

Article

[Tomohiro Iwamoto](#) · Jun 4, 2020 11m read

InterSystemsデータプラットフォームとパフォーマンス – パート2

前回の投稿では、pButtonsを使用してパフォーマンスメトリックを24時間収集する処理をスケジュールしました。この投稿では、収集対象の主なメトリックのいくつかと、それらの土台となるシステムハードウェアがどのように関連しているかを見ていきます。また、Cache（または任意のInterSystemsデータプラットフォーム）メトリックとシステムメトリックの関係についても調べます。さらに、これらのメトリックを使用してシステムの毎日の健全性を把握し、パフォーマンスの問題を診断する方法もご紹介します。

[このシリーズの他の記事のリストはこちら](#)

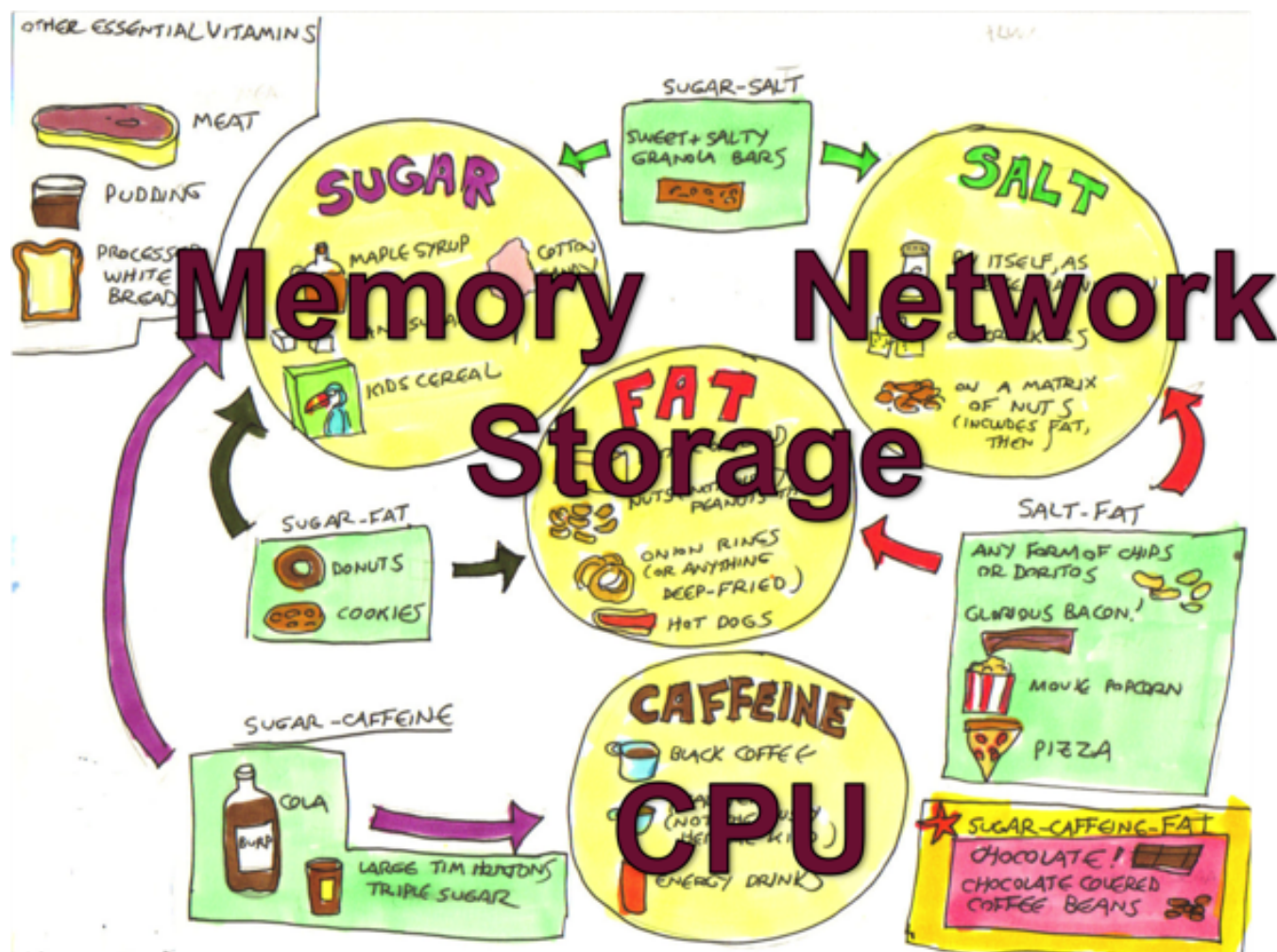
2016年10月に編集...

pButtonsデータを .csv ファイルに抽出するスクリプトの例はこちらで確認できます。

2018年3月に編集...

画像が消えていましたが、元に戻しました。

ハードウェア食品グループ



この連載を読み進めていくと、パフォーマンスに影響を与えるサーバーコンポーネントを次のように分類できることが分かってきます。

- CPU
- メモリ
- ストレージIO
- ネットワークIO

これらのコンポーネントのいずれかに負荷がかかっている場合、システムのパフォーマンスとユーザーエクスペリエンスが低下する可能性があります。これらのコンポーネントはすべて相互に関連しているため、1つのコンポーネントに対する変更が別のコンポーネントに影響する可能性があり、予期しない結果が生じることがあります。私はストレージアレイのIOボトルネックを解消するとCPU使用率が100%に跳ね上がり、ユーザーエクスペリエンスがさらに悪化した例を見たことがあります。これは、システムが突然自由に作業量を増やせるようになったものの、ユーザー活動とスループットの増加に対応するためのCPUリソースがなかったために発生したものです。

また、Cacheシステムの動作がサーバーコンポーネントにどのように直接影響を与えるかも説明します。ストレージのIOリソースが限られている場合はシステムメモリを増やし、Cacheグローバルのバッファに割り当てるメモリを増やすことで、好ましい変化をもたらすことが可能になります。これにより、システムストレージの読み取りIOを減らすことができます（ただし、CPU負荷が上がる可能性があります！）。

定期的に、あるいはユーザーが問題を報告したときに確認すべき最も分かりやすいシステムメトリックの1つにCPUの使用率が挙げられます。top や nmon(LinuxやAIXの場合)、またはWindowsパフォーマンスモニターを見てみましょう。ほとんどのシステム管理者はCPUデータを定期的に確認しているため、特にそのデータがグラフィカルに表示されている場合などは一目でシステムの現在の状態を把握できます。要はそのシステムが正常であるか、潜在的な異常動作によって突発的に負荷が上がっているか、あるいは問題が発生しているかということです。この投稿ではCPUメトリックをざっと確認しますが、Cacheメトリックを集中して取り上げます。まずは mgstat のデータを参照し、グラフィカルに表示したデータからシステムの正常性を一目で把握できることを確認します。

mgstatの概要

mgstatはpButtonsに含まれ、その中で実行されるCacheコマンドの1つです。mgstatは、システムの正常性を把握するのに役立つ基本的なパフォーマンスメトリックを収集するのに最適なツールです。この記事ではpButtonsを24時間実行して収集したmgstatのデータを確認しますが、pButtonの外部でデータを取り込みたい場合は、mgstatを必要に応じて対話形式で実行するか、Cacheターミナルからバックグラウンドジョブとして実行することができます。

オンデマンドで、%SYS名前空間からmgstatを実行するには、一般的に次の形式を使用します。

```
do mgstat(sample_time,number_of_samples,"/file_path/file.csv",page_length)
```

例えば、5秒のサンプル期間で1時間実行するバックグラウンドジョブを実行し、csvファイルに出力する場合は次のようになります。

```
job ^mgstat(5,720,"/data/mgstat_todays_date_and_time.csv")
```

一部のカラムを省略して結果を画面に表示する場合は、次のようにdsp132エントリを使用します。出力の違いについては、実際に皆さんで確かめてみてください。

```
do dsp132^mgstat(5,720,"",60)
```

mgstat の出力結果のカラムに関する詳細は、次のCachéの最新ドキュメントの「Caché監視ガイド」をご確認ください：

[InterSystemsオンラインドキュメント](#)

mgstatデータの参照

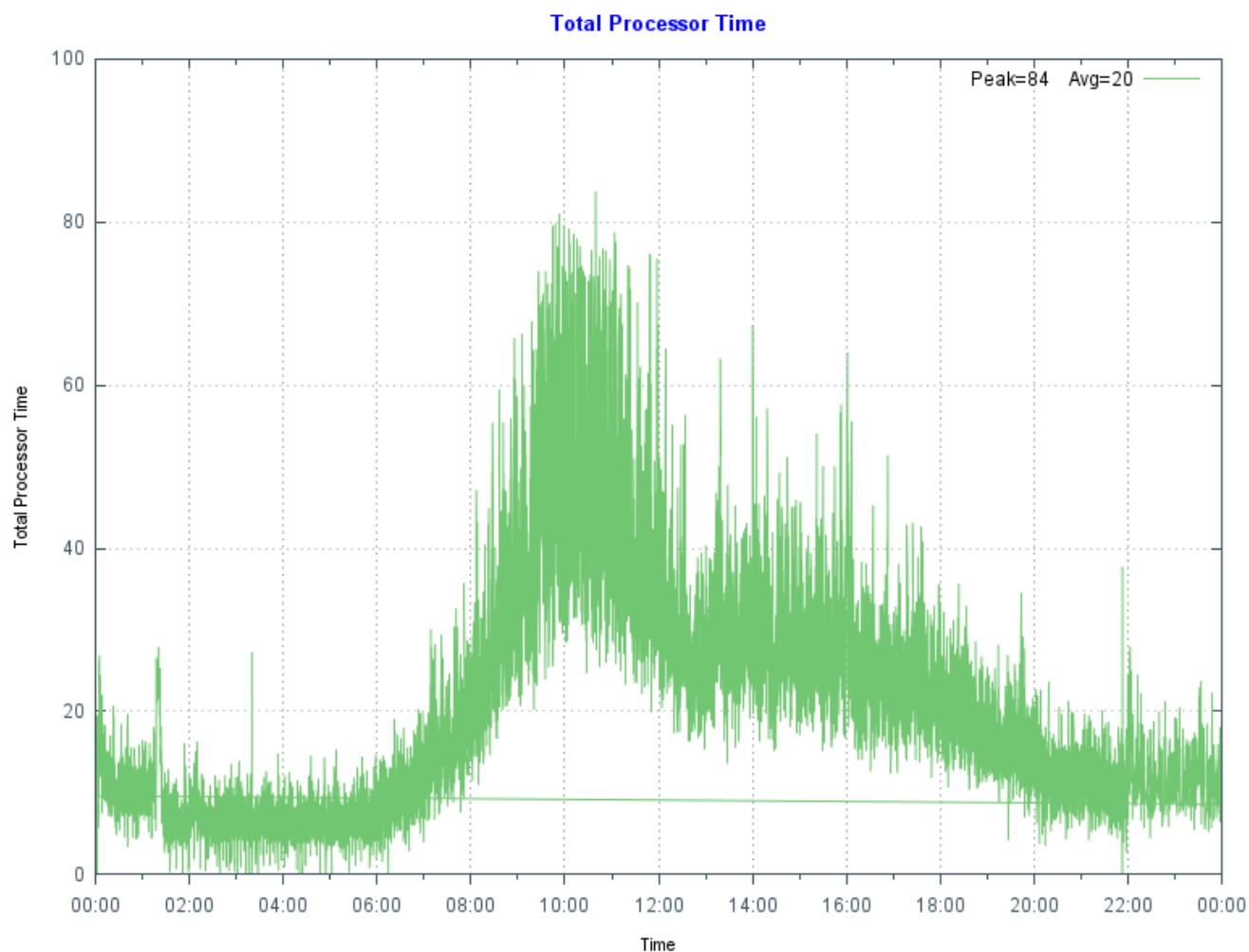
pButtonsは目的のデータを探しやすくし、パフォーマンスの問題を診断するWRCのサポート担当者に送信するためのパッケージ化を行うため、収集したデータを単一のHTMLファイルにまとめます。ただし、pButtonsを手動で実行してデータをグラフィカルに表示したい場合は、コマンドラインスクリプトや単純なカットアンドペーストを駆使してデータをcsvファイルに再分割し、Excelなどでグラフ化することができます。

この投稿では、mgstatメトリックのいくつかを掘り下げ、データを一目見ただけでシステムのパフォーマンスが優れているか、ユーザーエクスペリエンスに影響を与える発生中の問題や潜在的な問題が存在するかどうかを確認できることをご説明します。

GlorefsとCPU

次のグラフは、病院向けアプリケーションを高い処理頻度で実行しているサイトのデータベースサーバーのCPU使用率を示しています。多くの外来診療所では朝に活動のピークが訪れ、使用率は昼休みに急激に下がり、その後午後から夕方にかけて次第に低下していることが分かります。この例では、Windowsパフォーマンスモニターの「(Total)% Processor Time」（グラフの形状が稼働日の分布測定に適しています）からデータを取得しました。異常な山や谷が見つからないため、このサイトではこのデータは正常であると考えられます。

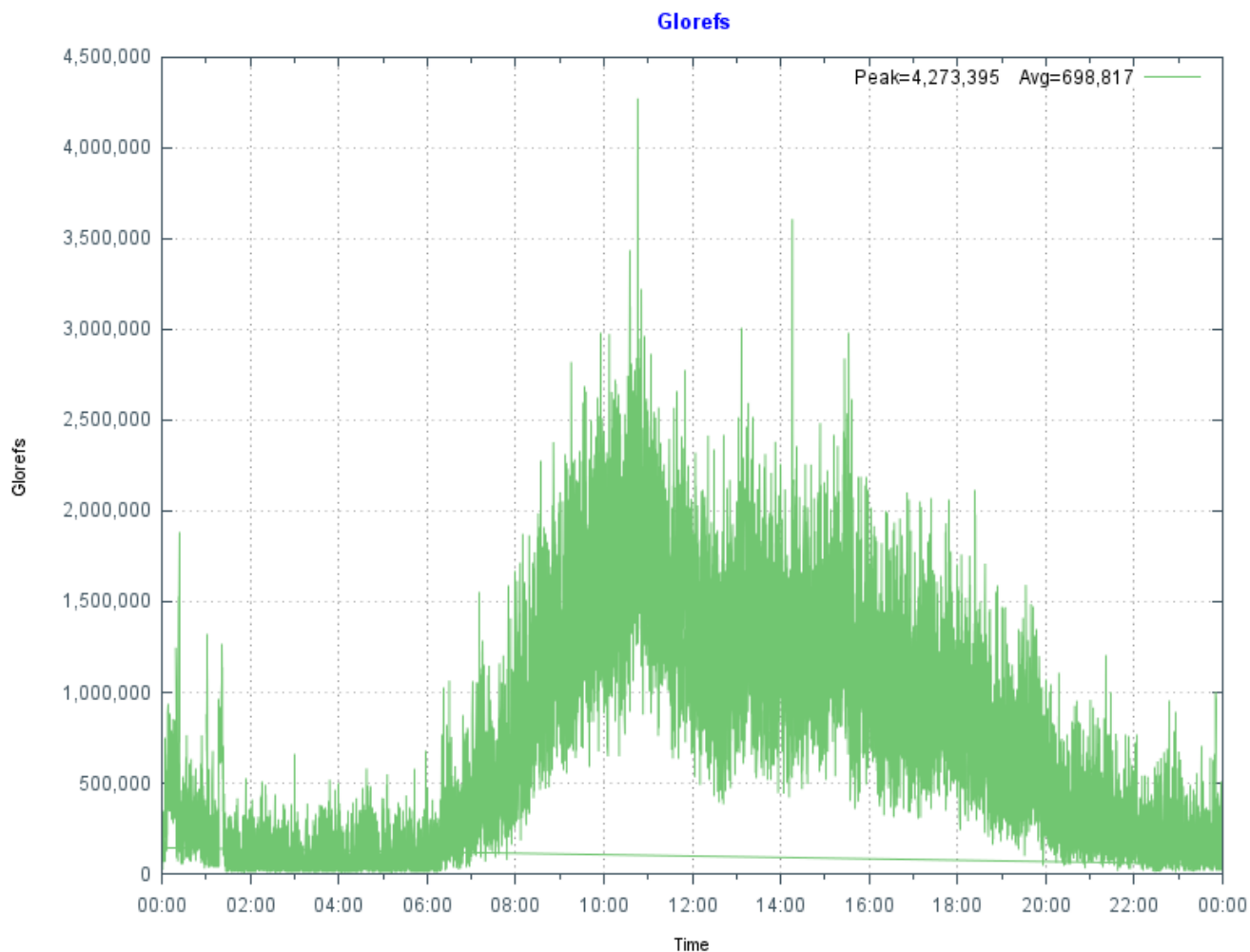
皆さんもご自身のサイトに対して同じことを行えば、「正常時」のベースラインを取得することができます。大幅な急上昇、特に異常な急上昇は問題が存在することを示している可能性があります。今後の投稿では、CPUに注目します。



参考までに、このデータベースサーバーは2つのE5-2670 8コアプロセッサを搭載したDell R720であり、サーバーには128 GBのメモリが搭載され、48 GBのグローバルバッファが割り当てられています。

次のグラフでは、CPUのグラフと同じ日のmgstat —

Glorefs (グローバル参照) またはデータベースアクセス数のデータをより詳細に示しています。Glorefsはその時点の負荷を代表するものとして、発生している作業量を示します。グローバルが参照されるとCPU時間が消費されますが、Cacheはグローバルメモリバッファプールを使用しているため、物理的な読み取り操作などの、他のシステムリソースが消費されるとは限りません。



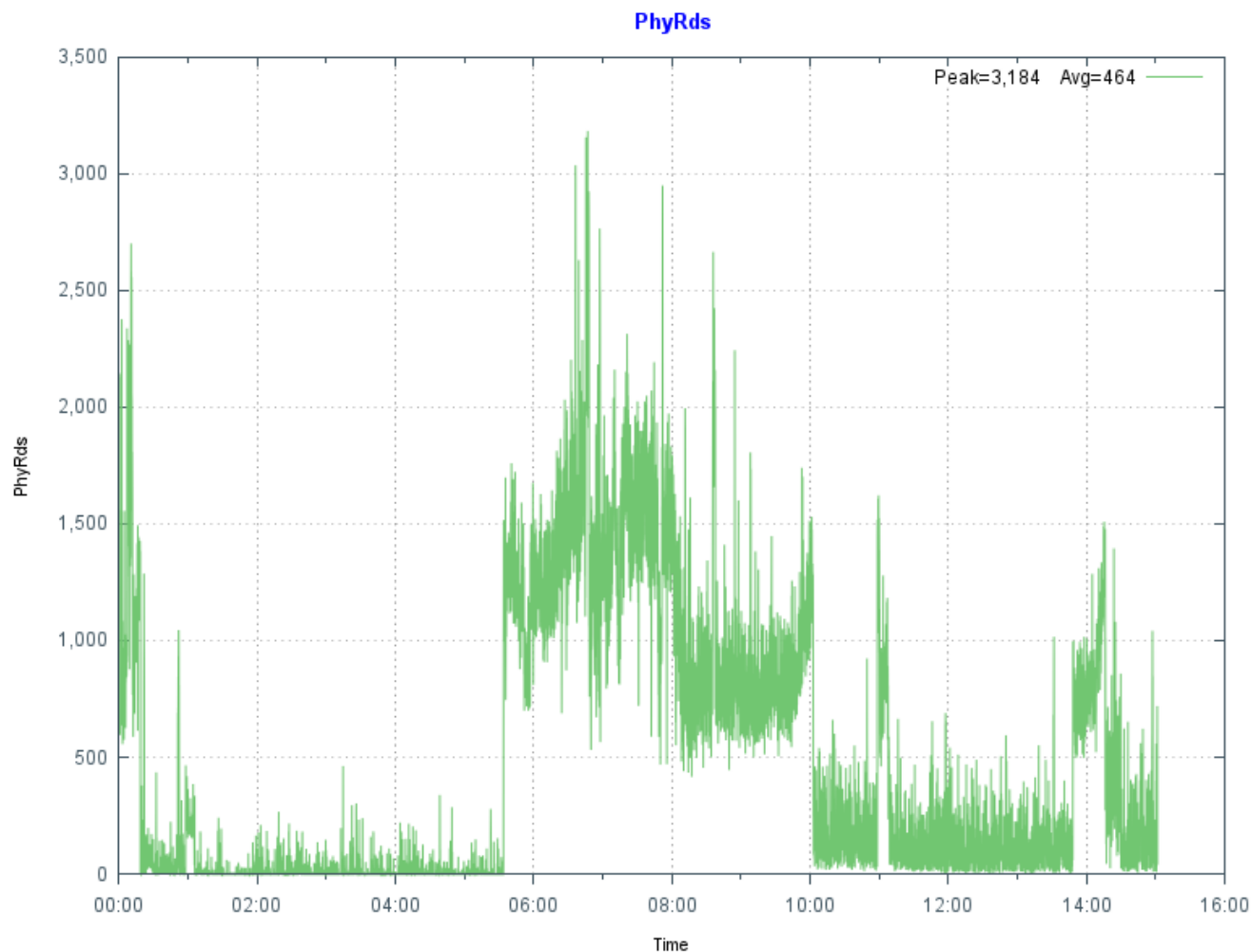
一般的なCacheアプリケーションでは、GlorefとCPU使用率の間に非常に強い相関があります。

このCPUとglorefのデータを別の視点から見た場合、glorefsを減らすとCPU使用率が減り、コア数の少ないサーバーにデプロイしたり、既存システムをさらに拡張したりできるようになるということが分かります。アプリケーションをさらに効率化することで、グローバル参照を削減する方法もありそうです。この考えは後の投稿で再検討します。

PhyRdsとRdratio

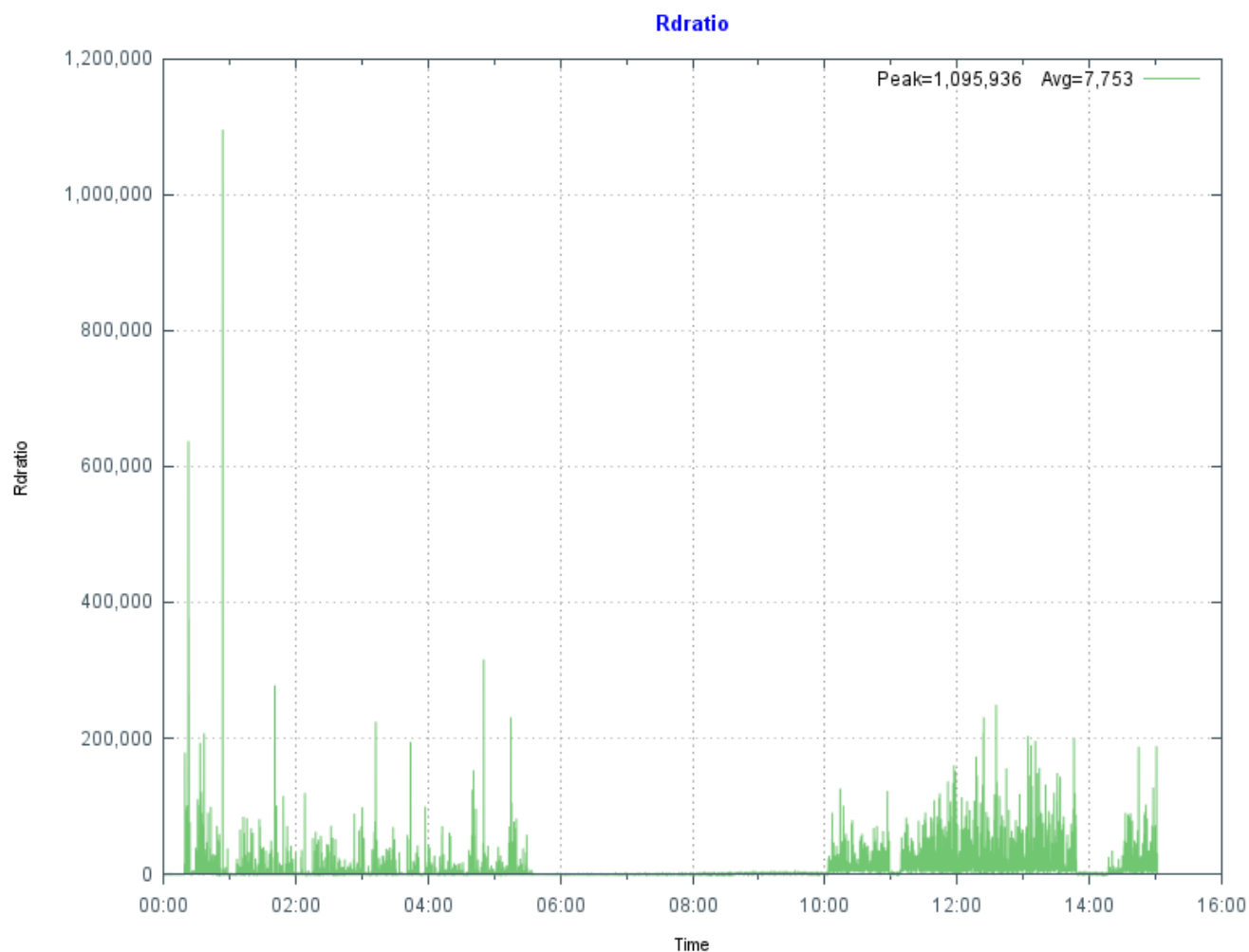
mgstatのデータをグラフ化したPhyRds（物理的読み込み）とRdratio（読み込み比率）はシステムパフォーマンスから予想される詳細な情報を把握し、キャパシティプランニングを立てるのに役立ちます。今後の投稿では、CacheのストレージIOについて詳しく説明します。

PhyRdsはディスクからCacheデータベースへの単純な物理読み取りIOPS（I/O毎秒）で、論理ディスクと物理ディスクに対応するオペレーティングシステムのメトリックに反映されているものと同じ値が表示されます。ただし、オペレーティングシステムのIOPSにはCache以外のアプリケーションに起因するIOPSも表示されている可能性があります。予想されるIOPSを考慮せずにストレージのサイジングを行うと確実に失敗することになります。適切なキャパシティプランニングを行うには、システムがピークに達している状態でのIOPSを知る必要があります。次のグラフは、午前0:00から15:30までのPhyRdsを示しています。



05:30から10:00の間に物理的読み込みが大幅に増加していることに注目してください。その他には11:00と14:00になる直前に短めのピークが発生しています。これらの原因は何だと思われますか？皆さんのサーバーではこのようなピークが見られますか？

Rdratioからは、より興味深い結果を得ることができます。このグラフは物理ブロックの読み取りに対する論理ブロックの読み取りの比率を表しています。つまり、メモリからのグローバルバッファ（論理ブロック）からの読み取り回数と、桁違いに遅いディスクからの読み取り回数の比率を表しています。Rdratioが高いほうが望ましく、長時間にわたってほぼゼロ近くまで低下するのは望ましくありません。



読み込みが多い時間帯に合わせてRdratioがゼロ近くに低下していることが分かります。私はこのサイトについて、システムが長時間にわたって重くなっているとの電話連絡がユーザーからIT部門に入り始めたため、調査するように依頼されました。

私がシステムの調査を依頼された最初の数週間は、この現象はランダムに発生しているように思われました。

しかし、私は比較的簡単にこのような高いPhyRdsと低いRdratioのパターンとサポートへの問い合わせ回数に相関関係があることを発見しました。pButtonsが毎日24時間実行されるようにスケジュールされていたおかげで、数週間前のデータに遡って調査することができたからです。

さらなる分析を続けた結果、新しいシフト勤務者が適切なインデックスを使用しない劣悪なクエリと「不正な」パラメーターを組み合わせで入力するレポートをいくつか実行し、データベースの読み込みを増やしていたことを突き止めました。それが、一見ランダムに見える遅延の原因だったのです。これらの実行時間の長いレポートはデータをグローバルバッファに読み込んでいるため、対話ユーザーのデータはメモリではなく物理ストレージから取得されています。その結果、読み取りを処理するストレージに負荷がかかっています。

PhyRdsとRdratioを監視すれば、システムの健全性に関する情報を取得できます。また、劣悪なレポートやクエリが存在を突き止めることもできるかもしれません。高いPhyRdsには妥当な原因があると思われます。そのような場合は、レポートを決まった時間に実行する必要があるでしょう。物理メモリ容量が大きい最新の64ビットオペレーティングシステムとサーバーを使えば、本番システムのPhyRdsを最小化できるはずです。

お使いのシステムでPhyRdsが高い値を示している場合は、次のようないくつかの対策を検討することができます。

- データベースの（グローバル）バッファ数（およびシステムメモリ）を増やしてパフォーマンスを向上さ

せる。

- 実行時間の長いレポートや抽出処理を営業時間外に実行するようにする。
- 実行時間の長い読み取り専用レポート、バッチジョブ、データ抽出処理を別のシャドウサーバーか非同期ミラーで実行し、対話ユーザーへの影響を最小限に抑え、CPUやIOPSなどのシステムリソースの利用負荷をオフロードする。

一般的に低いPhyRdsは望ましく、それはシステムをサイジングする際の目標でもあります。ただし、PhyRdsが低くてもユーザーがパフォーマンスに不満を抱いている場合は、ストレージがボトルネックではないことを確かめるため、他にもチェックできることがあります。システムが必要なサービスを提供できなくなっていることが原因で、読み取りが少なくなっている可能性があります。ストレージについては、今後の記事で詳しく見ていきます。

概要

この投稿では、pButtonsで収集されたメトリックをグラフ化し、システムの健全性を一目で確認する方法について説明しました。今後の投稿では、システムメトリックとCacheメトリックの関係に加えて、これらを使用して将来の計画を立てる方法について詳しく説明します。

[#System Administration](#) [#Performance](#) [#Cache](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

Source URL: <https://community.intersystems.com/node/477711>