

Artigo

[Murray Oldfield](#) · Dez. 8, 2020 8min de leitura

Plataformas de Dados InterSystems e desempenho - Parte 1

Sua aplicação está implantada e tudo está funcionando bem. Ótimo, bate aqui! Então, do nada, o telefone começa a tocar sem parar – são os usuários reclamando que, às vezes, a aplicação está "lenta". Mas o que isso significa? Às vezes? Quais ferramentas você tem e quais estatísticas você deve examinar para encontrar e resolver essa lentidão? A infraestrutura do seu sistema está à altura da tarefa de carga do usuário? Que perguntas de design de infraestrutura você deveria ter feito antes de entrar em produção? Como você pode planejar a capacidade de um novo hardware com confiança sem excesso de especificações? Como você pode parar o telefone de tocar? Como você poderia ter impedido o telefone de tocar em primeiro lugar?

[Veja aqui uma lista das outras postagens desta série](#)

Esta será uma jornada

Este é a primeira postagem de uma série que explorará as ferramentas e métricas disponíveis para monitorar, revisar e solucionar problemas de desempenho de sistemas, bem como considerações de design e arquitetura de sistema que afetam o desempenho. Ao longo do caminho, iremos percorrer algumas trilhas para entender o desempenho do Caché, sistemas operacionais, hardware, virtualização e outras áreas que se tornam tópicos a partir de seu feedback nos comentários.

Seguiremos o ciclo de feedback fornecido pelos dados de desempenho para visualizar as vantagens e limitações das aplicações e da infraestrutura implantadas e, em seguida, voltar para uma melhor concepção e planejamento de capacidade.

Não é preciso dizer que você deve revisar as métricas de desempenho constantemente. É lamentável o número de vezes que os clientes são surpreendidos por problemas de desempenho que estavam visíveis por um longo tempo, se eles tivessem olhando os dados. Mas, é claro, que a questão é: quais dados? Começaremos a jornada coletando algumas métricas básicas do Caché e do sistema para que possamos ter uma ideia da integridade do seu sistema atual. Em postagens posteriores, mergulharemos no significado das principais métricas.

Há muitas opções disponíveis para o monitoramento do sistema – internamente do Caché e externamente. Vamos explorar um monte deles nesta série.

Para começar, veremos minha ferramenta favorita para coleta contínua de dados que já está instalada em todos os sistemas Caché – o pButtons.

Para ter certeza de que você tem a última cópia do pButtons, reveja a seguinte postagem:

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

Coletando métricas de desempenho do sistema – pButtons

O utilitário Caché pButtons gera um relatório de desempenho em HTML legível a partir dos arquivos de log que ele cria. A saída das métricas de desempenho pelo pButtons pode ser facilmente extraída, mapeada e revisada.

Os dados coletados no arquivo HTML pelo pButtons incluem:

- Configuração do Caché: com configuração, mapeamentos de unidades, etc.
- mgstat: Métricas de desempenho do Caché – a maioria dos valores são médios por segundo.
- Unix: vmstat e iostat – recursos do sistema operacional e métricas de desempenho.
- Windows: monitor de desempenho – recursos do Windows e métricas de desempenho.
- Outras métricas que serão úteis.

A coleta de dados do pButtons tem muito pouco impacto no desempenho do sistema, as métricas já estão sendo coletadas pelo sistema, o pButtons simplesmente as empacota para facilitar o arquivamento e transporte.

Para manter uma linha de base, para análise de tendências e solução de problemas, é uma boa prática realizar uma coleta com o pButtons de 24 horas (meia-noite à meia-noite) todos os dias, para um ciclo de negócios completo. Um ciclo de negócios pode durar um mês ou mais, por exemplo, para capturar dados de processamento no final do mês. Se você não tiver nenhum outro monitoramento de desempenho externo ou coleta, você pode executar o pButtons durante o ano todo.

Os seguintes pontos-chave devem ser observados:

- Mude o diretório de log para um local longe dos dados de produção para armazenar arquivos de saída acumulados e evitar problemas de disco cheio!
- Execute um script do sistema operacional ou compacte e archive o arquivo pButtons regularmente. Isso é especialmente importante no Windows, pois os arquivos podem ser grandes.
- Revise os dados regularmente!

No caso de um problema que necessite de análise imediata, os dados do pButtons podem ser visualizados (coletados imediatamente) enquanto as métricas continuam a ser armazenadas para a coleta no final dos dias de execução.

Para obter mais informações sobre pButtons, incluindo visualização, interrupção de uma execução e adição de coleta de dados personalizados, veja o Guia de Monitoramento Caché na documentação mais recente do Caché:

<http://docs.intersystems.com>

Os dados do arquivo HTML pButtons podem ser separados e extraídos (para arquivos CSV, por exemplo) para processamento em gráficos ou outra análise por script ou simplesmente recortar e colar. Veremos exemplos da saída em gráficos posteriormente na próxima postagem.

É claro que, se você tiver problemas urgentes de desempenho, entre em contato com a Central de Suporte (WRC).

Agende a coleta de dados de 24 horas do pButtons

O ^pButtons pode ser iniciado manualmente a partir do prompt do terminal ou agendado. Para agendar uma coleta diária de 24 horas:

1.

Inicie o terminal Caché, mude para o %SYS namespace e execute o pButtons manualmente uma vez para configurar as estruturas de arquivo do pButtons:

```
%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
```

Selecione a opção 6. para teste, amostragem de execução de TESTE de 5 minutos a cada 30 segundos. Observe que sua numeração pode ser diferente, mas o teste deve ser óbvio.

Durante a execução, execute `Collect^pButtons` (conforme mostrado abaixo), você verá informações incluindo o `runid`. Neste caso, "201603031851test".

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

Observou que esta execução de 5 minutos leva 6 minutos e 48 segundos para finalizar? O `pButtons` adiciona um período de carência de 2 minutos a todas as execuções permitindo um tempo para a coleta e ordenação dos logs no formato HTML.

2. **IMPORTANTE!** Altere o diretório de saída do log `pButtons` - o local de saída padrão é a pasta `/mgr`. Por exemplo, no Unix, o caminho para o diretório de log pode ser assim:

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

Certifique-se de que o Cachê tenha permissões de gravação para o diretório e que haja espaço em disco suficiente disponível para acumular os arquivos de saída.

3. Crie um novo perfil de 24 horas com intervalos de 30 segundos executando o seguinte:

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

Verifique se o perfil foi adicionado ao `pButtons`:

```
%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:

Nota: Você pode variar o intervalo de coleta - 30 segundos é bom para monitoramento de rotina. Eu não usaria menos de 5 segundos para uma execução de rotina de 24 horas (... ",5,17280), pois os arquivos de saída podem se tornar muito grandes à medida que `pButtons` coleta dados a cada tique do intervalo. Se você está resolvendo problemas em um determinado momento do dia e deseja dados mais granulares, use um dos perfis padrão ou crie um novo perfil personalizado com um período de tempo mais curto, por exemplo, 1 hora com intervalo de 5 segundos (... ",5,720). Vários `pButtons` podem ser executados ao mesmo tempo, portanto, você pode ter `pButtons` curtos com intervalo de 5 segundos em execução ao mesmo tempo que os `pButtons` de 24 horas.

4. Dica Para sites UNIX, revise o comando `disk`. Os parâmetros padrão usados com o comando `'iostat'` podem não incluir os tempos de resposta do disco. Primeiro exiba quais comandos de disco estão configurados atualmente:

```
%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",">")
```

Para coletar estatísticas de disco, use o comando apropriado para editar a sintaxe de sua instalação do UNIX. Observe o espaço à direita. Aqui estão alguns exemplos:

```
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",">")
```

Você pode criar arquivos HTML pButtons muito grandes, executando os comandos iostat e sar. Para avaliações regulares de desempenho, geralmente uso apenas o iostat. Para configurá-lo há apenas um comando:

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

Mais detalhes sobre como configurar pButtons estão na documentação on-line.

5. Programe o pButtons para iniciar à meia-noite no Portal de Gerenciamento > Operação de Sistema > Gerenciado de Tarefas:

```
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
```

Coletando dados com pButtons

O pButtons disponível em versões mais recentes das Plataformas de Dados InterSystems incluem a coleta automática. Para coletar e ordenar manualmente os dados em um arquivo HTML no %SYS namespace, execute o seguinte comando para gerar quaisquer arquivos de saída HTML pButtons pendentes:

```
do Collect^pButtons
```

O arquivo HTML estará no logdir que você configurou no passo 2 (se você não configurou, vá e [faça agora!](#)). Caso contrário, o local padrão é <Cache install dir/mgr> <Cache install dir/mgr>

Os arquivos são nomeados como <hostnameinstanceNamedatetimeprofileName.html><hostnameinstanceNamedatetimeprofileName.html> Ex. vsan-tc-db1H2015201602180255test.html

Considerações sobre o Monitor de Desempenho do Windows

Se o sistema operacional for Windows, o Monitor de Desempenho do Windows (perfmon) pode ser usado para coletar dados em sincronia com as outras métricas coletadas. Em distribuições Caché mais antigas do pButtons, o perfmon do Windows precisa ser configurado manualmente. Se houver uma demanda nos comentários da postagem, escreverei uma postagem sobre a criação de um template perfmon para definir os contadores de desempenho a serem monitorados e agendar a execução para o mesmo período e intervalo que o pButtons.

Resumo

Esta postagem nos fez começar a coletar alguns dados para analisar. No final da semana, começarei a examinar alguns dados de amostra e o que isso significa. Você pode acompanhar os dados coletados em seus próprios sistemas. Vejo você então.

<http://docs.intersystems.com>

[#Administração do Sistema](#) [#Desempenho](#) [#Soluções de Negócio e Arquiteturas](#) [#Caché](#) [#InterSystems IRIS](#)
[#InterSystems IRIS for Health](#)

URL de
origem: <https://pt.community.intersystems.com/post/plataformas-de-dados-intersystems-e-desempenho-parte-1>