

# ブロック・ボリューム・サービス・メトリック を使用したブロック容量スループットおよび IOPSの計算

Oracle Cloud Infrastructureコンソールの使用

ORACLE WHITE PAPER | 2019年6月

## 免責事項

下記事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないで下さい。オラクルの製品に関して記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

## 改訂履歴

このホワイト・ペーパーは、初版の公開後、次の改訂がありました。

日付	改訂内容
2019年6月21日	初版

Oracle Cloud Infrastructureホワイト・ペーパーの最新のバージョンは、  
<https://cloud.oracle.com/iaas/technical-resources>で入手できます。



## 目次

概要	4
対象範囲と前提条件	4
ブロック・ボリュームのパフォーマンス・メトリック	4
スループットの計算	5
IOPSの計算	8
結論	13

## 概要

Oracle Cloud Infrastructure Block Volumesサービスは、ブロック・ボリュームのヘルス、容量およびパフォーマンスの表示を可能にするメトリックを提供します。このホワイト・ペーパーは、パフォーマンス・テスト・シナリオの手順を紹介し、このメトリックを使用して、ブロック・ボリュームのパフォーマンス特性を特定する方法を示します。

このシナリオでは、アタッチされたブロック・ボリュームに対して、Oracle Cloud Infrastructure ComputeインスタンスでFlexible I/O Tester (FIO)コマンドを実行します。次に、その出力を、コンソールで得られるBlock Volumesサービス・メトリックと比較して、パフォーマンス特性を相互に関連付けます。

## 対象範囲と前提条件

テストを目的としたI/Oワークロードのシミュレートに使用可能なツールであるFIOに精通している必要があります。

テストに使用されるインスタンス・シェイプはVM.Standard2.24です。このシェイプの詳細は、[コンピュート・シェイプ](#)を参照してください。スループット・テストに使用されるボリューム・サイズは1000GBです。IOPSテストに使用されるボリューム・サイズは600GBです。

Computeインスタンスで使用可能なネットワーク帯域幅は、そのサイズ(つまり、コアの数)に基づいて異なります。これらのテストでは、25-Gbpsネットワーク帯域幅もフルに使用されるため、24-コアVMが使用されます。

Block Volumesのパフォーマンス・メトリックおよび特性は、[サービス・ドキュメント](#)で公開されています。

## ブロック・ボリュームのパフォーマンス・メトリック

ブロック・ボリュームのパフォーマンス・メトリックは、スループット、I/O操作およびレイテンシです。

Block Volumesサービス・メトリックは、ブロック・ボリュームがアタッチされているComputeインスタンスに対応するボリュームの操作およびスループットの測定に役立ちます。ブロック・ボリュームのパフォーマンスに関連する、コンソールで得られるメトリックは、次のとおりです。

- ボリューム読取りスループット(単位: バイト)
- ボリューム書込みスループット(単位: バイト)
- ボリューム読取り操作(単位: バイト)
- ボリューム書込み操作(単位: バイト)

詳細は、[ブロック・ボリューム・メトリック](#)を参照してください。

## スループットの計算

次のFIOコマンドを実行して、インスタンスにアタッチされているブロック・ボリュームのいずれかでランダム読取りスループットを計算します。

```
sudo fio --direct=1 --rw=randread --bs=64k --ioengine=libaio --iodepth=64
--runtime=120 --numjobs=4 --time_based --group_reporting --name=throughput-test-
job --eta-newline=1 --filename=/dev/oracleoci/oraclelvd
```

次に、コマンドの出力の概要の例を示します。最後の数行は、読取りスループットが340MB/sであることを示しています。コマンドは、17:44:17から17:46:17まで、120秒間実行されました。

```
throughput-test-job: (groupid=0, jobs=4): err= 0: pid=22474: Wed May 29 17:46:17
2019
  read: IOPS=5185, BW=324MiB/s (340MB/s) (37.0GiB/120058msec)
    slat (usec): min=3, max=73323, avg=757.32, stdev=4254.17
    clat (usec): min=693, max=430115, avg=48603.48, stdev=24652.88
      lat (usec): min=699, max=430122, avg=49362.02, stdev=25030.76
    clat percentiles (msec):
      | 1.00th= [ 12], 5.00th= [ 18], 10.00th= [ 22], 20.00th= [ 27],
      | 30.00th= [ 34], 40.00th= [ 41], 50.00th= [ 46], 60.00th= [ 52],
      | 70.00th= [ 57], 80.00th= [ 66], 90.00th= [ 81], 95.00th= [ 94],
      | 99.00th= [123], 99.50th= [140], 99.90th= [203], 99.95th= [222],
      | 99.99th= [279]
    bw (KiB/s): min=64512, max=126722, per=24.99%, avg=82916.28, stdev=8082.73,
samples=960
    iops: min= 1008, max= 1980, avg=1295.43, stdev=126.32, samples=960
    lat (usec): 750=0.01%, 1000=0.01%
    lat (msec): 2=0.03%, 4=0.08%, 10=0.56%, 20=7.67%, 50=49.47%
    lat (msec): 100=38.73%, 250=3.43%, 500=0.02%
    cpu: usr=0.54%, sys=1.40%, ctx=28118, majf=0, minf=2084
    IO depths: 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
      submit: 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
      complete: 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.1%, >=64=0.0%
      issued rwat: total=622531,0,0, short=0,0,0, dropped=0,0,0
      latency: target=0, window=0, percentile=100.00%, depth=64

Run status group 0 (all jobs):
  READ: bw=324MiB/s (340MB/s), 324MiB/s-324MiB/s (340MB/s-340MB/s), io=37.0GiB
(40.8GB), run=120058-120058msec
```

ここで、コンソールのブロック・ボリューム・メトリックをチェックして、どのように比較されるかを確認します。コンソールからメトリックを表示するには、次のステップに従います。

1. ナビゲーション・メニューを開きます。「コア・インフラストラクチャ」で「コンピューター」に移動し、「インスタンス」をクリックします。
2. インスタンスをクリックして、その詳細を表示します。
3. 「アタッチされたブロック・ボリューム」をクリックします。
4. ボリュームをクリックして、その詳細を表示します。
5. 「リソース」で「メトリック」をクリックします。
6. 「開始時間」に、FIOコマンドが実行を開始した時間と同じ時間を指定します。この例では、値は2019-05-29 17:44となります。
7. 「終了時間」に、2019-05-29 17:50と指定します。
8. ボリュームの読取りスループットについて、「間隔」に「1分」を、「統計」に「合計」を選択します。

結果は、図1のようになります。図2から4は、FIOテストからの読取りスループットの計算に使用する3つのデータ・ポイントを示しています。

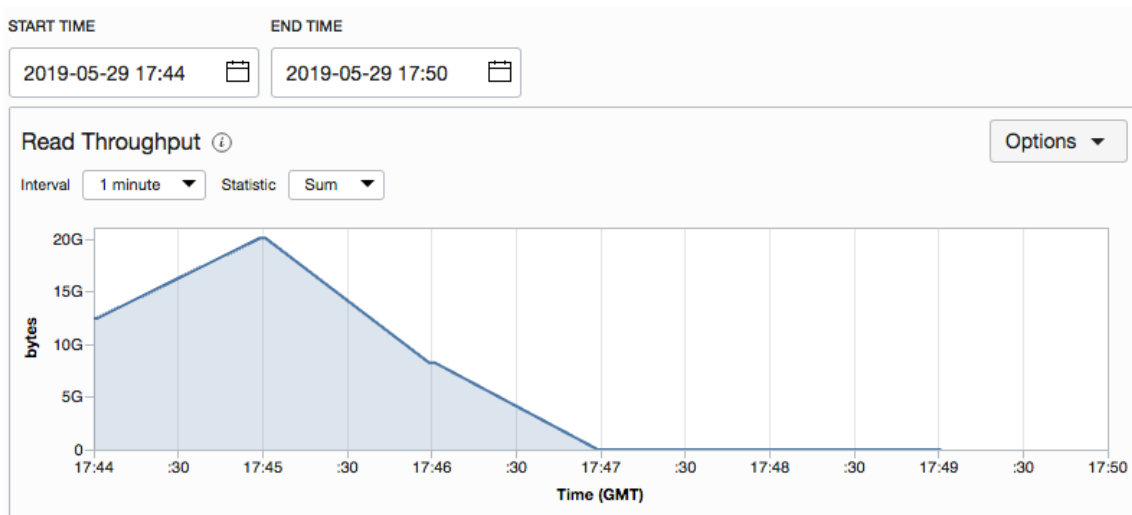


図1.2019-05-29の17:44から17:50のブロック・ボリュームの読取りスループット・メトリック

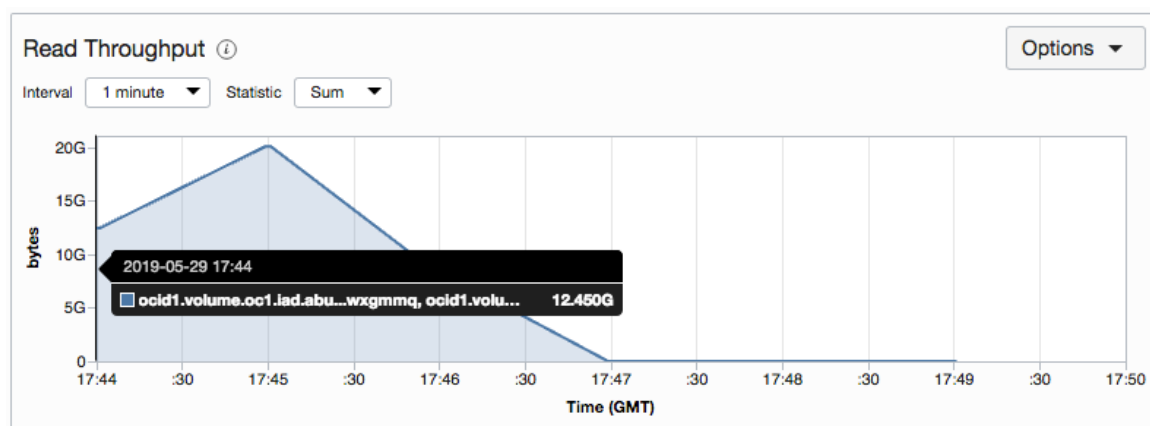


図2.2019-05-29 17:44の読取りスループット・メトリック・データ

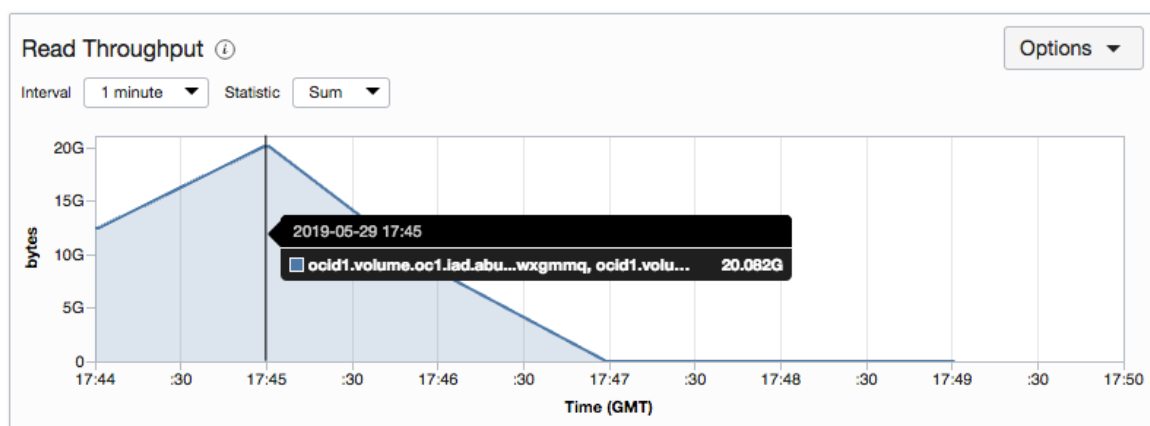


図3.2019-05-29 17:45の読取りスループット・メトリック・データ

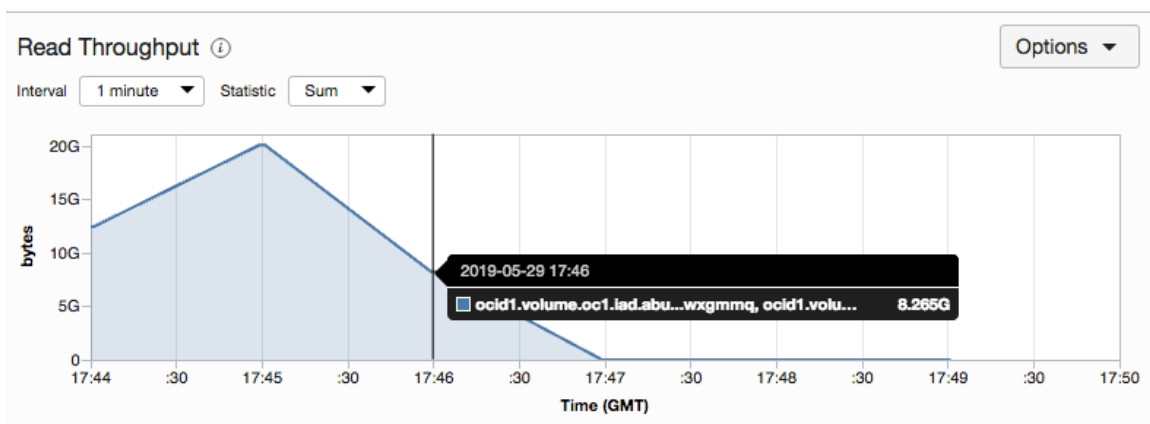


図4.2019-05-29 17:46の読取りスループット・メトリック・データ

3つのデータ・ポイントそれぞれに値を追加します。結果をテストの秒数(120)で除算し、次に1000を乗算し、スループットをMB/sで算出します。たとえば、次のようになります。

```
Throughput in MB/s = ((12.450 + 20.082 + 8.265) / 120) * 1000 = 339.975 MB/s  
~ 340 MB/s
```

## IOPSの計算

次のFIOコマンドを実行して、インスタンスにアタッチされているブロック・ボリュームのいずれかでランダム読取り/書込みIOPSを計算します。

```
sudo fio --direct=1 --rw=randrw --bs=4k --ioengine=libaio --iodepth=64 --  
runtime=120 --numjobs=4 --time_based --group_reporting --name=iops-test-job --  
eta-newline=1 --filename=/dev/oracleoci/oraclelvdb
```

次に、コマンドの出力の概要の例を示します。最後の数行は、読取りIOPSが12.5Kで、書込みIOPSが12.5Kであることを示しています。コマンドは、18:22:17から18:24:17まで、120秒間実行されました。

```
iops-test-job: (groupid=0, jobs=4): err= 0: pid=22927: Wed May 29 18:24:17 2019  
read: IOPS=12.5k, BW=48.9MiB/s (51.3MB/s) (5871MiB/120017msec)  
slat (usec): min=3, max=118179, avg=150.47, stdev=878.65  
clat (usec): min=392, max=163931, avg=6144.75,  
stdev=4087.94  
lat (usec): min=397, max=163936, avg=6296.45, stdev=4187.81  
clat percentiles (usec):  
| 1.00th= [ 1614], 5.00th= [ 2409], 10.00th= [ 3130], 20.00th= [ 3916],  
| 30.00th= [ 4621], 40.00th= [ 4883], 50.00th= [ 5211], 60.00th= [ 5932],  
| 70.00th= [ 6783], 80.00th= [ 7898], 90.00th= [ 9241], 95.00th= [ 10814],  
| 99.00th= [ 22414], 99.50th= [ 31327], 99.90th= [ 47973], 99.95th=  
[ 56886],  
| 99.99th= [100140]  
bw (KiB/s): min= 8248, max=14976, per=24.99%, avg=12520.27, stdev=938.78,  
samples=960  
iops: min= 2062, max= 3744, avg=3130.01, stdev=234.70, samples=960  
write: IOPS=12.5k, BW=48.9MiB/s (51.3MB/s) (5871MiB/120017msec)  
slat (usec): min=3, max=112365, avg=149.63, stdev=867.45  
clat (usec): min=478, max=179211, avg=13984.93, stdev=8679.88  
lat (usec): min=486, max=179217, avg=14135.79, stdev=8725.78  
clat percentiles (msec):  
| 1.00th= [ 3], 5.00th= [ 5], 10.00th= [ 6], 20.00th= [ 8],  
| 30.00th= [ 9], 40.00th= [ 11], 50.00th= [ 12], 60.00th= [ 15],  
| 70.00th= [ 18], 80.00th= [ 21], 90.00th= [ 25], 95.00th= [ 28],  
| 99.00th= [ 45], 99.50th= [ 53], 99.90th= [ 74], 99.95th= [ 89],  
| 99.99th= [ 130]  
bw (KiB/s): min= 8688, max=14856, per=24.99%, avg=12519.41, stdev=922.13,  
samples=960  
iops: min= 2172, max= 3714, avg=3129.80, stdev=230.54, samples=960  
lat (usec): 500=0.01%, 750=0.01%, 1000=0.02%  
lat (msec): 2=1.44%, 4=10.97%, 10=53.59%, 20=22.94%, 50=10.68%
```



```
lat (msec): 100=0.33%, 250=0.02%
cpu: usr=2.62%, sys=6.75%, ctx=163832, majf=0, minf=42
IO depths: 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit: 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete: 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.1%, >=64=0.0%
issued rwt: total=1503007,1503029,0, short=0,0,0, dropped=0,0,0
latency: target=0, window=0, percentile=100.00%, depth=64

Run status group 0 (all jobs):
  READ: bw=48.9MiB/s (51.3MB/s), 48.9MiB/s-48.9MiB/s (51.3MB/s-51.3MB/s),
io=5871MiB (6156MB), run=120017-120017msec
  WRITE: bw=48.9MiB/s (51.3MB/s), 48.9MiB/s-48.9MiB/s (51.3MB/s-51.3MB/s),
io=5871MiB (6156MB), run=120017-120017msec
```

ここで、コンソールのブロック・ボリューム・メトリックをチェックして、どのように比較されるかを確認します。コンソールからメトリックを表示するには、次のステップに従います。

1. ナビゲーション・メニューを開きます。「コア・インフラストラクチャ」で「コンピューター」に移動し、「インスタンス」をクリックします。
2. インスタンスをクリックして、その詳細を表示します。
3. 「アタッチされたブロック・ボリューム」をクリックします。
4. ボリュームをクリックして、その詳細を表示します。
5. 「リソース」で「メトリック」をクリックします。
6. 「開始時間」に、FIOコマンドが実行を開始した時間と同じ時間を指定します。  
この例では、値は2019-05-29 18:22となります。
7. 「終了時間」に、2019-05-29 18:26と指定します。
8. ボリュームの読取りIOPSについて、「間隔」に「1分」を、「統計」に「合計」を選択します。

結果は、図5のようになります。図6から8は、FIOテストからの読取りIOPSの計算に使用する3つのデータ・ポイントを示しています。

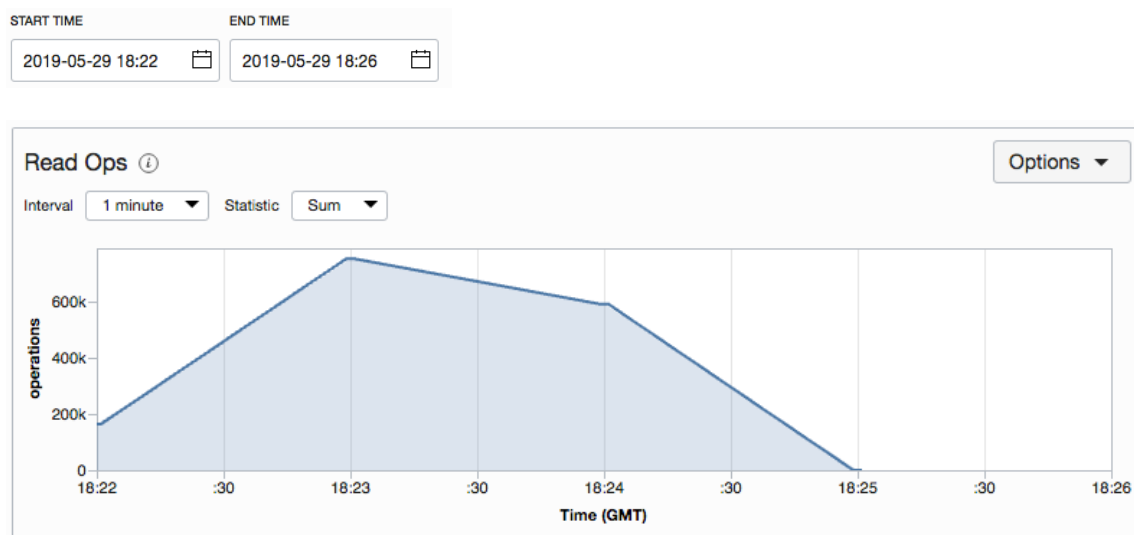


図5.2019-05-29の18:22から18:26のブロック・ボリュームの読取り操作メトリック

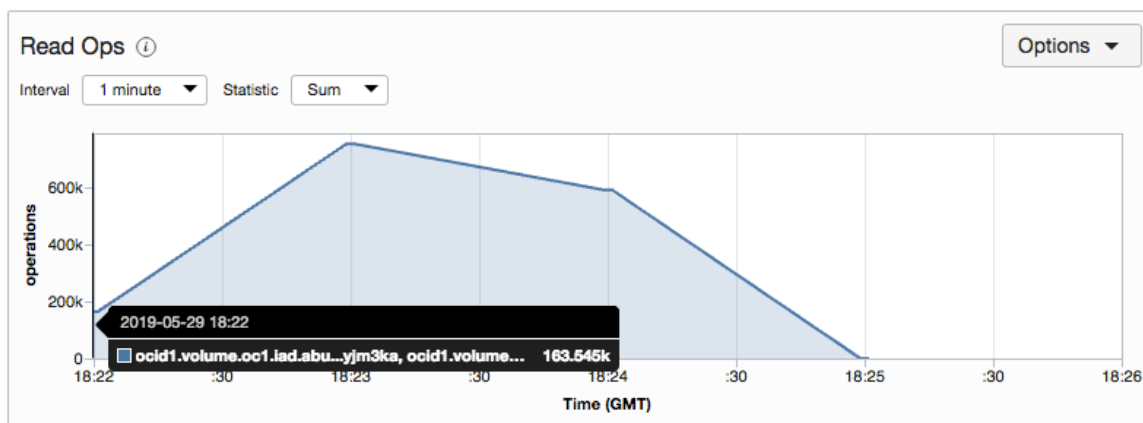


図6.2019-05-29 18:22の読取り操作メトリック・データ

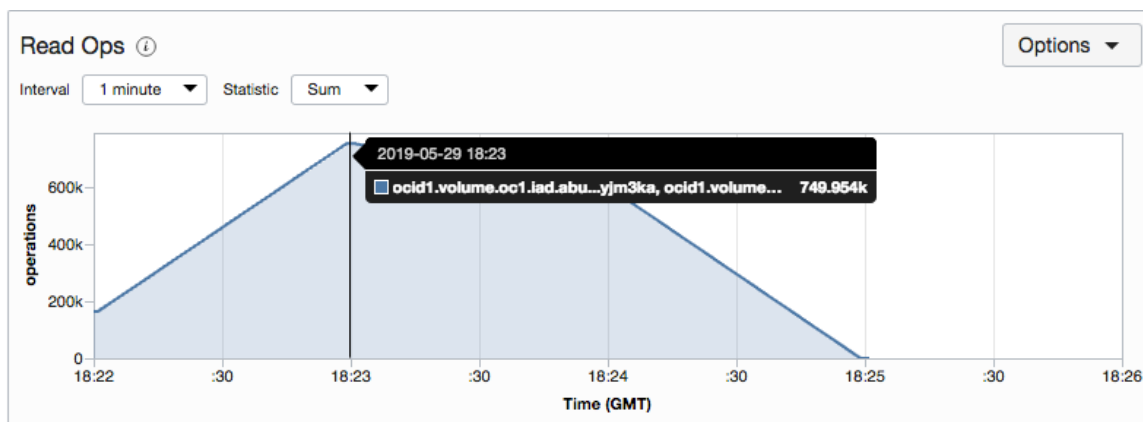


図7.2019-05-29 18:23の読取り操作メトリック・データ

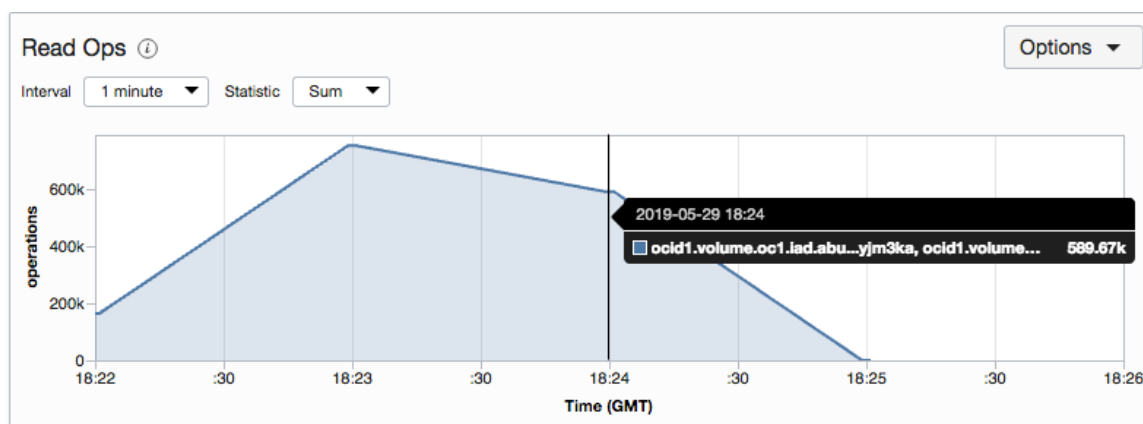


図8.2019-05-29 18:24の読取り操作メトリック・データ

3つのデータ・ポイントそれぞれに値を追加します。結果をテストの秒数(120)で除算し、次に1000を乗算し、読取りIOPS結果を算出します。たとえば、次のようになります。

```
Read IOPS = ((163.545k + 749.954k + 589.67k) / 120) * 1000 = 12.5K
```

書き込みIOPSを計算するための値を取得するには、ブロック・ボリュームのメトリックで同じ開始時間および終了時間の値を使用して、「ボリューム書き込み操作」メトリックに次を設定します。

- 「間隔」を「1分」に設定します。
- 「統計」を「合計」に設定します。

結果は、図9のようになります。図10から12は、FIOテストからの書き込みIOPSの計算に使用する3つのデータ・ポイントを示しています。

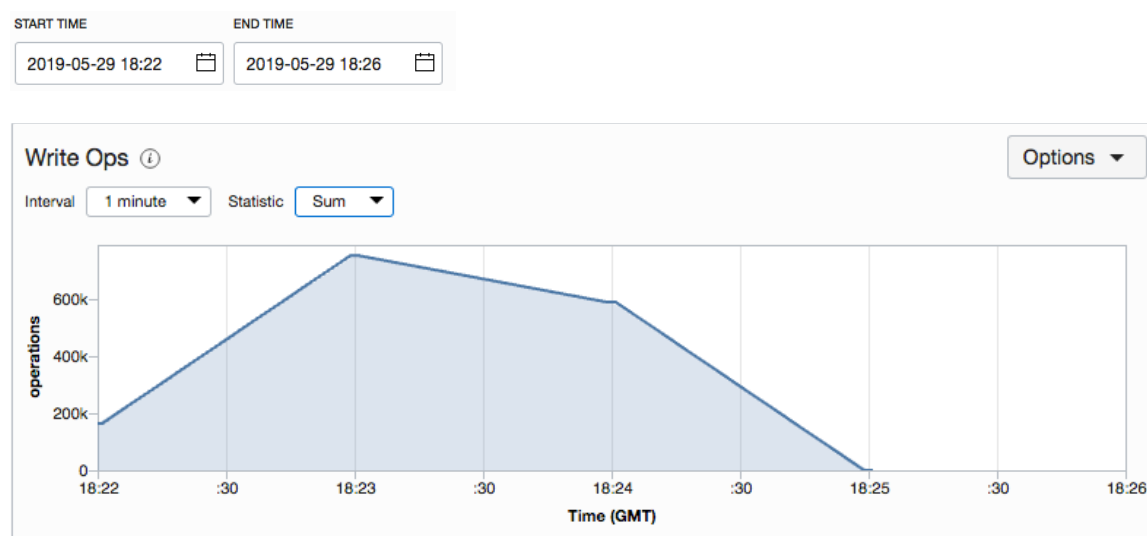


図9.2019-05-29の18:22から18:26のブロック・ボリュームの書き込み操作メトリック

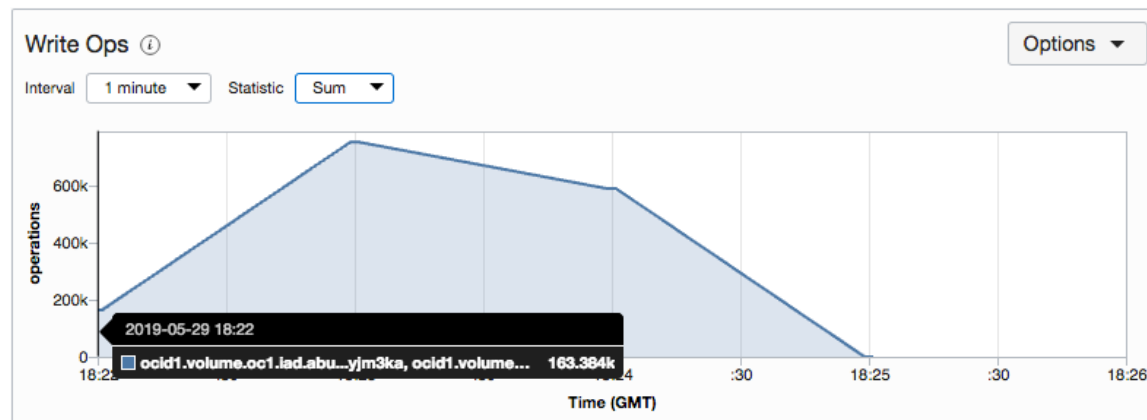


図10.2019-05-29 18:22の書き込み操作メトリック・データ

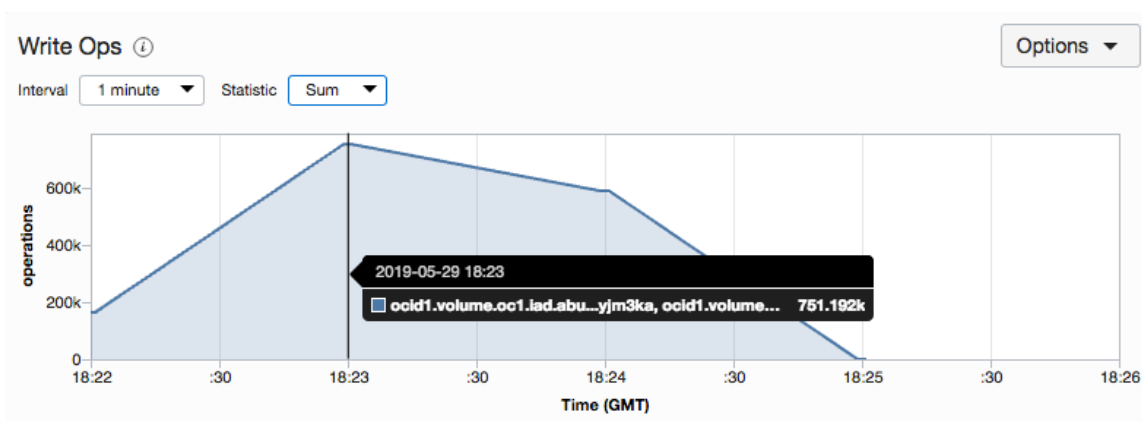


図11.2019-05-29 18:23の書き込み操作メトリック・データ

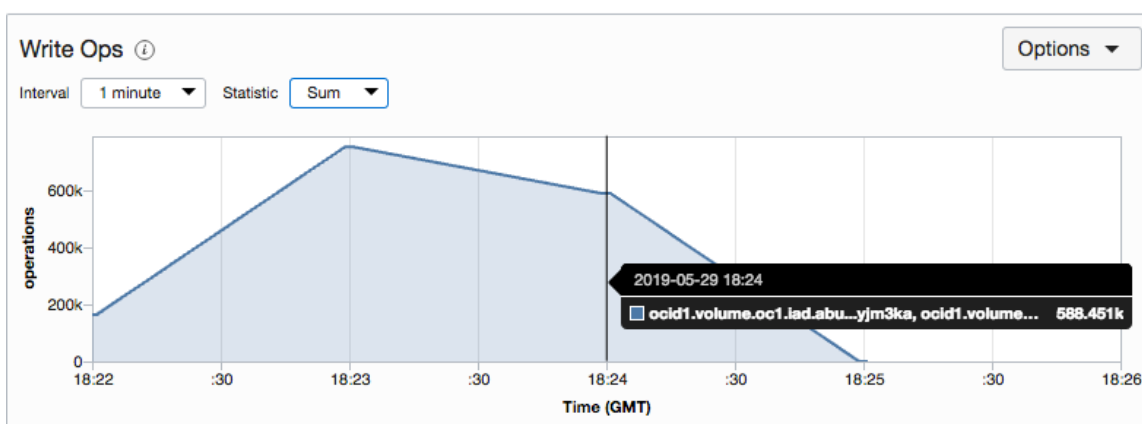


図12.2019-05-29 18:24の書き込み操作メトリック・データ

3つのデータ・ポイントそれぞれに値を追加します。結果をテストの秒数(120)で除算し、次に1000を乗算し、書き込みIOPS結果を算出します。たとえば、次のようになります。

```
Write IOPS = ((163.384k + 751.192k + 588.451k) / 120) * 1000 = 12.5K
```

## 結論

このホワイト・ペーパーでは、ブロック・ボリュームのパフォーマンス・メトリックをコンソールから取得し、その一貫性とパフォーマンスを確立する方法を説明しました。APIにより同じ情報を取得する方法の詳細は、[Block Volumesのドキュメント](#)を参照してください。



#### Oracle Corporation, World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065, USA

#### Worldwide Inquiries

Phone: +1.650.506.7000

Fax: +1.650.506.7200

#### CONNECT WITH US



[blogs.oracle.com/oracle](https://blogs.oracle.com/oracle)



[facebook.com/oracle](https://facebook.com/oracle)



[twitter.com/oracle](https://twitter.com/oracle)



[oracle.com](https://oracle.com)

### Integrated Cloud Applications & Platform Services

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否定し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel、Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

ブロック・ボリューム・サービス・メトリックを使用したブロック容量スループットおよびIOPSの計算

2019年6月

著者: Karan Vohra



Oracle is committed to developing practices and products that help protect the environment