

Article

[Tomohiro Iwamoto](#) · Jun 3, 2020 12m read

InterSystemsデータプラットフォームとパフォーマンス – パート1

アプリケーションがデプロイされ、すべてが問題なく動作しています。素晴らしいですね！しかし、その後突然電話が鳴り止まなくなりました。アプリケーションが時々「遅くなる」というユーザーからの苦情の電話です。これは一体どういうことなのでしょう？なぜ時々遅くなるのでしょうか？

このような速度低下を検出し、解決するにはどのようなツールと統計情報に注目すべきなのでしょう？

お使いのシステムのインフラはユーザーの負荷に対応できていますか？

本番環境を調べる前に、どのようなインフラ設計上の問題を問うべきなのでしょう？新しいハードウェアのキャパシティプランニングを、必要以上の設備投資を行うことなく自信を持って行うにはどうすればよいのでしょうか？どうすれば電話がかかってこなくなるのでしょうか？

そもそも電話がかかってこないようにするには、どうすればよかったのでしょうか？

[このシリーズの他の記事のリストはこちら](#)

長い道のりの始まり

これは、システムパフォーマンスの監視、確認、トラブルシューティングに使用できるツールとメトリック、およびパフォーマンスに影響を与えるシステムとアーキテクチャの設計上の考慮事項について説明する連載の最初の投稿です。途中、Caché、オペレーティングシステム、ハードウェア、仮想化、およびコメントのフィードバックで話題になっているその他の領域のパフォーマンスを理解するため、本題からそれる場面がかなりあります。

フィードバックグループに従い、パフォーマンスデータからデプロイ済みのアプリケーションとインフラのメリットと制限を詳しく確認し、その後を選びすぐれた設計とキャパシティプランニングに戻ります。

言うまでもなく、常にパフォーマンスメトリックを確認する必要があります。データに目を向けてさえいれば、ずっと前から分かっていたはずのパフォーマンスの問題に顧客が驚いているのは残念なことです。しかし、当然ながらここではどのデータが原因なのかということが問題です。まずは現在のシステムの健全性を感じ取れるよう、いくつかの基本的なCachéとシステムのメトリックを収集することから始めましょう。後の投稿では、主なメトリックの意味について詳しく説明します。

システム監視には、Cachéの内外から利用できる多くのオプションがあります。この連載では、それらの多くについて説明します。

まずは、すべてのCachéシステムに最初からインストールされている継続的データ収集ツールである、^pButtonsを見てみましょう。

次の投稿を確認し、pButtonsの最新コピーがあることを確認してください。

<https://community.intersystems.com/post/intersystems-data-platforms-and-performance-%E2%80%93-how-update-pbuttons>

システムパフォーマンスメトリックの収集 - ^pButtons

Caché pButtonsユーティリティは、作成したログファイルから読み取り可能なHTMLパフォーマンスレポートを生成します。pButtonsから出力されるパフォーマンスメトリックは、簡単に抽出、グラフ化、確認できます。

pButtons のhtmlファイルに収集されるデータには、次のものがあります。

- Caché の設定：構成、ドライブの割り当てなど。
- mgstat：Cachéのパフォーマンスメトリック - ほとんどの値は1秒あたりの平均です。
- Unix：vmstatとiostat：オペレーティングシステムのリソースとパフォーマンスに関するメトリック。
- Windows：パフォーマンスモニター：Windowsのリソースとパフォーマンスに関するメトリック。
- その他の有用なメトリック。

pButtons によるデータ収集はシステムのパフォーマンスにほとんど影響を与えません。メトリックはすでにシステムによって収集されており、pButtonsは単にこれらをパッケージ化し、整理して転送しやすくするだけです。

ベースラインを維持し、傾向分析と問題解決に役立てるため、業務サイクル全体を通して毎日pButtonで24時間（午前0時から午前0時）のデータを収集することをお勧めします。

例えば、月末処理からデータを取得する場合など、業務サイクルが1カ月以上になる場合もあります。他に外部パフォーマンス監視システムや収集システムを使用していない場合は、年間を通じてpButtonsを実行できます。

次の重要な点を考慮する必要があります。

- ディスク容量がいっぱいにならないようにするため、ログディレクトリを本番データとは異なる場所に變更し、蓄積した出力ファイルをそちらに保管するようにしてください！
- オペレーティングシステムのスクリプトを実行するか、定期的にpButtonsのファイルを圧縮してアーカイブしてください。このファイルは肥大化する可能性があるため、これはWindowsでは特に重要です。
- 定期的にデータを確認してください！

すぐに分析が必要な問題が発生した場合は、pButtonsのデータをプレビューすることができます（データはすぐに収集されます）が、メトリックは引き続き継続的に保存され、1日の終わりに収集されます。

プレビュー、実行の停止、独自のデータ収集の追加など、pButtonの詳細については、最新のCachéドキュメントの「Cachéモニタリングガイド」を参照してください。

<http://docs.intersystems.com>

pButtonsのHTMLファイルデータは（CVSファイルなどに）分離・抽出可能で、スクリプトを作成したり、単にカットアンドペーストしたりして、グラフの描画やその他の分析に利用することができます。次の投稿の後半では、出力例をグラフで確認します。

もちろん、パフォーマンスに緊急の問題がある場合は、WRCにご連絡ください。

pButtonsによる24時間のデータ収集をスケジュールする

^pButtonsはターミナルプロンプトから手動で開始するか、スケジュールすることができます。

毎日24時間の収集をスケジュールするには、以下の手順に従ってください。

1. pButtonsのファイル構造を準備するため、Cachéターミナルを起動して%SYSネームスペースに切り替え、pButtonsを手動で1回実行します。

```
%SYS>d ^pButtons
Current log directory: /db/backup/benchout/pButtonsOut/
Available profiles:
  1  12hours      - 12 hour run sampling every 10 seconds
  2  24hours      - 24 hour run sampling every 10 seconds
  3  30mins       - 30 minute run sampling every 1 second
  4  4hours       - 4 hour run sampling every 5 seconds
  5  8hours       - 8 hour run sampling every 10 seconds
  6  test         - A 5 minute TEST run sampling every 30 seconds
```

テストを行うには6番のオプションを選択してください。30秒ごとにサンプリングする5分間のテスト実行です。この番号は異なる場合がありますが、"test"というわかりやすい項目名になっています。

テスト中にCollect^pButtonsを実行すると(以下に例を示します)、runidを含む情報が表示されます。この場合は、「20160303_1851_test」です。

```
%SYS>d Collect^pButtons
Current Performance runs:
    20160303_1851_test      ready in 6 minutes 48 seconds nothing available to collect at the moment.
%SYS>
```

この5分間の実行が終わるまで、6分48秒かかることに注意してください。ログを収集してhtml形式にまとめる時間を確保するため、pButtonsでは実行のたびに2分の猶予期間が追加されます。

2. 重要！ pButtonsのログ出力ディレクトリを変更してください。デフォルトの出力先は <cache install path>/mgr フォルダーです。UNIXの場合、ログディレクトリのパスは例えば次のようになります。

```
do setlogdir^pButtons("/somewhere_with_lots_of_space/perflogs/")
```

Cachéにディレクトリへの書き込み権限があり、出力ファイルの蓄積に必要なディスク容量が確保されていることを確認してください。

3. 次のコマンドを実行し、30秒間隔で新しい24時間プロファイルを作成します。

```
write $$addprofile^pButtons("My_24hours_30sec","24 hours 30 sec interval",30,2880)
```

pButtonsにプロファイルが追加されたことを確認してください。

```
%SYS>d ^pButtons
Current log directory: /db/backup/benchout/pButtonsOut/
Available profiles:
  1 12hours      - 12 hour run sampling every 10 seconds
  2 24hours      - 24 hour run sampling every 10 seconds
  3 30mins       - 30 minute run sampling every 1 second
  4 4hours       - 4 hour run sampling every 5 seconds
  5 8hours       - 8 hour run sampling every 10 seconds
  6 My_24hours_30sec- 24 hours 30 sec interval
  7 test        - A 5 minute TEST run sampling every 30 seconds
```

select profile number to run:

注意：収集間隔は変更できます。通常の監視では30秒で十分です。pButtonsは間隔ごとにデータを収集するため、出力ファイルが非常に大きくなる可能性があります。そのため、24時間定期的に行う場合は5秒（... " ,5,17280）未満の間隔を指定しないことをお勧めします。特定の時刻を対象に問題を解決しようとしており、より詳細なデータが必要な場合は、デフォルトプロファイルのいずれかを使用するか、5秒間隔で1時間（... " ,5,720）といったより短い期間で新しいカスタムプロファイルを作成してください。pButtonsは複数同時に実行できるため、24時間のpButtonsと同時に5秒間隔の短いpButtonsを実行することができます。

4. ヒント UNIXサイトの場合は、diskコマンドを確認してください。「iostat」コマンドで使用するデフォルトのパラメーターには、ディスクの応答時間が含まれていない場合があります。最初に、現在設定されているディスクコマンドを表示します。

```
%SYS>zw ^pButtons("cmds","disk")
^pButtons("cmds","disk")=2
^pButtons("cmds","disk",1)=$lb("iostat","iostat ","interval"," ","count"," > ")
```

```
^pButtons("cmds","disk",2)=$lb("sar -d","sar -d ","interval"," ","count"," > ")
```

ディスク統計を収集するには、適切なコマンドを使用し、運用中のUNIXに合わせて構文を編集してください。末尾のスペースに注意してください。以下にいくつかの例を示します。

```
LINUX:      set $li(^pButtons("cmds","disk",1),2)="iostat -xt "  
AIX:        set $li(^pButtons("cmds","disk",1),2)="iostat -sadD "  
VxFS:       set ^pButtons("cmds","disk",3)=$lb("vxstat","vxstat -g DISKGROUP -i ","interval"," -c ","count"," > ")
```

iostatコマンドとsarコマンドの両方を実行すると、非常に大きなpButton htmlファイルを作成できます。筆者の場合、通常のパフォーマンスレビューではiostatのみを使用しています。コマンドを1つだけ設定するには、以下のように記述します。

```
set ^pButtons("cmds","disk")=1
```

pButtonsの設定に関する詳細は、オンラインドキュメントを参照してください。

5. Management Portal > System Operation > Task Managerで、pButtonsを真夜中に起動するようにスケジュールします。

```
Namespace: %SYS  
Task Type: RunLegacyTask  
ExecuteCode: Do run^pButtons("My_24hours_30sec")  
Task Priority: Normal  
User: superuser  
How often: Once daily at 00:00:01
```

pButtonsデータの収集

InterSystemsデータプラットフォームのより新しいバージョンのpButtonsには、自動収集機能があります。データを手動で収集してhtmlファイルにまとめるには、%SYSネームスペースで次のコマンドを実行し、未処理のpButtons html出力ファイルを生成します。

```
do Collect^pButtons
```

htmlファイルは、手順2で設定したlogdirにあります（設定していない場合は、今すぐ設定してください！）。設定していない場合、デフォルトの場所は <Cache install dir/mgr> になります。

ファイル名は <hostnameinstanceNamedatetimeprofileName.html> になります（例：vsan-tc-db1H2015201602180255test.html）。

Windowsパフォーマンス 모니터の考慮事項

オペレーティングシステムがWindowsの場合、Windowsパフォーマンスモニター(perfmon)を使用して、収集対象の他のメトリックと同期してデータを収集できます。pButtonsの古いCacheディストリビューションでは、Windowsパフォーマンスモニターを手動で構成する必要があります。この投稿のコメントで要望があれば、パフォーマンスカウンターを定義してpButtonsと同じ期間と間隔で監視とスケジュール実行を行うためのパフォーマンスモニター用のテンプレートを作成する方法について記事を書きます。

概要

この投稿により、考察すべきデータの収集が始まりました。
週の後半には、サンプルデータとその意味を見ていきます。
皆さんは自分のシステムで収集したデータを追跡できます。 それではまた。

<http://docs.intersystems.com>

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

Source URL: <https://community.intersystems.com/node/477621>