURL:

<https://jp.community.intersystems.com/post/%E3%83%A9%E3%83%B3%E3%83%80%E3%83%A0%E8%AA%AD%E3%81%BF%E5%87%BA%E3%81%97io%E3%82%B9%E3%83%88%E3%83%AC%E3%83%BC%E3%82%B8%E3%83%91%E3%83%95%E3%82%A9%E3%83%BC%E3%83%9E%E3%83%B3%E3%82%B9%E3%83%84%E3%83%BC%E3%83%AB>