文章
Michael Lei · 七月 6, 2021 阅读大约需 16 分钟

精华文章--虚拟化大型数据库 - VMware CPU 容量规划

供应商或内部团队要求说明如何为 VMware vSphere 上运行的*大型生产数据库*进行 CPU 容量规划。

总的来说，在调整大型生产数据库的 CPU 规模时，有几个简单的最佳做法可以遵循：

- 为每个物理 CPU 核心规划一个 vCPU。
- 考虑 NUMA 并按理想情况调整虚拟机规模，以使 CPU 和内存对于 NUMA 节点是本地的。
- 合理调整虚拟机规模。 仅在需要时才添加 vCPU。

通常，这会引出几个常见问题：

- 由于使用超线程技术，VMware 创建的虚拟机的 CPU 数量可以是物理 CPU 数量的两倍。 那不就是双倍容量吗？ 创建的虚拟机不应该有尽可能多的 CPU 吗？
- 什么是 NUMA 节点？ 我应该在意 NUMA 吗？
- 虚拟机应该合理调整规模，但我如何知道什么时候合理？

我以下面的示例回答这些问题。 但也要记住，最佳做法并不是一成不变的。 有时需要做出妥协。 例如，大型生产数据库虚拟机很可能不适合 NUMA 节点，但我们会看到，其实是没问题的。 最佳做法是指必须针对应用程序和环境进行评估和验证的准则。

虽然本文中的示例是在 InterSystems 数据平台上运行的数据库，但概念和规则通常适用于任何大型（怪兽）虚拟机的容量和性能规划。

有关虚拟化最佳做法以及有关性能和容量规划的更多帖子，请参见 InterSystems 数据平台和性能系列的其他帖子列表。

怪兽虚拟机

本帖主要是关于部署*怪兽虚拟机*，有时也称为 *Wide 虚拟机*。 高事务数据库的 CPU 资源要求意味着它们通常部署在怪兽虚拟机上。

> 怪兽虚拟机是指虚拟 CPU 或内存多于物理 NUMA 节点的虚拟机。

CPU 架构和 NUMA

当前的英特尔处理器架构采用非统一内存架构 (NUMA)。 例如，本帖中用来运行测试的服务器有：

- 两个 CPU 插槽，每个插槽一个 12 核处理器（英特尔 E5-2680 v3）。
- 256 GB 内存（16 条 16GB RDIMM）

每个 12 核处理器都有自己的本地内存（128GB RDIMM 及本地高速缓存），还可以访问同一主机中其他处理器上的内存。 每个由 CPU、CPU 高速缓存和 128GB RDIMM 内存组成的 12 核套装都是一个 NUMA 节点。 为了访问其他处理器上的内存，NUMA 节点通过快速互连来连接。

处理器上运行的进程访问本地 RDIMM 和缓存内存的延迟比跨互连访问其他处理器上的远程内存的延迟要低。
跨互连访问会增加延迟，因此性能不一致。 同样的设计也适用于具有两个以上插槽的服务器。
一台四插槽英特尔服务器有四个 NUMA 节点。

ESXi 了解物理 NUMA，ESXi CPU 调度器设计为优化 NUMA 系统的性能。 ESXi 使性能最大化的方法之一是在物理 NUMA 节点上创建数据本地性。 在我们的示例中，如果虚拟机有 12 个 vCPU，并且内存不到 128GB，ESXi 将分配该虚拟机在一个物理 NUMA 节点上运行。 这就形成了规则：

> 如果可能，将虚拟机规模调整为使 CPU 和内存对于 NUMA 节点是本地的。

如果需要比 NUMA 节点规模大的怪兽虚拟机也没有问题，ESXi 可以很好地自动计算和管理要求。 例如，ESXi 将创建能够智能调度到物理 NUMA 节点上的虚拟 NUMA 节点 (vNUMA)，以获得最佳性能。 vNUMA 结构对操作系统公开。 例如，如果您有一台具有两个 12 核处理器的主机服务器和一个具有 16 个 vCPU 的虚拟机，ESXi 可能会使用每个处理器上的 8 个物理核心来调度虚拟机 vCPU，操作系统（Linux 或 Windows）将看到两个 NUMA 节点。

同样重要的是，应合理调整虚拟机的规模，并且分配的资源不要超过所需的资源，否则会导致资源浪费和性能损失。 除了有助于调整 NUMA 的规模，具有高（但安全的）CPU 利用率的 12 vCPU 虚拟机比具有中低 CPU 利用率的 24 vCPU 虚拟机更高效、性能更好，特别是该主机上还有其他虚拟机需要调度并且争用资源时。
这也再次强化了该规则：

> 合理调整虚拟机规模。

注意：英特尔和 AMD 的 NUMA 实现有区别。 AMD 每个处理器有多个 NUMA 节点。
我已经有一段时间没有在客户服务器中看到 AMD 处理器了，但是如果你有这些处理器，请检查 NUMA 布局，作为规划的一部分。

Wide 虚拟机和授权

为实现最佳 NUMA 调度，请配置 Wide 虚拟机； 2017 年 6 月更正：~~按每个插槽 1 个 vCPU 配置虚拟机。 例如，默认情况下，一个具有 24 个 vCPU 的虚拟机应配置为 24 个 CPU 插槽，每个插槽一个核心。~~

> 遵守 VMware 最佳做法规则。

请参见 [VMware 博客上的这篇文章以查看示例。]

该 VMware 博客文章进行了详细介绍，但是作者 Mark Achtemichuk 建议遵循以下经验法则：

- 虽然有许多高级 vNUMA 设置，但只有极少数情况下需要更改其默认值。
- 总是将虚拟机 vCPU 数配置为反映每插槽核心数，直到超过单个物理 NUMA 节点的物理核心数。
- 当需要配置的 vCPU 数量超过 NUMA 节点中的物理核心数量时，将 vCPU 均匀分配到最少数量的 NUMA 节点上。
- 当虚拟机规模超过物理 NUMA 节点时，不要分配奇数数量的 vCPU。
- 不要启用 vCPU 热添加，除非您不介意禁用 vNUMA。
- 不要创建规模大于主机物理核心总数的虚拟机。

Caché 授权以核心数为准，因此这不是问题，但是对于除 Caché 以外的软件或数据库，指定虚拟机有 24 个插槽可能会对软件授权产生影响，因此必须与供应商核实。

超线程和 CPU 调度器

超线程 (HT) 经常在讨论中出现，我听过 “ 超线程使 CPU 核心数量翻倍 ”。
这在物理层面上显然是不可能的，物理核心有多少就是多少。 超线程应该被启用，并会提高系统性能。
预计应用程序性能可能会提高 20% 或更多，但实际数字取决于应用程序和工作负载。 但肯定不会翻倍。

正如我在 VMware 最佳实践中所述，*调整大型生产数据库虚拟机规模*的一个很好的起点是假定 vCPU
拥有服务器上完整的物理核心专用资源 — 在进行容量规划时基本忽略超线程。 例如：

> 对于一台 24 核主机服务器，可规划总共多达 24 个 vCPU 的生产数据库虚拟机，且可能还有余量。

在您花时间监测应用程序、操作系统和 VMware
在峰值处理期间的性能后，您可以决定是否进行更高度的虚拟机整合。 在最佳做法帖子中，我将规则表述为：

> 一个物理 CPU（包括超线程）= 一个 vCPU（包括超线程）。

为什么超线程不会使 CPU 翻倍

英特尔至强处理器上的超线程是在一个物理核心上创建两个*逻辑* CPU 的方法。
操作系统可以有效地针对两个逻辑处理器进行调度 — 如果一个逻辑处理器上的进程或线程正在等待，例如等待
IO，则物理 CPU 资源可以被另一个逻辑处理器使用。
在任何时间点都只能有一个逻辑处理器运行，因此虽然物理核心得到了更有效的利用，但*性能并没有翻倍*。

在主机 BIOS 中启用超线程后，当创建虚拟机时，可以为每个超线程逻辑处理器配置一个 vCPU。
例如，在一台启用了超线程的物理 24 核服务器上，可以创建具有多达 48 个 vCPU 的虚拟机。 ESXi CPU
调度器将通过首先在独立的物理核心上运行虚拟机进程来优化处理（同时仍然考虑 NUMA）。
在以后的帖子中，我将探讨在怪兽数据库虚拟机上分配比物理核心数更多的 vCPU 是否有助于扩展。

协同停止和 CPU 调度

在监测主机和应用程序性能后，您可以决定是否让主机 CPU 资源过载。
这是否是一个好主意在很大程度上取决于应用程序和工作负载。
了解调度器和要监测的关键指标有助于确保没有使主机资源过载。

我有时听说，要让虚拟机正常运行，空闲逻辑 CPU 的数量必须与虚拟机中的 vCPU 数量相同。 例如，一个 12
vCPU 虚拟机必须 “ 等待 ” 12 个逻辑 CPU “ 可用 ”，才能继续执行。 不过应该注意，ESXi 在版本 3 之后就不是这样了。
ESXi 对 CPU 使用宽松的协同调度，以提高应用程序性能。

由于多个协作线程或进程经常相互同步，不一起调度它们可能会增加操作的延迟。
例如，在自旋循环中，一个线程等待被另一个线程调度。 为了获得最佳性能，ESXi 尝试将尽可能多的同级 vCPU
一起调度。 但是，当有多个虚拟机在整合环境中争用 CPU 资源时，CPU 调度器可以灵活地调度 vCPU。 如果一些
vCPU 的进展比同级 vCPU 领先太多（这个时间差称为偏移），领先的 vCPU 将决定是否停止自身（协同停止）。
请注意，协同停止（或协同启动）的是 vCPU，不是整个虚拟机。
这种机制即使在资源有些过载的情况下也非常有效，但正如您所预期，CPU 资源过载太多将不可避免地影响性能。
我在后面的示例 2 中展示了一个过载和协同停止的例子。

记住，这不是虚拟机之间全力争夺 CPU 资源的竞赛；ESXi CPU 调度器的工作是确保 CPU
共享、保留和限制等策略被遵守，同时最大限度地提高 CPU
利用率，并确保公平性、吞吐量、响应速度和可伸缩性。
关于使用保留和共享来确定生产工作负载优先级的讨论不在本帖范围之内，而且取决于应用程序和工作负载组合。
如果我以后发现任何特定于 Caché 的建议，我可能会重新讨论这个话题。 有许多因素会影响到 CPU
调度器，本节只是简单提一下。 要深入了解，请参见帖子末尾的参考资料中的 VMware 白皮书及其他链接。

示例

为了说明不同的 vCPU 配置，我使用一个基于浏览器的高事务速率医院信息系统应用程序运行了一系列基准测试。
与 VMware 开发的 DVD 商店数据库基准测试的概念类似。

基准测试的脚本是根据现场医院实施的观测值和指标创建的，包括高使用率的工作流程、事务和使用最多系统资源的组件。 其他主机上的驱动虚拟机以设置的工作流程事务速率执行具有随机输入数据的脚本，来模拟 Web
会话（用户）。 1 倍速率的基准为基线。 速率可以按比例递增和递减。

除了数据库和操作系统指标外，一个很好的用来衡量基准数据库虚拟机性能的指标是在服务器上测量的组件（也可以是事务）响应时间。 一个组件示例是一部分最终用户屏幕。
组件响应时间增加意味着用户将开始看到应用程序响应时间变差。
性能良好的数据库系统必须为最终用户提供*一致的*高性能。
在下面的图表中，我针对一致的测试性能进行测量，并通过对 10
个最慢的高使用率组件的响应时间取平均值来表示最终用户体验。
预计平均组件响应时间为亚秒级，用户屏幕可能由一个组件组成，或者复杂的屏幕可能有多个组件。

> 请记住，您始终针对峰值工作负载进行规模调整，并且为意外的活动峰值留出缓冲区。 我通常以平均 80% 的峰值 CPU 利用率为目标。

基准测试硬件和软件的完整列表在帖子末尾。

示例 1. 合理调整规模 - 每个主机一个怪兽虚拟机

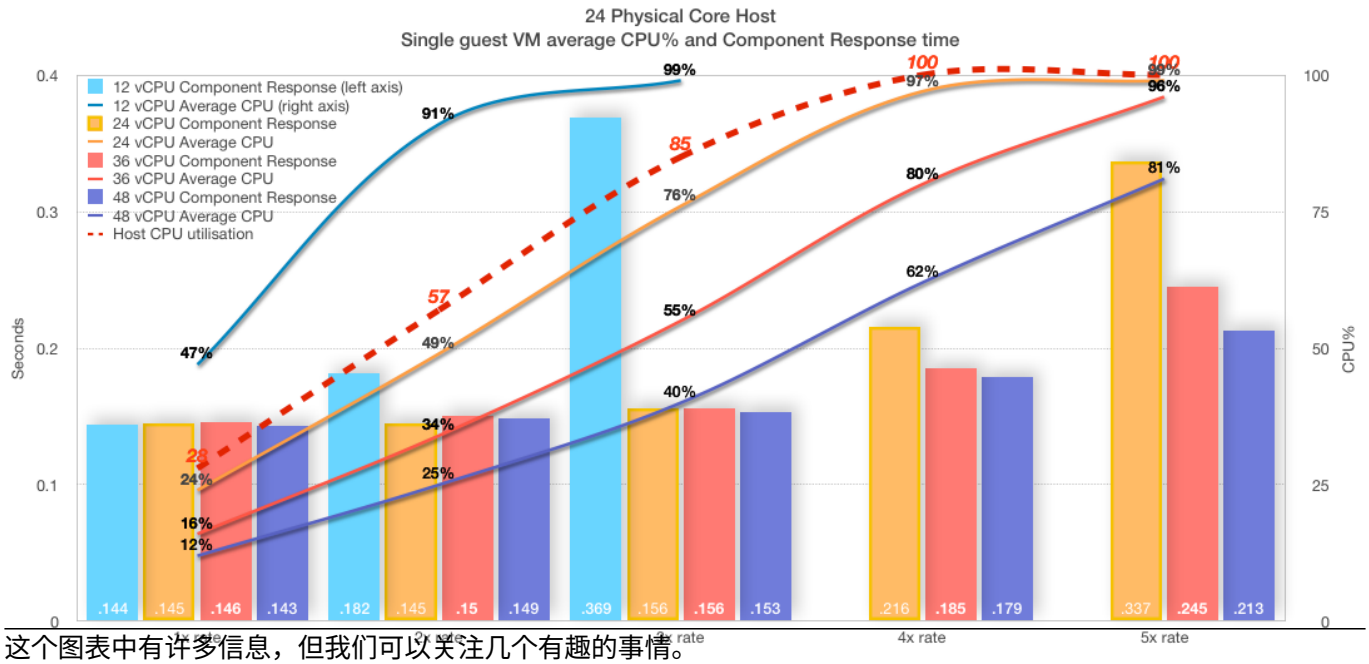
可以创建一个可以使用主机服务器所有物理核心的数据库虚拟机，例如 24 物理核心主机上的 24 vCPU 虚拟机。
数据库虚拟机不会在 Caché 数据库镜像中 “ 裸机 ” 运行服务器以实现
HA，也不会引入操作系统故障转移集群的复杂性，而是包含在 vSphere 集群中实现管理和 HA，例如 DRS 和 VMware HA。

我见过有客户遵循老派的思维，根据五年硬件寿命结束时的预期容量来确定主数据库虚拟机的规模，但从上文可知，最好合理调整规模；如果虚拟机没有过度调整，性能和整合度会更好，并且管理 HA 将更容易；如果需要维护或主机出现故障，并且数据库怪兽虚拟机必须迁移或在其他主机上重启，想想俄罗斯方块的玩法就知道了。
如果预计事务速率显著增加，可以在计划维护期间提前增加 vCPU。

> 注意，“ 热添加 ” CPU 选项会禁用 vNUMA，因此不要将其用于怪兽虚拟机。

考虑下图显示的在 24 核主机上进行的一系列测试。 对于这个 24 核系统，3 倍事务速率是甜蜜点和容量规划目标。

- 主机上运行一个虚拟机。
- 使用了四种虚拟机规模来展示 12、24、36 和 48 vCPU 的性能。
- 尽可能对每种虚拟机规模都运行一系列事务速率（1 倍、2倍、3 倍、4 倍、5 倍）。
- 性能/用户体验以组件响应时间（条形图）的形式显示。
- 客户机虚拟机的 CPU 利用率百分比为平均值（线条）。
- 所有虚拟机规模中，主机 CPU 利用率都在 4 倍速率时达到 100%（红色虚线）。



这个图表中有许多信息，但我们可以关注几个有趣的事情。

- 24 vCPU 虚拟机（橙色）平稳地增加到目标 3 倍事务速率。 在 3 倍速率时，客户机内虚拟机的平均 CPU 利用率为 76%（峰值为 91% 左右）。 主机 CPU 利用率并不比客户机虚拟机高多少。 在 3 倍速率之前，组件响应时间非常稳定，因此用户很满意。 就我们的目标事务速率而言 — *这个虚拟机已合理调整规模*。

关于合理规模调整先说这么多，那么增加 vCPU 也就是使用超线程又会如何。 性能和可伸缩性有可能翻倍吗？
简短回答是*不可能！*

在这种情况下，可以通过查看 4 倍以上速率的组件响应时间来了解答案。 虽然在分配了更多逻辑核心 (vCPU) 后性能“ 更好 ”，但仍然不平稳，不像 3 倍速率之前那样一致。 4 倍速率时，用户将报告响应时间变慢，无论分配多少个 vCPU。 请记住，在 4 倍速率时，*主机*曲线已经持平于 100% CPU 利用率，如 vSphere 所报告。 在 vCPU 数量较多的情况下，即使客户机内 CPU 指标 (vmstat) 报告低于 100% 利用率，对于物理资源来说情况也并非如此。
请记住，客户机操作系统不知道它是虚拟化的，它只是报告它所看到的资源。
另外，客户机操作系统也看不到超线程，所有 vCPU 都表现为物理核心。

关键是，数据库进程（在 3 倍事务速率时有 200 多个 Caché 进程）非常繁忙，并且非常高效地使用处理器，逻辑处理器没有很多空闲资源来调度更多工作，或将更多虚拟机整合到该主机。 例如，很大一部分 Caché 处理是在内存中进行的，因此没有很多 IO 等待。 所以，虽然可以分配比物理核心更多的 vCPU，但由于主机已经被 100% 利用，并不会获益许多。

Caché 非常擅长处理高工作负载。 即使主机和虚拟机的 CPU 利用率达到 100%，应用程序仍在运行，并且事务速率仍在提高 — 扩展不是线性的，如我们所见，响应时间越来越长，用户体验将受到影响 — 但应用程序不会“ 一落千丈 ”，尽管情况不是很好，但用户仍可以工作。
如果您的应用程序对响应时间不是那么敏感，那么很高兴地告诉您，您可以将其推向边缘甚至更远，并且 Caché 仍然可以安全地工作。

> 请记住，您不会想要以 100% CPU 运行数据库虚拟机或主机。
> 您需要容量来应对虚拟机的意外峰值和增长，而 ESXi 虚拟机监控程序需要资源来进行所有网络、存储和其他活动。

我总是针对 80% CPU 利用率的峰值进行规划。 即便如此，vCPU 的规模最多也只调整到物理核心数，这样即使在极端情况下，仍然有余量让 ESXi 虚拟机监控程序处理逻辑线程。

> 如果您运行超融合 (HCI) 解决方案，还必须考虑主机级别的 HCI CPU 要求。
> 有关详细信息，请参见我先前关于 HCI 的帖子。 部署在 HCI 上的虚拟机的基本 CPU
> 规模调整与其他虚拟机相同。

请记住，您必须在您自己的环境中使用您的应用程序验证和测试所有内容。

示例 2. 资源过载

我看到过客户站点报告应用程序性能“ 慢 ”，而客户机操作系统却报告有空闲的 CPU 资源。

记住，客户机操作系统并不知道它是虚拟化的。 不幸的是，客户机内指标（例如 vmstat 在 pButtons
中报告的指标）可能具有欺骗性，您还必须获得主机级指标和 ESXi 指标（例如
esxtop）才能真正了解系统运行状况和容量。

如上面的图表所示，当主机报告 100% 利用率时，客户机虚拟机可能报告较低的利用率。 36 vCPU
虚拟机（红色）在 4 倍速率时报告 80% 平均 CPU 利用率，而主机报告 100%。
即使规模调整合理的虚拟机也可能出现资源短缺的情况，例如，如果在启动后有其他虚拟机迁移到主机上，或者由于 DRS 规则配置不当而导致资源过载。

为了显示关键指标，在下面的一系列测试中，我进行了以下配置：

- 主机上运行两个数据库虚拟机。
    - 一个 24 vCPU 虚拟机以恒定的 2 倍事务速率运行（图表上未显示）。
    - 一个 24 vCPU 虚拟机以 1 倍、2 倍、3 倍事务速率运行（图表上显示这些指标）。

在另一个数据库使用资源的情况下；在 3 倍速率时，客户机操作系统 (RHEL 7) vmstat 只报告 86% 平均 CPU
利用率，运行队列大小平均只有 25。
然而，该系统的用户将大声抱怨，因为组件响应时间随着进程变慢而迅速增加。

如下图所示，协同停止和就绪时间说明了为什么用户性能如此糟糕。 就绪时间 (%RDY) 和协同停止 (%CoStop)
指标显示 CPU 资源在目标 3 倍速率下大幅过载。 这实际并不奇怪，因为*主机*以 2
倍速率运行（其他虚拟机），*而*该数据库虚拟机以 3 倍速率运行。

该图表明，当主机上的 总 CPU 负载增加时，就绪时间也会增加。

> 就绪时间是指虚拟机已准备好运行，但由于 CPU 资源不可用而无法运行的时间。

协同停止也会增加。 没有足够的空闲逻辑 CPU
来允许数据库虚拟机运行（正如我在上面的超线程部分详细说明的那样）。 最终结果是由于对物理 CPU
资源的争用而导致处理延迟。

我曾在一个客户站点看到过这种情况，当时通过 pButtons 和 vmstat 获取的支持视图只显示了虚拟化的操作系统。
虽然 vmstat 报告还有 CPU 余量，但用户的性能体验非常糟糕。

这里的教训是，直到 ESXi 指标和主机级视图可用，才能诊断出真正的问题；一般的集群 CPU 资源短缺导致的 CPU
资源过载，以及使情况变得更糟的不良 DRS 规则，会使高事务数据库虚拟机一起迁移并使主机资源不堪重负。

示例 3. 资源过载

在此示例中，我使用了一个以 3 倍事务速率运行的基准 24 vCPU 数据库虚拟机，然后使用两个以恒定 3
倍事务速率运行的 24 vCPU 数据库虚拟机。

虚拟机的平均基准 CPU 利用率（见上面的示例 1）为 76%，主机则为 85%。 单个 24 vCPU
数据库虚拟机会使用全部 24 个物理处理器。 运行两个 24 vCPU
虚拟机意味着这两个虚拟机将争用资源，并使用服务器上的全部 48 个逻辑执行线程。

请记住，在运行单个虚拟机时，主机并没有被 100% 利用，我们仍然可以看到，当两个非常繁忙的 24 vCPU
虚拟机试图使用主机上的 24 个物理核心（即使开启了超线程）时，吞吐量和性能显著下降。 尽管 Caché
非常有效地使用了可用的 CPU 资源，但每个虚拟机的数据库吞吐量仍然下降了
16%，更重要的是，组件（用户）响应时间增加了 50% 以上。

总结

本帖的目的是回答几个常见问题。 要深入了解 CPU 主机资源和 VMware CPU 调度器，请参见下面的参考部分。

虽然有许多专业级的调整，并且要深入研究 ESXi 才能榨干系统的最后一点性能，但基本规则非常简单。

对于*大型生产数据库*：

- 为每个物理 CPU 核心规划一个 vCPU。
- 考虑 NUMA 并按理想情况调整虚拟机规模，以使 CPU 和内存对于 NUMA 节点是本地的。
- 合理调整虚拟机规模。 仅在需要时才添加 vCPU。

如果您想要整合虚拟机，请记住，大型数据库非常繁忙，在高峰期会大量使用 CPU（物理和逻辑）。
在您的监视系统告诉您安全之前，不要超额预定 CPU。

参考

- VMware 博客 - 怪兽虚拟机何时过载 vCPU:pCPU
- 2016 NUMA 深入研究系列介绍
- VMware vSphere 5.1 中的 CPU 调度器

测试

我在一个 vSphere 集群上运行了本帖中的示例，该集群包括连接到一个全闪存阵列的双处理器 Dell R730。
在示例运行期间，网络或存储没有出现瓶颈。

- Caché 2016.2.1.803.0

PowerEdge R730

- 2 个 Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz
- 16 条 16GB RDIMM，2133 MT/s，双列，x4 数据宽度
- SAS 12Gbps HBA 外部控制器
- 超线程 (HT) 开启

PowerVault MD3420，12G SAS，2U-24 驱动器

- 24 个 960GB 固态硬盘 SAS 读取密集型 MLC 12Gbps 2.5 英寸热拔插驱动器，PX04SR
- 2 个控制器，12G SAS，2U MD34xx，8G 缓存

VMware ESXi 6.0.0 build-2494585

- 按照最佳实践配置虚拟机；VMXNET3、PVSCSI 等

RHEL 7

- 大页面

基准 1 倍速率下平均每秒 700,000 gloref（每秒数据库访问次数）。 24 vCPU 在 5 倍速率下平均每秒超过 3,000,000 gloref。 测试以老化方式进行，直到达到稳定的性能，然后进行 15 分钟采样并取平均值。

> 这些示例只是为了说明理论，您必须使用自己的应用程序进行验证！

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

源 URL: https://cn.community.intersystems.com/post/%E7%B2%BE%E5%8D%8E%E6%96%87%E7%AB%A0-%E8%99%9A%E6%8B%9F%E5%8C%96%E5%A4%A7%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93-vmware-cpu-%E5%AE%B9%E9%87%8F%E8%A7%84%E5%88%92