文章
Michael Lei · 五月 12, 2021 阅读大约需5 分钟

InterSystems 数据平台和性能 - 第 4 篇 - 关 注 内存

本帖将展示为 InterSystems 数据平台上运行的数据库应用调整共享内存需求（包括 global 和例程缓冲区、gmheap 以及 locksize）的方法，以及在配置服务器和虚拟化 Caché 应用程序时应考虑的一些性能提示。
和以往一样，当我谈到 Caché 时，我指的是所有数据平台（Ensemble、HealthShare、iKnow 和 Caché）。

本系列其他帖子的列表

当我最初开始使用 Caché 时，大多数客户的操作系统是 32 位的，Caché 应用程序的内存有限且昂贵。
通常部署的英特尔服务器只有几个核心，唯一的扩展方式是选择大型服务器，或者使用 ECP 横向扩展。
现在，即使是基本的生产级服务器也具有多个处理器、几十个核心，并且最小内存为 128 或 256 GB，可能达到 TB。对于大多数数据库安装，ECP 已被遗忘，我们现在可以在单台服务器上大幅提高应用事务处理速率。

Caché 的一个关键特性是我们在共享内存（通常称为数据库缓存或 global 缓冲区）中使用数据的方式。
简单地说，如果适当调整 global 缓冲区的大小并为其分配“更多”内存，通常会提高系统性能 - 访问内存中的数据要比访问磁盘上的数据快得多。当年，当 32 位系统占主导地位时，对问题 *我应该为 global 缓冲区分配多少内存？* 的回答很简单 - *尽可能多！*
反正也没有很多内存可用，所以人们努力计算操作系统需求、操作系统和 Caché 进程的数量和大小以及每个进程使用的实际内存，以找到剩余的内存来分配尽可能大的 global 缓冲区。

趋势已经转变

如果您在当代服务器上运行应用程序，可以为 Caché 实例分配大量内存，并且由于内存现在既“便宜”又充足，因此通常放任其使用。
然而，趋势再次转变，除了最大型的系统，现在部署的几乎所有系统都是虚拟化的。
所以，虽然在裸机时可以为“怪兽”虚拟机分配大量内存，但重点仍然回到适当调整系统规模上。
为了最大程度地利用服务器整合，需进行容量计划以充分利用可用主机内存。

你在使用内存？

通常，Caché 数据库服务器上有四个主要的内存消耗者：

- 操作系统，包括文件系统缓存。
- 其他非 Caché 应用程序（如果已安装）。
- Caché 进程。
- Caché 共享内存（包括 global 和例程缓冲区以及 GMHEAP）。

从高级别上，所需物理内存量只是将列表上每一项的需求都加起来。上面的所有项目都使用实际内存，但它们也可以使用虚拟内存，容量计划的一个关键部分是将系统规模调整为有足够的物理内存，使分页不会发生或尽量少发生，或者至少尽量减少或消除必须从磁盘恢复内存的硬页面故障。

在本帖中，我将重点介绍 Caché 共享内存的大小调整以及优化内存性能一些通用规则。
操作系统和内核的要求因操作系统而异，但大多数情况下为几个 GB。
文件系统缓存视情况而定，将是列表上其他项目获得内存分配之后的其余任何可用内存。

Caché 主要是进程 - 如果在应用程序运行时查看操作系统统计信息，则会看到缓存进程（例如 cache 或

cache.exe）。因此，观测应用程序内存要求的一个简单方法是查看操作系统指标。例如，通过 Linux 上的 vmstat 或 ps 或者 Windows 进程资源管理器，以及使用中的实际内存总量，推断出增长需求和峰值需求。
请注意，某些指标会报告包含共享内存的虚拟内存，因此收集实际内存需求时务必小心。

调整 global 缓冲区大小 - 一种简单的方法

对于高事务数据库，容量计划的目标之一是将 global 缓冲区的大小调整为使内存中容纳尽可能多的应用数据库工作集。这将最大程度地减少读取 IOPS，并且通常会使应用程序的性能更好。我们还需取得平衡，使其他内存用户（如操作系统和 Caché 进程）不会被移出分页，并且有足够的内存用于文件系统缓存。

我在 本系列的第 2 部分 中展示了一个示例，来说明如果对磁盘过度读取会发生什么情况。在该示例中，高读取数是由不良的报告或查询引起的，但是如果 global 缓冲区太小而迫使应用程序不断从磁盘读取数据块，则会产生同样的效果。
另外，值得注意的是，存储的格局一直在变化 - 随着固态硬盘技术的进步，存储速度越来越快，但靠近运行中的进程的内存数据仍然是最好的。

当然，每个应用程序都是不同的，所以必须指出 **“您的情况可能有所不同”**，但有一些通用规则可以帮助您开始为应用程序的共享内存制定容量计划。
之后，您可以根据您的具体要求进行调整。

从哪里开始？

很遗憾，并没有万能答案。但是，正如我在之前的帖子谈到的，一个好的做法是确定系统 CPU 容量规模，使得在达到所需峰值事务处理速率的情况下，峰值处理期间的 CPU 使用率为大约 80%。为短期增长或意外的活动激增留出 20% 余量。

例如，当我调整 TrakCare 系统的规模时，我通过基准测试和查看客户站点指标来了解已知事务处理速率的 CPU 要求，并且可以针对基于英特尔处理器的服务器使用普遍的经验法则：

经验法则：每个 CPU 核心对应 *n* GB 大小的物理内存。

- 对于 TrakCare 数据库服务器，*n* 为 8 GB。对于小型 Web 服务器则为 4 GB。

经验法则：为 Caché global 缓冲区分配 *n*% 的内存。

- 对于中小型 TrakCare 系统，*n*% 为 60%，其余 40% 的内存留给操作系统、文件系统缓存和 Caché 进程。
  如果您需许多文件系统缓存或有许多进程，可以更改该值，例如改为 50%。
  或者，在大型系统上使用超大内存配置时，将其调整为更高的百分比。
- 此经验法则假定服务器上只有一个 Caché 实例。

所以如果应用程序需10 个 CPU 核心，则虚拟机将有 80 GB 内存，其中 48 GB 为 global 缓冲区，32 GB 用于其他所有用途。

内存大小调整规则适用于物理或虚拟化系统，因此对于 TrakCare 虚拟机，1 vCPU : 8 GB 内存的比例同样适用。

调整 global 缓冲区

要查看规模调整的效果如何，需观察几个项目。您可以使用操作系统工具观察 Caché 之外的可用内存。根据最佳计算结果进行设置，然后观察一段时间内的内存使用情况，如果一直有可用内存，则可以重新配置系统以增加 global 缓冲区或适当调整虚拟机规模。

global 缓冲区大小调整适当的另一个关键指标是读取 IOPS 尽可能低 - 这意味着 Caché 缓存效率会很高。
您可以使用 mgstat 观察不同的 global 缓冲区大小对 PhyRds 和 RdRatio 的影响，本系列的第 2 部分 提供了查看这些指标的示例。除非您的整个数据库都在内存中，否则总会有一些对磁盘的读取，目的只是保持尽可能

低的读取数。

请记住您的硬件食物群并做好平衡 - 为 global 缓冲区分配更多内存会降低读取 IOPS，但可能会提高 CPU 使用率，因为您的系统现在可以在更短时间内完成更多工作。不过降低 IOPS 终归是好事，您的用户会因为更快响应时间而更满意。

有关如何将您的要求应用于 **物理内存** ，请参见下面的部分。

对于虚拟服务器计划， *不要* 超额预定生产虚拟机内存， *尤其是 Caché 共享内存* ，更多信息也请参见文。

您的应用程序甜蜜点是每 CPU 核心 8GB 物理内存吗？我说不准，但要提是否有相似的方法适合您的应用程序。
不管每核心 4GB 还是 10GB，如果您找到了另一种调整 global 缓冲区大小的方法，请在下面留下评论。

监视 Global 缓冲区使用情况

Caché 实用工具 ^GLOBUFF 可显示 global 缓冲区在任意时间点的使用情况统计。例如，按百分比显示前 25 名：

```
do display^GLOBUFF(25)
```

例如，输出可能如下所示：

```
Total buffers: 2560000    Buffers in use: 2559981  PPG buffers: 1121 (0.044%)

Item  Global                                Database          Percentage (Count)
1     MyGlobal                              BUILD-MYDB1        29.283 (749651)
2     MyGlobal2                             BUILD-MYDB2        23.925 (612478)
3     CacheTemp.xxData                      CACHETEMP          19.974 (511335)
4     RTx                                   BUILD-MYDB2        10.364 (265309)
5     TMP.CachedObjectD                     CACHETEMP          2.268 (58073)
6     TMP                                   CACHETEMP          2.152 (55102)
7     RFRED                                 BUILD-RB           2.087 (53428)
8     PANOTFRED                             BUILD-MYDB2        1.993 (51024)
9     PAPi                                  BUILD-MYDB2        1.770 (45310)
10    HIT                                   BUILD-MYDB2        1.396 (35727)
11    AHOMER                                BUILD-MYDB1        1.287 (32946)
12    IN                                    BUILD-DATA         0.803 (20550)
13    HIS                                   BUILD-DATA         0.732 (18729)
14    FIRST                                 BUILD-MYDB1        0.561 (14362)
15    GAMEi                                 BUILD-DATA         0.264 (6748)
16    OF                                    BUILD-DATA         0.161 (4111)
17    HISLast                               BUILD-FROGS        0.102 (2616)
18    %Season                               CACHE              0.101 (2588)
19    WooHoo                                BUILD-DATA         0.101 (2573)
20    BLAHi                                 BUILD-GECKOS       0.091 (2329)
21    CTPCP                                 BUILD-DATA         0.059 (1505)
22    BLAHi                                 BUILD-DATA         0.049 (1259)
23    Unknown                               CACHETEMP          0.048 (1222)
24    COD                                   BUILD-DATA         0.047 (1192)
25    TMP.CachedObjectI                     CACHETEMP          0.032 (808)
```

这在几个方面很有用，例如，查看哪些工作集保留在内存中。
如果你觉得此实用工具很有用，请在下面发表评论，让其他社区用户了解它为何对您有帮助。

调整例程缓冲区大小

应用程序运行的例程（包括编译的类）存储在例程缓冲区中。
调整例程缓冲区共享内存大小的目标是让所有例程代码都在例程缓冲区中加载并驻留。与 global 缓冲区一样，从磁盘读取例程的成本很高且效率低下。例程缓冲区的最大大小为 1023 MB。
一般来说，您都获得的例程缓冲区比需要的，因为将例程缓存总是可以大幅提高性能。

例程缓冲区由不同大小的缓冲区组成。默认情况下，Caché 确定每种大小的缓冲区数量，在安装时，2016.1 的默认缓冲区为 4、16 和 64 KB。可以更改不同大小的内存分配，但是要开始容量计划，建议保留 Caché 默认值，除非您有特殊原因需要进行更改。有关更多信息，请参见 Caché 文档 中的《Caché 参数文件参考》的“config”附录中的 routines，以及《Caché 系统管理指南》的“配置 Caché”一章中的“内存和启动设置”。

应用程序运行时，例程从磁盘加载，并存储在可容纳该例程的最小缓冲区中。例如，如果某个例程为 3 KB，理想情况下将存储在一个 4 KB 缓冲区中，如果没有可用的 4 KB 缓冲区，则使用更大的缓冲区。大于 32 KB 的例程将根据需要使用多达 64 KB 的例程缓冲区。

检查常规缓冲区的使用

mgstat 指标 RouLas

了解例程缓冲区是否足够大的一个方法是 mgstat 指标 RouLas（例程加载次数和保存次数）。一个 RouLas 是指从磁盘获取或保存磁盘一次。
例程的加载/保存次数可能表示存在性能问题，在这种情况下，可以增加例程缓冲区数量来提高性能。

cstat

如果您已将例程缓冲区增加到最大值 1023 MB，但仍然发现 RouLas 很高，那么可以使用 cstat 命令进行更详细的检查，以了解缓冲区中有哪些例程以及它们使用了多少内存。

```
ccontrol stat cache -R1
```

这将生成个例程指标列表，其中包括例程缓冲区列表和缓存中的所有例程。例如，默认安装 Caché 后，该列表的一部分如下所示：

```
Number of rtn buf: 4 KB-> 9600, 16 KB-> 7200, 64 KB-> 2400,
gmaxrouvec (cache rtns/proc): 4 KB-> 276, 16 KB-> 276, 64 KB-> 276,
gmaxinitalrouvec: 4 KB-> 276, 16 KB-> 276, 64 KB-> 276,

Dumping Routine Buffer Pool Currently Inuse
 hash   buf  size sys sfn inuse old type   rcrc     rtime   rver rctentry rouname
   22: 8937  4096   0   1     1   0  D  6adcb49e  56e34d34    53 dcc5d477  %CSP.UI.Po
rtal.ECP.0
   36: 9374  4096   0   1     1   0  M  5c384cae  56e34d88    13 908224b5  %SYSTEM.Wo
rkMgr.1
   37: 9375  4096   0   1     1   0  D  a4d44485  56e34d88    22 91404e82  %SYSTEM.Wo
rkMgr.0
   44: 9455  4096   0   0     1   0  D  9976745d  56e34ca0    57 9699a880  SYS.Monito
r.Health.x
 2691:16802 16384   0   0     7   0  P  da8d596f  56e34c80    27 383da785  START
   etc
   etc
```

上面第 2 行中的“rtns/proc”表示，默认情况下，每种大小的缓冲区可以缓存 276 个例程。

使用此信息来调整例程缓冲区大小的另一种方法是，运行应用程序并使用 cstat -R1 列出正在运行的例程。
然后可以计算使用中的例程大小，例如将此列表放在 Excel 中，按大小排序，并精确地查看哪些例程正在使用中。
如果每种大小的缓冲区没有被全部使用，则说明有足够的例程缓冲区，或者如果每种大小的缓冲区都已全部使用，则
需增加例程缓冲区，或者可以更直接地配置每个存储桶的大小。

锁表大小

locksiz 配置参数是为管理并发控制锁而分配的内存大小（以字节为单位），该锁用于防止不同的进程同时更改一个
特定的数据元素。在内部，内存中的锁表包含当前锁，以及持有这些锁的进程的信息。

由于用于分配锁的内存取自 GMHEAP，所以用于锁的内存不能超过 GMHEAP 中的内存。如果增加 locksiz
的大小，请按照下面 GMHEAP 部分中的公式增加 GMHEAP 的大小以进行匹配。
有关应用程序使用锁表的信息可以使用系统管理门户 (SMP) 进行监视，或者更直接地使用 API 来监视：

```
set x=##class(SYS.Lock).GetLockSpaceInfo()?
```

该 API 返回三个值："可获空间、可用空间、已用空间"。
检查可用空间和已用空间可粗略计算合适的值（某些锁空间为锁结构预留）。更多信息，请参见 Caché 文档 。

注：如果编辑 locksiz 设置，更改会立即生效。

GMHEAP

GMHEAP（通用内存堆）配置参数定义为：Caché 的通用内存堆的大小（以 KB 为单位）。锁表、NLS 表和 PID
表也是根据它来进行分配的。

注：更改 GMHEAP 需重启 Caché。

为了帮助您确定应用规模，可以使用 API 来检查 GMHEAP 的使用信息：

```
%SYSTEM.Config.SharedMemoryHeap
```

该 API 还提供获取可用的通用内存堆的功能，并推荐了用于配置的 GMHEAP 参数。例如，DisplayUsage
方法显示每个系统组件使用的所有内存以及可用堆内存的数量。更多信息，请参见 Caché 文档 。

```
write $system.Config.SharedMemoryHeap.DisplayUsage()
```

要了解任何时间点的 GMHEAP 使用情况和建议，可以使用 RecommendedSize 方法。
但是，您将需要运行该方法才能为系统建立基线和建议。

```
write $system.Config.SharedMemoryHeap.RecommendedSize()
```

经验法则：再次说明，您的应用情况会有所不同，但仍可以使用以下选项之一来开始调整大小：

```
??? 128MB???64 MB * ???????2 ? locksiz??????????
```

记住，必须调整 GMHEAP 的大小以包括锁表。

大页内存

Mark Bolinsky 写了一篇非常好的帖子，解释了为什么 在 Linux 中开启大页内存可大幅提高性能。

危险！Windows 大页内存和共享内存

Caché 在所有平台上的所有版本都使用共享内存，它可以大幅提高性能，包括在 Windows 中也总是使用，但您需注意一些特定于 Windows 的问题。

Caché 在启动时会分配一大块共享内存，用于数据库缓存（global 缓冲区）、例程缓存（例程缓冲区）、共享内存堆、日志缓冲区和其他控制结构。在 Caché 启动时，可以使用小页或大页来分配共享内存。在 Windows 2008 R2 及更高版本中，Caché 默认使用大页，但如果系统已长时间运行，由于内存已碎片化，可能无法在 Caché 启动时分配连续内存，Caché 可能会改用小页。

意外地以小页启动 Caché 可导致 Caché 启动时的共享内存小于配置中定义的值，或者 **Caché 可能需很长时间才能启动或启动失败**。
我在具有故障转移集群的站点上遇到过这种情况，这些站点中的备份服务器已经很长时间没有用作数据库服务器了。

提示：一种缓解策略是定期重启离线的 Windows 集群服务器。另一个策略是使用 Linux。

物理内存

物理内存取决于处理器的最佳配置。不良的内存配置可能会严重影响性能。

英特尔内存配置最佳实践

以下信息只适用于 **英特尔**处理器。请与供应商确认哪些规则适用于其他处理器。

决定 DIMM 最优性能因素包括：

- DIMM 类型
- DIMM rank
- 时钟速度
- 相对于处理器的位置（最近/最远）
- 内存通道数
- 所需冗余功能。

例如，在 Nehalem 和 Westmere 服务器（至强 5500 和 5600）上，每个处理器有三个内存通道，因此应以三个为一组的方式安装内存。对于当前处理器（例如 E5-2600），每个处理器有四个内存通道，因此应以四个为一组的方式安装内存。

当存在不平衡的内存配置时（没有以三个/四个为一组的方式安装内存，或者内存 DIMM 的大小不同），不平衡的内存可能导致 23% 的内存性能损失。

请记住，Caché 的特点之一是在内存中进行数据处理，因此获得最佳的内存性能很重要。
另外值得 注 意的是，为了获得最大带宽，应该为服务器配置最快的内存速度。对于至强处理器，只有每通道分达 2 条 DIMM 时才支持最大内存性能。因此对于常见的具有 2 个 CPU 的服务器来说，最大内存配置取决于包括 CPU 频率和 DIMM 大小（8GB、16GB 等）在内的因素。

经验法则：

- 使用平衡的平台配置：为每个通道和每个插槽填充相同数量的 DIMM
- 在整个平台中使用相同的 DIMM 类型：相同的大小、速度和 rank 数。
- 对于物理服务器，根据这些英特尔处理器最佳实践，将主机服务器的物理内存总数四舍五入到自然断点 - 64GB、128GB 等。

VMware 虚拟化注意事项

我会在将来另写一篇帖子来继续介绍虚拟化 Caché 的更多规则。不过，内存分配应考虑以下关键规则：

规则：在生产系统上设置 VMware 内存预留。

如上文所述，Caché 在启动时会分配一大块共享内存，用于 global
和例程缓冲区、GMHEAP、日志缓冲区和其他控制结构。

您想要避免任何共享内存交换，所以将 *生产数据库虚拟机* 内存预留设置为至少是 Caché 共享内存的大小加上 Caché 进程和操作系统及内核服务的内存。如果有疑问，请预留全部生产数据库虚拟机内存。

一般来说，如果在相同系统上混合使用生产服务器和非生产服务器，则不要在非生产系统上设置内存预留。
让非生产服务器争夺剩下的内存 ;)。VMware 通常将具有 8 个以上 CPU 的虚拟机称为“怪兽虚拟机”。高事务 Caché 数据库服务器通常是怪兽虚拟机。在怪兽虚拟机上设置内存预留还有其他考虑因素，例如，如果要迁移某个怪物虚拟机进行维护，或者由于高可用触发重启，那么目标主机服务器必须有足够的可用内存。
我会在将来的帖子中讨论这方面的计划策略，以及其他内存考虑因素，如计划充分利用 NUMA。

总结

这是对内存进行容量计划（一个混乱的领域）的开始，当然不像调整 CPU 规模那样明确。
如果您有任何疑问或意见，请留下评论。

发布本帖后，我将去参加 Global Summit 2016。
如果您今年也参加，欢迎观看我的两场关于性能主题的演讲，我也很乐意在开发者区域与您当面交流。

#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-%E7%AC%AC-4-%E7%AF%87-%E5%85%B3%E6%B3%A8%E5%86%85%E5%AD%98