文章
Michael Lei · 五月 24, 2021 阅读大约需 7 分钟

InterSystems 数据平台和性能 – 第 1 篇

您的应用程序已部署，一切运行正常。 很好，击个掌！ 然后电话突然响个不停 – 用户投诉应用程序有时很 “ 慢 ” 。
这是什么意思？ 有时？ 您有哪些工具，查找和解决这个缓慢问题应查看哪些统计数据？
您的系统基础架构是否能承担用户负载的任务？ 在投入生产之前，应该询问哪些基础架构设计问题？
如何自信地为新硬件规划容量，而不会过度规定？ 如何停止电话铃声？ 如何一开始就不让它响？

本系列其他帖子的列表

这将是一次旅程

这是本系列的第一个帖子，本系列将探讨可用于对系统性能进行监视、审查和故障排除的工具和指标，以及影响性能的系统和架构设计注意事项。 在旅程中，我们将朝多个方向前进，了解
Caché、操作系统、硬件、虚拟化以及根据评论中的反馈变为热门的其他方面的性能。

我们将关注反馈循环，其中性能数据提供了一个视角，通过它可查看已部署的应用程序和基础架构的优势和局限性，然后形成反馈以实现更好的设计和容量计划。

不言而喻，您应该不断检查性能指标，但不幸的是，如果客户只是查看数据，他们会多次对长期可见的性能问题感到惊讶。 但问题当然是 - 什么数据？ 我们将从收集一些基本的 Caché
和系统指标来开始这一旅程，这样我们可以对您的系统当前的运行状况有一个感觉。
在以后的帖子中，我们将深入了解主要指标的含义。

有多个可用于系统监视的选项 – 来自 Caché 内部和外部，我们将在本系列中探讨其中的许多选项。

首先，我们将了解我最喜欢的用于连续收集数据的必备工具 ^pButtons，它已经安装在每个 Caché 系统上。

为确保您拥有最新的 pButtons 副本，请查看以下帖子：

https://cn.community.intersystems.com/post/intersystems-数据平台和性能---如何更新-pbuttons

收集系统性能指标 - ^pButtons

Caché 的 pButtons 实用工具可根据其创建的日志文件生成可读的 HTML 性能报告。 pButtons
输出的性能指标可以被轻松提取、绘图和审核。

pButtons html 文件中收集的数据包括：

- Caché 设置：配置、驱动器映射等。
- mgstat：Caché 性能指标 - 大多数的值是每秒平均数。
- Unix：vmstat 和 iostat：操作系统资源和性能指标。
- Windows：性能监视器：Windows 资源和性能指标。
- 其他有用的指标。

pButtons 数据收集对系统性能的影响非常小，这些指标已经由系统收集，pButtons
只是将它们打包以便归档和传输。

要保存基准以进行趋势分析和故障排除，好的做法是在一个完整的业务周期内每天收集 24 小时

pButtons（午夜到午夜）。 一个业务周期可能是一个月或更长时间，例如，从月末处理中捕获数据。
如果没有任何其他外部性能监视或收集，则可以全年运行 pButtons。

应注意以下关键点：

- 将日志目录更改为远离生产数据的位置，以存储累积的输出文件，避免出现磁盘满的问题！
- 定期运行操作系统脚本或以其他方式压缩和存档 pButtons 文件，这在 Windows 上尤其重要，因为文件可能很大。
- 定期审核数据！

如果出现需要立即分析的问题，可以预览（立即收集）pButtons 数据，同时继续存储指标，以便在当天运行结束时进行收集。

有关 pButtons 的更多信息（包括预览、停止运行和添加自定义数据收集），请参见最新的 Caché 文档中的 *Caché 监视指南*：

http://docs.intersystems.com

可以将 pButtons HTML 文件数据分离并提取（例如提取为 CSV 文件），以通过脚本或简单的剪切和粘贴将其处理成图形或进行其他分析。
我们以后将在下一个帖子中看到图表输出的例子。

当然，如果您有紧急的性能问题，请联系 WRC。

计划 24 小时 pButtons 数据收集

^pButtons 可以手动从终端提示符启动，也可以按计划启动。 要计划每天 24 小时收集：

1. 启动 Caché 终端，切换到 %SYS 命名空间，然后手动运行一次 pButtons 以设置 pButtons 文件结构：

   ```
   %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
   ```

选择选项 6 test - 5 minute TEST run sampling every 30 seconds。 请注意，您的编号可能有所不同，但 test 应该是显而易见的。

在运行期间，运行 Collect^pButtons（如下所示），您将看到包括 runid 在内的信息。
在此例中为 “ 20160303_1851_test ”。

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

注意到这个 5 分钟运行还剩 6 分 48 秒吗？ pButtons 为所有运行增加了 2

分钟宽限期，以便有时间将日志收集和整理成 html 格式。

2. 重要！更改 pButtons 日志输出目录 – 默认的输出位置为 <Caché安装路径>/mgr 文件夹。 例如，在 unix 上，日志目录的路径可能如下所示：

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

确保 Caché 有该目录的写权限，并且有足够的磁盘空间用于累积输出文件。

3. 运行以下命令以创建一个新的 24 小时配置文件，间隔为 30 秒：

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

检查该配置文件是否已添加到 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:
```

注意：您可以更改收集间隔 – 30 秒对于例行监视是可以的。 对于例行 24 小时运行，我不会选择低于 5 秒的间隔 (… ” ,5,17280)，因为 pButtons 在每个间隔都会收集数据，而导致输出文件变得非常大。 如果您要对一天中的某个特定时间进行故障排除，并需要更细化的数据，请使用默认的配置文件之一，或创建一个新的自定义配置文件，取较短的时间段，例如 1 小时，间隔为 5 秒 (...",5,720)。 多个 pButtons 可以同时运行，因此可以同时运行一个间隔为 5 秒的短 pButtons 和一个 24 小时 pButtons。

4. 提示 对于 UNIX 站点，可审查磁盘命令。 “ iostat ” 命令使用的默认参数可能未包括磁盘响应时间。 首先显示当前配置了哪些磁盘命令：

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

要收集磁盘统计数据，请使用适合您的 UNIX 安装版的命令来编辑语法。 注意尾部空格。 以下是一些示例：

```
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 ","int
erval"," -c ","count"," > ")
```

您可以同时运行 iostat 和 sar 命令来创建非常大的 pButtons html 文件。 对于定期的性能审查，我通常只使用 iostat。 要仅配置一个命令：

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

有关配置 pButtons 的更多详细信息，请参见在线文档。

5. 在 Management Portal > System Operation > Task Manager 中安排 pButtons 在午夜启动：

```
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
```

收集 pButtons 数据

最新版的 InterSystems 数据平台中随附的 pButtons 包括自动收集功能。 要手动收集数据并整理成一个 html 文件，请在 %SYS 命名空间中运行以下命令以生成未完成的 pButtons html 输出文件：

```
do Collect^pButtons
```

该 html 文件将位于您在步骤 2 设置的日志目录中（如果您未设置，就**现在去设置！**）。 否则默认位置为 <Caché 安装目录/mgr>

文件将命名为 <主机名实例名称日期时间配置文件名.html>，例如 vsan-tc-db1H20152016021800255test.html

Windows 性能监视器注意事项

如果操作系统是 Windows，则可以使用 Windows 性能监视器 (perfmon) 收集数据，与所收集的其他指标同步。 在 pButtons 的旧版 Caché 发行版中，需要手动配置 Windows perfmon。
如果帖子评论有需求，我将写一篇帖子介绍如何创建 perfmon 模板来定义要监视的性能计数器，并安排与 pButtons 相同的运行期间和间隔。

总结

我们从本帖开始收集一些要研究的数据。 在本周晚些时候，我将开始研究一些样本数据及其含义。
您可以使用您在自己的系统上收集的数据。 到时见。

http://docs.intersystems.com

#InterSystems 业务解决方案和架构 #性能 #系统管理 #Caché #InterSystems IRIS #InterSystems IRIS for Health

---

源 URL: https://cn.community.intersystems.com/post/intersystems-%E6%95%B0%E6%8D%AE%E5%B9%B3%E5%8F%B0%E5%92%8C%E6%80%A7%E8%83%BD-%E2%80%93-%E7%AC%AC-1-%E7%AF%87