文章
Michael Lei · 七月 4, 2021 阅读大约需 9 分钟

数据平台和性能 - 第 7 部分
用于确保性能、可伸缩性和可用性的企业缓存协议ECP

(ECP) Caché 出色的可用性和扩展特性之一是企业缓存协议 (ECP)。 在应用程序开发过程中，如对使用 ECP 的分布式处理加以考虑，可以横向扩展 Caché 应用程序的架构。
应用程序处理可以调整为非常高的速率，处理能力从单个应用程序服务器扩展到最多 255 个应用程序服务器，并且不需要任何应用程序更改。

在我参与的 TrakCare 部署中，ECP 已广泛使用多年。 十年前，主要供应商之一的一台 “ 大型 ” x86 服务器可能总共只有八个核心。 对于大型部署来说，ECP 是横向扩展商业服务器处理能力的方式，不适合单台昂贵的大型企业服务器。
即使是高核心数的企业服务器也有限制，因此 ECP 也用于扩展这些服务器上的部署。

如今，大多数的新 TrakCare 部署或升级到当前硬件*不需要 ECP* 即可扩展。 目前的双插槽 x86 生产服务器可以拥有数十个核心和巨大容量的内存。 我们看到，在最近的 Caché 版本中，TrakCare 以及许多其他 Caché 应用程序具有可预测的线性扩展能力，能够随着单台服务器中 CPU 核心数量和内存的增加而支持逐渐增多的用户和事务。
在现场，我看到大多数的新部署都是虚拟化的，即使如此，虚拟机也可以根据需要扩展到主机服务器的规模。
如果资源需求超过单个物理主机可以提供的资源，则使用 ECP 进行横向扩展。

- ***提示：****为了简化管理和部署规模，在部署 ECP 之前，先在单台服务器内扩展。*

在本帖中，我将展示一个示例架构以及 ECP 工作原理的基础知识，然后评论性能注意事项，重点是存储。

有关配置 ECP 和应用程序开发的具体信息，请参见在线的 Caché 分布式数据管理指南，并且社区上有一个 ECP 学习轨迹。

ECP 的其他关键特性之一是提高了应用程序可用性，有关详细信息，请参见 Caché 高可用性指南中的 ECP 部分。

本系列其他帖子的列表

ECP 架构基础知识

ECP 的架构和运行在概念上很简单，ECP 提供了在多个服务器系统之间有效共享数据、锁定和可执行代码的方法。
从应用程序服务器角度看，数据和代码远程存储在*数据服务器*上，但缓存在*应用程序服务器*的本地内存中，以提供对活动数据的有效访问，同时尽可能减少网络流量。

数据服务器管理对磁盘上持久性存储的数据库读写，而多个应用程序服务器是解决方案的主力，执行大多数应用程序处理。

多层架构

ECP 采用多层架构。 描述处理层和它们扮演的角色有多种不同的方式，以下是我在描述基于 Web 浏览器的 Caché 应用程序时发现很有用的方式，也是我的帖子的模型和术语。
我知道可能有不同的方法来细分层级，但现在先使用我的方法 :)

基于浏览器的应用程序（例如 Caché Server Pages

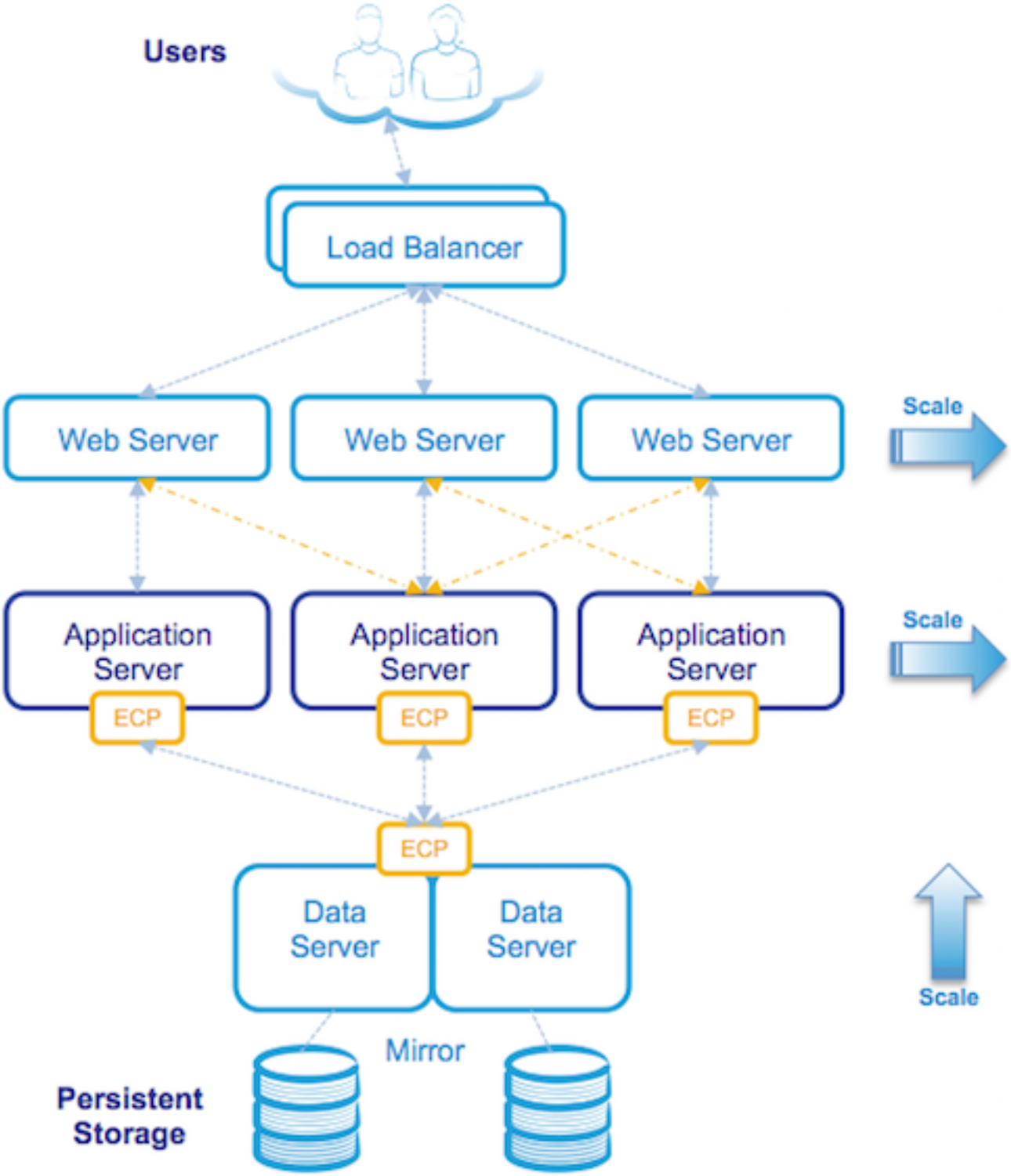
(CSP)）使用多层架构，其中表示、应用程序处理和数据管理功能在逻辑上是分开的。
具有不同角色的**逻辑**“ 服务器 ”填充各层。 逻辑服务器不必保留在单独的物理主机或虚拟服务器上，出于成本效益和可管理性的考虑，部分甚至全部逻辑服务器可能位于单个主机或操作系统实例上。
随着部署规模的扩展，服务器可以通过 ECP
划分到多个物理或虚拟主机上，从而可根据需要分散处理工作负载，而无需更改应用程序。

主机系统可以是物理的或虚拟化的，具体取决于容量和可用性要求。 以下层和逻辑服务器构成了一个部署：

- *表示层：*包括在基于浏览器的客户端和应用程序层之间充当网关的 Web 服务器。
- *应用程序层：*这是 ECP 应用程序服务器所在的位置。 如上文所述，这是一个逻辑模型，其中应用程序服务器不必与数据服务器分开，而且除了最大型的站点外，所有情况下通常都不需要分开。
  该层还可能包括进行专门处理的其他服务器，如报告服务器。
- *数据层：*这是数据服务器所在的位置。 数据服务器执行事务处理，是存储在 Caché
  数据库中的应用程序代码和数据存储库。 数据服务器负责读写持久性磁盘存储。

逻辑架构

下图是一个基于浏览器的应用程序在部署为三层架构时的逻辑视图：



尽管初看之下该架构可能很复杂，但构成它的组件仍然与安装在单台服务器上的 Caché 系统的组件相同，只是逻辑组件安装在多个物理或虚拟服务器上。 服务器之间的所有通信都通过 TCP/IP 进行。

逻辑视图中的 ECP 操作

上图从顶部开始，显示用户安全地连接到多个已进行负载平衡的 Web 服务器。 这些 Web 服务器在客户端和应用程序层（应用程序服务器）之间传递 CSP 网页请求，应用程序层进行所有处理，允许动态创建内容，并通过 Web 服务器将完成的页面返回给客户端。

在这个三层模型中，应用程序处理通过 ECP 分散到多个应用程序服务器上。

应用程序只将数据（您的应用程序数据库）视为应用程序服务器的本地数据。

当应用程序服务器发出数据请求时，它将尝试从本地缓存满足请求，如果不能满足，ECP
将向数据服务器请求必要的数据，数据服务器自己的缓存可能会满足请求，否则将从磁盘获取数据。
数据服务器对应用程序服务器的回复包括存储该数据的数据库块。
这些块将被使用，并且此时将缓存到应用程序服务器上。 ECP
自动负责管理整个网络中的缓存一致性，并将变化传播回数据服务器。
客户端会体验到快速响应，因为它们经常使用本地缓存的数据。

默认情况下，Web 服务器与首选的应用程序服务器通信，确保同一应用程序服务器满足相关数据的后续请求，因为这些数据可能已经在本地缓存中。

- ***提示：*** *如 Caché 文档*
  中详述，在循环或负载平衡方案中，应避免用户连接到应用程序服务器，因为这会影响应用程序服务器上缓存的优势。 理想情况下，相同的用户或用户组保持连接到同一应用程序服务器。

该解决方案通过在表示层添加 Web 服务器和在应用程序层添加其他应用程序服务器来进行扩展，无需用户停机。
数据层通过增加数据服务器的 CPU 和内存来进行扩展。

物理架构

下图显示了与三层逻辑架构示例相同的三层部署中使用的物理主机示例：

请注意，在每一层部署物理或虚拟主机时均采用 n+1 或 n+2 模式，以确保在主机故障或计划维护时保持 100%
能力。 由于用户分布在多个 Web
和应用程序服务器上，单个服务器故障只会影响少量用户，他们会自动重新连接到其余服务器之一。

数据管理层具有高度可用性，例如，位于连接到一个或多个存储阵列的故障转移集群上（例如，虚拟化
HA、InterSystems 数据库镜像或传统的故障转移集群）。
如果硬件或服务出现故障，集群将在其中一个幸存节点上重启服务。 ECP 的一个附加好处是内置弹性，并且在数据库节点集群发生故障转移时能保持事务完整性，应用程序用户将观察到处理暂停，直到故障转移和自动恢复完成，随后用户将无缝继续，不会断开连接。

同样的架构也可以映射到虚拟化服务器，例如，VMware vSphere 可用于虚拟化应用程序服务器。

ECP 容量规划

如上文所述，数据服务器管理对持久性磁盘的数据库读写，而多个应用程序服务器是解决方案的主力，执行大多数应用程序处理。 这是考虑系统资源容量规划时的一个关键概念，总结来说：

- ***数据服务器***（有时称为数据库服务器）通常执行很少的应用程序处理，因此*对 CPU 要求低*，但该服务器执行大部分存储 IO，因此可能有*非常高的存储 IOPS*，即数据库读写以及日志写入（稍后将详细介绍日志 IO）。
- ***应用程序服务器***执行大多数应用程序处理，因此*对 CPU 要求高*，但存储 IO 非常少。

通常，调整 ECP 服务器 CPU、内存和 IO 要求的规则与调整非常大的单服务器解决方案的规则相同，同时考虑 N+1 或 N+2 台服务器以确保高可用性。

基本 CPU 和存储规模调整：

假设 MyApplication 需要最多 72 个 CPU
核心进行应用程序处理（记得还要考虑余量），并且预计在写入守护进程周期期间需要 20,000 次写入，以及 10,000 次随机数据库读取的持续峰值。

一个简单的虚拟或物理服务器规模调整方案为：

- 4 台 32 CPU 应用程序服务器（3 台服务器 + 1 台服务器用于确保 HA）。 低 IOPS 要求。
- 2 台 10 CPU 数据服务器（镜像或集群以确保 HA）。 低延迟 IOPS 要求为 20K 写入、10K 读取，加上 WIJ 和日志。

虽然数据服务器只执行非常少的处理，但考虑到系统和 Caché 进程，将其规模调整为 8-10 个 CPU。
应用程序服务器的规模可以根据每台物理主机的最佳性价比和/或可用性来进行调整。
横向扩展时会有一些效率损失，但通常可以在服务器块中增加处理能力，并预计吞吐量有近乎线性的增长。
限制更有可能首先在存储 IO 中出现。

- *_ 提示：与确保 HA 一样，要考虑主机、机箱或机架故障的影响。 在 VMWare 上虚拟化应用程序和数据服务器时，确保应用 vSphere DRS 和相关性规则以分散处理负载并确保可用性。*

日志同步 IO 要求

ECP 部署的另一个容量规划注意事项是，由于日志同步，它们需要较高
IO，并且对存储响应时间的要求非常严格，以保持数据服务器上日志记录的可伸缩性 。
同步请求可以触发对日志中最后一个块的写入，以确保数据耐久性。

不过您的情况可能有所不同；在一个典型的以高事务处理速率运行的客户站点上，我经常看到非 ECP
配置上的日志写入 IOPS 为每秒十几次。 在繁忙的系统上使用 ECP 时，由于 ECP
强制日志同步，可以在日志磁盘上看到 100 到 1000 的写入 IOPS。

- *_ 提示：如果在繁忙的系统上显示 mgstat 或查看 pButtons 中的 mgstat，您将看到 Jrnwrts（日志写入次数），您将在存储 IO 资源规划中对其加以考虑。 在 ECP 数据服务器上，还有未显示在 mgstat 中的对日志磁盘的日志同步写入，要了解这些信息，您需要查看日志磁盘的操作系统指标，例如使用 iostat 查看。*

什么是日志同步？

需要日志同步的原因：

- 确保在数据服务器发生故障时数据的耐久性和可恢复性。
- 它们也是确保应用程序服务器之间的缓存一致性的触发器。

在非 ECP 配置中，对 Caché 数据库的修改将写入日志缓冲区（128 x 64K
缓冲区），当日志缓冲区满时或每两秒由日志守护程序写入磁盘上的日志文件。 Caché 为整个缓冲区分配
64k，并且这些缓冲区总是被重复使用，而不是被销毁和重新创建，Caché 只是跟踪末尾偏移量。
在大多数情况下（除非一次进行大量更新），日志写入次数非常小。

ECP 系统中也有日志同步。
日志同步可以定义为将当前日志缓冲区的相关部分重新写入磁盘，以确保磁盘上的日志始终是最新的。
因此，日志同步会请求多次重新写入同一日志块的某个部分（大小在 2k 到 64k 之间）。

ECP 客户端上可以触发日志同步请求的事件为更新（SET 或 KILL）或 LOCK。 例如，对于每个 SET 或
KILL，都会将当前日志缓冲区写入（或重新写入）磁盘。
在非常繁忙的系统中，单次同步操作中的日志同步可能被捆绑或延迟为多个同步请求。

日志同步的容量规划

为确保持续的吞吐量，日志同步的平均写入响应时间必须：

- *<=0.5 毫秒，最长 <=1 毫秒。*

有关详细信息，请参见以下帖子中的 IO 要求表格：第 6 部分 - Caché 存储 IO 配置文件。

- 提示：当通过 ECP 使用 Caché 数据库镜像时，主要和备份镜像节点日志磁盘上都应用日志同步。
  这应该不是问题，因为镜像配置规则是将两个节点配置成对于存储 IO 是相等的。

对于您自己的系统，您必须验证具体的 IO
指标，本部分的目的是让您知道有非常严格的响应时间要求，并了解在何处查找指标。

- _ *提示：在* Red Hat Linux 和 SuSE 上，建议对 Caché 系统使用 XFS 文件系统。
  但是测试表明，对于较少的写入（如日志同步），ext4 可提供更高的 IOPS 能力。 对于日志磁盘考虑使用 ext4。

总结

本帖介绍了 ECP 以及在容量规划过程中要考虑的其他指标。 我希望在不久的将来，我们可以分享 Caché 和 ECP 在一些非常大型的系统上的最新基准测试结果。
和往常一样，如果您有任何问题或需要添加的内容，请通过评论告诉我。 我的推特 @murrayoldfield

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

---

源 URL:
https://cn.community.intersystems.com/post/%E6%95%B0%E6%8D%AE%E5%B9%B3%E5%8F%B0%E5%92%8C%E6%80%A7%E8%83%BD-%E7%AC%AC-7-%E9%83%A8%E5%88%86-%E7%94%A8%E4%BA%8E%E7%A1%AE%E4%BF%9D%E6%80%A7%E8%83%BD%E3%80%81%E5%8F%AF%E4%BC%B8%E7%BC%A9%E6%80%A7%E5%92%8C%E5%8F%AF%E7%94%A8%E6%80%A7%E7%9A%84%E4%BC%81%E4%B8%9A%E7%BC%93%E5%AD%98%E5%8D%8F%E8%AE%AEecp